

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

## **ІНФОРМАТИКА І КОМП'ЮТЕРНА ТЕХНІКА**

**Методичні рекомендації  
до лабораторних робіт  
для студентів спеціальності  
186 «Видавництво та поліграфія»  
першого (бакалаврського) рівня**

**Частина 2**

**Харків  
ХНЕУ ім. С. Кузнеця  
2018**

УДК 004.3(07)

I-74

**Укладач** О. Б. Бережна

Затверджено на засіданні кафедри комп'ютерних систем і технологій.  
Протокол № 10 від 10.04.2018 р.

*Самостійне електронне текстове мережеве видання*

**Інформатика** і комп'ютерна техніка : методичні рекомендації до лабораторних робіт для студентів спеціальності 186 «Видавництво та поліграфія» першого (бакалаврського) рівня. Частина 2 [Електронний ресурс] / уклад. О. Б. Бережна. – Харків : ХНЕУ ім. С. Кузнеця, 2018. – 59 с.

Подано методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни. Для змістового модуля «Оброблення та зберігання інформації за допомогою СУБД Microsoft Access» надано основний навчальний матеріал, наведено порядок виконання лабораторних робіт, запропоновано контрольні запитання для самоперевірки.

Рекомендовано для студентів спеціальності 186 «Видавництво та поліграфія» першого (бакалаврського) рівня всіх форм навчання.

**УДК 004.3(07)**

© Харківський національний економічний університет імені Семена Кузнеця, 2018

## Вступ

Навчальна дисципліна «Інформатика і комп'ютерна техніка» вивчається протягом двох семестрів студентами першого курсу за навчальним планом спеціальності 186 «Видавництво та поліграфія». Навчальна дисципліна належить до групи базових навчальних дисциплін і викладається згідно з навчальним планом підготовки фахівців освітнього ступеня «бакалавр». Подано рекомендації до виконання лабораторних робіт, які виконують протягом першого модуля першого семестру.

Мета навчальної дисципліни: набуття студентами необхідних теоретичних основ і практичних навичок використання сучасних інформаційних технологій для розв'язання різноманітних завдань у практичній діяльності за фахом; формування у студентів сучасного рівня інформаційної та комп'ютерної культури, спроможності самостійно використовувати отримані знання під час вивчення нових програмних продуктів у практичній діяльності; підготовка кваліфікованих користувачів персонального комп'ютера, які володіють основними теоретичними положеннями, методами та практичними прийомами аналізу й оброблення комп'ютерної інформації та мають уявлення і вміють професійно працювати з сучасними програмними продуктами.

Результатами виконання лабораторних робіт є засвоєння студентами основних методів та інструментів оброблення інформації різного типу, набуття практичних навичок використання сучасного спеціалізованого програмного забезпечення для розв'язання різноманітних завдань у практичній діяльності за фахом, формування у студентів сучасного рівня інформаційної та комп'ютерної культури, спроможності самостійно використовувати отримані знання під час вивчення нових програмних продуктів у практичній роботі.

Запропоновані методичні рекомендації до лабораторних робіт, які виконують у рамках змістового модуля «Оброблення та зберігання інформації за допомогою СУБД *Microsoft Access*», надають студентам можливість опанувати основні засоби й інструменти та набути практичних навичок роботи з базами даних.

За результатами виконання завдання на лабораторному занятті студенти оформлюють індивідуальні звіти про його виконання та захищають їх перед викладачем.

## **Змістовий модуль 2**

### **Оброблення та зберігання інформації за допомогою СУБД *Microsoft Access***

#### **Лабораторна робота 7**

#### **Зберігання та оброблення інформації за допомогою баз даних на прикладі БД «Борей»**

**Мета** – ознайомитися з основними об'єктами бази даних (БД) *MS Access*: формами та звітами, з'ясувати їх призначення.

Лабораторна робота сприяє напрацюванню таких компетентностей відповідно до Національної рамки кваліфікації:

**знання:**

основних критеріїв вибору СУБД;

призначення основних об'єктів баз даних (форм і звітів) та інструментів для перегляду, пошуку, фільтрації та оброблення даних; елементів управління, розміщених на формах;

**уміння:**

обирати раціональний склад об'єктів бази даних;

визначати набір інструментів для роботи із записами;

здійснювати відбір даних за допомогою фільтрів і пошук даних;

виконувати додавання нових записів і вилучення даних з бази;

**комунікації:**

ефективно формувати комунікаційну стратегію щодо вибору структури бази даних і визначення набору інструментальних, апаратних і програмних засобів;

**автономність і відповідальність:**

відповідальність за вибір СУБД за обраними критеріями.

#### **Основні положення**

До найважливіших функцій ПК належить зберігання інформації (даних). Практично жодне сучасне підприємство не може обійтися без використання баз даних.

Під базою даних (БД) розуміють сукупність взаємозв'язаних даних, призначених для спільного застосування. Інформація в БД структурована та логічно зв'язана таким чином, що занесення, оброблення та вилучення даних з БД за допомогою спеціальних методів доступу до даних відбувається максимально ефективно.

Система управління базами даних (СУБД) – це комплекс програм, що забезпечують взаємодію користувача з базою даних. За допомогою СУБД забезпечується вирішення таких основних завдань:

- створення БД;
- введення, коригування та видалення даних;
- вибір сукупності даних, що відповідають заданим критеріям;
- оформлення вихідних даних і т. д.

Підхід до зберігання та оброблення даних за допомогою БД має такі переваги:

- можливість виключення надмірності збережених даних;
- одноразове введення даних і багаторазове їх використання;
- цілісність, актуальність, достовірність і несуперечливість інформації, що зберігається;
- адаптація інформаційно-логічної моделі до змін предметної області;
- видача інформації в формі, яка визначається користувачем;
- задовільнення інформаційних потреб користувачів різних типів (керівник, виконавець, оператор та ін.);
- санкціонований доступ до даних тощо.

Навчальний модуль присвячується розробленню, створенню та веденню реляційних баз даних засобами *MS Access*.

### **Опис БД «Борей»**

Лабораторна робота 7 ознайомлює з готовою навчальною базою даних «Борей» з метою продемонструвати призначення та можливості форм і звітів як основних об'єктів, з якими працює кінцевий користувач.

Навчальна база даних розроблена для фірми «Борей», яка займається продажем оптовим клієнтам продуктів харчування, тобто основною метою створення та ведення цієї бази є оформлення та облік замовлень на поставку товарів.

Для реалізації цієї мети в базі даних міститься довідкова інформація:  
про товари: назва (або марка) товару, категорія (або тип товару),  
ціна тощо;

про постачальника: назва фірми-постачальника, адреса, контактна  
особа тощо;

про клієнтів: назва фірми-покупця, адреса і т.п.;

про співробітників фірми «Борей»: ПІБ, посада, дата народження  
та інші дані.

На підставі цих даних створюється замовлення.

Користувач бази отримує доступ до необхідної інформації через  
екранні форми. Доступ до форм здійснюється через головну екранну  
форму бази. За допомогою форм можна переглядати та фільтрувати  
записи, також додавати нові записи в базу або видаляти існуючі.

### Хід роботи

1. Відкрийте файл Борей.mdb.

*Увага! З відкриттям файлу на екрані з'являються попередження  
системи безпеки про блокування небезпечних виразів.*

На екрані з'явиться головна екранна форма БД (рис. 1).

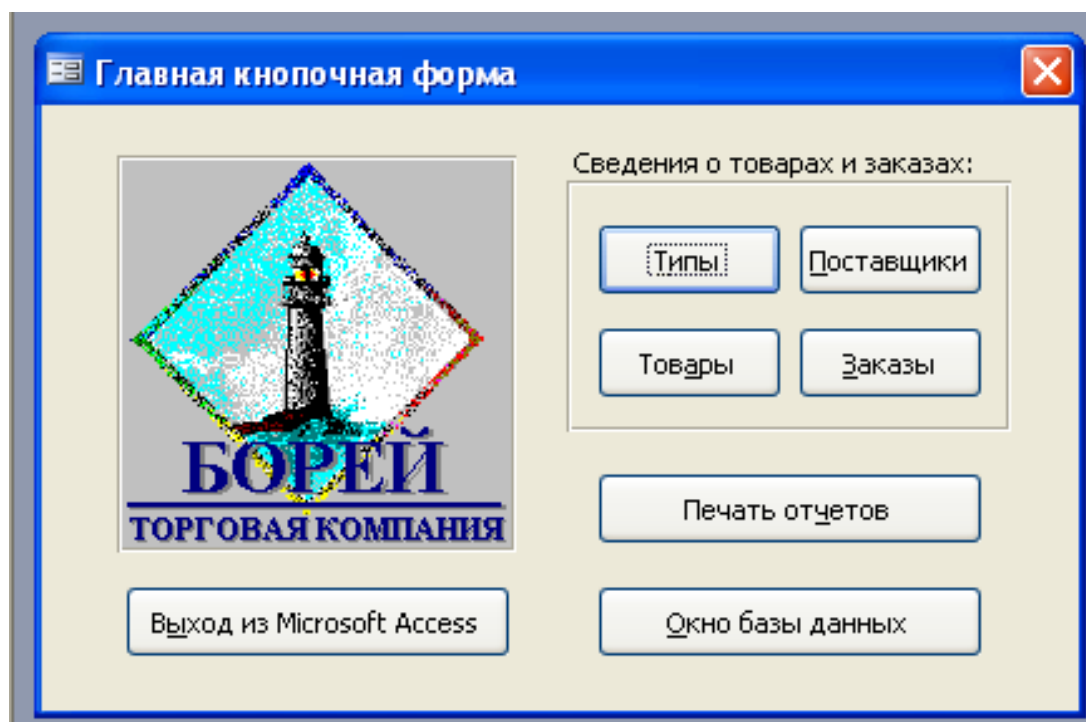


Рис. 1. Головна екранна форма БД «Борей»

Кнопки <Типи>, <Постачальники>, <Товари> та <Замовлення> дозволяють відкрити відповідні форми та отримати доступ до інформації, що зберігається в базі.

Відкрийте форму «Товари» (рис. 2).

The screenshot shows a window titled "Товари" (Goods) with a blue title bar. Inside the window, there are two buttons at the top right: "Просмотр списка товаров" (View list of goods) and "Вывод списка в файл HTML" (Export list to HTML file). The main area contains a form with the following fields:

Товар:	1
Марка:	Genen Shouyu
Поставщик:	Mayumi's
Тип:	Приправы
Единица измерения:	24 бутылки по 250 мл
Цена:	697,50р.
На складе:	39
Ожидается:	0
Минимальный запас:	5
Поставки прекращены:	<input checked="" type="checkbox"/>

At the bottom left, there is a "Запись:" (Record:) label followed by navigation buttons (back, forward, search) and a page indicator "1 из 77" (1 of 77).

Рис. 2. Форма «Товари»

Таку форму називають «формою в стовпець» (стовпчиковою), тому що вся інформація про один товар (тобто про один запис бази даних) виводиться в текстових полях (TextBox) і полях зі списком (ComboBox), які розташовані вертикально один під одним в один стовпець. Зліва від полів на формі знаходяться написи (Label), які зазвичай містять підписи до полів. Для переходу по записах (тобто для навігації) використовують кнопки, розташовані в рядку стану (див. рис. 2). Тут же відображається загальна кількість записів, доступних через форму.

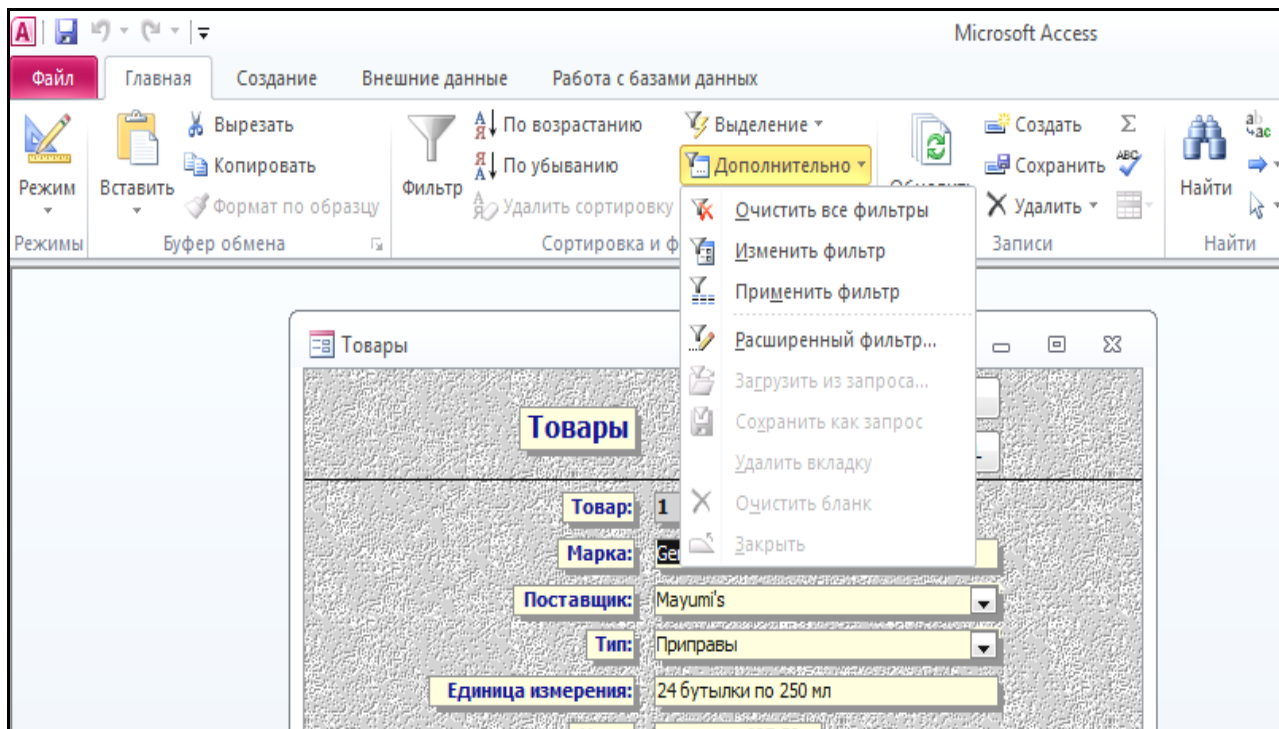
2. Визначте, які товари знаходяться під номерами 11, 54, 60, а також їх ціну, одиницю вимірювання та запас на складі; занесіть їх у звіт.

Для швидкого переходу до потрібного запису введіть її номер у поле номера поточного запису та натисніть <Enter>.

3. За допомогою кнопок сортування визначте найдорожчий (сортування за ціною за зменшенням і перехід на запис № 1) і найдешевший

товар. Запишіть у звіт дані про ці товари. Під час сортування не забудьте встановити курсор у поле «Ціна».

Крім навігації по записах, форми дозволяють упорядковувати (сортувати) дані та знаходити потрібні дані за допомогою фільтрів (рис. 3).



**Рис. 3. Палітра «Сортування та фільтрація»,  
меню кнопки «Додатково»**

Для того, щоб відновити вихідний порядок записів і скасувати сортування, виконайте команду «Додатково – Очистити всі фільтри» або натисніть кнопку <Видалити фільтр>.

У формах MS Access є такі види фільтрів: автофільтр (рис. 4), фільтр за виділенням, фільтр за формою і розширений фільтр.

Фільтр за виділенням дозволяє вибрати записи, для яких значення обраного поля співпадає з тим значенням, яке знаходиться в поточному (виділеному) полі.

Фільтр за формою визивають командою «Додатково – Змінити фільтр» (рис. 5), він дозволяє у пустій формі обирати умови відбору записів за допомогою операторів порівняння та символи підстановки, (наприклад, «\*»), задавати кілька умов фільтрації за допомогою ярличків «Або».



