**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**КАФЕДРА МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №2**

**по дисциплине «Базы данных»**

**Тема: Реализация базы данных в СУБД PostgreSQL**

Студент гр. 1303 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Насонов Я. К.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Заславский М. М.

Санкт-Петербург

2023

**Цель работы.**

Создать локальную базу данных PostgreSQL, создать необходимые таблицы, заполнить их данными. После этого написать запросы, с помощью которых выводится необходимая информация из таблиц и добавляется новая.

**Задание.**

Вариант 15

Пусть требуется создать программную систему, предназначенную для работника деканата Такая система должна обеспечивать хранение сведений о группах и студентах, а также о результатах текущей сессии. Таким образом, для каждого студента должны храниться такие данные, как фамилия, имя отчество студента, номер его зачетной книжки, адрес постоянной прописки и адрес, по которому студент проживает, получает или нет стипендию, а также оценки, полученные в текущей сессии, и отметки о сданных зачетах. Сведения о группе – это номер группы, факультет, кафедра, специальность, к которым она относится, год формирования группы.

Работник деканата может вносить в БД следующие изменения:

* Удалить или добавить в базу студента;
* Поменять студенту номер группы, специальность, кафедру, номер зачетки;
* Занести оценки, полученные студентами на экзаменах по каждому предмету;
* По результатам сессии начислить стипендии студентам, не имеющим троек или иногородним студентам, которые имеют не более одной тройки. Право на 50 % повышение стипендии имеют студенты, получившие в сессию не более двух четверок, а на 100 % повышение – студенты, сдавшие сессию на все пятерки.

Работнику деканата могут потребоваться следующие сведения:

* Студенты, обучающиеся на определенной кафедре и не сдавшие хотя бы один экзамен, с указанием группы и предмета, по которому оценка отсутствует или равна 2?
* Средний балл студентов каждой группы указанного факультета?
* Средний балл по каждому предмету?
* Список студентов указанной кафедры, которые по итогам сессии могут получать стипендию?

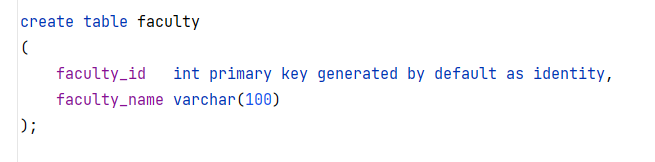
Необходимо развернуть PostgreSQL локально:

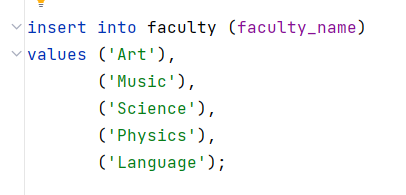
* Написать запросы для создания таблиц из предыдущей лабораторный работы
* Заполнить тестовыми данными: 5-10 строк на каждую таблицу, обязательно наличие связи между ними, данные приближены к реальности.
* Написать запросы к БД, отвечающие на вопросы из предыдущей лабораторной работы
* Исходный код выложить на www.db-fiddle.com для проверки работоспособности
* Исходный код в виде .sql файла запушить в виде PR в репо
* В отчете описать:
  + Цель
  + Текст задания в соответствии с вариантом
  + Скриншоты работы с СУБД PostgreSQL (psql / DBeaver / Datagrip, ...)
  + Скриншоты на каждый запрос (или группу запросов) на изменение/таблицы с выводом результатов (ответ)
  + Исходный код в приложении
  + Ссылку на исходный код www.db-fiddle.com в приложении
  + Ссылка на PR в приложении
  + Вывод

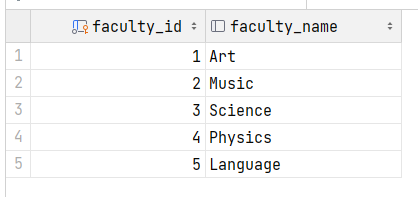
**Выполнение работы.**

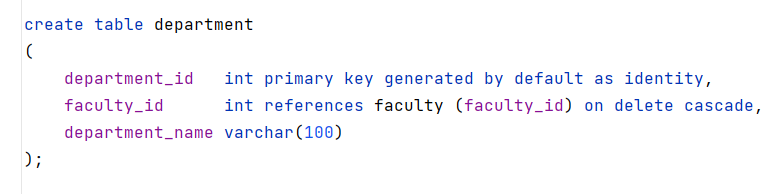
1. **Создание БД:**

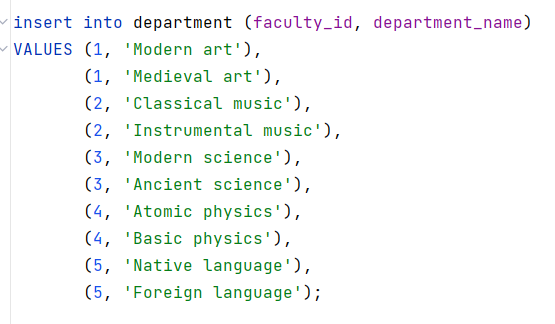
Для создания и заполнения таблиц базы данных написаны соответствующие запросы, также указаны в приложении Б:

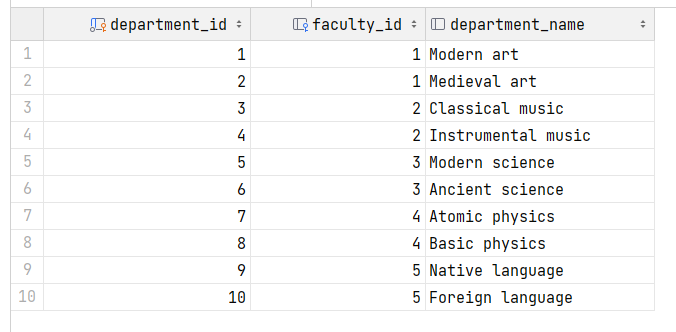


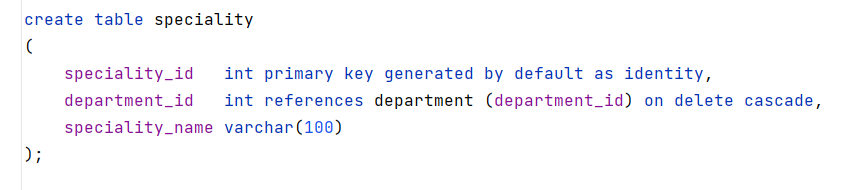


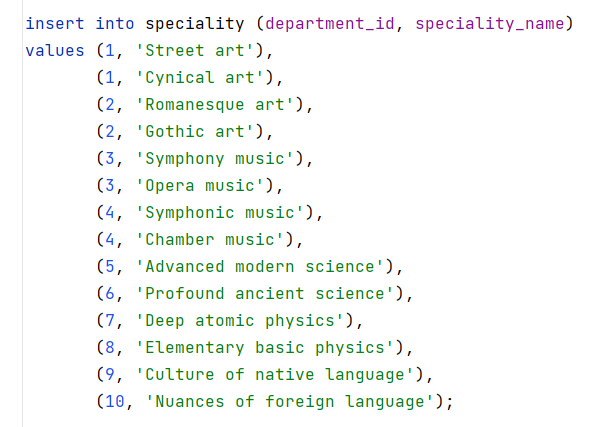


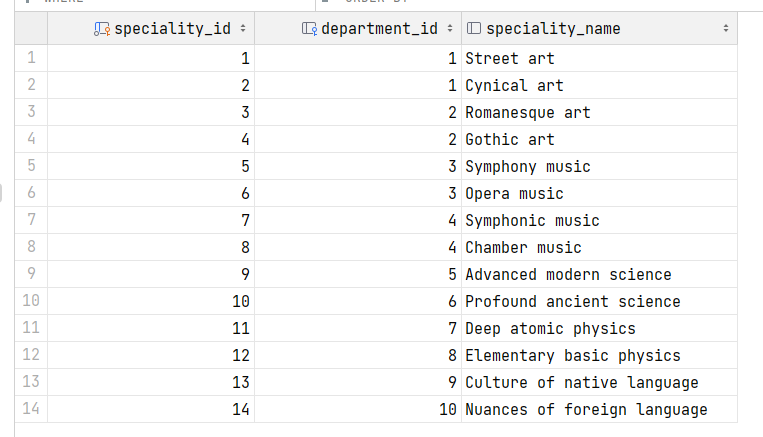


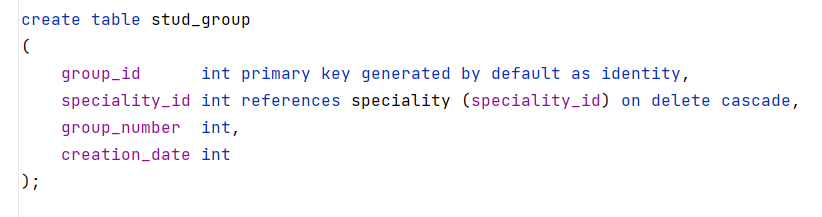


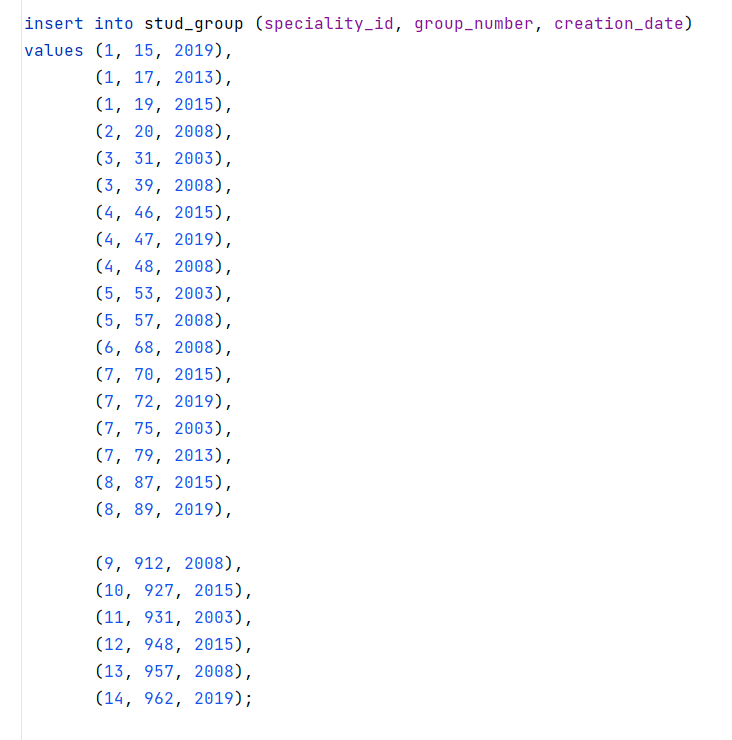


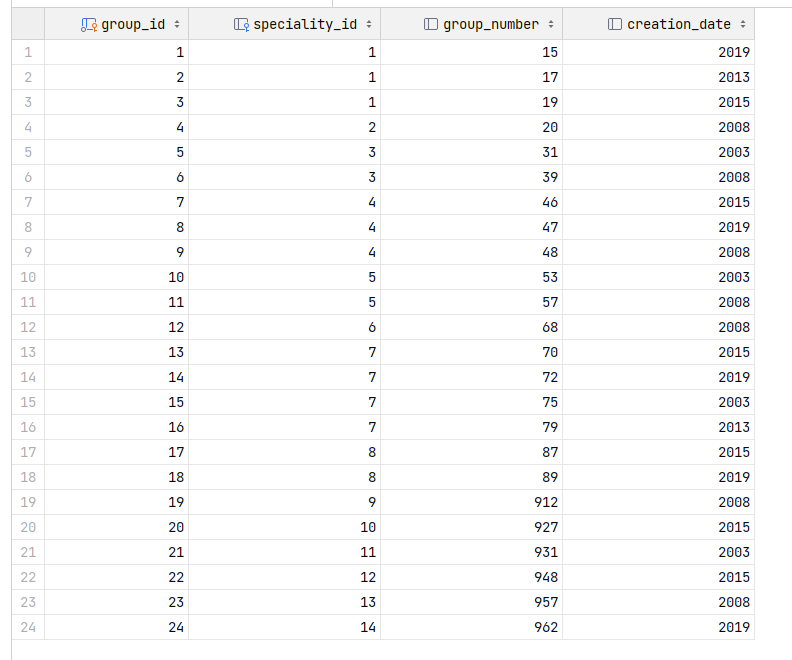


