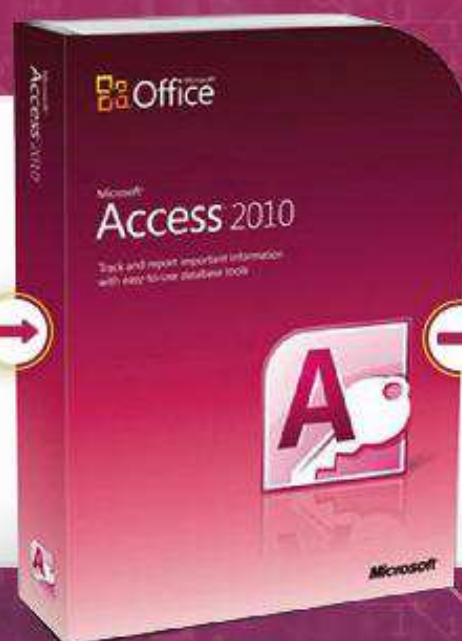


О. В. Шпортько, Л. В. Шпортько

# РОЗРОБКА БАЗ ДАНИХ В СУБД MICROSOFT ACCESS 2010/2013

**ПРАКТИКУМ**  
для студентів вищих та  
учнів професійно-технічних  
навчальних закладів



*Людина в XXI столітті,  
яка не буде вміти користуватися ЕОМ,  
буде подібна людині XX століття,  
що не вміла ні читати, ні писати*

**В. М. Глушков**



О. В. Шпортько, Л. В. Шпортько

# Розробка баз даних в СУБД Microsoft Access 2010/2013

Практикум  
для студентів вищих та  
учнів професійно-технічних  
навчальних закладів

з питань придбання  
друкованих примірників  
за оптовими цінами  
чи електронного варіанту  
цього посібника без реклами  
звертайтеся на e-mail  
books\_shportko@ukr.net

Рівне – 2017

УДК 681.3.016  
Ш 84

Схвалено для використання у професійно-технічних навчальних закладах  
(Лист Державної наукової установи "Інститут модернізації змісту освіти"  
МОН України від 15.09.2017 № 21.1/12-Г-665)

Рекомендовано до видання Вченою радою  
Рівненського державного гуманітарного університету  
(протокол № 11 від 28.09.2017 р.)

Автори: О. В. Шпортько, доцент кафедри економічної кібернетики РДГУ, кандидат технічних наук, доцент  
Л. В. Шпортько, викладач ДВНЗ "Рівненський коледж економіки та бізнесу".

Рецензенти: А. Я. Бомба, завідувач кафедри інформатики та прикладної математики РДГУ, доктор технічних наук, професор;  
А. П. Власюк, завідувач кафедри інформаційних систем та обчислювальних методів МЕГУ, доктор технічних наук, професор;  
Й. В. Джунь, завідувач кафедри математичного моделювання МЕГУ, дійсний член Міжнародної педагогічної академії, доктор фізико-математичних наук, професор;  
П. С. Янчук, декан факультету кібернетики МЕГУ, кандидат фізико-математичних наук, професор.

**Шпортько О. В. Розробка баз даних в СУБД Microsoft Access 2010/2013 :**  
Практикум для студентів вищих та учнів професійно-технічних навчальних закладів / О. В. Шпортько, Л. В. Шпортько. – Рівне: О. Зень, 2017. – 184 с.

**ISBN 978-617-601-166-8**

Практикум містить впорядкований набір інструкційних карт з завданнями і вказівками до виконання лабораторних робіт та самостійного опрацювання матеріалу кожної теми. Основна увага приділяється вивченню засобів СУБД MS Access для розробки цілісних інформаційно-пошукових систем на прикладі ІС обробки БД гуртового складу. Призначений для учнів, студентів, викладачів та всіх, хто прагне оволодіти можливостями та засобами СУБД MS Access для ефективного створення ІС обробки систематизованих даних.

УДК 681.3.016

ISBN 978-617-601-166-8

© О. В. Шпортько, Л. В. Шпортько, 2017

## Зміст

<b>ПЕРЕДМОВА.....</b>	<b>4</b>
1. ПРОЕКТУВАННЯ ВЛАСНОЇ БД НА ЗОВНІШНЬОМУ, ІНФОЛОГІЧНОМУ ТА ДАТАЛОГІЧНОМУ РІВНЯХ. ....	6
2. СТВОРЕННЯ БД В СЕРЕДОВИЩІ СУБД MS ACCESS. ОСНОВНІ СЕРВІСНІ ФУНКЦІЇ ОБРОБКИ БД. ....	18
3. СТВОРЕННЯ ТАБЛИЦЬ ЗАСОБАМИ СУБД MS ACCESS.....	24
4. РЕДАГУВАННЯ ВЛАСТИВОСТЕЙ ПОЛІВ ТАБЛИЦЬ В РЕЖИМІ КОНСТРУКТОРА. ВСТАНОВЛЕННЯ СТРУКТУРНИХ ЗВ'ЯЗКІВ МІЖ ТАБЛИЦЯМИ МАЙСТРОМ ПІДСТАНОВОК ТА В СХЕМІ ДАНИХ. ....	28
5. РЕДАГУВАННЯ, СОРТУВАННЯ, ІНДЕКСУВАННЯ, ПОШУК, ФІЛЬТРУВАННЯ ТА КОПИЮВАННЯ ДАНИХ ТАБЛИЦЬ. ЕКСПОРТ ТА ІМПОРТ ТАБЛИЦЬ. ВСТАНОВЛЕННЯ ЗВ'ЯЗКІВ З ТАБЛИЦЯМИ ІНШИХ БД. ....	45
6. СТВОРЕННЯ ЗВИЧАЙНИХ, ПІДПОРЯДКОВАНИХ ТА ПОВ'ЯЗАНИХ ФОРМ ЗА ДОПОМОГОЮ МАЙСТРА. ВИКОРИСТАННЯ ФОРМ ДЛЯ ВВЕДЕННЯ ТА АНАЛІЗУ ДАНИХ. ....	54
7. СТВОРЕННЯ ТА КОРЕГУВАННЯ ВЛАСТИВОСТЕЙ ФОРМ І ЇХ ЕЛЕМЕНТІВ КЕРУВАННЯ В РЕЖИМІ КОНСТРУКТОРА. РОЗРОБКА ЗАСОБІВ ПОШУКУ У ФОРМАХ.....	63
8. СТВОРЕННЯ ВЛАСНОЇ ТА СТАНДАРТНОЇ ГОЛОВНИХ КНОПКОВИХ ФОРМ. КОРЕГУВАННЯ ВЛАСТИВОСТЕЙ СТВОРЕНИХ ФОРМ ТА ЕЛЕМЕНТІВ КЕРУВАННЯ В РЕЖИМІ КОНСТРУКТОРА. СТВОРЕННЯ ОБЧИСЛЮВАЛЬНИХ ПОЛІВ У ФОРМАХ.....	80
9. РОЗРОБКА ТА ВИКОРИСТАННЯ ЗАПИТІВ НА ВИБІРКУ БЕЗ ГРУПУВАННЯ ЗАПИСІВ. ....	90
10. СТВОРЕННЯ І ВИКОРИСТАННЯ ЗАПИТІВ НА ВИБІРКУ З ГРУПУВАННЯМ ЗАПИСІВ ТА ПЕРЕХРЕСНИХ ЗАПИТІВ. ВИКОРИСТАННЯ ПАРАМЕТРІВ ПОЄДНАНЬ ТАБЛИЦЬ ТА ВЛАСТИВОСТЕЙ ЗАПИТІВ ДЛЯ ЕФЕКТИВНОГО АНАЛІЗУ ДАНИХ. ....	106
11. СТВОРЕННЯ ЗАПИТІВ НА ВИБІРКУ ТА НА ПОЄДНАННЯ В РЕЖИМІ SQL. ....	123
12. РОЗРОБКА ТА ВИКОРИСТАННЯ ЗАПИТІВ НА КОРЕГУВАННЯ ДАНИХ. ....	129
13. СТВОРЕННЯ ЗВІТІВ ЗА ДОПОМОГОЮ МАЙСТРА. ....	141
14. РОЗРОБКА ТА КОРЕГУВАННЯ ВЛАСТИВОСТЕЙ ЗВІТІВ І ЇХ ЕЛЕМЕНТІВ КЕРУВАННЯ В РЕЖИМІ КОНСТРУКТОРА. ФОРМУВАННЯ БАГАТОСТОВПЦЕВИХ ЗВІТІВ. ....	153
15. КОМПЛЕКСНЕ ВИКОРИСТАННЯ ФОРМ, ЗАПИТІВ ТА ЗВІТІВ ДЛЯ АВТОМАТИЗАЦІЇ ОБРОБКИ ДАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ.....	170
16. ПУБЛІКАЦІЯ СТАТИЧНОЇ ІНФОРМАЦІЇ З БД В ЛОКАЛЬНИХ ТА ГЛОБАЛЬНИХ МЕРЕЖАХ.....	179
<b>ПРОГРАМНІ ПИТАННЯ ПОТОЧНОГО ТА ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ .....</b>	<b>182</b>
<b>РЕКОМЕНДОВАНА ЛІТЕРАТУРА.....</b>	<b>183</b>

## Передмова

На сьогоднішній день важко уявити собі комп'ютерну техніку, локальні та глобальні мережі без баз даних (надалі БД) та невід'ємної частини прикладного програмного забезпечення – систем управління ними (надалі СУБД). З БД ми стикаємося всюди: у банку та відділі бухгалтерії, в пенсійному фонді та податковій інспекції, у паспортному столі та на залізничному вокзалі, в ЖЕКу та Укртелекомі. Вже стало звичним для посвідчення особи поряд з паспортом подавати ідентифікаційний номер, під яким зберігаються особисті дані в БД Державного реєстру фізичних осіб. Все частіше використовуються БД для автоматизації діяльності лікарень, бібліотек, автозаправних станцій, туристичних агенцій, підвищення якості підготовки спеціалістів у закладах освіти. Навіть у глобальній мережі Інтернет використовуються БД для пошуку інформації та зберігання вмісту більшості Web-сторінок, даних про відвідувачів сайтів та про власників поштових скриньок. Висловлюючись образно, світом зараз правлять БД.

Поширюватися БД та інформаційні системи (надалі ІС) їх обробки будуть і в майбутньому, закріплюючись у сферах мобільного зв'язку, телерадіокомунікацій та проникаючи у наш побут. Як в наш час комп'ютери витіснили друкарські машинки так, можливо, в недалекій перспективі комп'ютерні БД витіснять блокноти та записнички.

Метою зберігання даних на електронних носіях є швидка та якісна їх обробка. І саме реляційні СУБД дозволяють зберігати інформацію без дублювання, надлишковості та неузгодженості, забезпечують оперативний доступ до даних, створення та експлуатацію засобів для їх швидкого пошуку, фільтрування та сортування, незалежність структур даних від прикладних програм, коректне використання інформації в мережі. Ось чому більшість ІС обробки даних в наш час створюються на основі однієї з відомих сучасних СУБД. Отже, поширення БД, СУБД, ІС обробки БД – це об'єктивний процес сучасності, викликаний науково-технічним прогресом.

В цих умовах кожен випускник ВНЗ, що має використовувати комп'ютерну техніку в своїй професійній діяльності, повинен не лише розуміти внутрішню структуру реляційних БД, а й вміти проектувати їх, володіти засобами однієї з сучасних СУБД для оперативної обробки даних, а в разі потреби – і для розробки якісних цілісних інформаційно-пошукових систем. Адже тільки вірно спроектована та вдало реалізована система здатна економити час та підвищувати ефективність роботи спеціаліста. Етапи створення моделі однієї з таких систем детально розглядаються в цьому практикумі.

Основними завданнями практикуму є змістовне і глибоке вивчення студентами і учнями методології створення та ведення БД,



отримання навичок використання СУБД для швидкої та якісної автоматичної обробки систематизованих даних і створення цілісних інформаційно-пошукових систем.

Доцільність використання практикуму визначається здобутими знаннями студентів і учнів стосовно:

- створення та вдосконалення структури БД;
- забезпечення цілісності атрибутів, таблиць, реляційних та зв'язкових відношень БД;
- розробки засобів введення даних, їх обробки та публікації результатів на паперових і електронних носіях.

Даний практикум містить впорядкований набір інструкційних карт з завданнями до виконання лабораторних робіт та самостійного опрацювання матеріалу по створенню ІС обробки БД. Завдання кожної лабораторної роботи дозволяють вдосконалити інформаційну систему, розроблену протягом попередніх робіт. Під час лабораторних занять студенти і учні розробляють типову інформаційну систему гуртового складу, що допомагає їм порівнювати та колективно обговорювати отримані проміжні та загальні результати кожної роботи, а при самостійному опрацюванні матеріалу дисципліни кожен зі студентів і учнів розробляє власну систему згідно тематики першої лабораторної роботи. Це, з одного боку, підвищує рівень самоконтролю студента і учня, адже недоліки, допущені при виконанні роботи, можуть датися взнаки при виконанні наступних робіт, а з іншого – мотивує вивчення матеріалу для здобуття та закріплення навичок використання засобів СУБД MS Access.

В тексті практикуму різними шрифтами позначаються:

- **напівжирним шрифтом** – означувані поняття, важливі вислови, назви розділів та тем у викладеному матеріалі;
- *курсивним шрифтом* – пункти меню, підменю, заголовки вікон і кнопок, значення полів СУБД MS Access та назви об'єктів.

На початку кожної лабораторної роботи наведено короткі теоретичні відомості для розуміння суті її завдань, а в кінці – контрольні запитання для систематизації та закріплення отриманих знань, умінь і навичок.

Авторство практикуму не поширюється на теоретичні відомості до лабораторних робіт, в яких співавтори є укладачами. Особливу подяку автори практикуму висловлюють проф. Ситник (Єр'юміній) Н. В. за працю [4], з якої процитовано чи запозичено суть основних понять цих теоретичних відомостей. Саме цю працю ми настійливо рекомендуємо читачам для поглибленого вивчення теорії проектування БД.

## Лабораторна робота № 1

**Тема.** Проектування власної БД на зовнішньому, інфологічному та даталогічному рівнях.

**Мета.** Систематизація та застосування знань з теорії проектування БД. Усвідомлення ролі кожного рівня проектування БД. Формування вмінь та навичок проектування власної БД на різних рівнях.

### Теоретичні відомості

**База даних (БД)** – це поіменована структурована сукупність відомостей про об'єкти певної предметної області та окремих стандартизованих засобів для автоматизації їх обробки. БД розробляють для прискорення задоволення інформаційних потреб користувача (скажімо, визначення вільних місць на конкретний сеанс в кінотеатрі чи на обраний рейс потяга Укрзалізниці). Концепція застосування БД базується на однократному введенні даних і багатократному їх використанні. Існують не лише комп'ютерні БД (наприклад, прайс-листи чи бібліотечні каталоги), але саме електронні БД дають змогу максимально швидко перетворювати вхідну інформацію у вихідну.

Об'єкти предметної області характеризуються своїми властивостями (наприклад, для кожного співробітника організації можна вказати прізвище, ім'я, по батькові, ідентифікаційний номер, дані паспорта, домашню адресу). У свою чергу, властивості об'єкта складаються з неподільних атрибутів (наприклад, з властивості *Дані паспорта* доцільно виділити атрибути *Серія паспорта, Номер паспорта, Дата видачі паспорта, Ким видано паспорт*). В БД зберігаються значення лише тих атрибутів, які дають змогу відрізнити один об'єкт від іншого. Це зменшує розмір БД і, як наслідок, прискорює їх обробку.

На практиці найчастіше використовують реляційні (табличні) БД. В них дані однотипних об'єктів групують по окремих таблицях (сутностях), а логічно пов'язані атрибути різних таблиць поєднують зв'язками. Кожен рядок (запис, кортеж) реляційної таблиці описує один однотипний об'єкт, а кожен стовбець (поле) містить значення одного атрибута для всіх таких об'єктів. Приклад фрагменту реляційної таблиці наведено на рис. 1, а перелік її атрибутів міститься у заголовках полів.

Чітка структуризація даних реляційної таблиці з одного боку прискорює їх обробку, а з іншого – дає змогу отримати однозначні результати опрацювання. Наприклад, при пошуку співробітників, які проживають на вулиці *Кавказька*, співставлення значень буде здійснюватися лише для поля *Вулиця проживання* кожного запису (а не для всіх полів) і тому не буде віднайдено співробітниць з прізвищем *Кавказька*.

КодТовару	Назва товару	Марка товару	Одиниця вимірювання
1	згущене молоко 'Бахмач'	Бахмач	шт.
2	майонез 'Делікатесний'	Торчин продукт	шт.
3	цукерки 'Червоний мак'	Світоч	кг
4	цукерки 'Столичні'	ЕКО продукт	кг
5	цукерки 'Зоряне сйво'	ЕКО продукт	кг
6	кава 'Галка'	Львівська кава	шт.

Рис. 1. Фрагмент реляційної таблиці

Для організації взаємодії з електронними БД використовують системи управління базами даних (СУБД), наприклад, такі, як Oracle, MS Access чи SQL Server. **СУБД** – це комплекс програмних і мовних засобів загального і спеціального призначення необхідний для створення БД, підтримування її в актуальному стані, організації маніпулювання даними та доступу до них різних користувачів. Використання СУБД дає змогу розробнику зосередитися на створенні прикладних програм і обробляти дані реляційних таблиць на рівні окремих полів, не замислюючись над тим, де і як ці дані фізично зберігаються у файлах БД. Структура даних реляційних таблиць разом з даними зберігається в БД і автоматично використовується СУБД, що дозволяє вносити в неї зміни без внесення корегувань в прикладні програми, а новим програмам – використовувати накопичені раніше дані (рис. 2). Структура даних входить у **словник даних**, де також можуть міститися відомості про предметну область, користувачів, їх права доступу та журнал внесення змін. Фактично, опрацювання записів БД виконується СУБД в два проходи: на першому – зчитується структура даних, а на другому – відбувається їх обробка згідно зчитаної структури.

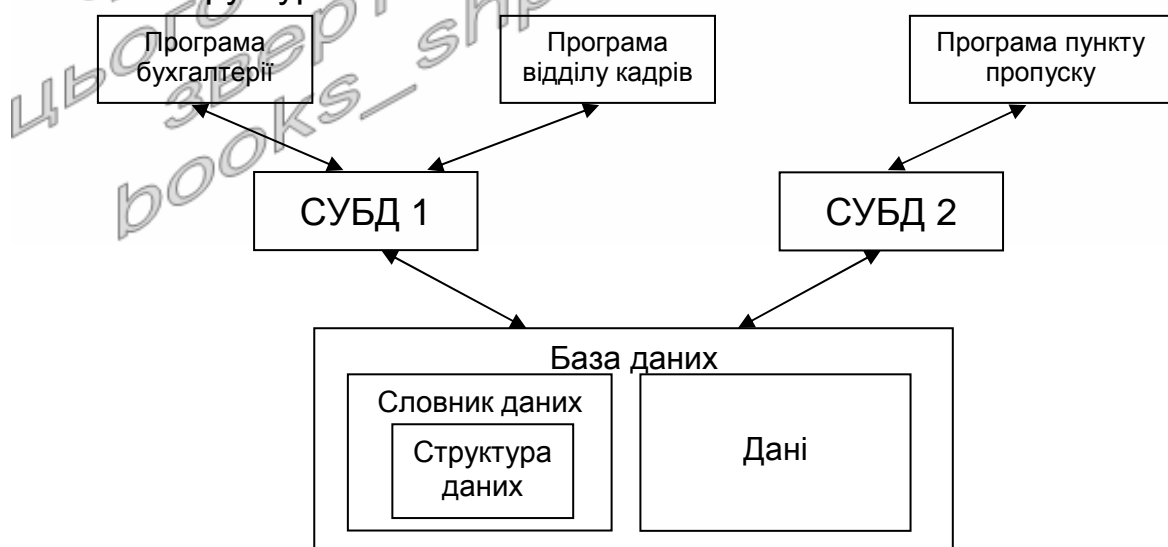


Рис. 2. Організація обміну даними при незалежній структурі БД від прикладних програм



Ми будемо розробляти електронні БД та прикладні програми для їх обробки за допомогою СУБД MS Access, оскільки вона дозволяє опрацьовувати дані в інтерактивному режимі без безпосереднього застосування програмних кодів.

Але виділення окремих атрибутів об'єктів, зведення їх до табличного вигляду та опису фізичного розміщення даних недостатньо для забезпечення однократного введення всіх даних. Розглянемо, наприклад, фрагмент даних таблиці обліку замовлених товарів (рис. 3). При її використанні у випадку замовлення клієнтом декількох товарів (записи 2-5) крім самого товару, його кількості та ціни доводиться щоразу вказувати відомості про того самого клієнта, співробітника, який прийняв замовлення та дату замовлення. Таке **дублювання** даних призводить до **надлишкових** витрат пам'яті і може призвести до **неузгодженості** інформації у різних записах таблиці (наприклад, дані клієнта у п'ятому записі відрізняються від трьох попередніх). Тому під час проектування БД після виділення окремих атрибутів необхідно усунути ці **три аномалії ненормалізованих відношень**.

Співробітник	Клієнт	Дата зам	Назва товару	Кільк	Ціна
Мулявка Снепа	Листопад Євге	03.01.2016	згущене молоко	3	12,50 грн.
Кравцов Петро	Кликов Михай.	05.01.2016	цукерки 'Столич	1	100,00 грн.
Кравцов Петро	Кликов Михай.	05.01.2016	кава 'Галка'	3	50,00 грн.
Кравцов Петро	Кликов Михай.	05.01.2016	морозиво 'Плом	5	5,00 грн.
Кравцов Петро	Дудяк Олекса	05.01.2016	цукерки 'Черво	1	80,00 грн.
Андрюшенко Л	Тітова Тетяна А	06.01.2016	цукерки 'Столич	1	100,00 грн.

Рис. 3. Структура та фрагмент даних реляційної таблиці з дублюваннями, надлишковостями та неузгодженостями

Отже, проектування БД загалом виконують на чотирьох рівнях:

1. На **зовнішньому** рівні визначають перелік задач, котрі будуть розв'язуватися з використанням прикладної програми, та перелік атрибутів, які необхідно зберігати в БД. Таке проектування найчастіше виконує постановник.
2. На **інфологічному** рівні виділяють типові інформаційні об'єкти (сутності) та встановлюють зв'язки між ними з метою мінімізації дублювань, надлишковостей та неузгодженостей ще на етапі проектування, до використання СУБД для створення електронної БД. Найчастіше проектування на цьому рівні виконують проектні менеджери.
3. При проектуванні на **даталогічному** рівні на основі інфологічної моделі будують інформаційно-логічну модель предметної області з урахуванням особливостей та обмежень обраної СУБД. Тут формують концептуальну модель даних, визначають їх типи та характеристики. Як правило, виконують таке проектування розробники програмного забезпечення.

4. На **внутрішньому** рівні формують опис фізичного розміщення даних в пам'яті ПК: визначають формати записів, порядок їх фізичного впорядкування, характеристики та шляхи доступу до даних. Від параметрів фізичної моделі залежить обсяг пам'яті, який займає БД та час реакції системи. Параметри фізичної моделі можуть змінюватися в процесі експлуатації, але ці зміни не повинні зумовлювати зміни на вищих рівнях. І взагалі, зміни в проектах нижчих рівнів не повинні викликати зміни проектів вищих рівнів. Проектування на внутрішньому рівні виконують розробники СУБД та адміністратор. В Access БД та засоби для їх обробки найчастіше зберігаються в одному файлі, тому нам виконувати проектування БД на внутрішньому рівні не потрібно.

В процесі проектування БД розробляють лише структуру таблиць та визначають зв'язки між ними, а дані в таблиці вносять лише після фізичного створення БД в середовищі обраної СУБД. Для прискорення проектування БД на сьогодні широко використовують спеціалізовані інструментальні засоби (наприклад, ERwin), які дають змогу зберігати проміжні результати проектування та автоматизують перехід між його рівнями. Детально основні етапи проектування на перших трьох рівнях описані в завданнях та прикладі виконання цієї лабораторної роботи. Тут наголосимо лише на принципах виділення окремих таблиць (сутностей) і встановлення зв'язків між ними на інфологічному рівні та на виборі типів даних для окремих полів на даталогічному рівні проектування.

В основі процедури агрегації атрибутів по окремих сутностях на інфологічному рівні лежить аналіз співвідношень між ними. Розрізняють чотири типи співвідношень між атрибутами:

1. Атрибути  $A1$  і  $A2$  перебувають у співвідношенні  $1:1$  (один до одного), якщо кожному представнику атрибута з сукупності  $A1$  відповідає не більше одного представника атрибута з сукупності  $A2$  і навпаки (рис. 4а). Якщо між атрибутами виявлено співвідношення  $1:1$ , то такі атрибути відносять до однієї таблиці та присвоюють їй назву. В кожній таблиці виділяють **первинний ключ** – це поле (рідше – сукупність полів), яке однозначно ідентифікує описаний об'єкт предметної області. Таке поле має обов'язково бути унікальним і непорожнім. При використанні об'єкта вказують лише значення з його ключового поля. Наприклад, в таблиці *Товари* (див. рис. 1) ключовим є поле *КодТовару*, яке однозначно ідентифікує окремих товар. Тому для опису замовлень вказаного товару в таблицях обліку замовлень достатньо зберігати лише код цього товару.

Наголосимо, що співвідношення встановлюються між кількостями, а не між значеннями атрибутів, адже, наприклад, декілька співробітників можуть народитися в один і той самий день, але кожен з них має одне прізвище і одну дату народження, тому й співвідношення між цими атрибутами  $1:1$ .

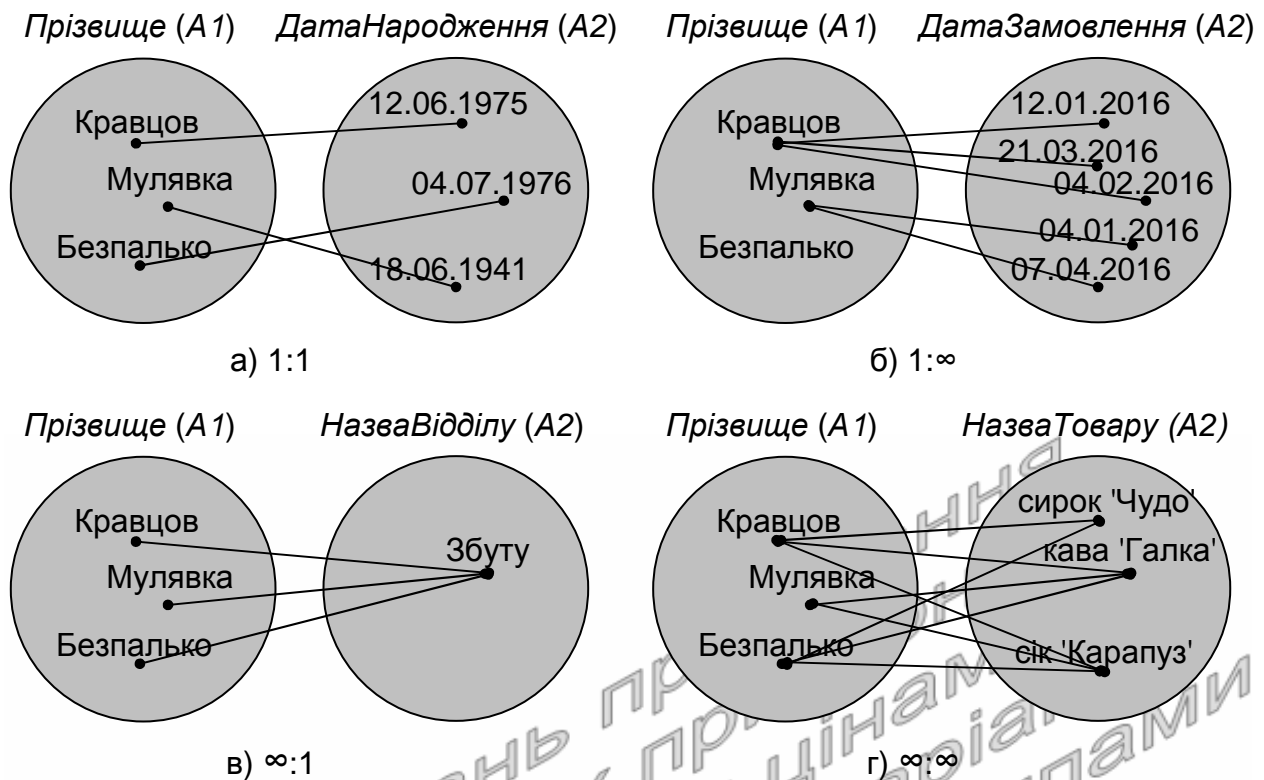


Рис. 4. Різновиди співвідношень між атрибутами

2. Атрибути  $A1$  і  $A2$  перебувають у співвідношенні  $1:\infty$  (один до багатьох), якщо кожному представнику атрибута з сукупності  $A1$  відповідає нуль або декілька представників з сукупності  $A2$ , але кожному представнику атрибута з сукупності  $A2$  відповідає не більше одного представника з сукупності  $A1$ . Наприклад, один співробітник може обслуговувати багато замовлень (рис. 4б), але за кожне замовлення відповідає лише один визначений співробітник. У випадку виявлення між атрибутами співвідношення  $1:\infty$  їх розміщують в різних таблицях і між цими таблицями встановлюють структурний зв'язок. Зв'язок встановлюється між полями з однаковим змістовим навантаженням, причому на стороні  $1$ , як правило, знаходиться первинний ключ, який ідентифікує об'єкт, а на стороні  $\infty$  – числове поле, яке вказує на використання обраного об'єкта (**вторинний ключ**, значення якого в таблиці на стороні  $\infty$  може повторюватися). Тобто для опису взаємодії об'єкта з іншим об'єктом в таблиці на стороні  $\infty$  зберігають лише значення ключового поля цього об'єкта з таблиці на стороні  $1$ . Наприклад, в таблиці *ОблікЗамовлень* (див. рис. 3) атрибути *КодКлієнта* і *КодТовару* перебувають у співвідношенні  $1:\infty$ , оскільки в одному замовленні клієнта може міститися декілька замовлених товарів. Тому цю таблицю необхідно розбити на дві таблиці – *ЗаголовкиЗамовлень* і *ПунктиЗамовлень* (рис. 5). Перша з них має фіксувати факт оформлення замовлення, а друга – містити дані про товари кожного замовлення. Інший приклад: як зазначалося вище,

атрибути *ПІБ* (співробітника) та *КодЗамовлення* перебувають у співвідношенні 1:∞ (див. рис. 4б). Тому ці атрибути мають міститися в різних таблицях (на рис. 5 – відповідно *Співробітники* та *ЗаголовкиЗамовлень*). Ці таблиці пов'язані по полях *КодСпівробітника*. В таблиці *Співробітники* це поле ідентифікує кожного співробітника, а в таблиці *ЗаголовкиЗамовлень* – вказує на того співробітника, який оформив конкретне замовлення.

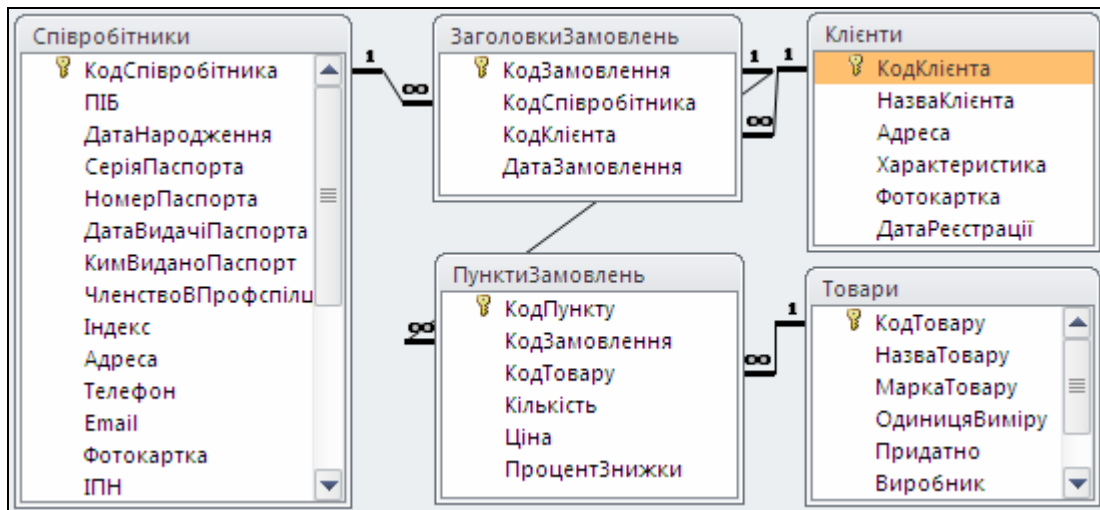


Рис. 5. Структура таблиць для обліку замовлень та пов'язаних об'єктів

3. Співвідношення  $\infty:1$  (рис. 4в) формулюється як обернене до попереднього. Воно реалізується структурним зв'язком  $\infty:1$  між двома таблицями (на рис. 5 такий зв'язок встановлений між таблицями *ЗаголовкиЗамовлень* та *Клієнти* і *ПунктиЗамовлень* та *Товари*).
4. Атрибути *A1* і *A2* перебувають у співвідношенні  $\infty:\infty$  (багато до багатьох), якщо кожному представнику атрибута з сукупності *A1* відповідає нуль або декілька представників з сукупності *A2* і кожному представнику атрибута з сукупності *A2* відповідає нуль або декілька представників з сукупності *A1*. При виявленні між атрибутами співвідношення  $\infty:\infty$  їх розміщують в різних таблицях і створюють ще й одну чи декілька проміжних таблиць, між якими встановлюють структурні зв'язки  $1:\infty$ . Наприклад, між атрибутами *Прізвище* (співробітника) і *НазваТовару* існує співвідношення  $\infty:\infty$ , оскільки кожен співробітник може оформляти замовлення на різні товари і кожен товар може замовлятися у різних співробітників (рис. 4г). Тому ці атрибути розміщують в різних таблицях (на рис. 5 – відповідно *Співробітники* та *Товари*), які пов'язуються через дві проміжні таблиці *ЗаголовкиЗамовлень* та *ПунктиЗамовлень*.

Після виділення сутностей та структурних зв'язків між ними (як на рис. 5) виконують їх перевірку на відповідність умовам нормалізації, з основними положеннями якої ми пропонуємо читачу ознайомитися самостійно [4, с. 60-67].

Вслід за проектуванням на інфологічному рівні основним етапом проектування на даталогічному рівні є визначення **типу даних** для кожного поля з переліку, пропонованого обраною СУБД. Саме тип даних в основному визначає дані, які можуть зберігатися в полі. В Access застосовуються типи даних, наведені в табл. 1.

Таблиця 1

**Типи даних СУБД MS Access та їх призначення**

Тип даних	Характеристика даних
Счетчик	Числове унікальне і непорожнє поле-лічильник, значення якого генеруються автоматично. Найчастіше використовується для первинних ключів
Текстовый / Короткий текст	Короткий текст (до 255 символів), який зберігається безпосередньо в таблиці
Поле MEMO / Длинный текст	Довгий текст, що зберігається поза таблицею, а в самій таблиці міститься лише посилання на нього
Числовой	Як цілі, так і дійсні числа
Денежный	Дійсні числа з максимум чотирма цифрами після коми
Дата/время	Дійсне число, ціла частина якого вказує на кількість днів, які минули до введеної дати, а дробова – на частину доби, що минула до вказаного часу
Логический	Може містити одне з двох значень – <i>Істинно</i> (-1) чи <i>Хибно</i> (0)
Поле объекта OLE	Вбудований OLE-об'єкт (наприклад, графічний, аудіо чи відео фрагмент). Зберігається за принципом MEMO-поля
Гиперссылка	Містить текст для користувача, шлях та місце у файлі чи на сторінці, на яке вказує посилання
Вложение	Один чи декілька вкладених файлів. Зберігаються за принципом MEMO-поля
Вычисляемый	Містить результати обчислень над іншими полями таблиці, застосовується для прискорення розрахунків

Лише після визначення типу даних для кожного атрибута всіх таблиць доцільно приступати до створення БД в середовищі обраної СУБД згідно розробленого проекту (як на рис. 5).

Література: [4, С. 5-80; 5, С. 464-471]

### Підготовчий етап заняття

1. Ознайомтеся з завданнями та прикладом виконання лабораторної роботи. Оберіть тему з наведеної нижче тематики для проектування власної БД та зареєструйте її у викладача.

### Тематика для розробки власного додатку обробки БД

Створення інформаційної системи...

- |                           |  |
|---------------------------|--|
| 1. Служби зайнятості.     | 9. Автозаправної станції.                  |
| 2. Абонементу бібліотеки. | 10. Деканату ВУЗу.                         |
| 3. Бібліотечних фондів.   | 11. Кафедри ВУЗу.                          |
| 4. Кадрової агенції.      | 12. Гуртожитку.                            |
| 5. Музею.                 | 13. Музичного відділу радіостанції.        |
| 6. Центру статистики.     | 14. Автостоянки.                           |
| 7. Архіву.                | 15. Агентства з торгівлі цінними паперами. |
| 8. Проведення аукціонів.  |  |



16. Залізничних перевезень.
17. Автовокзалу.
18. Вантажних перевезень.
19. Авіаперевезень.
20. Метрополітену.
21. Рекламної агенції.
22. Страхової компанії.
23. Банківської установи.
24. Лікарні.
25. Поліклініки.
26. Стоматологічної поліклініки.
27. Аптеки.
28. Готелю.
29. Кінотеатру.
30. Воєнкомату.
31. Хімчистки.
32. Відділення зв'язку.

33. Поштового відділення.
34. Кур'єрської пошти.
35. Податкової адміністрації.
36. Пенсійного фонду.
37. Автомобільного заводу.
38. Диспетчерської радіотаксі.
39. Заводу з виробництва продовольчої продукції.
40. Друкарні.
41. Продовольчого магазину.
42. Промислового магазину.
43. Магазину з торгівлі косметикою.
44. Магазину з торгівлі оргтехнікою.
45. Перукарні.
46. Фотоцентру.
47. Ринку.
48. Туристичної фірми.

### **Проектування БД на зовнішньому рівні**

2. Здійсніть проектування на зовнішньому рівні БД обраної предметної області, виконуючи наступні дії:
  - 2.1. Визначте перелік задач, що мають вирішуватися ІС обробки БД;
  - 2.2. Вивчіть та проаналізуйте первинні документи. Сформууйте перелік реквізитів документів для таблиць оперативної інформації;
  - 2.3. Вивчіть та проаналізуйте нормативно-довідкові документи. Сформууйте перелік реквізитів документів для таблиць умовно-постійної інформації;
  - 2.4. Вивчіть процес перетворення вхідної інформації у вихідну.

### **Проектування БД на інфологічному рівні**

3. Виконайте проектування БД обраної предметної області на інфологічному рівні. Для цього:
  - 3.1. Ліквідуйте синонімію (один і той же атрибут в різних таблицях має різну назву) та омонімію (різні атрибути в різних таблицях мають однакову назву) між атрибутами. Узагальніть окремі атрибути;
  - 3.2. Виконайте агрегацію атрибутів для виділення інформаційних об'єктів (не менше восьми таблиць);
  - 3.3. Сформулюйте запити системи та опишіть хід їх розв'язання;
  - 3.4. Побудуйте графічне зображення інфологічної моделі. Зобразіть та обґрунтуйте письмово кожен структурний зв'язок;

### **Проектування БД на даталогічному рівні**

4. Здійсніть часткове проектування БД обраної предметної області на даталогічному рівні, вказуючи для кожного атрибута, виділеного на інфологічному рівні, тип даних поля СУБД MS Access.

5. Захистіть розроблені інфологічну і даталогічну моделі перед викладачем.

## Приклад проектування БД гуртового складу

### 1. Аналіз предметної області.

#### Проектування БД на зовнішньому рівні

Як відомо, основними видами діяльності будь-якої торгівельної організації, у тому числі і гуртового складу, є отримання та реалізація товарів. Завдяки різниці між цінами продажу та купівлі товарів утворюються прибутки таких організацій. Отримані в результаті діяльності оборотні кошти та чисті прибутки торгівельної організації мають бути спрямовані на придбання тих товарів, запасів яких недостатньо для задоволення реального та прогнозованого попиту. В умовах розмаїтого асортименту товарів гуртового складу ведення обліку обігу продукції без ІС займає багато часу. Ще складніше в цих умовах спрогнозувати попит клієнтів та розрахувати необхідні обсяги закупок. Саме тому при проектуванні БД та створенні ІС гуртового складу слід насамперед автоматизувати облік обігу товарів.

З метою створення проекту БД на зовнішньому рівні спочатку проаналізуємо типову накладну постачання товарів (рис. 6).

від «___» _____ 200__ р.					
Накладна № _____					
Кому _____			_____		
<small>посада</small>			<small>П, І, Б.</small>		
Від кого _____			_____		
<small>посада</small>			<small>П, І, Б.</small>		
№ з-п	Найменування	Од. вим.	К-ть	Ціна	Сума
<b>Всього:</b>					
Здав: _____			Прийняв: _____		
<small>підпис</small>			<small>підпис</small>		
<small>П, І, Б.</small>			<small>П, І, Б.</small>		

Рис. 6. Типовий бланк накладної постачання товару

Як видно з наведеного документа, в накладній постачання насамперед фіксуються реквізити постачальника та отримувача продукції, дата постачання, назва, одиниця виміру, кількість, ціна та сума кожного поставленого товару. В процесі діяльності гуртового складу від кожного постачальника може бути здійснено декілька закупок, тому їх дані доцільно зберігати в окремій таблиці та використовувати при потребі. Крім того, в накладній постачання фіксуються дані співробітника, відповідального за отримання товару. У кожному торгівельному складі є визначене коло співробітників, відповідальних за отримання товарів. Кожен з них приймає декілька постачань в день, тому їх дані доцільно зберігати в окремій таблиці та використовувати

при обліку постачань. Окремий товар постачається на гуртовий склад після того, як сукупний реальний та прогнозований попит перевищить запаси складу, тобто в процесі діяльності кожен товар може надходити декілька разів. Саме тому характеристики товарів доцільно зберігати в окремій таблиці і посилатися на них при потребі.

Так само, аналізуючи накладну відвантаження товарів, яка має вигляд, аналогічний до рис. 6, приходимо до необхідності зберігання даних клієнтів в окремій таблиці. При цьому дані співробітників та товарів мають вказуватися з тих самих таблиць, що й при формуванні постачань. Для аналізу ефективності роботи окремих відділів гуртового складу необхідно також створити окрему таблицю для зберігання їх даних і для кожного співробітника вказати посилання на відділ, де він працює.

Отже, в ІС обліку обігу товарів гуртового складу мають автоматизуватися наступні задачі:

- облік відділів;
- облік співробітників;
- облік постачальників;
- облік клієнтів;
- облік товарів;
- облік постачань товарів;
- облік замовлень товарів.

< Далі вказується перелік атрибутів для зберігання даних кожної задачі і описується процес перетворення вхідної інформації у вихідну. >

## 2. Проектування БД на інфологічному рівні

В процесі проектування БД обліку обігу товарів гуртового складу на інфологічному рівні нами було виділено дев'ять таблиць-сутностей. Графічне зображення структури виділених таблиць та зв'язків між ними наведено на рис. 7. При виділенні таблиць було ліквідовано синонімію і омонімію атрибутів, атрибути *Прізвище*, *Ім'я* та *ПоБатькові* в таблиці *Співробітники* узагальнено в атрибуті *ПІБ*, оскільки їх окреме опрацювання не передбачається.

Між атрибутами кожної окремої таблиці встановлено співвідношення 1:1. Окремі атрибути інформаційних об'єктів потрапили не в одну, а в декілька таблиць, але в кожній з них вони мають різне функціональне призначення:

- атрибут *КодСпівробітника* в таблиці *Співробітники* містить унікальний внутрішній ідентифікаційний номер кожного співробітника, в таблиці *ЗаголовкиПостачань* цей самий атрибут містить код того співробітника, що приймав постачання, а в таблиці *ЗаголовкиЗамовлень* – код співробітника, що виконав замовлення;
- < Аналогічно обґрунтовуються інші дублювання назв атрибутів різних таблиць >.

Для зберігання даних постачань товарів в БД виділено дві таблиці: *ЗаголовкиПостачань* та *ПунктиПостачань*. Перша з них містить дані про сам факт постачання, а друга зберігає відомості про окремі поставлені товари. Між таблицями *ЗаголовкиПостачань* та

*ПунктиПостачань* встановлено співвідношення 1:∞, оскільки за одне постачання може надійти багато товарів.

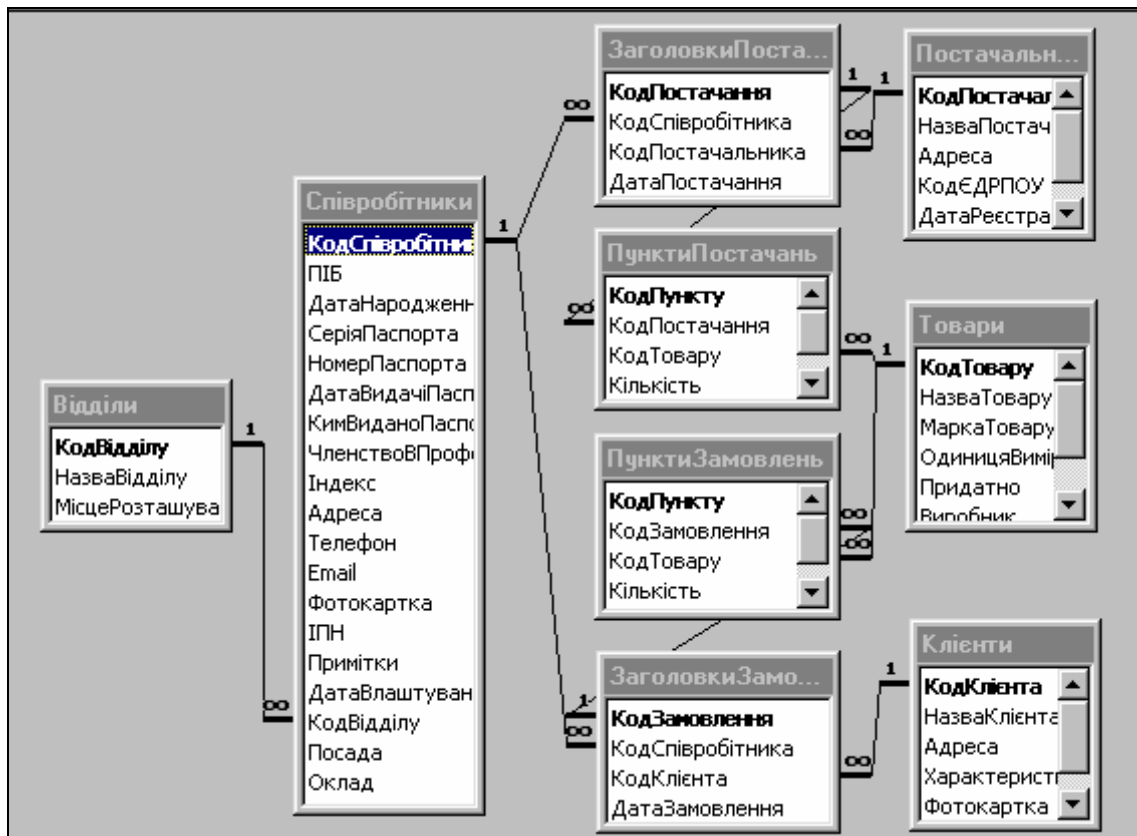


Рис. 7. Схема даних БД обліку обігу товарів гуртового складу на інфологічному рівні

Дані замовлень товарів в БД також зберігаються в двох таблицях: *ЗаголовкиЗамовлень* та *ПунктиЗамовлень*. Перша з них містить дані про сам факт замовлення, а друга зберігає відомості про окремі замовлені товари. Між таблицями *ЗаголовкиЗамовлень* та *ПунктиЗамовлень* встановлено співвідношення 1:∞, оскільки за одне замовлення клієнт може закупити багато товарів.

*< Аналогічно обґрунтовуються інші структурні зв'язки між таблицями схеми даних. Далі описуються запити системи, хід їх розв'язання та атрибути таблиць, які при цьому використовуються. >*

### 3. Проектування БД на даталогічному рівні

Для створення ІС згідно розробленого проекту БД на інфологічному рівні нами було обрано СУБД MS Access, оскільки вона дозволяє:

- розробляти нескладні додатки обробки БД в інтерактивному режимі без створення програмного коду;
- ефективно модифікувати створені об'єкти в режимі конструктора;
- копіювати, імпортувати та експортувати дані таблиць;
- перевіряти і забезпечувати цілісність реляційних відношень між таблицями та даними окремих записів таблиць;
- створювати ефективні засоби для редагування даних таблиць, здійснення пошуку, сортування та фільтрування даних, організації взаємодії між об'єктами.

Недоліки СУБД MS Access пов'язані з проблемами надійного захисту даних та продуктивної одночасної роботи з ІС багатьох користувачів. Але для ІС гуртового складу вони не мають особливого значення, оскільки з такою системою працює обмежене коло співробітників з декількох осіб.

Таблиці ІС розроблені згідно інформаційно-логічної моделі БД (див. рис. 7). Структура та типи даних полів таблиці *Співробітники* наведені в табл. 2.

Таблиця 2

**Структура та типи даних полів таблиці *Співробітники***

Ім'я поля	Тип даних
КодСпівробітника	Счетчик
ПІБ	Текстовый
ДатаНародження	Дата/время
СеріяПаспорта	Текстовый
НомерПаспорта	Числовой
КимКолиВиданоПаспорт	Текстовый
ЧленствоВПрофспілці	Логический
Індекс	Числовой
Адреса	Текстовый
Телефон	Текстовый
Email	Текстовый
Фотокартка	Поле OLE
ІПН	Текстовый
Примітки	Поле MEMO
ДатаВлаштування	Дата/время
КодВідділу	Числовой
Посада	Текстовый
Оклад	Денежный

< Аналогічно описуються структури та типи даних полів інших таблиць ІС >.

### Контрольні запитання

- Що таке БД? СУБД? Чому на практиці найчастіше використовують реляційні БД?
- На яких рівнях виконують проектування БД? З чим це пов'язано? Як узгоджуються між собою моделі кожного рівня проектування? Який результат проектування БД на кожному рівні?
- Які кроки необхідно виконати при проектуванні БД на зовнішньому рівні?
- Які кроки необхідно виконати при проектуванні БД на інфологічному рівні?
- Чому атрибути предметної області, як правило, неможливо розмістити в одній таблиці? На що вказують структурні зв'язки між таблицями?
- Що таке первинний і вторинний ключ? Які типи даних обирають для таких полів?
- На якому рівні проектування БД враховують особливості та обмеження обраної СУБД? Який найважливіший етап цього рівня проектування?



## Лабораторна робота № 2

**Тема.** Створення БД в середовищі СУБД MS Access. Основні сервісні функції обробки БД.

**Мета.** Формування вмінь та навичок створення і обробки БД різними способами, зберігання та конвертування БД. Закріплення навичок використання ОС та ППЗ для керування файлами та програмами.

### Теоретичні відомості


Для початку розробки БД в Access застосовуються два основні режими: за допомогою шаблону та з нової (порожньої) БД. Перший з них рекомендується застосовувати, коли існує шаблон, споріднений з предметною областю для розробки БД. Тоді створену на основі обраного шаблону БД достатньо лише вдосконалити, а не розробляти її "з нуля". Якщо ж потрібного шаблону віднайти не вдалося, то використовується другий режим. БД Access можуть зберігатися як на віддалених серверах (серверна частина), так і на локальному диску (клієнтська частина). В учбових цілях ми будемо створювати лише локальні БД (за допомогою піктограм створення середовище Access без зображення глобуса).

В СУБД MS Access в одному файлі БД крім взаємопов'язаних таблиць найчастіше зберігаються ще й інструментальні засоби для їх обробки, тобто в одному файлі БД може міститися декілька об'єктів. Кожен об'єкт належить до одного з визначених типів. Основні типи об'єктів БД Access такі:

- **таблиці** – це поіменовані реляційні відношення, що використовуються для зберігання інформації про певні сутності (тобто про об'єкти предметної області або про їх взаємодію). Зв'язки між таблицями Access відображаються та корегуються в **схемі даних**;
- **форми** – це спеціалізовані вікна чи вкладки, що використовують для коректного і зручного введення та аналізу даних таблиць, а також для управління іншими об'єктами додатку;
- **запити** – це команди, записані по стандарту **SQL** (Structured Query Language – структурована мова запитів), які використовують для пакетної обробки записів таблиць чи їх поєднань (вибірки даних, створення нових, оновлення чи видалення існуючих записів, автоматичного створення нових таблиць та ін.);
- **звіти** – це паперові чи електронні документи з результатами обробки даних таблиць БД;
- **макроси** – це поіменовані послідовності команд для автоматизації обробки даних в додатку, що можуть викликатися багаторазово;
- **модулі** – це сукупності підпрограм, написаних мовою програмування VBA, які використовуються для автоматизації роботи додатків чи обробки даних.

Для підвищення ефективності обробки даних у формах і звітах в Access використовуються події. **Подія** – це будь-яка дія, що розпізнається системою і на яку можна створити відгук. Відгуком на подію може бути макрос чи підпрограма з модуля.

Наголосимо, що дані про об'єкти предметної області чи про їх взаємодію зберігаються тільки у взаємопов'язаних реляційних таблицях. Всі інші об'єкти – це інструментальні засоби для підвищення ефективності обробки даних.

Для переходів між об'єктами в БД Access використовується *Область переходов (навігації)*, яка розміщується зліва від робочого поля (рис. 8). З метою організації впорядкованого доступу до всіх об'єктів БД доцільно, по-перше, розгорнути цю область за допомогою кнопки  у її верхньому правому кутку, по-друге, згрупувати об'єкти за типами і, по-третє, відобразити всі об'єкти БД (як на рис. 8).

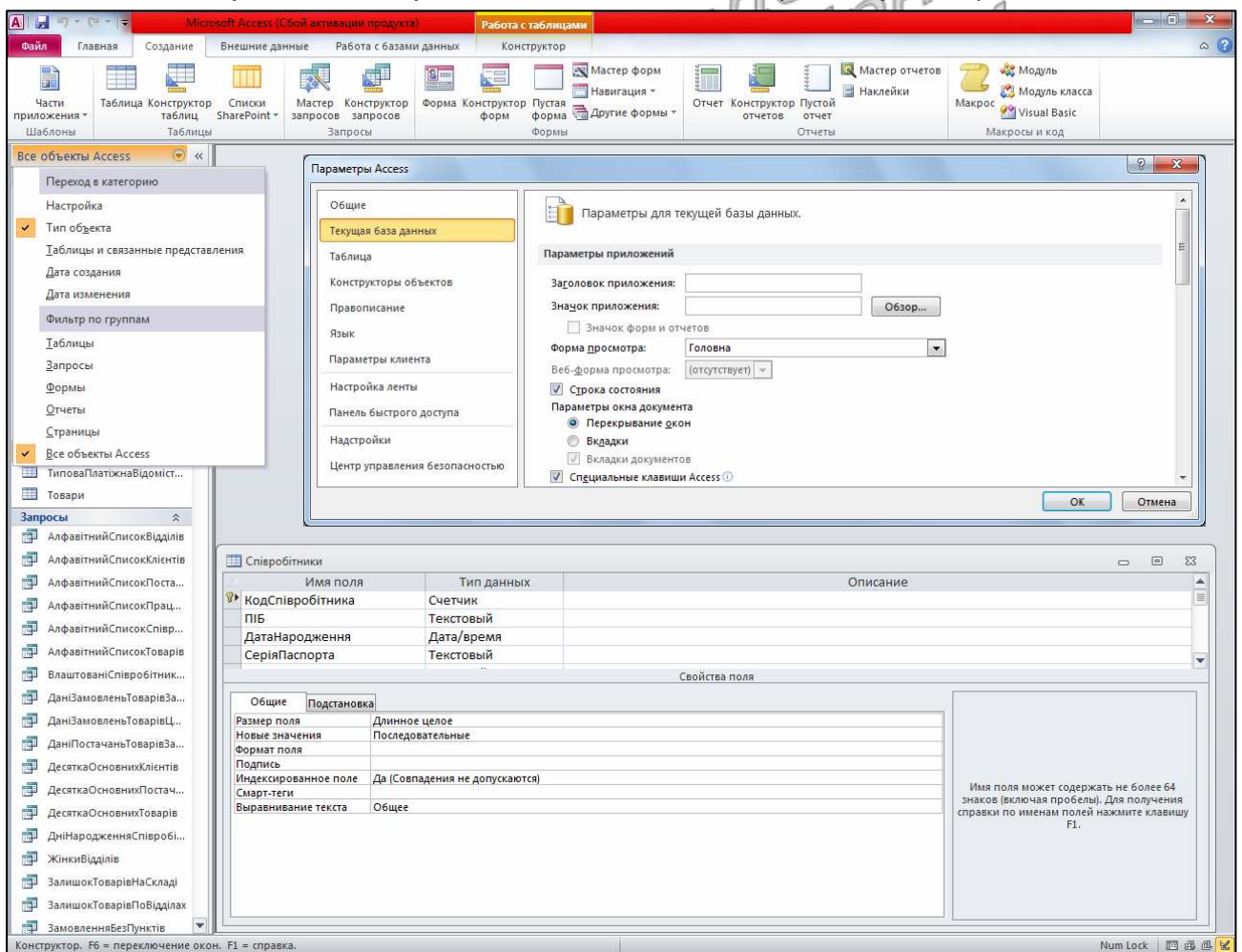


Рис. 8. Область переходів, вікно параметрів та вікно таблиці в режимі конструктора

Всі об'єкти БД Access можуть відображатися в окремих вікнах або вкладках. Встановлення режиму відображення об'єктів здійснюється у вікні параметрів Access (*Файл ► Параметри*) на вкладці *Текущая база данных* (див. рис. 8). Ми рекомендуємо відобразити об'єкти у

вигляді окремих вікон, оскільки це дозволяє ефективніше співставляти їх дані.

Використовуючи область переходів, кожен об'єкт БД Access можна відразу завантажити в **режимі застосовування** (для таблиць і запитів – це режим таблиці, для форм – режим форми, для звітів – режим попереднього перегляду чи його подання). Цей самий режим використовується для об'єктів БД Access при виборі в контекстному меню їх назви пункту *Открыть*. Структура візуальних об'єктів БД Access змінюється в режимі **конструктора** (див. нижню частину рис. 8). Для переходу в цей режим потрібно використати однойменний пункт контекстного меню об'єкта в області переходів чи заголовка вікна завантаженого об'єкта. Інші режими використання об'єктів залежать від їх типу (наприклад, для форм і звітів використовується також режим **макету**). Для швидкого переходу між режимами використання відкритого об'єкта ми рекомендуємо застосовувати першу кнопку *Режим* з вкладки *Главная* стрічки меню та її список (на місці кнопки *Части приложения* на рис. 8), оскільки ця кнопка дає змогу відразу перейти до найуживанішого альтернативного режиму (наприклад, для режиму застосовування – це режим конструктора, і тому ця кнопка має




вигляд **Режим**), а її список – до довільного іншого режиму. При цьому об'єкт не потрібно щоразу закривати і відкривати знову.

