

a.

The screenshot shows the SQL Studio interface. The top menu bar includes 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons. The main editor area contains two lines of SQL code:

```
1 • use education;  
2 • SELECT * FROM university;
```

The bottom pane displays the 'Result Grid' with the following data:

univID	nameUniv
1	ЛУ ім. Ів. Франка
2	НЛТУУ
3	УКУ
4	НУ ЛП
5	ЛьвДУВС
6	ЛНМУ ім. Д.Галицького
NULL	NULL

On the right side of the bottom pane, there are buttons for 'Result Grid', 'Form Editor', and 'Field Types'. The status bar at the bottom shows 'university 1' and buttons for 'Apply' and 'Revert'.

b.

The screenshot shows the SQL Studio interface. The top menu bar includes 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons. The main editor area contains four lines of SQL code:

```
1 • use education;  
2 • SELECT * FROM university;  
3  
4 • SELECT firstName, secondName from students
```

The bottom pane displays the 'Result Grid' with the following data:

firstName	secondName
Брик	Юля
Сухар	Степан
Бала	Оля
Горішній	Володимир
Кравець	Данило
Лаба	Ірина
Ярема	Софія
Берлін	Марко
Головка	Ірина
Сидоренко	Злата

On the right side of the bottom pane, there are buttons for 'Result Grid', 'Form Editor', and 'Field Types'. The status bar at the bottom shows 'students 2' and a 'Read Only' indicator.

C.

```
1 • use education;
2 • SELECT * FROM university;
3
4 • SELECT firstName, secondName from students;
5
6 • SELECT firstName from teachers where firstName = 'Иван'
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

firstName
Иван
Иван

teachers 4 x | Read Only

d.

```
3
4 • SELECT firstName, secondName from students;
5
6 • SELECT firstName from teachers where firstName = 'Иван';
7
8 • SELECT nameGroup, numberGroup from groupsSt where numberGroup > 300;
9
```

Limit to 1000 rows |

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

nameGroup	numberGroup
Мед	301
Фарм	302
Ек	502
Юр	402
Ф	501

groupsSt 5 x | Read Only

e.

The screenshot shows a database management tool interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, a SQL editor contains three queries:

```
5  
6 • SELECT firstName from teachers where firstName = 'Іван';  
7  
8 • SELECT nameGroup, numberGroup from groupsSt where numberGroup > 300;  
9  
10 • SELECT nameUniv from university where nameUniv like '%a%';  
11
```

Below the SQL editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid itself displays a table with the following data:



nameUniv
ЛУ ім. Ів. Франка
ЛНМУ ім. Д.Галицького

On the right side of the interface, there is a vertical toolbar with buttons for "Result Grid", "Form Editor", and "Field Types". At the bottom left, a tab labeled "university 6" is visible. At the bottom right, there is a "Read Only" status indicator.

```

1 • CREATE DATABASE education;
2
3 • create table education.university (
4     univID int auto_increment primary key,
5     nameUniv varchar (60) not null
6 );
7
8 • insert into education.university (nameUniv)
9     values
10     ("ЛУ ім. Ів. Франка"),
11     ("НЛТУУ"),
12     ("УКУ"),
13     ("НУ ЛП"),
14     ("ЛьвДУВС"),
15     ("ЛНМУ ім. Д.Галицького");
16
17 • select*from education.university;
18

```

Result Grid   Filter Rows:

	univID	nameUniv
▶	1	ЛУ ім. Ів. Франка
	2	НЛТУУ
	3	УКУ
	4	НУ ЛП
	5	ЛьвДУВС
	6	ЛНМУ ім. Д.Галицького
•	NULL	NULL

university 13 ×

Query Database Server Tools Scripting Help

Limit to 1000 rows

```
19 • create table education.faculties (  
20     facultyID int auto_increment primary key,  
21     nameFaculty varchar (60) not null  
22 );  
23  
24 • insert into education.faculties (namefaculty)  
25 values  
26     ("медичний"),  
27     ("фармацевтичний"),  
28     ("економічний"),  
29     ("юридичний"),  
30     ("архітектурний"),  
31     ("філологічний");  
32  
33 • create table education.UnivFacul (  
34     UnivFacultyID int auto_increment primary key,  
35     univID int,  
36     facultyID int,  
37     foreign key (univID)
```

Output

Action Output

#	Time	Action	Message
---	------	--------	---------

Result Grid | Filter Rows:

	facultyID	nameFaculty
▶	1	медичний
	2	фармацевтичний
	3	економічний
	4	юридичний
	5	архітектурний
	6	філологічний
✱	NULL	NULL

faculties 14 x

