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: 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:

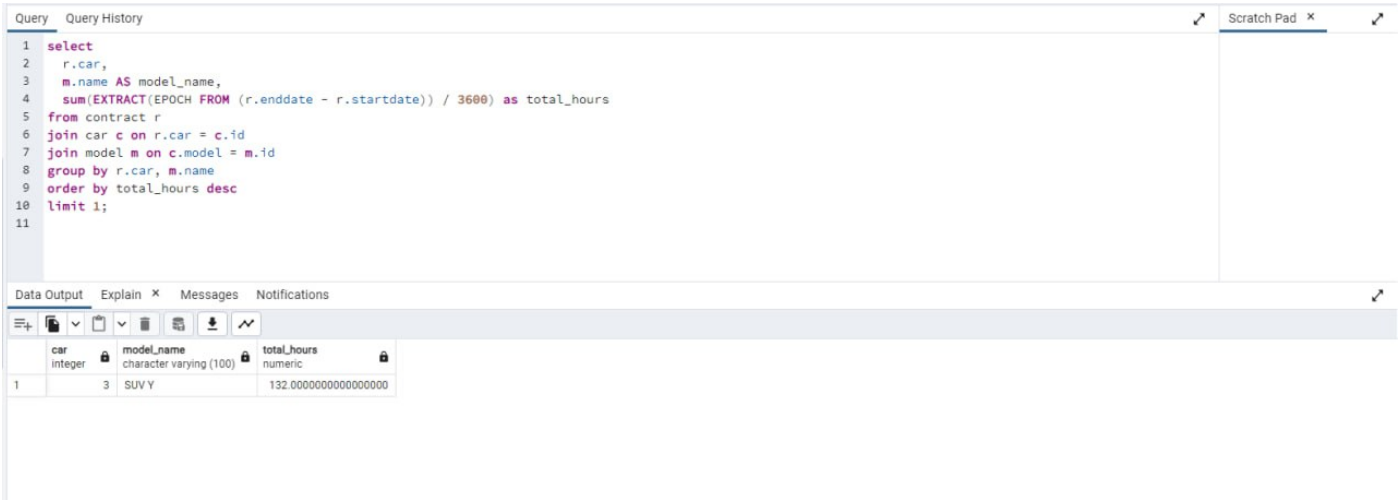
1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) **с использованием подзапросов**.
3. Изучить графическое представление запросов и просмотреть историю запросов.
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:

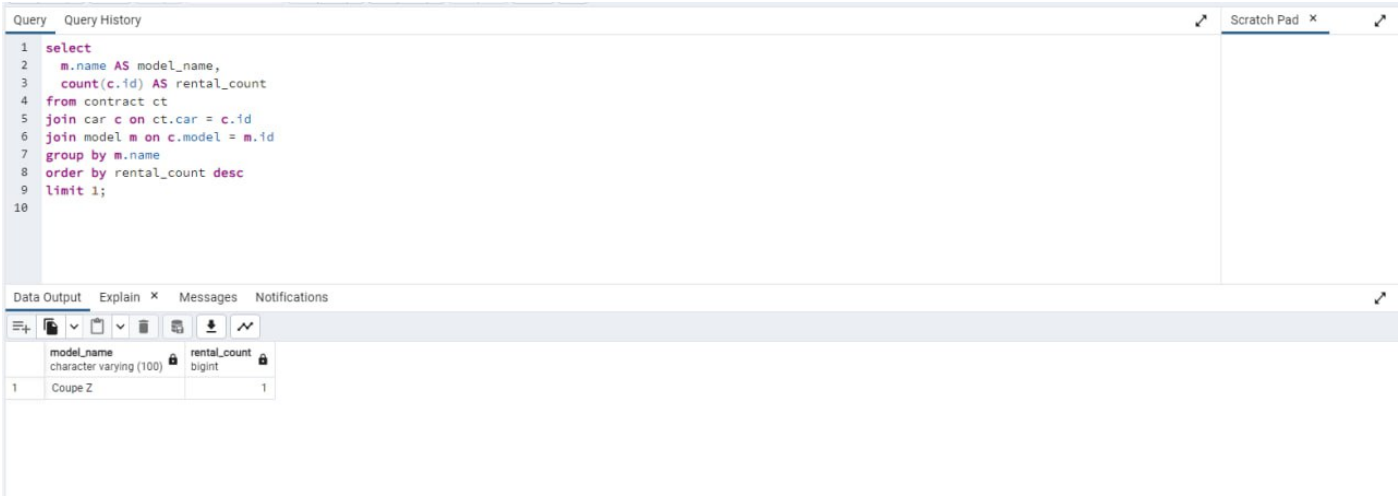
```
select
  r.car,
  m.name AS model_name,
  sum(EXTRACT(EPOCH FROM (r.enddate - r.startdate)) / 3600) as total_hours
from contract r
join car c on r.car = c.id
join model m on c.model = m.id
group by r.car, m.name
order by total_hours desc
limit 1;
```



Автомобили какой марки чаще всего брались в прокат?

Листинг:

