

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



Звіт до лабораторної роботи №7

з дисципліни
“ОБДЗ”

Виконав:
ст. гр. КН-211
Ільків Андрій

Викладач:
Якимишин Х.М.

Львів – 2020

“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 кортежів, і з результату вибирає лише ті, для яких виконується умова відбору (порівняння) ON. LEFT JOIN

Вказує на те, що результати вибору будуть містити всі рядки з таблиці, яка стоїть зліва від слова JOIN і тільки відповідні їм рядки з таблиці справа (ті, для яких виконується вказана умова). Якщо відповідний рядок відсутній, виводяться значення NULL. RIGHT JOIN

Вказує на те, що результати вибору будуть містити всі рядки з таблиці, яка вказана справа від JOIN і тільки відповідні їм рядки з таблиці зліва. Для сумісності на практиці використовують в основному LEFT JOIN. ON умова

Вказує поля, за якими слід з'єднувати таблиці. Замість ON можна також використовувати USING перелік_спільних_полів. В цьому випадку спільне поле буде відображене в результатах запиту лише один раз. NATURAL JOIN

Еквівалент внутрішньому з'єднанню за всіма однаковими полями (з опцією USING *).

У таблиці нижче описано основні функції порівняння, які можна використовувати при формуванні складних критеріїв вибору.

Функція Опис

STRCMP(*рядок1*, *рядок2*)

Порівнює два рядки. Повертає значення 0 (False) якщо рядки однакові, -1 якщо перший рядок менший за другий, і 1 (True) в усіх інших випадках.

LIKE *рядок*

Порівняння з рядком-шаблоном. В шаблоні можна використовувати знаки % (довільні символи) і _ (довільний символ).

REGEXP *рядок* Порівняння з рядком з використанням регулярних виразів. Функція-синонім – RLIKE.

MATCH (*поля*) AGAINST (*рядок*) Здійснює пошук рядка у вказаних текстових полях таблиці. (Тільки для MyISAM-таблиць.)

BETWEEN ... AND ... Повертає 1, якщо значення належить даному діапазону. NOT BETWEEN ... AND ... Повертає 1, якщо значення не належить діапазону.

IN(*arg1, arg2, ...*)

Перевірка належності множині. Повертає 1, якщо значення співпадає хоча б із одним аргументом, і 0 – у протилежному випадку. Повертає NULL, якщо значення є NULL, або якщо співпадиння не знайдено, а один із аргументів є NULL.

NOT IN(*arg1, arg2, ...*)

Повертає 1, якщо значення не міститься у множині аргументів, і 0 – у протилежному випадку. Повертає NULL аналогічно до функції IN(). IS NULL, IS NOT NULL Перевірка визначеності значення.

LEAST(*arg1, arg2, ...*) Повертає мінімальне значення серед аргументів.

Повертає NULL, якщо хоча б один із аргументів є NULL.

GREATEST(*arg1, arg2, ...*) Повертає максимальне значення серед аргументів.

Повертає NULL, якщо хоча б один із аргументів є NULL.

Для формування критеріїв вибору та підзапитів також використовують наступні оператори порівняння:

= <=> ня 1 (True) коли обидва вирази рівні, інакше Ікщо хоча б один з виразів приймає значення NULL.

ті двох виразів. Якщо відбувається порівняння двох не

Перевірка рівності виразів, яке враховує NULL значення. Повертає 1, якщо обидва вирази приймають значення NULL, або рівні значення. Повертає 0, якщо один із виразів приймає значення NULL, або значення виразів не рівні. >, >= Порівняння двох виразів.

Результатом є 1, якщо ліве значення більше (більше рівне) ніж праве, інакше результатом є 0. Якщо хоча б один з виразів приймає значення NULL, то результатом теж стає NULL. <, <= Порівняння двох виразів. Результатом є 1, якщо ліве значення менше (менше рівне) ніж праве, інакше результатом є 0. Якщо хоча б один з виразів

приймає значення NULL, то результатом теж є NULL. !=, <> Перевірка на не рівність.