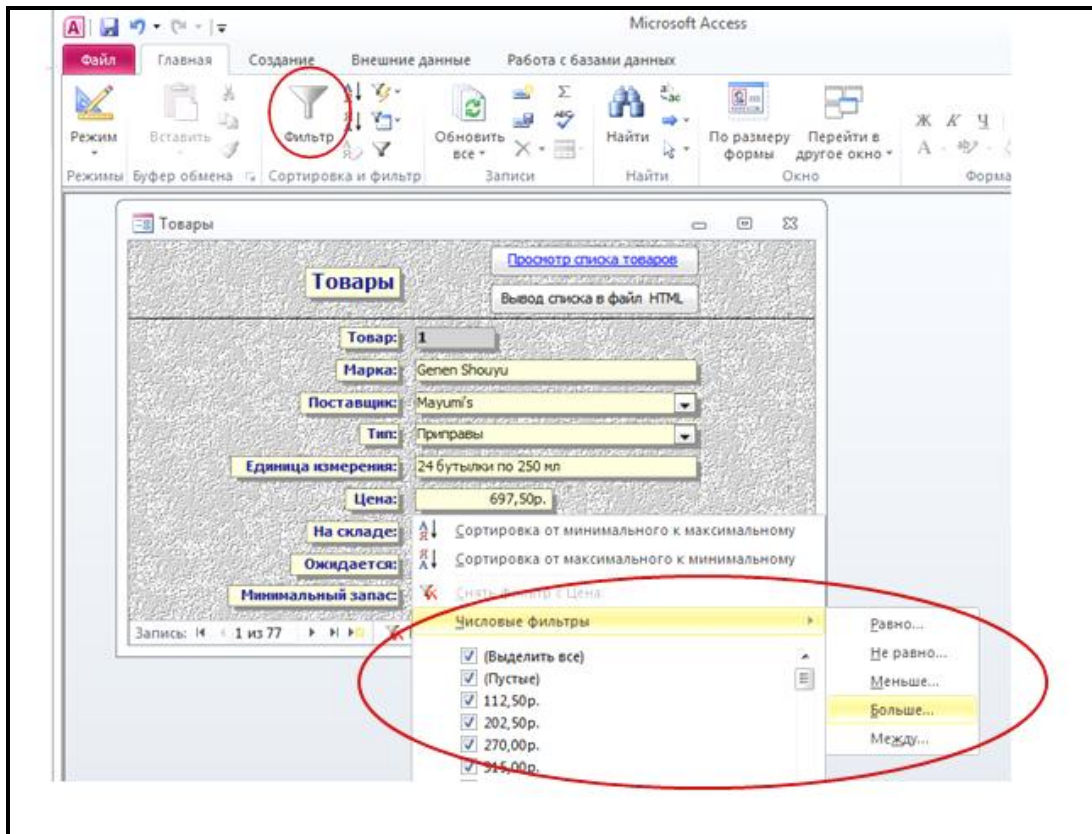


Рис. 4. Автофильтр

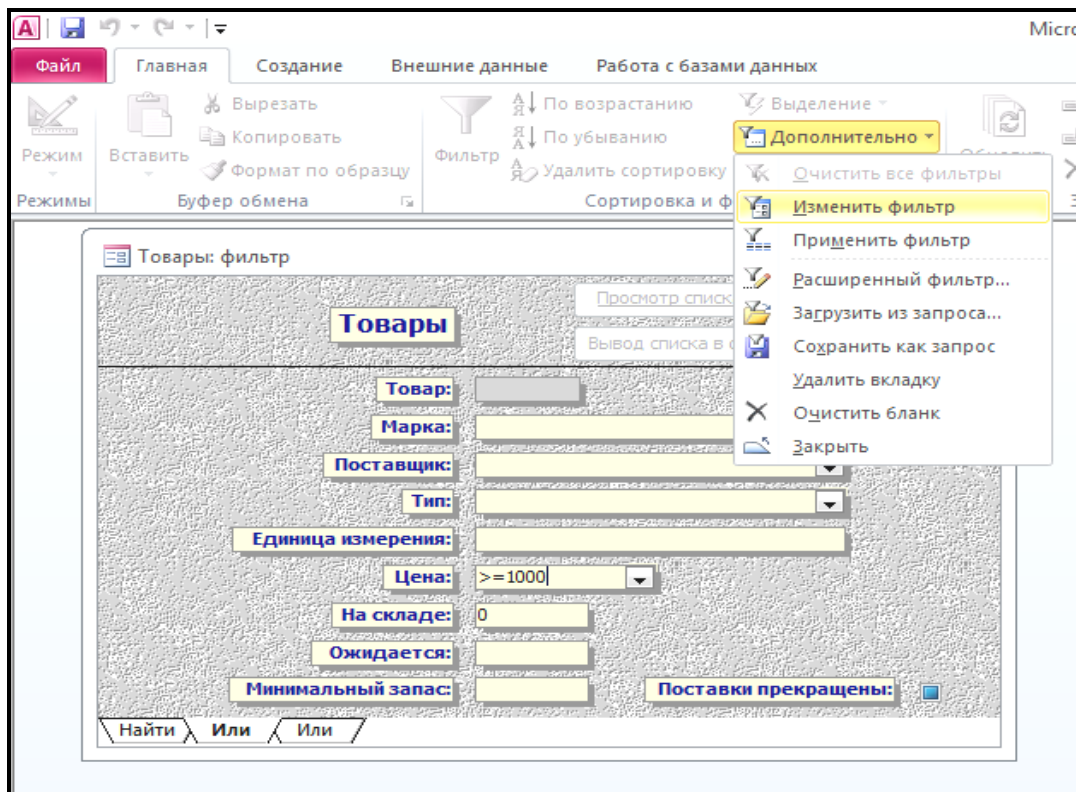


Рис. 5. Фильтр за форму (кнопка «Змінити фільтр»)

4. Ознайомтеся з фільтрацією.

4.1. За допомогою фільтра на виділеному визначте, яку кількість різних марок товарів представлено в таких категоріях (тип): Напої, Приправи, Фрукти, Хлібобулочні вироби.

*Увага! Перед черговим застосуванням фільтра попередній фільтр слід видалити.*

4.2. Визначте кількість різних марок товарів за категоріями Напої і Приправи. Порівняйте отримані в пунктах 4.1 і 4.2 результати, і якщо є розбіжності, то з'ясуйте, де зроблено помилку.

4.3. Знайдіть товари з ціною дорожче 4 000 р.

4.4. Визначте загальну кількість товарів, яких вже немає на складі (поле «На складі»), а також тих товарів, постачання яких припинено (поле «Поставки припинено»). Для вирішення цього завдання використовуйте ярличок «Або» (див. рис. 5).

Не забудьте видалити фільтр після розв'язання задачі.

5. За допомогою форми «Товари» додайте ще один товар. Для цього можна використати команду «Головна – Записи – Створити» або відповідну кнопку в рядку стану форми.

Під час додавання нового запису зверніть увагу на таке:

номер товару (поле «Товар» на формі) заблокований для введення і має тип Лічильник. Це означає, що значення поля автоматично збільшується на одиницю з додаванням нового запису;

для полів зі списком («Тип» і «Постачальник») значення обмежені списком.

Нові записи автоматично зберігаються з переходом на інший запис або із закриттям форми.

6. Закрийте та знову відкрийте форму «Товари», переконайтеся, що в ній відображається доданий вами запис (він матиме останній номер товару, сформований автоматично полем типу Лічильник). Видаліть цей запис командою «Головна – Видалити запис».

Зверніть увагу, що номер товару в останньому записі тепер на одиницю більше, ніж у видаленому записі.

7. Самостійно вивчіть форми «Типи», «Постачальники» і «Замовлення».

Форма «Типи» має більш складну структуру, тому що для кожного типу (категорії) товару можна переглядати список товарів, які належать до даного типу. Таку форму називають складною підпорядкованою (рис. 6). У неї вбудована підпорядкована форма «Товари».

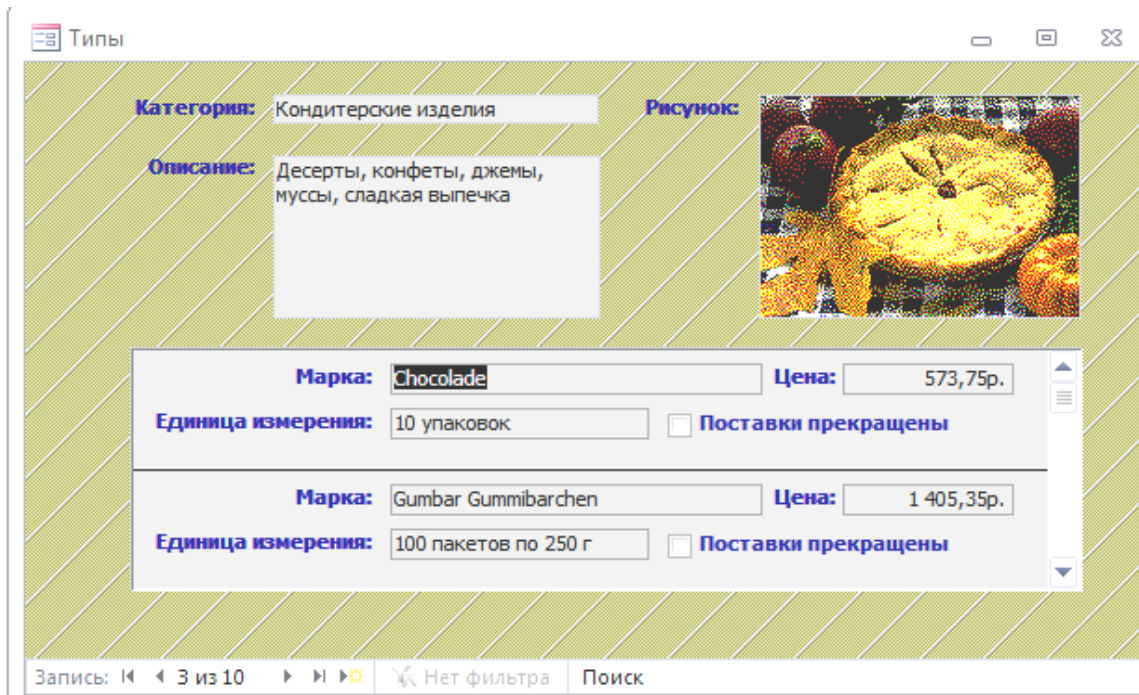


Рис. 6. Складна форма «Типи»

7.1. Визначте призначення кнопок <Перегляд товарів> і <Введення товарів> на формі «Постачальники». Запишіть свій висновок до звіту роботи.

7.2. Визначте, інформація про яку кількість замовлень зберігається в базі (форма «Замовлення»).

7.3. Оформіть два нових замовлення. У полі «До оплати» виберіть замовника (або клієнта), для якого робите замовлення. За замовчуванням він же є одержувачем (тому всі дані про одержувача заповнюються автоматично). Замовлення оформляє співробітник фірми «Борей» (поле «Продавець»). Номер замовлення (поле «Замовлення») присвоюється автоматично, тому дане поле заблоковане для введення даних. Виберіть спосіб доставки та дати призначення та виконання. Виберіть товари для замовлення, їх кількість, знижку. Запишіть у звіт дані про зроблені замовлення: код замовлення, кількість та вартість товарів, вартість доставки та замовлення. Зробіть висновок про додаткові функції форм (крім розглянутих).

8. Натисніть на головній кнопковій формі кнопку <Друк звітів>, а потім команду «Перегляд» і перегляньте кілька звітів. Зробіть і запишіть у звіт висновок про їх призначення. Інтервал дат, який можна вводити у ході перегляду даних про співробітників і країни, знаходиться в діапазоні від 01.01.1996 до 30.07.1998.

Звіти відобразяться на екрані, вони не будуть роздруковані на принтері.

## Контрольні запитання

1. Які дії можна здійснити за допомогою форм?
2. Які види фільтрів застосовують у *MS Access*?
3. Порівняйте призначення форм і звітів.
4. Що таке складна форма? Наведіть приклади складних форм.
5. Як зберегти новий запис, створений через форму?
6. Які елементи можуть бути присутніми на формі?
7. В якому випадку поля форми можуть бути заблоковані для введення даних?

## Лабораторна робота 8

### Проектування і створення індивідуальних баз даних

**Мета** – отримати навички з розроблення складу таблиць БД і зв'язків між ними; навчитися створювати таблиці різними способами; ознайомитися з основними властивостями полів різних типів даних.

Лабораторна робота сприяє напрацюванню таких компетентностей відповідно до Національної рамки кваліфікації:

**знання:**

типів даних полів таблиць *MS Access*;  
зв'язків (відношень) між таблицями реляційних БД;  
переваг та обмежень, що надає встановлення параметру «Забезпечення цілісності даних»;

**уміння:**

визначати перелік таблиць і склад їх полів для певних предметних областей;

установлювати відношення між таблицями та їх параметри;

визначати типи даних полів таблиць та їх властивості;

**комунікації:**

презентувати результати визначення структури баз даних та аргументувати їх доцільність;

**автономність і відповідальність:**

відповідальність за точність, коректність та оптимальність вибору структури таблиць БД.

## Основні положення

Кожна БД містить інформацію про певні об'єкти. Сукупність таких об'єктів називають предметною областю БД (це можуть бути, наприклад, дані про клієнтів, відомості про вироблювану продукцію тощо). Інформація про об'єкти предметної області міститься в різних і, як правило, табличних документах (картки обліку кадрів, картки складського обліку, накладні та ін.).

Для створення комп'ютерної бази даних необхідний формалізований опис цих документів (назви документів, колонки, з яких вони складаються) і логічних зв'язків між ними. Для табличних документів подібну схему називають реляційною моделлю даних, а самі відносини – реляційними (від англ. relation – відношення, зв'язок). В Access графічне зображення реляційної моделі прийнято називати схемою даних.

Рядок реляційної таблиці, яка містить сукупність відомостей про один об'єкт БД, називають *записом*. Стовець реляційної таблиці, що містить однорідні відомості про всі об'єкти таблиці, називають *полем*. Коли мова йде про введення даних в певний запис, окрему клітинку на перетині шпальти та рядок таблиці також називають *полем*.

Структура таблиці визначається сукупністю її полів. *Поле* – елементарна одиниця структури БД. У кожному полі (стовпці) міститься однорідна інформація про всі об'єкти таблиці (наприклад, тільки ціна виробу). Дані кожного поля можуть бути тільки одного з типів: рядок символів (текст), число, дата та ін.

У кожному рядку таблиці містяться логічно зв'язані різномірні дані про один об'єкт БД. Наприклад, запис про конкретний товар на складі може містити код товару і його залишок на поточний момент, запис про певного працівника підприємства – табельний номер, ПІБ працівника, домашню адресу, номер телефону тощо.

Основу реляційної моделі складають двовимірні (плоскі) таблиці, які повинні мати такі властивості:

- кожен стовець таблиці має унікальне (не повторюється в цій таблиці) ім'я, не допускається об'єднання осередків у «шапці» таблиці;

- усі дані в стовпці (поле) однорідні за змістом і за формою подання;

- у таблиці не повинно бути однакових рядків (записів), тобто значення хоча б одного поля має бути унікальними;

- дублювання інформації в таблиці повинно бути зведене до мінімуму;

- порядок проходження полів і записів у таблиці не важливий (все визначається лише логічними зв'язками між полями).

Щоб СУБД «знала», з яким записом у даний момент вона працює, запис потрібно ідентифікувати (відрізнити від інших). Реляційна модель будується таким чином, щоб у жодній таблиці не було двох однакових записів. На практиці для ідентифікації записів використовують ключ (синоніми – первинний ключ, ключове поле).

*Первинним ключем* називають поле, кожне з унікальних значень якого точно визначає один запис.

*Ключове поле* вибирається з існуючих у таблиці полів, значення яких не співпадуть за жодних умов (наприклад, номер паспорта), або додатково вноситься в таблицю. Після детального розроблення логічної структури бази даних можна приступати до її безпосередньої реалізації в *MS Access*.

Реальні реляційні БД містять багато пов'язаних між собою таблиць і можуть оброблятися тільки засобами СУБД. Подання БД кількома таблицями зводить до мінімуму дублювання інформації, спрощує технологію її оброблення. Стає технічно простіше реалізувати санкціонований доступ користувачів до закритої інформації.

Робота в *MS Access* починається зі створення та збереження нової БД.

У *MS Access* для створення будь-якого нового об'єкта (в тому числі і таблиць) використовують меню «Створення» та такі інструменти: «Таблиця» – для створення таблиці шляхом введення даних і «Конструктор таблиць», де користувач самостійно визначає типи даних полів та їх властивості (рис. 7).

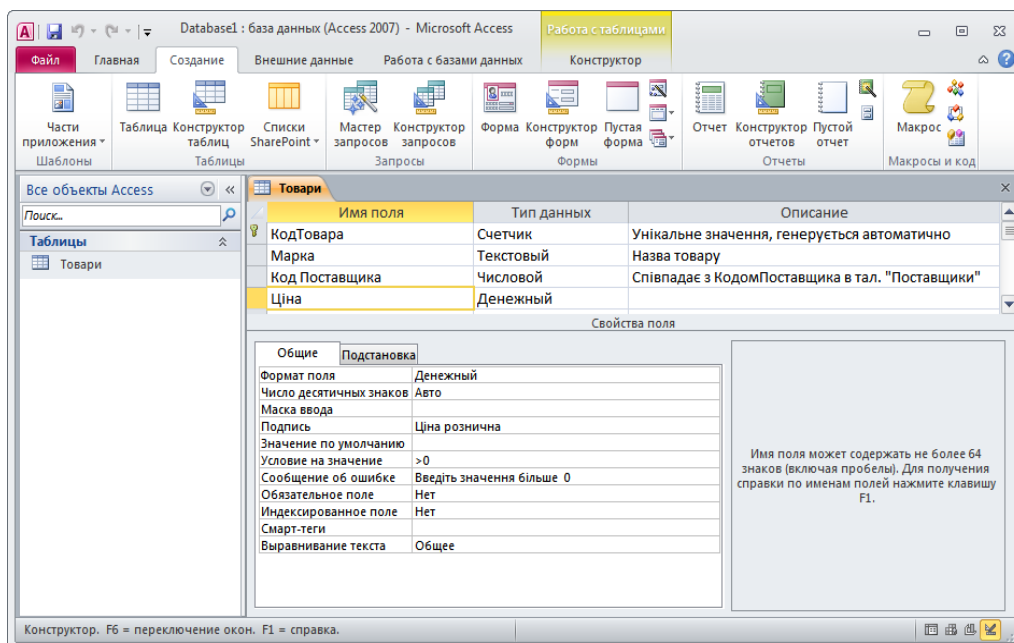


Рис. 7. Створення таблиці «Товари» за допомогою Конструктора таблиць

Бланк конструктора таблиць по горизонталі розділений на дві частини. У верхній частині знаходяться три стовпці: «Ім'я поля», «Тип даних», «Опис». Стовпець «Опис» заповнювати не обов'язково.

Нижня частина вікна Конструктора таблиць призначена для визначення властивостей поточного поля (зокрема, на рис. 7 показані властивості поля «Ціна», тому що воно є поточним). Загальні властивості поля залежать від обраного типу даних.

Після введення імені поля в сусідньому стовпці Тип даних за замовчуванням відображається тип даних Текстовий. Якщо в поле будуть зберігатися дані іншого типу, його потрібно вибрати зі списку (рис. 8).

