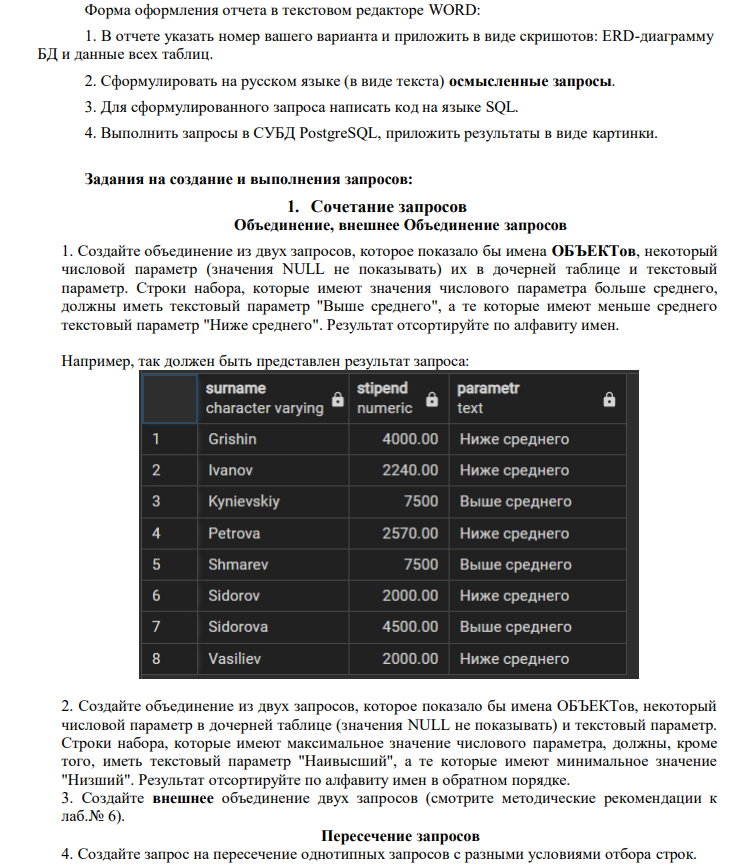
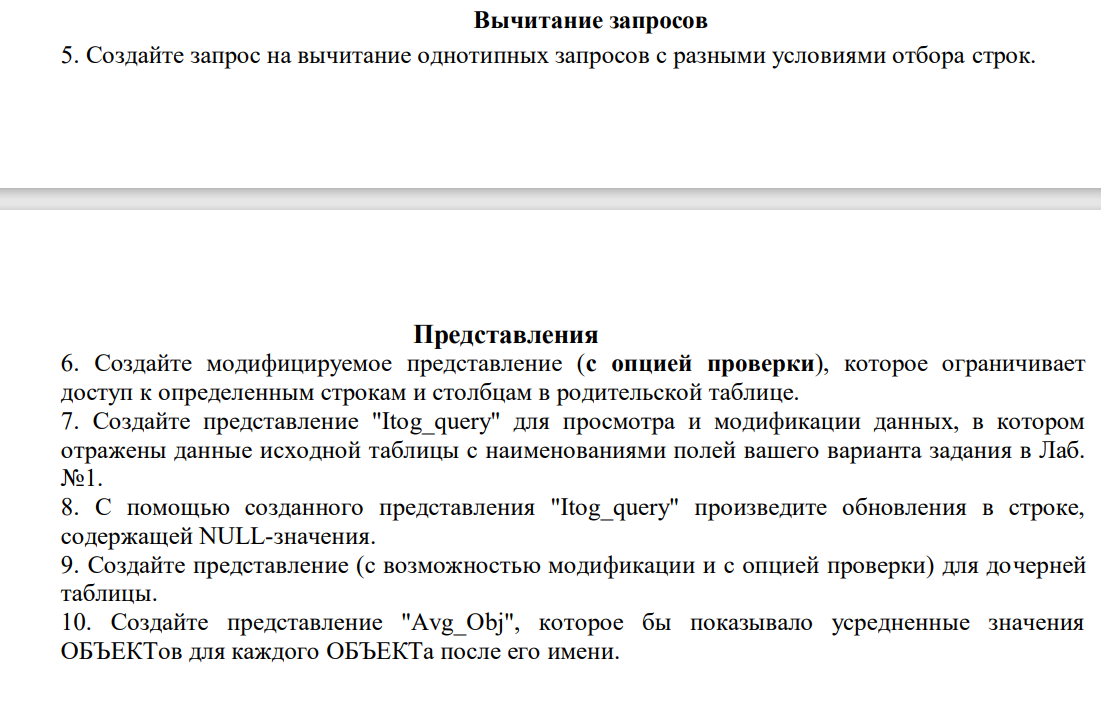
**Отчет по лабораторной работе 5.**

