

Розділ 9°

Основи мови запитів

Повторення

- ◆ Що таке мова маніпулювання даними?
- ◆ Яка мова маніпулювання даними найбільш поширена в реляційній моделі?
- ◆ Який вигляд має вікно конструктора запитів?
- ◆ Що таке віртуальна таблиця?

Як ми вже зазначали, практично в усіх реляційних СКБД для маніпулювання даними застосовують мову SQL (англ. Structured Query Language — мова структурованих запитів), а такі автоматизовані засоби створення запитів, як ми розглядали в попередньому розділі, представлені лише в поодиноких системах керування базами даних. Тобто основний спосіб конструювання запитів до баз даних — це їх запис у текстовому вигляді, чимось подібний до написання невеличких програм. Однак мова SQL не є мовою програмування, вона декларативна, тобто дозволяє користувачу описати, **що** він хоче отримати, не описуючи, **як** саме комп'ютер має обчислити потрібний результат. Тому писати запити мовою SQL значно легше, ніж програми будь-якою мовою програмування. Загалом є кілька різновидів запитів: на вибирання даних, їх додавання, видалення, оновлення та деякі інші. Сьогодні ви навчитеся описувати мовою SQL нескладні запити на вибирання даних.

Загальна структура SQL-запиту

Ви вже вмієте створювати *запити на вибирання даних*, або *вибіркові запити* за допомогою конструктора запитів MS Access.

Відмінною рисою цього типу запитів є те, що вони не змінюють дані в базі, а лише вибирають їх з таблиць за певними умовами. Результатом виконання будь-якого вибіркового запиту, нагадаємо, є віртуальна таблиця, що існує нетривалий час, поки ви її не закриєте.

Зрозуміти структуру вибіркового SQL-запиту найлегше на конкретному прикладі, розглянутому у вправі 9.1.

Вправа 9.1

Запишіть мовою SQL і виконайте запит *визначити прізвища та імена учнів-хлопців*.

1. Відкрийте конструктор запитів. У вікні **Добавление таблицы** виберіть таблицю **Учні**, оскільки запит стосується учнів, після чого клацніть кнопку **Добавить** і закрийте це вікно.
2. У Access 2003 клацніть правою кнопкою миші будь-яке місце вікна конструктора, крім бланка запиту, й виберіть з контекстного меню команду **Режим SQL**, а в Access 2007/2010 з меню кнопки **Режим** на стрічці **Конструктор** виберіть команду **Режим SQL**. Буде відображено заготовку SQL-коду запиту:

```
SELECT  
FROM Учні;
```

У цій заготовці ви бачите два ключових слова: **SELECT** і **FROM**. Після слова **FROM** вказують назви таблиць, з яких вибиратимуться дані. У нашому випадку це буде таблиця **Учні**; її назву вже введено автоматично.


3. Після слова **SELECT** через кому запишіть назви полів, значення яких відобразатимуться. У даному запиті це поля **прізвище** та **ім'я**. Оскільки назва поля **ім'я** містить символ апострофа, її потрібно взяти у квадратні дужки:

```
SELECT прізвище, [ім'я]  
FROM Учні;
```

Зазначимо, що у квадратні дужки беруться також назви, які містять пробіл.

4. В кінці запиту (але до символу ;) запишіть ключове слово WHERE, а після нього — умову відбору записів (серед усіх учнів нам потрібно відібрати хлопців). Остаточний вигляд запиту буде таким:

```
SELECT прізвище, [ім'я]
FROM Учні
WHERE стать="ч";
```

5. Запустіть запит на виконання кнопкою  (Виконати) та переконайтеся, що в таблиці результатів є відомості тільки про хлопців.
6. Закрийте вікно запиту, зберігши його під іменем учні-хлопці.

Загалом вибірковий SQL-запит має такий формат:

```
SELECT список полів, значення яких потрібно отримати
FROM список таблиць, з яких вибираються дані
WHERE умова, яку мають задовольняти записи,
      що вибираються
```

Зазначимо, що вирази, які починаються з ключових слів, часто називають *фразами*, наприклад фраза SELECT. Регістр ключових слів неважливий, але їх прийнято записувати великими літерами.

