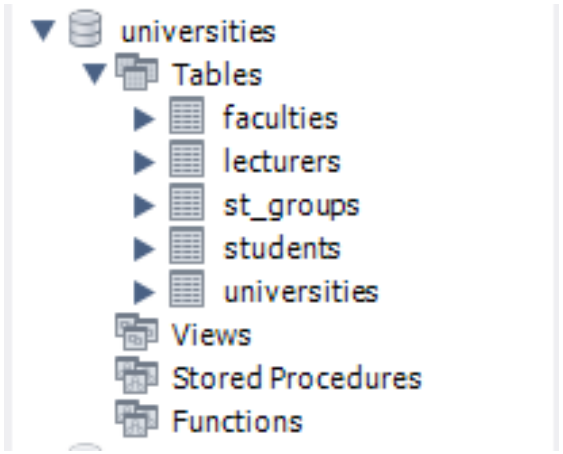
1. In the databases, create the tables Universities, Faculties, Teachers, Students, Groups, and links between them and fill them with data.



university_id	name_of_university	Accreditation
1	V. N. Karazin Kharkiv National University	IV level
2	Kyiv National University of Culture and Arts	IV level
NULL	NULL	NULL

universities 17 | Faculties 18 | St_groups 19 | Lecturers 20 | Students 21

faculty_id	name_of_faculty	university_Id
1	Faculty of Economics	1
2	Faculty of Computer Sciences	1
3	Faculty of hotel and restaurant business	2
4	Faculty of Film and Stage Theatre	2
NULL	NULL	NULL

universities 17 | Faculties 18 | St_groups 19 | Lecturers 20 | Students 21

st_group_id	name_of_group	faculty_Id	university_Id
1	MANAGE11	1	1
2	IT22	2	1
3	REST33	3	2
4	ACTOR44	4	2
NULL	NULL	NULL	NULL

universities 17 | Faculties 18 | St_groups 19 | Lecturers 20 | Students 21

Lecturer_id	name_of_Lecturer	last_name	university_Id	st_group_id
1	Levi	Ackerman	2	3
2	Hange	Zoë	1	2
3	Erwin	Smith	2	4
4	Dot	Pixis	1	1
NULL	NULL	NULL	NULL	NULL

universities 17 | Faculties 18 | St_groups 19 | Lecturers 20 | Students 21

Student_id	name_of_Student	last_name	university_Id	faculty_Id	st_group_ic
1	Reiner	Braun	2	4	4
2	Annie	Leonhart	1	2	2
3	Armin	Arlert	1	1	1
4	Bertolt	Hoover	1	2	2
5	Jean	Kirstein	1	1	1
6	Connie	Springer	2	3	3
7	Sasha	Blouse	2	3	3
8	Mikasa	Ackerman	1	2	2
9	Eren	Yeager	1	2	2
10	Marco	Bott	1	1	1
11	Historia	Reiss	2	4	4
12	Ymir	Reiss	2	4	4
13	Hitch	Dreyse	2	3	3
14	Floch	Forster	1	1	1
15	Marlowe	Freudenb...	2	3	3
16	Franz	Kefka	2	4	4
NULL	NULL	NULL	NULL	NULL	NULL

2. Write the selections in the tables:
    a. select all fields from universities,
    b. select only first and last names of students,
    c. select only those teachers whose name is Ivan,
    d. select only those groups whose numbers are greater than 300,
    e. select only those universities that contain the letter U in the name.

a.

```
108 •  SELECT university_id, name_of_university, Accreditation FROM universities.universities;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

university_id	name_of_university	Accreditation
1	V. N. Karazin Kharkiv National University	IV level
2	Kyiv National University of Culture and Arts	IV level
NULL	NULL	NULL

b.

```
110 •  SELECT name_of_Student, last_name FROM universities.students;
```

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

name_of_Student	last_name
Reiner	Braun
Annie	Leonhart
Armin	Arlert
Bertolt	Hoover
Jean	Kirstein
Connie	Springer
Sasha	Blouse
Mikasa	Ackerman
Eren	Yeager
Marco	Bott
Historia	Reiss
Ymir	Reiss
Hitch	Dreyse
Floch	Forster
Marlowe	Freudenb...
Franz	Kefka

c.

```
112 •  SELECT Lecturer_id, name_of_Lecturer, last_name, university_id, st_group_id FROM universities.lecturers WHERE name_of_Lecturer = 'Hange';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Lecturer_id	name_of_Lecturer	last_name	university_id	st_group_id
2	Hange	Zoë	1	2
NULL	NULL	NULL	NULL	NULL

```
112 •  SELECT Lecturer_id, name_of_Lecturer, last_name, university_id, st_group_id FROM universities.lecturers WHERE name_of_Lecturer = 'Ivan';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Lecturer_id	name_of_Lecturer	last_name	university_id	st_group_id
NULL	NULL	NULL	NULL	NULL

d.