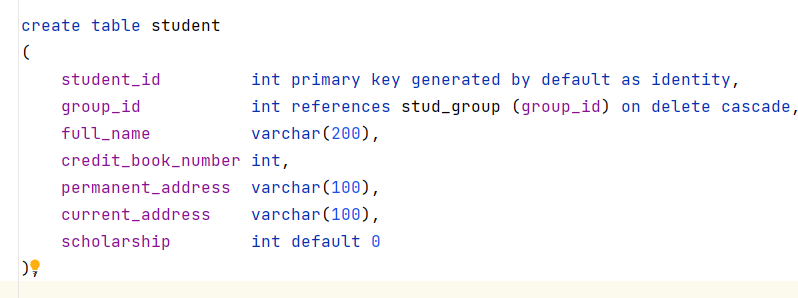


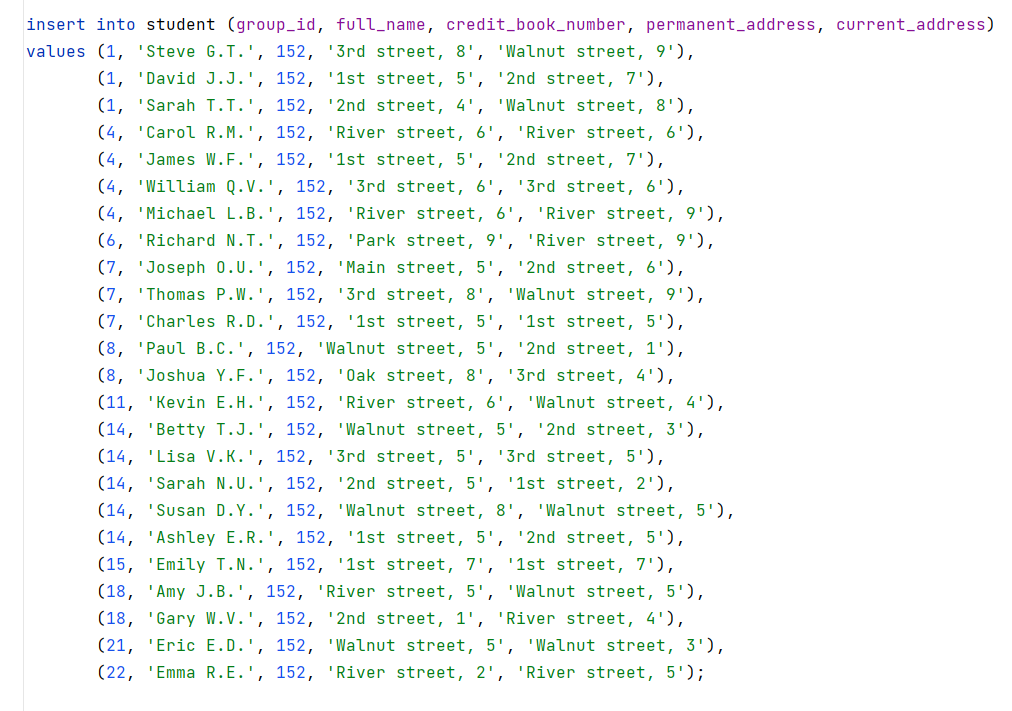


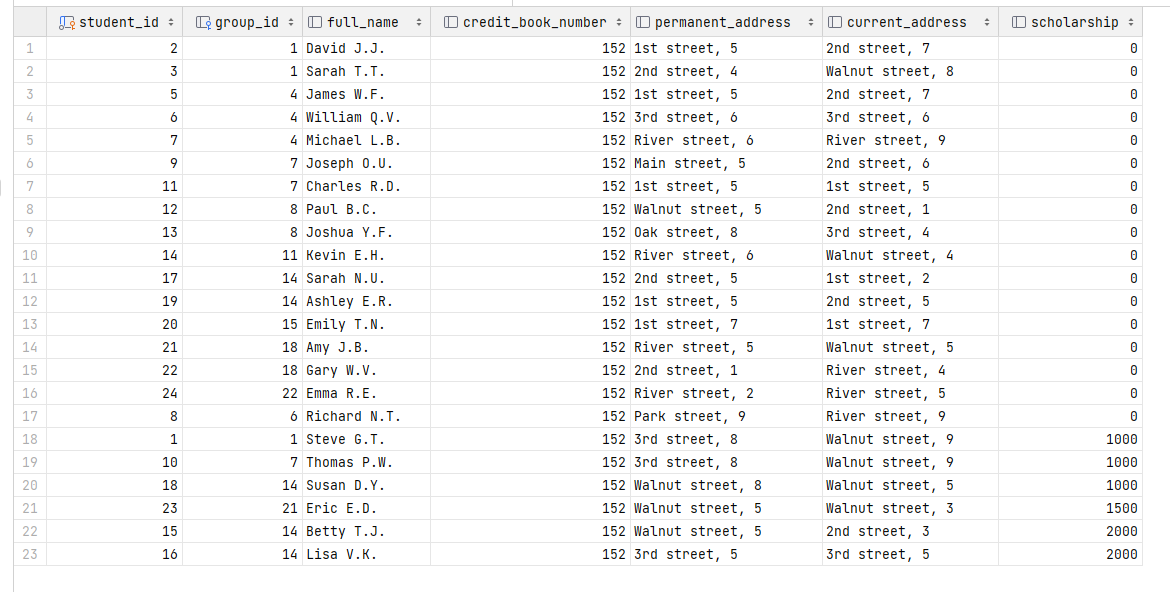


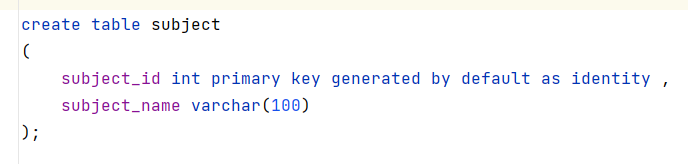


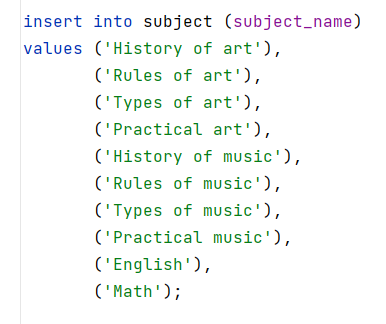


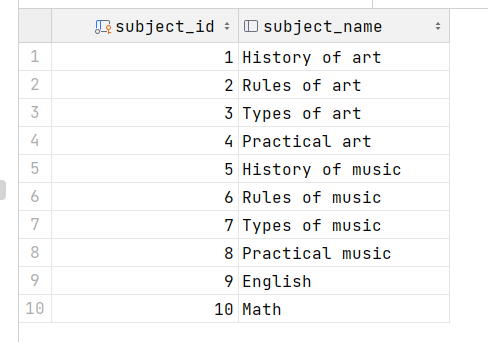


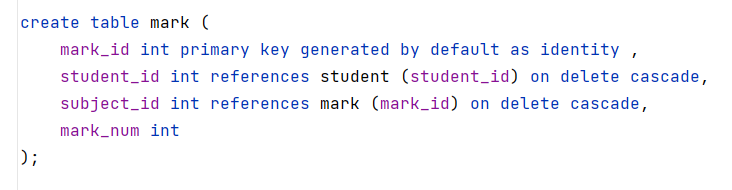