```

32
33 • create table education.UnivFacul (
34     UnivFacultyID int auto_increment primary key,
35     univID int,
36     facultyID int,
37     foreign key (univID)
38     references education.university (univID),
39     foreign key (facultyID)
40     references education.faculties (facultyID)
41 );
42 • select*from education.UnivFacul;
43
44 • insert into education.UnivFacul (univID,facultyID)
45 values
46 (1, 3),
47 (1, 6),
48 (2, 3),
49 (2, 5),

```

Result Grid | Filter Rows: Edit:

	UnivFacultyID	univID	facultyID
▶	1	1	3
	2	1	6
	3	2	3
	4	2	5
	5	3	3
	6	3	6
	7	4	3
	8	4	6
	9	5	3
	10	5	4
	11	6	2
	12	6	1
•	NULL	NULL	NULL

UnivFacul 15 ×

```
Limit to 1000 rows
1 • create table education.teachers (
2   teacherID int auto_increment primary key,
3   firstName varchar (60) not null,
4   secondName varchar (60) not null,
5   univID int,
6   foreign key (univID)
7   references education.university (univID)
8 );
9
10 • insert into education.teachers (firstName, secondName,univID )
11 values
12 ("Вербицька", "Поліна", 2),
13 ("Мельник", "Ростислав", 6),
14 ("Хома", "Іван", 3),
15 ("Савченко", "Ольга", 1),
16 ("Кравець", "Данило", 2),
17 ("Салюк", "Андрій", 5),
18 ("Петушинська", "Олександра", 4),
19 ("Бруфен", "Іван", 3);
20
```

```
80
81 • select * from education.teachers;
82
```

	teacherID	firstName	secondName	univID
	1	Вербицька	Поліна	2
▶	2	Мельник	Ростислав	6
	3	Хома	Іван	3
	4	Савченко	Ольга	1
	5	Кравець	Данило	2
	6	Салюк	Андрій	5
	7	Петушинська	Олександра	4
	8	Бруфен	Іван	3
•	NULL	NULL	NULL	NULL

teachers 17 ×

```

• create table education.students (
  studentID int auto_increment primary key,
  firstName varchar (60) not null,
  secondName varchar (60) not null
);

• insert into education.students (firstName,secondName)
  values
    ("Брик", "Юля"),
    ("Сухар", "Степан"),
    ("Бала", "Оля"),
    ("Горішній", "Володимир"),
    ("Кравець", "Данило"),
    ("Лаба", "Ірина"),
    ("Ярема", "Софія"),
    ("Берлін", "Марко"),
    ("Головко", "Ірина"),
    ("Сидоренко", "Злата");

• select*from education.students;

```

Result Grid			
Filter Rows: <input type="text"/>			
Edi			
	studentID	firstName	secondName
▶	1	Брик	Юля
	2	Сухар	Степан
	3	Бала	Оля
	4	Горішній	Володимир
	5	Кравець	Данило
	6	Лаба	Ірина
	7	Ярема	Софія
	8	Берлін	Марко
	9	Головко	Ірина
	10	Сидоренко	Злата
•	NULL	NULL	NULL

students 18 x


```

102
103
104 • create table education.groupsSt (
105     groupID int auto_increment primary key,
106     nameGroup varchar (60) not null
107 );
108
109 • insert into education.groupsSt (nameGroup)
110 values
111     ("Мед-101"),
112     ("Мед-301"),
113     ("Ст-302"),
114     ("Ек-201"),
115     ("Ек-502"),
116     ("Ф-102"),
117     ("Юр-402"),
118     ("Арх-102"),
119     ("Ф-102"),
120     ("Ф-501");
121
122

```

	groupID	nameGroup
▶	1	Мед-101
	2	Мед-301
	3	Ст-302
	4	Ек-201
	5	Ек-502
	6	Ф-102
	7	Юр-402
	8	Арх-102
	9	Ф-102
	10	Ф-501
●	NULL	NULL

```
125 • create table education.informationOfStudents(  
126     inforID int auto_increment primary key,  
127     studentID int,  
128     groupID int,  
129     UnivFacultyID int,  
130     foreign key (studentID)  
131     references education.students (studentID),  
132     foreign key (groupID)  
133     references education.groupsSt (groupID),  
134     foreign key (UnivFacultyID)  
135     references education.UnivFacul (UnivFacultyID)  
136 );  
137  
138 • insert into education.informationOfStudents (studentID,groupID,UnivFacultyID )  
139 values  
140 (10, 4, 7 ),  
141 (9, 1, 11 ),  
142 (8, 3, 12 ),  
143 (7, 5, 5 ),  
144 (6, 6, 2 ),  
145 (5, 7, 10 ),  
146 (4, 8, 4 ),  
147 (3, 2, 11 ),  
148 (2, 9, 6 );
```

SQL File 1* SQL File 3* SQL File 4* x teachers

Limit to 1000 rows

