

**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: Полухин А. В.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

## Оглавление

<b>Цель работы</b>	<b>3</b>
<b>Практическое задание (Вариант 19. БД «Пассажир»)</b>	<b>3</b>
Описание предметной области:	3
Задание 2. Создать запросы:	3
Задание 3. Создать представление:	4
<b>Выполнение лабораторной работы</b>	<b>4</b>
1. Запросы к базе данных:	4
• Свободные места на все поезда, отправляющиеся с вокзала в течение следующих суток.	4
• Список пассажиров, отправившихся в Москву всеми рейсами за прошедшие сутки.	5
• Номера поездов, на которые проданы все билеты на следующие сутки.	6
• Свободные места в купейные вагоны всех рейсов до Москвы на текущие сутки.	7
• Выручка от продажи билетов на все поезда за прошедшие сутки.	7
• Общее количество билетов, проданных по всем направлениям в вагоны типа “СВ”.	8
2. Создание представлений	8
• Для пассажиров о наличии свободных мест на заданный рейс (номер поезда, название поезда, время отправления, станция отправления, станция прибытия, количество мест в вагоне);	8
• Количество непроданных билетов на все поезда, формирующиеся за прошедшие сутки (номер поезда, тип вагона, количество).	9
3. Запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов	9
• Если поезд отменён, то удаляем места из поезда	9
• Обновление статуса билета, если пассажир его вернул	9
• Добавляем места, которые были недоступны до определённой даты	10
4. Создание простого и составного индексов для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN	10
• Простой индекс для запроса	10
• Составной индекс для запроса	11
<b>Вывод</b>	<b>11</b>

## 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 (Вариант 19. БД «Пассажир»)

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.

### Описание предметной области:

Информационная система служит для продажи железнодорожных билетов. Билеты могут продаваться на текущие сутки или предварительно (не более чем за 45 суток). Цена билета при предварительной продаже снижается на 5%. Билет может быть приобретен в кассе или онлайн. Если билет приобретен в кассе, необходимо знать, в какой. Для каждой кассы известны номер и адрес. Кассы могут располагаться в различных населенных пунктах.

Поезда курсируют по расписанию, но могут назначаться дополнительные поезда на заданный период или определенные даты.

По всем промежуточным остановкам на маршруте известны название, тип населенного пункта, время прибытия, отправления, время стоянки.

Необходимо учитывать, что местом посадки и высадки пассажира могут быть промежуточные пункты по маршруту.

БД должна содержать следующий минимальный набор сведений: Номер поезда. Название поезда. Тип поезда. Пункт назначения. Пункт назначения для проданного билета. Номер вагона. Тип вагона. Количество мест в вагоне. Цена билета. Дата отправления. Дата прибытия. Дата прибытия для пункта назначения проданного билета. Время отправления. Номер вагона в поезде. Номер билета. Место. Тип места. Фамилия пассажира. Имя пассажира. Отчество пассажира. Паспортные данные.

### Задание 2. Создать запросы:

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

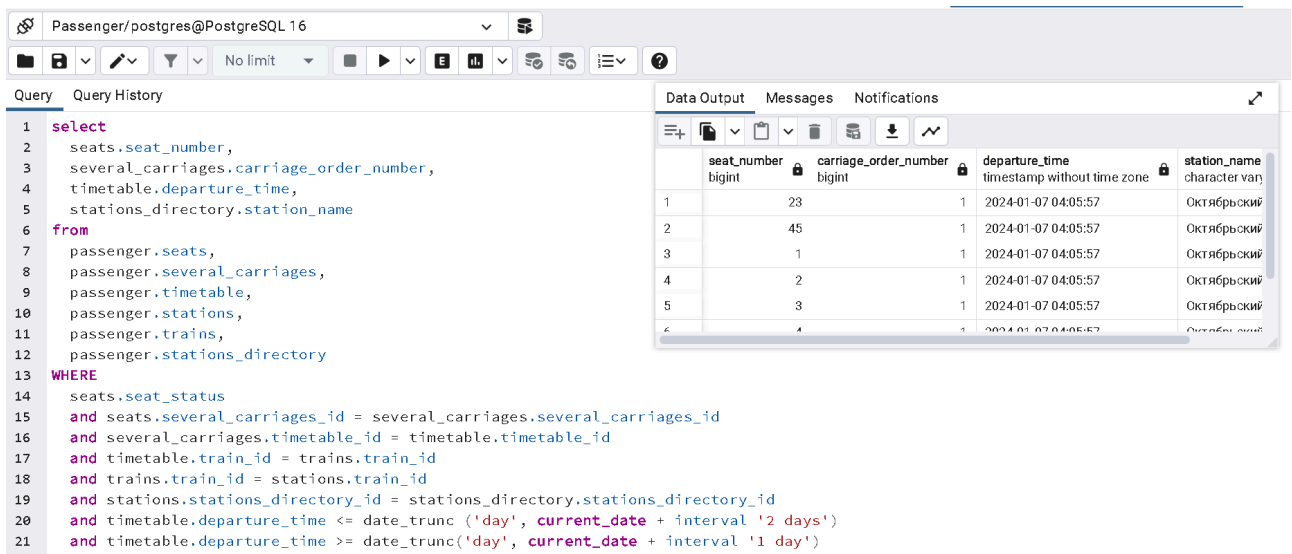
### Задание 3. Создать представление:

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

## Выполнение лабораторной работы

### 1. Запросы к базе данных:

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



The screenshot shows the PostgreSQL interface with a query editor on the left and a data output table on the right.

**Query:**

```

1 select
2   seats.seat_number,
3   several_carriages.carriage_order_number,
4   timetable.departure_time,
5   stations_directory.station_name
6 from
7   passenger.seats,
8   passenger.several_carriages,
9   passenger.timetable,
10  passenger.stations,
11  passenger.trains,
12  passenger.stations_directory
13 WHERE
14   seats.seat_status
15   and seats.several_carriages_id = several_carriages.several_carriages_id
16   and several_carriages.timetable_id = timetable.timetable_id
17   and timetable.train_id = trains.train_id
18   and trains.train_id = stations.train_id
19   and stations.stations_directory_id = stations_directory.stations_directory_id
20   and timetable.departure_time <= date_trunc('day', current_date + interval '2 days')
21   and timetable.departure_time >= date_trunc('day', current_date + interval '1 day')

```

**Data Output:**

	seat_number bigint	carriage_order_number bigint	departure_time timestamp without time zone	station_name character var
1	23	1	2024-01-07 04:05:57	Октябрьский
2	45	1	2024-01-07 04:05:57	Октябрьский
3	1	1	2024-01-07 04:05:57	Октябрьский
4	2	1	2024-01-07 04:05:57	Октябрьский
5	3	1	2024-01-07 04:05:57	Октябрьский
6	4	1	2024-01-07 04:05:57	Октябрьский

Листинг кода:

select

```

seats.seat_number,
several_carriages.carriage_order_number,
timetable.departure_time,
stations_directory.station_name
from
passenger.seats,
passenger.several_carriages,
passenger.timetable,
passenger.stations,
passenger.trains,
passenger.stations_directory
WHERE
seats.seat_status
and seats.several_carriages_id = several_carriages.several_carriages_id
and several_carriages.timetable_id = timetable.timetable_id
and timetable.train_id = trains.train_id
and trains.train_id = stations.train_id
and stations.stations_directory_id = stations_directory.stations_directory_id
and timetable.departure_time <= date_trunc('day', current_date + interval '2 days')
and timetable.departure_time >= date_trunc('day', current_date + interval '1 day')

```

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

Query
Query History

```

1 select
2     DISTINCT seats.seat_id,
3     passengers.name,
4     passengers.surname,
5     passengers.middle_name
6 from
7     passenger.passengers,
8     passenger.tickets,
9     passenger.seats,
10    passenger.several_carriages,
11    passenger.timetable,
12    passenger.stations,
13    passenger.stations_directory
14 where
15     tickets.passenger_id = passengers.passenger_id
16 and tickets.seat_id = seats.seat_id
17 and seats.several_carriages_id = several_carriages.several_carriages_id
18 and several_carriages.timetable_id = timetable.timetable_id
19 and timetable.train_id = stations.train_id
20 and stations.stations_directory_id = stations_directory.stations_directory_id
21 and stations_directory.station_address = 'Комсомольская площадь, 3, Москва'
22 and timetable.departure_time <= date_trunc('day', current_date)
23 and timetable.departure_time >= date_trunc('day', current_date - interval '1 day');

```

Data Output Messages Notifications

	seat_id bigint	name character varying (40)	surname character varying (40)	middle_name character varying (40)
1	0	Elena	Petrova	Vladimirovna
2	5	Aleksandr	Aleksandrov	Aleksandrovich
3	6	Ivan	Ivanov	Ivanovich

