

Отчет по лабораторной работе 4.

Выполнил: Золотухин Андрей Александрович КС-36

Преподаватель: Семёнов Геннадий Николаевич

ЗАДАНИЕ

Форма оформления отчета в текстовом редакторе WORD.

1. Сформулировать на русском языке (в виде текста) **осмысленные запросы**.
2. Для сформулированного запроса написать код на языке SQL.
3. Выполнить запросы в СУБД PostgreSQL, приложить результаты в виде картинки.

Задания на создание и выполнения запросов:

№	Тип запроса
	Оконные функции
1.	Придумать запрос на применение агрегирующих оконных функций
2.	Придумать запрос на применение ранжирующих оконных функций
	Объединение таблиц самих с собой
3.	Вывести пары атрибутов одной сущности при определенном условии.
4.	Тоже, что и в пункте 3 с устранением избыточности без оператора DISTINCT.
	Подзапросы
5.	Придумать однотабличный запрос, использующий подзапрос в условии отбора строк поле фразы WHERE.
6.	Придумать многотабличный запрос, использующий подзапрос в условии отбора строк поле фразы WHERE.
7.	Придумать запрос, использующий подзапрос с агрегатной функцией в условии отбора строк поле фразы WHERE.
8.	Придумать запрос на использование подзапросов, которые выдают много строк с помощью оператора IN
9.	Придумать запрос, использующий подзапрос в предложении HAVING
10.	Придумать запрос, использующий подзапрос в предложении FROM
	Соотнесенные подзапросы
11.	Придумать запрос на использование соотнесенного подзапроса, который выдает много строк с помощью оператора IN
12.	Придумать запрос на сравнение таблицы с собой

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 ОКОННЫЕ ФУНКЦИИ

1) Придумать запрос на применение агрегирующих оконных функций:

Формирование списка средней стипендии студентов по группам:

```
SELECT students.student_name, groups.group_name,
students.scholarship, AVG(students.scholarship)
OVER(PARTITION BY students.group_name) AS
avg_scholarship
FROM public.students
INNER JOIN public.groups ON students.group_name =
groups.group_id;
```



```

INNER JOIN public.accommodations AS accommodation_2
ON accommodation_1.room_number =
accommodation_2.room_number
INNER JOIN public.students AS a ON
accommodation_1.student_name = a.student_id
INNER JOIN public.students AS b ON
accommodation_2.student_name = b.student_id
WHERE a.student_id <> b.student_id;

```

student_1	student_2
Vasilyev	Grishin
Grishin	Vasilyev
Petrova	Sidorova
Sidorova	Petrova
Sidorov	Ivanov
Ivanov	Sidorov
(6 rows)	

4) Вывести пары атрибутов одной сущности при определенном условии с устранением избыточности без оператора DISTINCT:

Формирование списка пар студентов, которые проживают в одной комнате, без повторов:

```

SELECT a.student_name AS student_1, b.student_name AS
student_2
FROM public.accommodations AS accommodation_1
INNER JOIN public.accommodations AS accommodation_2
ON accommodation_1.room_number =
accommodation_2.room_number
INNER JOIN public.students AS a ON
accommodation_1.student_name = a.student_id
INNER JOIN public.students AS b ON
accommodation_2.student_name = b.student_id
WHERE a.student_id < b.student_id;

```

student_1	student_2
Vasilyev	Grishin
Petrova	Sidorova
Sidorov	Ivanov
(3 rows)	

ПОДЗАПРОСЫ

- 5) Придумать однотабличный запрос, использующий подзапрос в условии отбора строк поле фразы **WHERE**:

Формирование списка студентов, у которых стипендия выше, чем у Иванова:

```
SELECT student_name, scholarship
FROM public.students
WHERE scholarship > (SELECT scholarship
                     FROM public.students
                     WHERE student_name = 'Ivanov');
```

student_name	scholarship
Petrova	2570
Sidorova	4500
Grishin	4000
(3 rows)	

- 6) Придумать многотабличный запрос, использующий подзапрос в условии отбора строк поле фразы **WHERE**:

Формирование списка номеров комнат, где живет хотя бы один студент, получающий стипендию больше 2500:

```
SELECT DISTINCT rooms.room_number
FROM public.accommodations
INNER JOIN public.rooms ON accommodations.room_number
= rooms.room_id
WHERE student_name IN (SELECT student_id
                       FROM public.students
                       WHERE scholarship > 2500);
```

```

room_number
-----
          37
          14
(2 rows)

```

7) Придумать запрос, использующий подзапрос с агрегатной функцией в условии отбора строк поле фразы WHERE:

Формирование списка студентов, у которых стипендия выше средней стипендии среди всех студентов:

```
SELECT student_name, scholarship
FROM public.students
WHERE scholarship > (SELECT AVG(scholarship)
                     FROM public.students);
```

```

student_name | scholarship
-----+-----
Sidorova    |          4500
Grishin     |          4000
(2 rows)

```

8) Придумать запрос на использовании подзапросов, которые выдают много строк с помощью оператора IN:

Формирование списка студентов, которые живут с соседями, имеющими стипендию более 2500:

```
SELECT student_name  
FROM public.students  
WHERE student_id IN (SELECT neighbour_name  
                     FROM public.accommodations  
                     WHERE neighbour_name IN (SELECT student_id  
                                              FROM public.students  
                                              WHERE scholarship >  
                                              2500));
```

student_name
Petrova
Sidorova
Grishin
(3 rows)

9) Придумать запрос, использующий подзапрос в предложении **HAVING**:

Формирование списка групп, в которых количество студентов такое же, какое количество студентов, получающих стипендию больше 2300:

```
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
HAVING COUNT(students.student_id) = (SELECT COUNT(*)
FROM public.students
WHERE scholarship > 2300);
```

group_name	amount_of_students
CS-10	3
CS-14	3
(2 rows)	

10) Придумать запрос, использующий подзапрос в предложении **FROM**:

Формирование списка студентов, которые находятся в группе, где есть хотя бы один студент со стипендией выше 4000:

```
SELECT students.student_name, students.scholarship
FROM public.students
INNER JOIN (SELECT group_name
FROM public.students
WHERE scholarship > 4000) AS high_scholarship_groups
ON students.group_name = high_scholarship_groups.group_name;
```

student_name	scholarship
Vasilyev	2000
Ivanov	2240
Sidorova	4500
(3 rows)	

СООТНЕСЕННЫЕ ПОДЗАПРОСЫ

- 11) Придумать запрос на использование соотнесенного подзапроса, который выдает много строк с помощью оператора IN:

Формирование списка номеров комнат, в которых живут студенты, получающие стипендию выше 2400:

```
SELECT DISTINCT room_number
FROM accommodations
WHERE student_name IN (SELECT student_id
                        FROM public.students
                        WHERE scholarship > 2400);
```

room_number
37
14

(2 rows)

- 12) Придумать запрос на сравнение таблицы с собой:

Формирование списка пар студентов, которые получают одинаковую стипендию:

```
SELECT a.student_name AS student1, b.student_name AS student2
FROM public.students AS a
INNER JOIN public.students AS b ON a.scholarship =
b.scholarship
WHERE a.student_id <> b.student_id;
```


student1	student2
Vasilyev	Sidorov
Sidorov	Vasilyev

(2 rows)