Серед сервісних функцій обробки БД, які надає СУБД MS Access, використаємо в цій лабораторній роботі дві найуживаніших:


- **стиснення БД** – це автоматичне виконання модифікацій БД на фізичному рівні з метою зменшення розміру файла БД без внесення змін в логічну модель. При цьому самі дані не стискаються, а з файла вилучаються об'єкти і дані, що вже не використовуються (наприклад, при видаленні записів з таблиць вони лише помічаються як видалені, а фізично знищуються при стисненні БД). За рахунок зменшення розміру файла БД стиснення пришвидшує обробку її об'єктів. Ми рекомендуємо стискувати БД в процесі розробки після кожного сеансу роботи, а під час експлуатації – хоча б раз на тиждень;
- **конвертування БД** – це автоматичне зберігання вмісту БД у новому файлі іншого формату без внесення змін в дані та логічну модель. Зрозуміло, що СУБД може виконувати конвертування до і з одного з попередніх форматів. Найефективніше СУБД працює з активним форматом, а для перетворення БД до попереднього формату версій Access необхідно вилучити з неї всі складові частини, що використовують нові можливості, які не підтримуються цим попереднім форматом.

Література: [4, С. 93-95; 6, С. 115-120]

## Підготовчий етап заняття. Актуалізація знань

1. Завантажте Access п'ятьма способами:
  - 1.1. За допомогою ярлика на робочому столі (за наявності);
  - 1.2. При допомозі відповідного підпункту головного меню *Все программы* ► *Microsoft Office*;
  - 1.3. З вікна обробки результатів пошуку файлу *Msaccess.exe*;
  - 1.4. З вікна *Выполнить* чи *Запуск программы* (шлях до програми слід вписати з області результатів вікна пошуку після виконання попереднього підпункту);
  - 1.5. З вікна програми *Проводник* при перегляді вмісту папки, що містить файл *Msaccess.exe* (шлях до папки слід вписати з області результатів вікна пошуку після виконання підпункту 1.3).
2. Завершіть роботу Access п'ятьма способами:
  - 2.1. Комбінацією клавіш Alt-F4;
  - 2.2. Натисненням кнопки вікна ;
  - 2.3. Вибором пункту меню *Файл* ► *Выход*;
  - 2.4. Вибором пункту *Закреть* системного меню вікна Access;
  - 2.5. Вибором пункту *Закреть окно* чи *Закреть* контекстного меню піктограми Access на панелі завдань.

## Створення БД з використанням шаблонів

3. Створіть БД для обліку виконання завдань за допомогою шаблону. Для цього:
  - 3.1. Після завантаження Access оберіть на вкладці *Файл* ► *Создать* посилання *Образцы шаблонов*;
  - 3.2. На сторінці вибору шаблонів оберіть серед запропонованих шаблон *Задачи* (без зображення глобуса), біля рядка вводу імені файла за допомогою кнопки  перейдіть у власну папку (назва цієї папки має містити шифр групи та ваше прізвище. Якщо така папка відсутня, то створіть її за допомогою контекстного меню), у рядку вводу імені файла введіть *Proba1* та натисніть кнопку *Создать*;
  - 3.3. Дочекайтеся завершення створення БД з використанням шаблону.
4. З автоматично завантаженої форми перейдіть у форму *Список контактов* за допомогою відповідної кнопки справа. Де відкрилася ця додаткова форма? Перейдіть до попередньої форми та знову поверніться до форми контактів.
5. Внесіть у список контактів дані п'яти ваших друзів-співробітників. Для переходу між записами використовуйте кнопки переходу у нижній частині форми.
6. Внесіть додаткові відомості про співробітників (не менше одного поля для кожного запису на вибір), відкривши відповідну форму за



допомогою посилання-ідентифікатора на початку кожного запису. Звідки у цій формі взялися дані?

7. Перейдіть до форми списку контактів та поверніться до форми списку завдань. Внесіть не менше 7 завдань з різним рівнем важливості та ступенем виконання для різних співробітників. Як обирається співробітник, якому призначено завдання? Куди і для чого виконується перехід при подвійному натисненні лівою кнопкою мишки на цьому полі?
8. Ознайомтесь зі звітами системи за допомогою списку у верхній частині форми обліку завдань. Продемонструйте викладачу створену БД.
9. Закрийте відкриті форми та впорядкуйте область навігації зліва за типом об'єктів. З'ясуйте призначення кожного типу об'єктів, переглянувши кожен об'єкт як в режимі використання, так і в режимі конструктора за допомогою контекстного меню. Закрийте всі відкриті об'єкти БД. Якщо при цьому випадково закрилася створена БД, то завантажте її знову зі своєї папки.
10. Перегляньте схему зв'язків між таблицями БД, натиснувши на вкладці *Работа с базами данных* кнопку *Схема данных*. Логічно обгрунтуйте окремі зв'язки між таблицями.
11. Закрийте Access, віднайдіть створену БД у вікні програми *Проводник* та визначте її розмір.
12. Знову завантажте Access та відкрийте створену БД (пункт меню *Файл ► Открыть*) або відкрийте її безпосередньо з вашої папки.
13. Для оптимізації структури та зменшення розміру БД виконайте її стиснення (кнопка стрічки *Файл ► Сведения ► Сжать и восстановить базу данных*).
14. Повторно виконайте пункт 11. Чи зменшився розмір БД?
15. Створіть за допомогою майстра подібну БД *Proba2*, не вносячи в неї дані.
16. Повторно виконайте пункт 11. Чому розмір файла БД *Proba2* менший від розміру файла БД *Proba1*?
17. Завантажте Access та створіть **нову** БД *Proba3*, обравши на вкладці *Файл ► Создать* тип *Новая база данных* (без зображення глобуса). Чому в цій БД відсутні таблиці, форми та звіти?
18. Завантажте Access, відкрийте БД *Proba3* та конвертуйте її до версії Access-2002, створивши БД *Proba4* (для цього використайте можливість вкладки *Файл ► Сохранить и опубликовать*).
19. Відкрийте та перегляньте БД *Proba4*;
20. Закрийте БД *Proba4* та самостійно конвертуйте її до поточної версії Access, створивши файл *Proba5*. Порівняйте розміри файлів *Proba3* та *Proba5*.
21. Самостійно порівняйте розміри файла *Proba1* та інших створених БД. Зробіть висновки.



### **Завершальний етап заняття. Повторення вивченого матеріалу**

22. Завантажте БД *Proba1* та збережіть її резервну копію на власний флеш-носій за допомогою пункту меню *Файл – Сохранить базу данных как*.
23. Створіть архіви *Proba1.zip*, *Proba2.zip*, *Proba3.zip*, *Proba4.zip* та *Proba5.zip* з відповідних БД у власній папці.
24. Створіть архів *Proba.zip* з чотирьох створених БД.
25. Порівняйте розмір архіву *Proba.zip* з загальним розміром архівів *Proba1.zip*, *Proba2.zip*, *Proba3.zip*, *Proba4.zip* та *Proba5.zip*. Зробіть висновки.
26. Збережіть бази даних та створені архіви на власний флеш-носій у новій папці.
27. Після захисту роботи знищіть створені БД та архіви у власній папці на комп'ютері в аудиторії.
28. Розархівуйте архів *Proba1.zip* з флеш-носія відразу у власну папку на комп'ютер в аудиторії. Знищіть цю БД. Знищіть створені архіви та папку на флеш-носії.

### **Контрольні запитання**

1. Які основні режими створення БД в середовищі СУБД MS Access?
2. Коли слід створювати БД з використанням шаблону?
3. Коли слід створювати нову БД?
4. Чим відрізняються БД Access від класичних реляційних БД?
5. Як здійснити стиснення БД? Чому при послідовному повторному стисненні розмір файла БД не зменшується?
6. Що таке конвертування БД? Як здійснити конвертування БД в Access?
7. Як скопіювати БД в іншу папку?
8. Як слід зберігати БД на змінних носіях?
9. Чому недоцільно працювати з БД безпосередньо зі змінного носія?

## Лабораторна робота № 3

**Тема.** Створення таблиць засобами СУБД MS Access.

**Мета.** Формування вмінь та навичок створення, редагування та обробки даних таблиць БД різними способами. Вдосконалення навичок використання ОС та ППЗ для керування файлами та програмами.

### Теоретичні відомості

В СУБД MS Access **таблиці створюють такими способами:**

- **шляхом імпорту з інших БД**, електронних таблиць чи текстових документів. Цей спосіб використовують, коли структура чи дані вже існуючої таблиці в іншому файлі подібні створюваній таблиці. Після імпорту не потрібно розробляти таблицю з самого початку, достатньо відкоригувати її структуру чи дані;
- **в режимі таблиці**, де відразу створюються нові поля-стовпці, а в записах вносяться **дані**. Цей режим застосовується, коли відомі дані об'єктів предметної області, але до кінця не визначена структура таблиці. При цьому окремі (але не всі) додаткові властивості поля можна задати з вкладки інструментів стрічки меню;
- **в режимі конструктора**, де вказується **структура** нової таблиці. Дані в таку таблицю вносяться після її збереження і переходу в режим таблиці. Цей режим дозволяє задати всі властивості як полів, так і таблиці в цілому. Даний спосіб створення використовується, коли структура нової таблиці вже відома.

Після створення таблиць БД визначити, яким способом кожна з них була створена неможливо, адже кожна з них може експлуатуватися як в режимі таблиці, де корегуються дані, так і в режимі конструктора, де модифікується її структура.


Для кожної таблиці в БД Access рекомендується визначити первинний ключ (ключове поле), який ідентифікує кожен запис. Як правило, таке поле має тип даних *Счетчик*, і тому його значення генеруються автоматично та є унікальними і непорожніми.

Література: [4, С. 98-102; 6, С. 123-154]

### Підготовчий етап заняття. Актуалізація знань

1. Створіть у власній папці **нову** БД *Sklad*.

### Створення таблиць шляхом імпорту з інших БД

2. Для зберігання інформації про співробітників складу перенесіть у власну БД таблицю *Сотрудники* з БД *Борей* та перейменуйте створену таблицю на *Співробітники*. Для цього:
  - 2.1. Створіть та відкрийте нову БД з шаблону *Борей*. З цією метою оберіть послідовно *Файл* ► *Создать* ► *Образцы шаблонов* ► *Борей*, перейдіть у власну папку для збереження БД за допомогою кнопки  та натиснувши кнопку *Создать*;

- 2.2. У створеній на основі шаблону БД *Борей*:
  - 2.2.1. Ввімкніть активні елементи БД, використовуючи рядок попереджень;
  - 2.2.2. У вікні входу оберіть дані довільного співробітника та натисніть *Вход*;
  - 2.2.3. Закрийте вікно головної вкладки за допомогою її контекстного меню;
  - 2.2.4. У лівій частині вікна розгорніть область переходів. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип объекта*;
  - 2.2.5. Розгорніть розділ *Таблицы* в області переходів. Віднайдіть там таблицю *Сотрудники* та скопіюйте її в буфер обміну;
- 2.3. У тому ж програмному елементі *Access* відкрийте власну БД *Склад*. При цьому БД *Борей* має закритися автоматично;
- 2.4. Вставте скопійовану таблицю з буфера обміну під назвою *Співробітники*. Якщо вставити скопійовану таблицю через буфер обміну не вдалося, то імпортуйте її безпосередньо. Для цього натисніть кнопку *Внешние данные* ► *Access*, використовуючи кнопку *Обзор*, виберіть БД *Борей* у власній папці, після натиснення *OK* виберіть для імпорту таблицю *Сотрудники*, закрийте майстер імпорту та перейменуйте імпортовану таблицю на *Співробітники*.
3. У власній БД *Склад* в лівій частині вікна розгорніть область переходів. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип объекта*. Використовуючи меню області переходів, відобразіть всі об'єкти БД.
4. Віднайдіть та завантажте таблицю *Співробітники*. Звідки в ній взяли дані?
5. Введіть змістовні дані ще про двох співробітників організації. В які поля дані вносяться некоректно?
6. Завершіть внесення даних в таблицю, заклавши активне вікно (а не вікно БД).

### **Створення таблиці шляхом введення даних**

7. Для зберігання інформації про товари, що реалізуються та отримуються складом, створіть таблицю *Товари* шляхом внесення даних. Для цього:
  - 7.1. На вкладці стрічки меню *Создание* в групі *Таблицы* натисніть кнопку *Таблица*. Врахуйте, що в цьому режимі інформація про окремий товар вноситься в одному рядку; кожен критерій характеристики всіх товарів описується в одному стовпці; для створення нового стовпця необхідно натиснути кнопку *Щелкните для добавления* у заголовку таблиці та обрати відповідний тип даних. Створіть таблицю (як на рис. 1, але з

вашими даними), вказавши для п'яти різних товарів у відповідних **стовпцях** наступні характеристики:

- 7.1.1. Код товару (поле-лічильник, значення якого генеруються автоматично);
  - 7.1.2. Назву товару;
  - 7.1.3. Марку товару;
  - 7.1.4. Одиницю виміру;
  - 7.1.5. Максимальний термін придатності в днях;
  - 7.1.6. Фірму-виробника товару;
  - 7.1.7. ДСТУ (номер державного стандарту України, якому відповідає товар);
- 7.2. Перейменуйте стовпці, давши їм скорочені назви згідно попереднього підпункту;
- 7.3. Завершіть створення таблиці, закривши її вікно. Підтвердіть збереження таблиці, дайте їй назву *Товари* та дозвольте Access при потребі самостійно створити ключове поле.

#### Створення таблиці в режимі конструктора

8. Для зберігання інформації про клієнтів складу створіть таблицю *Клієнти* в режимі конструктора. Для цього:
- 8.1. На вкладці стрічки меню *Создание* в групі *Таблицы* натисніть кнопку *Конструктор таблиць*;
  - 8.2. Вкажіть назви та типи даних полів цієї таблиці згідно табл. 3.

Таблиця 3

#### Структура та типи даних полів таблиці *Клієнти*

Ім'я поля	Тип даних
КодКлієнта	Счетчик
НазваКлієнта	Текстовый
МісцеРоботиКлієнта	Текстовый
ДатаРеєстрації	Дата/Время
Фотокартка	Поле объекта OLE
Характеристика	Поле MEMO

- 8.3. Для редагованих полів задайте відповідну характеристику в стовпці *Описание* (підказку, що виводиться в режимі таблиці в рядку стану при введенні значення поля, наприклад, *Введіть основне місце роботи клієнта*);
- 8.4. Перемістіть поле *Характеристика* вище поля *Фотокартка*. Для цього:
- 8.4.1. Виділіть рядок опису поля *Характеристика*, відмітивши прямокутник перед описом поля;
  - 8.4.2. Перетягніть мишкою цей прямокутник вище поля *Фотокартка*;
- 8.5. Відмітивши при натиснутій клавіші *Shift* поля *Характеристика* та *Фотокартка* аналогічно попередньому пункту перемістіть їх вище поля *ДатаРеєстрації*;

- 8.6. Задайте для поля *КодКлієнта* властивість ключового поля, відмітивши в контекстному меню цього поля властивість *Ключевое поле*. Як ще можна встановити цю властивість?
- 8.7. Відмітьте одночасно декілька полів з типами даних *Счетчик*, *Текстовый* чи *Дата/Время* при натиснутій клавіші *Ctrl* та аналогічно встановіть для них властивість сукупного ключового поля;
- 8.8. Відмініть всі встановлені властивості ключового поля, віднайшовши відповідну кнопку на вкладці *Главная* стрічки меню. Встановіть властивість ключового поля лише для поля *КодКлієнта*;
- 8.9. Завершіть створення таблиці, заклавши її вікно. Підтвердіть збереження таблиці та дайте їй назву *Клієнти*. Відкрийте цю таблицю з області навігації в режимі таблиці та внесіть в неї змістовні дані п'яти клієнтів;
- 8.10. Перейдіть в режим конструктора цієї таблиці безпосередньо з режиму використання, натиснувши кнопку *Режим* вкладки *Главная* стрічки меню. Який вигляд цієї кнопки? Як він змінився в режимі конструктора таблиці? Чому? Проаналізуйте призначення інших пунктів списку цієї кнопки.

#### **Завершальний етап заняття. Повторення вивченого матеріалу**

9. Стисніть створену БД.
10. Завершіть роботу Access.
11. Віднайдіть створену БД та заархівуйте її вміст.
12. Збережіть створений архів на власному носії у двох екземплярах.

#### **Контрольні запитання**

1. Які способи створення таблиць використовуються в Access?
2. Як імпортувати таблицю іншої БД через буфер обміну? Як виконати цю операцію безпосередньо?
3. Які властивості полів неможливо задати при створенні таблиці шляхом введення даних?
4. В чому функціональна різниця між клавішами *Ctrl* та *Shift* в режимі конструктора таблиці?
5. Для чого використовується на стрічці меню у вкладці *Главная* кнопка *Режим*?
6. Оцініть ефективність та гнучкість різних способів створення таблиць.



## Лабораторна робота № 4

**Тема.** Редагування властивостей полів таблиць в режимі конструктора. Встановлення структурних зв'язків між таблицями майстром підстановок та в схемі даних.

**Мета.** Формування вмінь та навичок корегування властивостей полів таблиць в режимі конструктора, встановлення структурних зв'язків між таблицями БД. Систематизація та застосування знань стосовно забезпечення цілісності інформації БД в процесі її розробки. Закріплення навичок створення таблиць та використання сервісних функцій обробки БД.

### Теоретичні відомості

Для комплексного задання чи корегування структури таблиць в Access найчастіше використовують режим конструктора (рис. 9).

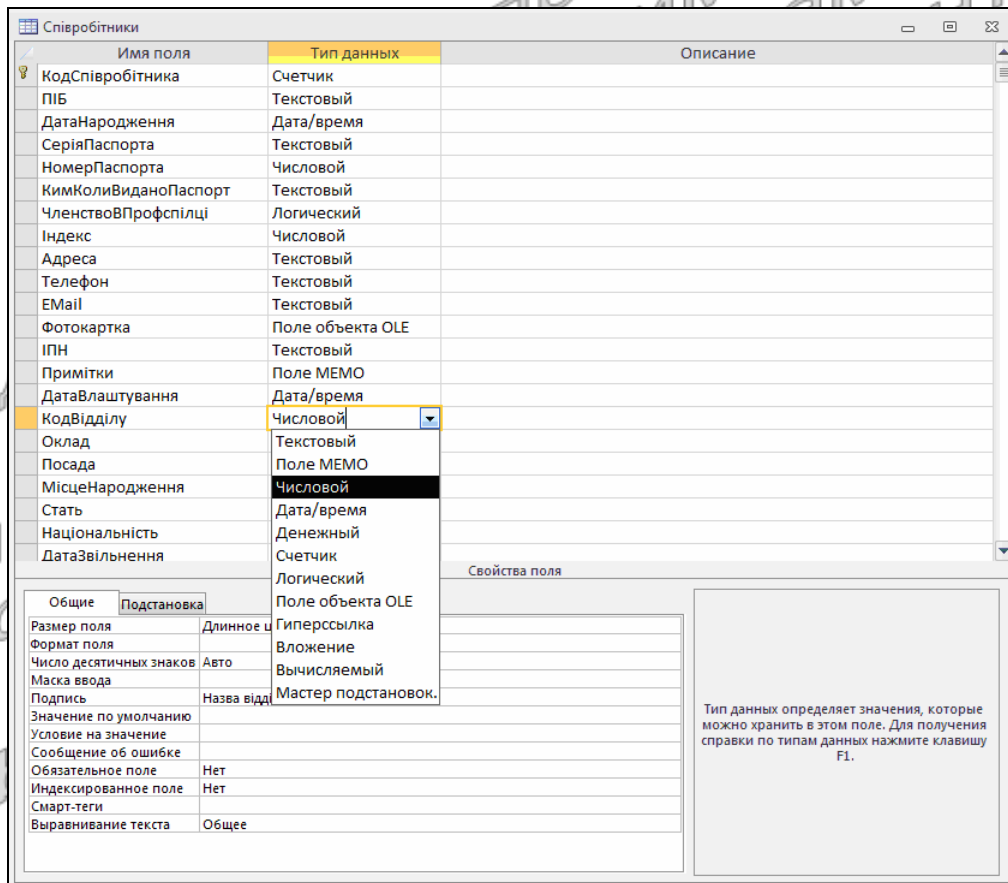


Рис. 9. Вигляд таблиці *Співробітники* в режимі конструктора

У верхній частині режиму конструктора задають базові властивості полів таблиці:

➤ **имя поля** – це унікальний ідентифікатор поля в межах таблиці. Воно може містити не більше 64 знаків крім ` , [ , ] , . , !. Рекомендується слова в імені набирати без пробілів, починаючи кожне з великої літери (наприклад, *ДатаПостачання*), адже імена з пробілами, що

використовуються в інших об'єктах, для однозначності звертань потрібно брати в квадратні дужки. Найчастіше ім'я поля співпадає з назвою відповідного атрибута сутності;

- **тип данных** – визначає значення, що можуть зберігатися в полі. Різновиди типів даних Access детально охарактеризовані в табл. 1;
- **описание** – це необов'язковий текст, що пояснює призначення поля, а також виводиться в рядку стану при активізації поля в режимі таблиці чи у формі;
- **ключевое поле** – це позначення (ключем біля імені) одного чи декількох полів, що в сукупності задають первинний ключ таблиці. Наголосимо, що позначення декількох полів не означає створення декількох ключів. Так позначається первинний ключ, що забезпечує непорожність і унікальність набору значень полів у кожному записі. Встановлюється таке позначення після виділення полів з їх контекстного меню або кнопкою *Ключевое поле* вкладки інструментів *Конструктор* стрічки меню. Найчастіше дана властивість задається для поля з типом даних *Счетчик* якщо не користувачем, то автоматично Access.

**Додаткові властивості полів таблиці** містяться в нижній частині режиму конструктора. Їх перелік залежить від обраного типу даних, а для числових полів – ще й від довжини поля. Стандартні значення цих властивостей (крім підпису), як правило, підходять для більшості полів, тому й змінюються порівняно рідко. Ми будемо використовувати такі додаткові властивості:

- **підпис** – це короткий текст, що супроводжує поле в режимі таблиці, у формах і звітах. Він пояснює призначення поля користувачу. Коли підпис не вказано, то замість нього використовують ім'я поля. Якщо ж підпис містить декілька слів, то вони розмежовуються пробілами (наприклад, *Дата постачання*). Взагалі кажучи, ім'я поля відповідає даним, які в ньому зберігаються, а підпис – тому що відображається. Наприклад, у полі *КодВідділу* таблиці *Співробітники* має зберігатися код того відділу, де працює співробітник, але відобразитися відповідна назва відділу, тому ім'я поля буде *КодВідділу*, а підпис – *Назва відділу*;
- **размер поля** – для коротких текстових полів ця властивість визначає максимальну кількість знаків, що можуть вноситися в поле (до 255), а для числових – тип чисел та їх розмір в байтах;
- **значение по умолчанию** – це значення, що автоматично заноситься в поле при створенні нового запису. Наголосимо, що встановлене у властивості значення не впливає на раніше створені записи;
- **маска ввода** – це набір символів, що задає шаблон для зручного введення даних;
- **формат поля** – задає формат для виводу даних на екран і принтер (наприклад, для числових полів може додатково відображати

позначення грошової одиниці). Ця властивість впливає на відображення, а не на значення поля. Найчастіше формат обирається зі списку, але може й задаватися за допомогою спеціальних знаків. Ці знаки відрізняються для текстових, числових полів і полів дати/часу;

- **число десятичних знаків** – це кількість знаків, що відображаються (а не зберігаються) для числа після коми;
- **условие на значение / правило проверки** – вираз, що задає умову яку має задовольняти значення поля для його зберігання. Умови для перевірки узгодженості даних різних полів задаються у вікні властивостей таблиці, що викликається з контекстного меню верхньої частини режиму конструктора;
- **сообщение об ошибке** – містить текст-повідомлення, що з'являється на екрані у випадку порушення умови на значення;
- **обязательное поле** – вказує, чи обов'язково вносити значення в поле. При відсутності значення видається стандартне повідомлення Access;
- **пустые строки** – дозволяє чи забороняє введення порожніх рядків (нульової довжини). У випадку заборони порожній рядок автоматично перетворюється на Null;
- **индексированное поле** – використовується для створення ієрархічного дерева (індекса) по значеннях поля з метою прискорення їх пошуку, сортування і фільтрування. Встановлення значення властивості рівним *Да (совпадения не допускаются)* дає змогу забезпечити унікальність значень поля у кожному записі.

Перед створенням структурних зв'язків між таблицями в Access необхідно визначитися, які таблиці будуть виводитися в одному, а які – в різних вікнах (вкладках). **Якщо таблиці планується редагувати в одному вікні, то зв'язок між ними створюється перетягуванням в схемі даних. Коли ж таблиці мають редагуватися у різних вікнах, то зв'язок між ними створюється майстром підстановок.** Наприклад, у різних вікнах виводяться таблиці оперативної і умовно-постійної інформації. Тому для розробленої нами схеми (див. рис. 7) у різних вікнах будуть корегуватися дані відділів, співробітників, постачальників, клієнтів, товарів відносно постачань і замовлень. З іншого боку, в одному вікні можуть корегуватися таблиці, що відображають один процес. Ось чому таблиця *ЗаголовкиЗамовлень* буде редагуватися в одному вікні з таблицею *ПунктиЗамовлень*, а таблиця *ЗаголовкиПостачань* – разом з таблицею *ПунктиПостачань* і зв'язки між ними доцільно створювати перетягуванням в схемі даних.

Майстер підстановок Access (останній пункт розгорнутого списку на рис. 9), організовуючи зв'язки між таблицями, забезпечує вибір (**зберігання**) у полі вторинного ключа (на стороні ∞) значення з поля первинного ключа (зі сторони 1) та **відображення** основних інформаційних полів обраного об'єкта. Саме тому тип значень і розмір

поля вторинного ключа має співпадати з цими ж параметрами первинного ключа. Наприклад, якщо на стороні 1 знаходиться тип даних *Счетчик*, то на стороні ∞ потрібно обрати тип даних *Числовой* (адже на цій стороні значення можуть повторюватися) і розмір поля *Длинное целое*.

Отже, майстер підстановок потрібно завантажувати після вибору типу даних поля на стороні ∞, виконуючи підстановку з поля зі сторони 1. Наприклад, для кожного співробітника гуртового складу, потрібно вказати відділ, куди його зараховано (хоча в одному відділі може працювати багато співробітників). Для цього в таблиці *Співробітники* створюється числове поле *КодВідділу*, для якого викликається майстер підстановок (див. рис. 9). Підстановка формується з таблиці *Відділи*, з якої обирається два поля: *КодВідділу* – для виконання підстановки, *НазваВідділу* – для відображення у списку підстановки. Після цього при внесенні даних кожного співробітника користувач зможе розгорнути список і обрати назву відділу, де він працює, і тільки розробник БД знає, що при цьому зберігається не довга назва відділу, а її код.

Всі встановлені зв'язки між таблицями в Access відображаються і корегуються в схемі даних. Використовуючи встановлені зв'язки, Access додатково може організувати **перевірку посилкової цілісності**. Ця перевірка гарантує існування для кожного встановленого значення вторинного ключа (на стороні ∞) відповідного значення первинного ключа (на стороні 1). При цьому редагування пов'язаного значення на стороні 1 або забороняється, якщо **режим каскадного оновлення** не встановлено, або виконується автоматично для всіх пов'язаних записів на стороні ∞ в протилежному випадку. Аналогічно, видалення запису з пов'язаним значенням на стороні 1 або забороняється, якщо **режим каскадного видалення** не встановлено, або виконується автоматично разом з пов'язаними записами на стороні ∞ в протилежному випадку.

Майстер підстановок Access може також організувати вибір з фіксованого набору, якщо такий набір не змінюється з часом (наприклад, одиниця виміру для кожного товару). Всі встановлені майстром параметри зберігаються на вкладці *Подстановка* поряд з вкладкою *Общие* у нижній частині вікна конструктора (див. рис. 9). Для повторного завантаження майстра підстановок необхідно попередньо розірвати всі встановлені зв'язки з цим полем в схемі даних.

Література: [4, С. 94-95, 102; 1, С. 100-105]

### **Підготовчий етап заняття. Актуалізація знань**

1. Завантажте Access, відкрийте розроблену раніше БД *Sklad* (вона має міститися у вашій папці).
2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип объекта*, відобразіть у ній всі об'єкти (всі групи) та

перемістіться до розділу *Таблицы*.

### Створення та редагування властивостей таблиці в режимі конструктора. Встановлення структурних зв'язків між таблицями майстром підстановок

3. Створіть таблицю *Відділи* в режимі конструктора. Для цього:
  - 3.1. На вкладці стрічки меню *Создание* в групі *Таблицы* натисніть кнопку *Конструктор таблиць*;
  - 3.2. Задайте структуру та окремі властивості полів цієї таблиці згідно табл. 4 (перші дві властивості вводяться у верхній, а наступні дві – у нижній частині режиму конструктора таблиці після вибору типу даних);

Таблиця 4

**Структура та окремі властивості полів таблиці *Відділи***

Ім'я поля	Тип даних	Розмір поля	Підпис
КодВідділу	Счетчик	Длинное целое	
НазваВідділу	Текстовый	30	Назва відділу
МісцеРозташування	Текстовый	30	Місце розташування відділу

- 3.3. Задайте значення додаткових властивостей полів таблиці відповідно до табл. 5. Усвідомте призначення встановлених значень. Врахуйте, що ім'я поля обирається у верхній, а властивість міститься у нижній частині режиму конструктора таблиці. При відсутності властивості чи значення необхідно перевірити встановлений тип даних або віднайти їх синоніми. Наприклад, синонімічними є типи даних *Поле MEMO* та *Длинный текст* чи властивості *Условие на значение* та *Правило проверки*;

Таблиця 5

**Додаткові властивості полів таблиці *Відділи***

Ім'я поля	Властивість	Значення	Призначення
КодВідділу	Новые значения	Последовательный	Значення поля в нових записах збільшується послідовно
	Индексированное поле	Да (Совпадения не допускаются)	Забезпечує швидку обробку і унікальність
НазваВідділу	Условие на значение	Is Not Null	Значення поля після редагування має бути непорожнім
	Сообщение об ошибке	Введіть назву відділу	Текст повідомлення, яке з'являється, якщо значення поля не введено
	Индексированное поле	Да (Совпадения не допускаются)	

- 3.4. Встановіть властивість ключового поля для поля *КодВідділу*;
- 3.5. Закрийте режим конструктора таблиці зі збереженням змін та введіть її назву;



3.6. Відкрийте створену таблицю в режимі таблиці та введіть в неї дані п'яти відділів.

4. В області переходів відмітьте таблицю *Співробітники* та відкрийте її в режимі конструктора за допомогою контекстного меню.
5. Відредагуйте структуру та окремі властивості полів таблиці *Співробітники* згідно табл. 6. При цьому близькі за призначенням поля потрібно перейменувати, зайві – знищити, а відсутні – ввести, доповнюючи структуру таблиці новими рядками (для виділення рядка поля структури використовується прямокутник перед його ім'ям).

Таблиця 6

**Структура та окремі властивості полів таблиці *Співробітники***

Ім'я поля	Тип даних	Розмір поля	Підпис
КодСпівробітника	Счетчик	Длинное целое	
ПІБ	Текстовый	30	Прізвище, ім'я, по батькові
ДатаНародження	Дата/время		Дата народження
СеріяПаспорта	Текстовый	4	Серія паспорта
НомерПаспорта	Числовой	Длинное целое	Номер паспорта
КимКолиВидано Паспорт	Текстовый	40	Дата та місце видачі паспорта
ЧленствоВПрофспілці	Логический		Член профспілки?
Індекс	Числовой	Длинное целое	Індекс місця проживання
Адреса	Текстовый	40	Домашня адреса
Телефон	Текстовый	12	Домашній телефон
Email	Текстовый	30	Адреса електронної пошти
Фотокартка	Поле OLE		Фотокартка співробітника
ІПН	Текстовый	10	Ідивідуальний податковий №
Примітки	Поле МЕМО		Додаткові відомості
ДатаВлаштування	Дата/время		Дата влаштування
КодВідділу	Числовой	Длинное целое	Назва відділу
Посада	Текстовый	30	Посада співробітника
Оклад	Денежный		Посадовий оклад

Скільки байт і чому відводиться для зберігання полів *ДатаНародження*, *ЧленствоВПрофспілці*, *Фотокартка*, *Примітки*? Чому для поля *КодСпівробітника* не потрібно задавати підпис? Чому для поля *КодВідділу* вводиться підпис *Назва відділу*?

6. Введіть властивість *Описание* для полів таблиці, в яких підпис не характеризує повністю призначення поля.
7. Задайте значення додаткових властивостей полів таблиці *Співробітники* відповідно до табл. 7 (при формуванні значення по замовчуванню для поля *Телефон* між лапками вводиться 12 пробілів). Усвідомте призначення встановлених значень, адже однакові призначення в таблиці не повторюються, а аналогічне застосування значень буде вимагатися надалі.

Додаткові властивості полів таблиці *Співробітники*

Ім'я поля	Властивість	Значення	Призначення
КодСпівробітника	Новые значения	Последовательный	
	Индексированное поле	Да (совпадения не допускаются)	
ПІБ	Условие на значение	Is Not Null	
	Сообщение об ошибке	Введіть прізвище та ініціали співробітника	
	Индексированное поле	Да (допускаются совпадения)	
ДатаНародження	Формат поля	Краткий формат даты	Забезпечує відображення двох цифр дня, двох – місяця і чотирьох цифр року
	Маска ввода	00.00.0000	Задає шаблон для обов'язкового (0) введення компонентів дати аналогічно формату поля
	Условие на значение	Is Not Null	
	Сообщение об ошибке	Введіть дату народження співробітника	
СеріяПаспорта	Маска ввода	>&&CC	Забезпечує введення двох обов'язкових (&) і двох необов'язкових (C) символів у верхньому (>) регістрі
НомерПаспорта	Значение по умолчанию		Типового значення для номера паспорта немає, тому стандартне значення по замовчуванню для числових полів (0) необхідно стерти
	Условие на значение	>999	
	Сообщение об ошибке	Введіть номер паспорта (мінімум 4 цифри)	

Ім'я поля	Властивість	Значення	Призначення
КимКоли ВиданоПаспорт	Значение по умолчанию	Рівненським МВ УМВС України	
ЧленствоВ Профспілці	Формат поля	Да/Нет	
	Значение по умолчанию	Да	
Індекс	Маска ввода	900000	Задає шаблон для введення однієї необов'язкової (9) та п'яти обов'язкових цифр
	Значение по умолчанию	33000	
Телефон	Формат поля	!(@@@)@@@- @@-@@@	Відображає телефон з відділенням коду від номера, хоча у полі зберігаються лише цифри
	Маска ввода	\(9999\)999\-99\-999	Задає шаблон вводу, де всі цифри необов'язкові
	Значение по умолчанию	“”	
ІПН	Маска ввода	0000000000	
ДатаВлаштування	Формат поля	Краткий формат даты	
	Значение по умолчанию	Date()	У новому записі функція встановлює в поле системну дату
	Условие на значение	Is Not Null	
	Сообщение об ошибке	Введіть дату влаштування співробітника	
Оклад	Формат поля	С разделителями разрядов	Відображає оклад без тексту грошової одиниці
	Число десятичных знаков	2	Забезпечує відображення двох цифр після коми
	Значение по умолчанию	2400	

- Самостійно встановіть властивість ключового поля лише для поля *КодСпівробітника*.
- Здайте для поля *КодВідділу* цієї ж таблиці *Співробітники* підстановку значень поля *КодВідділу* з таблиці *Відділи* з відображенням відповідної назви відділу. Для цього:

- 9.1. Розгорніть список властивості *Тип даних* поля *КодВідділу* таблиці *Співробітники* та оберіть у ньому пункт *Мастер подстановок*;
- 9.2. На першому кроці майстра підстановок встановіть перемикач *Объект "поле подстановки" получит значения из другой таблицы или другого запроса* та натисніть кнопку *Далее*;
- 9.3. На другому кроці вкажіть, що підстанова виконується з таблиці *Відділи* та знову натисніть кнопку *Далее*;
- 9.4. На третьому кроці з переліку доступних перенесіть в перелік обраних полів з допомогою кнопок '>', '<', '>>', '<<' наступні поля таблиці *Відділи*:
  - 9.4.1. *КодВідділу* – для виконання підстановки;
  - 9.4.2. *НазваВідділу* – для відображення в полі підстановки.
- 9.5. На четвертому кроці забезпечте сортування відображуваних елементів списку за зростанням значень поля *НазваВідділу*;
- 9.6. На п'ятому кроці переконайтеся в тому, що значення перемикача *Скрыть ключевой столбец* встановлено;
- 9.7. На останньому кроці майстра підстановок задайте для створеного стовпця ім'я *КодВідділу*. Інші прапорці не встановлюйте та натисніть кнопку *Готово*. Врахуйте, що для поля *КодВідділу* введено підпис *Назва відділу*, оскільки при здійсненні підстановки користувач буде бачити саме назву відділу, хоча в полі буде зберігатися її код (що відповідає імені поля), але про це має знати лише розробник БД. Як і де можливо змінити встановлені параметри підстановки?
- 9.8. Перейдіть в режим редагування даних таблиці безпосередньо з режиму конструктора (як?), підтвердивши при цьому збереження встановлених параметрів таблиці. Відкоригуйте і доповніть дані внесених співробітників та завершіть редагування цієї таблиці.
10. Змініть структуру таблиці *Товари* згідно табл. 8, перейменовуючи близькі за призначенням поля (нагадаємо, що в поле *ДСТУ* вноситься номер Державного стандарту України (при його наявності), якому відповідає товар).

Таблица 8

**Структура таблиці *Товари***

Ім'я поля	Тип даних
КодТовару	Счетчик
НазваТовару	Текстовый
МаркаТовару	Текстовый
ОдиницяВиміру	Числовой
Придатно	Числовой
Виробник	Текстовый
ДСТУ	Текстовый

11. Встановіть для визначеного вами поля властивість ключового поля.
12. Оформіть для редагованих полів властивості *Подпись* та *Описание*.

13. Задайте додаткові властивості полів, забезпечивши:

- 13.1. Обов'язкове введення назви товару;
- 13.2. Можливість швидкого сортування та пошуку згідно назви товару;
- 13.3. Стівосьмидесятиденний термін придатності по замовчуванню;
- 13.4. Обов'язкове введення назви фірми–виробника товару.

14. Задайте для поля *ОдиницяВиміру* підстановку одного з можливих фіксованих значень. Для цього:

- 14.1. Використайте властивість *Тип данных* цього поля, обравши в її списку пункт *Мастер подстановок*;
- 14.2. На першому кроці майстра підстановок оберіть пункт *Будет введен фиксированный набор значений* та натисніть кнопку *Далее*;
- 14.3. На другому кроці **встановіть два стовпці** таблиці підстановки та задайте значення її стовпців, як на рис. 10.

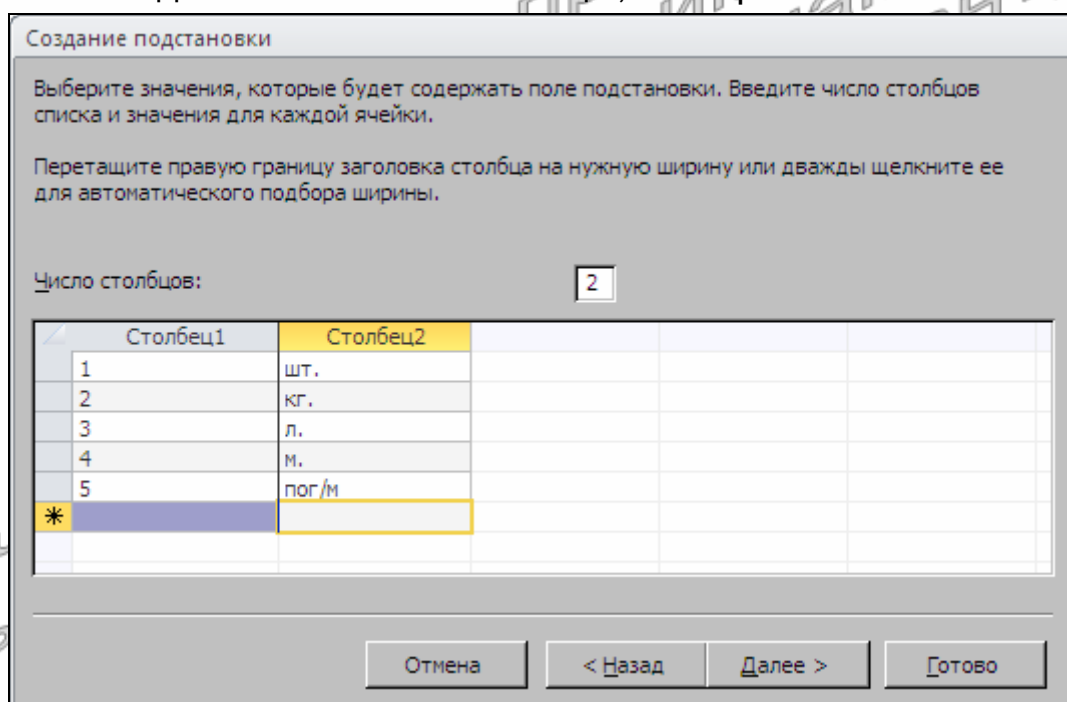


Рис. 10. Другий крок майстра підстановок з фіксованого набору значень для поля *ОдиницяВиміру*

- 14.4. Встановіть нульову ширину першого стовпця, затягнувши межу у заголовку між стовпцями за лівий край таблиці підстановки та натисніть кнопку *Далее*;
- 14.5. На третьому кроці вкажіть, що значення для підстановки містяться в першому стовпці (*Col1*) сформованого списку;
- 14.6. На останньому кроці майстра підстановок задайте для створеного стовпця підпис *ОдиницяВиміру*, інші прапорці не встановлюйте та натисніть кнопку *Готово*;
- 14.7. Підтвердіть збереження встановлених параметрів таблиці;

- 14.8. Встановіть для поля *ОдиницяВиміру* значення по замовчуванню 1. Яке значення по замовчуванню **відображається** при заповненні поля? Що при цьому **зберігається** в полі? Чому в числовому полі відображаються текстові величини? Як доповнити фіксований набір значень списку підстановки? Як змінити інші параметри цієї підстановки?
15. Задайте додаткові властивості полів створеної раніше таблиці *Клієнти*, забезпечивши:
- 15.1. Обов'язкове введення ініціалів клієнта;
- 15.2. Встановлення активної дати по замовчуванню в полі *ДатаРеєстрації* при введенні даних нового клієнта.
16. Самостійно задайте та обґрунтуйте інші доцільні значення властивостей полів таблиці *Клієнти*.
17. Створіть в режимі конструктора таблицю *Постачальники* з структурою згідно табл. 9 (в поле *КодЄДРПОУ* вноситься восьмизначний код Єдиного державного реєстру підприємств, організацій, установ відповідної організації).

Таблиця 9

**Структура таблиці *Постачальники***

Ім'я поля	Тип даних
КодПостачальника	Счетчик
НазваПостачальника	Текстовый
Адреса	Текстовый
КодЄДРПОУ	Текстовый
ДатаРеєстрації	Дата/время
Характеристика	Поле МЕМО

18. Встановіть для визначеного вами поля властивість ключового поля.
19. Оформіть для редагованих полів властивості *Подпись* та *Описание*.
20. Задайте додаткові властивості полів таблиці, забезпечивши:
- 20.1. Обов'язкове введення назви постачальника;
- 20.2. Можливість швидкого сортування та пошуку згідно назви постачальника;
- 20.3. Введення в полі *КодЄДРПОУ* восьми обов'язкових цифр;
- 20.4. Встановлення активної дати по замовчуванню в полі *ДатаРеєстрації* при введенні даних нового постачальника.
21. Завершіть формування структури таблиці, перейдіть в режим таблиці та внесіть в неї п'ять записів.
22. Створіть в режимі конструктора таблицю *ЗаголовкиПостачань* з структурою відповідно до табл. 10.

Таблиця 10

**Структура та типи даних полів таблиці *ЗаголовкиПостачань***

Ім'я поля	Тип даних
КодПостачання	Счетчик
КодСпівробітника	Числовой
КодПостачаньника	Числовой
ДатаПостачання	Дата/время



23. Забезпечте підстановку поля *КодСпівробітника* з таблиці *Співробітники*. Для цього:
- 23.1. Розгорніть список властивості *Тип даних* поля *КодСпівробітника* з таблиці *ЗаголовкиПостачань* та оберіть у ньому пункт *Мастер подстановок*;
  - 23.2. На першому кроці майстра підстановок виберіть пункт *Объект "поле подстановки" получит значения из другой таблицы или другого запроса* та натисніть кнопку *Далее*;
  - 23.3. На другому кроці вкажіть, що підстановка виконується з таблиці *Співробітники* та знову натисніть кнопку *Далее*;
  - 23.4. На третьому кроці з переліку доступних в перелік обраних полів з допомогою кнопок '>', '<', '>>', '<<' перенесіть наступні поля таблиці *Співробітники*:
    - 23.4.1. *КодСпівробітника* – для виконання підстановки;
    - 23.4.2. *ПІБ* – для відображення в полі підстановки;
    - 23.4.3. *Телефон* – для додаткового відображення при розгортанні списку підстановки;
  - 23.5. На четвертому кроці майстра забезпечте сортування відображуваних елементів списку за зростанням значень поля *ПІБ*;
  - 23.6. На п'ятому кроці переконайтеся в тому, що значення перемикача *Скрыть ключевой столбец* встановлено;
  - 23.7. На останньому кроці майстра підстановок задайте для створеного стовпця назву *КодСпівробітника* та натисніть кнопку *Готово*.
24. Для поля *КодПостачальника* цієї ж таблиці *ЗаголовкиПостачань* самостійно задайте підстановку значень з поля *КодПостачальника* таблиці *Постачальники*, забезпечивши при цьому відображення відповідного значення поля *НазваПостачальника*. Яку властивість поля слід для цього змінити? Які стовпці зберігаються, а які – відображаються при використанні підстановки? Де беруться дані для цих полів?
25. Встановіть для поля *КодПостачання* властивість ключового поля.
26. Тут і в наступних трьох таблицях оформіть для редагованих полів властивості *Подпись* та *Описание*. Врахуйте, що властивість *Подпись* має містити текст, який пояснює відображені дані користувачу. Тому, наприклад, для поля *ДатаПостачання* у цій властивості потрібно встановити значення *Дата постачання*. Для поля *КодПостачальника* необхідно встановити підпис *Назва постачальника*, оскільки у списку підстановки користувач буде бачити саме назву постачальника, хоча у полі буде зберігатися його код. Запам'ятайте: ім'я поля відповідає значенню, яке зберігається у ньому, а підпис – суті, яка в цьому полі відображається.
27. Задайте додаткові властивості полів таблиці, забезпечивши:

- 27.1. Обов'язковий вибір даних співробітника і постачальника та введення дати постачання;
- 27.2. Встановлення активної дати по замовчуванню в полі *ДатаПостачання* при введенні даних нового постачання.
28. Створіть в режимі конструктора таблицю *ПунктиПостачань* з структурою згідно табл. 11.

Таблиця 11

**Структура та типи даних полів таблиці *ПунктиПостачань***

Ім'я поля	Тип даних
КодПункту	Счетчик
КодПостачання	Числовой
КодТовару	Числовой
Кількість	Числовой
Ціна	Денежный

29. Забезпечте для поля *КодТовару* таблиці *ПунктиПостачань* підстановку значень з поля *КодТовару* таблиці *Товари*, відображаючи в полі зі списком відповідне значення поля *НазваТовару*, а при розгортанні списку – ще й значення поля *МаркаТовару*. Чому для поля *КодПостачання* не потрібно створювати підстановку з таблиці *ЗаголовкиПостачань*?
30. Встановіть для обраного вами поля властивість ключового поля.
31. Задайте додаткові властивості полів таблиці, забезпечивши:
- 31.1. Обов'язковий вибір даних отриманого товару;
- 31.2. Встановлення для поля *Кількість* значення по замовчуванню рівним 1;
- 31.3. Відображення ціни з двома знаками після коми без назви грошової одиниці;
- 31.4. Неможливості введення нульової кількості (для цього у властивості *Условие на значение* потрібно вказати  $<>0$ ) чи від'ємної ціни (значення властивості *Условие на значение* задається у вигляді  $>=0$ ) товару з відповідними повідомленнями при помилковому введенні значень цих полів.
32. Створіть в режимі конструктора таблицю *ЗаголовкиЗамовлень* з структурою відповідно до табл. 12.

Таблиця 12

**Структура та типи даних полів таблиці *ЗаголовкиЗамовлень***

Ім'я поля	Тип даних
КодЗамовлення	Счетчик
КодСпівробітника	Числовой
КодКлієнта	Числовой
ДатаЗамовлення	Дата/время

33. Забезпечте підстановку поля *КодСпівробітника* з таблиці *Співробітники*, поля *КодКлієнта* з таблиці *Клієнти*, відображаючи при цьому значення відповідно поля *ПІБ* та *НазваКлієнта*.
34. Встановіть для обраного вами поля властивість ключового поля.

35. Задайте додаткові властивості полів таблиці, забезпечивши:
- 35.1. Обов'язковий вибір даних співробітника і клієнта та введення дати замовлення;
  - 35.2. Встановлення активної дати по замовчуванню в полі *ДатаЗамовлення* при введенні даних нового замовлення.
36. Створіть в режимі конструктора таблицю *ПунктиЗамовлень* з структурою згідно табл. 13.

Таблиця 13

**Структура та типи даних полів таблиці *ПунктиЗамовлень***

Ім'я поля	Тип даних
КодПункту	Счетчик
КодЗамовлення	Числовой
КодТовару	Числовой
Кількість	Числовой
Ціна	Денежный

37. Забезпечте підстановку поля *КодТовару* з таблиці *Товари*, відображаючи в полі зі списком значення поля *НазваТовару*, а при розгортанні списку – ще й значення поля *МаркаТовару*. Чому для поля *КодЗамовлення* не потрібно створювати підстановку з таблиці *ЗаголовкиЗамовлень*?
38. Встановіть для обраного вами поля властивість ключового поля.
39. Задайте додаткові властивості полів таблиці, забезпечивши:
- 39.1. Обов'язковий вибір даних замовленого товару;
  - 39.2. Встановлення по замовчуванню значення поля *Кількість* рівним 1;
  - 39.3. Відображення ціни з двома знаками після коми без назви грошової одиниці;
  - 39.4. Неможливості введення нульової кількості чи від'ємної ціни товару з відповідними повідомленнями при помилковому введенні значень цих полів.

**Встановлення структурних зв'язків між таблицями в схемі даних**

40. Для забезпечення можливості блокування таблиць при встановленні структурних зв'язків та їх цілісності закрийте всі відкриті вікна чи вкладки таблиць.
41. Відкрийте вікно відображення схеми даних, натиснувши на стрічці меню у вкладці *Работа с базами данных* кнопку *Схема данных*.
42. У вікні *Добавление таблицы*, що автоматично з'являється при першому відкритті схеми даних чи відкривається відповідним пунктом контекстного меню вільного поля вікна схеми даних, відмітьте назви таблиць, які ще не відображені в схемі (як на рис. 11), натисніть кнопку *Добавить* та закрийте це вікно. Коли і як створилися вже наявні зв'язки між таблицями?
43. Для виведення образів таблиць, що ще не відобразилися в схемі даних, знову відкрийте вікно *Добавление таблицы* з контекстного

меню вільної області схеми даних та виконайте дії попереднього пункту.

44. Дублюючи зображення таблиць в схемі даних знищіть за допомогою їх виділення та натиснення клавіші *Del*, **попередньо знищивши встановлені з ними зв'язки**.

45. Забезпечте цілісність даних кожного встановленого зв'язку. Для цього послідовно:

45.1. Виділіть окремий зв'язок між таблицями;

45.2. Оберіть в контекстному меню зв'язка пункт *Изменить связь...*;

45.3. У вікні *Связи* встановіть відповідні прапорці для забезпечення цілісності даних та каскадного оновлення (як на рис. 12);

45.4. Збережіть внесені зміни характеру зв'язку, натиснувши кнопку *OK*.

46. Якщо для якогось зв'язка забезпечити цілісність посилань не вдається, то:

46.1. Закрийте схему даних зі збереженням внесених змін;

46.2. Відкрийте таблицю в режимі таблиці на стороні зв'язка 1, запам'ятайте значення, які у ній зустрічаються у полі зв'язка (первинному ключі) та закрийте цю таблицю;

46.3. Відкрийте таблицю на стороні зв'язка ∞, перевірте, а при необхідності – встановіть чи виберіть у полі зв'язка (вторинному ключі) лише такі значення, які зустрічалися на стороні 1;

46.4. Закрийте таблицю на стороні зв'язка ∞, поверніться у схему даних та спробуйте забезпечити цілісність посилань знову.

47. Розмістіть таблиці згідно схеми рис. 11.

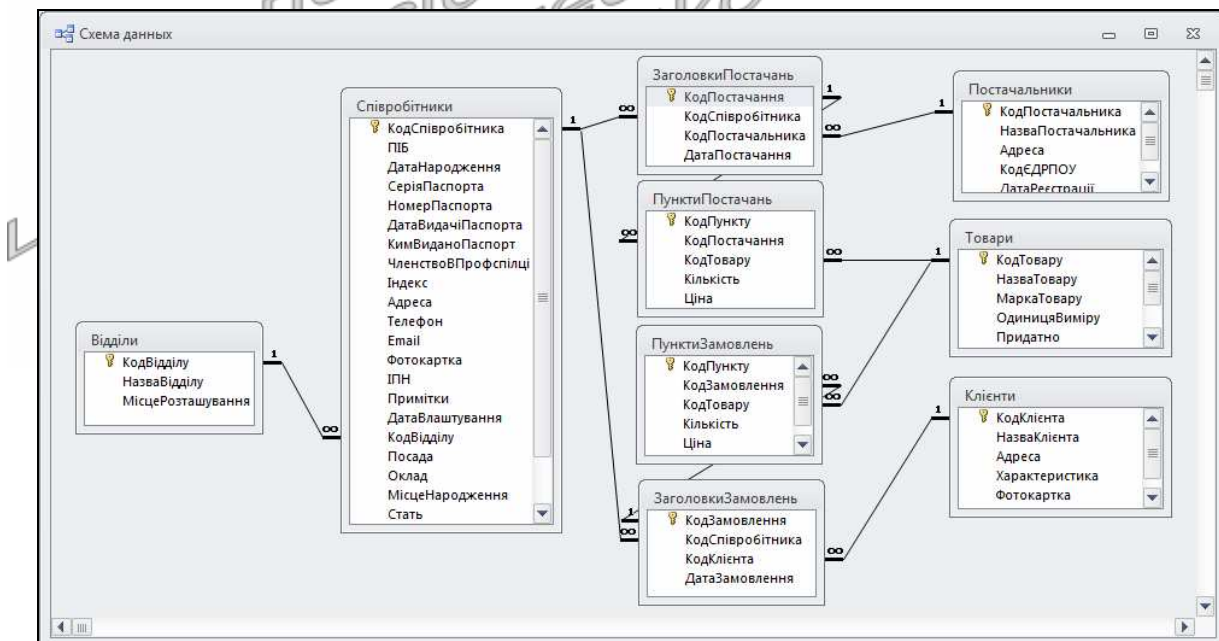


Рис. 11. Схема даних БД гуртового складу на даталогічному рівні в MS Access

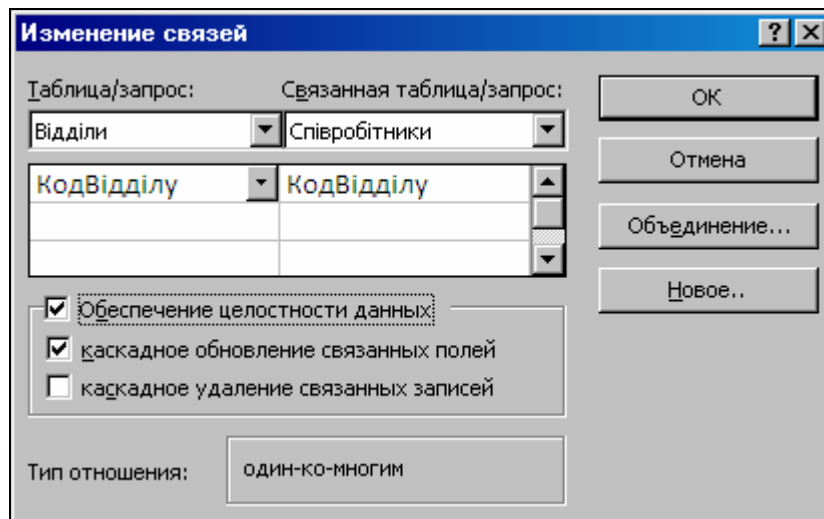


Рис. 12. Забезпечення цілісності даних між таблицями *Відділи* та *Співробітники*

48. Встановіть відсутні зв'язки між таблицями, виконуючи наступні дії:
- 48.1. Відмітьте поле першої таблиці, що входить до структурного зв'язку;
  - 48.2. Не відпускаючи лівої кнопки мишки, перемістіть її покажчик до відповідного поля другої таблиці зв'язку;
  - 48.3. Відпустіть ліву кнопку мишки. У вікні *Связи*, що з'явиться на екрані, забезпечте цілісність посилань аналогічно попередньому пункту з додатковим встановленням каскадного знищення лише між таблицями заголовків і пунктів постачань та замовлень (Чому?);
49. Закрийте вікно схеми даних та підтвердіть внесення відповідних змін.

#### **Завершальний етап заняття. Повторення вивченого матеріалу**

50. Стисніть створену БД.
51. Завершіть роботу Access.
52. Віднайдіть створену БД та заархівуйте її вміст.
53. Збережіть створений архів на власному носії в двох екземплярах.
54. Розробіть таблиці власної БД у відповідності з проектами інфологічного і даталогічного рівня лабораторної роботи № 1. Обгрунтуйте спосіб задання (майстром підстановок чи в схемі даних) та встановіть структурні зв'язки між таблицями. Продемонструйте розроблені БД викладачу.

#### **Контрольні запитання**

1. Як можна створити таблицю в режимі конструктора?
2. Яких вимог слід дотримуватися при введенні назви поля?
3. Яка різниця між властивостями полів *Имя поля*, *Подпись* та *Описание*?
4. Які властивості поля задаються в режимі конструктора? Яка роль та призначення кожної властивості?

5. Коли для поля встановлюється властивість підстановки значень з іншої таблиці, а коли – з фіксованого набору значень?
6. Які властивості мають співпадати для полів первинного і вторинного ключа? Чому?
7. Для яких цілей в майстрах Access використовуються кнопки '>' та '>>'. Яка різниця між їх діями? Яка різниця між діями кнопок '>' та '<', '>>' та '<<'?
8. Які властивості підстановки задаються в режимі конструктора? Де вони описуються?
9. Значення якого стовпця підставляється при використанні підстановки по замовчуванню? Де вказується номер приєднаного стовпця?
10. Значення якого стовпця відображається при використанні підстановки? Як відобразити в стовпці підстановки значення іншого стовпця?
11. Що відображається в схемі даних?
12. Які зв'язки встановлюються автоматично в схемі даних?
13. Які зв'язки між таблицями слід встановлювати самостійно в схемі даних?
14. Як забезпечується цілісність даних з використанням структурних зв'язків між таблицями в БД Access? Що вона означає?



## Лабораторна робота № 5

**Тема.** Редагування, сортування, індексування, пошук, фільтрування та копіювання даних таблиць. Експорт та імпорт таблиць. Встановлення зв'язків з таблицями інших БД.

**Мета.** Формування вмінь та навичок обробки даних таблиць, експорту та імпорту даних між таблицями та БД, сумісного використання таблиць БД. Вдосконалення вмінь та навичок обробки БД. Застосування на практиці знань структури ієрархічної моделі зв'язків між даними при створенні індексів таблиць.