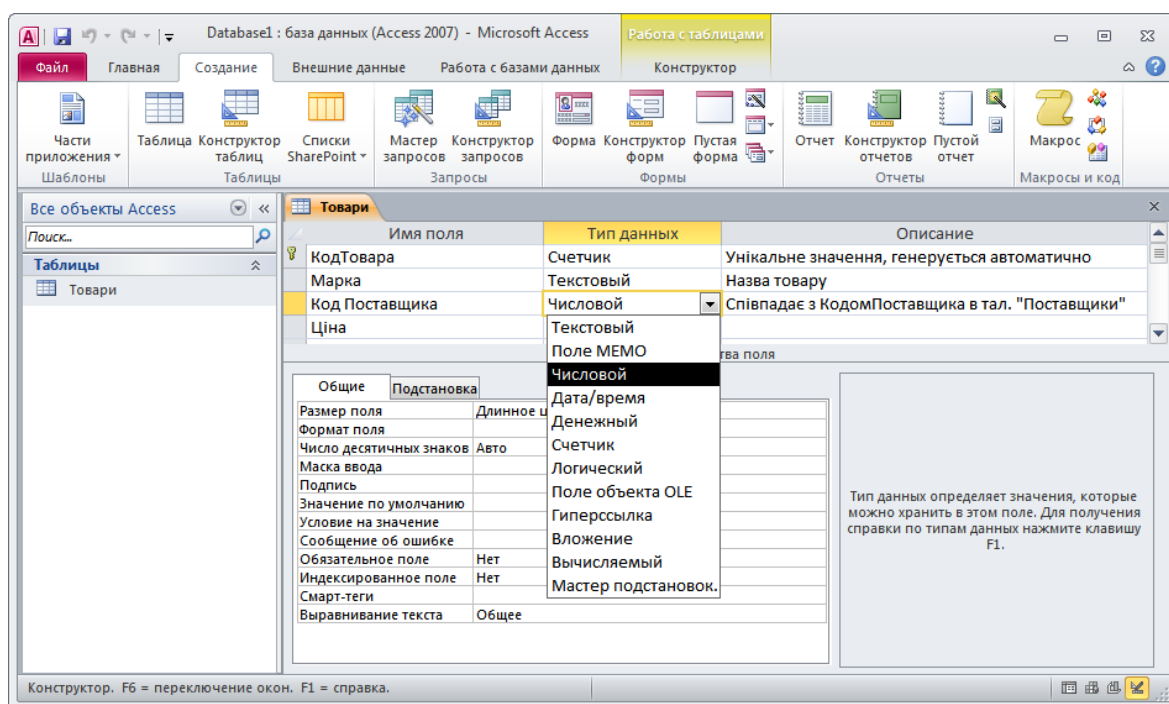


Рис. 8. Вибір необхідного типу даних

У нижній частині бланка Конструктора таблиць встановлюють необхідні загальні властивості поля (зверніть увагу, що список властивостей залежить від типу даних).

Потім призначають первинний ключ. Для цього виділяють рядок з ім'ям поля, яке повинно бути ключовим, і натискають відповідну кнопку на панелі інструментів.

Відповідність записів визначається збігом значень пов'язаних полів.

Для кожної з пари пов'язаних таблиць розрізняють головну та підпорядковану таблиці. Головними в зв'язках роблять, як правило, довідкові таблиці, інформація яких змінюється порівняно рідко. Головні таблиці

бажано заповнювати першими. Підлеглі таблиці зазвичай зберігають часто оновлювані оперативні дані та використовують для заповнення даних у головних таблицях.

Найбільш поширеним типом зв'язку між таблицями є відношення «один-до-багатьох». Це таке відношення, за якого одному запису головної таблиці відповідає до декількох записів підпорядкованої таблиці (причому ніякому запису підпорядкованої таблиці не повинно відповідати більше одного запису головної таблиці).

Установлення зв'язку між таблицями здійснюється у вікні «Схема даних» команди «Робота з базами даних» шляхом перетягування ключового поля головної таблиці на відповідне поле підпорядкованої таблиці. Приклад вікна схеми даних для БД «Борей» наведено на рис. 9.

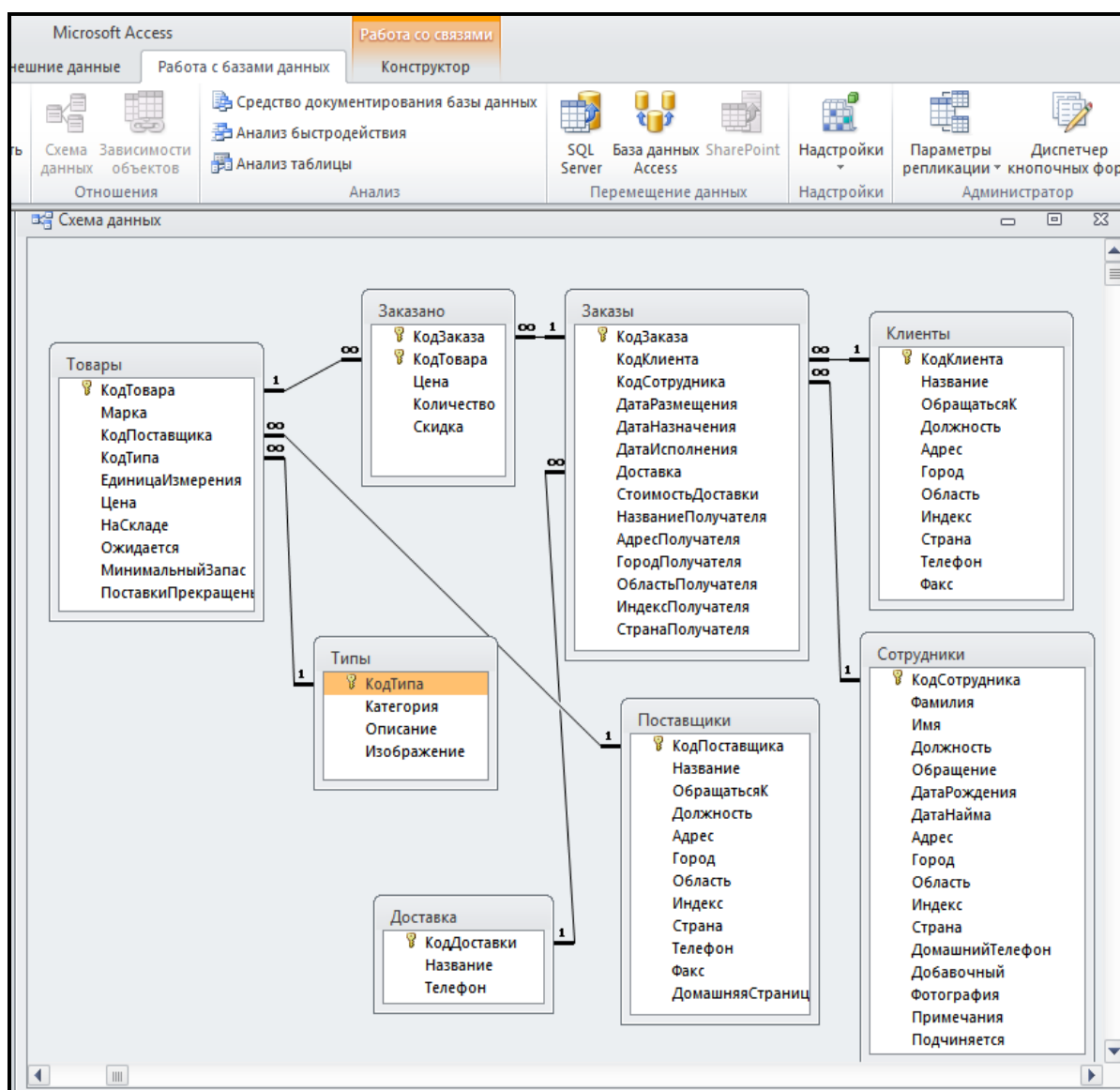


Рис. 9. Схема даних БД «Борей»



Вікно «Схема даних» призначене для перегляду, створення та редагування зв'язків між полями таблиць, а також для визначення властивостей зв'язків між таблицями (рис. 10).

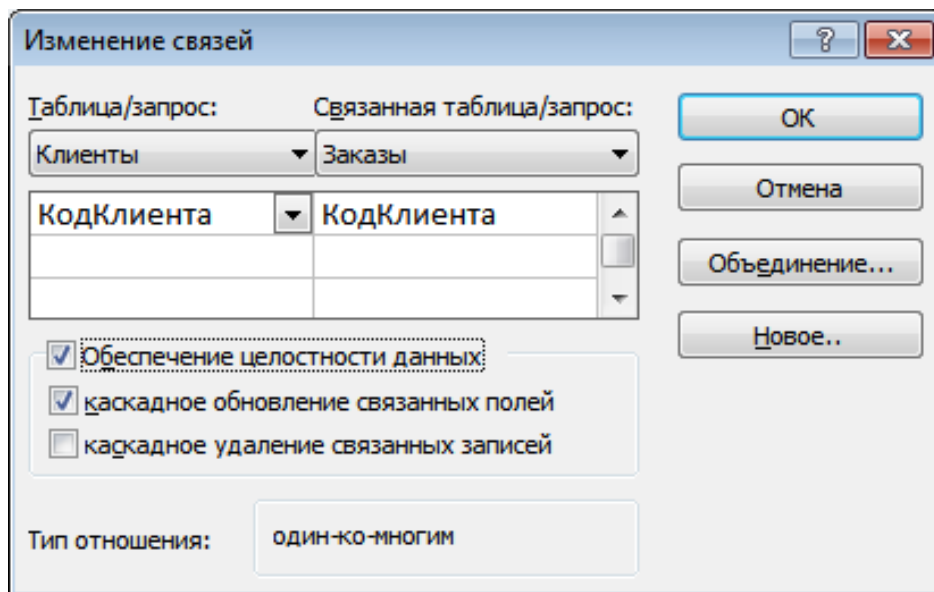


Рис. 10. Вікно «Змінення зв'язків» таблиць «Клієнти» та «Замовлення» БД «Борей»

Наприклад, між таблицями «Клієнти» та «Замовлення» не встановлена властивість каскадного видалення пов'язаних записів. Тому видалення запису про певного клієнта з таблиці «Клієнти» не призводить до видалення відомостей про його замовлення з таблиці «Замовлення».

У табл. 1. наведено перелік можливих предметних областей для індивідуальних БД студентів.

Таблиця 1

### Предметні області створення індивідуальних БД

№ варіанта	Назва БД	Загальна постановка задачі
1	2	3
1	Автомайстерня	Облік виконаних ремонтних робіт в автомайстерні для поточного та підсумкового аналізу роботи майстерні. Ведеться облік клієнтів і автомобілів

1	2	3
2	Мережа АЗС	Облік операцій з продажу різних видів палива в мережі АЗС. Ведеться облік поставок і поставників
3	Аптека	Облік операцій з реалізації медичних препаратів у роздрібній торгівлі в окремій аптеці міста. Ведеться облік поставок препаратів і поставників
4	Депозитні вклади клієнтів філії банку	Облік операцій обслуговування клієнтів. Різні види вкладів. Списки боржників
5	Бібліотека	Облік операцій обслуговування читачів. Ведеться облік читачів, літератури, а також роботи персоналу бібліотеки
6	Товарна біржа	Облік біржових операцій. У базі зберігаються дані про фірми-продавці, фірми-покупці, а також дані про брокерів, які обслуговують складені угоди
7	Сировинна біржа	Облік біржових операцій. В базі зберігаються дані про фірми-продавці, фірми-покупці, а також дані про брокерів, які обслуговують складені угоди
8	Відеотека	Оперативний облік роботи пункту прокату фільмів на різних носіях
9	Клієнти готелю	Оперативний облік обслуговування клієнтів готелю. Ведеться облік клієнтів, персоналу, номерів
10	Деканат. Успішність студентів одного курсу	Оперативний облік успішності студентів за дисциплінами одного курсу
11	Кадрова агенція	Ведеться облік резюме здобувачів і заявок роботодавців, а також облік з виконаного підбору персоналу (укладені контракти)
12	Магазин «Книжковий світ»	Ведеться облік роздрібного продажу книжок у книжковому магазині. Необхідно мати інформацію про поставників і поставки літератури різних жанрів
13	Меблі на замовлення	Види меблів, матеріалів, виробників, замовлення

1	2	3
14	Заходи в організації	Заходи, співробітники, план проведення заходів
15	Обмін валют	Ведеться облік роздрібного продажу та купівлі валют у мережі пунктів обміну. У базі фіксується кожна проведена операція з обміну валюти, а також поточний курс купівлі та продажу
16	Продаж авіаквитків	Рейси, пасажери, продаж білетів
17	Продаж залізничних квитків	Ж/д маршрути, зони, продаж білетів
18	«Продсервіс» – продаж продуктів харчування за дисконтними картками	Робочі місця продавців (номер касового апарату), товари, клієнти, які мають дисконтну картку, фіскальні чеки (облік продажу)
19	Розклад занять у ВНЗ	Аудиторії, дисципліни, викладачі, групи, розклад
20	Склад – Облік матеріалу	Постачальники, споживачі, матеріали, облік прийому/відпуску матеріалів
21	Митниця – Ввезення автомобілів	Мита на марки автомобілів, громадяни, проходження митного контролю
22	Телеательє	Вид техніки, клієнти, облік ремонтних операцій
23	Туристична фірма	Опис турів, клієнти, продаж путівок
24	Облік робочого часу персоналу організації	Співробітники, таблиця реєстрації часу початку та завершення робочого процесу
25	Фірма з продажу комп'ютерів	Монітори, процесори, відеокарти, ПЗУ, ОЗУ, формування конфігурації
26	Вибори	Виборчі дільниці, кандидати, результати голосування
27	Видавництво	Штат, заплановані видання, етапи підготовки до видання, виконання видавничого плану
28	Телестудія	Співробітники, опис проектів, виходи в ефір
29	Пошта. Підписка клієнтів на періодичні видання	Періодичні видання, передплатники, оформлення підписки
30	Поліклініка	Оперативний облік звернень пацієнтів до різних лікарів

## Хід роботи

1. Виберіть тематику своєї БД за погодженням з викладачем – самостійно або зі запропонованої табл. 1 з варіантами завдань.
2. Продумайте перелік вихідних документів і звітів.
3. Складіть перелік завдань, що вирішуються за допомогою даної БД.
4. З'ясуйте, які вихідні дані необхідні для вирішення поставлених завдань, які розрахунки необхідно виконати та за якими формулами.
5. Складіть перелік таблиць, що містять вихідні дані (не менше п'яти таблиць).
6. Продумайте структуру таблиць – склад полів, тип даних, діапазон можливих значень, ключові поля.
7. Накресліть на папері в звіті схему даних БД – таблиці зі складом полів і зв'язок між ними, визначте тип зв'язків.
8. Виконайте нормалізацію таблиць з метою уникнення дублювання даних.
9. Узгодьте з викладачем склад таблиць та їх полів, ключові поля, зв'язки між таблицями.
10. Створіть базу даних за своїм варіантом, необхідні таблиці, згідно з предметною областю, встановіть типи даних (лічильник, текстовий, числовий і т. д.), опис та інші необхідні властивості полів (розмір поля, маску введення, підпис, значення за замовчуванням тощо) створених таблиць. Визначте первинні ключі в створених таблицях.
11. Визначте необхідні зв'язки між таблицями, задайте необхідні параметри забезпечення цілісності даних і вид об'єднання. Проаналізуйте й обґрунтуйте у звіті, які параметри потрібні для кожного зв'язку між таблицями.
12. Заповніть створені таблиці даними (мінімум десять записів у кожній таблиці).

## Контрольні запитання

1. Як ви розумієте термін «відношення» в реляційній БД?
2. Яким вимогам повинен задовільнювати первинний ключ таблиці?
3. Чи обов'язково повинні збігатися імена пов'язаних полів? Які властивості повинні бути однаковими у пов'язаних полів (у первинного і у зовнішнього ключів)?

4. У чому полягає сенс відносини типу «один-до-багатьох»? Наведіть свої приклади.

5. У чому полягає сенс відносини типу «багато-до-багатьох»? Наведіть свої приклади.

6. Як ви розумієте цілісність даних? Чим вона забезпечується?

7. Чи є обов'язковими властивості зв'язку «каскадне оновлення пов'язаних полів» і «каскадне видалення зв'язаних записів»? Обґрунтуйте свою відповідь. Наведіть приклади.

## Лабораторна робота 9

**Робота з запитам на вибірку. Запити з групуванням даних.**

**Перехресні запити. Запити з параметрами.**

**Модифікуючі запити**

**Мета** – ознайомитися з основними можливостями та різновидами запитів на вибірку; освоїти правила побудови умов фільтрації даних; ознайомитися з призначенням, синтаксисом та особливостями використання модифікуючих запитів; отримати навички створення запитів на створення, додавання, поновлення та видалення даних у БД.

Лабораторна робота сприяє напрацюванню таких компетентностей відповідно до Національної рамки кваліфікації:

**знання:**

синтаксису запитів на вибірку, модифікуючих і групових запитів та запитів з параметрами;

правил побудови виразів і параметрів основних вбудованих функцій *MS Access*;

особливостей застосування різних типів запитів;

**уміння:**

здійснювати вибір інструментарію для вирішення конкретного прикладного завдання з ведення БД;

створювати запити з використанням складних умов фільтрації;

застосовувати основні вбудовані функції *MS Access*;

обчислювати підсумки за групами записів;

зберігати потрібні дані у вигляді нових таблиць;  
редагувати значення полів таблиць для записів, що відповідають певним умовам;

видаляти записи з однієї або кількох пов'язаних таблиць;

додавати записи в кінець існуючих таблиць;

**комунікації:**

презентувати результати розв'язання прикладних завдань з фільтрації даних, створення нових таблиць, видалення або додавання нових записів, редагування значень полів таблиць; обчислення виразів і підсумків за групами значень;

аргументована взаємодія з командою учасників проекту щодо застосованого інструментарію для побудови запитів;

рекомендації співробітникам підприємства щодо реакції системи у випадку застосування некоректних запитів;

**автономність і відповідальність:**

самостійно приймати рішення та відповідати за коректність і адекватність використаного інструментарію;

прийняття рішення щодо доцільного вибору типу запитів і математичного апарату.

