

Міністерство освіти і науки України
Запорізький національний технічний університет

ОРГАНІЗАЦІЯ БАЗ ДАНИХ І ЗНАНЬ

**методичні вказівки
до лабораторних робіт**
для студентів напрямку підготовки
6.050101 "Комп'ютерні науки"
усіх форм навчання



2015

Організація баз даних і знань. Методичні вказівки до лабораторних робіт для студентів напрямку підготовки 6.050101 “Комп’ютерні науки” усіх форм навчання / Уклад. С.К.Корнієнко – Запоріжжя: ЗНТУ, 2015. - 66 с.

Укладач: С.К. Корнієнко, доцент, к.т.н.,

Рецензент: В.П. Пінчук, доцент, к.ф.-м.н.

Відповідальний
за випуск:

В.І.Дубровін, професор, к.т.н.

Затверджено
на засіданні кафедри
"Програмні засоби"

Протокол № 5 від 26.12.2014 р.

ЗМІСТ

1 Лабораторна робота № 1 "Створення схеми бази даних"	4
2 Лабораторна робота № 2 "Створення простих запитів"	13
3 Лабораторна робота № 3 "Створення складних запитів"	24
4 Лабораторна робота № 4 "Створення та використання форм даних"	37
5 Лабораторна робота № 5 "Розробка звітів"	52
6 Лабораторна робота № 6 "Використання макросів"	58
Література.....	66

1 ЛАБОРАТОРНА РОБОТА № 1 "СТВОРЕННЯ СХЕМИ БАЗИ ДАНИХ"

1.1 Мета роботи

Метою роботи є ознайомлення з пакетом MS Access, придбання початкових навичок роботи з ним і створення схеми бази даних.

1.2 Завдання до лабораторної роботи

Ознайомитися зі змістом пункту 1.3 даних методичних указівок. Відповідно до індивідуального завдання розробити схему бази даних.

Ввести дані.

1.3 Основні теоретичні відомості

1.3.1 Специфікація структури таблиці

При проектуванні бази даних треба визначити, яка саме інформація повинна входити до бази даних і чи повинна вся збережена в базі даних інформація розташовуватися в одній таблиці або її краще розділити на декілька таблиць.

Створення таблиці виконується у наступним чином

- запустіть MS Access;
- задайте ім'я файла бази даних і натисніть кнопку **Створити**;
- у вікні бази даних, що з'явилося, виберіть закладку **Таблиці** та натисніть кнопку **Створити**. На екрані з'явиться вікно **Створення таблиці**, у правій частині якого знаходиться список можливих засобів створення таблиці;
 - виберіть засіб **Конструктор** і, натиснувши кнопку **ОК**, приступайте до проектування нової таблиці за допомогою конструктора таблиць

Специфікація таблиці містить опис полів. Для кожного поля необхідно зазначити його ім'я та тип даних (типи даних, які використовуються в Access, наведені в таблиці 1.1). Можна також прокоментувати призначення поля в колонці **Опис**.

Таблиця 1.1 - Типи даних Access

Тип даних	Опис
<i>Текстовий</i>	Текст, максимальна довжина якого дорівнює 255 символів або визначається значенням властивості "Розмір поля"
<i>Поле МЕМО</i>	Текст, максимальна довжина якого дорівнює 64 Кбайт
<i>Числовий</i>	Числові дані, які використовуються для проведення розрахунків (1, 2, 4 або 8 байт)
<i>Дата/час</i>	Дати та час (8 байт)
<i>Грошовий</i>	Числа, що зберігаються з точністю до 15 знаків у цілій та до 4 знаків у дробовій частині (8 байт)
<i>Лічильник</i>	Число, що автоматично збільшується при додаванні в таблицю кожного нового запису. Значення цих полів не можуть бути змінені (4 байта)
<i>Логічний</i>	Логічні значення, а також поля, що можуть містити одне з двох припустимих значень.
<i>Поле об'єкта OLE</i>	Об'єкт, який створений іншим додатком і пов'язаний або введений у таблицю (до 1 Гбайта).
<i>Майстер підстановок</i>	Відображає дані, що підставляються з іншої таблиці або списку, що задається користувачем. При цьому запускається майстер підстановок, за допомогою якого визначається тип даних і організується зв'язок з іншою таблицею.

Обов'язковою умовою нормального опрацювання файла бази даних є однозначна ідентифікація кожного запису даних, для чого одне з полів призначається у якості ідентифікатора (або *первинного ключа*) запису.

За допомогою піктограми ключа на стандартній панелі інструментів можна оголосити потрібне поле ключовим.

1.3.2 Властивості полів

Вкладка "Загальні"

Після визначення імені та розміру поля треба вказати його властивості на вкладці "Загальні" в лівому нижньому куті екрану. Опис властивостей полів наведений у таблиці 1.2

Таблиця 1.2 - Властивості полів Access

Властивість	Пояснення
Розмір поля	Текстовий: обмежує розмір поля заданим числом символів (1-255); по замовчанню приймається значення 50. Числовий: дозволяє задавати тип числа.
Формат поля	Змінює зовнішній вигляд даних після вводу значень.
Маска вводу	Використовується для організації вводу даних у жорстко визначеному форматі (номера телефонів, поштові індекси, тощо).
Число десяткових знаків	Задає кількість десяткових знаків (тільки в даних числового й грошового типів).
Підпис	Необов'язковий ярлик для полів форм і звітів (що замінює ім'я поля).
Значення по замовчанню	Значення, що автоматично з'являється в полі перед введенням даних.
Умова на значення	Забезпечує перевірку припустимості даних на підставі правил, які створені за допомогою виразів або макросів.
Повідомлення про помилку	Повідомляє про неприпустимість уведеного значення (задається користувачем).
Обов'язкове поле	Визначає, чи обов'язково вводити в цьому полі
Індексоване поле	Прискорює доступ до даного та при необхідності обмежує дані, що вводяться, тільки унікальними значеннями.

Формати дозволяють указувати спосіб відображення на екрані тексту, чисел, значень дат і часу. Деякі типи даних мають стандартний формат, інші - тільки визначений користувачем формат, а треті – обидва формати. ***Формати впливають тільки на відображення даних, а не на спосіб їхнього вводу або зберігання в таблиці.***

Для завдання форматів користувача у полях даних типу **Текстовий** і **Мето** використовуються чотири символи (таблиця 1.3).

Таблиця 1.3 - Символи завдання форматів користувача

Символ	Опис
@	обов'язковий текстовий символ або прогалина
&	необов'язковий текстовий символ
>	перетворить усі символи в прописні
<	перетворить усі символи в рядкові

Символи @ і & впливають на окремі символи, а > та < - на усі введені символи. Якщо, наприклад, необхідно, щоб введене ім'я відображалось прописними буквами, то треба ввести значення > для властивості **Формат поля**. При введенні, наприклад, номера телефону можна задати такий формат (@@(@))@@-@@-@@. Якщо потім ввести 612344586, то дані будуть відображені у вигляді (612)34-45-86.

Маска призначена для полегшення вводу даних у поле (табл. 1.4).

Таблиця 1.4 –Символи, які використовуються в масці вводу

Символ	Опис
1	2
0	Цифра (0-9, обов'язковий символ; знаки плюс (+) і мінус (–) не дозволені)
9	Цифра або прогалина (0-9, необов'язковий символ; знаки плюс (+) і мінус (–) не дозволені)
#	Цифра, знаки плюс (+), мінус (–) або прогалина (необов'язковий символ; незаповнені позиції виводяться як прогалини в режимі редагування, але вилучаються при зберіганні даних)
L	Буква (обов'язковий символ)
?	Буква (необов'язковий символ)
A	Буква або цифра (обов'язковий символ)
a	Буква або цифра (необов'язковий символ)
&	Будь-який символ або прогалина (обов'язковий символ)
C	Будь-який символ або прогалина (необов'язковий символ)
<	Всі введені після нього символи перетворюються в рядкові
>	Всі введені після нього символи перетворюються в прописні

Продовження таблиці 1.4

1	2
!	Вказує, що маска вводу заповнюється справа наліво. Варто використовувати, якщо в лівій частині маски знаходяться позиції, заповнювати котрі необов'язково. Символ можна розмішувати в будь-якій позиції маски вводу.
\	Наступний за ним символ відображається в такому ж вигляді, у якому він був уведений (наприклад, \{ призведе до появи символу {).
.,:;- /	Десятковий роздільник, роздільники груп розрядів, часу або дати. (Використані символи роздільників визначаються у вікні <i>Мова та стандарти</i> панелі керування Windows).

Наприклад, для поля **Телефон** можна задати таку маску вводу, що дозволяла б вводити тільки цифри й автоматично добавляла проміжні символи: (999)900-00-00. Access автоматично додасть символ "\" для кожного проміжного символу: \ (999)900\00\00.



Увага

Якщо визначити для даних маску вводу та властивість **Формат поля**, то при відображенні даних у Access властивість **Формат поля** буде мати перевагу. Це означає, що навіть після зберігання маски вводу разом із даними, вона буде ігноруватися при форматуванні даних.

Вкладка "Підстановка"

У цій вкладці знаходиться тільки властивість **Тип елемента керування**, для якого є три варіанти вибору: Поле, Список, Поле зі списком.

При завданні **Списку** або **Поля зі списком** необхідно зазначити джерело, в якому зберігаються варіанти даних, які вводяться (Таблиця/запит, ім'я стовпчика або стовпчиків). При введенні даних користувач може вибирати значення з готового списку, що полегшує вводу інформації, а також підвищує її достовірність.

🔗 Примітка

Поле зі списком відрізняється від **Списку** тим, що дозволяє вводити значення, відсутні в списку.

Завдання типу підстановки здійснюється або вручну на відповідній вкладці, або за допомогою **Майстра підстановки**, що викликається в стовпчику **Тип даних**.

🔗 Примітка

Припускається також створення в таблиці поля підстановок із посиланням на інше поле в тій же таблиці.

1.4 Встановлення зв'язків

1.4.1 Ключі

Для встановлення зв'язків між таблицями необхідно встановити зв'язок між тими полями, у яких міститься загальна (за сенсом) інформація. Ці поля можуть мати різні імена, але тип даних і довжина полів, а також (що особливо важливо) інформація в обох полях відповідних записів повинні бути однаковими в обох таблицях. Як правило, зв'язок устанавлюється з'єднанням **ключових полів** таблиць: **первинного ключа** однієї таблиці та **зовнішнього ключа** - в іншій.

Первинний ключ

Кожна таблиця повинна містити **первинний ключ** - одне або декілька полів, уміст яких унікальний для кожного запису.

При створенні таблиць Access завжди пропонує задати первинний ключ із типом даних **Лічильник**, значення якого буде автоматично збільшуватися на одиницю при додаванні нового запису.

🔗 Примітка

Поля первинного ключа повинні бути як можна коротшими, оскільки їхні розміри впливають на швидкість виконання операцій у базах даних.

Зовнішні ключі. Поняття посилальної цілісності

Коли поле однієї таблиці посилається на інше поле (в іншій таблиці), воно зветься **зовнішнім ключем (foreign key)**. Поле, на який він посилається, зветься його **батьківським ключем (parent key)**. Імена зовнішнього та батьківського ключів можуть бути неоднаковими.

Зовнішній ключ, як і первинний, може бути визначений на будь-якій кількості полів, які усі разом розглядаються як єдине ціле (складовий ключ). Зовнішній ключ і батьківський, на який він посилається, повинні бути визначені на однаковій множині полів (за кількістю полів, типам полів і порядку слідування полів).

Таким чином, **основна ідея посилальної цілісності** полягає в тому, що всі значення зовнішніх ключів відсилають до визначеного рядку батьківського ключа. Як тільки значення розміщується в зовнішньому ключі, батьківський ключ перевіряється, щоб переконатися в тому, що таке значення в ньому є присутнім; інакше команда відхиляється.

1.4.2 Конструктор зв'язків

Конструктор зв'язків викликається командою **Сервіс⇒Схема даних** або за допомогою кнопки **Схема даних**  панелі інструментів.