**Листинг кода:**

```

select
    DISTINCT seats.seat_id,
    passengers.name,

```

```

passengers.surname,
passengers.middle_name
from
passenger.passengers,
passenger.tickets,
passenger.seats,
passenger.several_carriages,
passenger.timetable,
passenger.stations,
passenger.stations_directory
where
tickets.passenger_id = passengers.passenger_id
and tickets.seat_id = seats.seat_id
and seats.several_carriages_id = several_carriages.several_carriages_id
and several_carriages.timetable_id = timetable.timetable_id
and timetable.train_id = stations.train_id
and stations.stations_directory_id = stations_directory.stations_directory_id
and stations_directory.station_address = 'Комсомольская площадь, 3, Москва'
and timetable.departure_time <= date_trunc('day', current_date)
and timetable.departure_time >= date_trunc('day', current_date - interval '1 day');
```

- **Номера поездов, на которые проданы все билеты на следующие сутки.**

Query
Query History

```

1 with trains_without_seats
2 as (
3   select distinct a.several_carriages_id from passenger.seats a
4   where
5     not exists (
6       select * from passenger.seats b where b.seat_status and b.several_carriages_id = a.several_carriages_id
7     )
8 )
9 select
10  distinct trains.train_number
11 from
12  trains_without_seats, passenger.trains, passenger.timetable
13 where
14  trains_without_seats.several_carriages_id = timetable.timetable_id
15  and timetable.train_id = trains.train_id
16  and timetable.departure_time <= date_trunc('day', current_date + interval '2 days')
17  and timetable.departure_time >= date_trunc('day', current_date + interval '1 day');
```

Data Output Messages Notifications

train_number	bigint
1	214

#### Листинг кода:

```

with trains_without_seats
as (
    select distinct a.several_carriages_id from passenger.seats a
    where
        not exists (
            select * from passenger.seats b where b.seat_status and b.several_carriages_id =
a.several_carriages_id
        )
)
select
    distinct trains.train_number
from
    trains_without_seats, passenger.trains, passenger.timetable
where
    trains_without_seats.several_carriages_id = timetable.timetable_id
    and timetable.train_id = trains.train_id
    and timetable.departure_time <= date_trunc('day', current_date + interval '2 days')
    and timetable.departure_time >= date_trunc('day', current_date + interval '1 day');
```

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

Query Query History

```

1 select distinct train_name, several_carriages.carriage_order_number, seat_number
2 from
3 passenger.timetable,
4 passenger.several_carriages,
5 passenger.carriages,
6 passenger.carriages_directory,
7 passenger.seats,
8 passenger.trains
9 where
10 timetable.departure_time <= date_trunc('day', current_date + interval '1 day')
11 and timetable.departure_time >= date_trunc('day', current_date)
12 and timetable.timetable_id = several_carriages.timetable_id
13 and several_carriages.carriages_id = carriages.carriage_id
14 and carriages.carriage_directory_id = carriages_directory.carriage_directory_id
15 and name_carriage_type = 'compartment'
16 and seats.several_carriages_id = timetable.timetable_id
17 and seats.seat_status
18 and timetable.train_id = trains.train_id;
19

```

Data Output Messages Notifications

	train_name character varying (100)	carriage_order_number bigint	seat_number bigint
1	Ночной экспресс (Санкт-Петербург - Москва)	3	1

**Листинг кода:**

```

select distinct train_name, several_carriages.carriage_order_number, seat_number
from
passenger.timetable,
passenger.several_carriages,
passenger.carriages,
passenger.carriages_directory,
passenger.seats,
passenger.trains
where
timetable.departure_time <= date_trunc('day', current_date + interval '1 day')
and timetable.departure_time >= date_trunc('day', current_date)
and timetable.timetable_id = several_carriages.timetable_id
and several_carriages.carriages_id = carriages.carriage_id
and carriages.carriage_directory_id = carriages_directory.carriage_directory_id
and name_carriage_type = 'compartment'
and seats.several_carriages_id = timetable.timetable_id
and seats.seat_status
and timetable.train_id = trains.train_id;

```

- **Выручка от продажи билетов на все поезда за прошедшие сутки.**

Query Query History

```

1 select sum(seats.price)
2 from
3 passenger.seats,
4 passenger.tickets
5 where
6 tickets.buying_time <= date_trunc('day', current_date)
7 and tickets.buying_time >= date_trunc('day', current_date - interval '1 day')
8 and tickets.seat_id = seats.seat_id;
9

```

Data Output Messages Notifications

	sum money
1	15 693,00 ?

