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:
«Базы данных»

Выполнила:
Редькина Любовь Александровна,
группа **К32422**
Преподаватель:
Говорова Марина Михайловна

Санкт-Петербург
2023

СОДЕРЖАНИЕ:

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.

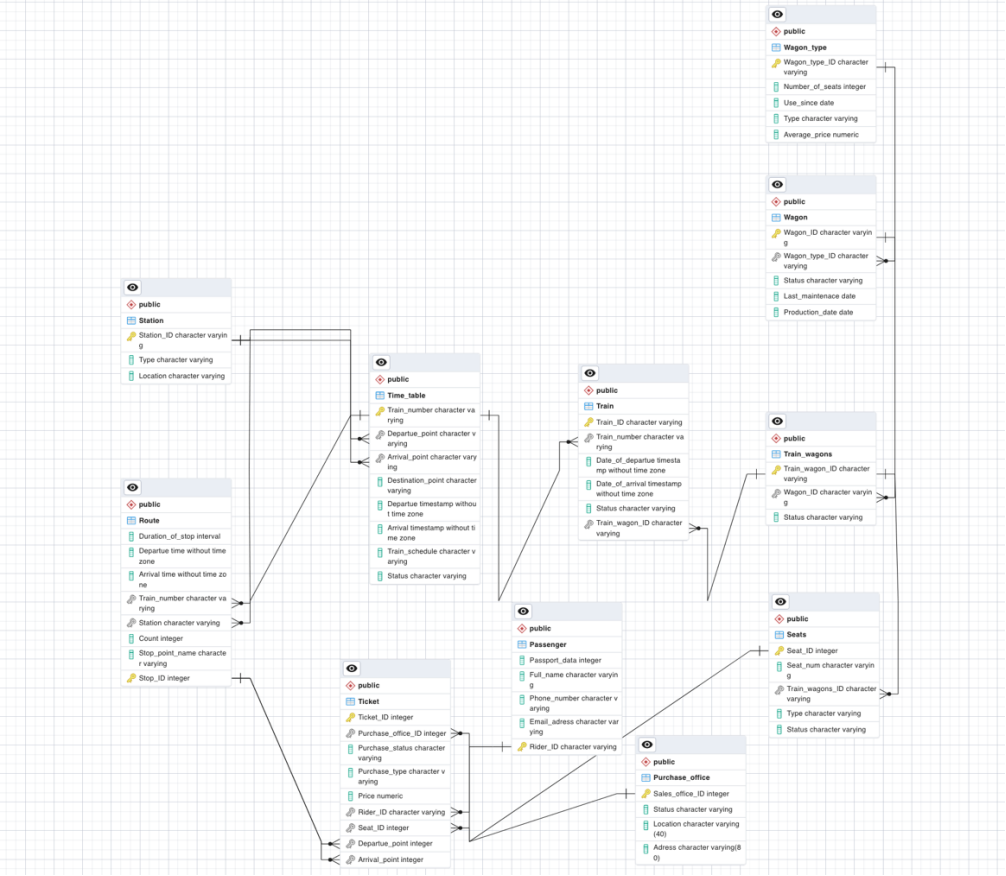
Программное обеспечение: СУБД PostgreSQL, pgadmin 4.

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.

Индивидуальное задание:
Используя данные предыдущей лабораторной работы (Создание базы данных для варианта 6 «Пассажир»), реализовать различные sql-запросы

Рис. 1 Графическое представление реализованной ранее БД



Выполнение:

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

Query | Query History

```
SELECT
    "Type",
    "Number_of_seats"
FROM
    public."Wagon_type"
WHERE
    "Number_of_seats" > (SELECT AVG("Number_of_seats") FROM public."Wagon_type");
```

Data Output | Messages | Notifications

	Type character varying	Number_of_seats integer
1	эконом	54
2	купейный	48
3	эконом	61
4	купейный	54
5	плацкарт	60
6	эконом	56

1.2 SELECT-запрос для вывода пассажиров, не оплативших билет

Query | Query History

```
SELECT
    "Full_name", "Rider_ID"
FROM
    public."Passenger"
WHERE
    "Rider_ID" IN
    (SELECT DISTINCT "Rider_ID" FROM public."Ticket" WHERE "Purchase_status" = 'забронировано')
```