На початку роботи вікно **Схема даних** не містить таблиць. Додавати таблиці до вікна можна у такий спосіб:

– використавши діалогове вікно **Додавання таблиці**; що відчиняється автоматично, якщо вікно **Схема даних** для бази даних відчиняється вперше;

– за допомогою кнопки **Додати таблицю**  панелі інструментів;

– за допомогою команди **Зв'язки⇒Додати таблицю**;

– у вікні **Схема даних** викликати контекстне меню та вибрати команду **Додати таблицю**.

Установка зв'язків між таблицями

Перш за все, слід мати на увазі, що таблиці зв'язуються лише за полем, яке є загальним (за вмістом , а не за назвою) для обох таблиць. Для установки зв'язків між таблицями потрібно вибрати загальне поле в одній таблиці та перетягнути його мишкою в загальне поле тієї таблиці, яку необхідно зв'язати з першою. При цьому викликається діалогове вікно Зв'язки, показане на рисунку 1.1.

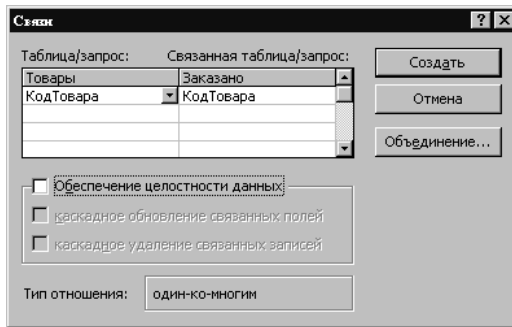


Рисунок 1.1 - Діалогове вікно *Зв'язки*

У вікні *Зв'язки* міститься інформація про поля зв'язку головної та підпорядкованої таблиць, параметрах забезпечення цілісності даних і тип зв'язку, який встановлюється.

У верхній частині вікна виводяться найменування таблиць. Таблиця, що знаходиться зліва, рахується первинною в цьому відношенні. Нижче розміщується поля зв'язку обох таблиць. Після встановлення зв'язку можна включити підтримку цілісності даних.

На замітку

Цілісність даних - це набір правил, що охороняють пов'язані таблиці від внесення некоректних значень.

Каскадне оновлення пов'язаних полів

Якщо включена опція *Забезпечення цілісності даних* у вікні *Зв'язки*, Access дозволяє включити опцію *Каскадне оновлення пов'язаних полів*. При цьому зміна значення ключа в батьківській

таблиці автоматично викликає оновлення відповідних полів в підлеглих таблицях.



Увага

Якщо опція **Каскадне оновлення пов'язаних полів** не включена, змінити значення ключового поля первинної таблиці не можна.

Каскадне вилучання пов'язаних записів

Ця опція дає можливість після вилучення запису у первинній таблиці автоматично вилучати відповідні записи в підлеглих таблицях.



Увага

Будьте обережні з опцією вилучення пов'язаних записів! Access не попереджує про каскадне вилучення

Для вилучення існуючого зв'язку потрібно зайти у вікно **Схема даних**, виділити потрібний зв'язок і вилучити його за допомогою або кнопки на панелі інструментів, або команди **Вилучити** в контекстному меню

Контрольні питання

- ? Які типи даних використовуються в Access?
- ? Як задається режим автоматичної нумерації записів?
- ? Як можна задати значення по замовчанню в числові поля?
- ? Що таке формат даних, як він задається?
- ? Для чого використовується маска вводу та як вона задається?
- ? Як задається ключове поле?
- ? Які властивості поля підтримує Access?
- ? Як можна задати список значень для підстановки в поле?
- ? Що таке **первинні** та **зовнішні ключі**?
- ? Як встановлюються та вилучаються зв'язки між таблицями?
- ? Поняття **цілісності даних**; як воно підтримується в Access?
- ? Як встановлюється **Каскадне вилучення пов'язаних полів**?
- ? Як здійснити **Каскадне оновлення пов'язаних полів**?

2 ЛАБОРАТОРНА РОБОТА № 2 "СТВОРЕННЯ ПРОСТИХ ЗАПИТІВ"

2.1 Мета роботи

Метою роботи є ознайомлення з основними прийомами упорядкування і використання запитів.

2.2 Завдання до лабораторної роботи

Для бази даних, створеної в попередній лабораторній роботі, за узгодженням із викладачем розробити різноманітні типи запитів, використовуючи можливі варіанти їхнього створення.

2.3 Основні теоретичні відомості

За допомогою запитів користувач може вибрати з таблиці необхідні дані й відобразити їх на екрані. Результатом опрацювання запиту є таблиця, яка має назву *динамічного, тимчасового набору даних*.

При зберіганні запиту зберігається тільки структура запиту, тобто лише перелік таблиць, список полів, порядок сортування, обмеження на записи, тип запиту, тощо. Динамічний набір даних при цьому не зберігається. При кожному виконанні запиту він будується знову на базі "свіжих" табличних даних.

2.3.1 Типи запитів

Access підтримує такі основні типи запитів:

Запит на вибірку	Найпоширеніший тип запиту. Витягає дані з однієї або декількох таблиць і результати відображає в режимі таблиці.
Груповий запит	Це спеціальна версія запиту на вибірку. Дозволяє обчислювати суми, підраховувати кількість записів і виконувати розрахунки підсумкових значень. При виборі цього типу запиту Access додає до бланку запиту рядок <i>Групова операція</i>

Запит на зміну	Дозволяє створювати нові таблиці (команда Створення таблиці) або змінювати дані в існуючих таблицях (команди Вилучання , Оновлення та Додавання).
Перехресний запит	Відображає результати статистичних розрахунків (такі як суми, кількість записів і середнє значення), які виконані за даними одного поля. Ці результати групуються за двома наборами даних у форматі перехресної таблиці (аналогічно таблиці Excel).
SQL	Це запит, який створений за допомогою інструкції SQL.
Запит із обмеженням	Цей обмежувач запиту можна використовувати тільки в кон'юнкції з іншими п'ятьма типами запитів. Він дозволяє задавати кількість перших записів або частину загальної кількості записів у відсотках, яку користувач хотів би одержати в будь-якому виді запитів.

2.3.2 Створення запитів на вибірку

Запит на вибірку називають також **запитом за зразком (QBE - Query By Example)**. Для його проектування потрібно у вікні бази даних вибрати розділ **Запити**, а потім натиснути на кнопки **Створити**.

На екрані з'явиться діалогове вікно **Новий запит** із списком варіантів створення запиту:

- **Конструктор** - новий запит створюється "вручну", без допомоги майстра;

- **Простий запит** - на основі обраних полів створюється простий запит;

- **Перехресний запит** - створюється запит, у якому дані надаються в компактному форматі, подібному формату відображення даних в електронних таблицях;

- **Повторювані записи** - створюється запит, у якому відображаються лише дублікати значень полів у таблиці або простому запиті;

- **Записи без підпорядкованих** - створюється запит, у якому

відображаються ті записи з однієї таблиці, що не мають зв'язків із записами іншої таблиці (підпорядкованої).

Для створення запиту треба вибрати пункт **Конструктор**, після чого Access відобразить порожній проект запиту, а також діалогове багатосторінкове вікно **Додавання таблиці**, за допомогою якого можна вибрати таблиці або інші запити, що будуть використовуватися при створенні даного запиту.

Для вибору таблиць або запитів потрібно виділити відповідний елемент і натиснути кнопку **Додати**. По закінченні вибору треба натиснути кнопку **Закрити**.

Вікно Конструктора запитів складається з двох частин (рис. 2.1):

Область таблиць запиту - це місце, де розміщуються таблиці або запити й встановлюються зв'язки між ними.

Бланк запиту призначений для визначення полів і умов, що будуть використані для витягу динамічного набору даних. У кожному стовпчику бланка запиту міститься інформація про одне поле з таблиці або запиту у верхній частині екрана.

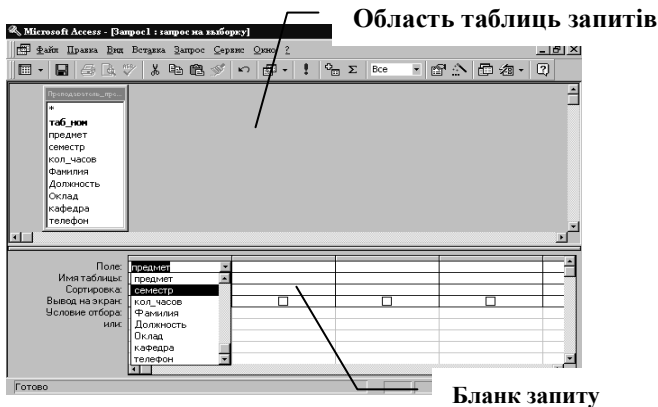


Рисунок 2.1 - Вікно конструктора запитів

Бланк запиту містить шість рядків:

1. **Поле** – ім'я поля;
2. **Ім'я таблиці** – ім'я таблиці;

3. **Сортування** – місце запровадження інструкцій сортування;
4. **Виведення на екран** – визначає, чи буде присутнім поле в динамічному наборі даних;
5. **Умова відбору** – містить першу умову, що обмежує набір записів;
6. **Або** – інші умови обмеження набору записів.

Вставка та вилучання полів

Ім'я поля вводиться до бланка запиту або перетягуванням його з відповідної таблиці, або зі списку, що розкривається, у потрібному стовпчику бланка запиту. Щоб додати до бланка запиту всі поля таблиці, потрібно виділити поле (*) у таблиці, а потім перетягнути піктограму *Поле* до першої чарунки бланку запиту.

Для вилучання стовпчиків із бланка запиту можна скористатися командами *Правка⇒Вилучити стовпчики* або *Правка⇒Очистити бланк*, а також шляхом виділення стовпчика та натискання клавіші .

Зміна порядку сортування

Сортування даних за окремими полями може виконуватися як по зростанню, так і по зменшенню. Сортування за декількома полями залежить від порядку розміщення полів у бланку запиту (зліва на право).

Примітка

Поля типу *Мето* та *OLE* не сортуються

Відбір записів

Умови відбору записів - це набір визначених у Access або заданих користувачем правил. Умови задаються за допомогою виразу. Вираз може задаватися за зразком або використовувати складні функції вибору.

Наприклад, можна вибрати товари, які поставлені у поточному кварталі постачальниками, розташованими у визначеному регіоні.

Зв'язування таблиць у запиті

Дуже часто виникає необхідність вибірки інформації з декількох таблиць.



Увага

Всі таблиці в запиті повинні бути пов'язані, принаймні, з одною таблицею. Якщо, наприклад, помістити в запит дві таблиці та не з'єднати їх, то Access, створить запит на основі **декартового добутку** цих двох таблиць (повний перебір).

Зв'язки між таблицями можна створити таким чином:

- при створенні макета бази даних;
- вибрати для запиту дві таблиці з однаковими полями одного типу, одне з яких також є **ключовим** полем в одній з цих таблиць;
- створити зв'язки у вікні конструктора запитів.

У перших двох способах зв'язки встановлюються автоматично. Проте іноді необхідно додати до запиту незв'язані таблиці, наприклад:

- дві таблиці мають загальне поле, яке назване різними іменами;
- таблиця не пов'язана (й не може бути пов'язана) з іншою таблицею.

Зв'язати дві таблиці, що не були автоматично пов'язані, можна у вікні конструктора запитів. Проте таке з'єднання не призводить до створення постійного зв'язку між цими таблицями. Такий зв'язок буде функціонувати тільки в запиті, з яким ви працюєте.

Типи зв'язків між таблицями

Між таблицями й запитамі можна встановити такі типи зв'язків:

- один-до-одного;
- один-до-багатьох;
- багато-до-одного.
- багато-до-багатьох.

При завданні зв'язку між таблицями встановлюються тільки правила зв'язку, але не засіб перегляду даних на основі цього зв'язку.

Для перегляду даних у двох таблицях їх необхідно з'єднати загальним полем (або групою полів) в обох таблицях. Такий метод з'єднання таблиць називається **з'єднанням (joining)**. Існують різноманітні типи з'єднань:

- рівні з'єднання (внутрішні з'єднання);
- зовнішні з'єднання;
- самоз'єднання;
- перехресні (декартові) з'єднання.