Результат набуває значення 1, якщо ліве значення менше або більше ніж праве, інакше результатом є 0. Якщо хоча б один з виразів приймає значення NULL, то результатом теж є NULL. ALL, SOME, ANY

Оператори, які можна використовувати після операторів порівняння. Задають необхідність виконання оператора хоча б для одного (SOME, ANY) чи всіх (ALL) елементів, отриманих в результаті підзапиту. На відміну від функцій IN(), NOT IN() оператори не працюють зі списками значень. [NOT] EXISTS

Оператор, який використовують після ключового слова WHERE. Повертає 1, якщо підзапит повертає хоча б одне визначене значення, і 0 – у протилежному випадку.

Хід роботи.

Для вивчення роботи директив вибору даних з таблиць розробимо та виконаємо такі запити над таблицями student, studentinfo, teacher.

1. Показати користувачів і їхні коментарі (ліве з'єднання таблиць).
2. Показати перелік користувачів у групі Guests (натуральне з'єднання).
3. Показати всі коментарі користувачів з груп Guests та SUGroup2 (умовне з'єднання).
4. Показати останні 3 коментарі користувачів з груп Guests та SUGroup2 (підзапит).
5. Визначити користувачів, які не написали жодного повідомлення.
6. Визначити користувачів, паролі яких не відповідають вимогам безпеки (менші за 8 символів або не містять цифр).

1. Виберемо всіх користувачів з їхніми даними. Для цього потрібно виконати ліве з'єднання.

SELECT

*

FROM

student

LEFT JOIN

studentinfo **ON** student.student_id =
studentinfo.student_id;

Результат запиту:

	student_id	first_name	last_name	group_id	login	password	identification_num...	civil_number	medical_info	created	student_id
►	1	Vitalii	Solyridze	1	vtko	123qwe	16140228 4549	73841748499	In nec orci.	2015-04-20	1
	2	Viktor	Kachmaryk	4	bestlol	kiko	16760316 4968	66575980999	elit, a feugiat	2021-06-20	2
	3	Melanii	Volskiy	3	amazoff	stronk23	16130202 8079	37611934999	cursus luctus, ipsum	2020-11-18	3
	4	Klasniy	Perec	2	uberli	987yh	16540530 6100	40724679599	vulputate, lacus. Cras	2018-03-20	4

2. Виберемо викладачів з їхніми категоріями автомобілів. Для цього виконаємо умовне з'єднання таблиць Teacher і Car за атрибутом *teacher_id*, використовуючи директиву INNER JOIN.

SELECT

teacher.first_name, teacher.last_name, car.category

FROM

teacher

INNER JOIN

car **ON** teacher.car_id = car.car_id;

Результат запиту:

	first_name	last_name	category
►	Gabriel	De Santa	C
	Entity	File	B
	Kekowish	Rofluckyi	A

3. Виберемо всі коментарі учнів які навчаються на автомобілях класу C і номер лекції, на якій вони знаходяться. Для цього виконаємо умовне з'єднання таблиць student і studentsgroup за атрибутом *student_id*, та таблиці category використовуючи директиву INNER JOIN.

SELECT

```

    student.first_name,
    student.last_name,
    studentsgroup.lectionId,
    category.category_description
FROM
    (student
    INNER JOIN studentsgroup)
    INNER JOIN
    category ON student.group_id = studentsgroup.group_id
    AND studentsgroup.category_id = category.category_id
WHERE
    category.category_description IN ('big cars');

```

Результат запиту:

	first_name	last_name	lectionId	category_descripti...
►	Vitalii	Solyridze	12	big cars
	Klasniy	Perec	15	big cars

4. Визначимо викладачів, які не мають автомобіля для викладання.

```

SELECT
    teacher.first_name, teacher.last_name
FROM
    teacher
WHERE
    NOT EXISTS( SELECT
        *
        FROM
            car
        WHERE
            car.car_id = teacher.car_id);

```


Результат запиту:

first_name	last_name

Отже, таких викладачів немає.

5. Визначимо користувачів, паролі яких не відповідають вимогам безпеки (менші за 8 символів або не містять цифр).

```
SELECT
    student.login, student.password
FROM
    student
WHERE
    CHAR_LENGTH(student.`password`) < 4
    OR student.password NOT REGEXP '[0-9]';
```

Результат запиту:

login	password
bestiol	kiko

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.