### Теоретичні відомості

Завантажувати дані в таблиці Access можна не лише шляхом безпосереднього введення чи за допомогою буфера обміну, а й використовуючи операції імпорту (перенесення з зовнішнього джерела в БД) чи експорту (перенесення з БД в зовнішнє джерело) даних (рис. 13). Причому обмін даними можливий не тільки з електронними таблицями чи файлами інших СУБД, а й з текстовими файлами. Головне, щоб дані були подані у табличній формі. Типові дії для організації експорту/імпорту даних можна зберегти в БД, дати цим діям назву (наприклад, *Резервне копіювання таблиць БД гуртового складу*) і надалі, використовуючи кнопки *Сохраненные операции импорта /экспорта* (див. рис. 13) та дану назву, виконувати ці дії багаторазово.

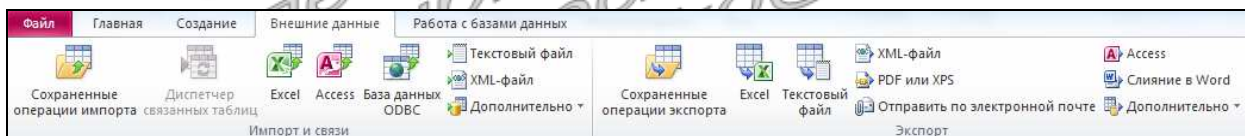



Рис. 13 Основні кнопки вкладки *Внешние данные* на стрічці меню

При перенесенні даних слід пам'ятати, що перед зберіганням Access перевіряє цілісність не лише відкоригованих полів, а й цілого запису. Тому, завантажуючи дані в нові записи, слід забезпечити не лише однакові типи даних і розміри полів-джерел і полів-приймачів даних, а й заповнення **всіх** обов'язкових полів таблиці.

Крім згаданих операцій з зовнішніми даними, Access має також засоби для встановлення зв'язків з таблицями інших БД. **Зв'язок з таблицею** – це посилання на таблицю, найчастіше – з іншої БД, яке відображається на вкладці *Таблицы* (наприклад, ). Його аналогом є ярлик в операційній системі. Обробка даних пов'язаної таблиці в режимі таблиці нічим не відрізняється від опрацювання таблиці, розміщеної в БД. А ось в режимі конструктора структуру пов'язаної таблиці можна лише переглядати. Змінити ж її можливо лише в базі-джерелі. Зрозуміло, що для використання пов'язаної

таблиці необхідно, щоб база-джерело містилася за тим самим шляхом та, як і таблиця, мала те саме ім'я, що й в момент створення зв'язка. Зв'язки з таблицями встановлюють, коли потрібно забезпечити одночасну обробку даних багатьма користувачами. Для архівного ж зберігання даних використовують імпорт таблиць.

Розглянемо основні операції над даними записів таблиць:

- **пошук** – це переміщення на запис та виділення в ньому даних поля, що задовольняє заданим критеріям. Вікно параметрів пошуку Access майже аналогічне вікнам цього процесу в інших додатках (рис. 14), але, завдяки структуризації даних, виконувати цей процес можна як по всіх полях, так і в активному полі (параметр *Поиск в*), вимагаючи співпадіння з фрагментом, цілим полем або його початком;

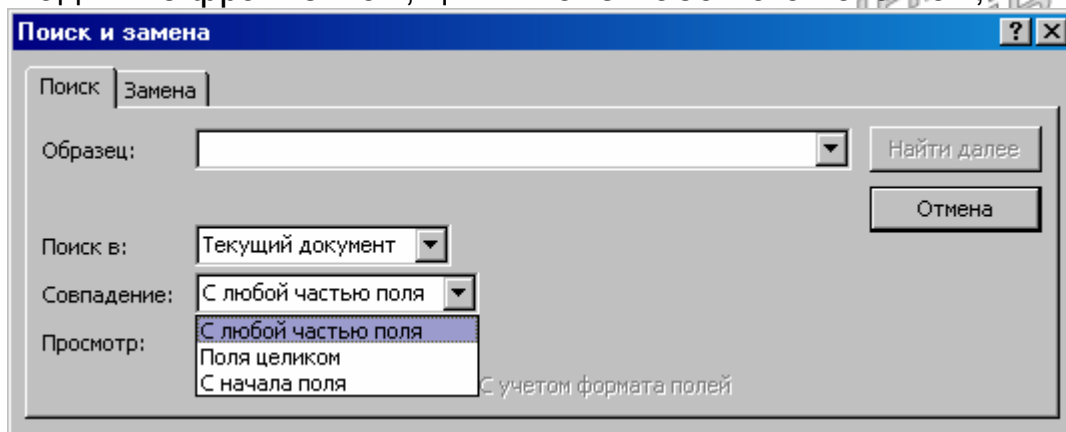


Рис. 14. Вікно задання параметрів пошуку

- **сортування** – це впорядкування записів таблиці (а не окремих полів) за зростанням (спаданням) значення одного чи декількох полів. Основним при виконанні сортування є значення першого (найлівішого) виділеного чи активного поля. Саме за ним в основному виконується впорядкування записів. Значення наступних полів сортування впливає на впорядкування між собою лише для тих записів, в яких значення попередніх полів сортування однакові (записи 3-4 та 7-9 на рис. 15).

Марка товару	Назва товару
Бахмач	згущене молоко 'Бахмач'
Геснерія	сік 'Карапуз'
ЕКО продукт	цукерки 'Зоряне сяйво'
ЕКО продукт	цукерки 'Столичні'
Київське морози	морозиво 'Пломбір'
Львівська кава	кава 'Галка'
Світоч	цукерки 'Білочка'
Світоч	цукерки 'Ромашка'
Світоч	цукерки 'Червоний мак'
Торчин продукт	майонез 'Делікатесний'

Рис. 15. Результати сортування за зростанням значень полів *МаркаТовару* і *НазваТовару*

Наголосимо, що при сортуванні записи фізично не переміщуються, а змінюється лише їх порядок відображення для користувача;

- **фільтрування** – це відбір записів, що відповідають встановленим критеріям. При цьому решта записів (на відміну від операції пошуку) не відображаються на екрані. Для повторного відображення всіх записів виконують видалення фільтру. В таблицях Access використовуються чотири типи фільтрів: за значеннями стовпця, за виділеним фрагментом поля стовпця, звичайний фільтр (його вікно має структуру запису, де можливо задати умови відбору для кожного поля) та розширений фільтр, що має структуру, аналогічну запиту.

Для прискорення пошуку, сортування і фільтрування даних в таблицях створюють ієрархічні індекси не лише за значеннями окремих полів (як це здійснює властивість поля таблиці *Индексированное поле*), а й за їх сукупностями. Формуються такі індекси у режимі конструктора таблиць в одноіменному вікні. Це вікно використовують також, коли необхідно знищити індекси з однаковими полями, але різними назвами. Необхідно пам'ятати, що створення кожного індекса збільшує розмір БД та час реакції системи, тому формувати їх доцільно лише для полів, за якими найчастіше виконується обробка записів.

Література: [4, С. 105-107; 1, С. 125-126]

### **Підготовчий етап заняття. Актуалізація знань**

1. Перевірте наявність БД *Sklad* у власній папці. При відсутності БД віднайдіть на власному носії збережений архів *Sklad* та розархівуйте його вміст у власну папку.
2. Віднайдіть на жорсткому диску БД *Борей* (*Nwind*). При відсутності цієї БД на жорсткому диску самостійно встановіть її на комп'ютері використовуючи шаблон *Файл ► Создать ► Образцы шаблонов... ► "Борей"* (без позначення глобуса). Скопіюйте цю БД у власну папку.
3. Переіменуйте файл скопійованої БД *Борей.accdb* (*Nwind.accdb*) на *Zrazok.accdb*.
4. Завантажте Access, відкрийте розроблену раніше БД *Sklad*.
5. Перейдіть на вкладку *Таблицы*.

### **Імпорт та експорт таблиць. Встановлення зв'язків з таблицями інших БД**

6. Імпортуйте у власну БД таблицю *Сотрудники* з БД *Zrazok.accdb*.  
Для цього:
  - 6.1. Перейдіть на вкладку меню *Внешние данные* та натисніть на ній кнопку *Импорт и связи ► Access*;
  - 6.2. Оберіть у виведеному вікні пункт *Импорт таблиц, запросов, форм, отчетов макросов и модулей в текущую базу данных*;

- 6.3. Вкажіть БД для імпорту об'єктів: у вікні *Импорт базы данных Access* перейдіть у власну папку, відмітьте у ній БД *Zrazok.accdb* та натисніть кнопку *Импорт*;
- 6.4. Оберіть імпортований об'єкт: на вкладці *Таблицы* БД *Zrazok.accdb* відмітьте таблицю *Сотрудники*;
- 6.5. На тому ж кроці майстра оберіть спосіб імпортування таблиці разом з даними: натисніть кнопку *Параметры >>* для відображення додаткових параметрів імпорту та в групі перемикачів *Импорт таблиц* встановіть перемикач *структура и данные*;
- 6.6. Імпортуйте обраний об'єкт, натиснувши кнопку *ОК*.
7. Самостійно імпортуйте з БД *Zrazok.accdb* структуру та дані таблиці *Клиенты*.
8. У власній БД *Sklad* встановіть зв'язок з таблицею *Товары* БД *Zrazok.accdb*. Для цього:
  - 8.1. Перейдіть на вкладку меню *Внешние данные* та натисніть на ній кнопку *Импорт и связи* ► *Access*;
  - 8.2. Оберіть пункт *Создать связанную таблицу для связи с источником данных*;
  - 8.3. Вкажіть БД для встановлення зв'язку: у вікні *Импорт базы данных Access* перейдіть у власну папку, відмітьте у ній БД *Zrazok.accdb* та натисніть кнопку *Импорт*;
  - 8.4. Оберіть таблицю та встановіть з нею зв'язок: на вкладці *Таблицы* БД *Zrazok.accdb* відмітьте таблицю *Товары* і натисніть *ОК*.
9. Відкрийте пов'язану таблицю в режимі конструктора. Поясніть зміст повідомлень системи.
10. Експортуйте дані з таблиці *Товары* в робочий лист книги Excel. Для цього:
  - 10.1. Завантажте Excel та створіть порожню книгу;
  - 10.2. Збережіть файл створеної книги у власній папці під назвою *Export.xlsx* та завершіть роботу Excel;
  - 10.3. У власній БД відмітьте пов'язану таблицю *Товары*;
  - 10.4. Для виклику майстра експорту даних натисніть кнопку головного меню *Внешние данные* ► *Экспорт* ► *Excel*;
  - 10.5. З метою експорту даних у файл Excel, у вікні параметрів майстра встановіть формат файла *Книга Excel*;
  - 10.6. У цьому ж вікні, використовуючи кнопку *Обзор*, відмітьте створений файл *Export.xlsx* у власній папці та натисніть *ОК*.
11. Завантажте Excel та відкрийте для редагування файл *Export.xlsx*.
12. Віднайдіть у цьому файлі експортовані дані (зверніть увагу на назви листів) та створіть новий стовпець *Сума* для підрахунку загальної вартості збережених товарів. Для формування значень стовпця введіть в одну з його комірок відповідну формулу та поширте її на

всі рядки таблиці (загальна вартість запасу товару рівна добутку ціни на кількість товару на складі). Перевірте коректність значень сформованого стовпця та закрийте Excel.

13. Імпортуйте дані з книги *Export.xlsx* у власну БД. Для цього:
  - 13.1. Оберіть пункт стрічки меню *Внешние данные* ► *Импорт и связи* ► *Excel*;
  - 13.2. Для імпорту даних з файла Excel виберіть пункт *Импортировать данные источника в новую таблицу в текущей базе данных*;
  - 13.3. Відмітьте (кнопка *Обзор*) файл *Export.xlsx* у власній папці та натисніть кнопку *OK*;
  - 13.4. У вікні першого кроку майстра імпорту даних відмітьте лист *Товары* для імпорту даних та натисніть кнопку *Далее*;
  - 13.5. Встановіть прапорець *Первая строка содержит заголовки столбцов* та натисніть кнопку *Далее*;
  - 13.6. Виділяючи послідовно кожен стовпець в нижній частині вікна майстра, перевірте та відкоригуйте властивості всіх полів імпортованої таблиці і натисніть кнопку *Далее*;
  - 13.7. Встановіть для нової таблиці ключове поле *ID* та натисніть кнопку *Далее*;
  - 13.8. Дайте новій таблиці назву *Товари1* та натисніть кнопку *Готово*.
14. Самостійно імпортуйте дані з файла *Export.xlsx* у власну БД для доповнення таблиці *Товари*. Чи вдалося імпортувати дані? Чому?
15. Самостійно імпортуйте дані з файла *Export.xlsx* у власну БД для доповнення таблиці *Товари1*. Чи вдалося імпортувати дані? Чому?


### **Редагування даних таблиць**

16. Доповніть імпортовану таблицю *Сотрудники* в режимі конструктора полем *ІПН* для зберігання індивідуального податкового номера співробітника. Забезпечте для даного поля обов'язкове введення 10 цифр. Помістіть це поле в кінці таблиці.
17. Перейдіть в режим таблиці при допомозі кнопки *Режим* та створіть умови для зручного введення поля *ІПН*:
  - 17.1. Приховайте стовпець *КодСотрудника*. Для цього:
    - 17.1.1. Виділіть відповідний стовпець, відмітивши його заголовком натисненням лівої кнопки мишки;
    - 17.1.2. В контекстному меню стовпця оберіть пункт *Скрыть поля*;
  - 17.2. Закріпіть стовпці *Фамилия* та *Имя* для відображення цих даних в будь-якій активній позиції таблиці. Для цього:
    - 17.2.1. Виділіть відповідні стовпці при натиснутій клавіші *Shift*;
    - 17.2.2. В контекстному меню виділених стовпців оберіть пункт *Закрепить поля*;



- 17.3. Розташуйте стовпець *ІПН* перед стовпцем *Вложення*. Для цього:
- 17.3.1. Виділіть стовпець *ІПН*, відмітивши його заголовком;
  - 17.3.2. Перетягніть стовпець за заголовок вліво в потрібну позицію при натиснутій лівій кнопці мишки.
18. Заповніть поле *ІПН* для всіх співробітників.
19. Дайте українські назви стовпцям таблиці. Стовпцю *Індекс* дайте назву *Номер паспорта*. При цьому для перейменування стовпців використовуйте три способи:
- 19.1. Відповідний пункт контекстного меню стовпця;
  - 19.2. Подвійне натиснення лівої кнопки мишки на назві стовпця;
  - 19.3. Пункт меню *Поля* ► *Імя и подпись*.
20. Перегляньте в режимі конструктора структуру таблиці *Сотрудники*. Як відобразилися в структурі таблиці проведені маніпуляції з стовпцями в режимі таблиці?
21. Поверніться в режим таблиці та відобразіть приховані стовпці. Для цього:
- 21.1. Виділіть правою кнопкою мишки заголовком першого стовпця. В його контекстному меню виберіть пункт *Отображение столбцов*;
  - 21.2. Встановіть всі прапорці у виведеному вікні;
  - 21.3. Натисніть кнопку *Закрывать*.
22. Відмініть закріплення всіх стовпців. Для цього в контекстному меню заголовка першого стовпця оберіть пункт *Отменить закрепление всех столбцов*.

### Сортування даних таблиць

23. Відсортуйте записи таблиці *Сотрудники* за зростанням значення поля *Прізвище*. Для цього:
- 23.1. Виділіть стовпець *Прізвище*;
  - 23.2. У стрічці меню на вкладці *Главная* натисніть кнопку  чи оберіть відповідний пункт контекстного меню стовпця.
24. Відсортуйте записи таблиці за зростанням значення поля *Посада*. Дані співробітників, що обіймають однакові посади, відсортуйте за зростанням значення поля *Прізвище*. Для цього:
- 24.1. Розташуйте стовпець *Посада* перед стовпцем *Прізвище*;
  - 24.2. Виділіть стовпці *Посада* та *Прізвище*;
  - 24.3. Відсортуйте виділені дані аналогічно попередньому пункту.
25. Самостійно відсортуйте дані таблиці за сукупним зростанням значень полів *ЗвертатисяДо* та *Ім'я*.




### Пошук даних таблиць

26. Відкрийте в режимі таблиці таблицю *Клиенты*. Віднайдіть першого клієнта, що обіймає посаду бухгалтера. Для організації пошуку перемістіться в поле *Должность* та використовуйте кнопку



- Главная* ► *Найти*. Віднайдіть всіх клієнтів, що обіймають цю посаду.
27. Відшукайте клієнтів, в даних яких міститься буквосполучення *ко*. Для цього у вікні задання параметрів пошуку в полі зі списком *Совпадение* оберіть значення *С любой частью поля*, а у полі зі списком *Поиск в – Текущий документ*. Самостійно введіть текст та здійсніть пошук.

### Фільтрування даних таблиць


28. Серед всіх клієнтів виберіть співробітників відділу постачання. Для цього:
- 28.1. Віднайдіть та виділіть назву посади *Сотрудник отдела снабжения*;
- 28.2. В контекстному меню виділеної комірки оберіть пункт *Равно «Сотрудник отдела снабжения»*.
29. Для відображення всіх записів розгорніть список фільтрації у правому кутку заголовка відфільтрованого поля  та зніміть прапорець з пункту контекстного меню *Сотрудник отдела снабжения* або на стрічці меню відтисніть кнопку фільтрування .
30. Серед всіх клієнтів відберіть тих, що не обіймають посаду *Сотрудник отдела снабжения*. Для цього:
- 30.1. Віднайдіть та виділіть посаду *Сотрудник отдела снабжения*;
- 30.2. В контекстному меню виділеної комірки оберіть пункт *Не равно «Сотрудник отдела снабжения»*.
31. Самостійно відберіть:
- 31.1. Клієнтів з іменем *Олександр*, які проживають в будинку № 4;
- 31.2. Бухгалтерів з іменами, що містять букву *а*.
32. Відберіть клієнтів, що обіймають посаду *співробітник відділу постачання*, для яких не введено адресу електронної пошти. Для цього:
- 32.1. Введіть декільком клієнтам правдоподібні адреси електронної пошти;
- 32.2. У групі *Главная* ► *Сортировка и фильтр* розгорніть список кнопки  та оберіть у ньому пункт *Изменить фильтр* для переходу у вікно задання фільтру таблиці;
- 32.3. В полі *Должность* цього вікна оберіть зі списку значення *Сотрудник отдела снабжения*;
- 32.4. В поле *Электронная почта* введіть *Is Null*;
- 32.5. Для відбору записів таблиці згідно сформованого фільтру оберіть в його контекстному меню пункт *Применить фильтр* або на стрічці меню натисніть кнопку .
33. Самостійно відберіть дані керівників відділу постачання без номера факсимільного пристрою.

34. Для відображення всіх даних таблиці відмініть встановлення фільтру, відтиснувши кнопку *С фільтром* біля області навігації у нижній частині вікна.

### Індексування даних таблиць

35. Для швидкого сортування та пошуку членів профспілки за прізвищами створіть комплексний індекс таблиці *Співробітники*. Для цього:

35.1. Відкрийте таблицю *Співробітники* в режимі конструктора;

35.2. Для переходу в вікно індексів таблиці на стрічці меню натисніть кнопку ;

35.3. Встановіть параметри індексу *ЧлениПрофспілки* згідно рис. 16. Обгрунтуйте значення окремих параметрів.

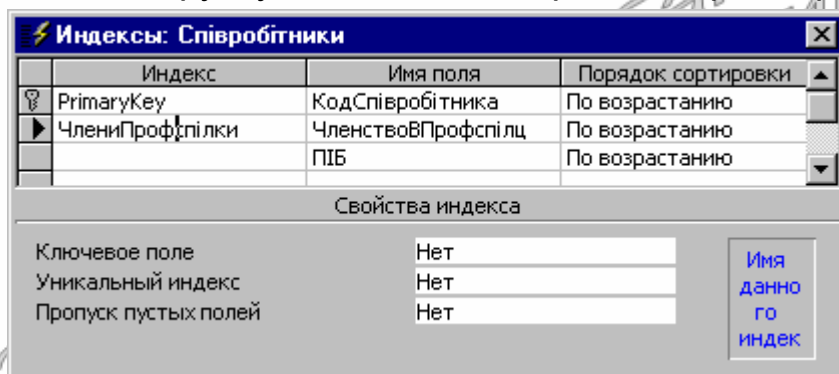


Рис. 16. Індеси таблиці *Співробітники*

36. Самостійно створіть індекс таблиці *Товари* для швидкого сортування та пошуку за сукупністю назви та марки товару.

### Копіювання даних таблиць

37. Перенесіть прізвища, дати народження та номери паспортів співробітників з таблиці *Сотрудники* в таблицю *Співробітники*. Для цього:

37.1. В імпортованій таблиці *Сотрудники* розмістіть в режимі таблиці поруч зліва направо стовпці *Прізвище*, *ДатаНародження*, *НомерПаспорта*;

37.2. Виділіть всі **дані** цих трьох стовпців (а не цілі стовпці). Для виділення полів покажчик мишки перемістіть на початок першого поля першого запису (покажчик мишки має набути вигляду жирного білого хрестика), натисніть ліву кнопку мишки та перемістіть покажчик до поля *НомерПаспорта* останнього запису;

37.3. Скопіюйте відмічені дані в буфер обміну;

37.4. Закрийте таблицю *Сотрудники*;

37.5. Відкрийте в режимі таблиці таблицю *Співробітники*, розмістіть поруч зліва направо стовпці *Прізвище*, *ДатаНародження*, *НомерПаспорта* та виділіть в ній ці три поля у новому записі;

- 37.6. Вставте дані з буфера обміну та підтвердіть створення нових записів. Якщо дані вставити не вдалося, то збільшіть розміри полів для копіювання і повторіть перенесення знову.
38. Спробуйте перенести окремо з таблиці *Сотрудники* в нові записи таблиці *Співробітники* прізвища, дати влаштування, номери паспорта. Чи вдалося сформувати нові записи? Чому?
39. Перенесіть в сформовані записи з таблиці *Сотрудники* в таблицю *Співробітники* домашні адреси, домашні телефони, фотокартки та примітки співробітників.

### **Завершальний етап заняття. Повторення вивченого матеріалу**

40. Знищіть БД *Zrazok.accdb*.
41. Завантажте власну БД та спробуйте відкрити пов'язану таблицю. Поясніть причину діагностичних повідомлень системи.
42. Після захисту роботи знищіть книгу *Export.xlsx*, а у власній БД – імпортовані, експортовані та пов'язані таблиці.
43. Стисніть створену БД.
44. Завершіть роботу Access.
45. Віднайдіть створену БД та заархівуйте її вміст.
46. Збережіть створений архів на власному носії в двох екземплярах.

### **Контрольні запитання**

1. З яких файлів може імпортувати та в які – експортувати дані Access?
2. В яких випадках імпортують, а в яких – пов'язують дані?
3. Яка умова має обов'язково виконуватися при доповненні імпортованими даними існуючих таблиць за допомогою майстра? Чому?
4. Де зберігається назва таблиці та назви полів при експорті даних у книги Excel?
5. Які поля мають обов'язково копіюватися при доповненні даних однієї таблиці інформацією іншої таблиці через буфер обміну?
6. Як виглядає кнопка *Режим* в різних режимах використання таблиці? Чому?
7. Як приховати, закріпити, відобразити, відмінити закріплення, перемістити, перейменувати стовпці в режимі таблиці? Які з цих операцій впливають на структуру таблиць в режимі конструктора?
8. Як відсортувати дані таблиці за багатьма полями?
9. Які способи фільтрування використовуються для відбору даних таблиць в Access? Коли слід використовувати кожен з цих способів?
10. Як і коли здійснюється створення додаткових індексів в таблицях Access?
11. Чому не доцільно створювати багато індексів за даними таблиці? Які індекси вважаються зайвими? Коли вони створюються і як їх вилучити?

## Лабораторна робота № 6

**Тема.** Створення звичайних, підпорядкованих та пов'язаних форм за допомогою майстра. Використання форм для введення та аналізу даних.

**Мета.** Формування вмінь та навичок створення форм за допомогою майстра для корегування та аналізу даних. Закріплення вмінь використання майстрів для формування об'єктів додатку. Вдосконалення навичок редагування даних. Усвідомлення ролі зв'язків між таблицями при створенні складних форм.

### Теоретичні відомості

Форми – це вікна або вкладки, які можуть відображати і корегувати дані таблиць чи організувати взаємодію між об'єктами додатку. В них не зберігаються, а лише візуалізуються дані БД. Для кожної таблиці можна розробити декілька форм різноманітного вигляду з різними можливостями корегування для різних категорій користувачів. У формі можуть одночасно виводитися дані декількох таблиць, але між цими таблицями мають бути попередньо встановлені структурні зв'язки у схемі даних або створений відповідний запит. Основні складові частини та розділи форм наведені на рис. 17.

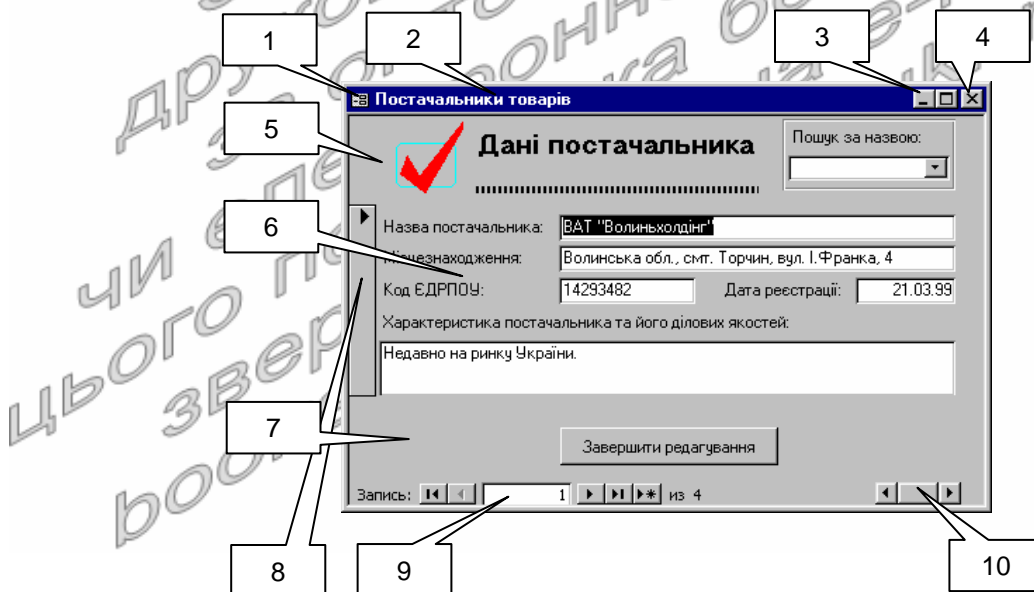


Рис. 17. Основні складові частини та розділи форм:


- |                                  |                              |
|----------------------------------|------------------------------|
| 1 – меню вікна;                  | 6 – область даних (розділ);  |
| 2 – заголовок вікна;             | 7 – примітки форми (розділ); |
| 3 – кнопки зміни розмірів вікна; | 8 – область виділення;       |
| 4 – кнопка закриття;             | 9 – кнопки навігації;        |
| 5 – заголовок форми (розділ);    | 10 – смуги прокрутки.        |

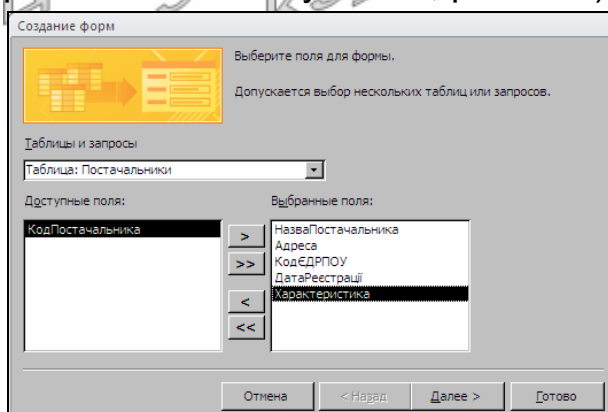
Як таблиці складаються з полів, так і у формах містяться окремі **елементи керування**. У властивостях редагованих елементів керування вказуються не лише назви полів, які вони опрацьовують, а й



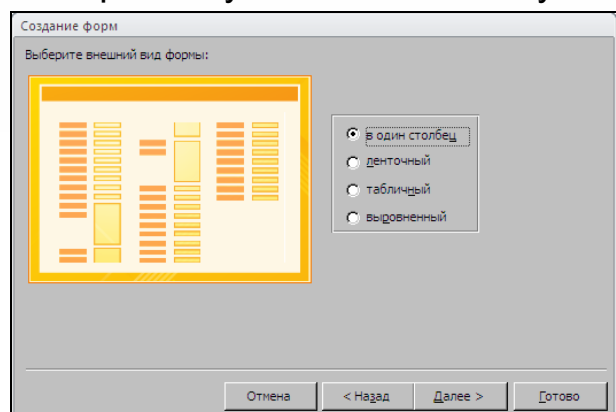
задаються параметри їх зовнішнього вигляду. Елементи керування розміщуються в одному з п'яти **розділів форми**: верхньому колонтитулі (виводиться при друці зверху на кожній сторінці), заголовку форми (рис. 17(4), відображається один раз у верхній частині форми), області даних (рис. 17(6), виводиться один раз для кожного запису, кількість виведених областей залежить від обраного режиму за замовчуванням), примітках форми (рис. 17(7), відображається у нижній частині форми) чи нижньому колонтитулі (виводиться при друці знизу на кожній сторінці). Для виділення цілого запису джерела даних у формі використовується **область виділення** (рис. 17(8)), а для переходу між записами – **кнопки навігації** (рис. 17(9)).

Форми, як і інші інструментальні засоби обробки даних, в СУБД MS Access найчастіше створюють двома основними способами: за допомогою майстра і в режимі конструктора. Використовуючи перший спосіб, розробник переносить з списку доступних в список обраних, всі поля, які підлягають редагуванню, обирає будову форми, вказує її назву, а сам процес створення форм виконує майстер. Цей спосіб дає змогу швидко створити форму стандартизованого вигляду. При другому способі користувач сам задає джерело даних і формує елементи керування, тому й розробляються такі форми довше, але мають досконаліший вигляд. На практиці рекомендується початковий вигляд форми задавати за допомогою майстра, доопрацювати – в режимі конструктора, а остаточно довершувати в режимі макету, оцінюючи вигляд вже введених даних.

Для завантаження майстра створення форм необхідно на стрічці меню перейти у вкладку *Создание* і у групі *Формы* натиснути кнопку  (*Мастер форм*). В процесі створення звичайних форм за допомогою майстра необхідно на першому його кроці обрати таблицю чи запит-джерело і перенести з списку доступних в список обраних всі поля, що мають редагуватися чи відображатися у формі (рис. 18а), на другому – вибрати варіант зовнішнього вигляду форми (фактично – задати режим по замовчуванню, рис. 18б), а на третьому – вказати її назву.



а) вікно першого кроку майстра



б) вікно другого кроку майстра

Рис. 18. Початкові кроки майстра створення звичайних форм для таблиці *Постачальники*

Проаналізуємо особливості будови кожного варіанту зовнішнього вигляду форм, створених майстром (рис. 19):

- *в один стовбець* – одночасно відображає дані лише одного запису (лише одну область даних), розміщуючи підпис кожного поля зліва від нього, а самі поля – одне під одним (рис. 19а). Використовується, коли потрібно корегувати багато полів з джерела даних;
- *выровненный* – аналогічно відображає дані одного запису, але підписи розміщуються безпосередньо над полями, а самі поля розтягуються до ширини форми (рис. 19б);
- *табличний* – відображає всі поля у вигляді таблиці в порядку перенесення на першому кроці майстра (рис. 19в). Застосовується, коли необхідно відобразити максимальну кількість записів;
- *ленточний* – одночасно відображає дані багатьох записів (виводить область даних для декількох записів джерела), розміщуючи підпис кожного поля зверху у заголовку форми, а самі поля – одне біля одного (рис. 19г). При цьому у верхній частині форми виводиться розділ заголовка форми, у нижній – примітки, а решта простору заповнюється областями даних для різних записів джерела. Цей варіант вигляду використовують, коли поля джерела можливо розмістити в одній стрічці.

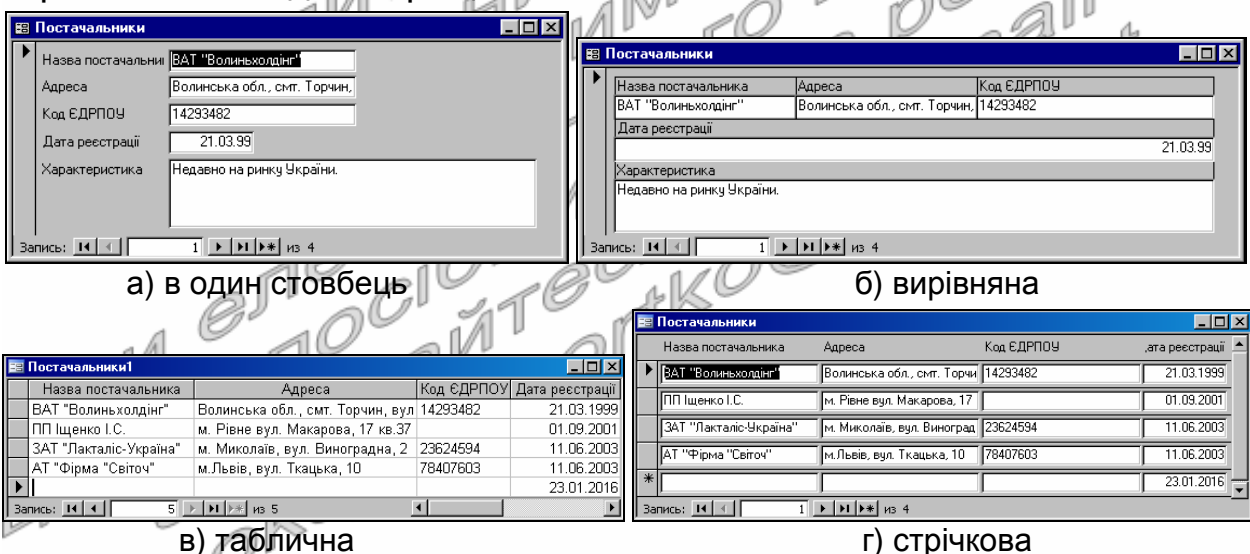


Рис. 19. Варіанти зовнішнього вигляду форм, створених майстром для таблиці *Постачальники*

Для уникнення неузгодженостей даних у кожній формі рекомендується редагувати дані лише однієї таблиці. Якщо ж необхідно забезпечити корегування даних одночасно декількох пов'язаних таблиць (наприклад, *ЗаголовкиЗамовлень* і *ПунктиЗамовлень*), то для таблиці на стороні 1 створюють **основну** форму, а для таблиць на стороні ∞ розробляють **підпорядковані** (відображаються в основній, як на рис. 20) чи **пов'язану** (викликається кнопкою з основної) форми. При цьому від кожної з таблиць, що використовується майстром, може відходити лише один зв'язок на стороні 1. Після створення форм в підпорядкованих і пов'язаних



формах відображаються лише записи, пов'язані з активним записом основної форми згідно структурного зв'язка між відповідними таблицями.

ЗаголовкиЗамовлень

Співробітник: Воронова Ольга Степанівна

Клієнт: Дудяк Олександр Васильович

Дата замовлення: 01.06.2016

ПунктиЗамовленьПП

Назва товару	Кількість	Ціна
згущене молоко 'Бахмач'	3	2,50 грн.
цукерки 'Столичні'	1	10,00 грн.
кава 'Галка'	3	5,00 грн.
морозиво 'Пломбір'	5	1,00 грн.
цукерки 'Червоний мак'	6	1,40 грн.

Записи: 1 из 7

Рис. 20. Форма для редагування даних таблиці *ЗаголовкиЗамовлень* з підпорядкованою формою для редагування даних таблиці *ПунктиЗамовлень*

Для створення основної і підпорядкованих чи пов'язаних форм за допомогою майстра необхідно на його першому кроці перенести з списку доступних в список обраних всі поля, що підлягають редагуванню, як з таблиці на стороні 1, так і з таблиць на стороні ∞ (рис. 21). **Не переносяться лічильники** (їх значення встановлюються автоматично) та поля, пов'язані з лічильником основної форми (адже їх значення встановлюються згідно структурних зв'язків). На другому кроці майстра обирається вигляд подання згідно таблиці на стороні 1 та різновид форми для таблиці на стороні ∞ (підпорядкована чи пов'язана), на третьому – варіант будови зовнішнього вигляду підпорядкованої форми (при її формуванні), а на четвертому – вказуються назви для створюваних форм.

Создание форм

Выберите поля для формы.  
Допускается выбор нескольких таблиц или запросов.

Таблицы и запросы  
Таблица: ЗаголовкиЗамовлень

Доступные поля: КодЗамовлення  
Выбранные поля: КодСпівробітника, КодКлієнта, ДатаЗамовлення

Создание форм

Выберите поля для формы.  
Допускается выбор нескольких таблиц или запросов.

Таблицы и запросы  
Таблица: ПунктиЗамовлень

Доступные поля: КодПункту, КодЗамовлення  
Выбранные поля: КодСпівробітника, КодКлієнта, ДатаЗамовлення, КодТовару, Кількість, Ціна, ПроцентЗнижки

а) перенесення полів з таблиці на стороні 1 б) перенесення полів з таблиці на стороні ∞

Рис. 21. Перший крок майстра створення основної форми *ЗаголовкиЗамовлень* та підпорядкованої форми *ПунктиЗамовлень*

Якщо знищити підпорядковану форму, то в основній на її місці буде порожній білий прямокутник. При знищенні основної форми підпорядкована чи пов'язана форма теж не зможе використовуватися

для введення, бо не буде встановлюватися значення її поля на стороні ∞ структурного зв'язка. Тобто підпорядкована чи пов'язана форми можуть використовуватися лише в комплексі з основною.


СУБД MS Access до версії 2013 мала також засоби створення діаграм і зведених таблиць для ефективного аналізу даних, принципи застосування яких також розглянемо в лабораторній роботі.

Література: [4, С. 125-129; 1, С. 213-239]

### **Підготовчий етап заняття. Актуалізація знань**

1. Завантажте Access, відкрийте з власної папки розроблену раніше БД *Sklad*.
2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип объекта*.
3. Перейдіть в області переходів до розділу *Формы*.

### **Створення та використання звичайних форм**

4. Створіть звичайну форму для редагування назв відділів. Для цього:
  - 4.1. Завантажте майстер створення форм, натиснувши у стрічці меню на вкладці *Создание* у групі *Формы* кнопку  *Мастер форм*;
  - 4.2. Оберіть в полі *Таблицы и запросы* таблицю *Відділи* та перенесіть з списку доступних в список обраних полів всі поля, призначені для редагування чи перегляду. В даному випадку слід перенести всі поля, крім поля *КодВідділу* (значення поля-лічильника встановлюються автоматично, не редагуються і призначені для внутрішнього використання). Після формування списку обраних полів натисніть кнопку *Далее*;
  - 4.3. На другому кроці майстра оберіть зовнішній вигляд форми в *один столбец* та натисніть кнопку *Далее*;
  - 4.4. Введіть назву форми *Відділи1* та натисніть кнопку *Готово*.
5. Відкоригуйте та доповніть список відділів (5-7 записів). Чи дозволяє сформована форма переглянути відразу назви багатьох відділів?
6. Самостійно створіть за допомогою майстра форми *Відділи2*, *Відділи3* та *Відділи4*, обираючи відповідно на другому кроці майстра вигляди *ленточный*, *табличный*, *выровненный*. При перегляді стрічкової форми відсортуйте назви відділів за зростання з допомогою контекстного меню відповідного поля. Чому стрічковий вигляд форми найкраще підходить для редагування назв відділів? Чому у всіх формах відображаються однакові дані? Де зберігаються дані цих форм?
7. Створіть форми *Співробітники1*, *Співробітники2*, *Співробітники3* та *Співробітники4*, обираючи відповідно на другому кроці майстра вигляди *в один столбец*, *ленточный*, *табличный* *выровненный* (на першому кроці майстра слід встановити в полі *Таблицы и запросы* значення *Таблица: Співробітники* та перенести в список обраних

полів всі поля, крім поля *КодСпівробітника*). Відкоригуйте та внесіть дані 7-8 співробітників, використовуючи для переміщення між полями даних співробітників клавішу *Tab*. Чому стовпцевий вигляд форми найкраще підходить для редагування даних співробітників? Чому поле *КодВідділу* переноситься в список обраних полів на першому кроці майстра?

8. Створіть різні варіанти форм та оберіть з них найкращий для таблиць *Товари*, *Клієнти* та *Постачальники*. Обґрунтуйте зроблений вибір. Внесіть в кожну з обраних форм по 7-8 записів.
9. Створіть форму з підпорядкованою формою для відображення даних постачань. Для цього:
  - 9.1. Самостійно завантажте майстер створення форм;
  - 9.2. На першому кроці майстра в полі зі списком *Таблиці і запити* виберіть таблицю *ЗаголовкиПостачань* та перенесіть з списку доступних в список обраних полів поля *КодСпівробітника*, *КодПостачальника* та *ДатаПостачання* (редаговані поля таблиці). Чому в список обраних полів не переноситься поле *КодПостачання*?
  - 9.3. **На тому ж кроці майстра**, додатково, в полі зі списком *Таблиці і запити* виберіть таблицю *ПунктиПостачань* та перенесіть зі списку доступних в список обраних полів поля *КодТовару*, *Кількість* та *Ціна*. Чому в список обраних полів не переносяться поля *КодПункту* та *КодПостачання*? Після перенесення полів натисніть кнопку *Далее*;
  - 9.4. На другому кроці оберіть вид подання даних по таблиці *ЗаголовкиПостачань*, встановіть перемикач *Подчиненные формы* та натисніть кнопку *Далее*;
  - 9.5. На третьому кроці оберіть вигляд підпорядкованої форми – *ленточный*. Самостійно завершіть створення форми з підпорядкованою формою, не змінюючи макет та замінивши назву основної форми на *ЗаголовкиПостачань*, а підпорядкованої – на *ПунктиПостачаньПП*. Чому на останньому кроці задаються назви двох, а не однієї форми?
10. Заповніть дані про 5-7 постачань. Для кожного постачання оформіть 4-5 пунктів.
11. Створіть форму з пов'язаною формою для відображення даних постачань аналогічно пункту 9, але на другому кроці встановіть перемикач *Связанные формы* і дайте формам інші назви. Якщо зі створеної основної форми неможливо викликати натисненням відповідної кнопки пов'язану форму, то перейдіть в режим конструктора та відсуньте надпис, який накладається на кнопку і знову поверніться в режим форми. Оцініть ефективність створених форм.

12. Цей та два наступних пункти не виконуються в СУБД MS Access, починаючи від версії 2013. Створіть діаграму для відображення термінів придатності різних товарів (як на рис. 22). Для цього:

- 12.1. В області переходів перейдіть до розділу таблиць та виділіть, як джерело даних, таблицю *Товари*;
- 12.2. Створіть форму в режимі зведеної діаграми, натиснувши у стрічці меню на вкладці *Создание* у групі *Формы* в списку *Другие формы* кнопку *Сводная диаграмма*;
- 12.3. В режимі зведеної діаграми перенесіть з вікна списку полів в область міток осі *OX* (область категорій) поле *НазваТовару*, а в область діаграми (область даних) – поле *Придатно*;



Рис. 22. Діаграма термінів придатності товарів

- 12.4. Для створеної діаграми в меню відтисніть кнопку *Зони* та оберіть найкращий тип подання даних на відповідній вкладці вікна її властивостей (це вікно відкривається при натисненні на стрічці меню у вкладці *Конструктор* кнопки *Изменить тип диаграммы* чи виборі аналогічного пункту контекстного меню вільної області діаграми);
  - 12.5. Закрийте і збережіть створену форму під назвою *Діаграма Термінів Придатності Товарів*.
13. Для відображення форми у вигляді зведеної таблиці про кількості отриманих товарів за різні дати (як на рис. 23) виконайте наступні дії:
- 13.1. Завантажте майстер створення форм з метою створення форми-джерела для розробки зведеної таблиці. Перенесіть на першому кроці майстра в список обраних полів поля:
    - 13.1.1. *НазваТовару* з таблиці *Товари*;
    - 13.1.2. *ДатаПостачання* з таблиці *ЗаголовкиПостачань*;
    - 13.1.3. *Кількість* з таблиці *ПунктиПостачань*;

Зведена Таблиця Кількостей Поставлених Товарів По Датах

Переташите сюда поля фильтра

НазваТовару	ДатаПостачання					Общие итоги
	17.06.2016	21.06.2016	25.06.2016	07.07.2016		
	Кількість	Кількість	Кількість	Кількість	Нет итогов	
згущене молоко 'Бахмач'		250			140	
кава 'Галка'	150	400			100	
					220	
майонез 'Делікатесний'		300			250	
сік 'Карапуз'		200			120	
цукерки 'Білочка'	200		100		150	
цукерки 'Зоряне сяйво'			100			
цукерки 'Ромашка'	200				150	
цукерки 'Столичні'			100			
цукерки 'Червоний мак'	150		100		150	
Общие итоги						

Рис. 23. Вигляд форми  
Зведена Таблиця Кількостей Поставлених Товарів По Датах  
в режимі зведеної таблиці

- 13.2. На другому кроці майстра оберіть вид подання даних згідно таблиці *ПунктиПостачань* для відображення всіх обраних полів в одній формі;
- 13.3. Самостійно завершіть створення форми-джерела та збережіть її під назвою *Зведена Таблиця Кількостей Поставлених Товарів По Датах*;
- 13.4. Для перетворення створеної форми у зведену таблицю виділіть її назву в області переходів та, **не завантажуючи її для редагування**, натисніть у стрічці меню на вкладці *Создание* у групі *Формы* в списку *Другие формы* кнопку *Сводная таблица*;
- 13.5. Перенесіть з вікна списку полів в область заголовків рядків поле *НазваТовару*, в область заголовків стовпців – поле *ДатаПостачання*, а в область даних таблиці – поле *Кількість* (див. рис. 23). Чому для створення цієї зведеної таблиці необхідно було спочатку створювати проміжну форму?
- 13.6. Проаналізуйте вибір полів для створення зведеної таблиці. Оцініть ефективність подання даних в отриманій формі. Закрийте форму та збережіть її під тією ж назвою *Зведена Таблиця Кількостей Поставлених Товарів По Датах*.
14. Самостійно створіть форму у вигляді зведеної таблиці про кількості замовлених товарів за різні дати.

### Завершальний етап заняття. Повторення вивченого матеріалу

15. Самостійно створіть форму з підпорядкованою формою для відображення даних замовлень, використовуючи редаговані поля



таблиць *ЗаголовкиЗамовлень* та *ПунктиЗамовлень*. При цьому основній формі дайте назву *ЗаголовкиЗамовлень*, а підпорядкованій – *ПунктиЗамовлень*ПП. Заповніть дані про 5-7 замовлень. Для кожного замовлення оформіть 4-5 пунктів. Аналогічно створіть форму з пов'язаною формою, але під іншими назвами. Дані яких таблиць модифікуються основними, підпорядкованою та пов'язаною формами?

16. Після захисту роботи залишіть для кожної з таблиць *Відділи*, *Співробітники*, *Постачальники*, *Товари*, *Клієнти* найкращу відповідну форму. Переіменуйте залишені форми згідно назв відповідних таблиць.
17. Знищіть **пов'язані** форми та відповідні їм основні форми, що редагували дані постачань та замовлень, **не знищуючи при цьому підпорядковані та їх основні форми**.
18. Стисніть створену БД.
19. Завершіть роботу Access.
20. Віднайдіть створену БД та заархівуйте її вміст.
21. Збережіть створений архів на власному носії в двох екземплярах.
22. Розробіть форми власної БД за допомогою майстра. Обґрунтуйте вибір для кожної таблиці звичайної, основної, підпорядкованої чи пов'язаної форми. Продемонструйте розроблені БД викладачу.

#### Контрольні запитання

1. Як завантажити майстер створення форм? Скільки форм можна створити для однієї таблиці?
2. Які поля переносяться в список обраних полів на першому кроці майстра?
3. Які поля не переносяться в список обраних полів на першому кроці майстра? Чому?
4. Які варіанти вигляду форм пропонуються майстром на другому кроці? В яких випадках використовується кожен з цих варіантів?
5. Коли доцільно створювати багатотабличні форми?
6. Чим звичайна форма відрізняється від основної? Основна від підпорядкованої? Підпорядкована від пов'язаної?
7. Звідки бере інформацію майстер створення форм Access про зв'язки між таблицями?
8. Як впливають структурні зв'язки на створення підпорядкованих і пов'язаних форм?
9. В яких випадках доцільно використовувати пов'язані, а в яких – підпорядковані форми?
10. Для чого створюють форми-діаграми та форми-зведені таблиці? Чи можливе редагування даних в цих об'єктах? Чому?
11. Звідки встановлюються початкові значення властивостей даних редагованих полів при створенні форми?





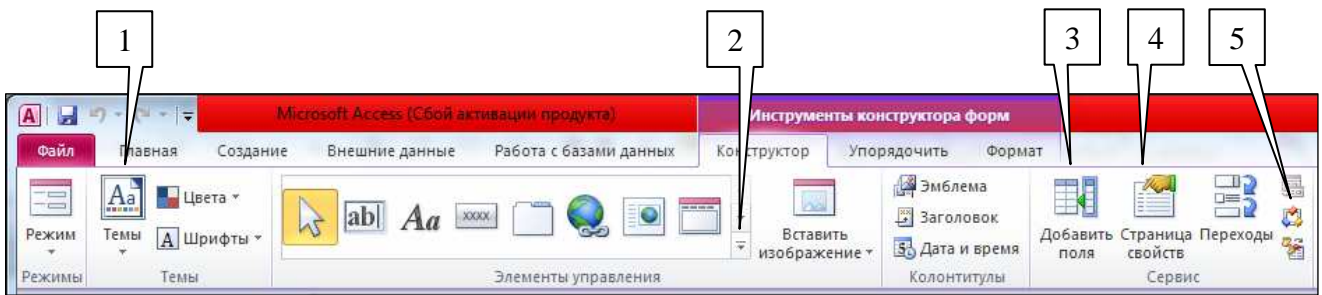


Рис. 25. Основні кнопки вкладки *Конструктор* режиму конструктора форми:

- 1 – вибір теми для оформлення стандартизованих форм;
- 2 – група елементів керування;
- 3 – режим видимості списку полів джерела даних форми;
- 4 – режим видимості вікна властивостей форми, її розділів чи елементів керування;
- 5 – відображення модуля процедур обробки подій форми.

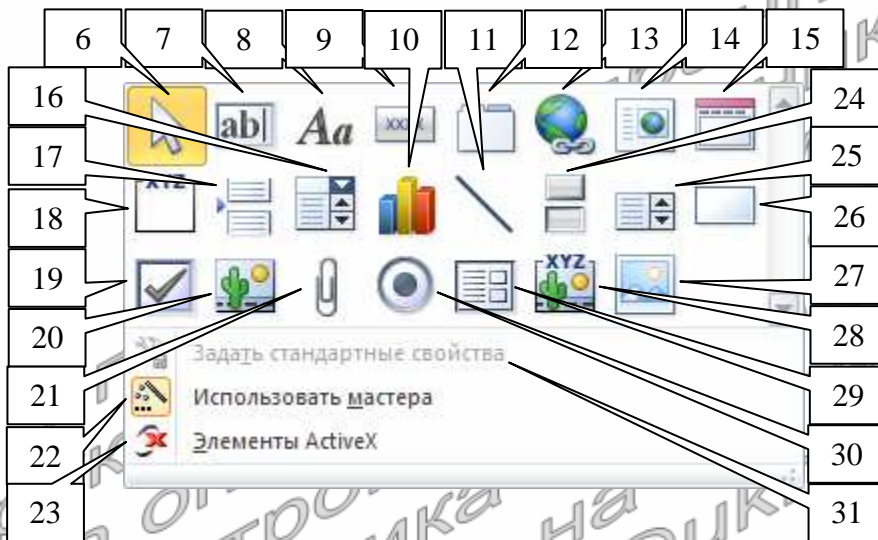



Рис. 26. Кнопки елементів керування режиму конструктора форми:

- 6 – кнопка групового виділення (вибору) об'єктів мишкою;
- 7 – кнопка для створення поля;
- 8 – кнопка для створення надпису;
- 9 – кнопка для створення кнопки;
- 10 – кнопка для створення діаграми;
- 11 – кнопка для створення лінії;
- 12 – кнопка для створення набору вкладок;
- 13 – кнопка для створення статичного гіперпосилання;
- 14 – кнопка для створення елемента типу «Веб-браузер»;
- 15 – кнопка формування елемента навігації;
- 16 – кнопка для створення поля зі списком;
- 17 – кнопка для вставки розриву сторінки;
- 18 – кнопка для створення групи перемикачів;
- 19 – кнопка для створення прапорця;
- 20 – кнопка для вставки вільної рамки об'єкта;
- 21 – кнопка для прикріплення вкладених об'єктів;
- 22 – кнопка виклику майстрів при створенні елементів;
- 23 – кнопка для вставки додаткового елемента ActiveX;
- 24 – кнопка для створення вимикача;
- 25 – кнопка для створення списку;
- 26 – кнопка для створення прямокутника;
- 27 – кнопка для вставки малюнка в форму;
- 28 – кнопка для вставки приєднаної рамки об'єкта;
- 29 – кнопка для створення підпорядкованої форми;
- 30 – кнопка для створення перемикача;
- 31 – кнопка для задання стандартних властивостей для нових елементів керування.

Задання параметрів елементів (форми загалом, її розділів чи окремих елементів керування) найчастіше виконується з **вікна властивостей**, яке викликається з контекстного меню чи кнопкою (4) після виділення відповідної складової частини (тут і надалі в теоретичних відомостях цієї лабораторної роботи для економії місця у круглих дужках наведено номери виносков з рис. 25-26). Для виділення окремого елемента керування за допомогою лівої кнопки мишки необхідно традиційно попередньо перемістити її покажчик на цей елемент, для виділення розділу – на смугу над ним (наприклад, на рис. 24 виділена смуга області даних), а для виділення цілої форми – на квадратик  між горизонтальною та вертикальною лініями. На практиці використовується також інший спосіб виділення елементів – вибір їх **назви** у списку верхньої частини вікна властивостей (рис. 27). Крім цього, властивості форми завжди можна вивести пунктом *Свойства формы* контекстного меню будь-якого активного елемента (як на рис. 24).

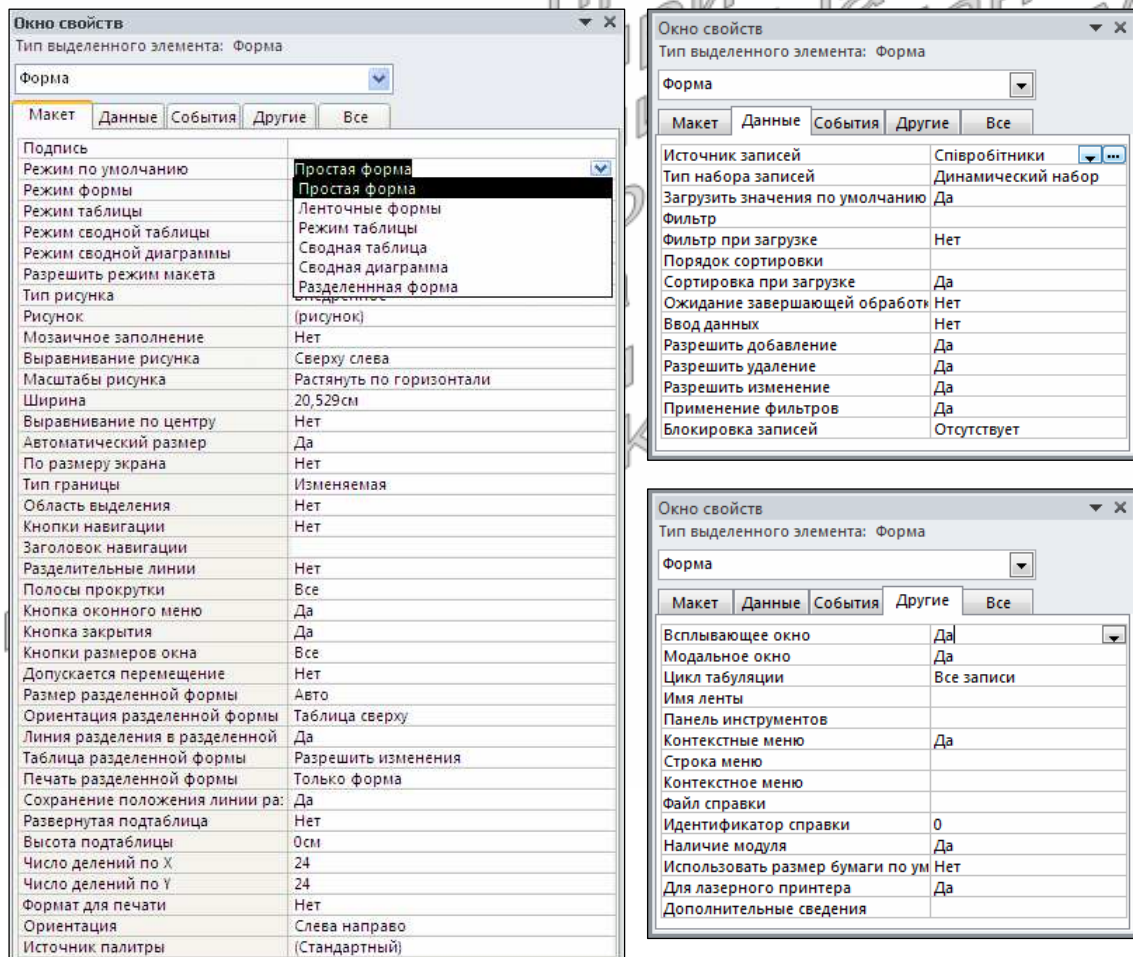


Рис. 27. Окремі вкладки вікна властивостей форми

Вікно властивостей режиму конструктора як форм, так і звітів (див. рис. 27) має п'ять стандартних вкладок для відображення всіх властивостей активного елемента чи спільних властивостей для групи виділених елементів керування:

1. *Макет* – містить параметри зовнішнього вигляду;
2. *Данные* – вміщує параметри джерела даних та можливостей його опрацювання;
3. *События* – містить перелік подій на які може реагувати активний елемент чи виділені елементи керування;
4. *Другие* – вміщує додаткові властивості, які не ввійшли в попередні вкладки, наприклад, ім'я елемента;
5. *Все* – містить параметри всіх попередніх вкладок.

В режимі конструктора завжди відображається структура лише однієї області даних. А кількість записів, які виводяться в режимі експлуатації, залежить від встановленого режиму відображення по замовчуванню (див. рис. 27). В Access 2010 використовуються шість режимів експлуатації:

- *простая форма* – забезпечує відображення в режимі експлуатації лише однієї області даних (для активного запису). Автоматично встановлюється майстром створення форм при виборі варіантів вигляду *в один столбец та выровненный*;
- *ленточные формы* – відображає заголовок і примітки форми, а решту робочого поля заповнює декількома областями даних, у тому числі – і для активного запису. Цей і наступний режими використовуються майстром для варіантів вигляду з аналогічними назвами;
- *режим таблицы* – відображає всі поля у вигляді таблиці в порядку послідовності переходу. Заголовок і примітки форми не відображаються, а над кожним полем виводиться текст приєднаного надпису;
- *разделенная форма* – розбиває форму на дві частини, в одній з яких виводить дані у табличному вигляді, а в іншій – активний запис в режимі простої форми;
- *сводная таблица* – забезпечує відображення відібраних даних записів з групуванням як по рядках, так і по стовпцях;
- *сводная диаграмма* – відображає аналізовані дані у вигляді динамічної діаграми. В Access 2013 цей і попередній режими експлуатації не використовуються.

