

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

Факультет післядипломної освіти

“Проектування баз даних”

Методичні вказівки до
лабораторних робіт

Частина 1

для слухачів спеціальності 7.080403
“Програмне забезпечення автоматизованих систем”

2003

Проектування баз даних. Методичні вказівки до лабораторних робіт. Частина 1. Для слухачів спеціальності 7.080403 “Програмне забезпечення автоматизованих систем” факультету післядипломної освіти /Укл.: С.К.Корнієнко. – Запоріжжя: ЗНТУ, 2003. - 46 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

Методичні вказівки, призначені для використання на лабораторних заняттях із дисципліни "Проектування баз даних", а також в процесі самостійної роботи слухачів факультету післядипломної освіти.

Методичні вказівки містять також практичні рекомендації стосовно проектування баз даних, які можуть бути корисними під час курсового та дипломного проектування.

ЗМІСТ

ВСТУП.....	4
1 Лабораторна робота № 1	
СТВОРЕННЯ ТАБЛИЦЬ БАЗИ ДАНИХ.....	5
2 Лабораторна робота № 2	
РЕДАГУВАННЯ БАЗИ ДАНИХ.....	13
3 Лабораторна робота № 3	
ВСТАНОВЛЕННЯ ЗВ'ЯЗКІВ МІЖ ТАБЛИЦЯМИ ТА АНАЛІЗ ТАБЛИЦЬ У ACCESS.....	22
4 Лабораторна робота № 4	
СТВОРЕННЯ ПРОСТИХ ЗАПИТІВ.....	31
ЛІТЕРАТУРА.....	46

ВСТУП

Microsoft Access – це одна з найбільш популярних на сьогодні настільна система керування базами даних. Система Access – це набір інструментів кінцевого користувача для керування базами даних. До її складу входять конструктори таблиць, форм, запитів і звітів. Цю систему можна розглядати ще й як середовище розробки додатків.

Використовуючи макроси чи модулі для автоматизації рішення задач, можна створювати орієнтовані на користувача додатки такими ж потужними, як і додатки, що написані безпосередньо на мові програмування. При цьому вони будуть включати кнопки, меню та діалогові вікна. Програмуючи на мові VBA, можна створювати такі ж потужні програми, як і сама система Access.

Реляційна обробка даних в Access за рахунок гнучкої архітектури системи здатна задовольнити будь-які потреби. При цьому Access може використовуватися як автономна СКБД, в режимі файл-сервера або клієнтського компонента таких продуктів, як SQL Server. Крім того, Access підтримує протокол **ODBC (Open Database Connectivity)**, що дозволяє підключатися до баз даних різноманітних форматів, таких як **SQL Server, Oracle, Sybase** і навіть **DB/2** для великих ЕОМ фірми IBM.

Система Access підтримує обробку *транзакцій* з гарантією їхньої цілісності. Крім того, передбачений захист на рівні користувача, що дозволяє контролювати доступ до даних окремих користувачів і цілих груп.

Access дозволяє імпортувати та експортувати файли багатьох відомих форматів, включаючи **dBASE, FoxPro, Excel, SQL Server, Oracle, Btrieve**, багато які текстові формати ASCII, а також дані у форматі HTML. В результаті імпортування створюється таблиця Access; в результаті експортування таблиці Access створюється файл у заданому форматі.

1 ЛАБОРАТОРНА РОБОТА № 1

СТВОРЕННЯ ТАБЛИЦЬ БАЗИ ДАНИХ

1.1 Мета роботи

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

1.2 Завдання на лабораторну роботу

1.2.1 Ознайомитися зі змістом пункту 1.3 даних методичних указівок.

1.2.2 Відповідно до індивідуального завдання сформувати структуру бази даних.

1.2.3 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.
- Виберіть спосіб створення бази даних (за допомогою **Конструктора** чи **Майстра**).
- Задайте ім'я файла бази даних і натисніть кнопку **Створити**.
- У вікні бази даних, що з'явилося, виберіть закладку **Таблиці** і 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
Текстовий	Текст, максимальна довжина якого дорівнює 255 символам або визначається значенням властивості "Розмір поля"
Поле МЕМО	Текст, максимальна довжина якого дорівнює 64 Кбайт
Числовий	Числові дані, які використовуються для проведення розрахунків (1, 2, 4 або 8 байт)
Дата/час	Дати та час (8 байт)
Грошовий	Числа, що зберігаються з точністю до 15 знаків у цілій та до 4 знаків у дробовій частині (8 байт)
Лічильник	Число, що автоматично збільшується при додаванні в таблицю кожного нового запису. Значення цих полів не можуть бути змінені (4 байта)
Логічний	Логічні значення, а також поля, що можуть містити одне з двох припустимих значень.
Поле об'єкта OLE	Об'єкт (наприклад, електронна таблиця, малюнок, тощо), якій створений іншим додатком і пов'язаний або введений у таблицю (до 1 Гбайта).
Майстер підстановок	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 кожному з них приписується власний *номер запису*.



Примітка

Номер запису зветься також **первинним ключем**. За допомогою ключів установлюється **відношення (relation)** між двома файлами

Access може призначати такий номер автоматично, приписуючи кожному знову введеному запису індивідуальний номер. Задати автоматичну нумерацію можна за допомогою вибору в якості типу поля значення **Лічильник**.

За допомогою піктограми ключа на стандартній панелі інструментів можна оголосити поле номера запису ключовим полем. Таке оголошення автоматично встановлює для параметра **Індексоване поле** значення **Так (Збіги не пропускаються)**. Завдяки цьому два різні записи в базі не можуть мати однакові номери.



Примітка

Записи можна пронумерувати й самостійно. У цьому випадку треба вибрати для поля номера запису тип **Числовий**, а для параметра **Індексоване поле** - значення **Так (Збіги не пропускаються)**.

При введенні даних у таблицю користувач може вносити в це поле будь-які номери. Access контролює ці дані та при спробі ввести вже використаний номер запису видає попереджуваче повідомлення.



Увага !

Треба уважно поставитися до вибору типу поля при специфікації, тому що зміна типу поля у вже заповненій базі даних може призвести до втрати даних.

1.3.2 Властивості полів

Після вводу імені поля, його типу й опису необхідно задати властивості поля (таблиця 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.