**Листинг кода:**

```

select sum(seats.price)
from
    passenger.seats,
    passenger.tickets
where
    tickets.buying_time <= date_trunc('day', current_date)
    and tickets.buying_time >= date_trunc('day', current_date - interval '1 day')
    and tickets.seat_id = seats.seat_id;

```

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

Query
Query History

```

1 select count(DISTINCT tickets.ticket_id)
2 from
3 passenger.tickets,
4 passenger.seats,
5 passenger.several_carriages,
6 passenger.carriages,
7 passenger.carriages_directory
8 where
9 tickets.seat_id = seats.seat_id
10 and seats.several_carriages_id = several_carriages.several_carriages_id
11 and several_carriages.carriages_id = carriages.carriage_id
12 and carriages.carriage_directory_id = carriages_directory.carriage_directory_id
13 and carriages_directory.name_carriage_type = 'business_class';
14

```

Data Output

Messages

Notifications

	count	bigint
1		0

Листинг кода:

```

select count(DISTINCT tickets.ticket_id)
from
passenger.tickets,
passenger.seats,
passenger.several_carriages,
passenger.carriages,
passenger.carriages_directory
where
tickets.seat_id = seats.seat_id
and seats.several_carriages_id = several_carriages.several_carriages_id
and several_carriages.carriages_id = carriages.carriage_id
and carriages.carriage_directory_id = carriages_directory.carriage_directory_id
and carriages_directory.name_carriage_type = 'business_class';

```

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

Query
Query History

```

1 with
2 occupied_seats as (
3     select
4         several_carriages.timetable_id,
5         count(*) as total_seats,
6         sum(case when seats.seat_status then 0 else 1 end) as occupied_seats
7     from
8         passenger.seats,
9         passenger.several_carriages
10    where
11        seats.several_carriages_id = several_carriages.several_carriages_id
12    group by several_carriages.timetable_id
13 )
14 select
15     distinct
16     trains.train_name, trains.train_number, timetable.departure_time
17 from
18     occupied_seats, passenger.timetable, passenger.trains
19 where
20     occupied_seats.occupied_seats < occupied_seats.total_seats / 2
21 and occupied_seats.timetable_id = timetable.timetable_id
22 and timetable.train_id = trains.train_id
23 and timetable.departure_time between date_trunc('day', current_date - interval '1 day')
24 and date_trunc('day', current_date);

```

Data Output

Messages

Notifications

train_name	train_number	departure_time
character varying (100)	bigint	timestamp without time zone
Стандарт (Санкт-Петербург - Воронеж)	201	2024-01-16 04:05:57

Листинг кода:

```

with
occupied_seats as (
    select

```



```

several_carriages.timetable_id,
count(*) as total_seats,
sum(case when seats.seat_status then 0 else 1 end) as occupied_seats
from
    passenger.seats,
    passenger.several_carriages
where
    seats.several_carriages_id = several_carriages.several_carriages_id
group by several_carriages.timetable_id
)
select
    distinct
    trains.train_name, trains.train_number, timetable.departure_time
from
    occupied_seats, passenger.timetable, passenger.trains
where
    occupied_seats.occupied_seats < occupied_seats.total_seats / 2
and occupied_seats.timetable_id = timetable.timetable_id
and timetable.train_id = trains.train_id
and timetable.departure_time between date_trunc('day', current_date - interval '1 day')
and date_trunc('day', current_date);

```

## 2. Создание представлений

- Для пассажиров о наличии свободных мест на заданный рейс (номер поезда, название поезда, время отправления, станция отправления, станция прибытия, количество мест в вагоне);

Query Query History

```

1 create or replace view passenger.available_seats_for_passengers as
2 with
3     available_seats as (
4         select
5             b.timetable_id,
6             count(*) as total_seats,
7             sum(case when a.seat_status then 0 else 1 end) as available_seats
8         from
9             passenger.seats a,
10            passenger.several_carriages b
11        where
12            a.several_carriages_id = b.several_carriages_id
13        group by b.timetable_id
14    )
15 select
16     c.train_name,
17     c.train_number,
18     b.departure_time,
19     e.station_name as departure_station,
20     b.arrival_time,
21     g.station_name as arrival_station

