

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА
ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту



Лабораторна робота №4
З курсу “Організація баз даних та знань”

Виконав:
студент групи КН-210
Марій Павло

Викладач:
Мельникова Наталя Іванівна

Тема: Запити на додавання, зміну та вилучення даних.

Мета: Розробити SQL-запити для внесення нових значень в таблиці в режимі одиничного та групового доповнення; розробити SQL-запити для внесення змін в рядки таблиць; розробити SQL-запити для вилучення вибраних рядків.

Теоретичні відомості

Для внесення значень в таблиці використовується директива **INSERT**:

INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]

[**INTO**] *назва_таблиці* [(*назва_поля*,...)]

| {**VALUES** | **VALUE**} ({*вираз* | **DEFAULT**},...),(...),...

| [**SET** *назва_поля*={*вираз* | **DEFAULT**}, ...]

| [*вибірка_даних*]

[**ON DUPLICATE KEY UPDATE** *назва_поля*=*вираз* [, *назва_поля*=*вираз*]...]

Наступна директива MySQL дозволяє заносити дані в таблицю із зовнішнього файлу:

LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] **INFILE**

'*назва_файлу*'

[**REPLACE** | **IGNORE**]

INTO TABLE *назва_таблиці*

[**CHARACTER SET** *кодування*]

[{**FIELDS** | **COLUMNS**}

[**TERMINATED BY** '*рядок_завершення*']

[**[OPTIONALLY] ENCLOSED BY** '*символ*']

[**ESCAPED BY** '*символ*']]

[**LINES** [**STARTING BY** '*рядок_початку*']

[**TERMINATED BY** '*рядок_завершення*']]

[**IGNORE** *кількість* **LINES**]

Для внесення змін в записи таблиці використовується директива **UPDATE**:

UPDATE [LOW_PRIORITY] [IGNORE] *назва_таблиці*

SET *назва_поля1*={*вираз*|**DEFAULT**} [, *назва_поля2*={*вираз*|**DEFAULT**}]...

[**WHERE** *умова_відбору*]

[**ORDER BY** ...]

[**LIMIT** *кількість_рядків*]

Для видалення записів з таблиці використовується директива **DELETE**:
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] **FROM** *назва_таблиці*
 [WHERE *умова_відбору*]
 [ORDER BY ...]
 [LIMIT *кількість_рядків*]

Хід роботи

Виконаємо такі запити для занесення даних у таблиці і їх подальшої модифікації.

1. Заповнимо таблицю type в режимі одиночного і групового доповнення.
 - Одиничне доповнення.

Код скрипта:

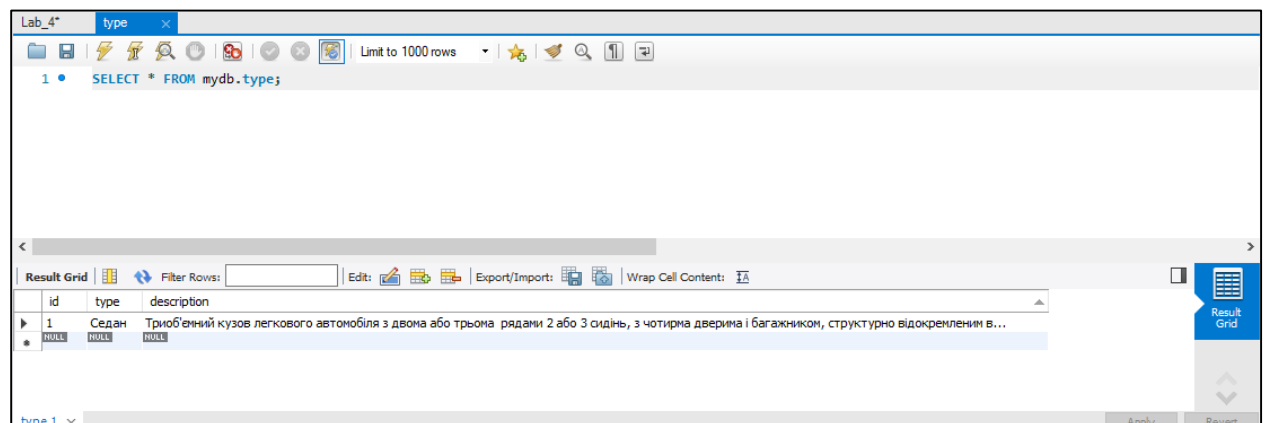
INSERT INTO mydb.type VALUES (

1, 'Седан', 'Триоб'ємний кузов легкового автомобіля з двома або трьома рядами 2 або 3 сидінь, з чотирма дверима і багажником, структурно відокремленим від пасажирського салону.'

