

Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение  
высшего образования  
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

**Практическая работа № 4**  
**«Запросы на выборку данных и представления в PostgreSQL»**

по дисциплине «Проектирование и реализация баз данных»

**Выполнила:**  
студентка II курса ИКТ  
группы К3242  
Тостановская Анна Евгеньевна

**Проверила:**  
*Говорова Марина Михайловна*

Санкт-Петербург  
2021

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

**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.
4. Просмотреть историю запросов.

**Индивидуальное задание (Вариант 14. БД «Таксопарк»)**

Описание предметной области: Система должна фиксировать все вызовы такси. Каждому водителю ежедневно начисляется заработная плата в зависимости от количества вызовов и их тарифа (50% от заработанной им суммы).

БД должна содержать следующий минимальный набор сведений: Код сотрудника. ФИО сотрудника. Адрес сотрудника. № телефона сотрудника. Паспортные данные сотрудника. Должность сотрудника. Категория сотрудника. Наименование марки автомобиля. Технические характеристики. Стоимость. Код тарифа. Наименование тарифа. Цена за километр. Код автомобиля. Код марки. Год выпуска. Пробег. Дата последнего ТО. Дата вызова. Время посадки пассажира. Время высадки пассажира. Номер телефона пассажира. Откуда. Куда. Расстояние.

**Задание 2. Создать запросы:**

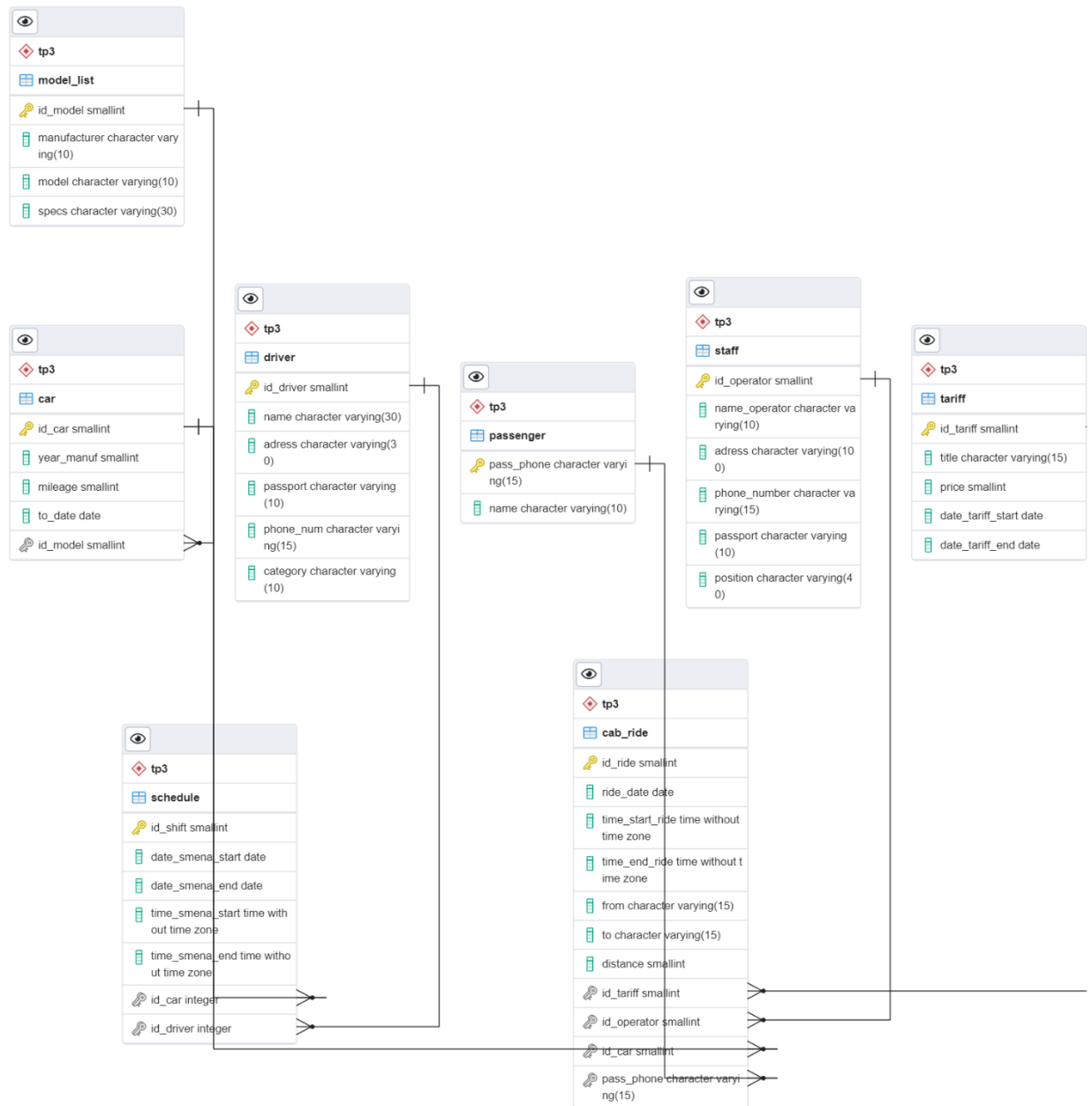
- Вывести данные о водителе, который чаще всех доставляет пассажиров на улицу Чкалова.
- Вывести данные об автомобилях, которые имеют пробег более 250 тысяч километров и которые не проходили ТО в текущем году.
- Сколько раз каждый пассажир воспользовался услугами таксопарка?
- Вывести данные пассажира, который воспользовался услугами таксопарка максимальное число раз.
- Вывести данные о водителе, который ездит на самом дорогом автомобиле.
- Вывести данные пассажира, который всегда ездит с одним и тем же водителем.
- Какие автомобили имеют пробег больше среднего пробега для своей марки.

