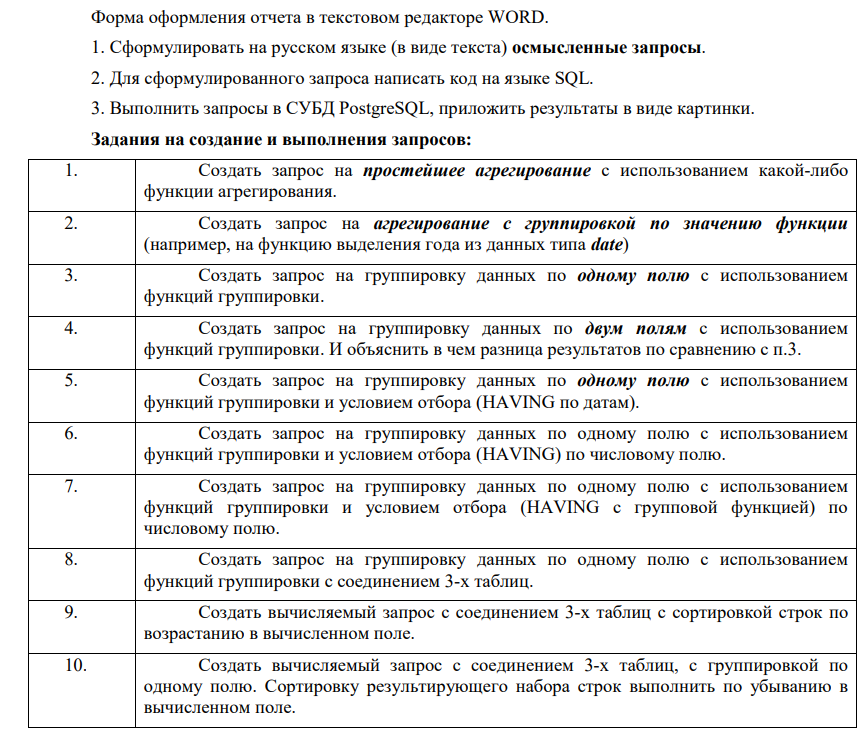
**Отчет по лабораторной работе 3.**

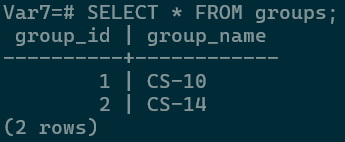
Выполнил: Золотухин Андрей Александрович КС-36

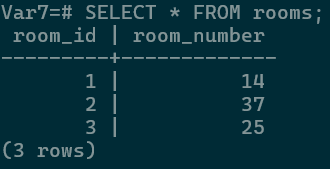
Преподаватель: Семёнов Геннадий Николаевич

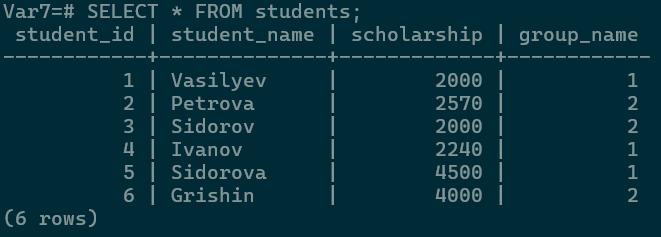
**ЗАДАНИЕ**

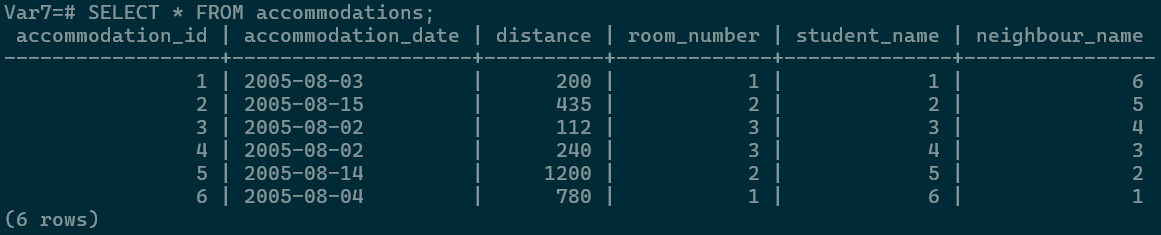
****

**Результат создания таблиц лабораторной работы 1**









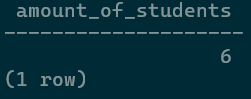
**Выполнение задания**

1. **Создать запрос на простейшее агрегирование с использованием какой-либо функции агрегирования:**

Формирование общего количества студентов:

SELECT COUNT(\*) AS amount\_of\_students

FROM public.students;



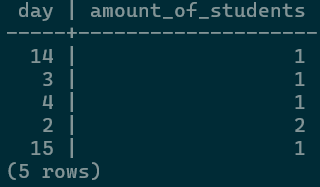
1. **Создать запрос на агрегирование с группировкой по значению функции (например, на функцию выделения года из данных типа date):**

Формирование количества студентов, размещенных в определенный день месяца:

SELECT EXTRACT(DAY FROM accommodation\_date) AS day, COUNT(\*) AS amount\_of\_students

FROM public.accommodations

GROUP BY day;



1. **Создать запрос на группировку данных по одному полю с использованием функций группировки:**

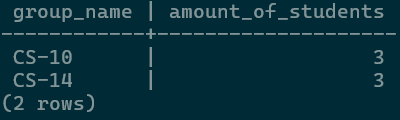
Формирование количества студентов из каждой группы:

SELECT groups.group\_name, COUNT(students.student\_id) AS amount\_of\_students

FROM public.groups

INNER JOIN public.students ON groups.group\_id = students.group\_name

GROUP BY groups.group\_name;



1. **Создать запрос на группировку данных по двум полям с использованием функций группировки. И объяснить, в чем разница результатов по сравнению с п.3:**

Формирование количества студентов из каждой группы и из каждой комнаты:

SELECT groups.group\_name, rooms.room\_number, COUNT(students.student\_id) AS amount\_of\_students

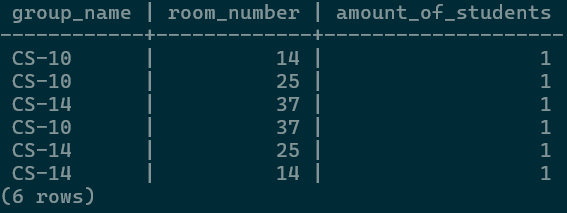
FROM public.groups

INNER JOIN public.students ON groups.group\_id = students.group\_name

INNER JOIN public.accommodations ON students.student\_id = accommodations.student\_name

INNER JOIN public.rooms ON rooms.room\_id = accommodations.room\_number

GROUP BY groups.group\_name, rooms.room\_number;



1. **Создать запрос на группировку данных по одному полю с использованием функций группировки и условием отбора (HAVING по датам):**

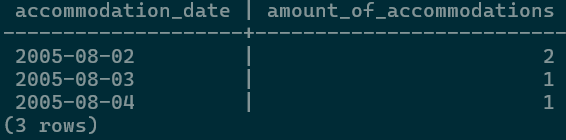
Формирование по определенным датам количества размещений, произошедших раньше 2005-08-14:

SELECT accommodation\_date, COUNT(\*) AS amount\_of\_students

FROM public.accommodations

GROUP BY accommodation\_date

HAVING accommodation\_date < '2005-08-14';



1. **Создать запрос на группировку данных по одному полю с использованием функций группировки и условием отбора (HAVING) по числовому полю:**

Формирование списка групп, в которых количество студентов больше 2:

SELECT groups.group\_name, COUNT(students.student\_id) AS amount\_of\_students

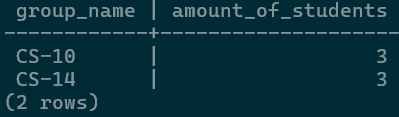
FROM public.groups

INNER JOIN public.students ON groups.group\_id = students.group\_name

WHERE students.scholarship < 4000

GROUP BY groups.group\_name

HAVING COUNT(students.student\_id) > 2;



1. **Создать запрос на группировку данных по одному полю с использованием функций группировки и условием отбора (HAVING с групповой функцией) по числовому полю:**

Формирование списка групп, в которых суммарная стипендия всех студентов больше 8600:

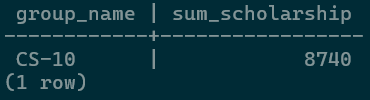
SELECT groups.group\_name, SUM(students.scholarship) AS sum\_scholarship

FROM public.groups

INNER JOIN public.students ON groups.group\_id = students.group\_name

GROUP BY groups.group\_name

HAVING SUM(students.scholarship) > 8600;



1. **Создать запрос на группировку данных по одному полю с использованием функций группировки с соединением 3-х таблиц:**

Формирование количества размещений и студентов из каждой группы:

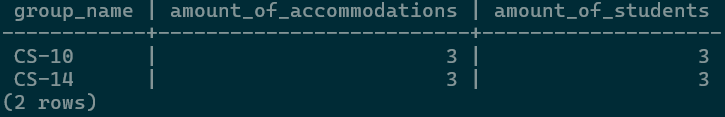
SELECT groups.group\_name, COUNT(accommodations.accommodation\_id) AS amount\_of\_accommodations, COUNT(students.student\_name) AS amount\_of\_students

FROM public.groups

INNER JOIN public.students ON groups.group\_id = students.group\_name

INNER JOIN public.accommodations ON students.student\_id = accommodations.student\_name

GROUP BY groups.group\_name;



1. **Создать вычисляемый запрос с соединением 3-х таблиц с сортировкой строк по возрастанию в вычисленном поле:**

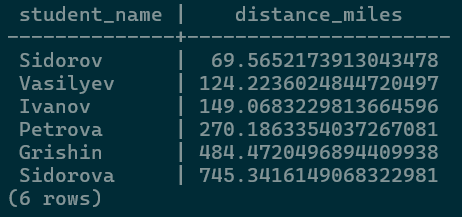
Формирование списка студентов в порядке возрастания дальности от общежития, выраженной в милях:

SELECT students.student\_name, (accommodations.distance / 1.61) AS distance\_miles

FROM public.students

INNER JOIN public.accommodations ON students.student\_id = accommodations.student\_name

ORDER BY average\_distance ASC;



1. **Создать вычисляемый запрос с соединением 3-х таблиц, с группировкой по одному полю. Сортировку результирующего набора строк выполнить по убыванию в вычисленном поле:**

Формирование списка групп в порядке убывания средней стипендии студентов из определенной группы, учитывая, что размер стипендии выводится, исходя из налоговой ставки (13%):

SELECT groups.group\_name, AVG(students.scholarship \* (1 - 0.13)) AS average\_scholarship

FROM public.groups

JOIN public.students ON groups.group\_id = students.group\_name

GROUP BY groups.group\_name

ORDER BY average\_scholarship DESC;

