

**МІНІСТЕРСТВО АГРАРНОЇ ПОЛІТИКИ УКРАЇНИ
ВІННИЦЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ**

Кафедра економічної
кібернетики та інформатики

Навчально-методичний посібник

Частина III:

“Створення і зміна запитів Access”

*з дисципліни: “Інформатика та комп’ютерна техніка”, “Інформатика”
та “Комп’ютери та комп’ютерні технології”
для виконання самостійної роботи студентами денної та заочної форми
навчання факультетів механізації сільського господарства,
обліково-фінансового, економічного, агрономічного, та факультету
технології виробництва та переробки продукції рослинництва
Вінницького державного аграрного університету*

Вінниця - 2006

ЗМІСТ

СТВОРЕННЯ І ЗМІНА ЗАПИТІВ	3
ПРОСТИЙ ЗАПИТ	3
КОНСТРУКТОР ЗАПИТІВ	6
“ПОСТРОИТЕЛЬ ВЫРАЖЕНИЙ” (БУДІВНИК ВИРАЗІВ)	9
ПЕРЕХРЕСНИЙ ЗАПИТ	13
ВИБИРАННЯ ПОВТОРЮВАНИХ ЗАПИСІВ.....	17
РЕДАГУВАННЯ ДАНИХ ЗА ДОПОМОГОЮ ЗАПИТІВ.....	21
ЗАПИТ НА ДОБАВЛЯННЯ	21
ЗАПИТ НА ВІДНОВЛЕННЯ	24
ЗАПИТ НА СТВОРЕННЯ ТАБЛИЦІ	25
СТВОРЕННЯ ОБЧИСЛЮВАНИХ ПОЛІВ	26
ФІЛЬТРИ.....	28

Створення та зміна запитів

Створення і зміна запитів

У першу чергу під *запитом* розуміється операція відбору даних із таблиці, але це лише одна з багатьох можливостей запитів. У сучасних програмах вирішені практично всі проблеми, пов'язані зі зберіганням даних, і на перший план виступає задача представлення користувачам зрозумілих і ефективних засобів доступу до тих даних, що їх цікавлять у певний момент. Проблема тут полягає в тому, що більшість кінцевих користувачів, що не мають спеціальної підготовки в області інформатики, сприймають комп'ютер і працюючі на ньому програми як якісь дуже складноорганізовані "чорні ящики", у надрах яких незбагненим чином схована інформація, на введення якої вони витратили стільки часу і сил. Незважаючи на те, що в новітніх СКБД існують дуже прості засоби фільтрації і відбору даних, для використання цих засобів користувач повинен перебороти певний психологічний бар'єр. Як це не дивно, багатьом людям складно зрозуміти, що в сучасній програмі є все, необхідне для успішного розв'язання більшості задач, і що нема рації звати на поміч програміста, щоб отримати доступ до своїх даних.

Простий запит

Ми можемо використовувати базу даних **Test1.mdb** у якості вихідного матеріалу для створення запиту.

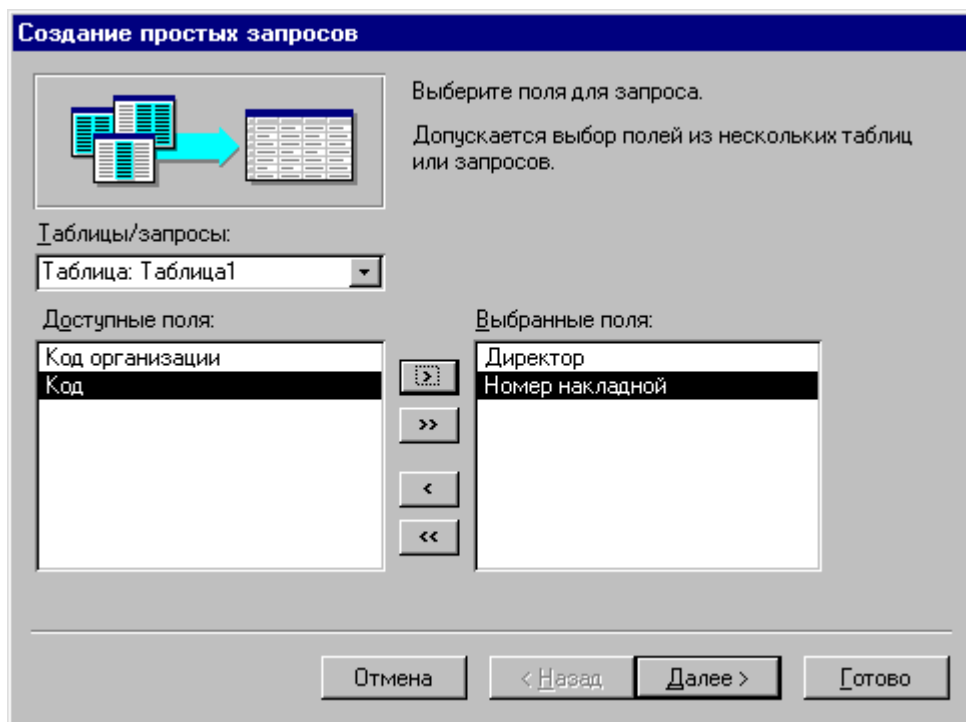


Рис. 4.1. Перший крок Майстра простих запитів

Відкриємо вкладку **Запросы** у вікні бази даних, натиснемо кнопку **Создать**, виберемо **Простой запрос** і клацнемо кнопку **ОК**. Перший крок Майстра, що дозволяє побудувати простий запит, показаний на рис. 4.1. На цьому етапі треба

вибрати таблиці, до яких буде адресований запит, і потрібні поля цих таблиць. Ми вкажемо поля [Таблица2]![Директор] і [Таблица 2]![Номер накладной] .

На другому кроці Майстра, показаному на рис.4.2, є можливість визначити вид звіту: докладний ("подробный") або підсумковий ("итоговый").

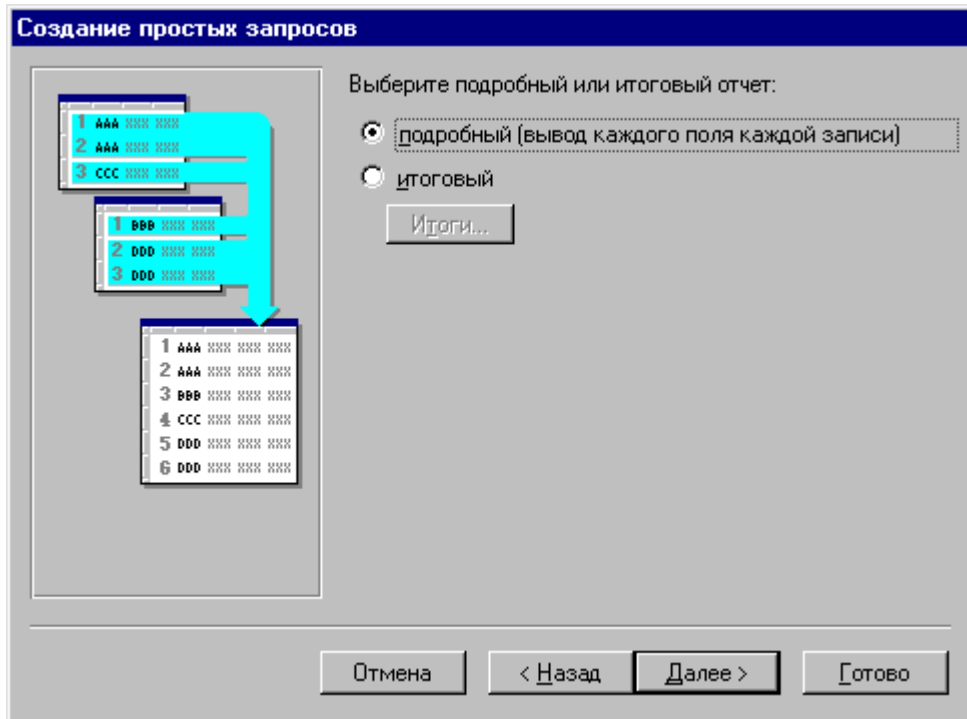


Рис.4.2. Другий крок Майстра простих запитів

Виберемо варіант "подробный" і перейдемо до наступного кроку. Він показаний на рис.4.3. Можна змінити ім'я нового запиту, але в даному випадку ми залишимо те, що пропонує Майстер.

Тепер саме час подивитися на те, що в нас вийшло. Натиснемо кнопку ГОТОВО. Після того, як робота Майстра завершиться, програма відкриє знову створену таблицю-запит. Її вміст показаний на рис.4.4. За термінологією Access - це "подробный отчет". Ми бачимо, що програма створила нову таблицю, у яку ввійшли вибрані нами поля. Використавши існуючий між вихідними таблицями зв'язок, СКБД змогла визначити значення полів для всіх записів. Кожному полю [Таблица1]![Номер накладної] було поставлено у відповідність поле [Таблица 2]![Директор]. На результат виконаної операції варто звернути увагу. Використовуючи дуже прості засоби ми об'єднали поля двох таблиць. Вийшла нова, *віртуальна* таблиця, що має властивості запиту. Цей метод варто взяти на озброєння.

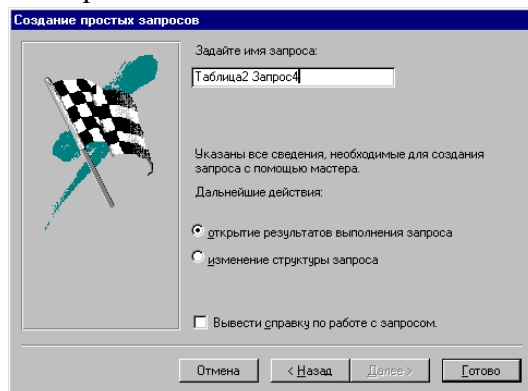


Рис.4.3. Третій крок Майстра простих запитів

Директор	Номер накладной
Петров	19
Петров	15
Петров	3
Иванов	2
Иванов	4
Иванов	8
Иванов	9
Сидоров	7
Сидоров	22
Сидоров	11

Рис.4.4. Результат роботи Майстра простих запитів

Якби ми вибрали на другому кроці майстра простих запитів варіант "итоговый отчет", це дозволило б нам підрахувати, скільки накладних оформлено для кожної організації і порівняти ці дані з прізвищами директорів організацій. У цьому випадку на другому кроці Майстра крім встановлення перемикача в положення "итоговый" потрібно натиснути кнопку "Результаты" і встановити прапорець "Подсчет записей" у вікні **Результаты**, зображеному на рис.4.5. Зверніть увагу, що жодна з доступних функцій (сума, середнє арифметичне, мінімум, максимум) для обчислення результатів у даному випадку непридатна. У результаті запит буде мати такий вигляд, який показано на рис.4.6. Ми бачимо, що програма провела групування й узагальнення даних. У полі **First_Номер накладной** вказані номери найближчих до початку таблиці **Таблица1** номерів документів, а в полі **Count_Таблица1** - число накладних для кожної організації.

Рис.4.5. Вікно *Результаты*

Директор	First_Номер	Count_Таблица1
Иванов	4	3
Петров	2	4
Сидоров	7	3

Рис.4.6. Підсумковий звіт за запитом

Вправа 1. Створіть за допомогою Майстра запитів запит по виборці альбомів конкретного виконавця в БД “Фонотека”.

Конструктор запитів

Повернемося до запиту на вибірку, який показано на рис.4.4., виділимо його, натиснемо кнопку **Конструктор** на вкладці **Запросы**. Вікно, яке з'явиться після цієї дії, зображене на рис.4.7. Це основний засіб роботи з запитом. Він дозволяє не лише сформулювати новий запит, але і зрозуміти, за яким принципом побудований будь-який із уже існуючих. У верхній половині вікна показані зв'язки між таблицями. Як бачимо, програма використала лише частину схеми даних нашого прикладу - ту, що має безпосереднє відношення до запиту. Користувач може редагувати цю схему, додаючи до неї нові об'єкти: таблиці або запити.