Використання форматів

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 введені ім'я відображалося 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 вводу

Маска призначена для полегшення вводу даних у поле (таблиця 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)

Продовження таблиці 1.4

1	2
&	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
!	Вказує, що маска вводу заповнюється справа наліво. Варто використовувати, якщо в лівій частині маски знаходяться позиції, заповнювати котрі необов'язково. Символ можна розміщувати 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 уводити 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 елементу керування**, для якого є три варіанти вибору: Поле, Список, Поле зі списком.

При завданні **Списку** або **Поля зі списком** необхідно зазначити

джерело, у якому зберігаються варіанти даних, що 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.

1.3.3 Введення даних у таблицю

Після завершення специфікації поля можна приступити до введення даних. Щоб заповнити таблицю, треба перейти з режиму проектування таблиці в режим заповнення. Такий перехід можна здійснити двома шляхами:

- натисканням кнопки **Представлення таблиці** в рядку піктограм (перша зліва в цьому рядку);
- вибором **Режиму таблиці** в пункті **Вид** головного меню.

 Примітка

Специфікація структури таблиці (і внесення змін у цю структуру) проводиться в режимі проектування, а вводу даних (і внесення змін у вміст) - у режимі заповнення.

Access-таблиця завжди містить один порожній запис. Саме до нього й вводиться інформація.

Поле **лічильника** заповнюється програмою автоматично й вже містить змінене значення. Access самостійно збільшує розмір цієї

змінної, привласнюючи кожному такому запису новий номер. Користувачу немає потреби турбуватися про зміст даного поля.

❏ Примітка

При вилучанні запису із таблиці номери записів, які залишилися, не перераховуються

Для вводу інформації в інші поля треба спочатку вибрати поле запису, що заповнюється, за допомогою клавіатури або шляхом фіксації на ньому курсору миші. Перехід до іншого поля фіксує введену в попереднє поле інформацію.

При додаванні до вже заповненої таблиці інших записів можна скористатися командою **Введення даних** у пункті **Записи** головного меню. Завдяки цій команді вже наявні дані не будуть відображатися, і можна приступати до вводу нових даних. При цьому будуть відображатися тільки нові дані.

Відновити перегляд усіх записів можна в пункті **Записи** за допомогою команди **Вилучити фільтр**. Пункт **Записи** головного меню доступний тільки в режимі заповнення таблиці.

1.4 Контрольні питання

- ✓ Які типи даних використовуються в Access?
- ✓ Якою уявою ідентифікуються окремі записи?
- ✓ Як задається режим автоматичної нумерації записів?
- ✓ Як можна додати нові записи?
- ✓ 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?

2 ЛАБОРАТОРНА РОБОТА № 2

РЕДАГУВАННЯ БАЗИ ДАНИХ

2.1 Мета роботи

Метою роботи є ознайомлення з основними прийомами редагування структури та вмісту бази даних, а також виконання операцій сортування, фільтрації та стиску баз даних.

2.2 Завдання на лабораторну роботу

- 2.2.1 Ознайомитися зі змістом пункту 2.3 даних методичних указівок.
- 2.2.2 Змінити структуру бази даних, підготовлену в лабораторній роботі №1.
- 2.2.3 Зробити пошук записів, що задовольняють визначеному критерію.
- 2.2.4 Виконати функцію автоматичної заміни знайденої інформації.
- 2.2.5 Здійснити функції заглушення та заморожування стовпчиків.
- 2.2.6 Зробити сортування та фільтрацію записів за різноманітними критеріями.
- 2.2.7 Зробити стиск бази даних.

2.3 Основні теоретичні відомості

2.3.1 Редагування бази даних

Переміщення по таблиці (навігація)

У Access користувачеві доступні різноманітні засоби ***переміщення по таблиці (навігації)***, під якими розуміються як перехід від запису до запису, так і перехід у записі від поля до поля.

Найбільш ефективною є навігація за допомогою навігаційного блока, розташованого в лівому нижньому куті таблиці. Для переходу на визначений запис достатньо ввести його номер у поле індикатора номера запису.

Редагування записів

Дані можуть редагуватися в режимі заміни або режимі вставки.

У **режимі заміни** поле запису вибирається за допомогою клавіш керування курсором або клавіші **[Tab]** і маркується цілком. Нові дані можна відразу вводити на місце старих або попередньо вилучити за допомогою клавіші **[Del]**.

У **режимі вставки** поле обирається щигликом миші, а потім на місце курсору, що з'явився, вставляється потрібний символ.

Переключення між зазначеними режимами провадиться за допомогою клавіші **[Ins]**.

Операції пошуку

Пошук інформації проводиться за допомогою команди **Знайти** меню **Правка** головного меню або за допомогою кнопки із зображенням бінокля в рядку піктограм. Коло пошуку інформації можна істотно розширити, використовуючи символи шаблонів **"*" і "?"**. По замовчанню пошук проводиться тільки в активній колонці (опція **Тільки в поточному полі** включена), що дозволяє швидше одержати результат.

Якщо необхідно зробити пошук по всій таблиці, то опцію **Тільки в поточному полі** треба відключити. Якщо об'єкт пошуку буде знайдений, Access виділить уміст знайденого поля, а в статусному рядку з'явиться повідомлення **Пошук виконаний**.

Пошук і заміна інформації провадиться за допомогою команди **Замінити** меню **Правка**.



Примітка

Якщо після виконання пошуку на екрані не з'явиться марковане поле, треба перемістити діалогове вікно пошуку. Можливо, знайдене поле знаходиться в тій області таблиці, що перекрита цим вікном.

Вилучання даних

Для видалення запису з бази даних (таблиці) потрібно попередньо маркірувати його, а потім скористатися командою **Правка/Видалити** або клавішею **[Del]**. Для вилучання стовпчиків таблиці (тобто полів у записах) треба звернутися до режиму проектування таблиці, оскільки при цьому змінюється специфікація

таблиці.

Зміни в структурі таблиці

Зміни в структурі таблиці, що стосуються типу окремих полів, їхнього порядку, формату й інших параметрів, виконуються в режимі проектування таблиці.



Увага!

Усі зміни в структурі запису вже заповненої таблиці чреваті втратою даних. Тому до внесення таких змін варто удаватися тільки у випадку крайньої необхідності.

