

Министерство образования и науки Российской Федерации  
федеральное государственное автономное образовательное учреждение высшего  
образования  
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,  
МЕХАНИКИ И ОПТИКИ»  
(УНИВЕРСИТЕТ ИТМО)  
Факультет среднего профессионального образования

ОТЧЁТ  
О ПРАКТИЧЕСКОЙ РАБОТЕ № 5  
по теме: Анализ данных. Создание таблиц бд PostgreSQL.  
Заполнение таблиц рабочими данными. Создание запросов  
по дисциплине: Основы проектирования баз данных

Специальность 09.02.07 Информационные системы и программирование

Проверил:

\_\_\_\_\_ Говоров А.И.

Дата: « \_\_\_\_ » \_\_\_\_\_ 2020г.

Оценка \_\_\_\_\_

Выполнил:

студент группы Y2336

\_\_\_\_\_ Морозов В.А.

## ПОСТАНОВКА ЗАДАЧИ

Цель лабораторной работы №5: овладеть практическими навыками создания таблиц базы данных PostgreSQL 12, заполнения их рабочими данными, резервного копирования и восстановления баз данных.

## ЗАДАНИЕ

Необходимо реализовать некоторое количество SQL-запросов.  
Примерный набор требуемых запросов:

- выбор значений, заданных атрибутов из более, чем двух таблиц, с сортировкой – от 1 балла;
  - использование условий WHERE, состоящих из более, чем одного условия – от 1 балла;
  - использование функций для работы с датами – от 2 баллов;
  - использование строковых функций – от 3 баллов;
  - запрос с использованием подзапросов – от 2 баллов (многострочный подзапрос - от 3 баллов);
  - вычисление групповой (агрегатной) функции – от 1 балла (с несколькими таблицами – от 3 баллов);
  - вычисление групповой (агрегатной) функции с условием HAVING – от 2 баллов;
  - использование предикатов EXISTS, ALL, SOME и ANY - от 4 баллов;
  - использование запросов с операциями реляционной алгебры (объединение, пересечение и т.д.) - от 3 баллов;
  - использование объединений запросов (inner join и т.д.) - от 3 баллов.
- Запросов должно быть не менее 10 (30 баллов на 5, 25 баллов на 4, 20 баллов на 3)(5, 4, 3 - это оценки, а не суммы баллов в журнале).

## ВЫПОЛНЕНИЕ

### Схема

Схема физической модели бд, спроектированной с помощью СА Erwin Data Moduler представлена на рисунке 1.