Поле:	Директор	Номер накладной	
Имя таблицы:	Таблица2	Таблица1	
Групповая операция:	Группировка	Группировка	
Сортировка:		по возрастанию	
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:	"Петров"		
или:			

Рис.4.7. Режим конструктора для запиту

Нижня частина вікна - *бланк запиту* - містить опис запиту в табличній формі.

Кожна колонка в ньому відповідає одному полю. Рядки **Поле** й **Имя таблицы** містять списки, що дозволяють визначити потрібне поле. Їхні елементи відповідають

розміщеній угорі схемі, тому користувач завжди може легко дізнатися, які поля йому доступні. Таблиця розширюється вправо автоматично, коли користувач додає в запит нові поля. Контекстне меню, пов'язане з бланком запиту, дозволяє приховувати або, навпаки, показувати рядки **Имя таблицы** і **Групповая операция**.

Заповнивши рядок **Условие отбора** ми зможемо, нарешті, провести вибірку з бази даних. Припустимо, ми хочемо переглянути всі записи, що стосуються пана Петрова. У колонку **Директор** треба ввести умову відбору - «**Петров**», як показано на рис.4.7. Цієї інформації достатньо, щоб програма відібрала всі рядки, де в полі **Директор** записане «Петров». Символьні константи належить брати в лапки. Але навіть якби ви ввели прізвище без лапок, програма все рівно зрозуміла б Вас і поставила лапки сама. За аналогією, у колонку бланка запиту **Номер накладной** можна було б записати умову **>10**, і програма відібрала б усі записи, у яких значення вказаного поля більше, ніж 10. Access на рідкість «розумна» програма. Для того, щоб побачити результат запиту, достатньо переключити режим його перегляду за допомогою списку **Вид**, розташованого на панелі **Конструктор запитів**. У режимі конструктора ця панель присутня на екрані за замовчуванням. Результат вибірки показаний на рис. 4.8.

Директор	Номер накладной
Петров	2
Петров	3
Петров	15
Петров	19

Рис.4.8. Результат виконання запиту

Для того, щоб освоїти технологію запитів, не потрібні більш складні приклади. Але в реальних завданнях зустрічаються набагато складніші запити. Їх складність полягає у формулюванні умови відбору записів. Вона може являти собою дуже запутаний логічний вираз. Для того, щоб скласти його, використовують "построитель выражений". Він запускається через контекстне меню, пов'язане з рядком **Условие отбора**, за командою **Построить**. Роботі з "построителем выражений" присвячений наступний розділ.

Рядок **Групповая операция** містить список функцій і багато в чому визначає вид даних запиту. Під **груповою операцією** розуміється оброблення й узагальнення значення даного поля за допомогою певної функції. У результаті в таблиці запиту змінюється розташування або кількість записів. Наприклад, запити, показані на рис.4.4 і рис.4.6, використовують ту саму вихідну таблицю, але відрізняються тим, що в першому для поля **Номер накладной** в якості групової операції визначена функція **Группировка**, а в другому для поля **Count_Таблица1** (яке є перетвореним полем **Номер накладной**) - функція **Count** (підрахунок числа записів).

Групова операція дозволяє легко створити обчислюване поле. Припустимо, ми хочемо підрахувати в запиті, скільки разів зустрічається кожне значення деякого поля. Для цього потрібно створити в бланку ще одну колонку для цього поля і визначити для неї групову операцію **Count** (підрахунок).

На рис. 4.6. ми бачимо, що програма автоматично створює нові імена полів, комбінуючи назви функції для групування й імена вихідних полів запиту. Якщо вам не подобається, що таблиця-запит має такі «комп'ютеризовані» імена полів і хочеться

бачити в заголовках колонок що-небудь більш осмислене, потрібно виконати таку послідовність дій:

1. Клацніть правою кнопкою миші на бланку запиту в потрібній колонці (у режимі конструктора).
2. З контекстного меню виберіть команду **Свойства**.
3. На вкладці **Общие**, що відкрилася, в рядку **Подпись** введіть бажаний заголовок. Тепер, переключившись у режим перегляду запиту, Ви побачите ім'я стовпця те, яке Вам потрібне.

Призначення рядка **Сортировка** на бланку запиту очевидне – він дозволяє відсортувати отримані в результаті виконання запиту дані. Наприклад, у запиті, показаному на рис.4.8., номери накладних відсортовані за зростанням, тому що на бланці запиту був заданий режим сортування (рис.4.7).

Ще один спосіб "конструювання" запиту - зміна його формулювання мовою SQL. Коли користувач конструює запит, програма перекладає останній на мову SQL. Текст запиту можна побачити, якщо вибрати команду **Режим SQL** зі списку **Вид** або ту саму команду з меню **Вид**. Цей текст можна відредагувати і, отже, змінити запит. Наприклад, запит, зображений на рис.4.7, має таке формулювання мовою SQL:

```
SELECT DISTINCTROW Таблица2. Директор, Count (Таблица1. [Номер накладной]) AS [Counter накладной]
FROM Таблица 2 LEFT JOIN Таблица1 ON Таблица2. Код = Таблица1.
[Код организации]
GROUP BY Таблица2. Директор;
```

Весь запит помістився в одну команду. Тому, вона вийшла досить довгою через ряд опцій. Дуже велике число запитів може бути виражене за допомогою команди **SELECT**. Вважають, що це головна команда мови.

Вправа 2. Створіть параметричний запит (з застосуванням поля "Условие отбора" в бланку запиту) за виборкою виконавця, який виконує даний (введіть назву) альбом.

"Построитель выражений" (Будівник виразів)

У попередньому прикладі нам потрібно було відібрати ті записи з таблиці, у яких в поле **Директор** було введено "Петров". Інакше сформулювати умову відбору можливо було, використовуючи "построитель выражений", показаний на рис. 4.9. Він запускається командою **Построить** з контекстного меню, пов'язаного з рядком **Условие отбора** на бланці запиту.

При складанні виразу необхідно пам'ятати декілька простих правил. По-перше, вираз завжди має логічний тип. Тобто в будь-якому випадку його значення повинно бути **Да** або **Нет**. У бланку запиту звичайно опускається частина виразу, що містить ім'я поля, тому що воно задане в тій самій колонці. Знак рівності також не пишеться за замовчуванням. Таким чином, умова відбору **[Директор] = "Петров"** записується як *літерал* "Петров". По-друге, існують певні вимоги до синтаксису виразу. Наприклад, імена об'єктів і полів поміщаються у квадратні дужки, символічні рядки – в лапки, дати – в знаки #.

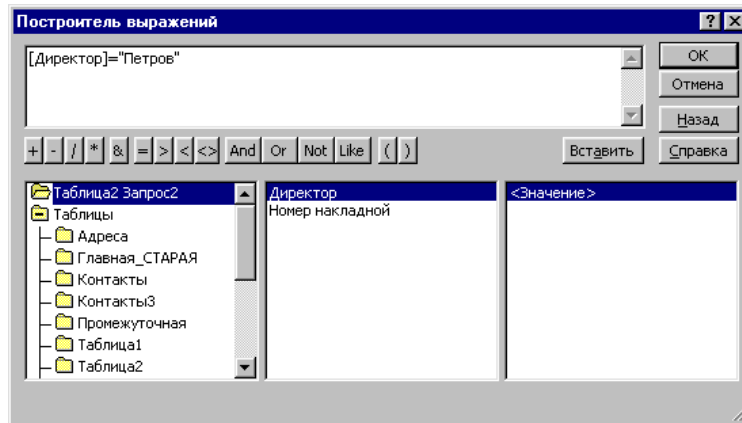


Рис.4.9. Вікно "построителя виражень"

Ім'я об'єкта бази даних відокремлюється від імені поля знаком оклику. По-третє, є декілька "нематематичних" операторів, використання яких добре відоме програмістам, але не завжди – кінцевим користувачам.

Коли Ви будете вирази для запиту, то повинні уявляти собі, що створюєте нову таблицю, використовуючи вже наявну інформацію в базі даних. Причому конструюєте таблицю насправді не Ви, а програма, яка використовує введений Вами логічний вираз як критерій відбору даних. Тому вираз повинен бути точно визначеним і містити всю інформацію про те, які записи потрібно включити в таблицю-запит.

Вікно "построителя" має чотири області зі своїми смугами прокручування. У верхній області розташовується утворюваний вираз. Три нижні використовуються для вибору елементів. Вони заповнюються за ієрархічним принципом. Ліва область містить список усіх джерел даних для запиту. Як очевидно з рис.4.9, ними можуть бути об'єкти бази даних Access: таблиці, форми, запити і т.д. Для полегшення пошуку потрібної інформації список поданий у тому ж вигляді, як дерево каталогів у Windows. Крім цього, у лівий список включені папки, які містять функції, константи й оператори, можуть бути використані у виразі. Середній список служить для показу елементів, які входять у вибраний об'єкт із лівого списку. Наприклад, якщо в лівій області вказане ім'я конкретної таблиці (наприклад, Таблица2) - не плутати з рядком Таблицы - то середній список буде містити імена її полів. Якщо вибрана папка з функціями, то середній список буде містити перелік груп функцій. І, нарешті, правий список використовується для вибору найпримітивніших об'єктів. Для того, щоб перенести ім'я об'єкта у вираз, потрібно два рази клацнути ім'я об'єкта в одному із нижніх списків. Кнопка **Назад** дозволяє скасувати введення даних у вираз. Текст виразу можна редагувати і звичайним способом. Кнопки з зображеними на них знаками математичних операцій дозволяють швидко вводити у вираз відповідні символи.

Крім традиційних знаків математичних дій існує ще декілька операторів:

Between And - Замінює знаки "більше або дорівнює" і "менше або дорівнює".

Наприклад, умова **Between 1981 And 1994** еквівалентна умові **>=1981 And <=1994**.

Оператор **In** дозволяє використовувати *списки значень*. Наприклад, вираз **In ("Петров"; "Сидоров")** дозволяє відібрати записи з вказаними прізвищами.

Вираз **Not Null (Is Not Null)** використовується для відбору *непустих* записів у вказаному полі.

Вираз **Is Null** використовується для відбору *порожніх* записів у вказаному полі.

\ - Обернена коса риска позначає ділення цілих частин ділимого і дільника. Результат округляється до цілого.

^ - Позначає операцію піднесення до степеня.

Mod - Операція одержання залишку від ділення цілих частин аргументів.

Знак **&** використовується для "додавання" даних символного типу. Для з'єднання рядків можна використовувати і більш звичний знак "плюс". Наприклад, наступні три вирази еквівалентні:

"Петров"

"Петро" & "ов"

"Петро" + "ов"

Оператор **Like** використовується для створення *масок* при визначенні рядків із невідомими символами і потребує додаткових спеціальних символів і правил синтаксису:

? - Знак питання позначає будь-який поодинокий символ. Наприклад, якщо невідомо, як написати: Carl або Karl, то умову можна задати як **Like "? arl"**.

* - Зірочка позначає будь-яку послідовність символів.

- Позначає будь-яку невідому цифру.

[] - Квадратні дужки використовують, якщо перед виконанням запиту повинно бути активізоване вікно діалогу з запрошенням ввести параметр, що є критерієм відбору. Вираз у дужках виводиться в цьому вікні у якості повідомлення-підказування.

Наприклад, при створенні запиту для таблиці **Телефон1** із БД **Test1** можна було б записати таку умову відбору для поля код станції: **[Введи код станції від 100 до 999]**.

Логічні оператори також використовуються для побудови умов відбору. Вони можуть застосовуватися до двох або декількох виразів і використовуються з дужками. У табл.4.1 показані значення основних логічних функцій. Крім них у Access використовується оператор логічного заперечення **Not**. Умова **Not "Іванов"** дозволяє вибрати всі рядки, які не містять вказаний літерал у відповідному полі. Результат виконання логічної операції **Imp** (імплікація) залежить від порядку слідування операторів.

