

Лабораторна робота 3

Створення запитів до бази даних

Мета роботи: Навчитися створювати прості запити до бази даних та створювати запити за умовою.

ЗАВДАННЯ

1. Запустити програму MS SQL Management Studio
2. Виконати завдання, перераховані в «Порядку виконання»

ЗМІСТ ЗВІТУ

Найменування і мета роботи
Завдання

ТЕОРЕТИЧНА ЧАСТИНА

Припустимо є БД forum, в якій є три таблиці: users (користувачі), topics (теми) і posts (повідомлення). І ми хочемо подивитися, які дані в них містяться. Для цього в SQL існує оператор **SELECT**. Синтаксис його виконання наступний:

```
SELECT что_выбрать FROM откуда_выбрать;
```

Замість "что_выбрать" ми повинні вказати або ім'я стовпця, значення якого хочемо побачити, або імена кількох стовпців через кому, або символ зірочки (*), що означає вибір всіх стовпців таблиці. Замість "откуда_выбрать" слід вказати ім'я таблиці.

Наприклад, нас цікавлять всі стовпці з таблиці users:

```
SELECT * FROM users;
```

Припустимо, що ми хочемо подивитися тільки стовпець id_user. Для цього в запиті ми вкажемо ім'я цього стовпця:

```
SELECT id_user FROM users;
```

Ну, а якщо ми захочемо подивитися, наприклад, імена та e-mail наших користувачів, то ми перерахуємо стовпці які цікавлять через кому:

```
SELECT name, email FROM users;
```

Що якщо ми хочемо щоб наші дані виводилися, наприклад, за алфавітом. Для цього в SQL існує ключове слово **ORDER BY** після котрого вказується ім'я стовпця, по якому буде відбуватися сортування. Синтаксис наступний:

```
SELECT ім'я_стовпця FROM ім'я_таблиці ORDER BY  
ім'я_стовпця_сортування;
```

За замовчуванням сортування йде по зростанню, але це можна змінити, додавши ключове слово **DESC**. Тепер наші дані будуть відсортовані в порядку спаданням.

Слід зазначити, що вертикальна вибірка може містити дублікати рядків в тому випадку, якщо вона не містить потенційного ключа, якій однозначно визначає запис. Якщо потрібно отримати тільки унікальні рядки, то можна використовувати ключове слово **DISTINCT**:

Якщо необхідно виконати вибірку з умовою, тоді використовується конструкція

```
SELECT ім'я_стовпця FROM ім'я_таблиці WHERE умова;
```

Для нашого прикладу умовою є ідентифікатор користувача, тобто нам потрібні тільки ті рядки, в стовпці `id_author` яких стоять 4:

```
SELECT * FROM topics WHERE id_author = 4;
```

Або ми хочемо дізнатися, хто створив тему "фітнес":

```
SELECT * FROM topics WHERE topic_name = 'фітнес';
```

Звичайно, було б зручніше, щоб замість `id` автора, виводилося його ім'я, але імена зберігаються в іншій таблиці. У наступних уроках ми дізнаємося, як вибирати дані з декількох таблиць. А поки дізнаємося, які умови можна задавати, використовуючи ключове слово **WHERE**.

Оператор	Опис
= (Равно)	Відбираються значення рівні зазначеному <i>приклад:</i> SELECT * FROM topics WHERE id_author = 4;
> (Більше)	Відбираються значення більше зазначеного <i>приклад:</i> SELECT * FROM topics WHERE id_author > 2;
<(Менше)	Відбираються значення менше зазначеного <i>приклад:</i> SELECT * FROM topics WHERE id_author <3;
> = (Більше або дорівнює)	Відбираються значення великі і рівні зазначеному <i>приклад:</i> SELECT * FROM topics WHERE id_author > = 2;
<= (Менше або дорівнює)	Відбираються значення менші і рівні зазначеному <i>приклад:</i> SELECT * FROM topics WHERE id_author <= 3;
! = (Не дорівнює)	Відбираються значення не рівні зазначеному <i>приклад:</i> SELECT * FROM topics WHERE id_author! = 2;
IS NOT NULL	Відбираються рядки, які мають значення в зазначеному полі <i>приклад:</i> SELECT * FROM topics WHERE id_author IS NOT NULL;
IS NULL	Відбираються рядки, які не мають значення в зазначеному поле <i>приклад:</i> SELECT * FROM topics WHERE id_author IS NULL;
BETWEEN (між)	Відбираються значення, що знаходяться між зазначеними <i>приклад:</i> SELECT * FROM topics WHERE id_author BETWEEN 1 AND 3;