Зміна зображення таблиці

Зовнішній вигляд таблиці, тобто те, що користувач бачить на екрані, можна змінювати, не вносячи змін у структуру та не звертаючись до режиму проектування.

Наприклад, якщо потрібно встановити ширину колонки за максимальним розміром даних, треба встановити стрілку миші на праву межу заголовка стовпчика та двічі натиснути кнопку миші.

Можна також викликати команду ***Ширина стовпчика*** меню ***Формат*** і натиснути кнопку ***За шириною даних***. Аналогічно можна змінити висоту рядка, скориставшись командою ***Висота рядка*** меню ***Формат***.

Зміна розташування рядків/колонок

У процесі опрацювання вже готової таблиці може виникнути необхідність зміни, вставки, переносу та вилучання колонок.

Для переміщення колонки таблиці в інше місце необхідно виділити її та, утримуючи натиснутою ліву кнопку миші на заголовку виділеної колонки, перетягнути її на нове місце. При цьому введені в таблицю дані не спотворюються та не загублюються.

Зміна послідовності полів можна виконати й у режимі проектування таблиці.

Крім того, для прискорення процесу проектування таблиці, що має поля з ідентичними параметрами, можна використовувати команду копіювання. Для цього необхідно:

- активізувати режим проектування таблиці;
- маркувати рядок (поле) специфікації, що копіюється;
- вибрати команду **Правка/Копіювати**;
- маркувати рядок, перед яким повинна бути вставлена копія;
- активізувати команду вставки **Правка/Вставити**.

Вставлена копія з'явиться перед маркованим рядком. Необхідно подбати про зміну її імені, оскільки в таблиці не можуть розміщатися поля з однаковими іменами.

Функції копіювання або переносу допомагають при створенні однотипних таблиць. Якщо структури, що вже були визначені в одній таблиці, будуть використані й в інших таблицях, то користувач може скопіювати довільні області з однієї таблиці до іншої.

Копіювати з однієї специфікації до іншої можна в такий спосіб:

- створити нову таблицю за допомогою кнопки **Створити** вікна бази даних.
- звернути вікно бази даних до піктограми;
- розмістити вікна таблиць на екрані за допомогою команди **Вікно/Зліва праворуч**;
- маркірувати область із декількох полів у вихідній таблиці і скопіювати її за допомогою команди **Правка/Копіювання** в буфер проміжного збереження;
- виконати щиголь на новій таблиці та вибрати команду **Правка/Вставити**.

При копіюванні полів із специфікації однієї таблиці в специфікацію іншої немає необхідності в зміні імен, оскільки мова йде про дві незалежні таблиці.

При цьому можна скопіювати як цілі записи, так і вміст одного поля. Копіювання вмісту декількох полів одного запису неможливо.

Необхідною умовою для копіювання цілих записів є ідентичність структур обох таблиць, приймаючих участь у процесі копіювання. Для вставки даних у таблицю, що приймає, використовується команда **Правка/Спеціальна вставка**.



Увага !

Записи таблиці, що не можуть бути вставлені в цільову таблицю, тимчасово, щоб не загубилися, зберігаються в окремій таблиці з ім'ям **Таблиця помилок вставки**. Відтіля вони пізніше можуть бути вставлені в цільову таблицю, наприклад, після редагування таблиці.

Зміна типу поля

Якщо в таблицю були внесені дані, то внесення змін до її специфікації може призвести до появи неточностей, пов'язаних із конвертацією даних.



Примітка

При зміні імені поля дані не конвертуються.



Увага !

Access забороняє конвертування будь-яких типів даних у тип **Лічильник**. Цей тип даних використовується й опрацьовується спеціальним чином.

Проблеми виникають у тому випадку, коли провадиться перетворення текстового типу даних у числовий. У цьому випадку буде вилучений уміст усіх полів, що містять хоча б одну букву.

При оберненому перетворенні проблем не виникає, оскільки текстові поля можуть містити як цифри, так і букви, а числові поля - винятково числа.

У випадку перетворення поля з типом даних **Мето** у текстове поле дані можуть бути загублені, якщо розмір текстового поля виявиться недостатнім, щоб прийняти весь уміст Мето-поля. У цьому випадку "зайві" символи просто відкидаються.

Індикація та заглушення стовпчиків

У базах даних із великою кількістю полів у запису частину інформації не видно на екрані. Для рішення цієї проблеми можна зменшити розмір шрифту або впорядкувати стовпчики, згрупувавши разом найбільш необхідні.

Крім того, непотрібні в даний момент стовпчики можна заглушити (виключити індикацію). Для цього необхідно:

- Маркувати стовпчик або декілька стовпчиків, що повинні бути заглушені.
- Вибрати команду **Формат/Сховати стовпчики**. Маркований стовпчик вилучиться з екрану, але не з таблиці.



Увага !

Сховані стовпчики не беруть участь у виконанні операцій пошуку й опрацюванні запитів.

Щоб сховані стовпчики зробити знову видимими, треба вибрати команду **Формат/Показати стовпчики**. Після цього в діалоговому вікні відобразиться список стовпчиків активної таблиці. Видимі стовпчики будуть відзначені. У цьому вікні можна виконати як заглушення, так і відновлення індикації - достатньо вилучити або встановити контрольний індикатор проти відповідного стовпчика.

Інший спосіб одержання кращого вигляду даних у таблиці полягає в застосуванні функції **закріплення (фіксації) стовпчиків**, що дозволяє зафіксувати на лівому краї екрану один або групу стовпчиків, незалежно від того, у якій області таблиці в даний момент працює користувач. Для цього потрібно:

- Маркувати стовпчик у таблиці.
- Вибрати команду **Формат/Закріпити стовпчики**.

За допомогою цієї команди маркований стовпчик переміщується до зовнішньої (лівої) позиції в таблиці. Тепер, якщо користувач буде рухатися за допомогою лінійки прокручування в праву область таблиці, фіксований стовпчик залишиться видимим на екрані. Скасування фіксації виконується командою **Формат/Звільнити всі стовпчики**.

Примітка

Після скасування фіксації автоматичне повернення стовпчика в його вихідну позицію не відбувається. Тому користувач повинний перенести його у відповідну позицію за допомогою миші.

