

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

Факультет післядипломної освіти

“Проектування баз даних”

Методичні вказівки до
лабораторних робіт

Частина 2

для слухачів спеціальності 7.080403
“Програмне забезпечення автоматизованих систем”

Проектування баз даних. Методичні вказівки до лабораторних робіт. Частина 2. Для слухачів спеціальності 7.080403 “Програмне забезпечення автоматизованих систем” факультету післядипломної освіти /Укл.: С.К.Корнієнко. – Запоріжжя: ЗНТУ, 2003. - 62 с.

Укладач:

С.К.Корнієнко, доцент, к.т.н.,

Рецензент:

Пінчук В.П., доцент, к.т.н

Відповідальний за випуск: С.К.Корнієнко

Затверджено
на засіданні кафедри ПЗ

Протокол № 2 від 13.12.2002

СОДЕРЖАНИЕ

5 Лабораторна робота № 6 СТВОРЕННЯ СКЛАДНИХ ЗАПИТІВ.....	4
6 Лабораторна робота № 6 СТВОРЕННЯ Й ВИКОРИСТАННЯ ФОРМ ДАНИХ.....	20
7 Лабораторна робота № 7 РОЗРОБКА ЗВІТІВ.....	41
8 Лабораторна робота № 8 ВИКОРИСТАННЯ МАКРОСІВ.....	49
ЛІТЕРАТУРА.....	60
ДОДАТОК А СЛОВНИК ОСНОВНИХ ТЕРМІНІВ.....	61

5 ЛАБОРАТОРНА РОБОТА № 5 СТВОРЕННЯ СКЛАДНИХ ЗАПИТІВ

5.1 Мета роботи

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

5.2 Завдання на лабораторну роботу

5.2.1 Ознайомитися зі змістом пункту 8.3 методичних вказівок.

5.2.2 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.

5.3 Основні теоретичні відомості

5.3.1 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* (рисунок 5.1).

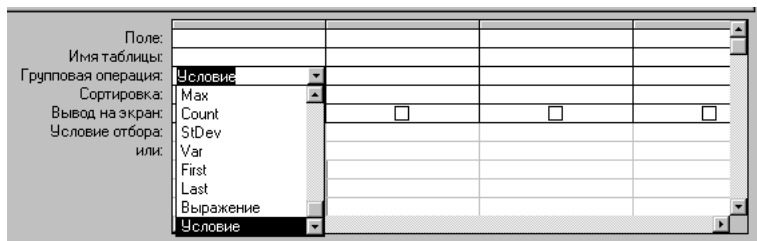


Рисунок 5.1 - Список, що розкривається, рядку *Группова операция*

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 замовчанню для всіх чарунк рядка *Группова операция*.

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 аргументу групи записів і виконує деякі математичні дії над ними (таблиця 5.1). Це може бути як проста функція, так і складний вираз, що базується на інших функціях і задається користувачем.

Таблиця 5.1 - 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 запити	AutoNumber, Number, Currency, Date/Time, Yes/No, Text, Memo, OLE
Last	Значення поля в останньому запису таблиці або запити	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 встановлена 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 5.2).

* 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:	Название	Цена	Цена	МинимальныйЗапас	МинимальныйЗапас
Имя таблицы:	Товары	Товары	Товары	Товары	Товары
Групповая операция:	Группировка	Sum	Avg	Группировка	Условие
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:					<200
или:					

Рисунок 5.2 - 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 5.3). У термінах **SQL** така умова задається оператором **Having**.

Поле:	Название	Цена	Цена	МинимальныйЗапас	
Имя таблицы:	Товары	Товары	Товары	Товары	
Групповая операция:	Группировка	Sum	Avg	Группировка	
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:	Like "крупн"	>1000			
или:					

Рисунок 5.3 - 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 відрізнитися. 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 **Груповая операція** даного стовпчика вибрати опцію **Выраз**. На малюнку 8.4 показаний приклад упорядкування запиту, що визначає уцінку товару на складі (усіх видів круп) 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.

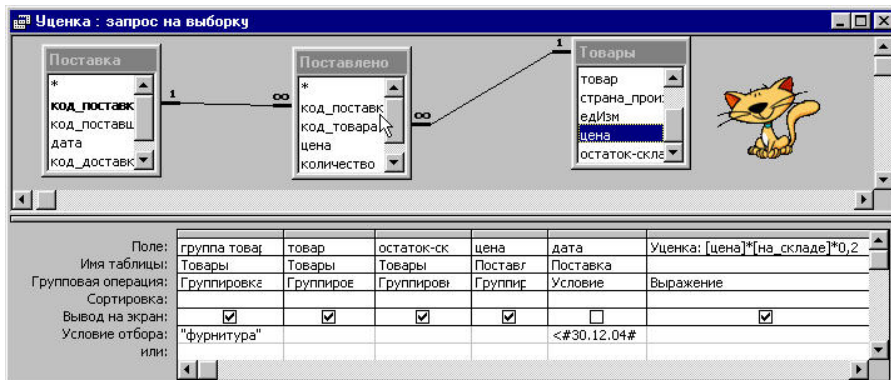


Рисунок 5.4 - Запит із використанням виразу в груповій операції

5.3.2 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.

Дані параметри вказуються в призначеному для цього рядку **Перехресна таблиця**.

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 5.5 приведений приклад конструювання даного запиту, результат виконання запиту показаний на рисунку 5.6.

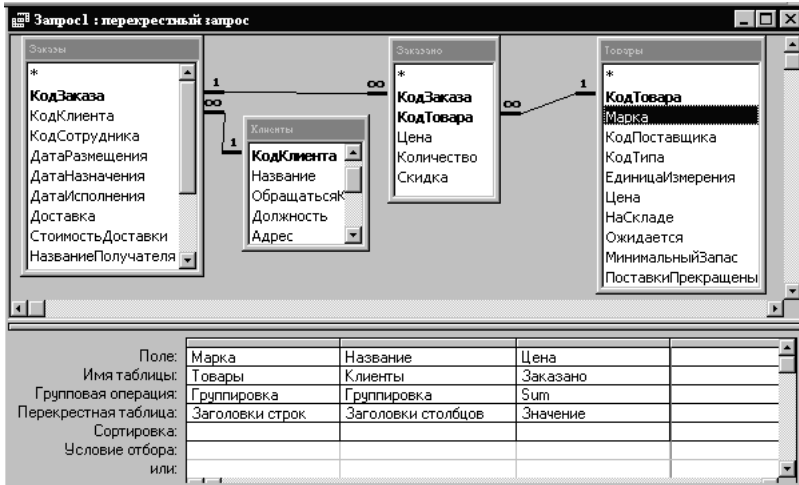


Рисунок 5.5 - Створення перехресного запиту

	Марка	4-й роддом	АО ЭВА	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

Рисунок 5.6 - 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.)