Внутрішнє з'єднання (рівне з'єднання)

Прийнято в Access по умовчанням. Означає, що з таблиць вибираються тільки записи, що мають однакове значення в обох таблицях.

При з'єднанні таблиці одержують деякі нові риси (або властивості) зв'язку. **Властивість зв'язку** – це правило, що управляє відображенням усіх записів (для заданих полів) і яким керується Access при інтерпретації будь-яких винятків (а можливо, й помилок) у таблицях).

Access дозволяє змінити тип з'єднання шляхом зміни його властивостей. Для зміни властивостей з'єднання треба вибрати лінію зв'язку та двічі клікнути на ній лівою кнопкою миші. На екрані з'явиться діалогове вікно **Параметри з'єднання** (рис. 2.2).

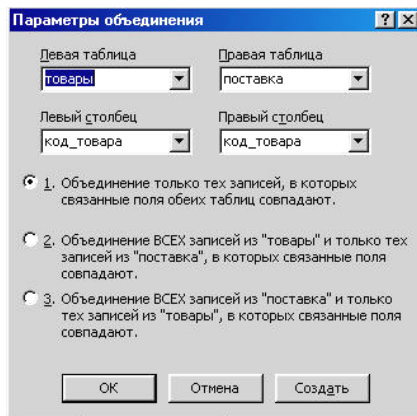


Рисунок 2.2 - Діалогове вікно "Параметри з'єднання"

У вікні є три перемикачі. Перший варіант зазвичай називають **внутрішнім** (INNER JOIN – у термінах мови SQL) **з'єднанням**, а два інших – зовнішнім.

Зовнішні з'єднання відображають усі записи, що мають відповідні друг другу дані в обох полях. Вони також показують записи з однієї таблиці, що не мають підпорядкованих їм даних в іншій таблиці. Існує два види зовнішніх з'єднань: **ліві** (LEFT JOIN) і **праві** (RIGHT JOIN).

✎ Примітка

Слід мати на увазі, що при перекладі мови інтерфейсу Access термін **Join** (операція реляційної алгебри **з'єднання**) був помилково перекладений як **об'єднання** (а це вже принципово інша операція – **Union**). Ця прикра помилка також дуже часто зустрічається у технічній літературі.

2.3.3 Вибір за складними умовами відбору записів

Оператор Like і символи підстановки

У полі **Умови відбору** бланка запиту можна задавати маску відбору за допомогою оператора **Like**, у якому використовуються символи підстановки (таблиця 2.1 і таблиця 2.2).

💡* Увага

Оператор **Like** і символи підстановки можуть використовуватися тільки для трьох типів полів: текстового, **Мето** та дати. При роботі з іншими типами полів це може призвести до помилки.

Таблиця 2.1 - Символи підстановки, які використовуються оператором Like

Спеціальний символ	Призначення
?	Будь-який символ (0-9, Aa-Zz, Aa-Яя)
*	Будь-яка кількість символів (0-n)
#	Будь-яка цифра

Таблиця 2.2 – Приклади використання символів підстановки разом з оператором Like

Вираз	У яких полях використане	Результат
Like "34*"	Клієнти.Телефон	Знаходить усі записи про клієнтів, телефони яких починаються на 34.
Like "[A-M]*"	Клієнти.Прізвища	Знаходить прізвища клієнтів, що починаються на літери від А до М
Like "*ОЗП*"	Товар.Опис	Знаходить усі записи, що містять слово ОЗП у будь-якому місці поля Опис.
Like "/*.01.03"	Замовлення.Дата	Знаходить усі записи за січень 2003 року.
Like "!"Монітор"	Товар.Опис	Усі товари, що відрізняються від "Монітор

Завдання незбіжних значень

Для завдання незбіжних значень використовуються оператори **NOT** або оператор "не дорівнює" (<>). Вони записуються перед тим виразом, у якому записаний тип розбіжності.

Наприклад, **NOT Ріо-де-Жанейро** або <> **фонтан**.

Запровадження декількох умов відбору записів у одному полі

Для створення складних умов відбору записів можна використовувати декілька операторів. У основному такі умови відбору записів складаються з декількох операторів **AND** і **OR**. Для пошуку записів, що не задовольняють деякому значенню, можна використовувати оператор **NOT** із цим значенням.

При роботі з запитами ім'я поля треба брати в квадратні дужки, як при створенні полів, які обчислюються, або вказівці умов відбору записів, наприклад, [Дата Візиту]+30.



Увага

Якщо при вводі імені поля в умови відбору записів не включити його в квадратні скобки, Access автоматично візьме це ім'я в лапки й буде сприймати його, як текст, а не як ім'я поля.

Функції оператора *OR*

Оператор **OR** можна використовувати або в одному осередку поля бланка запиту (рис. 2.3), або використовуючи рядок *або* (рис. 2.4).

Поле: Имя таблицы: Сортировка: Вывод на экран: Условие отбора: или:	наименование	цена	поставщик	
	Товары	Товары	Товары	
	☒	☒	☒	
	"монитор" Or "принтер" Or "процессор"			

Рисунок 2.3 - Використання оператора *OR* в умовах відбору записів

Поле: Имя таблицы: Сортировка: Вывод на экран: Условие отбора: или:	наименование	цена	поставщик	
	Товары	Товары	Товары	
	☒	☒	☒	
	"монитор"			
	"принтер"			

Рисунок 2.4 - Використання рядка *або* (*OR*) бланка запиту



Увага

Крім рядків *Умова відбору* та *або (OR)* у Access передбачено ще п'ять додаткових рядків *або (OR)* для запровадження умов відбору записів. При необхідності запровадження більшої кількості умов, можна використовувати оператор **OR** між окремими примірниками даних.

Використання списку значень і оператора *IN*

Для створення умов відбору для одного поля з використанням декількох примірників даних можна скористатися оператором **IN**. При виконанні запиту будуть відображені записи, що містять у зазначеному

полі значення, що перераховані в списку оператора **IN**. Наприклад, **IN** ("принтер"; "монітор") для поля **Найменування**.

Завдання діапазону значень

Діапазон значень можна задати за допомогою операторів **AND** або **BETWEEN...AND...**

Оператор **AND** використовується в тих запитах, у яких значення поля повинно одночасно задовольняти декільком умовам.

Наприклад, **>=100 And <=200** або **Between 100 And 200**.

Запровадження умов відбору записів за декількома полями

Для використання операторів **AND** і **OR** у запиті за декількома полями треба погоджувати розміщення примірників даних (або масок) в осередках **Умова відбору** й **або (OR)** одного поля з їхнім розміщенням в іншому полі.

Якщо ви хочете зв'язати декілька полів за допомогою оператора **AND**, розміщайте примірники даних (або маски) в одному й тому ж рядку. Для зв'язку умов оператором **OR** їх слід розміщувати в різних рядках бланка запиту (рис. 2.5).

Поле:	наименование	цена	поставщик	
Имя таблицы:	Товары	Товары	Товары	
Сортировка:				
Вывод на экран:	☒	☒	☒	☐
Условие отбора:	"монитор"	>500	"LG"	
или:	"принтер"	>300	"HP"	
	"процессор"	<300	"Intel"	

Рисунок 2.5 - Приклад використання операторів AND і OR у запиті

У даному прикладі умова задається формулою:

("монітор" **And** >500 **And** "LG") **Or**
 ("принтер" **And** >300 **And** "HP") **Or**
 ("процесор" **And** <300 **And** "Intel")

Створення в запиті поля, що обчислюється

При створенні запиту можна використовувати не тільки поля з таблиць, але й **поля, що обчислюються**. Наприклад, можна створити обчислюване поле **Премія**, у якому буде відображатися результат множення значення поля **Оклад** на значення поля **Кв_Премія**. Для цього у відповідну чарунку (осередок) бланка запиту потрібно ввести оператор **Премія: [Оклад] * [Кв_Премія]**.

Примітка

Ім'я для поля, що обчислюється, рекомендується задавати з двох основних причин: по-перше, для заголовка стовпчика таблиці запиту, й, по-друге, для звертання до цього поля у формі, звіті або іншому запиті.

Контрольні питання

- ? Що таке "запит"?
- ? Які типи запитів підтримує Access?
- ? Призначення і склад бланка запитів.
- ? Вставка і вилучання полів із бланку запиту.
- ? Як відсортувати таблицю по окремих полях?
- ? Яким чином задаються критерії відбору записів?
- ? Зв'язування таблиць у запиті.
- ? Які існують типи зв'язків між таблицями
- ? Які існують типи з'єднань між таблицями?
- ? Що називається внутрішнім з'єднанням?
- ? Що називається зовнішнім з'єднанням?
- ? Що таке "віртуальна таблиця", та її відмінності від базової.
- ? Використання функцій **OR** і **AND** при завданні умов відбору записів.
- ? Призначення оператора **Like**?
- ? Призначення оператора **IN**?
- ? Завдання діапазону значень.
- ? Як створити в запиті поле, що обчислюються?

3 ЛАБОРАТОРНА РОБОТА № 3 "СТВОРЕННЯ СКЛАДНИХ ЗАПИТІВ"

3.1 Мета роботи

Метою роботи є ознайомлення з основними прийомами упорядкування та використання складних запитів.

3.2 Завдання до лабораторної роботи

3.2.1 Ознайомитися зі змістом пункту 8.3 методичних вказівок.

3.2.2 Для бази даних, створеної в попередніх лабораторних роботах, за узгодженням із викладачем розробити різноманітні типи запитів, використовуючи можливі варіанти їхнього створення:

- групові;
- перехресні;
- параметричні;
- запити на зміну.

3.3 Основні теоретичні відомості

3.3.1 Створення групових запитів

Для виконання групового запиту необхідно в режимі конструктора запиту активізувати рядок *Групова операція* одним із двох засобів:

- вибрати команду *Вид⇒Групові операції*;
- клацнути на кнопці *Групові операції* (Σ) панелі інструментів.

У Access можна виконувати групові запити для всіх записів відразу або для групи записів в одній або декількох таблицях. Для виконання обчислень необхідно для кожного поля, що включене до запиту, вибрати відповідну опцію зі списку, що розкривається, у рядку *Групова операція* (рис. 3.1).

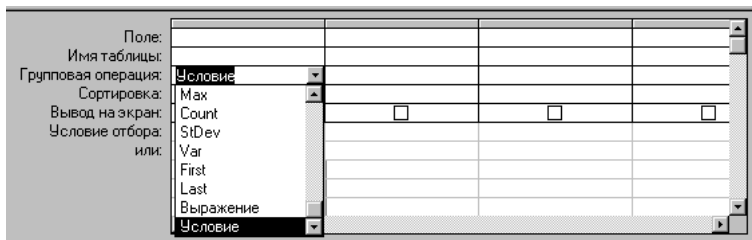


Рисунок 3.1 - Список, що розкривається, рядку *Групова операція*

Всі 12 функцій цього списку можна розділити на чотири категорії:

- групування;
- вирази;
- умови;
- підсумкові функції.

Групування

Ця категорія складається тільки з функції *Групування (GROUP BY* у термінах *SQL*), що служить для вказівки поля, яке буде використовуватися в якості групуючого для подальших підсумкових обчислень по кожній групі. Ця функція вибирається по умовчанням для всіх чарунок рядка *Групова операція*.

Вирази

Дана функція вказує Access, що необхідно створити поле, значення якого буде обчислюватися. Наприклад, можна створити запит, що буде, базуючись на знижці для клієнта, показувати зекономлену їм суму.

Умова

Дана функція (у термінах *SQL* – *Where*) повідомляє Access критерії відбору полів для обчислень. Наприклад, потрібно створити запит для перерахування всіх типів товарів вартістю нижче 500\$.

Оскільки поле вартості не використовується для угруповання й не може застосовуватися для підсумкових обчислень, користувач визначає параметр умови, повідомляючи Access, що використовувати

це поле слід як поле обмежуючого критерію перед виконанням обчислень, тобто умова визначає, над якими записами будуть провадитися обчислення.

