

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

## **ЗАПРОСЫ НА ВЫБОРКУ ДАННЫХ**

Отчёт о лабораторной работе № 4  
по дисциплине «Базы данных»

Выполнил: студент гр. 431-3

\_\_\_\_\_ Бекиш Е.П.

« \_\_\_\_ » \_\_\_\_\_ 2023 г.

Проверил: ассистент каф. АСУ

\_\_\_\_\_ Яблонский Я. В.

« \_\_\_\_ » \_\_\_\_\_ 2023 г.

Томск 2023

## 1 Цель лабораторной работы

- познакомиться с основными командами для формирования запросов в SQL;
- сформировать запросы для индивидуального задания в SQL;

## 2 Описание работы

Используя либо `psql`, либо `pgAdmin` создать нижеперечисленные запросы.

1. Получить список студентов первого года обучения с указанием комнат, в которых они проживают. Нужно выбрать столбец с некоторыми полями из определенной таблицы, необходимо использовать `SELECT` имя\_поля1, имя\_поля2, ..., имя\_поляN `FROM` имя\_таблицы1, имя\_таблицы2, ..., имя\_таблицыN. Теперь, чтобы задать только некоторую часть таблицы, которая соответствует определенному условию, нужно использовать ключевое слово `WHERE` имя\_поля.

Данный запрос выглядит следующим образом:

```
select snam, rnum from student where ann='1';
```

Выбираем из таблицы имя и комнату студента, в которой он проживает соответствующему году обучения. Результат индивидуального задания представлен на рисунке 2.1.

```
tusur=# select snam, rnum from student where ann='1';
          snam                | rnum
-----+-----
 Баранов Алексей Андреевич   | 000
 Барашкин Алексей Андреевич  | 000
 Быков Алексей Андреевич     | 000
(3 строки)
```

Рисунок 2.1 – успешный результат индивидуального задания

2. Для заданной дисциплины получить список сдавших её студентов с указанием экзаменатора и полученной оценки. Здесь применяем те же действия, что и в первом индивидуальном варианте, сейчас необходимо совершить внешнее соединение полей из разных таблиц. Это можно сделать с помощью оператора `JOIN`. Имеет следующий формальный синтаксис:

SELECT имя\_поля1, имя\_поля2, ..., имя\_поляN FROM имя\_таблицы1, имя\_таблицы2, ..., имя\_таблицыN JOIN имя\_таблицы2 ON условие1 JOIN таблица2 ON условие2. Перед оператором JOIN указывается одно из ключевых слов LEFT, RIGHT или FULL, которые определяют тип соединения:

- LEFT: выборка будет содержать все строки из первой или левой таблицы.
- RIGHT: выборка будет содержать все строки из второй или правой таблицы.
- FULL: выборка будет содержать все строки из обеих таблиц.

Перед оператором JOIN может указываться ключевое слово OUTER, но его применение необязательно. После JOIN указывается присоединяемая таблица, а затем идет условие соединения после оператора ON.

Данный запрос выглядит следующим образом:

```
select snam, pnam, ball from (select snum, exams.pnam, ball from "pr-dis"
join exams on exams.pnam="pr-dis".pnam and dnam='ПЯВУ') temp join student
on student.snum = temp.snum and ball > 2.
```

Создадим таблицу `select snum, exams.pnam, ball from "pr-dis" join exams on exams.pnam="pr-dis".pnam and dnam='ПЯВУ'`, в которой будет находиться поля с номером студенческого билета, имена преподавателей и полученный балл за дисциплину. Из данной таблицы будем выбирать имена студентов, имена преподавателей, оценку в том случае, если в данной таблицы имеется студенческий билет, который присутствует в таблицы STUDENT и полученная оценка за дисциплину выше двойки. Результат индивидуального задания представлен на рисунке 2.2.

```
tusur=# select snam, pnam, ball from
tusur-# (select snum, exams.pnam, ball from "pr-dis" join exams on exams.pnam="pr-dis".pnam and dnam='ПЯВУ') temp
tusur-# join student on
tusur-# student.snum = temp.snum and ball > 2;
```

snam	pnam	ball
Баранов Алексей Андреевич	Белоусова София Романовна	4
Барашкин Алексей Андреевич	Белоусова София Романовна	4
Воловский Алексей Андреевич	Белоусова София Романовна	3
Гатов Николай Саввич	Белоусова София Романовна	5
Батов Иван Константинович	Белоусова София Романовна	4