2.3.2 Сорткування та фільтрація записів

При пошуку деякого значення Access знаходить і відображає відповідний запис. Але якщо заданому критерію задовольняє декілька записів, то для зручності роботи можна відобразити на екрані тільки їх за допомогою кнопок сорткування та фільтрації, що знаходяться на панелі інструментів (або команд **Сорткування** та **Фільтр** із меню **Записи**). Для сорткування всієї таблиці використовуються дві кнопки сорткування, а для вибору визначених записів - три кнопки фільтрації.

Швидке сорткування

Для вибору поля сорткування достатньо помістити в нього курсор на рівні будь-якого запису. Після чого потрібно клацнути на відповідній кнопці, й дані будуть миттєво відсортовані. Для сорткування даних по декількох полях необхідно виділити відповідні стовпчики. Якщо стовпчики несумісні, то потрібно їх перемістити, щоб вони розташовувалися поруч.

Для відновлення початкового порядку розташування записів потрібно використовувати команду **Вилучити фільтр** у меню **Записи**.

Фільтр за виділенням

Фільтр за виділенням - це засіб швидкого відбору записів за виділенням зразком. Наприклад, нам необхідно виділити всі записи, у яких атрибут **Тип товару** має значення "продукти". Для цього потрібно встановити курсор на поле будь-якого запису, що містить значення "продукти" та натиснути кнопку **Фільтр за виділенням**. При цьому на екрані залишаться тільки записи, що містять зазначене значення.

Функція **Фільтр за виділенням** має властивість накопичення критеріїв вибору. Це означає, що продовжуючи вибирати значення за

допомогою кліків на кнопці **Фільтр за виділенням**, можна добавляти до фільтра нові критерії.

Примітка

Функція фільтрації за виділенням має декілька обмежень, найголовніше з яких полягає у додаванні критеріїв один до одного, тобто можна здійснювати добір тільки тих записів, що задовольняють зразу усім заданим критеріям вибору.

Розширений фільтр

Розширений фільтр дозволяє вводити більш складні запити. Він викликається командою **Записи** ⇒ **Фільтр** ⇒ **Розширений фільтр**. Далі виконується відбір потрібних полів таблиці та завдання їм певних значень, як показано на рисунку 2.1.

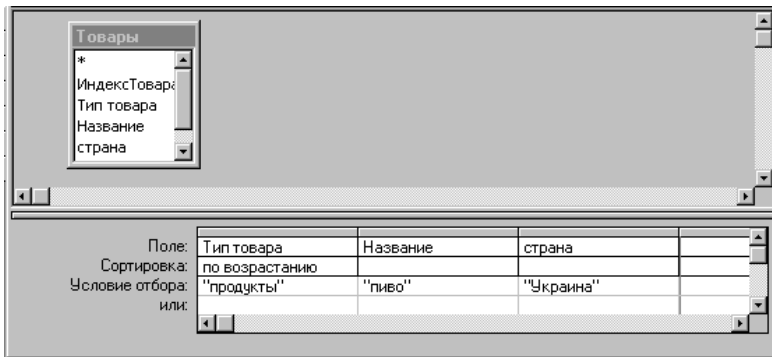


Рисунок 2.1 - Вікно розширеного фільтра

Запуск фільтра здійснюється за допомогою кнопки **Запуск Фільтра**  панелі інструментів.

2.3.3 Стиск бази даних

Операція стиску бази даних використовується для звільнення місця на диску та прискорення доступу. Її доцільно періодично

застосовувати до бази після внесення доповнень і вилучання непотрібних записів. Стиск упорядковує базу, усуває фрагментацію та зменшує місце, що займається. Після стиску стара база не вилучається, а створиться її нова версія.

Стиск виконується командою **Сервіс⇒Службові програми⇒Стиснути базу даних**. Якщо стискується відкрита база або при стиску закритої бази вона зберігається під старим ім'ям, то після успішного закінчення стиску Access автоматично вилучить початкову (незжату) базу.

Крім того, "Access може автоматично стискувати базу даних після закінчення роботи з нею. Ця функція задається командою **Сервіс⇒Параметри⇒Загальні⇒Стискувати при закритті**.

2.4 Контрольні питання

- ✓ Яким чином здійснюється навігація по таблиці?
- ✓ Чим відрізняється режим заміни від режиму вставки даних?
- ✓ Як здійснити пошук потрібної інформації?
- ✓ Як зробити заміну одних об'єктів іншими?
- ✓ Як здійснити видалення записів?
- ✓ Чим відрізняються команди Видалити і Вирізувати?
- ✓ Як переставити рядки, стовпчики?
- ✓ Як видалити рядки, стовпи?
- ✓ Як копіювати інформацію з однієї таблиці до іншої?
- ✓ Особливості процедури заміни типу поля.
- ✓ Як виконати операції придушення та заморожування стовпчиків?
- ✓ Як проводиться сортування даних?
- ✓ Якими засобами здійснюється фільтрація?
- ✓ Як можна стиснути базу даних?

3 ЛАБОРАТОРНА РОБОТА № 3

ВСТАНОВЛЕННЯ ЗВ'ЯЗКІВ МІЖ ТАБЛИЦЯМИ ТА АНАЛІЗ ТАБЛИЦЬ У ACCESS

3.1 Мета роботи

Метою роботи є одержання практичних навичок розробки багатотабличних баз даних шляхом встановлення і редагування зв'язків між таблицями, а також вивчення методології автоматичної нормалізації баз даних.

3.2 Завдання на лабораторну роботу

3.2.1 Ознайомитися з теоретичним матеріалом.

3.2.2 Розробити макет схеми даних.

3.2.3 Вивчити можливості реалізації зв'язків **1:1**, **1:M** і **M:N**.

3.2.4 За допомогою аналізатора таблиць зробити декомпозицію складної таблиці на ряд пов'язаних таблиць.

3.3 Основні теоретичні відомості

3.3.1 Ключі

Для встановлення зв'язків між таблицями необхідно встановити зв'язок між тими полями, у яких міститься загальна 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 установлюється 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.

