

Отчет по лабораторной работе 5.

Выполнил: Золотухин Андрей Александрович КС-36

Преподаватель: Семёнов Геннадий Николаевич

ЗАДАНИЕ

Форма оформления отчета в текстовом редакторе WORD:

1. В отчете указать номер вашего варианта и приложить в виде скришотов: ERD-диаграмму БД и данные всех таблиц.
2. Сформулировать на русском языке (в виде текста) **осмысленные запросы**.
3. Для сформулированного запроса написать код на языке SQL.
4. Выполнить запросы в СУБД PostgreSQL, приложить результаты в виде картинки.

Задания на создание и выполнения запросов:

1. Сочетание запросов

Объединение, внешнее Объединение 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.

Например, так должен быть представлен результат запроса:

	surname character varying	stipend numeric	parametr text
1	Grishin	4000.00	Ниже среднего
2	Ivanov	2240.00	Ниже среднего
3	Kynievskiy	7500	Выше среднего
4	Petrova	2570.00	Ниже среднего
5	Shmarev	7500	Выше среднего
6	Sidorov	2000.00	Ниже среднего
7	Sidorova	4500.00	Выше среднего
8	Vasiliev	2000.00	Ниже среднего

2. Создайте объединение из двух запросов, которое показало бы имена **ОБЪЕКТОВ**, некоторый числовой параметр в дочерней таблице (значения NULL не показывать) и текстовый параметр. Строки набора, 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.
3. Создайте **внешнее** объединение двух запросов (смотрите методические рекомендации к лаб. № 6).

Пересечение запросов

4. Создайте запрос на пересечение однотипных запросов с разными условиями отбора строк.

Вычитание запросов

5. Создайте запрос на вычитание однотипных запросов с разными условиями отбора строк.

Представления

6. Создайте модифицируемое представление (с опцией проверки), которое ограничивает доступ к определенным строкам и столбцам в родительской таблице.
7. Создайте представление "Itog_query" для просмотра и модификации данных, в котором отражены данные исходной таблицы с наименованиями полей вашего варианта задания в Лаб. №1.
8. С помощью созданного представления "Itog_query" произведите обновления в строке, содержащей NULL-значения.
9. Создайте представление (с возможностью модификации и с опцией проверки) для дочерней таблицы.
10. Создайте представление "Avg_Obj", которое бы показывало усредненные значения **ОБЪЕКТОВ** для каждого **ОБЪЕКТА** после его имени.

Результат создания таблиц лабораторной работы 1

```
Var7=# SELECT * FROM groups;
 group_id | group_name
-----+-----
          1 | CS-10
          2 | CS-14
(2 rows)
```

```
Var7=# SELECT * FROM rooms;
 room_id | room_number
-----+-----
          1 |          14
          2 |          37
          3 |          25
(3 rows)
```

```
Var7=# SELECT * FROM students;
 student_id | student_name | scholarship | group_name
-----+-----+-----+-----
          1 | Vasilyev     |          2000 |          1
          2 | Petrova      |          2570 |          2
          3 | Sidorov      |          2000 |          2
          4 | Ivanov       |          2240 |          1
          5 | Sidorova     |          4500 |          1
          6 | Grishin      |          4000 |          2
(6 rows)
```

```
Var7=# SELECT * FROM accommodations;
 accommodation_id | accommodation_date | distance | room_number | student_name | neighbour_name
-----+-----+-----+-----+-----+-----
          1 | 2005-08-03        |          200 |          1 |          1 |          6
          2 | 2005-08-15        |          435 |          2 |          2 |          5
          3 | 2005-08-02        |          112 |          3 |          3 |          4
          4 | 2005-08-02        |          240 |          3 |          4 |          3
          5 | 2005-08-14        |         1200 |          2 |          5 |          2
          6 | 2005-08-04        |          780 |          1 |          6 |          1
(6 rows)
```

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;

name	stipend	parametr
Grishin	4000	Выше среднего
Ivanov	2240	Ниже среднего
Petrova	2570	Ниже среднего
Sidorov	2000	Ниже среднего
Sidorova	4500	Выше среднего
Vasilyev	2000	Ниже среднего
(6 rows)		

2) 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;

name	stipend	parametr
Vasilyev	2000	Низший
Sidorova	4500	Наивысший
Sidorov	2000	Низший
(3 rows)		

3) 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;

student_name	room_number	neighbour_name
Grishin	14	Vasilyev
Ivanov	25	Sidorov
Petrova	37	Sidorova
Sidorov	25	Ivanov
Sidorova	37	Petrova
Vasilyev	14	Grishin
(6 rows)		

ПЕРЕСЕЧЕНИЕ ЗАПРОСОВ

- 4) 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;
```

```
student_name  
-----  
Sidorova  
(1 row)
```

ВЫЧИТАНИЕ ЗАПРОСОВ

- 5) 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;
```

student_id	student_name
1	Vasilyev
(1 row)	

ПРЕДСТАВЛЕНИЯ

- 6) 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;
```

CREATE VIEW		
student_id	student_name	group_name
1	Vasilyev	1
4	Ivanov	1
5	Sidorova	1
(3 rows)		

- 7) 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;
```

CREATE VIEW				
ID Студента	Имя Студента	Стипендия	ID Группы	
1	Vasilyev	2000	1	
2	Petrova	2570	2	
3	Sidorov	2000	2	
4	Ivanov	2240	1	
5	Sidorova	4500	1	
6	Grishin	4000	2	
(6 rows)				

8) 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;
```

UPDATE 1				
ID Студента	Имя Студента	Стипендия	ID Группы	
2	Petrova	2570	2	
3	Sidorov	2000	2	
4	Ivanov	2240	1	
5	Sidorova	4500	1	
6	Grishin	4000	2	
1	Vasilyev	3000	1	
(6 rows)				

9) 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;
```

CREATE VIEW				
accommodation_id	accommodation_date	distance	room_number	
1	2005-08-03	200	1	
2	2005-08-15	435	2	
3	2005-08-02	112	3	
4	2005-08-02	240	3	
5	2005-08-14	1200	2	
6	2005-08-04	780	1	
(6 rows)				

10) 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;
```

```
CREATE VIEW
```

```
  ID Группы | Средняя Стипендия
```

```
-----+-----  
          2 | 2856.6666666666666667  
          1 | 3246.6666666666666667
```

```
(2 rows)
```