Выполнил: Золотухин Андрей Александрович КС-36

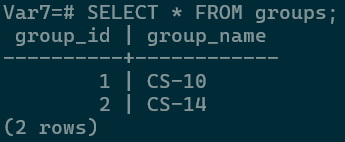
Преподаватель: Семёнов Геннадий Николаевич

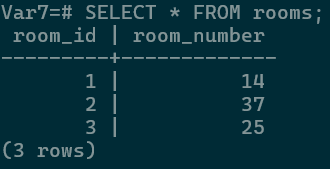
**ЗАДАНИЕ**

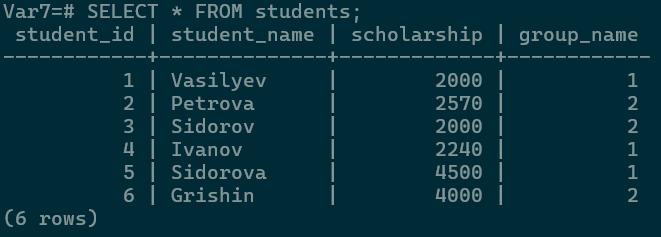


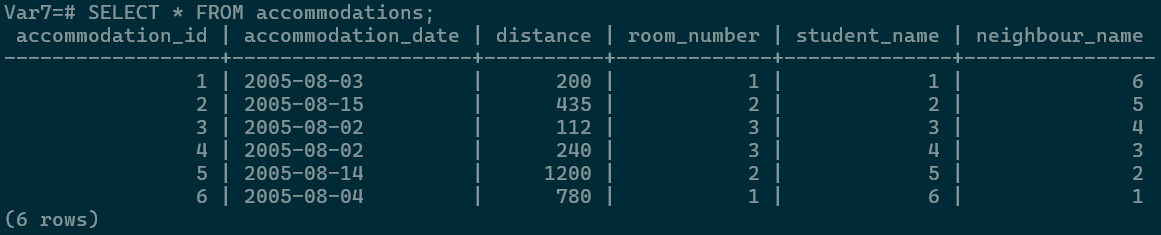


**Результат создания таблиц лабораторной работы 1**









**Выполнение задания**

**СОЧЕТАНИЕ ЗАПРОСОВ**

1. **Создайте объединение из двух запросов, которое показало бы имена ОБЪЕКТов, некоторый числовой параметр (значения NULL не показывать) их в дочерней таблице и текстовый параметр. Строки набора, которые имеют значения числового параметра больше среднего, должны иметь текстовый параметр “Выше среднего”, а те, которые имеют меньше среднего, текстовый параметр “Ниже среднего”. Результат отсортируйте по алфавиту имен:**

*Формирование списка студентов по алфавиту имен с отображением текстового параметра стипендии “Выше ..” или “Ниже средней”*:

SELECT students.student\_name AS name, students.scholarship AS stipend, 'Выше среднего' AS parametr

FROM public.students

WHERE students.scholarship > (SELECT AVG(students.scholarship) FROM public.students)

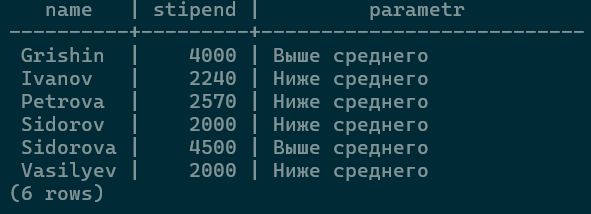
UNION

SELECT students.student\_name AS name, students.scholarship AS stipend, 'Ниже среднего' AS parametr

FROM public.students

WHERE students.scholarship <= (SELECT AVG(students.scholarship) FROM public.students)

ORDER BY name;



