Лабораторная №3

Часть 1

Цель: изучить конструкции языка SQL для манипулирования данными в СУБД MSSQL.

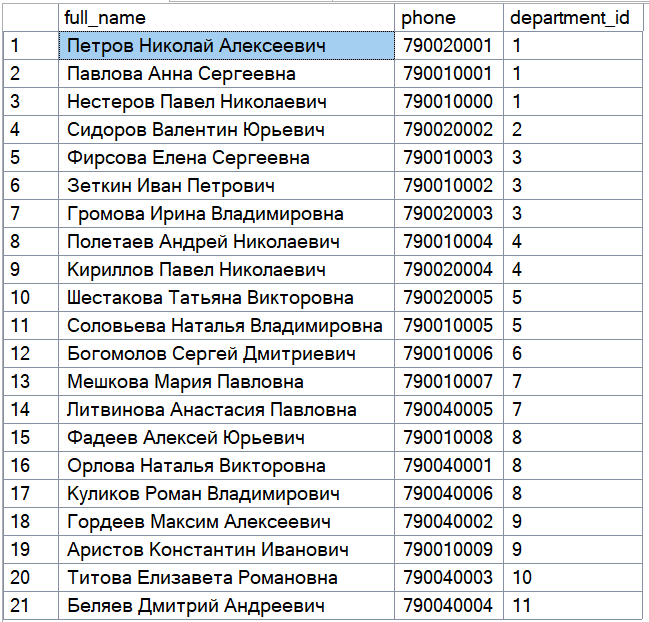
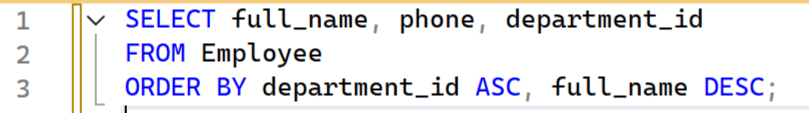
Задания и краткое описание работы:

**1. Выборка из одной таблицы.**

1.1 Выбрать из произвольной таблицы данные и

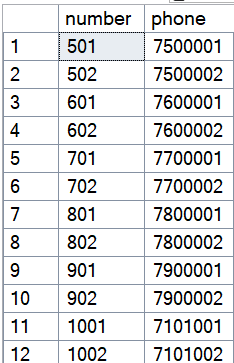
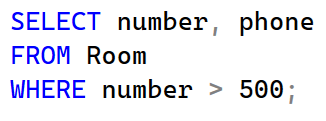
отсортировать их по двум произвольным имеющимся в таблице признакам (разные направления сортировки).

*Список сотрудников, отсортированный сначала по id кафедры(по возрастанию), затем по имени сотрудника (по убыванию).*

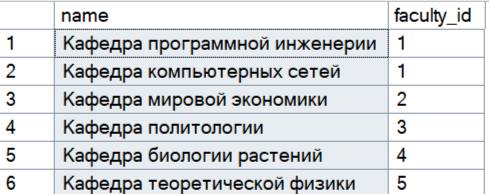
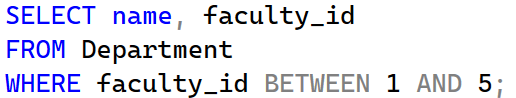


1.2 Выбрать из произвольной таблицы те записи, которые удовлетворяют условию отбора (where). Привести 2-3 запроса.

*Кабинеты с номером больше 500*

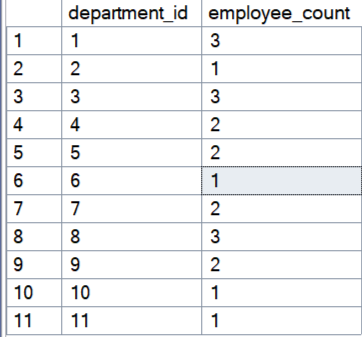
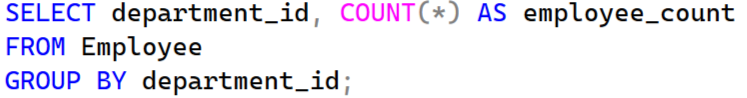


*Кафедры, относящиеся к факультетам с id от 1 до 5*

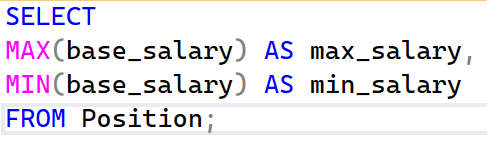


1.3 Привести примеры 2-3 запросов с использованием агрегатных функций (count, max, sum и др.) с группировкой и без группировки.

*количество сотрудников на каждой кафедре*

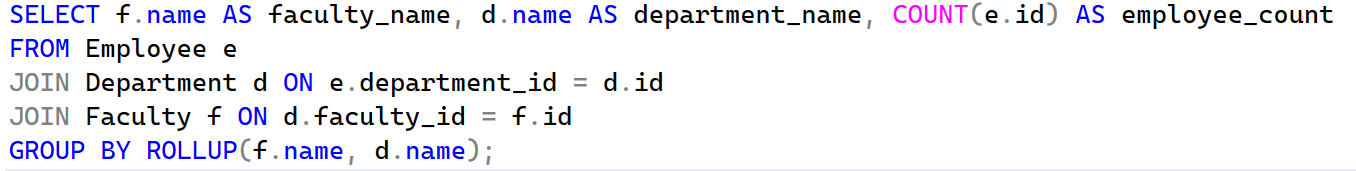


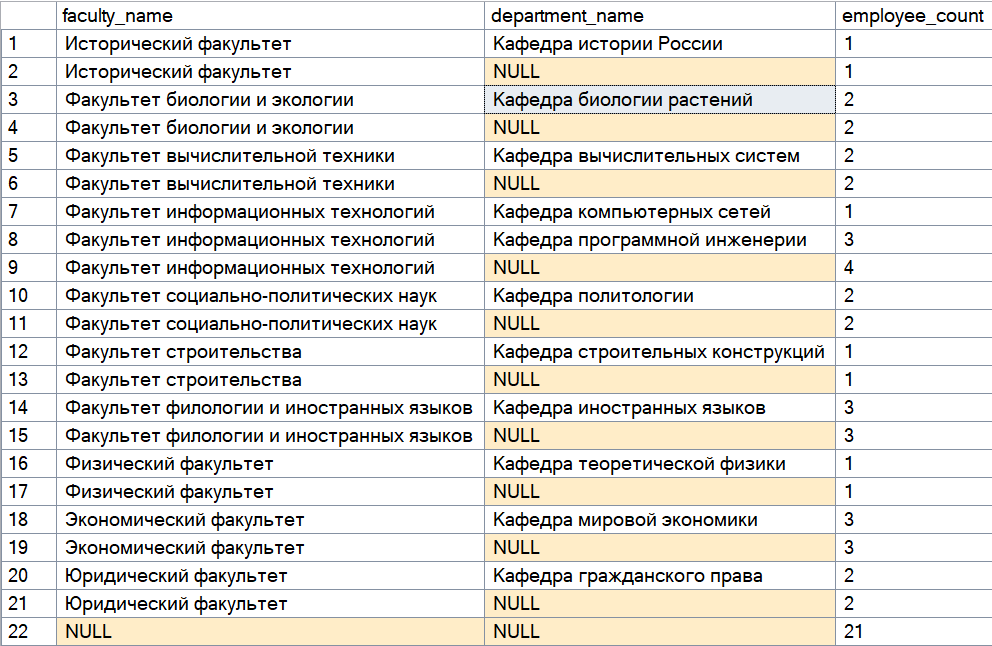
*максимальная и минимальная базовая зарплата среди всех должностей*