5.3.3 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 (див. Лабораторну роботу №1). 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.

5.3.4 Створення запитів на зміну

Запити на зміну (*action query*) 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 само необхідно вибрати поля для запиту та зазначити умови відбору. Крім того, необхідно вказати визначені властивості цих запитів - *Додавання, Створення таблиці, Оновлення* або *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 результат.

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*).

Майстер запитів на пошук дубльованих записів

Цей майстер допомагає створити запит, що вказує на однакові записи в таблиці, на підставі значень поля в цій таблиці. Access запитує, за якими полями потрібно виконувати пошук дублікатів і які поля повинні бути присутніми у результуючій таблиці.

Майстер запитів на пошук записів без підпорядкованих записів

Цей майстер допомагає створити запит, що дозволяє знайти в двох пов'язаних таблицях *запис-сироту* та *запис-вдову*.

Запис-сирота - це запис у таблиці з боку *багато*, що не має відповідного запису в таблиці з боку *один*.

Запис-вдова - це запис у таблиці з боку *один* зв'язку один-до-одного або один-до-багатьох, що не має відповідного запису в іншій таблиці.

При запуску цього майстра Access запитує найменування таблиць, для котрих необхідно виконати порівняння, ім'я поля, за яким створений зв'язок між таблицями, імена полів першої таблиці, що повинні виводитися в результуючій таблиці запиту, та назву результуючої таблиці. Потім створюється запит, що дозволяє відшукати записи, що не мають відповідних записів в іншій таблиці.

Якщо ви створюєте зв'язок між таблицями й намагаєтеся встановити опцію підтримки цілісності даних, але Access не в змозі зробити це, мабуть, виникла проблема, пов'язана з записами. Цю проблему й допоможе вирішити майстер запитів на пошук записів без підпорядкованих.

Проблеми, що виникають при роботі з запитами на зміну

При роботі з запитами на зміну необхідно пам'ятати про деякі можливі проблеми. При виконанні запиту можуть з'явитися повідомлення про те, що, наприклад, загублено декілька записів через руйнування ключа або що записи заблоковані на час виконання запиту. Нижче розповідається про деякі проблеми й про те, як їх уникнути.

Помилки даних при додаванні та оновленні

Access не дозволяє вводити значення, що не підходять для заданого типу поля. При цьому СКБД просто ігнорує те, що вводиться, й **присвоює** полю значення Null. При роботі з запитами на додавання це означає, що записи будуть додані, але вони можуть виявитися порожніми!

Руйнування ключа при роботі з запитами на зміну

Додати записи з ключовим полем в іншу базу даних, у котрій також зазначене ключове поле, неможливо: Access не припускає оновлення запису та зміни ключового поля, якщо воно вже задано. Значення ключового поля можна змінити тільки в наступних випадках:

- Значення ключового поля ще не зазначено.
- Значення змінюваного поля не пов'язано з полями інших таблиць.

Access не дозволяє вилучити поле з боку **один** у зв'язку **один-до-багатьох** до вилучання записів із боку **багато**.

Access не дозволяє додати або оновити значення поля, якщо в результаті з'явиться повторний запис у **індексному полі**, збіги в якому не припускаються.

Блокування полів у багатокористувацькому середовищі

Access не буде виконувати запити на зміну записів, які заблоковані іншими користувачами. У цьому випадку при виконанні

запитів на оновлення або на додавання можна продовжити роботу над іншими записами; проте при цьому складно буде визначити, які записи були змінені, а які ні.

Текстові поля

При додаванні або оновленні текстових полів, довжина котрих менше поточної довжини поля, відбудеться усікання текстових даних, про що Access **не повідомить**.

Резюме

- ✗ Перед виконанням запитів на зміну необхідно зробити резервні копії таблиць.
- ✗ Перед створенням запитів на зміну необхідно створити запит на вибірку для перевірки слушності відбору даних.
- ✗ Запит на додавання може працювати одночасно тільки з одною таблицею.
- ✗ Для виконання запиту на додавання необхідна таблиця, до якої будуть додані записи (запит на додавання такої таблиці не створює).
- ✗ Для додавання всіх записів однієї таблиці в іншу краще скопіювати таблицю в буфер обміну, а потім уставити її в іншу таблицю, чим використовувати запит на додавання.
- ✗ Запити на створення таблиці можуть використовувати поля з однієї або декількох таблиць для створення нової.
- ✗ Запити на вилучання дозволяють вилучати записи з декількох таблиць, пов'язаних відношенням **один-до-багатьох**.
- ✗ При використанні запитів на вилучання для таблиць, пов'язаних відношенням **один-до-багатьох**, спочатку треба вилучити записи з боку **багато**, а потім - із боку **один**.
- ✗ Не рекомендується зберігати запит на зміну, якщо ви не збираєтеся використовувати його багаторазово.
- ✗ При виконанні запитів на зміну дотримуються умов цілісності даних. Якщо при виконанні запиту на зміну відбудеться порушення умов цілісності даних, Access припинить виконання даної операції.

5.4 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.

5 ЛАБОРАТОРНА РОБОТА № 6 СТВОРЕННЯ Й ВИКОРИСТАННЯ ФОРМ ДАНИХ

6.1 Мета роботи

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

6.2 Завдання на лабораторну роботу

6.2.1 Ознайомитися зі змістом пункту 9.3 методичних указівок.

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

6.3 Основні теоретичні відомості

Форми зручні для перегляду, введення, зміни або вилучення 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**.

6.3.1 Створення форми за допомогою Автоформи

Щоб швидко створити форму, можна скористатися кнопкою **Створити**, а потім у діалоговому вікні, що з'явилося (рис.6.1), 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.

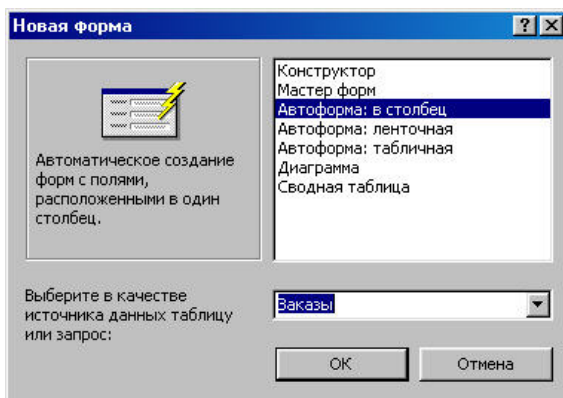


Рисунок 6.1 - Діалогове вікно *Нова форма*

6.3.2 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 6.2-6.5 наведені приклади форм, створених на підставі таблиці "Товари".