Data Output | Messages | Notifications

	Full_name character varying	Rider_ID [PK] character varying
1	Королева Екатерина	Rider01
2	Иванов Иван Иванович	Rider02
3	Валерьева Валерия	Rider09

1.3 SELECT-запрос для вывода средней цены для типа места

Query | Query History

```
SELECT
    S."Type" AS "Seat Type",
    AVG(T."Price") AS "Average Ticket Price"
FROM
    public."Wagon" AS W
JOIN
    public."Seats" AS S ON W."Wagon_ID" = W."Wagon_ID"
JOIN
    public."Ticket" AS T ON S."Seat_ID" = T."Seat_ID"
GROUP BY
    S."Type"
ORDER BY
    AVG(T."Price") DESC;
```

Data Output | Messages | Notifications

	Seat Type character varying	Average Ticket Price numeric
1	сидячее	4930.1000000000000000
2	спальное	3225.3750000000000000

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

Query | Query History

```
SELECT station_name, total_passengers
FROM (
    SELECT station_name, COUNT(*) AS total_passengers
    FROM (
        SELECT "Arrival_point" AS station_name
        FROM public."Ticket"
        UNION ALL
        SELECT "Departue_point" AS station_name
        FROM public."Ticket"
    ) AS station_counts
    GROUP BY station_name
) AS station_passengers
ORDER BY total_passengers DESC
```

Data Output | Messages | Notifications

	station_name integer	total_passengers bigint
1	7	4
2	8	3
3	11	2
4	13	2
5	12	2
6	10	1

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

Query | Query History

```
SELECT
    r."Full_name",
    SUM(t."Price") AS total_spent
FROM
    public."Passenger" r
JOIN
    public."Ticket" t ON r."Rider_ID" = t."Rider_ID"
GROUP BY
    r."Full_name"
HAVING
    SUM(t."Price") = (SELECT MAX(total_spent) FROM
    (SELECT
        r."Full_name",
        SUM(t."Price") AS total_spent
    FROM public."Passenger" r
    JOIN
        public."Ticket" t ON r."Rider_ID" = t."Rider_ID"
    GROUP BY r."Full_name") AS subquery);
```

Data Output | Messages | Notifications

	Full_name character varying	total_spent numeric
1	Иванов Иван Иванович	17500.50

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

Query | Query History

```
SELECT
    "Departue_point",
    "Arrival_point",
    COUNT(*) AS "Ticket_count"
FROM
    public."Ticket"
GROUP BY
    "Departue_point",
    "Arrival_point"
HAVING COUNT(*) = (SELECT MAX(ticket_count) FROM
    (SELECT
        "Departue_point",
        "Arrival_point",
        COUNT(*) AS ticket_count
     FROM
        public."Ticket"
     GROUP BY
        "Departue_point",
        "Arrival_point") AS subquery);
```

Data Output | Messages | Notifications

	Departue_point integer	Arrival_point integer	Ticket_count bigint
1	7	11	2
2	7	8	2

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

Query | Query History

```
SELECT
    r."Station",
    r."Train_number",
    r."Duration_of_stop"
FROM
    public."Route" r
JOIN
    (
        SELECT
            "Station",
            MAX("Duration_of_stop") AS max_duration
        FROM
            public."Route"
        GROUP BY
            "Station"
    ) subquery
ON
    r."Station" = subquery."Station" AND r."Duration_of_stop" = subquery.max_duration;
```

Data Output | Messages | Notifications

	Station character varying	Train_number character varying	Duration_of_stop interval
1	ТВР-1	000	01:00:00
2	БЛГ-1	000	01:00:00
3	ВЕЛ-1	000	00:30:00
4	ПЕТР-1	000	01:00:00
5	АПАТ-2	000	01:00:00
6	РЯЗ-1	999	01:00:00
7	МСК-2	999	02:00:00
8	СПБ-2	777	05:00:00

2.1 Запрос на представление расходов на покупку билета по пассажирам

Query | Query History

```
1  CREATE VIEW passenger_total_cost AS
2  SELECT p."Rider_ID", p."Full_name", SUM(t."Price") AS total_cost
3  FROM  public."Passenger" p
4  JOIN public."Ticket" t ON p."Rider_ID" = t."Rider_ID"
5  GROUP BY p."Rider_ID", p."Full_name";
```