Деякі логічні функції двох змінних. Таблиця 4.1.

Функція	Значення операндів							
	Ні	Ні	Ні	Так	Так	Ні	Так	Так
And	Ні		Ні		Ні		Так	
Egv	Так		Ні		Ні		Так	
Or	Ні		Так		Так		Так	
Xor	Ні		Так		Так		Ні	

Оператори "менше" і "більше" можуть використовуватися не лише з числовими даними, але і з датами і/або текстом. Стосовно до текстових рядків, вони дозволяють відбирати записи, що ідуть перед вказаним рядком або за ним. У цих випадках враховується алфавітний порядок значень поля. Наприклад, вираз **[Город]>"Москва"** припускає, що в полі **Город** треба відібрати всі назви, що стоять в алфавітному списку після Москви.

Якби сформульована умова відбору однозначно визначала, що вийде при представленні запиту, то користуватися Access була б легко і приємно. На жаль, на

результат запиту впливають і зв'язки між таблицями. Перевіримо це на простому прикладі.

Створимо запит на основі таблиці **Контакты** з бази даних **Test1**. Відберемо ті записи, що містять букву "н" у полі **Фамилия**. Вікно конструктора цього запиту показано на рис.4.10, а результат його виконання - на рис.4.11. Умова відбору в даному випадку: **Like "*н*"**. Як і слід було очікувати, відібрані три прізвища.

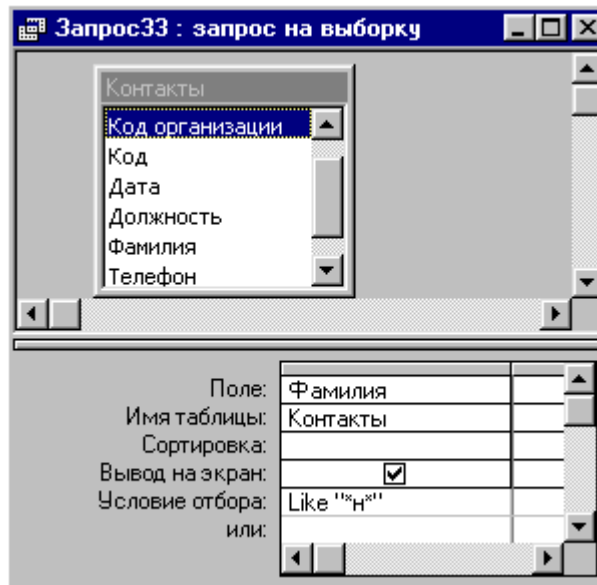


Рис.4.10. Запит із використанням одиночної таблиці

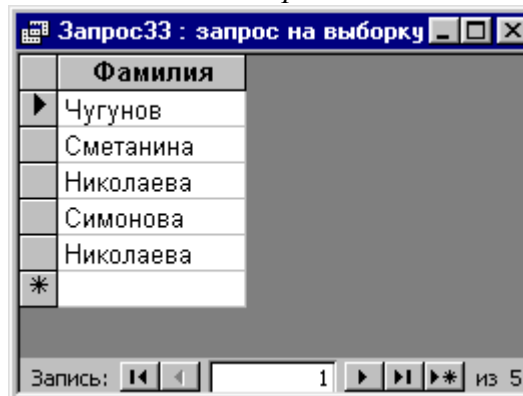


Рис.4.11. Результат запиту з використанням одиночної таблиці

Тепер додамо в запит ще одну таблицю - **Таблица2**. Це можна зробити таким чином. Якщо тепер перед вами результат виконання попереднього запиту (див. рис.4.11), то спочатку необхідно повернутися в режим конструктора, виконавши команду **Вид Конструктор**. Натисніть кнопку **Добавить таблицу**, розташовану на панелі інструментів (на цій кнопці зображений знак "плюс" і значок таблиці). Виберіть ім'я таблиці і натисніть кнопку **Добавить**, а потім кнопку **Закри́ть** у вікні **Добавление таблицы**. Список полів таблиці буде приміщений у схему запиту, як показано на рис.4.12. Оскільки в базі даних між цими таблицями існує зв'язок, то він відображений також й у вікні запиту. Незважаючи на те, що в сам бланк не були внесені ніякі зміни, результат запиту буде тепер іншим - таким, як показано на рис.4.13. На реальній схемі даних таблиці **Контакты** і **Таблица2** не зв'язані напруму.

Дійсно, програма автоматично встановила зв'язок між ними, коли друга таблиця була додана в схему запиту. Ви можете відновити правильну схему даних, спостерігаючи, як змінюється результат запиту до таблиці **Контакты** після додавання в схему чергової таблиці. Працювати зі схемою запиту можна так само, як і зі схемою бази даних (тобто редагувати зв'язки, добавляти і видаляти таблиці).

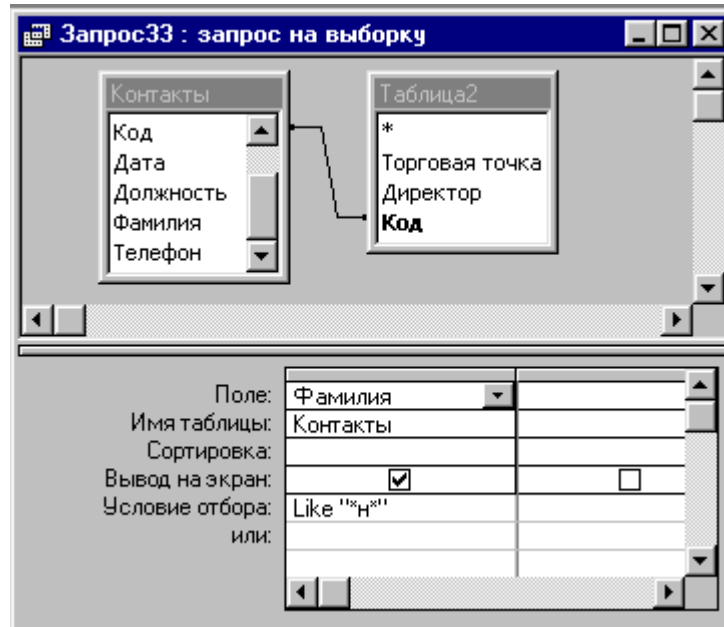


Рис.4.12. Запит з урахуванням зв'язку між таблицями

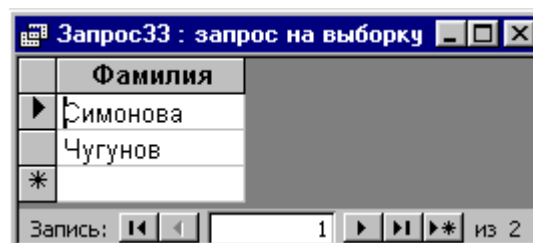


Рис.4.13. Результат запиту з урахуванням зв'язку між таблицями

У реальній базі даних схема може бути досить складною, а таблиці - великими, тому вплив зв'язків на результат запиту не завжди легко простежити. Для перевірки виконання запиту під час розроблення бази даних можна створювати тимчасові запити з використанням однієї таблиці, як було показано вище.

Вправа 3. Створіть запит за виборкою стилів, які містять в своїй назві літеру "к".

Перехресний запит

Перехресний запит - це операція побудови таблиці для обчислення підсумкових значень на основі існуючої таблиці або запиту. Перехресний запит створюється в припущенні, що вихідна таблиця містить необхідні дані для формування заголовків рядків і стовпців нової таблиці. Для ілюстрації Майстра перехресних запитів розглянемо таблицю **Контакты** з бази даних **Test1**. За допомогою перехресного запиту можна простежити кількість звертань до співробітників різних фірм, використовуючи поле **Дата** (мається на увазі дата звертання до співробітника фірми).

Для запуску Майстра перехресного запиту потрібно натиснути кнопку **Создать** на вкладці **Запросы**, вибрати зі списку **Перекрестный запрос** і натиснути кнопку **ОК**.

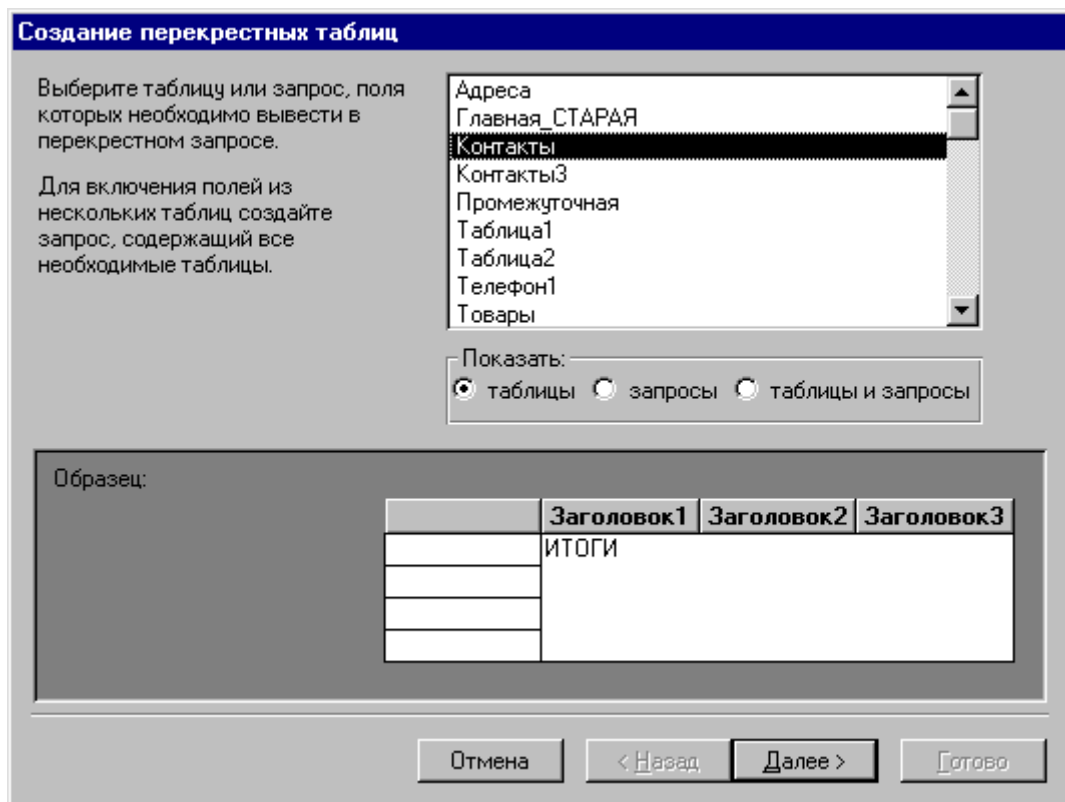


Рис.4.14. Перший крок Майстра перехресних запитів

На першому кроці Майстра, показаному на рис.4.14, потрібно вибрати таблицю. Якщо необхідні для запиту поля знаходяться в різних таблицях, то спочатку необхідно створити запит, який створить віртуальну таблицю з усіма потрібними полями. Ми ж виберемо **Контакты** і перейдемо до наступного кроку, натиснувши на кнопку **Далее**.

На другому кроці Майстра (рис.4.15) необхідно вибрати поле (поля), значення якого (яких) будуть використані для створення рядків таблиці. Якщо поле містить повторювані значення, то в перехресному запиті буде виконане групування даних. Ми вкажемо поле **Фамилия**. Недосить виділити потрібне поле - його необхідно перенести в список **Выбранные поля** зі списку **Доступные поля**, натиснувши кнопку зі значком **>**. Програма відразу ж показує, як будуть розташовуватися в таблиці значення поля. На третьому кроці Майстра (рис.4.16) необхідно вибрати поле (поля), значення якого (яких) будуть використані для створення стовпців таблиці.