Товары

код_товара: 2

группа_товара: фурнитура

товар: Ручка для шкафа

страна_произв: Италия

ед. изм.: шт

цена: 5,00грн

остаток-склад: 5

Запись: 2 из 25

Рисунок 6.2 – Зразок одностовпчкової форми

Товары

код	группа	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

Запись: 1 из 25

Рисунок 6.3 – Зразок стрічкової форми

код_тов	группа_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

Запись: 1 из 25

Рисунок 6.4 – Зразок табличної форми

код_товара: 1

группа_товара: фурнитура

товар: Ручка дверная

страна_производитель: Польша

ед. изм.: шт

цена: 24,00грн

остаток-склад: 1

Запись: 1 из 25

Рисунок 6.5 – Зразок вирівняної форми

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

6.3.3 Елементи керування та їхні властивості

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

Типи елементів керування

Таблиця 6.1 - Елементи керування форм і звітів

Елемент керування	Призначення
1	2
Основні елементи керування	
Надпис	Служить для відображення тексту
Поле	Дозволяє вводити дані
Розширені елементи керування введення даних	
Група перемикачів	Містить декілька вимикачів, перемикачів або прапорців
Вимикач	Може знаходитися в двох станах: натиснутому або віджатому, нагору або униз. Зазвичай на ньому розміщують картинки або піктограми
Перемикач	Відображається у вигляді кружка з точкою усередині, що позначає активізацію цього елемента. Його також називають <i>радіокнопкою</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 6.1

1	2
<i>Елементи керування графікою</i>	
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 за допомогою спеціальної панелі (рисунок 6.6), 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



Рисунок 6.6 - Панель елементів

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 оновлюють вміст полів таблиці.

Використання елементів керування

Вимикачі, перемикачі та прапорці

Використовуються для відображення даних логічного типу. Вид цих елементів можна змінювати (рис. 6.7) за допомогою кнопки **Оформлення** панелі інструментів або відповідного пункту контекстного меню.

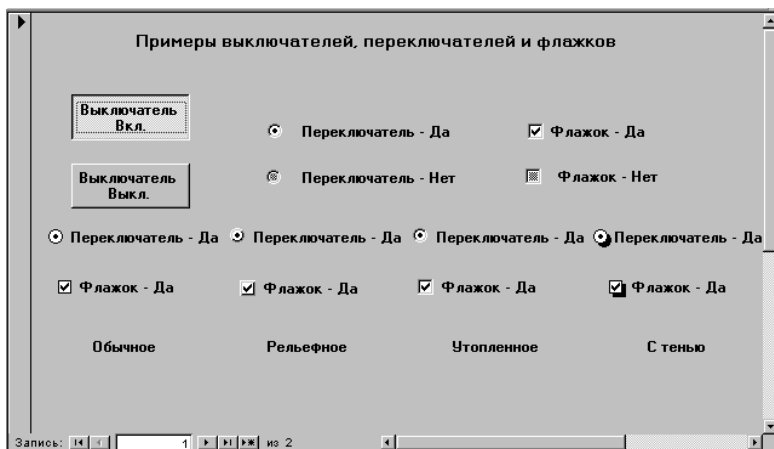


Рисунок 6.7 - Приклади оформлення вимикачів, перемикачів і прапорців

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) (рисунок 6.8,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. 6.8,б).



Рисунок 6.8 - 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. 6.9). Для додавання, вилучання або заміни місцями вкладок варто скористатися відповідними опціями в контекстному меню.

Способи створення елементів керування

Елемент керування можна створити двома способами:

- Натисканням на кнопку *Список полів* панелі інструментів і перетягуванням потрібного поля в макет форми. У результаті буде створений приєднаний елемент керування.
- Вибором і перетягуванням елемента з панелі елементів. У результаті буде створений вільний елемент керування

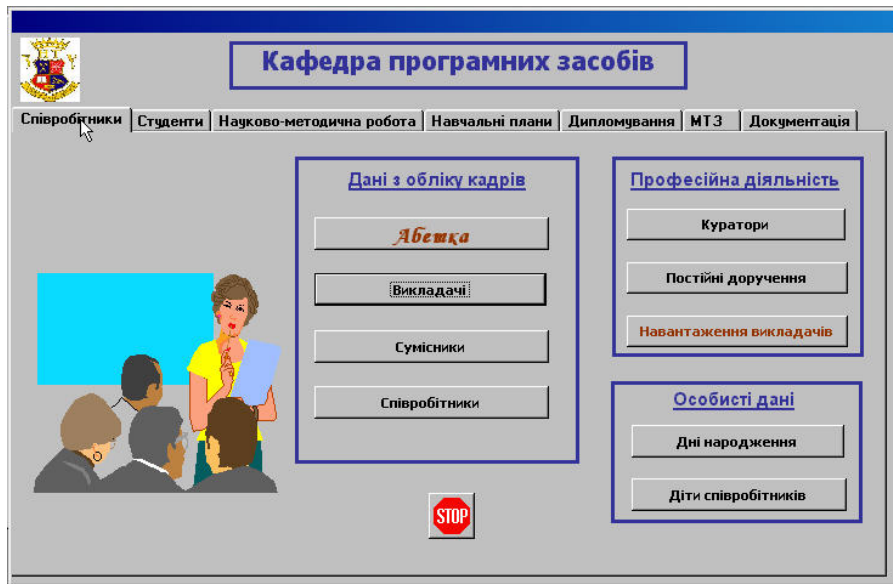


Рисунок 6.9 – Приклад використання вкладок

Вирівнювання елементів керування

Для вирівнювання елементів керування по одній лінії потрібно спочатку виділити їх, утримуючи натиснутою клавішу *Shift* або перетягуючи покажчик миші через потрібні елементи, укладаючи їх у рамку, а потім використовувати команду *Формат*⇒*Вирівняти* або опцію *Вирівняти* в контекстному меню.

Вирівнювати елементи керування можна також, прив'язавши їх до сітки або задавши інтервали по горизонталі або вертикалі за допомогою команди *Формат*⇒*Інтервал по вертикалі* або *Формат*⇒*Інтервал по горизонталі*.



Порада

При вставці об'єкта OLE (наприклад, малюнка) він може не вміщуватися цілком у рамку. Необхідно вибрати значення властивості *Установка розмірів* для даного елемента рівним *За розміром рамки*.

Властивості елементів керування

Властивостями називають параметри елементів керування, полів і об'єктів баз даних, що визначають характеристики об'єктів, наприклад, розмір, колір, зовнішній вигляд і ім'я. За допомогою властивостей можна також змінювати тип поведінки елементів керування, визначаючи, наприклад, чи буде елемент керування схований або чи можливо в ньому редагування.

Деякі властивості, наприклад, формат поля, маска введення, успадковуються з поля таблиці, зв'язаного з елементом керування. При цьому зміна властивостей елемента керування не викликає зміни властивостей поля в таблиці-джерелі.