```
1 • select count(workExperience), workExperience from teachers group by workExperience;
2
3 • select count(salary), salary from teachers group by salary having salary > 9000;
4
5 • SELECT * FROM education.teachers order by secondName;
6
7 • SELECT * FROM education.teachers order by secondName DESC;
8
9 • SELECT * FROM education.teachers order by salary DESC limit 3;
10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	count(workExperience)	workExperience
▶	2	5
	2	3
	1	10
	1	4
	1	2
	1	9

Result 8 x Read Only

Output

SQL File 1* SQL File 3* SQL File 4* x teachers

Limit to 1000 rows

```
1 • select count(workExperience), workExperience from teachers group by workExperience;
2   Execute the selected portion of the script or everything, if there is no selection
3 • select count(salary), salary from teachers group by salary having salary > 9000;
4
5 • SELECT * FROM education.teachers order by secondName;
6
7 • SELECT * FROM education.teachers order by secondName DESC;
8
9 • SELECT * FROM education.teachers order by salary DESC limit 3;
10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	count(salary)	salary
▶	1	15000
	1	10000
	1	9200
	1	12000

Result 9 x Read Only

SQL File 1 SQL File 3 SQL File 4 teachers

Limit to 1000 rows

```

1 • select count(workExperience), workExperience from teachers group by workExperience;
2
3 • select count(salary), salary from teachers group by salary having salary > 9000;
4
5 • SELECT * FROM education.teachers order by secondName;
6
7 • SELECT * FROM education.teachers order by secondName DESC;
8
9 • SELECT * FROM education.teachers order by salary DESC limit 3;
10

```

Result Grid

	teacherID	firstName	secondName	univID	salary	workExperience
▶	8	Іван	Бруфен	3	12000	9
	1	Полина	Вербицька	2	7000	5
	5	Данило	Кравець	2	9200	4
	2	Ростислав	Мельник	6	8500	3
	7	Олександра	Петушинська	4	7300	3
	4	Ольга	Савченко	1	10000	5
	6	Андрій	Салюк	5	7000	2
	3	Іван	Хома	3	15000	10

teachers 10 x

Apply

Output

Action Output

SQL File 1 SQL File 3 SQL File 4 teachers

Limit to 1000 rows

```

1 • select count(*) from education.groupsst;
2
3 • select count(firstName) from education.teachers where firstName like '%1%';
4
5 • select *, min(numberGroup) from education.groupsst;
6
7 • select *, max(nameUniv) from university;
8
9 • select avg(numberGroup) as AVGnumber from groupsst;

```

Result Grid

	count(*)
▶	10

Result 12 x

Read Only

Output

SQL File 1* SQL File 3* SQL File 4* teachers

Limit to 1000 rows

```
1 • select count(*) from education.groupsst;
2
3 • select count(firstName) from education.teachers where firstName like "%i%";
4
5 • select *, min(numberGroup) from education.groupsst;
6
7 • select *, max(nameUniv) from university;
8
9 • select avg(numberGroup) as AVGnumber from groupsst;
```

Result Grid

count(firstName)
4

Filter Rows: Export: Wrap Cell Content:

Result 13 x Read Only

Output

SQL File 1* SQL File 3* SQL File 4* teachers

Limit to 1000 rows

```
3 • select count(firstName) from education.teachers where firstName like "%i%";
4
5 • select *, min(numberGroup) from education.groupsst;
6
7 • select *, max(nameUniv) from university;
8
9 • select avg(numberGroup) as AVGnumber from groupsst;
10
11 • select sum(numberGroup) from groupsst where numberGroup < 300;
```

Result Grid

sum(numberGroup)
708

Filter Rows: Export: Wrap Cell Content:

Result 14 x Read Only

Output

SQL File 1* SQL File 3* SQL File 4* teachers

Limit to 1000 rows

```
1 • select count(*) from education.groupsst;
2
3 • select count(firstName) from education.teachers where firstName like "%1%";
4
5 • select *, min(numberGroup) from education.groupsst;
6
7 • select *, max(nameUniv) from university;
8
9 • select avg(numberGroup) as AVGnumber from groupsst;
```

Result Grid

AVGnumber
271.6

Result 15 x Read Only

SQL File 1* SQL File 3* SQL File 4* teachers

Limit to 1000 rows

```
1 • select count(*) from education.groupsst;
2
3 • select count(firstName) from education.teachers where firstName like "%1%";
4
5 • select *, min(numberGroup) from education.groupsst;
6
7 • select *, max(nameUniv) from university;
8
9 • select avg(numberGroup) as AVGnumber from groupsst;
```

Result Grid

groupID	nameGroup	numberGroup	min(numberGroup)
1	Мед	101	101

Result 16 x Read Only

Output

Server Tools Scripting Help

SQL File 1* SQL File 3* SQL File 4* teachers SQL File 5* x students

Limit to 1000 rows