Для того, щоб форма опрацьовувала дані бази, у вікні її властивостей на вкладці *Данные* необхідно задати значення поля *Источник записей* (див. рис. 27), сформувавши тим самим джерело даних. У цій властивості може бути обрана назва існуючої таблиці чи запиту або введена команда запиту за стандартом SQL. Можливості редагування даних джерела (внесення змін у записи, їх доповнення чи видалення, введення лише нових записів та ін.) також встановлюються на цій самій вкладці вікна властивостей форми. Лише після формування джерела даних з'явиться можливість за допомогою кнопки (3) відображати список полів і перетягувати з нього поля джерела у форму для швидкого створення елементів керування. При

цьому тип створюваного елемента керування залежить від типу даних та наявності підстановки значень поля джерела даних. Наприклад, для логічного поля створюється прапорець, для поля з підстановкою значень – поле зі списком або список, для більшості решти полів – звичайне поле. Створювати елементи керування для полів джерела даних можна й безпосередньо за допомогою кнопок панелі елементів (2), але при цьому на вкладці *Данные* необхідно самостійно задавати значення властивості *Данные* для формування значень цих елементів. Наголосимо, що декілька елементів керування можуть мати однакове значення властивості *Данные* і, відповідно, синхронно виводити однакові значення в режимі форми, але всі вони будуть мати різні імена (задаються на вкладці *Другие* вікна властивостей), оскільки саме за іменами виконується звертання до різних елементів. Для прискорення створення елементів керування за допомогою кнопок панелі елементів (2) в режимі конструктора форм і звітів використовуються відповідні майстри, але для їх завантаження необхідно, щоб кнопка (22) була обов'язково натиснута.

В режимі конструктора форми у кожному полі відображаються дані, які воно опрацьовує (рис. 28), або текст *Свободный*, якщо поле не має джерела даних, тому заповнювати поля в режимі конструктора недоречно. Навколо кожного з виділених елементів керування виводяться вісім маркерів. Верхній лівий з них дає змогу переміщувати активний елемент, навіть коли виділено декілька елементів керування. Решту маркерів застосовують для зміни розмірів виділених елементів.



Рис. 28. Виділений елемент керування форми в режимі конструктора

Для задання однакових значень спільним властивостям декількох елементів керування чи виконання їх одночасного переміщення необхідно виконати попереднє виділення цих елементів. З цією метою в режимі конструктора можливо використати не лише традиційне виділення лівою кнопкою мишки при натиснутій клавіші *Shift*, а й обхват елементів натягуванням умовного прямокутника мишкою при натиснутій кнопці (6) чи перетин елементів з лінією покажчика мишки, яку зпозиціоновано на горизонтальну чи вертикальну лінійку. Після виділення декількох елементів керування у вікні властивостей відображаються лише їх спільні властивості і саме тому задання значення у цьому вікні автоматично встановлює його для всіх виділених елементів.

Література: [4, С. 132-146; 1, С. 240-287]



### Підготовчий етап заняття. Актуалізація знань

1. Завантажте Access, відкрийте розроблену раніше БД *Склад* (вона має міститися у вашій папці).
2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип об'єкта*.
3. Перейдіть в області переходів до розділу *Таблиця*.
4. Доповніть таблицю *Співробітники* полями *МісцеНародження*, *Стать*, *Національність*, *ДатаЗвільнення*, *ПричиниЗвільнення*. Для поля *Стать* встановіть логічний тип даних і забезпечте вибір у полі зі списком значень *чоловіча* або *жіноча*. При цьому в самому полі мають зберігатися відповідно числа 0 та -1. По замовчуванню забезпечте відображення в даному полі значення *жіноча*. Самостійно вкажіть тип даних та, при потребі, значення додаткових властивостей для інших створених полів.
5. Створіть таблицю *ОсвітаСпівробітників* з структурою згідно табл. 14.

Таблиця 14

#### Структура та типи даних полів таблиці *ОсвітаСпівробітників*

Ім'я поля	Тип даних
КодОсвіти	Счетчик
КодСпівробітника	Числовой
РікВступу	Числовой
РікЗакінчення	Числовой
НазваЗакладу	Текстовый
ТипНавчання	Числовой
ТипОсвіти	Числовой
Спеціальність	Текстовый
Кваліфікація	Текстовый

6. Встановіть для обраного вами поля з унікальними значеннями властивість *ключового поля*.
7. Оформіть для редагованих полів властивості *Подпись* та *Описание*.
8. Задайте додаткові властивості полів, забезпечивши:
  - 8.1. Введення в поля *РікВступу* та *РікЗакінчення* чотирьох обов'язкових цифр. Недопустимість введення в ці поля значень, менших 1930 та відповідне повідомлення про помилку;
  - 8.2. Введення в текстові поля не більше сорока символів;
  - 8.3. Обов'язкове введення назви закладу;
  - 8.4. Можливість швидкого сортування та пошуку згідно з назвою закладу;
  - 8.5. Вибір зі списку при редагуванні поля *ТипНавчання* одного з трьох можливих значень: *денна*, *вечірня*, *дистанційна*. При цьому в самому полі мають зберігатися відповідно цифри 1, 2 чи 3. По замовчуванню забезпечте відображення в даному полі значення *денна*;
  - 8.6. Вибір зі списку при редагуванні поля *ТипОсвіти* одного з п'яти



можливих значень: *початкова, середня, середня спеціальня, незакінчена вища, вища*. При цьому в самому полі мають зберігатися відповідно цифри 1, 2, 3, 4 чи 5. По замовчуванню забезпечте відображення в даному полі значення *вища*.

9. Перейдіть у вікно схеми даних та встановіть між таблицями *Співробітники* і *ОсвітаСпівробітників* зв'язок, об'єднавши між собою поля *КодСпівробітника*. Забезпечте цілісність посилань між цими таблицями.
10. Перейдіть в області переходів до розділу *Формы*.
11. Знищіть всі форми, що базувалися на таблиці *Співробітники* (використовували цю таблицю як джерело даних, наприклад, *Співробітники, Співробітники1* і т.д.).

### **Створення форм та корегування їх властивостей в режимі конструктора**

12. Створіть форму, аналогічну наведеній на рис. 29, для редагування даних співробітників. Для цього:
  - 12.1. Створіть порожню форму одним з двох способів: на вкладці стрічки меню *Создание* у групі *Формы* натисніть кнопку *Конструктор форм* або у цій же групі натисніть кнопку *Пустая форма* і перейдіть в режим конструктора;
  - 12.2. Для відображення області заголовку та приміток форми натисніть в контекстному меню вільного місця робочої області форми кнопку *Заголовок/примечание формы*. Аналогічно забезпечте відображення лінійки та сітки для вирівнювання елементів керування;
  - 12.3. Встановіть висоту області *Заголовок формы* рівною приблизно 1,7 см. Для цього:
    - 12.3.1. Розташуйте покажчик мишки над областю даних (покажчик мишки набуде вигляду лінії з двома стрілками);
    - 12.3.2. Натисніть ліву кнопку мишки та змініть висоту розділу;
  - 12.4. Встановіть точне значення висоти розділу у вікні властивостей. Для цього:
    - 12.4.1. Активізуйте вікно властивостей натисненням кнопки (4) (тут і надалі у цій лабораторній роботі в круглих дужках наведено номери виносков з рис. 25-26 та рис. 30. Тому для прискорення виконання роботи ці рисунки настійливо рекомендуємо роздрукувати);
    - 12.4.2. Зробіть активною область *Заголовок формы*, виділивши її заголовок;
    - 12.4.3. На вкладці *Макет* вікна властивостей віднайдіть параметр *Высота* та встановіть його значення рівним 1,7 см;
  - 12.5. Самостійно встановіть висоту області даних рівною 8,3 см, а області приміток форми – 1 см;

Співробітники ринку

**Особова картка співробітника ринку**

Пошук співробітника:

Ініціали:

Стать:  чоловік  жінка  
 Дата народження:   
 Місце народження:

Національність:  Член профспілки?


Освіта | Семейний стан | Дані паспорта | Стаж | Призначення | Відпустки

Рік вступу	Рік закінчення	Назва закладу	Форма навчання	Тип освіти
▶ 1970	1975	КНЕУ	денна	середня спе
* 0	0			

Спеціальність:  Кваліфікація:

Запись:  из 1

Дата звільнення:  Причини та підстави звільнення:

Закордонна фотокартка: 

Характеристика:

Завершити редагування

Запись:  из 1

Рис. 29. Форма для редагування даних співробітників в режимі форми

Співробітники ринку : форма

32

Особова картка співробітника ринку

Пошук співробітника:

33 34 35

36 37

38 39 40


41 42 43 44

45 46 47

48

49 50


51

Закордонна фотокартка: 

Характеристика:

Завершити редагування

Рис. 30. Форма для редагування даних співробітників в режимі конструктора

- 12.6. Встановіть ширину форми рівною приблизно 16,9 см. Для цього перемістіть покажчик мишки на праву межу робочого поля області форми та відтягніть її до позначки 16,9 см;
- 12.7. Встановіть точне значення ширини форми у вікні властивостей. Для цього:
- 12.7.1. При відсутності вікна властивостей активізуйте його натисненням кнопки (4);
- 12.7.2. Зробіть активною цілу форму, відмітивши область (32) або обравши в контекстному меню робочої області пункт *Свойства формы*;
- 12.7.3. На вкладці *Макет* вікна властивостей віднайдіть параметр *Ширина* та встановіть його значення рівним 16,9 см;
- 12.8. Задайте заголовок створеної форми. А саме: у вікні властивостей форми перейдіть на вкладку *Макет* та введіть у значення властивості *Подпись* текст *Співробітники ринку*;
- 12.9. Оцініть вигляд створеної форми у режимі форми. Для переходу в цей режим скористайтеся відповідним пунктом контекстного меню вікна форми або натисніть кнопку  на вкладках *Главная* чи *Конструктор* стрічки меню. Де виводиться заголовок форми? Як називається область з трикутником, що розміщена з лівого боку вікна? Чому у формі не відображаються дані? Поверніться в режим конструктора. Поекспериментуйте з різними темами оформлення (1). Встановіть початкову тему оформлення;
- 12.10. Встановіть у вікні властивостей форми на вкладці *Макет* значення параметрів згідно табл. 15;

Таблиця 15

**Значення окремих властивостей форми на вкладці *Макет***

Властивість	Значення
Режим по умолчанию	Простая форма
Полосы прокрутки	Отсутствуют
Область выделения	Нет
Разделительные линии	Нет
Автоматический размер	Да
Выравнивание по центру	Да

Закрийте форму та дайте їй назву *Співробітники*. Відкрийте форму в режимі форми. Як кожен з встановлених параметрів вплинув на вигляд форми? Знову поверніться в режим конструктора;

- 12.11. Встановіть для форми джерело даних – таблицю *Співробітники*. Для цього у вікні властивостей форми перейдіть на вкладку *Данные* та у полі зі списком *Источник записей* оберіть таблицю *Співробітники*. Натисніть кнопку

списку полів джерела даних форми (3). Звідки взяли назви полів у виведеному вікні?

- 12.12. Ознайомтеся зі всіма властивості форми на вкладках *Макет*, *Данные*, *Другие*. Призначення незнайомих властивостей віднайдіть у довідковій системі. За результатами ознайомлення для невідомих властивостей заповніть в конспекті таблицю за зразком табл. 16.

Таблиця 16

**Властивості форми**

Вкладка	Властивість	Можливі значення	Призначення
Макет	Подпись	Всемажливі текстові рядки	Заголовок вікна форми в режимі форми
	Режим по умовчаниу	Простая форма, Ленточные формы, Режим таблицы ...	Варіант завантаження форми в режимі форми ...
	...	...	...

**Створення елементів керування форм та корегування їх властивостей в режимі конструктора**

13. Оснастіть створену форму елементами керування згідно рис. 29. Для цього:

- 13.1. В заголовку форми створіть напис (34), виконуючи наступні дії:

- 13.1.1. Переконайтеся, що кнопка (22) натиснута;
- 13.1.2. Для створення надпису натисніть кнопку (8);
- 13.1.3. Натягніть у заголовку форми прямокутник для розміщення тексту надпису;
- 13.1.4. Введіть текст надпису. Виділіть надпис за допомогою мишки. На вкладці стрічки меню *Формат* встановіть розмір шрифту – 14 пт. та натисніть кнопку Ж;

- 13.1.5. Перемістіть текст надпису у відведене місце (див. рис. 29) за допомогою мишки. Для цього:

- 13.1.5.1. Виділіть створений надпис;
- 13.1.5.2. Перемістіть його за верхній лівий маркер (мишка повинна мати вигляд двох перпендикулярних двонапрямлених стрілок);

- 13.1.6. Здійсніть переміщення надпису, змінюючи значення властивостей *От левого края* та *От верхнего края* на вкладці *Макет* вікна його властивостей;

- 13.1.7. Поекспериментуйте з різними значеннями властивостей надпису у вікні властивостей. Дію кожної незрозумілої властивості проаналізуйте в режимі форми. Нові властивості опишіть в конспекті у таблиці *Особливі властивості надписів* згідно зразка табл. 16;

13.2. Під розробленим надписом створіть лінію (35), виконуючи наступні дії:

13.2.1. Переконайтеся, що кнопка (22) натиснута;

13.2.2. Для створення лінії натисніть кнопку (11);

13.2.3. Натягніть у заголовку форми лінію;

13.2.4. У вікні властивостей лінії на вкладці *Макет* встановіть значення параметрів у відповідності з табл. 17;

Таблиця 17

**Значення окремих властивостей лінії (35) на вкладці *Макет***

Властивість	Значення
Ширина	9,8
Высота	0
Тип границы	Точечная
Ширина границы	4 пункта

13.2.5. Перемістіть лінію у відведене місце (див. рис. 29);

13.2.6. Поекспериментуйте з різними значеннями властивостей лінії у вікні властивостей. Дію кожної незрозумілої властивості проаналізуйте в режимі форми. Нові властивості опишіть в конспекті у таблиці *Особливі властивості ліній* згідно зразка табл. 16;

13.3. Поряд з надписом та лінією в заголовку форми вставте обраний вами малюнок-емблему організації на місці (33), виконуючи наступні дії:

13.3.1. Віднайдіть на комп'ютері чи завантажте з Інтернету і перемістіть у власну папку малюнок-емблему на ваш вибір розміром біля 50 Кб у форматі BMP;

13.3.2. Для вставки малюнка натисніть кнопку (27);

13.3.3. Натягніть у заголовку форми прямокутник для розміщення малюнка;

13.3.4. У вікні *Выбор рисунка*, перейдіть у власну папку, оберіть потрібний малюнок та натисніть кнопку *OK*;

13.3.5. Перемістіть малюнок у відведене місце (див. рис. 29). У вікні властивостей малюнка на вкладці *Макет* віднайдіть властивість *Установка размеров* та задайте у ній значення *Вписать в рамку*;

13.3.6. Поекспериментуйте з різними значеннями властивостей малюнка у вікні властивостей. Дію кожної незрозумілої властивості проаналізуйте в режимі форми. Нові властивості опишіть в конспекті у таблиці *Особливі властивості малюнків*;

13.4. Активізуйте вікно списку полів джерела даних, натиснувши кнопку (3);

13.5. Створіть у формі поле для відображення прізвища, імені та по батькові співробітника. Для цього:

- 13.5.1. У вікні списку полів джерела даних віднайдіть поле *ПІБ* та перетягніть його при натиснутій лівій кнопці мишки в область даних на місце реквізиту (38);
- 13.5.2. Якщо поле опинилося не на своєму місці, то виділіть його та перемістіть у відведене місце (див. рис. 29);
- 13.5.3. Збільшіть розміри поля, перетягуючи маркери при натиснутій лівій кнопці мишки. Вказівник мишки при цьому повинен мати вигляд двонаправленої стрілки;
- 13.5.4. Поекспериментуйте з різними значеннями властивостей поля у вікні властивостей. Дію кожної незрозумілої властивості проаналізуйте в режимі форми. Нові властивості опишіть в конспекті у таблиці *Особливі властивості поля*;
- 13.6. Відредагуйте надпис для поля *ПІБ*, замінюючи його текстом *Ініціали*:. Для цього:
  - 13.6.1. Виділіть надпис біля розглядуваного поля;
  - 13.6.2. Змініть текст надпису, редагуючи властивість *Подпись* у вікні властивостей надпису, або відмітивши ще раз мишкою текст надпису (при цьому безпосередньо в тексті надпису з'явиться курсор для редагування його тексту);
  - 13.6.3. Виділіть надпис, змініть його розміри та перемістіть у відведене місце (див. рис. 29). Який початковий текст містив надпис для поля (38)? Звідки скопіювалося його значення? Чи впливає надпис на властивості поля *ПІБ* базової таблиці?
- 13.7. Самостійно створіть у формі поля для редагування полів *ДатаНародження* (39), *МісцеНародження* (43), *Національність* (42), *Примітки* (48), *ДатаЗвільнення* (49), *ПричиниЗвільнення* (50) таблиці *Співробітники*. Встановіть положення та розміри створених полів згідно зразка. Поясніть значення властивості *Формат поля* кожного оформленого поля. Змініть тексти надписів полів. Встановіть положення та розміри надписів полів аналогічно рис. 29;
- 13.8. Перетягніть в область даних форми поле *Фотокартка* з вікна списку полів джерела даних. Встановіть його розмір та положення згідно реквізиту (40). Чому створився елемент керування *Присоединенная рамка объекта*? Поекспериментуйте з різними значеннями властивостей елемента керування у вікні властивостей. Дію кожної незрозумілої властивості проаналізуйте в режимі форми. Нові властивості опишіть в конспекті у таблиці *Особливі властивості приєднаної рамки малюнка*;
- 13.9. Перетягніть у форму поле *ЧленствоВПрофспілці* та розташуйте його на місці реквізиту (44). Відкоригуйте надпис та положення поля. Чому для даного поля автоматично



створено елемент керування *Флажок*? Яке значення по замовчуванню встановлено для даного елемента керування? Звідки скопійовано це значення? Поекспериментуйте з різними значеннями властивостей елемента керування у вікні властивостей. Дію кожної незрозумілої властивості проаналізуйте в режимі форми. Нові властивості опишіть в конспекті в таблиці *Особливі властивості приєднаної прапора*;

13.10. Створіть групу перемикачів для редагування статі співробітника (41). Для цього:

13.10.1. Переконайтеся, що кнопка (22) панелі елементів натиснута;

13.10.2. Натисніть кнопку (18) панелі елементів;

13.10.3. Натягніть в області даних форми область для створення групи перемикачів (41). При цьому завантажиться майстер створення групи перемикачів (перехід до кожного наступного кроку майстра здійснюється кнопкою *Далее*);

13.10.4. На першому кроці майстра введіть текст групи перемикачів: в першому рядку наберіть *чоловіча*, а в другому – *жіноча*;

13.10.5. На другому кроці майстра вкажіть, що по замовчуванню встановлюється значення перемикача *жіноча*;

13.10.6. На третьому кроці майстра вкажіть унікальні значення кожного перемикача: для перемикача *чоловіча* – значення –1, для перемикача *жіноча* – значення 0;

13.10.7. На четвертому кроці майстра вкажіть, що значення групи перемикачів **зберігається в полі**, а в списку полів оберіть поле *Стать*;

13.10.8. На п'ятому кроці оберіть тип елемента керування – *перемикачі* та тип оформлення групи – *втиснута*;

13.10.9. На шостому кроці майстра завершіть створення групи перемикачів, натиснувши кнопку *Готово*;

13.10.10. Виділіть групу перемикачів, змініть їх розмір та перемістіть у відведене місце (див. рис. 29). Перемістіть надпис цієї групи у відведене місце, розмістивши покажчик мишки на його верхньому лівому кутку;

13.10.11. Поекспериментуйте з різними значеннями властивостей елемента керування у вікні властивостей. Дію кожної незрозумілої властивості проаналізуйте в режимі форми. Нові властивості опишіть в конспекті у таблиці *Особливі властивості групи перемикачів*;

13.11. Створіть у формі набір вкладок (46). Для цього:

13.11.1. Натисніть кнопку (12) панелі елементів та натягніть в

- області даних форми область для розміщення вкладок згідно зразка;
- 13.11.2. Дайте першій вкладці назву *Освіта*. Для цього у вікні властивостей **цієї вкладки** (а не цілого набору) змініть значення властивості *Подпись*;
  - 13.11.3. Самостійно дайте другій вкладці назву *Сімейний стан*;
  - 13.11.4. Доповніть набір вкладок новою вкладкою *Дані паспорта*. Для цього в контекстному меню наявного **набору вкладок** оберіть пункт *Добавить вкладку* та змініть її підпис;
  - 13.11.5. Самостійно *доповніть існуючий* набір вкладок вкладками *Стаж, Призначення, Відпустки*;
  - 13.11.6. Змініть розміри та перемістіть у відведене місце (див. рис. 29) створений набір вкладок. Поекспериментуйте з новими пунктами контекстного меню та різними значеннями властивостей створеного елемента керування у вікні властивостей. Дію кожної незрозумілої властивості проаналізуйте в режимі форми. Нові пункти контекстного меню та властивості опишіть в конспекті у таблиці *Особливі властивості набору вкладок*;
- 13.12. Створіть на вкладці *Освіта* підпорядковану форму (47) для внесення даних про освіту співробітника. Для цього:
- 13.12.1. На панелі елементів натисніть кнопку (29);
  - 13.12.2. Натягніть в області даних форми прямокутну область для відображення підпорядкованої форми з даними етапів освіти співробітника. При цьому завантажиться майстер створення підпорядкованих форм (перехід до кожного наступного кроку майстра здійснюється кнопкою *Далее*);
  - 13.12.3. На першому кроці майстра вкажіть тип джерела даних – таблиця чи запит;
  - 13.12.4. На другому кроці майстра вкажіть джерело даних підпорядкованої форми – таблиця *ОсвітаСпівробітників* та перенесіть з списку доступних в список обраних всі поля, крім *КодОсвіти* та *КодСпівробітника* (Чому?);
  - 13.12.5. На третьому кроці оберіть перший тип зв'язку між основною та підпорядкованою формою;
  - 13.12.6. На четвертому кроці задайте назву підпорядкованої форми – *ОсвітаСпівробітників*;
  - 13.12.7. Після створення підпорядкованої форми змініть її розміри та перемістіть у відведене місце (див. рис. 29);
  - 13.12.8. Встановіть для створеної підпорядкованої форми у вікні властивостей (викликається з контекстного меню області (45)) на вкладці *Макет* режим по замовчуванню

*Ленточные формы* (Чому?);

- 13.12.9. Перенесіть в підпорядкованій формі з використанням буфера обміну поля *Спеціальність* і *Кваліфікація* разом з підписами з області даних в примітки форми, розмістивши їх, як на рис. 29. Перейдіть в режим макету та встановіть розміри цих полів так, щоб коректно відображати їх дані. Знову поверніться до режиму конструктора;
- 13.12.10. Виріжте **надписи** решти полів з області даних та вставте їх у заголовок підпорядкованої форми (47), як на рис. 29;
- 13.12.11. Розмістіть поля підпорядкованої форми в області даних під своїми надписами. Зменшіть розмір області даних підпорядкованої форми (47) до однієї стрічки, як на рис. 30;
- 13.13. В примітках основної форми створіть кнопку (51) для закриття форми після редагування. Для цього:
  - 13.13.1. На панелі елементів натисніть кнопку (9);
  - 13.13.2. Натягніть в області приміток форми прямокутну область кнопки. При цьому завантажиться майстер створення кнопок;
  - 13.13.3. На першому кроці майстра оберіть категорію *Робота с формой* та дію *Закрытие формы*;
  - 13.13.4. На другому кроці майстра введіть текст кнопки: *Завершення редагування*;
  - 13.13.5. На третьому кроці завершіть створення кнопки, натиснувши кнопку *Готово*;
  - 13.13.6. Змініть розміри та перемістіть кнопку згідно рис. 29;
  - 13.13.7. Перейдіть в режим форми та переконайтеся в дієздатності кнопки.
  - 13.13.8. Поверніться в режим конструктора. Поекспериментуйте з різними значеннями властивостей кнопки у вікні властивостей. Дію кожної незрозумілої властивості проаналізуйте в режимі форми. Нові властивості опишіть в таблиці *Особливі властивості кнопки*.

#### **Форматування груп елементів керування**

14. Встановіть однакові розміри елементів керування полів *ДатаНародження* (39) та *МісцеНародження* (43). Для цього:
  - 14.1. При натиснутій клавіші Shift відмітьте елементи керування (39) та (43) почерговим натисненням лівої кнопки мишки;
  - 14.2. На вкладці стрічки меню *Упорядочить* відкрийте список кнопки *Размер или интервал* та оберіть у ньому пункт *По самому широкому*.
15. Вирівняйте елементи керування полів *ПІБ* (38), *ДатаНародження* (39) та *МісцеНародження* (43) по правому краю. Для цього:

- 15.1. При натиснутій клавіші Shift відмітьте елементи керування (38), (39) та (43) почерговим натисненням лівої кнопки мишки;
- 15.2. На цій самій вкладці стрічки меню у списку кнопки *Вирівнять* оберіть пункт *Справа*.
16. Встановіть між елементами керування полів *ПІБ* (38), *Дата Народження* (39) та *Місце Народження* (43) однакові інтервали по вертикалі. Для цього:
  - 16.1. Виділіть елементи керування (38), (39) та (43);
  - 16.2. На вкладці стрічки меню *Упорядочить* знову відкрийте список кнопки *Размер или интервал* та оберіть у ньому пункт *Распределить по вертикали*;
  - 16.3. Самостійно встановіть однакові розміри, інтервали по вертикалі та вирівняйте елементи керування згідно рис. 29.

### **Створення елементів керування для пошуку даних в формах**

17. Оснастіть створену форму полем зі списком (37) для організації швидкого пошуку співробітника за ініціалами. Для цього:
  - 17.1. Натисніть на панелі елементів кнопку (16);
  - 17.2. Натягніть в області заголовка форми область для створення поля зі списком (37). При цьому завантажиться майстер створення полів зі списками;
  - 17.3. На першому кроці майстра встановіть перемикач вибору варіантів отримання даних полем зі списком в третє положення (організація пошуку в формі). Цей перемикач наявний лише тоді, коли для форми вказано джерело даних;
  - 17.4. На другому кроці майстра перенесіть з списку доступних в список обраних поля *Код Співробітника* (для проведення пошуку) та *ПІБ* (для відображення в полі зі списком);
  - 17.5. На третьому кроці переконайтеся, що прапорець *Скрыть ключевой столбец* встановлено;
  - 17.6. На четвертому кроці завершіть створення поля зі списком, натиснувши кнопку *Готово*;
  - 17.7. Перемістіть та змініть розміри поля зі списком (37) та його надпису згідно рис. 29. Змініть текст надпису поля зі списком;
  - 17.8. Перегляньте властивості створеного поля зі списком у вікні властивостей. Про кожну незрозумілу властивість дізнайтеся з довідкової системи. Нові властивості опишіть в конспекті у таблиці *Особливі властивості поля зі списком*.
18. Для виділення поля зі списком (37) створіть в заголовку форми прямокутник (36). Для цього:
  - 18.1. Натисніть кнопку (26) панелі елементів керування;
  - 18.2. Натягніть в області заголовка форми прямокутник згідно рис. 29.

### **Завершальний етап заняття**

19. Самостійно створіть на інших вкладках набору вкладок (46)

елементи керування для редагування інших полів таблиці *Співробітники* (всі поля цієї таблиці, крім лічильника, повинні мати відповідні елементи керування для редагування їх значень). Розмістіть, вирівняйте та встановіть інтервали між ними, використовуючи операції форматування груп елементів.

20. Перейдіть в режим форми та внесіть з її допомогою 2-3 записи в таблицю *Співробітники*. Для кожного співробітника внесіть інформацію про отриману освіту (в середньому по дві освіти для кожного співробітника).
21. Закрийте форму та перевірте наявність введених даних в таблицях *Співробітники* та *ОсвітаСпівробітників*. Зверніть увагу на значення пов'язаних полів.

### **Повторення вивченого матеріалу**

22. Стисніть створену БД.
23. Завершіть роботу Access.
24. Віднайдіть створену БД та заархівуйте її вміст.
25. Збережіть створений архів на власному носії в двох екземплярах.
26. Вдоскональте форми власної БД в режимах конструктора та макета. Продемонструйте розроблені БД викладачу.

### **Контрольні запитання**

1. Чим елемент конструктора форми відрізняється від елемента керування?
2. Як змінити висоту розділів форми двома способами?
3. Як змінити ширину форми двома способами?
4. В якому режимі відображається введений заголовок для вікна форми?
5. Як встановити чи змінити джерело даних форми?
6. Якими основними способами можна змінити властивості форм та елементів керування?
7. Чому форми та елементи керування мають різні властивості? Чому елементи керування різних типів мають різні властивості? Які властивості притаманні всім елементам керування? Чому?
8. Які особливі властивості мають окремі типи елементів керування?
9. Звідки копіюється текст для надписів кожного поля, що перетягується з вікна списку полів джерела даних форми?
10. Чому кожен перемикач групи перемикачів повинен мати унікальне значення? Як змінити значення по замовчуванню для створеної групи перемикачів?
11. При створенні якого елемента керування створюється нова форма?
12. Яка різниця між іменем та джерелом даних елемента керування?
13. Які види форматування груп елементів керування використовуються в режимі конструктора? Як одночасно виділити декілька елементів керування?
14. Які різновиди полів зі списками використовуються в формах?
15. Чи може одна форма вносити дані відразу в декілька таблиць?

## Лабораторна робота № 8

**Тема.** Створення власної та стандартної головних кнопоквих форм. Корегування властивостей створених форм та елементів керування в режимі конструктора. Створення обчислювальних полів у формах.

**Мета.** Формування вмінь та навичок задання параметрів завантаження додатку. Закріплення вмінь та навичок створення елементів керування в формах. Вдосконалення вмінь та навичок корегування форм в режимі конструктора.

### Теоретичні відомості

Для організації систематизованого доступу до різних об'єктів в ІС найчастіше використовують **головне меню** (вкладки на стрічці меню) або **головну кнопку форму**. Перший варіант застосовують у традиційних додатках для ПК, другий – при розробці інтерфейсів, подібних до оформлення мобільних пристроїв. Наголосимо, що як з меню так і з кнопоквих форм викликаються лише об'єкти, призначені для кінцевого користувача чи адміністратора (форми, звіти, запити на корегування, операції експорту/імпорту), а не для зберігання чи проміжної обробки даних (таблиці, запити на вибірку, модулі).

В сучасних версіях Access для створення кнопок на стрічці меню, які відкривають інші об'єкти (найчастіше – форми і звіти) спочатку розробляють макроси відкриття цих об'єктів, а потім, використовуючи можливості вкладки *Файл* ► *Параметри* ► *Настройка лент*, додають нову вкладку на стрічку меню та створюють у цій вкладці групи і кнопки, які викликають створені макроси.

У випадку ж створення головних кнопоквих форм використовують два підходи: або розробляють власну кнопку форму в режимі конструктора, або використовують диспетчер кнопоквих форм для створення стандартної кнопкової форми. В таких формах, як правило, розмежовують обробку оперативної, умовно-постійної та звітної інформації і команд автоматичного корегування даних, що підвищує рівень орієнтованості користувача в ІС. З цією метою у власній кнопковій формі створюють відповідний набір вкладок (як, наприклад, на рис. 31, 38), а в стандартній – окремі сторінки кнопкової форми. Після цього в кнопковій формі обов'язково розробляють кнопку для завершення роботи з ІС.

Крім цього, у власній кнопковій формі необхідно приховати область виділення, кнопки навігації та розмежовуючі лінії областей даних, адже така форма не виконує редагування джерела даних (тому й не містить записів з даними, між якими можливо виконувати переходи чи які доцільно виділяти), а лише організовує взаємодію між об'єктами додатку.



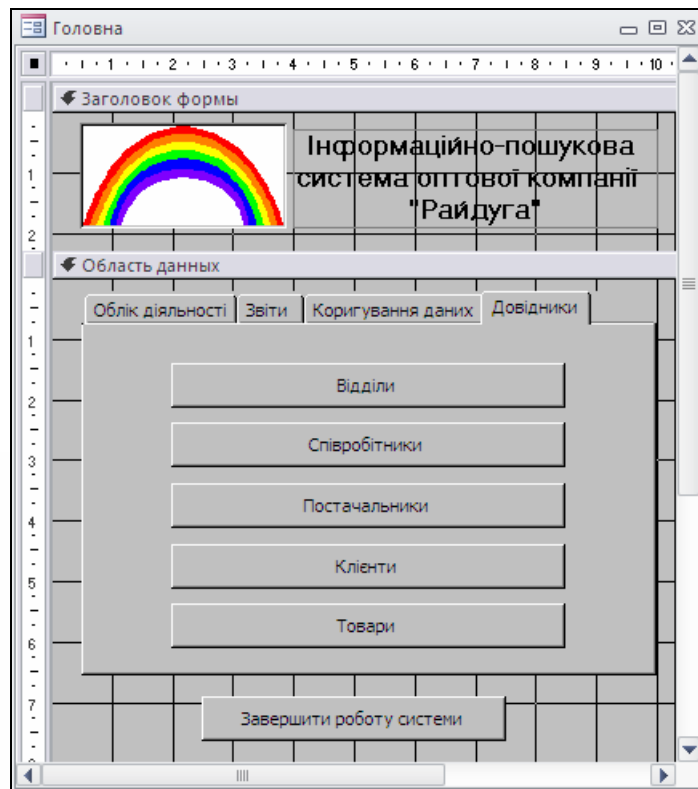


Рис. 31. Приклад власної головної кнопкової форми в режимі конструктора

Якщо ж розробляється стандартна кнопкова форма, то необхідно пам'ятати, що в головному вікні диспетчера кнопкових форм виводяться наявні сторінки кнопкової форми. Дані цього вікна при експлуатації стандартної кнопкової форми не відображається, тому й послідовність сторінок у ньому не має значення. Серед цих сторінок необхідно обрати сторінку по замовчуванню, яка буде відобразитися першою при завантаженні стандартної кнопкової форми.

Диспетчер кнопкових форм автоматично створює стандартну кнопку форму *Кнопочная форма*, яка відображає розроблені сторінки, і таблицю *Switchboard Items*, де зберігаються параметри сторінок стандартної кнопкової форми. Якщо знищити цю таблицю, то диспетчер створить нову при наступному завантаженні. Коли ж знищити дану форму, то для її повторного створення необхідно попередньо перейменувати ще й таблицю *Switchboard Items*, а після завантаження і закриття диспетчера – повернути їй попередню назву, знищивши перед цим новостворену порожню таблицю з тією ж назвою.

Після створення головної кнопкової форми необхідно забезпечити її відкриття при завантаженні додатку. Для цього на вкладці *Файл* ► *Параметри* ► *Текущая база данных* у полі зі списком *Форма просмотра* вибирають назву цієї форми.

Література: [6, С. 353-355, 382-383; 1, С. 267-268, 414-420]

### Підготовчий етап заняття. Актуалізація знань

1. Завантажте Access, відкрийте з власної папки розроблену раніше БД *Sklad*.

2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип об'єкта*.
3. Перейдіть в області переходів до розділу *Форми*.

### Корегування властивостей створених форм та елементів керування в режимі конструктора

4. Створіть форми згідно наведених зразків (рис. 32-34). При відсутності потрібних даних для форми самостійно змініть структуру базової таблиці та задайте додаткові властивості для створених полів. Під час створення форм:
  - 4.1. Для вирівнювання розмірів та відносних положень елементів керування використайте можливості кнопок групи *Размер и порядок* вкладки стрічки меню *Упорядочить*;
  - 4.2. Для всіх простих форм приховайте область виділення та розмежовуючі лінії між розділами, встановивши у вікні властивостей форм на вкладці *Макет* значення *Нет* для властивостей *Область выделения* та *Разделительные линии*.

Назва відділу	Місце розташування відділу
Збугу	Корпус №1
Кадрів	Адміністративна будівля
Бухгалтерія	Адміністративна будівля
Технічний	Корпус №1
Харчоблок	Корпус №2

Рис. 32. Форма *Відділи* в режимі форми

Рис. 33. Форма *Постачальники* в режимі форми


Рис. 34. Форма *Товари* в режимі форми

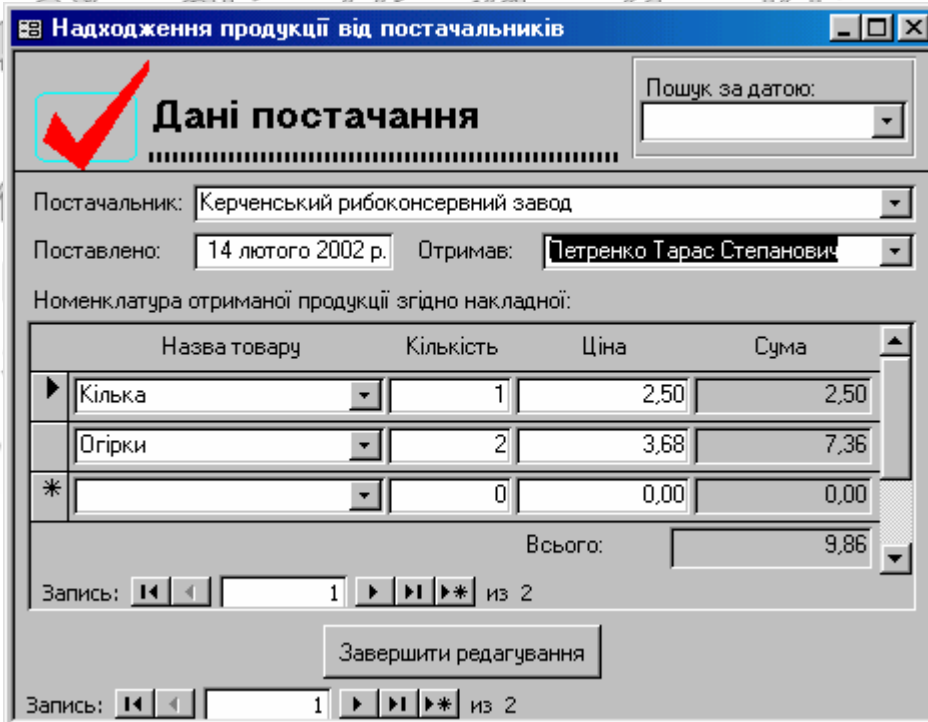
5. Відкоригуйте форму *Клієнти* згідно наведеного зразка (рис. 35). Для організації початку та продовження пошуку в активному полі створіть у формі дві відповідні кнопки. При цьому:
  - 5.1. При розробці першої кнопки на першому кроці майстра оберіть категорію дії *Переходи по записам* та дію *Найти запись*. Завершіть створення кнопки самостійно;
  - 5.2. Розробляючи другу кнопку, оберіть в тій самій категорії дію *Найти далее* та завершіть створення кнопки.

Рис. 35. Вигляд форми *Клієнти* в режимі форми

### Створення обчислювальних полів в формах

6. Відкоригуйте форми *ЗаголовкиПостачань* та *ЗаголовкиЗамовлень* разом з підпорядкованими формами згідно наведених зразків (рис. 36-37). У підпорядкованих формах *ПунктиПостачаньПП* та *ПунктиЗамовленьПП* створіть обчислювальні поля для підрахунку суми кожної позиції. Для цього в режимі конструктора кожної з них:

- 6.1. Натягніть у відведеному місці згідно зразка вільне поле при допомозі кнопки  панелі елементів;
  - 6.2. У вікні властивостей створеного поля на вкладці *Данные* задайте значення параметра *Данные* формулою  $=\text{Кількість} * \text{Ціна}$ ;
  - 6.3. Забороніть редагування створеного поля. Для цього встановіть у вікні його властивостей на вкладці *Данные* у властивості *Доступ* значення *Нет* а у властивості *Блокировка* – значення *Да*;
  - 6.4. Для створення візуального ефекту заборони редагування цього поля встановіть у вікні його властивостей на вкладці *Макет* у властивості *Тип фона* значення *Прозрачный*;
  - 6.5. З метою забезпечення коректного сприйняття вмісту поля встановіть для нього на тій самій вкладці у властивості *Формат поля* значення *С разделителями разрядов*, а у списку *Число десятичных знаков* оберіть значення 2.
7. В області приміток підпорядкованих форм *ПунктиПостачаньПП* та *ПунктиЗамовленьПП* створіть обчислювальні поля для підрахунку загальної суми постачання чи замовлення відповідно. Для цього після створення полів задайте формулу для підрахунку їх значень  $=\text{Sum}(\text{Кількість} * \text{Ціна})$ . Інші властивості створених обчислювальних полів задайте аналогічно полям для підрахунку суми кожної позиції.



Надходження продукції від постачальників

**Дані постачання**

Пошук за датою:

Постачальник: Керченський рибоконсервний завод

Поставлено: 14 лютого 2002 р. Отримав: Петренко Тарас Степанович

Номенклатура отриманої продукції згідно накладної:

Назва товару	Кількість	Ціна	Сума
Кілька	1	2,50	2,50
Огірки	2	3,68	7,36
*	0	0,00	0,00
Всього:			9,86

Запись: 1 из 2

Завершити редагування

Запись: 1 из 2

Рис. 36. Форми *ЗаголовкиПостачань* та *ПунктиПостачаньПП* в режимі форми

Рис. 37. Форми *ЗаголовкиЗамовлень* та *ПунктиЗамовленьПП* в режимі форми

### Створення власної головної кнопкової форми

- В режимі конструктора створіть форму *Головна* згідно зразка, наведеного на рис. 38. Для цього, обираючи на першому кроці майстра створення кнопок категорію дії *Робота с формой* і дію *Открыть форму*, а на другому – назву потрібної форми, створіть у формі наступні кнопки:

Рис. 38. Форма *Головна* в режимі форми



- 8.1. На вкладці *Облік діяльності* – кнопки *Облік постачань* (відкриває форму *ЗаголовкиПостачань*) та *Облік замовлень* (відкриває форму *ЗаголовкиЗамовлень*);
- 8.2. На вкладці *Довідники* – кнопки *Відділи*, *Співробітники*, *Постачальники*, *Клієнти*, *Товари* (див. рис. 31). Кожна з цих кнопок повинна відкривати форму, що відповідає її назві.
9. В нижній частині форми створіть кнопку *Завершити роботу системи*, що закриває базу даних разом з Access. Для цього на першому кроці майстра створення кнопок слід в категорії *Приложение* обрати вид дії *Выйти из приложения*.
10. Забезпечте автоматичне завантаження форми *Головна* при відкритті БД. Для цього у вікні *Файл – Параметри* перейдіть на вкладку *Текущая база данных* та у полі зі списком *Форма просмотра* виберіть назву форми *Головна*.
11. Закрийте та відкрийте знову БД *Sklad*. Чому при відкритті БД відразу завантажується форма *Головна*? З довідкової системи Access дізнайтеся про призначення інших елементів розділу *Текущая база данных* з вікна *Файл – Параметри*.
12. Використовуючи можливості вікна диспетчера кнопок форм, створіть стандартну кнопку форму системи:
- 12.1. Для відображення кнопки виклику диспетчера кнопок форм у вікні *Файл – Параметри* перейдіть до розділу *Настройка ленты*. Створіть на вкладці *Работа с базами данных* групу *Дополково*, віднайдіть серед всіх команд позицію *Диспетчер кнопочных форм* та додайте цю команду у створену групу. Закрийте вікно параметрів Access зі збереженням змін;
- 12.2. Віднайдіть нову кнопку *Диспетчер кнопочных форм* на вкладці *Работа с базами данных* стрічки меню. Завантажте з її допомогою диспетчер кнопок форм;
- 12.3. Створіть у вікні диспетчера кнопок форм за допомогою кнопки *Создать* три сторінки: *Головне меню системи*, *Облік діяльності* та *Довідники* (рис. 39).

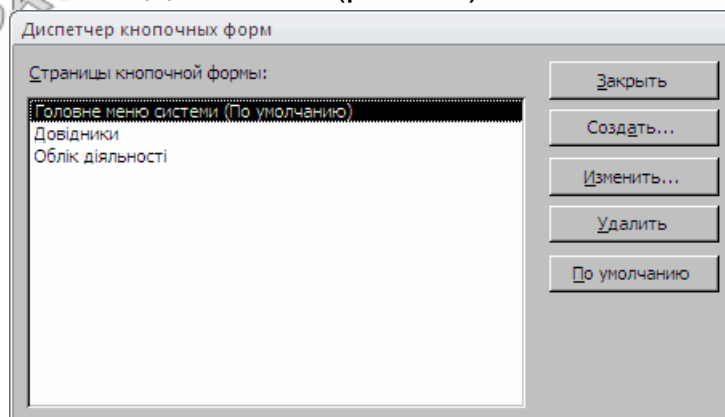


Рис. 39. Перелік створених сторінок диспетчера кнопок форм



Встановіть активною по замовчуванню сторінку *Головне меню системи*. Саме ця сторінка буде з'являтися при відкритті стандартної кнопкової форми;

- 12.4. Створіть кнопки сторінки *Головне меню системи*. Для цього у вікні диспетчера кнопкових форм оберіть назву цієї сторінки та натисніть кнопку *Изменить*. Створіть на виведеній сторінці за допомогою кнопки *Создать* три кнопки (рис. 40): перша – для переходу до сторінки *Облік діяльності* (рис. 41), друга – для переходу до сторінки *Довідники* (аналогічно першій), третя – для виходу з системи (рис. 42). Зверніть увагу на команди для кнопок, які при цьому обираються. Ознайомтеся з іншими командами, які можуть призначатися кнопкам. Завершіть редагування цієї сторінки, натиснувши кнопку *Закричь* та поверніться до головної сторінки диспетчера кнопкових форм;

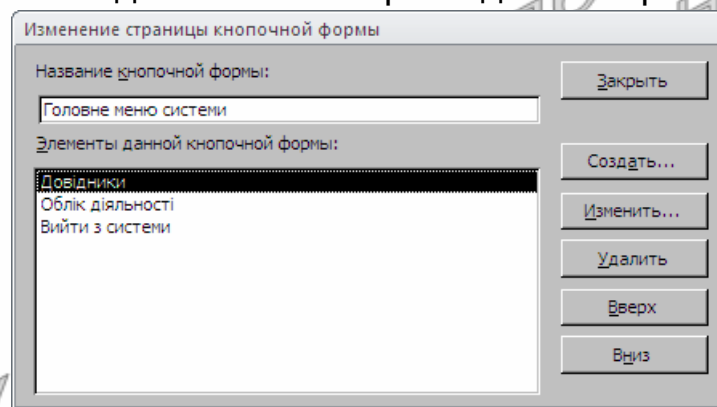


Рис. 40. Кнопки сторінки *Головне меню системи* стандартної кнопкової форми

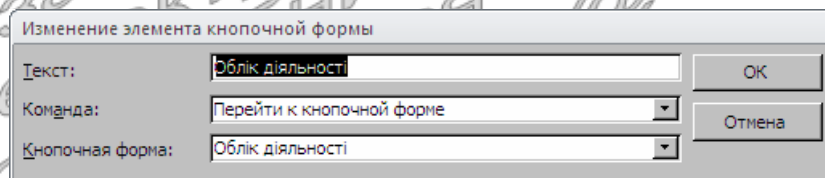


Рис. 41. Створення кнопки для переходу з сторінки по замовчуванню до сторінки *Облік діяльності* в стандартній кнопковій формі

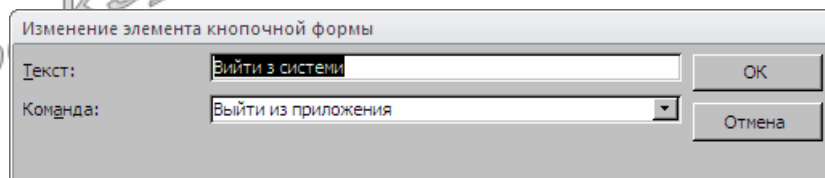


Рис. 42. Розробка кнопки стандартної кнопкової форми для виходу з додатку

- 12.5. Аналогічно створіть три кнопки сторінки *Облік діяльності* (рис. 43): перша – для відкриття форми *ЗаголовкиПостачань* (рис. 44), друга – для відкриття форми *ЗаголовкиЗамовлень*, третя – для повернення до сторінки *Головне меню системи* (рис. 45);

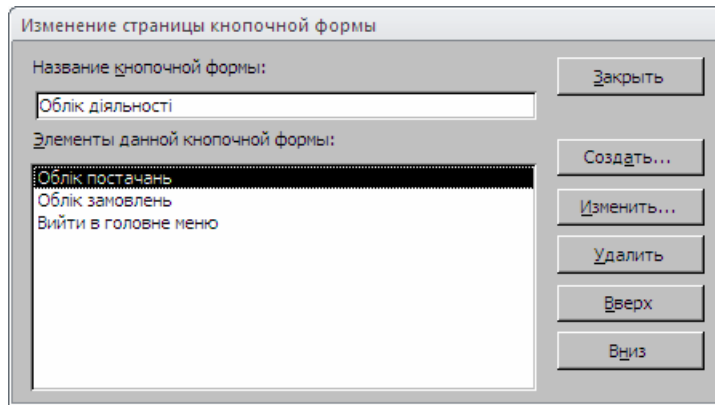


Рис. 43. Кнопки сторінки *Облік діяльності* стандартної кнопочної форми

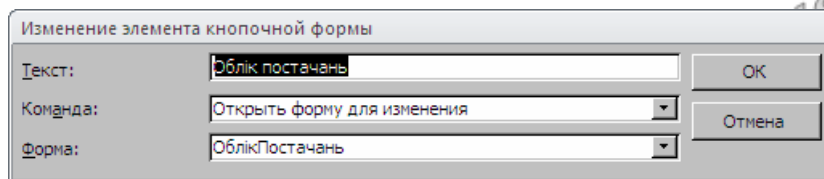


Рис. 44. Створення кнопки для виклику форми *ОблікПоставчань* з сторінки *Облік діяльності* стандартної кнопочної форми

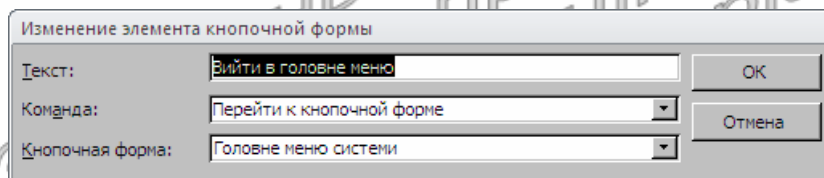


Рис. 45. Розробка кнопки для повернення до сторінки *Головне меню системи* з сторінки *Облік діяльності* стандартної кнопочної форми

12.6. Самостійно створіть у сторінці *Довідники* стандартної кнопочної форми п'ять кнопок для відкриття форм довідників і шосту кнопку – для повернення до сторінки *Головне меню системи* (рис. 46);

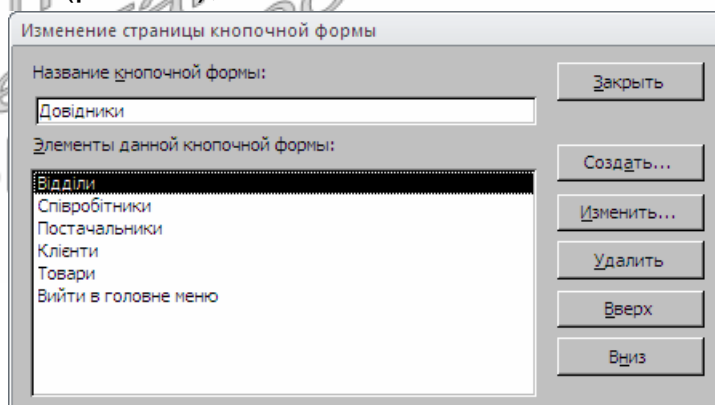


Рис. 46. Кнопки сторінки *Довідники* стандартної кнопочної форми

12.7. Закрийте диспетчер головної кнопочної форми. Визначте, яка нова форма та таблиця з'явилися в системі. Перевірте функціональність стандартної кнопочної форми. Як забезпечити відкриття саме цієї форми при завантаженні

системи? Як ще раз створити стандартну головну кнопку форму, якщо вона була випадково видалена?

### Повторення вивченого матеріалу

13. Стисніть вдосконалену БД.
14. Завершіть роботу Access.
15. Віднайдіть створену БД та заархівуйте її вміст.
16. Збережіть створений архів на власному носії в двох екземплярах.
17. Розробіть власну і стандартну кнопку форми у власній БД. Оснастіть, де це доцільно, форми для редагування з власної БД обчислювальними елементами керування. Продемонструйте розроблені БД викладачу.

### Контрольні запитання

1. Навіщо при оформленні засобів для пошуку в активному полі форми створюють дві кнопки?
2. Чому у формах *ЗаголовкиПостачань* та *ЗаголовкиЗамовлень* недоцільно створювати засоби для пошуку в активному полі форми?
3. Де задається формула для обчислювального елемента керування?
4. Обґрунтуйте формули створених обчислювальних полів в формах. За якою ознакою Access відрізняє обчислювальні поля від звичайних?
5. Чому при введенні формул для обчислювальних елементів керування назви полів базового джерела даних не бралися в квадратні дужки? Коли слід обов'язково брати назви полів в квадратні дужки?
6. В якому режимі створюється власна, а в якому – стандартна головна кнопка форми? Як забезпечити автоматичне відкриття цих форм при завантаженні БД?
7. Які вкладки доцільно створювати у власній головній кнопковій формі?
8. Де зберігаються параметри стандартної головної кнопкової форми?

## Лабораторна робота № 9

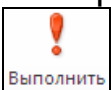
**Тема.** Розробка та використання запитів на вибірку без групування записів.

**Мета.** Формування вмінь та навичок створення запитів на вибірку без групування записів в режимі конструктора для комплексного аналізу даних таблиць. Усвідомлення послідовності етапів виконання запитів на вибірку даних. Вдосконалення вмінь створення та корегування об'єктів додатків СУБД MS Access.

### Теоретичні відомості

Запити дають змогу відбирати, групувати, сортувати, корегувати записи однієї чи одночасно декількох таблиць, виконувати обчислення як в межах запису, так і для кожної групи, а також ефективно обробляти дані в мережі. Запит може обробляти дані таблиць чи інших запитів. Але запити – це лише команди для обробки даних. В них не зберігаються дані записів. Створюються ці команди в режимах конструктора чи SQL. При перегляді ж запиту в режимі таблиці згідно його команди відображаються в табличному вигляді або результати виконання аналізу даних або записи, над якими **буде виконуватися** корегування. Ці тимчасові таблиці не зберігаються у файлах БД, а лише відображаються на екрані, тому й називаються **динамічними наборами даних**. Експлуатувати ці набори можна майже так само, як і таблиці, але при цьому потрібно чітко усвідомлювати, що внесені зміни зберігаються не в запиті, а в базовій таблиці, і тому стають доступними іншим користувачам в мережі.

Формувати більшість запитів в Access доцільно в режимі конструктора, де використовується стандарт **QBE** (Query by Example – запити по зразку). У цьому режимі, використовуючи інтерактивні засоби та регіональні національні стандарти, користувач формує бланк, де вказує, які поля необхідно опрацювати. Розробити ж будь-який запит можна в режимі SQL по однойменному стандарту. Тут команда кожного запиту записується за регіональним американським стандартом текстовими реченнями, тому запити в режимі SQL створюються і відлагоджуються довше, ніж в режимі конструктора, зате дозволяють задати оптимальніші умови відбору. Для виконання запиту необхідно завантажити його з вікна БД або в режимі конструктора у вкладці стрічки меню *Конструктор* натиснути кнопку



Наголосимо, що виконання запитів відбувається шляхом послідовної реалізації їх окремих речень згідно стандарту SQL, тому запити, створені по стандарту QBE в режимі конструктора перед виконанням всерівно перетворюються до стандарту SQL. І навпаки, більшість запитів, створених в режимі SQL, можна відкрити для

подальшого корегування в режимі конструктора. Всі запити виконуються **пакетно**, тобто СУБД, реалізуючи кожен запит, або повністю опрацює дані згідно його команди або, при неможливості виконання перерве обробку і всі задіяні таблиці поверне до початкового стану.

За призначенням запити Access поділяються на дві групи – запити **аналізу даних** та **запити на корегування**. В режимі таблиці для запитів аналізу даних фактично відображається результат їх виконання, а для запитів на корегування – записи, над якими буде здійснено дію. До першої групи запитів належать:

- **запити на вибірку** – найуживаніші запити, які використовуються для відбору, групування, сортування записів джерела та виконання обчислень над його даними (наприклад, підрахунок загальних сум оформлених замовлень по кожному співробітнику за минулий квартал);
- **перехресні запити** – виконують групування відібраних записів як по рядках, так і по стовпцях та проводять розрахунки над відповідними даними на їх перетині (наприклад, розрахунок сум замовлень (дані на перетині) по кожному товару (заголовки рядків) за кожен місяць (заголовки стовпців) в поточному році (умова відбору));
- **запити на поєднання** – здійснюють поєднання результатів відбору декількох запитів на вибірку. Такі запити можливо розробити лише в режимі SQL, оскільки вони обробляють дані декількох джерел;
- **запити до сервера** – виконують аналіз даних таблиць, розміщених на віддаленому сервері. Створюються лише в режимі SQL. Дані сервера мають бути попередньо записані в джерелах ODBC панелі керування операційної системи.

До групи запитів на корегування належать:

- **запити на оновлення** – змінюють дані, які містяться в записах таблиць з збереженням цілісності;
- **запити на додавання** – використовуються для пакетного створення нових записів в обраній таблиці. При цьому перевіряється цілісність не лише задіяних полів, а й даних записів в цілому;
- **запити на видалення** – вилучають записи, що відповідають критеріям відбору, з обраної таблиці. При встановленій властивості каскадного видалення для зв'язків з обраною таблицею на стороні 1 автоматично вилучаються також пов'язані записи в таблицях на стороні ∞;
- **запити на створення нових таблиць** – автоматично зберігають дані, відібрані командою кожного запиту, в новій таблиці. Якщо назва нової таблиці співпадає з назвою існуючої таблиці чи запиту, то такий об'єкт замінюється створюваною таблицею після отримання згоди від користувача;



➤ **запити керування** – використовуються для створення чи видалення таблиць або для зміни їх структури чи індексів. Створюються лише в режимі SQL.

У цій лабораторній ми розглянемо лише найпростіші запити на вибірку без групування записів, а в наступних – складніші запити.

Виконання запиту на вибірку відбувається в такій послідовності:

- формується джерело даних (динамічна таблиця з даних базових таблиць);
- відбираються записи з сформованого джерела;
- виконується групування відібраних записів і обчислення в групах;
- здійснюється відбір з сформованих груп;
- виконується сортування результатів відбору;
- формуються значення полів і виводяться результати дії запиту.

Обов'язковими з цих етапів є лише перший та останній. Наявність решти етапів залежить від команди запиту.

Для створення команди запиту на вибірку в режимі конструктора потрібно на стрічці меню натиснути кнопку **Создание** ► **Конструктор запросов**, після чого на екрані з'являється порожній бланк запиту і вікно для формування джерела (рис. 47). Верхня частина вікна запиту в режимі конструктора називається **областю джерела даних**, а нижня – **бланком запиту**.