Для заданого типу елемента керування можна створити набір стандартних властивостей (див. нижче), клацнувши на кнопці елемента обраного типу панелі елементів, а потім на кнопці **Властивості** панелі інструментів. Змінюючи набір властивостей, прийнятих за замовчуванням, можна створювати форми набагато швидше, ніж змінюючи їх окремо для кожного елемента керування.

6.3.4 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 перетаскувати відразу можна групу полів, попередньо виділивши їх звичайним чином.

Зміна послідовності переходу між елементами

За замовчуванням послідовність переходу між елементами форми завжди відповідає порядку, у якому вони додавалися в цю форму. Для зміни послідовності переходу у вікні конструктора форм виберіть команду **Вид** —> **Послідовність переходу**. Кнопка **Авто** дозволяє розмістити поля в порядку ліворуч, а потім праворуч і зверху вниз, відповідно до їхнього розташування у формі.

Додавання фонового рисунка

Для додавання фонового рисунка потрібно:

1. Вибрати форму, клацнувши в її верхньому лівому куті (у місці перетинання двох лінійок).
2. Викликати вікно властивостей і клацнути на властивості **Рисунок**.
3. Ввести шлях до файлу з рисунком.
4. Клацнути на властивості **Масштаби рисунка** та задати нове значення.

Зміна розміру форми

Сітка в центрі вікна режиму конструктора форми (рис. 6.10) представляє розмір форми при її запуску. Щоб змінити розмір форми, виділіть край сітки і розтягніть її до потрібного розміру. Зверніть

увагу на лінійки уздовж верхнього і лівого країв сітки: вони допомагають задати остаточний розмір форми.

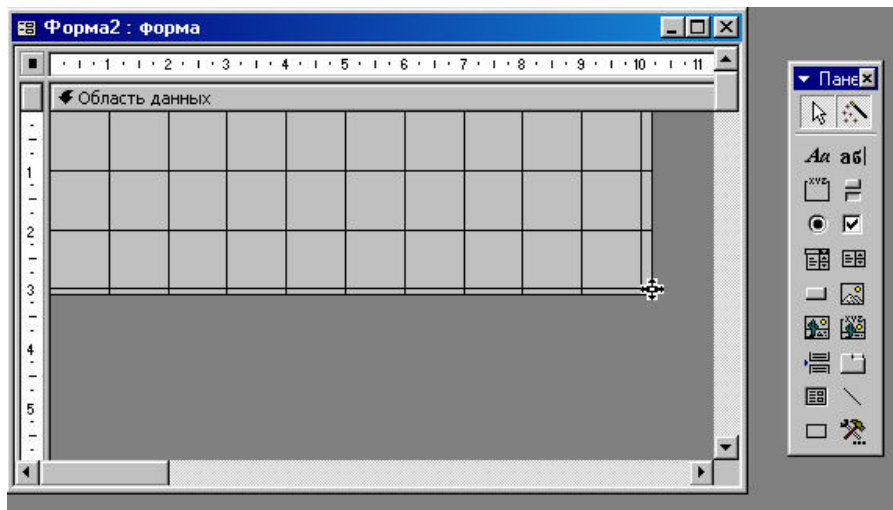


Рисунок 6.10 – Вікно Конструктора форм

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. 6.11).

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 6.11).

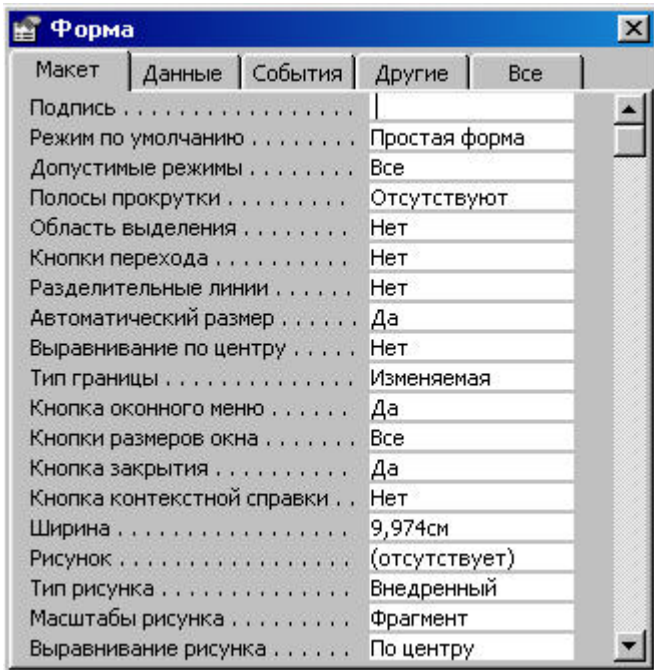


Рисунок 6.11 – Вікно властивостей форми

- **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 щиклика 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.
3. Виберіть текстове поле біля властивості, що змінюється, та введіть нове значення. Деякі властивості дозволяють запустити *Побудовник виразів*.
 4. Закрийте вікно властивостей форми. При цьому внесені зміни зберігаються.

Сітка, на якій розміщаються елементи керування, за замовчуванням служить для фіксації елемента керування. Це допомагає розмістити елементи керування на формі в лінію.

Іноді таке захоплення сіткою елементів керування заважає роботі. Елементи продовжують зберігати невірне розташування, крім того, буває важко змінити їхній розмір. Щоб блокувати захоплення сіткою елементів керування, скиньте прапорець *Прив'язка до сітки* в меню **Формат**.

Щоб тимчасово деактивізувати опцію *Прив'язати до сітки*, утримуйте натиснутої клавішу <Ctrl> при переносі чи зміні розміру елемента керування на сітці.

Додавання до форми елементів керування

Перелік елементи керування наведений у таблиці 6.1. Для розміщення елемента керування на формі (чи в сітці) треба спочатку вибрати його за допомогою миші на панелі елементів (рисунок 6.6).

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. Коли елемент керування розміщується на сітці, покажчик миші відновлює свою звичайну форму.

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 6.12).
3. В лівому списку виберіть категорію дії, що повинна виконувати кнопка.
4. В правому списку виберіть необхідну дію та клацніть на кнопці **Далі**.
5. Введіть необхідний текст напису на кнопці або виберіть для неї необхідний рисунок.
6. Введіть ім'я кнопки та клацніть на кнопці **Готово**.

