

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: Цыпандин А.П.

Факультет: ИКТ

Группа: К3239

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

ЦЕЛЬ РАБОТЫ.....	3
ПРАКТИЧЕСКОЕ ЗАДАНИЕ	3
ВАРИАНТ 14. БД «СЛУЖБА ЗАКАЗА ТАКСИ».....	3
ВЫПОЛНЕНИЕ	3
Нотация IDEF1X и ЧЕНА-КИРИЛЛОВА	5
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.

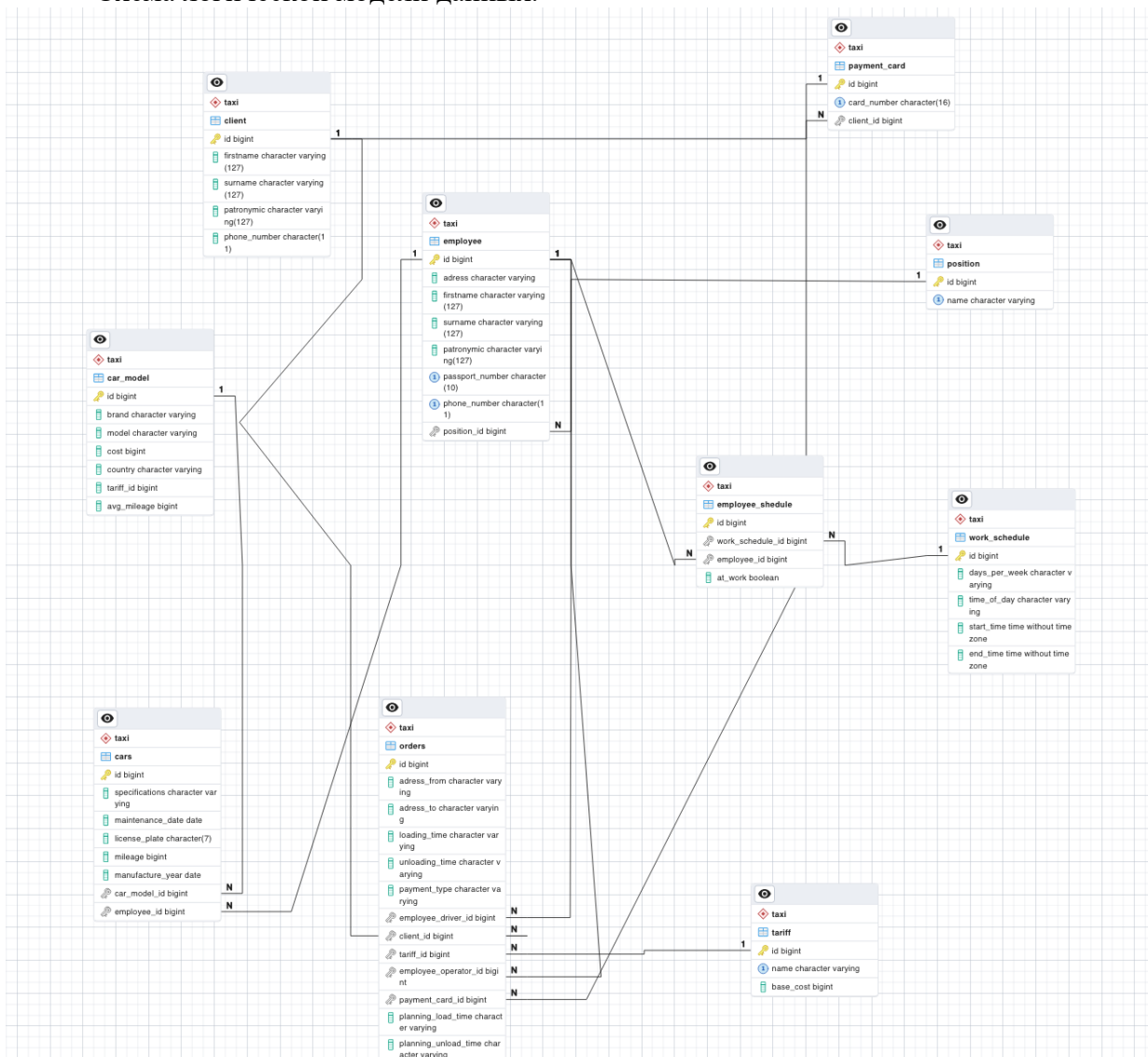
Практическое задание

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.

Выполнение работы

Название БД: Сервис для заказа такси.