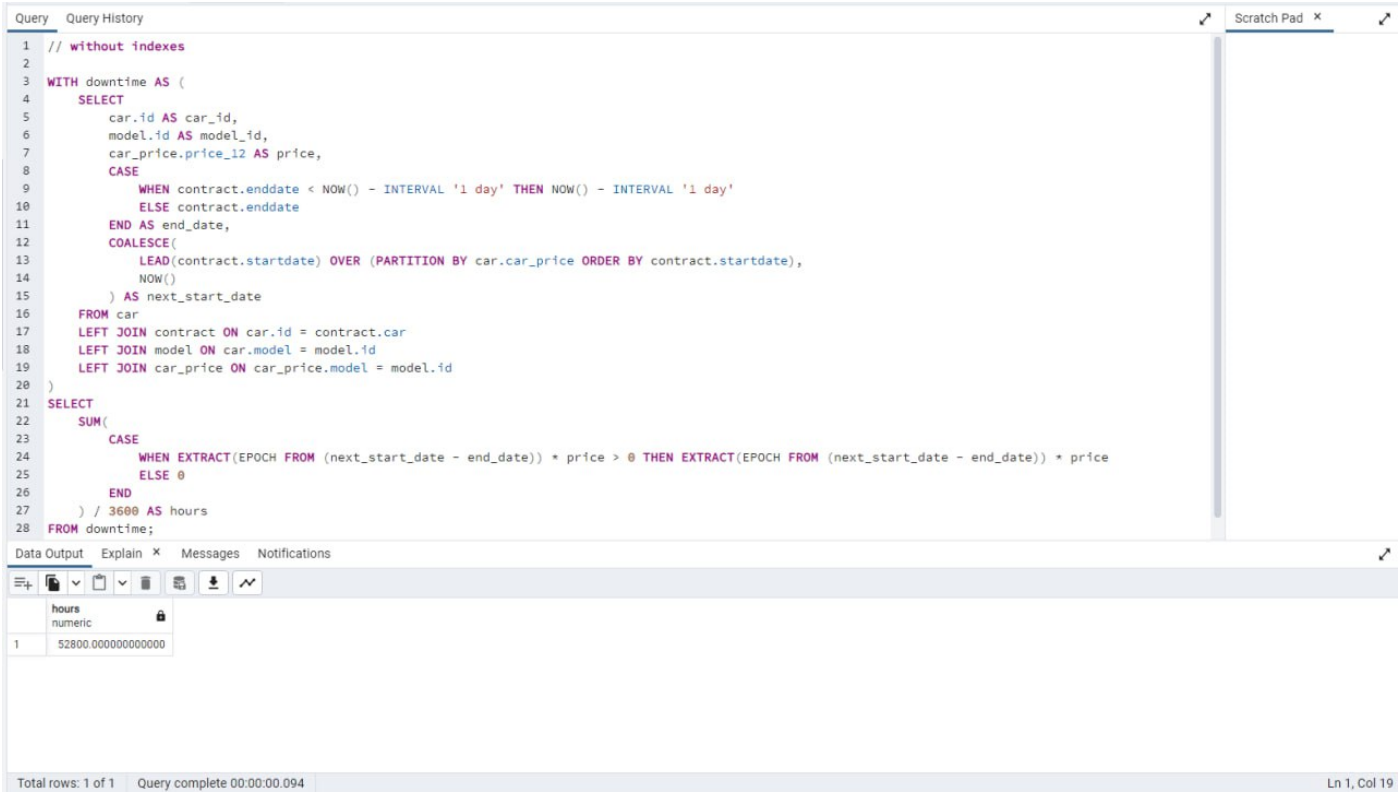
```
select
  m.name AS model_name,
  count(c.id) AS rental_count
from contract ct
join car c on ct.car = c.id
join model m on c.model = m.id
group by m.name
order by rental_count desc
limit 1;
```



Определить убытки от простоя автомобилей за вчерашний день.

Листинг:

```sql
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;
```



Вывести данные автомобиля, имеющего максимальный пробег.

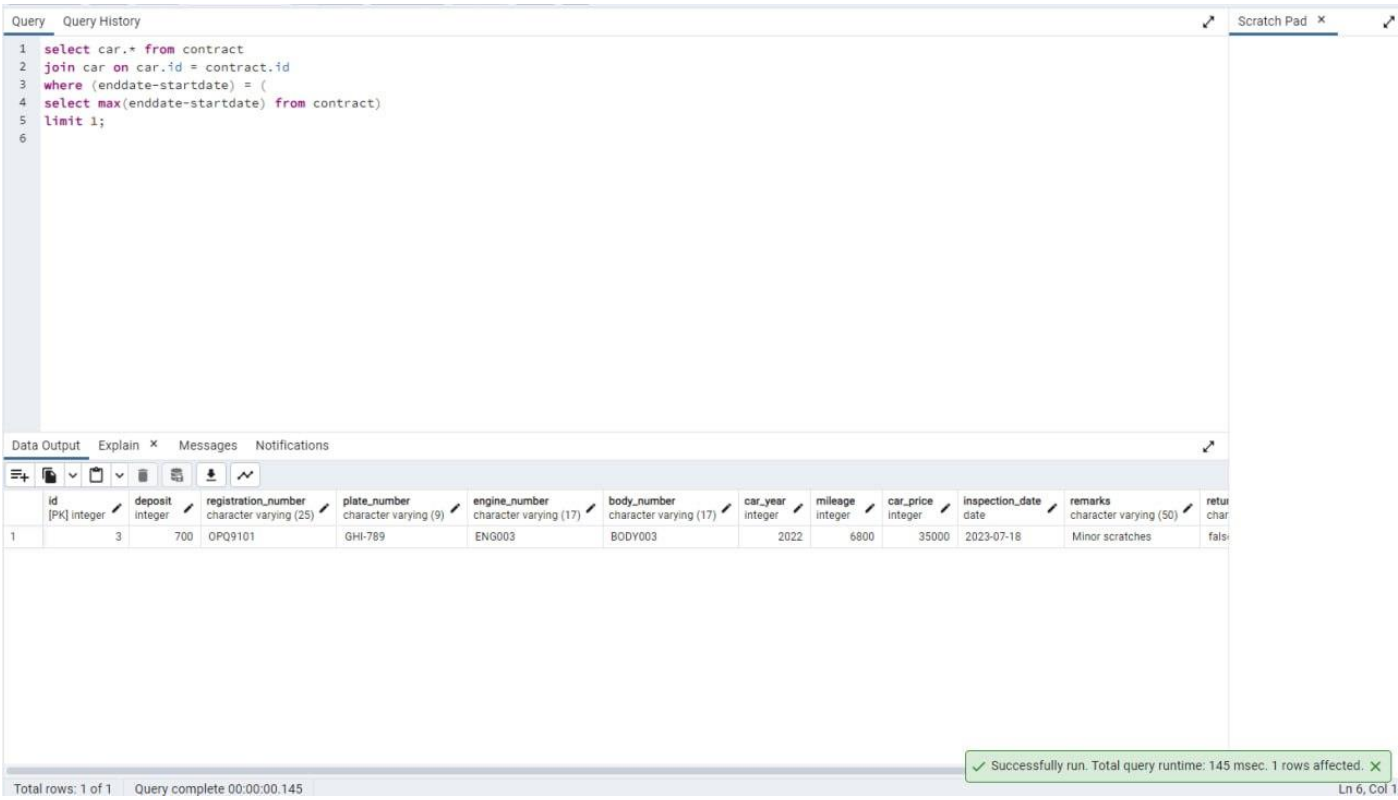
Листинг:

```sql
select * from car
order by mileage desc
limit 1;
```

Какой автомобиль суммарно находился в прокате дольше всех.