IN	Відбираються значення, що відповідають зазначеним <i>приклад:</i> SELECT * FROM topics WHERE id_author IN (1, 4);
NOT IN	Відбираються значення, крім зазначених <i>приклад:</i> SELECT * FROM topics WHERE id_author NOT IN (1, 4);
LIKE (відповідність)	Відбираються значення, відповідні зразком <i>приклад:</i> SELECT * FROM topics WHERE topic_name LIKE 'фит%'; Можливі метасимволи оператора LIKE розглянути нижче.
NOT LIKE	Відбираються значення, що не відповідають зразку <i>приклад:</i> SELECT * FROM topics WHERE topic_name NOT LIKE 'вів%';

Метасимволи оператора *LIKE*

Пошук з використанням метасимволів може здійснюватися тільки в текстових полях.

Найпоширеніший метасимвол - **%**. Він означає будь-які символи. Наприклад, якщо нам треба знайти слова, що починаються з літер "фит", то ми напишемо LIKE 'фит%', а якщо ми хочемо знайти слова, які містять символи "клуб", то ми напишемо LIKE '% клуб%'.

Ще один часто використовуваний метасимвол - **_**. На відміну від%, котрий позначає кілька або жодного символу, нижнє підкреслення позначає рівно один символ.

Предикат **LIKE** порівнює рядок, зазначений в першому вираженні, для обчислення значення рядка, зі зразком, який визначений в другому вираженні для обчислення значення рядка. У зразку дозволяється використовувати два трафаретних символу:

- символ підкреслення (**_**), який можна застосовувати замість будь-якого одиничного символу в перевіряється значенні;

- символ відсотка (%) замінює послідовність будь-яких символів (число символів в послідовності може бути від 0 і більше) в значенні.

Якщо значення що перевіряється відповідає зразку з урахуванням трафаретних символів, то значення предиката буде **TRUE**. Нижче наводиться декілька прикладів написання зразків.

Зразок	Опис
'Abc%'	Будь-які рядки, які починаються з літер «abc»
'Abc_ '	Рядки довжиною строго 4 символу, причому першими символами рядка повинні бути «abc»
'% Z'	Будь-яка послідовність символів, яка обов'язкове закінчується символом «Z»
'% Rost%'	Будь-яка послідовність символів, що містить слово «Rost» в будь-якій позиції рядка
'% % %'	Текст, що містить не менше 2-х прогалін, наприклад, "World Wide Web"

Якщо шукана стрічка містить універсальний шаблон, то слід поставити керуючий символ в реченні ESCAPE. Цей керуючий символ повинен використовуватися в зразку перед трафаретним символом, повідомляючи про те, що останній слід трактувати як звичайний символ. Наприклад, якщо в деякому полі слід відшукати всі значення, що містять символ «_», то шаблон '% _%' призведе до того, що будуть повернуті всі записи з таблиці. В даному випадку шаблон потрібно записати таким чином:

```
'% # _%' ESCAPE '#'
```

Істинне значення предиката **LIKE** присвоюється відповідно до наступних правил:

- якщо або значення що перевіряється, або зразок, або керуючий символ є **NULL**, істинне значення дорівнює **UNKNOWN**;
- в іншому випадку, якщо значення що перевіряється і зразок мають нульову довжину, істинне значення дорівнює **TRUE**;

- в іншому випадку, якщо значення що перевіряється відповідає шаблону, то предикат **LIKE** дорівнює **TRUE**;
- якщо не дотримується жодна з перерахованих вище умов, предикат **LIKE** дорівнює **FALSE**.

Предикат **LIKE** в його стандартної редакції не підтримує регулярних виразів, хоча ряд реалізацій (зокрема, Oracle) допускає їх використання, розширюючи можливості стандарту.

В **Transact-SQL**, крім стандартних символів-шаблонів ("%" і "_"), існує ще пара символів, які роблять цей предикат **LIKE** більш гнучким інструментом. Цими символами є:

- [] - одиночний символ з набору символів (наприклад, [zxy]) або діапазону ([az]), зазначених у квадратних дужках. При цьому можна перерахувати відразу кілька діапазонів (наприклад, [0-9a-z]);
- ^ - який в поєднанні з квадратними дужками виключає з пошукового зразка символи з набору або діапазону.