```

1 select * from students join groupsst on students.groupID = groupsst.groupID;
2
3 • select * from students inner join groupsst on students.groupID = groupsst.groupID;
4
5 • select * from students left join groupsst on students.groupID = groupsst.groupID;
6
7 • select * from students right join groupsst on students.groupID = groupsst.groupID;
8

```

Result Grid Filter Rows: Export: Wrap Cell Content:

	studentID	firstName	secondName	groupID	groupID	nameGroup	numberGroup
▶	1	Брик	Юля	10	10	Ф	501
	2	Сухар	Степан	9	9	Ф	202
	3	Бала	Оля	2	2	Мед	301
	4	Горішній	Володимир	8	8	Арх	102
	5	Кравець	Данило	7	7	Юр	402
	6	Лаба	Ірина	6	6	Ф	102
	7	Ярема	Софія	5	5	Ек	502
	8	Берлін	Марко	3	3	Фарм	302
	9	Головко	Ірина	4	4	Ек	201
	10	Сидоренко	Злата	1	1	Мед	101

Result 16 x Read Only

SQL File 1* SQL File 3* SQL File 4* teachers SQL File 5* x students

Limit to 1000 rows

```

1 select * from students join groupsst on students.groupID = groupsst.groupID;
2
3 • select * from students inner join groupsst on students.groupID = groupsst.groupID;
4
5 • select * from students left join groupsst on students.groupID = groupsst.groupID;
6
7 • select * from students right join groupsst on students.groupID = groupsst.groupID;
8

```

Result Grid Filter Rows: Export: Wrap Cell Content:

	studentID	firstName	secondName	groupID	groupID	nameGroup	numberGroup
▶	1	Брик	Юля	10	10	Ф	501
	2	Сухар	Степан	9	9	Ф	202
	3	Бала	Оля	2	2	Мед	301
	4	Горішній	Володимир	8	8	Арх	102
	5	Кравець	Данило	7	7	Юр	402
	6	Лаба	Ірина	6	6	Ф	102
	7	Ярема	Софія	5	5	Ек	502
	8	Берлін	Марко	3	3	Фарм	302
	9	Головко	Ірина	4	4	Ек	201
	10	Сидоренко	Злата	1	1	Мед	101

Result 17 x Read Only

Limit to 1000 rows

```

1 select * from students join groupsst on students.groupID = groupsst.groupID;
2
3 • select * from students inner join groupsst on students.groupID = groupsst.groupID;
4
5 • select * from students left join groupsst on students.groupID = groupsst.groupID;
6
7 • select * from students right join groupsst on students.groupID = groupsst.groupID;
8

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	studentID	firstName	secondName	groupID	groupID	nameGroup	numberGroup
▶	1	Брик	Юля	10	10	Ф	501
	2	Сухар	Степан	9	9	Ф	202
	3	Бала	Оля	2	2	Мед	301
	4	Горішній	Володимир	8	8	Арх	102
	5	Кравець	Данило	7	7	Юр	402
	6	Лаба	Ірина	6	6	Ф	102
	7	Ярема	Софія	5	5	Ек	502
	8	Берлін	Марко	3	3	Фарм	302
	9	Головко	Ірина	4	4	Ек	201
	10	Сидоренко	Злата	1	1	Мед	101
	11	Іванів	Іван	NULL	NULL	NULL	NULL

Result 20 x Read Only

Limit to 1000 rows

```

1 select * from students join groupsst on students.groupID = groupsst.groupID;
2
3 • select * from students inner join groupsst on students.groupID = groupsst.groupID;
4
5 • select * from students left join groupsst on students.groupID = groupsst.groupID;
6
7 • select * from students right join groupsst on students.groupID = groupsst.groupID;
8

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	studentID	firstName	secondName	groupID	groupID	nameGroup	numberGroup
▶	10	Сидоренко	Злата	1	1	Мед	101
	3	Бала	Оля	2	2	Мед	301
	8	Берлін	Марко	3	3	Фарм	302
	9	Головко	Ірина	4	4	Ек	201
	7	Ярема	Софія	5	5	Ек	502
	6	Лаба	Ірина	6	6	Ф	102
	5	Кравець	Данило	7	7	Юр	402
	4	Горішній	Володимир	8	8	Арх	102
	2	Сухар	Степан	9	9	Ф	202
	1	Брик	Юля	10	10	Ф	501
	NULL	NULL	NULL	NULL	11	Юр	502

Result 21 x Read Only