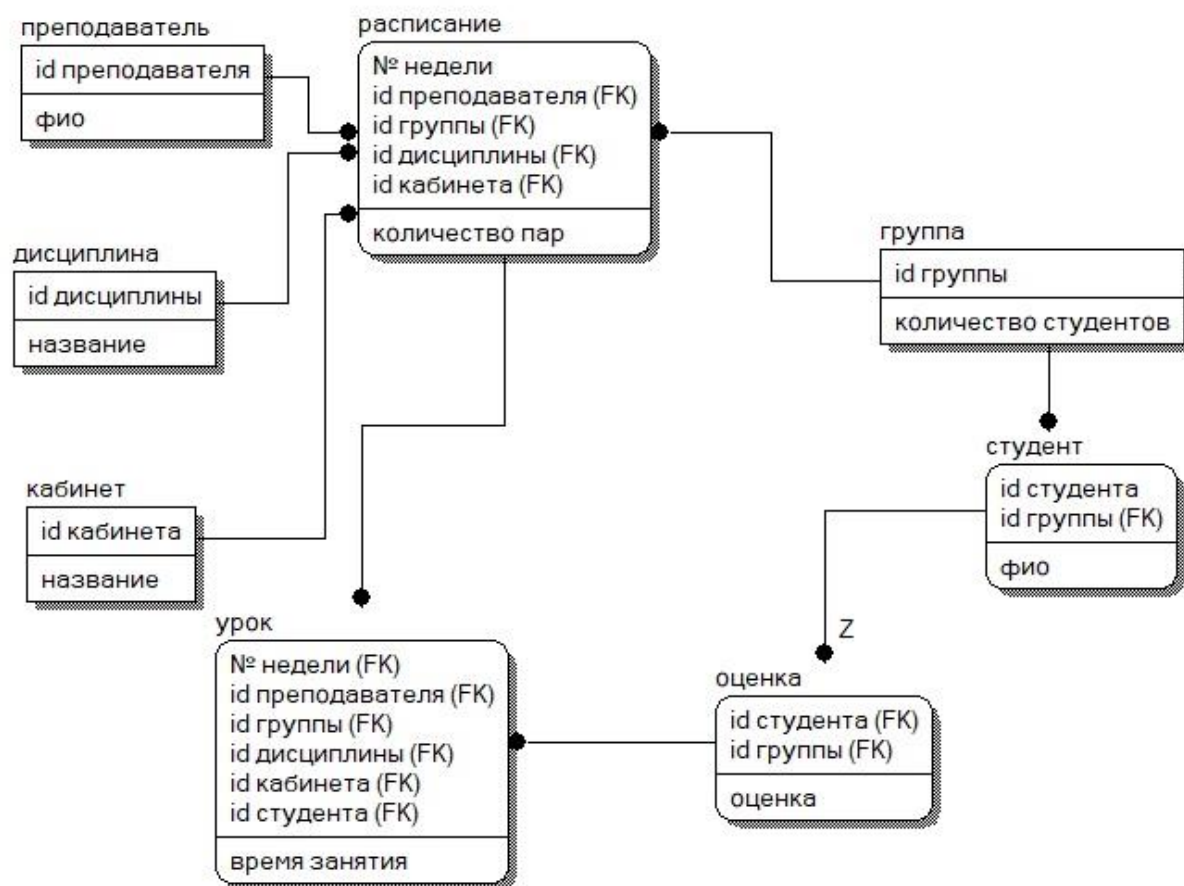


Рисунок 1 Схема информационной модели БД

## Запросы

1) Вывести какой предмет будет в заданной группе в заданный день недели на заданном уроке:

```
SELECT public."Subject".name FROM public."Lesson" inner join
public."Subject" on (public."Lesson".subject_id = public."Subject".id) where group_id
= 1 and teacher_id = 1 and lesson = 'T' and day = 'Monday';
```

	name text
1	Math

2) Вывести кто из преподавателей преподает в заданной группе:

```
SELECT public."Group".number, public."Teacher".name,
public."Teacher".surname
FROM public."TeachersGroups"
inner join public."Group" on (public."Group".id =
public."TeachersGroups".group_id)
inner join public."Teacher" on (public."Teacher".id =
public."TeachersGroups".teacher_id)
where public."Group".number = 'Y2121';
```

	number text	name character varying	surname character varying
1	Y2121	Tanya	Ivanova
2	Y2121	Ivan	Ivanov
3	Y2121	Danya	Konev

3) Вывести в каких группах преподает заданный предмет заданный преподаватель:

```
SELECT          public."Group".number,          public."Teacher".name,
public."Teacher".surname, public."Subject".name

FROM public."Teacher"

    inner join public."TeachersGroups" on (public."Teacher".id =
public."TeachersGroups".teacher_id)

    inner join public."Group" on (public."Group".id =
public."TeachersGroups".group_id)

    inner join public."TeachersSubjects" on (public."Teacher".id =
public."TeachersSubjects".teacher_id)

    inner join public."Subject" on (public."Subject".id =
public."TeachersSubjects".subject_id)

where public."Teacher".id = 2 and public."Subject".name = 'History';
```

	<b>number</b> text	<b>name</b> character varying	<b>surname</b> character varying	<b>name</b> text
1	Y2121	Ivan	Ivanov	History
2	Y2122	Ivan	Ivanov	History

