

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»  
Факультет инфокоммуникационных технологий

**ОТЧЕТ**  
**О ЛАБОРАТОРНОЙ РАБОТЕ № 4**  
по теме: Запросы на выборку данных к БД PostgreSQL.  
Представления в PostgreSQL.  
по дисциплине: Проектирование и реализация баз данных

Специальность:  
45.03.04 Интеллектуальные системы в гуманитарной сфере

Проверил:  
Горова М.М. \_\_\_\_\_  
Дата: «\_\_» \_\_\_\_\_ 20\_\_ г.  
Оценка \_\_\_\_\_

Выполнил:  
студент группы k3243  
Новикова В.В.

Санкт-Петербург 2020/2021

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

**Оборудование:** компьютерный класс.

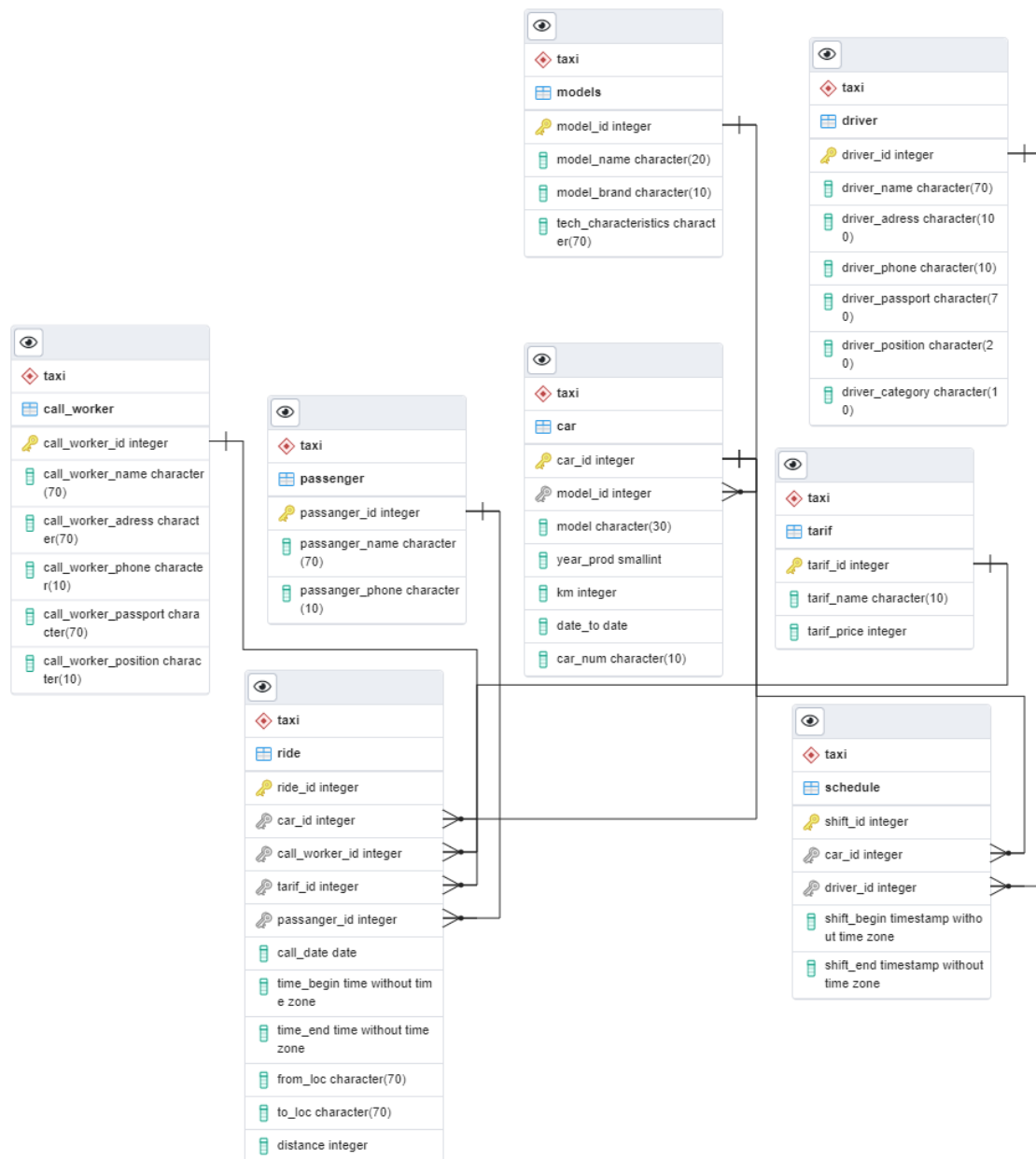
**Программное обеспечение:** СУБД PostgreSQL, pgadmin 4.