Якщо ми хочемо дізнатися, хто створив тему "фітнес", і зробимо відповідний запит:

```
SELECT * FROM topics WHERE topic_name = 'фітнес';
```

Те замість імені автора, ми отримаємо його ідентифікатор. Це і зрозуміло, адже ми робили запит до однієї таблиці - Теми, а імена авторів зберігаються в іншій таблиці - Користувачі. Тому, дізнавшись ідентифікатор автора теми, нам треба зробити ще один запит - до таблиці Користувачі, щоб дізнатися його ім'я.

У SQL передбачена можливість об'єднувати такі запити в один шляхом перетворення одного з них в підзапит (вкладений запит). Отже, щоб дізнатися, хто створив тему "фітнес", ми зробимо наступний запит:

```
SELECT name FROM users WHERE id_user IN  
(SELECT id_author FROM topics WHERE topic_name = 'фітнес');
```

Тобто, після ключового слова **WHERE**, в умову ми записуємо ще один запит. SQL спочатку обробляє підзапит, повертає id_author = 3, і це значення передається в пропозицію **WHERE** зовнішнього запиту.

В одному запиті може бути кілька підзапитів, синтаксис у такого запиту наступний:

```
SELECT ім'я_стовпця FROM ім'я_таблиці WHERE частина умови
IN
(SELECT ім'я_стовпця FROM ім'я_таблиці WHERE частина умови
IN
(SELECT ім'я_стовпця FROM ім'я_таблиці WHERE умова)
);
```

Зверніть увагу, що підзапити можуть вибирати тільки один стовбець, значення якого вони будуть повертати зовнішнім запитам. Спроба вибрати кілька стовпців призведе до помилки.

Тепер ускладнити завдання, дізнаємося, в яких темах залишав повідомлення автор теми "нічні клуби":

```
SELECT topic_name FROM topics WHERE id_topic IN
(SELECT id_topic FROM posts WHERE id_author IN
(SELECT id_author FROM topics WHERE topic_name = 'нічні
клуби'));
```

Давайте розберемося, як це працює.

- Спочатку SQL виконає найглибший запит:

```
SELECT id_author FROM topics WHERE topic_name = 'нічні клуби'
```

- Отриманий результат (id_author = 2) передасть у зовнішній запит, який матиме вигляд:

```
SELECT id_topic FROM posts WHERE id_author IN (2);
```

- Отриманий результат (id_topic: 4,1) передасть у зовнішній запит, який матиме вигляд:

```
SELECT topic_name FROM topics WHERE id_topic IN (4,1);
```

- І видасть остаточний результат (topic_name: про рибалку, про риболовлю). Тобто автор теми "нічні клуби" залишав повідомлення в темі "Про рибалці", створеної Сергієм (id = 1) і в темі "Про риболовлю", створеної Світланою (id = 4).

Є момент, на якій варто звернути увагу: не рекомендується створювати запити зі ступенем вкладення більше трьох. Це призводить до збільшення часу виконання і до складності сприйняття коду.