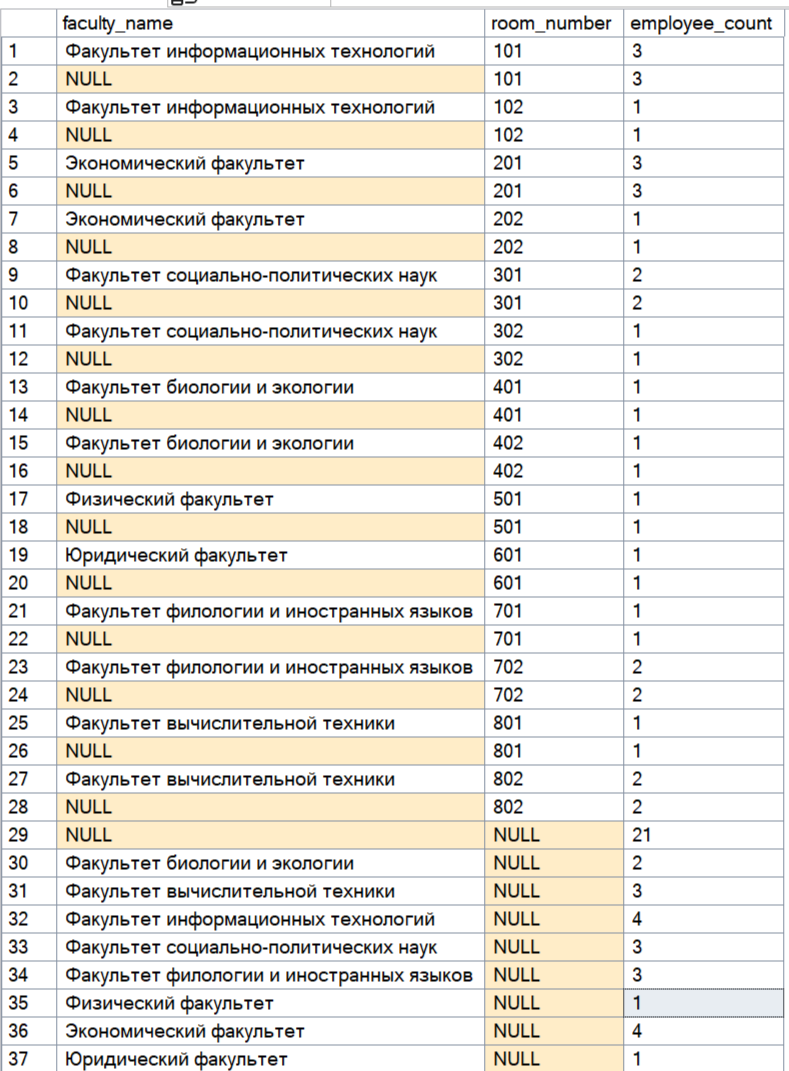
1.4  Привести примеры подведения подытога с использованием GROUP BY [ALL] [ CUBE | ROLLUP](2-3 запроса). В ROLLUP и CUBE использовать не менее 2-х столбцов.

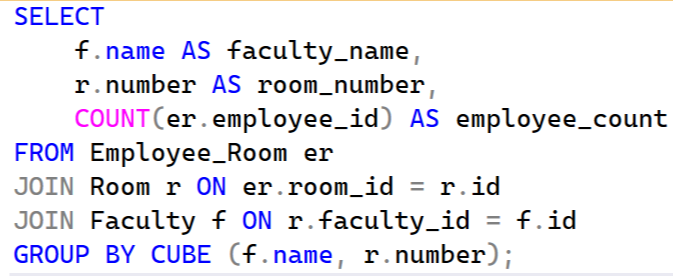
*количество сотрудников по факультетам и кафедрам с* ***ROLLUP***



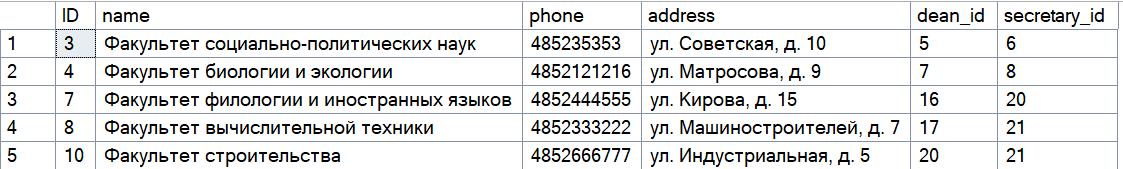


*Количество сотрудников по факультетам и номерам комнат с CUBE*





1.5 Выбрать из таблиц информацию об объектах, в названиях которых нет заданной последовательности букв (LIKE).



***2. Выборка из нескольких таблиц.***

2.1 Вывести информацию подчиненной (дочерней) таблицы, заменяя коды

(значения внешних ключей) соответствующими символьными значениями из

родительских таблиц. Привести 2-3 запроса с использованием классического

подхода соединения таблиц (where).

