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

Факультет инфокоммуникационных технологий

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

тема: Запросы на выборку и модификацию данных, представления и индексы в PostgreSQL.
дисциплина: «ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ БАЗ ДАННЫХ»

Выполнил:
Костенников Данил Вячеславович
Группа K3241
Проверила:
Говорова Марина Михайловна

Санкт-Петербург
2022

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

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL 1X, pgAdmin 4.

Практическое задание:

1. Создать запросы и представления на выборку данных к базе данныхPostgreSQL (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 историюзапросов
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.

Выполнение:

БД 14 вариант «такси»

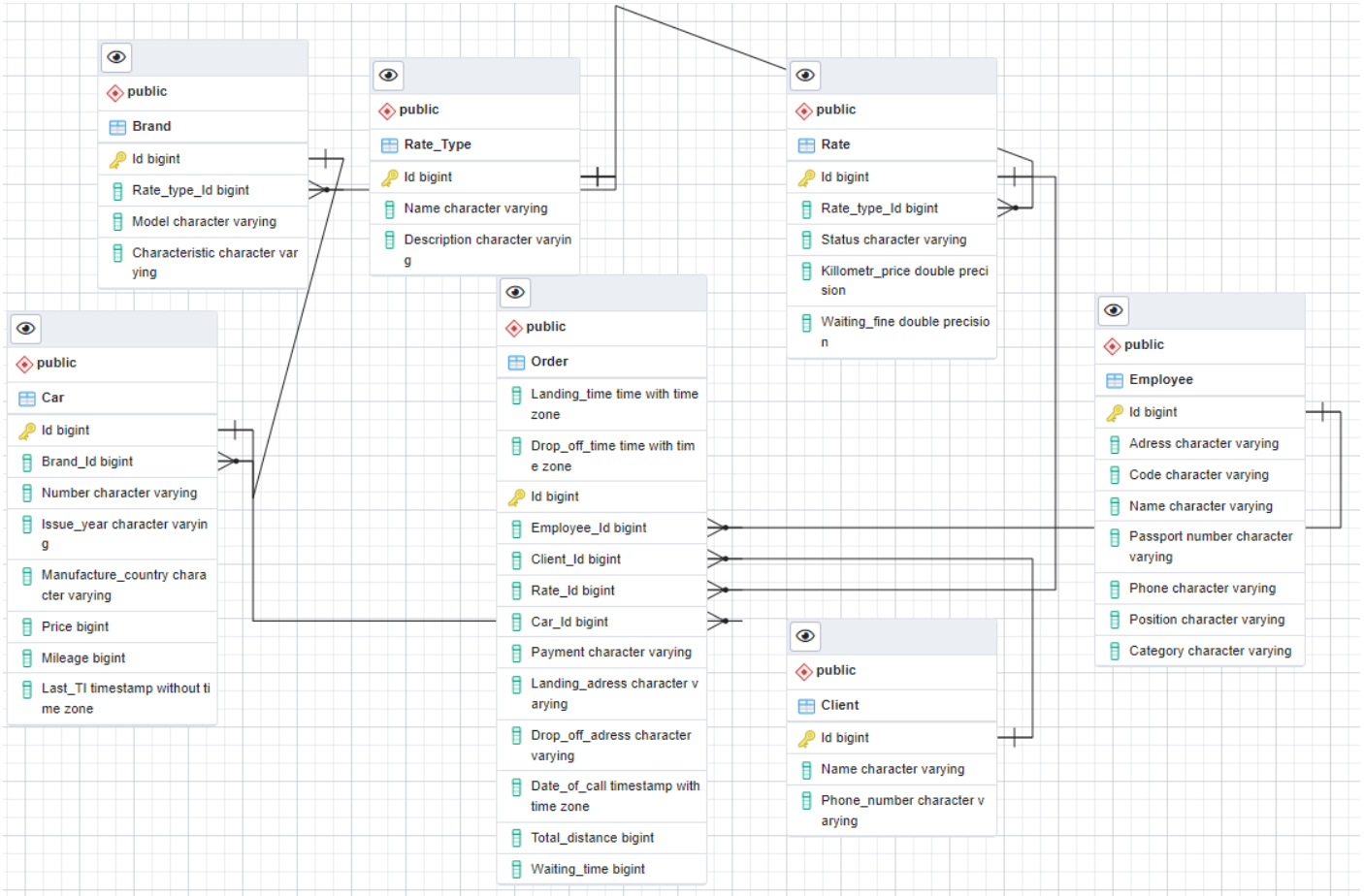


Рис 1. Схема БД

Запросы:

1. Вывести данные о водителе, который чаще всех доставляет пассажиров на заданную улицу:

Query Editor

```
SELECT "Employee"."Id", "Employee"."Adress", "Employee"."Code", "Employee"."Name", "Employee"."Passport number",
    "Employee"."Phone", "Employee"."Position", "Employee"."Category", Count("Employee"."Id")
    FROM public."Employee", public."Order"
WHERE public."Order"."Drop_off_adress" LIKE 'Alp str%'
AND "Employee"."Id" = "Order"."Employee_Id"
Group By ("Employee"."Id")
Order By count Desc Limit 1
```

Notifications | Query History

Recorded time	Event	Process ID	Payload
No data found			

Data Output | Explain | Messages

	Id [PK] bigint	Adress character varying	Code character varying	Name character varying	Passport number character varying	Phone character varying	Position character varying	Category character varying	count bigint
1	2	Spb	001	Dima	0002	+7904000002	Driver	B	

2. Вывести данные об автомобилях, которые имеют пробег более 250 тысяч. километров и которые не проходили ТО в текущем году.

Query Editor

```
SELECT "Id", "Brand_Id", "Number", "Issue_year", "Manufacture_country", "Price", "Mileage", "Last_TI"
    FROM public."Car"
    WHERE "Mileage" > 250000
    AND date_part('year', "Last_TI") < date_part('year', Current_date)
```

Notifications | Query History

Recorded time	Event	Process ID	Pay
No data found			

Data Output | Explain | Messages

	Id [PK] bigint	Brand_Id bigint	Number character varying	Issue_year character varying	Manufacture_country character varying	Price bigint	Mileage bigint	Last_TI timestamp without time zone
1	4	2	000AA003	2021	South Korea	8500000	250001	2021-02-15 00:00:00

3. Сколько раз каждый пассажир воспользовался услугами таксопарка?

Query Editor

```
1  SELECT "Client"."Id", COUNT("Client"."Id")
2      FROM public."Client", public."Order"
3  WHERE "Client"."Id" = "Order"."Client_Id"
4  Group By "Client"."Id"
```

Notifications | Query History

Recorded time

Data Output | Explain | Messages

	Id [PK] bigint	count bigint
1	5	2
2	1	2
3	3	2

4. Вывести данные пассажира, который воспользовался услугами таксопарка максимальное число раз.

Query Editor

```
1  SELECT "Client"."Id", COUNT("Client"."Id")
2      FROM public."Client", public."Order"
3  WHERE "Client"."Id" = "Order"."Client_Id"
4  Group By "Client"."Id"
5  Order By count Desc Limit 1
```

Notifications | Query History

Recorded time

Data Output | Explain | Messages

	Id [PK] bigint	count bigint
1	5	2

5. Вывести данные о водителе, который ездит на самом дорогом автомобиле.

Query Editor

```
SELECT "Employee"."Id", "Employee"."Adress", "Employee"."Code", "Employee"."Name", "Employee"."Passport number",
       "Employee"."Phone", "Employee"."Position", "Employee"."Category"
    FROM public."Employee", public."Car", public."Order"
    WHERE "Employee"."Id" = "Order"."Employee_Id"
    AND "Car"."Id" = "Order"."Car_Id"
    AND "Car"."Price" = (SELECT MAX ("Car"."Price") FROM public."Car")
```

Notifications | Query History

ecorded time	Event	Process ID	Payload
No data found			

Data Output | Explain | Messages

	Id [PK] bigint	Adress character varying	Code character varying	Name character varying	Passport number character varying	Phone character varying	Position character varying	Category character varying
1	4	Spb	004	Alena	0004	+7904000004	Driver	B

6. Вывести данные пассажира, который всегда ездит с одним и тем же водителем

Query Editor

```
SELECT "Client"."Id", "Client"."Name", "Client"."Phone_number"
    FROM public."Client", public."Employee", public."Order"
    WHERE "Order"."Employee_Id" = "Employee"."Id"
    AND "Order"."Client_Id" = "Client"."Id"
Group By ("Client"."Id")
Having Count (Distinct "Order"."Employee_Id") = 1
```