Якщо поле є зовнішнім ключем, то воно пов'язано з таблицею, на яку посилається, тобто кожне значення в цьому полі (зовнішньому ключі) безпосередньо пов'язано зі значенням в іншому полі (батьківському ключі). Це значить, що **таблиця знаходиться в стані посилальної цілісності**.

Зовнішній ключ, як і первинний, може бути визначений на будь-якій кількості полів, що 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. Це означає, що кожне значення зовнішнього ключа повинно бути подане в батьківському ключі один і тільки один раз. Як тільки значення розміщується в зовнішньому ключі, батьківський ключ перевіряється, щоб переконатися в тому, що таке значення в ньому є присутнім; у іншому випадку команда відхиляється.

Поле ключа, первинного чи зовнішнього, перевіряється незалежно від того, добавляється воно, змінюється або вилючається. Якщо зміна ключа порушує зв'язок, виходить, що воно порушує цілісність даних.

3.3.2 Встановлення зв'язків

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**.

Для виклику вікна **Зв'язки** та додавання до нього таблиць потрібно:

1. Клацнути на кнопці **Схема даних** панелі інструментів. При цьому Access відчинить діалогове вікно **Додавання таблиці**.
2. Вибрати потрібні таблиці.
3. Клацнути на кнопці **Закрити** вікна **Додавання таблиці**. Екран буде виглядати як на рисунку 3.1.

Установка зв'язків між таблицями

Перш за все, слід мати на увазі, що таблиці зв'язуються лише за полем, яке є загальним (за змістом, 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**, показаний на рисунку 3.2.

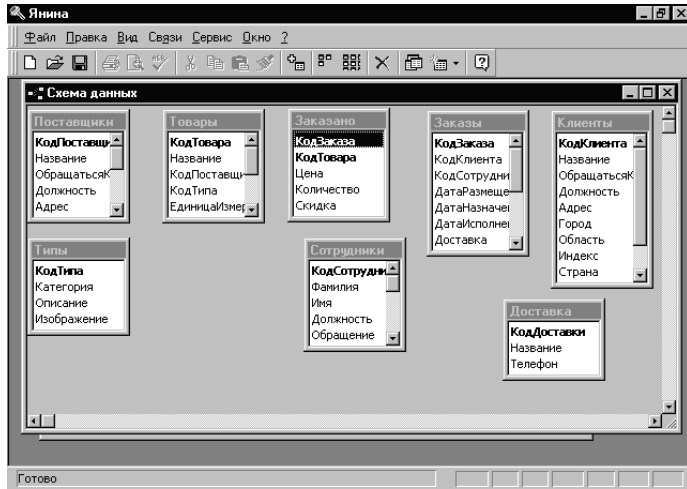


Рисунок 3.1 - Початкове вікно "Схема даних"

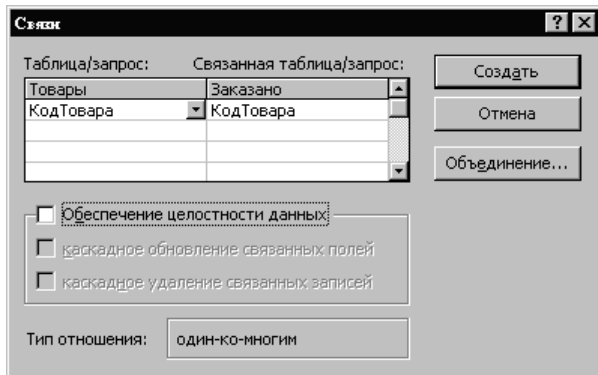


Рисунок 3.2 - Діалогове вікно Зв'язки

У вікні **Зв'язки** міститься інформація про поля зв'язку головної та підпорядкованої таблиць, параметрах забезпечення цілісності даних і типі 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**. Ця опція дає можливість змінювати вміст поля зв'язку (ключове поле в первинній таблиці, наприклад, **Код товару** в таблиці **Товари** на рисунку 6.3).

Якщо користувач змінює вміст ключового поля в первинній таблиці, Access перевіряє коректність значення в цьому полі (у первинній таблиці не повинно бути однакових записів) і змінює значення відповідних полів у всіх пов'язаних таблицях. Таким чином, зміна значень ключового поля в первинній таблиці поширюється на всі пов'язані таблиці.

**Увага**

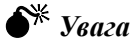
Якщо опція **Каскадне оновлення пов'язаних полів** не включена, змінити значення ключового поля первинної таблиці не можна.

Каскадне вилучання пов'язаних записів

При включеній опції **Забезпечення цілісності даних** у вікні **Зв'язки**, Access дозволяє включити опцію **Каскадне вилучання пов'язаних записів**. Ця опція дає можливість вилучити записи в пов'язаних таблицях, а потім - запис у первинній таблиці.

🔍 Примітка

Якщо поле ключа в первинній таблиці є полем зв'язку між декількома таблицями, то для правильної роботи опції каскадного відновлення пов'язаних полів і видалення пов'язаних записів необхідно включити у всіх пов'язаних таблицях.

**Увага**

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

Для вилучання існуючого зв'язку потрібно зайти у вікно *Схема даних*, виділити зв'язок, що вилучають, і вилучити цей зв'язок за допомогою або кнопки на панелі інструментів, або команди **Видалити** в контекстному меню. Остаточний вид схеми бази даних показаний на рисунку 3.3.

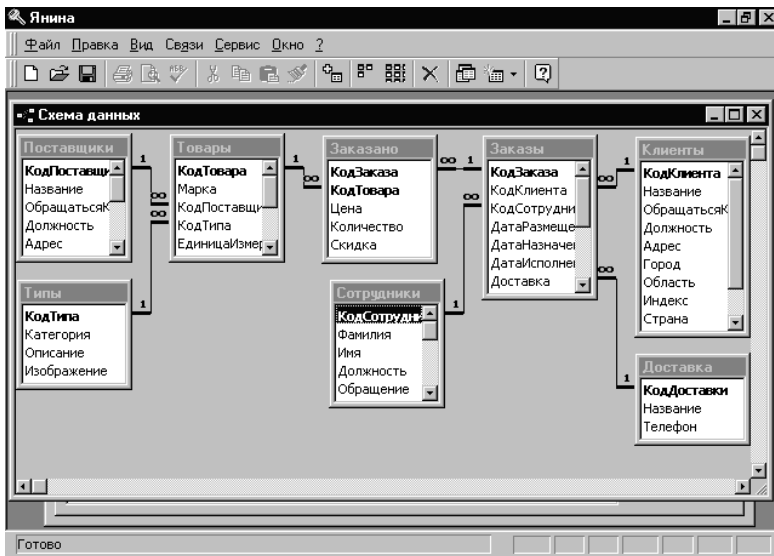


Рисунок 3.3 - Відображення зв'язків між таблицями

3.3.3 Аналіз таблиць у Access

Бази даних можна імпортувати до СКБД Access із файлів формату Excel або текстових файлів. Після імпортування дані в Access розміщуються в одну плоску таблицю. В Access є засіб аналізу та приведення єдиної таблиці до нормалізованого виду. Аналізатор таблиць допомагає розбити плоску таблицю даних на декілька невеличких пов'язаних таблиць (усунувши надмірність), створити

первинні й зовнішні ключі та перевірити орфографію.