## Основні положення

Запити є основним інструментом СУБД для аналізу й управління даними: це інструкції (вираження), записані спеціальною мовою SQL (Structured Query Language).

Запит на вибірку даних – найпоширеніший вид запитів у *MS Access*. За допомогою запитів на вибірку можна виконувати такі дії з даними:

об'єднувати поля з різних таблиць в одній тимчасовій (динамічній) таблиці;

обмежувати набір даних, що відображаються за допомогою логічних виразів – умов відбору даних;

будувати обчислювані поля.

Для створення запитів на вибірку в *MS Access* передбачені різні способи: за допомогою майстрів зі створення запитів; у режимі конструктора; в режимі SQL.

Режим конструктора надає користувачам простий, інтуїтивно зрозумілий інтерфейс для створення запитів будь-якої складності.

Результати виконання запиту подаються в табличній формі.

Доступ користувачів до таблиць зазвичай закривають, оскільки для однієї і тієї ж таблиці можна створити безліч різних запитів, що надають користувачам тільки необхідну їм інформацію. Наприклад, бухгалтер заводу матиме можливість використати запит, хто зі співробітників скільки днів прохворів, проте доступ до запиту, що дозволяє дізнатися, чим співробітник хворів і де лікувався, може бути тільки у лікаря медичної частини заводу.

Запити можуть бути побудовані на основі однієї (однотабличні) або кількох таблиць (багатотабличні). Результат виконання багатотабличних запитів формується на основі збігу значень пов'язаних полів базових таблиць. Якщо такого збігу немає, результатом запиту буде порожня таблиця.

Вікно конструктора запитів складається з двох частин (рис. 11).

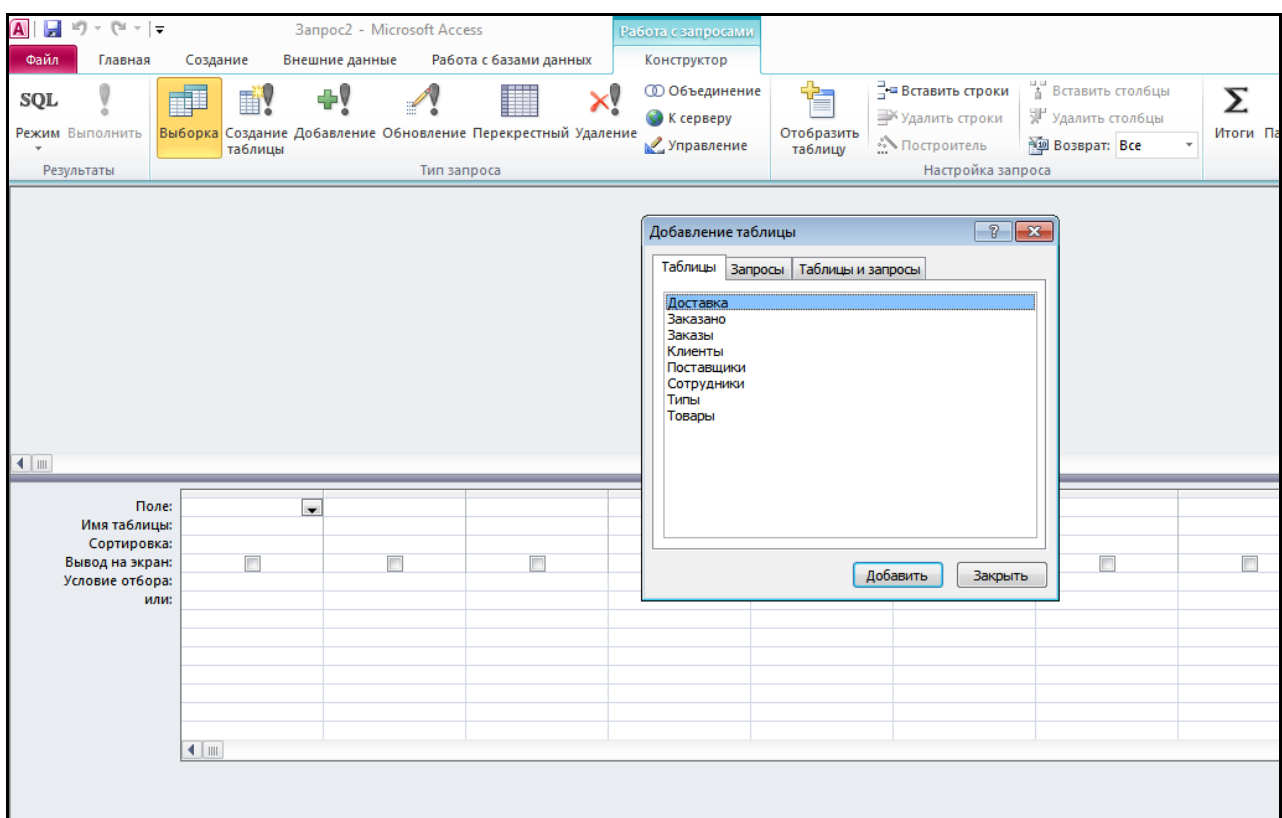


Рис. 11. Вікно бланка запиту на вибірку

У верхню область вікна бланка запиту додаються таблиці, на підставі яких будується запит, тобто ця область відображає частину схеми даних.

У нижній області бланку запиту вказують:

імена полів, що відбираються запитом (можна з різних таблиць);

ім'я таблиці, з якої вибирається дане поле;

можливість задати сортування на даному полі;

чи потрібно виведення поля на екрані в результаті виконання запиту;

умови фільтрації у вигляді логічних виразів, відповідно до яких вибираються тільки ті записи, які задовільнюють умовам. Рядок АБО та всі рядки, що розташовані нижче, дозволяють задавати додаткові умови до поля. Умови будуть виконуватися незалежно одна від іншої, тобто реалізують логічну функцію АБО.

*Увага! Слід пам'ятати, що запит – це послідовність інструкцій з фільтрації записів. З виконанням запиту щоразу створюється так звана динамічна таблиця для виведення записів, із закриттям запиту вона не зберігається.*

У табл. 2 наведено спеціальні символи, які можна використовувати для побудови умов фільтрації.

Таблиця 2

### Спеціальні символи, використовувані для побудови умов фільтрації

Символи	Опис
*	Будь-яка кількість символів
?	Один будь-який символ
#	Будь-яка цифра від 0 до 9
[A-Z]	Будь-який символ, що входить у зазначений діапазон
[!A-Z]	Будь-який символ, який не входить у зазначений діапазон. Також використовується в поєднанні з шаблонами «*» або «?»
[A, D, H]	Будь-який символ із зазначеного списку



У табл. 3 подані приклади з використання спеціальних символів, використовуваних для побудови умов фільтрації.

Таблиця 3

### Приклади використання оператора Like

Послідовність символів	Приклад умови фільтрації	Результат
“aBBa”	Like “a*a”	Істина
“Форма”	Like [А-Я]*	Істина
“a2a”	Like “a#a”	Істина
“BAT123 khg”	Like “B?T*”	Істина
“Форма”	Like [А, М, Ф]*	Істина
1.01.19 – 31.01.09	Like ##.01.19	Істина

Запит з параметром дозволяє користувачеві самому змінювати умови відбору записів або використовувати змінні величини в обчислюваних виразах за багаторазового використання запиту. Його формат:

*[Текст звернення до користувача].*

Текст звернення до користувача, з одного боку, є ім'ям параметра, з іншого – містить підказку, яку саме інформацію необхідно ввести. У бланк запиту (зазвичай у рядок «Умова відбору») поміщається текст запиту з параметром у тому полі, за яким задається критерій відбору. Можна сформулювати запит з кількома параметрами. Вони будуть з'являтися на екрані в тому порядку, в якому розташовані в бланку запиту (зліва направо).

Із запуском такого запиту спочатку з'являється діалогове вікно для введення значення, і тільки після введення значення параметра запит буде виконаний.

Параметри можна включати до складу логічних виразів.

Наприклад, за запитом користувач може відібрати відомості про замовлення з ціною не менше заданого значення, причому граничне значення визначає сам користувач у ході виконання запити:

$> = [Введіть мінімальну ціну]$ .

Вікно запити з параметром наведено на рис. 12.

Можна задати виконання кількох умов одночасно, наприклад, обрати товари з ціною у заданих межах:

$> = [Введіть мінімальну ціну] \text{ And } < = [Введіть максимальну ціну]$ .

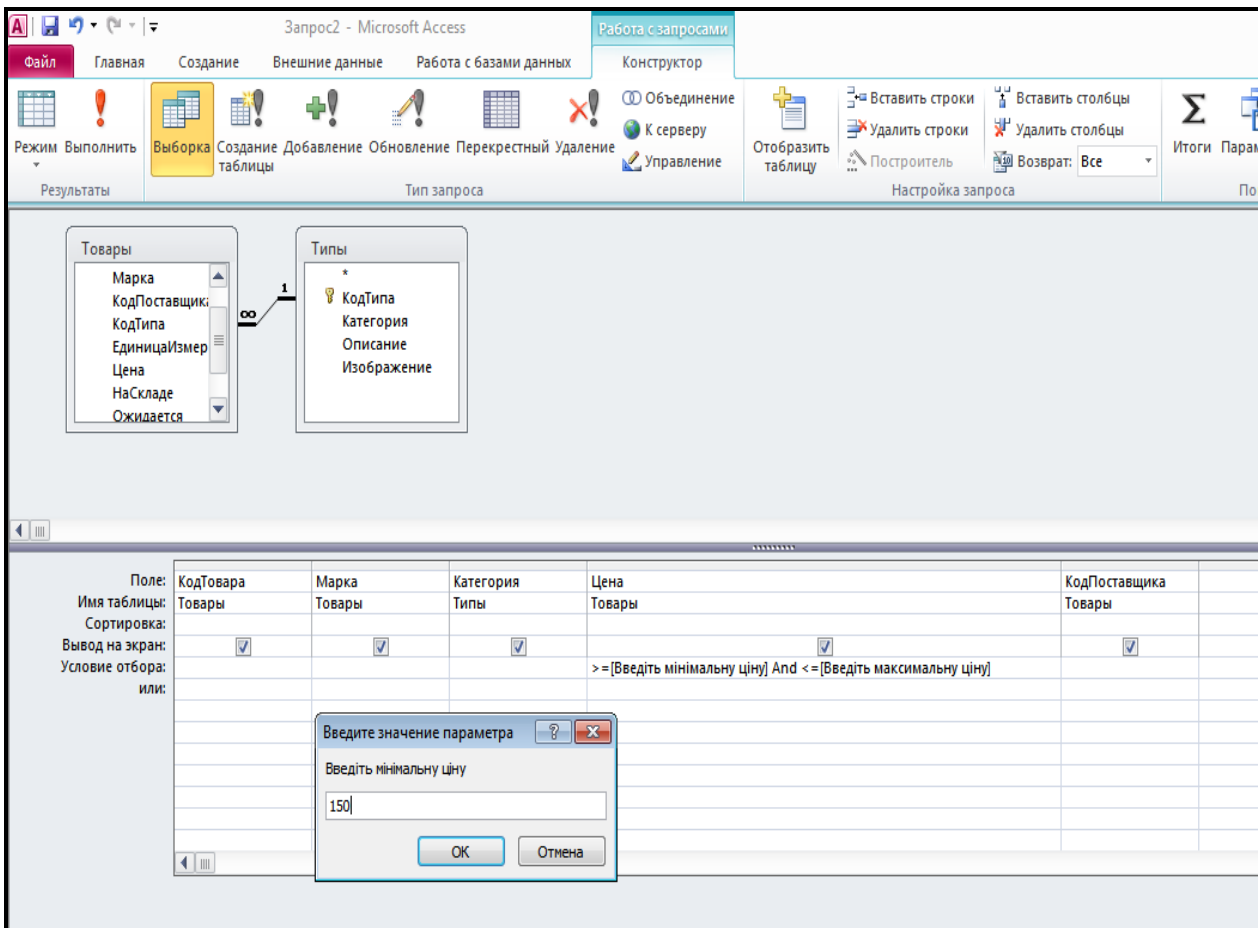


Рис. 12. Приклад виконання запити з параметром

Підсумкові запити створюються на основі запитів на вибірку, але містять угруповання даних, що дозволяє використовувати статистичні функції (категорія «Статистичні»). Підсумкові запити аналогічні проміжним підсумкам у MS Excel.

Список статистичних функцій наведено в табл. 4.


Таблиця 4

### Перелік вбудованих функцій категорії «Статистичні»

Функції	Призначення функції
Avg	Обчислює середнє арифметичне набору значень, що містяться в певному полі запиту
Count	Обчислює кількість записів, повернутих запитом
Min, Max	Повертають мінімальне або максимальне зі значень, що містяться в заданому полі запиту
Sum	Повертає суму значень, що містяться в заданому полі запиту
StDev	Повертає значення стандартного відхилення вибірки даних, що містяться в заданому полі запиту
Var	Повертають оцінку дисперсії вибірки значень, що містяться в заданому полі запиту
First, Last	Повертає значення з першої або останньої записи набору. Якщо записи не впорядковані, то повертаються значення, що виявляються випадковими

На відміну від *MS Excel*, у *MS Access* кількість рівнів угруповання необмежена. Зі створенням підсумкового запиту є можливість задавати умови відбору записів і створювати обчислювані поля, використовуючи будь-які функції *MS Access*.

Створення підсумкових запитів у *MS Access* можливе за допомогою майстра та в режимі конструктора.

Груповий запит у режимі конструктора створюють за допомогою кнопки «Підсумки»  на панелі інструментів. У результаті в бланк запиту додається рядок «Групова операція».

На рис. 13. наведено приклад групового запиту, що обчислює кількість товарів за кожною категорією.

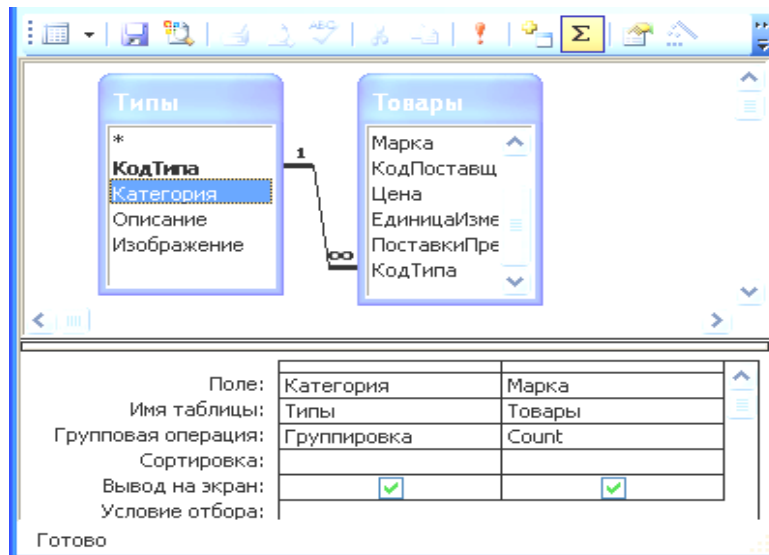


Рис. 13. Приклад группового запроса

На рис. 14 поданий результат выполнения группового запроса.

Категория	Count-Марка
Кондитерские изделия	14
Молочные изделия	10
Мясо/птица	7
Напитки	10
Приправы	11
Рыбопродукты	12
Фрукты	5
▶ Хлебобулочные изделия	8

Запись: 8 из 8

Рис. 14. Результат выполнения группового запроса, что обчислює кількість товарів кожної категорії

Перехресний запит є окремим видом підсумкового запиту (він аналогічний зведеній таблиці в *MS Excel*). У перехресному запиті угруповання здійснюється у двох полях. Створюють перехресний запит на основі запиту на вибірку за допомогою зміни типу запиту кнопкою «Перехресний» на панелі інструментів.

Наприклад, за допомогою перехресного запиту зручно визначати загальну кількість товарів кожної категорії, реалізованих за кожен рік. Результат виконання такого запиту поданий на рис. 15.

Категория	Итого	1996	1997	1998
Кондитерские изделия	7 687	1 154	3 338	3 195
Молочные изделия	6 840	1 274	3 284	2 282
Мясо/птица	5 248	909	3 128	1 211
Напитки	8 069	1 702	3 761	2 606
Приправы	7 997	1 753	4 072	2 172
Рыбопродукты	8 211	1 256	4 434	2 521
Фрукты	3 178	513	1 800	865
Хлебобулочные изделия	4 140	1 042	1 703	1 395

Запись: 8 из 8

**Рис. 15. Результат виконання перехресного запиту, що обчислює кількість товарів кожної категорії за роками**

Розглянемо основні функції категорії «Дата/Час».

1. Now () повертає поточну системну дату та час. Аргументів не має. Значення функції автоматично оновлюється.

2. Date () повертає поточну дату в числовому форматі або в одному з форматів, явно зазначеному користувачем у властивостях поля.

3. Year («number») повертає номер року в вигляді цілого числа (діапазон допустимих значень залежить від вибору системи літочислення (для Григоріанського календаря: від 2.08.718 до 31.12.9999). Наприклад, Year (1.09.18) = 2018. Тут і далі в якості аргументу «number» можна використовувати ім'я поля таблиці.

4. Month («number») повертає номер місяця в році, приймає значення від 1 до 12.

5. Day («number») повертає номер дня у місяці, приймає значення від 1 до 31.

Аналогічні функції Access існують і для більш дрібних інтервалів часу. Ознайомтеся з ними самостійно.