Зауважимо також, що мова SQL *англійована*, тобто її вирази дещо нагадують англійські речення. Слово SELECT перекладається як «вибрати», слово FROM — «з», а WHERE — «де», або «для яких». У цілому простий SQL-запит потрібно читати за такою схемою.

Запит SELECT x FROM y WHERE z слід інтерпретувати так: «вибрати поля x тих записів таблиці y, які задовольняють умову z».

Для допитливих. Якщо запит із вправи 9.1 скласти в конструкторі, а потім перейти в режим SQL, то ми побачимо такий текст, автоматично сконструйований СКБД:

```
SELECT Учні.прізвище, Учні.[ім'я]
FROM Учні
WHERE (((Учні.стать)="ч"));
```

За змістом він нічим не відрізняється від тексту, записаного нами в SQL-редакторі, але виглядає складнішим через те, що під час автоматичної побудови SQL-виразів MS Access вводить деякі зайві позначення. Так, після слова WHERE поставлено непотрібні дужки, а перед назвами атрибутів записано назву таблиці з символом крапки. Вираз Учні.прізвище можна читати як «поле прізвище таблиці Учні». Проте якщо у запиті всього одна таблиця, то поле прізвище ніякій іншій таблиці не може належати і тому специфікатор Учні. можна опустити.

Завдання 9.1

Запишіть мовою SQL запит *визначити прізвища та номери паспортів учителів-математиків*. Як перейти в режим введення SQL-коду, описано у вправі 9.1.

З'єднання таблиць

Розглянемо запит *визначити прізвища та імена вчителів, що викладають в 11А класі*, який ми вже реалізували за допомогою конструктора запитів (див. рис. 8.7). Якщо створити цей запит у конструкторі, а потім відобразити його SQL-код та видалити зайві дужки і специфікатори, результат буде таким:

```
SELECT прізвище, [ім'я]
FROM Учителі INNER JOIN Викладання ON
      Учителі.паспорт = Викладання.учитель
WHERE клас="11А"
```

З'єднання таблиць

У фразі FROM ми бачимо не назву таблиці і не список назв, а оператор INNER JOIN (у перекладі з англійської — внутрішнє з'єднання), призначений для з'єднання таблиць. Загальний синтаксис цього оператора такий:

Таблиця1 INNER JOIN Таблиця2 ON умова

Умова зазвичай стосується обох таблиць, вказаних до і після слів INNER JOIN (у прикладі вище — таблиць Учителі та Викладання). Оператор «зчіплює» ті записи таблиці 1 і таблиці 2, які, разом узяті, відповідають умові. У результаті з таких зчіпок формується нова таблиця, до якої вже застосовується відбір рядків згідно з фразою WHERE, а потім — відбір стовпців згідно з фразою SELECT.

Опишемо детальніше алгоритм, за яким СКБД виконує запит.

1. Виконується оператор INNER JOIN у фразі FROM. А саме, зчіплюються всі такі пари записів з таблиць Учителі та Викладання, для яких виконується рівність `Учителі.паспорт = Викладання.учитель`, тобто значення поля `паспорт` у таблиці Учителі збігається зі значенням поля `учитель` у таблиці Викладання (рис. 9.1, а). У результаті отримуємо таблицю, що містить усі такі зчіпки (рис. 9.1, б).

Оскільки у зв'язку між таблицями Учителі та Викладання таблиця Учителі є головною, а Викладання — підлеглою, то оператор Учителі INNER JOIN Викладання можна інтерпретувати так: «до кожного вчителя дописати інформацію про його викладання».