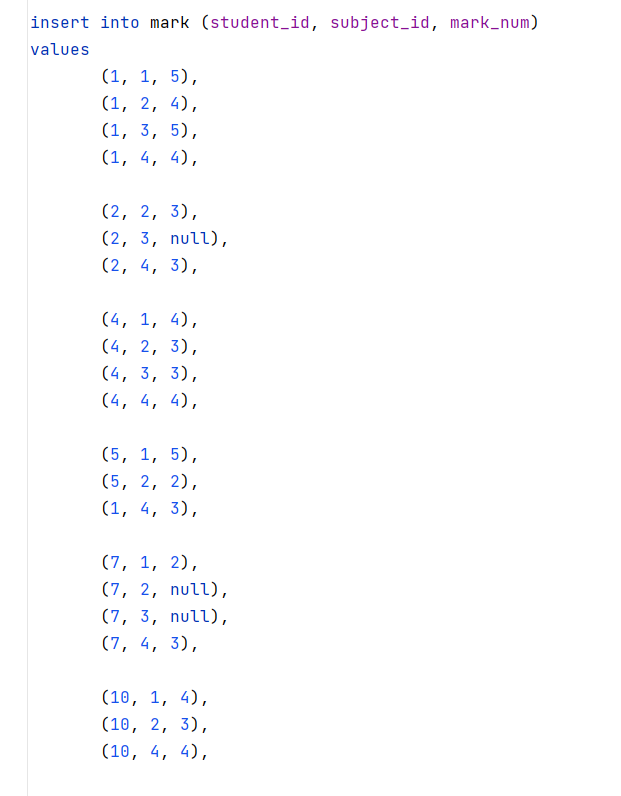


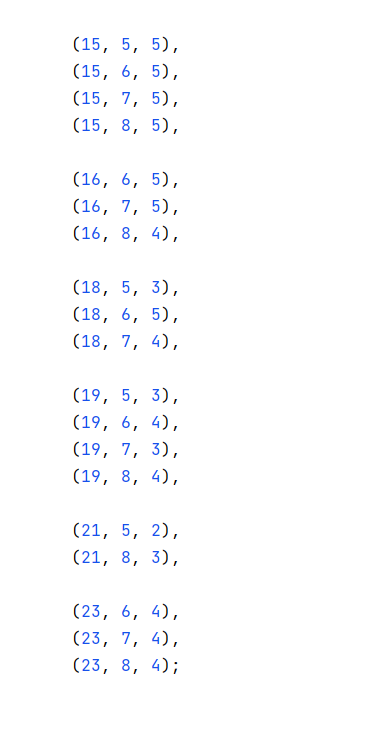


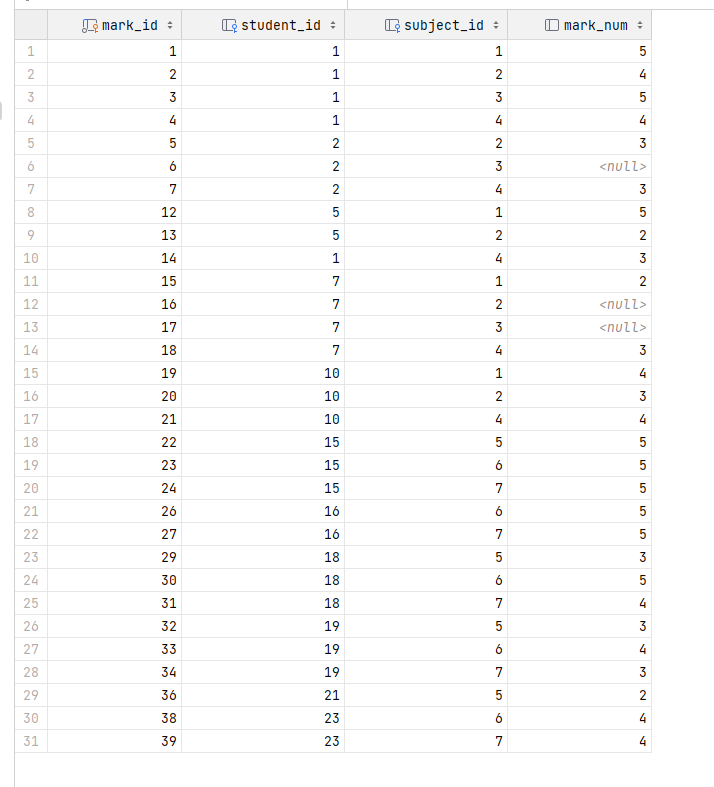








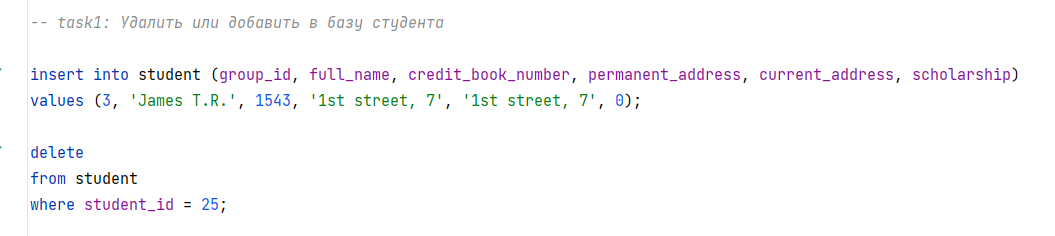


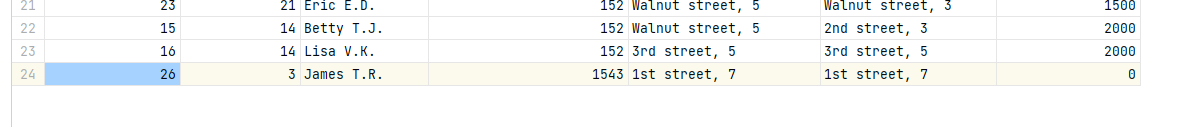


**Запросы к БД:**

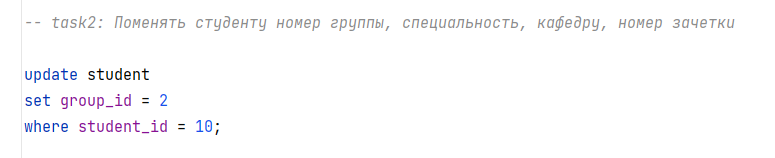
Запросы к БД находятся в приложении Б.