**Задание 3. Создать представление:**

- содержащее сведения о незанятых на данный момент водителях;
- зарплата всех водителей за вчерашний день.

## Ход работы:

1. Наименование БД: tp3 (таксопарк)
2. Схема логической модели базы данных, сгенерированная в Generate ERD



### 3. Запросы к базе данных

Выполнить запросы согласно индивидуальному заданию, часть 2. В отчете привести формулировку запроса, команду, скриншот выполнения запроса.

- Вывести данные о водителе, который чаще всех доставляет пассажиров в пункт А.

```
Query Editor    Query History

1  SELECT schedule.id_driver, COUNT(cab_ride.to)
2  FROM tp3.cab_ride, tp3.schedule
3  WHERE (schedule.id_car = cab_ride.id_car
4  AND cab_ride.to = 'A')
5  GROUP BY schedule.id_driver HAVING COUNT(cab_ride.to) = (SELECT COUNT(cab_ride.to) AS count_d
6  FROM tp3.cab_ride, tp3.schedule
7  WHERE (schedule.id_car = cab_ride.id_car AND cab_ride.to = 'A'))
8  GROUP BY schedule.id_driver
9  ORDER BY count_d DESC
10 LIMIT 1)
```

Data Output   Explain   Messages   Notifications

	id_driver integer	count bigint	
1	1	4	

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

```
Query Editor  Query History

1  SELECT tp3.car.id_car, tp3.car.year_manuf, tp3.car.mileage, tp3.car.to_date, tp3.car.id_model
2  FROM tp3.car
3  WHERE (car.mileage > 250000
4         AND date_part('year', "to_date") < date_part('year', CURRENT_DATE))

Data Output  Explain  Messages  Notifications

id_car [PK] smallint  year_manuf smallint  mileage integer  to_date date  id_model smallint
1 562 2005 251006 2019-04-07 4
```

- Сколько раз каждый пассажир воспользовался услугами таксопарка?

Query Editor

Query History

```

1 SELECT tp3.cab_ride.pass_phone, COUNT(pass_phone) as count_p
2 FROM tp3.cab_ride
3 GROUP BY pass_phone

```

Data Output

Explain

Messages

Notifications

	pass_phone character varying (15)		count_p bigint	
1	9098765432		1	
2	9213254780		2	
3	9007795544		1	

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

Query Editor

Query History

```
1 SELECT tp3.cab_ride.pass_phone, COUNT(pass_phone)
2 FROM tp3.cab_ride
3 GROUP BY pass_phone HAVING COUNT(pass_phone) = (SELECT COUNT(pass_phone) as count_p
4 FROM tp3.cab_ride
5 GROUP BY pass_phone
6 ORDER BY count_p DESC
7 LIMIT 1)
```

Data Output

Explain

Messages

Notifications

	pass_phone character varying (15)	count bigint	
1	9213254780	2	

- Вывести данные о водителе, который ездит на самом новом автомобиле.