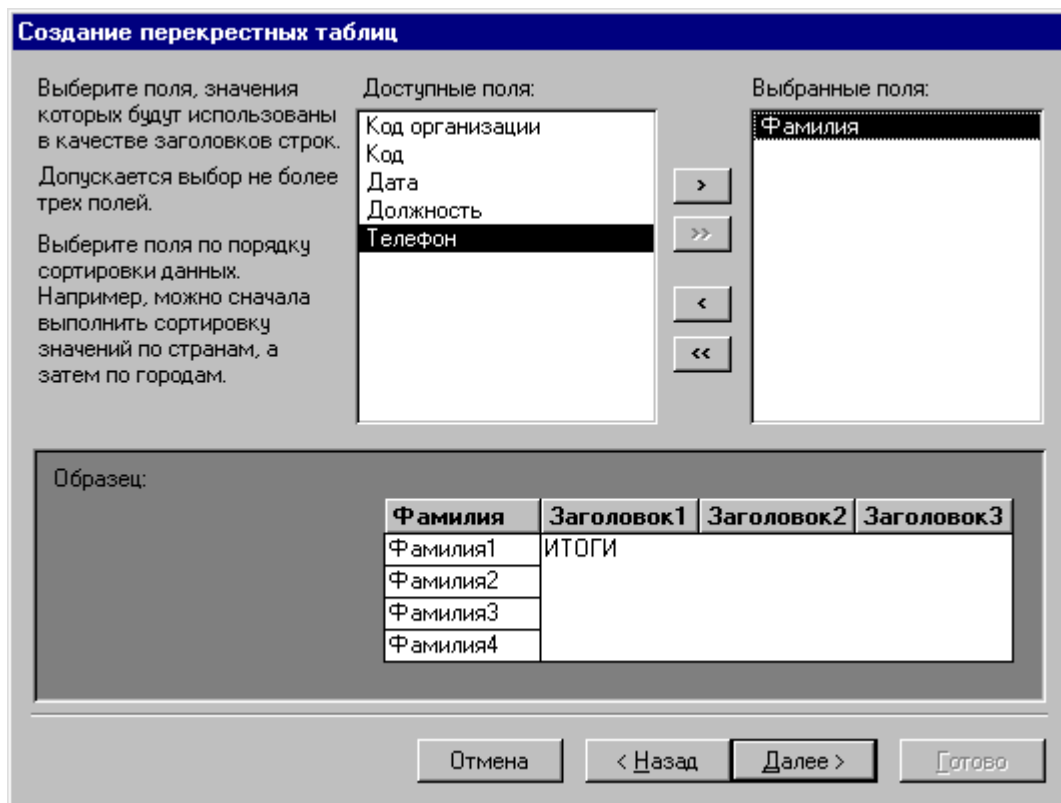


Рис.4.15. Другий крок Майстра перехресних запитів

Четвертий крок Майстра логічно пов'язаний із попереднім. Цей крок змінюється в залежності від того, яке поле вибрав користувач. У нашому випадку Майстер не став автоматично присвоювати заголовкам колонок значення дат, а запропонував розбити часовий інтервал на рівні проміжки. Людина повинна лише вибрати протяжність інтервалу: рік, квартал або місяць. Звичайно, можна вказати й ім'я поля (у нашому випадку **Дата**) і тоді в заголовках колонок будуть вказані конкретні дати. Виберемо "Місяць" і перейдемо до наступного кроку Майстра.

На п'ятому кроці Майстра (рис.4.18) необхідно вибрати функцію, яка буде використовуватися для підведення результатів. У нашому випадку можна застосувати функцію **Count** і підрахувати кількість значень для поля **Код організації**.

На шостому, останньому кроці Майстра (див. рис.4.19) не потрібно погоджуватися з запропонованим варіантом назви запиту "Перекрестный запрос", а краще використати інше ім'я, наприклад, **Контакты22**. Тепер можна натискати кнопку **Готово**.

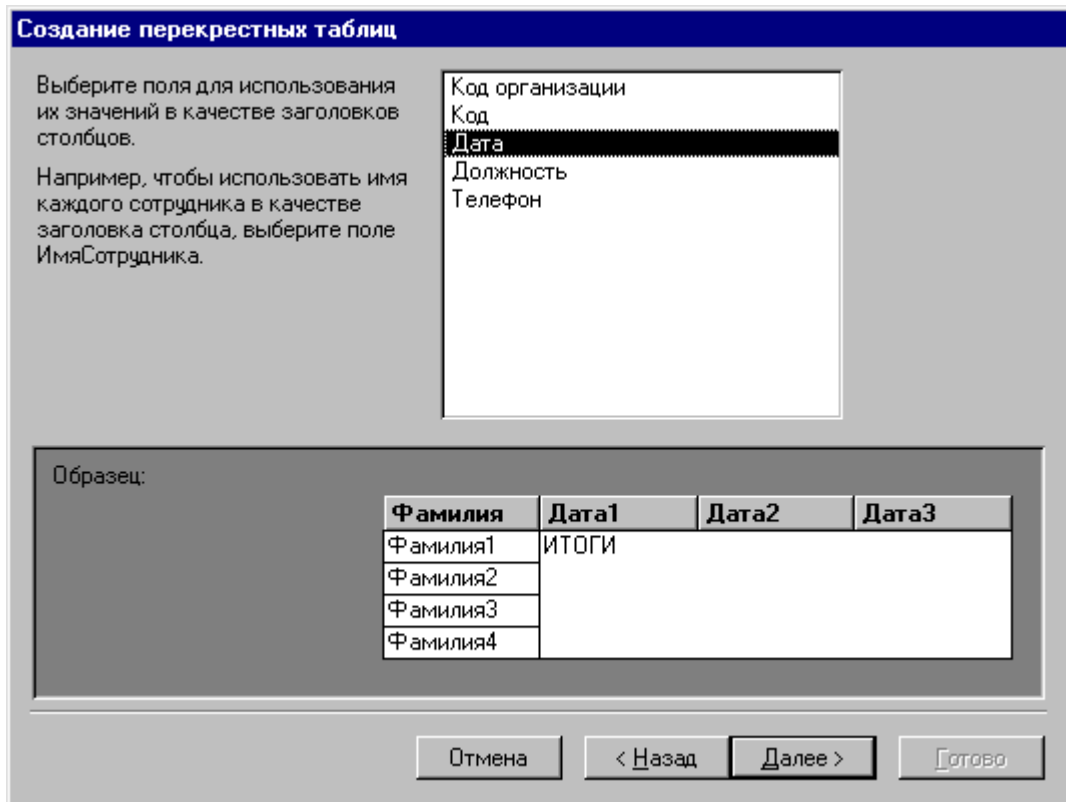


Рис 4.16. Третий шаг Мастера перекрестных запросов

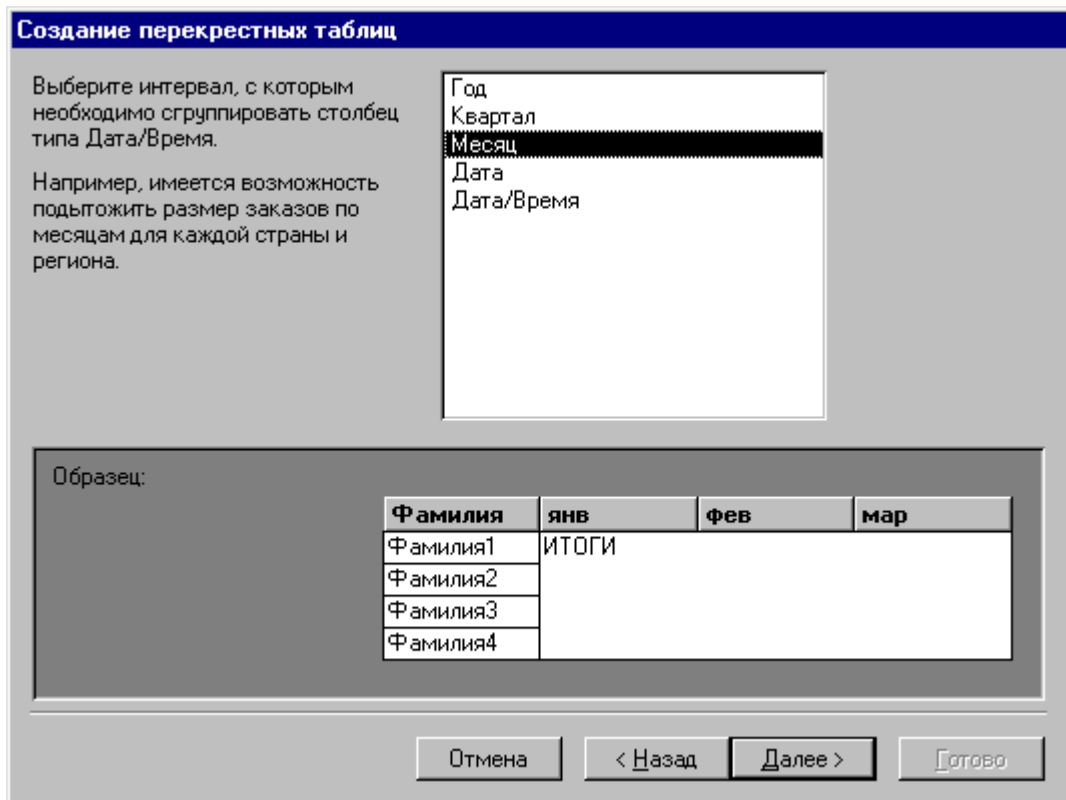


Рис.4.17. Четвертый шаг Мастера перекрестных запросов

Отриманий перехресний запит показаний на рис.4.20. Програма створила в новій таблиці поле результатів із досить довгим ім'ям, у якому підсумувала комірки таблиці по рядках. У нашому прикладі поле **Итоговое значение_Код организации** містить число нагадувань про того чи іншого суб'єкта у вихідній таблиці. Кожна графа з назвою місяця в заголовку містить число звертань до співробітника організації за цей місяць.