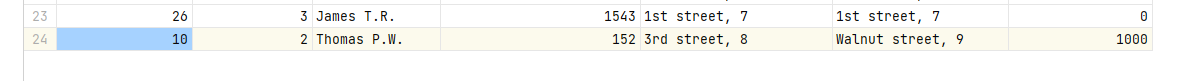
1. Удалить или добавить в базу студента.



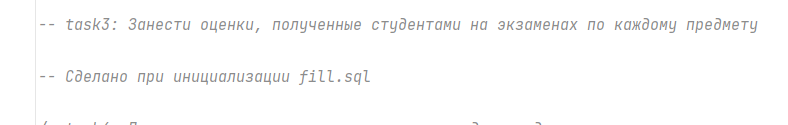


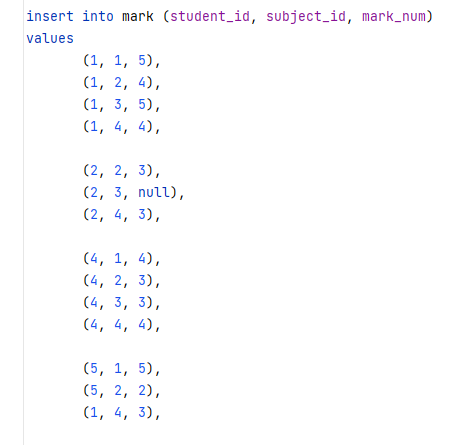
1. Поменять студенту номер группы, специальность, кафедру, номер зачетки.

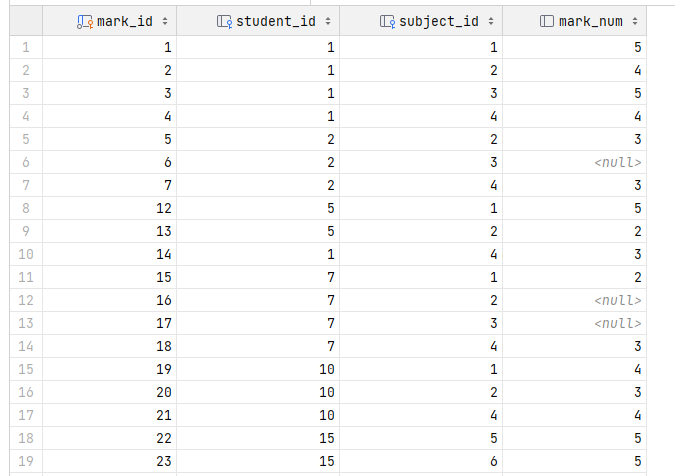
****

****

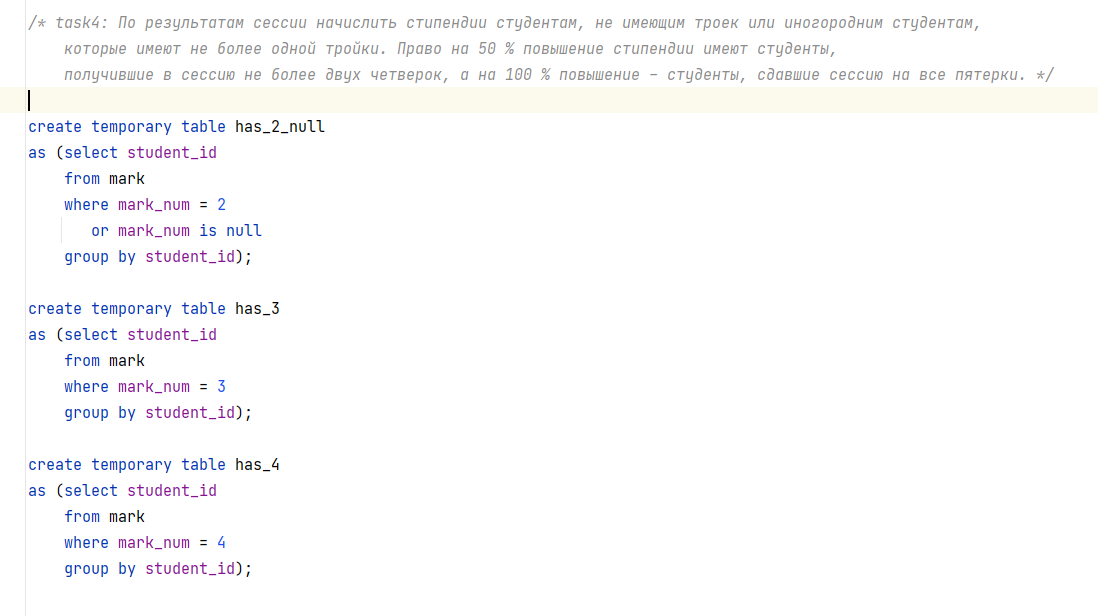
1. Занести оценки, полученные студентами на экзаменах по каждому предмету