Підсумкові функції

Підсумкова (або агрегатна) функція означає функцію, що одержує в якості аргумента групи записів і виконує деякі математичні дії над ними (таблиця 3.1). Це може бути як проста функція, так і складний вираз, що базується на інших функціях і задається користувачем.

Таблиця 3.1 - Підсумкові (агрегатні) функції

Функція	Опис	Підтримувані типи полів
Count	Кількість непорожніх полів	AutoNumber, Number, Currency, Date/Time, Yes/No, Text, Memo, OLE
Sum	Сума значень у полі	AutoNumber, Number, Currency, Date/Time, Yes/No
Avg	Середнє значення у полі	AutoNumber, Number, Currency, Date/Time, Yes/No
Max	Максимальне значення у полі	AutoNumber, Number, Currency, Date/Time, Yes/No, Text
Min	Мінімальне значення у полі	AutoNumber, Number, Currency, Date/Time, Yes/No, Text
StDev	Стандартне відхилення значень у полі	AutoNumber, Number, Currency, Date/Time, Yes/No
Var	Дисперсія розподілу значень у полі	AutoNumber, Number, Currency, Date/Time, Yes/No
First	Значення поля в першому запису таблиці або запиту	AutoNumber, Number, Currency, Date/Time, Yes/No, Text, Memo, OLE
Last	Значення поля в останньому запису таблиці або запиту	AutoNumber, Number, Currency, Date/Time, Yes/No, Text, Memo, OLE

Виконання групових операцій для групи записів

При необхідності одержання інформації з окремих груп записів потрібно згрупувати записи за відповідним полем, а потім задати тип потрібної агрегатної функції.

Наприклад, можна згрупувати записи по окремим замовникам, а потім для кожного замовника визначити кількість замовлень, середню вартість тощо.

Access дозволяє виконувати групові операції для багатьох полів і багатьох таблиць так само, як і для одного поля однієї таблиці. Наприклад, можна згрупувати записи по постачальникам і по типу товару для визначення суми поставлених товарів кожного типу для окремих постачальників.

Примітка

Access групує записи на підставі порядку проходження полів, у яких установлена функція **Групування**, зліва направо. Порядок проходження не змінює результат, але визначає вид представлення даних у результуючій таблиці.

Визначення критерію для групового запиту

Крім групування записів для запитів, можна зазначити критерій (умову відбору) для обмеження кількості записів, що будуть розглядатися при обчисленнях і відображенні результату. Іншими словами, можна задати умову вибірки записів **до** обчислень і **після** обчислень.

У першому випадку умова задається в стовпчику, у якому в рядку **Групова операція** коштує опція **Умова** (рисунок 3.2).

*** Увага**

Якщо для поля визначається опція **Умова**, то побачити це поле неможливо: Access використовує його винятково для перевірки умови до обчислень. При необхідності виведення поля на екран його необхідно продублювати в бланку запиту з опцією **Групування**.

Поле: Имя таблицы: Групповая операция: Сортировка: Вывод на экран: Условие отбора: или:	Название	Цена	Цена	МинимальныйЗапас	МинимальныйЗапас
	Товары	Товары	Товары	Товары	Товары
	Группировка	Sum	Avg	Группировка	Условие
	☒	☒	☒	☒	☐
					<200

Рисунок 3.2 - Визначення критерію відбору даних до обчислень

При необхідності фільтрації підсумкових даних критерії відбору задаються у відповідних стовпчиках бланка запиту, у яких задані опції **Групування** або конкретні підсумкові функції (рисунок 3.3). У термінах **SQL** така умова задається оператором **Having**.

Поле: Имя таблицы: Групповая операция: Сортировка: Вывод на экран: Условие отбора: или:	Название	Цена	Цена	МинимальныйЗапас	
	Товары	Товары	Товары	Товары	
	Группировка	Sum	Avg	Группировка	
	☒	☒	☒	☒	☐
	Like "крупн"	>1000			

Рисунок 3.3 - Визначення критерію відбору даних після обчислення

Примітка

Якщо в запиті використовуються агрегатні функції, то оператор **Where** виконується перед ними, тоді як оператор **Having** застосовується після розбивки на групи й обчислення значень функцій для кожної групи.

Увага

Результати виконання операторів **Where** і **Having** можуть відрізнятися. Це пов'язано з тим, що **Where** відсіває рядки перед угрупованням даних, а **Having** - після.

Створення виразу для групових операцій

У Access користувач має можливість створювати власні вирази. Для цього в порожній чарунці рядку **Поле** потрібно записати вираз, а в рядку **Групова операція** даного стовпчика вибрати опцію **Вираз**. На рисунку 3.4 показаний приклад упорядкування запиту, що визначає уцінку товару на складі (усіх видів фурнітури) як 20% від добутку ціни товару на кількість. При цьому такі обчислення провадяться тільки для тих товарів, що надійшли на склад до 30.12.04.

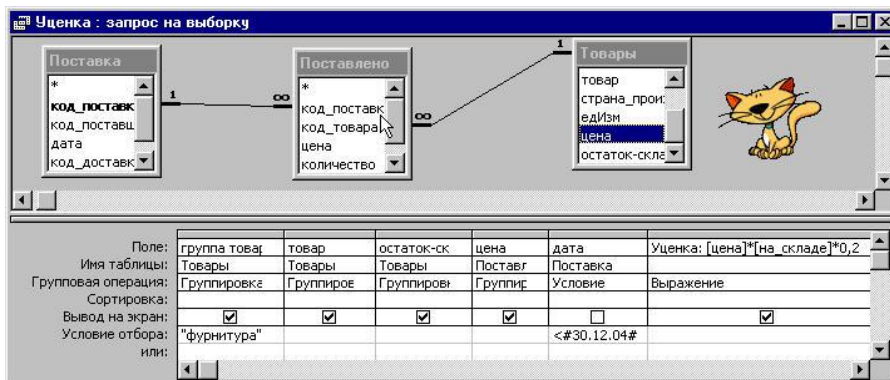


Рисунок 3.4 - Запит із використанням виразу в груповій операції

3.3.2 Створення перехресних запитів

Перехресний запит – це вибірка даних, записаних у вигляді двовимірної матриці, що створена з таблиці Access. Цей запит представляє визначені дані з обраних полів у форматі, схожому на формат електронної таблиці.

Для створення перехресного запиту необхідно визначити, як мінімум, три параметри:

- поле заголовків рядків;
- поле заголовків стовпчиків;
- поле для вибору значень.

Припустимо, необхідно створити запит, який відображав би в результуючій таблиці типи товарів як заголовки рядків, клієнтів як заголовки стовпчиків, а в кожній чарунці таблиці містилися б дані про

кількість поставленого товару кожному клієнту. На рисунку 3.5 наведений приклад конструювання даного запиту, результат виконання запиту показаний на рисунку 3.6.

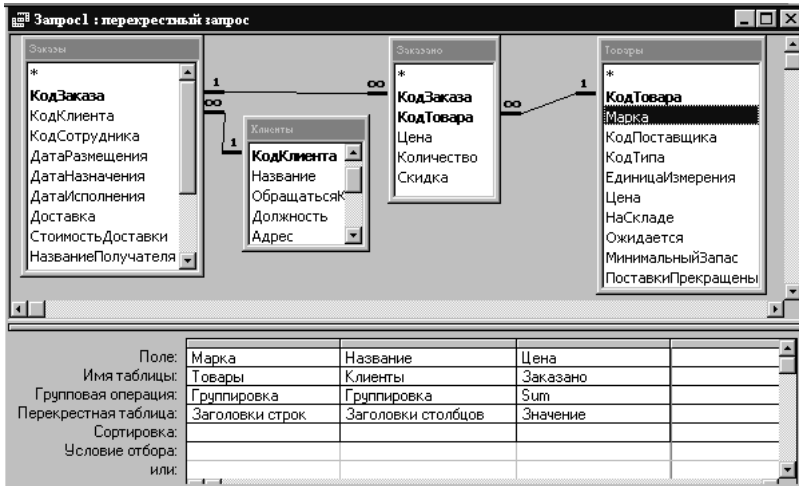


Рисунок 3.5 - Створення перехресного запиту

Запрос1 : перекрестный запрос

	Марка	4-й роддом	АО ЗВА	АО ЭКЗ	АО ЭТЗ	Школа
▶	орех		0,60			
	Йогурт сливочный		2,80		2,80	
	Крабовые палочки	8,80				8,80
	Крупа гречневая	1,05		1,05	1,05	
	Крупа кукурузная			0,55		
	Крупа пшеничная				0,50	
	Крупа ячневая		0,35		0,35	
	Макароны		0,85	0,85		
	Минтай мороженный		4,50	4,50		
	Молоко	1,20				
	Молоко сгущеное			4,00		4,00
	Окорочка			7,20		
	Рис	1,90		1,90		1,90

Запись: 1 из 17

Рисунок 3.6 - Результат виконання перехресного запиту



Порада

Перехресний запит може мати декілька заголовків рядків і тільки один заголовок стовпчиків. Якщо необхідно створити запит з одним заголовком рядків і декількома заголовками стовпчиків, поміняйте місцями рядки та стовпчики.

Визначення критеріїв для перехресного запиту

Критерій може бути визначений для таких типів полів:

- поле Заголовка стовпчика;
- поле Заголовка рядка;
- нове поле.

Крім критеріїв відбору для стовпчиків і рядків Access дозволяє ввести критерій відбору для поля, не задіяного безпосередньо в перехресному запиті. Проте при цьому чарунка ***Перехресна таблиця*** для даного стовпчика повинна залишатися порожньою. Якщо це не так, необхідно вибрати опцію (***не відображається***) із списку, що розкривається.



Порада

Визначити умову відбору у полі, використовуваному в якості значень, не можна. Проте, якщо це необхідно, скопіюйте це поле й опишіть умову в ньому, як у нове поле.

Майстер перехресного запиту

Майстер викликається у вікні ***Новий запит*** і дозволяє створити перехресний запит на підставі інформації про:

- початкову таблицю або запит;
- поля для заголовків рядків;
- поля для заголовків стовпчиків;
- полях для значень;
- заголовок.

При використанні цього майстра необхідно враховувати декілька обмежень.

- Якщо для створення перехресного запиту використовується більш однієї таблиці, необхідно створити окремі запити для таблиць. (*Підказка*: створіть запит для таблиць, обравши всі необхідні поля, а потім збережіть цей допоміжний запит, після чого можете використовувати його як джерело записів для майстра..)
- Не можна визначати обмежуючі умови для запиту, створеного майстром. (*Підказка*: після створення майстром запиту модифікуйте його.)
- Не можна вказувати заголовки стовпчиків і порядок їхнього слідування. (*Підказка*: після створення майстром запиту модифікуйте його. Для цього виберіть команду **Вид⇒Властивості** та у полі **Заголовки стовпчиків** уведіть ці заголовки в потрібному порядку.)

3.3.3 Створення параметричних запитів

Параметричний запит – це запит, при виконанні якого користувачу пропонується ввести значення якогось параметра, з урахуванням котрого й виконується запит. Іншими словами, це завдання одного або декількох критеріїв відбору, але не в режимі конструктора таблиць, а динамічно в процесі виконання запиту.

Наприклад, створений запит, що відображає інформацію про товари, наявні на складі. Якщо цей запит використовується часто, то можна створити параметричний запит, що запитував би користувача про назву товару. Для створення такого запиту необхідно в наявний запит на вибірку в стовпчику **Назва_товару** (наприклад) у чарунку **Умова відбору** ввести потрібну фразу. Наприклад, [**Введіть назву товару**].



Порада

Ви можете створити параметричні запити з використанням будь-якого припустимого оператора, включаючи оператор **Like** із шаблонами. Наприклад, параметр **Like [ведіть назву товару або <Enter>]&*** дозволяє користувачу запустити запит для одного товару або для всіх.

Аналогічно створюються багатопараметричні запити. При цьому варто мати на увазі, що вводити потрібно коректні значення, інакше в результуючій таблиці не виявиться жодного запису.

3.3.4 Створення запитів на зміну

Запити на зміну дозволяють змінювати значення полів записів:

- вилучити записи з таблиці або групи таблиць;
- додати записи з однієї таблиці в іншу;
- оновити інформацію в групі записів;
- створити таблиці.



