

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**

## **СОДЕРЖАНИЕ:**

<b>ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ:</b> .....	<b>3</b>
<b>ВЫПОЛНЕНИЕ:</b> .....	<b>4</b>
<b>ВЫВОДЫ:</b> .....	<b>17</b>

**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.

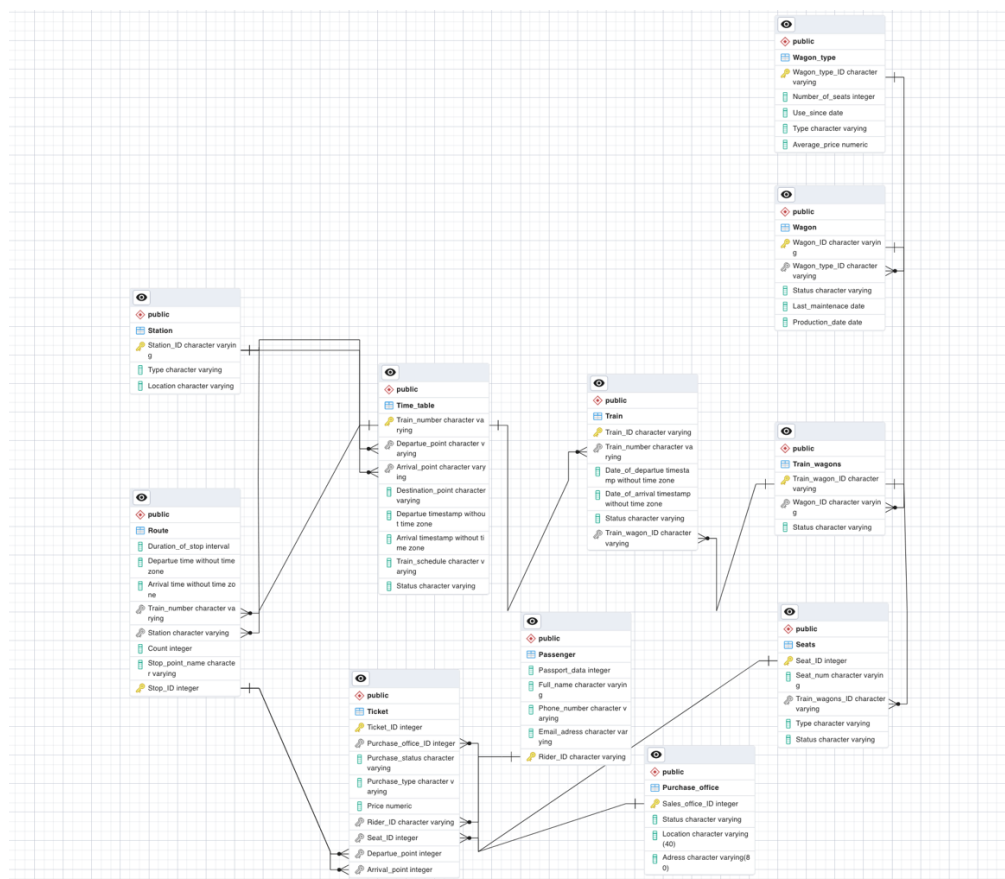
**Практическое задание:**

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

```
1 SELECT
2     "Type",
3     "Number_of_seats"
4 FROM
5     public."Wagon_type"
6 WHERE
7     "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

```
1 SELECT
2     "Full_name", "Rider_ID"
3 FROM
4     public."Passenger"
5 WHERE
6     "Rider_ID" IN
7     (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

QueryQuery History

1SELECT

2S."Type" AS "Seat Type",

3AVG(T."Price") AS "Average Ticket Price"

4FROM

5public."Wagon" AS W

6JOIN

7public."Seats" AS S ON W."Wagon\_ID" = W."Wagon\_ID"

8JOIN

9public."Ticket" AS T ON S."Seat\_ID" = T."Seat\_ID"

10GROUP BY

11S."Type"

12ORDER BY

13AVG(T."Price") DESC;

Data OutputMessagesNotifications

	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

QueryQuery History

1SELECT station\_name, total\_passengers

2FROM (

3SELECT station\_name, COUNT(\*) AS total\_passengers

4FROM (

5SELECT "Arrival\_point" AS station\_name

6FROM public."Ticket"

7UNION ALL

8SELECT "Departue\_point" AS station\_name

9FROM public."Ticket"

