

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИТМО»**

**Отчет**

по лабораторной работе «Запросы на выборку и модификацию данных,  
представления и индексы в PostgreSQL»  
по дисциплине «**Базы данных**»

Автор: Пронина Мария Владимировна

Факультет: ИКТ

Группа: K32392

Преподаватель: Говорова М. М.

Дата: 04.05.2023



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.

**Практическое задание:**

1. Запросы к базе данных. Выполнить запросы согласно индивидуальному заданию, часть 2.

Создать запросы:

- Свободные места на все поезда, отправляющиеся с вокзала в течение следующих суток.
- Список пассажиров, отправившихся в Москву всеми рейсами за прошедшие сутки.
- Номера поездов, на которые проданы все билеты на следующие сутки.
- Свободные места в купейные вагоны всех рейсов до Москвы на текущие сутки.
- Выручка от продажи билетов на все поезда за прошедшие сутки.
- Общее количество билетов, проданных по всем направлениям в вагоны типа “СВ”.
- Номера и названия поездов, все вагоны которых были заполнены менее чем наполовину за прошедшие сутки.

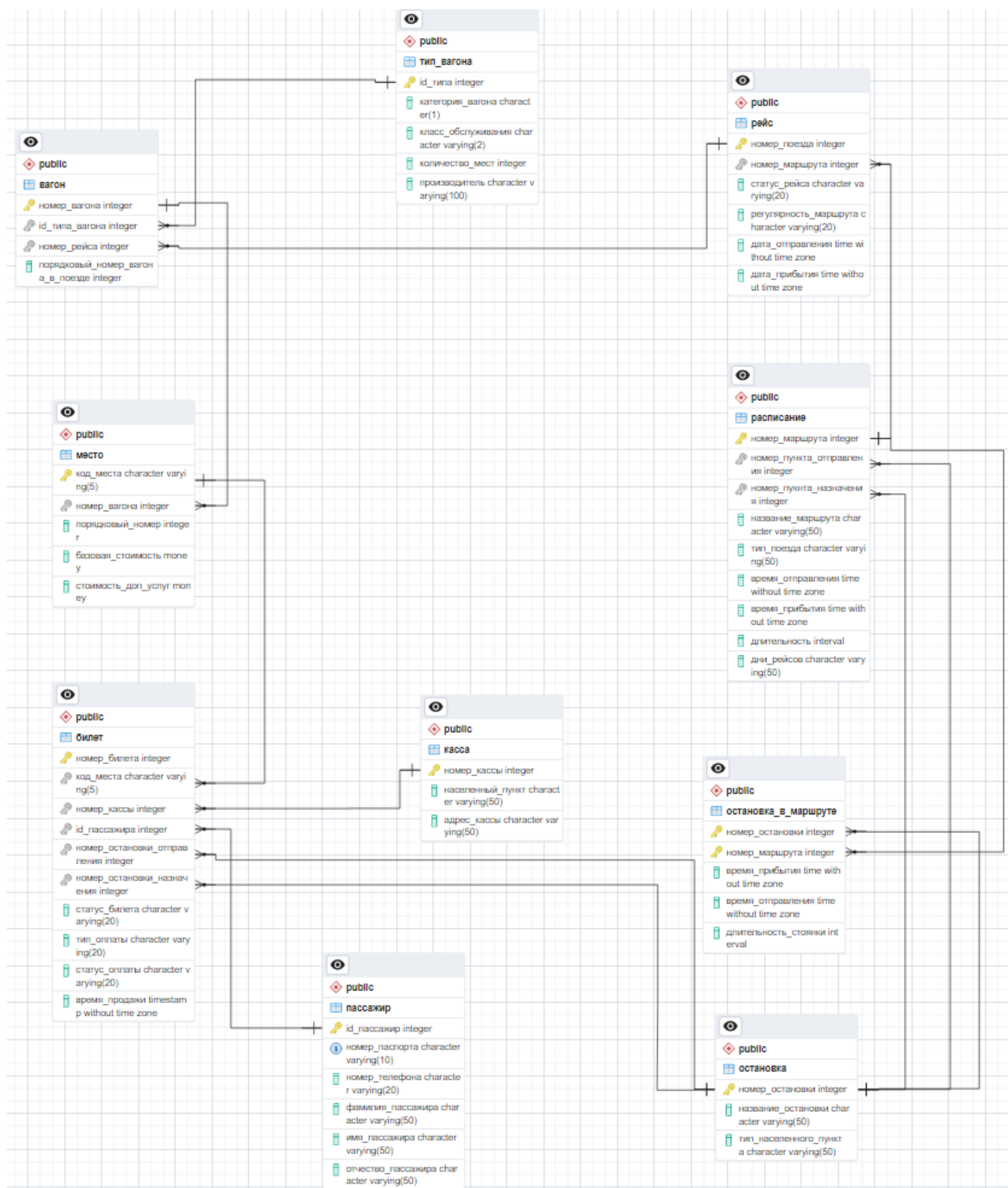
2. Представления. Выполнить запросы на создание представлений согласно индивидуальному заданию, часть 3.