Порада Завжди зберігайте резервну копію даних перед виконанням запиту на оновлення

Створення запитів на зміну схоже на створення запитів на вибірку. Так саме необхідно вибрати поля для запиту та зазначити умови відбору. Крім того, необхідно вказати визначені властивості цих запитів: *Додавання, Створення таблиці, Оновлення* чи *Вилучення*.

Створення запитів на оновлення

Для створення запиту на оновлення виконайте такі дії:

- створіть запит на вибірку; перегляньте необхідні дані в режимі таблиці;
- перетворіть запит на вибірку в запит на оновлення. Для цього потрібно вибрати пункт меню *Запит* ⇨ *Оновлення*. При цьому Access додасть рядок властивостей *Оновлення* у вікні конструктора запитів. У цьому рядку потрібно зазначити необхідні зміни.

🔗 Примітка

Зміни можна задавати для декількох полів відразу, включаючи поля, що використовуються для завдання умов відбору.

Побудова запиту на створення таблиці

Запит на зміну можна застосувати для створення нових таблиць (на основі наявних), використовуючи умову відбору записів. Для цього необхідно:

- створити новий запит, використовуючи необхідні таблиці;
- вибрати пункт **Запит**⇒**Створення таблиці** та у діалоговому вікні, що з'явилося, задати назву таблиці, що формується;
- вибрати необхідні поля з наявних таблиць;
- зазначити умови відбору;
- запустити запит на виконання (кнопка **Запуск** або пункт **Запит**⇒**Запуск**).

Примітка

При створенні таблиці за допомогою запиту типи даних і розміри полів зберігаються такими ж, якими вони були в базовій таблиці запиту. Проте ніякі інші властивості полів і таблиці не успадковують!

Побудова запиту на додавання записів

Даний тип запиту дозволяє додавати записи в зазначену (обов'язково існуючу) таблицю, причому як у таблицю в поточній базі даних, так і в будь-яку іншу базу даних Access. Використання такого запиту можливо, наприклад, коли необхідно з наявної базової таблиці перенести інформацію в архівну таблицю з наступним їхнім вилученням із базової таблиці.

При роботі з цими запитамі необхідно пам'ятати наступне:

- Якщо таблиця, у яку додаються записи, має ключове поле, то записи, що додаються, не можуть мати значення **Null** або повторювані значення в цьому полі.
- При додаванні записів до таблиці, розташованої в іншій базі даних, необхідно знати точне місце розташування та назву цієї бази даних.
- При використанні символу "*" у рядку конструктора запитів, використовувати окремі поля з цієї таблиці не можна - це сприймається як спроба додавання одного й того ж запису двічі.

- Якщо додаються записи з полем *Лічильник* (ключове поле, створене Access), додавати саме поле *Лічильник* не слід (за умови, що в таблиці є це поле (див. п. 1)).

Побудова запиту на додавання аналогічна побудові запиту на створення таблиці – спочатку необхідно створити запит на вибірку й переконатися, що вибираються потрібні записи, а потім перетворити його в запит на додавання. Після виконання запиту на додавання необхідно відчинити таблицю (таблиці), до якої додані записи, й перевірити результати.

Побудова запиту на вилучення записів

З усіх запитів на зміну запит на вилучення найбільше небезпечний. На відміну від інших запитів він вилучає записи раз і назавжди.

Подібно іншим запитам на зміну, запит на вилучення працює з групами записів на підставі умов відбору записів. Запит на вилучення може одночасно вилучати записи з декількох таблиць. Проте для вилучення пов'язаних записів із декількох таблиць необхідно виконати наступні дії:

- визначити зв'язки між таблицями у вікні *Схема даних*.
- встановити опцію *Забезпечення цілісності даних* для заданого типу зв'язку між таблицями.
- встановити опцію *Каскадне вилучення пов'язаних записів* для заданого типу зв'язку між таблицями.

При роботі зі зв'язками *один-до-багатьох* без завдання зв'язків і без умикання параметра *Каскадне вилучення пов'язаних записів* Access вилучить записи тільки з однієї таблиці. Порядок такий: Access спочатку вилучає записи з боку *багато* цих зв'язків, а потім вам належить вилучити записи з боку *один*.



Увага

Оскільки запит на вилучення робить необоротний руйнуючий вплив на таблиці, перед початком роботи з ними завжди варто виконувати архівне копіювання таблиць.

Примітка

Запит на вилучання цілком вилучає записи, а не тільки зазначені поля даних. Для вилучання даних з окремих полів необхідно використовувати запит на оновлення, щоб замінити значення цих полів на порожні значення (**Null**).

Контрольні питання

- ? Поняття групового запиту.
- ? Категорії групових запитів.
- ? Що називається підсумковими функціями, для чого вони використовуються?
- ? Склад і призначення підсумкових функцій.
- ? Визначення критеріїв для групового запиту.
- ? Призначення операторів **Where** та **Having**.
- ? Створення виразу для групових операцій.
- ? Послідовність створення перехресного запиту.
- ? Як задаються критерії для перехресного запиту.
- ? Робота з *Майстром перехресного запиту*.
- ? Призначення та технологія створення параметричного запиту.
- ? Типи запитів на зміну.
- ? Створення запитів на оновлення.
- ? Побудова запиту на створення таблиці.
- ? Побудова запиту на додавання записів.
- ? Побудова запиту на вилучання записів.
- ? Майстер запитів на пошук повторюваних записів.
- ? Майстер запитів на пошук записів без підпорядкованих записів.
- ? Проблеми, що виникають при роботі з запитами на зміну, та можливі шляхи їхнього вирішення.

4 ЛАБОРАТОРНА РОБОТА № 4 "СТВОРЕННЯ ТА ВИКОРИСТАННЯ ФОРМ ДАНИХ"

4.1 Мета роботи

Метою роботи є придбання практичних навичок створення й використання форм даних.

4.2 Завдання до лабораторної роботи

4.2.1 Ознайомитися зі змістом пункту 4.3 методичних указівок.

4.2.2 Для таблиць і запитів, створених у попередніх лабораторних роботах, за узгодженням із викладачем розробити різноманітні типи форм уведення та відображення даних.

4.3 Основні теоретичні відомості

Форми зручні для перегляду, введення, зміни або видалення форматуваних даних. Режим форми забезпечує максимальну гнучкість перегляду даних, дозволяючи відображати одночасно усі поля одного або декількох записів. У Access використовуються шість основних типів форм:

- у стовпчик (називається також повноекранною формою);
- стрічкова;
- таблична;
- головна/підпорядкована;
- зведена таблиця;
- діаграма.

Форми можна створювати трьома способами:

- автоматично за допомогою Автоформи;
- напівавтоматично за допомогою Майстра форм;
- вручну, використовуючи режим Конструктора форм.

4.3.1 Створення форми за допомогою Автоформи

Щоб швидко створити форму, можна скористатися кнопкою **Створити**, а потім у діалоговому вікні, що з'явилося, вибрати один із варіантів **Автоформи**, а також таблицю або запит, на підставі яких створюється форма. У цьому випадку форма з'являється відразу ж, без яких-небудь додаткових дій.

4.3.2 Створення форми за допомогою майстра форм

Після вибору опції **Майстер форм** потрібно вибрати поля, зовнішній вигляд форми та стиль форми. При цьому можливі наступні варіанти зовнішнього вигляду форми:

- в один стовпчик;
- стрічкова;
- таблична;
- вирівняна.

На рисунках 4.1-4.4 наведені приклади форм, створених на підставі таблиці "Товари".

The image shows a screenshot of a software window titled "Товар" (Goods). The window contains a form with the following fields and values:

Field	Value
Код	40
Наименование	3D-акселератор
Группа	мультимедиа
Цена	134,00 грн.
КолСклад	100
МинЗапас	12

At the bottom of the window, there is a status bar with the text "Запись:" followed by navigation icons, the number "1", more navigation icons, and the text "из 32".

Рисунок 4.1 – Зразок одностовпчикової форми

код	група	товар	страна	ед. изм.	цена	остаток
1	фурнитура	Ручка дверная	Польша	шт	24,00грн	1
2	фурнитура	Ручка для шкафа	Италия	шт	5,00грн	5
3	фурнитура	Петли стальные	Турция	шт	6,00грн	5
5	фурнитура	Замок дверной	Украина	шт	25,00грн	16
6	фурнитура	Замок гаражный	Украина	шт	35,00грн	106
7	лакокрасоч	Эмаль белая	Украина	1 кг	22,00грн	40
8	лакокрасоч	Эмаль белая	Польша	1 кг	26,00грн	65
9	инструмент	Молоток 400гр	Белоруссия	шт	5,00грн	34
10	инструмент	Молоток 600гр	Белоруссия	шт	7,00грн	54
11	инструмент	Зубило	Белоруссия	шт	3,00грн	23

Записи: 1 из 25

Рисунок 4.2 – Зразок стрічкової форми

код тов	група товара	товар	страна произ	едизм	цена	остаток
1	фурнитура	Ручка дверная	Польша	шт	24,00грн	1
2	фурнитура	Ручка для шкафа	Италия	шт	5,00грн	5
3	фурнитура	Петли стальные	Турция	шт	6,00грн	5
5	фурнитура	Замок дверной врезной	Украина	шт	25,00грн	16
6	фурнитура	Замок гаражный	Украина	шт	35,00грн	106
7	лакокрасочные	Эмаль белая	Украина	1 кг	22,00грн	40
8	лакокрасочные	Эмаль белая	Польша	1 кг	26,00грн	65
9	инструменты	Молоток 400гр	Белоруссия	шт	5,00грн	34
10	инструменты	Молоток 600гр	Белоруссия	шт	7,00грн	54
11	инструменты	Зубило	Белоруссия	шт	3,00грн	23
12	инструменты	Плоскогубцы	Белоруссия	шт	8,00грн	76
13	инструменты	Рубанок малый	Украина	шт	15,00грн	54
14	инструменты	Рубанок большой	Украина	шт	24,00грн	43
15	инструменты	Рубанок средний	Украина	шт	20,00грн	45

Записи: 1 из 25

Рисунок 4.3 – Зразок табличної форми

код товара		
1		
група товара	товар	страна производитель
фурнитура	Ручка дверная	Польша
ед. изм.	цена	остаток-склад
шт	24,00грн	1

Записи: 1 из 25

Рисунок 4.4 – Зразок вирівняної форми

Використання останнього типу форми доцільне при великій кількості атрибутів або коли потрібно переглядати один рядок таблиці.

4.3.3 Елементи керування та їхні властивості

У Access *елементом керування* (табл. 4.1) називається будь-який елемент форми або звіту, наприклад, надпис або поле.

Типи елементів керування

Таблиця 4.1 - Елементи керування форм і звітів

Елемент керування	Призначення
1	2
<i>Основні елементи керування</i>	
Надпис	Служить для відображення тексту
Поле	Дозволяє вводити дані
<i>Розширені елементи керування введення даних</i>	
Група перемикачів	Містить декілька вимикачів, перемикачів або прапорців
Вимикач	Може знаходитися в двох станах: натиснутому або віджатому, нагору або униз. Зазвичай на ньому розміщують картинку або піктограми
Перемикач	Відображається у вигляді кружка з точкою усередині, що позначає активізацію цього елемента. Його також називають <i>радіокнопкой</i>
Прапорець	Може знаходитися в двох станах. Відображається у вигляді квадрата з галочкою усередині, що позначає активізацію цього елемента
Поле зі списком	Список значень, що розкривається, у який можна вводити й інші значення
Список	Список значень, що відображаються у формі або звіті
Кнопка	Звичайно використовується для виклику макроса або запуску програми мовою Basic для виконання якоїсь дії
Підпорядкована форма	Призначений для відображення ще однієї форми або звіту у вихідній формі або звіті
Набір вкладок	Дозволяє відображати декілька сторінок у вигляді папки з вкладками

Продовження таблиці 4.1

1	2
Елементи керування графікою	
Рисунок	Відображає на екрані растрове зображення, використовуючи незначні системні ресурси
Вільна рамка об'єкта	Містить об'єкт OLE або рисунок, що не пов'язаний із полем таблиці. Крім того, може містити діаграми, звук і відеозображення
Приєднана рамка об'єкта	Містить об'єкт OLE або рисунок, пов'язаний із полем таблиці.
Лінія	Лінія, товщину та колір якої можна змінювати. Використовується для поділу елементів керування
Прямокутник	Прямокутник, розміри та колір якого можна змінювати. Може бути порожнім або заповненим і використовуватися для виділення елементів форм
Розрив сторінки	Зазвичай використовується в звітах і позначає початок нової сторінки

Елементи керування можна створити за допомогою спеціальної панелі (рис. 4.5), що викликається командою **Вид⇒Панель елементів** або за допомогою кнопки **Панель елементів** .

