Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет: **Инфокоммуникационных технологий**
Образовательная программа: **Интеллектуальные системы в гуманитарной сфере**
Направление подготовки: **45.03.04 Интеллектуальные системы в гуманитарной сфере**

Лабораторная работа №2

«Запросы на выборку и модификацию данных, представления и индексы в PostgreSQL»

по дисциплине:
«ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ БАЗ ДАННЫХ»

Выполнила:
Чагина Вероника Александровна,
группа К32422
Преподаватель:
Говорова Марина Михайловна

Санкт-Петербург
2023

`

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:

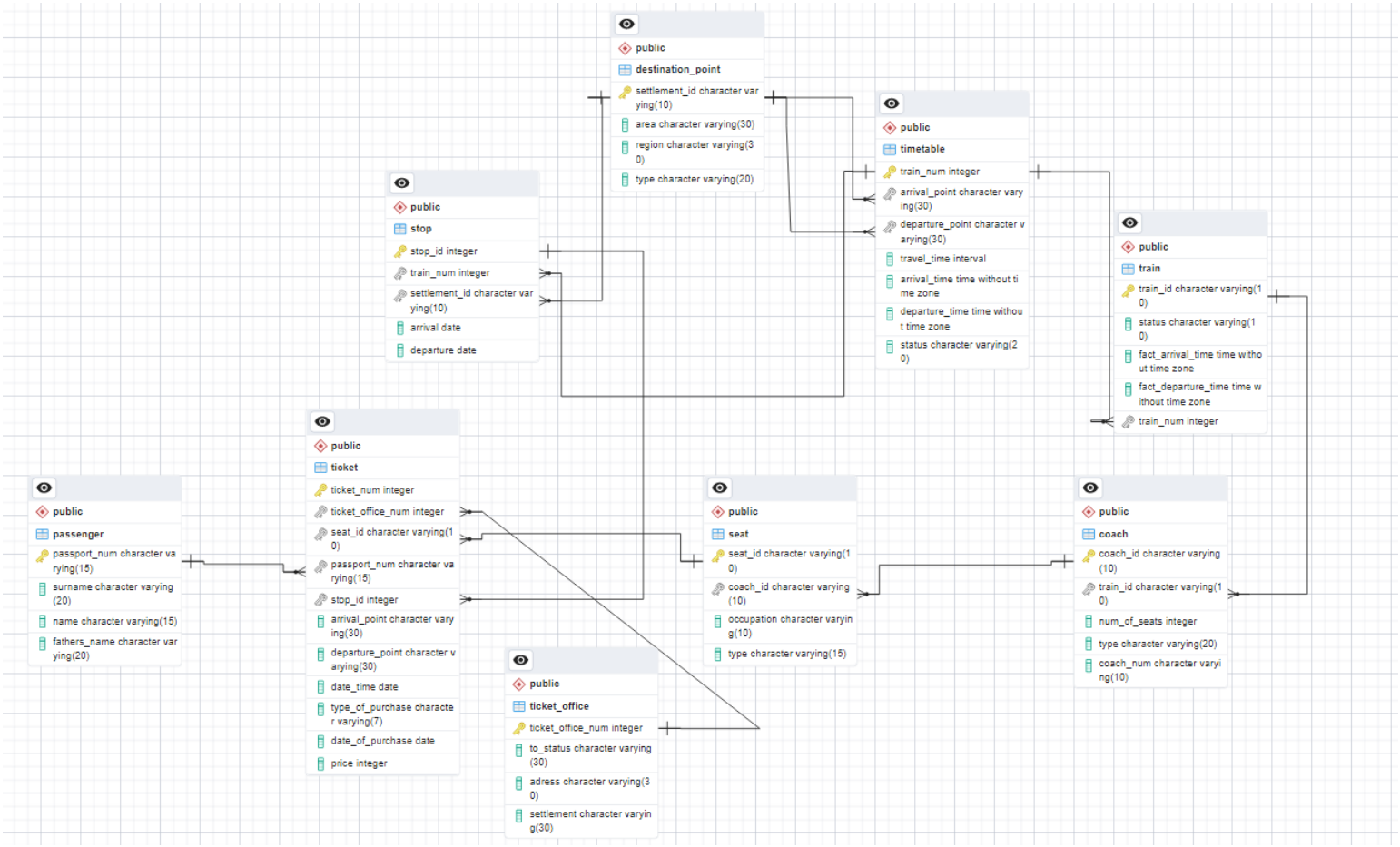
1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и просмотреть историю запросов.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

