

**Министерство науки и высшего образования Российской Федерации**  
**федеральное государственное автономное образовательное учреждение высшего образования**  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе №4 «Запросы на выборку и модификацию данных. Представления. Работа с индексами»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Юркин А.С.

Факультет: ИКТ

Группа: К3240

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

## Оглавление

Цель работы.....	Ошибка! Закладка не определена.
Практическое задание .....	Ошибка! Закладка не определена.
Вариант 12. БД «Прокат автомобилей» .....	Ошибка! Закладка не определена.
Рисунок 1 – Схема логической модели базы данных. ....	Ошибка! Закладка не определена.
Листинг дампа .....	Ошибка! Закладка не определена.
Вывод .....	4

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

**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 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.

## 1. Запросы к БД.

Какой автомобиль находился в прокате максимальное количество часов?

Листинг:

```
select
  r.car,
  m.name AS model_name,
  sum(EXTRACT(EPOCH FROM (r.enddate - r.startdate))) / 3600 as total_hours
from contract r
join car c on r.car = c.id
join model m on c.model = m.id
group by r.car, m.name
order by total_hours desc
limit 1;
```

The screenshot shows a database query interface with a 'Query' tab and a 'Data Output' tab. The SQL query is as follows:

```
1 select
2   r.car,
3   m.name AS model_name,
4   sum(EXTRACT(EPOCH FROM (r.enddate - r.startdate))) / 3600 as total_hours
5 from contract r
6 join car c on r.car = c.id
7 join model m on c.model = m.id
8 group by r.car, m.name
9 order by total_hours desc
10 limit 1;
```

The 'Data Output' tab shows the results of the query in a table with the following columns: car, model\_name, and total\_hours. The results are as follows:

car	model_name	total_hours
3	SUV Y	132.00000000000000

Автомобили какой марки чаще всего брались в прокат?

Листинг:

```
select
  m.name AS model_name,
  count(c.id) AS rental_count
from contract ct
join car c on ct.car = c.id
join model m on c.model = m.id
group by m.name
order by rental_count desc
limit 1;
```

The screenshot shows a database query interface with a 'Query' tab and a 'Data Output' tab. The SQL query is as follows:

```
1 select
2   m.name AS model_name,
3   count(c.id) AS rental_count
4 from contract ct
5 join car c on ct.car = c.id
6 join model m on c.model = m.id
7 group by m.name
8 order by rental_count desc
9 limit 1;
```

The 'Data Output' tab shows the results of the query in a table with the following columns: model\_name and rental\_count. The results are as follows:

model_name	rental_count
Coupe Z	1

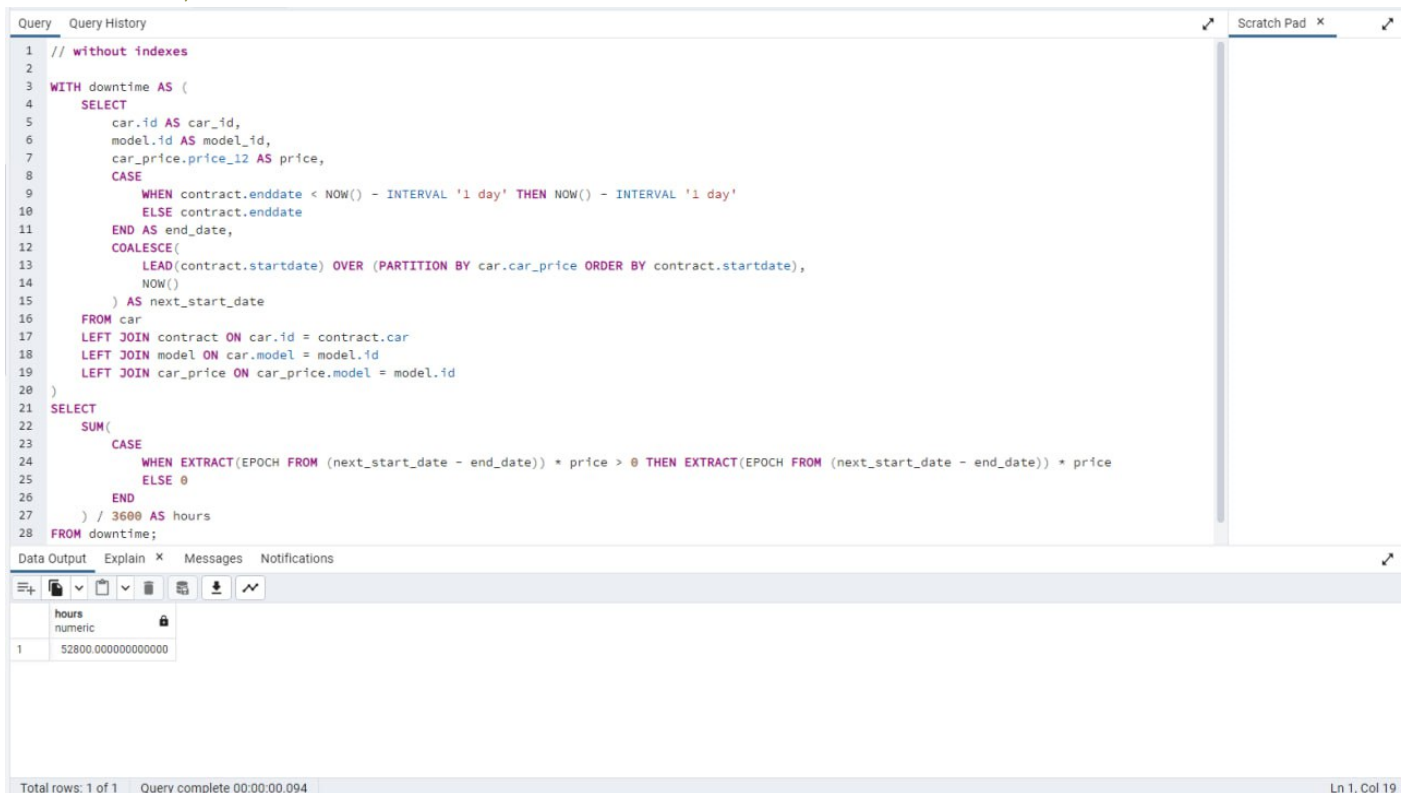
Определить убытки от простоя автомобилей за вчерашний день.

Листинг:

```

with downtime as (
    select
        car.id as car_id,
        model.id as model_id,
        car_price.price_12 as price,
        case
            when contract.enddate < NOW() - INTERVAL '1 day' then NOW() - INTERVAL
'1 day'
            else contract.enddate
        end as end_date,
        COALESCE(
            lead(contract.startdate) over (partition by car.car_price order by
contract.startdate),
            NOW()
        ) as next_start_date
    from car
    left join contract on car.id = contract.car
    left join model on car.model = model.id
    left join car_price on car_price.model = model.id
)
select
    sum(
        case
            when EXTRACT(EPOCH FROM (next_start_date - end_date)) * price > 0 then
EXTRACT(EPOCH FROM (next_start_date - end_date)) * price
            else 0
        end
    ) / 3600 as hours
from downtime;

```



The screenshot shows a SQL IDE interface. The main editor displays the SQL query from the previous block. Below the editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with two columns: 'hours' and 'numeric'. The first row of data shows a value of 52800.000000000000. The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.094'.

hours	numeric
1	52800.000000000000

Вывести данные автомобиля, имеющего максимальный пробег.

Листинг:

```

select * from car
order by mileage desc
limit 1;

```

Какой автомобиль суммарно находился в прокате дольше всех.