2. З таблиці, зображеної на рис. 9.1, б відбираються записи за умовою `клас="11А"`, тобто ті записи, які в полі `клас` містять значення 11А. У результаті отримуємо таблицю, зображену на рис. 9.2.
3. У таблиці, зображеній на рис. 9.2, залишаються тільки два поля, вказані після слова SELECT: `прізвище` та `ім'я`. Отримуємо таблицю, зображену на рис. 9.3, — це і є остаточний результат виконання запиту.

клас	учитель
11Б	СН 410268
11А	СО 211517
11Б	СО 211517
10А	СР 652320
11А	СР 652320
10Б	КН 200125
11Б	КН 200125
10А	СО 927453
11А	СО 927453
11Б	СО 927453

паспорт	прізвище	ім'я	по батькові	стать	спеціальність
СН 410268	Михайлюк	Дмитро	Семенович	ч	математик
СО 211517	Корбут	Василь	Петрович	ч	математик
СР 652320	Сошко	Катерина	Миколаївна	ж	біолог
КН 200125	Томчишин	Віктор	Георгійович	ч	історик
СО 927453	Петрова	Ніна	Володимирівна	ж	фізик

а

паспорт	прізвище	ім'я	по батькові	стать	спеціальність	учитель	клас
СН 410268	Михайлюк	Дмитро	Семенович	ч	математик	СН 410268	11Б
СО 211517	Корбут	Василь	Петрович	ч	математик	СО 211517	11А
СО 211517	Корбут	Василь	Петрович	ч	математик	СО 211517	11Б
СР 652320	Сошко	Катерина	Миколаївна	ж	біолог	СР 652320	10А
СР 652320	Сошко	Катерина	Миколаївна	ж	біолог	СР 652320	11А
КН 200125	Томчишин	Віктор	Георгійович	ч	історик	КН 200125	10Б
КН 200125	Томчишин	Віктор	Георгійович	ч	історик	КН 200125	11Б
СО 927453	Петрова	Ніна	Володимирівна	ж	фізик	СО 927453	10А
СО 927453	Петрова	Ніна	Володимирівна	ж	фізик	СО 927453	11А
СО 927453	Петрова	Ніна	Володимирівна	ж	фізик	СО 927453	11Б

б

Рис. 9.1. З'єднання таблиць Учителі та Викладання: а — зчеплення записів; б — результат з'єднання

паспорт	прізвище	ім'я	по батькові	стать	спеціальність	учитель	клас
СО 211517	Корбут	Василь	Петрович	ч	математик	СО 211517	11А
СР 652320	Сошко	Катерина	Миколаївна	ж	біолог	СР 652320	11А
СО 927453	Петрова	Ніна	Володимирівна	ж	фізик	СО 927453	11А

Рис. 9.2. Відбір записів зі з'єднаної таблиці

прізвище	ім'я
Корбут	Василь
Сошко	Катерина
Петрова	Ніна

Рис. 9.3. Проекція таблиці на поля, вказані після слова SELECT

У загальному випадку SQL-запит виконується так:

1. Виконується операція з'єднання таблиць із фрази FROM.
2. До отриманої таблиці застосовується операція вибірки: відбираються ті записи, що задовольняють умову, вказану у фразі WHERE.
3. Таблиця, отримана на попередньому кроці, проектується на поля, вказані у фразі SELECT, тобто ці поля залишаються, а всі інші відкидаються.

Більшість вибірових запитів до реляційних баз даних (але не всі) виконуються саме за цією схемою: з'єднання таблиць — вибірка записів — проекція на поля.

Для допитливих: З'єднання, вибірка і проекція — три з восьми операцій реляційної алгебри Кодда. Як уже згадувалося в розділі 1, реляційна алгебра — це мова маніпулювання даними, запропонована автором реляційної моделі Е. Коддом. Багато її рис увібрала в себе мова SQL.

Завдання 9.2

Запишіть оператори INNER JOIN, які дозволять отримати в базі даних школа такі з'єднання таблиць, як показано на рис. 9.4, а-б.