`

Выполнение:

1. *Название: БД «Пассажир»*
2. *Схема логической модели базы данных, сгенерированная в Generate ERD*

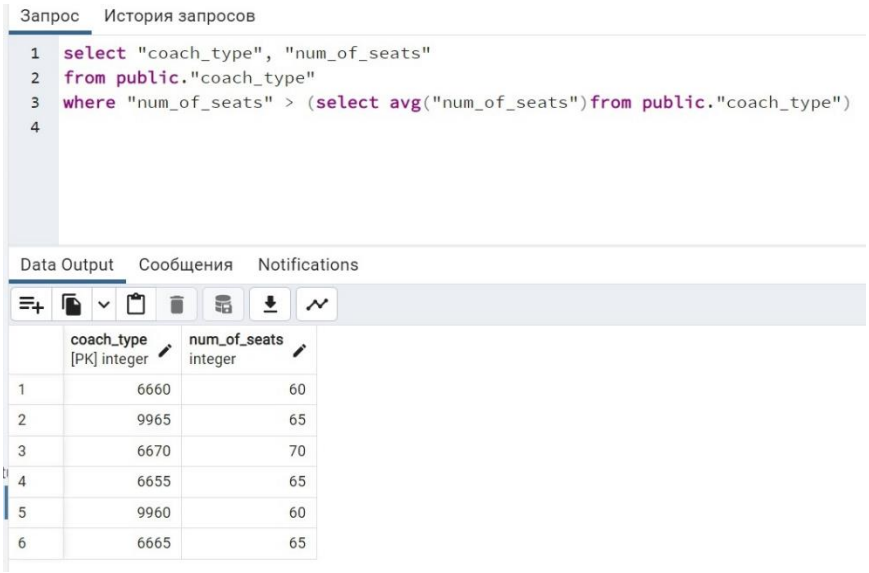
Рисунок 1



`

1. Запросы к базе данных

1.1 SELECT-запрос для вывода типа вагона с числом мест выше среднего



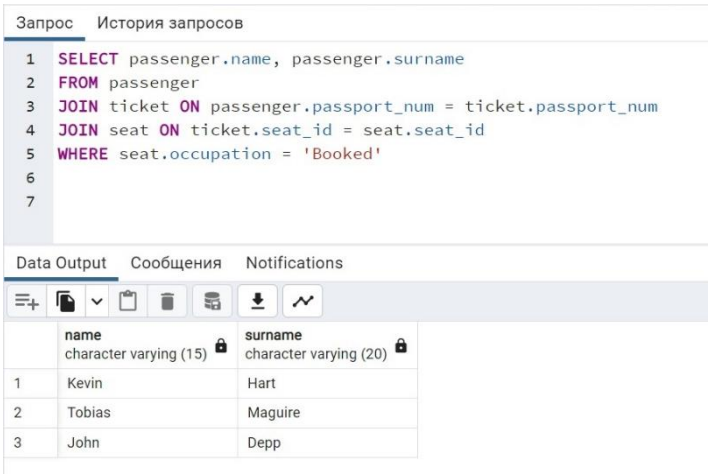
Запрос История запросов

```
select "coach_type", "num_of_seats"
from public."coach_type"
where "num_of_seats" > (select avg("num_of_seats")from public."coach_type")
```

Data Output Сообщения Notifications

	coach_type [PK] integer	num_of_seats integer
1	6660	60
2	9965	65
3	6670	70
4	6655	65
5	9960	60
6	6665	65