Листинг:

```

select car.* from contract
join car on car.id = contract.id
where (enddate-startdate) = (
select max(enddate-startdate) from contract)
limit 1;

```

Query Query History

```

1 select car.* from contract
2 join car on car.id = contract.id
3 where (enddate-startdate) = (
4 select max(enddate-startdate) from contract)
5 limit 1;
6

```

Data Output Explain Messages Notifications

	id [PK] integer	deposit integer	registration_number character varying (25)	plate_number character varying (9)	engine_number character varying (17)	body_number character varying (17)	car_year integer	mileage integer	car_price integer	inspection_date date	remarks character varying (50)	return_status boolean
1	3	700	OPQ9101	GHI-789	ENG003	BODY003	2022	6800	35000	2023-07-18	Minor scratches	false

Total rows: 1 of 1 Query complete 00:00:00.145

✓ Successfully run. Total query runtime: 145 msec. 1 rows affected. ✕

Ln 6, Col 1

Определить, каким количеством автомобилей каждой марки и модели владеет компания.

Листинг:

```

select model.name as "model", count(*) AS "count"
from car
join model on car.model = model.id
group by model.name;

```

Query

Query History

1

2

3

4

SELECT

model.name

AS

"model",

COUNT(\*)

AS

"count"

FROM

car

JOIN

model

ON

car.model

=

model.id

GROUP

BY

model.name;

Data Output

Explain

×

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	model		count	
	character varying (100)	🔒	bigint	🔒
1	Coupe Z		1	
2	SUV Y		2	
3	Sedan X		1	

Определить средний “возраст” автомобилей компании.

Листинг:

```
select avg(date_part('year', CURRENT_DATE) - car_year) FROM car;
```

The screenshot shows a SQL IDE with a query editor at the top containing the following SQL statement:

```
1 select avg(date_part('year', CURRENT_DATE) - car_year) FROM car;
```

Below the editor is a toolbar and a 'Data Output' tab. The 'Data Output' tab displays a single row of results:

avg
double precision

The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.084'. A green notification box on the right states: 'Successfully run. Total query runtime: 84 msec. 1 rows affected. X'.

## 2. Создание представлений:

Какой автомобиль ни разу не был в прокате?

Листинг:

```
create view NotInRentCar AS  
select car.*  
from car  
left join contract on car.id = contract.car  
where contract.car is NULL;
```



Query
Query History

```

1 CREATE VIEW NotInRentCar AS
2 SELECT car.*
3 FROM car
4 LEFT JOIN contract ON car.id = contract.car
5 WHERE contract.car IS NULL;
6

```

Data Output
Explain
Messages
Notifications

CREATE VIEW

Query returned successfully in 99 msec.

Total rows: 2 of 2    Query complete 00:00:00.099

Query
Query History

```

1 select * from NotInRentCar;

```

Data Output
Explain
Messages
Notifications

	id integer	deposit integer	registration_number character varying (25)	plate_number character varying (9)	engine_number character varying (17)	body_number character varying (17)	car_year integer	mileage integer	car_price integer	inspection_date date	remarks character varying (50)	return_receipt character varying (25)	time_in_rent integer
1	4	800	OPQ9444	GHI-666	ENG004	BODY004	2020	16800	55000	2023-07-18	Minor scratches	false	0

Вывести данные клиентов, не вернувших автомобиль вовремя.

Листинг:

```

create view ExpiredClient as
select
    c.id as client_id,
    c.phone as phone,
    contract.enddate as enddate
from
    client c
join

```

```
contract on c.id = contract.client
where
contract.enddate < NOW()
and contract.status = 'Active';
```

Query

Query History

1

CREATE VIEW ExpiredClient AS

2

SELECT

3

c.id AS client\_id,

4

c.phone AS phone,

5

contract.enddate AS enddate

6

FROM

7

client c

8

JOIN

9

contract ON c.id = contract.client

10

WHERE

11

contract.enddate < NOW()

12

AND contract.status = 'Active';

13

Data Output

Explain

×

Messages

Notifications

CREATE VIEW

Query returned successfully in 47 msec.