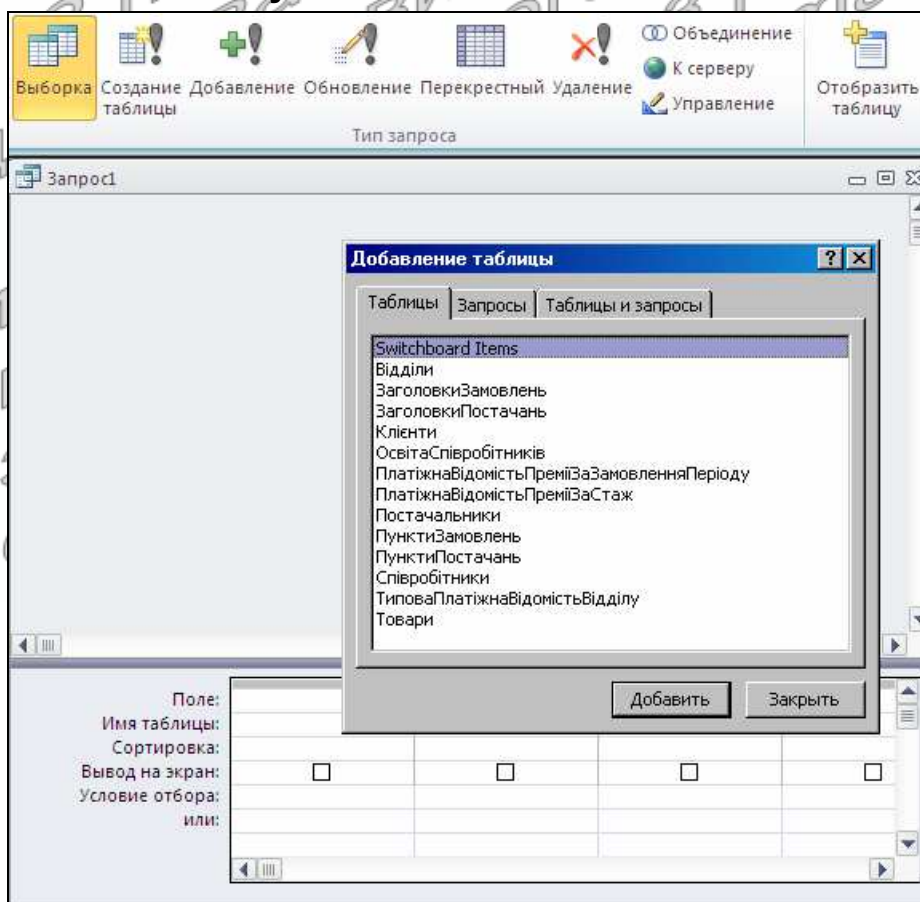


Рис. 47. Порожній бланк запиту на вибірку та вікно для доповнення джерела новими таблицями чи запитамі в режимі конструктора



По замовчуванню в режимі конструктора створюється саме запит на вибірку. Інший тип запиту обирається натисненням відповідної кнопки на вкладці стрічки меню *Конструктор* (верхня частина рис. 47).

Для формування джерела даних у вікні *Добавление таблицы* необхідно відмітити потрібні таблиці чи запити і натиснути кнопку *Добавить*. Це саме вікно можна надалі викликати з контекстного меню вільного місця області джерела даних або кнопкою стрічки меню *Отобразить таблицу* (див. рис. 47). Початкові зв'язки між таблицями джерела формуються з схеми даних, але надалі розробник в режимі конструктора запиту може встановити нові чи знищити існуючі зв'язки і ці зміни ніяк не впливають на схему даних (і це логічно, адже запитальні зв'язки не повинні впливати на структурні зв'язки інфологічної моделі). Наголосимо, що в джерелі даних не слід використовувати зайві таблиці чи запити, адже їх наявність може в рази збільшити кількість записів джерела даних і, як наслідок, спотворити результати виконання запиту. Знищення образу таблиці в області джерела даних запиту не призводить до її вилучення з схеми даних, чи, тим більше, до знищення самої таблиці. В області джерела даних режиму конструктора відображається структура обраних таблиць чи запитів, тобто їх поля, оскільки тут формується команда обробки даних (див. рис. 47).

Бланк запиту складається з таких рядків:


➤ *Поле* – містить назви полів з таблиць чи запитів джерела або вирази для обчислень значень, які використовуються запитом. Для вказування поля з джерела можна використати один з таких способів: перетягнути ім'я поля з області джерела даних в бланк; обрати чи набрати це ім'я безпосередньо в рядку *поле*; двічі натиснути ЛКМ або клавішу *Enter* на імені поля в джерелі (при цьому поле перенесеться у вільний стовбець бланку запиту). Крім цього, для вибору всіх полів таблиці чи запиту (наприклад, для формування джерела форми чи звіту) достатньо перетягнути в бланк \* з її образу. Тоді навіть перейменування чи створення нового поля у базовій таблиці не призведе до втрати дієздатності запиту.

Для формування ж виразу не достатньо вказати лише формулу його обчислення за національним стандартом. Справа в тому, що кожен вираз у бланку повинен мати унікальне ім'я (псевдонім), яке в подальшому використовується для звертання до значень його поля. Ім'я записується перед виразом і відділяється від нього двокрапкою (наприклад, *Сума: Кількість\*Ціна*). Якщо розробник не вказує ім'я виразу, то Access автоматично генерує імена типу *Выраженный1*, *Выражение2* і т.д. У полі двокрапка двічі використовуватися не може, тобто кожен стовбець запиту може мати лише один заголовок;

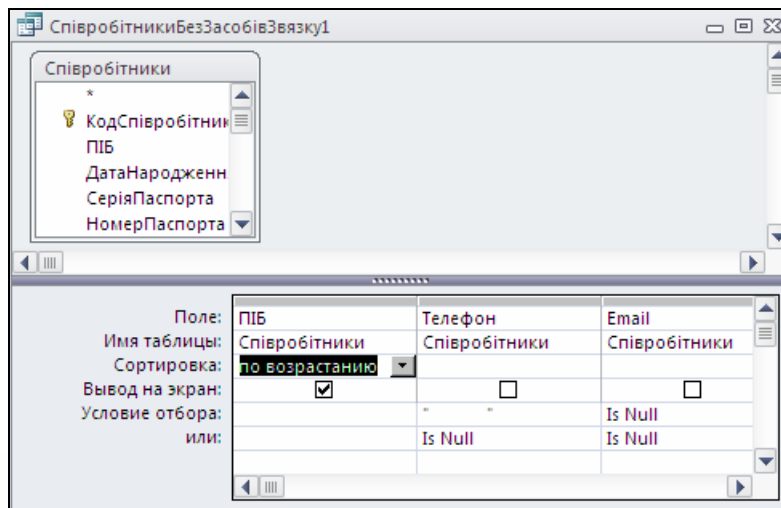
➤ *Имя таблицы* – вказує, з якої таблиці береться *Поле*. Обов'язково зазначається лише у випадках, коли ім'я поля присутнє у декількох

таблицях чи запитах, які формують джерело даних і не зазначається для виразів;

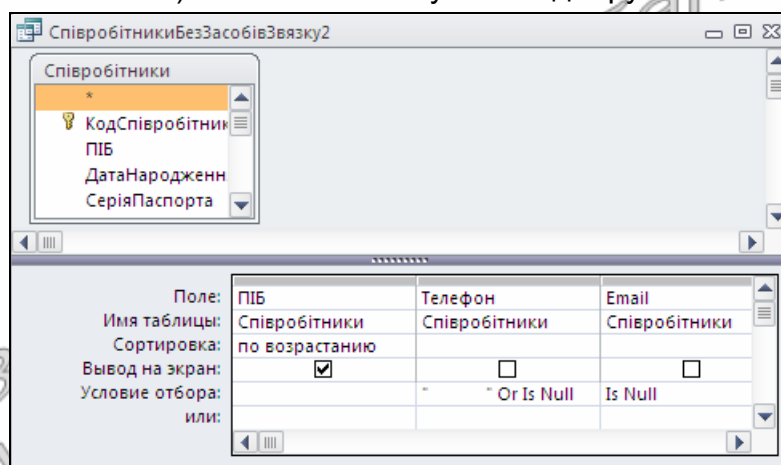
- *Сортировка* – забезпечує впорядкування записів (а не тільки значень поля), сформованих запитом. Це поле зі списком може набувати одного з трьох значень: *по возрастанию*, *по убыванию* та *отсутствует* (порожнє, застосовується по замовчуванню). Найважливішим є перше зліва поле сортування. Саме за ним впорядковуються записи динамічної таблиці результатів дії запиту. І лише ті записи, у яких значення першого поля сортування однакове, впорядковуються між собою за другим полем сортування і т.д.;
- *Групповая операция* – використовується для групування відібраних записів та виконання обчислень в групах. З'являється в бланку лише після натиснення в його контекстному меню чи на вкладці

*Конструктор* стрічки меню кнопки :

- *Вывод на экран* – прапорець, що встановлюється для полів, які мають відобразитися при виконанні запиту. Запит на вибірку має виводити хоча б одне поле;
- *Условие отбора / или* – містять умови для відбору записів з джерела чи з сформованих груп. Умови, записані в одному рядку, мають для відібраних записів чи груп виконуватися одночасно (поєднуються логічним сполучником **And**). У випадку ж вказування умов в різних рядках (застосовуючи рядки бланку *или*) для відбору запису джерела чи групи достатньо виконання умов хоча б з одного рядка. Тобто такі умови поєднуються логічним сполучником **Or**. Отже, запис чи група відбираються запитом тоді і лише тоді, коли для нього виконуються **всі умови хоча б одного** з рядків умов відбору. Наприклад, для відбору співробітників без зазначеного номера телефону та адреси електронної пошти необхідно врахувати, що телефон може бути не вказаний у двох випадках: або це поле ще не заповнювалося, тоді у ньому буде зберігатися значення по замовчуванню (12 пробілів, тобто " "), або телефон ввели, а потім видалили (тоді значення поля буде порожнім, тобто *Is Null*). Для кожного з цих випадків, згідно з умовою запиту, адреса електронної пошти має бути порожньою (*Is Null*, що відповідає як значенню по замовчуванню, так і значенню поля після видалення). Вигляд режиму конструктора цього варіанту запиту наведено на рис. 48а. Зверніть увагу, що поля *Телефон* та *Email* не виводяться на екран, оскільки при виконанні в них не відображаються дані (слід завжди дотримуватися цього підходу, щоб не відволікати увагу користувача на такі поля). Оптимізувати умови відбору цього запиту можливо, явно використавши логічний оператор *Or* для поля *Телефон* (рис. рис. 48б). Тоді при виконанні запиту для кожного запису джерела буде виконуватися максимум 3, а не 4 операції порівняння (Чому?).



а) неоптимізовані умови відбору



б) оптимізовані умови відбору

Рис. 48. Два варіанти структури запиту *СпівробітникиБезЗасобівЗв'язку* в режимі конструктора

Для відбору записів, що відповідають різним умовам, без внесення змін в структуру запиту в режимі конструктора чи SQL в умовах запиту використовують **параметри** (наприклад, замість конкретного номера місяця доцільно ввести текст *[Введіть номер місяця для аналізу даних]*. Чому текст параметра береться в квадратні дужки?). При виконанні запиту Access не віднаходить кожен параметр в джерелі даних і виводить вікно для введення його значення від користувача (ось чому поява зайвих вікон для введення даних при виконанні запиту найчастіше свідчить про неправильне введення імені поля). Безпосередній відбір записів виконується лише після введення всіх параметрів. Для кожного параметра можна забезпечити контроль типу даних, як це описано в лабораторній роботі, щоб уникнути аварійного завершення виконання запиту.

Надалі, описуючи синтаксис запитів і елементів виразів, в < > будемо позначати місця для ідентифікаторів та виразів користувача, в [ ] – необов'язкові елементи, а через | будемо перераховувати елементи, серед яких користувач має обрати лише один.

В режимі SQL запити на вибірку записуються згідно синтаксису:

```
SELECT <поля результатів відбору>  
FROM <джерело даних>  
[WHERE <умови відбору записів>]  
[GROUP BY <поля групування>]  
[HAVING <умови відбору груп>]  
[ORDER BY <поля сортування>];
```

Бачимо, що обов'язковими речення в інструкції відбору є лише FROM та SELECT, які відповідають першому та останньому етапу виконання запиту на вибірку. Окремі речення в середині інструкції вибору, як правило, набираються з нового рядка, хоча й можуть вводитися через пробіл після попереднього речення. В режимі SQL ім'я таблиці вказується перед іменем її поля і відділяється від нього крапкою. Ідентифікатори та вирази користувача у кожному з речень перераховуються через кому. Завершується кожна інструкція в режимі SQL крапкою з комою.

Таблиці, запити і зв'язки між ними, які формують джерело даних і в режимі конструктора відображаються в його верхній частині, в режимі SQL описуються після FROM. Якщо джерело містить лише одну таблицю чи запит, то після FROM вказується лише його назва (наприклад, FROM Відділи), якщо ж більше – то крім назв базових об'єктів ще й описуються зв'язки між ними.

Після слова SELECT перераховуються лише ті поля, що виводяться при виконанні запиту (для них в режимі конструктора встановлюється прапорець *Вивод на екран*). Якщо формується вираз, то його ім'я вказується після формули виразу і відділяється від нього сполучником AS. Наприклад, SELECT ЗаголовкиЗамовлень . ДатаЗамовлення, Кількість\*Ціна AS Сума.

Окремі умови відбору записів, які в режимі конструктора записуються в рядку *Умовие отбора* та нижчих і можуть міститися у різних стовпцях, в режимі SQL послідовно вказуються після WHERE, але поєднуються між собою сполучниками AND чи OR (Коли і як саме?).

Як і в режимі конструктора (рядок *Сортировка*), найважливішим полем сортування в режимі SQL є перше поле, яке зазначається відразу після ORDER BY, адже саме за ним впорядковуються записи. Лише ті записи, в яких значення першого поля сортування однакові, між собою впорядковуються за другим полем сортування і т.д. По замовчуванню сортування виконується за зростанням значення поля. Для явного сортування за зростанням після виразу поля вказують ASC. При необхідності ж сортування за спаданням після виразу поля вказують DESC. Наприклад, ORDER BY Відділи.НазваВідділу DESC, Співробітники.ПІБ.

Особливості формування та відбору груп розглянемо в наступній лабораторній роботі. Тут же висвітлимо принципи формування виразів, які застосовуються у всіх частинах запиту для виконання обчислень.

## Конструювання виразів в Access

**Вирази** – це конструкції, що містить хоча б один оператор і хоча б одну константу, ідентифікатор чи функцію.

Вирази використовують при конструюванні умов відбору в запитах, під час формування обчислювальних полів, в мовах програмування для виконання обчислень і формування умов для операторів. Запис виразу залежить від того, в якому режимі він створюється, оскільки в режимі конструктора діють національні стандарти, а в режимі SQL – американський. При переході з режиму в режим зміна синтаксису виразу відбувається автоматично.

Основні правила запису **констант**:

- рядкові константи беруться в подвійні лапки, наприклад, "Факультет";
- дати записуються між символами #, причому в режимі конструктора – у форматі #dd.mm.yyyy#, а в режимі SQL – #mm.dd.yyyy#;
- ціла частина числа відмежується від дробової в режимі конструктора комою, а в режимі SQL – крапкою. Вона може містити мантису, яка записується після E.

**Ідентифікатори** – це імена об'єктів, які повертають активне значення. Наприклад, для запитів ідентифікаторами є імена полів з джерела даних. Якщо назва ідентифікатора містить декілька слів, то вони беруться в квадратні дужки. Для уникнення помилок назви ідентифікаторів рекомендується перетягувати, вибирати зі списку або вводити за допомогою будівничого виразів.

**Функції** – це послідовності команд, які повертають обчислення значення в місце виклику. В Access після назви функції вказують дужки, у яких при потребі перераховують її аргументи. Аргументи функцій відділяються між собою в режимі конструктора ;, а в SQL – ,. Різновиди функцій:

**а) функції обробки дат:**

- **Date()** – повертає активну дату в місце виклику;
- **Now()** – повертає активний момент часу;
- **Day(<дата>)** – повертає число дня в місяці від вказаної дати;
- **Month(<дата>)** – повертає місяць від значення аргумента;
- **Year(<дата>)** – повертає номер року від вказаної дати;
- **Weekday(<дата>)** – повертає номер дня в тижні від зазначеної дати;
- **Hour(<дата>)** – повертає годину дати в місце виклику;

**б) функції обробки рядків:**

- **Len(<рядок>)** – обчислює кількість символів у рядку;
- **Left(<рядок>; <кількість>)** – вирізає з рядка зліва вказану кількість символів;
- **Right(<рядок>; <кількість>)** – вирізає з рядка справа вказану кількість символів;
- **Mid(<рядок>; <позиція>; <кількість>)** – вирізає з рядка, починаючи з вказаної позиції, необхідну кількість символів;
- **Format(<змінна>; <шаблон>)** – форматує значення змінної по заданому шаблону;
- **InStr(<рядок>; <шаблон>)** – виконує пошук в рядку тексту згідно шаблону і повертає першу позицію входження (при наявності);

**в) Функції перетворення типів:**

- **CBool(<значення>)** – перетворює значення до логічного типу, якщо це можливо;
- **CStr(<значення>)** – перетворює значення до тексту;
- **Cdbl(<значення>)** – переводить значення до дійсного числа;
- **CLng(<значення>)** – перетворює значення до довгого цілого числа;
- **CCur(<значення>)** – перетворює значення до грошового типу;
- **CDate(<значення>)** – перетворює значення до дати;
- **Nz(<вираз>; <значення>)** – якщо вираз непорожній, то повертає його значення, інакше повертає значення другого аргумента;

#### г) математичні функції:

- **Int**(<значення>) – повертає цілу частину числа. Наприклад, вираз для визначення кількості років, які пройшли від дати влаштування співробітника на роботу:  $Int((Date() - [Дата влаштування]) / 365,25)$ ;
- **Round**(<значення>; <точність>) – заокруглює число до вказаної у точності кількості знаків. Наприклад,  $Round(Оклад/27,2;2)$ ;

#### д) функції змішаного типу:

- **IIf**(<умова>; <значення1>; <значення2>) – ця функція перевіряє істинність умови. Якщо умова істинна, то повертається *значення1*, інакше повертається *значення2*. Наприклад, для обчислення суми прибуткового податку, якщо до 340 грн. він не нараховується, а після 340 складає 19,5% від суми, що перевищує 340 грн. зазначають:  $IIf(Оклад \leq 340; 0; 0,195 * (Оклад - 340))$ ;
- **IsNull**(<значення>) – повертає *істинно*, якщо значення порожнє;
- **IsNumeric**(<значення>) – повертає *істинно*, якщо значення можна перетворити до числа;
- **IsDate**(<значення>) – повертає *істинно*, якщо значення можна перетворити до дати.

**е) Статистичні функції SQL** – це функції, які в режимі конструктора вибирають у рядку групової операції. Вони детально розглядаються в наступній лабораторній роботі;

**ж) Статистичні функції обробки підмножин** – це функції, що використовуються для обробки записів зовнішнього джерела. Всі функції цієї групи мають 3 аргументи: перший вказує, яке значення повертає функція; другий – описує джерело даних; третій – задає умови відбору записів джерела. Наприклад, функція **DLookup** повертає перше значення, яке відповідає умовам відбору. Назви інших функцій, обробки підмножин аналогічні функціям SQL, лише спереду додається буква D.

**Оператори**, що використовуються у виразах:

- а) **арифметичні**: +; -; \*; /; **mod** – остача від ділення, \ - ціла частина від ділення, ^ - степінь;
- б) **порівняння**: (<, ≤, >, ≥, =, <>);
- в) **логічні**: **NOT** – заперечення, **AND**, **OR**, **XOR** – оператор істинний, коли істинні обидва операнди;
- г) **злиття рядкових величин (конкатенації)**: &;
- д) **ідентифікації**: ! – використовують для відмежування класу об'єкта від самого об'єкта і об'єкта від вкладеного об'єкта); . – відмежовує стандартне поле запису від назви таблиці з джерела та стандартну властивість від самого об'єкта. В Access використовують такі типи об'єктів: Tables (таблиці); Queryes (запити); Forms (форми); Reports (звіти).

#### з) Інші оператори:

- **IS** – використовують для перевірки того, чи є поле порожнім чи ні (в поєднанні з *Null*, а саме *Is Null* або *Is Not Null*);
- **LIKE** <шаблон> – встановлює відповідність поля шаблону. В шаблоні можуть використовуватися символи \* (довільна кількість символів) та ? (один довільний символ);
- **IN** (<список>) – перевіряє належність поля до множини, елементи списку перераховують як аргументи функцій – в режимі конструктора через ;, а в SQL – через ,;
- **BETWEEN** <нижня межа> **AND** <верхня межа> – перевіряє на належність до діапазону. Межі в діапазон включаються, одна з меж може не вказуватися, але вживання **AND** обов'язкове.

Література: [4, С. 108-114, 120-121; 1, С. 157-167, 172-174]



### Підготовчий етап заняття. Актуалізація знань

1. Завантажте Access, відкрийте розроблену раніше БД *Sklad*.
2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип объекта*.
3. Перейдіть в області переходів до розділу *Запросы*. По можливості, для прискорення формування виразів, роздрукуйте дві попередні сторінки теоретичних відомостей.

### Сортування та відбір даних за допомогою запитів

4. Створіть запит *АлфавітнийСписокСпівробітників* в режимі конструктора для формування алфавітного списку співробітників з зазначенням дати народження. Для цього:
  - 4.1. Перейдіть в режим конструктора для створення запиту, натиснувши на вкладці стрічки меню *Создание* у групі *Запросы* кнопку *Конструктор запросов*;
  - 4.2. Для формування джерела даних запиту у вікні *Добавление таблицы* виділіть таблицю *Співробітники* та натисніть кнопку *Добавить*. Закрийте вікно *Добавление таблицы*;
  - 4.3. З метою формування списку співробітників в перший стовпець бланку запиту внесіть поле *ПІБ* одним з двох способів: перетягніть поле при натиснутій лівій кнопці мишки з образу таблиці *Співробітники* у верхній частині вікна в бланк запиту у нижній його частині або виберіть назву поля зі списку в рядку *Поле* першого стовпця бланку запиту;
  - 4.4. В другий стовпець бланку запиту внесіть одним з двох описаних вище способів поле *ДатаНародження*;
  - 4.5. В рядку *Сортировка* для поля *ПІБ* оберіть зі списку значення *по возрастанию* для відповідного впорядкування списку співробітників при виконанні запиту;
  - 4.6. Закрийте вікно конструктора та збережіть запит під назвою *АлфавітнийСписокСпівробітників*.
5. Виділіть створений запит та завантажте його на виконання різними способами:
  - 5.1. Подвійним натисненням лівої кнопки мишки по назві запиту;
  - 5.2. Натисненням клавіші *Enter* після виділення назви запиту;
  - 5.3. За допомогою пункту *Открыть* контекстного меню запиту;
  - 5.4. Перейдіть в режим конструктора, обравши в контекстному меню запиту пункт *Конструктор*, та оберіть, перебуваючи у ньому, у списку кнопки *Режимы* пункт *Режим таблицы* (цей спосіб рекомендується використовувати надалі);
  - 5.5. Перейдіть в режим конструктора та натисніть на вкладці інструментів стрічки меню кнопку *Выполнить*.
6. Перегляньте текст створеного запиту в режимі SQL. Для цього перейдіть в режим конструктора та виберіть у списку кнопки *Режимы* пункт *режим SQL*. Обгрунтуйте структуру тексту запиту.

7. Завантажте запит в режимі конструктора та з'ясуйте призначення всіх пунктів списку кнопки *Режими*.
8. Самостійно створіть аналогічні запити:
  - 8.1. *АлфавітнийСписокВідділів* – для відображення в алфавітному порядку назв відділів з зазначенням місць розташування;
  - 8.2. *АлфавітнийСписокКлієнтів* – для відображення в алфавітному порядку назв клієнтів з зазначенням дати реєстрації та місця роботи;
  - 8.3. *АлфавітнийСписокПостачальників* – для відображення в алфавітному порядку назв постачальників з зазначенням коду ЄДРПОУ та дати реєстрації;
  - 8.4. *АлфавітнийСписокТоварів* – для відображення назв товарів в алфавітному порядку з зазначенням марки, одиниці виміру та виробника.

### Створення обчислювальних полів в запитах

9. Доповніть бланк запиту *АлфавітнийСписокСпівробітників* полем для зазначення його віку. Для цього:
  - 9.1. Відкрийте обраний запит в режимі конструктора;
  - 9.2. В рядку *Поле* третього стовпця бланку запиту введіть формулу *Date()-ДатаНародження* для підрахунку кількості прожитих днів;
  - 9.3. Для перевірки дієздатності запити завантажте його в режимі таблиці. Знову поверніться в режим конструктора. Чому перед формулою з'явився підпис *Выражение 1*? Яку роль він відіграє?
  - 9.4. Відредагуйте текст третього стовпця до вигляду *Вік: (Date()-ДатаНародження)/365,25* (оскільки середня тривалість року складає 365,25 дня). Перевірте функціональність запити;
  - 9.5. Для відкидання дробової частини відредагуйте текст третього стовпця до вигляду *Вік: Int((Date()-ДатаНародження)/365,25)*. Перевірте функціональність запити;
  - 9.6. Закрийте запит та збережіть внесені зміни.
10. Створіть запит *ДніНародженняСпівробітників* з переліком прізвищ співробітників, місяців і днів їх народження та кількості років, що виповнилися (виповняться) співробітнику в цьому році. Для цього:
  - 10.1. Розпочніть створення запити в режимі конструктора;
  - 10.2. Сформуєте джерело даних запити з записів таблиці *Співробітники*;
  - 10.3. Перетягніть в перший стовпець бланку запити поле *ПІБ*;
  - 10.4. В другий стовпець запити введіть вираз *Місяць народження: Month(ДатаНародження)*;
  - 10.5. Самостійно сформуєте вираз третього стовпця для виводу на екран числа дня народження співробітника;
  - 10.6. В четвертий стовпець запити введіть вираз *Виповниться (виповнилося): Year(Date())-Year(ДатаНародження)*;

- 10.7. Самостійно встановіть ознаки сортування за сукупним зростанням значень полів *Місяць народження* та *День народження*;
- 10.8. Збережіть запит та переконайтеся в його дієздатності.
11. Самостійно доповніть запит *АлфавітнийСписокТоварів* полем *Мінімальна дата виготовлення* для відображення дати виготовлення товару, починаючи з якої він ще є придатним до вживання (значення цього поля дорівнює різниці поточної дати та терміну придатності товару в днях).
12. Самостійно створіть запит *ОкладиСпівробітників* з переліком прізвищ співробітників, окладу в гривнях, доларах (рівний окладу в гривнях, поділеному на поточний курс долара) та євро. Відсортуйте його записи за спаданням значення окладу в гривнях.

### **Конструювання умов відбору записів в запитах**

13. Створіть запит *ЖінкиВідділів* для відображення прізвищ жінок, що працюють у кожному відділі. Для цього:
  - 13.1. Розпочніть створення запиту в режимі конструктора;
  - 13.2. Сформууйте джерело даних запиту з записів таблиць *Відділи* та *Співробітники*;
  - 13.3. Перетягніть в перший стовпець бланку запиту поле *НазваВідділу* з таблиці *Відділи*;
  - 13.4. Перетягніть в другий стовпець бланку запиту поле *ПІБ* з таблиці *Співробітники*;
  - 13.5. Для виводу на екран лише співробітників-жінок перетягніть в третій стовпець бланку запиту поле *Стать* з таблиці *Співробітники*, зніміть для нього прапорець *Вывод на экран* та вкажіть в рядку *Условие отбора* значення *0*;
  - 13.6. Оскільки працюючі співробітники – це, насамперед, ті, для яких не встановлено дату звільнення, то для виводу на екран при виконанні запиту лише працюючих співробітників перетягніть в четвертий стовпець бланку запиту поле *ДатаЗвільнення* з таблиці *Співробітники*, зніміть для нього прапорець *Вывод на экран* та введіть в рядок *Условие отбора* вираз *Is Null*;
  - 13.7. Для першого і другого стовпця встановіть ознаку сортування за зростанням;
  - 13.8. Збережіть запит та переконайтеся в його дієздатності.
14. Аналогічно забезпечте відображення в запитах *АлфавітнийСписокСпівробітників*, *ДніНародженняСпівробітників*, *ОкладиСпівробітників* даних лише працюючих співробітників.
15. Самостійно створіть наступні запити:
  - 15.1. *СпівробітникиБезТелефонів* – для відображення алфавітного списку співробітників без вказаних номерів телефонів з зазначенням відділу (не введений номер телефону

позначається дванадцятьма пробілами (чому?), а видалений – *Is Null*, тому загальна умова відбору для цього поля має вигляд " " *Or Is Null*);

15.2. *СпівробітникиЗEmail* – для відображення алфавітного списку співробітників з зазначенням відділу, що мають непорожні адреси електронної пошти.

16. Створіть запит *ПрацюючіПенсіонери* для відображення алфавітного списку працюючих пенсіонерів кожного відділу (пенсіонери – це жінки, старші 60 років та чоловіки, старші 62 років). Для цього:

16.1. Щоб уникнути повторного введення формули для обчислення віку співробітника, скопіюйте в буфер обміну запит *АлфавітнийСписокСпівробітників* та вставте його під назвою *ПрацюючіПенсіонери*;

16.2. Відкрийте скопійований запит в режимі конструктора та перетягніть в четвертий стовпець бланку поле *Стать*, а в п'ятий – поле *ДатаЗвільнення*;

16.3. Для відбору пенсіонерів–жінок внесіть в рядок *Условие отбора* для стовпця *Вік* вираз  $\geq 60$ , а для стовпця *Стать* – значення 0;

16.4. Для відбору пенсіонерів–чоловіків внесіть в рядок *или* для стовпця *Вік* вираз  $\geq 62$ , а для стовпця *Стать* – значення -1;

16.5. Для відбору лише працюючих пенсіонерів внесіть в рядки *Условие отбора* та *или* для стовпця *ДатаЗвільнення* вираз *Is Null*. Самостійно забезпечте відсутність відображення на екрані даного стовпця при виконання запиту;

16.6. Збережіть створений запит та перевірте його дієздатність.

17. Створіть запит *ОсвітченіСпівробітники* для відображення алфавітного списку співробітників з зазначенням відділу, посади та окладу, що мають середню спеціальну, незакінчену вищу або вищу освіту. Для цього:

17.1. Розпочніть створення запиту в режимі конструктора;

17.2. Сформууйте джерело даних запиту з записів таблиць *Відділи*, *Співробітники* та *ОсвітаСпівробітників*;

17.3. Перетягніть в перший стовпець бланку запиту поле *ПІБ* з таблиці *Співробітники*;

17.4. Перетягніть в другий стовпець бланку запиту поле *НазваВідділу* з таблиці *Відділи*;

17.5. Перетягніть в третій та четвертий стовпці бланку запиту відповідно поля *Посада* та *Оклад* з таблиці *Співробітники*;

17.6. Для виводу на екран лише співробітників, що мають середню спеціальну, незакінчену вищу або вищу освіту виконайте наступні дії:

17.6.1. Перетягніть в п'ятий стовпець бланку запиту поле *ТипОсвіти* з таблиці *ОсвітаСпівробітників*;

- 17.6.2. Оскільки для поля *ТипОсвіти* задано підстановку значень з фіксованого набору так, що відображуваному тексту *середня спеціальна* відповідає значення поля 3, тексту *незакінчена вища* – значення 4, тексту *вища* – значення 5, то для відбору записів джерела даних запиту лише з вказаними типами освіти задайте для даного поля в рядку *Условие отбора* значення 3, в рядку *или* – значення 4, а в наступному рядку *или* – значення 5;
- 17.7. Для виводу на екран при виконанні запиту лише працюючих співробітників перетягніть в шостий стовпець бланку запиту поле *ДатаЗвільнення* з таблиці *Співробітники*, зніміть для нього прапорець *Вывод на экран* та введіть в рядок *Условие отбора* вираз *Is Null*;
- 17.8. Для першого і другого стовпця встановіть ознаку сортування за зростанням;
- 17.9. Збережіть запит та переконайтеся в його дієздатності. Чому окремі прізвища співробітників виведено декілька разів?
- 17.10. Перегляньте текст створеного запиту в режимі SQL. Як в цьому режимі записано умову відбору записів джерела даних запиту?
- 17.11. Поверніться в режим конструктора. Для стовпця *ТипОсвіти* в рядок *Условие отбора* введіть еквівалентну до попередньої умову відбору записів *>=3 and <=5* та знищіть умови відбору в нижчих рядках;
- 17.12. Завантажте відредагований запит в режимі таблиці. Чому на екран при зміненому варіанті умови відбору записів виведено аналогічні дані?
- 17.13. Перегляньте текст відредагованого запиту в режимі SQL. Як змінився текст умови відбору записів джерела даних запиту?
- 17.14. Поверніться в режим конструктора. Для стовпця *ТипОсвіти* в рядок *Условие отбора* введіть еквівалентну до попередньої умову відбору записів *between 3 and 5*;
- 17.15. Завантажте відредагований запит в режимі таблиці та проаналізуйте отримані дані;
- 17.16. Перегляньте текст відредагованого запиту в режимі SQL. Як змінився текст умови відбору записів джерела даних запиту?
- 17.17. Теоретично проаналізуйте та обґрунтуйте у звіті швидкість виконання трьох варіантів запиту.
18. Самостійно створіть наступні запити:
- 18.1. *СпівробітникиЗВищоюОсвітою* – для відображення алфавітного списку працюючих співробітників з зазначенням відділу, що здобули вищу освіту;

- 18.2. *СпівробітникиБезЗасобівЗв'язку* – для відображення алфавітного списку працюючих співробітників з зазначенням відділу без введених номерів телефонів та адрес електронної пошти;
- 18.3. *СтажРоботиВТоваристві* – для відображення списку працюючих співробітників з зазначенням відділу та кількості відпрацьованих років в товаристві відносно активної дати. Відсортуйте виведені записи за спаданням кількості відпрацьованих років.

### **Створення запитів з параметрами відбору записів**

19. Створіть запит *ДаніПостачаньТоварівЗаПеріод* для відображення алфавітного списку назв та марок отриманих товарів за введений період з зазначенням відповідної назви постачальника, дати постачання, кількості, ціни та суми. Для цього:
  - 19.1. Розпочніть створення запиту в режимі конструктора;
  - 19.2. Сформууйте джерело даних запиту з записів таблиць *Постачальники*, *ЗаголовкиПостачань*, *ПунктиПостачань* та *Товари*;
  - 19.3. Перетягніть в перший та другий стовпці бланку запиту відповідно поля *НазваТовару* та *МаркаТовару* з таблиці *Товари*;
  - 19.4. Перетягніть в третій стовпець бланку запиту поле *ДатаПостачання* з таблиці *ЗаголовкиПостачань*;
  - 19.5. Перетягніть в четвертий стовпець бланку запиту поле *ПІБ* з таблиці *Постачальники*;
  - 19.6. Для першого, другого та третього стовпця встановіть ознаку сортування за зростанням;
  - 19.7. Для відбору даних постачань за 2016 рік введіть в рядку *Условие отбора* для поля *ДатаПостачання* вираз *between #01.01.2016# and #31.12.2016#*;
  - 19.8. Збережіть запит та переконайтеся в його дієздатності;
  - 19.9. Для відбору даних постачань за довільний проміжок часу поверніться в режим конструктора та змініть в рядку *Условие отбора* для поля *ДатаПостачання* умову відбору на *between [Введіть початкову дату] and [Введіть кінцеву дату]*;
  - 19.10. Завантажте запит на виконання, ввівши у першому вікні початкову, а в другому – кінцеву дати.
  - 19.11. Для забезпечення коректності введених параметрів задайте для них тип даних. Для цього відкрийте вікно *Параметры запроса* за допомогою кнопки *Параметры* вкладки інструментів стрічки меню або за допомогою аналогічного пункту контекстного меню вільного поля області джерела даних запиту, заповніть виведену таблицю згідно табл. 18 та натисніть *OK*.



Типи даних для параметрів запиту *ДаніПостачаньТоварівЗаПеріод*

Параметр	Тип даних
[Введіть початкову дату]	Дата/время
[Введіть кінцеву дату]	Дата/время

- 19.12. Завантажте запит на виконання та переконайтеся у функціонуванні перевірки введених параметрів на відповідність вказаним типам даних.
20. Самостійно доповніть створений запит полями для виводу на екран кількості, ціни та суми поставленого товару.
21. Самостійно створіть запит *ДаніЗамовленьТоварівЗаПеріод* для відображення алфавітного списку назв та марок замовлених товарів з зазначенням відповідної дати замовлення, назви клієнта, кількості, ціни та суми.

**Завершальний етап заняття. Повторення вивченого матеріалу**

22. Стисніть створену БД.
23. Завершіть роботу Access.
24. Віднайдіть створену БД та заархівуйте її вміст.
25. Збережіть створений архів на власному носії в двох екземплярах.
26. Розробіть аналогічні запити до запитів цієї лабораторної роботи у власній БД. Продемонструйте розроблені БД викладачу.

**Контрольні запитання**

1. Що спільного між стандартами мов запитів QBE та SQL? Чим вони відрізняються? Які регіональні стандарти у них застосовуються?
2. Яка різниця між дією запиту та операцією пошуку?
3. Як перейти в режим конструктора для створення нового запиту?
4. Які поля відображаються в результаті виконання запиту? Де записуються назви цих полів в режимі SQL?
5. Як здійснити сортування полів при виконанні запиту? Як сортуються дані при вказуванні ознаки сортування для багатьох полів? Де записуються назви цих полів в режимі SQL?
6. Як здійснити відбір записів джерела даних при виконанні запиту? Де описуються критерії відбору записів в режимі SQL?
7. Як оформляються константи різних типів даних при конструюванні умов відбору?
8. Як створити обчислювальне поле в запиті? Значення яких полів може використовуватися при формуванні обчислювального поля? Коли слід створювати обчислювальне поле, а коли – задавати параметри відбору записів джерела даних?
9. Для чого використовуються параметри при виконанні запиту? Чому текст параметра з декількох слів необхідно брати в квадратні дужки?
10. Як забезпечити перевірку типу даних для параметра запиту? Навіщо це робити? Коли виконується така перевірка?

## Лабораторна робота № 10

**Тема.** Створення і використання запитів на вибірку з групуванням записів та перехресних запитів. Використання параметрів поєднань таблиць та властивостей запитів для ефективного аналізу даних.

**Мета.** Формування вмінь та навичок створення запитів на вибірку з групуванням даних, перехресних запитів, корегування параметрів об'єднань таблиць та властивостей запитів. Закріплення вмінь і навичок створення та корегування запитів в режимі конструктора. Систематизація та застосування знань варіантів поєднань таблиць БД при конструюванні запитів. Усвідомлення ролі зв'язків між таблицями при формуванні джерела даних запиту.

### Теоретичні відомості

Розглянемо детальніше механізм формування джерела даних запиту. Єдина динамічна таблиця джерела даних містить всі поля з базових таблиць чи запитів (назвемо їх базовими таблицями, оскільки обробка в запиті іншого запиту нічим не відрізняється від обробки таблиці). Зрозуміло, що якщо запит обробляє одну базову таблицю, то джерело даних буде співпадати з нею. Якщо ж джерело містить більше базових таблиць, то його записи формуються згідно встановлених **поєднань**. Поєднання між базовими таблицями джерела даних найчастіше реалізуються за допомогою зв'язків. **Кожен зв'язок між базовими таблицями фактично вказує на поєднання записів, в яких значення пов'язаних полів співпадають** (як і в схемі даних БД). Крім цього, на кількість записів джерела впливають встановлені **параметри поєднань**. Дослідимо роль зв'язків та параметрів поєднань при формуванні джерела на прикладі двох базових таблиць для обліку даних студентів та їх пропусків (рис. 49).

The image shows two screenshots of a database interface. The left screenshot shows a table named 'Студенти' with columns 'КодСтудента' and 'ПІБ'. The right screenshot shows a table named 'Пропуски' with columns 'КодПропу', 'КодСтудента', 'ДатаПропуска', and 'НомерПари'.

КодСтудента	ПІБ
1	Семешук
2	Бондарець
3	Собчук
4	Козачок
*	(№)

КодПропу	КодСтудента	ДатаПропуска	НомерПари
1	2	19.09.2016	1
2	2	19.09.2016	2
3	3	23.09.2016	1
*	(№)		

Рис. 49. Дані базових таблиць *Студенти* та *Пропуски* для формування джерела даних запиту

Дані пропусків студентів зберігаються в окремій таблиці, оскільки один студент може мати багато пропусків (саме тому між цими таблицями

встановлено структурний зв'язок по полях *КодСтудента*, які перебувають у співвідношенні 1:∞). Бачимо (див. рис. 49), що студенти з кодами 1 і 4 взагалі не мають пропусків, студент з кодом 2 має два пропуски, а студент з кодом 3 має один пропуск. Отож, в Access застосовується чотири типи параметрів поєднань:

➤ **поєднання тільки тих записів, в яких пов'язані поля обох базових таблиць співпадають.** Цей тип поєднання використовується на практиці найчастіше. Якщо поєднати базові таблиці, наведені на рис. 49, по співпадаючих значеннях полів *КодСтудента*, то отримаємо джерело, наведене на рис. 50. Зверніть увагу, що, по-перше, дані студентів з кодами 1 і 4 не ввійшли в джерело, оскільки для них немає відповідних записів у другій таблиці і тому ці дані не будуть виведені запитом; по-друге, дані студента з кодом 2 ввійшли в джерело двічі, оскільки за ним зафіксовано два пропуски; по-третє, для ідентифікації полів з однаковими іменами, які входять в різні базові таблиці, перед цими іменами вказується назва базової таблиці, яка відділяється крапкою.

Студенти.КодСтудє	ПІБ	КодПропуска	Пропуски.КодСту	ДатаПропус	НомерПари
2	Бондарець	1	2	19.09.2016	1
2	Бондарець	2	2	19.09.2016	2
3	Собчук	3	3	23.09.2016	1

Рис. 50. Джерело даних з таблиць *Студенти* та *Пропуски*, поєднаних по співпадаючих значенням поля *КодСтудента*

Якщо ж поєднати ці самі базові таблиці по співпадаючих значенням поля *КодСтудента* з першої таблиці і *НомерПари* з другої таблиці (а це можна зробити, наприклад, в режимі конструктора перетягуванням між полями, як в схемі даних), то отримаємо джерело, наведене на рис. 51. Тобто ті самі базові таблиці з тими самими параметрами поєднання, але іншими зв'язками можуть породити відмінне джерело даних.

Студенти.КодСтудє	ПІБ	КодПропуска	Пропуски.КодСту	ДатаПропус	НомерПари
1	Семешук	1	2	19.09.2016	1
2	Бондарець	2	2	19.09.2016	2
1	Семешук	3	3	23.09.2016	1

Рис. 51. Джерело даних з таблиць *Студенти* та *Пропуски*, поєднаних по співпадаючих значенням поля *КодСтудента* з таблиці *Студенти* і *НомерПари* з таблиці *Пропуски*

В режимі конструктора цей параметр поєднання обирається для зв'язків по замовчуванню і виглядає як ламана, проведена між полями базових таблиць (див., наприклад, рис. 11). Для перетворення зв'язків інших типів до даного типу необхідно виділити зв'язок і в його контекстному меню обрати пункт *Параметри об'єднання*, після чого в однойменному вікні, що з'явиться на екрані (як на рис. 52), обрати перший перемикач. Між собою можуть бути пов'язані по декілька полі з кожної базової таблиці, але всі зв'язки між довільними парами таблиць мають бути одного типу.

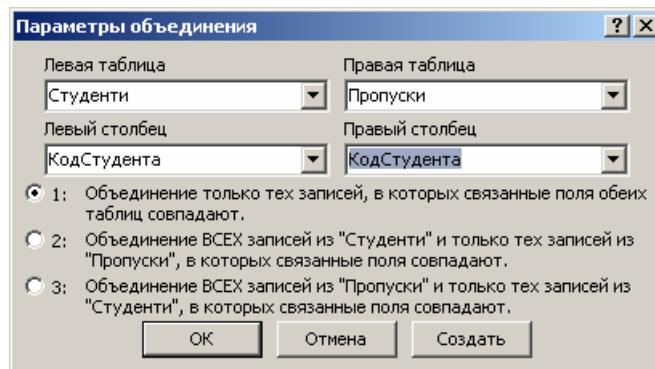


Рис. 52. Вікно вибору параметра поєднання в режимі конструктора

В режимі SQL джерело даних з двома базовими таблицями та зв'язком між ними записується згідно синтаксису

FROM <Таблиця 1> INNER JOIN | LEFT JOIN | RIGHT JOIN  
 <Таблиця 2> ON <поле з Таблиці 1>=<поле з Таблиці 2> .

Зокрема, якщо застосовується описуване поєднання записів по співпадаючих значеннях полів обох базових таблиць, то між іменами цих таблиць використовується оператор INNER JOIN. В стандартах SQL такий тип поєднання називається **внутрішнім**. Наприклад, джерело рис. 50 в режимі SQL має такий вигляд:

FROM Студенти INNER JOIN Пропуски ON Студенти.КодСтудента =  
 Пропуски.КодСтудента .

При необхідності опису джерела більш ніж з двох базових таблиць спочатку у круглих дужках записується поєднання перших двох таблиць, яке надалі сприймається як єдина таблиця, після чого дописується поєднання до третьої таблиці і т.д. Наприклад:

FROM (Студенти INNER JOIN Пропуски ON Студенти.  
 КодСтудента = Пропуски.КодСтудента) INNER JOIN  
 ПричиниПропусків ON Пропуски.КП=ПричиниПропусків.КП ;

- **поєднання ВСІХ записів з першої таблиці і тільки тих записів з другої таблиці, в яких пов'язані поля співпадають.** У випадку використання цього параметра поєднання в джерело даних запиту крім записів, утворених поєднанням записів базових таблиць, в яких значення пов'язаних полів співпадають, увійдуть ще й записи першої таблиці, для яких відсутні відповідні записи у другій таблиці. При цьому в джерелі даних для таких додаткових записів у всіх полях другої таблиці буде встановлено *Null*.

Студенти.КодСтудента	ПІБ	КодПропуска	Пропуски.КодСтудента	ДатаПропуску	НомерПари
1	Семешук				
2	Бондарець	1	2	19.09.2016	1
2	Бондарець	2	2	19.09.2016	2
3	Собчук	3	3	23.09.2016	1
4	Козачок				

Рис. 53. Джерело даних з усіх записів таблиці *Студенти* і тільки тих записів таблиці *Пропуски*, в яких значення полів *КодСтудента* співпадають





другої до першої базової таблиці, а для задання такого параметра поєднання в режимі SQL необхідно між назвами таблиць використати оператор RIGHT JOIN;

- **поєднання кожного запису першої таблиці з кожним записом другої таблиці** (Декартів добуток записів). При використанні цього параметра поєднання в режимі конструктора зв'язки між базовими таблицями не встановлюються (саме тому додаткові непов'язані таблиці в джерелі призводять до створення додаткових записів), а в режимі SQL назви базових таблиць перераховуються через кому і поля з двох таблиць не порівнюються між собою (наприклад, FROM Студенти, Пропуски). Якщо в першій таблиці міститься  $N$ , а в другій –  $M$  записів, то в джерелі буде сформовано  $M*N$  записів (як на рис. 55).

Студенти.КодСтудє	ПІБ	КодПропуска	Пропуски.КодСту	ДатаПропус	НомерПари
1	Семещук	1	2	19.09.2016	1
1	Семещук	2	2	19.09.2016	2
1	Семещук	3	3	23.09.2016	1
2	Бондарець	1	2	19.09.2016	1
2	Бондарець	2	2	19.09.2016	2
2	Бондарець	3	3	23.09.2016	1
3	Собчук	1	2	19.09.2016	1
3	Собчук	2	2	19.09.2016	2
3	Собчук	3	3	23.09.2016	1
4	Козачок	1	2	19.09.2016	1
4	Козачок	2	2	19.09.2016	2
4	Козачок	3	3	23.09.2016	1

Рис. 55. Джерело даних при поєднанні кожного запису таблиці *Студенти* з кожним записом таблиці *Пропуски*

В стандарті SQL використовується також оператор повного зовнішнього поєднання FULL JOIN, який повертає поєднання записів, в яких пов'язані поля співпадають, записи з першої базової таблиці без відповідних записів у другій і записи другої таблиці без пов'язаних записів у першій, але цей параметр поєднання в Access не реалізований.

Розглянемо тепер особливості створення і використання запитів на вибірку з групуванням записів. Як вже зазначалося в попередній лабораторній роботі, для забезпечення групування відібраних записів і виконання обчислень у групах в режимі конструктора необхідно натиснути кнопку *Ітоги*. Після цього в бланку запиту з'явиться додатковий рядок *Групповая операція*, в якому для кожного поля можна обрати одне з значень:

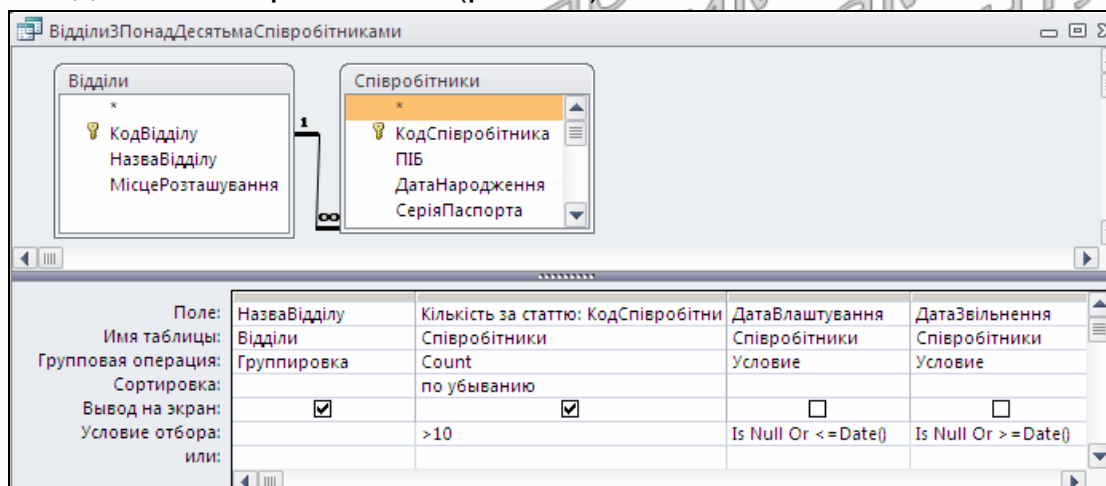
- **Группировка** – використовується для групування відібраних записів. Якщо це значення встановлюється для декількох полів, то групи створюються згідно унікальних комбінацій їх значень. Коли ж це значення не встановлено для жодного поля, то всі записи джерела буде віднесено до однієї групи. Поля, для яких в режимі



конструктора обрана *Групування*, в режимі SQL послідовно перераховуються в реченні GROUP BY. Решта з перерахованих нижче значень списку, крім останнього, - це статистичні функції SQL, що застосовуються до значення поля;

- **Sum** – обчислює суму значень поля в кожній групі;
- **Count** – обчислює кількість непорожніх значень поля в кожній групі;
- **Min** – обчислює мінімальне значення в кожній групі;
- **Max** – обчислює максимальне значення поля в кожній групі;
- **Avg** – обчислює середнє арифметичне в кожній групі;
- **Var** – обчислює дисперсію в кожній групі;
- **First** – повертає перше значення з джерела в кожній групі;
- **Last** - повертає останнє значення з джерела в кожній групі;
- **умовие** – містить умову відбору окремих записів з джерела і тому не виводиться на екран.

Створимо, наприклад, запит для відбору відділів, в яких працює більше десяти співробітників (рис. 56).



```
SELECT Відділи.НазваВідділу,
Count(Співробітники.КодСпівробітника) AS [Кількість за статтю]
FROM Відділи INNER JOIN Співробітники ON Відділи.КодВідділу =
Співробітники.КодВідділу
WHERE (Співробітники.ДатаВлаштування Is Null Or
Співробітники.ДатаВлаштування<=Date()) AND (Співробітники.
ДатаЗвільнення Is Null Or Співробітники.ДатаЗвільнення>=Date())
GROUP BY Відділи.НазваВідділу
HAVING (((Count(Співробітники.КодСпівробітника))>10))
ORDER BY Count(Співробітники.КодСпівробітника) DESC;
```

Рис. 56. Запит *ВідділиЗПонадДесятьмаСпівробітниками* в режимах конструктора та SQL

Зрозуміло, що джерелом даних цього запиту будуть таблиці *Відділи* та *Співробітники*, поєднані по співпадаючих значенням полів *КодВідділу* (перший етап виконання). У такому джерелі назва кожного відділу буде повторюватися стільки разів, скільки є записів, пов'язаних з цим відділом. На другому етапі виконання запиту необхідно серед всіх співробітників відібрати працюючих зараз. На третьому – згрупувати відібраних співробітників по групах і в кожній групі

підрахувати кількість співробітників (групова операція *Count*). На четвертому етапі виконання потрібно відібрати групи (відділи), в які входить більше десяти записів (співробітників), на п'ятому – відсортувати сформовані групи за спаданням кількості працюючих співробітників, а на останньому – вивести результати відбору на екран.

Бачимо, що умови відбору груп в режимі конструктора задаються у тих самих рядках, що й умови відбору записів, хоча другі діють до, а перші – після групування записів. В режимі ж SQL умови відбору записів перераховуються в реченні *WHERE*, а груп – в реченні *HAVING*.

Ще одним типом запитів, які виконують групування даних, є перехресні запити, які групують записи джерела не тільки по рядках, а й по стовпцях. Майстер створення перехресних запитів для прискорення цього процесу дає змогу послідовно обрати джерело, поля групування по рядках, поле (може бути лише одне) та функцію групування по стовпцях, поле та групову функцію для обчислень в середині перехресної таблиці. Якщо у перехресному запиті необхідно проаналізувати дані декількох таблиць, то з цих таблиць необхідно попередньо створити проміжний запит-джерело, в якому вивести не менше трьох полів (Чому?). Міжнародний стандарт SQL на такі запити не затверджений, тому обмежимося розглядом їх структури в режимі конструктора (як на рис. 57).

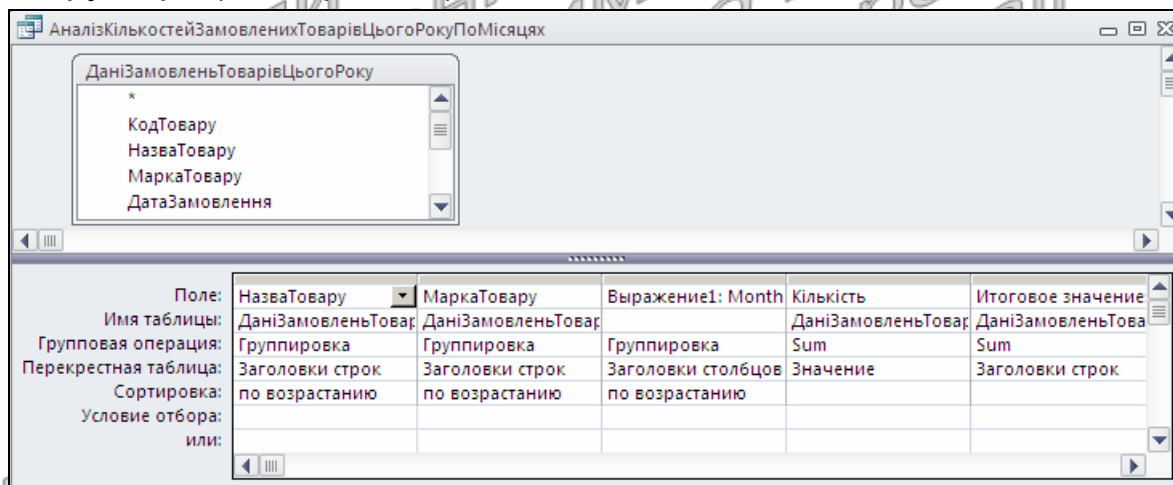


Рис. 57. Перехресний запит

*АналізКількостейЗамовленихТоварівЦьогоРокуПоМісяцях* в режимі конструктора Бачимо, що в цьому режимі у бланку перехресного запиту відображається додатковий рядок *Перекрестная таблица*. У ньому розробник для кожного поля може обрати одне з чотирьох значень: *Заголовки строк*, *Заголовки столбцов* (лише для одного поля), *Значение* (вираз для обчислення на перетині рядків і стовпців, може бути лише один), *не отображается* (містить умови для відбору записів з джерела). Наприклад, для аналізу загальних кількостей замовлень кожного товару у цьому році по місяцях доцільно дані замовлень цього року згрупувати у рядках по окремих товарах, а в стовпцях – по місяцях, підрахувавши на їх перетині відповідні загальні замовлені кількості (див. рис. 57). Звернемо увагу, що в режимі

конструктора по кожному рядку перехресного запиту можна виконати обчислення декількох підсумків, але для виразу такого підсумку у рядку *Перекрестная таблица* необхідно встановити значення *Заголовки строк* (див., наприклад, останній стовбець бланку на рис. 57).

При виконанні перехресного запиту відображаються лише ті рядки і стовпці, для яких є хоча б одне обчислене значення (як на рис. 58, де відображені дані лише двох місяців).

Назва товару	Марка това	Итоговое зна	6	7	8
взгущенное молоко 'Бахмач'	Бахмач	13	8	5	
кава 'Галка'	Львівська кав	6	5	1	
майонез 'Деликатесний'	Торчин проду	2		2	
морозиво 'Пломбір'	Київське морс	11	6	5	
сік 'Карапуз'	Геснерія	20	10	10	
цукерки 'Білочка'	Світоч	6	1	5	
цукерки 'Зоряне сяйво'	ЕКО продукт	3	1	2	
цукерки 'Ромашка'	Світоч	2	1	1	
цукерки 'Столичні'	ЕКО продукт	1	1		
цукерки 'Червоний мак'	Світоч	12	7	5	

Рис. 58. Перехресний запит *АналізКількостейЗамовленихТоварівЦьогоРокуПоМісяцях* в режимі таблиці

Але для остаточного оформлення даних, виведених перехресним запитом, недостатньо створити лише цей запит, а на його основі потрібно розробити форму чи звіт. В цих же об'єктах неможливо відобразити змінну кількість полів (фактично, стовпців перехресного запиту). Тому перехресні запити розробляють так, щоб заздалегідь передбачити всі можливі стовпці, які можуть з'явитися при його виконанні і щоб назви цих стовпців не залежали від національних стандартів. Наприклад, для розглянутого вище перехресного запиту – це номери місяців від першого до дванадцятого (а не їх скорочені чи повні назви). Заголовки цих стовпців перераховують в режимі конструктора у полі *Заголовки столбцов* вікна властивостей перехресного запиту (як на рис. 59). Саме ці заголовки надалі будуть стовпцями перехресного запиту і полями для створення на його основі форм чи звітів.

