

**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»**

**Обучающийся Малахов Алексей Витальевич**

**Факультет прикладной информатики**

**Группа К3239**

**Направление подготовки 09.03.03 Прикладная информатика**

**Образовательная программа Мобильные и сетевые технологии 2023**

**Преподаватель Говорова Марина Михайловна**

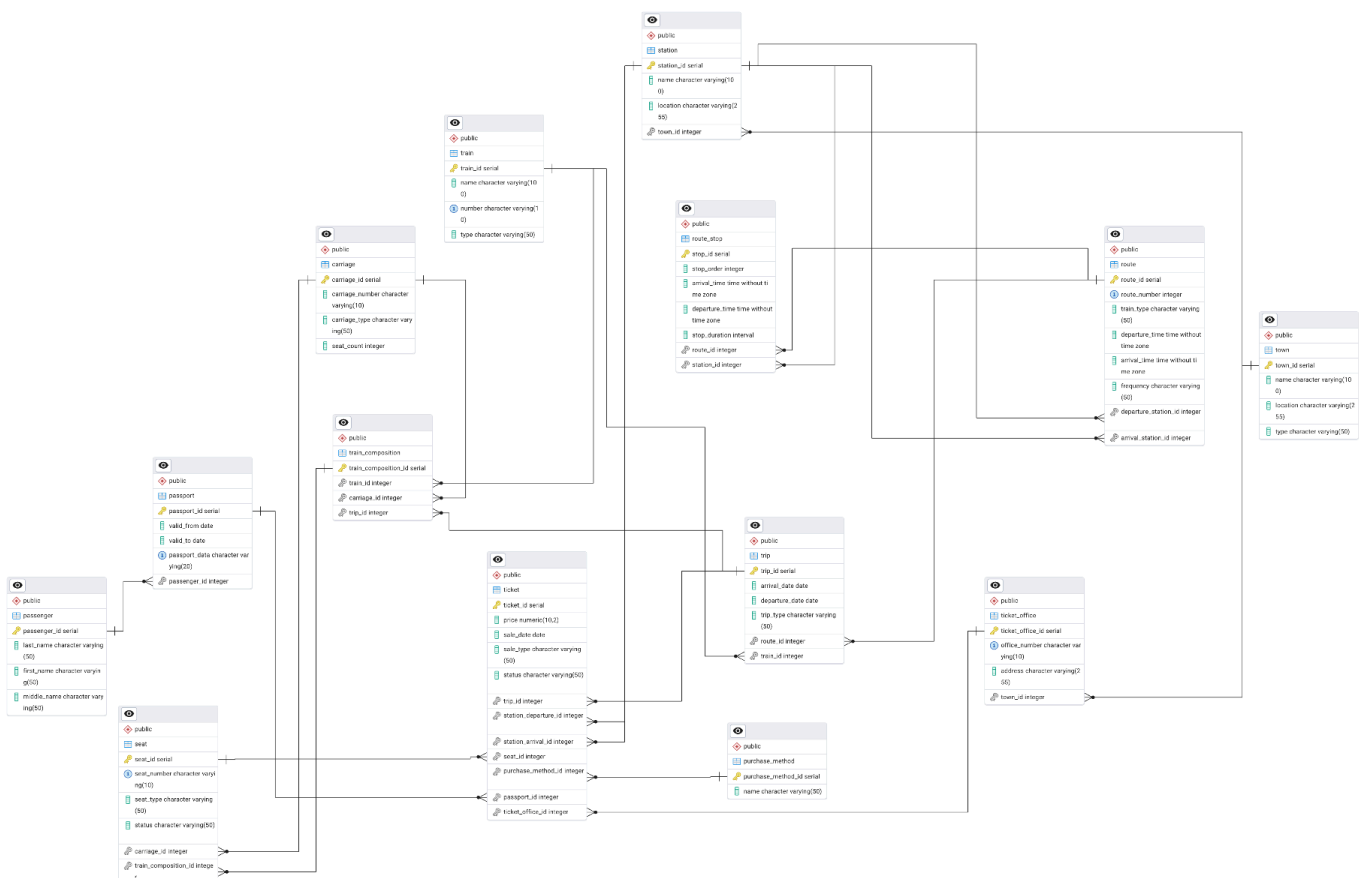
Санкт-Петербург  
2025/2026

1. **Цель работы:** овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.
2. **Практическое задание:**
  - Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию лабораторной работы №2, часть 2 и 3).
  - Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
  - Изучить графическое представление запросов и просмотреть историю запросов.
  - 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.