1. **Создайте объединение из двух запросов, которое показало бы имена ОБЪЕКТов, некоторый числовой параметр в дочерней таблице (значения NULL не показывать) и текстовый параметр. Строки набор, которые имеют максимальное значение числового параметра, должны, кроме того, иметь текстовый параметр “Наивысший”, а те, которые имеют минимальное значение, “Низший”. Результат отсортируйте по алфавиту имен в обратном порядке:**

*Формирование списка студентов по алфавиту имен в обратном порядке, у которых наивысшая или низшая стипендия*:

SELECT students.student\_name AS name, students.scholarship AS stipend, 'Наивысший' AS parametr

FROM public.students

WHERE students.scholarship = (SELECT MAX(students.scholarship) FROM public.students)

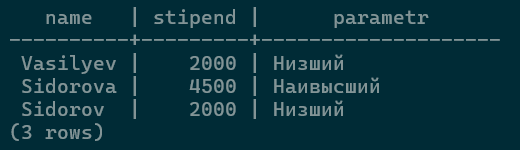
UNION

SELECT students.student\_name AS name, students.scholarship AS stipend, 'Низший' AS parametr

FROM public.students

WHERE students.scholarship = (SELECT MIN(students.scholarship) FROM public.students)

ORDER BY name DESC;



1. **Создайте внешнее объединение двух запросов:**

*Формирование списка студентов, их комнат, а также соседей, если они есть*:

SELECT s.student\_name AS student\_name, rooms.room\_number, students.student\_name AS neighbour\_name

FROM public.students s

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

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

LEFT OUTER JOIN public.students ON accommodations.neighbour\_name = students.student\_id

UNION

SELECT students.student\_name AS student\_name, rooms.room\_number, 'Нет соседа' AS neighbour\_name

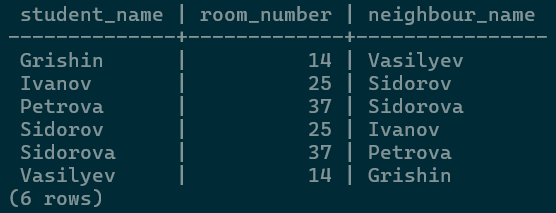
FROM public.students

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

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

WHERE accommodations.neighbour\_name IS NULL

ORDER BY student\_name;



**ПЕРЕСЕЧЕНИЕ ЗАПРОСОВ**

1. **Создайте запрос на пересечение однотипных запросов с разными условиями отбора строк:**

*Формирование списка студентов, которые имеют стипендию выше 3000 и которые из группы КС-10*:

SELECT students.student\_name

FROM public.students

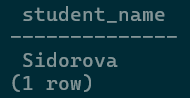
WHERE students.scholarship > 3000

INTERSECT

SELECT students.student\_name

FROM public.students

WHERE students.group\_name = 1;



**ВЫЧИТАНИЕ ЗАПРОСОВ**

1. **Создайте запрос на вычитание однотипных запросов с разными условиями отбора строк:**

*Формирование списка студентов, которые проживают в комнате номер 14, за исключением тех, кто из группы КС-14*:

SELECT students.student\_id, students.student\_name

FROM public.students

WHERE students.student\_id IN (SELECT accommodations.student\_name

FROM public.accommodations

WHERE accommodations.room\_number = 1

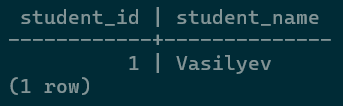
)

EXCEPT

SELECT students.student\_id, students.student\_name

FROM public.students

WHERE students.group\_name = 2;



**ПРЕДСТАВЛЕНИЯ**

1. **Создайте модифицируемое представление (с опцией проверки), которое ограничивает доступ к определенным строкам и столбцам в родительской таблице:**

*Формирование списка студентов из группы КС-10*:

CREATE OR REPLACE VIEW students\_view AS

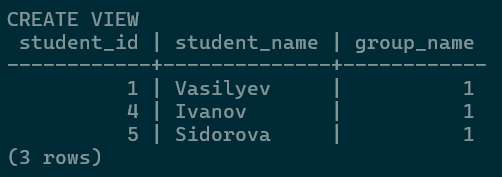
SELECT students.student\_id, students.student\_name, students.group\_name

FROM public.students

WHERE students.group\_name = 1

WITH CHECK OPTION;

SELECT \* FROM students\_view;