**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.

БД: taxi service

ERD диаграмма:



Запросы:

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

```

1 select count(ride.from_loc), schedule.driver_id
2 from taxi.ride, taxi.schedule where(schedule.car_id = ride.car_id
3         and ride.to_loc like 'Chkalova%')
4 group by schedule.driver_id having count(ride.to_loc)= (select count(ride.to_loc) as count_
5 from taxi.ride,taxi.schedule where(schedule.car_id = ride.car_id and ride.to_loc like 'Chkalova%')
6 group by schedule.driver_id
7 order by count_ desc
8 limit 1)

```

Data Output Explain Messages Notifications

	count bigint	driver_id integer
1	2	1

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

```

1 select car.car_id,car.model,car.car_num from taxi.car
2 where (car.km >250000
3         and date_part('year',"date_to") < date_part('year',current_date))

```

Data Output Explain Messages Notifications

	car_id [PK] integer	model character (30)	car_num character (10)
1	7	Rio	y543gf

### 3. Сколько раз каждый пассажир воспользовался услугами таксопарка

```
1 select passenger.passanger_id, passenger.passanger_name, count(ride.passanger_id)
2 from taxi.passenger, taxi.ride
3 where (ride.passanger_id = passenger.passanger_id)
4 group by passenger.passanger_id
```

Data Output Explain Messages Notifications

	passanger_id [PK] integer	passanger_name character (70)	count bigint
1	3	Antonova Anita Ant...	2
2	5	Danilova Elizaveta ...	2
3	4	Vernov Alexey Vladi...	2
4	2	Antonov Anton Ant...	3
5	1	Victorova Alina Ant...	2

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

```
1 select passenger.passanger_id, passenger.passanger_name, count(ride.passanger_id)
2 from taxi.passenger, taxi.ride
3 where (ride.passanger_id = passenger.passanger_id)
4 group by passenger.passanger_id
5 order by count(ride.passanger_id) desc
6 limit 1
7
```

Data Output Explain Messages Notifications

	passanger_id [PK] integer	passanger_name character (70)	count bigint
1	2	Antonov Anton Ant...	3

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

Самый дорогой автомобиль – Toyota Fortuner

```
1 select schedule.driver_id, driver.driver_name from taxi.schedule, taxi.car, taxi.driver
2 where (schedule.car_id=car.car_id and driver.driver_id=schedule.driver_id
3        and car.model='Fortuner')
4
```

Data Output Explain Messages Notifications

	driver_id integer	driver_name character (70)
1	2	Sidorov Ivan Sidor...

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

```
1 select passenger.passanger_name from (select distinct ride.passanger_id as pass_id,
2                                       schedule.driver_id as dr_id
3 from taxi.ride, taxi.schedule
4 where (schedule.car_id = ride.car_id)) as aa, taxi.passenger
5 where (pass_id = passenger.passanger_id)
6 group by passenger.passanger_name having count(dr_id)=1
7
```

Data Output Explain Messages Notifications

	passanger_name character (70)
1	Antonov Anton Ant...