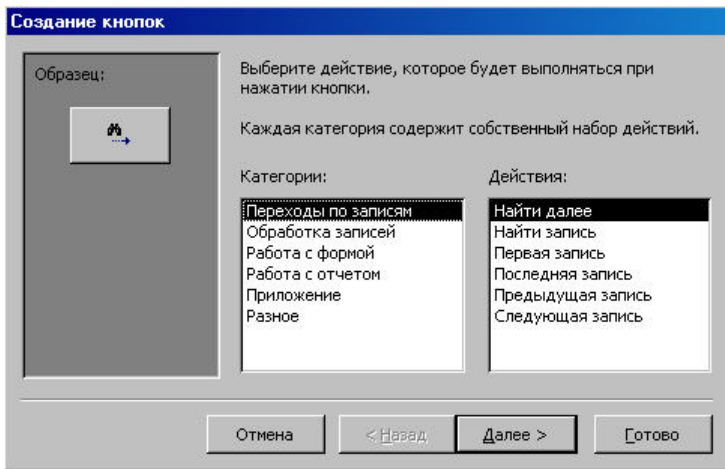


Рисунок 6.12 – Вікно Майстра кнопок

Створення властивостей елемента керування

Для кожного елемента керування є таблиця властивостей, подібна з таблицею властивостей форми. На рис. 6.13 показано вікно властивостей елемента керування **Кнопка**.

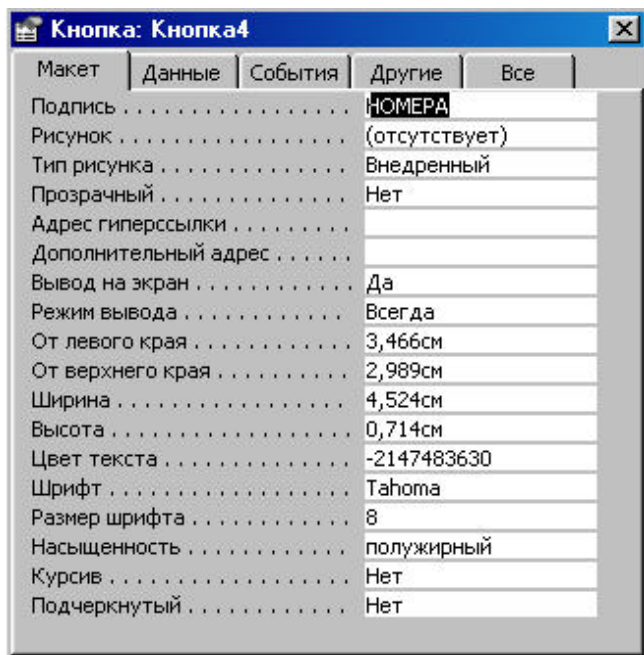


Рисунок 6.13 – Вікно властивостей елемента керування **Кнопка**

Вікна властивостей різних елементів дещо відрізняються друг від друга, але в цілому їхня структура подібна. Це полегшує недосвідченим користувачам заміну установок властивостей. Для ілюстрації на рис.6.14 показано вікно властивостей елемента керування **Лінія**.

Як видно з цього рисунка, загальне компонування вікон властивостей елементів керування **Лінія** і **Кнопка** однакові. Розходження обумовлено тим, що з елементом керування **Кнопка** асоційовано більше властивостей.

Щоб змінити властивість у вікні властивостей, виконасте наступні дії:

1. Відкрийте вікно властивостей, клацнувши правою кнопкою миші на елементі керування та вибравши в контекстному меню команду **Властивості**.

2. Клацніть на вкладці, що містить властивість, яку треба змінити.

3. Виберіть текстове поле біля властивості, що змінюється та введіть його значення. Деякі властивості дають можливість відкрити вікно макросів чи запустити побудовника виразів.

4. Закрийте вікно властивостей. При цьому внесені зміни зберігаються.



Рисунок 6.14 – Вікно властивостей елемента керування *Лінія*

6.3.5 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**, клацнувши на вкладці **Форми** у вікні бази даних, а потім клацнувши на кнопці **Конструктор**.
2. Виберіть в меню Вікно команду Ліворуч праворуч. При цьому вікно бази даних і вікно Режим конструктора форми розташовуються по горизонталі без перекриття.
3. Перетягніть з вкладки **Форми** у вікні бази даних і опустіть в потрібному місці головної форми ту форму, що повинна стати підлеглою
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 6.15 наведений приклад використання головної форми **Поставка** та підлеглої форми **Склад поставки**.

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 6.15 – Приклад використання підлеглої форми

6.4 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**.

7 ЛАБОРАТОРНА РОБОТА № 7 РОЗРОБКА ЗВІТІВ

7.1 Мета роботи

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

7.2 Завдання до лабораторної роботи

- 8.2.1 Ознайомитися зі змістом пункту 7.3 даних методичних указівок.
- 8.2.2 Створити звіти та відредагувати їх у відповідності до індивідуального завдання

7.3 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**.

7.3.1 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.

7.3.2 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.7.1), задати необхідні дії для потрібних полів.

Тут можна також відобразити або сховати дані в розділі даних. Якщо обрана опція **Дані та підсумки**, в звіті будуть показані всі дані, а якщо **Тільки підсумки**, то розділ даних буде прихований і у звіті будуть показані тільки підсумки.

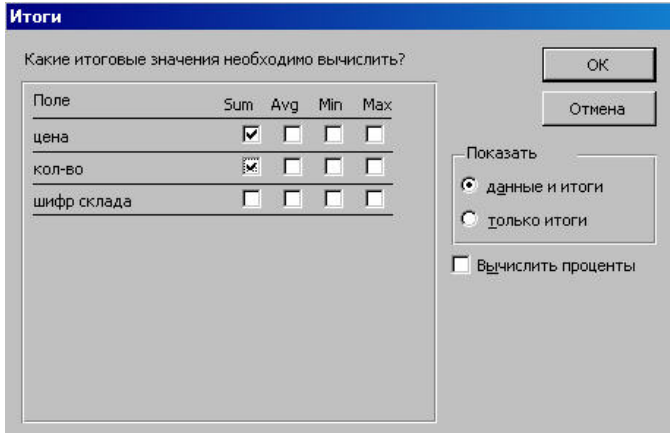


Рисунок 7.1 – Вікно завдання підсумків

9. Наступне вікно (рис. 7.2) призначене для завдання макету даних і орієнтації сторінок звіту.

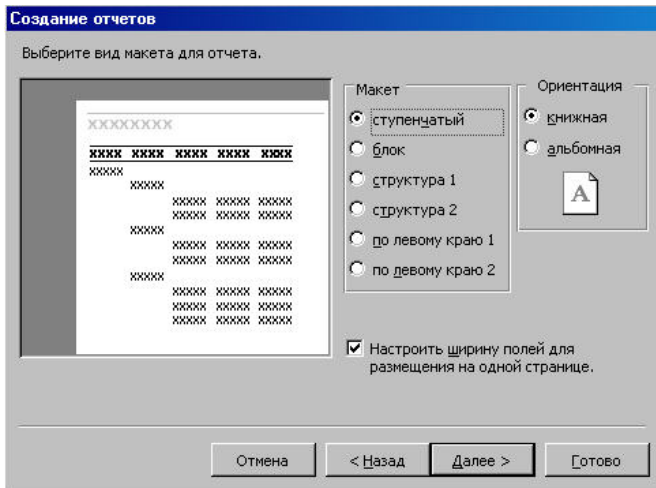


Рисунок 7.2 – Вікно завдання макету даних

10. У наступному вікні (рис. 7.3) треба вибрати необхідний стиль оформлення звіту.