Total rows: 2 of 2

Query complete 00:00:00.047

Query

Query History

1

select \* from ExpiredClient;

2

Data Output

Explain

Messages

Notifications

+

📄

▼

📋

▼

🗑️

📦

📥

📈

	client_id integer	phone character varying (11)	enddate timestamp with time zone
1	1	+1234567890	2023-10-25 00:00:00+03
2	2	+0987654321	2023-10-25 00:00:00+03

Total rows: 2 of 2

Query complete 00:00:00.088

✓ Succ

### 3. Запросы на модификацию данных:

### Добавление автомобиля с моделью с конкретным названием

ЛИСТИНГ:

```
insert into car (id, mileage, license_plate, vin, engine_number, body_number, year,
purchase_price, current_value, inspection_date, notes, available, rented_times,
model_id)
values (
    5, 800, 'OPQ9411', 'GHI-777', 'ENG005', 'BODY005', 2010,
    36800, 15000, NOW(), 'Minor scratches', 'false', 0,
    (select id from model where name = 'Sedan X')
);
```

Query
Query History
Scratch Pad

```

1 INSERT INTO car (id, deposit, registration_number, plate_number, engine_number, body_number, car_year, mileage, car_price, inspection_date, remarks, return_receipt, time_in_rent)
2 VALUES (
3     5, 800, 'OPQ9411', 'GHI-777', 'ENG005', 'BODY005', 2010,
4     36800, 15000, NOW(), 'Minor scratches', 'false', 0,
5     (SELECT id FROM model WHERE name = 'Sedan X')
6 );
7

```

Data Output
Explain
Messages
Notifications

```

INSERT 0 1
Query returned successfully in 123 msec.

```

Total rows: 1 of 1
Query complete 00:00:00.123
Ln 1, Col 66

Query
Query History
Scratch Pad

```

1 select * from car;

```

Data Output
Explain
Messages
Notifications

id	deposit	registration_number	plate_number	engine_number	body_number	car_year	mileage	car_price	inspection_date	remarks	return_receipt	time_in_rent
1	1	500	XYZ1234	ABC-123	ENG001	BODY001	2023	1200	25000	2023-10-01	No remarks	false
2	2	600	LMN5678	DEF-456	ENG002	BODY002	2023	5400	30000	2023-09-20	Good condition	false
3	3	700	OPQ9101	GHI-789	ENG003	BODY003	2022	6800	35000	2023-07-18	Minor scratches	false
4	4	800	OPQ9444	GHI-666	ENG004	BODY004	2020	16800	55000	2023-07-18	Minor scratches	false
5	5	800	OPQ9411	GHI-777	ENG005	BODY005	2010	36800	15000	2023-11-09	Minor scratches	false

Total rows: 5 of 5
Query complete 00:00:00.134

Successfully run. Total query runtime: 134 msec. 5 rows affected.

## Изменение отметок клиентов-должников, которые не вернули вовремя машину

Листинг:

```

insert into car (id, mileage, license_plate, vin, engine_number, body_number, year,
purchase_price, current_value, inspection_date, notes, available, rented_times,
model_id)
values (
    5, 800, 'OPQ9411', 'GHI-777', 'ENG005', 'BODY005', 2010,
    36800, 15000, NOW(), 'Minor scratches', 'false', 0,
    (select id from model where name = 'Sedan X')
);

```

Query    Query History
Scratch Pad ✕

```
1 select * from client;
```

Data Output   Explain ✕   Messages   Notifications
↗

≡
📄
▼
🗑️
📄
📄
📄
📄

	id [PK] integer	phone character varying (11)	name character varying (100)	address character varying (100)	country character varying (35)	remarks character varying (100)	sale integer	time_in_rent integer	passport character varying (10)
1	1	+1234567890	John Doe	123 Elm St, Springfield	USA	[null]	10	0	P123456789
2	2	+0987654321	Alice Brown	456 Maple St, Metropolis	Canada	[null]	5	0	P234567890
3	3	+1122334455	Charlie White	789 Pine St, Gotham	UK	[null]	7	0	P345678901