*Таблица с ФИО сотрудников, их телефонами, названиями кафедр и факультетов вместо ID*

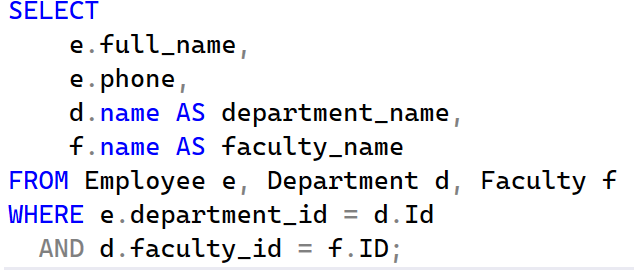
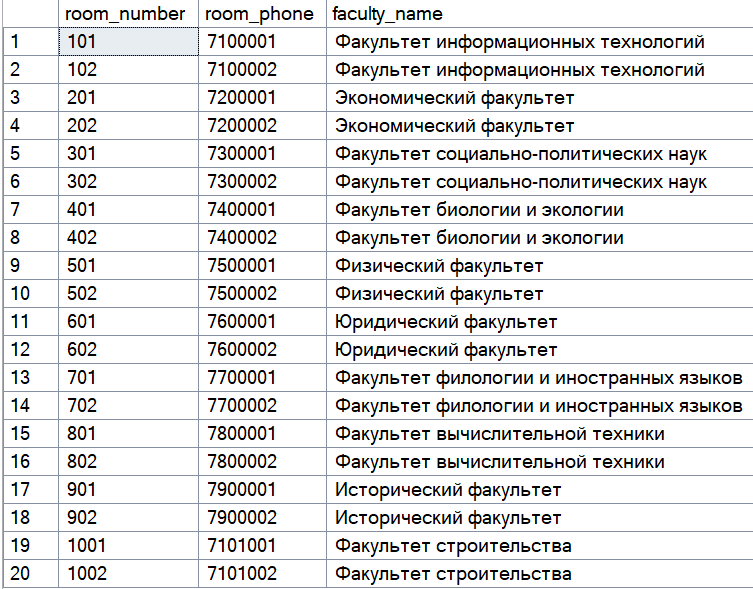
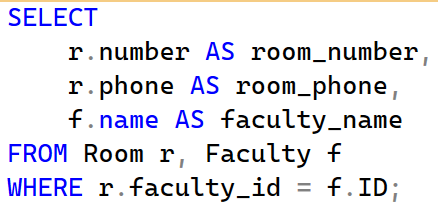
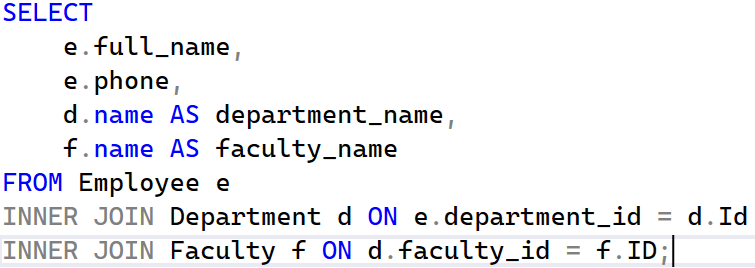




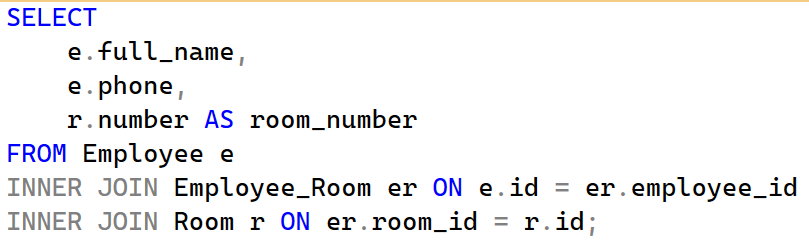
Таблица с номерами помещений, их телефонами и названиями факультетов



2.2. Реализовать запросы пункта 2.1 через внутреннее соединение inner join.

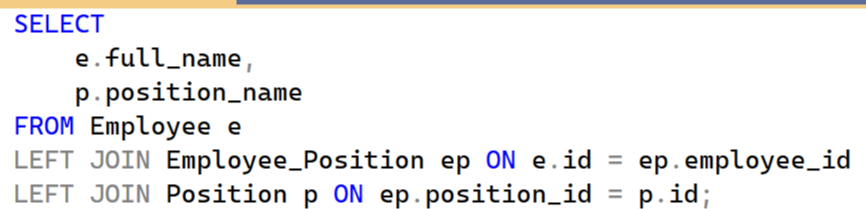


****

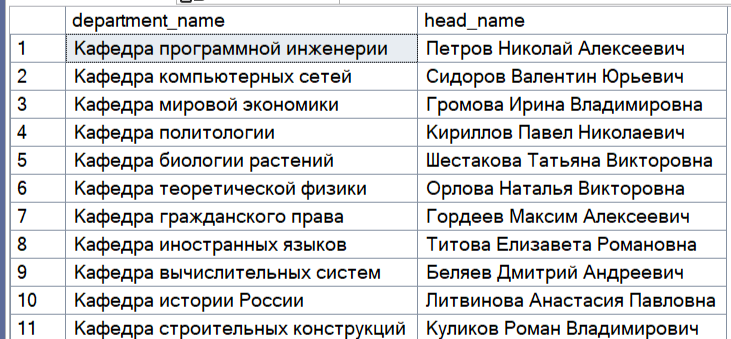
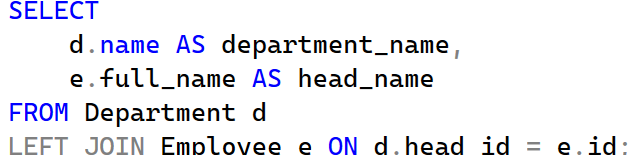
****

2.3. Левое внешнее соединение left join. Привести 2-3 запроса.