Создать представление:

- для пассажиров о наличии свободных мест на заданный рейс;
- количество непроданных билетов на все поезда, формирующиеся за прошедшие сутки (номер поезда, тип вагона, количество).

3. Запросы на модификацию данных. Составить и выполнить запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
4. Создание индексов. Выполнить запросы без индекса и создать планы запросов. Выполнить создание индексов. Выполнить запросы с индексами и создать планы запросов. Сравнить время выполнения запросов. Удалить индексы.

### Схема базы данных:



- Запросы к базе данных

1. Свободные места на все поезда, отправляющиеся с вокзала в течение следующих суток.

```
select место.* from место, вагон, рейс where
((рейс.дата_отправления >= cast(now() as date) + interval '1 day')
and (рейс.дата_отправления < cast(now() as date) + interval '2
day'))
and вагон.номер_рейса = рейс.номер_поезда and вагон.номер_вагона =
место.номер_вагона
and место.код_места not in (select код_места from билет where
статус_билета IN ('оплачен', 'забронирован')))
```

```
1 select место.* from место, вагон, рейс where
2 ((рейс.дата_отправления >= cast(now() as date) + interval '1 day')
3 and (рейс.дата_отправления < cast(now() as date) + interval '2 day')
4 and вагон.номер_рейса = рейс.номер_поезда and вагон.номер_вагона = место.номер_вагона
5 and место.код_места not in (select код_места from билет where статус_билета IN ('оплачен', 'забронирован'))
6 |
```

Data Output   Сообщения   Notifications

	код_места [PK] character varying (5)	номер_вагона integer	порядковый_номер integer	базовая_стоимость money	стоимость_доп_услуг money
1	20	13	2	\$30.00	\$5.00

2. Список пассажиров, отправившихся в Москву всеми рейсами за прошедшие сутки.

```
select пассажир.* from пассажир, билет, место, вагон, рейс where
билет.id_пассажира = пассажир.id_пассажир
and билет.номер_остановки_назначения = (select номер_остановки from
остановка where название_остановки='Москва')
and билет.код_места = место.код_места and вагон.номер_вагона =
место.номер_вагона and рейс.номер_поезда = вагон.номер_рейса
and рейс.дата_отправления >= cast(now() as date) - interval '1 day' and
рейс.дата_отправления < cast(now() as date);
```

```

12 select пассажир.* from пассажир, билет, место, вагон, рейс where билет.id_пассажира = пассажир.id_пассажира
13 and билет.номер_остановки_назначения = (select номер_остановки from остановка where название_остановки='Москва')
14 and билет.код_места = место.код_места and вагон.номер_вагона = место.номер_вагона and рейс.номер_поезда = вагон.номер_рейса
15 and рейс.дата_отправления >= cast(now() as date) - interval '1 day' and рейс.дата_отправления < cast(now() as date);
16
17