Total rows: 3 of 3    Query complete 00:00:00.074

✓
Successfully run. Total query runtime: 74 msec. 3 rows affected. ✕

Ln 1, Col 21



Query

Query History

1

```
select * from contract;
```

2

Data Output

Explain

Messages

Notifications

	id [PK] integer	price integer	startdate timestamp with time zone	enddate timestamp with time zone	status character varying (25)	payment_status character varying (25)	client integer	employee integer	car integer	insurance integer
1	1	250	2023-10-21 00:00:00+03	2023-10-25 00:00:00+03	Active	Paid	1	1	1	1
2	2	280	2023-10-22 00:00:00+03	2023-10-25 00:00:00+03	Active	Paid	2	2	2	2
3	3	320	2023-11-05 00:00:00+03	2023-11-09 00:00:00+03	Completed	Paid	3	3	3	3

Query

Query History

1

```
UPDATE contract AS c
SET enddate = c.enddate + COALESCE(
(
SELECT p.hours
FROM prolongation AS p
WHERE p.contract = c.id
) * INTERVAL '1 hour',
INTERVAL '0 hour'
);
```

10

Scratch Pad

Data Output

Explain

Messages

Notifications

UPDATE 3

Query returned successfully in 230 msec.

✓ Query returned successfully in 230 msec. ✕

Total rows: 3 of 3    Query complete 00:00:00.230    Ln 9, Col 2



[illegible]

```
delete from prolongation as p
where contract in (
    select id
    from contract
    where startdate > p.date
);
```

QueryQuery History

1

select \* from prolongation;

2

Data OutputExplain ×MessagesNotifications

	id [PK] integer	contract integer	date timestamp with time zone	hours integer
1		1	2023-10-07 00:00:00+03	24
2		2	2023-10-13 00:00:00+03	48
3		3	2023-10-10 00:00:00+03	36

QueryQuery History

1

DELETE FROM prolongation as p

2

WHERE contract IN (

3

SELECT id

4

FROM contract

5

WHERE startdate > p.date

6

);

7

Data OutputExplain ×MessagesNotifications

DELETE 3

Query returned successfully in 106 msec.

Ln 4, Col 18

✓ Query returned successfully in 106 msec. ✕

Total rows: 3 of 3Query complete 00:00:00.106

Query Query History

```
1 select * from prolongation;
```

Scratch Pad

Data Output Explain Messages Notifications

id	contract	date	hours
[PK] Integer	Integer	timestamp with time zone	Integer

Total rows: 0 of 0 Query complete 00:00:00.105

✓ Successfully run. Total query runtime: 105 msec. 0 rows affected. ✕

Ln 1, Col 28

## 4. Создание индексов

Листинг:

```
CREATE INDEX idx_car_model_id ON car (model);

CREATE INDEX idx_price_model_id ON car_price (model);

CREATE INDEX idx_contract_client_id ON contract (client);

CREATE INDEX idx_prolongation_contract ON prolongation (contract);
```

Query Query History

```
1 CREATE INDEX idx_car_model_id ON car (model);
2
3 CREATE INDEX idx_price_model_id ON car_price (model);
4
5 CREATE INDEX idx_contract_client_id ON contract (client);
6
7 CREATE INDEX idx_prolongation_contract ON prolongation (contract);
8
```

Scratch Pad

Data Output Explain Messages Notifications

CREATE INDEX

Query returned successfully in 72 msec.