Data Output | Messages | Notifications

```
CREATE VIEW

Query returned successfully in 41 msec.
```

Query | Query History

```
1  SELECT * FROM passenger_total_cost
```

Data Output | Messages | Notifications

	Rider_ID character varying	Full_name character varying	total_cost numeric
1	Rider02	Иванов Иван Иванович	13500.50
2	Rider08	Макаров Макар	2450.75
3	Rider09	Валерьева Валерия	8000
4	Rider05	Иванов Иван Иванович	4000
5	Rider01	Королева Екатерина	3150

2.2 Запрос на представление остановки поезда с наибольшей продолжительностью

Query | Query History

```
1  CREATE VIEW longest_route_stop_view AS
2  SELECT r."Stop_ID", r."Departue", r."Arrival", r."Train_number", r."Station", r."Duration_of_st
3  FROM "Route" r
4  WHERE r."Duration_of_stop" = (SELECT MAX("Duration_of_stop") FROM "Route");
```

Data Output | Messages | Notifications

```
CREATE VIEW

Query returned successfully in 40 msec.
```

Query | Query History

```
1  SELECT * FROM longest_route_stop_view
```

Data Output | Messages | Notifications

	Stop_ID integer	Departue time without time zone	Arrival time without time zone	Train_number character varying	Station character varying	Duration_of_stop interval	Stop_p charact
1	20	15:00:00	10:00:00	777	СПБ-2	05:00:00	САНКТ

2.3 Запрос на представление поезда с наибольшим числом забронированных мест

Query | Query History

```
CREATE OR REPLACE VIEW train_most_reserved_seats_view AS
SELECT t."Train_ID", t."Train_number", t."Date_of_departue", t."Date_of_arrival", t."Status"
FROM "Train" t
JOIN "Train_wagons" tw ON t."Train_wagon_ID" = tw."Train_wagon_ID"
JOIN "Seats" s ON tw."Train_wagon_ID" = s."Train_wagons_ID"
JOIN "Ticket" tk ON s."Seat_ID" = tk."Seat_ID"
WHERE tk."Purchase_status" = 'забронировано'
GROUP BY t."Train_ID", t."Train_number", t."Date_of_departue", t."Date_of_arrival", t."Status"
HAVING COUNT(tk."Ticket_ID") = (SELECT MAX(seat_count) FROM (SELECT COUNT(tk."Ticket_ID") AS se
JOIN "Train_wagons" tw ON t."Train_wagon_ID" = tw."Train_wagon_ID"
JOIN "Seats" s ON tw."Train_wagon_ID" = s."Train_wagons_ID"
JOIN "Ticket" tk ON s."Seat_ID" = tk."Seat_ID"
WHERE tk."Purchase_status" = 'забронировано'
GROUP BY t."Train_ID") AS seat_counts);
```

Data Output | Messages | Notifications

```
CREATE VIEW

Query returned successfully in 39 msec.
```

Query | Query History

```
SELECT * FROM train_most_reserved_seats_view
```

Data Output | Messages | Notifications

	Train_ID character varying	Train_number character varying	Date_of_departue timestamp without time zone	Date_of_arrival timestamp without time zone	Status character varying
1	0	000	2023-06-28 19:00:00	2023-06-30 07:00:00	посадка
2	2	222	2023-05-17 13:00:00	2023-05-18 10:00:00	задерживается
3	6	666	2024-01-02 15:30:00	2024-01-07 15:30:00	задерживается
4	9	999	2022-12-01 08:00:00	2022-12-02 08:20:00	прибывает

3.1 INSERT-запрос для добавления данных в таблицу Маршрута

Query | Query History

```
INSERT INTO public."Route" ("Duration_of_stop", "Departue", "Arrival", "Train_number", "Station
VALUES (
  '05:00:00',
  '15:00:00',
  '10:00:00',
  '777',
  'СПБ-2',
  1,
  'САНКТ-ПЕТЕРБУРГ');
```