```
116 •  SELECT * FROM universities.st_groups WHERE name_of_group > 3;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

st_group_id	name_of_group	faculty_Id	university_Id
5	3355	1	1
NULL	NULL	NULL	NULL

e.

```
114 •  SELECT * FROM universities.universities WHERE name_of_university LIKE '%a%';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Con

university_id	name_of_university	Accreditation
1	V. N. Karazin Kharkiv National University	IV level
2	Kyiv National University of Culture and Arts	IV level
NULL	NULL	NULL

3. Create a request with aggregate functions

```
127 •  SELECT AVG(faculty_id) FROM universities.students;
128
```

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

AVG(faculty_id)
2.5000

```
121 •  SELECT COUNT(name_of_Student) as names_with_r FROM universities.students WHERE name_of_student LIKE "%r%";
```

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

names_with_r
9

```
125 •  SELECT MAX(faculty_id) FROM universities.students
```

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

MAX(faculty_id)
4

```
123 •  SELECT MIN(st_group_id) FROM universities.students
124
125
```

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

MIN(st_group_id)
1

```
129 •  SELECT SUM(university_id) FROM universities.students;
```

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

SUM(university_id)
24

4. Sort the table of students alphabetically and in reverse order.

```
133 •  SELECT * FROM universities.students ORDER BY name_of_Student
```

Result Grid | Filter Rows: | Edit: | Export/Import:

Student_id	name_of_Student	last_name	university_Id	faculty_Id	st_group_id
2	Annie	Leonhart	1	2	2
3	Armin	Arlert	1	1	1
4	Bertolt	Hoover	1	2	2
6	Connie	Springer	2	3	3
9	Eren	Yeager	1	2	2
14	Floch	Forster	1	1	1
16	Franz	Kefka	2	4	4
11	Historia	Reiss	2	4	4
13	Hitch	Dreyse	2	3	3
5	Jean	Kirstein	1	1	1
10	Marco	Bott	1	1	1
15	Marlowe	Freudenb...	2	3	3
8	Mikasa	Ackerman	1	2	2
1	Reiner	Braun	2	4	4
7	Sasha	Blouse	2	3	3
12	Ymir	Reiss	2	4	4
NULL	NULL	NULL	NULL	NULL	NULL

students 8 ×

```
135 •  SELECT * FROM universities.students ORDER BY name_of_Student DESC;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

Student_id	name_of_Student	last_name	university_Id	faculty_Id	st_group_id
12	Ymir	Reiss	2	4	4
7	Sasha	Blouse	2	3	3
1	Reiner	Braun	2	4	4
8	Mikasa	Ackerman	1	2	2
15	Marlowe	Freudenberg	2	3	3
10	Marco	Bott	1	1	1
5	Jean	Kirstein	1	1	1
13	Hitch	Dreyse	2	3	3
11	Historia	Reiss	2	4	4
16	Franz	Kefka	2	4	4
14	Floch	Forster	1	1	1
9	Eren	Yeager	1	2	2
6	Connie	Springer	2	3	3
4	Bertolt	Hoover	1	2	2
3	Armin	Arlert	1	1	1
2	Annie	Leonhart	1	2	2
NULL	NULL	NULL	NULL	NULL	NULL

students 9 ×

```
131 •  SELECT COUNT(st_group_id), st_group_id FROM universities.students group by st_group_id;
```

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

COUNT(st_group_id)	st_group_id
4	1
4	2
4	3
4	4

5.Create a request to join the students and groups tables using join

```
137 •  SELECT * FROM universities.St_groups INNER JOIN universities.students ON universities.St_groups.university_id = universities.students.university_Id;
```

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

st_group_id	name_of_group	faculty_Id	university_Id	Student_id	name_of_Student	last_name	university_Id	faculty_Id	st_group_id
1	MANAGE11	1	1	2	Annie	Leonhart	1	2	2
1	MANAGE11	1	1	3	Armin	Arlert	1	1	1
1	MANAGE11	1	1	4	Bertolt	Hoover	1	2	2
1	MANAGE11	1	1	5	Jean	Kirstein	1	1	1
1	MANAGE11	1	1	8	Mikasa	Ackerman	1	2	2
1	MANAGE11	1	1	9	Eren	Yeager	1	2	2
1	MANAGE11	1	1	10	Marco	Bott	1	1	1
1	MANAGE11	1	1	14	Floch	Forster	1	1	1
2	IT22	2	1	2	Annie	Leonhart	1	2	2
2	IT22	2	1	3	Armin	Arlert	1	1	1
2	IT22	2	1	4	Bertolt	Hoover	1	2	2
2	IT22	2	1	5	Jean	Kirstein	1	1	1
2	IT22	2	1	8	Mikasa	Ackerman	1	2	2
2	IT22	2	1	9	Eren	Yeager	1	2	2
2	IT22	2	1	10	Marco	Bott	1	1	1
2	IT22	2	1	14	Floch	Forster	1	1	1
3	REST33	3	2	1	Reiner	Braun	2	4	4
3	REST33	3	2	6	Connie	Springer	2	3	3
3	REST33	3	2	7	Sasha	Blouse	2	3	3

Result 16 ×