1. **Создайте представление "Itog\_query" для просмотра и модификации данных, в котором отражены данные исходной таблицы с наименованиями полей вашего варианта задания в Лаб. №1:**

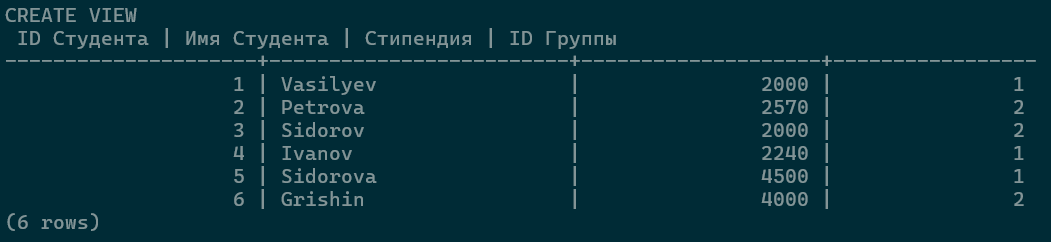
*Формирование списка студентов с их стипендиями и группы, в которых они обучаются*:

CREATE OR REPLACE VIEW Itog\_query AS

SELECT students.student\_id AS "ID Студента", students.student\_name AS "Имя Студента", students.scholarship AS "Стипендия", students.group\_name AS "ID Группы"

FROM public.students;

SELECT \* FROM Itog\_query;



1. **С помощью созданного представления "Itog\_query" произведите обновления в строке, содержащей NULL-значения:**

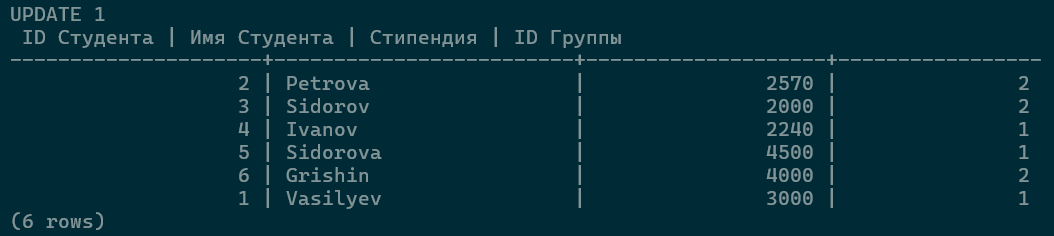
*Установка стипендии номиналом в 3000 рублей студенту, ID которого номер 1*:

UPDATE Itog\_query

SET "Стипендия" = 3000

WHERE "ID Студента" = 1;

SELECT \* FROM Itog\_query;



1. **Создайте представление (с возможностью модификации и с опцией проверки) для дочерней таблицы:**

*Формирование списка размещений по датам, дистанции студентов от общежития, комнатам*:

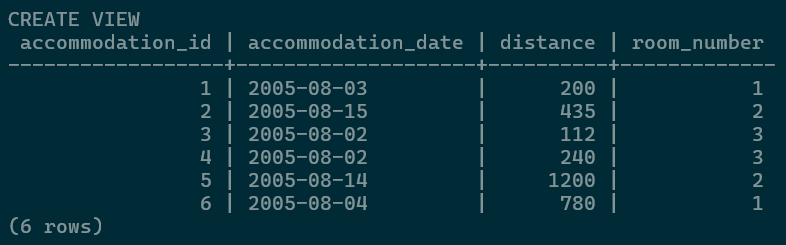
CREATE OR REPLACE VIEW accommodations\_view AS

SELECT accommodations.accommodation\_id, accommodations.accommodation\_date, accommodations.distance, accommodations.room\_number

FROM public.accommodations

WITH CHECK OPTION;

SELECT \* FROM accommodations\_view;



1. **Создайте представление "Avg\_Obj", которое бы показывало усредненные значения ОБЪЕКТов для каждого ОБЪЕКТа после его имени:**

*Формирование списка групп, в каждой из которых рассчитана средняя стипендия*:

CREATE OR REPLACE VIEW Avg\_Obj AS

SELECT students.group\_name AS "ID Группы", AVG(students.scholarship) AS "Средняя Стипендия"

FROM public.students

GROUP BY students.group\_name;

SELECT \* FROM Avg\_Obj;