10) AS station\_counts

11GROUP BY station\_name

12) AS station\_passengers

13ORDER BY total\_passengers DESC

Data OutputMessagesNotifications

	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

QueryQuery History

1SELECT

2r."Full\_name",

3SUM(t."Price") AS total\_spent

4FROM

5public."Passenger" r

6JOIN

7public."Ticket" t ON r."Rider\_ID" = t."Rider\_ID"

8GROUP BY

9r."Full\_name"

10HAVING

11SUM(t."Price") = (SELECT MAX(total\_spent) FROM

12(SELECT

13r."Full\_name",

14SUM(t."Price") AS total\_spent

15FROM public."Passenger" r

16JOIN

17public."Ticket" t ON r."Rider\_ID" = t."Rider\_ID"

18GROUP BY r."Full\_name") AS subquery);

Data OutputMessagesNotifications

	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

QueryQuery History

1SELECT

2"Departue\_point",

3"Arrival\_point",

4COUNT(\*) AS "Ticket\_count"

5FROM

6public."Ticket"

7GROUP BY

8"Departue\_point",

9"Arrival\_point"

10HAVING COUNT(\*) = (SELECT MAX(ticket\_count) FROM

11(SELECT

12"Departue\_point",

13"Arrival\_point",

14COUNT(\*) AS ticket\_count

15FROM

16public."Ticket"

17GROUP BY

18"Departue\_point",

19"Arrival\_point") AS subquery);

Data OutputMessagesNotifications

	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

QueryQuery History

1SELECT

2r."Station",

3r."Train\_number",

4r."Duration\_of\_stop"

5FROM

6public."Route" r

7JOIN

8(

9SELECT

10"Station",

11MAX("Duration\_of\_stop") AS max\_duration

12FROM

13public."Route"

14GROUP BY

15"Station"

16) subquery

17ON

18r."Station" = subquery."Station" AND r."Duration\_of\_stop" = subquery.max\_duration;

Data OutputMessagesNotifications

	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

QueryQuery 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 OutputMessagesNotifications

CREATE VIEW

Query returned successfully in 41 msec.

QueryQuery History

1

SELECT \* FROM passenger\_total\_cost

Data OutputMessagesNotifications

	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

QueryQuery 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\_stop"

3

FROM "Route" r

4

WHERE r."Duration\_of\_stop" = (SELECT MAX("Duration\_of\_stop") FROM "Route");

Data OutputMessagesNotifications

CREATE VIEW

Query returned successfully in 40 msec.

QueryQuery History

1

SELECT \* FROM longest\_route\_stop\_view

Data OutputMessagesNotifications

	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	САЖК



2.3 Запрос на представление поезда с наибольшим числом забронированных мест

QueryQuery History

1CREATE OR REPLACE VIEW train\_most\_reserved\_seats\_view AS

2SELECT t."Train\_ID", t."Train\_number", t."Date\_of\_departue", t."Date\_of\_arrival", t."Status"

3FROM "Train" t

4JOIN "Train\_wagons" tw ON t."Train\_wagon\_ID" = tw."Train\_wagon\_ID"

5JOIN "Seats" s ON tw."Train\_wagon\_ID" = s."Train\_wagons\_ID"

6JOIN "Ticket" tk ON s."Seat\_ID" = tk."Seat\_ID"

7WHERE tk."Purchase\_status" = 'забронировано'

8GROUP BY t."Train\_ID", t."Train\_number", t."Date\_of\_departue", t."Date\_of\_arrival", t."Status"

9HAVING COUNT(tk."Ticket\_ID") = (SELECT MAX(seat\_count) FROM (SELECT COUNT(tk."Ticket\_ID") AS seat\_count

10JOIN "Train\_wagons" tw ON t."Train\_wagon\_ID" = tw."Train\_wagon\_ID"

11JOIN "Seats" s ON tw."Train\_wagon\_ID" = s."Train\_wagons\_ID"

12JOIN "Ticket" tk ON s."Seat\_ID" = tk."Seat\_ID"

13WHERE tk."Purchase\_status" = 'забронировано'

14GROUP BY t."Train\_ID") AS seat\_counts);

Data OutputMessagesNotifications

CREATE VIEW

Query returned successfully in 39 msec.