Листинг:

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



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

Листинг:

```sql
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  SELECT model.name AS "model", COUNT(*) AS "count"
2  FROM car
3  JOIN model ON car.model = model.id
4  GROUP BY model.name;
```

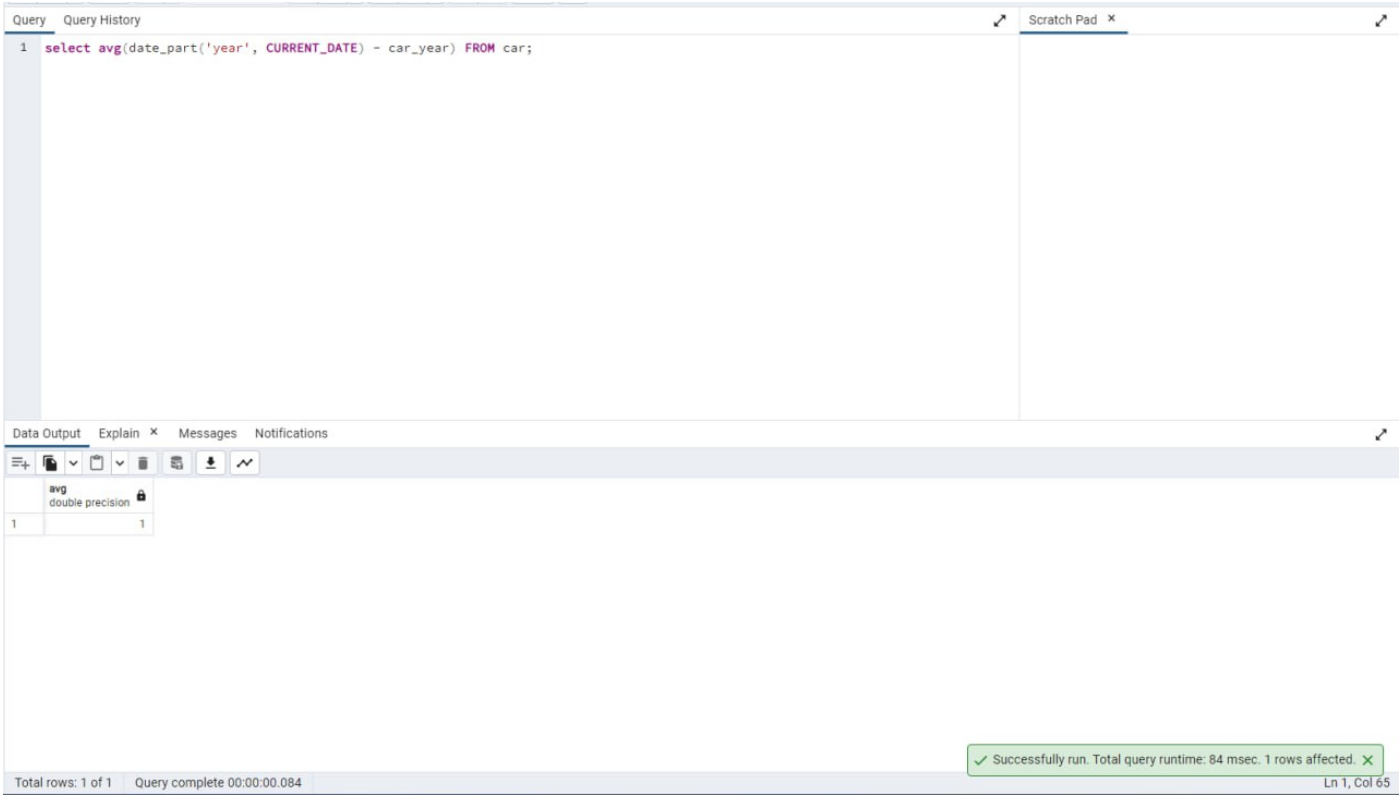
Data Output | Explain | Messages | Notifications

	model character varying (100)	count 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;
```



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;
```

Scratch Pad ×

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';
```

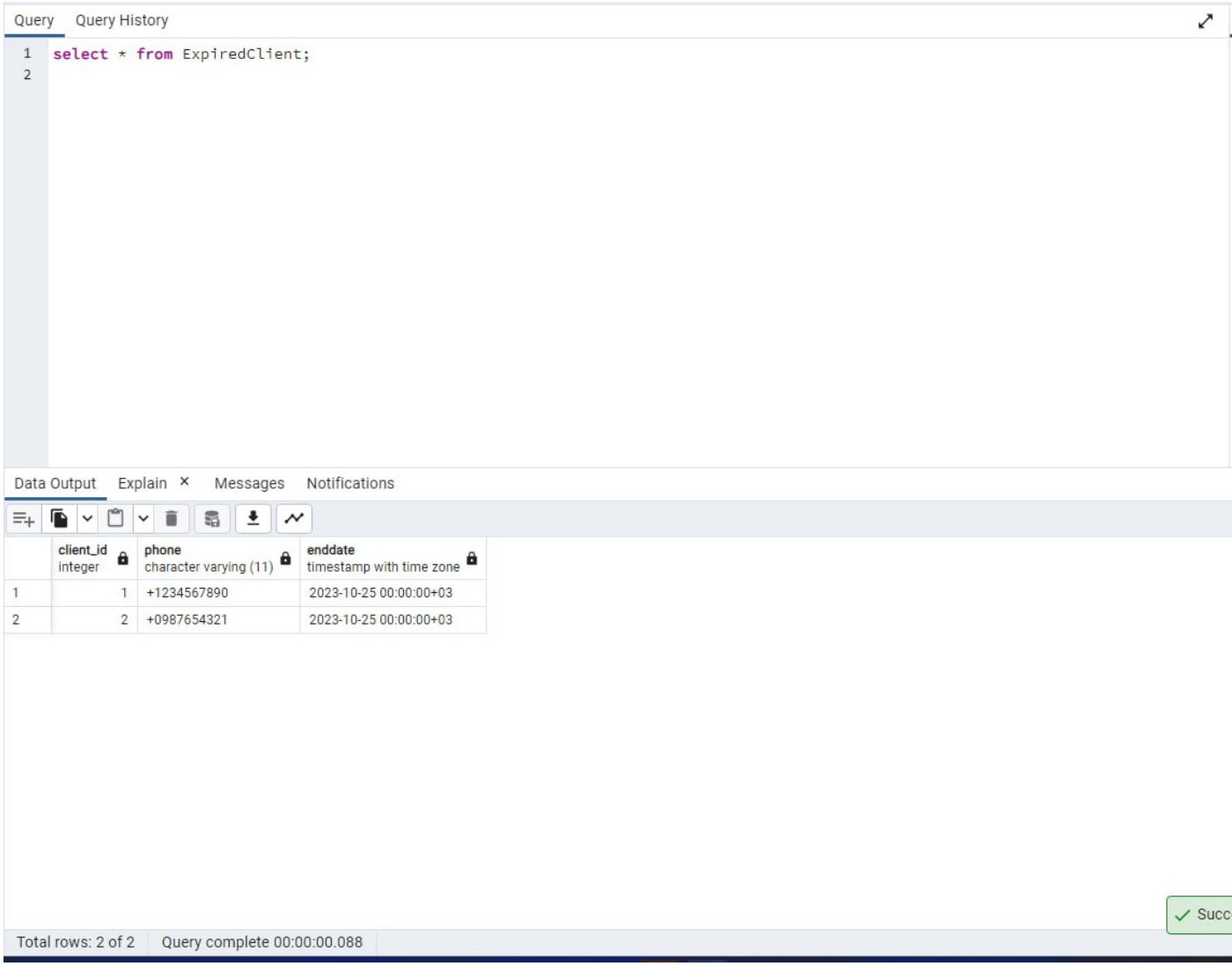
Query | Query History

