

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Звіт до лабораторної роботи №7
З ПРЕДМЕТУ “ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ”

Виконав:

ст. гр. КН-211

Шебеко Андрій

Викладач:

Якимишин Х.М.

Львів – 2020

Лабораторна робота №7
з курсу "ОБДЗ"
на тему:
"Запити на вибір даних з таблиць бази даних"

Мета роботи: Розробити SQL запити відбору даних з одиничних та з'єднаних таблиць, в тому числі з використанням підзапитів, натурального, умовного та лівого з'єднання, із застосуванням у критеріях вибірки функцій та операторів, в т. ч. LIKE, BETWEEN, IS NULL, IS NOT NULL, IN (...), NOT IN (...), ALL, SOME, ANY, EXISTS.

Короткі теоретичні відомості.

Для вибирання даних з таблиць використовується директива SELECT, яка може містити інші директиви SELECT (підзапити, або вкладені запити) та директиви з'єднання таблиць.

SELECT

```
[ALL | DISTINCT | DISTINCTROW ]
[STRAIGHT_JOIN]
[SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
елемент_вибірки [, елемент_вибірки ...]
[FROM перелік_таблиць]
[WHERE умова_відбору]
[GROUP BY {ім'я_поля | синонім | позиція_поля}
[ASC | DESC], ...]
[HAVING умова_відбору]
[ORDER BY {ім'я_поля | синонім | позиція_поля}
[ASC | DESC], ...]
[LIMIT {к-сть_рядків [OFFSET зміщення]}]
[PROCEDURE ім'я_процедури(аргументи)]
[INTO OUTFILE 'ім'я_файлу' опції_експорту
| INTO DUMPFILE 'ім'я_файлу'
| INTO змінна [, змінна]]
```

Параметри:

SELECT

Вказує поля, константи та вирази, що будуть відображатися у результатах запиту.

Директива вимагає чіткого дотримання порядку ключових слів FROM, WHERE і т.д.

елемент_вибірки

Вказує елемент, який буде включатися в результати відбору. Такими елементами можуть бути: ім'я поля, константа або вираз. Кожному елементу можна присвоїти ім'я-псевдонім, яке буде відображатись у результатах запиту. Для цього після назви елемента слід дописати AS *псевдонім*.

перелік_таблиць

Назви таблиць, з яких здійснюється вибір значень. Тут можна задавати синоніми назвам таблиць (*ім'я_таблиці AS синонім*), використовувати підзапити SELECT для формування таблиці з вказаним синонімом, з'єднувати декілька таблиць.

WHERE

Вказує критерії порівняння (або підзапити) для відбору рядків.

GROUP BY

Групує (і одночасно сортує) рядки за вказаними полями. Поля можна вказувати за іменами, синонімами або порядковими номерами в таблиці.

ORDER BY

Сортує рядки за вказаними полями. За замовчуванням – за зростанням значень (ASC).

HAVING

Дає можливість застосування до значень полів агрегатних функцій (COUNT, AVG, MIN, MAX тощо) при відборі чи групуванні рядків. Після слова WHERE ці функції не працюють, однак у всіх інших випадках слід використовувати саме WHERE.

LIMIT

Обмежує кількість рядків, повернутих в результаті запиту.

OFFSET

Вказує зміщення для LIMIT – з якого рядка в результатах запиту почати відбирати потрібну кількість рядків.

PROCEDURE

Задає назву збереженої процедури, яка повинна обробляти результат запиту.

INTO

Вказує місце, куди будуть збережені результати запиту. Це може бути як зовнішній файл, так і параметри чи змінні, визначені користувачем. Кількість змінних має бути рівна кількості полів у результаті.

DISTINCT | DISTINCTROW

Видалення з результату рядків-дублікатів. За замовчуванням вибираються всі рядки.

STRAIGHT_JOIN

Опція, яка строго задає порядок вибирання кортежів зі з'єднаних таблиць в порядку переліку таблиць. (Оптимізатор запитів MySQL іноді змінює цей порядок.)