1.2 SELECT-запрос для вывода пассажиров, не оплативших билет



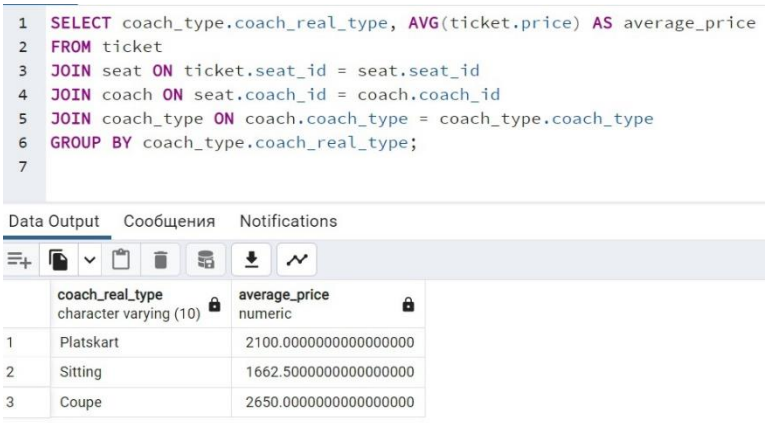
Запрос История запросов

```
SELECT passenger.name, passenger.surname
FROM passenger
JOIN ticket ON passenger.passport_num = ticket.passport_num
JOIN seat ON ticket.seat_id = seat.seat_id
WHERE seat.occupation = 'Booked'
```

Data Output Сообщения Notifications

	name character varying (15)	surname character varying (20)
1	Kevin	Hart
2	Tobias	Maguire
3	John	Depp

1.3 SELECT-запрос для вывода средней цены для типа места



```
SELECT coach_type.coach_real_type, AVG(ticket.price) AS average_price
FROM ticket
JOIN seat ON ticket.seat_id = seat.seat_id
JOIN coach ON seat.coach_id = coach.coach_id
JOIN coach_type ON coach.coach_type = coach_type.coach_type
GROUP BY coach_type.coach_real_type;
```

Data Output Сообщения Notifications

	coach_real_type character varying (10)	average_price numeric
1	Platskart	2100.0000000000000000
2	Sitting	1662.5000000000000000
3	Coupe	2650.0000000000000000

`

1.4 SELECT-запрос для упорядочивания станций по количеству отбывающих и прибывающих на них пассажиров

Запрос История запросов

```
SELECT name_of_station, passengers
from (select name_of_station, count(*) as passengers
     from (select "arrival_point" as name_of_station from public."ticket"
          union all
          select "departure_point" as name_of_station from public."ticket")
     as station_cnt
     group by name_of_station)
     as passenger order by passengers
```

Data Output Сообщения Notifications

	name_of_station character varying (10)	passengers bigint
1	RAH	3
2	MSC	7
3	SPB	10

1.5 SELECT-запрос для вывода пассажира, потратившего наибольшую сумму на билет

Запрос История запросов

```
SELECT passenger.name, passenger.surname, ticket.price
FROM passenger
JOIN ticket ON passenger.passport_num = ticket.passport_num
WHERE ticket.price = (SELECT MAX(price) FROM ticket);
```

Data Output Сообщения Notifications

	name character varying (15)	surname character varying (20)	price integer
1	Sandra	Bullock	2800

1.6 SELECT-запрос для вывода маршрута, на который купили наибольшее число мест

Запрос История запросов

```
select "departure_point", "arrival_point", count (*) as "Ticket_count"
from public."ticket"
group by "arrival_point", "departure_point"
having count (*) = (select max(ticket_count)
                    from (select "departure_point", "arrival_point", count (*) as ticket_count
                    from public."ticket"
                    group by "arrival_point", "departure_point") as subquery)
```

Data Output Сообщения Notifications

	departure_point character varying (10)	arrival_point character varying (10)	Ticket_count bigint
1	MSC	SPB	4

`

1.7 SELECT-запрос для номера поезда с наибольшей продолжительностью стоянки на станции

Запрос История запросов

```
1  SELECT s1.settlement_id, s1.train_num, (s2.departure - s1.arrival) AS duration
2  FROM stop s1
3  JOIN stop s2 ON s1.train_num = s2.train_num AND s2.departure > s1.arrival
4  ORDER BY (s2.departure - s1.arrival) DESC
```

Data Output Сообщения Notifications

	settlement_id character varying (10)	train_num integer	duration interval
1	TVR	101	00:06:00
2	KOL	301	00:04:00
3	SOL	201	00:03:00

2. Представления

2.1 Запрос на представление расходов на покупку билета по пассажирам

Запрос История запросов

```
1  CREATE VIEW expenses_per_passenger AS
2  SELECT passenger.name, passenger.surname, SUM(ticket.price) AS total_expenses
3  FROM passenger
4  JOIN ticket ON passenger.passport_num = ticket.passport_num
5  GROUP BY passenger.name, passenger.surname;
```