);

Результат:

✓	2	16:43:30	INSERT INTO mydb.type VALUES (1, 'Седан', 'Триоб'ємний кузов легкового автом...	1 row(s) affected	0.953 sec
✓	3	16:44:27	SELECT * FROM mydb.type LIMIT 0, 1000	1 row(s) returned	0.032 sec / 0.000 sec



- Групове доповнення

Код скрипта:

INSERT INTO mydb.type VALUES

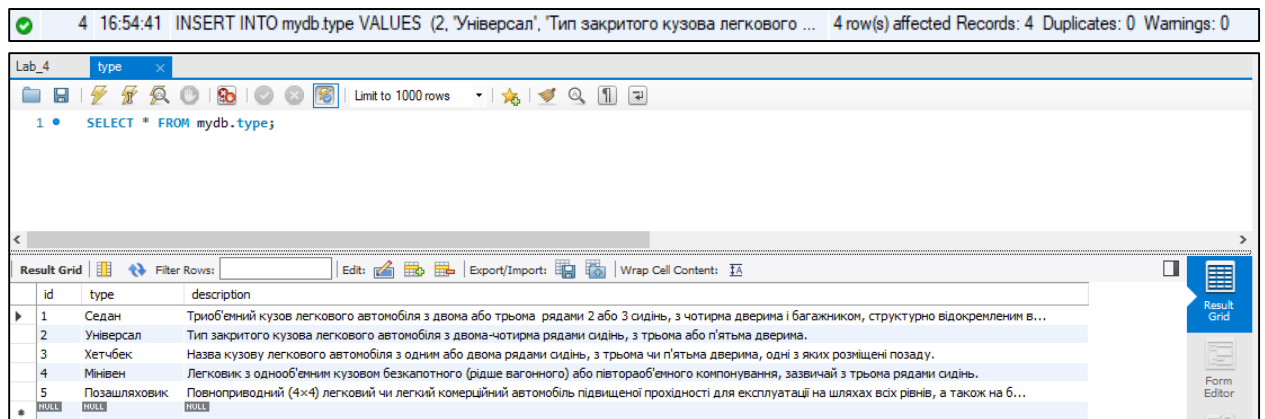
(2, 'Універсал', 'Тип закритого кузова легкового автомобіля з двома-чотирма рядами сидінь, з трьома або п'ятьма дверима.'),

(3, 'Хетчбек', 'Назва кузову легкового автомобіля з одним або двома рядами сидінь, з трьома чи п'ятьма дверима, одні з яких розміщені позаду.'),

(4, 'Мінівен', 'Легковик з однооб'ємним кузовом безкапотного (рідше вагонного) або півтораоб'ємного компонування, зазвичай з трьома рядами сидінь.'),

(5, 'Позашляховик', 'Повноприводний (4×4) легковий чи легкий комерційний автомобіль підвищеної прохідності для експлуатації на шляхах всіх рівнів, а також на бездоріжжі.');

Результат:



The screenshot shows a database management interface with a status bar at the top indicating a successful INSERT operation: "4 16:54:41 INSERT INTO mydb.type VALUES (2, 'Універсал', 'Тип закритого кузова легкового ... 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0". The main window displays the 'type' table with the following data:

id	type	description
1	Седан	Триоб'ємний кузов легкового автомобіля з двома або трьома рядами 2 або 3 сидінь, з чотирма дверима і багажником, структурно відокремленим в...
2	Універсал	Тип закритого кузова легкового автомобіля з двома-чотирма рядами сидінь, з трьома або п'ятьма дверима.
3	Хетчбек	Назва кузова легкового автомобіля з одним або двома рядами сидінь, з трьома чи п'ятьма дверима, одні з яких розміщені позаду.
4	Мінівен	Легковик з однооб'ємним кузовом безкапотного (рідше вагонного) або півтораоб'ємного компонування, зазвичай з трьома рядами сидінь.
5	Позашляховик	Повноприводний (4×4) легковий чи легкий комерційний автомобіль підвищеної прохідності для експлуатації на шляхах всіх рівнів, а також на б...

2. Створимо файли з даними і заповнимо ними решту таблиць баз даних.

Я створив текстові файли з даними, назви яких співпадають з назвами таблиць.