*Соединение сотрудников с должностями*

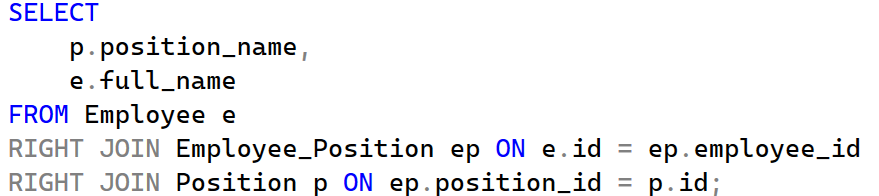


*Соединение кафедр с заведующими*

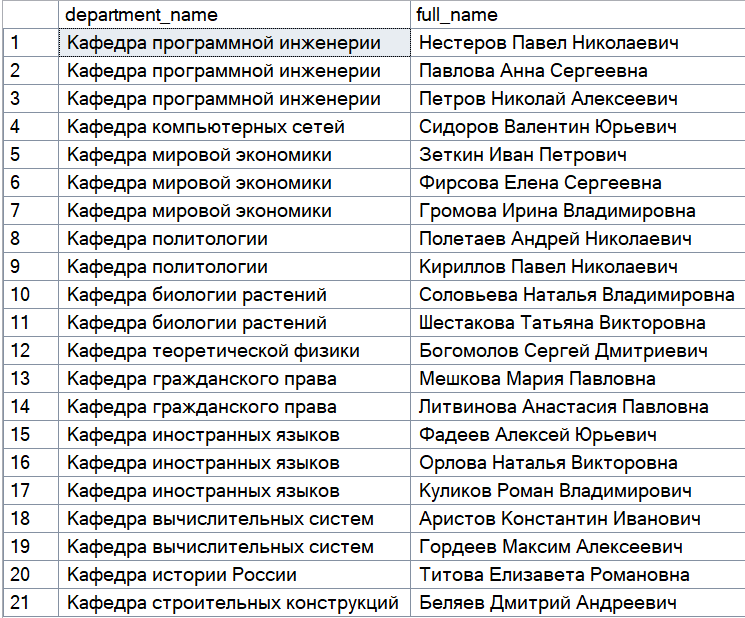
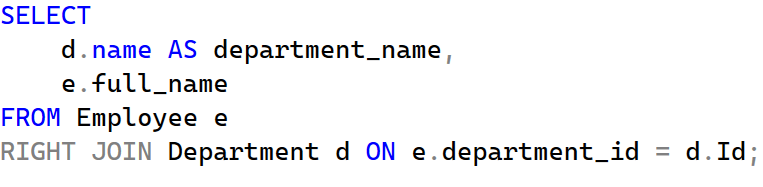
**

2.4. Правое внешнее соединение right join. Привести 2-3 запроса

*Соединение должностей с сотрудниками (все должности, даже незанятые)*

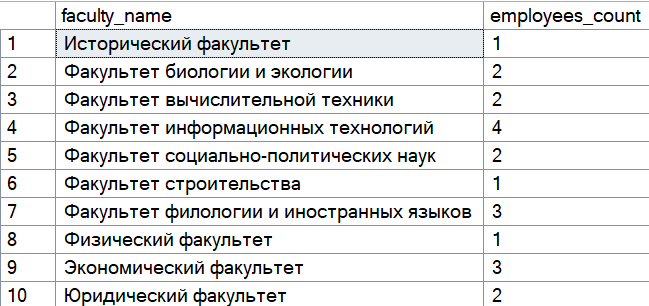
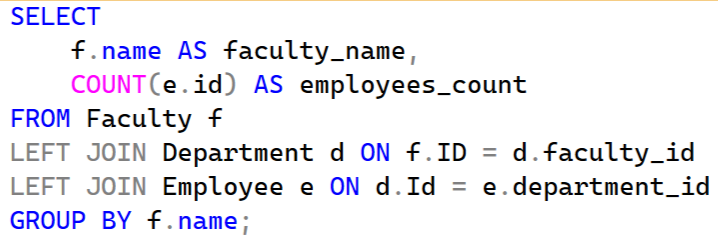
**

*Соединение кафедр с сотрудниками (все кафедры, даже без сотрудников)*

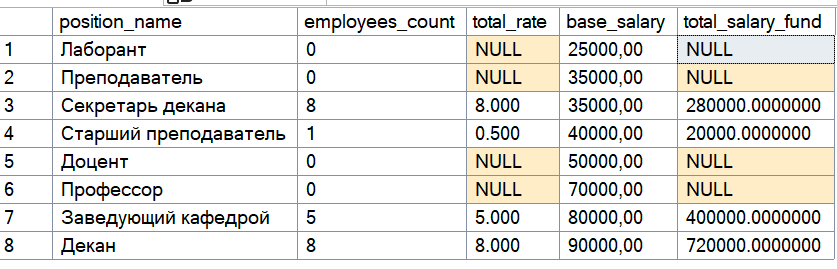
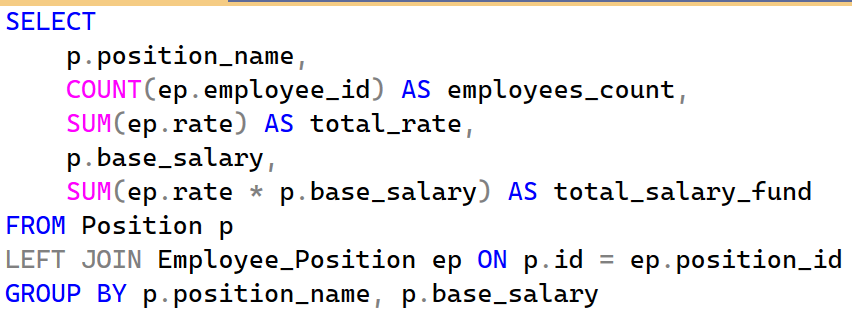
****

2.5. Привести примеры 2-3 запросов с использованием агрегатных функций и группировки.

*Количество сотрудников на каждом факультете*

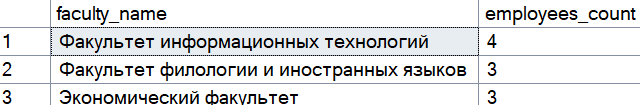
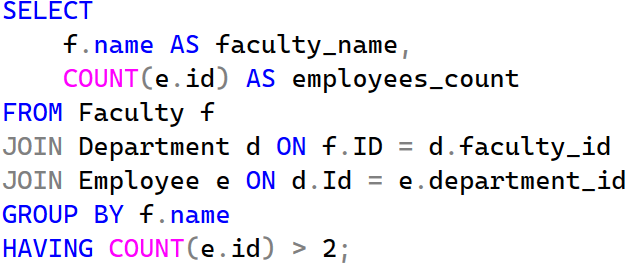


*Количество сотрудников и их суммарная зарплата на каждой должности*

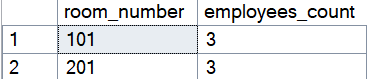
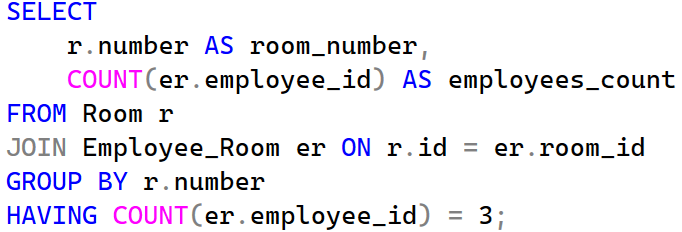


2.6. Привести примеры 2-3 запросов с использованием группировки и условия отбора групп (Having).