Data Output Сообщения Notifications

```
CREATE VIEW

Запрос завершён успешно, время выполнения: 90 msec.
```

Запрос История запросов

```
1  select * from expenses_per_passenger
```

Data Output Сообщения Notifications

	name character varying (15)	surname character varying (20)	total_expenses bigint
1	Sam	Smith	2000
2	Orlando	Bloom	2500
3	Nicholas	Cannon	1550
4	John	Depp	1600
5	Jacob	Gyllenhaal	2300
6	Kevin	Hart	1800
7	Keira	Knightley	2100
8	Tobias	Maguire	1700
9	Terry	Bollea	2000
10	Sandra	Bullock	2800

`

2.2 Запрос на представление остановки поезда с наибольшей продолжительностью

Запрос История запросов

```
1  CREATE VIEW longest_stop AS
2  SELECT s1.train_num, s1.settlement_id, (s2.departure - s1.arrival) AS duration
3  FROM stop s1
4  JOIN stop s2 ON s1.train_num = s2.train_num AND s2.departure > s1.arrival
5  ORDER BY (s2.departure - s1.arrival) DESC
6
```

Data Output Сообщения Notifications

```
CREATE VIEW

Запрос завершён успешно, время выполнения: 103 msec.
```

Запрос История запросов

```
1  select * from longest_stop
```

Data Output Сообщения Notifications

	train_num integer	settlement_id character varying (10)	duration interval
1	101	TVR	00:06:00
2	301	KOL	00:04:00
3	201	SOL	00:03:00

2.3 Запрос на представление поезда с наибольшим числом забронированных мест

Запрос История запросов

```
1  CREATE VIEW train_w_most_bookings AS
2  SELECT train.train_id, COUNT(seat.seat_id) AS num_bookings
3  FROM train
4  JOIN coach ON train.train_id = coach.train_id
5  JOIN seat ON coach.coach_id = seat.coach_id
6  WHERE seat.occupation = 'Booked'
7  GROUP BY train.train_id
8  ORDER BY COUNT(seat.seat_id) DESC
9
```

Data Output Сообщения Notifications

```
CREATE VIEW

Запрос завершён успешно, время выполнения: 87 msec.
```

Запрос История запросов

```
1  select * from train_w_most_bookings
2
```

Data Output Сообщения Notifications

	train_id character varying (10)	num_bookings bigint
1	HS1	3
2	IC1	1

`

3. Запросы на модификацию данных

3.1 INSERT-вставка нового пассажира, только если пассажир с таким номером паспорта не существует в таблице ‘passenger’

Запрос История запросов

```
1  SELECT * FROM passenger
2
```

Data Output Сообщения Notifications

	passport_num [PK] character varying (15)	surname character varying (20)	name character varying (15)	fathers_name character varying (20)
1	4016 510509	Cannon	Nicholas	Scott
2	4014 404014	Hart	Kevin	Darnell
3	4018 920202	Smith	Sam	Frederick
4	4017 444333	Bloom	Orlando	Jonathan
5	4014 212343	Bullock	Sandra	Annette
6	4020 228822	Depp	John	Christopher
7	4018 789987	Gyllenhaal	Jacob	Benjamin
8	4017 450054	Bollea	Terry	Jene
9	4015 501401	Knightley	Keira	Christina
10	4014 516517	Maguire	Tobias	Vincent

Запрос История запросов

```
1  INSERT INTO passenger (passport_num, surname, name, fathers_name)
2  SELECT '2022 022220', 'Doe', 'John', 'John'
3  WHERE NOT EXISTS (SELECT 1 FROM passenger WHERE passport_num = '2022 022220');
4
```

Data Output Сообщения Notifications

```
INSERT 0 1

Запрос завершён успешно, время выполнения: 91 msec.
```

`

Запрос История запросов

```
1  SELECT * FROM passenger
2
```

Data Output Сообщения Notifications

	passport_num [PK] character varying (15)	surname character varying (20)	name character varying (15)	fathers_name character varying (20)
1	4016 510509	Cannon	Nicholas	Scott
2	4014 404014	Hart	Kevin	Darnell
3	4018 920202	Smith	Sam	Frederick
4	4017 444333	Bloom	Orlando	Jonathan
5	4014 212343	Bullock	Sandra	Annette
6	4020 228822	Depp	John	Christopher
7	4018 789987	Gyllenhaal	Jacob	Benjamin
8	4017 450054	Bollea	Terry	Jene
9	4015 501401	Knightley	Keira	Christina
10	4014 516517	Maguire	Tobias	Vincent
11	2022 022220	Doe	John	John

Total rows: 11 of 11 Query complete 00:00:00.127

3.2 UPDATE. Все места, связанные с билетами, принадлежащими пассажиру с passenger_id равным 1 в таблице ticket, обновляются и помечаются как "Booked" в таблице seat.