Схема логической модели данных:



Запросы к базе данных:

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

```
SELECT e.surname, e.firstname, e.passport_number, COUNT(o.id) AS
orders_count
FROM taxi.employee e
JOIN taxi.orders o ON o.employee_driver_id = e.id
WHERE o.adress_to = 'ул. Тукая, д. 2'
GROUP BY e.surname, e.firstname, e.passport_number
ORDER BY orders_count desc
LIMIT 1;
```

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

```
SELECT c.specifications, c.manufacture_year, c.license_plate,
c.maintenance_date, c.employee_id
FROM taxi.cars c
WHERE c.maintenance_date <= '2023-01-05' AND c.mileage > 250000;
```

3) Сколько раз каждый пассажир воспользовался услугами таксопарка?

```
SELECT c.id, c.firstname, c.surname, COUNT(o.client_id) AS orders_count
FROM taxi.client c
LEFT JOIN taxi.orders o ON o.client_id = c.id
GROUP BY c.id
ORDER BY orders_count DESC;
```

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

```
SELECT c.id, c.firstname, c.surname, COUNT(o.client_id) AS orders_count
FROM taxi.client c
LEFT JOIN taxi.orders o ON o.client_id = c.id
GROUP BY c.id
ORDER BY orders_count DESC
LIMIT 1;
```

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

```
SELECT e.id, e.firstname, e.surname, e.phone_number, cm.cost
FROM taxi.employee e
JOIN taxi.cars c ON e.id = c.employee_id
JOIN taxi.car_model cm ON cm.id = c.car_model_id
GROUP BY e.id, e.firstname, e.surname, e.phone_number, cm.cost
ORDER BY cm.cost DESC
LIMIT 1;
```

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

```
SELECT c.id, c.firstname, c.surname
FROM taxi.client c
JOIN taxi.orders o ON o.client_id = c.id
GROUP BY c.id, c.firstname, c.surname
HAVING COUNT(DISTINCT o.employee_driver_id) = 1;
```

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

```
SELECT c.id, c.license_plate, c.specifications, c.employee_id
FROM taxi.cars c
JOIN taxi.car_model cm ON cm.id = c.car_model_id
WHERE cm.avg_mileage < (c.mileage / (2024 - EXTRACT(YEAR FROM
maintenance_date)));
```

Представления:

1) Содержащее сведения о незанятых на данный момент водителях

```
SELECT e.id, e.firstname, e.surname, e.phone_number
FROM taxi.employee e
JOIN taxi.employee_schedule es ON es.employee_id = e.id
WHERE at_work = true;
```

2) Зарплата всех водителей за вчерашний день.

```
SELECT e.id, e.firstname, e.surname, e.phone_number, (p.salary_per_hour * 4)
AS salary
FROM taxi.employee e
JOIN taxi.position p ON p.id = e.position_id
WHERE p.name IN ('Водитель', 'Водитель комфорт')
ORDER BY salary DESC;
```

Подзапросы:

1) Запрос, который удаляет тех клиентов, у которых не привязана карта оплаты, нет ни одного заказа или заказы которые оплачены наличным расчётом.

```
DELETE FROM taxi.client c
WHERE NOT EXISTS (
    SELECT 1
    FROM taxi.orders o
    WHERE o.client_id = c.id OR o.payment_card_id IS null
)
AND NOT EXISTS (
    SELECT 1
    FROM taxi.payment_card pc
    WHERE pc.client_id = c.id
);
```

Клиенты которые подходят под описание под индексом 150, 151, 152:

144	147	Щербakov	Всеслав	Валерьянович	76423468110
145	148	Афанасьев	Евгений	Валентинович	72154751678
146	149	Носкова	Эмилия	Эльдаровна	77232086108
147	150	Василий	Удаленко	Иосифович	89124238712
148	151	Петр	Удаленков	Валерьевич	89124238412
149	152	Петр	Удаленков	Петрович	89124271632
150	153	Иван	Иванов	Кириллов	89124295544

После выполнения запроса:

142	145	Алина	Федоровна	Рыбакова	76359694742
143	146	Королев	Никандр	Власович	76914077824
144	147	Щербаков	Всеслав	Валерьянович	76423468110
145	148	Афанасьев	Евгений	Валентинович	72154751678
146	149	Носкова	Эмилия	Эльдаровна	77232086108
147	153	Иван	Иванов	Кириллов	89124295544

2) Запрос, который добавляет для сотрудника – водителя автомобиль по заданным параметрам

```
WITH inserted_car_model AS (
    INSERT INTO taxi.car_model (brand, model, cost, country, tariff_id,
avg_mileage)
    VALUES ('Honda', 'Fit', '450000', 'Japan', 1, 15000)
    RETURNING id
)
```

```
INSERT INTO taxi.cars (id, specifications, maintenance_date, license_plate,
mileage, manufacture_year, car_model_id, employee_id)
```

```
SELECT
```

```
18,
'Motor 1.5, auto',
'2023-11-15',
'A413ГД',
111034,
'2017-02-15',
icm.id,
e.id
```

```
FROM taxi.employee e
```

```
JOIN inserted_car_model icm ON 1 = 1
```

```
WHERE
```

```
e.firstname || ' ' || e.surname || ' ' || e.patronymic = 'Василий
Петров Петрович'
```

```
AND passport_number = '9818765455'
```

```
AND e.position_id = 1;
```