*Факультеты, на которых более двух сотрудников*

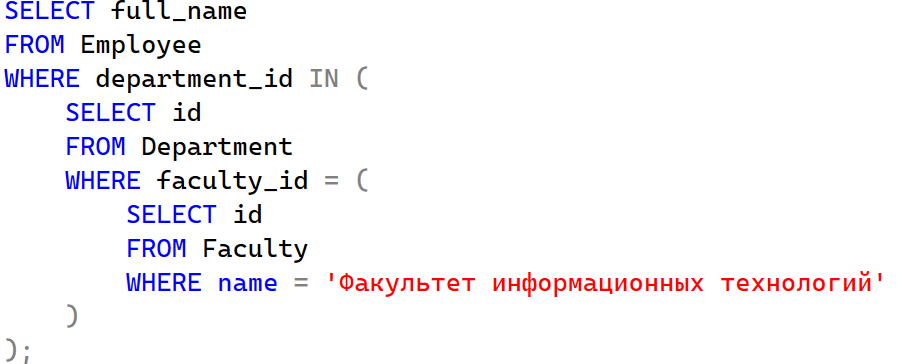
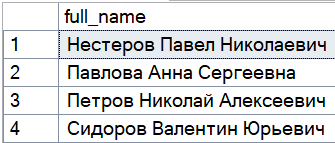
**

Помещения, в которых работают ровно 3 сотрудника

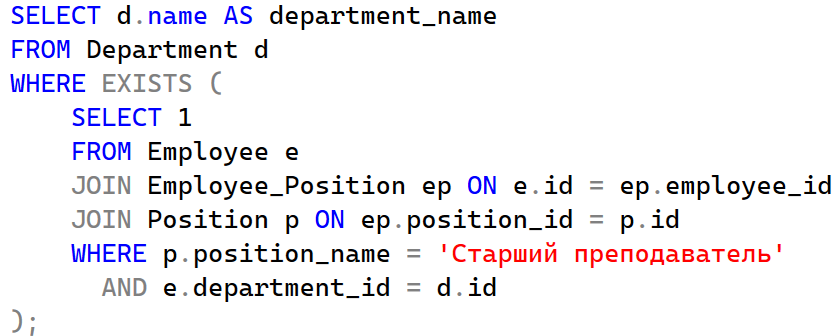
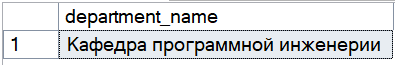


2.7. Привести примеры 3-4 вложенных (соотнесенных, c использованием IN, EXISTS) запросов.

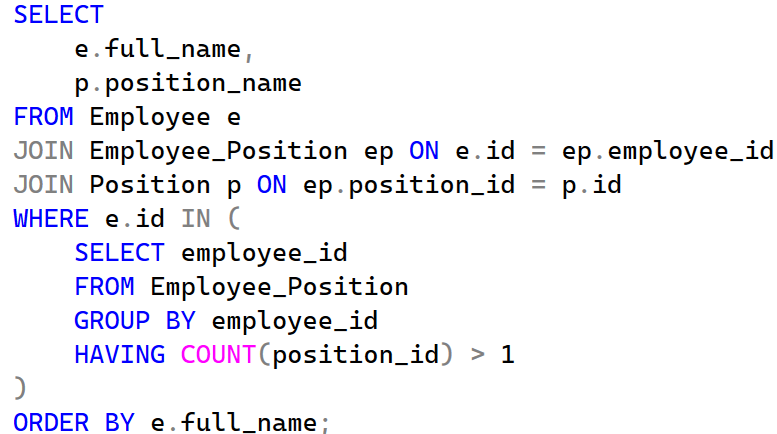
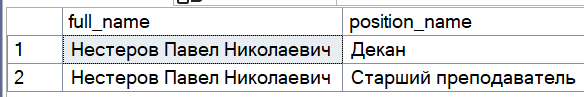
*Сотрудники, работающие на факультете "Факультет информационных технологий"*

*Кафедры, где есть хотя бы один профессор*

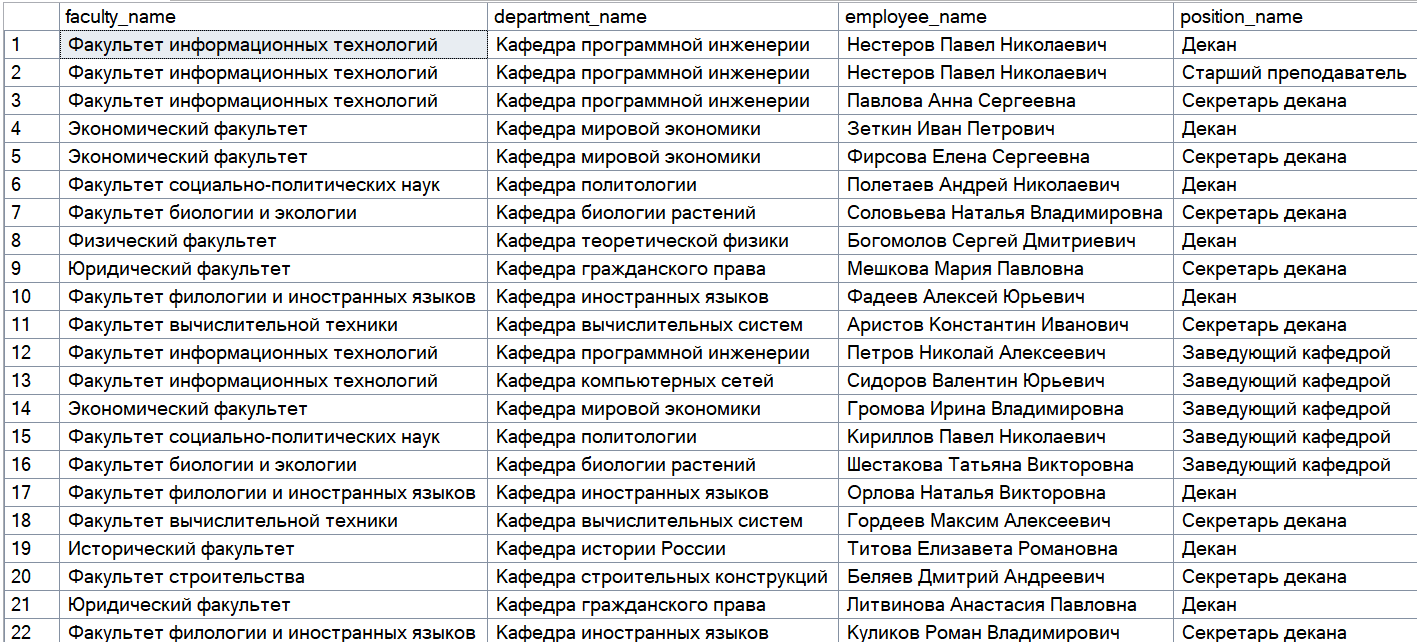
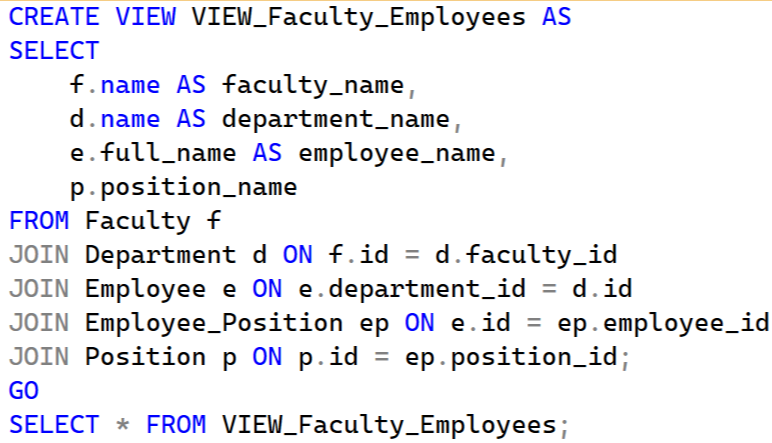
*Сотрудники, которые занимают более одной должности*