назва	класний керівник	паспорт	прізвище	ім'я	по батькові	стать	спеціальність
10А	СН 410268	СН 410268	Михайлюк	Дмитро	Семенович	ч	математик
11Б	СО 211517	СО 211517	Корбут	Василь	Петрович	ч	математик
10Б	КН 200125	КН 200125	Томчишин	Віктор	Георгійович	ч	історик

а

код	прізвище	ім'я	дата народж	стать	Учні.клас	назва	класний керівник	учитель	Вкля
1	Шлак	Максим	13.05.1998	ч	10А	10А	СР 652320	СР 652320	10А
1	Шлак	Максим	13.05.1998	ч	10А	10А	СР 652320	СО 92745	10А
2	Хоменко	Ірина	09.09.1995	ж	11А	11А	СО 927453	СР 652320	11А
2	Хоменко	Ірина	09.09.1995	ж	11А	11А	СО 927453	СО 211517	11А
2	Хоменко	Ірина	09.09.1995	ж	11А	11А	СО 927453	КН 200125	11А
3	Недригайло	Петро	01.07.1995	ч	11Б	11Б	СО 211517	СН 410268	11Б
4	Буджак	Оксана	14.01.1996	ж	11Б	11Б	СО 211517	СН 410268	11Б
3	Недригайло	Петро	01.07.1995	ч	11Б	11Б	СО 211517	СО 211517	11Б
4	Буджак	Оксана	14.01.1996	ж	11Б	11Б	СО 211517	СО 211517	11Б
3	Недригайло	Петро	01.07.1995	ч	11Б	11Б	СО 211517	СО 92745	11Б
4	Буджак	Оксана	14.01.1996	ж	11Б	11Б	СО 211517	СО 92745	11Б

б

Рис. 9.4. З'єднання таблиць

Завдання 9.3

Реалізуйте мовою SQL запит визначити, у якому класі класним керівником є Сошко Катерина Миколаївна.

Для допитливих. З'єднувати таблиці можна і без оператора INNER JOIN. Для цього слід перелічити їхні назви через кому у фразі FROM, а у фразі WHERE записати умову з'єднання записів, долучивши її до інших умов сполучником AND. Наприклад, запит визначити прізвища та імена вчителів, що викладають в 11А класі можна реалізувати так:

```
SELECT прізвище, [ім'я]
FROM Учителі, Викладання
WHERE клас="11А" AND Учителі.паспорт =
      Викладання.учитель
```

Підзапити

У попередньому розділі ми згадували про властивість реляційної замкненості мови SQL. Вона полягає в тому, що результатом будь-якого запиту є таблиця, а отже, одні запити можна підставляти в інші замість таблиць. Загалом є два способи підстановки одного запиту в інший:

- ◆ замість імені таблиці в запиті вказують ім'я іншого запиту;
- ◆ в одному запиті записують повний текст іншого запиту, взятий у круглі дужки.

Підстановку можна здійснювати у фразах FROM та WHERE. Особливо важливий і цікавий той випадок, коли у фразі WHERE одного запиту записують повний текст іншого запиту. Розглянемо цей випадок детально, реалізуючи запит *визначити прізвища та імена вчителів, що викладають в 11А класі*, з використанням підзапиту. SQL-текст запиту буде таким:

```
SELECT прізвище, [ім'я]
FROM Учителі
WHERE паспорт IN (SELECT учитель FROM Викладання
                  WHERE клас="11А")
```

У фразі WHERE цього запиту вжито оператор IN (англ. «в», «належить»). Його загальний формат такий:

значення IN (підзапит)

Результатом підзапиту має бути таблиця з одним стовпцем. Якщо вказане перед словом IN значення в цьому стовпці є, то весь вираз значення IN (підзапит) вважається істинним, інакше — хибним. Щоб краще зрозуміти принцип дії оператора IN, розглянемо крок за кроком виконання запиту *визначити прізвища та імена вчителів, що викладають в 11А класі* з точки зору СКБД.

1. Виконуємо фразу FROM. Оскільки в ній вказано лише одну таблицю Учителі, то на першому кроці ми просто «беремо» цю таблицю для подальших операцій.