```

Data Output Messages Notifications

	train_name character varying (100)	train_number bigint	departure_time timestamp without time zone	departure_station character varying
1	Стандарт (Санкт-Петербург - Воронеж)	214	2024-01-14 04:05:57	Московский вок:
2	Стандарт (Санкт-Петербург - Воронеж)	214	2024-01-14 04:05:57	Московский вок:
3	Стандарт (Санкт-Петербург - Воронеж)	214	2024-01-14 04:05:57	Московский вок:
4	Стандарт (Санкт-Петербург - Воронеж)	214	2024-01-14 04:05:57	Московский вок:
5	Стандарт (Санкт-Петербург - Воронеж)	214	2024-01-14 04:05:57	Московский вок:
6	Стандарт (Санкт-Петербург - Воронеж)	214	2024-01-14 04:05:57	Московский вок:

### Листинг кода:

```

create or replace view passenger.available_seats_for_passengers as
with
    available_seats as (
        select
            several_carriages.timetable_id,
            count(*) as total_seats,
            sum(case when seats.seat_status then 0 else 1 end) as available_seats
        from
            passenger.seats,

```

```

        passenger.several_carriages
    where
        seats.several_carriages_id = several_carriages.several_carriages_id
        group by several_carriages.timetable_id
    )
select
    trains.train_name,
    trains.train_number,
    timetable.departure_time,
    e.station_name as departure_station,
    timetable.arrival_time,
    g.station_name as arrival_station
from
    available_seats,
    passenger.timetable,
    passenger.trains,
    passenger.stations d,
    passenger.stations_directory e,
    passenger.stations f,
    passenger.stations_directory g
where
    available_seats.timetable_id = timetable.timetable_id
    and timetable.train_id = trains.train_id
    and trains.train_id = d.train_id
    and d.order_number = 1
    and f.station_id = d.station_id
    and trains.train_id = f.train_id
    and d.order_number = -1
    and f.station_id = f.station_id
    and timetable.departure_time > now()
;
select * from passenger.available_seats_for_passengers;

```

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

Query
Query History

```

1 create or replace view passenger.available_seats_on_yesterday as
2 with
3     available_seats as (
4         select
5             several_carriages.timetable_id,
6             several_carriages.several_carriages_id,
7             sum(case when seats.seat_status then 1 else 0 end) as available_seats
8         from
9             passenger.seats,
10            passenger.several_carriages
11        where
12            seats.several_carriages_id = several_carriages.several_carriages_id
13        group by several_carriages.timetable_id, several_carriages.several_carriages_id
14    )
15    select
16        trains.train_name,
17        trains.train_number,
18        timetable.departure_time,
19        carriages_directory.name_carriage_type,
20        sum(available_seats.available_seats) as available_seats
21    from
22        available_seats,
23        passenger.timetable,
24        passenger.trains,
25        passenger.carriages,
26        passenger.carriages_directory

```

Data Output
Messages
Notifications

	train_name character varying (100)	train_number bigint	departure_time timestamp without time zone	name chara
1	Ночной экспресс (Санкт-Петербург - Москва)	301	2024-03-06 04:05:57	comp
2	Ночной экспресс (Санкт-Петербург - Москва)	301	2024-03-06 09:35:32	comp
3	Стандарт (Москва - Сочи)	202	2024-03-06 11:12:42	comp
4	Стандарт (Санкт-Петербург - Воронеж)	201	2024-03-06 04:05:57	comp

✓ Data saved successfully. ✕

#### Листинг кода:

```

create or replace view passenger.available_seats_on_yesterday as
with

```

```

available_seats as (
select
    several_carriages.timetable_id,
    several_carriages.several_carriages_id,
    sum(case when seats.seat_status then 1 else 0 end) as available_seats
from
    passenger.seats,
    passenger.several_carriages
where
    seats.several_carriages_id = several_carriages.several_carriages_id
group by several_carriages.timetable_id, several_carriages.several_carriages_id
)
select
    trains.train_name,
    trains.train_number,
    timetable.departure_time,
    carriages_directory.name_carriage_type,
    sum(available_seats.available_seats) as available_seats
from
    available_seats,
    passenger.timetable,
    passenger.trains,
    passenger.carriages,
    passenger.carriages_directory,
    passenger.several_carriages
where
    available_seats.timetable_id = timetable.timetable_id
    and timetable.train_id = trains.train_id
    and timetable.departure_time <= date_trunc('day', current_date)
    and timetable.departure_time >= date_trunc('day', current_date - interval '1 day')
    and available_seats.several_carriages_id = several_carriages.several_carriages_id
    and carriages.carriage_id = several_carriages.carriages_id
    and carriages.carriage_directory_id = carriages_directory.carriage_directory_id
group by
    trains.train_name,
    trains.train_number,
    timetable.departure_time,
    carriages_directory.name_carriage_type
;
select * from passenger.available_seats_on_yesterday;