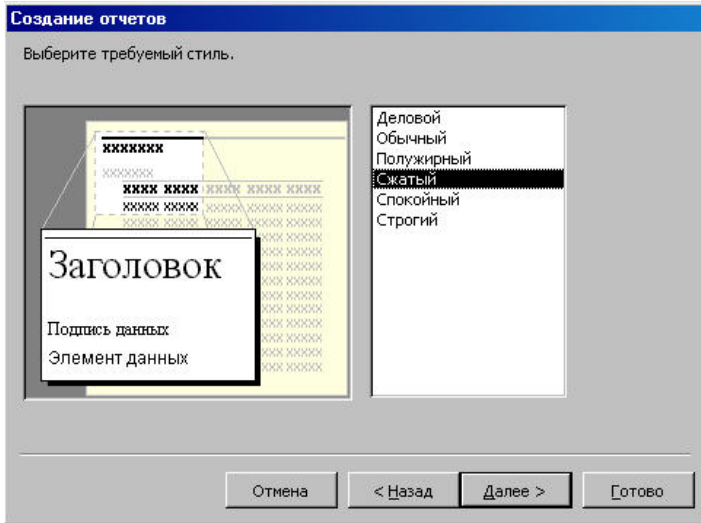


Рисунок 7.3 – Вікно вибору стилю звіту



Порада

Будь-який стиль можна налагодити. Крім того, є можливість створити власний стиль, скориставшись командою **Формат**⇒**Автоформат**.

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**.

7.3.2 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.

7.3.3 Розробка звітів за допомогою Конструктора

Панель інструментів *Конструктора звітів* (рисунок 7.4) відрізняється від панелі форматування й призначена для виконання інших задач. Наприклад, для вибору шрифту чи виду вирівнювання слід використовувати панель форматування.



Рисунок 7.4 – Панель інструментів режиму *Конструктора звітів*

Призначення окремих кнопок панелі інструментів наведено в таблиці 7.1.

Таблиця 7.1 – Елементи панелі інструментів *Конструктора звітів*

Кнопка панелі інструментів	Призначення
1	2
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

Продовження таблиці 7.1

1	2
Список полів	Відкриває/закриває вікно списку полів
Панель елементів	Відкриває чи закриває панель елементів
Сортування та групування	Відкриває чи закриває діалогове вікно <i>Сортування та Групування</i>
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 (таблиця 7.2). Заголовок звіту друкується на початку звіту на титульній сторінці.

Таблиця 7.2 – 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 запит 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 колонтитул**, **Область даних**, **Нижній колонтитул**. 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.

7.3.4 Друк Звіту

1. Перейдіть у вікно попереднього перегляду.
2. Виконаєте команду **Файл⇒Друк** або натисніть на кнопці **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.

7.3.5 Закриття звіту

Для закриття звіту потрібно виконати команду **Файл**⇒ **Збереження** або зробити два щиклика на кнопці системного меню. Якщо звіт раніше не був збережений, то ACCESS запитас для нього ім'я.

Створений звіт можна переглянути в режимі попереднього перегляду, надрукувати його, експортувати до **Word** або **Excel**. При необхідності модифікація звіту робиться за допомогою **Конструктора звітів**.

7.4 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?

8 ЛАБОРАТОРНА РОБОТА № 8 ВИКОРИСТАННЯ МАКРОСІВ

8.1 Мета роботи

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

8.2 Завдання до лабораторної роботи

8.2.3 Ознайомитися зі змістом пункту 8.3 даних методичних указівок.

8.2.4 Створити звіти та відредагувати їх у відповідності до індивідуального завдання

8.3 Основні теоретичні відомості

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

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

Макрос дозволяє виконати наступні задачі:

- зв'язати й запустити разом запити та звіти;
- одночасно відкрити декілька форм і/або звітів;
- контролювати правильність вводу 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.

8.3.1 Конструктор макросів

Вікно Конструктора макросів складається з чотирьох частин (рис. 8.1):

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

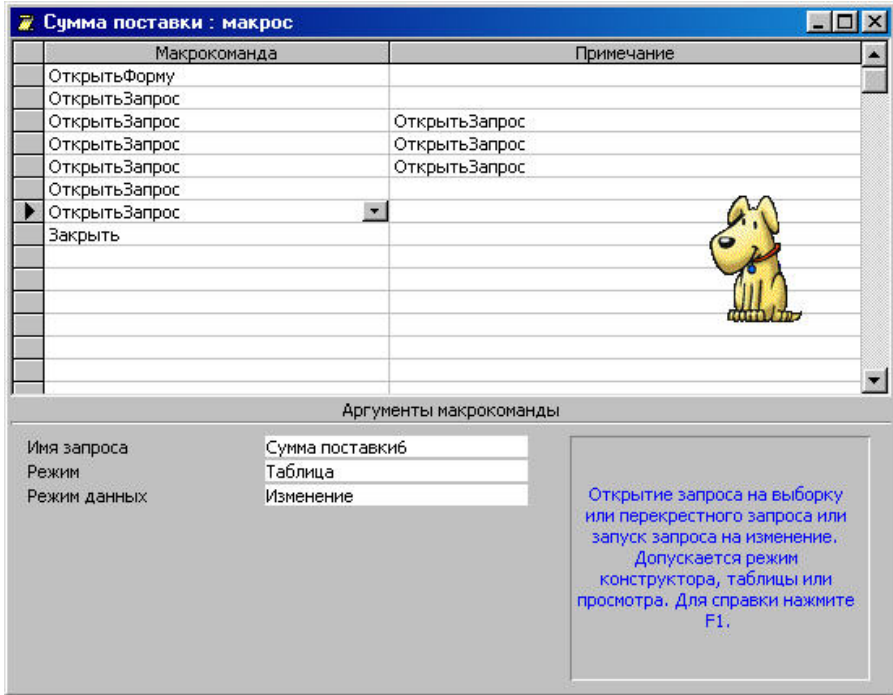


Рисунок 8.1 – Вікно Конструктора макросів

Панель макрокоманд

По умовчанняю 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 на зву макрокоманди **ВідкритиФорму** й встановить відповідні аргументи (наприклад, ім'я форми).