Data Output | Messages | Notifications

```
INSERT 0 1

Query returned successfully in 43 msec.
```

Запрос до выполнения вставки:

Query | Query History

```
SELECT * FROM public."Route"
```

Data Output | Messages | Notifications

	Duration_of_stop interval	Departue time without time zone	Arrival time without time zone	Train_number character varying	Station character varying	Count integer	Stop_p charact
1	01:00:00	06:00:00	05:00:00	000	ТВР-1	1	ТВЕРЬ
2	01:00:00	08:00:00	07:00:00	000	БЛГ-1	2	БОЛОГ
3	00:30:00	15:30:00	15:00:00	000	ВЕЛ-1	3	ВЕЛИК
4	01:00:00	20:30:00	19:30:00	000	СПБ-2	4	САНКТ
5	01:00:00	17:00:00	16:00:00	000	ПЕТР-1	5	ПЕТРО
6	01:00:00	21:00:00	20:00:00	000	АПАТ-2	6	АПАТИ
7	01:00:00	06:00:00	05:00:00	999	РЯЗ-1	2	РЯЗАН
8	02:00:00	01:00:00	23:00:00	999	МСК-2	1	МОСКВ

Запрос после выполнения вставки:

Query | Query History

```
SELECT * FROM public."Route"
```

Data Output | Messages | Notifications

	Duration_of_stop interval	Departue time without time zone	Arrival time without time zone	Train_number character varying	Station character varying	Count integer	Stop_p charact
1	01:00:00	06:00:00	05:00:00	000	ТВР-1	1	ТВЕРЬ
2	01:00:00	08:00:00	07:00:00	000	БЛГ-1	2	БОЛОГ
3	00:30:00	15:30:00	15:00:00	000	ВЕЛ-1	3	ВЕЛИК
4	01:00:00	20:30:00	19:30:00	000	СПБ-2	4	САНКТ
5	01:00:00	17:00:00	16:00:00	000	ПЕТР-1	5	ПЕТРО
6	01:00:00	21:00:00	20:00:00	000	АПАТ-2	6	АПАТИ
7	01:00:00	06:00:00	05:00:00	999	РЯЗ-1	2	РЯЗАН
8	02:00:00	01:00:00	23:00:00	999	МСК-2	1	МОСКВ
9	05:00:00	15:00:00	10:00:00	777	СПБ-2	1	САНКТ

3.2 DELETE-запрос удаления билетов на поезда, которые не были укомплектованы

Query | Query History

```
1  DELETE FROM public."Ticket"
2  WHERE "Seat_ID" IN (
3    SELECT "Seat_ID"
4    FROM "Seats"
5    WHERE "Train_wagons_ID" IN (
6      SELECT "Train_wagon_ID"
7      FROM "Train_wagons"
8      WHERE "Status" = 'не укомплектован'
9    )
10 );
```

Data Output | Messages | Notifications

```
DELETE 3

Query returned successfully in 53 msec.
```

Запрос до удаления данных

Query | Query History

```
1  SELECT * FROM public."Ticket"
2  ORDER BY "Ticket_ID" ASC
```

Data Output | Messages | Notifications

	Ticket_ID [PK] integer	Purchase_office_ID integer	Purchase_status character varying	Purchase_type character varying	Price numeric	Rider_ID character varying	Seat_ID integer	Depart integer
1	1	1	выкуплено	онлайн	3000.50	Rider02	1	
2	2	2	забронировано	онлайн	1500	Rider02	2	
3	3	7	выкуплено	касса	4000	Rider05	3	
4	4	4	забронировано	касса	999	Rider07	7	
5	5	3	выкуплено	онлайн	890	Rider04	5	
6	6	2	забронировано	онлайн	3150	Rider01	6	
7	7	8	забронировано	онлайн	1250	Rider03	4	
8	8	6	выкуплено	онлайн	2450.75	Rider08	8	
9	9	2	забронировано	касса	8000	Rider09	9	
10	10	9	забронировано	онлайн	9000.0	Rider02	10	

Запрос после удаления

Query | Query History

```
1  SELECT * FROM public."Ticket"
2  ORDER BY "Ticket_ID" ASC
```