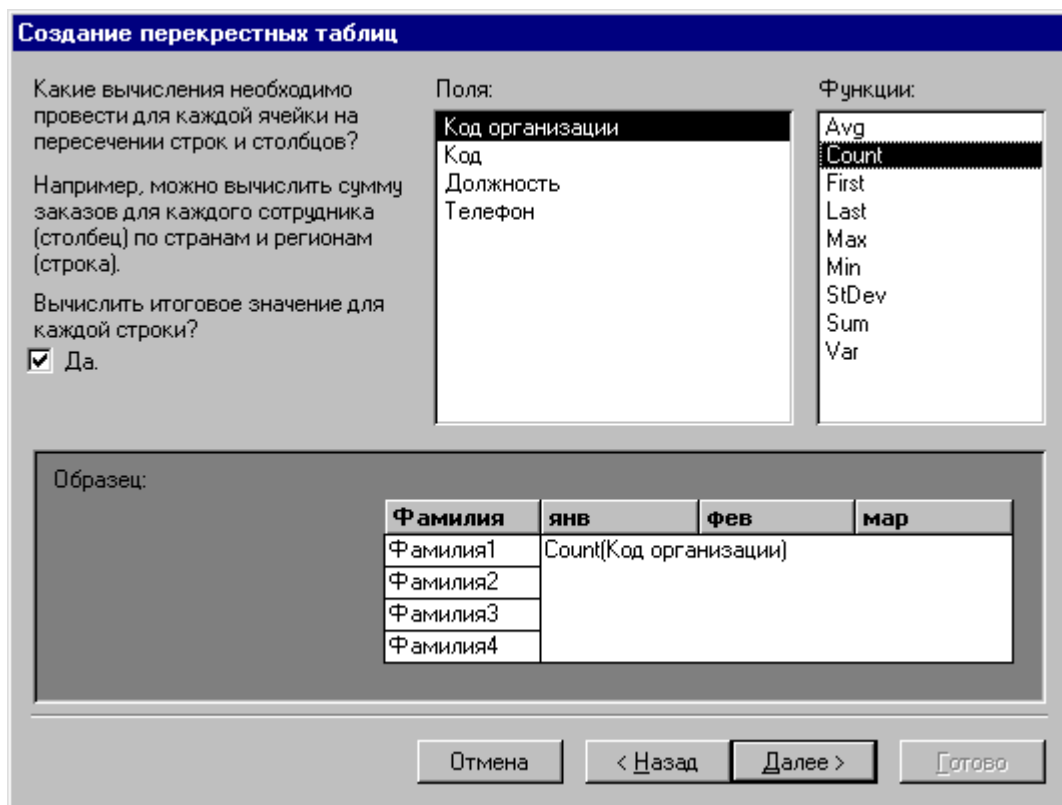


Рис.4.18. П'ятий крок Майстра перехресних запитів

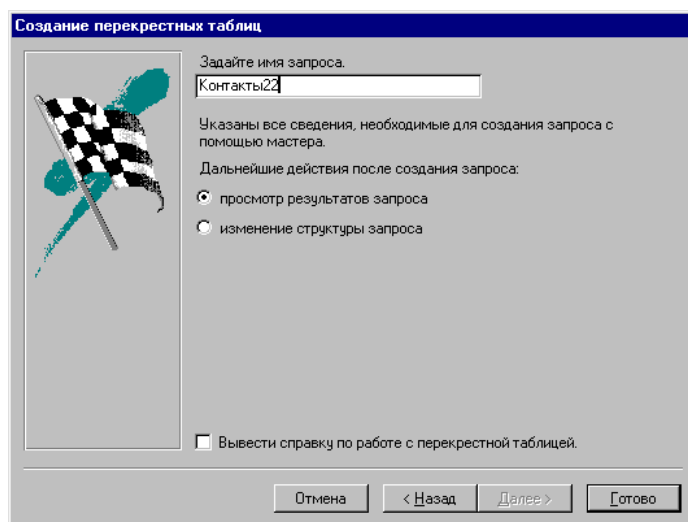


Рис.4.19. Шостий крок Майстра перехресних запитів

Фамилия	Итоговое значение	Код организации	январь	февраль	март	апрель
Васильев	1					1
Николаева	2				1	
Симонова	1					1
Сметанина	1			1		
Хлопов	1				1	
Чугунов	1			1		
Яковлев	3		2	1		

Рис.4.20. Приклад перехресного запиту

Вибрання повторюваних записів

Розглянемо, як приклад, таблицю Телефон1 з бази даних Test1. У полі Районы назва кожного району введена декілька разів. Іноді виникає необхідність згрупувати записи з однаковим значенням поля для того, щоб легше було вести їхнє подальше сортування. Найпростішим рішенням є сортування таблиці за відповідним полем. У Access передбачена також можливість створення спеціального запиту. Його використання нагадує операцію сортування, але надає користувачу додаткові можливості. Запит для пошуку повторюваних записів виконується за допомогою Майстра. Для його запуску потрібно натиснути кнопку **Создать** на вкладці **Запросы**, вибрати команду **Повторяющиеся записи** і клацнути кнопку **ОК**.

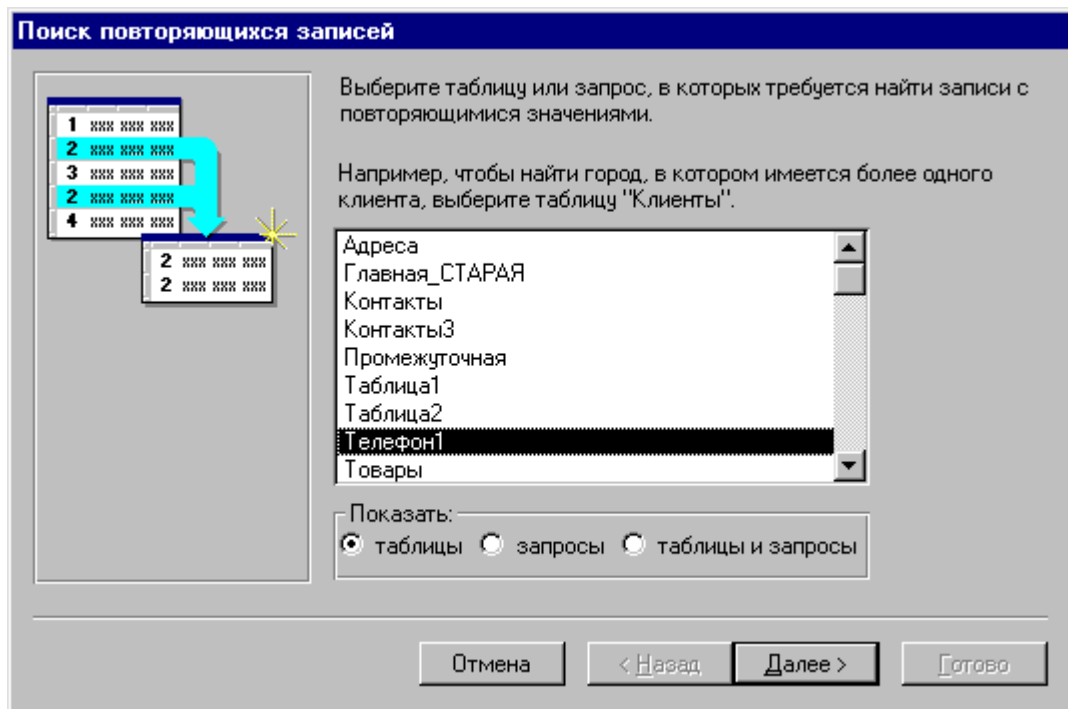


Рис.4.21. Перший крок Майстра повторюваних записів

Перший крок Майстра показаний на рис.4.21. Він дозволяє визначити джерело даних для запиту. У нашому прикладі - це таблиця Телефон1.

На другому кроці, показаному на рис.4.22, необхідно вибрати поле, у якому буде проводитися пошук повторюваних записів. Вкажемо поле Район.