Аналізатор таблиць можна запустити за допомогою команди Сервіс⇒Аналіз⇒Таблиця. При цьому з'явиться вікно постановки питань аналізатора таблиць, а потім вікно розв'язування питань аналізатора таблиць. Виведення цих вікон можна блокувати, вимкнувши опцію **"Вивести сторінки з поясненнями?"**.

Після ознайомлення зі змістом цих треба варто перейти до вікна вибору таблиці (рис. 3.4).

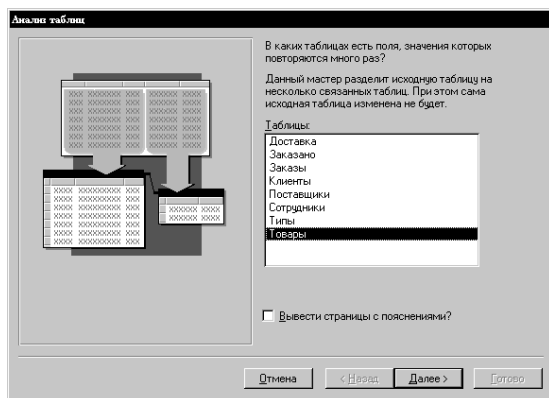


Рисунок 3.4 - Вибір таблиці для аналізу

Обравши таблицю, натисніть на кнопку *Далі*. У такому вікні аналізатор таблиць запропонує виконати аналіз таблиці. Аналіз таблиці може бути виконаний двома способами: у першому випадку аналізатор бере основну частину роботи на себе й іноді по ходу аналізу задає питання; у другому випадку аналізатор відчиняє вікно, схоже на вікно *Зв'язки*, у якому можна перетаскувати поля з однієї таблиці до іншої, автоматично встановлюючи зв'язок між таблицями.

Початкова таблиця показана на рисунку 3.5. Переміщати поля між таблицями можна за допомогою миші. Якщо при перетягуванні поля розмістити його на вільній області вікна аналізатора, буде створена нова таблиця. Переміщаючи поля усередині однієї таблиці, можна змінювати їхній порядок. На малюнку 3.6 показаний результат роботи аналізатора.

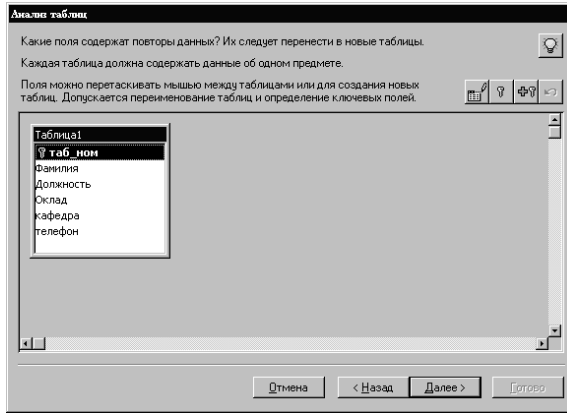


Рисунок 3.5 - Вихідна таблиця перед декомпозицією

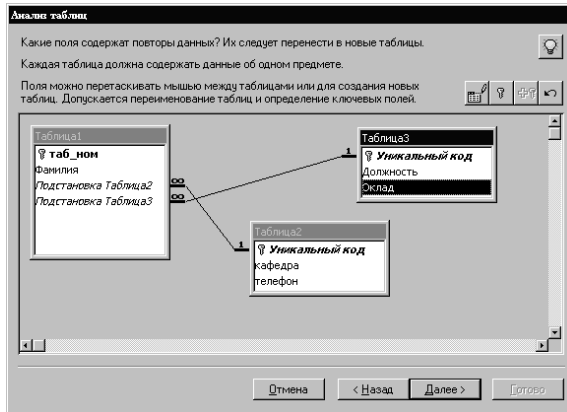


Рисунок 3.6 – Результати роботи аналізатора таблиць

Закінчивши роботу в цьому вікні, треба перейти до наступного, натиснувши кнопку *Далі*. У цьому вікні можна змінювати ключові поля, які створені аналізатором таблиць. На рисунку 3.7 показаний остаточний варіант оформлення розбивки вихідної таблиці.

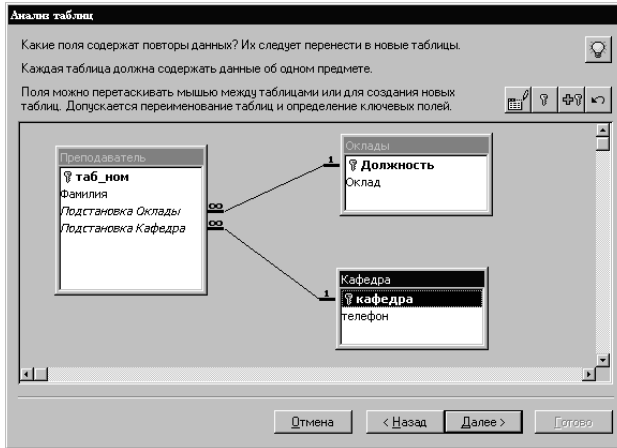


Рисунок 3.7 - Остаточный вариант разбивки таблицы

Після налаштування полів первинних ключів треба перейти далі для аналізу помилок. При виконанні аналізу помилок перевіряється також вигляд обраних вами полів.