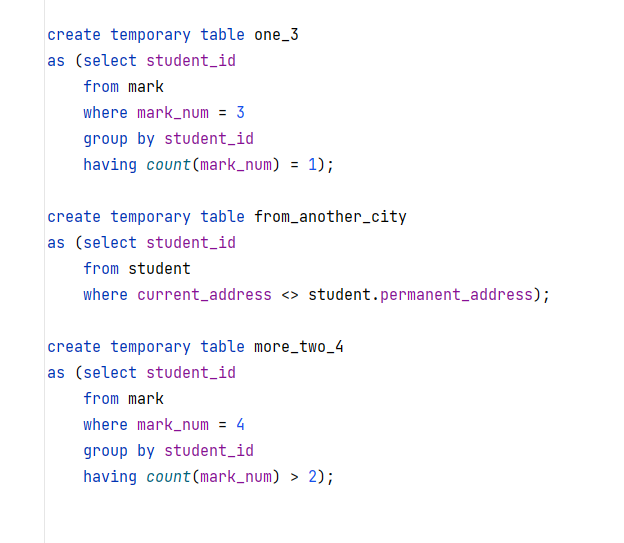


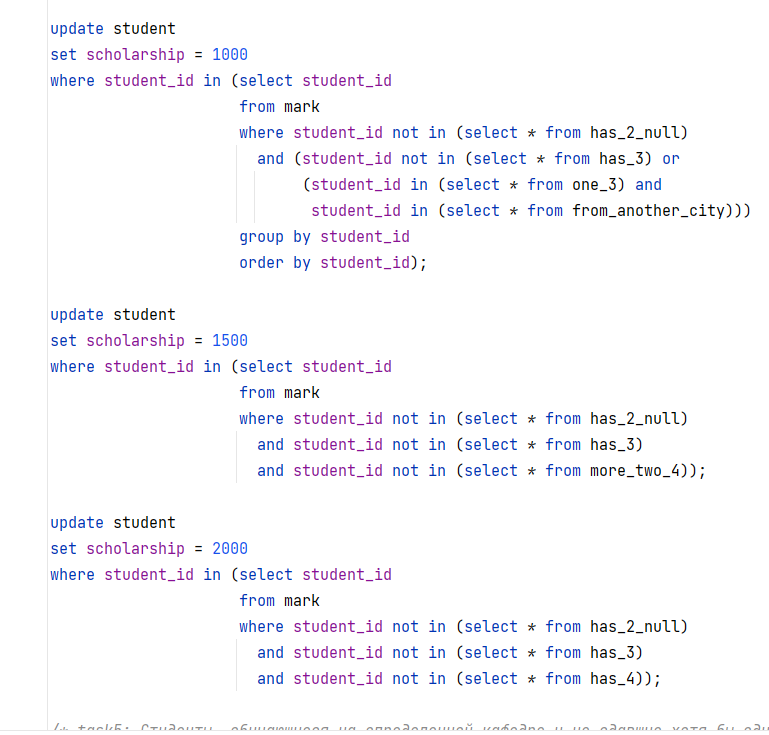


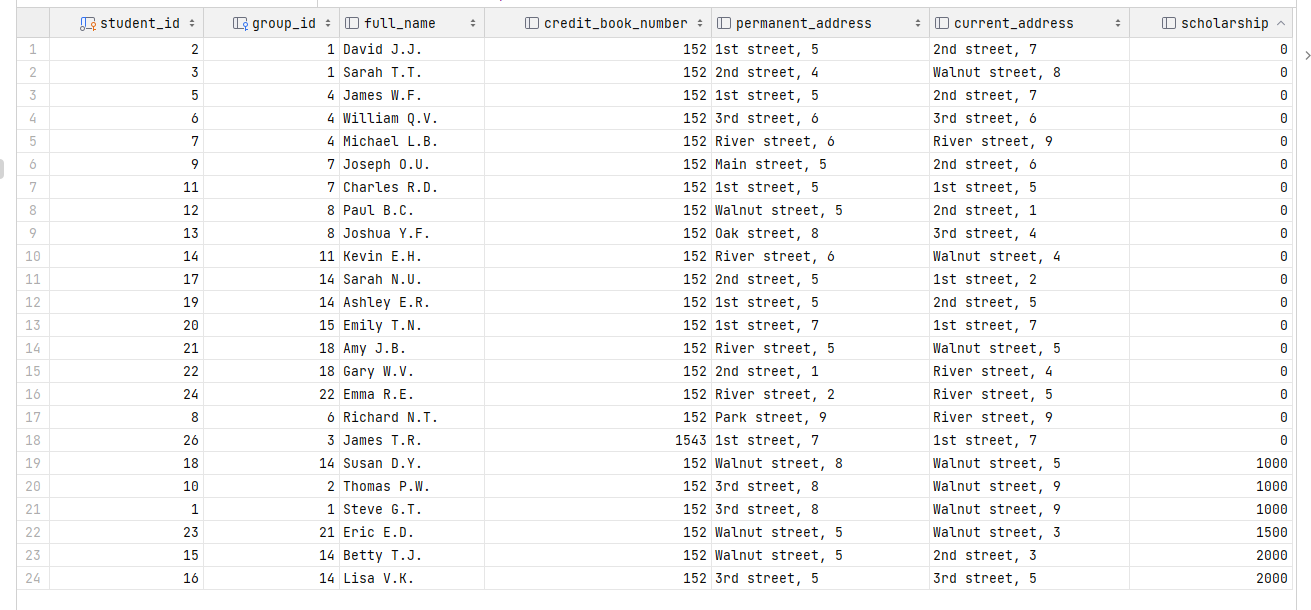


1. По результатам сессии начислить стипендии студентам, не имеющим троек или иногородним студентам, которые имеют не более одной тройки. Право на 50 % повышение стипендии имеют студенты, получившие в сессию не более двух четверок, а на 100 % повышение – студенты, сдавшие сессию на все пятерки

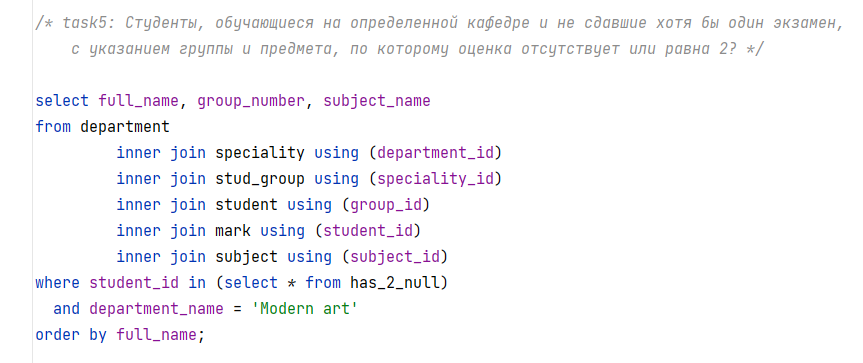


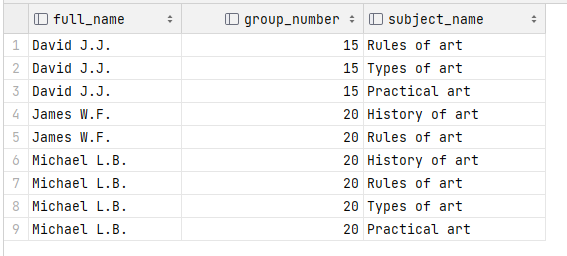




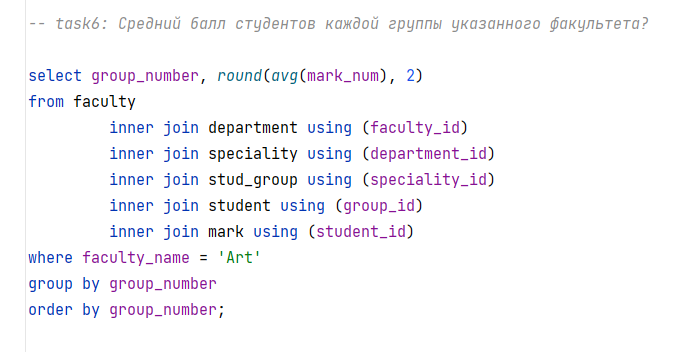


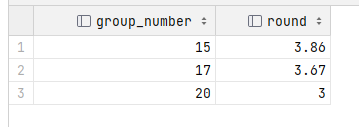
5. Студенты, обучающиеся на определенной кафедре и не сдавшие хотя бы один экзамен, с указанием группы и предмета, по которому оценка отсутствует или равна 2?