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

```
1 select car.car_id,car.km, meankm.model_m, meankm.mean_kms
2 from taxi.car, (select car.model_id as model_m,sum(car.km)/count(model_id) as mean_kms
3                 from taxi.car
4                 group by model_id having sum(car.km)/count(model_id)>0) as meankm
5                 where car.model_id=meankm.model_m and car.km > meankm.mean_kms
```

Data Output Explain Messages Notifications

	car_id [PK] integer	km integer	model_m integer	mean_kms bigint
1	6	6083	1	5068
2	7	250001	2	128027
3	8	11001	5	8000
4	9	76001	3	40916
5	10	54826	4	33454

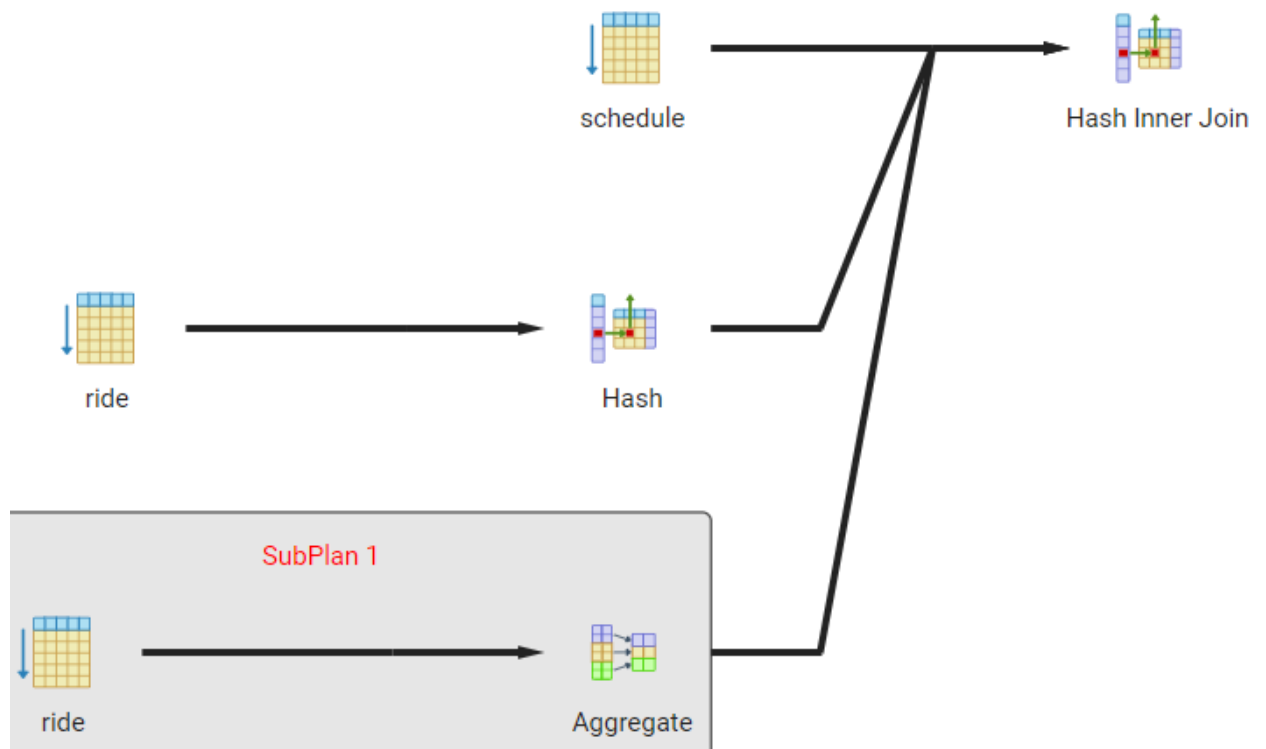
Представления:

1. содержащее сведения о незанятых на данный момент водителях

```
create view taxi.without_passens as
select ride.car_id,schedule.driver_id from taxi.ride, taxi.schedule
where ('2021-06-07 20:00:00'<schedule.shift_end and '2021-06-07 20:00:00'>schedule.shift_begin
      and ride.car_id=schedule.car_id and ride.time_end is not null
      and ride.time_begin = (select max(ride.time_begin) from taxi.ride
                             where(ride.car_id = schedule.car_id)))
```

Data Output

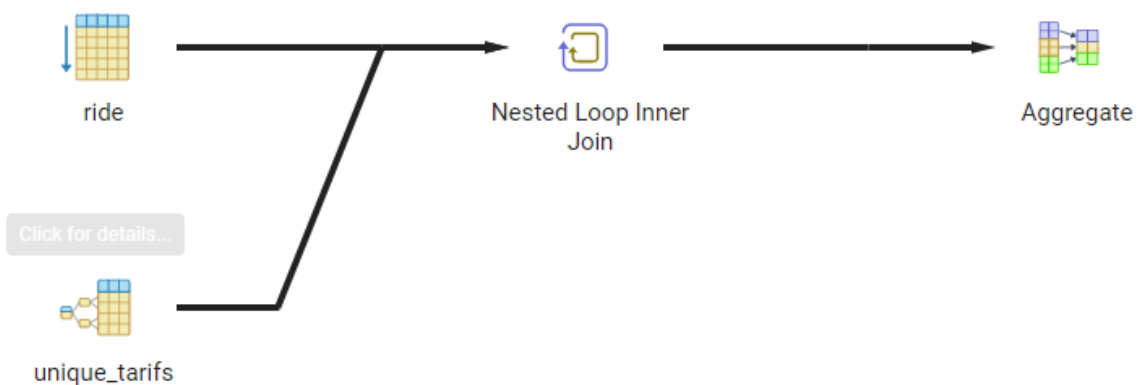
	car_id integer	driver_id integer
1	2	1



## 2. зарплата всех водителей за день

```
create view taxi.salary_daily as select sum(0.5*ride.distance*tarif.tarif_price)
from taxi.ride,taxi.tarif
where (tarif.tarif_id = ride.tarif_id and ride.call_date = '2021-06-08')
```

Data Output	
	sum numeric
1	4900.0



## Модификация данных:

### 1. INSERT



Создадим смену с тарифом Lux. Для нее выберем самую дорогую модель(Fortuner) и самую новую машину

```
insert into taxi.schedule (shift_id,car_id,driver_id,shift_begin,shift_end)
values (6,
       (select car_id from taxi.car
        where car.model = 'Fortuner'
        and car.km = (select min(car.km) from taxi.car where car.model = 'Fortuner')),
       6, '2021-06-10 10:00:00','2021-06-10 20:00:00')
```

Результат:

Data Output

	shift_id [PK] integer	car_id integer	driver_id integer	shift_begin timestamp without time zone	shift_end timestamp without time zone
1	1	1	2	2021-06-06 08:05:00	2021-06-06 20:00:00
2	2	2	1	2021-06-07 16:05:00	2021-06-08 02:00:00
3	3	3	4	2021-06-07 08:05:00	2021-06-07 16:00:00
4	4	4	5	2021-06-08 09:00:00	2021-06-08 19:00:00
5	5	5	3	2021-06-08 20:00:00	2021-06-09 03:00:00
6	6	1	6	2021-06-10 10:00:00	2021-06-10 20:00:00

	shift_id [PK] integer	car_id integer	driver_id integer	shift_begin timestamp without time zone	shift_end timestamp without time zone
1	1	1	2	2021-06-06 08:05:00	2021-06-06 20:00:00
2	2	2	1	2021-06-07 16:05:00	2021-06-08 02:00:00
3	3	3	4	2021-06-07 08:05:00	2021-06-07 16:00:00
4	4	4	5	2021-06-08 09:00:00	2021-06-08 19:00:00
5	5	5	3	2021-06-08 20:00:00	2021-06-09 03:00:00
6	6	1	6	2021-06-10 10:00:00	2021-06-10 20:00:00