Код скрипта:

```
LOAD DATA INFILE 'D:\\Education\\Education_2.2\\BD\\Labs\\Lab_4\\user.txt'  
INTO TABLE mydb.user LINES
```

```
TERMINATED BY '\\r\\n';
```

```
LOAD DATA INFILE 'D:\\Education\\Education_2.2\\BD\\Labs\\Lab_4\\mark.txt'  
INTO TABLE mydb.mark LINES
```

```
TERMINATED BY '\\r\\n';
```

```
LOAD DATA INFILE
```

```
'D:\\Education\\Education_2.2\\BD\\Labs\\Lab_4\\model.txt' INTO TABLE  
mydb.model LINES
```

```
TERMINATED BY '\\r\\n';
```

```
LOAD DATA INFILE 'D:\\Education\\Education_2.2\\BD\\Labs\\Lab_4\\auto.txt'  
INTO TABLE mydb.auto LINES
```

```
TERMINATED BY '\\r\\n';
```

```
LOAD DATA INFILE
```

```
'D:\\Education\\Education_2.2\\BD\\Labs\\Lab_4\\advertisement.txt' INTO  
TABLE mydb.advertisement LINES
```

```
TERMINATED BY '\\r\\n';
```

LOAD DATA INFILE

```
'D:\\Education\\Education_2.2\\BD\\Labs\\Lab_4\\advertisement_user.txt' INTO  
TABLE mydb.advertisement_user LINES
```

```
TERMINATED BY '\\r\\n';
```

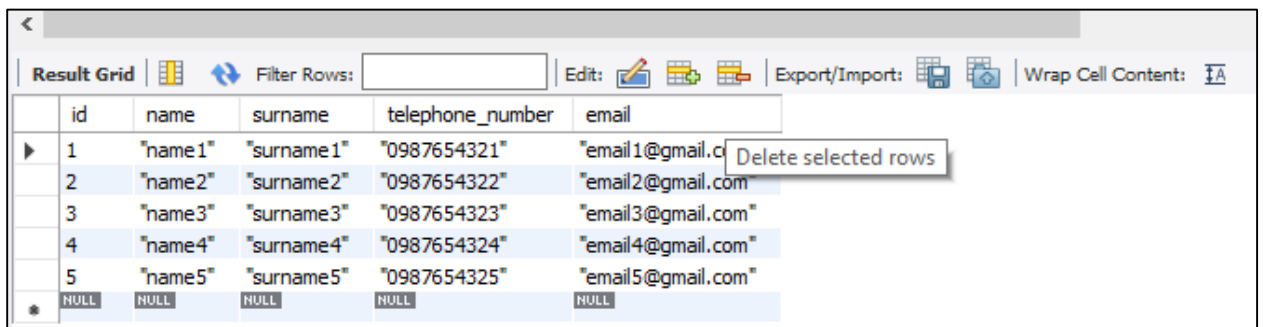
LOAD DATA INFILE

```
'D:\\Education\\Education_2.2\\BD\\Labs\\Lab_4\\comment.txt' INTO TABLE  
mydb.comment LINES
```

```
TERMINATED BY '\\r\\n';
```

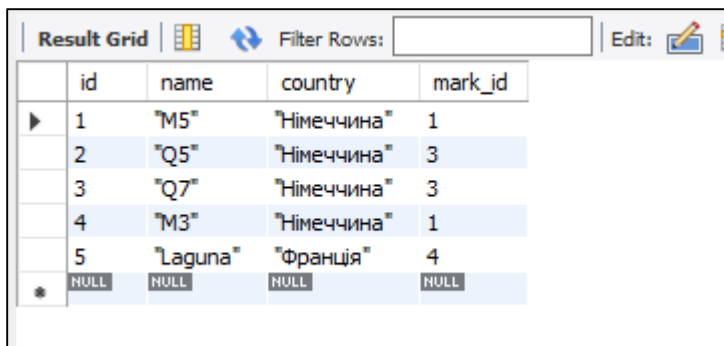
Результат:

Таблиця user:



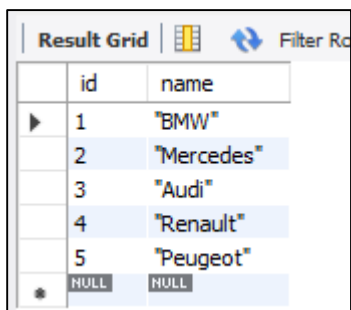
	id	name	surname	telephone_number	email
▶	1	"name1"	"surname1"	"0987654321"	"email1@gmail.c
	2	"name2"	"surname2"	"0987654322"	"email2@gmail.com"
	3	"name3"	"surname3"	"0987654323"	"email3@gmail.com"
	4	"name4"	"surname4"	"0987654324"	"email4@gmail.com"
	5	"name5"	"surname5"	"0987654325"	"email5@gmail.com"
*	NULL	NULL	NULL	NULL	NULL