Data Output | Messages | Notifications

	Ticket_ID [PK] integer	Purchase_office_ID integer	Purchase_status character varying	Purchase_type character varying	Price numeric	Rider_ID character varying	Seat_ID integer	Depart integer
1	1	1	выкуплено	онлайн	3000.50	Rider02	1	
2	2	2	забронировано	онлайн	1500	Rider02	2	
3	3	7	выкуплено	касса	4000	Rider05	3	
4	6	2	забронировано	онлайн	3150	Rider01	6	
5	8	6	выкуплено	онлайн	2450.75	Rider08	8	
6	9	2	забронировано	касса	8000	Rider09	9	
7	10	9	забронировано	онлайн	9000.0	Rider02	10	

3.3 UPDATE-запрос на добавление 10% скидки тем, кто еще не оплатил поездку в вагоне типа «семейный»

Query | Query History

```
UPDATE "Ticket"
SET "Price" = "Price" * 0.9
WHERE "Purchase_status" = 'забронировано'
AND "Ticket_ID" IN (
  SELECT "Ticket"."Ticket_ID"
  FROM "Ticket"
  JOIN "Seats" ON "Ticket"."Seat_ID" = "Seats"."Seat_ID"
  JOIN "Train_wagons" ON "Seats"."Train_wagons_ID" = "Train_wagons"."Train_wagon_ID"
  JOIN "Wagon" ON "Train_wagons"."Wagon_ID" = "Wagon"."Wagon_ID"
  JOIN "Wagon_type" ON "Wagon"."Wagon_type_ID" = "Wagon_type"."Wagon_type_ID"
  WHERE "Wagon_type"."Type" = 'семейный'
);
```

Data Output | Messages | Notifications

```
UPDATE 1

Query returned successfully in 64 msec.
```

Запрос до обновления данных

Query | Query History

```
SELECT * FROM public."Ticket"
ORDER BY "Ticket_ID" ASC
```

Data Output | Messages | Notifications

	Ticket_ID [PK] integer	Purchase_office_ID integer	Purchase_status character varying	Purchase_type character varying	Price numeric	Rider_ID character varying	Seat_ID integer	Departu integer
1	1	1	выкуплено	онлайн	3000.50	Rider02	1	
2	2	2	забронировано	онлайн	1500	Rider02	2	
3	3	7	выкуплено	касса	4000	Rider05	3	
4	4	8	выкуплено	онлайн	890	Rider04	4	
5	5	3	забронировано	касса	1250	Rider05	5	
6	6	2	забронировано	онлайн	3150	Rider01	6	
7	7	10	выкуплено	онлайн	300	Rider07	7	
8	8	6	выкуплено	онлайн	2450.75	Rider08	8	
9	9	2	забронировано	касса	8000	Rider09	9	
10	10	9	забронировано	онлайн	9000.0	Rider02	10	

Запрос после обновления данных

Query | Query History

```
SELECT * FROM "Ticket"
```

Data Output | Messages | Notifications

	Ticket_ID [PK] integer	Purchase_office_ID integer	Purchase_status character varying	Purchase_type character varying	Price numeric	Rider_ID character varying	Seat_ID integer	Departu integer
1	6	2	забронировано	онлайн	3150	Rider01	6	
2	1	1	выкуплено	онлайн	3000.50	Rider02	1	
3	3	7	выкуплено	касса	4000	Rider05	3	
4	9	2	забронировано	касса	8000	Rider09	9	
5	8	6	выкуплено	онлайн	2450.75	Rider08	8	
6	2	2	забронировано	онлайн	1500	Rider02	2	
7	4	8	выкуплено	онлайн	890	Rider04	4	
8	5	3	забронировано	касса	1250	Rider05	5	
9	7	10	выкуплено	онлайн	300	Rider07	7	
10	10	9	забронировано	онлайн	8100.00	Rider02	10	

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

Query Query History

```
1  CREATE INDEX idx_wagon_id ON public."Wagon" ("Wagon_ID");
2  CREATE INDEX idx_seat_id ON public."Seats" ("Seat_ID");
3  CREATE INDEX idx_price ON public."Ticket" ("Price");
4
```

Data Output Messages Notifications

```
CREATE INDEX

Query returned successfully in 138 msec.
```