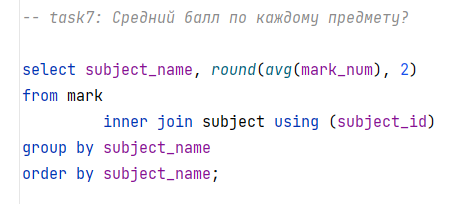


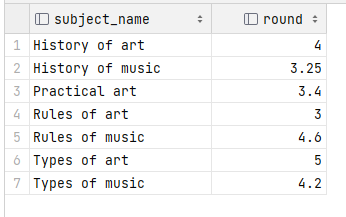
6. Средний балл студентов каждой группы указанного факультета?



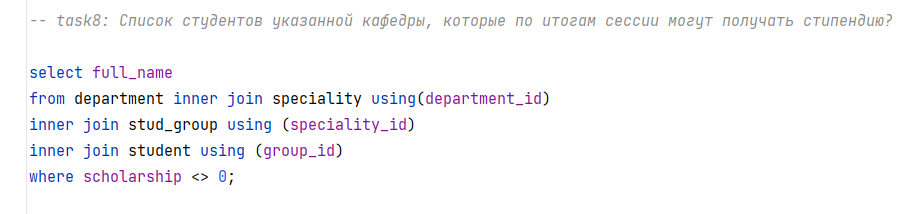


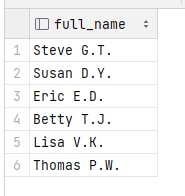
7. Средний балл по каждому предмету?





8. Список студентов указанной кафедры, которые по итогам сессии могут получать стипендию?





**Вывод.**

При выполнении работы создана локальная база данных PostgreSQL, созданы необходимые таблицы, заполнены данными. После этого написаны запросы, с помощью которых выводится необходимая информация из таблиц и добавляется новая.

**ПРИЛОЖЕНИЕ А**

PullRequest: https://github.com/moevm/sql-2023-1303/pull/24

db-fiddle : https://www.db-fiddle.com/f/vG4PgaVB9W7Rxsobwd6tkz/2

**ПРИЛОЖЕНИЕ Б**

-- creating tables

create table if not exists faculty

(

faculty\_id int primary key generated by default as identity,

faculty\_name varchar(100)

);

create table if not exists department

(

department\_id int primary key generated by default as identity,

faculty\_id int references faculty (faculty\_id) on delete cascade,

department\_name varchar(100)

);

create table speciality

(

speciality\_id int primary key generated by default as identity,

department\_id int references department (department\_id) on delete cascade,

speciality\_name varchar(100)

);

create table stud\_group

(

group\_id int primary key generated by default as identity,

speciality\_id int references speciality (speciality\_id) on delete cascade,

group\_number int,

creation\_date int

);

create table student

(

student\_id int primary key generated by default as identity,

group\_id int references stud\_group (group\_id) on delete cascade,

full\_name varchar(200),

credit\_book\_number int,

permanent\_address varchar(100),

current\_address varchar(100),

scholarship int default 0

);

create table subject

(

subject\_id int primary key generated by default as identity ,

subject\_name varchar(100)

);

create table mark (

mark\_id int primary key generated by default as identity ,

student\_id int references student (student\_id) on delete cascade,

subject\_id int references mark (mark\_id) on delete cascade,

mark\_num int

);

-- filling data

insert into faculty (faculty\_name)

values ('Art'),

('Music'),

('Science'),

('Physics'),

('Language');

insert into department (faculty\_id, department\_name)

VALUES (1, 'Modern art'),

(1, 'Medieval art'),

(2, 'Classical music'),

(2, 'Instrumental music'),

(3, 'Modern science'),

(3, 'Ancient science'),

(4, 'Atomic physics'),

(4, 'Basic physics'),

(5, 'Native language'),

(5, 'Foreign language');

insert into speciality (department\_id, speciality\_name)

values (1, 'Street art'),

(1, 'Cynical art'),

(2, 'Romanesque art'),

(2, 'Gothic art'),

(3, 'Symphony music'),

(3, 'Opera music'),

(4, 'Symphonic music'),

(4, 'Chamber music'),

(5, 'Advanced modern science'),

(6, 'Profound ancient science'),

(7, 'Deep atomic physics'),

(8, 'Elementary basic physics'),

(9, 'Culture of native language'),

(10, 'Nuances of foreign language');

insert into stud\_group (speciality\_id, group\_number, creation\_date)

values (1, 15, 2019),

(1, 17, 2013),

(1, 19, 2015),

(2, 20, 2008),

(3, 31, 2003),

(3, 39, 2008),

(4, 46, 2015),

(4, 47, 2019),

(4, 48, 2008),

(5, 53, 2003),

(5, 57, 2008),

(6, 68, 2008),

(7, 70, 2015),

(7, 72, 2019),

(7, 75, 2003),

(7, 79, 2013),

(8, 87, 2015),

(8, 89, 2019),

(9, 912, 2008),

(10, 927, 2015),

(11, 931, 2003),

(12, 948, 2015),

(13, 957, 2008),

(14, 962, 2019);

insert into student (group\_id, full\_name, credit\_book\_number, permanent\_address, current\_address)