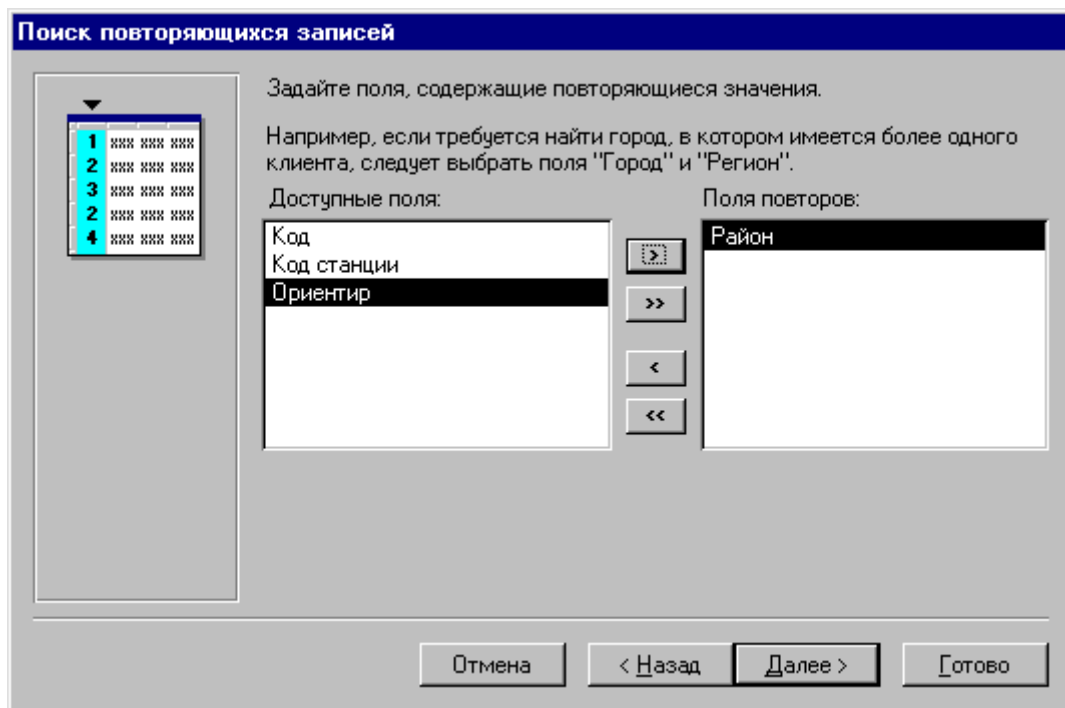


Рис.4.22. Другий крок Майстра повторюваних записів

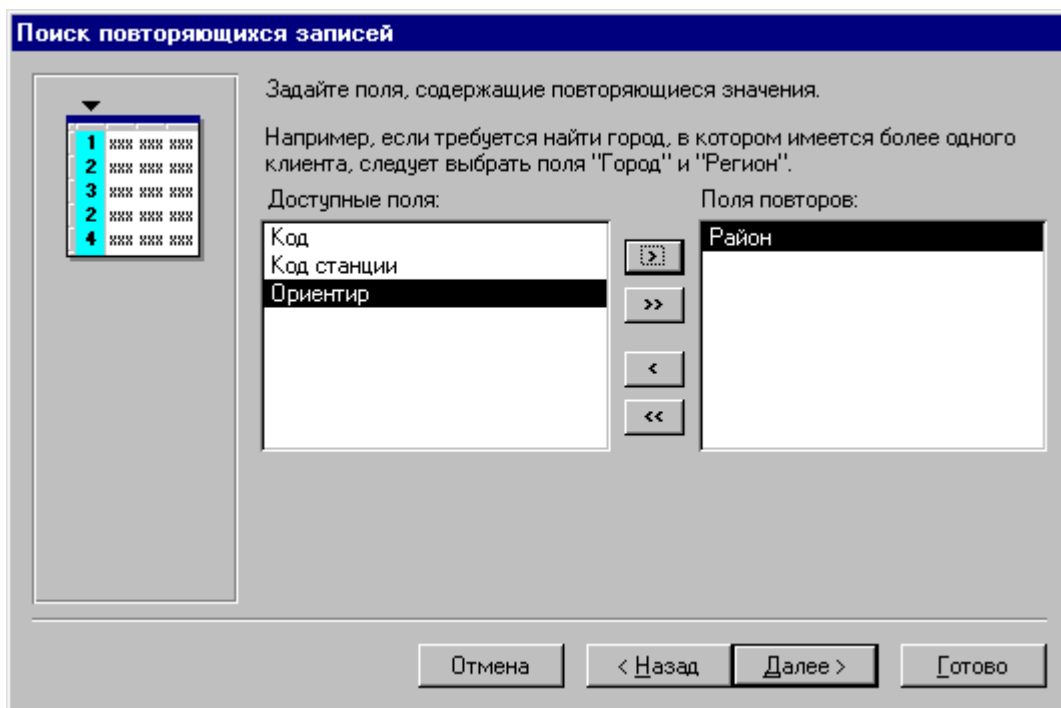


Рис.4.23. Третій крок Майстра повторюваних записів

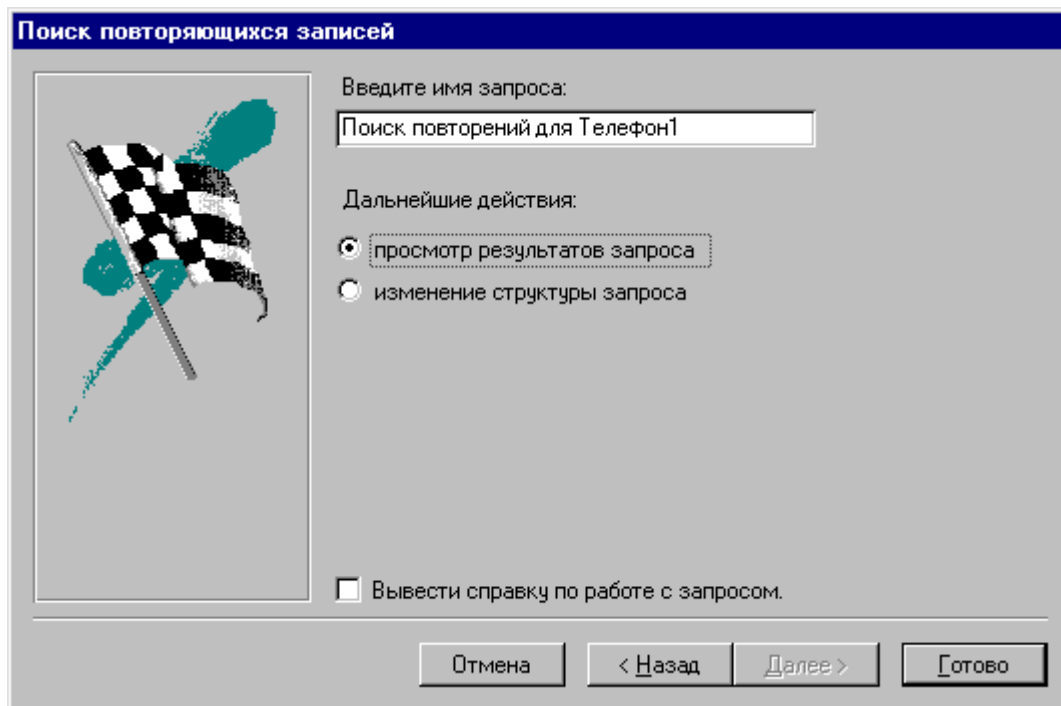


Рис.4.24. Четвертый шаг Мастера повторяющихся записей

Район	Код станции
Адмиралтейский	219
Адмиралтейский	259
Адмиралтейский	315
Адмиралтейский	314
Адмиралтейский	311
Адмиралтейский	214
Адмиралтейский	310
Адмиралтейский	112
Адмиралтейский	252
Адмиралтейский	251
Адмиралтейский	113
Адмиралтейский	114
Адмиралтейский	316
Василеостровский	356
Василеостровский	218
Василеостровский	217

Рис.4.25. Результат запиту на вибирання повторюваних записів

При необхідності можна вибрати і декілька полів. Наприклад, якщо вказати поля Район і Ориентир, то програма буде групувати записи, у яких зустрічаються однакові райони й однакові вулиці. Фактично, це еквівалентно сортуванню по декількох полях.

На третьому кроці Майстра, показаному на рис.4.23, необхідно вибрати поля, що повинні бути показані в таблиці запиту. Вкажемо поле Код станції.

На останньому, четвертому кроці Майстра, показаному на рис.4.24, можна задати ім'я нового запиту.

Фрагмент віртуальної таблиці, що є результатом запиту, показаний на рис 4.25. З рисунка видно, що назви районів у таблиці відсортовані, а коди станцій - ні. У даному випадку порядок слідування записів із тією самою назвою району може бути будь-яким.

Існує другий варіант використання Майстра повторюваних записів. Якщо на третьому кроці Майстра не вказувати жодного поля в списку **Дополнительное поле**, то програма автоматично створить у запиті поле **Повторы**, у якому вкаже, скільки разів зустрічається запис із тим або іншим значенням. В отриманому запиті вже не буде повторюваних записів - кожне значення поля повторів буде приведено в запиті один раз. Розмір таблиці-запиту буде менший, ніж розмір вихідної таблиці. Таким чином, ми маємо можливість створити *таблицю-зведення*. Потреба в таких таблицях дуже часто виникає в задачах користувачів. Запит у вигляді зведення, створений на основі таблиці **Телефон1**, показаний на рис.4.26.

Район поле	Повторы
Адмиралтейский	13
Василеостровский	8
Выборгский	21
г. Ломоносов	2
г. Пушкин	2
г. Сестрорецк	2
Калининский	15
Кировский	11
Колпинский	3
Красногвардейский	14
Красносельский	14
Кронштадский	3
Московский	13

Рис.4.26. Зведення, створене за допомогою Майстра повторюваних записів

Таке зведення має одну особливість: у ньому не вказуються записи, що не мають дублів за значенням заданого поля. Це значить, що в стовпці **Повторы** вам не зустрінеться одиниця. Якщо при редагуванні вихідної таблиці в ній стане менше повторюваних значень поля, то і таблиця-запит автоматично зменшиться. Якщо у вихідній таблиці зовсім не залишиться повторів, то і таблиця-запит стане порожньою. Корисно пам'ятати про цей ефект, коли потрібно відредагувати поле, для якого необхідно задати індексацію "без совпадений", а збіги є. Подібний запит наочно покаже, які значення зустрічаються в поле більше одного разу. Редагуйте поле, поки запит не стане порожнім, а потім сміливо індексуйте таблицю.

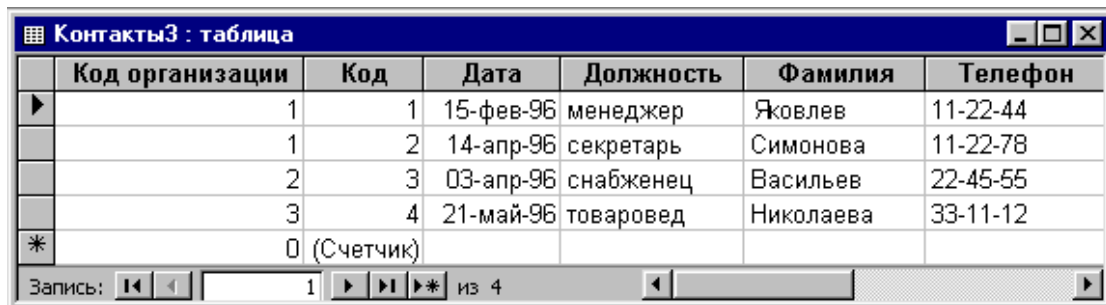
Вправа 4. Створіть запит по виборці повторень музичних стилів.

Редагування даних за допомогою запитів

Існують запити, які дозволяють автоматизувати зміну даних у таблицях. У великих базах даних не завжди зручно змінювати інформацію вручну. У деяких випадках необхідні спеціальні процедури для внесення змін за заданим алгоритмом. Розглянемо варіанти таких запитів.

Запит на додавання

Запит на додавання дозволяє скопіювати дані з однієї таблиці в іншу, не звертаючись за допомогою до буфера обміну. Записи можуть переміщатися між різними файлами баз даних і між базами даних різного типу. Для ілюстрації виконання запиту на додавання використаємо таблицю **Контакты** з бази даних **Test1**. Припустимо, у цю таблицю потрібно додати декілька нових записів, що знаходяться в таблиці **Контакты3** (див. рис.4.27).



	Код организации	Код	Дата	Должность	Фамилия	Телефон
▶	1	1	15-фев-96	менеджер	Яковлев	11-22-44
	1	2	14-апр-96	секретарь	Симонова	11-22-78
	2	3	03-апр-96	снабженец	Васильев	22-45-55
	3	4	21-май-96	товаровед	Николаева	33-11-12
*	0 (Счетчик)					

Рис.4.27. Дані для доповнення таблиці *Контакты*

Для виконання запиту на додавання потрібно виконати такі дії:

1. На вкладці **Запросы** натисніть кнопку **Создать** і виберіть режим **Конструктор**. Активізується вікно **Добавление таблицы**.
2. Виберіть із списку таблицю **Контакты3** і натисніть кнопку **Добавить**. Таблиця з'явиться у вікні запиту. Закрийте вікно. У такий спосіб ми визначили таблицю, що буде джерелом даних при виконанні запиту.
3. У режимі конструктора на панелі інструментів знаходиться список **Тип запроса**. Виберіть із нього елемент **Добавление**. Відкриється вікно **Добавление**, зображене на рис.4.28. У цьому вікні виберіть у списку ім'я таблиці, у яку потрібно додати дані - **Контакты**. Перемикач у вікні дозволяє вказати, у яку базу даних додається інформація: у поточну чи в якусь іншу. Після цього вікно можна закрити.

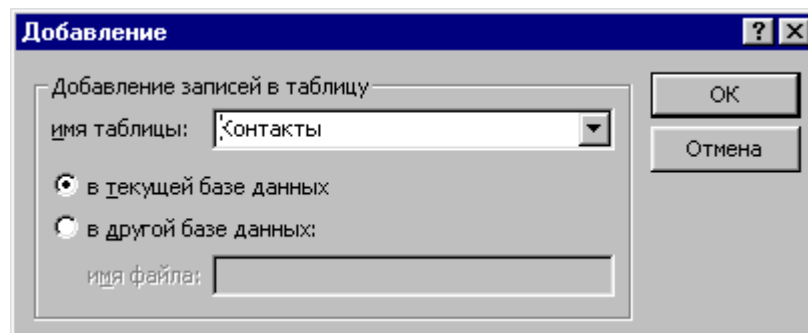


Рис.4.28. Вікно *Добавление*

4. Тепер необхідно визначити поля, які будуть добавлені в таблицю **Контакты**. Ідеальним варіантом є той, коли поля обох таблиць збігаються. Тоді, клацнувши два рази по символі "зірочка", ми вкажемо, що повинні бути скопійовані всі поля. Відповідна інформація з'явиться в бланку запиту, як показано на рис.4.29.
5. Залишилося натиснути кнопку **Запуск**, і можна буде переглянути результат. Він показаний на рис.4.30.

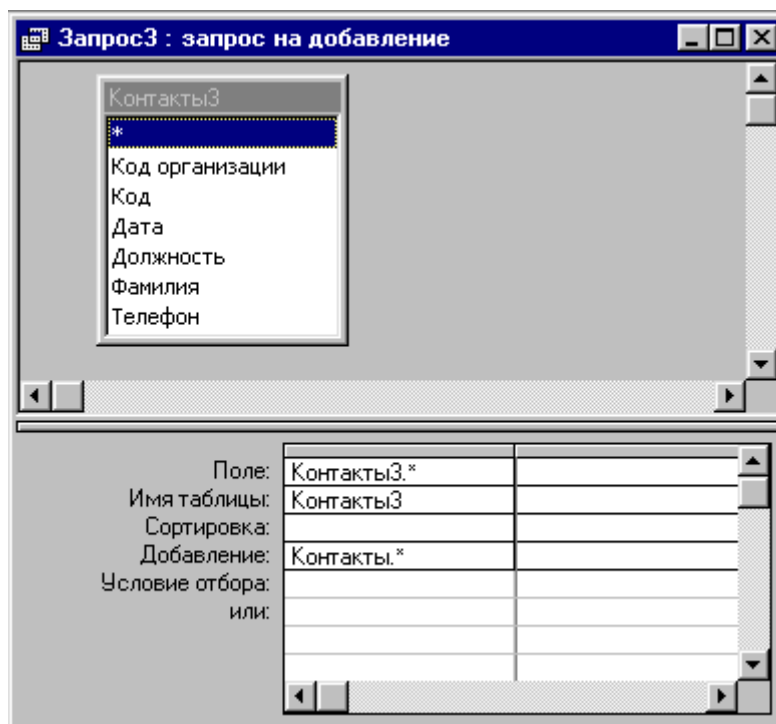


Рис.4.29. Вид запиту на добавляння

Код организации	Код	Дата	Должность	Фамилия	Телефон
	1	01-январь-96	менеджер	Яковлев	11-22-44
	1	15-февраль-96	менеджер	Яковлев	11-22-44
	1	15-февраль-96	менеджер	Яковлев	11-22-44
	1	02-январь-96	менеджер	Яковлев	11-22-44
	1	14-апрель-96	секретарь	Симонова	11-22-78
	1	14-апрель-96	секретарь	Симонова	11-22-78
	2	01-февраль-96	зам директора	Чугунов	22-45-45
	2	03-апрель-96	снабженец	Васильев	22-45-55
	2	03-апрель-96	снабженец	Васильев	22-45-55
	2	02-февраль-96	товаровед	Сметанина	22-45-66
	3	21-май-96	товаровед	Николаева	33-11-12
	3	21-май-96	товаровед	Николаева	33-11-12
	3	01-март-96	менеджер	Хлопов	33-11-12
	3	02-март-96	товаровед	Николаева	33-11-12
*	0	(Счетчик)			

Рис.4.30. Вид таблиці **Контакты** після виконання запиту

Декілька зауважень з приводу створення запитів на додавання:

Оскільки поле Код організації є індексованим, записи в таблиці-приймачі розташувалися у відсортованому порядку.