Таблица cars и car_model до запроса:

13	13	Motor 2.1i; Manual	2022-11-20	P970Ч0	192667	2000-05-13	6	26
14	14	Motor 1.5i; Manual	2021-08-18	K161BЭ	105459	2011-01-19	7	27
15	15	Motor 2.2i; Manual	2021-08-25	Ш577РК	244624	2012-05-17	7	28
16	16	Motor 1.9i; Manual	2020-03-25	3751ГО	229424	2017-11-26	8	29
17	17	Motor 1.7i; Manual	2021-08-22	E648X3	248532	2000-01-05	9	30

5	5	VAZ	21083	240000	Russia	1	10000
6	6	toyota	celica	650000	Japan	1	16000
7	7	Nissan	SkyLine	450000	Japan	1	15000
8	8	peugot	206	600000	France	1	18000
9	9	mercedes	s-class	4500000	Germany	2	14500

Таблица **cars** и **car_model** после запроса, а также таблица **employee**:

16	16	Motor 1.9; Manual	2020-03-25	3751ГО	229424	2017-11-26	8	29
17	17	Motor 1.7; Manual	2021-08-22	E648X3	248532	2000-01-05	9	30
18	18	Motor 1.5, auto	2023-11-15	A413ГД	111034	2017-02-15	10	32

8	8	peugot	206	600000	France	1	18000
9	9	mercedes	s-class	4500000	Germany	2	14500
10	10	Honda	Fit	450000	Japan	1	15000

27	27	наб. Лазо, д. 7/7 стр. 5/8, 50	Владлен	Колобова	Геннадиевич	2497237661	75110255942	1
28	28	алл. Шевченко, д. 4 стр. 67, 17	Лука	Киселев	Кузьминична	6079770925	79651988842	1
29	29	пер. Первомайский, д. 60 стр. 707, 92	Август	Потапова	Горбачева	7553613986	79840212470	2
30	30	наб. Заовражная, д. 3 стр. 3/3, 63	Твердислав	Моисеев	Павлова	9543666304	71474032276	2
31	32	Краснодыбская 1/2, кв.7	Василий	Петров	Петрович	9818765455	78645632451	1

Total rows: 31 of 31 Query complete 00:00:00 137

3) Запрос, который обновляет дату последнего ТО на новое, для автомобиля, который привязан к сотруднику с заданными данными

```

UPDATE taxi.cars
SET maintenance_date = '2024.01.10', license_plate = 'O184ГО'
WHERE employee_id = (
    SELECT id
    FROM taxi.employee e
    WHERE CONCAT(e.firstname, ' ', e.surname, ' ', e.patronymic) =
'Василий Петров Петрович'
    AND e.passport_number = '9818765455')
AND car_model_id = (
    SELECT id
    FROM taxi.car_model cm
    WHERE cm.brand = 'Honda'
    AND cm.model = 'Fit');
```

Таблица **cars** до запроса:

15	15	Motor 2.2; Manual	2021-08-25	Ш577РК	244624	2012-05-17	7	28
16	16	Motor 1.9; Manual	2020-03-25	3751ГО	229424	2017-11-26	8	29
17	17	Motor 1.7; Manual	2021-08-22	E648X3	248532	2000-01-05	9	30
18	18	Motor 1.5, auto	2023-11-15	A413ГД	111034	2017-02-15	10	32

Таблица cars и car_model после запроса:

16	16	Motor 1.9; Manual	2020-03-25	3751ГO	229424	2017-11-26	8	29
17	17	Motor 1.7; Manual	2021-08-22	E648X3	248532	2000-01-05	9	30
18	18	Motor 1.5, auto	2024-01-10	O184ГO	111034	2017-02-15	10	32

Индексы:

Тестовый запрос и план запроса:

EXPLAIN ANALYZE SELECT

```
o.id,
o.adress_to,
o.adress_from,
o.planning_load_time,
o.planning_unload_time,
c.firstname,
c.surname,
e.firstname,
e.surname,
t.name
```

FROM taxi.orders o

JOIN taxi.client c **ON** o.client_id = c.id

JOIN taxi.tariff t **ON** o.tariff_id = t.id

JOIN taxi.employee e **ON** o.employee_driver_id = e.id

WHERE e.position_id = 1;