```sql
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';
```

Data Output | Explain | Messages | Notifications

```
CREATE VIEW

Query returned successfully in 47 msec.
```

Total rows: 2 of 2 | Query complete 00:00:00.047

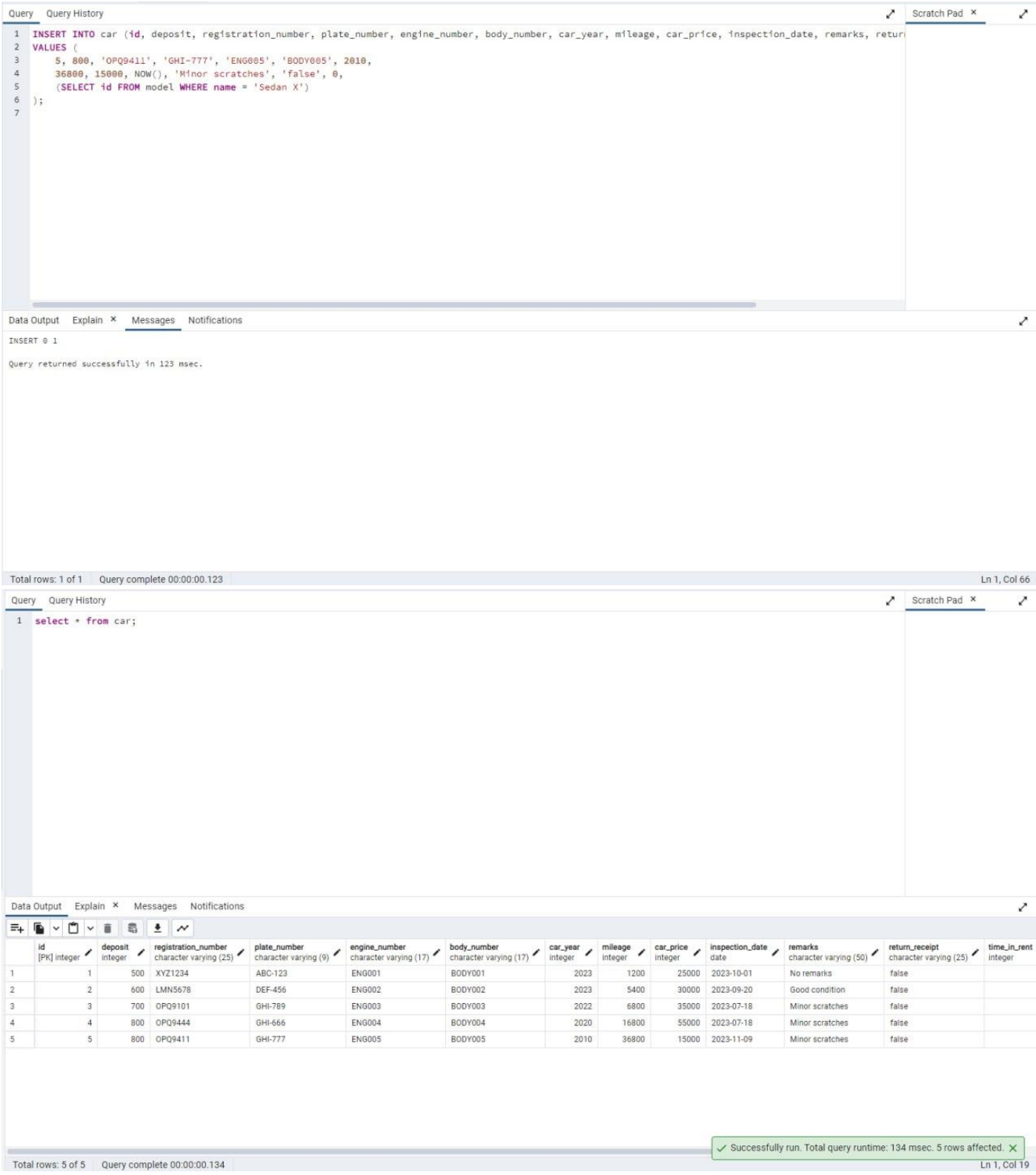


3. Запросы на модификацию данных:

Добавление автомобиля с моделью с конкретным названием

Листинг:

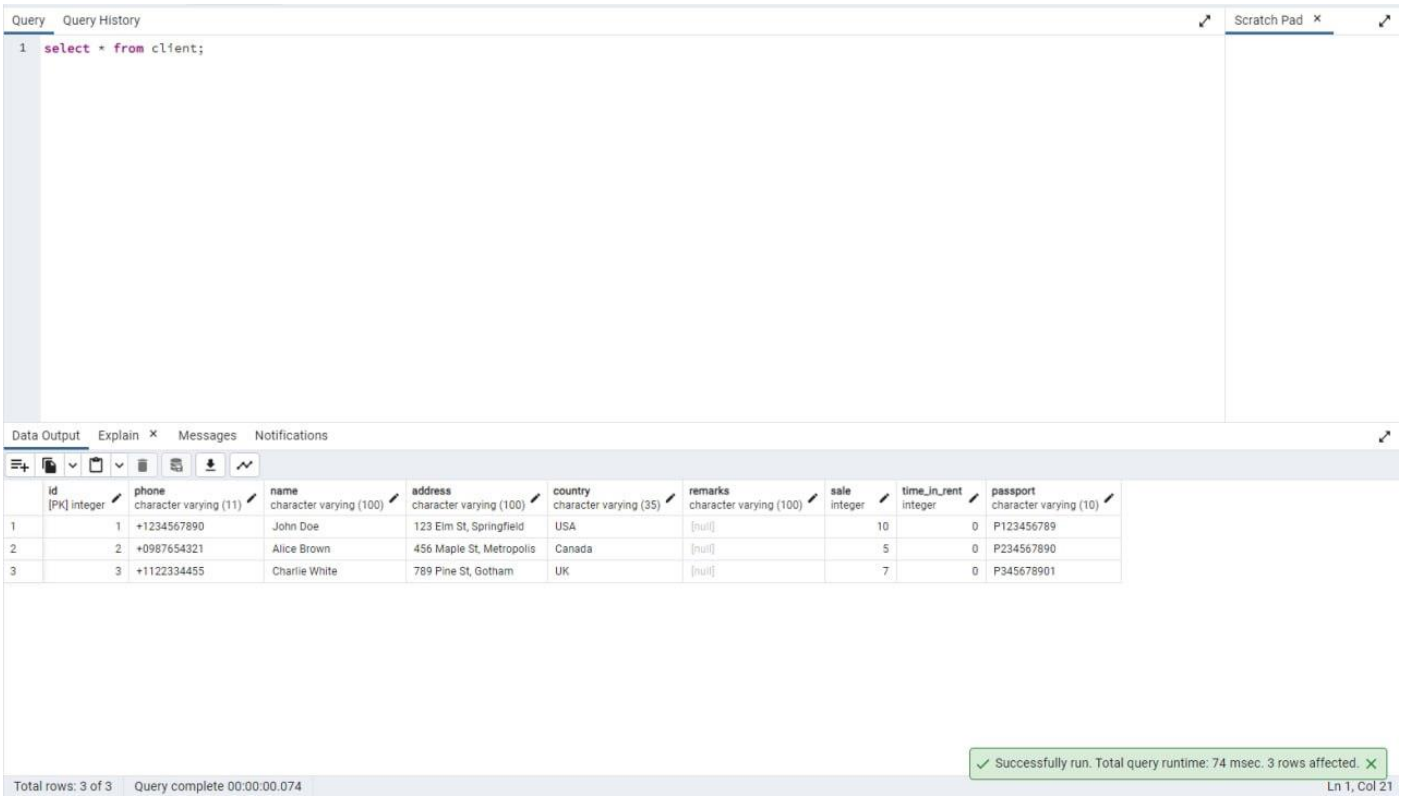
```
insert into car (id, mileage, license_plate, vin, engine_number, body_number, year,
purchase_price, current_value, inspection_date, notes, available, rented_times,
model_id)
values (
    5, 800, 'OPQ9411', 'GHI-777', 'ENG005', 'BODY005', 2010,
    36800, 15000, NOW(), 'Minor scratches', 'false',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