Приєднані, вільні та обчислювані елементи керування

Існують три основних типи елементів керування:

- приєднані;
- вільні;
- обчислювані

Приєднаними елементами керування називають елементи, що пов'язані з полем таблиці. При введенні значення в приєднаний елемент керування поле таблиці в поточного запису автоматично оновлюється.

Вільні елементи керування (перемінні, або перемінні пам'яті) зберігають уведений розмір без відновлення поля таблиці. Їх можна використовувати для відображення тексту, значень, що повинні бути передані макросам, ліній і прямокутників. Крім того, вони призначені для зберігання об'єктів OLE (наприклад, малюнків), що утримуються

не в таблиці, а в самій формі.

Обчислювані елементи керування створюють на основі таких виражень, як функції і формули. Оскільки вони не приєднані до полів таблиці, вони не обновлюють вміст полів таблиці.



Рисунок 4.5 - Панель елементів

Використання елементів керування

Вимикачі, перемикачі та прапорці

Використовуються для відображення даних логічного типу. Вид цих елементів можна змінювати (рис. 4.6) за допомогою кнопки **Оформлення** панелі інструментів або відповідного пункту контекстного меню.

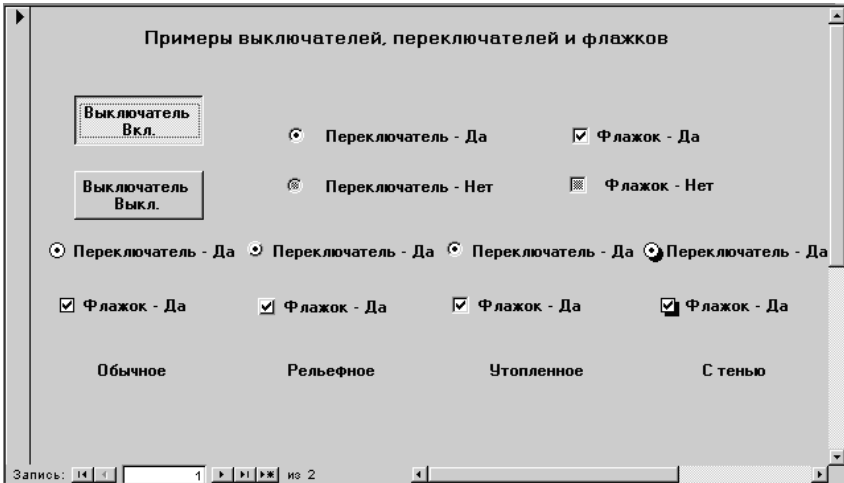


Рисунок 4.6 - Приклади оформлення вимикачів, перемикачів і прапорців

Група

У групі може знаходитися декілька вимикачів, перемикачів або прапорців. При цьому всі елементи керування усередині рамки групи функціонують узгоджено. Вони не просто подають дані логічного типу, а повертають число, що відповідає їхньому положенню в групі. Причому тільки один елемент керування можна активізувати усередині групи (у групі можна використовувати не більш 4 елементів) (рисунк 4.7,а).

 **Порада**

При необхідності використання більшого числа елементів можна використовувати поле зі списком.

Група зазвичай пов'язана з одним полем таблиці або виразом.

Якщо необхідно здійснити вибір із списку варіантів, що не є взаємовиключними, то ці елементи керування укладають у рамку (рис. 4.7,б).

Квартал

☒ I квартал

☐ II квартал

☐ III квартал

☐ IV квартал

Комплект поставки

☒ Компьютеры

☒ Комплектующие

☒ Офисное оборудование

☒ Программное обеспечение

а)
б)

Рисунок 4.7 - Приклад використання групи (а) та індивідуальних елементів керування (б)

Вкладки

При створенні набору вкладок по умовчання створюються дві вкладки. На кожній вкладці можна розташовувати свої елементи керування (рис. 4.8). Для додавання, вилучання або заміни місцями вкладок варто скористатися відповідними опціями в контекстному меню.

Рисунок 4.8 – Приклад використання вкладок

4.3.4 Створення та редагування форм у режимі Конструктора

Конструктор форм використовується, коли потрібно вручну створити форму на підставі певної таблиці (запиту) або коли створюється довільна форма, не пов'язана з певною таблицею БД.

У першому випадку треба вибрати пункт ***Створити*** вікна бази даних, у вікні нової форми вибрати режим ***Конструктор*** і вказати потрібну таблицю або запит.

У другому випадку вибирається пункт ***Створення форми в режимі конструктора***. Далі за допомогою елементів керування створюється відповідна форма.

Розміщення полів у формі

Для розміщення елементів керування у формі потрібно виконати три основні етапи:

- відобразити вікно списку полів за допомогою щиглика на кнопці ***Список полів*** панелі інструментів;
- визначити тип створюваного елемента керування за допомогою щиглика на потрібній кнопці панелі елементів;
- вибрати потрібні поля таблиці (запиту) та перетягнути їх у вікно конструктора форм (аналогічно побудові запиту). При цьому можна перетаскувати відразу групу полів, попередньо виділивши їх звичайним чином.

Додавання фонового рисунка

Для додавання фонового рисунка потрібно:

- вибрати форму, клацнувши в її верхньому лівому куті (у місці перетинання двох лінійок).

- викликати вікно властивостей і клацнути на властивості

Рисунок.

- ввести шлях до файлу з рисунком.
- клацнути на властивості ***Масштаби рисунка*** та задати нове значення.

Властивості форм

Щоб змінити властивості самої форми, клацніть правою кнопкою миші на області виділення форми чи звіту (прямокутник на перетинанні лінійок ліворуч угорі). У контекстному меню виберіть команду **Властивості**. Тоді відкриється вікно **Форма**, у якому можна задати усі властивості форми (рис. 4.9).

Зміна властивостей форми виконується наступним чином:

1. Відкрийте вікно **Форма**, якщо воно ще не відкрито.
2. Клацніть на вкладці, що відноситься до властивості, яка змінюється. Нижче проводиться список і короткий опис вкладок у вікні властивостей форми (рисунок 4.10).

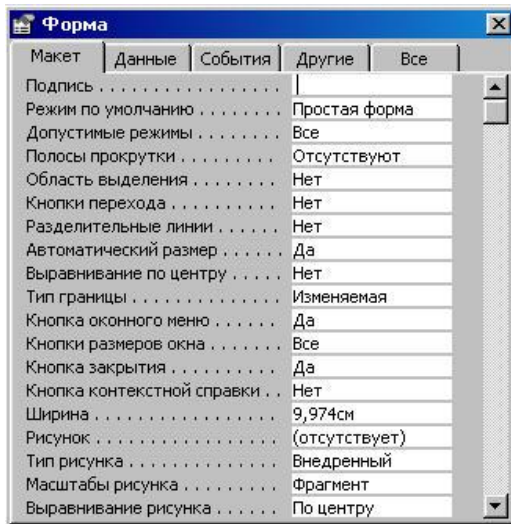


Рисунок 4.9 – Вікно властивостей форми

– **Макет**. На цій вкладці приведені властивості, що відносяться до усієї форми й стосуються її вигляду. Деякі властивості на вкладці **Макет** змінюються автоматично при зміні розміру форми за допомогою миші.

– **Дані.** На цій вкладці зазначене джерело даних, стандартні величини та діапазони значень

– **Події.** На цій вкладці можна контролювати, що відбувається після визначеної події. Наприклад, подія *Натискання кнопки* визначає, що буде після щиглика користувачем на формі.

– **Інші.** На цій вкладці містяться загальні властивості. Тут можна включати та відключати застосування деяких засобів, наприклад, тих, що відносяться до автокоректування.

– **Всі.** На цій вкладці всі перераховані вище категорії розташовуються в одному великому списку властивостей.

Додавання до форми елементів керування

Перелік елементи керування наведений у таблиці 4.1. Для розміщення елемента керування на формі треба спочатку вибрати його за допомогою миші на панелі елементів (рисунок 4.5).

При цьому покажчик миші перетворюється в перехрестя. За його допомогою елемент керування перетягується в потрібне місце на сітці. Потім при натиснутій лівій кнопці миші шляхом протягання потрібно задати розміри елемента керування.

Елемент керування **Кнопка** є, мабуть, найбільш розповсюдженим. Нижче описане додавання до форми кнопки та використання відповідного Майстра:

1. Виберіть на панелі елемент керування **Кнопка**.
2. Розташуєте елемент керування на формі. При цьому запускається *Майстер кнопка* (рисунок 4.10).
3. В лівому списку виберіть категорію дії, що повинна виконувати кнопка.
4. В правому списку виберіть необхідну дію та клацніть на кнопці **Далі**.
3. Введіть необхідний текст напису на кнопці або виберіть для неї необхідний рисунок.
4. Введіть ім'я кнопки та клацніть на кнопці **Готово**.

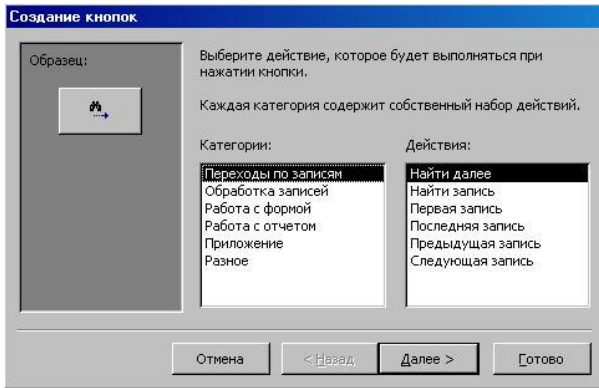


Рисунок 4.10 – Вікно Майстра кнопок

Завдання властивостей елемента керування

Для кожного елемента керування є таблиця властивостей, подібна з таблицею властивостей форми. На рис. 4.11 показане вікно властивостей елемента керування **Кнопка**.

Рисунок 4.11 – Вікно властивостей елемента керування *Кнопка*

Вікна властивостей різних елементів дещо відрізняються друг від друга, але в цілому їхня структура подібна.

4.3.5 Робота з підлеглими формами

Підлеглими формами називають форми, що розміщені всередині інших форм. Це дає можливість переглядати на одному екрані значно більше інформації. Форму, що містить всередині себе підлеглу форму, називають головною.

Підлеглу форму створюють окремо від головної. Вона приводиться в списку на вкладці форми вікна бази даних. Зв'язок між таблицями, на яких створена як головна, так і підлегла форми, забезпечує синхронізацію цих форм.

Приведемо деякі розуміння, корисні при роботі з підлеглими формами;

1. Таблиці або запити, що служать джерелом даних форми, зазвичай містять зв'язки типу «один-до-багатьох». Таблиця чи запит, що відноситься до підлеглої форми, є на боці «до багатьох» даного зв'язку. Таблиця ж або запит, які відповідають головній формі, знаходяться на боці «один».

2. Підлегла форма повинна бути створена в першу чергу. Перед переміщенням її в головну форму вона повинна бути збережена.

3. У форму можна помістити кілька підлеглих форм.

4. Підлеглі форми можуть бути вкладеними. Інакше кажучи, в підлеглий формі також може знаходитися підлегла форма.

5. У головній формі може знаходитися не більш двох підлеглих форм, що містять у собі підлеглі форми.