## 2. UPDATE

Обновим дату ТО у автомобилей, проходивших ТО в январе 2021

```
update taxi.car set "date_to" = '2021-06-06'
where car_id in (select car_id from taxi.car
                 where date_part('month',"date_to") = '01')
```

	car_id [PK] integer	model_id integer	model character (30)	year_prod smallint	km integer	date_to date	car_num character (10)
1	2	2	Rio	2017	6054	2021-02-16	t262ed
2	4	3	Duster	2019	5832	2021-03-09	x872nj
3	5	4	Sandero	2016	12083	2021-04-18	a624je
4	6	1	Solaris	2020	6083	2021-04-04	k835fr
5	8	5	Fortuner	2017	11001	2021-02-09	s287sy
6	9	3	Duster	2015	76001	2021-04-09	d726dy
7	10	4	Sandero	2016	54826	2021-03-24	g872ud
8	1	5	Fortuner	2015	5000	2021-01-01	k228ek
9	3	1	Solaris	2018	4054	2021-01-01	s724jx

Результат:

	car_id [PK] integer	model_id integer	model character (30)	year_prod smallint	km integer	date_to date	car_num character (10)
1	2	2	Rio	2017	6054	2021-02-16	t262ed
2	4	3	Duster	2019	5832	2021-03-09	x872nj
3	5	4	Sandero	2016	12083	2021-04-18	a624je
4	6	1	Solaris	2020	6083	2021-04-04	k835fr
5	8	5	Fortuner	2017	11001	2021-02-09	s287sy
6	9	3	Duster	2015	76001	2021-04-09	d726dy
7	10	4	Sandero	2016	54826	2021-03-24	g872ud
8	1	5	Fortuner	2015	5000	2021-06-06	k228ek
9	3	1	Solaris	2018	4054	2021-06-06	s724jx

### 3. DELETE

Удалим из списка машину, не проходившую ТО в этом году

```
delete from taxi.car
where car_id = (select car_id from taxi.car
                where date_part('year',"date_to") < 2021)
```

	car_id [PK] integer	model_id integer	model character (30)	year_prod smallint	km integer	date_to date	car_num character (10)
1	1	5	Fortuner	2015	5000	2021-01-16	k228ek
2	2	2	Rio	2017	6054	2021-02-16	t262ed
3	3	1	Solaris	2018	4054	2021-01-26	s724jx
4	4	3	Duster	2019	5832	2021-03-09	x872nj
5	5	4	Sandero	2016	12083	2021-04-18	a624je
6	6	1	Solaris	2020	6083	2021-04-04	k835fr
7	7	2	Rio	2012	250001	2020-06-09	y543gf
8	8	5	Fortuner	2017	11001	2021-02-09	s287sy
9	9	3	Duster	2015	76001	2021-04-09	d726dy
10	10	4	Sandero	2016	54826	2021-03-24	g872ud

Результат:

	car_id [PK] integer	model_id integer	model character (30)	year_prod smallint	km integer	date_to date	car_num character (10)
1	2	2	Rio	2017	6054	2021-02-16	t262ed
2	4	3	Duster	2019	5832	2021-03-09	x872nj
3	5	4	Sandero	2016	12083	2021-04-18	a624je
4	6	1	Solaris	2020	6083	2021-04-04	k835fr
5	8	5	Fortuner	2017	11001	2021-02-09	s287sy
6	9	3	Duster	2015	76001	2021-04-09	d726dy
7	10	4	Sandero	2016	54826	2021-03-24	g872ud
8	1	5	Fortuner	2015	5000	2021-06-06	k228ek
9	3	1	Solaris	2018	4054	2021-06-06	s724jx

**Вывод:** в ходе работы были выполнены запросы к БД, созданы представления, а также выполнены модификации данных с использованием подзапросов.