#### 4. **Выполнение:**

1. Наименование создаваемой БД: **Rail Ticket Database**

2. Схема логической модели базы данных, сгенерированная в Generate ERD:



### 3. Скрипты запросов:

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

```
SELECT
    tr.train_id,
    tr.number      AS train_number,
    tr.name        AS train_name,
    COUNT(s.seat_id)           AS total_seats,
    COUNT(tk.seat_id)          AS sold_seats,
    COUNT(s.seat_id) - COUNT(tk.seat_id) AS free_seats
FROM trip tp
JOIN route r ON tp.route_id = r.route_id
JOIN train tr ON tp.train_id = tr.train_id
JOIN train_composition tc ON tp.trip_id = tc.trip_id
JOIN seat s ON tc.train_composition_id = s.train_composition_id
LEFT JOIN ticket tk ON tk.trip_id = tp.trip_id AND tk.seat_id = s.seat_id AND
tk.status = 'Куплен'
WHERE r.departure_station_id = 2
    AND tp.departure_date = CURRENT_DATE + 1
GROUP BY tr.train_id, tr.number, tr.name;
```

2. Список поездов за прошедшие сутки с максимальной выручкой о продажи билетов.

```
WITH daily_revenue AS (
    SELECT
        tr.train_id,
        tr.number      AS train_number,
        tr.name        AS train_name,
        SUM(tk.price)   AS total_revenue
    FROM ticket tk
    JOIN trip tp
        ON tk.trip_id = tp.trip_id
    JOIN train tr
        ON tp.train_id = tr.train_id
    WHERE tk.sale_date = CURRENT_DATE - 1
        AND tk.status = 'Куплен'
    GROUP BY tr.train_id, tr.number, tr.name
)
SELECT
    train_id,
    train_number,
    train_name,
    total_revenue
FROM daily_revenue
WHERE total_revenue = (
    SELECT MAX(total_revenue)
    FROM daily_revenue
)
```

```
) ;
```

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

```
SELECT
    tr.number AS train_number
FROM trip tp
JOIN train tr ON tp.train_id = tr.train_id
JOIN train_composition tc ON tp.trip_id = tc.trip_id
JOIN seat s ON tc.train_composition_id = s.train_composition_id
LEFT JOIN ticket tk
    ON tk.trip_id = tp.trip_id
    AND tk.seat_id = s.seat_id
    AND tk.status = 'Куплен'
WHERE tp.departure_date = CURRENT_DATE + 1
GROUP BY tr.number
HAVING COUNT(s.seat_id) = COUNT(tk.ticket_id);
```

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

```
SELECT
    tp.trip_id,
    tr.number AS train_number,
    tr.name AS train_name,
    COUNT(s.seat_id) - COUNT(tk.ticket_id) AS free_coupe_seats
FROM trip tp
JOIN route r
    ON tp.route_id = r.route_id
JOIN station st
    ON r.arrival_station_id = st.station_id
JOIN town tw
    ON st.town_id = tw.town_id
AND tw.name = 'Москва'
JOIN train tr
    ON tp.train_id = tr.train_id
JOIN train_composition tc
    ON tp.trip_id = tc.trip_id
JOIN carriage c
    ON tc.carriage_id = c.carriage_id
AND c.carriage_type = 'Купе'
JOIN seat s
    ON s.train_composition_id = tc.train_composition_id
LEFT JOIN ticket tk
    ON tk.trip_id = tp.trip_id
    AND tk.seat_id = s.seat_id
    AND tk.status = 'Куплен'
WHERE tp.departure_date = CURRENT_DATE
GROUP BY tp.trip_id, tr.number, tr.name;
```

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

```

SELECT
    SUM(tk.price) AS total_revenue
FROM ticket tk
WHERE tk.sale_date = CURRENT_DATE - 1
AND tk.status = 'Куплен';

```

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

```

SELECT
    COUNT(*) AS total_sv_tickets
FROM ticket tk
JOIN seat s
    ON tk.seat_id = s.seat_id
JOIN carriage c
    ON s.carriage_id = c.carriage_id
WHERE c.carriage_type = 'CB'
AND tk.status = 'Куплен';

```

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

```

SELECT
    tr.number AS train_number,
    tr.name AS train_name
FROM trip tp
JOIN train tr
    ON tp.train_id = tr.train_id
JOIN train_composition tc
    ON tp.trip_id = tc.trip_id
JOIN seat s
    ON tc.train_composition_id = s.train_composition_id
LEFT JOIN ticket tk
    ON tk.trip_id = tp.trip_id
AND tk.seat_id = s.seat_id
AND tk.status = 'Куплен'
WHERE tp.departure_date = CURRENT_DATE - 1
GROUP BY tr.train_id, tr.number, tr.name
HAVING
    COUNT(tk.ticket_id)::float
    /
    NULLIF(COUNT(s.seat_id), 0)
    < 0.5;