Total rows: 0 of 0 Query complete 00:00:00.072

✓ Query returned successfully in 72 msec. ✕

Ln 5, Col 48

## Сравнение времени работы с индексами и без:

```
1 // with indexes
2
3 SELECT
4   r.car,
5   m.name AS model_name,
6   SUM(EXTRACT(EPOCH FROM (r.enddate - r.startdate)) / 3600) AS total_hours
7 FROM contract r
8 JOIN car c ON r.car = c.id
9 JOIN model m ON c.model = m.id
10 GROUP BY r.car, m.name
11 ORDER BY total_hours DESC
12 LIMIT 1;
```

Data Output Explain Messages Notifications

	car	model_name	total_hours
	integer	character varying (100)	numeric
1	3	SUV Y	132.0000000000000000

Total rows: 1 of 1 Query complete 00:00:00.062 Ln 1, Col 16

Query Query History

```
1 // without indexes
2
3 SELECT
4   r.car,
5   m.name AS model_name,
6   SUM(EXTRACT(EPOCH FROM (r.enddate - r.startdate)) / 3600) AS total_hours
7 FROM contract r
8 JOIN car c ON r.car = c.id
9 JOIN model m ON c.model = m.id
10 GROUP BY r.car, m.name
11 ORDER BY total_hours DESC
12 LIMIT 1;
```

Data Output Explain Messages Notifications

	car	model_name	total_hours
	integer	character varying (100)	numeric
1	3	SUV Y	132.0000000000000000

Total rows: 1 of 1 Query complete 00:00:00.121 Ln 1, Col 19

Query
Query History
Scratch Pad

```

1 // with indexes
2
3 WITH downtime AS (
4     SELECT
5         car.id AS car_id,
6         model.id AS model_id,
7         car_price.price_12 AS price,
8         CASE
9             WHEN contract.enddate < NOW() - INTERVAL '1 day' THEN NOW() - INTERVAL '1 day'
10            ELSE contract.enddate
11        END AS end_date,
12        COALESCE(
13            LEAD(contract.startdate) OVER (PARTITION BY car.car_price ORDER BY contract.startdate),
14            NOW()
15        ) AS next_start_date
16    FROM car
17    LEFT JOIN contract ON car.id = contract.car
18    LEFT JOIN model ON car.model = model.id
19    LEFT JOIN car_price ON car_price.model = model.id
20 )
21 SELECT
22     SUM(
23         CASE
24             WHEN EXTRACT(EPOCH FROM (next_start_date - end_date)) * price > 0 THEN EXTRACT(EPOCH FROM (next_start_date - end_date)) * price
25             ELSE 0
26         END
27     ) / 3600 AS hours
28 FROM downtime;

```

Data Output
Explain
Messages
Notifications

hours	numeric
1	52800.000000000000

Total rows: 1 of 1
Query complete 00:00:00.088
Ln 10, Col 16

Query
Query History
Scratch Pad

```

1 // without indexes
2
3 WITH downtime AS (
4     SELECT
5         car.id AS car_id,
6         model.id AS model_id,
7         car_price.price_12 AS price,
8         CASE
9             WHEN contract.enddate < NOW() - INTERVAL '1 day' THEN NOW() - INTERVAL '1 day'
10            ELSE contract.enddate
11        END AS end_date,
12        COALESCE(
13            LEAD(contract.startdate) OVER (PARTITION BY car.car_price ORDER BY contract.startdate),
14            NOW()
15        ) AS next_start_date
16    FROM car
17    LEFT JOIN contract ON car.id = contract.car
18    LEFT JOIN model ON car.model = model.id
19    LEFT JOIN car_price ON car_price.model = model.id
20 )
21 SELECT
22     SUM(
23         CASE
24             WHEN EXTRACT(EPOCH FROM (next_start_date - end_date)) * price > 0 THEN EXTRACT(EPOCH FROM (next_start_date - end_date)) * price
25             ELSE 0
26         END
27     ) / 3600 AS hours
28 FROM downtime;

```

Data Output
Explain
Messages
Notifications

hours	numeric
1	52800.000000000000

Total rows: 1 of 1
Query complete 00:00:00.094
Ln 1, Col 19

## **Вывод**

В ходе лабораторной работы я освоил работу с различными SQL-запросами к базе данных, также создание представлений и индексов. Также сравнил время работы SELECT запросов с индексами и без. Разумеется, с индексами время меньше.