Notifications | Query History

Recorded time	Event

Data Output | Explain | Messages

	Id [PK] bigint	Name character varying	Phone_number character varying

7. Какие автомобили имеют пробег больше среднего пробега для своей марки.

```
SELECT "Id", "Number", "Car"."Brand_Id", "Issue_year", "Manufacture_country", "Price", "Mileage", "Last_TI"
    FROM public."Car",
    (SELECT "Brand_Id", round(avg("Car"."Mileage")) as avg_mileage
    From public."Car"
     Group by "Brand_Id") as t_mileage
     Where "Car"."Brand_Id" = t_mileage."Brand_Id"
     And "Mileage" >= t_mileage.avg_mileage;
```

Notifications | Query History

Recorded time	Event	Process ID	P
No data found			

Data Output | Explain | Messages

	Id [PK] bigint	Number character varying	Brand_Id bigint	Issue_year character varying	Manufacture_country character varying	Price bigint	Mileage bigint	Last_TI timestamp without time zone
1	1	000AA000	1	2022	Germany	5000000	1000	2022-01-16 00:00:00
2	4	000AA003	2	2021	South Korea	8500000	250001	2021-02-15 00:00:00

Представления:

1. содержащее сведения о незанятых на данный момент водителях;

Query Editor

```
Create view public."free_drivers" AS
 Select Distinct "Employee"."Id", "Employee"."Name", "Employee"."Adress", "Employee"."Code",
        "Employee"."Passport number", "Employee"."Phone", "Employee"."Category"
From public."Employee", public."Order"
Where "Employee"."Position" = 'Driver'
and  "Employee"."Id" = "Order"."Employee_Id"
and ("Order"."Landing_time" < current_time and "Order"."Drop_off_time" < current_time)
or ("Order"."Landing_time" > current_time and "Order"."Drop_off_time" > current_time)

```

Notifications | Query History

Recorded time	Event	Process ID
No data found		

Data Output | Explain | Messages

```
CREATE VIEW

Query returned successfully in 101 msec.
```

Query Editor

```
1  SELECT * FROM public.free_drivers
2
```

Notifications | Query History

Recorded time	Event	Process ID
No data found		

Data Output | Explain | Messages

	Id bigint	Name character varying	Adress character varying	Code character varying	Passport number character varying	Phone character varying	Category character varying
1	3	Kutay	Spb	003	0003	+7904000003	B
2	4	Alena	Spb	004	0004	+7904000004	B
3	1	Danil	Spb	000	0001	+7904000000	B
4	2	Dima	Spb	001	0002	+7904000002	B

2. зарплата всех водителей за вчерашний день.

Query Editor

```
1  Create view public."salary_for_yeasterday" as
2      Select sum("Rate"."Killometr_price" * "Order"."Total_distance" + "Rate"."Waiting_fine" * "Order"."Waiting_time")
3      From public."Order", public."Rate"
4      Where "Order"."Rate_Id" = "Rate"."Id"
5                 And "Order"."Date_of_call" = current_date - 1
```

Notifications | Query History

Recorded time	Event	Process ID
No data found		

Data Output | Explain | Messages

```
CREATE VIEW

Query returned successfully in 131 msec.
```

Query Editor

```
1  SELECT * FROM public.salary_for_yeasterday
2
```

Notifications | Query History

Recorded time

Data Output | Explain | Messages

	sum double precision
1	1235

Запросы на модификацию данных:

1. INSERT. Добавить в таблицу Rate тариф “Rush day”

Запрос:

Query Editor

```
1  INSERT INTO public."Rate"(
2      "Id", "Rate_type_Id", "Status", "Killometr_price", "Waiting_fine")
3      VALUES ((SELECT MAX("Rate"."Id") + 1 from public."Rate"), 1,'Rush day', 50, 10);
```

Notifications | Query History

Recorded time	Event	Process ID
		No data found

Data Output | Explain | Messages

```
INSERT 0 1

Query returned successfully in 58 msec.
```

До:

Query Editor

```
1  SELECT * FROM public."Rate"
2  ORDER BY "Id" ASC
```

Notifications | Query History

Recorded time	Event
	No data f

Data Output | Explain | Messages

	Id [PK] bigint	Rate_type_Id bigint	Status character varying	Killometr_price double precision	Waiting_fine double precision
1	1	1	Night	25	20
2	2	2	Day	10	10

После:

Query Editor

```
1  SELECT * FROM public."Rate"
2  ORDER BY "Id" ASC
```

Notifications | Query History

Recorded time	Event
	No data f

Data Output | Explain | Messages

	Id [PK] bigint	Rate_type_Id bigint	Status character varying	Killometr_price double precision	Waiting_fine double precision
1	1	1	Night	25	20
2	2	2	Day	10	10
3	3	1	Rush day	50	10

2. UPDATE, повысить цены на ночной и дневной тариф в два раза

До:

Query Editor

```
1  SELECT * FROM public."Rate"
2  ORDER BY "Id" ASC
```

Notifications Query History

Recorded time	Event
	No data f

Data Output Explain Messages

	Id [PK] bigint	Rate_type_Id bigint	Status character varying	Killometr_price double precision	Waiting_fine double precision
1	1	1	Night	25	20
2	2	2	Day	10	10
3	3	1	Rush day	50	10

Запрос:

Query Editor

```
1  UPDATE public."Rate"
2      SET "Killometr_price"= "Killometr_price" * 2, "Waiting_fine"= "Waiting_fine" * 2
3      WHERE "Status" = 'Day' or "Status" = 'Night'
```

Notifications Query History

Recorded time	Event	Process ID
	No data found	

Data Output Explain Messages

```
UPDATE 2

Query returned successfully in 75 msec.
```

После:

Query Editor

```
1  SELECT * FROM public."Rate"
2  ORDER BY "Id" ASC
```

Notifications | Query History

Recorded time	Event
	No data

Data Output | Explain | Messages

	Id [PK] bigint	Rate_type_Id bigint	Status character varying	Killometr_price double precision	Waiting_fine double precision
1	1	1	Night	50	40
2	2	2	Day	20	20
3	3	1	Rush day	50	10

3. DELETE, удалить тариф “Rush day”

До:

Query Editor

```
1  SELECT * FROM public."Rate"
2  ORDER BY "Id" ASC
```

Notifications | Query History

Recorded time	Event
	No data

Data Output | Explain | Messages

	Id [PK] bigint	Rate_type_Id bigint	Status character varying	Killometr_price double precision	Waiting_fine double precision
1	1	1	Night	50	40
2	2	2	Day	20	20
3	3	1	Rush day	50	10

Запрос:

Query Editor

```
1  DELETE FROM public."Rate"
2      WHERE "Status" = 'Rush day';
```

Notifications | Query History

Recorded time

Data Output | Explain | Messages

```
DELETE 1

Query returned successfully in 61 msec.
```

После:

Query Editor

```
1  SELECT * FROM public."Rate"
2  ORDER BY "Id" ASC
```

Notifications | Query History

Recorded time	Event
	No data

Data Output | Explain | Messages

	Id [PK] bigint	Rate_type_Id bigint	Status character varying	Killometr_price double precision	Waiting_fine double precision
1	1	1	Night	50	40
2	2	2	Day	20	20

Индексы:

Запрос без индекса:

Query Editor

```
1  SELECT "Client_Id", Count("Client_Id")
2  FROM public."Order"
3  Group by "Client_Id"
```

Notifications | Query History

Recorded time

Data Output | Explain | Messages

```
Successfully run. Total query runtime: 54 msec.
3 rows affected.
```

Query Editor

```
1  SELECT "Client_Id", Count("Client_Id")
2  FROM public."Order"
3  Group by "Client_Id"
```

Notifications | Query History

Recorded time

Data Output | Explain | Messages

Graphical | Analysis | Statistics

Query Editor

```
1  SELECT "Client_Id", Count("Client_Id")
2  FROM public."Order"
3  Group by "Client_Id"
```

Notifications | Query History

Recorded time

Data Output | Explain | Messages

	QUERY PLAN json
1	[

С индексом:

Query Editor

```
1  create index index_counter on public."Order" ("Client_Id")
```

Notifications | Query History

Recorded time	Event

Data Output | Explain | Messages

```
CREATE INDEX

Query returned successfully in 55 msec.
```

Query Editor

```
1  create index index_counter on public."Order" ("Client_Id")
```

Notifications | Query History

Recorded time	Event

Data Output | Explain | Messages

	QUERY PLAN json
1	[

Query Editor

```
1  create index index_counter on public."Order" ("Client_Id")
```

Notifications | Query History

Recorded time	Event

Data Output | Explain | Messages

Graphical | Analysis | Statistics

Запрос №2 без индекса:

Query Editor

```
1  SELECT "Id", "Number"
2  FROM public."Car",
3  (SELECT "Brand_Id", round(avg("Car"."Mileage")) as avg_mileage
4  From public."Car"
5  Group by "Brand_Id") as t_mileage
6  Where "Car"."Brand_Id" = t_mileage."Brand_Id"
7  and "Mileage" >= t_mileage.avg_mileage;
```

Notifications | Query History

Recorded time	Event

Data Output | Explain | Messages

```
Successfully run. Total query runtime: 48 msec.
1 rows affected.
```

Query Editor

```
1  SELECT "Id", "Number"
2  FROM public."Car",
3  (SELECT "Brand_Id", round(avg("Car"."Mileage")) as avg_mileage
4  From public."Car"
5  Group by "Brand_Id") as t_mileage
6  Where "Car"."Brand_Id" = t_mileage."Brand_Id"
7  and "Mileage" >= t_mileage.avg_mileage;
```

Notifications | Query History

Recorded time	Event

Data Output | Explain | Messages

	QUERY PLAN json
1	[

Query Editor

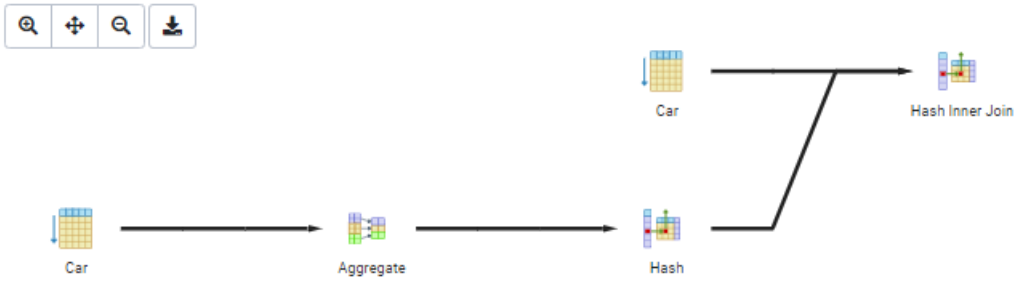
```
1  SELECT "Id", "Number"
2  FROM public."Car",
3  (SELECT "Brand_Id", round(avg("Car"."Mileage")) as avg_mileage
4  From public."Car"
5  Group by "Brand_Id") as t_mileage
6  Where "Car"."Brand_Id" = t_mileage."Brand_Id"
7  and "Mileage" >= t_mileage.avg_mileage;
```

Notifications | Query History

Recorded time	Event	Process ID
		No data found

Data Output | Explain | Messages

Graphical | Analysis | Statistics



Запрос №2 с индексом:

Query Editor

```
1  CREATE UNIQUE INDEX index_avg_mileage ON public."Car"("Id","Number")
```

Notifications | Query History

Recorded time	Event
	No

Data Output | Explain | Messages

```
CREATE INDEX

Query returned successfully in 54 msec.
```

Query Editor

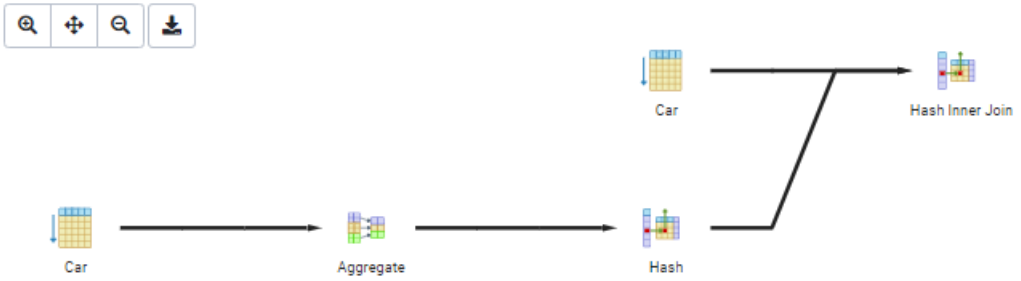
```
1  CREATE UNIQUE INDEX index_avg_mileage ON public."Car"("Id","Number")
```

Notifications | Query History

Recorded time	Event
	No data

Data Output | Explain | Messages

Graphical | Analysis | Statistics



Вывод

В ходе проделанной работы с помощью инструмента запросов Query Tool были составлены 7 запросов к базе данных, 2 запроса на создание представлений, 3 запроса на модификацию данных с подзапросами и создание индексов с оценкой времени выполнения запроса.