Query Editor

Query History

1

2

3

4

5

6

7

8

SELECT

schedule.id\_driver, driver.name, driver.passport, driver.phone\_num, driver.category

FROM

tp3.schedule, tp3.car, tp3.driver

WHERE

(schedule.id\_car = car.id\_car AND driver.id\_driver = schedule.id\_driver AND car.year\_manuf =

(SELECT

car.year\_manuf

FROM

tp3.car, tp3.schedule

WHERE

schedule.id\_car = car.id\_car

GROUP BY

car.year\_manuf

ORDER BY

car.year\_manuf DESC

Data Output

Explain

Messages

Notifications

	<div><div>id_driver</div><div>integer</div></div>	<div><div>name</div><div>character varying (30)</div></div>	<div><div>passport</div><div>character varying (10)</div></div>	<div><div>phone_num</div><div>character varying (15)</div></div>	<div><div>category</div><div>character varying (10)</div></div>	
1	3	Любовь	7809654476	9674351280		

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

Query Editor

Query History

1

SELECT pass

2

FROM (SELECT distinct cab\_ride.pass\_phone as pass, schedule.id\_driver as dr

3

FROM tp3.cab\_ride, tp3.schedule

4

WHERE (schedule.id\_car = cab\_ride.id\_car)) as carrr

5

GROUP BY pass having count(dr) = 1

Data Output

Explain

Messages

Notifications

pass

character varying (15)

1

9007795544

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

Query Editor

Query History

```
1 SELECT car.id_car, car.mileage, car.max_mileage, s_mil.mean_mil, s_mil.car_model FROM tp3.car,
2 (SELECT car.id_model as car_model, sum(car.max_mileage)/2 as mean_mil
3 FROM tp3.car
4 GROUP BY id_model having sum(car.mileage)/count(id_model) > 0) as s_mil
5 WHERE car.id_model = s_mil.car_model and car.mileage >= s_mil.mean_mil
6
```

Data Output

Explain

Messages

Notifications

	<div>id_car</div> <div>[PK] smallint</div>	<div>mileage</div> <div>integer</div>	<div>max_mileage</div> <div>integer</div>	<div>mean_mil</div> <div>bigint</div>	<div>car_model</div> <div>smallint</div>	
1	562	251006	200000	100000	4	
2	678	179590	150000	75000	3	

#### 4. Представления

Выполнить запросы на создание представлений согласно индивидуальному заданию, часть 3. В отчете привести формулировку запроса, команду, скриншот выполнения запроса и просмотр содержимого представления.

- содержащее сведения о незанятых на данный момент водителях

Создание:

Query Editor	Query History		
<pre>1 CREATE VIEW public.fr_drivers AS 2     SELECT DISTINCT schedule.id_driver, cab_ride.id_car 3     FROM tp3.schedule, tp3.cab_ride 4     WHERE schedule.id_car = cab_ride.id_car AND schedule.time_smena_end is NULL</pre>			
Data Output	Explain	Messages	Notifications
CREATE VIEW			
Query returned successfully in 197 msec.			

Использование:

Query Editor

Query History

1

SELECT \* FROM public.fr\_drivers

Data Output

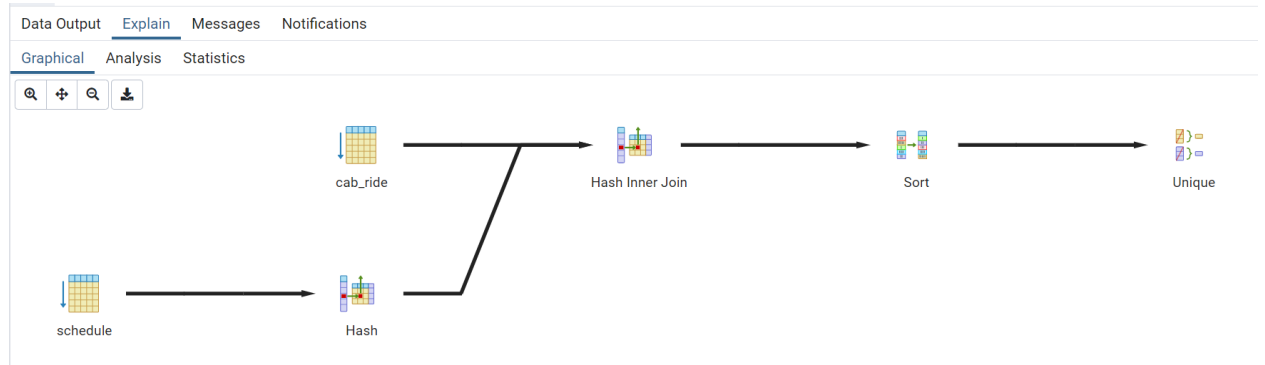
Explain

Messages

Notifications

	id_driver integer	id_car smallint	
1	1	678	
2	3	9	

## Схема:



- зарплата всех водителей за вчерашний день.

## Создание:

Query Editor
Query History

```

1 CREATE VIEW public.yesterday_income AS
2     SELECT sum(tariff.price * cab_ride.distance)
3     FROM tp3.tariff, tp3.cab_ride
4     WHERE (tariff.id_tariff = cab_ride.id_tariff
5           AND cab_ride.ride_date = CURRENT_DATE - 1)

```

Data Output
Explain
Messages
Notifications

CREATE VIEW

Query returned successfully in 148 msec.

## Использование:

Query Editor
Query History

```

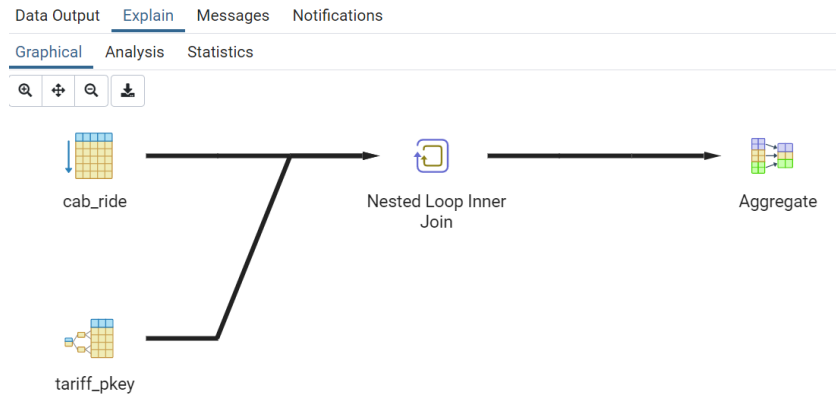
1 SELECT * FROM public.yesterday_income
2

```

Data Output
Explain
Messages
Notifications

	sum bigint	
1	333	

## Схема:



### 5. 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

Добавить заказ свободному на данный момент водителю

Query Editor Query History

```

1 INSERT INTO tp3.cab_ride(
2     id_ride, ride_date, time_start_ride, time_end_ride, "from",
3     "to", distance, id_tariff, id_operator, id_car, pass_phone)
4     VALUES (9, '2021-06-20', '14:44', Null, 'B', 'F', 11, 2, 1,
5             (SELECT id_car FROM public.fr_drivers LIMIT 1), 9213254780)
  
```

Data Output Explain Messages Notifications

INSERT 0 1









Query returned successfully in 177 msec.

## До:

Data Output		Explain	Messages	Notifications					
 id_ride [PK] smallint	 ride_date date	 time_start_ride time without time zone	 time_end_ride time without time zone	 from character varying (15)	 to character varying (15)	 distance smallint	 id_tariff smallint	 id_operator smallint	
1	1	2020-09-21	12:33:00	12:50:00	A	B	9	3	
2	2	2020-09-22	21:00:00	22:14:00	B	A	19	2	
3	3	2020-09-21	14:47:00	15:02:00	C	A	4	2	
4	4	2020-09-21	13:37:00	13:50:00	D	A	23	1	
5	5	2020-09-21	23:10:00	23:38:00	E	C	8	3	
6	6	2021-06-19	20:13:00	20:23:00	C	A	6	2	
7	7	2021-06-19	10:00:00	11:07:00	A	E	29	3	
8	8	2021-06-20	07:52:00	08:24:00	C	A	13	1	



После:

Data Output		Explain	Messages	Notifications				
	 id_ride [PK] smallint	 ride_date date	 time_start_ride time without time zone	 time_end_ride time without time zone	 from character varying (15)	 to character varying (15)	 distance smallint	 id_tariff smallint
1		1 2020-09-21	12:33:00	12:50:00	A	B		9
2		2 2020-09-22	21:00:00	22:14:00	B	A		19
3		3 2020-09-21	14:47:00	15:02:00	C	A		4
4		4 2020-09-21	13:37:00	13:50:00	D	A		23
5		5 2020-09-21	23:10:00	23:38:00	E	C		8
6		6 2021-06-19	20:13:00	20:23:00	C	A		6
7		7 2021-06-19	10:00:00	11:07:00	A	E		29
8		8 2021-06-20	07:52:00	08:24:00	C	A		13
9		9 2021-06-20	14:44:00	[null]	B	F		11

- UPDATE

Обновить дату тех.осмотра для машин, у которых был тех.осмотр в последний год, но которые требуют ремонта (указано в tp3.model\_list.specs).

Query Editor

Query History

```
1 UPDATE tp3.car
2     SET "to_date" = '2021-06-20'
3     WHERE id_car = (SELECT id_car FROM
4         (SELECT car.id_car, "to_date", model_list.specs
5         FROM tp3.car, tp3.model_list
6         WHERE (car.id_model = model_list.id_model
7             AND model_list.specs = 'требуется ремонта'
8             AND date_part('month', "to_date") > date_part('month', CURRENT_DATE)-12)) as n_TO);
```

Data Output

Explain








Messages

Notifications








UPDATE 1

Query returned successfully in 78 msec.

До:

Data Output		Explain	Messages	Notifications		
	id_car [PK] smallint 	year_manuf smallint 	mileage integer 	to_date date 	id_model smallint 	max_mileage integer 
1	9	2015	9070	2021-04-19	2	100000
2	562	2005	251006	2020-04-07	4	200000
3	678	2009	179590	2021-05-31	3	150000
4	777	2013	34907	2021-05-01	1	100000

После:

Data Output		Explain	Messages	Notifications		
	id_car [PK] smallint 	year_manuf smallint 	mileage integer 	to_date date 	id_model smallint 	max_mileage integer 
1	9	2015	9070	2021-06-20	2	100000
2	562	2005	251006	2020-04-07	4	200000
3	678	2009	179590	2021-05-31	3	150000
4	777	2013	34907	2021-05-01	1	100000

- DELETE

Удалить все записи из графика работы водителей до 2021 года.

Query Editor   Query History

```

1 DELETE FROM tp3.schedule
2 WHERE date_smena_end <= '2020-12-31';
3









```

Data Output   Explain   Messages   Notifications









DELETE 3

Query returned successfully in 73 msec.

До:

Data Output		Explain	Messages	Notifications				
	id_shift [PK] smallint 		date_smena_start date 	date_smena_end date 	time_smena_start time without time zone 	time_smena_end time without time zone 	id_car integer 	id_driver integer 
1	1	1	2020-09-21	2020-09-22	09:10:00	19:10:00	678	1
2	2	2	2020-09-21	2020-09-22	19:30:00	09:40:00	9	3
3	3	3	2020-09-22	2020-09-23	07:00:00	18:50:00	777	2
4	4	4	2021-06-20	2021-06-21	11:00:00	01:00:00	9	3
5	5	5	2021-06-19	2021-06-20	22:00:00	08:00:00	777	2
6	6	6	2021-06-20	2021-06-20	10:00:00	[null]	9	3
7	7	7	2021-06-20	2021-06-21	12:00:00	[null]	678	1

После:

Data Output		Explain	Messages	Notifications				
	id_shift [PK] smallint 		date_smena_start date 	date_smena_end date 	time_smena_start time without time zone 	time_smena_end time without time zone 	id_car integer 	id_driver integer 
1	4		2021-06-20	2021-06-21	11:00:00	01:00:00	9	3
2	5		2021-06-19	2021-06-20	22:00:00	08:00:00	777	2
3	6		2021-06-20	2021-06-20	10:00:00	[null]	9	3
4	7		2021-06-20	2021-06-21	12:00:00	[null]	678	1

**Вывод:** В ходе выполнения работы я стала более уверенным пользователем ПО pgadmin, выполнив запросы к базе данных и представления согласно индивидуальному заданию, а так же запросы на модификацию данных (INSERT, UPDATE, DELETE), составленные самостоятельно.