6. DateDiff («interval»; «date1»; «date2»; «firstweekday»; «firstweek») повертає різницю між двома датами (date2 – date1) у форматі, визначеному аргументом "interval". Найбільш часто використовують значення аргументу "interval" "y" і "d", які дозволяють визначити різницю між двома датами в днях. Результат буде негативним, якщо date1 > date2. Аргументи firstweekday та firstweek не є обов'язковими для визначення, тому що використовують деякі значення за замовчуванням. Firstweekday задає, який день тижня вважати першим (точніше «нульовим», для понеділка цей параметр = 2, тому що рахунок починається з суботи = 0). Значення firstweek задає,

який тиждень року вважати першим. За замовчуванням першим вважається тиждень, на який припадає 1 січня.

7. DatePart («interval»; «date»; «firstweekday»; «firstweek») повертає фрагмент дати, що відповідає значенням interval. Наприклад, DatePart ("ууу"; [ДатаРозміщення]) поверне значення 1996, 1997 або 1998, а DatePart ("m"; [ДатаРозміщення]) поверне значення від 1 до 12.

8. Format («expr»; «fmt»; «firstweekday»; «firstweek») також дозволяє отримати подібний результат. Аргумент expr задає вираз, який треба перетворити до формату, визначеному аргументом fmt. Слід пам'ятати, що функція формат перетворює дату до текстового типу; в цьому її суттєва відмінність від функції Format. Наприклад, Format (# 1.06.86 #; "mmm") поверне скорочену назву місяця, (скажімо, черв.).


9. If («expr»; «truepart»; «falsepart») перевіряє істинність логічного виразу expr і в разі його виконання повертає значення, що визначається аргументом truepart і, в іншому випадку, значення, що визначається аргументом falsepart. Якщо аргументи явно не визначені, функція поверне значення True і False відповідно (аналогічна функції ЕСЛИ в MS Excel).

10. CCur (expression) – функція перетворення формату, яка перетворює числове або рядкове вираження аргументу до грошового формату. Як аргумент може використовуватись ім'я поля, чий формат треба привести до грошового.

## Модифікуючі запити

MS Access пропонує такі типи модифікуючих запитів, що дозволяють внести якісь зміни у таблиці БД :

запит на створення нової таблиці  Созданиe таблицы...;

запит на видалення записів  Удаление;

запит на додавання записів у таблицю  Добавление...;

запит на оновлення значень полів у таблиці  Обновление.

### Запит на створення нової таблиці

Запит на створення нової таблиці зберігає результат запиту у вигляді нової таблиці. У результаті виконання такого запиту створюється нова таблиця з даними та структурою, взятими, можливо, з різних таблиць.

Завдання умов відбору даних для нової таблиці також можливе. Нова таблиця може бути, за вибором користувача, збережена як у поточній, так і в новій базі даних.

Збереження набору записів запиту використовується, наприклад, якщо необхідно довго зберігати підсумкову інформацію, а вихідні дані з БД, на яких базувався запит, будуть видалені, або дозволяє експортувати її в інші додатки. Наприклад, доцільно зберегти в окремій таблиці дані про відрахованих студентів у БД «Деканат» перед тим, як видалити інформацію про них з основних таблиць.

Для конструювання запиту на створення таблиці необхідно створити запит на вибірку або викликати в режимі конструктора раніше створений запит і змінити його тип на «Створення таблиці» (відповідна кнопка на панелі інструментів). У результаті з'явиться діалогове вікно «Створення таблиці», де необхідно ввести назву нової таблиці та клацнути на кнопці «ОК» (рис. 16).

Після цього запит необхідно виконати. З'явиться повідомлення про кількість рядків, що додаються в нову таблицю, та вимога підтвердити створення нової таблиці (рис. 17). Після натискання кнопки «Так» таблиця буде створена.

Створювана таблиця може містити також розрахункові поля.

Для того щоб побачити результат виконання запиту на створення нової таблиці, у вікні БД треба перейти до об'єктів «Таблиці» та переконатися, що у переліку таблиць з'явилася нова.

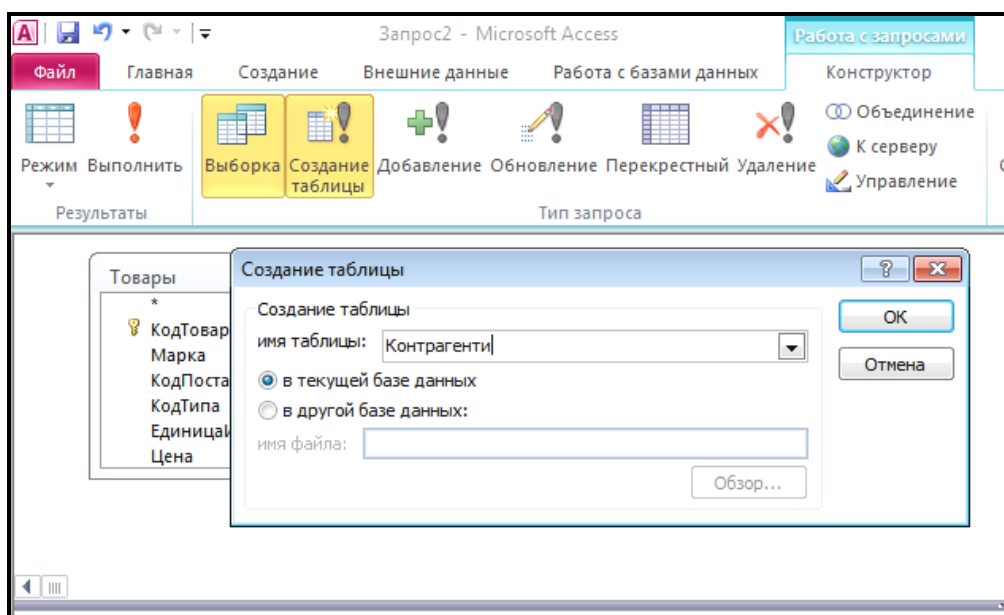


Рис. 16. Вікно «Створення таблиці» запиту на створення таблиці

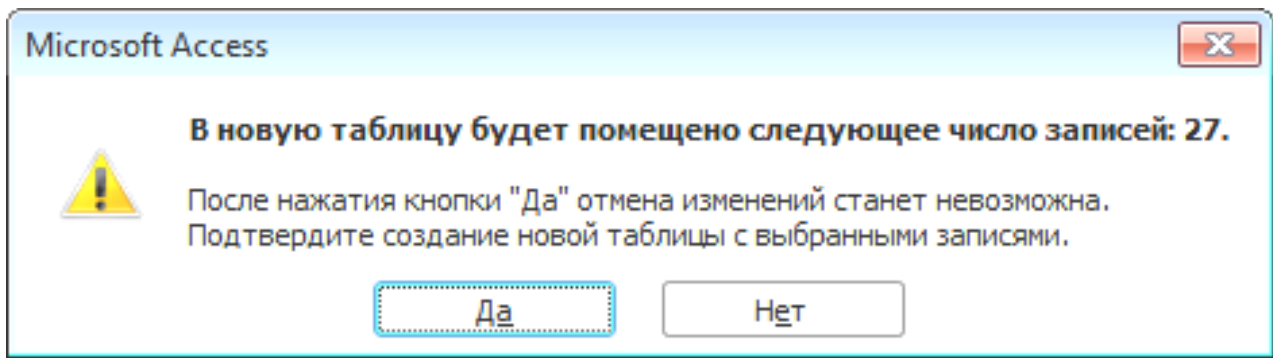


Рис. 17. Попередження про створення нової таблиці

Як ви думаєте, чи пов'язана нова таблиця з іншими таблицями у схемі даних? Перевірте своє припущення.

### Запит на оновлення полів

Запит на оновлення полів використовується для зміни значень груп записів, які відповідають заданим умовам відбору. Він дозволяє одночасно змінювати значення певних полів будь-якої кількості записів. Такий запит доцільно застосовувати, коли необхідно змінити велику кількість записів таблиці, які відповідають умовам вибору. Інакше записи можна змінювати безпосередньо в таблицях «вручну».

Наприклад, підприємство вирішило збільшити ціни на товари категорії «Рибопродукти» на 15 % у зв'язку зі збільшенням власних витрат. Інший приклад – потрібно підвищити ціну на всі вироби, що поставляються фірмами-постачальниками з кодами 132 і 151, на певну однакову кількість відсотків.

Для вирішення цього завдання спочатку створюється запит на вибірку, що відбирає записи з відомостями про товари заданих фірм. Потім вихідний запит перетворюється в запит на оновлення відповідною кнопкою з типом запиту. У нижній частині бланка запиту додається рядок «Оновлення», в якому користувач формує правила оновлення (наприклад, підвищення ціни на певний відсоток). Рекомендується за можливістю використовувати запит з параметром (рис. 18), що дозволяє змінювати умову підвищення цін.



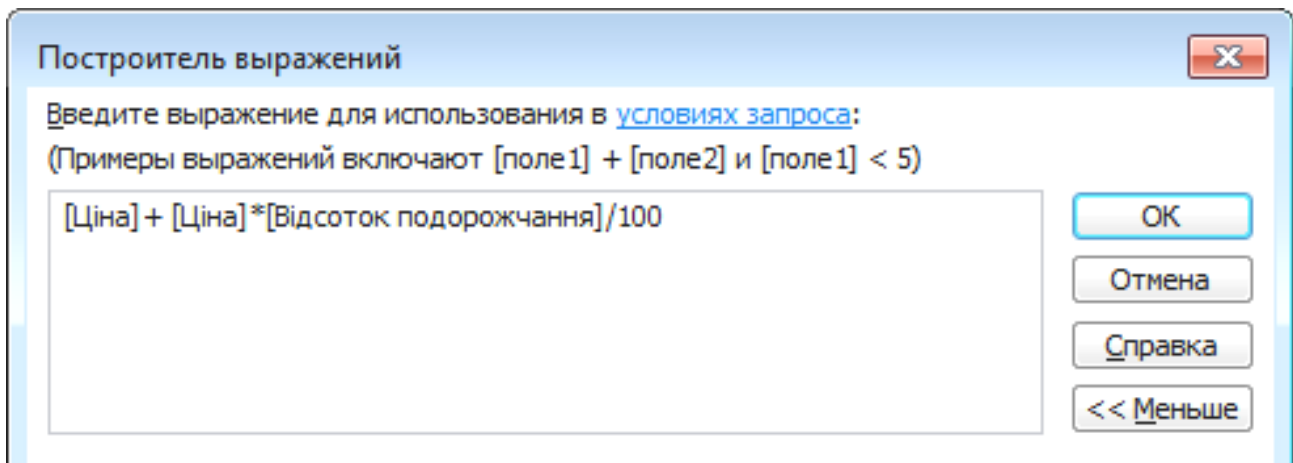


Рис. 18. Приклад формування виразу для підвищення ціни на певний відсоток у вікні Будівника виразів

У результаті виконання запиту на оновлення ціна для всіх записів, що підходять за умовами відбору, буде збільшена на певний відсоток, який введе користувач. Якщо запит буде виконаний повторно, то ціна збільшиться знову порівняно з її попереднім значенням.

Запит на оновлення не буде виконаний, якщо спроба його виконання може призвести до таких помилок:

- дублювання значень ключового поля;

- помилки перетворення даних (наприклад, у спробі додати до числових значень текстові);

- зміна записів, які в цей час оновлюються іншими користувачами та заблоковані ними;

- порушення умов на значення.

### Запит на додавання записів

Запит на додавання записів забезпечує додавання одного або декількох записів, що відповідають умовам відбору, з однієї таблиці в кінець однієї або декількох таблиць. Створюється на основі запиту на вибірку шляхом зміни типу запиту на «Додавання» (див. рис. 16) у режимі конструктора.

У результаті виконання такого запиту в існуючу таблицю додаються записи з іншої таблиці, що має подібну структуру.

## Запит на видалення записів

Запит на видалення записів дозволяє видаляти групу записів з однієї або декількох пов'язаних таблиць. Такий запит доцільно застосовувати, коли необхідно видалити велику кількість записів, які відповідають одній умові. Інакше, записи можна видалити безпосередньо з таблиці «вручну». Це створюється на підставі запиту на вибірку шляхом зміни типу запиту на «Видалення» (див. рис. 16) у режимі конструктора.

Якщо в схемі даних бланка запиту на видалення відкрито більше однієї таблиці, то необхідно вказати, з якої саме таблиці видаляти записи. Це здійснюють шляхом перетягування символу «\*» (що означає «всі поля») з переліку полів таблиці в порожній стовпець рядка «Видалення» бланка запиту. Якщо дана таблиця пов'язана з іншими таблицями і є для них головною та включено властивість «Каскадне видалення пов'язаних записів», то відповідні записи будуть видалені з усіх пов'язаних таблиць.

Наприклад, необхідно видалити записи із заданим номером анульованого замовлення (якщо такого запису немає, можна ввести 0 – запису з таким номером замовлення напевно не існує) та записи із замовленнями більш ніж річної давності (припустимо, відомості про замовлення зберігають тільки один рік). Бланк запиту на видалення зображений на рис. 19.

Поле:	Заказы.*	Номер заказа	Дата
Имя таблицы:	Заказы	Заказы	Заказы
Удаление:	Из	Условие	Условие
Условие отбора:	или: [Введите номер аннулированного заказа или 0] <Date()-365		

Рис. 19. Приклад запиту на видалення

Після натискання кнопки «Виконати» зазвичай з'являється діалогове вікно для підтвердження видалення. У ньому вказується кількість записів, що будуть видалені у разі натискання кнопки «Так» (рис. 20). Поновити записи після видалення неможливо, тому доцільно створювати заздалегідь резервну копію БД.

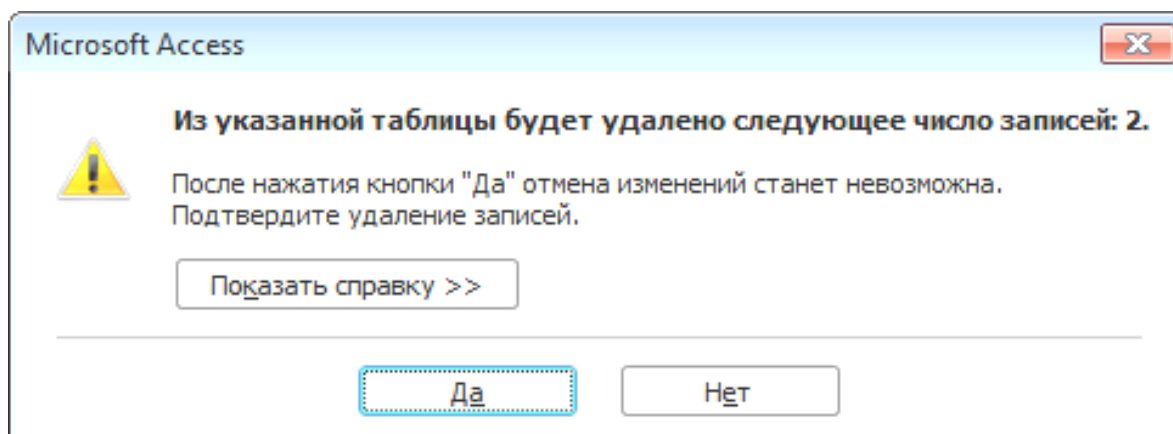


Рис. 20. Діалогове вікно видалення записів

Переконайтеся у виконанні запиту можна шляхом перегляду кількості записів у таблиці після виконання запиту (вона повинна зменшитися). Як ви думаєте, чи призведе до успіху спроба повторно виконати цей запит?

### Хід роботи

*Виконання завдань, помічених «\*», надає право на отримання оцінки «відмінно».*

1. Виведіть список клієнтів із зазначенням номера замовлення («КодЗаказу»), які розмістили свої замовлення в квітні 1998 р.

2. Виведіть прізвища співробітників і список замовлень (КодЗаказу), які ще не виконані (поле «ДатаВиконання» для таких замовлень не заповнене).

3. Виведіть список клієнтів з Норвегії, які замовили будь-які напої (поле «Категория» з таблиці «Типи») у постачальників з Австралії.

4. Виведіть список усіх постачальників, країна яких починається на букву Н, а мінімальний запас товарів, що поставляються ними, більше нуля.

5. Виведіть список марок товарів, кількість яких на складі виражається тризначним числом.

6. Виведіть список клієнтів, назва яких починається літерами А, В, С (англійські символи). Виконати завдання двома способами: спочатку використовуючи логічну функцію OR, а надалі – шаблон Like "[A-C] \*\*"

7. \*Виведіть список клієнтів, у номері телефону яких відсутній код (тобто номер телефону починається не на дужку).

8. Виведіть список замовлень, розміщених влітку 1997 року, в які увійшли товари категорії Напої з 10-відсотковою знижкою.

Для конструювання умови фільтрації на полі «ДатаРозміщення», доцільно скористатися оператором Between 1.01.97 And 31.01.97. Можна також застосувати вираз >= 1.01.97 And <= 31.01.97.

9. Визначте вартість поставок товарів з різних країн («Марка», «Ціна», «Очікується» – з таблиці «Товари»; «Країна» – з таблиці «Постачальники»). Урахувати тільки ті товари, щодо яких поставки очікуються (значення поля «Очікується» більше 0). Визначити найдорожчу поставку.

Вартість поставки розрахувати як [Ціна] \* [Очікується].

10. Визначте вартість кожного товару на складі.

11. \*Розрахуйте вік і стаж роботи кожного співробітника фірми (результат округлити до цілих). Яку посаду обіймає наймолодший та найстарший співробітник фірми?

12. Виведіть список товарів із зазначенням ціни, які належать до категорії, що задається користувачем (параметр по полю «Категорія»).

13. Виведіть список замовлень із зазначенням коду замовлення, країни клієнта, прізвища співробітника з обслуговування та способу доставки, причому країна клієнта та спосіб доставки задаються користувачем.