Щоб створити підлеглу форму, виконаєте наступне:

1. Відкрийте головну форму в режимі **Конструктор**.
2. На панелі інструментів виберіть елемент керування **Підлегла форма** та перетягніть його в потрібне місце головної форми.
3. Далі дійте згідно вказівок **Майстра підлеглих форм**.
4. Збережіть і закрийте головну форму.

Підпорядкування форм можна також зробити за допомогою майстра форм. Для цього необхідно виконати наступні дії:

1. Викликати **Майстра форм**.
2. В якості джерела даних вказати головну таблицю (з боку "одиниці" у зв'язку **I:M**). Вибрати необхідні поля даних.
3. Повернутися до переліку таблиць і вказати підлеглу таблицю. Вибрати необхідні поля даних.
4. Вибрати спосіб представлення підлеглої таблиці.
5. Вибрати стиль оформлення всієї форми.
6. Вказати найменування головної та підлеглої форми.

На рисунку 4.12 наведений приклад використання головної форми **Постачання** та підлеглої форми **Склад постачання**.

Поставка

код поставки: 1 дата: 07.01.04

поставщик: РостСельМаш метод доставки: Укртранс

Сумма поставки: 1 250,00грн Телефоны поставщиков

Статистика поставок Корректировка товаров

Товары в поставке

товар	кол-во	цена	сумма
Замок дверной врезной	40	25	1 000,00грн
Ручка для шкафа	50	5	250,00грн
Петли стальные	10	6	60,00грн
Эмаль белая	15	22	330,00грн

Запись: 4 из 4

1. Корректировка складских запасов 2. Сумма поставки

Запись: 1 из 7

Рисунок 4.12 – Приклад використання підлеглої форми

Контрольні питання

- ? Які основні типи форм використовуються в Access?
- ? Які способи створення форм існують?
- ? Як автоматично створити форму документа?
- ? Які існують основні типи елементів керування?
- ? Що таке приєднані, вільні та обчислювані елементи керування.
- ? Способи вирівнювання елементів керування.
- ? Властивості елементів керування.
- ? Як змінити послідовність переходу між елементами керування?
- ? Додавання фонового рисунку до форми.
- ? Властивості форм.
- ? Створення підлеглої форми.
- ? Створення елемента керування *Кнопка*.

5 ЛАБОРАТОРНА РОБОТА № 5 "РОЗРОБКА ЗВІТІВ"

5.1 Мета роботи

Метою роботи є придбання навичок створення та редагування звітів у середовищі MS Access.

5.2 Завдання до лабораторної роботи

5.2.1 Ознайомитися зі змістом пункту 5.3 даних методичних указівок.

Створити звіти та відредагувати їх у відповідності до індивідуального завдання

5.3 Основні теоретичні відомості

Звіт – це засіб перегляду та друку підсумкової інформації. В ньому можна групувати та сортувати дані в будь-якому порядку, отримувати підсумкові значення, середні значення та інші статистичні величини, а також розміщувати графічні діаграми.

Головна відміна форм від звітів – це їхнє призначення. Якщо форми переважно використовуються для зручності введення даних, то звіт – для їхнього перегляду (на екрані чи на папері).

Все, що можна зробити з формою (за виключенням уводу даних), можна зробити й зі звітом. Наприклад, форму можна зберегти у вигляді звіту, а потім змінити властивості елементів керування у вікні **Конструктора звітів**.

5.3.1 Типи звітів

За допомогою Access можна створити наступні типи звітів:

Стрічкові звіти	дані роздруковуються у вигляді рядків і стовпців із групуванням і підсумками. Їхні різновиди включають звіти з підсумковим і груповим підсумовуванням.
Звіти в стовпець	дані роздруковуються у виді форми з можливим включенням підсумків і графіків.

Діаграми створюються звіти у вигляді різноманітних діаграм.

Поштові наклейки створюються наклейки для конвертів.

Стрічковий звіт

Він схожий на таблицю. Для поділу даних у стрічкових звітах можна використовувати номери сторінок, дати складання звітів, лінії та прямокутники. В них можуть використовуватися обрамлення й заливання, рисунки, ділова графіка та МЕМО-поля.

Звіти в стовпець

Вони відображають одну чи кілька записів на сторінці по вертикалі. В нього можуть входити розділ інформації з записів однієї таблиці та розділ з декількома зв'язаними записами з іншої таблиці.

5.3.2 Розробка звітів за допомогою Майстра

Щоб створити звіт за допомогою **Майстра**, потрібно:

1. Відкрити базу даних.
 2. Вибрати вкладку **Звіт**.
 3. Вибрати команду **Створити**.
 4. Клацнувши на стрілці, розкрити список і вибрати таблицю чи запит, по якому буде створюватися звіт.
 5. Клацнувши по кнопці **Майстер**, вивести на екран діалогове вікно **Майстра звітів**.
 6. Вибрати необхідні для звіту поля зі списку **Наявні поля**, перемістити їх у список полів звіту кнопкою ">", а всі поля відразу — кнопкою ">>". Вилучити окреме поле з обраного списку можна кнопкою "<", а всі поля - кнопкою "<<".
 7. У наступному вікні **Майстра** вибрати при необхідності поля для додаткових рівнів групування.
 8. Далі вказати режим сортування. В цьому ж вікні, якщо необхідно підрахувати підсумкові значення по окремим числовим полям, слід натиснути клавішу **Підсумки** та у вікні, що з'явилося (рис.5.1), задати необхідні дії для потрібних полів.
- Тут можна також відобразити або сховати дані в розділі даних.

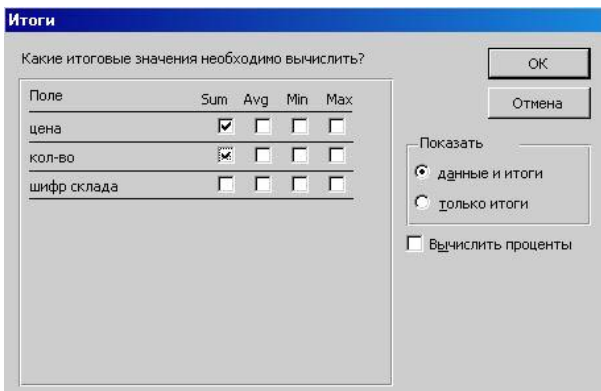


Рисунок 5.1 – Вікно завдання підсумків

9. Наступне вікно призначене для завдання макету даних і орієнтації сторінок звіту.

10. У наступному вікні треба вибрати необхідний стиль оформлення звіту.

11. Задати ім'я звіту. По умовчанням пропонується ім'я таблиці або звіту, на базі яких формувався звіт.

12. Переглянути звіт у режимі попереднього перегляду.

Зазвичай звіт, створений за допомогою **Майстра**, потребує подальшого редагування вручну в режимі **Конструктора**.

5.3.3 Автозвіт

Створити звіт без питань **Майстра** можна за допомогою кнопки **Створити**⇒**Новий об'єкт**⇒**Автозвіт**. Інший метод міститься у використанні команди **Вставка**⇒**Звіт** і щиклику мишею на одному з двох варіантів вибору автозвіту в діалоговому вікні, що з'явиться. При використанні кнопки **Автозвіт** звіт з'являється відразу, без додаткових зусиль із вашого боку.

5.3.4 Розробка звітів за допомогою Конструктора

Панель інструментів **Конструктора звітів** відрізняється від панелі форматування й призначена для виконання інших задач.

Наприклад, для вибору шрифту чи виду вирівнювання слід використовувати панель форматування.

Призначення окремих кнопок панелі інструментів наведено в таблиці 5.1.

Таблиця 5.1 – Елементи панелі інструментів *Конструктора звітів*

Кнопка панелі інструментів	Призначення
Вигляд	Відображає в списку, що розкривається, три типи представлення звіту
Зберегти	Зберігає поточний макет звіту
Друк	Друкує форму, таблицю, запит або звіт
Попередній перегляд	Включає режим попереднього перегляду
Орфографія	Перевіряє орфографію обраної області чи документа
Вилучити до буфера	Вилучає фрагмент документа до буфера
Копіювати до буфера	Копіює обраний об'єкт до буфера обміну
Вставити з буфера	Копіює вміст буфера обміну в документ
Формат за зразком	Копіює формат одного елемента керування в інший
Скасувати	Скасовує попередню дію
Список полів	Відкриває/закриває вікно списку полів
Панель елементів	Відкриває чи закриває панель елементів
Сортування та групування	Відкриває чи закриває діалогове вікно Сортування та Групування
Автоформат	Застосовує готові стилі до форми чи звіту
Програма	Відкриває чи закриває вікно модуля
Властивості	Відображає вікно властивостей для обраного елемента
Побудувати	Викликає Майстра чи Будівника для обраного елемента
Вікно бази даних	Викликає вікно бази даних
Новий об'єкт	Створює новий об'єкт
Помічник	Надає довідку

Записи в звіті зазвичай обробляються послідовно. У залежності від макета звіту елементи даних обробляються по-різному. Звіти поділяються на розділи (табл. 5.2). Заголовок звіту друкується на початку звіту на титульній сторінці.

Таблиця 5.2 – Призначення розділів звіту

Розділ	Призначення
Верхній колонтитул	Друкується вгорі кожної сторінки
Заголовок групи	Друкується перед обробкою кожної групи
Область даних	Друкується кожен запис таблиці або динамічного набору даних запиту
Примітка групи	Друкується після обробки останнього запису групи
Нижній колонтитул	Друкується внизу кожної сторінки
Примітка звіту	Друкується наприкінці звіту після обробки всіх записів

Першим кроком на шляху створення звіту є створення його *макета*. Макет створюється вручну. В ньому вказуються розміщення основних елементів звіту. Далі відповідно до макета підбираються дані з таблиць. Якщо для підбора даних необхідно створити запит, то спочатку створюється запит, а потім він зв'язується з новим звітом.

Для того, щоб зв'язати звіт із запитом:

1. Натисніть клавішу **F11** для відображення вікна БД якщо воно ще не відображено.
2. Клацніть на вкладці **Звіти**
3. У вікні БД клацніть на кнопці **Створити**. При цьому з'явиться діалогове вікно **Новий звіт**.
4. У діалоговому вікні **Новий звіт** виберіть опцію **Конструктор**.
5. Клацніть на стрілці списку, що розкривається, з написом **"Виберіть як джерело даних таблицю чи запит"**. При цьому на екрані з'явиться список всіх таблиць та запитів поточної БД.
6. Виберіть необхідний запит і клацніть на кнопці **ОК**.
7. Розгорніть до максимального розміру вікно конструктора звітів.

Після цього з'явиться порожнє вікно конструктора звітів. Зверніть увагу на три розділи: **Верхній колонтикул**, **Область даних**, **Нижній колонтикул**. Оскільки цей звіт зв'язаний із запитом, то дані з запиту будуть використані при перегляді та роздрукуванні звіту. Поля з запиту будуть доступні для використання в макеті звіту та з'являться у вікні списку полів.

5.3.5 Друк Звіту

1. Перейдіть у вікно попереднього перегляду.
2. Виконаєте команду **Файл⇒Друк** або натисніть на кнопки **Друк**, на екрані з'явиться діалогове вікно **Друк**.
3. В розділі **Роздрукувати** потрібно вказати, яку частину звіту потрібно вивести на папір. Наприклад, сторінки з 1 по 5.
4. В списку **Розрізняння** можна вказати якість друку.
5. В полі **Число копій** вкажіть кількість копій.

Контрольні питання

- ? Призначення звіту.
- ? Які звіти бувають?
- ? З якими об'єктами можна зв'язати звіт?
- ? Засоби створення звітів.
- ? Основні кроки роботи **Майстра звітів**.
- ? Яким чином можна згрупувати дані в звіті?
- ? Чи можна в звіті вивести підсумкові значення? Як це зробити?
- ? Призначення макету даних.
- ? Які стилі звітів бувають?
- ? Призначення елементів панелі інструментів **Конструктора звітів**.
- ? Призначення розділів звіту.
- ? Як створити автозвіт?
- ? Як можна використати звіт?

6 ЛАБОРАТОРНА РОБОТА № 6 "ВИКОРИСТАННЯ МАКРОСІВ"

6.1 Мета роботи

Метою роботи є придбання навичок створення та використання макросів у середовищі MS Access.