QueryQuery History

1SELECT \* FROM train\_most\_reserved\_seats\_view

Data OutputMessagesNotifications

	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

9

3.1 INSERT-запрос для добавления данных в таблицу Маршрута

QueryQuery History

1

INSERT INTO public."Route" ("Duration\_of\_stop", "Departue", "Arrival", "Train\_number", "Station

2

VALUES (

3

'05:00:00',

4

'15:00:00',

5

'10:00:00',

6

'777',

7

'СПБ-2',

8

1,

9

'САНКТ-ПЕТЕРБУРГ');

Data OutputMessagesNotifications

INSERT 0 1

Query returned successfully in 43 msec.

Запрос до выполнения вставки:

QueryQuery History

1

SELECT \* FROM public."Route"

Data OutputMessagesNotifications

	Duration_of_stop interval	Departue time without time zone	Arrival time without time zone	Train_number character varying	Station character varying	Count integer	Stop_pr 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	МОСКИ

Запрос после выполнения вставки:

QueryQuery History

1

SELECT \* FROM public."Route"

Data OutputMessagesNotifications

	Duration_of_stop interval	Departue time without time zone	Arrival time without time zone	Train_number character varying	Station character varying	Count integer	Stop_pr 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	МОСКИ
9	05:00:00	15:00:00	10:00:00	777	СПБ-2	1	САНКТ

10

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

QueryQuery 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 OutputMessagesNotifications

DELETE 3

Query returned successfully in 53 msec.

Запрос до удаления данных

QueryQuery History

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

Data OutputMessagesNotifications

	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

QueryQuery History

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

Data OutputMessagesNotifications

	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

1

UPDATE "Ticket"

2

SET "Price" = "Price" \* 0.9

3

WHERE "Purchase\_status" = 'забронировано'

4

AND "Ticket\_ID" IN (

5

SELECT "Ticket"."Ticket\_ID"

6

FROM "Ticket"

7

JOIN "Seats" ON "Ticket"."Seat\_ID" = "Seats"."Seat\_ID"

8

JOIN "Train\_wagons" ON "Seats"."Train\_wagons\_ID" = "Train\_wagons"."Train\_wagon\_ID"

9

JOIN "Wagon" ON "Train\_wagons"."Wagon\_ID" = "Wagon"."Wagon\_ID"

10

JOIN "Wagon\_type" ON "Wagon"."Wagon\_type\_ID" = "Wagon\_type"."Wagon\_type\_ID"

11

WHERE "Wagon\_type"."Type" = 'семейный'

12

);

Data Output

Messages

Notifications

UPDATE 1

Query returned successfully in 64 msec.

### Запрос до обновления 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	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

1

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	Depart 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

QueryQuery History

1CREATE INDEX idx\_wagon\_id ON public."Wagon" ("Wagon\_ID");

2CREATE INDEX idx\_seat\_id ON public."Seats" ("Seat\_ID");

3CREATE INDEX idx\_price ON public."Ticket" ("Price");

4

Data OutputMessagesNotifications

CREATE INDEX

Query returned successfully in 138 msec.

Время выполнения запроса до индексирования

QueryQuery History

1SELECT

2S."Type" AS "Seat Type",

3AVG(T."Price") AS "Average Ticket Price"

4FROM

5public."Wagon" AS W

6JOIN

7public."Seats" AS S ON W."Wagon\_ID" = W."Wagon\_ID"

8JOIN

9public."Ticket" AS T ON S."Seat\_ID" = T."Seat\_ID"

10GROUP BY

11S."Type"

12ORDER BY

13AVG(T."Price") DESC;

Data OutputMessagesNotifications

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

✓ Query returned successfully

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

13

## Время выполнения запроса после индексирования

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

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

Total rows: 2 of 2

Query complete 00:00:00.050

✓ Query returned

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

QueryQuery History

```
1 CREATE INDEX IF NOT EXISTS multiple_idx
2   ON public."Passenger" USING btree
3   ("Full_name" COLLATE pg_catalog."default" ASC NULLS LAST,
4    "Rider_ID" COLLATE pg_catalog."default" ASC NULLS LAST)
5   TABLESPACE pg_default;
6
7 CREATE INDEX IF NOT EXISTS multiple_index
8   ON public."Ticket" USING btree
9   ("Rider_ID" COLLATE pg_catalog."default" ASC NULLS LAST)
10  TABLESPACE pg_default;
```

Время выполнения до индексирования:

QueryQuery 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 OutputMessagesNotifications

≡+📄▼📋▼🗑️🗄️⬇️📈

	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 7Query 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.