	QUERY PLAN	
	text	
1	Hash Join (cost=7.56..13.18 rows=23 width=242) (actual time=0.237..0.368 rows=31 loops=1)	
2	Hash Cond: (o.tariff_id = t.id)	
3	-> Hash Join (cost=6.52..11.94 rows=23 width=218) (actual time=0.207..0.323 rows=31 loops=1)	
4	Hash Cond: (c.id = o.client_id)	
5	-> Seq Scan on client c (cost=0.00..4.46 rows=146 width=39) (actual time=0.012..0.041 rows=147 loops=1)	
6	-> Hash (cost=6.23..6.23 rows=23 width=195) (actual time=0.146..0.148 rows=31 loops=1)	
7	Buckets: 1024 Batches: 1 Memory Usage: 14kB	
8	-> Hash Join (cost=2.56..6.23 rows=23 width=195) (actual time=0.070..0.113 rows=31 loops=1)	
9	Hash Cond: (o.employee_driver_id = e.id)	
10	-> Seq Scan on orders o (cost=0.00..3.51 rows=51 width=174) (actual time=0.013..0.037 rows=35 loops=1)	
11	-> Hash (cost=2.39..2.39 rows=14 width=37) (actual time=0.033..0.034 rows=15 loops=1)	
12	Buckets: 1024 Batches: 1 Memory Usage: 10kB	
13	-> Seq Scan on employee e (cost=0.00..2.39 rows=14 width=37) (actual time=0.007..0.019 rows=15 loop...	
14	Filter: (position_id = 1)	
15	Rows Removed by Filter: 16	
16	-> Hash (cost=1.02..1.02 rows=2 width=40) (actual time=0.022..0.022 rows=2 loops=1)	
17	Buckets: 1024 Batches: 1 Memory Usage: 9kB	
18	-> Seq Scan on tariff t (cost=0.00..1.02 rows=2 width=40) (actual time=0.015..0.017 rows=2 loops=1)	
19	Planning Time: 2.469 ms	
20	Execution Time: 0.463 ms	

Создание индексов и план запроса с индексами:

CREATE INDEX idx_firstname **ON** taxi.employee(firstname);

CREATE INDEX idx_surname **ON** taxi.employee(surname);

CREATE INDEX idx_phone_number **ON** taxi.employee(phone);

	QUERY PLAN text	
1	Hash Join (cost=7.26..12.56 rows=16 width=242) (actual time=0.189..0.288 rows=31 loops=1)	
2	Hash Cond: (o.tariff_id = t.id)	
3	-> Hash Join (cost=6.22..11.38 rows=16 width=218) (actual time=0.151..0.233 rows=31 loops=1)	
4	Hash Cond: (c.id = o.client_id)	
5	-> Seq Scan on client c (cost=0.00..4.46 rows=146 width=39) (actual time=0.006..0.035 rows=147 loops=1)	
6	-> Hash (cost=6.02..6.02 rows=16 width=195) (actual time=0.137..0.139 rows=31 loops=1)	
7	Buckets: 1024 Batches: 1 Memory Usage: 14kB	
8	-> Hash Join (cost=2.56..6.02 rows=16 width=195) (actual time=0.061..0.103 rows=31 loops=1)	
9	Hash Cond: (o.employee_driver_id = e.id)	
10	-> Seq Scan on orders o (cost=0.00..3.35 rows=35 width=174) (actual time=0.005..0.029 rows=35 loops=1)	
11	-> Hash (cost=2.39..2.39 rows=14 width=37) (actual time=0.033..0.033 rows=15 loops=1)	
12	Buckets: 1024 Batches: 1 Memory Usage: 10kB	
13	-> Seq Scan on employee e (cost=0.00..2.39 rows=14 width=37) (actual time=0.007..0.020 rows=15 loop...	
14	Filter: (position_id = 1)	
15	Rows Removed by Filter: 16	
16	-> Hash (cost=1.02..1.02 rows=2 width=40) (actual time=0.019..0.019 rows=2 loops=1)	
17	Buckets: 1024 Batches: 1 Memory Usage: 9kB	
18	-> Seq Scan on tariff t (cost=0.00..1.02 rows=2 width=40) (actual time=0.012..0.014 rows=2 loops=1)	
19	Planning Time: 1.292 ms	
20	Execution Time: 0.368 ms	

Вывод

В ходе выполнения данной лабораторной работы мы рассмотрели использование запросов SELECT, UPDATE и DELETE с применением подзапросов в контексте базы данных с несколькими связанными таблицами. Также, в ходе лабораторной работы были рассмотрены индексы и их влияние на производительность запросов. Индексы были созданы для полей, используемых в условиях JOIN, чтобы ускорить поиск и сопоставление данных. В результате использования индексов мы получили улучшение производительности запросов. При работе с большими объемами данных, отличие было бы еще больше.