4) Вывести расписание на заданный день недели для указанной группы:

```
SELECT      public."Teacher".name,          public."Teacher".surname,
public."Group".number,      public."Subject".name,      public."Lesson".lesson,
public."Lesson".day

FROM public."Lesson"

      inner      join      public."Teacher"      on      (public."Teacher".id      =
public."Lesson".teacher_id)

      inner      join      public."Subject"      on      (public."Subject".id      =
public."Lesson".subject_id)

inner join public."Group" on ((public."Group".id = public."Lesson".group_id))

where public."Group".number = 'Y2121' and public."Lesson".day = 'Monday';
```

	<b>name</b> character varying	<b>surname</b> character varying	<b>number</b> text	<b>name</b> text	<b>lesson</b> n_lesson	<b>day</b> weekday
1	Tanya	Ivanova	Y2121	Math	I	Monday
2	Tanya	Ivanova	Y2121	Math	II	Monday
3	Tanya	Ivanova	Y2121	Program...	III	Monday
4	Ivan	Ivanov	Y2121	History	IV	Monday

5) Вывести сколько студентов обучается на каждом курсе в указанном классе:

```
SELECT Count(*)

from public."Group"

inner join public."Student" on (public."Group".id = public."Student".group_id)

where public."Group".number = 'Y2121';
```

	<b>count</b> bigint
1	2

6) Вывести средние оценки студентов и сгруппировать их по именам:

```
SELECT public."Student".name, avg(public."Grade".grade) as Average
```

```
FROM public."Grade"
```

```
inner join public."Student" on (public."Student".id = public."Grade".student_id)
```

```
group by public."Student".name;
```

	<b>name</b> text	<b>average</b> numeric
1	Masha	5.0000000000000000
2	Tolik	4.5000000000000000
3	Mirron	4.5000000000000000
4	Pasha	5.0000000000000000
5	Nuta	4.5000000000000000
6	Alex	4.8333333333333333

7) Вывести кабинеты, за которыми не закреплен учитель:

```
SELECT id, "number", teacher_id
```

```
FROM public."Classroom"
```

```
where public."Classroom".teacher_id is null;
```

	<b>id</b> [PK] integer	<b>number</b> integer	<b>teacher_id</b> integer
1	5	201	[null]
2	7	313	[null]



8) Вывести учителей, за которыми не закреплены кабинеты:

```
SELECT public."Teacher".name, public."Teacher".surname
```

```
FROM public."Teacher"
```

```
where
```

```
(select count(*) from public."Classroom" where public."Classroom".teacher_id  
= public."Teacher".id) = 0;
```

	<b>name</b> character varying	<b>surname</b> character varying
1	John	Sieno
2	Dean	Iden

9) Вывести студентов и предметы, по которым у студентов незначит:

```
SELECT public."Student".name, public."Student".surname,  
public."Subject".name, avg(public."Grade".grade) as AvgGrade
```

```
FROM public."Grade"
```

```
inner join public."Subject" on (public."Grade".subject_id = public."Subject".id)
```

```
inner join public."Student" on (public."Grade".student_id = public."Student".id)
```

```
group by public."Student".name, public."Student".surname,  
public."Subject".name
```

```
having avg(public."Grade".grade) <= 2.5;
```

	<b>name</b> text	<b>surname</b> text	<b>name</b> text	<b>avggrade</b> numeric
1	Gosha	Merin	History	2.3333333333333333
2	Gosha	Merin	Engilsh	2.0000000000000000
3	Albert	Gaev	Program...	2.0000000000000000

10) Вывести студентов, у которых есть хотя бы один долг

```
select public."Student".name, public."Student".surname
```

```
from public."Student"
```

```
where
```

```
public."Student".id = any(
```

```
select public."Grade".student_id
```

```
from public."Grade"
```

```
group by public."Grade".student_id
```

```
having avg(public."Grade".grade) < 2.5
```

```
);
```