```

Data Output   Сообщения   Notifications



	id_пассажир [PK] integer	номер_паспорта character varying (10)	номер_телефона character varying (20)	фамилия_пассажира character varying (50)	имя_пассажира character varying (50)	отчество_пассажира character varying (50)
1	2	1111222222	+79811111111	Иванов	Иван	Иванович

3. Номера поездов, на которые проданы все билеты на следующие сутки.

```

create table places as
(select рейс.номер_поезда, count(место.*) from место, вагон, рейс
where место.номер_вагона = вагон.номер_вагона and рейс.номер_поезда
= вагон.номер_рейса
and (рейс.дата_отправления >= cast(now() as date) + interval '1
day')
and (рейс.дата_отправления < cast(now() as date) + interval '2
day')
group by рейс.номер_поезда);

create table tickets as
(select рейс.номер_поезда, count(билет.*) from билет, место, вагон,
рейс where билет.код_места = место.код_места and место.номер_вагона
= вагон.номер_вагона
and рейс.номер_поезда = вагон.номер_рейса
and (рейс.дата_отправления >= cast(now() as date) + interval '1
day')
and (рейс.дата_отправления < cast(now() as date) + interval '2
day')
group by рейс.номер_поезда);

select places.номер_поезда from places, tickets where
places.номер_поезда = tickets.номер_поезда and places.count =
tickets.count;

drop table places;
drop table tickets;

```

```

8 create table places as
9 (select рейс.номер_поезда, count(место.*) from место, вагон, рейс where место.номер_вагона = вагон.номер_вагона and рейс.номер_поезда = вагон.номер_рейса
10 and (рейс.дата_отправления >= cast(now() as date) + interval '1 day')
11 and (рейс.дата_отправления < cast(now() as date) + interval '2 day')
12 group by рейс.номер_поезда);
13
14 create table tickets as
15 (select рейс.номер_поезда, count(билет.*) from билет, место, вагон, рейс where билет.код_места = место.код_места and место.номер_вагона = вагон.номер_вагона
16 and рейс.номер_поезда = вагон.номер_рейса
17 and (рейс.дата_отправления >= cast(now() as date) + interval '1 day')
18 and (рейс.дата_отправления < cast(now() as date) + interval '2 day')
19 group by рейс.номер_поезда);
20
21 select places.номер_поезда from places, tickets where places.номер_поезда = tickets.номер_поезда and places.count = tickets.count;
22
23 drop table places;
24 drop table tickets;

```

Data Output   Сообщения   Notifications



	номер_поезда integer
1	6

4. Свободные места в купейные вагоны всех рейсов до Москвы на текущие сутки.

```
select место.* from место, вагон, тип_вагона, рейс, расписание,
остановка where место.код_места not in ('оплачен', 'забронирован')
and вагон.номер_вагона = место.номер_вагона and
вагон.id_типа_вагона=тип_вагона.id_типа
and substring(тип_вагона.класс_обслуживания, 1, 1) = '1'
and вагон.номер_рейса = рейс.номер_поезда and рейс.номер_маршрута
= расписание.номер_маршрута and расписание.номер_пункта_назначения
= остановка.номер_остановки
and остановка.название_остановки = 'Москва'
and cast(рейс.дата_отправления as date) = cast(now() as date)
```

```
1 select место.* from место, вагон, тип_вагона, рейс, расписание, остановка where место.код_места not in ('оплачен', 'забронирован')
2 and вагон.номер_вагона = место.номер_вагона and вагон.id_типа_вагона=тип_вагона.id_типа
3 and substring(тип_вагона.класс_обслуживания, 1, 1) = '1'
4 and вагон.номер_рейса = рейс.номер_поезда and рейс.номер_маршрута = расписание.номер_маршрута and расписание.номер_пункта_назначения = остановка.номер_остановки
5 and остановка.название_остановки = 'Москва'
6 and cast(рейс.дата_отправления as date) = cast(now() as date)
7
```