ПОРЯДОК ВИКОНАННЯ

Виконайте нижченаведені завдання

№	Завдання	Відповідь															
1	Знайдіть номер моделі, швидкість та розмір жорсткого диска для всіх ПК вартістю менше 500 дол. Вивести: model, speed та hd	<table border="1"> <thead> <tr> <th>model</th> <th>speed</th> <th>hd</th> </tr> </thead> <tbody> <tr> <td>1232</td> <td>500</td> <td>10.0</td> </tr> <tr> <td>1232</td> <td>450</td> <td>8.0</td> </tr> <tr> <td>1232</td> <td>450</td> <td>10.0</td> </tr> <tr> <td>1260</td> <td>500</td> <td>10.0</td> </tr> </tbody> </table>	model	speed	hd	1232	500	10.0	1232	450	8.0	1232	450	10.0	1260	500	10.0
model	speed	hd															
1232	500	10.0															
1232	450	8.0															
1232	450	10.0															
1260	500	10.0															
2	Знайдіть виробників принтерів. Вивести: maker	<table border="1"> <thead> <tr> <th>maker</th> </tr> </thead> <tbody> <tr> <td>A</td> </tr> <tr> <td>D</td> </tr> <tr> <td>E</td> </tr> </tbody> </table>	maker	A	D	E											
maker																	
A																	
D																	
E																	
3	Знайдіть номер моделі, обсяг пам'яті та розміри екранів ПК-блокнотів, ціна яких перевищує 1000 дол.	<table border="1"> <thead> <tr> <th>model</th> <th>ram</th> <th>screen</th> </tr> </thead> <tbody> <tr> <td>1750</td> <td>128</td> <td>14</td> </tr> <tr> <td>1298</td> <td>64</td> <td>15</td> </tr> <tr> <td>1752</td> <td>128</td> <td>14</td> </tr> </tbody> </table>	model	ram	screen	1750	128	14	1298	64	15	1752	128	14			
model	ram	screen															
1750	128	14															
1298	64	15															
1752	128	14															
4	Знайдіть усі записи таблиці Printer для кольорових принтерів.	<table border="1"> <thead> <tr> <th>code</th> <th>model</th> <th>color</th> <th>type</th> <th>price</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>1434</td> <td>y</td> <td>Jet</td> <td>290.0000</td> </tr> <tr> <td>2</td> <td>1433</td> <td>y</td> <td>Jet</td> <td>270.0000</td> </tr> </tbody> </table>	code	model	color	type	price	3	1434	y	Jet	290.0000	2	1433	y	Jet	270.0000
code	model	color	type	price													
3	1434	y	Jet	290.0000													
2	1433	y	Jet	270.0000													
5	Знайдіть номер моделі, швидкість та розмір жорсткого диска ПК, що мають 12x або 24x CD та ціну менше 600 дол.	<table border="1"> <thead> <tr> <th>model</th> <th>speed</th> <th>hd</th> </tr> </thead> <tbody> <tr> <td>1232</td> <td>500</td> <td>10.0</td> </tr> <tr> <td>1232</td> <td>450</td> <td>8.0</td> </tr> <tr> <td>1232</td> <td>450</td> <td>10.0</td> </tr> <tr> <td>1260</td> <td>500</td> <td>10.0</td> </tr> </tbody> </table>	model	speed	hd	1232	500	10.0	1232	450	8.0	1232	450	10.0	1260	500	10.0
model	speed	hd															
1232	500	10.0															
1232	450	8.0															
1232	450	10.0															
1260	500	10.0															
6	Знайдіть виробників ПК із процесором не менше 450 МГц. Вивести: Maker	<table border="1"> <thead> <tr> <th>maker</th> </tr> </thead> <tbody> <tr> <td>A</td> </tr> <tr> <td>B</td> </tr> <tr> <td>E</td> </tr> </tbody> </table>	maker	A	B	E											
maker																	
A																	
B																	
E																	

7	<p>У таблиці Product знайти моделі, які складаються лише з цифр або лише з латинських літер (A-Z, без урахування регістру).</p> <p>Вивести: номер моделі, тип моделі</p>	<table border="1"> <thead> <tr> <th>model</th> <th>type</th> </tr> </thead> <tbody> <tr><td>1121</td><td>PC</td></tr> <tr><td>1232</td><td>PC</td></tr> <tr><td>1233</td><td>PC</td></tr> <tr><td>1260</td><td>PC</td></tr> <tr><td>1276</td><td>Printer</td></tr> <tr><td>1288</td><td>Printer</td></tr> <tr><td>1298</td><td>Laptop</td></tr> <tr><td>1321</td><td>Laptop</td></tr> <tr><td>1401</td><td>Printer</td></tr> <tr><td>1408</td><td>Printer</td></tr> <tr><td>1433</td><td>Printer</td></tr> <tr><td>1434</td><td>Printer</td></tr> <tr><td>1499</td><td>Printer</td></tr> <tr><td>1750</td><td>Laptop</td></tr> <tr><td>1752</td><td>Laptop</td></tr> <tr><td>2112</td><td>PC</td></tr> <tr><td>2113</td><td>PC</td></tr> </tbody> </table>	model	type	1121	PC	1232	PC	1233	PC	1260	PC	1276	Printer	1288	Printer	1298	Laptop	1321	Laptop	1401	Printer	1408	Printer	1433	Printer	1434	Printer	1499	Printer	1750	Laptop	1752	Laptop	2112	PC	2113	PC
model	type																																					
1121	PC																																					
1232	PC																																					
1233	PC																																					
1260	PC																																					
1276	Printer																																					
1288	Printer																																					
1298	Laptop																																					
1321	Laptop																																					
1401	Printer																																					
1408	Printer																																					
1433	Printer																																					
1434	Printer																																					
1499	Printer																																					
1750	Laptop																																					
1752	Laptop																																					
2112	PC																																					
2113	PC																																					
8	<p>Вивести всі моделі та ціни ноутбуків, які випускає виробник В. Ціни сортувати у спадному порядку.</p>																																					
9	<p>Вивести всі моделі та ціни принтерів у ціновому діапазоні від 200\$ до 300\$.</p>																																					
10	<p>Сортувати таблицю Product: виробники за зростанням, моделі зі спадання.</p>																																					