5	4014 212343	Bullock	Sandra	Annette

	ticket_num [PK] integer	ticket_office_num integer	seat_id character varying (10)	passport_num character varying (15)
1	309	33	9	4018 920202
2	127	11	27	4016 510509
3	144	11	44	4014 404014
4	332	33	32	4018 789987
5	343	33	43	4017 450054
6	322	33	22	4015 501401
7	118	11	18	4014 516517
8	133	11	33	4020 228822
9	123	11	23	4017 444333
10	103	11	3	4014 212343

`

Запрос История запросов

```
1  SELECT * FROM seat
2
```

Data Output Сообщения Notifications

	seat_id [PK] character varying (10)	coach_id character varying (10)	occupation character varying (10)
1	18	010	Booked
2	27	011	Sold
3	45	020	Free
4	9	030	Sold
5	3	020	Sold

Запрос История запросов

```
1  UPDATE seat
2  SET occupation = 'Booked'
3  WHERE seat_id IN (SELECT seat_id FROM ticket WHERE passport_num = '4014 212343');
4
```

Data Output Сообщения Notifications

```
UPDATE 1

Запрос завершён успешно, время выполнения: 78 msec.
```

Запрос История запросов

```
1  SELECT * FROM seat
2
```

Data Output Сообщения Notifications

	seat_id [PK] character varying (10)	coach_id character varying (10)	occupation character varying (10)
2	27	011	Sold
3	45	020	Free
4	9	030	Sold
5	11	020	Free
6	23	020	Sold
7	32	030	Sold
8	43	030	Sold
9	13	033	Booked
10	33	010	Booked
11	44	010	Booked
12	22	030	Sold
13	2	020	Free
14	1	020	Free
15	3	020	Booked

`

3.3 DELETE- В этом примере все вагоны, принадлежащие поезду с train_id равным HS1, будут удалены из таблицы coach, только если они не имеют связанных мест.

Запрос История запросов

```
1  SELECT * FROM coach
2
```

Data Output Сообщения Notifications

	coach_id [PK] character varying (10)	train_id character varying (10)	coach_num character varying (10)	standard_price integer	coach_type integer
1	010	HS1	9901	1500	6670
2	011	HS1	9911	1500	6660
3	020	INT1	9902	2500	7745
4	030	IC1	9903	2000	9965
5	021	INT1	9912	2500	7740
6	031	IC1	9913	2000	9955
7	022	INT1	9922	2500	7750
8	032	IC1	9923	2000	9960
9	012	HS1	9921	1500	6665
10	033	IC1	9933	2000	9950

Запрос История запросов

```
1  DELETE FROM coach
2  WHERE train_id = 'HS1'
3    AND coach_id NOT IN (SELECT coach_id FROM seat);
4
```

Data Output Сообщения Notifications

```
DELETE 1

Запрос завершён успешно, время выполнения: 72 msec.
```

Запрос История запросов

```
1  SELECT * FROM seat
2
3
```

Data Output Сообщения Notifications

	seat_id [PK] character varying (10)	coach_id character varying (10)	occupation character varying (10)
1	18	010	Booked
2	27	011	Sold
3	45	020	Free

`

Запрос История запросов

```
1  SELECT * FROM coach
2
```

Data Output Сообщения Notifications

	coach_id [PK] character varying (10)	train_id character varying (10)	coach_num character varying (10)	standard_price integer	coach_type integer
1	010	HS1	9901	1500	6670
2	011	HS1	9911	1500	6660
3	020	INT1	9902	2500	7745
4	030	IC1	9903	2000	9965
5	021	INT1	9912	2500	7740
6	031	IC1	9913	2000	9955
7	022	INT1	9922	2500	7750
8	032	IC1	9923	2000	9960
9	033	IC1	9933	2000	9950