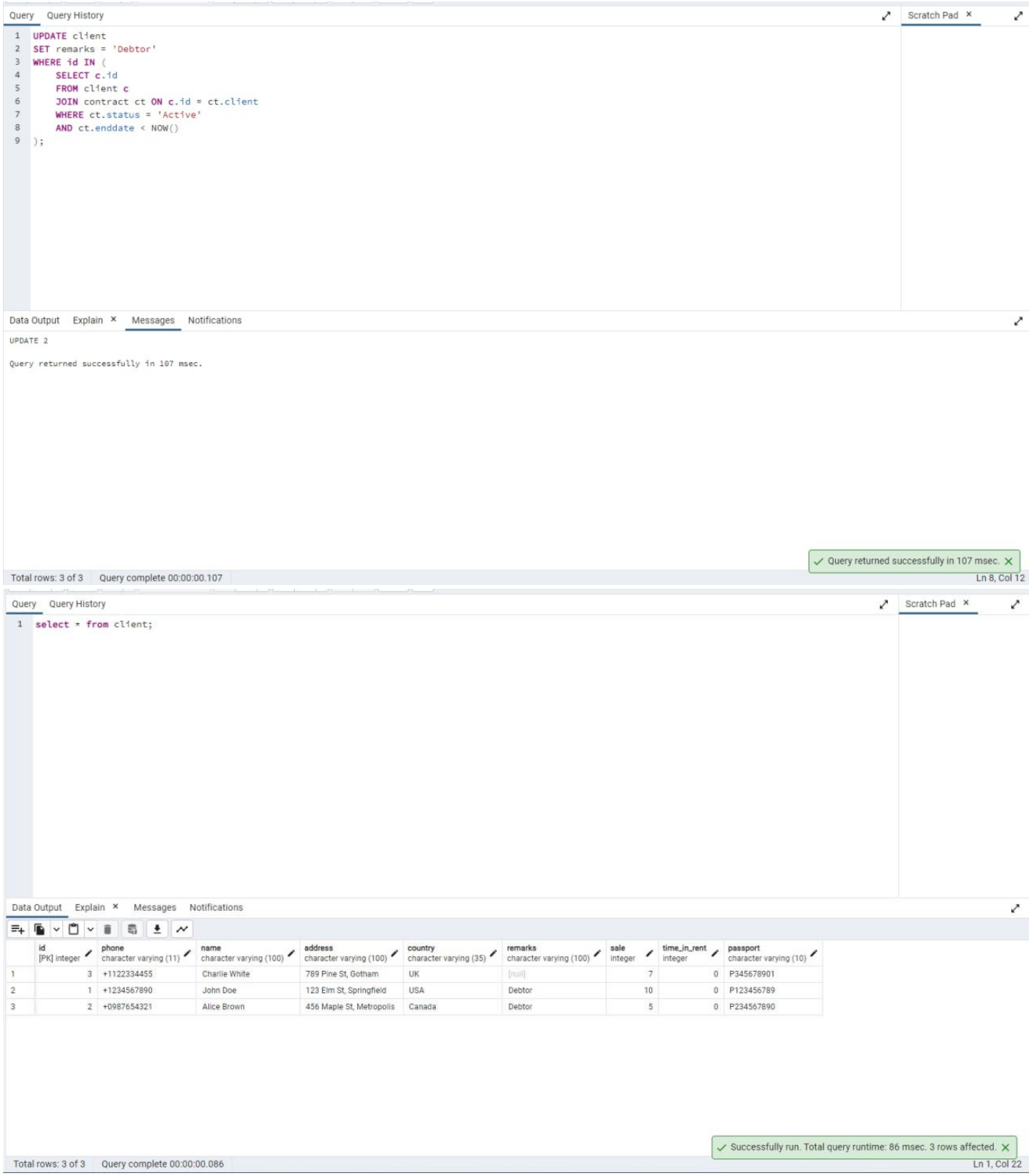


Изменение отметок клиентов-должников, которые не вернули вовремя машину

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
Successfully run. Total query runtime: 74 msec. 3 rows affected.
Total rows: 3 of 3
Query complete 00:00:00.074
Ln 1, Col 21