Таблиця model:



	id	name	country	mark_id
▶	1	"M5"	"Німеччина"	1
	2	"Q5"	"Німеччина"	3
	3	"Q7"	"Німеччина"	3
	4	"M3"	"Німеччина"	1
	5	"Laguna"	"Франція"	4
*	NULL	NULL	NULL	NULL

Таблиця mark:



	id	name
▶	1	"BMW"
	2	"Mercedes"
	3	"Audi"
	4	"Renault"
	5	"Peugeot"
*	NULL	NULL

Таблиця comment:

	id	text	advertisement_id	user_id
▶	1	"comment1"	2	4
	2	"comment2"	2	4
	3	"comment3"	4	3
	4	"comment4"	4	3
	5	"comment5"	5	1
*	NULL	NULL	NULL	NULL

Таблиця auto:

	id	volume_engine	volume_power	transmission	type_id	model_id
▶	1	1.6	140	"transmission1"	2	1
	2	1.8	160	"transmission2"	3	2
	3	2	180	"transmission3"	5	4
	4	2.5	250	"transmission4"	4	3
	5	3	340	"transmission5"	1	5
*	NULL	NULL	NULL	NULL	NULL	NULL

Таблиця advertisement_user:

	id	advertisement_id	user_id
▶	1	2	2
	2	3	3
	3	3	1
	4	4	1
	5	5	1
*	NULL	NULL	NULL

Таблиця advertisement:

	id	price	mileage	color	condition	pubdate	address	year_auto	text	auto_id	user_id
▶	1	100	1000	"color1"	"normal"	2020-05-24 10:10:10	"address1"	2001	"text1"	1	5
	2	200	2000	"color2"	"bad"	2020-05-24 21:21:22	"address2"	2002	"text2"	1	3
	3	300	3000	"color3"	"good"	2020-05-24 21:21:23	"address3"	2003	"text3"	3	3
	4	400	4000	"color4"	"brilliant"	2020-05-24 21:21:24	"address4"	2004	"text4"	3	1
	5	500	5000	"color5"	"normal"	2020-05-24 21:21:25	"address5"	2005	"text5"	5	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. Виконаємо модифікацію значень у таблиці advertisement.

Змінимо пробіг автомобіля з третього оголошення:

```
UPDATE mydb.advertisement SET mileage = 9999 WHERE id = 3;
```

Результат:

	id	price	mileage	color	condition	pubdate	address	year_auto	text	auto_id	user_id
▶	1	100	1000	"color1"	"normal"	2020-05-24 10:10:10	"address1"	2001	"text1"	1	5
	2	200	2000	"color2"	"bad"	2020-05-24 21:21:22	"address2"	2002	"text2"	3	3
	3	300	9999	"color3"	"good"	2020-05-24 21:21:23	"address3"	2003	"text3"	3	3
	4	400	4000	"color4"	"brilliant"	2020-05-24 21:21:24	"address4"	2004	"text4"	3	1
	5	500	5000	"color5"	"normal"	2020-05-24 21:21:25	"address5"	2005	"text5"	5	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

А тепер змінимо всі ціни оголошень, встановивши для них залежність від пробігу “ціна = пробіг + 1000”:

```
UPDATE mydb.advertisement SET price = mileage + 1000 WHERE id >= 0;
```

Результат:

	id	price	mileage	color	condition	pubdate	address	year_auto	text	auto_id	user_id
▶	1	2000	1000	"color1"	"normal"	2020-05-24 10:10:10	"address1"	2001	"text1"	1	5
	2	3000	2000	"color2"	"bad"	2020-05-24 21:21:22	"address2"	2002	"text2"	1	3
	3	10999	9999	"color3"	"good"	2020-05-24 21:21:23	"address3"	2003	"text3"	3	3
	4	5000	4000	"color4"	"brilliant"	2020-05-24 21:21:24	"address4"	2004	"text4"	3	1
	5	6000	5000	"color5"	"normal"	2020-05-24 21:21:25	"address5"	2005	"text5"	5	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. Видалимо записи з таблиці mark:

```
DELETE FROM mydb.mark WHERE id > 4 OR id=2;
```

Результат:

Цим запитом я видалив п'яту і другу марку машини.

	id	name
▶	1	"BMW"
	3	"Audi"
	4	"Renault"
*	NULL	NULL

Висновок: у цій лабораторній роботі було розглянуто способи наповнення і модифікації даних в таблицях БД та проведено модифікацію даних у двох таблицях.