Data Output   Сообщения   Notifications

	код_места [PK] character varying (5)	номер_вагона integer	порядковый_номер integer	базовая_стоимость money	стоимость_доп_услуг money
1	8		9	\$70.00	\$0.00

5. Выручка от продажи билетов на все поезда за прошедшие сутки.

```
select sum(место.базовая_стоимость)+sum(место.стоимость_доп_услуг)
from место, билет where место.код_места = билет.код_места
and билет.время_продажи >= cast(now() as date) - interval '1 day'
and билет.время_продажи < cast(now() as date)
and билет.статус_билета = 'оплачен';
```

```

1 select sum(место.базовая_стоимость)+sum(место.стоимость_доп_услуг) from место, билет where место.код_места = билет.код_места
2 and билет.время_продажи >= cast(now() as date) - interval '1 day' and билет.время_продажи < cast(now() as date)
3 and билет.статус_билета = 'оплачен';
4

```

Data Output   Сообщения   Notifications



	?column? money
1	\$35.00

6. Общее количество билетов, проданных по всем направлениям в вагоны типа “СВ”.

```

select count(билет.номер_билета) from билет, место, вагон, тип_вагона
where билет.код_места = место.код_места and место.номер_вагона =
вагон.номер_вагона
and вагон.id_типа_вагона = тип_вагона.id_типа and билет.статус_билета =
'оплачен'
and substring(тип_вагона.класс_обслуживания, 1, 1) = '1'

```

```

1 select count(билет.номер_билета) from билет, место, вагон, тип_вагона where билет.код_места = место.код_места and место.номер_вагона = вагон.номер_вагона
2 and вагон.id_типа_вагона = тип_вагона.id_типа and билет.статус_билета = 'оплачен'
3 and substring(тип_вагона.класс_обслуживания, 1, 1) = '1'
4
5

```

Data Output   Сообщения   Notifications



	count bigint
1	1



7. Номера и названия поездов, все вагоны которых были заполнены менее чем наполовину за прошедшие сутки.

```
create table places as
(select вагон.номер_вагона, count(место.*) from место, вагон, рейс
where место.номер_вагона = вагон.номер_вагона and вагон.номер_рейса
= рейс.номер_поезда
and (рейс.дата_отправления >= cast(now() as date) - interval '1
day')
and (рейс.дата_отправления < cast(now() as date))
group by вагон.номер_вагона);

create table tickets as
(select вагон.номер_вагона, count(билет.*) from билет, место, вагон,
рейс where билет.код_места = место.код_места and место.номер_вагона
= вагон.номер_вагона
and вагон.номер_рейса = рейс.номер_поезда
and (рейс.дата_отправления >= cast(now() as date) - interval '1
day')
and (рейс.дата_отправления < cast(now() as date) )
group by вагон.номер_вагона);

create table num as
(select рейс.номер_поезда, count(вагон.*) from рейс, places, вагон,
tickets where (рейс.номер_поезда = вагон.номер_рейса
and tickets.номер_вагона = вагон.номер_вагона and
places.номер_вагона = вагон.номер_вагона
and places.count - tickets.count > places.count / 2)
or (рейс.номер_поезда = вагон.номер_рейса and places.номер_вагона =
вагон.номер_вагона and places.номер_вагона not in
(tickets.номер_вагона))
group by рейс.номер_поезда);

create table vagon_num as
(select рейс.номер_поезда, count(вагон.*) from рейс, вагон where
рейс.номер_поезда = вагон.номер_рейса group by рейс.номер_поезда);

select рейс.номер_поезда, расписание.название_маршрута from рейс,
расписание, num, vagon_num where рейс.номер_поезда =
num.номер_поезда
and рейс.номер_маршрута = расписание.номер_маршрута and
vagon_num.номер_поезда = num.номер_поезда
and num.count = vagon_num.count
```

```
drop table places;
drop table tickets;
drop table num;
```

```
26 group by вагон.номер_вагона);
27
28 create table num as
29 (select рейс.номер_поезда, count(вагон.*) from рейс, places, вагон, tickets where (рейс.номер_поезда = вагон.номер_рейса
30 and tickets.номер_вагона = вагон.номер_вагона and places.номер_вагона = вагон.номер_вагона
31 and places.count - tickets.count > places.count / 2)
32 or (рейс.номер_поезда = вагон.номер_рейса and places.номер_вагона = вагон.номер_вагона and places.номер_вагона not in (tickets.номер_вагона))
33 group by рейс.номер_поезда);
34
35 create table вагон_num as
36 (select рейс.номер_поезда, count(вагон.*) from рейс, вагон where рейс.номер_поезда = вагон.номер_рейса group by рейс.номер_поезда);
37
38 select рейс.номер_поезда, расписание.название_маршрута from рейс, расписание, num, вагон_num where рейс.номер_поезда = num.номер_поезда
39 and рейс.номер_маршрута = расписание.номер_маршрута and вагон_num.номер_поезда = num.номер_поезда
40 and num.count = вагон_num.count
41
42 drop table places;
43 drop table tickets;
44 drop table num;
45
```