Время выполнения запроса до индексирования

Query Query History

```
1  SELECT
2      S."Type" AS "Seat Type",
3      AVG(T."Price") AS "Average Ticket Price"
4  FROM
5      public."Wagon" AS W
6  JOIN
7      public."Seats" AS S ON W."Wagon_ID" = W."Wagon_ID"
8  JOIN
9      public."Ticket" AS T ON S."Seat_ID" = T."Seat_ID"
10 GROUP BY
11     S."Type"
12 ORDER BY
13     AVG(T."Price") DESC;
```

Data Output Messages Notifications

	Seat Type character varying	Average Ticket Price numeric
1	сидячее	4008.4166666666666667
2	спальное	2147.6875000000000000

Query returne

Total rows: 2 of 2 Query complete 00:00:00.109

Время выполнения запроса после индексирования

Query | Query History

```
SELECT
    S."Type" AS "Seat Type",
    AVG(T."Price") AS "Average Ticket Price"
FROM
    public."Wagon" AS W
JOIN
    public."Seats" AS S ON W."Wagon_ID" = W."Wagon_ID"
JOIN
    public."Ticket" AS T ON S."Seat_ID" = T."Seat_ID"
GROUP BY
    S."Type"
ORDER BY
    AVG(T."Price") DESC;
```

Data Output | Messages | Notifications

	Seat Type character varying	Average Ticket Price numeric
1	сидячее	4008.4166666666666667
2	спальное	2147.6875000000000000

Query returne

Total rows: 2 of 2 | Query complete 00:00:00.050

Query plan остался без изменений, так как индекс не оказал значительного влияния на производительность запроса

4.2 CREATE-INDEX для запроса по выводу суммы потраченных на покупку билетов денег пассажирами

Query | Query History

```
CREATE INDEX IF NOT EXISTS multiple_idx
    ON public."Passenger" USING btree
    ("Full_name" COLLATE pg_catalog."default" ASC NULLS LAST,
     "Rider_ID" COLLATE pg_catalog."default" ASC NULLS LAST)
    TABLESPACE pg_default;

CREATE INDEX IF NOT EXISTS multiple_index
    ON public."Ticket" USING btree
    ("Rider_ID" COLLATE pg_catalog."default" ASC NULLS LAST)
    TABLESPACE pg_default;
```

Время выполнения до индексирования:

Query | Query History

```
SELECT p."Rider_ID", p."Full_name", SUM(t."Price") AS total_cost
FROM  public."Passenger" p
JOIN public."Ticket" t ON p."Rider_ID" = t."Rider_ID"
GROUP BY p."Rider_ID", p."Full_name";
```

Data Output | Messages | Notifications

	Rider_ID [PK] character varying	Full_name character varying	total_cost numeric
1	Rider02	Иванов Иван Иванович	12600.50
2	Rider08	Макаров Макар	2450.75
3	Rider04	Сидоров Сидор Сидорович	890
4	Rider05	Иванов Иван Иванович	5250
5	Rider09	Валерьева Валерия	8000
6	Rider07	Ульянова Ульяна	300
7	Rider01	Королева Екатерина	3150

Total rows: 7 of 7 | Query complete 00:00:00.126

Время выполнения после индексирования:

Query | Query History

```
1  SELECT p."Rider_ID", p."Full_name", SUM(t."Price") AS total_cost
2  FROM  public."Passenger" p
3  JOIN public."Ticket" t ON p."Rider_ID" = t."Rider_ID"
4  GROUP BY p."Rider_ID", p."Full_name";
```

Data Output | Messages | Notifications

	Rider_ID [PK] character varying	Full_name character varying	total_cost numeric
1	Rider02	Иванов Иван Иванович	12600.50
2	Rider08	Макаров Макар	2450.75
3	Rider04	Сидоров Сидор Сидорович	890
4	Rider05	Иванов Иван Иванович	5250
5	Rider09	Валерьева Валерия	8000
6	Rider07	Ульянова Ульяна	300
7	Rider01	Королева Екатерина	3150

Total rows: 7 of 7 | Query complete 00:00:00.049

ВЫВОДЫ:

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.

Индексы при больших запросах позволили значительно выиграть время выполнения, план запроса остался тем же.