SQL_CACHE | SQL_NO_CACHE

Явним чином вмикає/вимикає зберігання результатів запиту у кеші запитів MySQL. За замовчуванням, кешування запитів залежить від системної змінної query_cache_type.

SQL_CALC_FOUND_ROWS

Вказує, що при виконанні запиту слід обчислити загальну кількість рядків в результаті, ігноруючи опцію обмеження LIMIT. Цю кількість рядків потім можна отримати командою SELECT FOUND_ROWS().

Для вибору записів зі з'єднаних таблиць використовується директива SELECT разом із директивами JOIN у переліку таблиць. Наприклад:

```
SELECT * FROM author INNER JOIN comment
ON author.authorID = comment.authorID;
```

Параметри директиви:

INNER JOIN

Внутрішнє з'єднання. Результати вибору будуть містити тільки ті рядки, для яких існують один або більше відповідних рядків з іншої таблиці. В MySQL – є синонімом директиви CROSS JOIN. Слід зауважити, що вибір рядків директивою SELECT з кількох таблиць, вказаних через кому, є аналогічним до явного використання директиви INNER JOIN. В обох випадках MySQL формує декартовий добуток усіх

Хід роботи

Для вивчення роботи директив вибору даних з таблиць розробимо та виконаємо такі запити над таблицями Author, Role, Comment.

1. Виберемо коментар користувача з іменем Lera:

```
#1
select comment
from comment
where username = 'Lera';
```

comment
Terrible

2. Використаємо ліве з'єднання для вибору назви та опису продукту разом з коментарем та іменем користувача.

```
#2
select product.product_name, product.description,
       comment.username, comment.comment
from `product` left join `comment`
on comment.product_id = product.product_id;
```

product_name	description	username	comment
Table	Big round-shape table	Andrii	Great
Table	Big round-shape table	Katya	Top
Table	Big round-shape table	Zoya	Excelent
Table	Big round-shape table	Andrii	Bad
Chair	small light chair	Vitya	Super
Umbrella	Big square `1715` umbrella	Lera	Terrible
Table	Small round-shape table	NULL	NULL

3. Також об'єднаємо назву продукту та оцінку.

```
#3
select product.product_name, comment.rating
from product inner join comment
on comment.product_id = product.product_id
where comment.rating > 3;
```

product_name	rating
Table	4
Table	5
Chair	4
Table	4

4. Тепер об'єднаємо значення з 3х таблиць: продукт, коментар, клієнт.

```
#4
select product.product_name, comment.comment,
       comment.username, client.`second name`
from (product inner join comment) inner join client
on comment.product_id = product.product_id
and comment.username = client.`first name`;
```

product_name	comment	username	second name
Table	Great	Andrii	Sheba
Table	Bad	Andrii	Sheba

5. Виберемо ім'я та прізвище користувачів які залишили відгуки товарам дешевшим за 499од.

#5

```
select comment.comment, comment.username,
       client.`second name`, client.register_date
from comment inner join client
on comment.username = client.`first name`
where comment.product_id in (
    select product.product_id
    from product
    where product.price < 499)
order by client.`second name`;
```

comment	username	second name	register_date
Super	Vitya	Surname	2020-03-01 00:00:00
Terrible	Lera	Vitya	2020-04-01 00:00:00

6. Виберемо продукти до яких немає коментарів.

#6

```
select product.description from product
where not exists (
    select * from comment
    where comment.product_id = product.product_id
);
```

description
Small round-shape table

7. Виберемо лише коментарі коротші за 5 символів і які починаються з літери S.

#7

```
select comment from comment
where char_length(comment) <= 5
and left(comment, 1) = 'S';
```

comment
Super

Висновок: на цій лабораторній роботі було вивчено методи вибору даних зі з'єднаних таблиць БД засобами SQL та виконано запити до бази даних з використанням директив SELECT та JOIN, а також складних критеріїв в умові вибірки.