```

## Представления:

### 1) free\_seats\_view

```
CREATE OR REPLACE VIEW free_seats_view AS
SELECT
    tp.trip_id,
    tr.number          AS train_number,
    tr.name            AS train_name,
    c.carriage_id,
    c.carriage_number,
    c.carriage_type,
    s.seat_id,
    s.seat_number,
    s.seat_type
FROM trip tp
JOIN train tr
    ON tp.train_id = tr.train_id
JOIN train_composition tc
    ON tp.trip_id = tc.trip_id
JOIN carriage c
    ON tc.carriage_id = c.carriage_id
JOIN seat s
    ON tc.train_composition_id = s.train_composition_id
WHERE s.status = 'Свободно'
;
```

### 2) unsold\_tickets\_yesterday

```
CREATE OR REPLACE VIEW unsold_tickets_yesterday AS
SELECT
    tr.number          AS train_number,
    c.carriage_type,
    COUNT(s.seat_id)
        - COUNT(tk.ticket_id)    AS unsold_tickets
FROM trip tp
JOIN train tr
    ON tp.train_id = tr.train_id
JOIN train_composition tc
    ON tp.trip_id = tc.trip_id
JOIN carriage c
    ON tc.carriage_id = c.carriage_id
JOIN seat s
    ON tc.train_composition_id = s.train_composition_id
LEFT JOIN ticket tk
    ON tk.trip_id = tp.trip_id
AND tk.seat_id = s.seat_id
AND tk.status = 'Куплен'
```

```
WHERE tp.departure_date = CURRENT_DATE - 1
GROUP BY tr.number, c.carriage_type
;
```

Проверка:

```
SELECT *
FROM free_seats_view
WHERE trip_id = 1;
```

Создание запросов на модификацию данных с подзапросами:

До:

	ticket_id [PK] integer	price numeric (10, 2)	sale_date date	sale_type character varying (50)	status character	trip_id integer	station_departure_id integer	station_arrival_id integer	seat_id integer	purchase_method_id integer	passport_id integer	ticket_office_id integer
1	1	2000.00	2025-05-11	Онлайн	Куплен	1	2	1	1	1	1	1
2	2	2100.00	2025-05-12	Обычная продажа	Куплен	2	3	1	3	2	2	2
3	3	1900.00	2025-05-12	Предварительная	Куплен	2	3	1	4	2	3	2
4	4	3000.00	2025-05-12	Онлайн	Куплен	3	4	1	7	1	4	3
5	5	2800.00	2025-05-12	Онлайн	Куплен	5	5	3	9	1	5	4

```
INSERT INTO ticket (
    ticket_id, price, sale_date, sale_type, status,
    trip_id, station_departure_id, station_arrival_id,
    seat_id, purchase_method_id, passport_id, ticket_office_id
)
SELECT
    (SELECT COALESCE(MAX(ticket_id), 0) FROM ticket) + ROW_NUMBER() OVER () AS
ticket_id,
    2500.00 AS price,
    '2025-05-13' AS sale_date,
    'Онлайн' AS sale_type,
    'Куплен' AS status,
    3 AS trip_id,
    2 AS station_departure_id,
    1 AS station_arrival_id,
    s.seat_id,
    1 AS purchase_method_id,
    1 AS passport_id,
    1 AS ticket_office_id
FROM seat s
JOIN train_composition tc ON s.train_composition_id = tc.train_composition_id
JOIN carriage c ON tc.carriage_id = c.carriage_id
WHERE tc.trip_id = 3
    AND c.carriage_type = 'CB'
    AND s.status = 'Свободно';
```

После:

	ticket_id [PK] integer	price numeric	sale_date date	sale_type character varying (50)	status character va	trip_id integer	station_departure_id integer	station_arrival_id integer	seat_id integer	purchase_method_id integer	passport_id integer	ticket_office_id integer
1	1	2000.00	2025-05-11	Онлайн	Куплен	1	2	1	1	1	1	1
2	2	2100.00	2025-05-12	Обычная продажа	Куплен	2	3	1	3	2	2	2
3	3	1900.00	2025-05-12	Предварительная	Куплен	2	3	1	4	2	3	2
4	4	3000.00	2025-05-12	Онлайн	Куплен	3	4	1	7	1	4	3
5	5	2800.00	2025-05-12	Онлайн	Куплен	5	5	3	9	1	5	4
6	6	2500.00	2025-05-13	Онлайн	Куплен	3	2	1	11	1	1	1
7	7	2500.00	2025-05-13	Онлайн	Куплен	3	2	1	12	1	1	1

```
UPDATE ticket
SET status = 'Возврат'
WHERE ticket_id IN (
    SELECT ticket_id
    FROM ticket
    WHERE price > 2000
        AND status = 'Куплен'
);
```