Обновление даты окончания контракта в зависимости от продлений:

```
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'
)
```

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

Scratch Pad

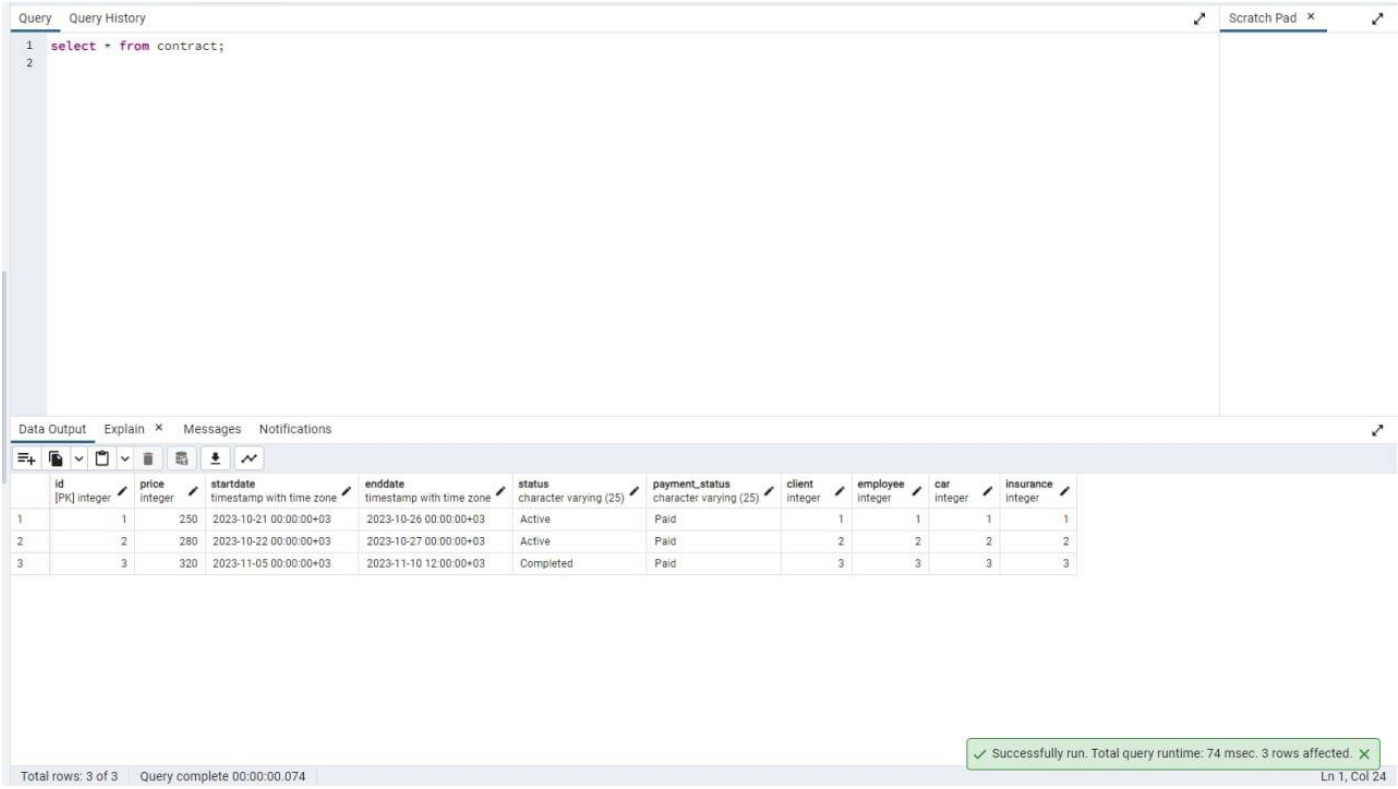
```
1  UPDATE contract AS c
2  SET enddate = c.enddate + COALESCE(
3      (
4          SELECT p.hours
5          FROM prolongation AS p
6          WHERE p.contract = c.id
7      ) * INTERVAL '1 hour',
8      INTERVAL '0 hour'
9  );
10
```

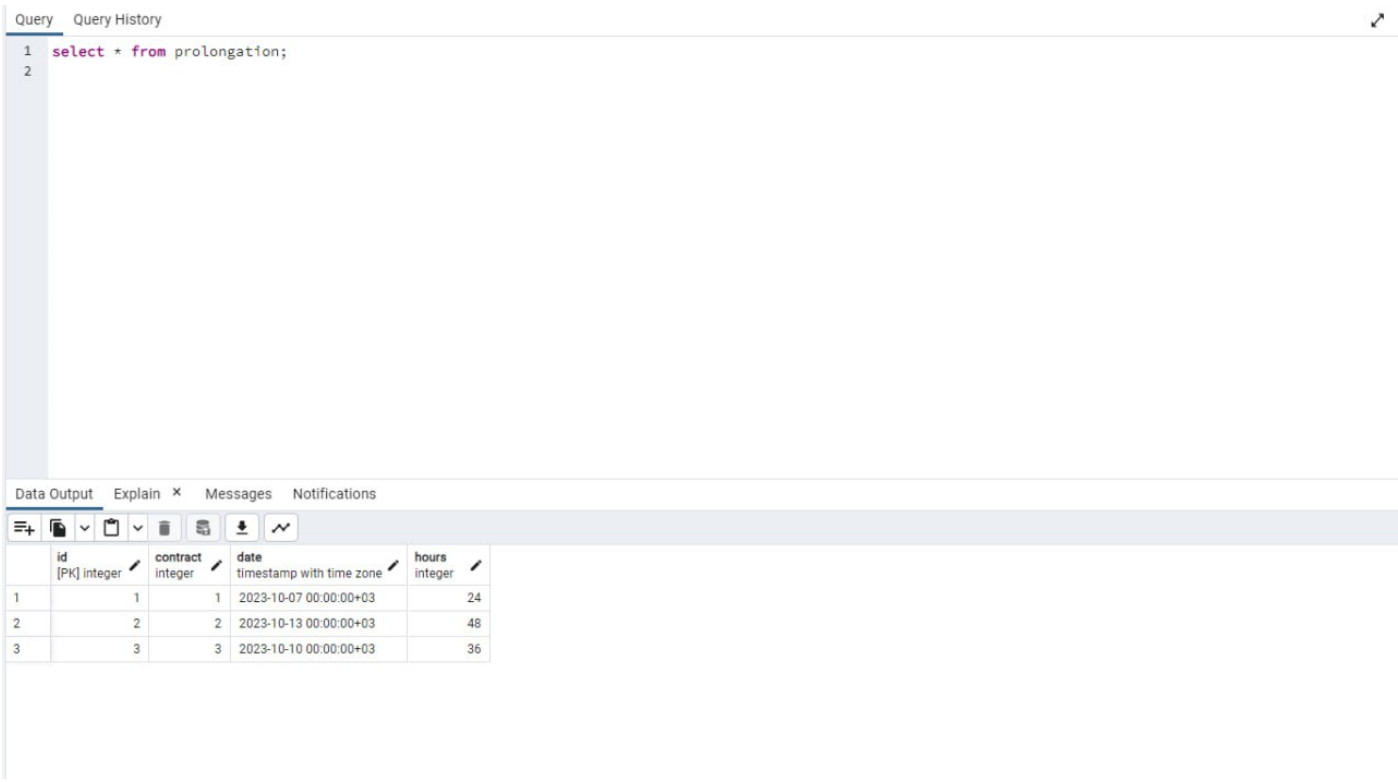
Data Output | Explain | Messages | Notifications

```
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





Удаление неактульных продлений:

Листинг:

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

Query | Query History