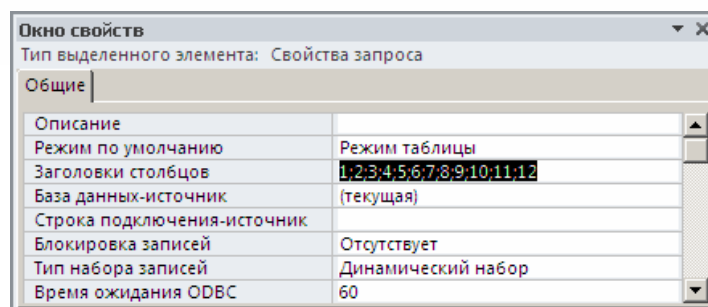


Рис. 59. Задання переліку змінних стовпців перехресного запиту *АналізКількостейЗамовленихТоварівЦьогоРокуПоМісяцях* у вікні його властивостей

Література: [4, С. 115-116; 1, С. 167-171]

### **Підготовчий етап заняття. Актуалізація знань**

1. Завантажте Access, відкрийте розроблену раніше БД *Sklad*.
2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип об'єкта*.
3. Перейдіть в області переходів до розділу *Запроси*.
4. Доповніть запит *ДаніЗамовленьТоварівЗаПеріод* полем *ДеньЗамовлення: Weekday([ДатаЗамовлення])* для відображення номера дня в тижні, у якому було оформлено замовлення. Створіть аналогічне поле *ДеньПостачання* у запиті *ДаніПостачаньТоварівЗаПеріод*.
5. Самостійно створіть запити:
  - 5.1. *СпівробітникуВідділу* – для відображення в алфавітному порядку прізвищ працюючих співробітників вказаного відділу з зазначенням посади, окладу та стажу роботи з дня зарахування в роках. Для введення назви потрібного відділу у бланку запити для поля *НазваВідділу* з таблиці *Відділи* вкажіть умову відбору *[Введіть назву відділу]*. Переконайтеся у дієздатності створеного запити;
  - 5.2. *СпівробітникуВказаноїНаціональності* – для відображення в алфавітному порядку прізвищ працюючих співробітників вказаної національності з зазначенням статі, відділу та посади;
  - 5.3. *ПостачанняВказаногоТовару* – для відображення дат постачання, кількості, ціни та суми обраного товару (параметром запити має бути не код, а назва товару);
  - 5.4. *ПоколінняСпівробітників* – для відображення в алфавітному порядку прізвищ співробітників, що народилися за вказаний проміжок часу з зазначенням статі, відділу та посади. Для параметрів запити забезпечте при введенні перевірку відповідності типу даних;
  - 5.5. *ЗвільненіСпівробітникиЗаПеріод* та *ВлаштованіСпівробітникиЗаПеріод* – для відображення в алфавітному порядку прізвищ співробітників, що звільнилися чи, відповідно, влаштувалися за вказаний проміжок часу з зазначенням статі, відділу та посади. Для параметрів запитів забезпечте при введенні перевірку відповідності типу даних.

### **Розробка запитів на вибірку даних з груповими операціями**

6. Створіть запит *КількостіСпівробітниківПоВідділах* для відображення алфавітного списку назв відділів та кількостей співробітників, що в них працюють, виконуючи наступні дії:
  - 6.1. Розпочніть створення запити в режимі конструктора та сформуєте його джерело даних з таблиць *Співробітники* та *Відділи*;
  - 6.2. У перший стовпець бланку запити перетягніть поле *НазваВідділу* з таблиці *Відділи*;

- 6.3. В другий стовпець бланку запиту перетягніть поле *КодСпівробітника* з таблиці *Співробітники*. Дайте йому назву *Кількість співробітників*;
- 6.4. Для першого стовпця встановіть ознаку сортування за зростанням;
- 6.5. З метою визначення кількості співробітників кожного відділу згрупуйте записи джерела даних за назвою відділу і підрахуйте для кожної групи кількість відповідних кодів співробітників. Для цього:
  - 6.5.1. Натисніть кнопку  $\Sigma$  (*Итоги*) в контекстному меню бланку запиту чи на вкладці стрічки меню *Конструктор* у групі *Показать или скрыть*;
  - 6.5.2. В рядку *Групповая операция*, що з'явиться на екрані, оберіть зі списку для поля *Назва відділу* значення *Группировка* (встановлюється по замовчуванню), а для поля *Кількість співробітників* – значення *Count*;
- 6.6. Вкажіть повну умову відбору для пакетної обробки даних лише працюючих співробітників (працюючий співробітник – це особа, дата влаштування якої не порожня або не перевищує сьогоднішню, а дата звільнення – не вказана, або не менша сьогоднішньої). Для цього:
  - 6.6.1. Перетягніть в третій та четвертий стовпці бланку запиту з таблиці *Співробітники* поля *ДатаВлаштування* та *ДатаЗвільнення* відповідно;
  - 6.6.2. Задайте для третього стовпця умову відбору записів *Is Null Or <=Date()*, а для четвертого зазначте *Is Null Or >=Date()*;
  - 6.6.3. Оскільки третій та четвертий стовпці задають лише умову для відбору записів, то в рядку *Групповая операция* вкажіть для них значення *Условие* та забезпечте відсутність відображення цих полів при виконанні запиту. Чому ці поля неможливо вивести на екран?;
- 6.7. Перегляньте текст сформованого запиту в режимі SQL. Обгрунтуйте його структуру;
- 6.8. Збережіть створений запит та переконайтеся в його дієздатності.
7. Для всіх запитів, пов'язаних з аналізом даних працюючих співробітників, вкажіть повну умову відбору записів аналогічно попередньому запиту.
8. Самостійно створіть запити:
  - 8.1. *КількостіПрацюючихСпівробітниківПоНаціональностям* – для відображення кількостей працюючих співробітників кожної національності;
  - 8.2. *КількостіПрацюючихСпівробітниківПоСтатях* – для відображення кількостей працюючих співробітників кожної статі.
9. Створіть запит *СумиОформленихЗамовленьПоСпівробітниках ЗаПеріод* для відображення впорядкованих за спаданням загальних сум оформлених замовлень кожним співробітником за введений період з зазначенням їх прізвищ. Для цього:



- 9.1. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиць *Співробітники*, *ЗаголовкиЗамовлень* та *ПунктиЗамовлень*;
- 9.2. В перший стовпець бланку запиту перетягніть поле *ПІБ* з таблиці *Співробітники*;
- 9.3. Для другого стовпця в рядку *Поле* введіть формулу для підрахунку суми пункту замовлення *Сума: Кількість\*Ціна* та встановіть для нього ознаку сортування за спаданням;
- 9.4. Перегляньте результати відбору даних запитом в режимі таблиці. Чому прізвища окремих співробітників виводяться декілька разів? Від чого залежить кількість таких записів?
- 9.5. Для підрахунку загальних сум оформлених замовлень кожним співробітником знову перейдіть в режим конструктора та виконайте наступні дії:
  - 9.5.1. Вкажіть, що створюється підсумковий запит. Для цього натисніть кнопку *Итоги* одним з описаних вище способів;
  - 9.5.2. Оскільки запит повинен групувати дані за прізвищами співробітника та для кожного з них підраховувати загальну суму оформлених замовлень, то в рядку *Групповая операция* для поля *ПІБ* оберіть зі списку значення *Группировка* (встановлюється по замовчуванню), а для поля *Сума* – значення *Sum*;
- 9.6. Так як запит повинен аналізувати замовлення лише за введеним періодом, то перетягніть в третій стовпець його бланку поле *ДатаЗамовлення* з таблиці *ЗаголовкиЗамовлень* та вкажіть для нього в рядку *Групповая операция* значення *Условие*, а в рядку *Условие отбора* введіть вираз *between [Введіть початкову дату] and [Введіть кінцеву дату]*;
- 9.7. Для параметрів запиту забезпечте при введенні перевірку відповідності типу даних. Для цього натисніть кнопку *Параметры* на вкладці стрічки меню *Конструктор* та вкажіть у вікні *Параметры запроса*, що з'явиться на екрані, для параметрів *[Введіть початкову дату]* та *[Введіть кінцеву дату]* тип даних *Дата/время* (див. табл. 18);
- 9.8. Перегляньте текст сформованого запиту в режимі SQL. Обгрунтуйте його структуру;
- 9.9. Збережіть створений запит та переконайтеся в його дієздатності.
10. Самостійно створіть запити, впорядкувавши їх записи за спаданням значень основних показників (призначення кожного запиту відповідає його назві):
  - 10.1. *КількостіОформленихЗамовленьПоСпівробітникахЗаПеріод*;
  - 10.2. *СумиОформленихПостачаньПоСпівробітникахЗаПеріод*;
  - 10.3. *КількостіОформленихПостачаньПоСпівробітникахЗаПеріод*;
  - 10.4. *СумиПостачаньПоПостачальникахЗаПеріод*;
  - 10.5. *КількостіПостачаньПоПостачальникахЗаПеріод*;
  - 10.6. *СумиЗамовленьПоКлієнтахЗаПеріод*;
  - 10.7. *КількостіЗамовленьПоКлієнтахЗаПеріод*;
  - 10.8. *КількостіНародженихСпівробітниківПоДняхТижня*.



## Корегування параметрів поєднань між таблицями джерела даних

11. Створіть запит *НеосвідченіСпівробітники* для відображення алфавітного списку співробітників, яким в таблиці *ОсвітаСпівробітників* не відповідає жоден запис, з зазначенням відповідних назв відділів та посад. Для цього:
  - 11.1. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиць *Відділи*, *Співробітники* та *ОсвітаСпівробітників*;
  - 11.2. Для аналізу даних всіх співробітників, а не лише тих, що мають відповідні записи в таблиці *ОсвітаСпівробітників*, змініть параметри поєднання таблиць *Співробітники* та *ОсвітаСпівробітників*, виконуючи наступні дії:
    - 11.2.1. Виділіть натисненням лівої кнопки мишки зв'язок між цими таблицями у верхній частині вікна запиту;
    - 11.2.2. В контекстному меню зв'язка оберіть пункт *Параметри об'єднання*;
    - 11.2.3. У вікні *Параметри об'єднання*, що з'явиться на екрані, встановіть перемикач поєднання в положення *Об'єднання ВСЕХ записей из 'Співробітники' и только тех из 'ОсвітаСпівробітників', в которых связанные поля совпадают* та натисніть кнопку *ОК*;
  - 11.3. Перетягніть в перший стовпець бланку запиту поле *НазваВідділу* з таблиці *Відділи*, а в другий та третій – відповідно поля *ПІБ* та *Посада* з таблиці *Співробітники*;
  - 11.4. Встановіть для першого та другого стовпця ознаку сортування за зростанням;
  - 11.5. Для відбору з базового джерела даних інформації про співробітників без відповідних записів в таблиці *ОсвітаСпівробітників* (тобто рядків джерела даних, в яких поля цієї таблиці не визначені) виконайте наступні дії:
    - 11.5.1. Перетягніть в четвертий стовпець бланку запиту поле *КодОсвіти* з таблиці *ОсвітаСпівробітників*;
    - 11.5.2. В рядку *Условие отбора* для цього стовпця введіть вираз *Is Null*;
  - 11.6. Самостійно забезпечте відображення на екрані даних лише перших трьох стовпців при виконанні запиту;
  - 11.7. Перегляньте текст сформованого запиту в режимі SQL. Обгрунтуйте структуру всіх частин запиту;
  - 11.8. Збережіть створений запит та переконайтеся в його дієздатності.
12. Самостійно створіть запити, впорядкувавши їх записи за зростанням значень основного поля (призначення кожного запиту відповідає його назві):
  - 12.1. *ТовариБезПостачань*;
  - 12.2. *ТовариБезПродажу*;
  - 12.3. *ПостачальникиБезПостачань*;
  - 12.4. *КлієнтиБезЗамовлень*;
  - 12.5. *ПостачанняБезПунктів*;
  - 12.6. *ЗамовленняБезПунктів*.

### Створення запитів на основі запитів

13. Створіть запит *ТовариБезПостачаньЗаПеріод* для формування алфавітного списку товарів, що не надходили протягом введеного періоду часу. Для формування такого списку слід спочатку визначити товари, що **надходили** за введений період, а потім з переліку всіх товарів виключити знайдені товари та вивести результати виключення на екран. Створений раніше запит *ДаніПостачаньТоварівЗаПеріод* відбирає дані постачань товарів за введений період, тому модифікуйте його структуру та використайте при створенні нашого запиту:
  - 13.1. Відкрийте запит *ДаніПостачаньТоварівЗаПеріод* в режимі конструктора, перетягніть на місце першого поля бланку запиту (а не замість нього) поле *КодТовару* з таблиці *ПунктиПостачань*, закрийте запит та погодьтеся з збереженням внесених змін;
  - 13.2. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиці *Товари* (містить дані всіх товарів) та запиту *ДаніПостачаньТоварівЗаПеріод* (містить дані товарів, які постачалися за період);
  - 13.3. Переконайтеся, що зв'язок між таблицями джерела даних встановлено поєднанням полів *КодТовару*. При відсутності такого зв'язку встановіть його перетягування відповідних полів між таблицями (аналогічно способу встановлення зв'язків між таблицями у схемі даних);
  - 13.4. Самостійно встановіть між таблицями джерела даних запиту параметр поєднання *Объединение ВСЕХ записей из 'Товари' и только тех из 'ДаніПостачаньТоварівЗаПеріод', в которых связанные поля совпадают*;
  - 13.5. Перетягніть в перший та другий стовпці бланку запиту поля *НазваТовару* та *МаркаТовару* з таблиці *Товари*. Встановіть для цих стовпців ознаку сортування за зростанням;
  - 13.6. Для відбору лише товарів, що не постачалися протягом вказаного періоду, перетягніть в третій стовпець бланку поле *КодТовару* з запиту *ДаніПостачаньТоварівЗаПеріод*, вкажіть для нього умову відбору *Is Null* та приховайте його відображення на екрані при виконанні запиту;
  - 13.7. Перегляньте текст сформованого запиту в режимі SQL. Обгрунтуйте структуру всіх частин запиту;
  - 13.8. Збережіть створений запит під назвою *ТовариБезПостачаньЗаПеріод* та переконайтеся в його дієздатності.
14. Самостійно створіть запити, впорядкувавши їх записи за зростанням значень основного поля (призначення кожного запиту відповідає його назві):
  - 14.1. *ТовариБезПродажуЗаПеріод*;
  - 14.2. *ПостачальникиБезПостачаньЗаПеріод*;
  - 14.3. *КлієнтиБезЗамовленьЗаПеріод*.
  - 14.4. *МаксимальніСумиПостачаньПоПостачальниках* (створивши перед цим запит *СумиПостачань*);

- 14.5. *МінімальніСумиПостачаньПоПостачальниках*;
- 14.6. *МаксимальніСумиЗамовленьПоКлієнтах* (створивши перед цим запит *СумиЗамовлень*);
- 14.7. *МінімальніСумиЗамовленьПоКлієнтах*.

#### Створення складних запитів на основі запитів

15. Створіть запит *ЗалишкиТоварівНаСкладі* для відображення алфавітного списку назв, марок і загальних отриманих, замовлених та наявних кількостей для кожного товару. Для цього:
  - 15.1. Створіть запит *КількостіОтриманихТоварів* з зазначенням коду та загальної отриманої кількості кожного товару (це поле повинно мати заголовок *ВсьогоОтримано*);
  - 15.2. Створіть запит *КількостіЗамовленихТоварів* з зазначенням коду та загальної замовленої кількості кожного товару (це поле повинно мати заголовок *ВсьогоЗамовлено*);
  - 15.3. Розпочніть створення запиту *ЗалишкиТоварівНаСкладі* в режимі конструктора та сформуєте його джерело даних з таблиці *Товари* та запитів *КількостіОтриманихТоварів* та *КількостіЗамовленихТоварів*;
  - 15.4. Оскільки окремі товари можуть взагалі не постачатися чи не реалізовуватися, то встановіть параметри поєднання запитів з таблицею *Товари* по полях *КодТовару* для забезпечення аналізу всіх товарів;
  - 15.5. Перетягніть в перший та другий стовпці бланку запиту відповідно поля *НазваТовару* та *МаркаТовару* з таблиці *Товари* (Чому?). Встановіть для цих стовпців ознаку сортування за зростанням;
  - 15.6. У рядок *Поле* третього стовпця бланку запиту введіть вираз *Отримано: IIF(IsNull(ВсьогоОтримано);0;ВсьогоОтримано)* для відображення кількості отриманого товару;
  - 15.7. У рядок *Поле* четвертого стовпця бланку запиту введіть вираз *Замовлено: IIF(IsNull(ВсьогоЗамовлено);0;ВсьогоЗамовлено)* для відображення кількості замовленого товару;
  - 15.8. У п'ятому стовпці самостійно введіть вираз для підрахунку залишку товарів, як різниці між отриманою та замовленою кількостями;
  - 15.9. Забезпечте відображення при виконанні запиту записів лише про ті товари, що надходили **або** замовлялися на складі;
  - 15.10. Перегляньте текст сформованого запиту в режимі SQL. Обгрунтуйте структуру всіх частин запиту;
  - 15.11. Збережіть створений запит та переконайтеся в його дієздатності.
16. Самостійно створіть запит *ЗалишкиТоварівПоВідділах* для відображення залишків товарів у кожному відділі. При цьому врахуйте, що кожен товар може знаходитися у кожному відділі, тому спочатку створіть запити *ТовариВідділів* (джерелом даних цього запиту будуть непоєднані таблиці *Відділи* та *Товари*), *КількостіОтриманихТоварівПоВідділах* (джерело даних цього

запиту має містити поєднані таблиці *Співробітники*, *ЗаголовкиПостачань* і *ПунктиПостачань*) та *КількостіЗамовленихТоварівПоВідділах*. У джерелі даних запиту *ЗалишкиТоварівПоВідділах* запит *ТовариВідділів* має поєднуватися з двома іншими запитами як по полю *КодТовару*, так і по полю *КодВідділу*, оскільки залишки мають обчислюватися для **кожного** товару у **кожному** з тих відділі, де він надходив **або** замовлявся.

### Використання властивостей запитів для ефективного аналізу даних

17. Створіть запит *ПостачанняНаМаксимальнуСуму* для відображення даних постачання на максимальну суму. Для цього:
  - 17.1. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних із запиту *СумиПостачань*;
  - 17.2. Перетягніть з джерела даних в бланк запиту поля *КодПостачальника*, *ДатаПостачання* та *Сума*;
  - 17.3. Для стовпця *Сума* встановіть ознаку сортування даних за спаданням;
  - 17.4. Для відображення даних лише першого постачання у відсортованому списку відкрийте вікно властивостей запиту, обравши відповідний пункт контекстного меню заголовка вікна чи вільного поля джерела даних, у виведеному вікні задайте для поля *Набор значень* значення *1* та закрийте вікно властивостей запиту;
  - 17.5. Перегляньте текст сформованого запиту в режимі SQL. Як відобразилася введена властивість запиту в цьому режимі;
  - 17.6. Збережіть створений запит та переконайтеся в його дієздатності (Цей самий запит можна було б створити з використанням групових операцій, але тоді довелося б створювати два запити: перший – для визначення максимальної суми постачання, а другий – для відбору даних постачання з цією сумою (в джерелі даних треба було б поєднати перший запит з запитом *СумиПостачань* по полю *Сума*)).
18. Перегляньте інші властивості запитів, полів та таблиць джерела даних, встановіть їх призначення та опишіть у письмовому звіті в таблиці згідно структури табл. 19.

Таблиця 19

Властивості запиту

Назва властивості	Призначення властивості	Відображення властивості в режимі SQL
...	...	...

19. Самостійно створіть запити:
  - 19.1. *ЗамовленняНаМаксимальнуСуму* – для відображення даних замовлення на максимальну суму;
  - 19.2. *ДесяткаОсновнихПостачальників* – для відображення десяти постачальників з максимальними загальними сумами постачань (створивши перед цим запит-джерело даних *РейтингПостачальників*);

- 19.3. *ДесяткаОсновнихКлієнтів* – для відображення десяти клієнтів з максимальними загальними сумами замовлень (створивши перед цим запит-джерело даних *РейтингКлієнтів*);
- 19.4. *ДесяткаОсновнихТоварів* – для відображення десяти товарів з максимальними загальними сумами замовлень (створивши перед цим запит-джерело даних *РейтингТоварів*).

### Створення перехресних запитів

20. Створіть перехресний (з групуванням по рядках і стовпцях) запит *АналізСумЗамовленихТоварівЦьогоРокуПоМісяцях* для відображення щомісячних сум замовлень кожного товару (в рядках запиту мають вказуватися дані замовлень окремих товарів, а в стовпцях – дані замовлень по кожному місяцю). Для цього:
- 20.1. Самостійно створіть запит-джерело даних *ДаніЗамовленьТоварівЦьогоРоку*, що відбирає дані замовлень поточного року, з аналогічною структурою до запиту *ДаніЗамовленьТоварівЗаПеріод*. З метою відбору даних створіть **додаткове** поле для обчислення року від дати замовлення та вкажіть для нього умову відбору *Year(Date())* і знищіть попередні умови та параметри відбору стосовно відбору дат за період;
- 20.2. Створіть заданий перехресний запит за допомогою майстра, виконавши наступні дії:
- 20.2.1. Завантажте майстер створення перехресних запитів, натиснувши на вкладці *Создание* стрічки меню кнопку *Мастер запросов* та обравши у виведеному вікні *Новый запрос* пункт *Перекрестный запрос*;
- 20.2.2. На першому кроці майстра оберіть запит *ДаніЗамовленьТоварівЦьогоРоку* для формування джерела даних перехресного запиту;
- 20.2.3. На другому кроці майстра перемістіть з списку доступних в список обраних поля *НазваТовару* та *МаркаТовару* для формування заголовків рядків;
- 20.2.4. На третьому кроці майстра оберіть поле *ДатаЗамовлення* для формування заголовків стовпців;
- 20.2.5. На четвертому кроці майстра оберіть зі списку інтервал групування *Місяць* для поля-заголовка стовпця;
- 20.2.6. На п'ятому кроці майстра оберіть поле *Сума* та групову функцію *Сумма* для здійснення підрахунків на перетині рядків і стовпців, оскільки запит повинен знаходити загальну щомісячну суму замовлень по кожному товару;
- 20.2.7. На останньому кроці майстра задайте назву запиту та завершіть його створення;
- 20.3. Переконайтеся у дієздатності створеного запиту;

- 20.4. Перегляньте текст сформованого запиту в режимах конструктора та SQL. Визначте призначення та обгрунтуйте структуру всіх частин запиту в цих режимах.
21. Самостійно створіть перехресні запити (призначення кожного запиту відповідає його назві):
- 21.1. Аналіз Сум Замовлених Товарів Цього Року По Кварталах;
  - 21.2. Аналіз Сум Замовлених Товарів Цього Року По Днях Тижня;
  - 21.3. Аналіз Кількостей Замовлених Товарів Цього Року По Місяцях;
  - 21.4. Аналіз Кількостей Замовлених Товарів Цього Року По Кварталах;
  - 21.5. Аналіз Кількостей Замовлених Товарів Цього Року По Днях Тижня;
  - 21.6. Аналіз Кількостей Замовлень Цього Року По Клієнтах Та Місяцях (створивши попередньо запит Замовлення Цього Року);
  - 21.7. Аналіз Кількостей Замовлень Цього Року По Клієнтах Та Кварталах;
  - 21.8. Аналіз Кількостей Замовлень Цього Року По Клієнтах Та Днях Тижня.

### **Завершальний етап заняття. Повторення вивченого матеріалу**

22. Стисніть створену БД.
23. Завершіть роботу Access.
24. Віднайдіть створену БД та заархівуйте її вміст.
25. Збережіть створений архів на власному носії в двох екземплярах.
26. Створіть у власній БД аналогічні до створених на лабораторній роботі запити з груповими операціями та перехресні запити.

### **Контрольні запитання**

1. Що позначає символ \* в першому рядку образів таблиць джерела даних запиту?
2. Як створити, перемістити чи видалити поля бланку запиту?
3. Які групові операції можна виконати над полями при конструюванні підсумкових запитів? Як кожна з цих операцій описується в режимі SQL?
4. Які параметри поєднань можна встановлювати між таблицями джерела даних? Який з цих параметрів встановлюється автоматично? Де описуються параметри поєднань в режимі SQL?
5. Для яких запитів результати виконання відрізняються від результатів перегляду в режимі таблиці?
6. Які з властивостей запиту, полів та таблиць джерела даних відображаються в режимі SQL, а які – ні? Чому?
7. Чим відрізняється перехресний запит від підсумкового? Які обмеження накладаються на джерело даних перехресного запиту? Які обмеження накладаються на структуру перехресного запиту?



## Лабораторна робота № 11

**Тема.** Створення запитів на вибірку та на поєднання в режимі SQL.

**Мета.** Формування вмінь та навичок створення запитів на вибірку даних в режимі SQL для аналізу даних таблиць. Закріплення вмінь і навичок використання запитів для аналізу даних. Усвідомлення послідовності етапів виконання запитів на вибірку та ролі зв'язків між таблицями при формуванні джерела даних. Вдосконалення вмінь створення та корегування об'єктів додатків СУБД MS Access.

### Теоретичні відомості

У випадках, коли необхідно одночасно вивести результати виконання декількох запитів, застосовують **запити на поєднання**. Ці запити поєднують результати відбору з декількох джерел (а не з одного, як, наприклад, запити на вибірку), тому створювати і змінювати їх структуру можливо лише в режимі SQL. Синтаксис запитів на поєднання в цьому режимі такий:

```
SELECT <поля результатів відбору 1>  
FROM <джерело даних 1>  
[WHERE <умови відбору записів 1>]  
[GROUP BY <поля групування 1>]  
[HAVING <умови відбору груп 1>]  
UNION | UNION ALL  
SELECT <поля результатів відбору 2>  
FROM <джерело даних 2>  
[WHERE <умови відбору записів 2>]  
[GROUP BY <поля групування 2>]  
[HAVING <умови відбору груп 2>]  
...  
[ORDER BY <поля сортування>];
```

Бачимо, що запити на поєднання можуть об'єднувати результати багатьох запитів на вибірку. Необхідно лише забезпечити, щоб кількість полів виводу у кожному з цих запитів-частин на вибірку була однаковою і кожне поле у всіх частинах мало сумісний тип даних.

Оператор **UNION** виводить лише унікальні записи з усіх частин (тобто автоматично відкидає записи-дублікати). Для відображення ж всіх записів з усіх запитів-частин необхідно між ними в режимі SQL вказувати оператор **UNION ALL**. Наголосимо, що речення сортування **ORDER BY** вказується при потребі в кінці всього запиту на поєднання лише один раз і забезпечує сортування між собою всіх записів з усіх частин. В цьому реченні записуються або поля з першої частини, або (що використовується частіше) номери стовпців.

Література: [2, С. 372-436]

### Підготовчий етап заняття. Актуалізація знань

1. Завантажте Access, відкрийте розроблену раніше БД Sklad.
2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип об'єкта*.
3. Перейдіть в цій області до розділу *Запроси*.

### Створення найпростіших запитів на вибірку з сортуванням записів в режимі SQL

4. Створіть запит *SQLАлфавітнийСписокПостачальників* в режимі SQL для формування алфавітного списку постачальників з зазначенням адреси та дати реєстрації. Для цього:
  - 4.1. Розпочніть створення запиту в режимі конструктора, натиснувши на вкладці стрічки меню *Создание* у групі *Запросы* кнопку *Конструктор запросов*;
  - 4.2. Закрийте вікно *Добавление таблицы*, оскільки текст запиту буде вводиться в режимі SQL;
  - 4.3. Для створення тексту запиту перейдіть в режим SQL одним з двох способів: за допомогою пункту контекстного меню *Режим SQL* вільного поля області джерела даних запиту чи з використанням аналогічного пункту кнопки *Главная* ► *Режим стрічки меню*;
  - 4.4. У вікні режиму SQL введіть текст запиту:

```
SELECT Постачальники.НазваПостачальника,  
Постачальники.Адреса, Постачальники.ДатаРеєстрації  
FROM Постачальники  
ORDER BY Постачальники.НазваПостачальника;
```
  - 4.5. Завантажте створений запит на виконання, обираючи в контекстному меню заголовка вікна запиту чи у списку кнопки *Режим* стрічки меню позицію *Режим таблиць*;
  - 4.6. Перегляньте структуру створеного запиту в режимі конструктора. Де в цьому режимі відображається кожна структурна частина запиту, створеного в режимі SQL?
  - 4.7. Закрийте вікно конструктора та збережіть запит під назвою *SQLАлфавітнийСписокПостачальників*.
5. Самостійно створіть аналогічний запит *SQLАлфавітний-СписокКлієнтів* для відображення в алфавітному порядку назв клієнтів з зазначення місця роботи (назви необхідних полів можна вписати чи безпосередньо скопіювати з вікна режиму конструктора таблиці *Клієнти*). Переконайтеся в дієздатності запиту та перегляньте і проаналізуйте його структуру в режимі конструктора.

### Створення обчислювальних полів та формування умов відбору записів джерела даних запитів в режимі SQL

6. Доповніть запит *SQLАлфавітнийСписокПостачальників* полем *День реєстрації* для зазначення дня тижня, в якому було зареєстровано постачальника. Для цього:

6.1. Відкрийте вказаний запит в режимі SQL;

6.2. Відредагуйте текст запиту до вигляду:

```
SELECT Постачальники.НазваПостачальника,  
Постачальники.Адреса, Постачальники.ДатаРеєстрації,  
Weekday([ДатаРеєстрації]) AS [День реєстрації]  
FROM Постачальники  
ORDER BY Постачальники.НазваПостачальника;
```

6.3. Завантажте відредагований запит на виконання, переглядаючи його в режимі таблиці;

6.4. Перегляньте запит в режимі конструктора. Як в цьому режимі відображається обчислювальне поле?

6.5. Закрийте запит та збережіть внесені зміни.

7. Створіть в режимі SQL запит *SQLНекоректноЗареєстрованіПостачальники* для відображення даних постачальників, зареєстрованих в неділю. Для цього:

7.1. Скопіюйте запит *SQLАлфавітнийСписокПостачальників* в буфер обміну та вставте його під назвою *SQLНекоректноЗареєстрованіПостачальники*, оскільки ці запити мають однакові джерела даних, поля для відображення та сортування;

7.2. Відкрийте запит *SQLНекоректноЗареєстрованіПостачальники* в режимі SQL та відредагуйте його текст до вигляду:

```
SELECT Постачальники.НазваПостачальника, Постачальники.  
Адреса, Постачальники.ДатаРеєстрації,  
Weekday([ДатаРеєстрації]) AS [День реєстрації]  
FROM Постачальники  
WHERE Weekday([ДатаРеєстрації])=1  
ORDER BY Постачальники.НазваПостачальника;
```

7.3. Завантажте відредагований запит на виконання;

7.4. Перегляньте запит в режимі конструктора. Як в цьому режимі відображається умова відбору записів з джерела даних?

7.5. Закрийте запит та збережіть внесені зміни.

8. Самостійно доповніть в режимі SQL запит *SQLАлфавітнийСписокКлієнтів* полями *Число реєстрації*, *Місяць реєстрації* та *День реєстрації* (для відображення дня тижня, в якому було зареєстровано клієнта).

9. Самостійно створіть запит *SQLНекоректноЗареєстрованіКлієнти* для відображення даних клієнтів, зареєстрованих в неділю.

10. Перегляньте запит *ПрацюючіПенсіонера* в режимах конструктора та SQL. Як поєднуються в режимі SQL умови відбору, записані в режимі конструктора в одному рядку? В різних рядках?

## Конструювання параметрів поєднання таблиць джерела даних в режимі SQL

11. Створіть запит *SQLСпівробітникиБезВідділів* для відображення прізвищ співробітників, у яких не вказаний відділ, де вони працюють. Для цього:

11.1. Розпочніть створення запиту та перейдіть в режим SQL;

11.2. У вікні режиму SQL введіть текст запиту:

```
SELECT Співробітники.ПІБ  
FROM Відділи RIGHT JOIN Співробітники ON  
Відділи.КодВідділу = Співробітники.КодВідділу  
WHERE Відділи.КодВідділу Is Null  
ORDER BY Співробітники.ПІБ;
```

11.3. Завантажте введений запит на виконання;

11.4. Перегляньте запит в режимі конструктора. Як в цьому режимі відображається обраний параметр поєднання між таблицями джерела даних? Встановіть в режимі конструктора інші параметри поєднання таблиць джерела даних та з'ясуйте, як ці зміни відображаються в режимі SQL;

11.5. Закрийте та збережіть створений запит.

12. Самостійно створіть в режимі SQL такі запити:

12.1. *SQLСпівробітникиБезПостачань*;

12.2. *SQLСпівробітникиБезПостачаньЗаПеріод*;

12.3. *SQLСпівробітникиБезЗамовлень*;

12.4. *SQLСпівробітникиБезЗамовленьЗаПеріод*.

## Створення запитів з групуванням даних в режимі SQL

13. Створіть запит *SQLКількостіЗареєстрованихПостачальників-ПодняхТижня* в режимі SQL для відображення кількостей зареєстрованих постачальників за кожен день тижня. Для цього:

13.1. Розпочніть створення запиту та перейдіть в режим SQL;

13.2. У вікні режиму SQL введіть текст запиту:

```
SELECT SQLАлфавітнийСписокПостачальників.[День реєстрації],  
Count(SQLАлфавітнийСписокПостачальників.НазваПостачальника)  
AS [Кількість постачальників]  
FROM SQLАлфавітнийСписокПостачальників  
GROUP BY SQLАлфавітнийСписокПостачальників.[День реєстрації]  
ORDER BY SQLАлфавітнийСписокПостачальників.[День реєстрації];
```

13.3. Завантажте введений запит на виконання;

13.4. Перегляньте даний запит в режимі конструктора. Як в цьому режимі відображаються поля групування і поля з груповими операціями?

13.5. Закрийте та збережіть створений запит.

14. Самостійно створіть в режимі SQL запити *SQLКількостіЗареєстрованихКлієнтівПодняхТижня*, та *SQLКількостіЗареєстрованихКлієнтівПоМісяцях*.

## Створення запитів на поєднання записів в режимі SQL

15. Створіть запит *SQLПостачальникиТаКлієнти* в режимі SQL для відображення алфавітного списку постачальників та клієнтів з зазначенням дати реєстрації. Для цього:
  - 15.1. Внесіть у форми *Постачальники* та *Клієнти* по запису з однаковими прізвищами та ініціалами і датами реєстрації;
  - 15.2. Розпочніть створення запиту та перейдіть в режим SQL;
  - 15.3. Не закриваючи вікна запиту, відкрийте в режимі SQL додатковий запит *АлфавітнийСписокПостачальників* та скопіюйте через буфер обміну його текст у вікно створюваного запиту. Закрийте додатковий запит;
  - 15.4. Не закриваючи вікна запиту, відкрийте в режимі SQL додатковий запит *АлфавітнийСписокКлієнтів* та скопіюйте через буфер обміну його текст у вікно створюваного запиту після скопійованого раніше тексту. Закрийте додатковий запит;
  - 15.5. Відредагуйте текст нового запиту до вигляду:

```
SELECT Постачальники.НазваПостачальника,  
Постачальники.ДатаРеєстрації  
FROM Постачальники  
UNION  
SELECT Клієнти.НазваКлієнта, Клієнти.ДатаРеєстрації  
FROM Клієнти  
ORDER BY 1;
```
  - 15.6. Завантажте введений запит на виконання та запам'ятайте кількість записів, що відображається при цьому на екрані;
  - 15.7. Спробуйте переглянути його структуру в режимі конструктора. Чому це не вдалося?
  - 15.8. Поверніться в режим SQL і після оператора UNION допишіть слово ALL. Знову завантажте введений запит на виконання і порівняйте кількість записів, що відображається на екрані, з кількістю записів, отриманою раніше. Віднайдіть ті записи, що призвели до розбіжностей результатів;
  - 15.9. Закрийте та збережіть створений запит.
16. Самостійно створіть запит *SQLФізичніОсоби* для відображення алфавітного списку унікальних прізвищ та ініціалів співробітників, постачальників і клієнтів.

## Завершальний етап заняття. Повторення вивченого матеріалу

17. Стисніть створену БД.
18. Завершіть роботу Access.
19. Віднайдіть створену БД та заархівуйте її вміст.
20. Збережіть створений архів на власному флеш-носії в двох екземплярах.
21. Доповніть власну БД запитам на поєднання, аналогічними до створених на лабораторній роботі.

### Контрольні запитання

1. Чи відрізняється послідовність виконання запитів, створених в режимі SQL, від послідовності виконання запитів, створених в режимі конструктора?
2. Яка загальна структура запитів на вибірку в режимі SQL?
3. Чим відрізняється синтаксис опису обчислювальних полів в режимах конструктора та SQL?
4. Як в режимі SQL описуються зв'язки та параметри поєднання між таблицями?
5. Де в режимі SQL описуються поля групування і де – поля з груповими операціями?
6. Де в режимі SQL описуються умови відбору записів джерела даних і де – умови відбору груп?
7. Як в Access слід створювати нові, але подібні до існуючих запити?
8. Яка загальна структура запитів на поєднання в режимі SQL?
9. Які основні вимоги висуваються до кожного складового запита на вибірку у загальному запиті на поєднання?

З питань продажів  
друкованих примірників  
за оптовими цінами  
чи електронного варіанту  
цього посібника без реклами  
звертайтеся на e-mail  
books\_shportko@ukr.net



## Лабораторна робота № 12

**Тема.** Розробка та використання запитів на корегування даних.

**Мета.** Формування вмінь та навичок конструювання запитів на створення таблиць, доповнення, оновлення та знищення записів, використання будівничого виразів. Вдосконалення вмінь і навичок створення та корегування структури запитів в режимі конструктора. Систематизація знань можливостей використання запитів для пакетного корегування даних.

### Теоретичні відомості

Дослідимо можливості пакетного корегування даних за допомогою відповідних запитів та їх синтаксис як в режимі конструктора, так і SQL. Нагадаємо, що для виконання дій, передбачених такими запитами, необхідно або завантажувати їх з вікна додатку Access або натискати кнопку стрічки меню *Конструктор* ► *Виконати*. Перед виконанням запиту на корегування рекомендується перейти в *Режим таблиць* щоб переглянути записи, над якими **буде виконуватися** корегування.

Для перетворення запиту на вибірку в запит на корегування, перебуваючи в режимі конструктора, необхідно у групі стрічки меню *Конструктор* ► *Тип запиту* змінити його тип (рис. 60), натиснувши відповідну кнопку. В режимі ж SQL для таких дій необхідно змінювати саму структуру запиту.

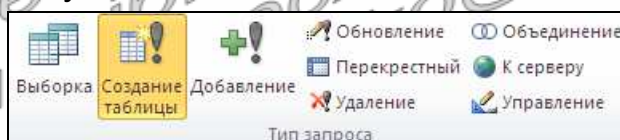


Рис. 60. Кнопки стрічки меню для зміни типу запиту в режимі конструктора

Серед запитів на корегування найменше відрізняються за структурою від запитів на вибірку **запити на створення таблиць** (📄!). Обираючи такий тип запиту в режимі конструктора (див. рис. 60), потрібно лише додатково вказати назву нової таблиці, в яку будуть записуватися результати вибірки даних цим запитом. В режимі ж SQL для перетворення запиту на вибірку у запит на створення таблиці необхідно після речення `SELECT` дописати тільки

```
INTO <ім'я нової таблиці> .
```

Наголосимо, що ім'я таблиці, яку створює запит, не має співпадати з назвою цього запиту, інакше при виконанні такого запиту буде створено нову таблицю, але сам запит буде знищено, навіть якщо при цьому він відкритий в режимі конструктора.

Запити на створення таблиць найчастіше застосовуються для автоматичного архівного зберігання поточного стану даних таблиць чи результатів їх аналізу. Наприклад, для зберігання сум замовлень кожного клієнта в кінці поточного місяця доцільно створити запит, який в режимі SQL запишеться так:

```
SELECT Клієнти.НазваКлієнта, Sum([Кількість]*[Ціна]) AS Сума
INTO СумиЗамовленьПоКлієнтахЗаПоточнийМісяць
FROM Клієнти INNER JOIN (ЗаголовкиЗамовлень INNER JOIN
ПунктиЗамовлень ON ЗаголовкиЗамовлень.КодЗамовлення =
ПунктиЗамовлень.КодЗамовлення) ON Клієнти.КодКлієнта =
ЗаголовкиЗамовлень.КодКлієнта
WHERE (((Month([ДатаЗамовлення])=Month(Date())) AND
((Year([ДатаЗамовлення])=Year(Date()))))
GROUP BY Клієнти.НазваКлієнта
ORDER BY Sum([Кількість]*[Ціна]) DESC;
```

Для того, щоб створювана таблиця не була автоматично знищена після наступного виконання цього запиту у наступному місяці, щоразу після виконання запиту її слід перейменовувати, додатково фіксуючи у назві поточний рік та місяць, а при потребі – і поточну дату.

Для запитів на оновлення (🔧) вказувати ім'я корегованої таблиці не потрібно, адже під час їх виконання змінюються записи джерела даних. В режимі конструктора після вибору цього типу запитів (див. рис. 60, кнопка *Обновление*) у бланку замість рядків *Групповая операция*, *Сортировка* та *Вывод на экран* виводиться рядок *Обновление*, де вказуються значення, що заносяться в обране поле з джерела. Для полів відбору значення в цьому рядку не вказується. Наприклад, для обнулення проценту знижки при поверненні товарів доцільно в режимі конструктора створити запит, аналогічний наведеному на рис. 61. Зверніть увагу: для прискорення виконання цього запиту обнулюється процент знижки не для всіх повернень (*Кількість < 0*), а лише для тих, де процент знижки не вказаний або додатний (*Is Null or >0*).

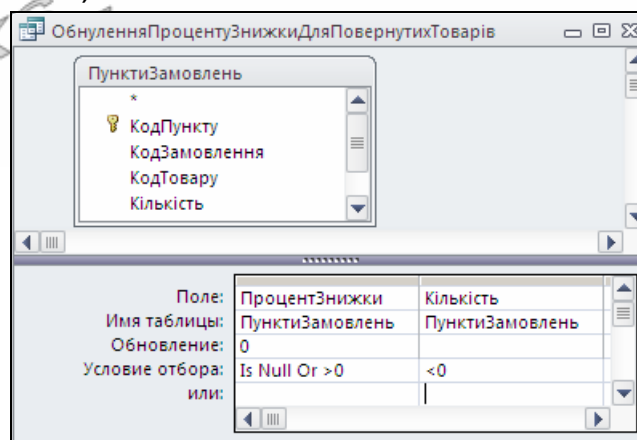


Рис. 61. Структура запиту *ОбнуленняПроцентуЗнижкиДляПовернутихТоварів* в режимі конструктора

В режимі SQL синтаксис запитів на оновлення такий:

```

UPDATE <джерело даних>
SET <ім'я поля> = <вираз> [, ...]
[WHERE <умови відбору записів>];

```

тобто поля і значення, які в них заносяться, перераховуються в реченні SET, а сам запит починається словом UPDATE. Наприклад, розглянутий вище запит в режимі SQL має вигляд:

```

UPDATE ПунктиЗамовлень SET ПунктиЗамовлень.ПроцентЗнижки = 0
WHERE (ПунктиЗамовлень.ПроцентЗнижки Is Null Or ПунктиЗамовлень.
ПроцентЗнижки > 0) AND ПунктиЗамовлень.Кількість < 0;

```

У запитах на оновлення ніколи не змінюються обчислювальні поля та поля-лічильники.

Кожен з **запитів на додавання** (👤) застосовується для доповнення новими записами однієї з існуючих таблиць. Тому в режимі конструктора після вибору даного типу запиту (див. рис. 60, кнопка *Добавление*) потрібно вказати назву цієї таблиці. Після цього у бланку запиту з'явиться додатковий рядок *Добавление*, де **обираються** поля таблиці для доповнення, в які записується значення з рядка *Поле*. В доповнювану таблицю записується стільки записів, скільки є у джерелі. Запит на додавання може й не мати джерела. Тоді він доповнює таблицю одним записом значеннями виразів з рядка *Поле*. У запитах на додавання, аналогічно запитам на оновлення, ніколи не записуються значення в обчислювальні поля та не рекомендується встановлювати поля-лічильники. Access перевіряє цілісність як окремих полів, так і цілих створюваних записів, тому всі обов'язкові поля доповнюваної таблиці повинні або мати значення по замовчуванню або заповнюватися таким запитом. Наприклад, запит на доповнення таблиці *Клієнти* записами з даних тих постачальників, які у цій таблиці відсутні, в режимі конструктора виглядає як на рис. 62. Зверніть увагу: у поле *КодКлієнта* значення не записуються. Воно лише використовується для відбору даних тих постачальників, які відсутні в таблиці *Клієнти*.

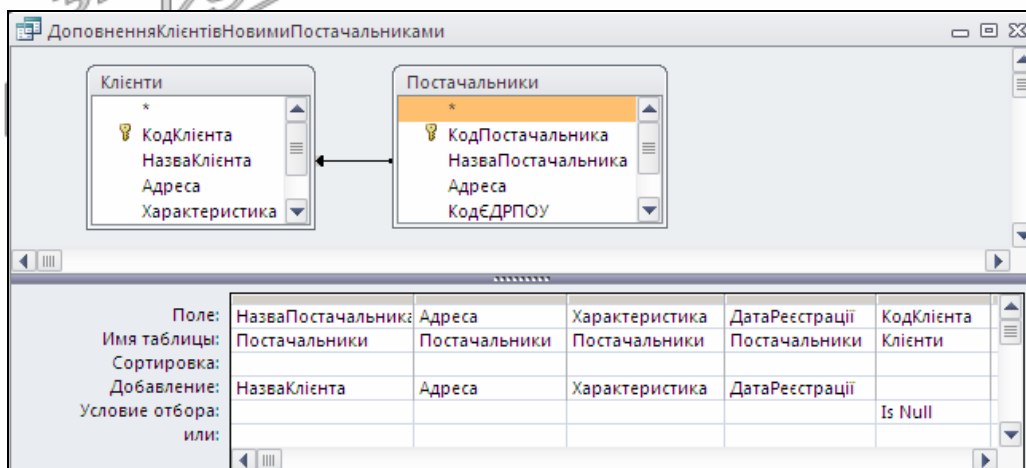


Рис. 62. Структура запиту *ДоповненняКлієнтівНовимиПостачальниками* в режимі конструктора

В режимі SQL запити на додавання записуються згідно синтаксису:

```
INSERT INTO <ім'я таблиці> [(<поле1>, [<поле2> ...])]  
VALUES ((<значення1>, [<значення2> ...]) |  
SELECT <поле1>, [<поле2> ...]  
[FROM <джерело даних>]  
[WHERE <умови відбору записів>]  
[GROUP BY <поля групування>]  
[HAVING <умови відбору груп>]  
[ORDER BY <поля сортування>]; .
```

Бачимо, що поля доповнюваної таблиці і значення, які в них заносяться, записуються в різних реченнях команди SQL. Поля для зберігання значень перераховуються в реченні INSERT INTO в круглих дужках після назви таблиці (коли послідовно встановлюються значення всіх полів, то їх можна не перераховувати). Значення ж для полів обчислюються з виразів після VALUES (якщо вставляється один запис) або послідовно беруться з полів результатів відбору команди SELECT. Наприклад, запит, наведений на рис. 62, в режимі SQL виглядає так:

```
INSERT INTO Клієнти (НазваКлієнта, Адреса, Характеристика,  
ДатаРеєстрації)  
SELECT Постачальники.НазваПостачальника, Постачальники.Адреса,  
Постачальники.Характеристика, Постачальники.ДатаРеєстрації  
FROM Постачальники LEFT JOIN Клієнти ON  
Постачальники.НазваПостачальника = Клієнти.НазваКлієнта  
WHERE Клієнти.КодКлієнта Is Null; .
```

І нарешті, для пакетного видалення записів з окремих таблиць, що формують відібрані записи джерела, створюють **запити на видалення** (✖). Кожен з таких запитів може видаляти записи з однієї чи декількох таблиць джерела. Тому при виборі цього типу запиту в режимі конструктора у бланку з'являється додатковий рядок *Удаление*, в якому для кожного стовпця можна обрати одне з двох значень: *Из* – для позначення кожної таблиці, з якої виконується вилучення (при цьому з даної таблиці в рядок *Поле* такого стовпця необхідно перетягнути \*) чи *Условие* – для вказування полів чи виразів, згідно яких виконується відбір записів джерела (для таких полів потрібно задавати умови відбору в розміщених нижче рядках). Наприклад, запит з рис. 63 використовується для знищення тих пунктів постачань (перший стовбець), в яких кількість не вказана або відсутня (другий стовбець).

Перед завантаженням запитів на видалення необхідно у схемі даних проаналізувати встановлення прапорця *каскадного видалення пов'язаних записів* тих зв'язків, відносно яких таблиці, з яких безпосередньо чи опосередковано вилучаються дані, знаходяться на стороні 1. Адже видалення записів з таблиці при встановленому цьому



прапорці автоматично призведе до видалення пов'язаних записів з таблиць на стороні  $\infty$ , а при зняттю – не вилучить ті записи на стороні 1, для яких є пов'язані записи на стороні  $\infty$ .

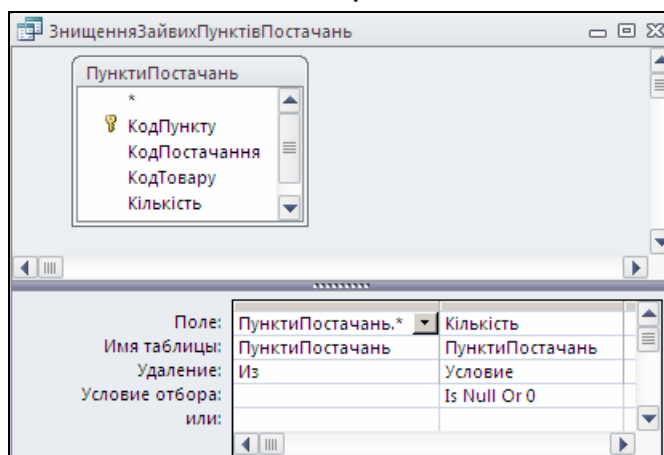


Рис. 63. Структура запити *ЗнищенняЗайвихПунктівПостачань* в режимі конструктора

Синтаксис запитів на видалення в режимі SQL такий (причому \* без назви таблиці можна вживати лише у випадку, коли джерело містить єдину таблицю):

```
DELETE * | <ім'я таблиці1>.* [, <ім'я таблиці2>.* ...]
FROM <джерело даних>
[WHERE <умови відбору записів>];
```

Наприклад, запит з рис. 63 в режимі SQL виглядає так:

```
DELETE ПунктиПостачань.*, ПунктиПостачань.Кількість
FROM ПунктиПостачань
WHERE ПунктиПостачань.Кількість Is Null Or
ПунктиПостачань.Кількість=0;
```

На завершення теоретичних відомостей нагадаємо, що для прискорення виконання найуживаніших запитів на корегування (як і всіх інших запитів) доцільно створювати індекси в базових таблицях за основними полями умов відбору.

Література: [4, С. 115-116; 1, С. 167-171]

### Підготовчий етап заняття. Актуалізація знань

1. Завантажте Access, відкрийте розроблену раніше БД *Sklad*.
2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип объекта*.
3. Перейдіть в цій області до розділу *Запросы*.

### Розробка запитів на створення таблиць.

#### Використання будівничого виразів при конструюванні запитів

4. Створіть запит *СтворенняТиповоїПлатіжноїВідомостіВідділу* для генерації таблиці *ТиповаПлатіжнаВідомістьВідділу*, що має містити алфавітний список працюючих співробітників вказаного відділу з зазначенням відповідних окладів, утримань (прибуткового

податку, профспілкового внеску, додаткового податку) та сум до видачі. Для цього:

- 4.1. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиць *Відділи* та *Співробітники*;
- 4.2. Перетворіть створюваний запит на вибірку в запит на створення таблиці, натискаючи на вкладці *Конструктор* стрічки меню у групі *Тип запроса* кнопку *Создание таблицы* або обираючи в контекстному меню вільної області джерела даних у підменю *Тип запроса* пункт *Создание таблицы*. У вікні *Создание таблицы*, що з'явиться на екрані, вкажіть назву створюваної таблиці – *ТиповаПлатіжнаВідомістьВідділу* та натисніть кнопку *ОК*;
- 4.3. В перший та другий стовпці бланку запиту перетягніть відповідно поля *ПІБ* та *Оклад* з таблиці *Співробітники*;
- 4.4. Для першого стовпця встановіть ознаку сортування за зростанням;
- 4.5. Припустимо, що прибутковий податок вираховується за формулою

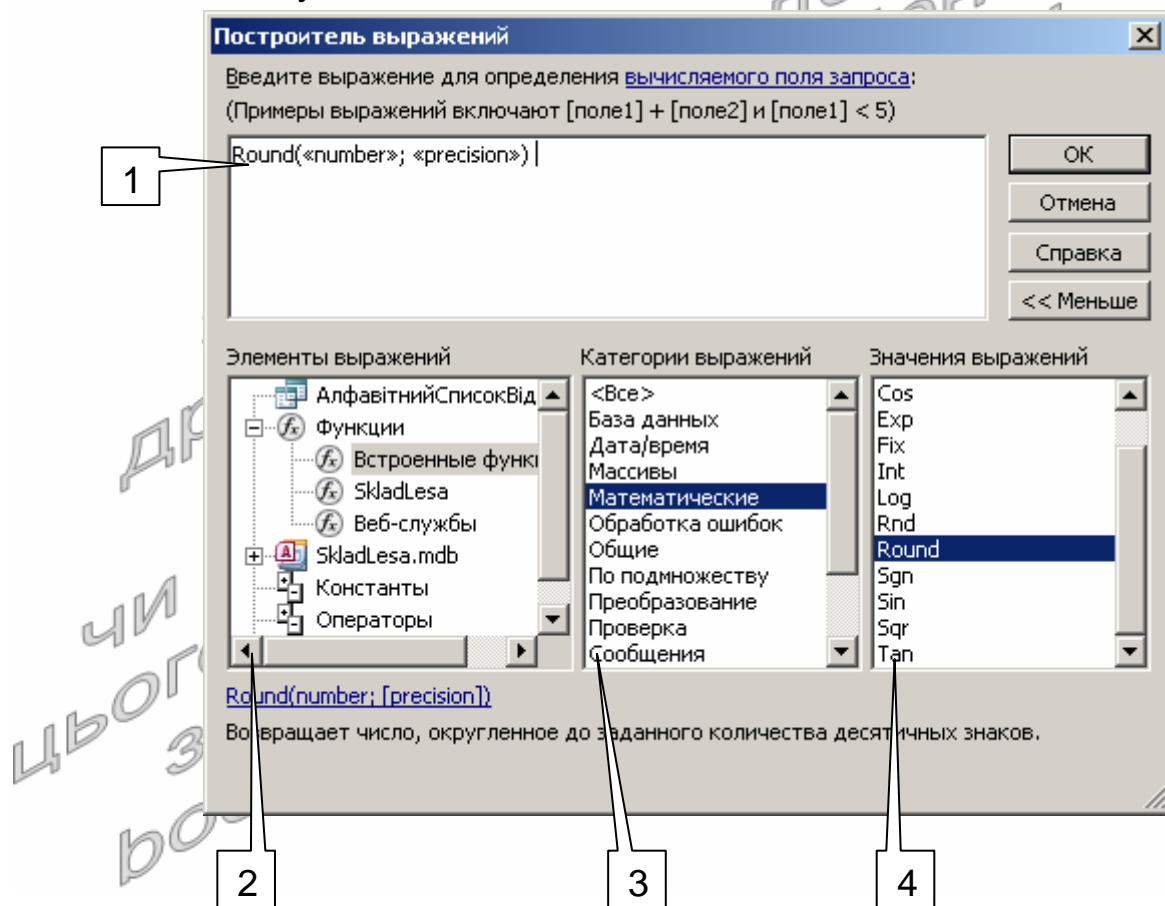
$$P = \begin{cases} 0, & H \leq 17 \\ 0,1 \times (H - 17), & 17 < H \leq 85 \\ 6,80 + 0,15 \times (H - 85), & 85 < H \leq 170 \\ 19,55 + 0,2 \times (H - 170), & 170 < H \leq 1020 \\ 189,55 + 0,3 \times (H - 1020), & H > 1020 \end{cases}$$

де  $P$  – прибутковий податок,  $H$  – нарахована сума. Тоді для автоматичного підрахунку прибуткового податку відносно окладу введіть в рядок *Поле* третього стовпця бланку запиту вираз *Прибутковий податок*:  $Round(If(Оклад \leq 17; 0; If(Оклад \leq 85; 0,1 * (Оклад - 17); If(Оклад \leq 170; 6,8 + 0,15 * (Оклад - 85); If(Оклад \leq 1020; 19,55 + 0,2 * (Оклад - 170); 189,55 + 0,3 * (Оклад - 1020))))); 2)$  за допомогою будівничого виразів, виконуючи наступні дії:

- 4.5.1. В контекстному меню рядка *Поле* третього стовпця бланку запиту оберіть пункт *Построить* для виклику вікна будівничого виразів (рис. 64. Надалі в цьому підпункті номер поля вказує на номер виноска цього рисунка);
- 4.5.2. В полі для редагування виразу введіть заголовок стовпця *Прибутковий податок*.;
- 4.5.3. Для введення у вираз функції *Round* в полі 2 відкрийте папку *Функции*, а в ній – папку *Встроенные функции*; в полі 3 оберіть категорію функцій *Математические* та серед математичних функцій двічі натисніть лівою кнопкою мишки у полі 4 на функції *Round*. Обрана функція вставиться у полі 1 в положення курсору (див. рис. 64);



- 4.5.4. Для введення вкладеної функції виділіть у полі 1 перший аргумент «*Выражение*» вставленої функції *Round* та вставте замість нього функцію *If* з категорії *Управление*. Знищіть елемент «*Выражение*» після вставленої функції;
- 4.5.5. Назву поля *Оклад* введіть безпосередньо з клавіатури або вставте з полів нижньої частини будівничого виразів, відмітивши в полі 2 таблицю *Склад* ► *Таблицы* ► *Співробітники* та обравши з неї в 3 поле *Оклад*;
- 4.5.6. Самостійно завершіть формування виразу, вводячи оператори безпосередньо з клавіатури чи за допомогою полів 2-4. Не забувайте при цьому знищувати зайві елементи «*Выражение*» в 1;
- 4.5.7. Для запису сформованого виразу у бланк запити натисніть кнопку *OK*;



- 1 – поле для редагування тексту виразу;
- 2 – структура об'єктів, констант, функцій, операторів та загальних виразів;
- 3 – категорії обраного елемента з (2);
- 4 – значення обраного елемента з двох попередніх полів.

Рис. 64. Структура будівничого виразів

- 4.6. В рядок *Поле* четвертого стовпця введіть вираз для обчислення профспілкового внеску (1 % з нарахувань співробітника);

4.7. В рядок *Поле* п'ятого стовпця бланку запиту введіть вираз для обчислення додаткового податку, що розраховується за формулою  $V = \begin{cases} 0.01 \times H, H \leq 150 \\ 0.02 \times H, H > 150 \end{cases}$ , де  $V$  – сума додаткового податку,

$H$  – нарахована сума;

4.8. В рядок *Поле* шостого стовпця самостійно введіть вираз для обчислення сум до видачі, як різниці між відповідними нарахованими (поле *Оклад*) і утриманими сумами;

4.9. Для формування типової платіжної відомості лише для співробітників окремого відділу в сьомий стовпець бланку запиту перетягніть поле *НазваВідділу* з таблиці *Відділи*, приховайте для нього вивід на екран в режимі таблиці та вкажіть умову відбору [*Введіть назву відділу*];

4.10. Перетягніть у восьмий та дев'ятий стовпці відповідно поля *ДатаВлаштування* та *ДатаЗвільнення* з таблиці *Співробітники*. Приховайте їх вивід на екран в режимі таблиці та вкажіть коректні умови відбору для аналізу даних лише працюючих співробітників;

4.11. У вікні властивостей числових полів запиту, що виводяться на екран, вкажіть потрібний формат поля та число десяткових знаків;

4.12. Збережіть запит під назвою *СтворенняТиповоїПлатіжноїВідомостіВідділу* та завантажте його на виконання. Після виконання запиту перегляньте вміст створеної таблиці *ТиповаПлатіжнаВідомістьВідділу*. Перегляньте структуру створеної таблиці в режимі конструктора.