Порада

Для зручності перетягування об'єкта з вікна бази даних до вікна конструювання макроса ці вікна доцільно розташовувати поруч, використовуючи властивість **Зліва направо** пункту **Вікно** головного меню Access (рис. 8.2).

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** (рис. 8.3), яке відкривається за допомогою команди **Сервіс⇒Параметри запуску**, можна задати ім'я форми, яка буде відкриватися автоматично при відкритті бази даних.

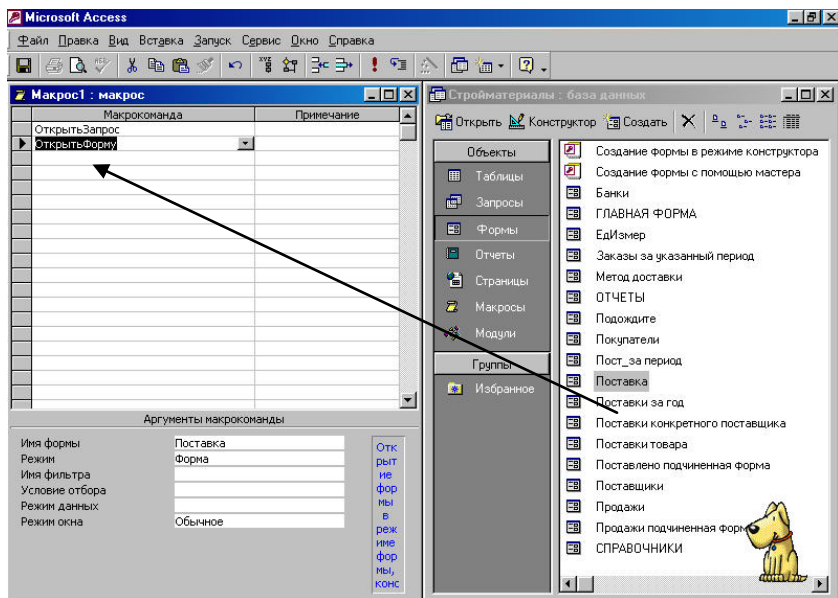


Рисунок 8.2 – Вікна бази даних і макросів, розташованих поруч

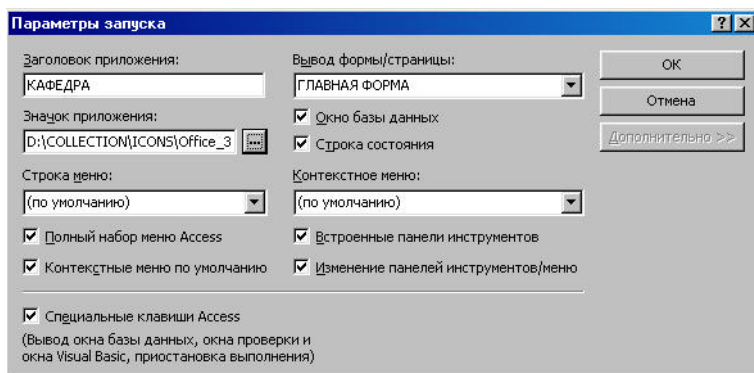


Рисунок 8.3 – Установка опций у диалоговому вікні параметрів запуску

8.3.2 Встановлення умов виконання макрокоманд

Іноді необхідно запусити окрему макрокоманду або їхню сукупність тільки при виконанні певних умов.

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])* (рисунок 8.4).

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.



До уваги!

Якщо в умовах макроса є посилання на ім'я елемента керування (ім'я поля), джерело (форма або звіт) цього елемента повинне бути відчинене.

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

Рисунок 8.4 – Приклад завдання умови виконання макроса

8.3.3 Відлагодження макросів

Access надає дві можливості відлагодження макросів:

- покрокове виконання макроса;
- використання діалогового вікна помилок виконання.

Для цього необхідно:

1. Вибрати макрос у вікні макросів.
2. Зробити щиклик на кнопці **По кроках** панелі інструментів або вибрати з меню команду **Запуск**⇒**По кроках**

3. Запустити макрос.

Access відкріє діалогове вікно *Покрокове виконання макроса* (рис. 8.5), в якому показані ім'я макроса, ім'я макрокоманди, що виконується, та її аргументи.

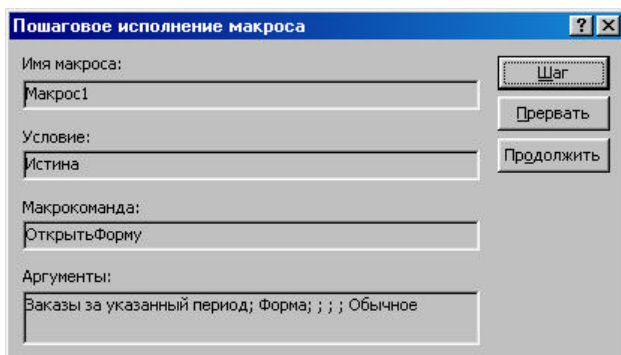


Рисунок 8.5 – Вікно Покрокове виконання макроса

8.3.4 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:

1. **Події вікна (форма, звіт)** (відкриття, закриття, зміна розміру, тощо).
2. **Подія даних** (активізація, вилучання, оновлення, тощо).
3. **Подія фокуса вводу** (активізація, вхід, вихід, тощо).
4. **Подія клавіатури** (натискання, відпускання клавіші, тощо).

5. **Подія миші** (щиглик, щиглик на кнопки, тощо).
6. **Подія друку** (форматування, вивід на друк, тощо).
7. **Подія помилки та подія таймера** (виконуються через певний проміжок часу або у випадку виникнення помилки).

Access може розпізнавати події за допомогою властивостей форм, елементів керування та звітів. Наприклад, за допомогою властивостей елемента керування *Поле* об'єкта *Форма* можна призначити виконання певного макроса, як показано на рисунку 8.6.

Наявність фокуса означає, що даний об'єкт у поточний момент часу є активним і знаходиться в стані чекання команди від користувача.