3.4 Контрольні питання

- ✓ Що таке *первинні* та *зовнішні ключі*?
- ✓ Яким чином встановлюються та вилучаються зв'язки між таблицями?
- ✓ Поняття *цілісності даних*; як воно підтримується в Access?
- ✓ Яким чином встановлюється *Каскадне вилучання пов'язаних полів*?
- ✓ Як здійснити *Каскадне оновлення пов'язаних полів*?
- ✓ Для чого використовується функція аналізу таблиць у Access?
- ✓ У якій послідовності виконується аналіз таблиць?

4 ЛАБОРАТОРНА РОБОТА № 4 СТВОРЕННЯ ПРОСТИХ ЗАПИТІВ

4.1 Мета роботи

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

4.2 Завдання на лабораторну роботу

4.2.1 Ознайомитися зі змістом пункту 4.3 даних методичних указівок.

4.2.2 Для бази даних, яка створена в попередніх лабораторних роботах, за узгодженням із викладачем розробити різноманітні типи запитів, використовуючи можливі варіанти їхнього створення.

4.3 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.

4.3.1 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. При цьому припускається зміна записів у базовій таблиці (при деяких обмеженнях)

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 додавляє до бланку запиту рядок Групова операція
Запит на зміну	Дозволяє створювати нові таблиці (команда Створення таблиці) або змінювати дані в існуючих таблицях (команди Вилучання , Оновлення та Додавання). Якщо в процесі виконання запиту на вибірку можна змінити тільки один запис, то запит-дія дозволяє вносити зміни в декілька записів відразу при виконанні однієї операції.
Перехресний запит	Відображає результати статистичних розрахунків (такі як суми, кількість записів і середнє значення), які виконані за даними з одного поля. Ці результати групуються за двома наборами даних у форматі перехресної таблиці. Перший набір виводиться в стовпчик зліва й утворює заголовки рядків, а другий виводиться у верхньому рядку й утворює заголовки стовпчиків.
SQL	Це запит, який створений за допомогою інструкції SQL. Прикладами SQL-запитів є <i>запит на з'єднання</i> , <i>запит до серверу</i> , <i>керуючий запит</i> , <i>тощо</i> .
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.

4.3.2 Створення запитів на вибірку

Запит на вибірку називають також **запитом за зразком (QBE - Query By Example)**. Він складається шляхом специфікації окремих параметрів у вікні проектування з використанням підказок (зразків).

QBE-запит є стандартним типом запиту. Для його проектування

потрібно у вікні бази даних вибрати розділ *Запити*, а потім натиснути на кнопці *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 запиту;
- *Записи без підпорядкованих* - створюється запит, у якому відображаються ті записи з однієї таблиці, що не мають зв'язків із записами іншої таблиці (підпорядкованої).

Щоб створити запит без допомоги майстра запитів 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 (рисунк 4.1):

- області таблиць запиту;
- бланка запитів.

Область таблиць запиту - це місце, де розміщаються 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.

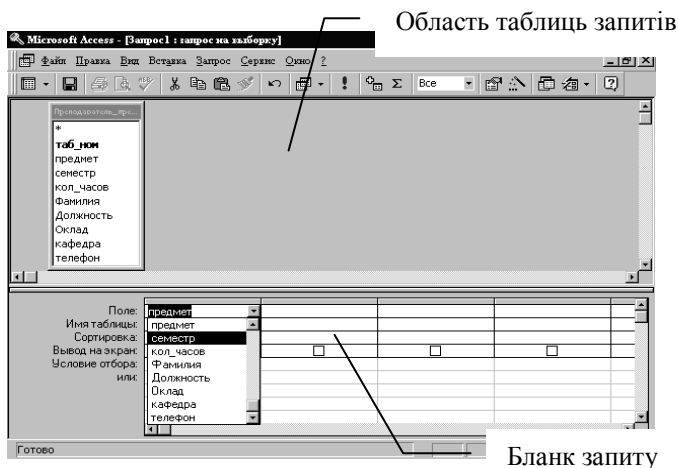


Рисунок 4.1 - Вікно конструктора запитів

Бланк запиту містить шість рядків:

Поле	- Ім'я поля
Ім'я таблиці	- Ім'я таблиці
Сортування	- Місце запровадження інструкцій сортування
Виведення на екран	- Визначає, чи буде присутнім поле в динамічному наборі даних
Умова відбору	- Містить першу умову, що обмежує набір записів
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 ****.

Для кращого сприйняття результатів виконання запиту можна перейменувати поля в динамічному наборі даних. Для цього в рядку **Поле** бланка запиту потрібно перед старим ім'ям уставити нове ім'я, розділивши їх символом ":".

Примітка

Зміна імені поля призводить лише до зміни заголовка цього поля в динамічному наборі. Ім'я поля в базовій таблиці залишається незмінним.

Відображення поля

При виборі поля для нього автоматично включається опція **Виведення на екран**. Скасування виведення проводиться вручну у відповідному рядку бланка запиту.

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 сортують

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

При створенні запиту на вибірку з декількох таблиць необхідно враховувати обмеження, що накладаються на можливість редагування полів. У принципі, дані можна змінювати в отриманому після виконання запиту динамічному наборі даних, і ці зміни будуть збережені у базових таблицях. Проте якщо встановлений прапорець **Забезпечення цілісності даних** і ключове поле є частиною зв'язку, то редагувати дані в цьому полі не можна (якщо не буде встановлений прапорець **Каскадне оновлення зв'язків**).

Для оновлення базової таблиці з запиту, величина у даному записі запиту повинна відповідати одному запису у базовій таблиці. Це значить, що не можна оновити поле в перехресних запитах або в запитах із виконання підсумкових обчислень, оскільки в них записи об'єднані для відображення групової інформації. Замість відображення даних із базової таблиці вони показують записи, що були відібрані та збережені у **віртуальній** (реально не існуючій) таблиці. Віртуальна таблиця є нібито миттєвим відображенням відібраних записів.

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, над яким ви працюєте.

**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.

Типи зв'язків між таблицями

Між таблицями й запитами 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 умовчання. Означає, що з таблиць вибираються тільки записи, що мають однакове значення в обох таблицях.