14. Виведіть список співробітників і назви клієнтів, для яких замовлення було виконано в призначений день.

15. Виведіть список кодів замовлень, оформлених будь-яким співробітником (параметр) у проміжок часу, що задається користувачем.

16. \*Розрахуйте вартість замовлених товарів в у.о. Використовуйте параметр для введення поточного курсу долара із розрахунком вартості.

17. Розрахуйте кількість клієнтів з кожної країни.

18. Розрахуйте сумарну вартість кожного замовлення.

19. Для кожного клієнта обчисліть сумарну вартість, сплачену за всі його замовлення. Упорядкувати список клієнтів у порядку спадання вартості замовлень.

20. Визначте максимальну знижку, отриману кожним клієнтом.

21. \*Для кожного співробітника розрахуйте суму забезпеченої ним виручки від продажу товарів у ту чи іншу країну (країна отримувача – параметр).

22. \*Розрахуйте загальну суму виручки від реалізації всіх товарів за місяцями 1997 року. Щоб виконати угруповання за місяцями, використовуйте або функцію Month ([ДатаРозміщення]) – у цьому випадку місяці будуть виводитися як число, або функцію Format для виведення назв місяців. Щоб місяць виводився повністю, треба задати формат "mmmm".

23. Для кожного клієнта визначте першу й останню дати звернення у фірму та кількість замовлень, розміщених протягом усього часу.

24. Визначте середню вартість доставки в кожну країну. Країни клієнтів – заголовки рядків; спосіб доставки – заголовки стовпців.

25. Для кожного співробітника розрахуйте вартість оформлених ним замовлень за місяцями 1997 року. Спочатку слід розрахувати вартість проданих товарів.

26. За допомогою перехресного запиту виведіть суму продажів товарів кожної категорії, проданих у кожному кварталі 1997 року.

27. Для кожного клієнта визначте найдорожче замовлення за кожен рік.

28. За допомогою запиту на створення таблиці виділіть замовлення клієнтів зі США в окрему таблицю. Спочатку створіть відповідний запит на вибірку на основі таблиць «Замовлення» та «Клієнти».

29. За допомогою запиту на додавання додайте в створену таблицю «Замовлення клієнтів з США» також клієнтів з Бразилії, Мексики та Венесуели. Назвіть її «Замовлення з Америки».

30. Видаліть замовлення за 1996 рік і пов'язані з ними записи таблиці «Замовлено».

31. Установіть знижку 25 % у разі замовлення, кількість якого для одного товару становить не менше 50-ти одиниць (таблиця «Замовлено»).

32. \*За допомогою запиту на оновлення оновіть дати розміщення, призначення та виконання замовлень – «збільшіть» їх на 10 років. Використовуйте функцію DateAdd для цих полів у запиті на оновлення.

33. Самостійно сформулюйте умови та виконайте запит на створення нової таблиці для індивідуальної БД.

34. Самостійно сформулюйте умови та виконайте два запити на оновлення для індивідуальної БД.

35. Самостійно сформулюйте умови та виконайте два запити на видалення записів для індивідуальної БД, причому один із запитів повинен видаляти записи одночасно з кількох пов'язаних таблиць.

36. Оформіть звіт і захистіть його перед викладачем.

### Контрольні запитання

1. Визначте призначення запитів на вибірку.

2. Що таке запит з параметром? Які переваги надає його застосування?

3. Назвіть відомі вам засоби в *MS Excel*, які відіграють подібну ролі до запитів на вибірку. Порівняйте їх можливості.

4. Які засоби відбору записів, крім запитів на вибірку, існують у *MS Access*? Порівняйте можливості відбору записів безпосередньо в таблицях і за допомогою запитів.

5. Охарактеризуйте обчислюване поле. Чи можна створити обчислюване поле безпосередньо в таблиці? Обґрунтуйте свій висновок.

6. Опишіть послідовність дій для створення запиту на вибірку в режимі конструктора.

7. Які способи створення запитів існують у *MS Access*?

8. Яке призначення групових запитів? Порівняйте їх із перехресними запитами.

9. Наведіть приклад завдання, яке доцільно розв'язати за допомогою перехресного запиту.

10. У чому полягає специфіка запитів на видалення порівняно з іншими модифікуючими запитами?

11. Як впливає властивість зв'язку «Каскадне видалення пов'язаних записів» на результати виконання такого запиту?

12. Як впливає властивість зв'язку «Каскадне оновлення видалення пов'язаних полів» на результати виконання такого запиту?

13. У яких випадках запит на видалення може призвести до помилки?

14. Як перевірити результати виконання запиту на оновлення?

15. Наведіть приклади, коли доцільно створювати запит на оновлення.

## **Лабораторна робота 10**

### **Створення форм. Налаштування елементів форми**

**Мета** – набути навички створення та використання форм у середовищі *MS Access*.

Лабораторна робота сприяє напрацюванню таких компетентностей відповідно до Національної рамки кваліфікації:

**знання:**

поняття та призначення форм;

основні способи створення форм;

структура форм і призначення її розділів;

різновиди керівних елементів для форми та їх властивості і параметри;

засоби обчислення значень і підсумків за групами записів на формі;

інструменти фільтрації та сортування даних на формі;

**уміння:**

обирати керівні елементи форми та налаштовувати їх параметри;

додавати, коректувати, видаляти дані таблиць за допомогою форм;

виконувати сортування та фільтрацію у формах;

засвоїти елементарні знання з проектування та модифікації форм у режимі конструктора;

розробляти кнопкові форми;

**комунікації:**

презентувати результати розв'язання прикладних завдань за допомогою форм;

надавати рекомендації співробітникам щодо вибору способів оформлення форм;

**автономність і відповідальність:**

самостійно приймати рішення та відповідати за коректність і адекватність використаного інструментарію;

самостійне прийняття рішення про доцільність подання даних у вигляді форм та їх оброблення і візуалізацію.

### **Основні положення**

За допомогою форми можна переглядати та коригувати дані, додавати та видаляти записи в таблицях, змінювати значення даних у полях, отримувати розрахункові дані, а також здійснювати контроль даних, що вводяться.

Конструкція форми визначається користувачем. Для цього вказується, з якої таблиці (запиту) та які поля потрібно відображувати, які елементи повинні бути включені в форму. Для створення форми потрібно відкрити вкладку «Форми» вікна бази даних і натиснути кнопку «Створити».

Форма – це інтерактивний бланк, через який може здійснюватися первісне введення даних у таблиці бази даних, перегляд даних, а також їх коригування. Працюючи з формою, користувач має можливість додавати і видаляти записи в таблицях, змінювати значення полів, отримувати розрахункові дані. У процесі роботи може здійснюватися контроль введених даних, перевірка обмежень на доступ до них.

Кнопкові форми можуть бути створені за допомогою майстра кнопкових форм або в режимі конструктора. В останньому випадку розробник має найширші можливості для створення форми будь-якої складності. Можливий також випадок, коли частина форми створюється за допомогою майстра форми, а частина, що залишилася, – в режимі конструктор.

Основні способи створення форм у MS Access (рис. 21): автоформа; конструктор форм; майстер форм; порожня форма.

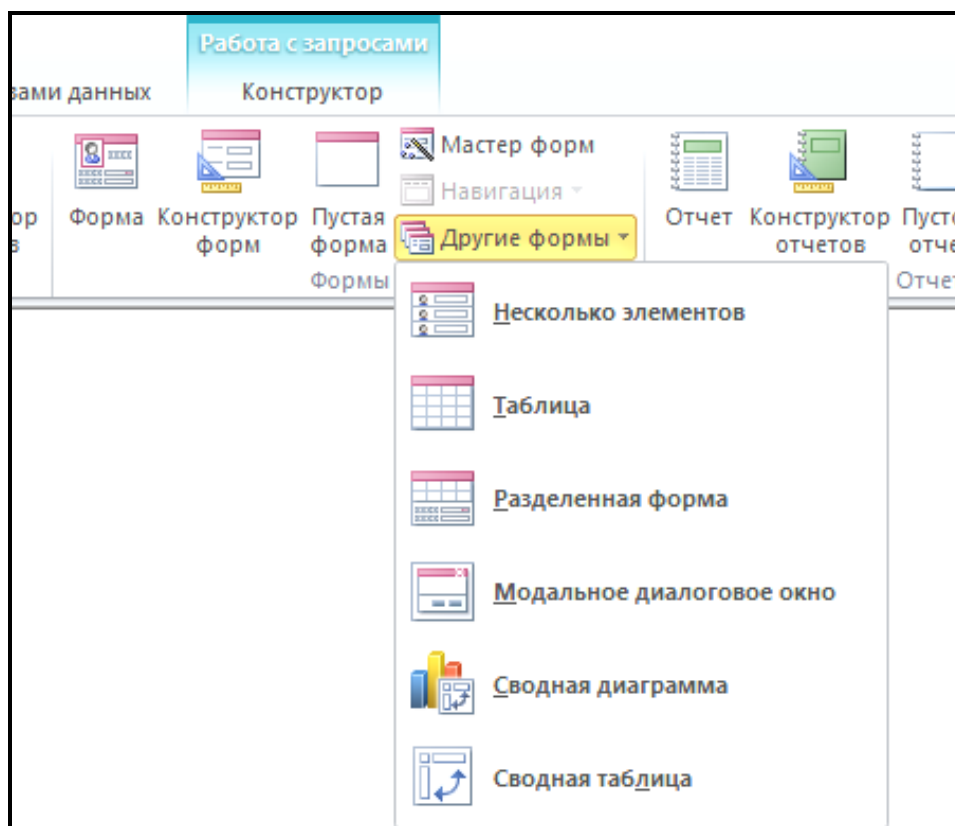


Рис. 21. Способи створення форм



У режимі конструктора форм можна виконати будь-які зміни створеної форми, зміни порядку розташування керівних елементів, розташування полів і їх властивостей, форматування полів, оформлення елементів тощо.

Майстер форм дозволяє обирати поля з однієї або декількох таблиць або запитів, визначати угруповання та сортування даних.

Інструмент «Порожня форма» створює нову порожню форму, яку можна налаштувати та відформатувати на свій розсуд.

У *MS Access* на формі можна розмістити такі типи елементів оформлення та управління: поле, напис, група перемикачів, вимикач, перемикач, прапорець, поле зі списком, список, кнопка, рисунок, вільна рамка об'єкта, приєднана рамка об'єкта, лінія, прямокутник тощо.

Призначення основних елементів форми:

напис – створення текстів, постійних заголовків, приміток;

поле – встановлення зв'язку з полем таблиці або запиту, а також створення обчислюваних полів;

група перемикачів – розміщення набору прапорців, перемикачів або вимикачів;

перемикач – вибір альтернативних значень;

прапорець – вибір набору з можливих значень;

поле зі списком – об'єднання поля та списку значень, які можуть вводитися як безпосередньо в полі, так і вибором зі списку;

список – створення постійно відображуваного списку значень, які є єдиним джерелом введення даних у поле;

кнопка – створення командної кнопки, за допомогою якої може бути виконаний перехід по записах, відкрита форма, надрукований звіт та інші функції;

рисунок – відображення малюнка, який не підлягає редагуванню;

вкладка – створення вкладок у формі, на кожній з яких можуть розміщуватися свої елементи управління;

підпорядкована форма – висновок даних з таблиць, пов'язаних з таблицею – джерелом основної частини форми;

лінія – розмежування розділів в формі.

Для кожного елемента форми можна змінювати його властивості.

Перелік властивостей залежить від типу обраного елемента форми. На рис. 22 наведено приклад вікна властивостей елемента «поле» форми.

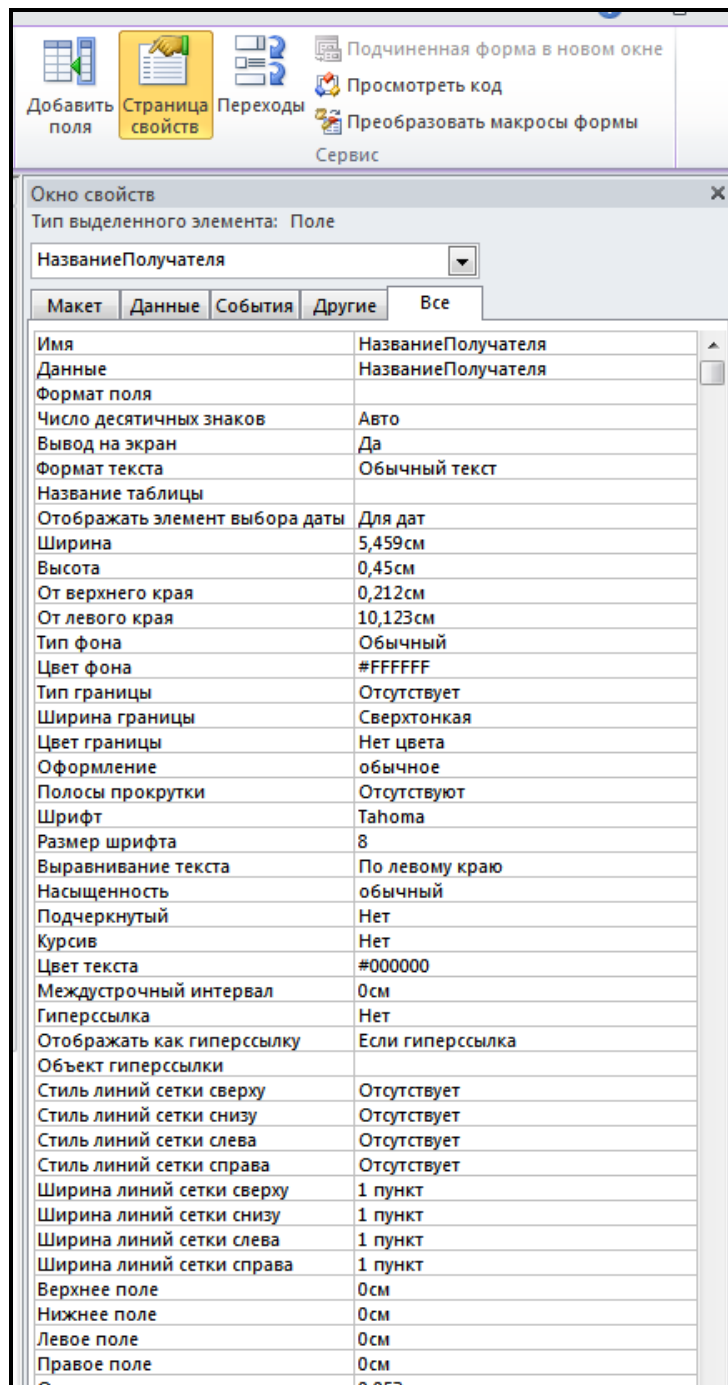


Рис. 22. Пример вікна властивостей поля форми

Розглянемо команду контекстного меню «Властивості» докладніше. З вибором команди для елемента «Поле» (в який вводиться інформація) або «Напис» (який містить назву, що відображається в формі), з'являється діалогове вікно властивостей елемента.

Рекомендується змінити деякі з властивостей.

На вкладці «Макет»: число десяткових знаків, виведення на екран, режим виведення, смуги прокрутки, геометричні розміри та властивості форматування (колір, курсив і т. д.).

На вкладці «Дані»: дані (як визначається значення поля), доступ (можливість зміни значень поля), маска введення, значення за замовчуванням. Щоб користувач випадково не змінив дані, які не підлягають зміні (наприклад, поле зв'язку «Код» у підпорядкованій таблиці), необхідно встановити такі властивості:

«Доступ» – «Ні» (курсор не встановлюється в поле);

«Блокування» – «Так» (неможливо змінити значення поля, навіть якщо доступ до нього відкритий);

на вкладці «Події»: імена процедур (програм на мові Visual Basic, які будуть викликані для оброблення відповідних подій, наприклад натискання кнопки миші);

на вкладці «Інші»: рядок стану, підказку;

на вкладці «Все» можна переглянути всі властивості елемента.

Форма може містити розділи «Заголовок форми», «Область даних» і «Примітка форми», причому заголовок і примітка форми можуть не виводитися на екран (рис. 23).



Рис. 23. Структура форми

З додаванням у форму елементів управління, пов'язаних з полями даних таблиць або запитів, рекомендується в режимі конструктора відкрити список полів відповідною кнопкою «Список полів» на панелі інструментів «Конструктор форм» і зробити буксирування потрібного поля в певне місце форми.

Додавання на форму незв'язаних елементів управління проводиться за допомогою панелі інструментів «Панель елементів». Для цього слід клацнути на кнопці обраного елемента на «Панелі елементів», а потім клацнути в місці розташування верхнього лівого кута цього елемента в формі.

Для введення обчислюваного виразу в полі викличте в його контекстному меню команду «Властивості – Дані» та в рядку «Дані» клацніть на кнопці «Побудувати» для введення формули за допомогою Будівника виразів або введіть формулу «вручну» (рис. 24).

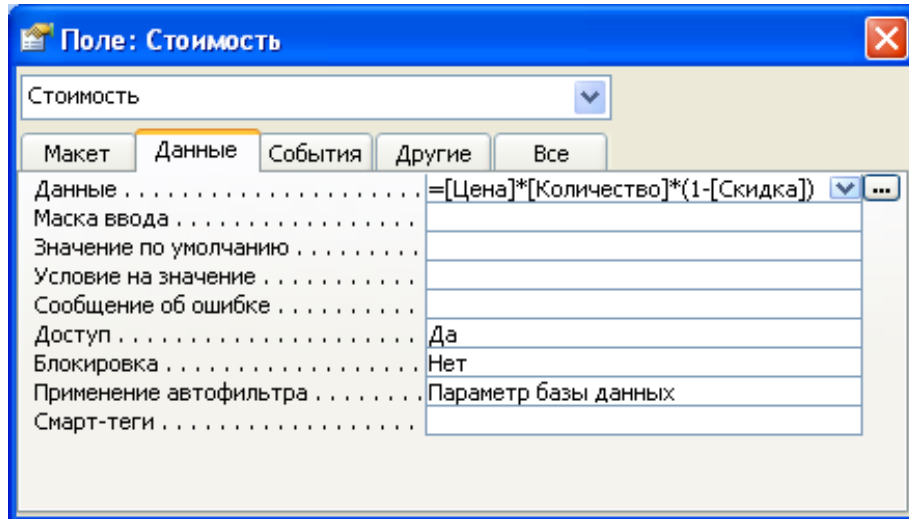


Рис. 24. Обчислення значення поля за формулою на формі

Аналогічно в форму можуть бути додані інші елементи управління (наприклад, прапорці, перемикачі, списки, кнопки, рисунки).