5. Самостійно створіть запити:

5.1. *СтворенняПлатіжноїВідомостіПреміїЗаСтаж* – для генерації таблиці *ПлатіжнаВідомістьПреміїЗаСтаж* з алфавітним списком працюючих співробітників та відповідними сумами нарахованої премії за стаж (за кожен рік стажу – п'ять гривень);

5.2. *СтворенняПлатіжноїВідомостіПреміїЗаЗамовленняПеріоду* – для генерації таблиці *ПлатіжнаВідомістьПреміїЗаЗамовленняПеріоду* з алфавітним списком працюючих співробітників та відповідними сумами нарахованої премії за оформлені замовлення протягом вказаного періоду (1% від загальної суми оформлених замовлень).

### **Створення запитів на оновлення даних таблиць**

6. Доповніть таблицю *ПунктиЗамовлень* цілочисельним полем *ПроцентЗнижки*, встановивши для нього значення по замовчуванню рівним 0. Для сформованих раніше записів цієї таблиці автоматично внесіть в створене поле значення 0 при відсутності в ньому будь-яких значень. Для цього створіть запит

Ініціалізація *ПроцентуЗнижкиВПунктахЗамовлень*, виконуючи наступні дії:

- 6.1. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиці *ПунктиЗамовлень*;
  - 6.2. Перетворіть запит на вибірку в запит на оновлення даних, натискаючи на вкладці *Конструктор* стрічки меню у групі *Тип запроса* кнопку *Обновление* або обираючи в контекстному меню вільної області джерела даних у підменю *Тип запроса* пункт *Обновление*;
  - 6.3. В перший стовпець бланку запиту перетягніть поле *ПроцентЗнижки* з таблиці *ПунктиЗамовлень*;
  - 6.4. Для редагування під час виконання запиту лише записів з відсутнім значенням в полі *ПроцентЗнижки* вкажіть для створеного першого стовпця бланку запиту умову відбору *Is Null*;
  - 6.5. З метою автоматичного внесення в поля *ПроцентЗнижки* відібраних записів значення 0 введіть в рядку *Обновление* першого стовпця запиту значення 0;
  - 6.6. Збережіть створений запит та переконайтеся в його дієздатності. Перегляньте та обгрунтуйте структуру запиту в режимі SQL.
7. Створіть запит *НарахуванняПроцентуЗнижкиВПунктахЗамовлень-ЗаПеріод* для автоматичного заповнення поля *ПроцентЗнижки* в записах таблиці *ПунктиЗамовлень*, що відповідають оформленим замовленням протягом вказаного періоду, згідно наступної шкали:
- $$P = \begin{cases} 0, H \leq 10 \\ 3, 10 < H \leq 100 \\ 5, H > 100 \end{cases}$$
- де  $P$  – процент знижки,  $H$  – сума пункту замовлення. Для цього:
- 7.1. Розпочніть створення запиту в режимі конструктора, сформуєте його джерело даних з таблиць *ЗаголовкиЗамовлень* та *ПунктиЗамовлень* і перетворіть його в запит на оновлення даних;
  - 7.2. В перший стовпець бланку запиту перетягніть поле *ПроцентЗнижки* з таблиці *ПунктиЗамовлень*;
  - 7.3. В рядку *Обновление* першого стовпця, використовуючи будівничого, введіть вираз для оновлення проценту знижки згідно шкали: *IIf([Кількість]\*[Ціна]<=10; 0; IIf((([Кількість]\*[Ціна]<= 100; 3; 5))*);
  - 7.4. Для аналізу пунктів, що стосуються лише оформлених замовлень протягом вказаного періоду, перетягніть в другий стовпець бланку запиту поле *ДатаЗамовлення* з таблиці *ЗаголовкиЗамовлень* та вкажіть для нього умову відбору

*Between [Введіть початкову дату] and [Введіть кінцеву дату];*

7.5. Збережіть створений запит та переконайтеся в його дієздатності.

8. Самостійно створіть запити:

8.1. *НарахуванняСвятковогоПроцентуЗнижкиВПунктахЗамовлень* – для автоматичного встановлення п'ятипроцентної знижки в пунктах замовлень, що оформлені 1 січня, 7 січня, 8 березня, 9 травня, 24 серпня будь-якого року;

8.2. *ПідвищенняОкладівНаДесятьПроцентів* – для автоматично підвищення окладів всіх працюючих співробітників на 10%.

### **Створення запитів на додавання до таблиць нових записів**

9. Створіть запит *СтворенняЗаголовокЗамовленняЗаЗразком* для створення в таблиці *ЗаголовкиЗамовлень* нового запису з даними обраного запису та активною датою замовлення (тобто для створення аналогічного заголовка замовлення, але від поточної дати). Для цього:

9.1. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиці *ЗаголовкиЗамовлень*;

9.2. Перетворіть створюваний запит на вибірку в запит на додавання записів до таблиці, натискаючи на вкладці *Конструктор* стрічки меню у групі *Тип запроса* кнопку *Добавление* або обираючи в контекстному меню вільної області джерела даних у підменю *Тип запроса* пункт *Добавление*. Після цього у вікні, що з'явиться на екрані, вкажіть назву таблиці для доповнення записами – *ЗаголовкиЗамовлень* та натисніть кнопку *ОК*;

9.3. Для копіювання даних співробітника та клієнта з обраного замовлення в новий запис перетягніть в перший та другий стовпці бланку запиту відповідно поля *КодСпівробітника* та *КодКлієнта* з таблиці *ЗаголовкиЗамовлень*;

9.4. З метою автоматичного встановлення в полі *ДатаЗамовлення* активної дати при створенні нового запису введіть для третього стовпця в рядку *Поле* вираз *Date()*, а в рядку *Добавление* оберіть зі списку поле *ДатаЗамовлення*;

9.5. Для доповнення таблиці даними лише обраного замовлення перетягніть в четвертий стовпець бланку запиту поле *КодЗамовлення*, **очистіть для нього значення в рядку *Добавление*** та вкажіть умову відбору *[Введіть код замовлення]*;

9.6. Збережіть створений запит та переконайтеся в його дієздатності. Перегляньте та обґрунтуйте структуру запиту в режимі SQL.

10. Самостійно створіть запит *СтворенняЗаголовкаПостачання-ЗаЗразком* для створення в таблиці *ЗаголовкиПостачань* нового запису з даними обраного запису та активною датою оформлення постачання.

### **Створення запитів на видалення записів таблиць**

11. Створіть запит *ЗнищенняЗайвихТоварів* для автоматичного знищення у таблиці *Товари* записів з даними товарів, що не постачалися і не замовлялися. Для цього:

11.1. В схемі даних відкоригуйте зв'язки між таблицею *Товари* та таблицями *ПунктиПостачань* і *ПунктиЗамовлень* так, щоб **заборонити** каскадне знищення даних для уникнення вилучення товарів, дані яких використовуються в постачаннях чи замовленнях. Для цього в контекстному меню відповідних зв'язків по чергово оберіть пункт *Изменить связь* і у вікні, що з'явиться на екрані, зніміть чи переконайтеся, що вже знятий прапорець *Каскадное удаление связанных записей*;

11.2. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиці *Товари*;

11.3. Перетворіть створюваний запит на вибірку в запит на знищення записів, натискаючи на вкладці *Конструктор* стрічки меню у групі *Тип запроса* кнопку *Удаление* або обираючи в контекстному меню вільної області джерела даних у підменю *Тип запроса* пункт *Удаление*;

11.4. Для знищення записів саме з таблиці *Товари* перетягніть в перший стовпець бланку запиту символ \* (позначення всіх полів) з її праобразу у верхній частині вікна;

11.5. Збережіть створений запит та переконайтеся в його дієздатності. Перегляньте та обгрунтуйте структуру запиту в режимі SQL.

12. Самостійно створіть запити на знищення записів *ЗнищенняЗайвихПунктівЗамовлень* (для знищення тих записів з таблиці *ПунктиЗамовлень*, в яких кількість не вказана або рівна нулю), *ЗнищенняЗайвихПунктівПостачань* (для знищення тих записів з таблиці *ПунктиПостачань*, в яких кількість не вказана або рівна нулю), *ЗнищенняЗайвихПостачальників*, *ЗнищенняЗайвихКлієнтів* (заборонивши перед цим відповідні каскадні знищення в схемі даних).

### **Керування запитом на корегування з головної кнопкової форми**

13. Для впорядкування всіх запитів на корегування даних забезпечте їх виклик безпосередньо з головної кнопкової форми. Для цього:

13.1. В режимі конструктора доповніть набір вкладок форми *Головна* новою вкладкою *Корегування даних*. За допомогою пункту контекстного меню набору вкладок

*Последовательность вкладок* розташуйте її перед вкладкою *Довідники*;

- 13.2. Для завантаження кожного запиту на корегування створіть на цій вкладці відповідні кнопки, обираючи у вікні *Создание кнопок* майстра створення кнопок категорію *Разное* та дію *Выполнить запрос*. Для кожного з запитів, назви яких не виводяться у вікні майстра (наприклад, для запитів на вилучення), створіть кнопку, яка викликає інший запит на корегування, а потім змініть назву завантажуваного запиту на потрібну в макросі *События* ► *Нажатие кнопки* вікна властивостей створеної кнопки;
- 13.3. Згрупуйте кнопки за типом операцій корегування, кожну групу кнопок помістіть в окремий прямокутник;
- 13.4. Збережіть зміни структури головної кнопкової форми та переконайтеся в її дієздатності.

### **Завершальний етап заняття. Повторення вивченого матеріалу**

14. Стисніть створену БД;
15. Завершіть роботу Access;
16. Віднайдіть створену БД та заархівуйте її вміст;
17. Збережіть створений архів на власному флеш-носії в двох екземплярах.
18. Доповніть власну БД запитамі на корегування, аналогічними до створених на лабораторній роботі.

### **Контрольні запитання**

1. У чому полягає основна відмінність між запитамі на вибірку і запитамі на корегування?
2. У чому полягає відмінність між дією запитів на корегування та самостійним корегуванням даних?
3. Як двома способами перетворити запит на вибірку в запит на корегування?
4. Які різновиди запитів на вибірку і запитів на корегування використовуються в Access?
5. Який синтаксис запису мають різні варіанти запитів на корегування в режимі SQL?
6. Які поля містить вікно будівничого виразів? Яке їх функціональне призначення?
7. Яке обмеження накладається на назви таблиць, що створюються запитамі?
8. Поля якого типу не змінюються запитамі на оновлення? Чому?
9. Які поля мають обов'язково заповнюватися запитамі на додавання записів? Як ще можливо реалізувати таке заповнення?
10. Як використовуються параметри зв'язків між таблицями при виконання запитів на вилучення?
11. Як здійснюється керування запитамі на корегування у розроблених додатках?



## Лабораторна робота № 13

**Тема.** Створення звітів за допомогою майстра.

**Мета.** Формування вмінь та навичок використання майстра для створення звітів. Закріплення вмінь та навичок використання майстрів для формування об'єктів додатку. Систематизація та застосування знань стосовно джерела даних, рівнів групування, порядку сортування даних, формування зовнішнього вигляду проміжних та загального підсумків звіту. Усвідомлення ролі встановлених зв'язків у схемі даних при формуванні звітів з даних декількох таблиць.

### Теоретичні відомості

Для публікації результатів обробки інформації БД найчастіше використовують **звіти**, оскільки вони дають змогу:

- подавати дані у вигляді, зручному для користувача;
- відображати і сортувати дані як **вкладених рівнів групування**, так і окремих записів джерела;
- підраховувати підсумки по кожній групі та загальні підсумки по всьому звіту.

Структура та властивості звітів подібні до форм: і ті і інші складаються з розділів, в яких розміщуються елементи керування, а джерелом даних цих типів об'єктів можуть бути таблиці чи запити. Але форми орієнтовані на корегування даних та активну взаємодію з користувачем, тому й можуть містити такі елементи керування, як *форма навігації* чи *веб-браузер* та мають ширший перелік подій у вікні властивостей. А звіти напрямлені на **публікацію результатів** обробки даних, тому й містять статичну інформацію, проте в них можливо відобразити як результати багаторівневого групування, так і дані вкладених записів. Крім цього, у формах, як правило, відображають всі редаговані поля джерела, а у звітах – лише ті, дані з яких потрібно опублікувати.

Для звітів використовуються два режими зміни структури (конструктора і макета) та два режими застосування – *Представление отчета*, де дані відображаються без масштабування, як в електронному документі та *Предварительный просмотр*, у якому дані виводяться на образах сторінок, як вони будуть надруковані. Параметри сторінки для друку звіту задаються на вкладці стрічки меню з такою ж назвою. Ширина звіту не повинна перевищувати ширини друкованого поля сторінки, бо інакше звіт буде виводитися по ширині на декількох сторінках.

Формуючи різні рівні сортування та групування, на основі однакових полів джерела, можливо розробити відмінні звіти для аналізу різних аспектів даних. Наприклад, якщо джерело даних з полів

НазваВідділу таблиці Відділи та ПІБ і Посада з таблиці Співробітники згрупувати на першому рівні за полем НазваВідділу, то отримаємо списки співробітників кожного відділу з зазначенням їх посад (рис. 65). Коли ж ті самі записи джерела згрупувати на першому рівні за полем Посада, то у звіті виведуться переліки співробітників, що обіймають однакові посади, та відділів, де вони працюють (рис. 66). При цьому другий (вкладений) рівень сортування та групування двох наведених звітів забезпечує сортування даних співробітників за полем ПІБ в межах кожної групи першого рівня.

Назва відділу	Прізвище, ім'я, по батькові	Посада співробітника
Відділ кадрів	Белова Антоніна Валеріївна	інспектор
	Зварич Олег Володимирович	начальник відділу
Бухгалтерія	Крилова Інна Станіславівна	голов.бухгалтер
	Петренко Тарас Степанович	бухгалтер
	Акбаєв Іван Петрович	агент
Відділ постачань	Акбаєв Іван Петрович	агент
	Безпалько Ірина Сергіївна	агент

Рис. 65. Фрагмент звіту *СпівробітникиВідділів* в режимі *Представление отчета*

Посада	Прізвище, ім'я, по батькові	Назва відділу
агент	Акбаєв Іван Петрович	Відділ постачань
	Безпалько Ірина Сергіївна	Відділ постачань
	Воронова Ольга Степанівна	Відділ збуту
	Кравцов Петро Олексійович	Відділ реклами
	Новиков Андрій Павлович	Відділ реклами
	Ясенева Олена Вікторівна	Відділ постачань
	Петренко Тарас Степанович	Бухгалтерія
бухгалтер	Петренко Тарас Степанович	Бухгалтерія

Рис. 66. Фрагмент звіту *ПосадиСпівробітників* в режимі *Представление отчета*

В Access можливо сформулювати до десяти вкладених рівнів сортування і групування. В режимі конструктора звітів, додатково до розділів,

притаманних формам, для кожного з цих рівнів можна створити як розділ заголовка, так і розділ приміток, де, відповідно, забезпечити вивід згрупованих даних та підсумків по кожній групі.

Створювати звіти в Access можна п'ятьма способами, кожен з яких викликається відповідною кнопкою групи стрічки меню *Создание* ► *Отчеты* (рис. 67):

- *Отчет* – забезпечує створення простого звіту табличного вигляду на основі даних об'єкта, виділеного в області переходів. Після створення звіт відкривається в режимі макету для подальшого вдосконалення його структури;
- *Конструктор отчетов* – створює порожній звіт та відкриває його для доопрацювання в режимі конструктора;
- *Пустой отчет* – аналогічно попередньому способу, генерує порожній звіт, але відкриває його для вдосконалення в режимі макету;
- *Мастер отчетов* – дає змогу користувачу обрати поля для відображення у звіті, задати рівні групування та параметри розрахунку підсумків як по даних окремих таблиць, так і по окремих полях, вказати поля сортування записів джерела даних, обрати макет, стиль оформлення та орієнтацію сторінки, ввести ім'я звіту, після чого майстер автоматично створить звіт відповідно до вказаних параметрів;
- *Наклейки* – завантажує майстер створення наклеюк (прямокутників стандартного розміру) на основі даних об'єкта, виділеного в області переходів. На першому кроці цього майстра обирається розмір окремої наклеюки, на другому – вказуються параметри шрифту для друку їх тексту, на третьому – формується прототип наклеюки з сталого тексту і полів джерела даних, на четвертому – задаються поля сортування джерела даних, а на останньому – вводиться ім'я звіту.

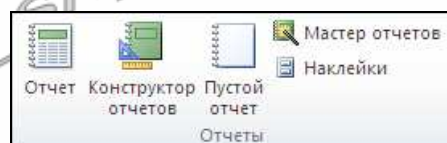


Рис. 67. Кнопки вкладки *Создание* стрічки меню для створення звітів

На практиці початковий вигляд звітів найчастіше задають за допомогою майстра, а вдосконалюють їх структуру в режимах конструктора та макету. Тому в цій лабораторній роботі вивчимо можливості створення звітів за допомогою майстра, а в наступній – навчимося використовувати засоби режиму конструктора звітів.

Література: [4, С. 147-149; 1, С. 289-295]

### Підготовчий етап заняття. Актуалізація знань

1. Завантажте Access, відкрийте розроблену раніше БД *Sklad*.
2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип объекта*.
3. Перейдіть в області переходів до розділу *Отчеты*.

### Створення звітів на основі даних таблиць

4. Створіть за допомогою майстра звіт *АлфавітнийСписокВідділів1* для відображення відсортованого за зростанням списку назв відділів з зазначенням їх місця розташування, виконуючи наступні дії:
  - 4.1. Завантажте майстер створення звітів, натиснувши на стрічці меню на вкладці *Создание* у групі *Отчеты* кнопку *Мастер отчетов*;
  - 4.2. На першому кроці майстра сформуєте перелік полів, що будуть відображатися в звіті. Для цього оберіть в списку *Таблицы и запросы* таблицю *Відділи* та перенесіть з списку доступних в список обраних полів всі поля, призначені для відображення даних звіту (в даному випадку – це поля *НазваВідділу* та *МісцеРозташування*). Після формування списку обраних полів натисніть кнопку *Далее*;
  - 4.3. На другому кроці майстра не формуйте рівні групування, оскільки в звіті має відображатися звичайний список відділів, а лише натисніть кнопку *Далее*;
  - 4.4. На третьому кроці оберіть поле *НазваВідділу*, згідно якого мають сортуватися дані записів звіту, вкажіть порядок сортування даних обраного поля за зростанням, встановивши кнопку-перемикач біля списку поля в положення *По возрастанию* та натисніть кнопку *Далее*;
  - 4.5. На четвертому кроці майстра оберіть на власний розсуд макет звіту, встановіть книжну орієнтацію сторінки та натисніть кнопку *Далее*;
  - 4.6. На останньому кроці майстра введіть назву звіту та натисніть кнопку *Готово*;
  - 4.7. Перегляньте отриманий звіт. Як називається режим використання звітів? Звідки майстер створення звітів скопіював текст заголовків полів? Яка різниця у відображенні даних режимами експлуатації форм та звітів?
5. Створіть аналогічні звіти *АлфавітнийСписокВідділів2* та *АлфавітнийСписокВідділів3*, обираючи на четвертому кроці майстра інші макети звітів. Яка різниця у відображенні даних створеними звітами? Серед трьох створених звітів оберіть той, що найкраще та найкомпактніше відображає дані базової таблиці (з табличним макетом), знищіть два інші подібні звіти та

перейменуйте залишений звіт, зберігши його під назвою *АлфавітнийСписокВідділів*.

6. Самостійно створіть звіти на основі відповідних таблиць:
  - 6.1. *АлфавітнийСписокКлієнтів* – для відображення в алфавітному порядку назв клієнтів з зазначенням дати реєстрації та місця роботи;
  - 6.2. *АлфавітнийСписокПостачальників* – для перегляду в алфавітному порядку назв постачальників з зазначенням коду ЄДРПОУ та дати реєстрації.
7. Порівняйте відображення даних створеними звітами та запитам з аналогічними назвами. Які об'єкти наочніше відображають дані таблиць?

### Створення звітів на основі запитів

8. Створіть звіт *АлфавітнийСписокСпівробітників* для перегляду та друку алфавітного списку співробітників з зазначенням дати народження та віку. Для цього:
  - 8.1. Розпочніть створення звіту за допомогою майстра;
  - 8.2. Оскільки всі необхідні дані для формування звіту містяться лише в запиті *АлфавітнийСписокСпівробітників*, то на першому кроці майстра, формуючи перелік полів для відображення в звіті, виберіть назву саме цього об'єкта в списку *Таблиці и запросы* та перенесіть з списку доступних в список обраних всі його поля;
  - 8.3. На другому кроці майстра не формуйте рівні групування;
  - 8.4. На третьому кроці оберіть поле *ПІБ* для сортування даних записів звіту та вкажіть порядок сортування даних обраного поля за зростанням;
  - 8.5. На четвертому кроці майстра оберіть *табличний* макет звіту та встановіть книжну орієнтацію сторінки;
  - 8.6. Самостійно завершіть створення звіту та переконайтеся в його дієздатності.
9. Самостійно створіть звіти на основі відповідних запитів:
  - 9.1. *ДніНародженняСпівробітників* – для відображення прізвищ співробітників, відсортованих за зростанням місяців і чисел їх народження та кількості років, що виповнилися (виповняться) кожному співробітнику в цьому році;
  - 9.2. *АлфавітнийСписокТоварів* – для перегляду назв товарів в алфавітному порядку з зазначенням для кожного з них відповідної марки, одиниці виміру, виробника та мінімальної дати виготовлення.
10. Самостійно **створіть три звіти** та обґрунтуйте їх структуру і призначення на основі відповідних запитів, обираючи з наведеного нижче переліку попередній, однаковий та наступний підпункти відносно вашого номера в журналі групи:

- 10.1. *ОкладиСпівробітників;*
- 10.2. *ОсвітченіСпівробітники;*
- 10.3. *ПрацюючіПенсіонери;*
- 10.4. *СпівробітникиБезЗасобівЗв'язку;*
- 10.5. *СпівробітникиБезТелефонів;*
- 10.6. *СпівробітникиЗEmail;*
- 10.7. *СпівробітникиЗВищоюОсвітою;*
- 10.8. *СтажРоботиВТоваристві;*
- 10.9. *ДесяткаОсновнихКлієнтів;*
- 10.10. *ДесяткаОсновнихПостачальників;*
- 10.11. *ДесяткаОсновнихТоварів;*
- 10.12. *ТовариБезПостачань;*
- 10.13. *ТовариБезЗамовлень;*
- 10.14. *ПостачальникиБезПостачань;*
- 10.15. *КлієнтиБезЗамовлень;*
- 10.16. *ПостачанняБезПунктів;*
- 10.17. *ЗамовленняБезПунктів;*
- 10.18. *ЗамовленняНаМаксимальнуСуму;*
- 10.19. *КількостіНародженихСпівробітниківПоДняхТижня;*
- 10.20. *КількостіПрацюючихСпівробітниківПоСтатях;*
- 10.21. *КількостіПрацюючихСпівробітниківПоВідділах;*
- 10.22. *КількостіПрацюючихСпівробітниківПоНаціональностях;*
- 10.23. *МаксимальніСумиЗамовленьПоКлієнтах;*
- 10.24. *МаксимальніСумиПостачаньПоПостачальниках;*
- 10.25. *МінімальніСумиЗамовленьПоКлієнтах;*
- 10.26. *МінімальніСумиПостачаньПоПостачальниках;*
- 10.27. *ПостачанняНаМаксимальнуСуму;*
- 10.28. *РейтингКлієнтів;*
- 10.29. *РейтингПостачальників;*
- 10.30. *РейтингТоварів;*
- 10.31. *ЗалишокТоварівНаСкладі;*
- 10.32. *КількостіОтриманихТоварів;*
- 10.33. *КількостіЗамовленихТоварів.*

#### **Групування даних та формування підсумків у звітах**

11. Створіть звіт *АсортиментВиробниківТоварів* для відображення асортименту товарів кожного зареєстрованого в системі виробника. Для цього:
  - 11.1. Розпочніть створення звіту за допомогою майстра;
  - 11.2. На першому кроці майстра оберіть таблицю *Товари* для формування джерела даних звіту та перенесіть з списку доступних в список обраних поля *НазваТовару*, *МаркаТовару*, *ОдиницяВиміру*, *Придатно*, *Виробник* та *ДСТУ*;



- 11.3. На другому кроці майстра створіть рівень групування даних за виробником товарів, перемістивши поле *Виробник* з списку полів в макет звіту з рівнями групування;
  - 11.4. На третьому кроці оберіть поля *НазваТовару* та *МаркаТовару* для сортування записів груп звіту. Забезпечте сортування даних за зростанням значень обраних полів;
  - 11.5. Самостійно завершіть створення звіту та переконайтеся в його дієздатності.
12. Самостійно створіть аналогічні звіти:
- 12.1. *ЖінкиВідділів* – на основі запиту з такою ж назвою для відображення алфавітного списку жінок кожного відділу;
  - 12.2. *НеосвідченіСпівробітникиВідділів* – на основі запиту *НеосвідченіСпівробітники* для відображення алфавітного списку співробітників кожного відділу, які не закінчили жодного закладу освіти;
  - 12.3. *СумиОкремихЗамовленьПоКлієнтах* – на основі запиту *СумиЗамовлень* для відображення по кожному клієнту загальних сум та дат окремих замовлень;
  - 12.4. *СумиОкремихПостачаньПоПостачальниках* – на основі запиту *СумиПостачань* для відображення по кожному постачальнику загальних сум та дат окремих постачань;
  - 12.5. *ЗалишкиТоварівПоВідділах* – на основі запиту з такою ж назвою для відображення по кожному відділу алфавітного списку назв та марок товарів, що знаходилися чи знаходяться в обігу, з зазначенням відповідних отриманих та замовлених кількостей і залишків продукції.
13. Створіть звіт *ДаніЗамовленьЦьогоРокуПоТоварах* для відображення алфавітного списку назв та марок замовлених товарів з зазначенням відповідної дати замовлення, назви клієнта, кількості, ціни та суми, загальних сум замовлень по кожному та по всіх товарах. Для цього:
- 13.1. Розпочніть створення звіту за допомогою майстра;
  - 13.2. На першому кроці майстра оберіть запит *ДаніЗамовленьТоварівЦьогоРоку* для формування джерела даних звіту, оскільки він містить всі необхідні дані, та перенесіть з списку доступних в список обраних поля *НазваТовару*, *МаркаТовару*, *ДатаЗамовлення*, *НазваКлієнта*, *Кількість*, *Ціна* та *Сума*;
  - 13.3. На другому кроці майстра, почергово перебираючи вигляди подання даних, перегляньте та проаналізуйте варіанти розміщення полів джерела даних у правій частині вікна. Для аналізу замовлень окремих товарів оберіть вид подання даних згідно таблиці *Товари*;

- 13.4. На третьому кроці майстра не формуйте рівні групування даних;
  - 13.5. На четвертому кроці майстра самостійно забезпечте сортування даних звіту за зростанням значення поля *ДатаЗамовлення*. Для відображення загальних сум замовлень по кожному та по всіх товарах натисніть кнопку *Итоги* та встановіть у відповідному вікні прапорець для обчислення групової операції *Sum* для поля *Сума*;
  - 13.6. Самостійно завершіть створення звіту та переконайтеся в його дієздатності.
14. Створіть звіт *ДаніЗамовленьТоварівЦьогоРокуПоДатах* для відображення по кожній даті алфавітного списку назв та марок замовлених товарів з зазначенням відповідної назви клієнта, кількості, ціни та суми і підсумків замовлень по кожній та по всіх датах з відображенням їх процентної частини стосовно загальної суми. Для цього:
- 14.1. Розпочніть створення звіту за допомогою майстра;
  - 14.2. На першому кроці майстра сформуєте джерело даних, аналогічне попередньому звіту;
  - 14.3. На другому кроці для аналізу окремих замовлень по кожній даті оберіть вид подання даних згідно таблиці *ПунктиЗамовлень* (всі інші варіанти видів подання даних або не містять поля *ДатаЗамовлення* на верхньому рівні групування або поєднують його з іншими полями);
  - 14.4. На третьому кроці майстра для групування даних звіту по окремих датах сформуєте відповідний рівень групування, перемістивши поле *ДатаЗамовлення* з переліку полів джерела даних в макет рівнів групувань звіту. Для забезпечення групувань саме по окремих датах (а не по місяцях, як це встановлюється для полів цього типу даних по замовчуванню) на цьому ж кроці майстра натисніть кнопку *Группировка*, вкажіть у відповідному вікні для поля *ДатаЗамовлення* інтервал групування *обычный* та натисніть ОК;
  - 14.5. На четвертому кроці майстра самостійно забезпечте сортування даних звіту за зростанням значень полів *НазваТовару* та *МаркаТовару* і відображення загальних сум замовлень за кожною датою з їх процентними частками стосовно загальної суми всіх замовлень (при цьому зверніть увагу на прапорець *Вычислить проценты* у вікні *Итоги*);
  - 14.6. Самостійно завершіть створення звіту та переконайтеся в його дієздатності.
15. Створіть звіт *СумиЗамовленьЦьогоРокуПоДняхТижня* на основі того ж запиту *ДаніЗамовленьТоварівЦьогоРоку* для відображення

номера дня в тижні, відповідної загальної суми замовлень всіх товарів та процентної частки кожної суми стосовно загальної суми всіх замовлень. З цією метою:

- 15.1. На першому кроці майстра перенесіть з списку доступних в список обраних полів лише поля *ДеньЗамовлення* та *Сума*;
  - 15.2. На другому кроці для аналізу всіх сум замовлень за кожен день тижня забезпечте вигляд подання даних згідно таблиці *ЗаголовкиЗамовлень* (оскільки значення поля *ДеньЗамовлення* розраховується з поля *ДатаЗамовлення* цієї таблиці);
  - 15.3. На третьому кроці майстра для відображення лише загальних сум замовлень кожного дня тижня самостійно перейдіть у вікно *Итоги*, встановіть прапорець обчислення для поля *Сума* групової операції *Sum* та відмітьте перемикач *только итоги* в групі перемикачів *Показать*. У цьому ж вікні самостійно забезпечте відображення процентної частки кожної суми стосовно загальної суми всіх замовлень;
  - 15.4. Самостійно завершіть створення звіту та переконайтеся в його дієздатності.
16. Створіть звіт *СумиЗамовленьЦьогоРокуДляКожногоТоваруПоКварталах* на основі того ж запиту *ДаніЗамовленьТоварівЦьогоРоку* для відображення по кожному товару сум замовлень, згрупованих по кварталах. Для цього:
- 16.1. На першому кроці майстра оберіть той самий запит *ДаніЗамовленьТоварівЦьогоРоку* для формування джерела даних звіту та перенесіть з списку доступних в список обраних поля *НазваТовару*, *МаркаТовару*, *ДатаЗамовлення* та *Сума*;
  - 16.2. На другому кроці майстра для аналізу замовлень окремих товарів оберіть вигляд подання даних згідно таблиці *Товари*;
  - 16.3. На третьому кроці майстра для групування даних замовлень по кварталах сформууйте відповідний рівень групування, перемістивши поле *ДатаЗамовлення* з переліку полів джерела даних в макет рівнів групувань звіту (створений рівень групування має бути другим). Для забезпечення групувань саме по кварталах натисніть кнопку *Группировка*, вкажіть у відповідному вікні для поля *ДатаЗамовлення* інтервал групування *по кварталам* та натисніть *ОК*;
  - 16.4. На четвертому кроці майстра для відображення лише загальних сум замовлень кожного дня кварталу самостійно перейдіть у вікно *Итоги*, встановіть прапорець для обчислення групової операції *Sum* для поля *Сума* та відмітьте перемикач *только итоги* в групі перемикачів *Показать*;
  - 16.5. Самостійно завершіть створення звіту та переконайтеся в його дієздатності. Чи виводяться у звіті підсумки по виглядах подання, рівнях групування та загальні підсумки звіту?

17. Самостійно створіть аналогічні звіти:

- 17.1. *ДаніЗамовленьТоварівЦьогоРокуПоКлієнтах* – для відображення по кожному клієнту алфавітного списку назв та марок замовлених товарів з зазначенням відповідної дати замовлення, кількості, ціни та суми і підсумків замовлень по кожному та по всіх клієнтах з відображенням їх процентної частини стосовно загальної суми;
- 17.2. *СумиЗамовленьЦьогоРокуПоКлієнтах* – для відображення по кожному клієнту загальної суми замовлень всіх товарів та процентної частки кожної суми стосовно загальної суми всіх замовлень;
- 17.3. *СумиЗамовленьЦьогоРокуДляКожногоКлієнтаПоМісяцях* – для відображення по кожному клієнту сум замовлень, згрупованих по місяцях.

### **Створення звітів на основі перехресних запитів**

18. Створіть звіт *АналізСумЗамовленихТоварівЦьогоРокуПоДняхТижня* на основі запиту з такою ж назвою для відображення сум замовлень окремих товарів кожного дня тижня (в рядках звіту мають вказуватися дані замовлень по кожному товару, а в стовпцях – по кожному дню тижня). Для цього:

- 18.1. Завантажте майстер створення звітів, оберіть відповідне джерело даних та переконайтеся, що у списку доступних полів не відображаються **всі можливі** номери днів тижня. Саме тому поки що відмовтеся від послуг майстра створення звітів, оскільки внесені дані за інші дні тижня не будуть відображатися у створеному звіті (через відсутність відповідних полів);
- 18.2. В області переходів перейдіть до розділу *Запросы* та завантажте на виконання запит *АналізСумЗамовленихТоварівЦьогоРокуПоДняхТижня*. Зробіть висновок стосовно полів, що відображалися майстром створення звітів для даного запиту;
- 18.3. Відкрийте розглядуваний запит в режимі конструктора. Для відображення при виконанні запиту номерів всіх днів тижня у вікні властивостей запиту, що викликається, наприклад, з контекстного меню вільної області джерела даних запиту, для параметра *Заголовки столбцов* введіть рядок цих номерів: 1;2;3;4;5;6;7. Збережіть корегування структури запиту, перегляньте результати його виконання та обгрунтуйте наявність і походження порожніх стовпців;
- 18.4. Поверніться до розділу *Отчеты* та самостійно створіть за допомогою майстра звіт без рівнів групувань на основі всіх полів запиту *АналізСумЗамовленихТоварівЦьогоРокуПоДняхТижня* з альбомною орієнтацією сторінки.

19. Самостійно створіть чотири звіти на вибір, беручи за основу наступні перехресні запити та забезпечивши попередньо відображення при їх виконанні всіх можливих стовпців (для кварталів – "Кв1"; "Кв2"..., для місяців – 1; 2; 3; ... 12):
- 19.1. Аналіз Сум Замовлених Товарів Цього Року По Місяцях;
  - 19.2. Аналіз Сум Замовлених Товарів Цього Року По Кварталах;
  - 19.3. Аналіз Кількостей Замовлених Товарів Цього Року По Місяцях;
  - 19.4. Аналіз Кількостей Замовлених Товарів Цього Року По Кварталах;
  - 19.5. Аналіз Кількостей Замовлених Товарів Цього Року По Днях Тижня;
  - 19.6. Аналіз Кількостей Замовлень Цього Року По Клієнтах Та Місяцях;
  - 19.7. Аналіз Кількостей Замовлень Цього Року По Клієнтах Та Кварталах;
  - 19.8. Аналіз Кількостей Замовлень Цього Року По Клієнтах Та Днях Тижня.

### **Створення звітів-поштових наклейок**

20. Створіть звіт *Поштові Адреси Співробітників* за допомогою майстра для відображення поштових наклейок з прізвищами та ініціалами співробітників та їх домашніми адресами. Для цього:
- 20.1. В області переходів відмітьте таблицю *Співробітники* для формування джерела даних звіту;
  - 20.2. Завантажте майстер створення звітів-наклейок, натиснувши у стрічці меню на вкладці *Создание* у групі *Отчеты* кнопку *Наклейки*;
  - 20.3. На першому кроці майстра створення поштових наклейок оберіть за смаком параметри наклейок у два стовпці;
  - 20.4. На другому кроці майстра встановіть власні параметри шрифту та кольору тексту наклейки;
  - 20.5. На третьому кроці майстра перемістіть з переліку доступних полів в прототип наклейки поле *ПІБ*. Перейдіть в третій рядок прототипу наклейки та перенесіть у нього поле *Адреса*. Аналогічно в четвертий рядок прототипу перенесіть поле *Індекс*;
  - 20.6. На четвертому кроці майстра оберіть поле *ПІБ* для сортування наклейок, а на п'ятому – введіть назву звіту та завершіть його створення. Перегляньте звіт в режимі попереднього перегляду.
21. Самостійно створіть аналогічні звіти *Поштові Адреси Клієнтів* та *Поштові Адреси Постачальників*, обираючи на першому кроці майстра відмінні від попередніх звітів параметри поштових наклейок.

### **Завершальний етап заняття. Повторення вивченого матеріалу**

22. Стисніть створену БД.
23. Завершіть роботу Access.
24. Віднайдіть створену БД та заархівуйте її вміст.
25. Збережіть створений архів на власному носії в двох екземплярах.
26. Розробіть у власній БД за допомогою майстра звіти, аналогічні до створених на лабораторній роботі.

### Контрольні запитання

1. Як завантажити майстер створення звітів?
2. Які об'єкти можуть бути джерелами даних при створенні звітів?
3. Які поля переносяться в список обраних полів на першому кроці майстра?
4. Навіщо у звітах використовуються рівні групування?
5. Коли обираються поля сортування області даних звіту при його створенні за допомогою майстра? Як змінюється принцип сортування по значенню поля?
6. Які варіанти вигляду звіту пропонуються майстром на четвертому кроці? У яких випадках доцільно використовувати кожен з цих варіантів?
7. Де бере інформацію майстер Access про зв'язки між таблицями при генерації звітів?
8. Звідки копіюються властивості полів при створенні звітів?
9. Коли доцільно створювати звіти на основі таблиць, а коли – на основі запитів?
10. За яким принципом при створенні багаторівневих звітів обирають вигляд подання та рівні групування?
11. Як встановити інтервал групування та різновиди групових операцій для окремих полів? Над полями яких типів можна виконувати ці операції? Як забезпечити відображення у звіті лише підсумків кожної групи, процентної частки кожного підсумку стосовно загальних підсумків звіту? Коли не формуються загальні підсумки звіту?
12. Чому при створенні звіту на основі перехресних запитів на першому кроці майстра у списку доступних полів мають відобразитися всі можливі варіанти значень стовпців? Як цього досягти?
13. Які параметри вказуються на кожному кроці майстра при створенні поштових наклейок? Коли для цих звітів обирається джерело даних?



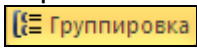
## Лабораторна робота № 14

**Тема.** Розробка та корегування властивостей звітів і їх елементів керування в режимі конструктора. Створення багатостовпцевих звітів.

**Мета.** Формування вмінь та навичок створення і корегування властивостей звітів та їх елементів керування в режимі конструктора, розробки засобів автоматичної нумерації рядків та відображення накопичувальних підсумків. Закріплення вмінь та навичок використання майстрів в БД. Вдосконалення вмінь та навичок створення запитів в режимі конструктора.

### Теоретичні відомості

Звіти створюють в режимі конструктора аналогічно формам: спочатку задають джерело даних (для цього у вікні властивостей звіту на вкладці *Данные* встановлюють значення поля *Источник записей*, обираючи існуючу таблицю чи запит або створюючи новий запит); потім виводять необхідні розділи; після цього створюють елементи керування в обраних розділах, перетягуючи поля зі списку полів джерела даних чи застосовуючи кнопки групи *Элементы управления*. Додатково у звітах після вибору чи формування джерела записів доцільно задати рівні групування та сортування. Звичайно, відсортувати записи джерела даних можливо безпосередньо у вікні властивостей звіту на вкладці *Данные*, у запиті згаданій властивості *Источник записей* чи вказуючи ім'я поля сортування у полі *Порядок сортировки*, але саме формування рівнів групування та сортування дає змогу виконати багаторівневе впорядкування записів джерела: рівні групування дозволяють відсортувати та вивести розділи заголовків і приміток для записів з однаковим значеннями полів чи виразів їх формування, в яких відображаються узагальнюючі надписи та підсумки кожної групи; рівні сортування виконують впорядкування записів, а в області даних відображаються дані кожного запису.

Задаються рівні групування та сортування в області *Группировка, сортировка и итоги* (рис. 68), яка активізується пунктом контекстного меню *Сортировка и группировка* цілого звіту чи довільного його розділу або за допомогою кнопки  вкладки *Конструктор* стрічки меню.

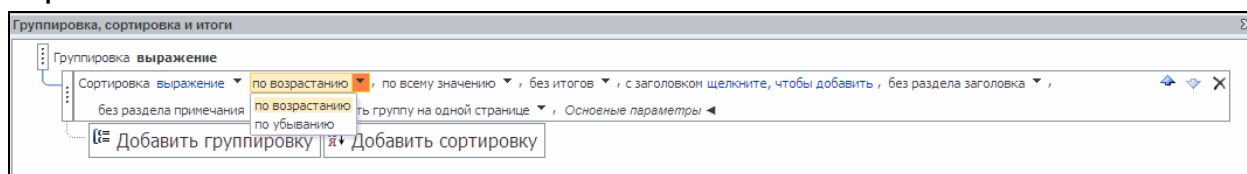


Рис. 68. Параметри другого рівня сортування в області *Группировка, сортировка и итоги* звіту *ДаніЗамовленьТоварівПоКлієнтам*

Для створення нового рівня групування (сортування) необхідно в нижній частині цієї області натиснути відповідну кнопку (див. рис. 68) та обрати поле чи ввести потрібний вираз у властивості *Группировка* (*Сортировка*) створеного рівня. З метою відображення всіх параметрів рівня групування (сортування) використовується його кнопка *Больше*, а для їх приховування – кнопка *Основные параметры*. Ці параметри дають змогу вказати порядок сортування (як на рис. 68), задати кількість знаків (для текстових) чи ширину діапазону (для числових полів) групування, сформувати нескладні підсумки та заголовки груп, обрати принцип відображення груп при переході на нову сторінку: розривати/не розривати заголовок/примітки групи та її записи або не відображати знизу сторінки заголовок групи без першого запису (рекомендується авторами практикуму).

Зрозуміло, що кожен вкладений рівень групування (сортування) впливає на опрацювання лише тих записів, в яких поля/вирази формування вищих рівнів однакові. Наприклад, для звіту *ДаніЗамовленьТоварівПоКлієнтах* доцільно сформувати перший рівень групування записів за даними клієнта, а за допомогою другого – відсортувати товари, замовлені кожним клієнтом, за зростанням.

В режимі конструктора звіту можна задавати властивості відображення не лише для цілого звіту чи елементів керування, а й для окремих розділів. Серед цих властивостей виділимо дві: *Сжатие* – дозволяє зменшити висоту елемента керування чи цілого розділу, якщо дані для відображення відсутні та *Расширение* – дає змогу збільшувати їх висоту, якщо дані повністю не відображаються в елементі керування.

Призначення різних типів елементів керування звітів такі самі, як у формах. Лише на вкладках *Данные* та *События* вікна властивостей вони містять значно менше параметрів, оскільки дані у звітах не редагуються. Зате у звітах для числових полів додатково використовується властивість *Сумма с накоплением* (з вкладки *Данные*). У списку цієї властивості можна встановити одне з трьох значень: *отсутствует* (накопичення не виконуються, встановлюється по замовчуванню), *для группы* (накопичення виконуються окремо для кожної групи першого рівня) чи *для всего* (накопичення виконуються для всіх записів звіту). Вибір другого чи третього значення властивості *Сумма с накоплением* забезпечує відображення у полі його значення з попереднього запису зі збільшенням на значення цього ж поля для поточного запису. Наприклад, задання в області даних для обчислювального поля з даними  $=1$  значення властивості *Сумма с накоплением* рівним *для всего* забезпечує наскрізну нумерацію записів звіту.

На завершення зазначимо, що результат обчислювального поля звіту залежить не лише від його виразу, а й від розділу, в якому це поле розміщується. Наприклад, обчислювальне поле

= $Sum(Кількість*Ціна)$  в примітках групи по даних товару у звіті *ДаніЗамовленьТоварівПоКлієнтах* виведе загальну суму закупки кожного товару окремим клієнтом, в примітках групи по даних клієнта – загальну суму закупки кожним клієнтом, а в примітках звіту – загальну суму закупки всіма клієнтами. Крім цього, у звітах (як і у формах) для кожного обчислювального поля у вікні властивостей на вкладці *Макет* необхідно вказувати коректний формат поля і число десяткових знаків.

Література: [4, С. 153-154; 1, С. 312-329]

### Підготовчий етап заняття. Актуалізація знань

1. Завантажте Access, відкрийте розроблену раніше БД *Sklad*.
2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип объекта*.
3. Перейдіть в області переходів до розділу *Запросы* та створіть запит *ДаніЗамовленьТоварівПоКлієнтах* в режимі конструктора для відображення алфавітного списку клієнтів, назв та марок замовлених ними товарів з зазначенням відповідних загальних кількостей та сум (цей запит в подальшому буде використано як джерело даних звітів). Для цього:
  - 3.1. Розпочніть створення запиту в режимі конструктора;
  - 3.2. Сформууйте джерело даних запиту з записів таблиць *Клієнти*, *ЗаголовкиЗамовлень*, *ПунктиЗамовлень* та *Товари*;
  - 3.3. Перетягніть в перший та другий стовпці бланку запиту відповідно поля *НазваКлієнта* та *МісцеРоботиКлієнта* з таблиці *Клієнти*;
  - 3.4. Перетягніть в третій та четвертий стовпці бланку запиту відповідно поля *НазваТовару* та *МаркаТовару* з таблиці *Товари*;
  - 3.5. У п'ятий стовпець бланку запиту перетягніть поле *Кількість* з таблиці *ПунктиЗамовлень*. Задайте для даного стовпця заголовок *ЗагальнаКількість*, який буде відображатися при виконанні запиту;
  - 3.6. Для шостого стовпця в рядку *Поле* введіть формулу для підрахунку суми пункту замовлення *Сума: Кількість\*Ціна*. Задайте для цього поля у вікні властивостей формат *С разделителями разрядов* та відображення двох десяткових знаків після коми;
  - 3.7. Оскільки запит повинен групувати записи за даними клієнтів та товарів і для кожної групи підраховувати загальну кількість та суму, то в рядку *Групповая операция* (його відображення забезпечте самостійно) для перших чотирьох полів оберіть зі списку значення *Группировка* (встановлюється по замовчуванню), а для поля п'ятого та шостого – значення *Sum*;
  - 3.8. Для першого, третього та четвертого стовпця встановіть ознаку сортування за зростанням;

3.9. Самостійно завершіть створення запиту та переконайтеся в його дієздатності.

### **Створення звітів та корегування їх властивостей в режимі конструктора**

4. В області переходів перейдіть до розділу *Отчеты* та розпочніть створення звіту *ДаніЗамовленьТоварівПоКлієнтах* для аналізу структури замовлень товарів окремими клієнтами, аналогічного наведеному на рис. 71. Для цього:
  - 4.1. Створіть порожній звіт одним з двох способів:
    - 4.1.1. Натиснувши у стрічці меню на вкладці *Создание* у групі *Отчеты* кнопку *Конструктор отчетов*;
    - 4.1.2. Натиснувши у стрічці меню на вкладці *Создание* у групі *Отчеты* кнопку *Пустой отчет* та перейшовши в режим конструктора.
5. Забезпечте відображення лінійки та сітки розмітки в режимі конструктора. Для цього в контекстному меню області даних встановіть відповідні прапорці.
6. Встановіть необхідні параметри сторінки для відображення звіту. З цією метою:
  - 6.1. На вкладці стрічки меню *Инструменты конструктора отчетов* ► *Параметры страницы* натисніть кнопку *Параметры страницы*;
  - 6.2. У виведеному вікні *Параметры страницы* перейдіть на вкладку *Параметры печати* та встановіть розміри всіх полів від краю сторінки до області виводу рівними 2 см (20 мм);
  - 6.3. У вкладці *Страница* цього ж вікна встановіть книжну орієнтацію сторінки та стандартні розміри паперу формату А4. Які ще параметри та з якою метою задаються на цій вкладці?
  - 6.4. На вкладці *Столбцы* цього ж вікна переконайтеся, що встановлено один стовпець для відображення даних звіту. Встановіть ширину цього стовпця рівною 16,9 см. (Чому саме стільки?);
  - 6.5. Натисніть *ОК* для збереження внесених змін параметрів сторінки при відображенні звіту. Чи можна аналогічні зміни вносити в режимі попереднього перегляду?
7. Встановіть ширину звіту рівною приблизно 16,9 см. Для цього перемістіть покажчик мишки на праву межу області звіту (покажчик мишки набуде вигляду лінії з двома стрілками) та відтягніть її до позначки 16,9 см.
8. Встановіть точне значення ширини звіту у вікні властивостей. Для цього:
  - 8.1. Зробіть активним цілий звіт, відмітивши його область виділення.

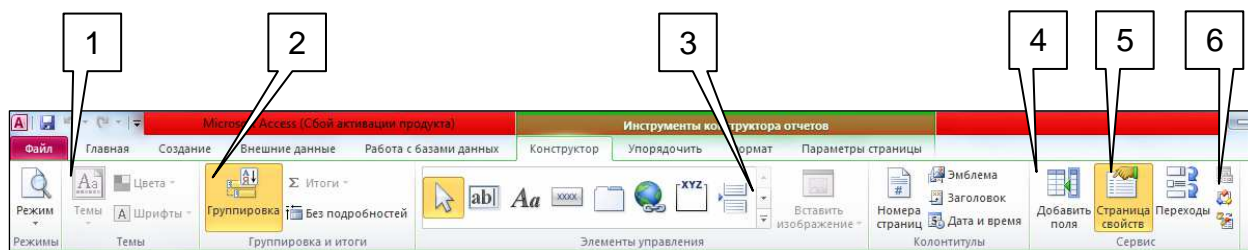


Рис. 69. Основні кнопки вкладки *Конструктор* режиму конструктора звітів:

- 1 – вибір теми для оформлення стандартизованих звітів;
- 2 – режим видимості вікна групування, сортування та підсумків даних;
- 3 – група елементів керування;
- 4 – режим видимості списку полів джерела даних звіту;
- 6 – режим видимості вікна властивостей звіту, розділів та елементів керування;
- 5 – відображення модуля процедур обробки подій звіту

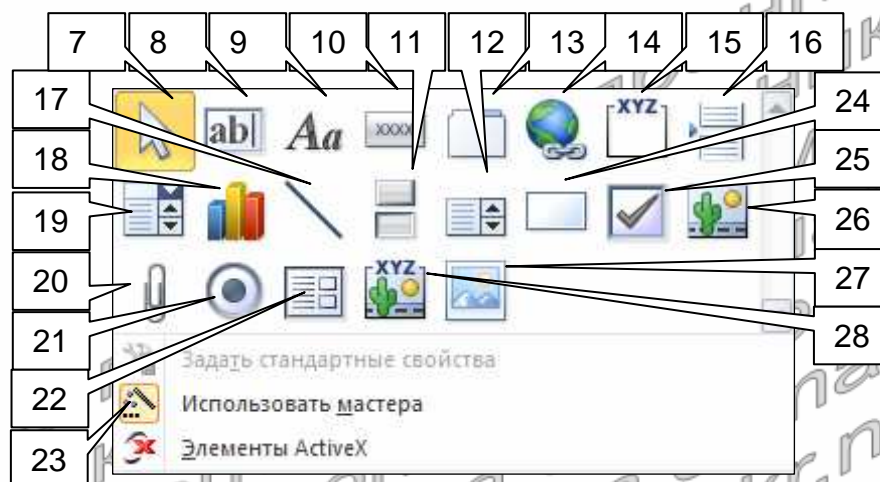


Рис. 70. Основні кнопки для створення елементів керування звітів:

- 7 – кнопка групового вибору об'єктів мишкою;
- 8 – кнопка для створення поля;
- 9 – кнопка для створення надпису;
- 10 – кнопка для створення кнопки;
- 11 – кнопка для створення перемикача;
- 12 – кнопка для створення списку;
- 13 – кнопка для створення набору вкладок;
- 14 – кнопки гіперпосилань та елементів навігації;
- 15 – кнопка для створення групи перемикачів;
- 16 – кнопка для вставки розриву сторінки;
- 17 – кнопка для створення лінії;
- 18 – кнопка для створення діаграми;
- 19 – кнопка для створення поля зі списком;
- 20 – кнопка для прикріплення вкладених об'єктів;
- 21 – кнопка для створення вимикача;
- 22 – кнопка для створення підпорядкованого звіту;
- 23 – кнопка для забезпечення виклику майстрів елементів керування;
- 24 – кнопка для створення прямокутника;
- 25 – кнопка для створення прапорця;
- 26 – кнопка для вставки малюнка;
- 27 – кнопка для вставки вільної рамки об'єкта;
- 28 – кнопка для вставки приєднаної рамки об'єкта



### Дані замовлень товарів по клієнтах станом на 22.03.2003

Клієнт: *Зінчук Т.Я.*

Місце роботи: *Бізнес-центр  
"Партнер"*

№ п/п	Назва та марка товару	Замовлена кількість	Замовлено на суму
1	Кава 'Галка'	2	8,00
2	Майонез 'Чумак'	10	12,50
3	Мінеральна вода 'Свалява'	2	3,00
4	Молоко 'Ласуня'	7	10,64
5	Хліб 'Дарницький'	6	6,00
6	Шоколад 'Світоч'	1	2,00
<b>Всього замовлено товарів клієнтом на суму:</b>			<b>42,14</b>

...

Сторінка 1 з 4

### Дані замовлень товарів по клієнтах станом на 22.03.2003 (продовження)

Клієнт: *Прохоров К.Н.*

Місце роботи: *спортивний клуб  
"Марафон"*

№ п/п	Назва та марка товару	Замовлена кількість	Замовлено на суму
1	Кава 'Галка'	2	12,00
2	Мінеральна вода 'Свалява'	1	3,00
3	Молоко 'Ласуня'	3	6,00
4	Хліб 'Дарницький'	1	1,00
5	Шоколад 'Світоч'	17	51,00
<b>Всього замовлено товарів клієнтом на суму:</b>			<b>73,00</b>

...

**Всього замовлено товарів всіма клієнтами на суму:**

**122,59**

Сторінка 4 з 4

Рис. 71. Фрагменти звіту *ДаніЗамовленьТоварівПоКлієнтах* в режимі попереднього перегляду



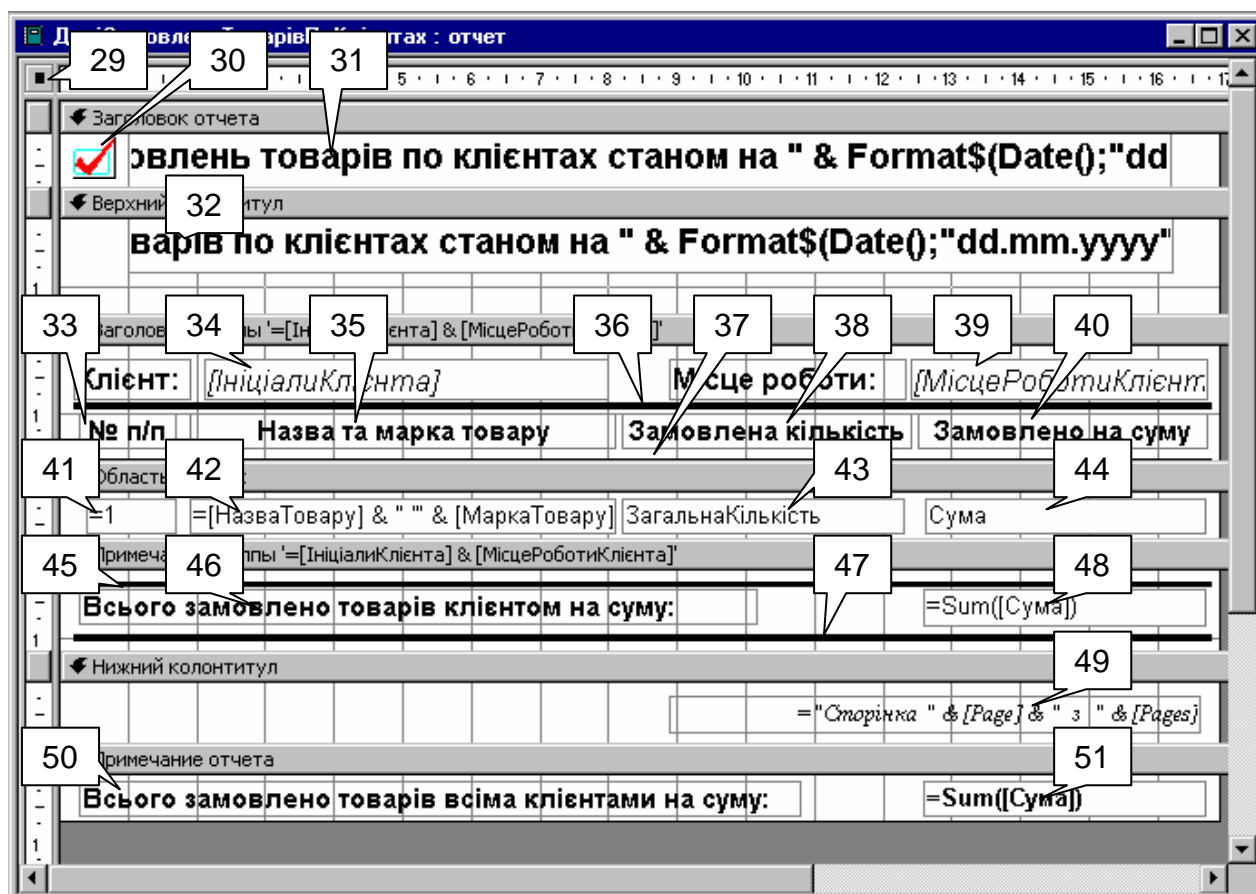


Рис. 72. Структура звіту *ДаніЗамовленьТоварівПоКлієнтах* в режимі конструктора

З цією метою активізуйте область (29). Тут і надалі у цій лабораторній в круглих дужках наведено номери виносок з рис. 69-72. Тому для прискорення виконання роботи ці рисунки настійливо рекомендуємо роздрукувати;

- 8.2. На вкладці *Макет* вікна властивостей віднайдіть параметр *Ширина* та встановіть його значення рівним 16,9 см.
9. Для відображення областей заголовка та приміток звіту встановіть прапорець *Заголовок/примечание отчета* в контекстному меню області даних. Самостійно приховуйте та відобразіть знову області верхнього та нижнього колонтитула.
10. Встановіть висоту області *Заголовок отчета* рівною приблизно 0,8 см. Для цього розташуйте покажчик мишки над областю заголовка верхнього колонтитула та перетягніть її на позначку 0,8 см по шкалі вертикальної лінійки.
11. Встановіть точне значення висоти цієї області у вікні властивостей. З цією метою:
  - 11.1. Зробіть активною область *Заголовок отчета*;
  - 11.2. На вкладці *Макет* вікна властивостей віднайдіть параметр *Висота* та встановіть його значення рівним 0,8 см.
12. Самостійно встановіть висоту верхнього колонтитула рівною 1,4 см, області даних – 0,7 см, нижнього колонтитула – 0,9 см, а області приміток звіту – 0,7 см.

13. Задайте заголовок вікна створеного звіту для режиму перегляду. З цією метою: у вікні властивостей звіту перейдіть на вкладку *Макет* та введіть у поле властивості *Подпись* текст *Дані замовлень товарів по клієнтах*.
14. Оцініть вигляд створеного звіту в режимі попереднього перегляду (на вкладці стрічки меню *Файл* ► *Печать* чи натиснувши відповідну кнопку в рядку стану). Де виводиться заголовок вікна звіту? Поверніться в режим конструктора.
15. Встановіть у вікні властивостей звіту на вкладці *Макет* значення параметрів згідно табл. 20.

