

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра систем штучного інтелекту

Лабораторна робота №7
із дисципліни
Бази даних

Виконав:
Ст. групи КН-207
Гуменчук Б.Р.
Прийняв:
Мельникова Н.І.

Львів – 2018 р.

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]]
```



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 <i>рядок</i>	Порівняння з рядком-шаблоном. В шаблоні можна використовувати знаки % (довільні символи) і _ (довільний символ).
REGEXP <i>рядок</i>	Порівняння з рядком з використанням регулярних виразів. Функція-синонім – RLIKE.
MATCH (<i>поля</i>) AGAINST (<i>рядок</i>)	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(<i>арг1, арг2, ...</i>)	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 співпадіння не знайдено, а один із аргументів є NULL.
NOT IN(<i>арг1, арг2, ...</i>)	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(<i>арг1, арг2, ...</i>)	Повертає мінімальне значення серед аргументів. Повертає NULL, якщо хоча б один із аргументів є NULL.
GREATEST(<i>арг1, арг2, ...</i>)	Повертає максимальне значення серед аргументів. Повертає NULL, якщо хоча б один із аргументів є NULL.

Хід роботи:



1. Виконаємо ліве з'єднання.

```
182 • select performer.idperformer, performer.name, performer.emeil, book.description, book.amount
183 from performer left join book on
184 performer.idperformer = book.performer_idperformer;
185
```

Result Grid					
Filter Rows: <input type="text"/>					
Export:  Wrap Cell Content: 					
	idperformer	name	emeil	description	amount
▶	1	Vasiliy	vasya_homilka@gmail.com	book_1	10
	1	Vasiliy	vasya_homilka@gmail.com	book_2	10
	1	Vasiliy	vasya_homilka@gmail.com	book_3	10
	3	Kolya	Kolyan@gmail.com	NULL	NULL
	4	Petro	petya_geniy@gmail.com	NULL	NULL



2. Виконаємо умовне з'єднання двох таблиць.

```
188 • select performer.name, performer.emeil, book.idbooks, book.amount, book.description
189 from performer inner join book on book.performer_idperformer = performer.idperformer
190 where performer.name = "Vasiliy";
```

Result Grid					
Filter Rows: <input type="text"/>					
Export:  Wrap Cell Content: 					
	name	emeil	idbooks	amount	description
▶	Vasiliy	vasya_homilka@gmail.com	4	10	book_1
	Vasiliy	vasya_homilka@gmail.com	5	10	book_2
	Vasiliy	vasya_homilka@gmail.com	6	10	book_3

3. Виконаємо умовне з'єднання трьох таблиць.

```
192 • select performer.name, performer.emeil, book.idbooks, book.amount, book.description, customer.name, customer.surname
193 from (performer inner join book) inner join customer
194 on book.performer_idperformer = performer.idperformer
195 and book.customer_idcustomer = customer.idcustomer
196 where performer.name in ("Vasiliy", "Kolya");
```

Result Grid						
Filter Rows: <input type="text"/>						
Export:  Wrap Cell Content: 						
	name	emeil	idbooks	amount	description	name surname
▶	Vasiliy	vasya_homilka@gmail.com	4	10	book_1	Ivan Franko
	Vasiliy	vasya_homilka@gmail.com	5	10	book_2	Ivan Franko
	Vasiliy	vasya_homilka@gmail.com	6	10	book_3	Ivan Franko

4. Виконаємо під запит в умові відбору.

```
198 • select performer.idperformer, performer.name, book.description, book.amount
199 from performer inner join book
200 on performer.idperformer = book.performer_idperformer
201 where book.performer_idperformer
202 in (select performer.idperformer from performer where performer.name in ("Vasiliy", "Kolya"))
203 order by performer.emeil desc limit 2;
```

<				
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:				
	idperformer	name	description	amount
▶	1	Vasiliy	book_1	10
	1	Vasiliy	book_2	10

Висновок: на цій лабораторній роботі було розглянуто операції реляційної алгебри та їх реалізація в MySQL.