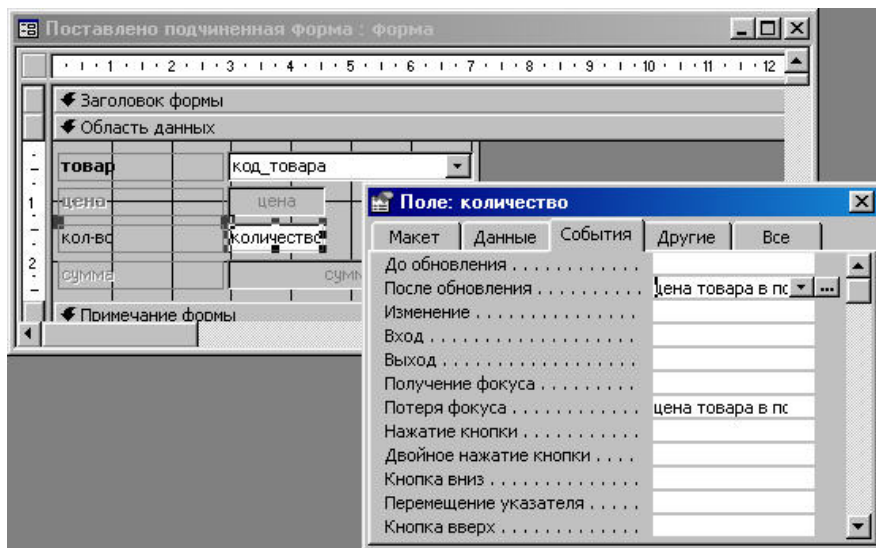


Рисунок 8.6 – Вікно властивостей поля форми

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 8.1.

Таблиця 8.1 – Події форми

Подія	Коли виконуватиметься 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
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

Продовження таблиці 8.1

1	2
Клавіша вниз	При натисканні клавіші на клавіатурі; при запуску макроса <i>КомандиКлавіатури</i>
Клавіша вверх	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

**До уваги!**

Якщо в умові визначена форма, яка невідкрита до моменту виконання макроса, то генерується помилка при виконанні макроса, але не при його створенні.

Макроси, що пов'язані з подіями елементів керування

Макроси можна запускати й на рівні окремих елементів керування, використовуючи для цього відповідні події (табл. 8.2).

Наприклад, можна перевірити правильність вводу 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 8.2 – Події елементів керування

Подія	Коли виконується макрос
1	2
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

Продовження таблиці 8.2

1	2
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. 8.3). 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. 8.4)

Таблиця 8.3 – Події рівня звіту

Подія	Коли виконувється 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 8.4 – Події рівня розділів звіту

Подія	Коли виконувється 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

8.4 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. Гетц К., Литвин П., Гилберт М. *Access 2000. Руководство разработчика. Том 1. Настольные приложения*. К.: ВНУ, 2000. – 1264 с.
2. Горев А., Ахьян Р., Макашарипов С. *Эффективная работа с СУБД*. -Санкт-Петербург: Питер, 1997. -704 с.
3. Карпов Б. *Microsoft Access 2000: Справочник*. – СПб.: Питер, 2000. – 416 с.
4. Кэмпбелл Мери. *Access. Ответы*. – М.: Восточная Книжная Компания, 1996. – 336 с.
5. Нортон П., Андерсон В. *Разработка приложений в Access 97*. Санкт-Петербург: ВНУ, 1999. – 656 с.
6. Праг К.Н., Ирвин М.Р. *Библия пользователя Access 97*. –К.: Диалектика, 1997. – 768 с.
7. Праг К.Н., Ирвин М.Р. *Библия пользователя Access 2000*. – Мю: Вильямс, 2001. – 1040 с.
8. Берлинер Э.М., Глазырин Б.Э., Глазырина И.Б. *Офис от Microsoft*. Москва: АБФ, 1997. - 751 с.



ДОДАТОК А СЛОВНИК ОСНОВНИХ ТЕРМІНІВ

Адміністратор бази даних – особа, яка відповідає за виконання функцій адміністрування бази даних, тобто координацію дій по збору відомостей, проектуванню та експлуатації бази даних, а також по забезпеченню захисту даних.

Альтернативний ключ – атрибут (або група атрибутів), які не співпадають із первинним ключем і унікально ідентифікують кожний рядок у таблиці.

Атрибут – інформаційне відображення властивостей об'єкту.

База даних – поійменована структурована сукупність взаємозв'язаних даних, які відносяться до конкретної предметної області та призначені для задоволення інформаційних потреб багатьох користувачів.

Безпека БД - захист від несанкціонованого доступу, зміни або руйнування даних.

Домен – набір значень елементів даних одного типу, який відповідає поставленим умовам.

Екземпляр сутності – опис конкретного об'єкту в наборі.

Зв'язок – функціональна залежність між сутностями.

Зв'язне відношення – відношення, яке зберігає ключі двох або більше об'єктних відношень, за якими встановлюється зв'язок між цими відношеннями.

Ключовий елемент даних – елемент, по якому можна визначити значення інших елементів даних.

Курсор (віртуальна таблиця) – об'єкт, який не містить власних даних. Це свого роду віртуальна таблиця, що не існує як незалежний об'єкт в базі даних і вміст якої береться з інших таблиць шляхом виконання запиту.

Нормалізація відношень – процес побудови оптимальної структури таблиць і зв'язків у реляційній базі даних.

Об'єкт – елемент інформаційної системи, інформацію про який зберігається.

Об'єктне відношення – відношення, яке зберігає дані про об'єкти (екземпляри сутності) предметної області.

Первинний ключ – атрибут (або група атрибутів), які однозначно ідентифікують кожний рядок у таблиці.

Посилальна цілісність – забезпечення співвідношення значень

зовнішнього ключа екземпляра дочірній сутності значенням первинного ключа в батьківській сутності.

Предметна область – частина реальної системи, яка представляє цікавість для даного дослідження.

Словник даних – централізоване сховище відомостей про об'єкти, елементи даних, які їх складають, взаємозв'язки між об'єктами, їх джерела, значення, використання та формати представлення.

Ступінь відношення – кількість атрибутів (стовпців) відношення.

Сутність – це збиральне поняття, деяка абстракція реально існуючого об'єкта навколишнього світу, процесу або явища.

Таблиця (відношення) – деяка регулярна структура, яка складається з кінцевого набору однотипних записів.

Тип сутності – набір однорідних об'єктів, які виступають як єдине ціле.

Транзакція – це логічна одиниця роботи, яка виконується без порушення логічної цілісності бази даних. Якщо в процесі виконання транзакції виникла помилка виконання, то система, яка підтримує процес транзакції, гарантує повернення до первинного стану.

Тригер – попередньо визначена дія або послідовність дій, які автоматично здійснюються під час виконання операцій оновлення, додавання або вилучання даних.

ODBC (Open Database Connectivity) – відкритий доступ до баз даних – загальне визначення мови та набір протоколів, які дозволяють клієнтському додатку працювати з командами та функціями, що підтримуються сервером.

OLE (Object Linking and Embedding) – зв'язування та впровадження об'єктів – технологія, яка дозволяє використовувати в додатку об'єкти, розроблені в іншому додатку. OLE-об'єктами можуть бути зву, рисунки, діаграми, відеокліпи тощо.

OLE-сервер – програма, яка може надавати іншим програмам можливість використання своїх об'єктів.

SQL (Structured Query Language) – мова структурованих запитів, офіційний стандарт мови для роботи з реляційними базами даних.