

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное
учреждение высшего образования
“Национальный исследовательский университет ИТМО”

Факультет инфокоммуникационных технологий

ЛАБОРАТОРНАЯ РАБОТА №2

ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ, ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В POSTGRESQL.

по дисциплине:

«Проектирование и реализация баз данных»

Выполнил:

студент 2 курса ИКТ

группы К3241

Кондратьев Алексей

Санкт-Петербург

2022

Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и просмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

Индивидуальное задание: Вариант 11. БД «Автомастерская»

Описание предметной области: Сеть автомастерских осуществляет ремонт автомашин, используя для этих целей штат мастеров и свои мастерские. Стоимость ремонта включает цену деталей и стоимость работы. Заработная плата мастеров составляет 50% стоимости работы.

БД должна содержать следующий минимальный набор сведений: Табельный номер сотрудника. ФИО сотрудника. Должность. Разряд мастера. Специализация. Адрес автомастерской. Дата заказа. Гос. Номер автомобиля. Марка. Мощность автомобиля. Год выпуска. Цвет автомобиля. Дата принятия в ремонт. Плановая дата окончания ремонта. Фактическая дата окончания ремонта. Вид ремонта. Стоимость вида ремонта. Название детали. Цена детали. Марка и модель автомобиля. Страна производителя. Госномер автомобиля. ФИО владельца. Номер телефона владельца. E-mail владельца.

Схема базы данных (из ЛР 1):

Query Editor
Query History

```

1 select name_worker from order_composition
2 join "order" on "order".order_number = order_composition.order_number
3 join car on car.state_number = "order".state_number
4 join worker on worker.id_worker = order_composition.id_worker
5 where brand = 'toyota' group by worker.id_worker
6 having count(*) = (select count(*) from order_composition
7 join "order" on "order".order_number = order_composition.order_number
8 join car on car.state_number = "order".state_number
9 join worker on worker.id_worker = order_composition.id_worker
10 where brand = 'toyota' group by worker.id_worker order by count(*) desc limit 1)

```

Data Output
Explain
Messages
Notifications

	name_worker character varying (20)
1	Vladov Vlad
2	Olegov Oleg

- Определить тех владельцев автомобилей, которых всегда обслуживает один и тот же механик. Вывести фамилии механика и его постоянного клиента.

Query Editor
Query History

```

1 select client.client_name, worker.name_worker
2 from worker, order_composition, client, "order"
3 where worker.id_worker=order_composition.id_worker
4 and client.client_name="order".client_name
5 and "order".order_number = order_composition.order_number
6 and client.client_name not in (select client.client_name
7 from worker, order_composition, client, "order"
8 where worker.id_worker=order_composition.id_worker
9 and client.client_name="order".client_name
10 and "order".order_number = order_composition.order_number
11 group by client.client_name, worker.name_worker
12 having count(client.client_name)=1
13 order by client.client_name, worker.name_worker)
14 group by client.client_name, worker.name_worker
15 order by client.client_name, worker.name_worker

```

Data Output
Explain
Messages
Notifications

	client_name character varying (20)	name_worker character varying (20)
1	Petrov_Petr	Olegov Oleg

- Вывести фамилии механиков, которые не выполняли работы в срок и количество дней просрочки выполнения заказа.

Query Editor

```
1 select name_worker from order_composition
2 join "order" on "order".order_number = order_composition.order_number
3 join client on client.client_name = "order".client_name
4 join worker on worker.id_worker = order_composition.id_worker
5 where date_to_return_car - actual_return_car < 0
```

Data Output Explain Messages Notifications

	name_worker character varying (20)	
1	Olegov Oleg	
2	Ivanov Ivan	

- Вывести данные механика, который выполнял все виды ремонта за прошедшую неделю.

Query Editor Query History

```
1 select worker.name_worker from order_composition
2 join worker on worker.id_worker = order_composition.id_worker
3 join repair_service on repair_service.repair_type = order_composition.repair_type
4 join price on price.repair_type = repair_service.repair_type
5 where payment_end_date between '2020-09-01' and '2020-09-30'
6 group by worker.name_worker
7 having count(*) = 3
```

Data Output Explain Messages Notifications




	name_worker character varying (20)	
1	Vladov Vlad	

- Сколько заработал каждый мастер за прошедший месяц?

Query Editor Query History

```
1 select worker.id_worker, price.value from order_composition
2 join worker on worker.id_worker = order_composition.id_worker
3 join repair_service on repair_service.repair_type = order_composition.repair_type
4 join price on price.repair_type = repair_service.repair_type
5 where date_to_return_car between '2020-09-01' and '2020-09-30'
6 group by worker.id_worker, price.id_price
```

Data Output Explain Messages Notifications

	 id_worker integer	 value integer	
1	1	5000	
2	2	5000	
3	2	8000	
4	2	6000	
5	3	6000	

Вывести данные владельцев автомобилей, которые обращались в ремонт больше одного раза.