2. Перебираємо всі записи таблиці Учителі, перевіряючи для кожного, чи виконується умова, вказана після слова WHERE. Ця перевірка здійснюється в два етапи.

- 2а. Виконуємо підзапит. Його результат — це набір номерів паспортів учителів 11А класу (згадайте структуру таблиці Викладання).

- 2б. Перевіряємо, чи належить значення поля паспорт з поточного запису таблиці Учителі набору значень, отриманому на кроці 2а. Якщо належить, то запис у таблиці Учителі залишаємо, інакше — відкидаємо.

3. Проектуємо отриману на кроці 2 таблицю за полями прізвище та ім'я, тобто ці поля залишаємо, а всі інші — відкидаємо.

Таким чином, оператор IN дає змогу перевірити, чи належить значення множині результатів підзапиту. Зазначимо, що у фразу WHERE підзапит можна вставляти лише за допомогою логічного оператора, формуючи вираз, значення якого істинне або хибне. Однак це не обов'язково має бути оператор IN; є ще оператори EXISTS (англ. «існує»), ANY (англ. «будь-який»), ALL (усі). Так, значення виразу EXISTS (підзапит) буде істинним, якщо результат підзапиту містить хоча б один запис, і хибним, якщо результат підзапиту порожній.

Завдання 9.4

Використовуючи підзапит в операторі IN, реалізуйте мовою SQL запит *визначити класи, у яких викладають вчителі на ім'я Петро*.

Віднімання множин записів

Можливості мови SQL значно ширші за можливості конструктора запитів MS Access і тому далеко не кожен SQL-запит може бути створений у вікні конструктора. Зараз ми розглянемо найпростіший різновид запитів, які не можуть бути створені за допомогою конструктора. Це запити з запереченням, наприклад *визначити прізвища та імена вчителів, які не викладають в 11А класі*. Цей запит відрізняється від того, який ми

розглядали раніше, наявністю частки «не». Якби її не було, ми би просто з'єднали таблицю вчителів із таблицею викладання, відібрали ті записи, які відповідають 10А класу, та спроектували результат на поля прізвище та ім'я. Але в запиті із часткою «не» нам потрібні не ці, а якраз всі інші вчителі. Тобто від множини всіх учителів нам потрібно відняти тих, які викладають у 10А класі. Найлегше це зробити за допомогою оператора NOT IN (англ. «не належить»):

```
SELECT прізвище, [ім'я]
FROM Учителі
WHERE паспорт NOT IN (SELECT учитель FROM Викладання
                       WHERE клас="11А")
```

Буквально запит читається так: «вибрати прізвища та імена тих учителів, які не належать множині вчителів, що викладають в 11А класі».

Завдання 9.5

Реалізуйте мовою SQL запит *визначити назви класів, у яких не вчиться жодної дівчини*.

Висновки

- ◆ Вибірковий запит у мові SQL має такий формат:

```
SELECT список полів, значення яких потрібно отримати
FROM список таблиць, з яких вибираються дані
WHERE умова, яку мають задовольняти записи,
      що вибираються
```

- ◆ Запит **SELECT x FROM y WHERE z** слід інтерпретувати так: «вибрати поля *x* тих записів таблиці *y*, які задовольняють умову *z*».
- ◆ Оператор **a INNER JOIN b ON c** використовується у фразі **FROM** і виконує з'єднання таблиць. Він «зчіплює» ті записи таблиць *a* і *b*, які, разом узяті, відповідають умові *c*.
- ◆ Оператор **IN** дає змогу сконструювати у фразі **WHERE** умову з підзапитом. Вираз значення **IN** (підзапит) вважається

істинним, якщо значення належить множині значень, отриманій у результаті виконання підзапиту, та хибним в іншому випадку.