Access може також знайти втрачені записи, тобто записи в одній таблиці, що не мають підпорядкованих записів в іншій таблиці. У цьому випадку при створенні запиту у вікні **Новий запит** варто вибрати опцію **Записи без підпорядкованих**.

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 (наприклад, чи повинні відображатися записи без підпорядкованих).

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.4.2).

У вікні є три перемикачі. Перший варіант зазвичай називають **внутрішнім** (**INNER JOIN** у термінах мови **SQL**) з'єднанням, а два інших - **зовнішнім**. Ці з'єднання управляють роботою Access при побудові динамічного набору даних.

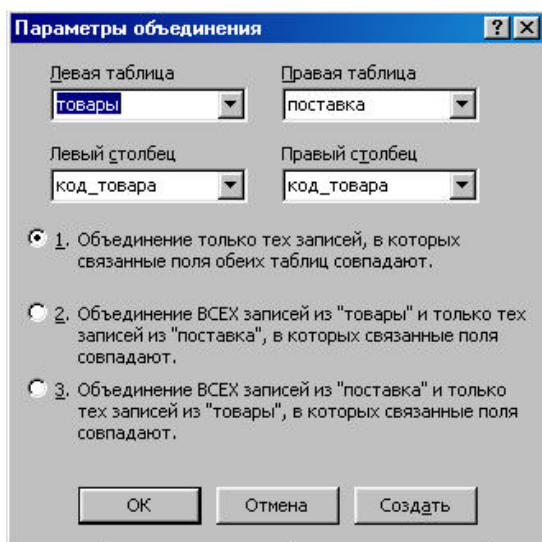


Рисунок 4.2 - Діалогове вікно "Параметри з'єднання"

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.

З'єднання двох копій однієї і тієї ж таблиці в запиті

З'єднання двох копій однієї таблиці означає наступне: будь-який рядок однієї таблиці (один в кожен момент часу) можна комбінувати з його ж копією або з будь-яким іншим рядком тієї ж таблиці.

СКБД виконує цю операцію так, як і над двома таблицями, що мають різні назви. Тому для таблиць треба задати псевдоніми, або *aliases* (*alias*). В якості аліаса можна використовувати будь-яке ім'я, навіть для зручності одну літеру.

При з'єднанні двох копій однієї таблиці користувач створює *самоз'єднання*, що з'єднує записи однієї й тієї ж таблиці, якщо поля, що з'єднуються, містять однакові значення.

Наприклад, є таблиця "*Співробітники*", у якій для кожного співробітника в поле "*Звітується*" замість прізвища вказаний код його менеджера. Самоз'єднання можна використовувати для відображення в записі кожного співробітника прізвища його менеджера. Для цього потрібно:

- У режимі конструктора двічі додайте таблицю до запиту.
- До імені другої таблиці автоматично додається "_1".
- Переіменувати таблицю так, щоб її назва була більш описовою, або більш зручною для вводу можна, встановивши для списку полів властивість *Псевдонім* (*Alias*), як на рисунку 4.3.

✎ Примітка

Переіменування копій таблиці (або запиту) в запиті не призводить до переіменування базової таблиці або запиту.

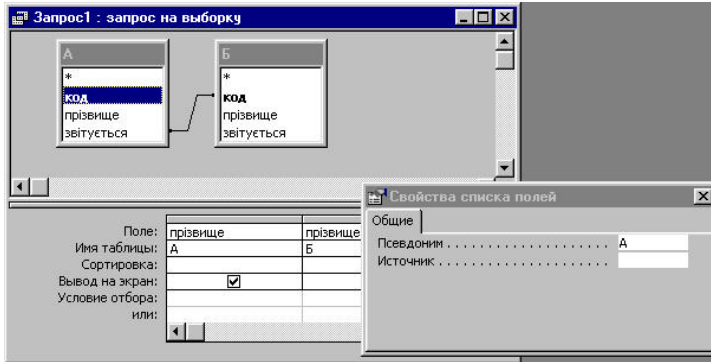


Рисунок 4.3 –Вікно Конструктора запитів при самоз'єднанні

4.3.3 Вибір за складними умовами відбору записів

Запровадження умов відбору записів за одним полем

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 4.1 і таблиця 4.2).



Увага

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

Таблиця 4.1 - 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 4.2 – 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 умови відбору записів для об'єкта OLE

Для об'єкта OLE також можна ввести умову відбору, а саме - умова наявності непорожнього запису **IS NOT NULL**. Наприклад, вибрати тільки ті записи, у яких значення в полі малюнка **NOT NULL**.

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 запитамі ім'я поля треба брати в квадратні дужки,

як при створенні полів, що 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 однім осередку поля бланка запиту (рис. 4.4), або використовуючи рядок *або* (рис. 4.5).

наименование	цена	поставщик
Товары	Товары	Товары
☒	☒	☒
"монитор" Or "принтер" Or "процессор"		

Рисунок 4.4 - Використання оператора **OR** в умовах відбору записів

наименование	цена	поставщик
Товары	Товары	Товары
☒	☒	☒
"монитор"		
"принтер"		
"процессор"		

Рисунок 4.5 - Використання рядка *або* (**OR**) бланка запиту

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 по декількох полях

Для використання операторів **AND** і **OR** у запиті по декількох полях 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. 4.6).

наименование	цена	поставщик	
Товары	Товары	Товары	
☒	☒	☒	☐
"монитор"	>500	"LG"	
"принтер"	>300	"HP"	
"процессор"	<300	"Intel"	

Рисунок 4. 6 - Приклад використання операторів **AND** і **OR** у запиті

У даному прикладі умова задається формулою:

("монитор" **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 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.

4.4 Контрольні питання

- ✓ Що таке "запит"?
- ✓ Які типи запитів підтримує Access?
- ✓ Призначення і склад бланка запитів.
- ✓ Вставка і вилучання полів із бланка 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 об'єднань між таблицями?
- ✓ Що називається внутрішнім з'єднанням?
- ✓ Що називається зовнішнім з'єднанням?
- ✓ Що називається самоз'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 обчислюється?

ЛІТЕРАТУРА

1. Гетц К., Литвин П., Гилберт М. *Access 2000. Руководство разработчика. Том 1. Настольные приложения*. К.: BHV, 2000. – 1264 с.
2. Горев А., Ахаян 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: BHV, 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.