(5 строк)

Рисунок 2.2 – успешный результат индивидуального задания

3. Подсчитать количество студентов указанного года обучения, сдавших указанную дисциплину. Теперь сформируем запрос на основе второго индивидуального задания только необходимо показать не поля определенных имен, а количество строк в полученной таблице. Это можно сделать при помощи агрегатной функции COUNT. Существует две формы использования:

- COUNT (\*): находит количество строк в запросе
- COUNT (expression): находит количество строк в запросе, для которых expression не содержит значение NULL.

Функция COUNT вычисляет количество строк в выборке. Есть две формы этой функции. Первая форма COUNT (\*) подсчитывает число строк в выборке: SELECT COUNT (\*) FROM имя\_таблицы.

Вторая форма функции вычисляет количество строк по определенному столбцу, при этом строки со значениями NULL игнорируются: SELECT COUNT (DISTINCT имя\_поля) FROM имя\_таблицы.

Данный запрос выглядит следующим образом:

```
select count(*) from (select snum, exams.pnam, ball from "pr-dis" join
exams on exams.pnam="pr-dis".pnam and dnam='ВУПЯ') temp join student on
student.snum = temp.snum and ann='1' and ball > 2.
```

Создадим таблицу select snum, exams.pnam, ball from "pr-dis" join exams on exams.pnam="pr-dis".pnam and dnam='ВУПЯ', в которой будут находиться поля с номером студенческого билета, имена преподавателей и полученный балл за дисциплину. Из данной таблицы будем выбирать имена студентов, имена преподавателей и оценку в том случае, если в этой таблицы имеется

студенческий билет, который присутствует в таблице STUDENT и полученная оценка за дисциплину выше двойки с указанным годом обучения. После получаем таблицу с определенным количеством строк и считаем их. Результат индивидуального задания представлен на рисунке 2.3.

```
tusur=# select count(*) from
tusur=# (select snum, exams.pnam, ball from "pr-dis" join exams on exams.pnam="pr-dis".pnam and dnam='ВУПЯ') temp
tusur=# join student on
tusur=# student.snum = temp.snum and ann='1' and ball > 2;
count
-----
      3
(1 строка)
```

Рисунок 2.3 – успешный результат индивидуального задания

4. Для каждого студента подсчитать средний балл. Еще раз повторим построение запроса из второго и третьего индивидуального задания. И для подсчета среднего значения тоже будем использовать агрегатную функцию AVG. Входной параметр должен представлять один из следующих типов: smallint, int, bigint, real, double precision, numeric, interval. Для целочисленных параметров результатом будет значение типа numeric, для параметров, которые представляют число с плавающей точкой, - значение типа double precision. Функция AVG возвращает среднее значение на диапазоне значений столбца таблицы. Найдем среднее значение таблицы из базы данных: SELECT AVG (имя\_поля) AS Average\_Price FROM имя\_таблицы.

Данный запрос выглядит следующим образом: select snam, avg(ball) from exams join student on exams.snum = student.snum group by snam. Из соединенных таблиц STUDENT и EXAMS необходима выбрать поля с именем студента и его среднеарифметической оценкой за все сданные дисциплины. Результат индивидуального задания представлен на рисунке 2.4.

```
tusur=# select snam, avg(ball) from exams join student on exams.snum = student.snum group by snam;
      snam          |      avg
-----+-----
Баранов Алексей Андреевич | 4.0000000000000000
Барашкин Алексей Андреевич | 3.5000000000000000
Батов Иван Константинович  | 3.8333333333333333
Быков Алексей Андреевич    | 4.0000000000000000
Воловский Алексей Андреевич | 3.7500000000000000
Гатов Николай Саввич       | 5.0000000000000000
(6 строк)
```

Рисунок 2.4 – успешный результат индивидуального задания

### **3 Выводы**

В ходе выполнения лабораторной работы я научился формировать запросы из перечисленных запросов, которые были представлены в индивидуальных заданиях.