- ◆ Запити, у яких перед дієсловом розташовано частку «не», реалізують за допомогою оператора **NOT IN** у фразі **WHERE**. Вираз значення **NOT IN** (підзапит) вважається істинним, якщо значення не належить множині значень, отриманій у результаті виконання підзапиту, та хибним в іншому випадку.

Завдання для самостійного виконання

Реалізуйте мовою SQL такі запити.

1. Відобразити всю інформацію про учнів, яких навчають учителі-чоловіки.
2. Для кожного вчителя відобразити всю інформацію про учнів, яких він навчає.
3. Відобразити прізвища та імена всіх учнів, крім тих, які вчаться в 11 класах.
4. Визначити прізвища та імена учнів, яких не навчає Василь Петрович Корбут.
5. Визначити прізвища вчителів, які викладають не тільки в 11 класах (зауважте, що вони можуть не викладати в 11 класах взагалі).
- 6*. Для кожного вчителя відобразити всю інформацію про учнів, яких він не навчає.
- 7*. Визначити прізвища вчителів, які викладають принаймні в одному тому ж класі, що і Сошко Катерина Миколаївна.
- 8*. Визначити пари прізвищ учителів, які викладають тільки в різних класах (тобто вчитель 1 не викладає в жодному з тих класів, де викладає вчитель 2, і навпаки).

Питання для роздумів

1. Нижче наведено формулювання та реалізації чотирьох запитів. Визначте, яка реалізація якому запиту відповідає.

Формулювання

- I. Визначити прізвища вчителів, які викладають принаймні в одному 11 класі.
- II. Визначити прізвища вчителів, які викладають в усіх 11 класах.
- III. Визначити прізвища вчителів, які викладають не в усіх 11 класах.
- IV. Визначити прізвища вчителів, які не викладають принаймні в одному 11 класі.

Реалізація

```
SELECT прізвище --А
FROM Учителі
WHERE NOT EXISTS (SELECT * FROM Викладання
                  WHERE клас Like "*11" AND
                  клас NOT IN (SELECT клас
                              FROM Викладання
                              WHERE учитель=
                              Учителі.паспорт))
```

```
SELECT прізвище --Б
FROM Учителі
WHERE EXISTS (SELECT * FROM Викладання
              WHERE клас Like "*11" AND
              клас NOT IN (SELECT клас
                           FROM Викладання
                           WHERE учитель=
                           Учителі.паспорт))
```

```
SELECT прізвище --В
FROM Учителі
WHERE EXISTS (SELECT * FROM Викладання
              WHERE клас Like "*11" AND
              клас IN (SELECT клас
                       FROM Викладання
                       WHERE учитель=
                       Учителі.паспорт))
```

```
SELECT прізвище --Г
FROM Учителі
WHERE NOT EXISTS (SELECT * FROM Викладання
                  WHERE клас Like "*11" AND
                  клас IN (SELECT клас
                           FROM Викладання
                           WHERE учитель=
                           Учителі.паспорт))
```

2. Припустимо, що предметною областю бази даних є механічні деталі. База містить лише одну таблицю, у якій є два поля: № деталі та № складової. У цій таблиці зберігаються відомості про те, які деталі з яких складаються. Наприклад, з поданої таблиці видно, що деталь 1 складається з деталей 2 і 3, а деталь 3 — з деталей 4 і 5.

№ деталі	№ складової
1	2
1	3
3	4
3	5

Сформулюйте для цієї бази даних запити, які:

- а) неможливо реалізувати за допомогою конструктора запитів;
- б*) неможливо реалізувати засобами мови SQL.

Завдання для досліджень

1. Реалізуйте завдання 9.3 та завдання для самостійного виконання 1–2 мовою SQL, але без оператора INNER JOIN.
- 2*. З'ясуйте, який синтаксис має та як використовується оператор EXISTS. Реалізуйте за допомогою операторів EXISTS і NOT IN запит визначити назви класів, у яких принаймні один вчитель не викладає.
- 3*. Реалізуйте за допомогою операторів NOT EXISTS і NOT IN запит визначити назви класів, у яких викладають усі вчителі.