```

### 3. Запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов

- **Если поезд удалён из расписания, то удаляем места из поезда**

```
1 delete from passenger.seats where seats.several_carriages_id in
2 (
3     select
4         several_carriages.several_carriages_id
5     from
6         passenger.timetable,
7         passenger.several_carriages
8     where
9         timetable.timetable_id = several_carriages.timetable_id
10        and timetable.status = 'canceled'
11 );
12
```

Data Output Messages Notifications

DELETE 1

Query returned successfully in 183 msec.

```
delete from passenger.seats where seats.several_carriages_id in
(
    select
        several_carriages.several_carriages_id
    from
        passenger.timetable,
        passenger.several_carriages
    where
        timetable.timetable_id = several_carriages.timetable_id
        and timetable.status = 'canceled'
);
```

- **Обновление статуса билета, если пассажир его вернул**

Query Query History

```
1 update passenger.seats
2 set seat_status = true
3 where seat_id in(
4     select
5         seats.seat_id
6     from
7         passenger.seats,
8         passenger.tickets
9     where
10        seats.seat_id = tickets.seat_id
11        and tickets.status = 'returned'
12 );
13
14
```

Data Output Messages Notifications

UPDATE 2

Query returned successfully in 93 msec.

```
update passenger.seats
set seat_status = true
where seat_id in(
    select
        seats.seat_id
    from
        passenger.seats,
```

```

        passenger.tickets
    where
        seats.seat_id = tickets.seat_id
        and tickets.status = 'returned'
);

```

- **Добавляем места, которые были недоступны до определённой даты**

```

2 delete from passenger.seats where several_carriages_id = 1;
3 insert into passenger.seats(seat_id, seat_number, seat_status, price, several_carriages_id)
4 select
5     nextval('passenger.seat_id_seq'::regclass),
6     generate_series(1, a.number_of_seats),
7     true,
8     1000::money,
9     c.several_carriages_id
10 from
11     passenger.carriages_directory a,
12     passenger.carriages b,
13     passenger.several_carriages c
14 where
15     c.several_carriages_id = 1
16     and c.carriage_id = b.carriage_id
17     and b.carriage_directory_id = a.carriage_directory_id;
18

```

Data Output   Messages   Notifications

CREATE SEQUENCE

Query returned successfully in 38 msec.

```

delete from passenger.seats where several_carriages_id = 1;
insert into passenger.seats(seat_id, seat_number, seat_status, price, several_carriages_id)
select
    nextval('passenger.seat_id_seq'::regclass),
    generate_series(1, a.number_of_seats),
    true,
    1000::money,
    c.several_carriages_id
from
    passenger.carriages_directory a,
    passenger.carriages b,
    passenger.several_carriages c
where
    c.several_carriages_id = 1
    and c.carriage_id = b.carriage_id
    and b.carriage_directory_id = a.carriage_directory_id;

```

#### 4. Создание простого и составного индексов для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN

- **Простой индекс для запроса**

Query

Query History

1

create index if not exists seats\_several\_carriages\_id\_index on

2

passenger.seats(several\_carriages\_id);

Data Output

Messages

Notifications

CREATE INDEX

Query returned successfully in 46 msec.

explain analyse

select distinct train\_name, several\_carriages.carriage\_order\_number, seat\_number  
from

passenger.timetable,  
passenger.several\_carriages,  
passenger.carriages,  
passenger.seats,  
passenger.trains

where

timetable.departure\_time <= date\_trunc('day', current\_date + interval '1 day')  
and timetable.departure\_time >= date\_trunc('day', current\_date)  
and timetable.timetable\_id = several\_carriages.timetable\_id  
and several\_carriages.carriages\_id = carriages.carriage\_id  
and carriage\_directory\_id = 1  
and seats.several\_carriages\_id = timetable.timetable\_id  
and seats.seat\_status  
and timetable.train\_id = trains.train\_id;

**Без использования индекса:** Execution Time: 0.325 ms

**С использованием индекса:** Execution Time: 0.286 ms

- **Составной индекс для запроса**

1

create unique index if not exists timetable\_uk\_1 on

2

passenger.timetable(train\_id, departure\_time, timetable\_id);

3

drop index passenger.timetable\_uk\_1;

4

explain analyse select train\_id, departure\_time, timetable\_id from

5

passenger.timetable;

**Без использования индекса:** Execution Time: 0.027 ms

**С использованием индекса:** Execution Time: 0.030 ms

## Вывод

Были изучены практические навыки создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.