Query Editor

Query History

```
1 select client.client_name, client.mail, client.mobile_number from order_composition
2 join "order" on "order".order_number = order_composition.order_number
3 join client on client.client_name = "order".client_name
4 group by client.client_name
5 having count(*) > 1
```

Data Output


Explain

Messages

Notifications

	<div>client_name</div> <div>[PK] character varying (20)</div>	<div>mail</div> <div>character varying (40)</div>	<div>mobile_number</div> <div>character varying (12)</div>	
1	Kondratev_Aleksey	kondratev-alesha69@mail.ru	+79108405702	
2	Varlamov_Ivan	var-ivan@gmail.com	+79227212123	
3	Morozova_Maria	morozova-masha@mail.ru	+79108882244	
4	Petrov_Petr	petr-petrov@mail.ru	+79108405522	

- За каждый день просрочки выполнения заказа механику назначается штраф в размере 5%. Рассчитать штраф каждого механика за прошедший месяц.

 car_service/postgres@PostgreSQL 13 ▾

Query Editor

Query History

```
1 select name_worker, -1 * (date_to_return_car - actual_return_car) * 0.05 * price.value from order_composition
2 join "order" on "order".order_number = order_composition.order_number
3 join client on client.client_name = "order".client_name
4 join worker on worker.id_worker = order_composition.id_worker
5 join repair_service on repair_service.repair_type = order_composition.repair_type
6 join price on price.repair_type = repair_service.repair_type
7 where date_to_return_car - actual_return_car < 0 and date_to_return_car between '2020-09-01' and '2020-09-30'
8
```

Data Output

Explain

Messages

Notifications

	name_worker character varying (20)	?column? numeric	
1	Ivanov Ivan	500.00	

2. Представления. Выполнить запросы на создание представлений согласно индивидуальному заданию, часть 3. В отчете привести формулировку запроса, команду, скриншот выполнения запроса и просмотр содержимого представления;
- для заказчиков (фамилию механика и модель автомобиля, которую он ремонтирует чаще всего)

Query EditorQuery History

```
1 select client.client_name, worker.name_worker
2 from worker, order_composition, client, "order"
3 where worker.id_worker=order_composition.id_worker
4       and client.client_name="order".client_name
5       and "order".order_number = order_composition.order_number
6 and client.client_name not in (select client.client_name
7                               from worker, order_composition, client, "order"
8                               where worker.id_worker=order_composition.id_worker
9                                     and client.client_name="order".client_name
10                                    and "order".order_number = order_composition.order_number
11                                   group by client.client_name, worker.name_worker
12                                   having count(client.client_name)=1
13                                   order by client.client_name, worker.name_worker)
14 group by client.client_name, worker.name_worker
15 order by client.client_name, worker.name_worker
```

Data OutputExplainMessagesNotifications

	client_name character varying (20)	name_worker character varying (20)
1	Petrov_Petr	Olegov Oleg

Diagram illustrating the query execution plan. It shows the flow of data from tables (worker, order_composition, client, order) through various operations (Hash, Hash inner join, Aggregate, Sort, Limit) to the final output.

Successfully run. Total que

- для менеджеров (рассчитать премию все механикам, которые за прошедший месяц все свои заказы выполнили своевременно - 10% от зарплаты).

Query Editor Query History

```
1 select name_worker, (sum(price.value) + worker.salary ) * 1.1
2 from order_composition
3 join "order" on "order".order_number = order_composition.order_number
4 join client on client.client_name = "order".client_name
5 join worker on worker.id_worker = order_composition.id_worker
6 join repair_service on repair_service.repair_type = order_composition.repair_type
7 join price on price.repair_type = repair_service.repair_type
8 where date_to_return_car - actual_return_car > 0 and date_to_return_car between '2020-09-01' and '2020-09-30'
9 group by name_worker, worker.salary
```

Data Output Explain Messages Notifications

	name_worker character varying (20)	?column? numeric
1	Olegov Oleg	50600.0
2	Vladov Vlad	64900.0

```
graph LR
    OC[order_composition] --> J1[Nested Loop Inner Join]
    O[order] --> J1
    J1 --> J2[Nested Loop Inner Join]
    C[client] --> J2
    J2 --> J3[Nested Loop Inner Join]
    W[worker] --> J3
    J3 --> J4[Nested Loop Inner Join]
    RS[repair_service] --> J4
    J4 --> J5[Nested Loop Inner Join]
    P[price] --> J5
    J5 --> J6[Nested Loop Inner Join]
    RO[repair_order] --> J6
    J6 --> S[Sort]
    S --> A[Aggregate]
```

3. Запросы на модификацию данных. Выполнить запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов (составить самостоятельно). В отчете привести формулировку запроса, команду, скриншот до и после выполнения запроса. Учитывается сложность запроса;
- INSERT с использованием подзапросов;

```
Query Editor  Query History
1 insert into client(client_name, mail, mobile_number)
2 values ('Barbosov_Bober', 'barbosov-bober@mail.ru', '+79112333225')
```

– UPDATE с использованием подзапросов;

```
Copy  Copy to Query Editor
update public.detail_composition set id_dc = 1
where composition_code = 1;
```

Выводы:

В результате выполненной работы:

- были выполнены запросы и представления
- были изучены и использованы запросы на модификацию данных
- были изучены и реализованы индексы, хотя мне не удалось сделать так, чтобы запрос работал через индекс, а не таблицу. Но самое главное усвоено.