Якби нам потрібно було перемістити лише частину полів, то кожне поле тоді необхідно переносити в бланк запиту окремо, подвійним клацанням миші.

При бажанні можна додавати не всі записи з таблиці-джерела, а лише деякі, відповідно до заданої умови. У цьому випадку потрібно перенести кожне поле в бланк запиту окремо і заповнити рядок "Условие отбора". Наприклад, для відбору записів, що відносяться до періоду після 1 березня, у колонку для поля Дата цього рядка потрібно ввести такий вираз: >#1/03/96#.

Перед виконанням запиту можна побачити дані, які будуть вставлені в іншу таблицю. Для цього досить переключити режим перегляду запиту: вибрати зі списку Вид елемент Режим таблиці.

Виконання запиту на додавання не повинно порушувати цілісності даних, тому не будь-який запит може бути виконаний. Ускладнення можуть виникнути, якщо таблиця-приймач зв'язана з іншими таблицями і виконання запиту призводить до порушення унікальності ключа. Запит не буде виконаний і в тому випадку, коли порушується принцип унікальності записів для індексованого поля, у якому заборонені збіги.

Запит на додавання (якщо макет його збережений) можна виконувати необмежену кількість разів. Якщо ви оновите дані в таблиці-джерелі і знову активізуєте запит, то нові дані будуть добавлені в таблицю-приймач і т.д.

Ви можете відкрити запит у режимі конструктора і модифікувати його. Сам запит при цьому виконуватися не буде. Виконати його в режимі конструктора можна шляхом натискання кнопки Запуск. Якщо ж Ви виберете запит і натиснете кнопку Открыть на вкладці Запросы, це призведе до негайного запуску (виконання) запиту.

Access надає останній шанс скасувати запит перед його виконанням. На екран виводиться вікно діалогу, у якому повідомляється, скільки записів буде добавлено. Якщо користувач підтвердить виконання операції, скасувати її дію згодом буде неможливо.

Вправа 5. Створіть запит на додавання нових альбомів.

Запит на відновлення

Запит на відновлення забезпечує одну з найважливіших сервісних функцій СКБД - автоматизовану обробку даних. Такі запити дозволяють істотно заощадити час оператора, що виконує редагування інформації. Можливості алгоритмічного відновлення даних закладені в запити Access з самого початку, тому правило створення таких запитів гранично просте: спочатку створюється звичайний запит, а потім у його бланку заповнюється рядок "Обновление".

Код	Код товара	Название	Количество	Цена
1	T001	Товар 1	25	10700
2	T001	Телевизор	50	1500000
3	T002	Товар 2	0	9000
5	T001	Товар 1	120	10700
6	T003	Товар 3	300	4000
7	T003	Товар 3	250	3500
8	Б0Ф1	Бумага	10	5000
*	(Счетчик)		0	0

Рис.4.31. Таблица Товары

Для того, щоб внести зміни не в усі записи, а лише в ті, де це необхідно, у бланку запиту потрібно сформулювати умову відбору. На рис.4.31 показана таблиця з даними про товари. Якщо для якогось товару потрібно змінити ціну, створення запиту буде мати такий вигляд:

1. Створіть запит і додайте в нього таблицю **Товари**, як було зроблено з таблицею **Контакты3** у попередньому прикладі.
2. Перетягніть поля **Код товара** і **Цена** зі списку полів на бланк запиту.
3. Виберіть із списку **Тип запроса** елемент відновлення. У бланку з'явиться рядок "Обновление".
4. Введіть умову відбору для поля **Код товара**. Наприклад: **T001**.
5. Заповніть рядок "Обновление" для поля **Цена**. Припустимо, ціна повинна складати **1000**. У результаті запит набуде вигляду, показаного на рис.4.32.
6. Переглянути дані, що піддаються відновленню, можна, якщо переключити запит у режим таблиці (командою Вид ⇨ Режим таблиці). Кнопка **Запуск** дозволяє виконати відновлення.

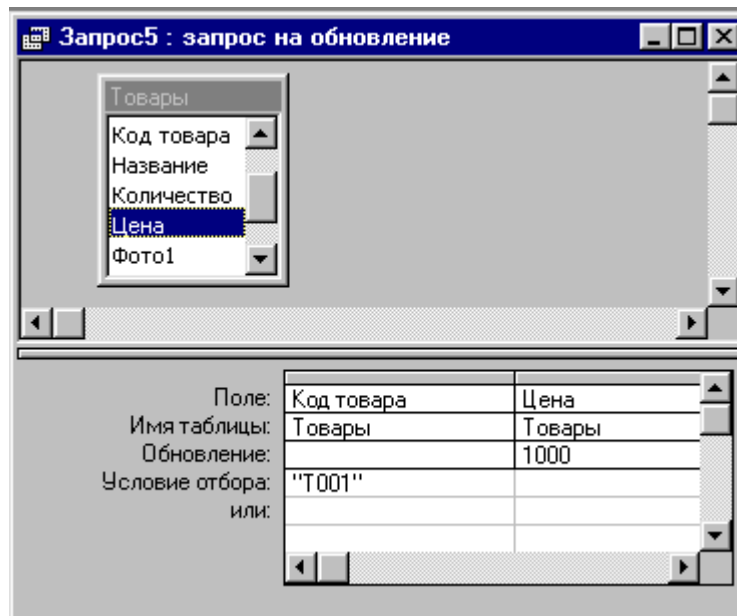


Рис.4.32. Запит на відновлення в режимі конструктора

У нашому прикладі ціни **8500** і **9500** у таблиці **Товари** після виконання запиту будуть замінені на **1000**.

Вправа 6. Створіть запит на відновлення таблиці "Виконавці".

Запит на створення таблиці

Запит може служити і засобом створення нової таблиці. У звичайному запиті користувач має можливість з'єднати поля і записи декількох таблиць, створивши тим самим об'єкт бази даних, що виконує функції таблиці. Всі дані, які використані в такому запиті, будуть розташовуватися у вихідних таблицях. Але таким самим способом можна створити і нову, реально існуючу таблицю з даними. Таким способом можна перетворити декілька зв'язаних таблиць в одну. Запит на створення таблиці виконується в декілька прийомів:

1. Створіть запит у режимі конструктора і включіть в нього всі необхідні таблиці.
2. Визначте зв'язки між таблицями. Програма намагається сама встановлювати зв'язки, якщо знаходить поля з однаковими іменами і типами. Якщо Access не може створити зв'язки автоматично, створіть їх вручну, перетягуючи мишею імена полів з одного списку в інший.
3. Відредагуйте бланк: створіть графі, що відповідають полям майбутньої таблиці, введіть (при необхідності) умови відбору і/або сортування.
4. Виберіть із списку Тип запроса елемент **Создание таблицы**.
5. Перегляньте отриману таблицю, переключившись у режим таблиці командою **Вид Режим таблиці**. Ви побачите таблицю в тому вигляді, якою вона буде створена після завершення операції. Якщо результат Вас влаштував, натисніть кнопку **Запуск**. Програма запросить ім'я нової таблиці і створить її.

Зауваження:

Розробка запиту на створення таблиці містить два етапи. Спочатку створюється звичайний запит будь-яким доступним і зручним засобом. На другому етапі з'являється нова таблиця, після того як звичайний запит перетвориться в запит на створення таблиці. Виконання другого етапу означає *копіювання* даних усередині однієї бази даних або в іншу базу даних.

Якщо в запит входять зв'язані таблиці і Ви перенесли всі їх поля в бланк за допомогою символу "зірочка" (тобто не вказуючи поля окремо), то програма автоматично залишить у новій таблиці тільки одне *поле* з двох, використаних для організації зв'язку. Якщо Ви переносите поля в бланк запиту окремо, то в новій таблиці з'являться *два* зв'язаних поля. Якщо вони мають однакові імена, то до них будуть додані імена таблиць-джерел.

Запит на створення таблиці можна використовувати неодноразово, причому двома способами. Якщо Ви спробуєте відкрити такий запит або натиснете кнопку **Запуск** у режимі конструктора, то запит буде запущений на виконання, і програма запропонує перезаписати отриману раніше таблицю, тобто створити її наново. Якщо ж ви виберете зі списку Тип запроса команду **Создание таблицы**, то Access запропонує Вам ввести нове ім'я для нової таблиці. Так можна створити необмежене число нових таблиць. Зрозуміло, дані в них будуть дублюватися.

У запит на створення таблиці можна включити обчислюване поле (див. нижче). У створеній таблиці воно буде перетворене в звичайне, але заповниться даними відповідно до заданого виразу.

Створення обчислюваних полів

У таблиці **Товари** (див. рис.4.31) є відомості про обсяг кожної партії товару (поле **Количество**) і ціни одиниці товару (поле **Цена**). Можна припустити, що в реальній задачі буде потрібно створити поле, у якому була б вказана вартість кожної партії товару. Для цього в таблиці потрібно створити *обчислюване поле*. Зробимо це, слідуючи покроковим інструкціям:

1. Спочатку необхідно перетворити таблицю в запит. Створіть новий запит і додайте в нього таблицю **Товари**.
2. Якщо Вам потрібно бачити всі поля таблиці **Товари**, клацніть два рази по символу "зірочка" у списку полів, щоб перенести всі поля в бланк. Або можна перенести лише частину полів.
3. У вільній графі бланку в рядку "Поле" введіть вираз, який визначить значення поля, що обчислюється:

[Товари! [Количество] * [Товари] ! [Цена]

Готовий запит показаний на рис.4.33.

4. Переключіть запит у режим таблиці і переконайтеся, що обчислюване поле створене (див. рис.4.34).
5. Закрийте запит. Програма запитає, під яким ім'ям його зберегти. Придумайте нове ім'я.

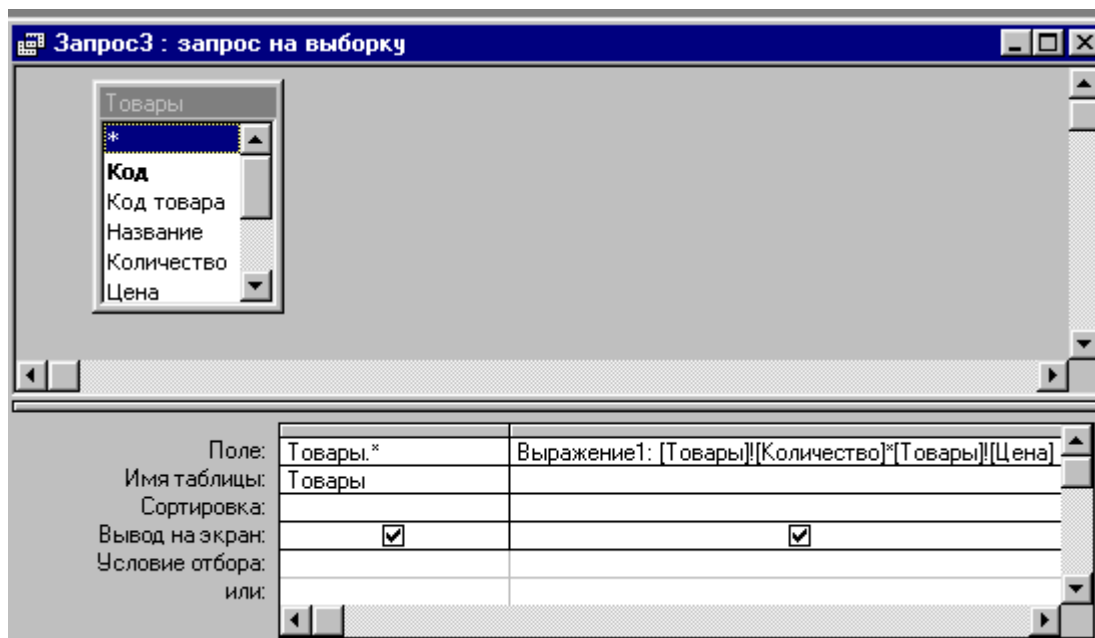


Рис.4.33. Створення обчислюваного поля в запиті

Ще декілька зауважень:

Ім'я обчислюваному полю програма призначила сама. Якщо вам не подобається заголовок колонки "Выражение1", то у Вас є два способи змінити його. Можна задати інше ім'я прямо в бланку запиту або вибрати з контекстного меню цього поля команду **Свойства** і визначити підпис.