После:

	ticket_id [PK] integer	price numeric (10,2)	sale_date date	sale_type character varying (50)	status character varying (50)	trip_id integer	station_departure_id integer	station_arrival_id integer	seat_id integer	purchase_method_id integer	passport_id integer	ticket_office_id integer
1	1	2000.00	2025-05-11	Онлайн	Куплен	1	2	1	1	1	1	1
2	3	1900.00	2025-05-12	Предварительная	Куплен	2	3	1	4	2	3	2
3	2	2100.00	2025-05-12	Обычная продажа	Возврат	2	3	1	3	2	2	2
4	4	3000.00	2025-05-12	Онлайн	Возврат	3	4	1	7	1	4	3
5	6	2500.00	2025-05-13	Онлайн	Возврат	3	2	1	11	1	1	1
6	7	2500.00	2025-05-13	Онлайн	Возврат	3	2	1	12	1	1	1
7	5	2800.00	2025-05-12	Онлайн	Возврат	5	5	3	9	1	5	4

```
DELETE FROM ticket
WHERE status = 'Возврат';
```

После:

	ticket_id [PK] integer	price numeric (10,2)	sale_date date	sale_type character varying (50)	status character varying (50)	trip_id integer	station_departure_id integer	station_arrival_id integer	seat_id integer	purchase_method_id integer	passport_id integer	ticket_office_id integer
1	1	2000.00	2025-05-11	Онлайн	Куплен	1	2	1	1	1	1	1
2	3	1900.00	2025-05-12	Предварительная	Куплен	2	3	1	4	2	3	2

Работа с индексами:

Удаляем старые индексы:

```
DROP INDEX IF EXISTS idx_trip_departure_date;
DROP INDEX IF EXISTS idx_route_departure_station;
DROP INDEX IF EXISTS idx_seat_train_composition;
DROP INDEX IF EXISTS idx_ticket_trip_seat_status;
```

Замер без индексов:

```
EXPLAIN (ANALYZE, BUFFERS, FORMAT TEXT)
SELECT
    tr.train_id,
    tr.number          AS train_number,
    tr.name            AS train_name,
```



```

COUNT(s.seat_id)          AS total_seats,
COUNT(tk.ticket_id)       AS sold_seats,
COUNT(s.seat_id) - COUNT(tk.ticket_id) AS free_seats
FROM trip tp
JOIN route r ON tp.route_id = r.route_id
JOIN train tr ON tp.train_id = tr.train_id
JOIN train_composition tc ON tp.trip_id = tc.trip_id
JOIN seat s ON tc.train_composition_id = s.train_composition_id
LEFT JOIN ticket tk
  ON tk.trip_id = tp.trip_id
 AND tk.seat_id = s.seat_id
 AND tk.status = 'Куплен'
WHERE r.departure_station_id = 2
  AND tp.departure_date = CURRENT_DATE + 1
GROUP BY tr.train_id, tr.number, tr.name;

```

Planning:

Buffers: shared hit=302

Planning Time: 2.527 ms

Execution Time: 0.515 ms

## Создадим индексы:

```

CREATE INDEX idx_trip_departure_date ON trip(departure_date);
CREATE INDEX idx_route_departure_station ON route(departure_station_id);

CREATE INDEX idx_seat_train_composition ON seat(train_composition_id);
CREATE INDEX idx_ticket_trip_seat_status
  ON ticket(trip_id, seat_id, status)
WHERE status = 'Куплен';

```

## Замер с индексами:

```

EXPLAIN (ANALYZE, BUFFERS, FORMAT TEXT)
SELECT
  tr.train_id,
  tr.number      AS train_number,
  tr.name        AS train_name,
  COUNT(s.seat_id)          AS total_seats,
  COUNT(tk.ticket_id)       AS sold_seats,
  COUNT(s.seat_id) - COUNT(tk.ticket_id) AS free_seats
FROM trip tp
JOIN route r ON tp.route_id = r.route_id
JOIN train tr ON tp.train_id = tr.train_id
JOIN train_composition tc ON tp.trip_id = tc.trip_id
JOIN seat s ON tc.train_composition_id = s.train_composition_id

```

```
LEFT JOIN ticket tk
  ON tk.trip_id = tp.trip_id
AND tk.seat_id = s.seat_id
AND tk.status = 'Куплен'
WHERE r.departure_station_id = 2
AND tp.departure_date = CURRENT_DATE + 1
GROUP BY tr.train_id, tr.number, tr.name;
```

Planning:

Buffers: shared hit=320 read=4

Planning Time: 2.491 ms

Execution Time: 0.305 ms

**Выводы:** В ходе лабораторной работы была спроектирована и наполнена тестовая БД с учётом CHECK/FOREIGN KEY, разработаны и оптимизированы сложные SELECT-запросы, реализованы DML-операции с подзапросами, созданы удобные представления и подтверждена значительная выгода от индексирования.