

## Отчет по лабораторной работе 3.

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:**

1.	Создать запрос на <i>простейшее агрегирование</i> с использованием какой-либо функции агрегирования.
2.	Создать запрос на <i>агрегирование с группировкой по значению функции</i> (например, на функцию выделения года из данных типа <i>date</i> )
3.	Создать запрос на группировку данных по <i>одному полю</i> с использованием функций группировки.
4.	Создать запрос на группировку данных по <i>двум полям</i> с использованием функций группировки. И объяснить в чем разница результатов по сравнению с п.3.
5.	Создать запрос на группировку данных по <i>одному полю</i> с использованием функций группировки и условием отбора (HAVING по датам).
6.	Создать запрос на группировку данных по одному полю с использованием функций группировки и условием отбора (HAVING) по числовому полю.
7.	Создать запрос на группировку данных по одному полю с использованием функций группировки и условием отбора (HAVING с групповой функцией) по числовому полю.
8.	Создать запрос на группировку данных по одному полю с использованием функций группировки с соединением 3-х таблиц.
9.	Создать вычисляемый запрос с соединением 3-х таблиц с сортировкой строк по возрастанию в вычисленном поле.
10.	Создать вычисляемый запрос с соединением 3-х таблиц, с группировкой по одному полю. Сортировку результирующего набора строк выполнить по убыванию в вычисленном поле.

### 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

- 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;
```

```
 amount_of_students
-----
                  6
(1 row)
```

**2) 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;
```

day	amount_of_students
14	1
3	1
4	1
2	2
15	1
(5 rows)	

**3) 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;
```

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

**4) 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;
```

group_name	room_number	amount_of_students
CS-10	14	1
CS-10	25	1
CS-14	37	1
CS-10	37	1
CS-14	25	1
CS-14	14	1
(6 rows)		

**5) 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';
```

accommodation_date	amount_of_accommodations
2005-08-02	2
2005-08-03	1
2005-08-04	1
(3 rows)	

**6) 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;
```

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

**7) 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;
```

group_name	sum_scholarship
CS-10	8740
(1 row)	

**8) 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;
```

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

**9) 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;
```

student_name	distance_miles
Sidorov	69.5652173913043478
Vasilyev	124.2236024844720497
Ivanov	149.0683229813664596
Petrova	270.1863354037267081
Grishin	484.4720496894409938
Sidorova	745.3416149068322981
(6 rows)	

- 10) 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;
```

group_name	average_scholarship
CS-10	2534.6000000000000000
CS-14	2485.3000000000000000
(2 rows)	