Таблиця 20

**Значення окремих властивостей звіту *ДаніЗамовленьТоварівПоКлієнтах***

Властивість	Значення
Автоматический размер	Да
Выравнивание по центру	Да
Верхний колонтитул	Без заголовка
Нижний колонтитул	Все страницы

Чому верхній колонтитул не повинен відображатися на першій сторінці звіту? Закрийте звіт та дайте йому назву *ДаніЗамовленьПоКлієнтах*. Відкрийте звіт в режимі попереднього перегляду. Як перші два встановлені параметри вплинули на вигляд звіту? Знову поверніться в режим конструктора.

16. Забезпечте можливість автоматичного розширення області даних при розширенні одного з її полів внаслідок перевищення розміру даних поля розмірів, відведених йому в макеті. З цією метою у вікні **властивостей області даних** перейдіть на вкладку *Макет* та оберіть у списку значень властивості *Расширение* параметр *Да*.
17. Встановіть для розроблюваного звіту джерело даних – запит *ДаніЗамовленьТоварівПоКлієнтах*. Для цього у вікні властивостей звіту перейдіть на вкладку *Данные* та у полі зі списком *Источник записей* оберіть назву зазначеного запиту.
18. Натисніть кнопку списку полів джерела даних звіту (4). Звідки беруться назви полів у виведеному вікні?

**Формування рівнів групування та сортування даних звіту в режимі конструктора**

19. Для забезпечення групування даних замовлень по окремих клієнтах та сортування даних товарів, придбаних кожним клієнтом за зростання значень назви та марки товару, виконайте наступні дії:
  - 19.1. Відкрийте область *Группировка, сортировка и итоги* звіту відповідним пунктом контекстного меню області виділення звіту (29) чи за допомогою кнопки (2) вкладки *Конструктор* стрічки меню;
  - 19.2. Для забезпечення групування даних замовлень по окремих клієнтах в полі *Группировка* **першого** рівня групування верхньої частини вікна (рис. 73) викличте будівничого виразів

та введіть у ньому вираз  $=[НазваКлієнта] \& [МісцеРоботиКлієнта]$  (рис. 74).

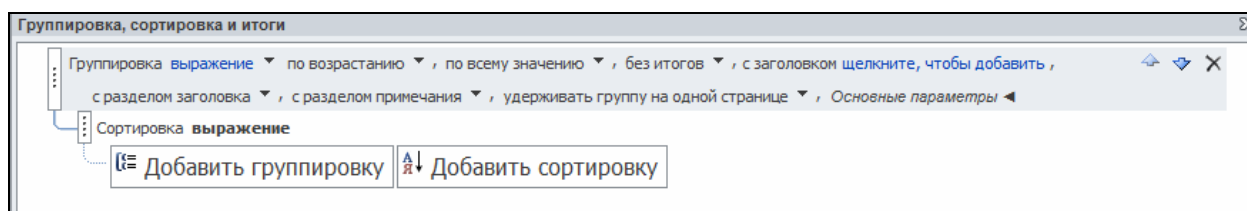


Рис. 73. Вигляд області *Групування, сортування та ітоги* звіту *ДаніЗамовленьТоварівПоКлієнтах*

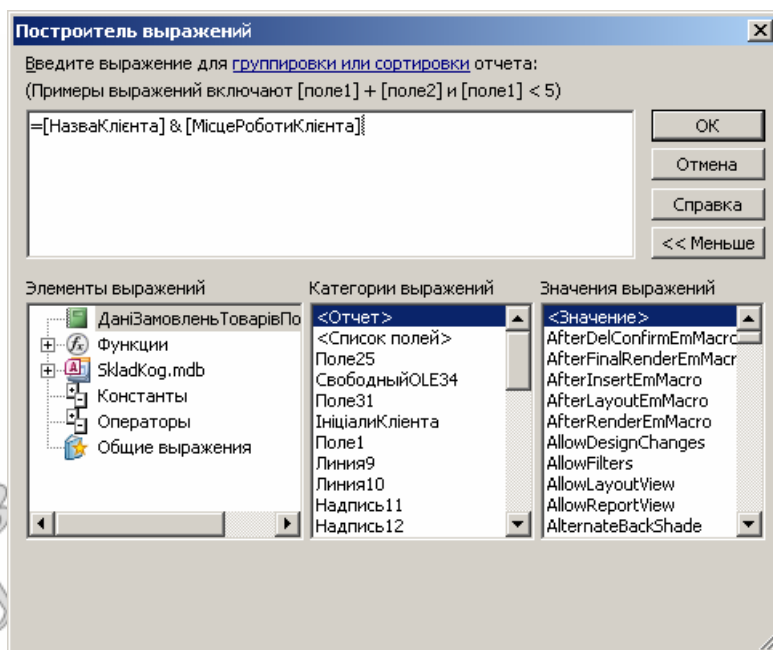


Рис. 74. Вигляд вікна будівничого виразів для першого рівня групування звіту

З метою забезпечення впорядкування даних клієнтів у звіті за зростанням значень введеного виразу переконайтеся, що значення наступного поля **цього ж рядка** рівне по **возрастанію**. Додаткові параметри рівня групування встановіть згідно перших двох стовпців табл. 21. Зверніть увагу на призначення значень встановлених параметрів;

Таблиця 21  
Значення окремих властивостей першого рівня групування звіту *ДаніЗамовленьТоварівПоКлієнтах*

Властивість	Значення	Призначення
Раздел заголовка	Да	Для відображення даних кожного клієнта
Раздел примечания	Да	Для відображення підсумків по замовленнях кожного клієнта
Группировка	По всему значению	Для відображення даних кожного клієнта окремо
Удерживать группу на одной странице	Да	Для недопущення розміщення даних кожного з клієнтів на різних сторінках

- 19.3. З метою встановлення сортування товарів, придбаних кожним клієнтом за зростанням значень відповідних назв та марок товарів, створіть вкладений (другий) рівень групування  $\text{=[НазваТовару]} \& \text{[МаркаТовару]}$  без відображення його заголовка та приміток. Обгрунтуйте значення встановлених вами параметрів;
- 19.4. Закрийте область *Групування, сортування та ітоги*. Які розділи і чому з'явилися у структурі звіту? Самостійно встановіть висоту заголовка групи  $\text{=[НазваКлієнта]} \& \text{[МісцеРоботиКлієнта]}$  рівною 1,7 см, а її приміток – 1,2 см. Забезпечте також можливість автоматичного розширення заголовка цієї групи при розширенні одного з її полів внаслідок перевищення розміру даних поля розмірів, відведених йому в макеті.

**Створення елементів керування звітів та корегування їх властивостей в режимі конструктора. Форматування груп елементів керування у звітах**

20. Оснастіть в режимі конструктора заголовки групи  $\text{=[НазваКлієнта]} \& \text{[МісцеРоботиКлієнта]}$  надписами згідно рис. 72. Для цього:
- 20.1. Створіть надпис (33), виконуючи наступні дії:
- 20.1.1. Для створення надпису натисніть кнопку (9);
  - 20.1.2. Натягніть у заголовку групи  $\text{=[НазваКлієнта]} \& \text{[МісцеРоботиКлієнта]}$  прямокутник для розміщення тексту надпису;
  - 20.1.3. Введіть текст надпису. На вкладці *Формат* встановіть розмір шрифту 11 пт. та натисніть кнопку Ж;
  - 20.1.4. Перемістіть текст надпису у відведене місце (див. рис. 72) за допомогою мишки. Для цього:
    - 20.1.4.1. Виділіть створений надпис;
    - 20.1.4.2. Перетягніть його за межу у відведене місце;
  - 20.1.5. Здійсніть переміщення надпису, змінюючи значення властивостей *От левого края* та *От верхнего края* на вкладці *Макет* вікна його властивостей. Встановіть на цій вкладці параметри згідно табл. 22.

Таблица 22

**Значення окремих властивостей надпису (33) звіту**

***ДаніЗамовленьТоварівПоКлієнтах***

Властивість	Значення
От левого края	0,3 см
От верхнего края	1 см
Ширина	1,4 см
Высота	0,5 см

- 20.1.6. Поекспериментуйте з різними значеннями властивостей надпису у вікні властивостей. Дію кожної незрозумілої

властивості проаналізуйте в режимі попереднього перегляду звіту;

20.2. Самостійно створіть надписи (35), (38) та (40).

21. Вирівняйте створені надписи (33), (35), (38) та (40) по верхньому краю. Для цього:

21.1. При натиснутій клавіші *Shift* відмітьте вказані елементи керування, по чергово виділяючи їх натисненням лівої кнопки мишки. Крім того, виділити зазначені елементи можна при натиснутій кнопці (7) за допомогою мишки, відмічаючи область вертикальної лінійки перед надписами або охоплюючи їх умовним прямокутником при натиснутій лівій кнопці маніпулятора;

21.2. На вкладці стрічки меню *Упорядочить* оберіть пункт *Вывровняют – По верхнему краю*;

21.3. Для встановлення точних значень відступів виділених надписів від верхнього краю у вікні властивостей виділених елементів на вкладці *Макет* встановіть значення поля *От верхнего края* рівним 1 см.

22. Встановіть однакову висоту для надписів (33), (35), (38) та (40). З цією метою:

22.1. Самостійно виділіть всі зазначені елементи;

22.2. Встановіть у вікні їх властивостей на вкладці *Макет* значення поля *Высота* рівним 0,5 см або у вкладці *Формат* оберіть пункт *Размер – По самому высокому*.

23. Створіть у звіті лінії згідно рис. 72. Для цього:

23.1. Над відформатованими надписами розмістіть лінію (36), виконуючи наступні дії:

23.1.1. Для створення лінії натисніть кнопку (17);

23.1.2. Натягніть у заголовку групи *=[НазваКлієнта] & [МісцеРоботиКлієнта]* лінію та перемістіть її мишкою у відведене місце;

23.1.3. У вікні властивостей лінії на вкладці *Макет* встановіть значення властивостей відповідно до табл. 23.

Таблиця 23

**Значення окремих властивостей лінії (36) звіту**

***ДаніЗамовленьТоварівПоКлієнтах***

<b>Властивість</b>	<b>Значення</b>
От левого края	0,2 см
От верхнего края	0,9 см
Ширина	16,6 см
Высота	0 см
Тип границы	Сплошная
Ширина границы	3 пункта

23.1.4. Поекспериментуйте з різними значеннями властивостей лінії у вікні властивостей. Дію кожної незрозумілої

властивості проаналізуйте в режимі попереднього перегляду звіту;

- 23.2. Самостійно створіть в макеті звіту лінії (37), (45) та (47) з такою ж (чому?) шириною, висотою, відступом від лівого краю та типом межі. Індивідуальні параметри ліній встановіть згідно табл. 24.

Таблиця 24

**Індивідуальні параметри ліній (37), (45) і (47) звіту  
ДаніЗамовленьТоварівПоклієнтах**

Номер реквізиту	Від верхнього краю	Ширина межі
37	1,7 см	2 пункта
45	0,2 см	2 пункта
47	1,0 см	3 пункта

24. У заголовку звіту вставте довільний малюнок на місці (30), виконуючи наступні дії:

- 24.1. Для вставки довільного малюнка натисніть кнопку (26);
- 24.2. Натягніть у заголовку звіту прямокутник для розміщення малюнка згідно рис. 72;
- 24.3. У вікні *Выбор рисунка*, що з'явиться після окреслення меж малюнка, перейдіть у свою папку, оберіть потрібний малюнок та натисніть кнопку ОК;
- 24.4. Перемістіть малюнок у відведене місце згідно зразка;
- 24.5. Поекспериментуйте з різними значеннями властивостей малюнка у вікні властивостей. Дію кожної незрозумілої властивості проаналізуйте в режимі попереднього перегляду.

25. Створіть у звіті поля для відображення результатів виконання запиту джерела даних. Для цього:

- 25.1. Відобразіть (якщо воно приховане) вікно списку полів джерела даних, натиснувши кнопку (4);

- 25.2. Створіть у звіті поле (38) для відображення загальної замовленої кількості кожного товару окремим клієнтом. З цією метою:

- 25.2.1. У вікні списку полів джерела даних віднайдіть поле *ЗагальнаКількість* та перетягніть його при натиснутій лівій кнопці мишки в область даних на місце реквізиту (38);

- 25.2.2. Якщо створене поле виявилось зміщеним відносно зразка, то виділіть та перемістіть його у відведене місце (див. рис. 72);

- 25.2.3. Збільшіть розміри поля, перетягуючи маркери навколо нього при натиснутій лівій кнопці мишки. Вказівник мишки при цьому повинен мати вигляд двонаправленої стрілки;

- 25.2.4. Поекспериментуйте з різними значеннями властивостей поля у вікні властивостей. Дію кожної незрозумілої



- властивості проаналізуйте в режимі попереднього перегляду;
- 25.3. Самостійно створіть в макеті звіту поля (44), (34) та (39);
- 25.4. Відредагуйте надпис поля (34) *НазваКлієнта*, замінюючи його текстом *Клієнт*:. Для цього:
- 25.4.1. Виділіть надпис розглядуваного поля.
- 25.4.2. Змініть текст надпису, редагуючи його властивість *Подпись* у вікні властивостей, або відмітивши **ще раз** мишкою текст цього надпису (при цьому безпосередньо в тексті надпису з'явиться курсор для редагування його вмісту);
- 25.4.3. Виділіть надпис, змініть його розміри та перемістіть у відведене місце (див. рис. 72). Який початковий текст містив надпис поля (34)? Звідки скопіювалося його значення? Чи впливає надпис на властивості поля *НазваКлієнта* базової таблиці?
- 25.5. Самостійно відкоригуйте надпис поля (39) *МісцеРоботиКлієнта*. Забезпечте також можливість автоматичного збільшення висоти полів (34) та (39) внаслідок перевищення розміру даних поля розмірів, відведених йому в макеті.
26. Створіть обчислювальні поля для форматowanego відображення в звіті значень результатів виконання запиту джерела даних. З цією метою:
- 26.1. Оснастіть заголовок звіту обчислювальним полем (31), виконуючи наступні дії:
- 26.1.1. Натягніть у відведеному місці згідно зразка вільне поле при допомозі кнопки (8) панелі елементів;
- 26.1.2. У вікні властивостей створеного поля на вкладці *Данные* задайте значення параметра *Данные*, вказавши ="*Дані замовлень товарів по клієнтах станом на " & Format\$(Date());"dd.mm.yyyy"*);
- 26.2. Самостійно створіть та відформатуйте інші обчислювальні поля звіту та їх підписи згідно рис. 72 (підпис реквізиту (49) слід знищити). Вирази для обчислень значень полів вкажіть згідно табл. 25.
- 26.3. Самостійно забезпечте можливість автоматичного збільшення висоти полів (32) та (42) внаслідок перевищення розміру даних поля розмірів, відведених йому в макеті.
27. Забезпечте автоматичну нумерацію назв товарів в межах кожного клієнта. Для цього у вікні властивостей реквізиту (41) на вкладці *Данные* встановіть значення поля зі списком *Сумма с накоплением* рівним *Для группы*.

**Значення виразів окремих обчислювальних полів звіту  
ДаніЗамовленьТоварівПоКлієнтах**

Номер реквізиту	Вираз для обчислення значення поля
32	= "Дані замовлень товарів по клієнтах станом на " & Format\$(Date(); "dd.mm.yyyy") & " (продовження)"
41	=1
42	= [НазваТовару] & " " & [МаркаТовару] & ""
48	= Sum([Сума])
49	= "Сторінка " & [Page] & " з " & [Pages]
51	= Sum([Сума])

28. Самостійно, використовуючи групові операції, вирівняйте всі поля області даних по верхньому краю та встановіть для них однакові розміри по вертикалі. Перегляньте створений звіт в режимі попереднього перегляду та переконайтеся в його дієздатності.

**Закріплення вивченого матеріалу**

29. Самостійно створіть звіт *ДаніЗамовленьКлієнтівПоТоварах* згідно рис. 75 для аналізу замовлень кожного товару по клієнтах на основі того ж запиту *ДаніЗамовленьТоварівПоКлієнтах*.

30. Самостійно створіть звіт *СумиЗамовлень* на основі запиту з тією ж назвою для перегляду сум замовлень кожного клієнта по кожному замовленню, передбачивши в ньому два рівні групування: по даних клієнта (з заголовками та примітками) та по даних замовлення (для сортування записів області даних). В області даних звіту забезпечте відображення дат та відповідних сум замовлень окремого клієнта, відсортованих за зростанням дат. Чому замість ініціалів клієнта відображається його код? Самостійно модифікуйте джерело даних звіту та його структуру для відображення не коду, а ініціалів клієнта.

31. Самостійно створіть аналогічний звіт *СумиПостачань* на основі запиту з тією ж назвою.

32. Самостійно створіть звіт *ДаніЗамовленьТоварівЦьогоРоку* на основі запиту з тією ж назвою для перегляду даних замовлень кожного товару окремими клієнтами у цьому році, передбачивши в ньому три рівні групування: по даних товару (з заголовками та примітками), по даних клієнта (з заголовками і примітками) та по датах оформлення замовлень (для сортування записів області даних). В області даних звіту забезпечте відображення дати, кількості, ціни, суми позиції та загальної суми замовлень з початку року окремого товару відповідним клієнтом.



### Дані замовлень клієнтів по товарах станом на 29.03.2003

Назва та марка товару: *Кава 'Галка'*

№ п/п	Ініціали та місце роботи клієнта	Замовлена кількість	Замовлено на суму
1	Зінчук Т.Я. (Бізнес-центр "Партнер")	2	8,00
2	Прохоров К.Н. (Спортивний клуб "Марафон")	2	12,00
<b>Всього замовлено товару по клієнтах:</b>		<b>4</b>	<b>20,00</b>

Назва та марка товару: *Майонез 'Чумак'*

№ п/п	Ініціали та місце роботи клієнта	Замовлена кількість	Замовлено на суму
1	Зінчук Т.Я. (Бізнес-центр "Партнер")	10	12,50
2	Попов Н.Н. (Магазин "Все для саду і городу")	1	1,45
<b>Всього замовлено товару по клієнтах:</b>		<b>11</b>	<b>13,95</b>

Сторінка 1 з 4

### Дані замовлень клієнтів по товарах станом на 29.03.2003 (продовження)

Назва та марка товару: *Шоколад 'Світоч'*

№ п/п	Ініціали та місце роботи клієнта	Замовлена кількість	Замовлено на суму
1	Зінчук Т.Я. (Бізнес-центр "Партнер")	1	2,00
2	Петрів Т.Я. (АТ "Оазис")	1	3,00
3	Прохоров К.Н. (Спортивний клуб "Марафон")	17	51,00
<b>Всього замовлено товару по клієнтах:</b>		<b>19</b>	<b>56,00</b>
<b>Всього замовлено товарів по клієнтах:</b>		<b>55</b>	<b>122,59</b>

Сторінка 6 з 6

...

Рис. 75. Фрагменти звіту *ДаніЗамовленьКлієнтівПоТоварах* в режимі попереднього перегляду

33. Самостійно створіть звіт *ДаніЗамовленьТоварівЦьогоРокуПоДняхТижня* на основі запиту *ДаніЗамовленьТоварівЦьогоРоку* для перегляду сум замовлень кожного товару по днях тижня у цьому році, передбачивши в ньому два рівні групування: по даних товару (з заголовками та примітками) та по днях тижня (лише з примітками для відображення загальних сум замовлень товару по кожному дню тижня). В області даних звіту не розміщуйте жодного поля та встановіть його висоту рівною нулю для уникнення відображення даних окремих пунктів замовлень. Чому суми замовлень по кожному дню тижня потрібно виводити в примітках, а не в заголовку групи?
34. Відкоригуйте структуру створеного раніше за допомогою майстра звіту *АлфавітнийСписокВідділів*, виконуючи наступні дії:
- 34.1. Відкрийте звіт в режимі конструктора. Самостійно встановіть ширину звіту рівною 16,9 см;
- 34.2. У вікні параметрів сторінки звіту встановіть розміри правого та лівого поля рівними 2 см, книжну орієнтацію та розміри сторінки стандарту А4 (чому?), а на вкладці *Столбцы* – параметри для відображення даних у два стовпці відповідно до табл. 26.

Таблиця 26

Значення окремих параметрів вкладки *Столбцы* параметрів сторінки звіту *ДаніЗамовленьТоварівПоКлієнтах*

Параметр	Значення параметру
Число столбцов	2
Интервал столбцов	0,3 см
Ширина столбца	8,3 см (чому?)
Макет столбца	слева направо

- 34.3. Збережіть встановлені параметри та перегляньте звіт в режимі попереднього перегляду. Що змінилося у зовнішньому вигляді звіту?
- 34.4. Знову перейдіть в режим конструктора та встановіть у вікні *Параметры страницы* на вкладці *Столбцы* значення параметри *Макет столбца* рівним *сверху вниз*. Збережіть встановлені параметри та перегляньте звіт в режимі попереднього перегляду. Обгрунтуйте структуру зовнішнього вигляду отриманого звіту. Самостійно встановіть попереднє значення макету стовпця та збережіть встановлені параметри.
35. Самостійно модифікуйте створені раніше звіти в режимі конструктора, забезпечивши:
- 35.1. Коректне відображення надписів звіту;
- 35.2. Розміщення всіх полів звіту по ширині в межах однієї сторінки;
- 35.3. Відображення на першій сторінці дати отримання звіту, а на всіх сторінках – їх номерів;

- 35.4. Графічне виділення заголовків та приміток груп. Відображення для кожної групи підсумків в області приміток;
- 35.5. Автоматичну нумерації записів кожної групи;
- 35.6. Коректне відображення загальних підсумків звіту.

### **Завершальний етап заняття**

- 36.Стисніть створену БД.
- 37.Завершіть роботу Access.
- 38.Віднайдіть створену БД та заархівуйте її вміст.
- 39.Збережіть створений архів на власному носії в двох екземплярах.
- 40.Відкоригуйте у власній БД звіти, створені за допомогою майстра, в режимі конструктора.

### **Контрольні запитання**

1. Як двома способами розпочати створення нового звіту?
2. Де задаються параметри сторінки для відображення звіту в режимі попереднього перегляду?
3. Як наближено вказати ширину звіту та висоту кожного розділу? Як встановити точні значення цих параметрів?
4. Як забезпечити можливість автоматичного збільшення висоти розділу звіту при збільшенні висоти одного з його полів внаслідок перевищення розміру даних поля розмірів, введених йому в макеті?
5. Як встановити чи змінити джерело даних звіту? Де відображаються назви полів джерела даних?
6. Де формуються рівні групування та ознаки сортування даних звіту? Які додаткові параметри можна встановити для кожного рівня групування та для чого вони використовуються?
7. Яка різниця між іменем та джерелом даних елемента керування?
8. Які види форматування груп елементів керування використовуються в режимі конструктора?
9. Якими трьома способами можна виділити групу елементів керування? Для чого використовується таке виділення?
10. Навіщо і як створюються обчислювальні поля у звіті?
11. Як створити у звіті поле для автоматичної нумерації записів групи чи звіту? Як відобразити накопичувальні підсумки у звіті?
12. Коли у звіті відображаються коди об'єктів замість відповідних характеристик? Як в таких випадках відобразити саме характеристики об'єктів?
13. Де встановити параметри багатостовпцевого відображення звіту? Які параметри і для чого при цьому використовуються?

## Лабораторна робота № 15

**Тема.** Комплексне використання форм, запитів та звітів для автоматизації обробки даних інформаційних систем

**Мета.** Формування вмінь та навичок комплексного використання форм, запитів та звітів. Закріплення вмінь і навичок створення вільних форм, корегування параметрів запитів, створення та корегування структури звітів. Вдосконалення вмінь та навичок створення та використання вільних елементів керування у формах.

### Теоретичні відомості

Під час аналізу у звітах показників діяльності в предметній області користувача, як правило, не цікавить вся інформація, що міститься в БД. Найчастіше йому потрібна лише частина даних, які відповідають визначеним критеріям відбору (наприклад, замовлення і постачання найчастіше аналізують за декілька попередніх місяців, а не за весь час функціонування ІС). Зрозуміло, що ці критерії можна задавати як параметри запитів-джерел відповідних звітів, але якщо відбувається аналіз різних даних з одними і тими самими параметрами, то вказувати їх декілька разів недоцільно. Крім цього, при введенні значення наступного параметра у запитах не відображаються вже введені значення попередніх параметрів і тому узгодити їх між собою неможливо. Ось чому типові параметри відбору даних для формування кожної групи звітів доцільно вводити в окремій вільній формі і викликати ці звіти з даної форми за допомогою кнопок. Наприклад, для отримання звітів за період доцільно розробити форму для введення і узгодження між собою початкової і кінцевої дат (рис. 76), забезпечити зчитування запитами-джерелами цих звітів діапазону відбору дат з розробленої форми, після чого створити у ній кнопки з відповідними назвами для завантаження даних звітів.

**Механізм дії** типового комплексу *Форма – Запит – Звіт* (див. відповідні виноски на рис. 76):

1. Користувач вводить параметри у форму і натискає кнопку для отримання відповідного звіту;
2. Перед початком формування активований звіт викликає запит-джерело даних;
3. Запит-джерело звіту звертається до форми за значеннями параметрів;
4. Значення параметрів зчитуються з форми і надходять у запит;
5. Запит-джерело виконується згідно параметрів, зчитаних з форми, і динамічна таблиця з результатами його виконання повертається у звіт;



6. СУБД формує звіт згідно його макету, використовуючи динамічну таблицю виконаного запиту-джерела. Сформований звіт відображається для користувача.

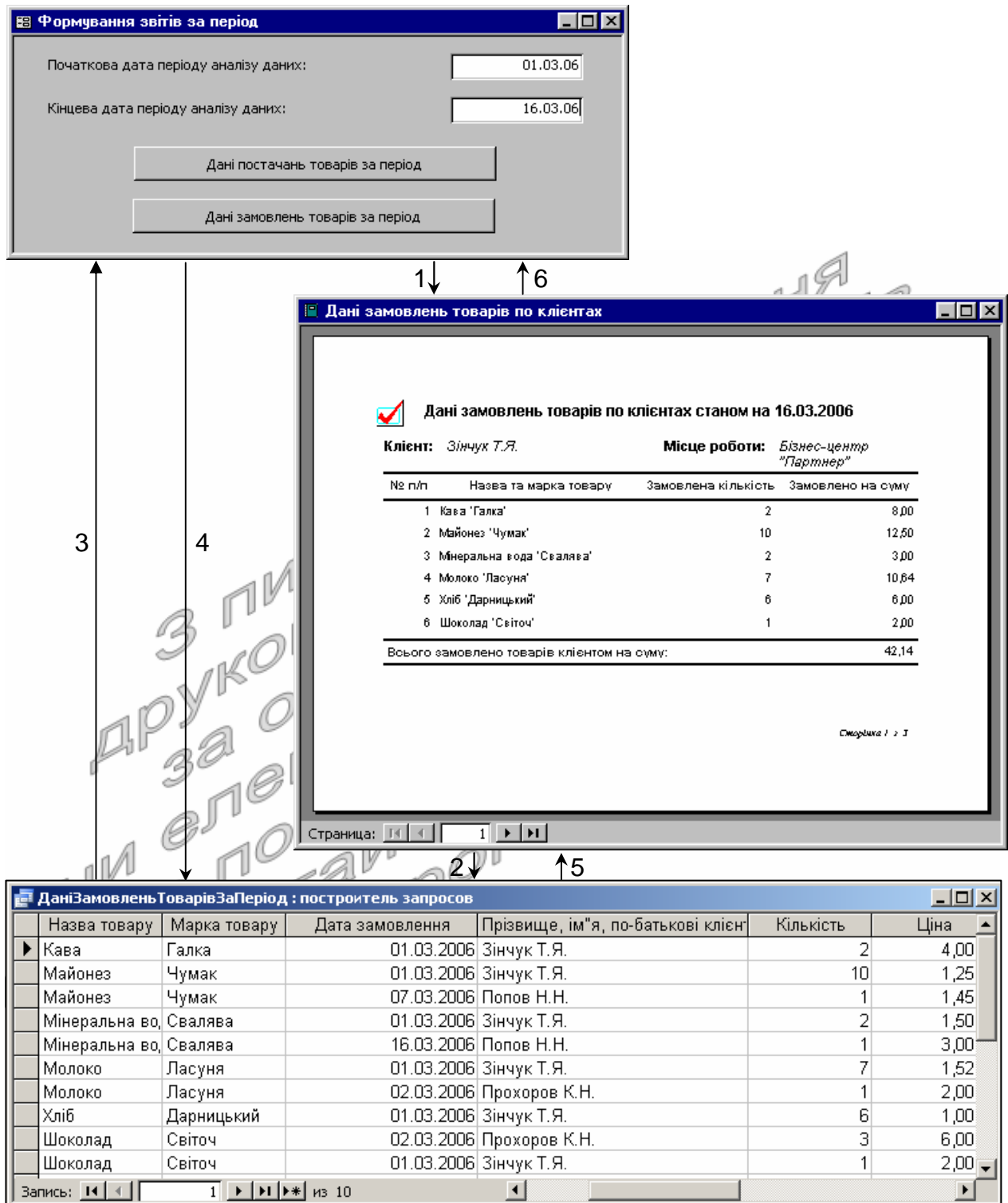


Рис. 76. Послідовність взаємодії об'єктів комплексу *Форма – Запит – Звіт*:

- 1 – виклик звіту з форми;
- 2 – виклик запиту для формування джерела даних звіту;
- 3 – звертання запиту для зчитування параметрів з форми;
- 4 – зчитування параметрів з форми і підстановка їх у запит-джерело;
- 5 – виконання запиту-джерела даних та повернення його результатів у звіт;
- 6 – формування та відображення звіту згідно параметрів форми.

Для забезпечення відбору запитом-джерелом даних згідно параметрів, введених у формі, необхідно у ньому використовувати параметри у форматі *Forms!<Ім'я форми>!<Ім'я елемента керування>*. Наприклад, для відбору даних за вказаний період у запиті комплексу рис. 76 по полю *ДатаЗамовлення* необхідно вказати умову відбору: *Between Forms!ФормуванняЗвітівЗаПеріод!ПочатковаДата AND Forms!ФормуванняЗвітівЗаПеріод!КінцеваДата*. Наголосимо також, що при модифікації запиту-джерела з комплексу *Форма – Запит – Звіт* в режимі конструктора потрібно не лише відкоригувати параметри відбору у бланку, а й видалити ці параметри з вікна *Параметри...*, а під час перегляду результатів виконання такого запиту в режимі таблиці потрібно, щоб форма з параметрами відбору була відкрита.

Якщо комплекс *Форма – Запит – Звіт* створюється для формування звітів по окремому об'єкту (наприклад, по відділу, співробітнику чи товару), то у формі з параметрами відбору для вибору такого об'єкта доцільно створювати майстром підстановок список чи поле зі списком, в якому зберігати унікальний код обраного об'єкта (для прискорення відбору запитом-джерелом), а відображати його назву (для користувача). Ім'я такого елемента керування має відповідати полю для зберігання значень (коду об'єкта), щоб полегшити процес розробки запиту-джерела для звіту. Наприклад, для аналізу даних окремого співробітника доцільно розробити форму *ЗвітиПоСпівробітнику*, в якій створити поле зі списком *КодСпівробітника* для підстановки значення з таблиці *Співробітники*. Джерело записів цього поля зі списком має містити принаймні два поля з цієї таблиці: *КодСпівробітника* – для зберігання та *ПІБ* – для відображення. Тоді запити-джерела, які відбирають дані окремих співробітників, необхідно модифікувати, вказати по полю *КодСпівробітника* умову відбору *Forms!ЗвітиПоСпівробітнику!-КодСпівробітника*, після чого розробити для цих запитів звіти і створити у формі *ЗвітиПоСпівробітнику* кнопки для виклику даних звітів.

На завершення зазначимо, що для елементів керування параметрів відбору кожної вільної форми комплексів *Форма – Запит – Звіт* потрібно задавати значення по замовчуванню та перевіряти їх коректність і взаємоузгодження перед викликом звітів, а самі такі форми доцільно викликати з вкладки *Звіти* головної кнопкової форми чи стрічки меню ІС.

Література: [2, С. 510-514; 1, С. 269-282]

### **Підготовчий етап заняття. Актуалізація знань**

1. Завантажте Access, відкрийте розроблену раніше БД *Sklad*.
2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип объекта*.
3. Перейдіть в області переходів до розділу *Формы*.

## Створення вільних форм для введення параметрів формування звітів


4. Створіть вільну форму *ФормуванняЗвітівЗаПеріод* для введення початкової та кінцевої дат відбору даних при формуванні звітів за період (рис. 77). Для цього:

Рис. 77. Вигляд форми *ФормуванняЗвітівЗаПеріод* в режимі форми

- 4.1. Розпочніть створення форми в режимі конструктора;  
 4.2. У вікні властивостей форми на вкладці *Макет* відкоригуйте значення окремих властивостей згідно табл. 27. Обґрунтуйте встановлені значення параметрів форми;

Таблиця 27  
**Значення окремих властивостей зовнішнього вигляду форми *ФормуванняЗвітівЗаПеріод***

Властивість	Значення
Подпись	Формування звітів за період
Полосы прокрутки	Отсутствуют
Область выделения	Нет
Кнопки навигации	Нет
Разделительные линии	Нет
Выравнивание по центру	Да
Ширина	12 см

- 4.3. Користуючись кнопкою  вкладки *Конструктор* стрічки меню, створіть у формі вільне поле для введення початкової дати аналізу даних у звітах за період. Чому у цьому полі відображається текст *Свободный*?

Таблиця 28  
**Значення окремих властивостей зовнішнього вигляду першого поля форми *ФормуванняЗвітівЗаПеріод***

Вкладка	Властивість	Значення
Макет	Формат поля	Краткий формат даты
	Выравнивание текста	По правому краю
Данные	Значение по умолчанию	Date()-Day(Date())+1
	Условие на значение	Is Not Null
	Сообщение об ошибке	Введіть початкову дату періоду аналізу даних
Другие	Имя	ПочатковаДата

Встановіть у вікні властивостей цього поля значення окремих властивостей у відповідності до табл. 28. Обґрунтуйте встановлені значення параметрів поля. Що забезпечує встановлення такого формату поля та значення по замовчуванню?

- 4.4. Відкоригуйте надпис біля поля, ввівши у нього текст *Початкова дата періоду аналізу даних*;
- 4.5. Самостійно створіть під полем *ПочатковаДата* вільне поле *КінцеваДата* для введення кінцевої дати аналізу даних у звітах за період. Встановіть у вікні властивостей поля значення параметрів згідно табл. 29. Обґрунтуйте встановлені значення властивостей поля;

Таблиця 29

**Значення окремих властивостей зовнішнього вигляду другого поля форми  
ФормуванняЗвітівЗаПеріод**

Вкладка	Властивість	Значення
Макет	Формат поля	Краткий формат дати
	Выравнивание текста	По правому краю
Данные	Значение по умолчанию	Date()
	Условие на значение	Is Not Null
	Сообщение об ошибке	Введіть кінцеву дату періоду аналізу даних
Другие	Имя	КінцеваДата

- 4.6. Відкоригуйте надпис біля поля *КінцеваДата*, ввівши у нього текст *Кінцева дата періоду аналізу даних*;
- 4.7. Самостійно розмістіть створені поля та їхні надписи згідно рис. 77.
- 4.8. Збережіть створену форму.

**Корегування параметрів запитів для відбору даних згідно значень, введених у форми**

5. Відкоригуйте структуру запиту *ДаніПостачаньТоварівЗаПеріод* так, щоб параметри відбору він не вимагав ввести з клавіатури, а зчитував з форми *ФормуванняЗвітівЗаПеріод*. Для цього:

- 5.1. Відкрийте даний запит в режимі конструктора;
- 5.2. Перейдіть у вікно *Параметри запроса*, що відкривається кнопкою *Параметри* зі стрічки меню або аналогічним пунктом контекстного меню вільного поля області джерела даних, знищіть в ньому всі введені назви параметрів у стовпці *Параметр* та збережіть зміни, внесені у поля вікна параметрів;
- 5.3. Змініть умову відбору поля *ДатаПостачання*, ввівши в рядку *Условие отбора* замість попередньої умови текст *Between Forms![ФормуванняЗвітівЗаПеріод]![ПочатковаДата] And Forms![ФормуванняЗвітівЗаПеріод]![КінцеваДата]*;

5.4. Збережіть внесені зміни в структуру запиту. При перевірці дієздатності запиту подбайте, щоб форма *ФормуванняЗвітівЗаПеріод* була відкрита в режимі форми.

#### **Створення звітів на основі відкоригованих запитів**

6. Самостійно створіть за допомогою майстра звіт *ДаніПостачаньТоварівЗаПеріод* на основі полів запиту *ДаніПостачаньТоварівЗаПеріод*, сформувавши у ньому рівень групування за даними товару. Відкоригуйте макет створеного звіту в режимі конструктора, забезпечивши:
  - 6.1. Коректне відображення надписів звіту;
  - 6.2. Розміщення всіх полів звіту по ширині в межах однієї сторінки;
  - 6.3. Відображення на першій сторінці дати отримання звіту, а на всіх сторінках – їх номерів;
  - 6.4. Графічне виділення заголовків та приміток групи. Відображення в області приміток групи загальних кількостей та сум по кожному товару;
  - 6.5. Автоматичну нумерації записів кожної групи;
  - 6.6. Коректне відображення загальних підсумків звіту.

#### **Створення засобів завантаження звітів з форми для введення параметрів їх формування**

7. Самостійно створіть у формі *ФормуванняЗвітівЗаПеріод* кнопку для виклику звіту *ДаніПостачаньТоварівЗаПеріод* (див. рис. 77). Чому при завантаженні звіту аналізуються дані саме за вказаний період? Який механізм взаємодії розробленого комплексу об'єктів?
8. Самостійно, згідно варіанту, змініть структуру трьох запитів, що аналізують дані за період так, щоб параметри відбору вони зчитували з форми *ФормуванняЗвітівЗаПеріод*. Номер варіанту рівний остачі від ділення на шістнадцять Вашого порядкового номера в журналі групи, збільшеній на два. Назви запитів для корегування оберіть з попереднього, однакового та наступного підпункту відносно номера варіанту:
  - 8.1. *ДаніЗамовленьТоварівЗаПеріод*;
  - 8.2. *ВлаштованіСпівробітникиЗаПеріод*;
  - 8.3. *ДаніЗамовленьТоварівЗаПеріод*;
  - 8.4. *ДаніПостачаньТоварівЗаПеріод*;
  - 8.5. *ЗвільненіСпівробітникиЗаПеріод*;
  - 8.6. *КількостіЗамовленьПоКлієнтахЗаПеріод*;
  - 8.7. *КількостіОформленихЗамовленьПоСпівробітникахЗаПеріод*;
  - 8.8. *КількостіОформленихПостачаньПоСпівробітникахЗаПеріод*;
  - 8.9. *КількостіПостачаньПоПостачальникахЗаПеріод*;
  - 8.10. *КлієнтиБезЗамовленьЗаПеріод*;
  - 8.11. *ПоколінняСпівробітників*;
  - 8.12. *ПостачальникиБезПостачаньЗаПеріод*;
  - 8.13. *СумиЗамовленьПоКлієнтахЗаПеріод*;

- 8.14. Суми Оформлених Заомвлень По Співробітниках За Період;
  - 8.15. Суми Оформлених Постачань По Співробітниках За Період;
  - 8.16. Суми Постачань По Постачальниках За Період;
  - 8.17. Товари Без Постачань За Період;
  - 8.18. Товари Без Заомвлень За Період.
9. Створіть звіти на основі відкоригованих запитів. У формі *Формування Звітів За Період* створіть кнопки для завантаження нових звітів.
10. Створіть в головній кнопковій формі на вкладці *Звіти* кнопку для завантаження форми *Формування Звітів За Період*.

### Закріплення вивченого матеріалу

11. Створіть вільну форму *Формування Звітів По Вказаному Відділу* для вибору відділу та завантаження звітів по аналізу пов'язаних з ним даних (рис. 78). Для цього:

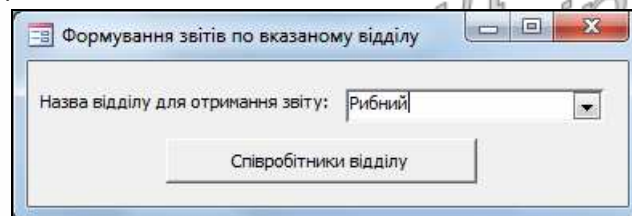



Рис. 78. Вигляд форми *Формування Звітів По Вказаному Відділу* в режимі форми

- 11.1. Розпочніть створення форми в режимі конструктора;
- 11.2. У вікні властивостей форми на вкладці *Макет* відкоригуйте значення окремих властивостей згідно табл. 30. Обґрунтуйте встановлені значення параметрів форми;

Таблиця 30

#### Значення окремих властивостей зовнішнього вигляду форми *Формування Звітів По Вказаному Відділу*

Властивість	Значення
Подпись	Формування звітів по вказаному відділу
Полосы прокрутки	Отсутствуют
Область выделения	Нет
Кнопки навигации	Нет
Разделительные линии	Нет
Выравнивание по центру	Да
Ширина	12 см

- 11.3. Створіть у формі поле зі списком для вибору потрібного відділу для отримання звітів. Для цього:
  - 11.3.1. Перегляньте таблицю *Відділи* та запам'ятайте код того відділу, дані якого плануєте аналізувати найчастіше;
  - 11.3.2. Поверніться в режим конструктора форми та переконайтеся, що кнопка виклику майстрів створення елементів керування (Яка?) натиснута.
  - 11.3.3. Користуючись кнопкою  групи елементів керування, натягніть у формі прямокутник для відображення поля зі



- списком та розпочніть його створення за допомогою майстра;
- 11.3.4. На першому кроці майстра забезпечте можливість підстановки значення поля зі списком з таблиці;
  - 11.3.5. На другому кроці майстра оберіть таблицю *Відділи* для формування джерела даних поля зі списком;
  - 11.3.6. На третьому кроці майстра перенесіть зі списку доступних в список обраних наступні поля: *КодВідділу* – для зберігання його значення в полі зі списком, *НазваВідділу* – для відображення його тексту у полі зі списком, *МісцеРозташування* – для однозначної ідентифікації відділу при виборі значення поля зі списком;
  - 11.3.7. На четвертому кроці забезпечте сортування даних списку за назвою відділу;
  - 11.3.8. На п'ятому кроці майстра переконайтеся, що значення ключового стовпця приховано та самостійно завершіть формування поля зі списком;
  - 11.3.9. Перейдіть на вкладку *Данные* вікна властивостей створеного поля зі списком та вкажіть у її властивості *Значение по умолчанию* код того відділу, який ви запам'ятали раніше. На вкладці *Другие* цього ж вікна встановіть для створеного поля зі списком ім'я *КодВідділу*. Чому обрано саме це ім'я? Забезпечте виконання умови непорожності цього поля та вивід попереджувального повідомлення при її порушенні;
  - 11.4. Відкоригуйте надпис біля поля, ввівши у нього текст *Назва відділу для отримання звіту*:. Самостійно розмістіть поле зі списком та його надпис згідно рис. 78;
  - 11.5. Збережіть створену форму.
12. Відкоригуйте структуру запити *СпівробітникиВідділу* так, щоб параметри відбору він не вимагав ввести з клавіатури, а зчитував з форми *ФормуванняЗвітівПоВказаномуВідділу*. Для цього:
- 12.1. Відкрийте даний запит в режимі конструктора;
  - 12.2. Самостійно знищіть всі встановлені умови та параметри відбору;
  - 12.3. Доповніть бланк запити поле *КодВідділу* та вкажіть для нього умову відбору *[Forms]![ФормуванняЗвітівПоВказаномуВідділу]![КодВідділу]*;
  - 12.4. Збережіть внесені зміни в структуру запити. При перевірці дієздатності запити подбайте, щоб форма *ФормуванняЗвітівПоВказаномуВідділу* була відкрита в режимі форми.
13. Самостійно створіть звіт *СпівробітникиВідділу* в режимі конструктора на основі одноіменного запити, забезпечивши при цьому коректність відображення даних.

14. Самостійно створіть у формі *ФормуванняЗвітівПоВказаномуВідділу* кнопку для завантаження звіту *СпівробітникиВідділу* (див. рис. 78) та переконайтеся у дієздатності створеного комплексу.
15. Створіть в головній кнопковій формі на вкладці *Звіти* кнопку для завантаження форми *ФормуванняЗвітівПоВказаномуВідділу*.

### **Завершальний етап заняття. Повторення вивченого матеріалу**

16. Самостійно створіть комплекс для коректного відображення результатів дії запитів *ПостачанняВказаногоТовару* та *ЗамовленняВказаногоТовару*.
17. Самостійно створіть комплекс *СпівробітникиВказаної Національності* (у формі параметрів цього комплексу відповідне поле зі списком створюється за допомогою підстановки з фіксованого набору значень, як в таблиці *Співробітники*).
18. Забезпечте можливість виклику з головної кнопкової форми всіх створених форм комплексів для введення параметрів формування звітів.
19. Стисніть створену БД.
20. Завершіть роботу Access.
21. Віднайдіть створену БД та заархівуйте її вміст.
22. Збережіть створений архів на власному носії в двох екземплярах.
23. Доповніть власну БД комплексами для автоматизації обробки даних.

### **Контрольні запитання**

1. В якій послідовності створюються комплекси для сумісного використання форм, запитів та звітів?
2. Як взаємодіють між собою форма, запит та звіт при їх сумісному використанні?
3. Чи можуть різні звіти зчитувати параметри відбору з однієї форми?
4. Чи може один звіт зчитувати параметри відбору з багатьох форм?
5. Навіщо для елементів керування форм, що впливають на створення звітів, вказувати значення по замовчуванню? Як слід обирати ці значення? Як забезпечити непорожність значень цих елементів керування?

## Лабораторна робота № 16

**Тема.** Публікація статичної інформації з БД в локальних та глобальних мережах.

**Мета.** Формування вмінь та навичок створення Web-сторінок та pdf-документів засобами СУБД MS Access. Закріплення вмінь і навичок використання запитів для аналізу інформації БД. Усвідомлення переваг, недоліків та можливостей використання і вдосконалення статичних Web-сторінок. Вдосконалення вмінь створення та коригування об'єктів додатків СУБД MS Access.

### Теоретичні відомості

Для публікації зафіксованої поточної інформації, відображеної в таблицях, запитах, формах та звітах БД Access, в локальних та глобальних мережах найчастіше використовуються статичні HTML-сторінки та PDF-документи. Створювати такі зовнішні файли в Access для виділеного чи завантаженого об'єкта можливо двома основними способами: за допомогою відповідних кнопок стрічки меню з групи *Внешние данные – Экспорт* та з використання аналогічних підпунктів пункту контекстного меню *Экспорт* цього об'єкта в області переходів (рис. 79).

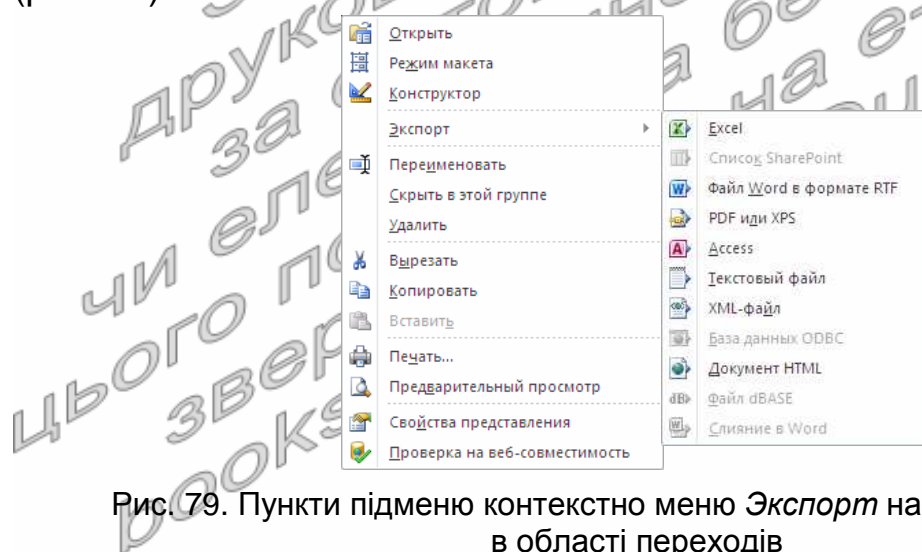


Рис. 79. Пункти підменю контекстно меню *Экспорт* назви звіту Access в області переходів

Після введення назви та вибору типу файла для експорту *Документ HTML* Access надає можливість додатково обрати шаблон оформлення сторінки, а для типу *PDF или XPS* – ще й вказати діапазон сторінок для публікування.

Типові послідовності кроків для публікування статичної інформації БД доцільно зберігати як окремі операції експорту, після чого викликати їх на виконання за допомогою кнопки *Сохраненные операции экспорта* групи стрічки меню *Внешние данные – Экспорт*.

Література: [2, С. 654-656; 1, С. 33-34]

### Підготовчий етап заняття. Актуалізація знань

1. Завантажте Access, відкрийте у власній папці розроблену раніше БД *Sklad*.
2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип об'єкта*.
3. Перейдіть в області переходів до розділу *Запроси*.

### Створення статичних Web-сторінок засобами СУБД MS Access

4. Створіть статичну Web-сторінку *ДніНародженняСпівробітників* для публікації в мережі даних про дні народження співробітників. Для цього:
  - 4.1. Віднайдіть та відмітьте в списку назв запитів запит *ДніНародженняСпівробітників*;
  - 4.2. Експортуйте результати дії відміченого запиту у Web-сторінку, виконуючи наступні дії:
    - 4.2.1. На вкладці стрічки меню *Внешние данные* у групі *Экспорт* розгорніть список кнопки *Дополнительно* та оберіть у ньому пункт *Документ HTML*;
    - 4.2.2. У виведеному вікні *Экспорт – Документ HTML*, використовуючи кнопку *Обзор*, перейдіть у папку власної БД, у полі *Имя файла* вкажіть назву сторінки – *ДніНародженняСпівробітників* та переконайтеся, що обрано тип файла *Документы HTML (\*.html \*.htm)*. Збережіть параметри для створення Web-сторінки;
    - 4.2.3. Не вказуючи інших параметрів, розпочніть процес експорту даних, натиснувши кнопку *OK*.
5. Завантажте та перегляньте отриману Web-сторінку з програми *Проводник* чи *Мой компьютер*. Чи можливо змінити вміст цієї сторінки? За допомогою якої програми виконується перегляд вмісту Web-сторінки?
6. Змініть в БД інформацію про день народження довільного співробітника. Чи відобразилися ці зміни у створеній Web-сторінці? Чому?
7. Аналогічно створіть з цього ж запиту статичну Web-сторінку *ДніНародженняСпівробітників1*, додатково встановивши у вікні експорту прапорець *Экспортировать данные с макетом и форматированием*. При цьому у вікні *Параметры вывода в формате HTML*, що з'явиться після початку експорту, не встановлюйте прапорці і не змінюйте значення перемикачів, а лише продовжіть експорт об'єкта.
8. Порівняйте вигляд отриманої Web-сторінки з попередньою. Що спільного і у чому різниця між вмістом цих Web-сторінок? Чим це пояснити?
9. Самостійно створіть статичну Web-сторінку *ДніНародженняСпівробітників2* з цього ж запиту з використанням іншого шаблону (довільний інший html-файл) та порівняйте її вигляд з попередніми.

10. Самостійно створіть з використання стандартного форматування різні статичні Web-сторінки з таблиці *Співробітники*, форми *Співробітники* та звіту *АлфавітнийСписокСпівробітників*. Порівняйте між собою вміст отриманих Web-сторінок. Як формується структура та дані кожної сторінки?
11. Самостійно створіть з використання стандартного форматування статичні Web-сторінки з форми *ЗаголовкиЗамовлень* та звіту *ДаніЗамовленьТоварівПоКлієнтах*. Яка інформація відображається у цих сторінках?

### **Створення pdf-документів засобами СУБД MS Access**

12. Створіть pdf-документ *ДніНародженняСпівробітників* для публікації в мережі даних про дні народження співробітників у цьому форматі. Для цього:
  - 12.1. Віднайдіть та відмітьте в області переходів запит *ДніНародженняСпівробітників*;
  - 12.2. Екпоруйте результати дії відміченого запиту у pdf-документ, виконуючи наступні дії:
    - 12.2.1. На вкладці стрічки меню *Внешние данные* у групі *Экспорт* натисніть кнопку *PDF или XPS*;
    - 12.2.2. У виведеному вікні *Опубликовать как PDF или XPS* перейдіть у папку власної БД, у полі *Имя файла* вкажіть назву документа – *ДніНародженняСпівробітників*. Опублікуйте результати виконання виділеного запиту;
    - 12.2.3. На останньому кроці майстра самостійно збережіть кроки експорту даних під назвою *ДніНародженняСпівробітників PDF*.
13. Використовуючи форму *Співробітника*, внесіть у БД дані ще одного співробітника.
14. Повторно створіть pdf-документ *ДніНародженняСпівробітників*, використовуючи збережену операцію експорту. Для цього на вкладці стрічки меню *Внешние данные* у групі *Экспорт* натисніть кнопку *Сохраненные операции экспорта*, оберіть назву збереженої послідовності дій *ДніНародженняСпівробітниківPDF* та виконайте збережені команди.
15. Самостійно створіть різні pdf-документи мінімального розміру з таблиці *Співробітники*, форми *Співробітники* та звіту *АлфавітнийСписокСпівробітників*. Порівняйте між собою вміст отриманих документів. Як формується їх структура та дані?
16. Самостійно створіть окремі pdf-документи стандартного розміру з форми *ЗаголовкиЗамовлень* та звіту *ДаніЗамовленьТоварів ПоКлієнтах*. Яка інформація відображається у цих документах?

### **Завершальний етап заняття. Повторення вивченого матеріалу**

17. Стисніть створену БД.
18. Завершіть роботу Access.

19. Віднайдіть створену БД та заархівуйте її вміст.
20. Збережіть створений архів на власному носії в двох екземплярах.
21. Самостійно створіть статичні html-сторінки та pdf-документи з даних власної БД.

### **Контрольні запитання.**

1. Які створити статичні Web-сторінки для публікації інформації з БД в локальних та глобальних мережах за допомогою Access?
2. Для чого при створенні Web-сторінок використовуються шаблони?
3. Які об'єкти можна експортувати в статичні Web-сторінки?
4. Як створити pdf-документ з даних об'єктів Access?
5. З яких типів об'єктів Access можна створити pdf-документи?
6. Що відображається для форм і звітів у web-сторінках і pdf-документах? Чому?
7. Як прискорити виконання типових операцій імпорту та експорту?

### **Програмні питання поточного та підсумкового контролю знань**

1. Бази даних, банки даних, інформаційні системи.
2. Класифікація та засоби автоматизованого банку даних.
3. Моделі зв'язків між даними.
4. Рівні подання даних. Етапи проектування БД.
5. Проектування БД на зовнішньому рівні.
6. Проектування БД на інфологічному рівні.
7. Проектування БД на даталогічному рівні.
8. Проектування БД на внутрішньому рівні.
9. Суть реляційного підходу до проектування БД.
10. Відношення та типи зв'язків між таблицями БД.
11. Нормалізація відношень БД.
12. Забезпечення цілісності посилань в БД.
13. Конвертування та актуалізація БД.
14. Відновлення, реорганізація та реструктуризація БД.
15. Створення та ведення розподілених БД.
16. Об'єкти БД MS Access. Режими функціонування об'єктів БД MS Access.
17. Таблиці БД MS Access та властивості їх полів.
18. Створення таблиць в БД MS Access. Індексування полів таблиць.
19. Забезпечення цілісності посилань в таблицях БД MS Access.
20. Експортування та імпортування даних таблиць БД MS Access. Створення зв'язків з таблицями інших БД.
21. Зберігання інформації про структурні зв'язки між таблицями в БД MS Access.
22. Призначення, типи та варіанти використання форм в БД MS Access.
23. Створення та корегування параметрів форм в БД MS Access.
24. Види та властивості елементів керування форм в БД MS Access.



25. Призначення та різновиди запитів.
26. Конструювання запитів на вибірку в БД MS Access.
27. Групові операції в запитах БД MS Access.
28. Роль та задання параметрів зв'язків між таблицями при конструюванні запитів БД MS Access.
29. Основні оператори SQL.
30. Конструювання запитів на корегування в БД MS Access.
31. Вирази БД MS Access.
32. Функції БД MS Access.
33. Сортування в таблицях, запитах та формах БД MS Access.
34. Пошук в таблицях, запитах та формах БД MS Access.
35. Фільтрування в таблицях, запитах та формах БД MS Access.
36. Призначення, типи та структура звітів в БД MS Access.
37. Створення та корегування параметрів звітів БД MS Access.
38. Види та властивості елементів керування звітів в БД MS Access.
39. Рівні групування даних в звітах. Розміщення на сторінці та експорт звітів в БД MS Access.
40. Комплексне використання форм, запитів та звітів в БД MS Access.
41. Призначення та створення діаграм в БД MS Access.
42. Створення та використання макросів в БД MS Access.
43. Створення та використання процедур обробки подій в БД MS Access.
44. Документування БД MS Access.
45. Автоматизація додатків БД MS Access.
46. Оптимізація об'єктів БД MS Access.
47. Адміністрування БД MS Access.

### Рекомендована література

1. Бекаревич Ю. Б. Самоучитель Microsoft Access 2013 / Ю. Б. Бекаревич, Н. В. Пушкина. – СПб: BHV-СПб, 2014. – 464 с.
2. Вейскас Дж. Эффективная работа с Microsoft Access 2000 / Дж. Вейскас. – СПб.: Издательство "Питер", 2000. – 1040 с.
3. Дженнингс Р. Использование Microsoft Access 2002. Специальное издание / Р. Дженнингс. – М.: Издательский дом "Вильямс", 2002. – 1012 с.
4. Єрьоміна Н. В. Проектування баз даних / Н. В. Єрьоміна. – К.: КНЕУ, 1998. – 204 с.
5. Інформатика: Комп'ютерна техніка. Комп'ютерні технології: Підручник для студентів вищих навчальних закладів / За ред. О. І. Пушкаря. – К.: Академія, 2002. – 704 с.
6. Кошелєв В. Е. Access 2007 / В. Е. Кошелєв. – М: ООО "Бином-Пресс", 2008. – 592 с. Режим доступу: <http://depositfiles.com/files/wslgqf4s3>.
7. Пасічник В. В. Сховища даних : Навч. посібн. / В. В. Пасічник, Н. Б. Шаховська. - Львів: Магнолія-2006, 2008. – 492 с.
8. Послед. Б. С. Access 2002. Приложения баз даних. Лекции и упражнения / Б. С. Послед. – СПб.: ООО "ДиаСофтЮП", 2002. – 656 с.

Навчальне видання

ШПОРТЬКО Олександр Володимирович

ШПОРТЬКО Леся Василівна

# Розробка баз даних в СУБД Microsoft Access 2010/2013.

Практикум  
для студентів вищих та  
учнів професійно-технічних  
навчальних закладів

Комп'ютерна верстка О. В. Шпортка

Підписано до друку 30 вересня 2017 р.  
Формат 60x84 1/16. Папір офсетний.  
Умовн. друк. арк. 10,7. Обл. вид. арк. 10,2.  
Наклад 300 примірників.

Видавець О. Зень  
Свідоцтво суб'єкта видавничої справи  
серії РВ № 26 від 6 квітня 2004 р.  
вул. Кн. Романа, 9/24, м. Рівне, 33022;  
тел. 0362-24-45-09, 068-025-067-4;  
olegzen@ukr.net

Віддруковано засобами оперативної поліграфії  
цифрової друкарні  
"VPM-Поліграф"  
вул. Буковинська, 3, м. Рівне, 33027;  
тел. 0362-64-21-34.