4. Создание индексов

4.1 CREATE-INDEX для запроса по выводу средней цены для разных типов мест

Запрос История запросов

```
1  SELECT coach_type.coach_real_type, AVG(ticket.price) AS average_price
2  FROM ticket
3  JOIN seat ON ticket.seat_id = seat.seat_id
4  JOIN coach ON seat.coach_id = coach.coach_id
5  JOIN coach_type ON coach.coach_type = coach_type.coach_type
6  GROUP BY coach_type.coach_real_type;
```

Data Output Сообщения Notifications

	coach_real_type character varying (10)	average_price numeric
1	Coupe	2150.0000000000000000
2	Platskart	1600.0000000000000000
3	Sitting	2700.0000000000000000

Total rows: 3 of 3 Query complete 00:00:00.242

`

Запрос История запросов

```
create index idx_coach on coach(coach_id)
```

Data Output Сообщения Notifications

```
CREATE INDEX

Запрос завершён успешно, время выполнения: 42 msec.
```

Запрос История запросов

```
SELECT coach_type.coach_real_type, AVG(ticket.price) AS average_price
FROM ticket
JOIN seat ON ticket.seat_id = seat.seat_id
JOIN coach ON seat.coach_id = coach.coach_id
JOIN coach_type ON coach.coach_type = coach_type.coach_type
GROUP BY coach_type.coach_real_type;
```

Data Output Сообщения Notifications

	coach_real_type character varying (10)	average_price numeric
1	Coupe	2150.0000000000000000
2	Platskart	1600.0000000000000000
3	Sitting	2700.0000000000000000

Total rows: 3 of 3 Query complete 00:00:00.075

Скорость выполнения команды увеличилась. Query plan остался без изменений, в связи с незначительным сокращением времени выполнения команды.

4.2 CREATE-INDEX для запроса по выводу пассажира, потратившего наибольшую сумму

Запрос История запросов

```
SELECT s1.settlement_id, s1.train_num, (s2.departure - s1.arrival) AS duration
FROM stop s1
JOIN stop s2 ON s1.train_num = s2.train_num AND s2.departure > s1.arrival
ORDER BY (s2.departure - s1.arrival) DESC
```

Data Output Сообщения Notifications

	settlement_id character varying (10)	train_num integer	duration interval
1	TVR	101	00:06:00
2	KOL	301	00:04:00
3	SOL	201	00:03:00

Total rows: 3 of 3 Query complete 00:00:00.321

`

Запрос История запросов

```
1  create index idx_train_num on stop(train_num);
2  create index idx_departure on stop(departure);
3  create index idx_arrival on stop(arrival);
```

Data Output Сообщения Notifications

```
CREATE INDEX

Запрос завершён успешно, время выполнения: 90 msec.
```

Запрос История запросов

```
1  SELECT s1.settlement_id, s1.train_num, (s2.departure - s1.arrival) AS duration
2  FROM stop s1
3  JOIN stop s2 ON s1.train_num = s2.train_num AND s2.departure > s1.arrival
4  ORDER BY (s2.departure - s1.arrival) DESC
```

Data Output Сообщения Notifications

	settlement_id character varying (10)	train_num integer	duration interval
1	TVR	101	00:06:00
2	KOL	301	00:04:00
3	SOL	201	00:03:00

Total rows: 3 of 3 Query complete 00:00:00.166

Индексы помогают сократить время сложного запроса, но на примере простых запросов мы видим, что планировщик считает, что лучше просканировать обычным способом. Индексы при больших запросах позволили значительно выиграть время выполнения, план запроса остался тем же.

`

Вывод:

SQL запросы позволяют изменять, добавлять или удалять данные, а также составлять различные выборки, подсчитывать числовые характеристики. Сравнив время выполнения запросов с индексами и без, можно сделать вывод, что с индексами запросы выполнялись примерно столько же. Это связано с небольшим количеством данных в таблице.

В рамках лабораторной работы были созданы запросы и представления на выборку данных к базе данных PostgreSQL, согласно индивидуальному заданию, часть 2 и 3. Были созданы 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов. Были изучены графические представления запросов. Были созданы простой и составной индексы для двух произвольных запросов.