Для створення виразу можна було використовувати "построитель выразений". Для цього потрібно вибрати команду **Построить** з контекстного меню. Для нашого прикладу дії у вікні **Построителя**:

1. Знайти в лівій нижній області вікна папку таблиці **Товары** і клацнути на ній два рази.
2. У середній нижній області вікна клацнути два рази на імені поля **Количество**.
3. Натиснути кнопку з зображенням знака множення.
4. У середній нижній області вікна клацнути два рази на імені поля **Цена**.
5. Натиснути кнопку **ОК**.

Вправа 7. Створіть запит по виборці конкретних альбомів з обчислюваним полем "Термін звучання альбома" (термін звучання запису * кількість записів).

Код	Код товара	Название	Количество	Цена	Выражение1
1	T001	Товар 1	25	10700	267500
2	T001	Телевизор	50	1500000	75000000
3	T002	Товар 2	0	9000	0
5	T001	Товар 1	120	10700	1284000
6	T003	Товар 3	300	4000	1200000
7	T003	Товар 3	250	3500	875000
8	Б0Ф1	Бумага	10	5000	50000
*	(Счетчик)		0	0	

Рис.4.34. Запит із обчислюваним полем, відкритий у режимі таблиці

Фільтри

Фільтр можна собі уявити як запит, але в спрощеному варіанті. Якщо запит є одним із найважливіших об'єктів (тобто складових частин) бази даних, то фільтр - це скоріше процедура пошуку і сортування записів. Проте, у кінцевому рахунку, і фільтри, і запити виконують однакову функцію: відбір даних. Існує декілька різновидів фільтрів:

1. Фільтр за виділенням;
2. Звичайний фільтр (або поле "Фільтр для");
3. Розширений фільтр.

Контекстні меню, зв'язані з заголовком вікна і з комірками таблиці, мають декілька команд для виконання фільтрації даних у таблиці.

Фільтр за виділенням - це використовуваний у Access спосіб швидкого відбору записів за виділенням зразком; він забезпечує запуск найпростішого фільтра, що тільки можна уявити. Критерієм фільтрації служить одна або декілька виділених комірок у таблиці чи запиті. Або просто комірка, у якій міститься курсор. Уявимо собі, наприклад, що в одному з полів знаходяться числа. Якщо Ви виділите комірку з числом 33 і виберете вказану команду, то в таблиці залишаться видимими лише ті записи, в яких у даному полі стоїть 33. Тобто Вам не потрібно починати ніяких дій по *створенню фільтра*. Фільтрацію можна здійснювати і за блоком виділених комірок. При цьому він не обов'язково повинний знаходитися в межах одного поля, але може містити декілька суміжних полів - у цьому випадку фільтрація буде проводитися у відповідності зі значеннями цих полів. Необхідно лише, щоб типи даних у полях збігалися. Команда **Исключить выделенное** також забезпечує фільтрацію записів, але в даному випадку критерій фільтрації зворотний - приховуються записи, значення полів яких збігаються зі значеннями виділених полів (комірок). Команда **Удалить фильтр** не видаляє сам використаний критерій із пам'яті, як можна подумати, а лише відмінняє дію фільтра, тобто всі записи стають доступними. Включити режим фільтрації знову можна командою **Применение фильтра**. На панелі інструментів за замовчуванням й у відповідних режимах виводиться кнопка, що змінює своє ім'я в залежності від ситуації: "Применение фильтра" або "Удалить фильтр". Команда **Изменить фильтр** дозволяє відредагувати умову відбору записів у вікні "обычного фильтра". За цією командою відкривається вікно, у якому критерій фільтрації по кожному полю поданий у вигляді списку. Список можна розкрити і змінити набір значень полів, або створити додаткові списки для тих полів, що спочатку не використовувалися для задання

критерію фільтрації. Приклад такого вікна показаний на рис.4.35. Фільтр задається для запиту Товари2, розглянутого вище. Вираз, уведений для поля **Название**, дозволяє відібрати всі рядки таблиці, за винятком записів про товар **Товар1**. Критерій фільтрації можна визначити для будь-якого поля, використовуючи відповідний список.

Наприклад, список для поля **Цена** містить значення всіх цін, що зустрічаються в запиті. Вкладки **Или** дозволяють створити додаткові критерії. Об'єднання за **ИЛИ** в даному випадку означає, що при фільтрації будуть показані спільно всі записи, які були відібрані за критеріями на кожній із вкладок.

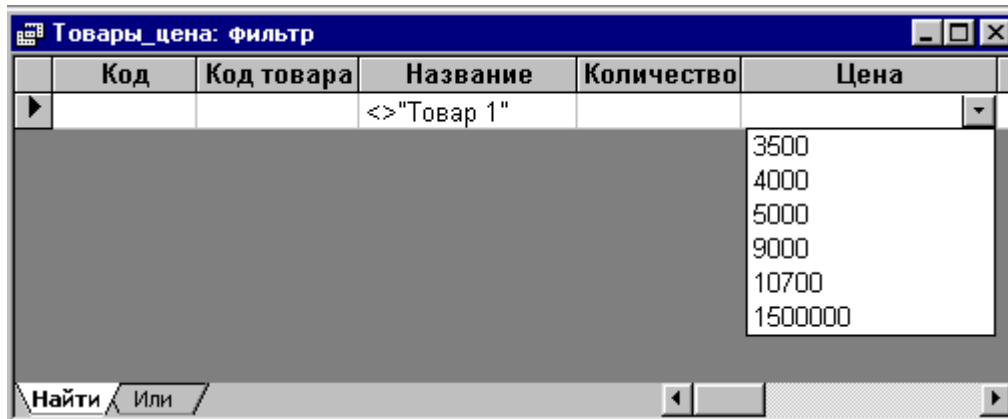


Рис.4.35. Редагування звичайного фільтра

У деяких випадках неможливо реалізувати переваги фільтра за виділенням або звичайного фільтра. Припустимо, нам потрібно виконати фільтрацію в таблиці **Телефон1** по полю **Код станции**. У цьому полі зібрані коди - тризначні числа. Їхні значення не повторюються. Сама ж таблиця містить біля двохсот записів. Зрозуміло, можна використовувати будь-який із фільтрів. Але фільтр за *виділенням* нам не допоможе: якщо значення коду можна виділити мишею, то шукати його вже не потрібно. Перевага *звичайного* фільтра полягає в можливості використовувати список значень поля. Але для поля **Код станции** список вийде таким самим довгим, як і таблиця: повторюваних значень поля немає. У подібній ситуації доцільніше усього використати поле **"Фільтр для"**, тому що це не потребує складних маніпуляцій із командами і вікнами програми. Достатньо встановити курсор на потрібне поле і натиснути праву кнопку миші. Контекстне меню для цього випадку показано на рис.4.36. У поле **Фільтр для** треба ввести потрібне значення і натиснути клавішу ENTER. Після цього фільтр відразу ж приводиться в дію. Скасувати фільтрацію можна, як і в попередніх випадках, командою **Удалить фильтр**.

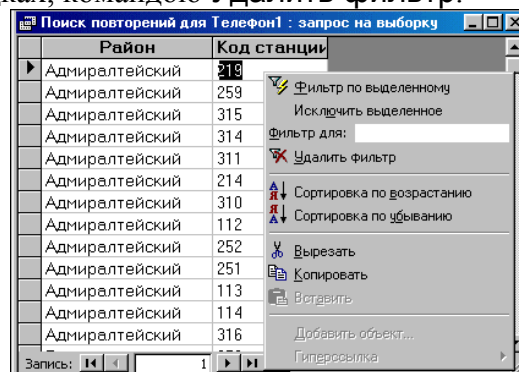


Рис. 4.36. Контекстне меню з вікном "Фільтр для"

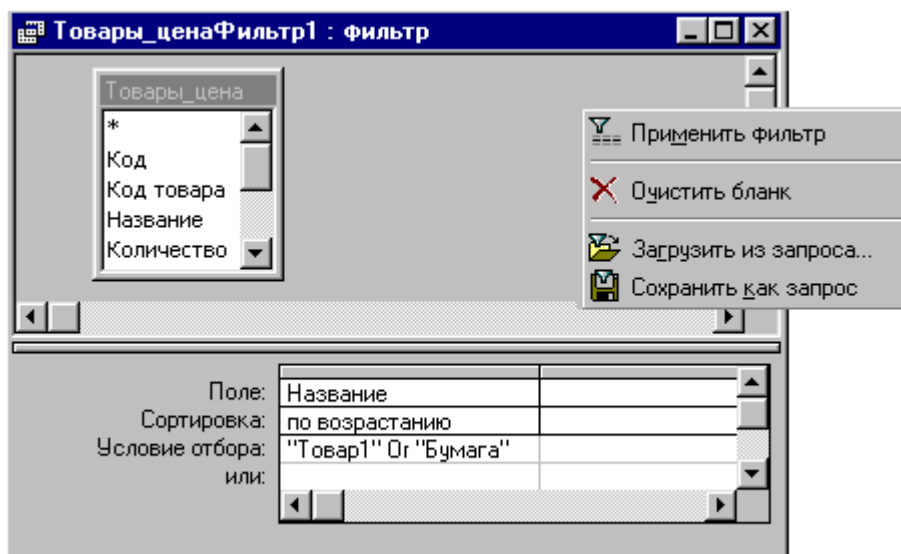


Рис. 4.37. Вікно розширеного фільтра з розкритим контекстним меню

Розширений фільтр запускається командою **Записи** **Фільтр** **Расширенный фильтр**. Його вікно показане на рис.4.37. У даному випадку для фільтрації був вибраний запит **Товары2**. Ми бачимо, що це вікно дуже схоже на вікно запиту в режимі конструктора. Дійсно, у запиті і розширеного фільтра дуже багато спільного. Умова відбору в цьому фільтрі задається точно так само, як і в запиті: за допомогою бланка. Режим сортування також може бути заданий у бланку: шляхом вибору зі списку.

У чому ж різниця між запитом і розширеним фільтром? Запит - це об'єкт бази даних, що створюється з використанням таблиць і інших запитів. Він може служити, у свою чергу, основою для створення форм і звітів. Нарешті, запит формулюється за допомогою мови програмування (SQL), а це передбачає високий рівень стандартизації і класичні підходи до використання додатка. Розширений фільтр являє собою більш "просту" конструкцію. У бланку фільтра відсутній рядок **"Вывод на экран"**, що присутній у бланку запиту. Фільтр - це властивість об'єкта бази даних, що може зберігатися в пам'яті комп'ютера.

Контекстне меню розширеного фільтра (див. рис.4.37) містить команди, що дозволяють перетворювати розширені фільтри в запити і навпаки.