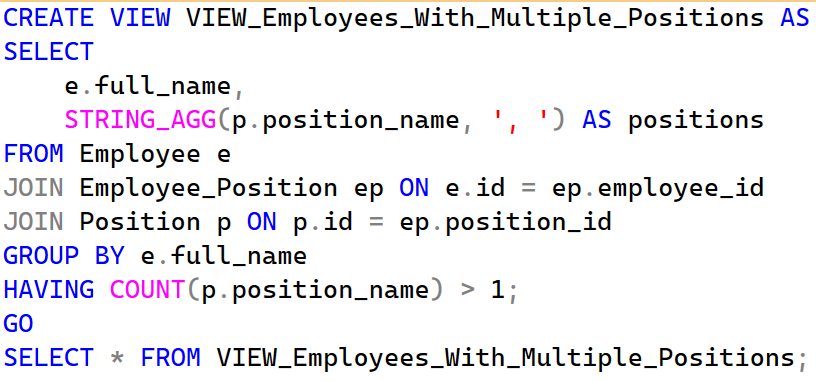
**3. Представления**

3.1 На основе любых запросов из п. 2 создать два представления (VIEW).

*все факультеты и сотрудники, которые на них работают (через кафедры)*

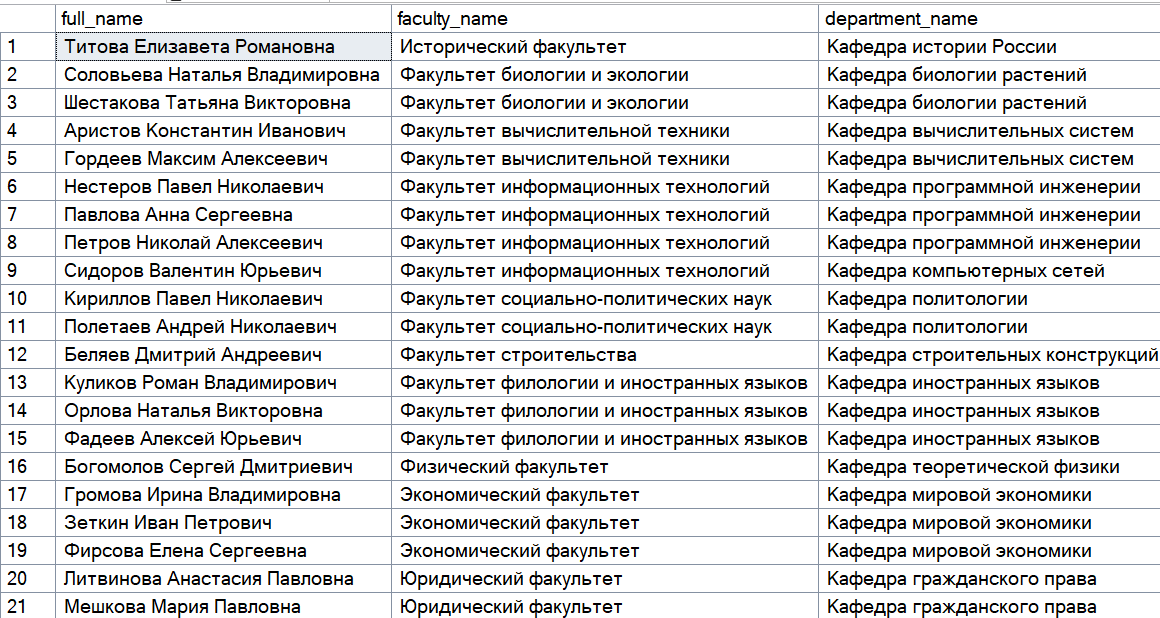
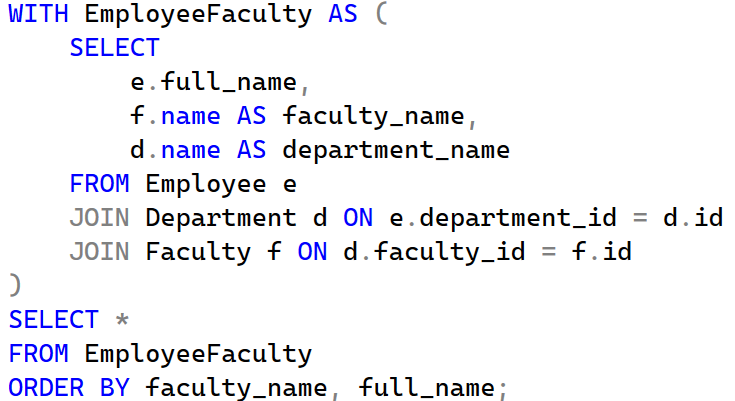


*сотрудники, у которых больше одной должности*

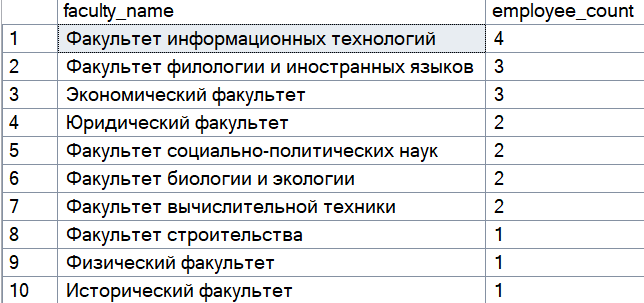
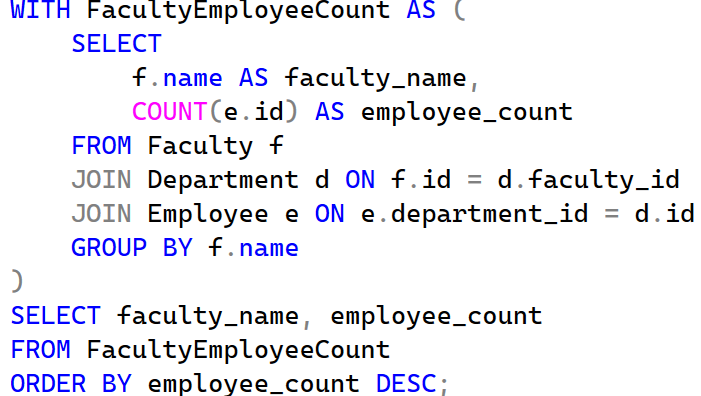