До виділеного елемента форми можна застосувати такі дії: зміну розміру (перетягуванням маркерів), форматування за допомогою панелі «Формат (форма/звіт)», зміну положення об'єкта (перетягуванням за лівий верхній маркер).

Контекстне меню кожного об'єкта містить команди, які стосуються стану, оформлення об'єкта, черговості переходу від об'єкта до об'єкта (від поля до поля), подій, пов'язаних з використанням об'єкта та ін.

### Хід роботи

1. Створіть просту автоформу на основі таблиці «Заказано».

Створіть ще одну, складну автоформу, що містить вбудовану підпорядковану форму, попередньо виділивши таблицю, що має бути джерелом даних. Поясніть у звіті, у якому випадку ви отримуєте просту, а в якому – складну автоформу.

Порівняйте отриману автоформу з формою *Excel* за можливостями і зовнішнім виглядом.

2. Спробуйте:

переміщення на записах (використовуючи кнопки внизу форми);  
введення записів у таблицю;  
видалення записів (наприклад, кнопкою «Видалити запис»);  
зміну значень полів.

3. Виконайте фільтрацію даних у формі. Створіть фільтр з однієї умови; з двох умов, що виконуються одночасно; з трьох умов, пов'язаних відношенням «Або».

4. Якщо розділи «Заголовок форми» та «Примітки форми» не відображаються, то підключіть їх через контекстне меню. Установіть (орієнтуючись по горизонтальній і вертикальній лінійці) такі розміри для розділів форми:  
ширина – 10 см;  
висота області заголовка – 0 см;  
висота області даних – 4 см;  
висота області приміток – 2 см.

5. В область приміток додайте з панелі елементів вільний елемент управління – поле. Додавання елемента з панелі елементів управління на форму виконується таким чином: мишкою виберіть відповідний елемент на панелі елементів, а потім клацніть лівою кнопкою мишки в обраному місці форми – це координата лівого верхнього кута поля, що додається.

В якості даних для створеного вільного поля створіть вираз, що обчислюється за формулою (через вкладку Дані властивостей поля).

6. Змініть шрифт і фон форми, додайте в область заголовка логотип і назву форми. Виділіть іншим накресленням, кольором і рамкою підсумкові значення на формі.

7. Застосуйте фільтр. Проаналізуйте результат і скасуйте фільтрацію кнопкою «Видалити фільтр».

8. Створіть кнопку форму для індивідуальної бази даних, самостійно оберіть її оформлення, фоновий рисунок, назву, логотип, розмір кнопок та рисунки (написи) на них. Кожна кнопка повинна активувати запит або іншу форму БД. Додайте кнопку для закінчення роботи з БД.

9. Оформіть звіт і захистіть його.

### **Контрольні запитання**

1. Яка сфера застосування різних способів створення форм?
2. Для чого можна застосовувати форми?
3. Як створити складну форму, що містить підпорядковану форму?

4. Як змінити оформлення форми?
5. Для чого використовують кнопочві форми?
6. Як задати обчислення за групами записів на формі?

## **Лабораторна робота 11**

### **Проектування звітів**

**Мета** – навчитися створювати та редагувати звіти на основі запитів, групувати дані в звітах, обчислювати підсумки.

Лабораторна робота сприяє напрацюванню таких компетентностей відповідно до Національної рамки кваліфікації:

**знання:**

- структури та призначення звітів;
- різновидів звітів;
- інструментів для створення звітів;

**уміння:**

- створювати звіти різного типу;
- задавати умови фільтрації, сортування для даних звітів;
- групувати дані;
- обчислювати підсумки за групами записів звіту;

**комунікації:**

- презентація структури й оформлення звітів;

**автономність і відповідальність:**

самостійно приймати рішення щодо переліку необхідних звітів та їх оформлення;

відповідати за коректність та адекватність розроблених звітів і періодичність їх створення.

### **Основні положення**

Звіт – це гнучкий засіб для організації даних з метою їх виведення на друкування.

Створення звітів зазвичай виконують *основними способами*:

- в режимі конструктора;
- за допомогою автозвіта;
- за допомогою майстра.

Звіти можуть бути відкриті в таких *режимах*:

у режимі конструктора, який використовують для створення нових звітів або зміни структури існуючих;

у режимі попереднього перегляду, який дозволяє переглядати дані в тому вигляді, в якому вони будуть розміщуватися на друкованих сторінках;

у режимі макета;

у режимі подання зразка, в якому виводяться основні елементи макета звіту зі зразками даних, що демонструють подання даних у звіті (наявність поділу на сторінки, колонтитулів та ін.).

Перемикання між режимами звіту проводиться з використанням кнопки <Режим> у лівому верхньому кутку на панелі інструментів.

Найпростішим є створення автозвіту. Користувачу достатньо вказати запит чи таблицю у якості джерела даних і натиснути кнопку <Звіт>. На рис. 25 наведено приклад автозвіту.

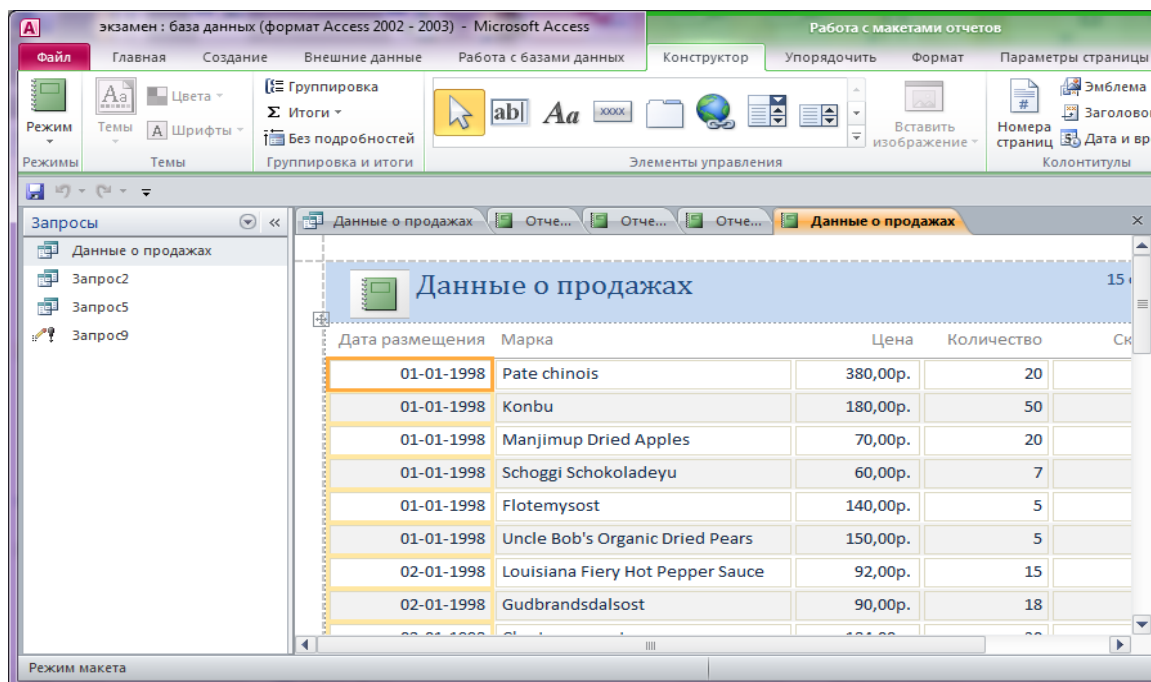


Рис. 25. Приклад автозвіту з джерелом даних – запит

Звіт може містити такі розділи:

область даних;

верхній та нижній колонтитули;

заголовок звіту;

примітки звіту.

Якщо дані звіту згруповано з метою сортування або обчислення підсумків, то він має містити заголовок і (або) примітку групи (рис. 26).

Для побудови виразів, що обчислюються за формулами, доцільно використовувати спеціалізований «Будівник виразів», приклад використання якого наведено на рис. 27.

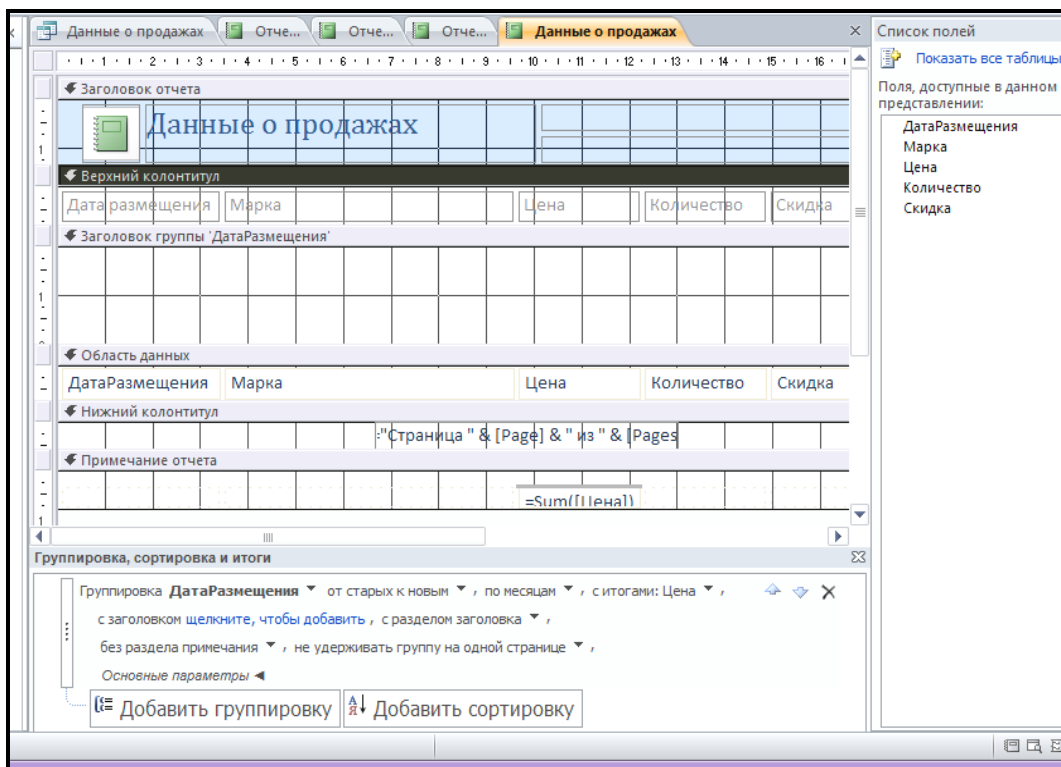


Рис. 26. Пример структуры отчета в режиме конструктора

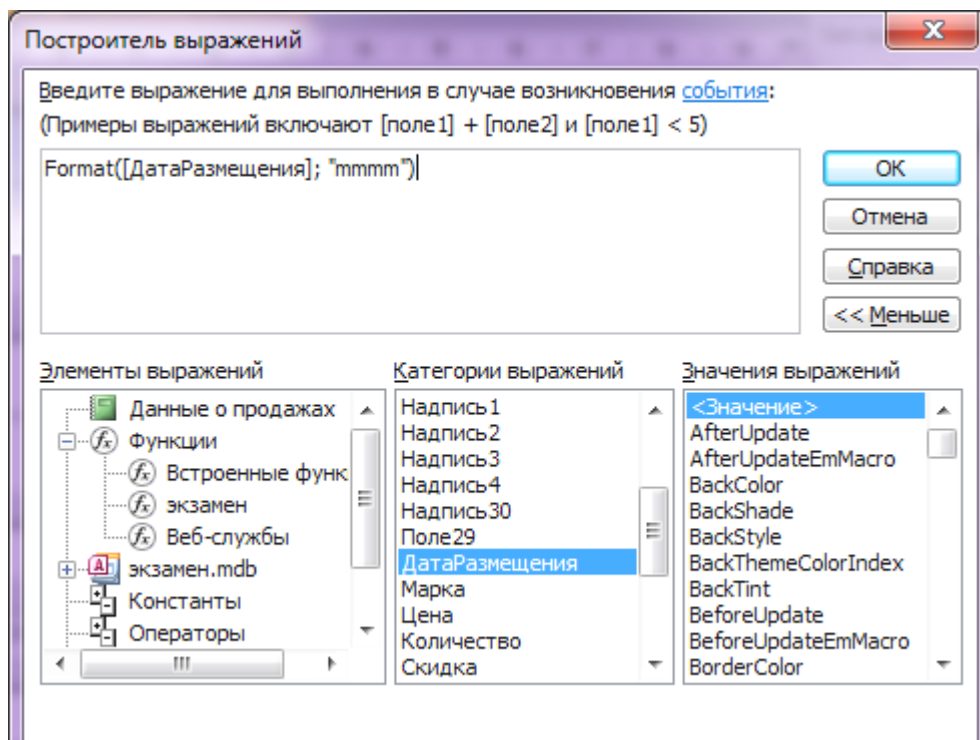


Рис. 27. Пример Будівника виразів



Приклад звіту з даними про продажі у певному році у режимі попереднього перегляду наведено на рис. 28.

Данные о продажах за 1998 год				15 февраля 2017 г.	
Марка	Количество	Сумма			
Январь					
Alice Mutton					
Alice Mutton	100,00р.	30	0%		
Alice Mutton	100,00р.	49	0%		
		79		7900	
Boston Crab Meat					
Boston Crab Meat	96,50р.	12	0%		
Boston Crab Meat	96,50р.	20	0%		
Boston Crab Meat	96,50р.	20	0%		
		52		5018	
Camembert Pierrot					
Camembert Pierrot	285,00р.	30	15%		
		30		7267,489948038	
Carnarvon Tigers					
Carnarvon Tigers	220,00р.	21	0%		
		21		4620	
Chai					
Chai	332,50р.	3	0%		
Chai	332,50р.	30	0%		
Chai	332,50р.	30	25%		
Chai	332,50р.	48	0%		

Страница 1 из 58

Рис. 28. Приклад звіту з даними про продажі товарів

Звіти можна не тільки друкувати, а і пересилати електронною поштою.

### Хід роботи

1. Побудуйте звіт про доставки товарів у кожну країну різними способами доставки. Для цього введіть у звіті назви країн клієнтів і розрахуйте кількість доставок у кожну країну за способами доставки.

2. Побудуйте звіт про роботу співробітників за кварталами одного року: для кожного співробітника необхідно розрахувати кількість оформлених замовлень і їх вартість. Додайте в заголовок звіту довільний рисунок.

3. Відформатуйте звіт – змініть ступінь точності полів там, де це необхідно; приберіть непотрібні поля; додайте логотип; змініть розмір та розміщення елементів, параметри шрифтів тощо з метою покращення його читаності та привабливості.

4. Акцентуйте увагу користувача на розрахункові дані та підсумки.

5. Роздрукуйте створені звіти.

### **Контрольні запитання**

1. Які способи формування звітів ви знаєте?
2. На основі яких об'єктів формуються звіти?
3. В яких режимах можна працювати зі звітами?
4. Для чого слугує режим попереднього перегляду звіту?
5. Як обчислити підсумки за групами записів у звіті?
6. Як задати умови фільтрації для даних звіту?
7. Від яких параметрів залежить читаність та привабливість звіту?

## **Лабораторна робота 12**

### **Створення макросів**

**Мета** – освоїти принципи створення макросів.

Лабораторна робота сприяє напрацюванню таких компетентностей відповідно до Національної рамки кваліфікації:

**знання:**

- понять макросу та макрокоманди;
- різновидів макросів;
- інструментів для створення макросів;

**уміння:**

- створювати групи макросів;
- задавати умови розгалуження для макрокоманд;
- налагоджувати роботу макросів;

аналізувати доцільність сформульованих обмежень;  
розширювати можливості БД за допомогою макросів;  
**комунікації:**

презентація результатів роботи макросів;

**автономність і відповідальність:**

самостійно приймати рішення щодо складу макросів, необхідних для оптимізації роботи БД;

відповідати за коректність та адекватність розроблених макросів.

### Основні положення

Макрос – це певна послідовність операцій, за допомогою якої можна визначати реакцію програми на різні події в формах і звітах, такі, як: натискання кнопки, зміна даних, відкриття та закриття форм і звітів тощо.

Макрос у *Microsoft Access* є структурою, що складається з однієї або декількох макрокоманд, які виконуються послідовно або в порядку, заданому певними умовами. Кожна макрокоманда має ім'я та, можливо, один або кілька аргументів, які задаються користувачем. Наприклад, для використання макрокоманди «ВідкритиФорму» в якості аргументів необхідно задати хоча б ім'я форми, що відкривається, та режим виведення її на екран.

Макроси дозволяють автоматизувати оброблення БД без застосування програмування на мові Visual Basic for Application (VBA). У результаті можна отримати додаток з досить великими функціональними можливостями та зручним інтерфейсом. Однак використання макросів має деякі недоліки:

можливості макрокоманд обмежені порівняно з можливостями мови VBA;

процедури VBA виконуються, як правило, швидше макросів, тому їх краще використовувати для створення багатокористувацьких додатків, в яких швидкість роботи програми критична;

макроси, на відміну від процедур VBA, це об'єкти, що існують окремо від форм і звітів. Тому, коли цих об'єктів стає багато, їх підтримка ускладнюється.