Data Output    Сообщения    Notifications

	номер_поезда integer	название_маршрута character varying (50)
1	7	Санкт-Петербург - Москва

- Представления:

1. Представление для пассажиров о наличии свободных мест

```
create or replace view empty_seats as
select train.departure_time, carriage_type.carriage_category,
count(seat.*), schedule.name_of_route from
schedule, train, carriage, carriage_type, seat where
not seat.if_occupied and seat.carriage_number =
carriage.carriage_number and carriage.train_number =
train.train_number
and train.route_number = schedule.route_number and
carriage.id_carriage_type = carriage_type.type_id and
train.departure_time > now()
group by train.train_number, carriage_type.type_id,
schedule.route_number
```

```

1 create or replace view empty_seats as
2 select train.departure_time, carriage_type.carriage_category, count(seat.*), schedule.name_of_route from
3 schedule, train, carriage, carriage_type, seat where
4 not seat.if_occupied and seat.carriage_number = carriage.carriage_number and carriage.train_number = train.train_number
5 and train.route_number = schedule.route_number and carriage.id_carriage_type = carriage_type.type_id and train.departure_time > now()
6 group by train.train_number, carriage_type.type_id, schedule.route_number

```

Data Output   Сообщения   Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 117 мсек.

	departure_time timestamp without time zone	carriage_category character (1)	count bigint	name_of_route character varying (50)
1	2023-06-04 00:00:00	П	1	Москва - Санкт-Петербург
2	2023-06-17 13:00:00	П	2	Санкт-Петербург - Выборг

- Количество непроданных билетов на все поезда, формирующиеся за прошедшие сутки (номер поезда, тип вагона, количество).

```

create or replace view unsold_tickets as
select рейс.номер_поезда, тип_вагона.категория_вагона,
count(место.*) from рейс, вагон, тип_вагона, место
where место.код_места not in (select код_места from билет where
статус_билета = 'оплачен') and место.номер_вагона =
вагон.номер_вагона
and вагон.id_типа_вагона = тип_вагона.id_типа and вагон.номер_рейса
= рейс.номер_поезда
and (рейс.дата_отправления >= cast(now() as date) - interval '1
day') and (рейс.дата_отправления < cast(now() as date))
group by рейс.номер_поезда, тип_вагона.категория_вагона;

```

```

1 create or replace view unsold_tickets as
2 select рейс.номер_поезда, тип_вагона.категория_вагона, count(место.*) from рейс, вагон, тип_вагона, место
3 where место.код_места not in (select код_места from билет where статус_билета = 'оплачен') and место.номер_вагона = вагон.номер_вагона
4 and вагон.id_типа_вагона = тип_вагона.id_типа and вагон.номер_рейса = рейс.номер_поезда
5 and (рейс.дата_отправления >= cast(now() as date) - interval '1 day') and (рейс.дата_отправления < cast(now() as date))
6 group by рейс.номер_поезда, тип_вагона.категория_вагона;
7

```

Data Output   Сообщения   Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 301 мсек.

