

Лабораторна робота №4

“Запити на додавання, зміну та вилучення даних”

з дисципліни

«Організація баз даних та знань»

Виконала:

студентка групи КН-208

Бокшо Каріна

Прийняла:

Мельникова Н.І.

Мета роботи: 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

1. Для внесення значень в таблиці використовується директива INSERT.

```
CREATE TABLE if not exists subject (  
    id INTEGER AUTO_INCREMENT PRIMARY KEY not null,  
    name VARCHAR(30) not null,  
    level INTEGER not null  
);  
  
CREATE TABLE if not exists term (  
    id INTEGER AUTO_INCREMENT PRIMARY KEY not null,  
    day enum('mon', 'tue', 'wed', 'thu', 'fri') not null,  
    time integer not null  
);  
  
CREATE TABLE if not exists teacher (  
    id INTEGER AUTO_INCREMENT PRIMARY KEY not null,  
    firstName varchar(15) not null,  
    lastName varchar(15) not null,  
    id_category integer  
);  
  
CREATE TABLE if not exists category (  
    id INTEGER AUTO_INCREMENT PRIMARY KEY not null,  
    name varchar(15) not null  
);  
  
CREATE TABLE if not exists class (  
    id INTEGER AUTO_INCREMENT PRIMARY KEY not null,  
    category varchar(15) not null  
);  
  
  
CREATE TABLE if not exists student_group (  
    id INTEGER AUTO_INCREMENT PRIMARY KEY not null,  
    name varchar(10) not null  
);  
  
CREATE TABLE if not exists student (  
    id INTEGER AUTO_INCREMENT PRIMARY KEY not null,  
    id_group integer not null,  
    name varchar(15) not null,  
    surname varchar(15)  
);  
  
CREATE TABLE if not exists schedule (  
    id INTEGER AUTO_INCREMENT PRIMARY KEY not null,  
    id_teacher INTEGER not null,  
    id_group integer not null,  
    id_subject integer not null,  
    id_room integer not null,  
    id_term integer not null  
);  
  
alter table schedule add constraint sch_sub_fk foreign key (id_subject) references subject(id);  
alter table schedule add constraint sch_term_fk foreign key (id_term) references term(id);
```

```
alter table schedule add constraint sch_class_fk foreign key (id_room) references class(id);
alter table schedule add constraint sch_techer_fk foreign key (id_teacher) references teacher(id);
alter table schedule add constraint sch_group_fk foreign key (id_group) references
student_group(id);
alter table student add constraint stdgr_stud_fk foreign key (id_group) references
student_group(id);
alter table teacher add constraint categ_tech_fk foreign key (id_category) references category(id);
```

```
insert into category(id, name) values(1, 'methodist');
insert into category(id, name) values(2, 'first');
insert into category(id, name) values(3, 'second');
```

```
insert into teacher (firstName, lastName, id_category) values ('Ivan', 'Bondarenko', 1);
insert into teacher (firstName, lastName, id_category) values ('Natalia', 'Kovalyk', 1);
insert into teacher (firstName, lastName, id_category) values ('Nelia', 'Boiko', 2);
insert into teacher (firstName, lastName, id_category) values ('Iryna', 'Oleksyn', 3);
insert into teacher (firstName, lastName, id_category) values ('Mark', 'Livin', 2);
```

```
insert into student_group(id, name) values (1, 'E-1');
insert into student_group(id, name) values (2, 'E-2');
insert into student_group(id, name) values (3, 'E-3');
insert into student_group(id, name) values (4, 'PLG');
insert into student_group(id, name) values (5, 'LPLG');
```

```
insert into student(name, surname, id_group) values('Katia', 'Babkina', 1);
insert into student(name, surname, id_group) values('Solomia', 'Kulishko', 1);
insert into student(name, surname, id_group) values('Sophia', 'Chyzh', 2);
insert into student(name, surname, id_group) values('Julia', 'Opoka', 3);
insert into student(name, surname, id_group) values('Ostap', 'Samahala', 2);
```

```
insert into class(id, category) values (1, 'math');
insert into class(id, category) values (2, 'chemistry');
insert into class(id, category) values (3, 'history');
insert into class(id, category) values (4, 'physics');
insert into class(id, category) values (5, 'sport');
```

```
insert into term(day, time) values('mon', 1);
insert into term(day, time) values('tue', 2);
insert into term(day, time) values('wed', 3);
insert into term(day, time) values('wed', 4);
insert into term(day, time) values('fri', 5);
```