6.2 Завдання до лабораторної роботи

6.2.1 Ознайомитися зі змістом пункту 6.3 даних методичних указівок.

6.2.2 Згідно з індивідуальним завданням створити макроси та продемонструвати їхню роботу.

6.3 Основні теоретичні відомості

Макрос – це об’єкт Access, який створюється для автоматичного виконання певних дій або їх сукупності.

Будь-яка команда, що виконується в Access, називається **макрокомандою**. Access пропонує на вибір біля 50 макрокоманд, які можна використовувати в макросах.

Макрос дозволяє виконати наступні дії:

- зв’язати та запустити разом запити та звіти;
- одночасно відкрити декілька форм і/або звітів;
- контролювати правильність уводу даних при заповненні форм;
- переміщувати дані між таблицями;
- виконувати дії після щиглика на кнопці макрокоманди.

В Access макроси можна використовувати з будь-яким об’єктом бази даних.

6.3.1 Конструктор макросів

Вікно Конструктора макросів складається з чотирьох частин:

- меню;
- панелі інструментів;
- панелі макрокоманд (у верхній частині вікна);
- панелі аргументів (у нижній частині вікна).

Панель макрокоманд

По умовчанню панель макрокоманд складається з двох стовпчиків: **Макрокоманда** та **Примітка**. При необхідності можна додати два додаткових стовпчика **Ім'я макроса** та **Умова**. Для цього необхідно вибрати пункт меню **Сервіс⇒Параметри**, а потім у вкладці **Вигляд** у групі команд **Конструктор макросів** встановити опції **Стовпчик імен** і **Стовпчик умов**.

Панель аргументів макрокоманд

Ця область призначена для завдання необхідних аргументів (властивостей, параметрів) для відповідних макросів.

Після вибору макрокоманд і завдання для них відповідних аргументів макрос може бути збереженим для наступного виконання.

Завдання макрокоманд і аргументів

Існують наступні способи завдання макрокоманд і аргументів:

- ввести ім'я макрокоманди до стовпчику **Макрокоманда** вікна макросів;
- у стовпчику **Макрокоманда** вибрати ім'я макрокоманди зі списку, який розкривається;
- перетягнути й потім відпустити потрібні об'єкти з вікна бази даних до чарунки макрокоманд. Наприклад, можна перетягнути до чарунки одну з форм. При цьому Access автоматично додасть назву макрокоманди **ВідкритиФорму** й встановить відповідні аргументи (наприклад, ім'я форми).

Запуск макросів

Створений макрос можна запускати з:

- вікна макросів;
- вікна бази даних;
- вікон інших об'єктів;
- інших макросів.

Автоматичний запуск макроса при відкритті бази даних

Для автоматичного запуску певного макроса його необхідно

назвати **AutoExec**.

Крім того, в діалоговому вікні *Параметри запуску*), яке відкривається за допомогою команди *Сервіс⇒Параметри запуску*, можна задати ім'я форми, що буде відкриватися автоматично при відкритті бази даних.

6.3.2 Встановлення умов виконання макрокоманд

Умова – це логічний вираз, який повертає значення *Істина* чи *Фальш*, від яких буде залежати подальше виконання макроса.

Наприклад, потрібно викликати звіт *Прострочені замовлення*, який побудований на підставі однойменного запиту, тільки за умовою, що в базі даних є потрібні записи для складання звіту. Інакше треба перервати виконання макроса. В цьому випадку умовою припинення виконання макроса є вираз *IsNull([Forms]![Замовлення за вказаний період]![Дата_розм])* (рис. 6.3).

Условие	Макрокоманда	Примечание
IsNull([Forms]![Заказы за указанный период]![Дата_разм])	ОткрытьФорму	Відкрити форму
...	Сообщение	(Then) Якщо записів немає, викликати повідомлення
	ОстановитьМакрос	та закінчити макрос
	ОткрытьОтчет	(Else) Якщо записи є, то запустити звіт

Рисунок 6.3 – Приклад завдання умови виконання макроса

Посилання на елемент керування записується наступним чином:
 для форми – *Forms!ім'я_форми! ім'я_поля*;
 для звіту – *Reports! ім'я_звіту! ім'я_поля*.

Якщо ім'я складається з декількох слів, воно береться до прямокутних дужок.

6.3.3 Події та прив'язка до них макросів

Для запуску макроса можна використати подію. Подію Access можна розглядати як результат виконання деякої послідовності дій користувача. Наприклад, подія може бути пов'язана з переходом від одного запису форми до іншого, з вибором командної кнопки на формі або закриттям звіту.

Примітка

Об'єктами, що володіють властивостями подій, є форми та звіти. Таблиці та запити властивостями подій не володіють.

Події Access можна розбити на наступні категорії:

- **Події вікна (форма, звіт)** (відкриття, закриття, зміна розміру, тощо).
- **Подія даних** (активізація, вилучання, оновлення, тощо).
- **Подія фокуса вводу** (активізація, вхід, вихід, тощо).
- **Подія клавіатури** (натискання, відпускання клавіш, тощо).
- **Подія миші** (щиглик, щиглик на кнопці, тощо).
- **Подія друку** (форматування, вивід на друк, тощо).
- **Подія помилки та подія таймера** (виконуються через певний проміжок часу або у випадку виникнення помилки).

Макроси, що пов'язані з подіями форми

Для призначення макроса певній події форми необхідно створити цей макрос і вказати його для відповідної властивості форми. Перелік властивостей форм, з якими можна зв'язати виконання макроса, наведений у таблиці 6.1.

Таблиця 6.1 – Події форми

Подія	Коли виконується макрос
1	2
Поточний запис	При переході до наступного запису (коли вона становиться поточною)
До вставки	Після першого вводу даних до нового запису, але до реального створення запису
Після вставки	Після додавання нового запису до таблиці
До оновлення	Перед оновленням змінених даних у запису
Після оновлення	Після оновлення запису
Вилучання	При вилучанні запису – перед вилучанням
До підтвердження	Перед тим, як Access виведе вікно підтвердження вилучання
Після підтвердження	Після підтвердження вилучання в діалозі з Access
Відчинення	При відкритті форми – до показу першого запису

Продовження таблиці 6.1

1	2
Завантаження	При завантаженні форми до пам'яті, але перед відчиненням
Зміна розміру	При зміні розміру форми
Вивантаження	Коли форма зачинена – перед вилучанням її з екрану
Зачинення	Після зачинення та вилучання форми з екрану
Включення	Коли відчинена форма стає активним вікном
Відключення	Коли друге вікно стає активним – до втрати фокуса
Одержання фокуса	При одержанні формою фокуса вводу
Втрата фокуса	При втраті формою фокуса вводу
Натискання кнопки	При щиглику лівою кнопкою миші на керуючому елементі форми
Подвійне натискання кнопки	При подвійному щиглику лівою кнопкою миші на керуючому елементі форми
Кнопка униз	При натисканні кнопки миші, коли покажчик знаходиться в межах форми
Переміщення покажчика	Переміщення покажчика миші в межах форми
Кнопка уверх	При відпусканні натиснутої кнопки миші, коли покажчик знаходиться в межах форми
Клавіша униз	При натисканні клавіші на клавіатурі; при запуску макроса <i>КомандиКлавіатури</i>
Клавіша уверх	При відпусканні клавіші на клавіатурі; зразу ж після виконання макроса <i>КомандиКлавіатури</i>
Натискання клавіші	При натисканні та відпусканні клавіші; при запуску макроса <i>КомандиКлавіатури</i>
Помилка	При помилці виконання
Фільтрація	При визначенні фільтра, але перед його використанням
Застосування фільтра	Після використання фільтра у формі
Таймер	По закінченні певного інтервалу часу

Макроси, що пов'язані з подіями елементів керування

Макроси можна запускати й на рівні окремих елементів керування, використовуючи для цього відповідні події (табл. 6.2).

Наприклад, можна перевірити правильність уводу складних даних безпосередньо на рівні поля (а не після виходу з цього запису), використовуючи для цього властивість даного поля, а не властивості на рівні всієї форми.

Таблиця 6.2 – Події елементів керування

Подія	Коли виконується макрос
До оновлення	Перед оновленням змінених даних у елементі керування
Після оновлення	Після оновлення змінених даних у елементі керування
Оновлення	При зміні вмісту текстового поля або комбінованого списку
При оновленні	При зміні даних OLE-об'єкта
Відсутність у списку	При вводі до комбінованого списку значення, що відсутнє в списку
Вхід	Перед одержанням фокуса вводу елементом керування від іншого елемента
Вихід	Перед передачею фокуса вводу іншому елементу керування
Одержання фокуса	При одержанні елементом керування фокуса вводу
Втрата фокуса	При втраті елементом керування фокуса вводу
Натискання кнопки	При щиглику лівою кнопкою миші на керуючому елементі
Подвійне натискання кнопки	При подвійному щиглику лівою кнопкою миші на керуючому елементі
Кнопка униз	При натисканні кнопки миші, коли покажчик знаходиться в межах елемента керування
Переміщення покажчика	Переміщення покажчика миші в межах елемента керування
Кнопка уверх	При відпусканні натиснутої кнопки миші, коли покажчик знаходиться в межах елемента керування
Клавіша униз	При натисканні клавіші на клавіатурі; при запуску макроса <i>КомандиКлавіатури</i>
Клавіша уверх	При відпусканні клавіші на клавіатурі; зразу ж після виконання макроса <i>КомандиКлавіатури</i>
Натискання клавіші	При натисканні та відпусканні клавіші; при запуску макроса <i>КомандиКлавіатури</i>

Макроси звіту

При роботі зі звітами можна також використовувати макроси, що будуть запускатися автоматично при виконанні вказаних подій (табл. 6.3). Макроси можуть бути використаними на рівні всього звіту або рівнях окремих розділів (областей) звіту.

Крім властивості подій, які відбуваються на рівні звітів, Access дозволяє використовувати три додаткові властивості для призначення макросів, які виконуються на рівні окремих розділів звіту (табл.. 6.4)

Таблиця 6.3 – Події рівня звіту

Подія	Коли виконується макрос
Відкриття	При відкритті звіту, але до друку
Закриття	При закритті звіту та вилучанні його з екрану
Включення	При одержанні звітом фокуса вводу (звіт стає активним вікном)
Відключення	При загубленні звітом фокуса вводу (активним стає інше вікно)
Відсутність даних	При відсутності переданих до звіту даних із таблиць або запитів
Сторінка	При зміні поточної сторінки
Помилка	При помилці часу виконання Access

Таблиця 6.4 – Події рівня розділів звіту

Подія	Коли виконується макрос
Форматування	При зміні макету сторінки – до форматування даних розділів звіту
Друк	При підготовці даних в розділі до друку – до виконання друку розділу
Повернення	Після форматування, але до друку – при багатопрохідному форматуванні

Контрольні питання

- ? Призначення макросів.
- ? Складові частини вікна Конструктора макросів;
- ? Способи завдання макрокоманд.
- ? Автоматичний запуск макроса при відкритті бази даних.
- ? Умови виконання макрокоманд.
- ? Відлагодження макросів.
- ? Поняття події.
- ? Поняття фокусу.
- ? Використання макросів у формах.
- ? Використання макросів у звітах

ЛІТЕРАТУРА

1. Баловсяк Р.Л. Система управління базами даних Access для самостійного вивчення. Навчальний посібник. – К.: КНТ, 2006. – 158 с.
2. Блюттман К. Анализ данных в Access. Сборник рецептов. – СПб: Питер, 2008. – 350 с.
3. Блюттман К. Access. Трюки. – СПб: Питер, 2006. – 332 с.
4. Гандерлой М., Харкинз С. Автоматизация Microsoft Access с помощью VBA. – М.: Вильямс, 2006. – 416 с.
5. Моисеенко А.О. Разработка приложений в Access. Краткое руководство. – К.: Диалектика, 2006. – 272 с.
6. Сенов А.П. Access 2007. Учебный курс. – СПб: Питер, 2008. – 267 с.
7. Тимошок В.Р. Access 2007. Самоучитель. – К.: Диалектика, 2008. – 464 с.