	номер_поезда integer	категория_вагона character (1)	count bigint
1	9	П	1
2	9	С	1

- Запросы на модификацию данных

1. **Insert.** Добавить билет на рейс Москва - Санкт-Петербург на 6 июня 2023 в вагон типа 2А, пассажир с номером паспорта '7777888888', оплата безналично.

```

insert into билет (код_места, id_пассажира,
номер_остановки_отправления, номер_остановки_назначения,
статус_билета, тип_оплаты, статус_оплаты, время_продажи)
values ((select место.код_места from место, вагон, тип_вагона, рейс,
расписание
where место.номер_вагона = вагон.номер_вагона and
вагон.id_типа_вагона = тип_вагона.id_типа and вагон.номер_рейса =
рейс.номер_поезда and рейс.номер_маршрута =
расписание.номер_маршрута and тип_вагона.класс_обслуживания = '2А'
and расписание.название_маршрута = 'Москва - Санкт-Петербург' and
место.код_места not in (select код_места from билет) and
cast(рейс.дата_отправления as date) = '2023-06-04'), (select
id_пассажир from пассажир where номер_паспорта = '7777888888'),
(select номер_остановки from остановка where название_остановки =
'Москва'), (select номер_остановки from остановка where
название_остановки = 'Санкт-Петербург'), 'оплачен', 'безналичная',
'успешно', now());

```

```

6 insert into билет (код_места, id_пассажира, номер_остановки_отправления, номер_остановки_назначения, статус_билета, тип_оплаты, статус_оплаты, время_продажи)
7 values ((select место.код_места from место, вагон, тип_вагона, рейс, расписание
8 where место.номер_вагона = вагон.номер_вагона and вагон.id_типа_вагона = тип_вагона.id_типа
9 and вагон.номер_рейса = рейс.номер_поезда and рейс.номер_маршрута = расписание.номер_маршрута
10 and тип_вагона.класс_обслуживания = '2A' and расписание.название_маршрута = 'Москва - Санкт-Петербург'
11 and место.код_места not in (select код_места from билет) and cast(рейс.дата_отправления as date) = '2023-06-04'),
12 (select id_пассажир from пассажир where номер_паспорта = '7777888888'),
13 (select номер_остановки from остановка where название_остановки = 'Москва'),
14 (select номер_остановки from остановка where название_остановки = 'Санкт-Петербург'),
15 'оплачен', 'безналичная', 'успешно', now());
16
17 select * from билет
18

```

Data Output Сообщения Notifications

	номер_билета [PK] integer	код_места character varying (5)	номер_кассы integer	id_пассажира integer	номер_остановки_отправления integer	номер_остановки_назначения integer	статус_билета character varying (20)	тип_оплаты character varying (20)	статус_оплаты character varying (20)	время_продажи timestamp without time zone
1	3	1		2	2	6	оплачен	безналичная	успешно	2023-01-24 18:18:00
2	4	2		[null]	3	6	оплачен	безналичная	успешно	2023-01-25 14:37:00
3	5	3		3	4	8	оплачен	наличная	успешно	2023-03-25 15:01:50
4	6	4		3	5	8	оплачен	наличная	успешно	2023-03-25 15:02:30
5	7	7		[null]	2	6	оплачен	безналичная	успешно	2023-05-01 09:00:00
6	8	6		[null]	3	6	оплачен	безналичная	успешно	2023-04-01 19:00:00
7	9	9		[null]	4	6	оплачен	безналичная	успешно	2023-05-02 20:18:00
8	10	8		[null]	5	6	оплачен	безналичная	успешно	2023-04-20 20:28:00
9	11	15		[null]	3	8	оплачен	безналичная	статус_оплаты	2023-04-30 13:05:00
10	15	18		[null]	5	8	оплачен	безналичная	успешно	2023-05-04 14:21:39.241114

(Добавился билет №15).