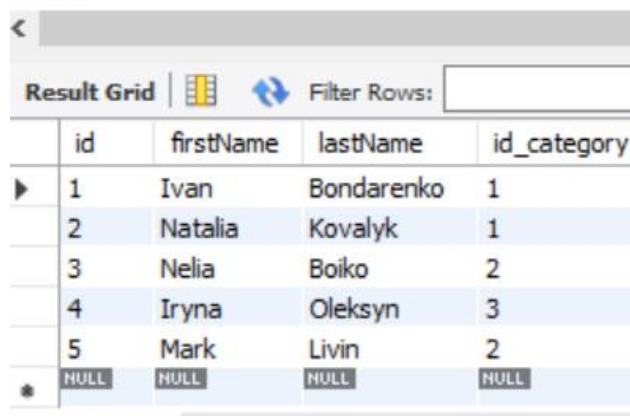
```
insert into subject(id, name, level) value(1, 'literature', 9);
insert into subject(id, name, level) value(2, 'history', 7);
insert into subject(id, name, level) value(3, 'chemistry', 8);
insert into subject(id, name, level) value(4, 'math', 6);
insert into subject(id, name, level) value(5, 'physics', 8);
```

```
insert into schedule(id_teacher, id_group, id_subject, id_room, id_term) values(1, 1, 1, 1, 1);
insert into schedule(id_teacher, id_group, id_subject, id_room, id_term) values(2, 2, 2, 2, 2);
insert into schedule(id_teacher, id_group, id_subject, id_room, id_term) values(3, 3, 3, 3, 3);
insert into schedule(id_teacher, id_group, id_subject, id_room, id_term) values(4, 4, 4, 4, 4);
```

insert into schedule(id_teacher, id_group, id_subject, id_room, id_term) values(5, 5, 5, 5, 5);

```
101 • select * from teacher;
```

102



The screenshot shows a database query result grid. The grid has a header row with columns: id, firstName, lastName, and id_category. Below the header, there are five data rows. The first row has id 1, firstName Ivan, lastName Bondarenko, and id_category 1. The second row has id 2, firstName Natalia, lastName Kovalyk, and id_category 1. The third row has id 3, firstName Nelia, lastName Boiko, and id_category 2. The fourth row has id 4, firstName Iryna, lastName Oleksyn, and id_category 3. The fifth row has id 5, firstName Mark, lastName Livin, and id_category 2. At the bottom of the grid, there is a row with NULL values for all columns.

| id | firstName | lastName | id_category |
|------|-----------|------------|-------------|
| 1 | Ivan | Bondarenko | 1 |
| 2 | Natalia | Kovalyk | 1 |
| 3 | Nelia | Boiko | 2 |
| 4 | Iryna | Oleksyn | 3 |
| 5 | Mark | Livin | 2 |
| NULL | NULL | NULL | NULL |

2. Заповнення бази даних з файлу:

```
1 • use lk;
```

2

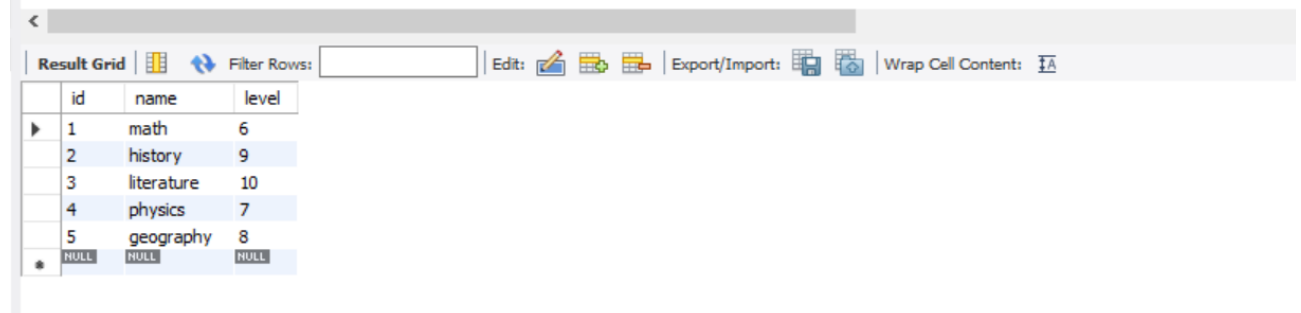
```
3 • load data infile 'C:\\Users\\ACER\\Desktop\\worker.txt' into table subject FIELDS TERMINATED by ","
```

```
4 LINES TERMINATED BY '\\n'
```

```
5 (name,level);
```