В однокористувацьких додатках використання макросів істотно спрощує та прискорює розроблення.

Для створення макросу використовують відповідну кнопку «Макрос» на панелі інструментів, що відкриває вікно конструктора макросів, розділене по горизонталі на дві частини – панель описів і панель аргументів.

Панель описів за замовчуванням складається з двох стовпців – «Макрокоманда» і «Примітка». Вона дозволяє визначити послідовність макрокоманд, якщо вибрати їх зі списку (рис. 29).

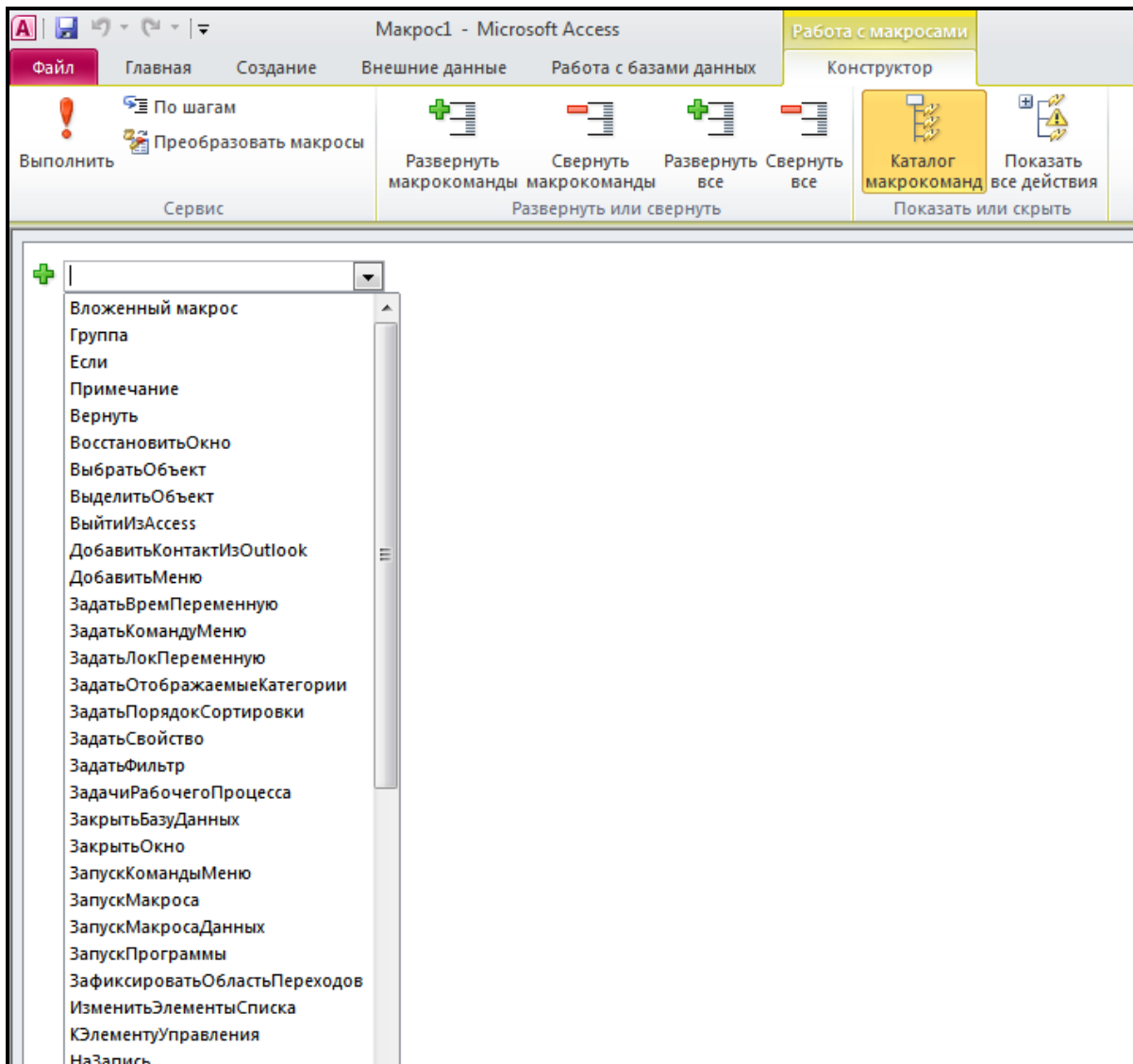


Рис. 29. Перелік вбудованих макрокоманд

За бажанням користувача в стовпці Примітка можна ввести коментарі, які описують їх дії.

Коли поле «Макрокоманда» заповнене, у нижній частині вікна Конструктора макросів з'являється панель аргументів, список полів якої залежить від обраної дії.

Можна також створити макрокоманду, що відкриває потрібний об'єкт БД, відобразивши одночасно на екрані вікно Конструктора макросів і вікна БД шляхом буксирування мишкою потрібного об'єкта (наприклад, форми) з вікна БД у поле «Макрокоманда» вікна Конструктора макросу або вибравши її з переліку. На рис. 30 наведені аргументи макрокоманди «ВідкритиФорму».

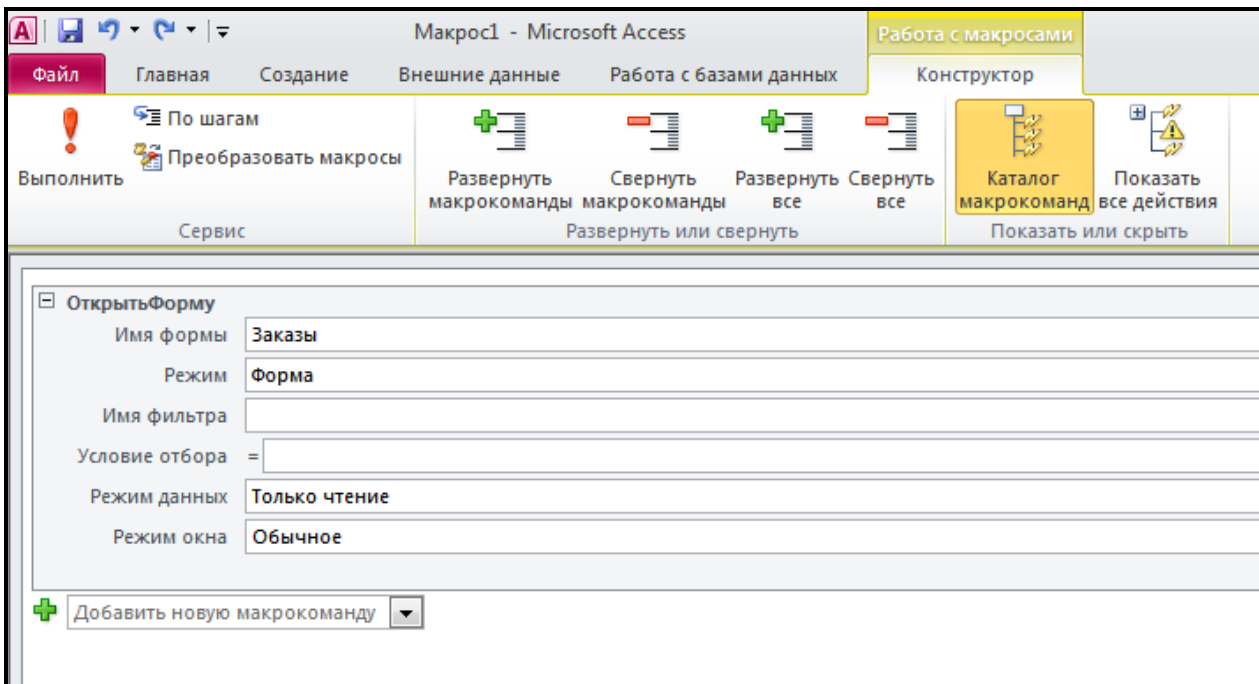


Рис. 30. Приклад макрокоманди «ВідкритиФорму»

У процесі розроблення програми кількість макросів може виявитися досить великою, тому важливо добре організувати доступ до потрібних макросів. Для цього в одному об'єкті можна об'єднати кілька макросів. Наприклад, рекомендується всі макроси, пов'язані з подіями в формі або звіті, об'єднати в окремий об'єкт, що відповідає даній формі або звіту. Кожен макрос групи повинен мати своє ім'я, а ім'я об'єкта макросу буде ім'ям групи макросів.

*Microsoft Access* продовжує виконання макросу, що входить до групи, поки не виявить рядок, який містить ім'я в стовпці Ім'я макросу, або поки не дійде до останнього рядка вікна макросів. Для поліпшення читаності складних макросів можна використовувати порожні рядки.

Для запуску одного з макросів групи використовується повне ім'я для посилання на макрос: Ім'яГрупи.Ім'яМакроса.

## **Запуск макросу**

Якщо макрос у вікні Конструктора макросу один, то для його запуску потрібно активізувати кнопку <Запуск> або вибрати команду Запуск – По кроках для налагодження макросу. Для запуску макросу з вікна БД можна також виконати подвійне клацання на потрібному макросі у вікні БД.

Однак якщо макросів декілька, то за допомогою цієї кнопки або команди можна запустити тільки перший макрос у групі. Для запуску будь-якого іншого макросу з групи використовується команда «Сервіс – Макрос – Виконати макрос».

Можна передбачити запуск макросу користувачем з форми за допомогою спеціальної кнопки. Для цього можна розмістити на екрані одночасно два вікна – вікно БД, відкрите на вкладці Макроси, та вікно форми, відкритої в режимі конструктора. У вкладці <Макроси> у вікні БД необхідно знайти потрібний макрос і перетягнути його за допомогою мишки в форму. На ній з'явиться кнопка, пов'язана з даними макросом, з підписом, відповідним імені макросу.

Можна також створити кнопку для запуску макросу в формі за допомогою майстра створення кнопок.

За допомогою діалогового вікна властивостей кнопки можна змінити текст підказки та задати комбінацію клавіш для її швидкого виклику. Для цього в режимі налаштування необхідно виділити кнопку на панелі інструментів, у вікні «Налаштування» натиснути кнопку <Змінити виділений об'єкт>. Потім потрібно вибрати в контекстному меню команду Властивості та в полі «Спливаюча підказка» ввести потрібний текст.

## **Створення макросів з умовами**

Для організації розгалужень у програмі в макросі задають умови виконання макрокоманд. Користувач запускає макрос на виконання, і далі все управління його виконанням здійснюється зсередини. Програма сама відкриває потрібні об'єкти, вибирає та обробляє дані, викликає інші об'єкти, дотримуючись алгоритму, що приводить до розв'язання завдання.

Для створення макросу, в якому виконуються ті чи інші макроси залежно від заданих умов, необхідно у вкладці Макроси вікна БД активізувати кнопку <Створити>, а потім натиснути кнопку <Умови> на панелі інструментів. У верхній частині вікна Конструктора макросів перед стовпцями Макрокоманда та Примітка з'явиться стовпець «Умова».

На рис. 31 наведений приклад макроса з розгалуженням, що виконує різні дії залежно від того, який пароль введений користувачем.

Условие	Макрокоманда	Примечание
[ГруппаВыбора]=1 And (IsNull([пароль])) Or [пароль]<>111	Сообщение	если пароль не введен или введен неправильно
...	КЭлементуУправления	повторный ввод пароля
...	ОстановитьМакрос	
[ГруппаВыбора]=2	ОткрытьФорму	открыть форму для покупателя
[ГруппаВыбора]=1 And [пароль]=111	ОткрытьФорму	открыть форму для сотрудника
	Закрыть	

Аргументы макрокоманды		
Сообщение	Введите правильный пароль	
Сигнал	Да	
Тип	Отсутствует	
Заголовок	Необходим ввод пароля	

Рис. 31. Приклад макроса з розгалуженням

### Автоматичне виконання макросу із завантаженням БД

Користувач має можливість створити макрос, що автоматично виконуватиметься з кожним відкриттям БД. Для цього достатньо привласнити йому ім'я AutoExec (перейменувати за допомогою контекстного меню). З кожним відкриттям бази даних *Microsoft Access* перевіряє, чи присутній макрос з ім'ям AutoExec, і якщо знаходить, то виконує його. Якщо ім'я AutoExec надане групі макросів, то з відкриттям БД буде виконаний тільки перший макрос цієї групи.

У макросі AutoExec доцільно помістити макрокоманди, які підготують потрібне робоче середовище для користувача; відкриють форми і, якщо необхідно, інші об'єкти бази даних; розмістять їх на екрані в зручному вигляді.

Макрос AutoExec не виконуватиметься, якщо, відкриваючи БД, ви утримуєте клавішу <Shift>.

За допомогою макросу з *MS Access* можна запустити додатки *Microsoft Office*, а також програми, що працюють під управлінням ОС *Windows* або *DOS* (наприклад, щоб виконати обчислення за допомогою табличного процесора *Excel* або зробити невеликі замітки за допомогою редактора *Word*).

Виклик полягає у вказівці імені потрібної програми. Повний шлях місцезнаходження викликаного файлу необхідно вказувати тільки в тому випадку, якщо він не заданий командою Path у файлі Autoexec.bat. Як пра-

вило, з викликом прикладних програм, що входять у комплект поставки *Microsoft Office*, місце їх розташування вказується в команді Path. Тому користувачеві задавати повний шлях немає необхідності.

### **Хід роботи**

1. Розробіть макрос, що містить кілька макрокоманд і дозволяє відкривати різні форми залежно від того, який пароль вводить користувач.
2. Якщо пароль не введений, має з'явитися відповідне повідомлення.
3. Якщо введений хибний пароль, то після повідомлення про помилку користувачу треба надати можливість повторно ввести пароль.
4. Оформіть звіт і захистіть його викладачу.

### **Контрольні запитання**

1. Які функції виконують макроси?
2. Яка структура макросів?
3. Як зберегти макрос?
4. Як задати автоматичний запуск макросів з відкриттям БД?
5. Як створити макрос для запуску певного додатка?
6. Чим макроси відрізняються від макрокоманд?
7. Які існують інструменти створення макросів?



## Рекомендована література

### Основна

1. Інформатика: Комп'ютерна техніка. Комп'ютерні технології: навч. посіб. / за ред. О. І. Пушкаря. – Київ : Видавничий центр «Академія», 2002. – 704 с.
2. Лабораторний практикум з інформатики та комп'ютерних технологій: навч. посіб. / за ред. О. І. Пушкаря. – Харків : ВД «ІНЖЕК», 2003. – 424 с.
3. Робоча програма навчальної дисципліни «Інформатика і комп'ютерна техніка» для студентів за спеціальністю 186 «Видавництво та поліграфія» / укл. О. Б. Бережна. – Харків : Вид. ХНЕУ ім. С. Кузнеця, 2016. – 54 с.

### Додаткова

4. Гурвиц Г. А. Microsoft Access 2010. Разработка приложений на реальном примере / Г. А. Гурвиц. – Санкт-Петербург : Изд. БХВ-Петербург, 2010 – 486 с.
5. Інформатика та комп'ютерна техніка (розділ «Проектування баз даних»). Конспект лекцій для студентів усіх спеціальностей всіх форм навчання. Ч. 1 / О. Б. Бережна, Ю. А. Сисоєва. – Харків : Вид. ХНЕУ, 2005. – 44 с.
6. Кузин А. В. Базы данных : учеб. пособ. для студ. высш. учеб. заведений / А. В. Кузин, С. В. Левонисова. – Москва : Academia, 2008. – 242 с.
7. Методические рекомендации к лабораторным работам по курсу «Информатика и компьютерная техника». Работа с базами данных в среде СУБД Microsoft Access для студентов специальностей 7.050108, 7.050201 всех форм обучения / сост. Е. Б. Бережная, М. В. Бутов. – Харьков : Изд. ХГЭУ, 2005. – 36 с.
8. Мюррей К. Первый взгляд на Office 2010 / К. Мюррей. – Москва : Microsoft, 2010. – 202 с.
9. Основы информационных технологий / под ред. В. В. Шкурко. – Минск : ГУО «Институт подготовки научных кадров Национальной академии наук Беларуси», 2010. – 253 с.
10. Румянцева Е. Л. Информационные технологии : учеб. пособ. / Е. Л. Румянцева, В. В. Слюсарь. – Москва : ИД «ФОРУМ»; ИНФРА-М, 2013. – 256 с.

## Зміст

Вступ.....	3
Змістовий модуль 2. Оброблення та зберігання інформації за допомогою СУБД <i>Microsoft Access</i> .....	4
Лабораторна робота 7. Зберігання та оброблення інформації за допомогою баз даних на прикладі БД «Борей» .....	4
Лабораторна робота 8. Проектування і створення індивідуальних баз даних .....	12
Лабораторна робота 9. Робота з запитам на вибірку. Запити з групуванням даних. Перехресні запити. Запити з параметрами. Модифікуючі запити.....	21
Лабораторна робота 10. Створення форм. Налаштування елементів форми .....	39
Лабораторна робота 11. Проектування звітів.....	46
Лабораторна робота 12. Створення макросів .....	50
Рекомендована література.....	57

НАВЧАЛЬНЕ ВИДАННЯ

# ІНФОРМАТИКА І КОМП'ЮТЕРНА ТЕХНІКА

**Методичні рекомендації  
до лабораторних робіт  
для студентів спеціальності  
186 «Видавництво та поліграфія»  
першого (бакалаврського) рівня**

## Частина 2

*Самостійне електронне текстове мережеве видання*

Укладач **Бережна** Олена Борисівна

Відповідальний за видання *О. І. Пушкар*

Редактор *Н. І. Ганцевич*

Коректор *Т. А. Маркова*

План 2018 р. Поз. № 289 ЕВ. Обсяг 59 с.

---

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А

*Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру*

**ДК № 4853 від 20.02.2015 р.**