- Update.** Изменить в последнем купленном билете пассажира, на пассажира с номером паспорта 1111222222.

```

update билет set id_пассажира = (select id_пассажир from пассажир
where номер_паспорта='1111222222')
where номер_билета = (select номер_билета from билет order by
время_продажи desc limit 1);

```

```

1 update билет set id_пассажира = (select id_пассажир from пассажир where номер_паспорта='1111222222')
2 where номер_билета = (select номер_билета from билет order by время_продажи desc limit 1);
3
4 select * from билет

```

Data Output Сообщения Notifications

	номер_билета [PK] integer	код_места character varying (5)	номер_кассы integer	id_пассажира integer	номер_остановки_отправления integer	номер_остановки_назначения integer	статус_билета character varying (20)	тип_оплаты character varying (20)	статус_оплаты character varying (20)	время_продажи timestamp without time zone
1	3	1		2	2	6	оплачен	безналичная	успешно	2023-01-24 18:18:00
2	4	2		[null]	3	6	оплачен	безналичная	успешно	2023-01-25 14:37:00
3	5	3		3	4	8	оплачен	наличная	успешно	2023-03-25 15:01:50
4	6	4		3	5	8	оплачен	наличная	успешно	2023-03-25 15:02:30
5	7	7		[null]	2	6	оплачен	безналичная	успешно	2023-05-01 09:00:00
6	8	6		[null]	3	6	оплачен	безналичная	успешно	2023-04-01 19:00:00
7	9	9		[null]	4	6	оплачен	безналичная	успешно	2023-05-02 20:18:00
8	10	8		[null]	5	6	оплачен	безналичная	успешно	2023-04-20 20:28:00
9	11	15		[null]	3	8	оплачен	безналичная	статус_оплаты	2023-04-30 13:05:00
10	15	18		[null]	2	8	оплачен	безналичная	успешно	2023-05-04 14:21:39.241114

### 3. Delete. Удалить записи обо всех билетах на рейсы 4 февраля 2023.

```
delete from билет where код_места in (select место.код_места from
место, вагон, рейс where место.номер_вагона = вагон.номер_вагона and
вагон.номер_рейса = рейс.номер_поезда and рейс.дата_отправления =
'2023-02-04');
```

```
1 delete from билет where код_места in (select место.код_места from место, вагон, рейс
2                                     where место.номер_вагона = вагон.номер_вагона and вагон.номер_рейса = рейс.номер_поезда
3                                     and рейс.дата_отправления = '2023-02-04');
4
5 select * from билет
```

Data Output										
Сообщения										
Notifications										
	номер_билета [PK] integer	код_места character varying (5)	номер_касы integer	id_пассажира integer	номер_остановки_отправления integer	номер_остановки_назначения integer	статус_билета character varying (20)	тип_оплаты character varying (20)	статус_оплаты character varying (20)	время_продажи timestamp without time zone
1	5	3	3	4	8	6	оплачен	наличная	успешно	2023-03-25 15:01:50
2	6	4	3	5	8	6	оплачен	наличная	успешно	2023-03-25 15:02:30
3	7	7	[null]	2	6	8	оплачен	безналичная	успешно	2023-05-01 09:00:00
4	8	6	[null]	3	6	8	оплачен	безналичная	успешно	2023-04-01 19:00:00
5	9	9	[null]	4	6	7	оплачен	безналичная	успешно	2023-05-02 20:18:00
6	10	8	[null]	5	6	8	оплачен	безналичная	успешно	2023-04-20 20:28:00
7	11	15	[null]	3	8	6	оплачен	безналичная	статус_оплаты	2023-04-30 13:05:00
8	15	18	[null]	2	8	6	оплачен	безналичная	успешно	2023-05-04 14:21:39.241114

- Создание индексов

- Простой индекс. Запрос – все билеты пассажира №2.