3.2 Привести примеры использования общетабличных выражений (СТЕ) (2-3 запроса)

*Имя, факультет и кафедра сотрудника*



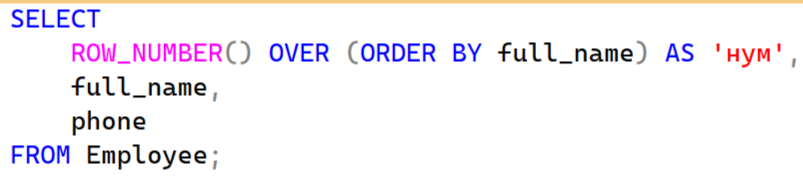
*Количество сотрудников по факультетам*



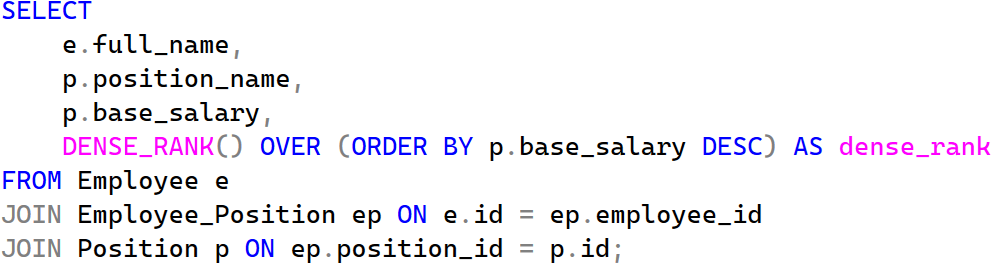
**4. Функции ранжирования**

4.1 Привести примеры 3-4 запросов с использованием ROW\_NUMBER, RANK, DENSE\_RANK (c  PARTITION BY и без)

*Присваиваем каждому сотруднику уникальный порядковый номер, сортируя по фамилии*

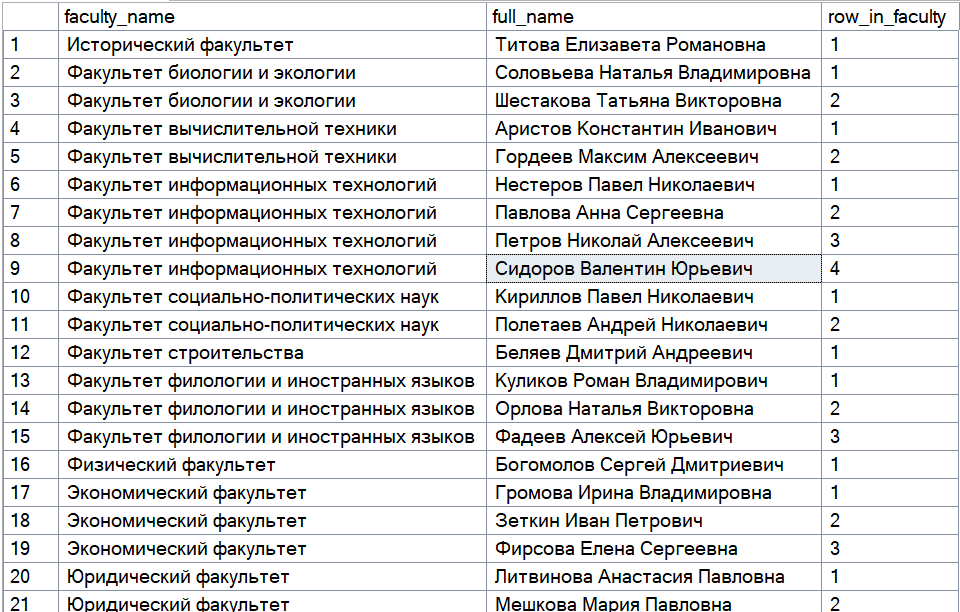
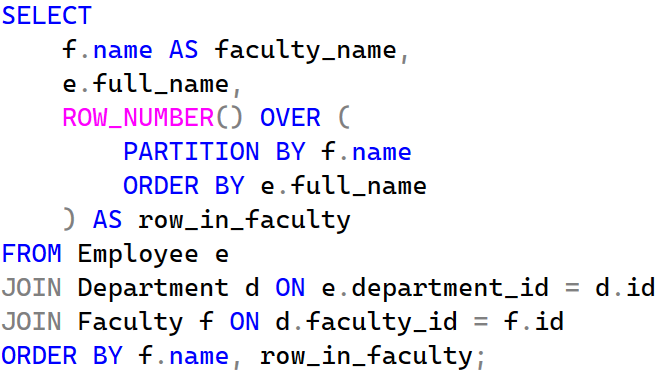


*Вывод всех сотрудников с учётом их должности и их последовательный рейтинг по зарплате*

**

**

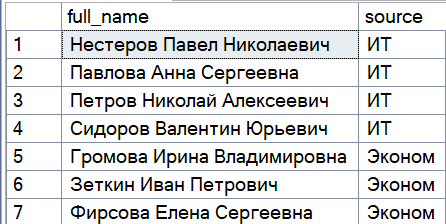
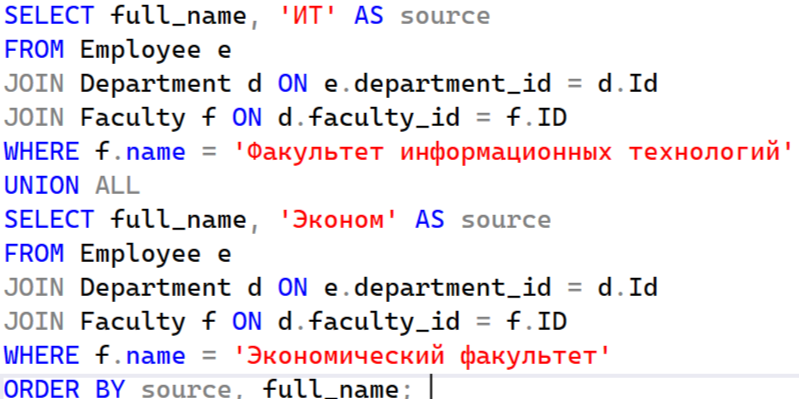
*Нумерация сотрудников внутри каждого факультета (нумерация начинается заново для каждого факультета)*