	<b>name</b> text	<b>surname</b> text
1	Albert	Gaev
2	Gosha	Merin

11) Вывести преподавателей, которые преподают больше, чем в 5 группах

```
select public."Teacher".name, public."Teacher".surname
```

```
from public."Teacher"
```

```
inner join public."TeachersGroups" on (public."TeachersGroups".teacher_id =  
public."Teacher".id)
```

```
group by public."Teacher".name, public."Teacher".surname
```

```
having count(
```

```
public."TeachersGroups".teacher_id
```

```
) > 5;
```

	<b>name</b> character varying	<b>surname</b> character varying
1	Tanya	Ivanova
2	Danya	Konev
3	Ivan	Ivanov

12) Вывести преподавателей, которые преподают один предмет

```
select public."Teacher".name, public."Teacher".surname
from public."Teacher"
inner join public."TeachersSubjects" on (public."TeachersSubjects".teacher_id
= public."Teacher".id)
group by public."Teacher".name, public."Teacher".surname
having count(
    public."TeachersSubjects".teacher_id
) = 1;
```

	<b>name</b> character varying	<b>surname</b> character varying
1	Ann	Postre

13) Вывести студентов, которые обучаются в группе Y2121

```
SELECT          public."Group".number,          public."Student".name,
public."Student".surname
from public."Group"
inner join public."Student" on (public."Student".group_id = public."Group".id)
where public."Group".number = 'Y2121';
```

	<b>number</b> text	<b>name</b> text	<b>surname</b> text
1	Y2121	Alex	McLaden
2	Y2121	Pasha	Sharik

14) Вывести расписание для группы Y2336 и сгруппировать по дням недели и паре

```

SELECT          public."Group".number,          public."Subject".name,
public."Lesson".lesson, public."Lesson".day

from public."Group"

inner join public."Lesson" on (public."Lesson".group_id = public."Group".id)

inner  join  public."Subject"  on  (public."Lesson".subject_id  =
public."Subject".id)

where public."Group".number = 'Y2121'

group by public."Lesson".day, public."Lesson".lesson, public."Group".number,
public."Subject".name;

```

	<b>number</b> text	<b>name</b> text	<b>lesson</b> n_lesson	<b>day</b> weekday
1	Y2121	Math	I	Monday
2	Y2121	Math	II	Monday
3	Y2121	Programmi...	III	Monday
4	Y2121	History	IV	Monday
5	Y2121	History	II	Thursday
6	Y2121	Engilsh	III	Thursday
7	Y2121	Math	I	Saturday

15) Вывести текущие средние оценки по каждому предмету студента Alex

```
SELECT      public."Student".name,      public."Student".surname,
public."Subject".name, avg(public."Grade".grade)

from public."Student"

inner join public."Grade" on (public."Student".id = public."Grade".student_id)

inner join public."Subject" on (public."Grade".subject_id = public."Subject".id)

where public."Student".name = 'Alex'

group      by      public."Student".name,      public."Student".surname,
public."Subject".name;
```

	<b>name</b> text	<b>surname</b> text	<b>name</b> text	<b>avg</b> numeric
1	Alex	McLaden	Engilsh	5.0000000000000000
2	Alex	McLaden	History	5.0000000000000000
3	Alex	McLaden	Math	4.6666666666666667
4	Alex	McLaden	Program...	5.0000000000000000

16) Вывести оценки, которые были поставлены за сегодня

```
SELECT      public."Student".name,      public."Student".surname,
public."Subject".name, public."Grade".grade, public."Grade".date

FROM public."Grade"

inner join public."Student" on (public."Student".id = public."Grade".student_id)

inner join public."Subject" on (public."Subject".id = public."Grade".subject_id)

where public."Grade".date = current_date;
```

	<b>name</b> text	<b>surname</b> text	<b>name</b> text	<b>grade</b> integer	<b>date</b> date
1	Pasha	Sharik	Program...	3	2020-04...
2	Albert	Gaev	Engilsh	4	2020-04...
3	Albert	Gaev	Engilsh	4	2020-04...