```
1  select * from prolongation;
2
```

Data Output | Explain | Messages | Notifications

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

Query | Query History | Scratch Pad

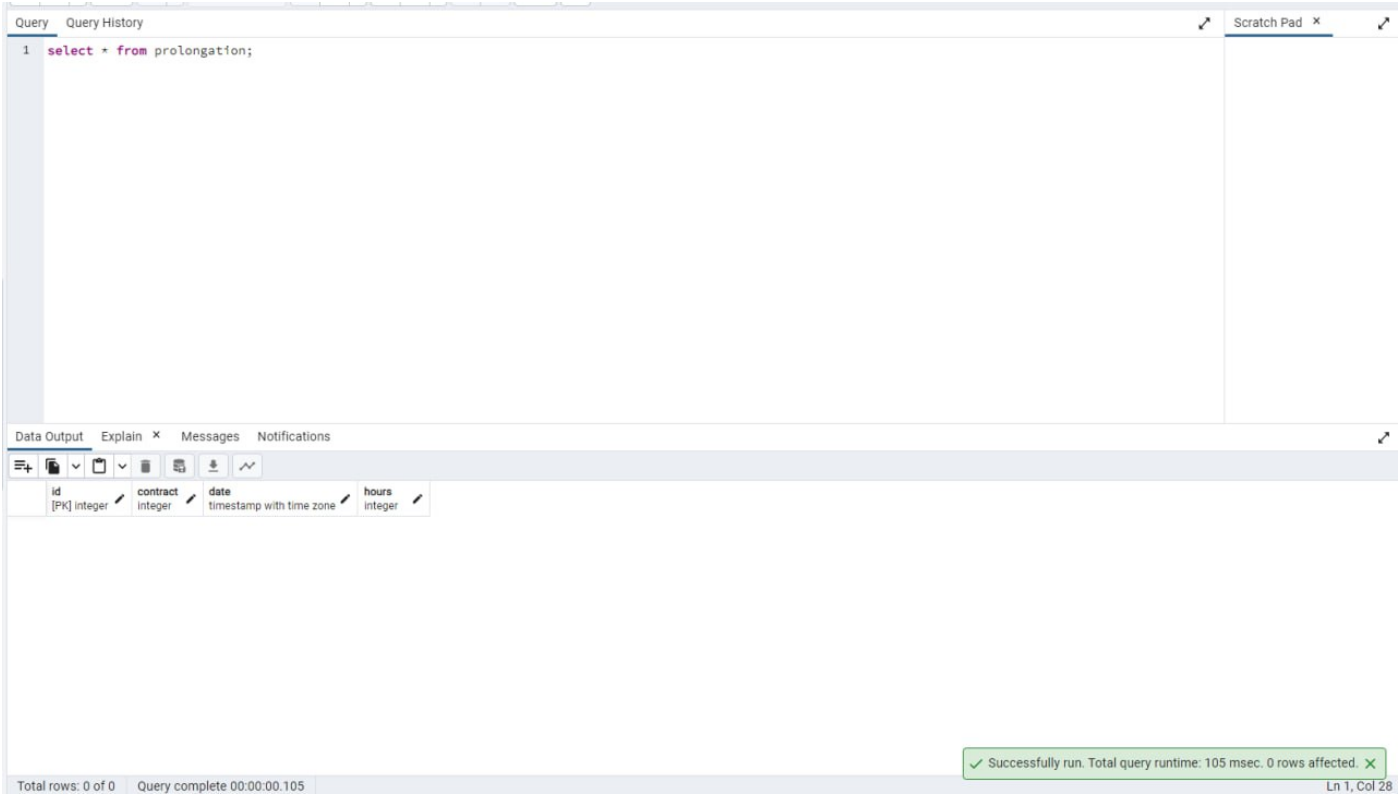
```
1  DELETE FROM prolongation as p
2  WHERE contract IN (
3      SELECT id
4      FROM contract
5      WHERE startdate > p.date
6  );
7
```

Data Output | Explain | Messages | Notifications