**

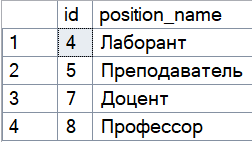
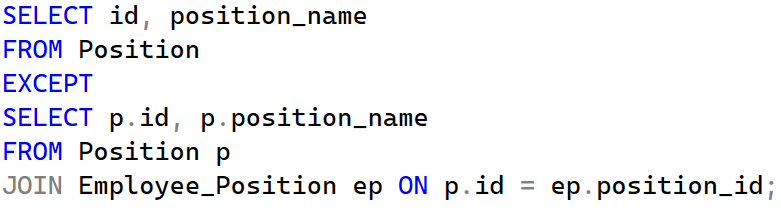
***5. Объединение, пересечение, разность***

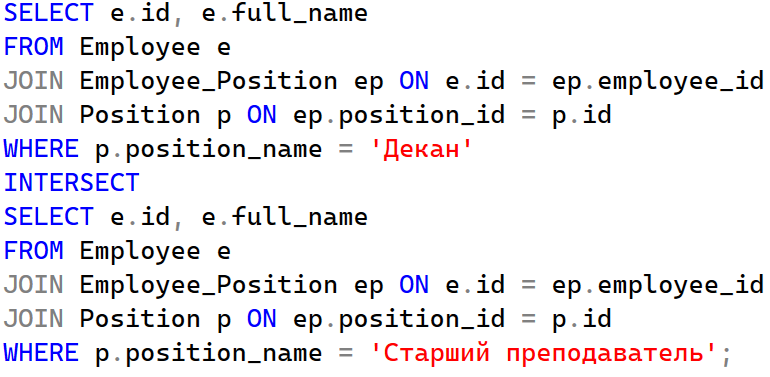
5.1 Привести примеры 3-4 запросов с использованием UNION / UNION ALL, EXCEPT, INTERSECT. Данные в одном из запросов отсортируйте по произвольному признаку.

*Объединение телефонов факультетов и кафедр без дубликатов*

**

*Должности, которые не занимает ни один сотрудник*

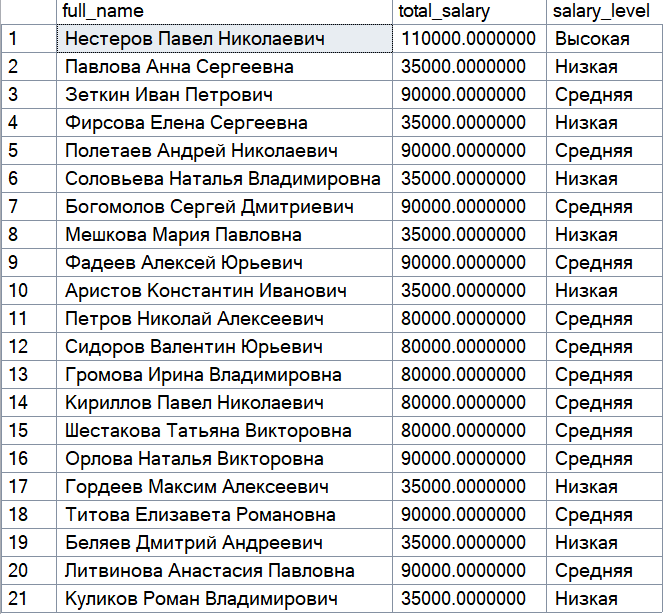
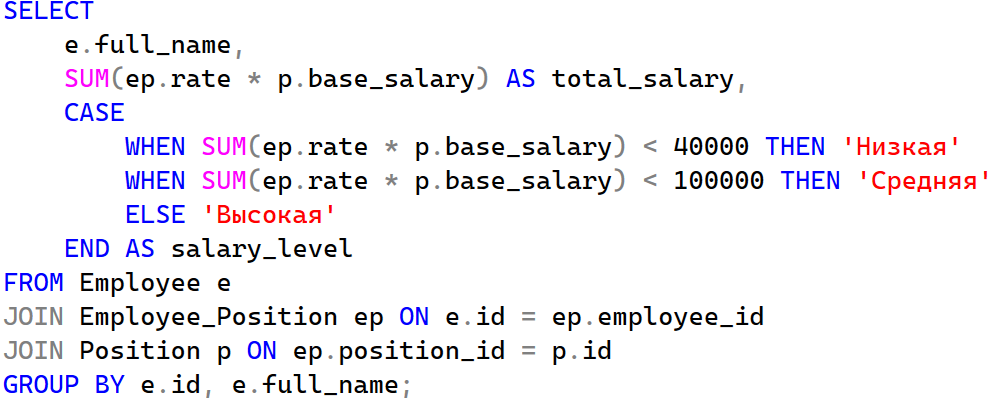
**

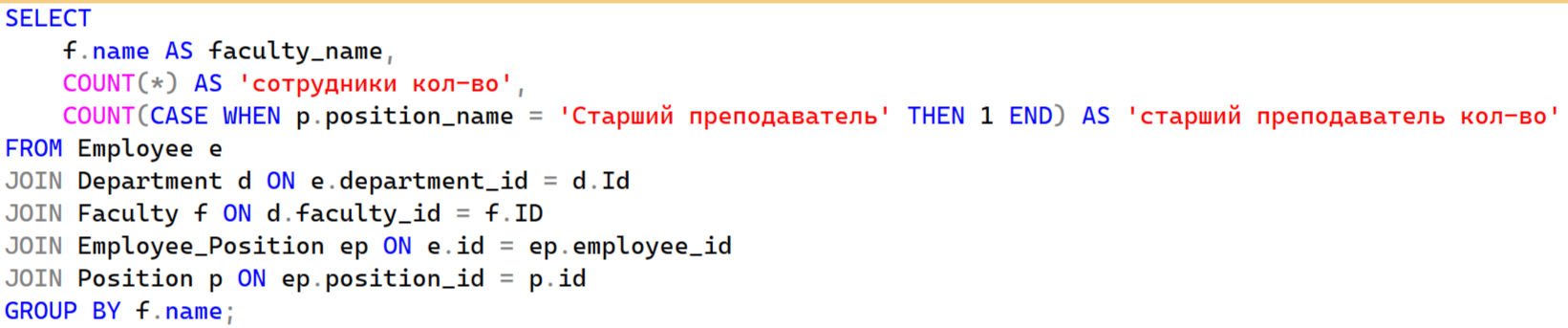
*Сотрудники, занимающие должность декана и старшего преподавателя*

***6. Использование CASE, PIVOT и UNPIVOT.***

6.1 Привести примеры получения сводных (итоговых) таблиц с использованием CASE

*Классификация зарплат*

******

******

******

6.2 Привести примеры получения сводных (итоговых) таблиц с использованием PIVOT и UNPIVOT.