values (1, 'Steve G.T.', 152, '3rd street, 8', 'Walnut street, 9'),

(1, 'David J.J.', 152, '1st street, 5', '2nd street, 7'),

(1, 'Sarah T.T.', 152, '2nd street, 4', 'Walnut street, 8'),

(4, 'Carol R.M.', 152, 'River street, 6', 'River street, 6'),

(4, 'James W.F.', 152, '1st street, 5', '2nd street, 7'),

(4, 'William Q.V.', 152, '3rd street, 6', '3rd street, 6'),

(4, 'Michael L.B.', 152, 'River street, 6', 'River street, 9'),

(6, 'Richard N.T.', 152, 'Park street, 9', 'River street, 9'),

(7, 'Joseph O.U.', 152, 'Main street, 5', '2nd street, 6'),

(7, 'Thomas P.W.', 152, '3rd street, 8', 'Walnut street, 9'),

(7, 'Charles R.D.', 152, '1st street, 5', '1st street, 5'),

(8, 'Paul B.C.', 152, 'Walnut street, 5', '2nd street, 1'),

(8, 'Joshua Y.F.', 152, 'Oak street, 8', '3rd street, 4'),

(11, 'Kevin E.H.', 152, 'River street, 6', 'Walnut street, 4'),

(14, 'Betty T.J.', 152, 'Walnut street, 5', '2nd street, 3'),

(14, 'Lisa V.K.', 152, '3rd street, 5', '3rd street, 5'),

(14, 'Sarah N.U.', 152, '2nd street, 5', '1st street, 2'),

(14, 'Susan D.Y.', 152, 'Walnut street, 8', 'Walnut street, 5'),

(14, 'Ashley E.R.', 152, '1st street, 5', '2nd street, 5'),

(15, 'Emily T.N.', 152, '1st street, 7', '1st street, 7'),

(18, 'Amy J.B.', 152, 'River street, 5', 'Walnut street, 5'),

(18, 'Gary W.V.', 152, '2nd street, 1', 'River street, 4'),

(21, 'Eric E.D.', 152, 'Walnut street, 5', 'Walnut street, 3'),

(22, 'Emma R.E.', 152, 'River street, 2', 'River street, 5');

insert into subject (subject\_name)

values ('History of art'),

('Rules of art'),

('Types of art'),

('Practical art'),

('History of music'),

('Rules of music'),

('Types of music'),

('Practical music'),

('English'),

('Math');

insert into mark (student\_id, subject\_id, mark\_num)

values

(1, 1, 5),

(1, 2, 4),

(1, 3, 5),

(1, 4, 4),

(2, 2, 3),

(2, 3, null),

(2, 4, 3),

(4, 1, 4),

(4, 2, 3),

(4, 3, 3),

(4, 4, 4),

(5, 1, 5),

(5, 2, 2),

(1, 4, 3),

(7, 1, 2),

(7, 2, null),

(7, 3, null),

(7, 4, 3),

(10, 1, 4),

(10, 2, 3),

(10, 4, 4),

(15, 5, 5),

(15, 6, 5),

(15, 7, 5),

(15, 8, 5),

(16, 6, 5),

(16, 7, 5),

(16, 8, 4),

(18, 5, 3),

(18, 6, 5),

(18, 7, 4),

(19, 5, 3),

(19, 6, 4),

(19, 7, 3),

(19, 8, 4),

(21, 5, 2),

(21, 8, 3),

(23, 6, 4),

(23, 7, 4),

(23, 8, 4);

-- some tasks on this data tables

-- task1: Удалить или добавить в базу студента

insert into student (group\_id, full\_name, credit\_book\_number, permanent\_address, current\_address, scholarship)

values (3, 'James T.R.', 1543, '1st street, 7', '1st street, 7', 0);

delete

from student

where student\_id = 25;

-- task2: Поменять студенту номер группы, специальность, кафедру, номер зачетки

update student

set group\_id = 6

where student\_id = 8;

-- task3: Занести оценки, полученные студентами на экзаменах по каждому предмету

-- Сделано при инициализации fill.sql

/\* task4: По результатам сессии начислить стипендии студентам, не имеющим троек или иногородним студентам,

которые имеют не более одной тройки. Право на 50 % повышение стипендии имеют студенты,

получившие в сессию не более двух четверок, а на 100 % повышение – студенты, сдавшие сессию на все пятерки. \*/

create temporary table has\_2\_null

as (select student\_id

from mark

where mark\_num = 2

or mark\_num is null

group by student\_id);

create temporary table has\_3

as (select student\_id

from mark

where mark\_num = 3

group by student\_id);

create temporary table has\_4

as (select student\_id

from mark

where mark\_num = 4

group by student\_id);

create temporary table one\_3

as (select student\_id

from mark

where mark\_num = 3

group by student\_id

having count(mark\_num) = 1);

create temporary table from\_another\_city

as (select student\_id

from student

where current\_address <> student.permanent\_address);

create temporary table more\_two\_4

as (select student\_id

from mark

where mark\_num = 4

group by student\_id

having count(mark\_num) > 2);

update student

set scholarship = 1000

where student\_id in (select student\_id

from mark

where student\_id not in (select \* from has\_2\_null)

and (student\_id not in (select \* from has\_3) or

(student\_id in (select \* from one\_3) and

student\_id in (select \* from from\_another\_city)))

group by student\_id

order by student\_id);

update student

set scholarship = 1500

where student\_id in (select student\_id

from mark

where student\_id not in (select \* from has\_2\_null)

and student\_id not in (select \* from has\_3)

and student\_id not in (select \* from more\_two\_4));

update student

set scholarship = 2000

where student\_id in (select student\_id

from mark

where student\_id not in (select \* from has\_2\_null)

and student\_id not in (select \* from has\_3)

and student\_id not in (select \* from has\_4));

/\* task5: Студенты, обучающиеся на определенной кафедре и не сдавшие хотя бы один экзамен,

с указанием группы и предмета, по которому оценка отсутствует или равна 2? \*/

select full\_name, group\_number, subject\_name

from department

inner join speciality using (department\_id)

inner join stud\_group using (speciality\_id)

inner join student using (group\_id)

inner join mark using (student\_id)

inner join subject using (subject\_id)

where student\_id in (select \* from has\_2\_null)

and department\_name = 'Modern art'

order by full\_name;

-- task6: Средний балл студентов каждой группы указанного факультета?

select group\_number, round(avg(mark\_num), 2)

from faculty

inner join department using (faculty\_id)

inner join speciality using (department\_id)

inner join stud\_group using (speciality\_id)

inner join student using (group\_id)

inner join mark using (student\_id)

where faculty\_name = 'Art'

group by group\_number

order by group\_number;

-- task7: Средний балл по каждому предмету?

select subject\_name, round(avg(mark\_num), 2)

from mark

inner join subject using (subject\_id)

group by subject\_name

order by subject\_name;

-- task8: Список студентов указанной кафедры, которые по итогам сессии могут получать стипендию?

select full\_name

from department inner join speciality using(department\_id)

inner join stud\_group using (speciality\_id)

inner join student using (group\_id)

where scholarship <> 0;