6

```
7 • SELECT * FROM subject;
```



The screenshot shows a database query result grid. The grid has a header row with columns: id, name, and level. Below the header, there are five data rows. The first row has id 1, name math, and level 6. The second row has id 2, name history, and level 9. The third row has id 3, name literature, and level 10. The fourth row has id 4, name physics, and level 7. The fifth row has id 5, name geography, and level 8. At the bottom of the grid, there is a row with NULL values for all columns.

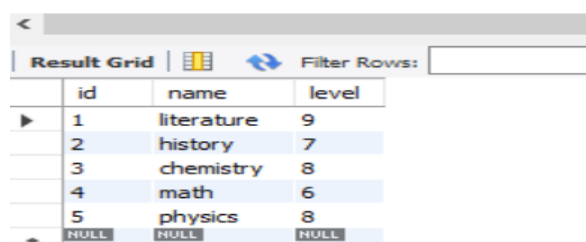
| id | name | level |
|------|------------|-------|
| 1 | math | 6 |
| 2 | history | 9 |
| 3 | literature | 10 |
| 4 | physics | 7 |
| 5 | geography | 8 |
| NULL | NULL | NULL |

3.Для внесення змін в записи таблиці використовується директива UPDATE.

```
1 • use dl33;
```

```
2 • describe subject;
```

```
3 • select * from subject;
```



The screenshot shows a database query result grid. The grid has a header row with columns: id, name, and level. Below the header, there are five data rows. The first row has id 1, name literature, and level 9. The second row has id 2, name history, and level 7. The third row has id 3, name chemistry, and level 8. The fourth row has id 4, name math, and level 6. The fifth row has id 5, name physics, and level 8. At the bottom of the grid, there is a row with NULL values for all columns.

| id | name | level |
|------|------------|-------|
| 1 | literature | 9 |
| 2 | history | 7 |
| 3 | chemistry | 8 |
| 4 | math | 6 |
| 5 | physics | 8 |
| NULL | NULL | NULL |

Рис1.До внесення змін

```

16 • UPDATE subject SET level = '11'
17   WHERE name='literature';

```

Result Grid

| | id | name | level |
|---|------|------------|-------|
| ▶ | 1 | literature | 11 |
| | 2 | history | 7 |
| | 3 | chemistry | 8 |
| | 4 | math | 6 |
| | 5 | physics | 8 |
| * | NULL | NULL | NULL |

subject 4 x

Output

Рис1.Після внесення змін

4.Для видалення записів з таблиці використовується директива DELETE

```

1 • use dl33;
2 • describe subject;
3 • select * from subject;

```

Result Grid

| | id | name | level |
|---|------|------------|-------|
| ▶ | 1 | literature | 9 |
| | 2 | history | 7 |
| | 3 | chemistry | 8 |
| | 4 | math | 6 |
| | 5 | physics | 8 |
| | NULL | NULL | NULL |

Рис3.До внесення змін

```

19 • delete from subject
20   where(level<'8');
21
22 • select * from subject;
23

```

Result Grid

| | id | name | level |
|---|----|------------|-------|
| ▶ | 1 | literature | 9 |
| | 3 | chemistry | 8 |
| | 5 | physics | 8 |

Рис4.Після внесення змін

Висновок: на цій лабораторній роботі було розглянуто SQL-запити для внесення нових значень в таблиці в режимі
одиночного та групового доповнення; розробити SQL-запити для внесення змін в рядки таблиць; розробити SQL-запити для вилучення вибраних рядків