```
DELETE 3

Query returned successfully in 106 msec.
```

Query returned successfully in 106 msec.

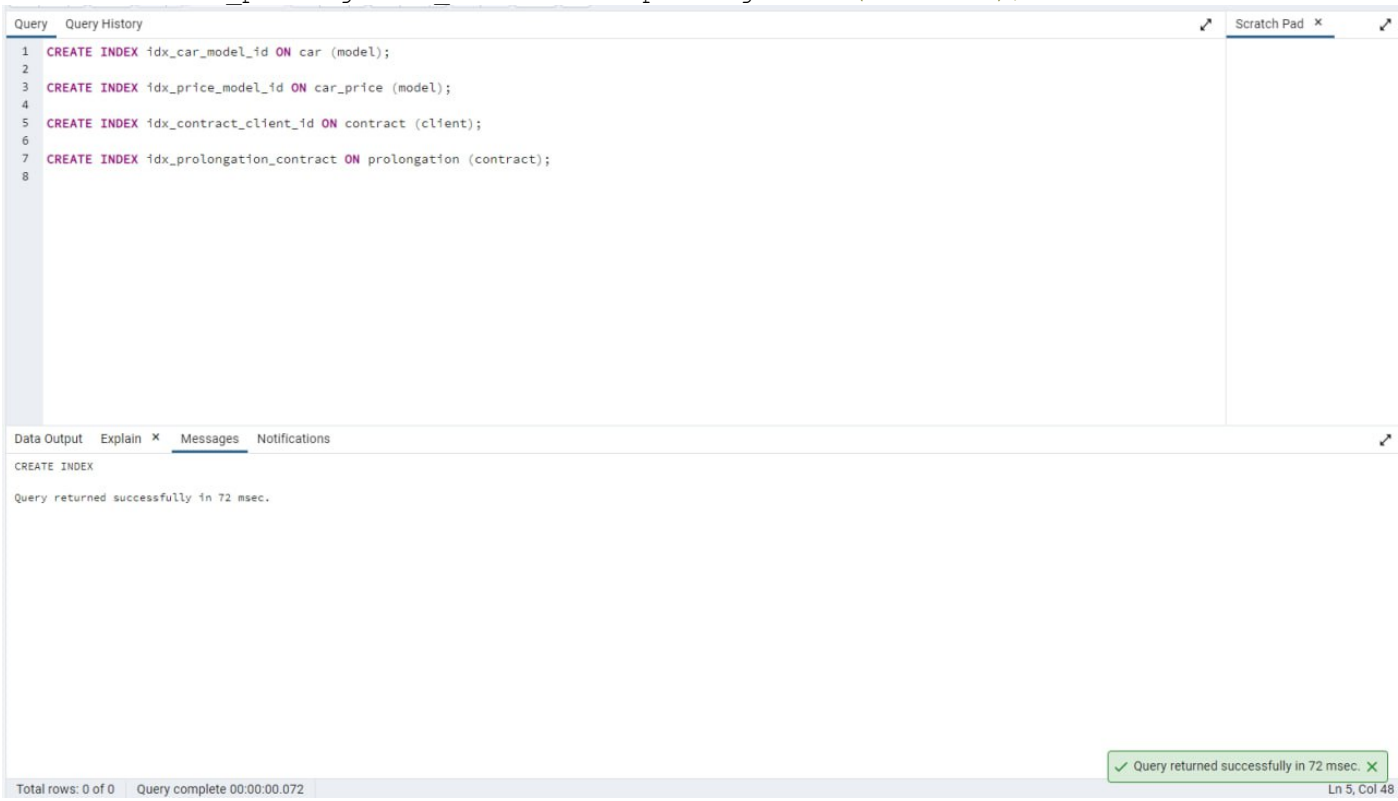
Total rows: 3 of 3 | Query complete 00:00:00.106 | Ln 4, Col 18



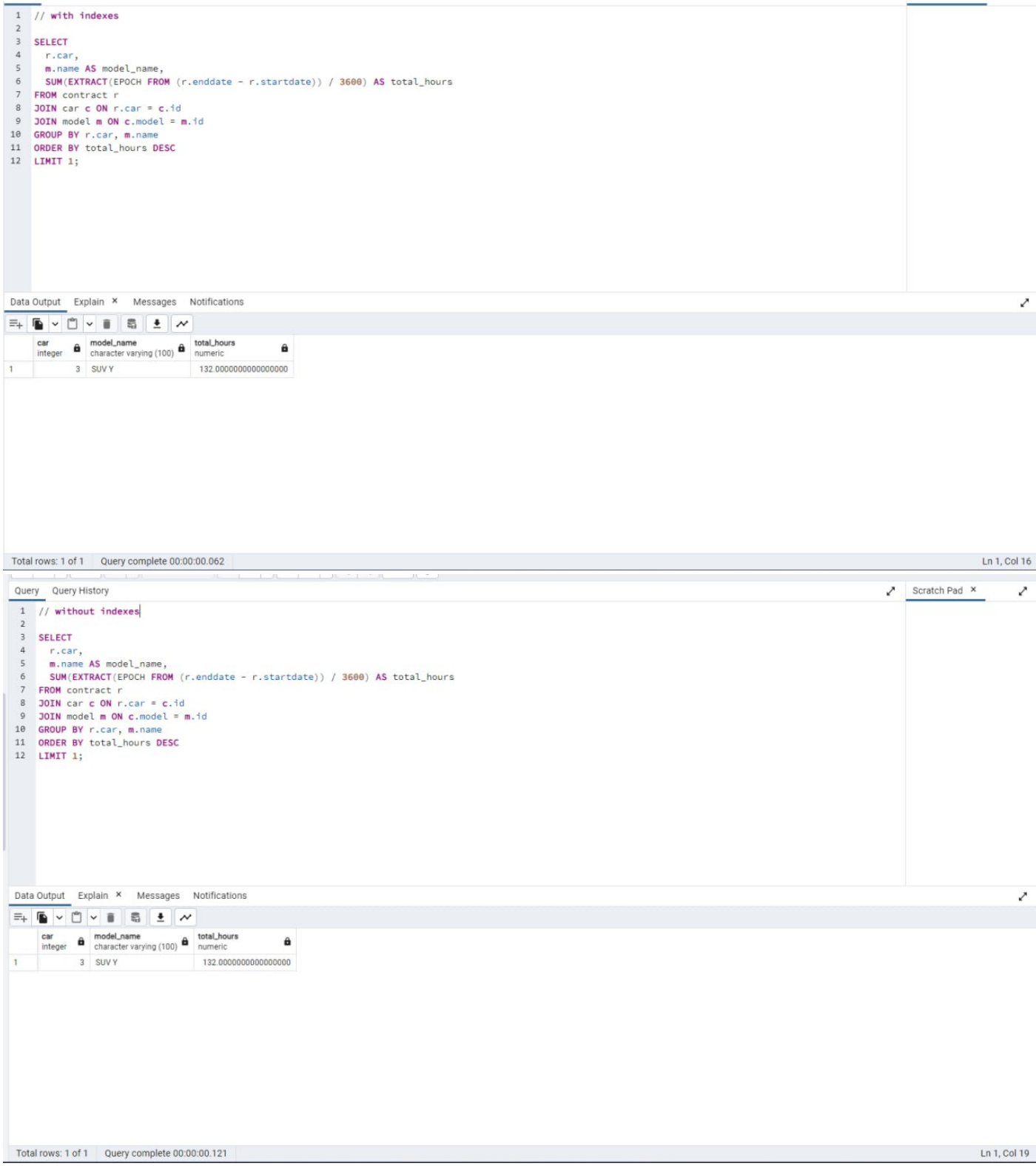
4. Создание индексов

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);
```



Сравнение времени работы с индексами и без:



```
// with indexes

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;
```

Data Output | Explain | Messages | Notifications

	car integer	model_name character varying (100)	total_hours 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 | Scratch Pad

```
// without indexes

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;
```

Data Output | Explain | Messages | Notifications

	car integer	model_name character varying (100)	total_hours 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

```sql
// with indexes

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;
```

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

```sql
// without indexes

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;
```

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.