До создания индекса запрос выполнялся 238 мс.

```
1 select * from билет where id_пассажира = 2
```

	номер_билета [PK] integer	код_места character varying (5)	номер_кассы integer	id_пассажира integer	номер_остановки_отправления integer	номер_остановки_назначения integer	статус_билета character varying (20)	тип_оплаты character varying (20)	статус_оплаты character varying (20)	время_продажи timestamp without time zone
1	7	7	[null]	2	6	8	оплачен	безналичная	успешно	2023-05-01 09:00:00
2	15	18	[null]	2	8	6	оплачен	безналичная	успешно	2023-05-04 14:21:39.241114

Total rows: 2 of 2    Query complete 00:00:00.238

✓ Запрос выполнен успешно. Общее время выполнения: 238 msec. обработано строк: 2. ✕

После создания индекса – 188 мс.

```
1 --drop index ticket_passanger;
2
3
4 --create index ticket_passanger on билет(id_пассажира);
5
6 select * from билет where id_пассажира = 2
```

	номер_билета [PK] integer	код_места character varying (5)	номер_кассы integer	id_пассажира integer	номер_остановки_отправления integer	номер_остановки_назначения integer	статус_билета character varying (20)	тип_оплаты character varying (20)	статус_оплаты character varying (20)	время_продажи timestamp without time zone
1	7	7	[null]	2	6	8	оплачен	безналичная	успешно	2023-05-01 09:00:00
2	15	18	[null]	2	8	6	оплачен	безналичная	успешно	2023-05-04 14:21:39.241114

✓ Запрос выполнен успешно. Общее время выполнения: 188 msec. обработано строк: 2. ✕

2. Составной индекс. Запрос – все билеты пассажира №2, купленные сегодня.

До создания индекса запрос выполняется 304 мс.

3

```
select * from билет where id_пассажира = 2 and cast(время_продажи as date) = cast(now() as date)
```

Data OutputСообщенияNotifications

	номер_билета [PK] integer	код_места character varying (5)	номер_кассы integer	id_пассажира integer	номер_остановки_отправления integer	номер_остановки_назначения integer	статус_билета character varying (20)	тип_оплаты character varying (20)	статус_оплаты character varying (20)	время_продажи timestamp without time zone
1	15	18	[null]	2	8	6	оплачен	безналичная	успешно	2023-05-04 14:21:39.241114

✓ Запрос выполнен успешно. Общее время выполнения: 304 мсек. обработано строк: 1. ✕

После создания индекса – 189 мс.

1

2

3

4

5

```
--create index ticket_passanger_today on билет(id_пассажира, время_продажи);  
  
select * from билет where id_пассажира = 2 and cast(время_продажи as date) = cast(now() as date)
```

Data OutputСообщенияNotifications

	номер_билета [PK] integer	код_места character varying (5)	номер_кассы integer	id_пассажира integer	номер_остановки_отправления integer	номер_остановки_назначения integer	статус_билета character varying (20)	тип_оплаты character varying (20)	статус_оплаты character varying (20)	время_продажи timestamp without time zone
1	15	18	[null]	2	8	6	оплачен	безналичная	успешно	2023-05-04 14:21:39.241114

✓ Запрос выполнен успешно. Общее время выполнения: 189 мсек. обработано строк: 1. ✕



## **Выводы**

Выполнив лабораторную работу, я овладела навыками создания запросов и представлений на выборку данных, использования подзапросов в модификации данных. Также я изучила графическое представление базы данных, просмотрела историю запросов, благодаря чему смогла оценить производительность базы данных. Также я изучила создание индексов и на практике увидела, как они улучшают производительность и ускоряют выполнение запросов.