

**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**

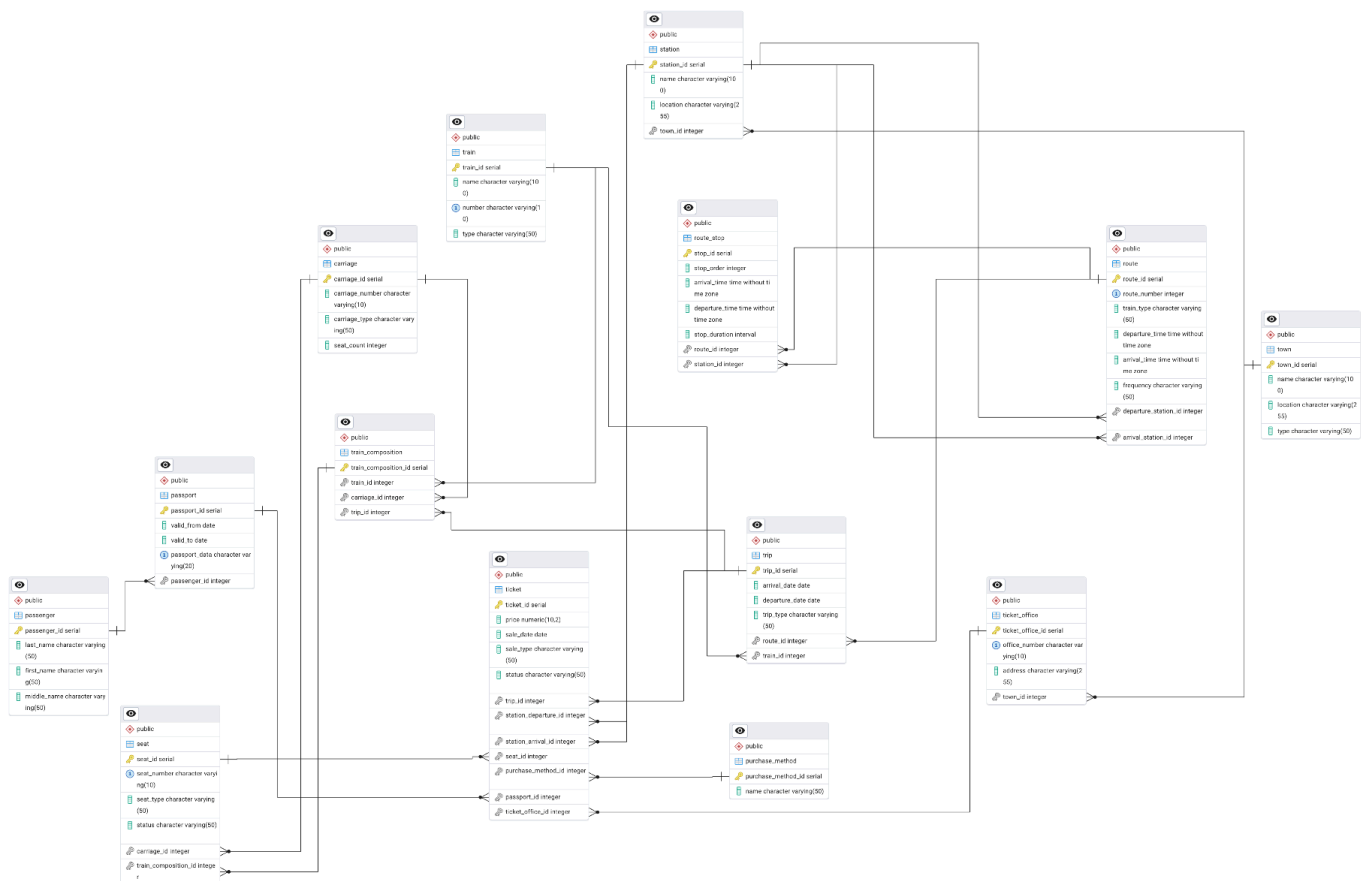
**Преподаватель Говорова Марина Михайловна**

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. **Запросы:**

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

```
SELECT r.route_number, s.seat_number, c.carriage_number, r.departure_time, r.arrival_time
FROM seat s
JOIN train_composition t_c ON s.train_composition_id = t_c.train_composition_id
JOIN trip tr ON t_c.trip_id = tr.trip_id
JOIN route r ON tr.route_id = r.route_id
JOIN carriage c ON s.carriage_id = c.carriage_id
WHERE s.status = 'Свободно' AND arrival_station_id = 1 AND tr.departure_date =
(CURRENT_DATE + 1)
```

-- (места со статусом «Свободно», рейсы departure\_date = current\_date+1)

	route_number integer	seat_number character varying (10)	carriage_number character varying (10)	departure_time time without time zone	arrival_time time without time zone
1	101	01	10	08:00:00	12:00:00
2	101	02	10	08:00:00	12:00:00
3	101	03	10	08:00:00	12:00:00
4	101	04	10	08:00:00	12:00:00
5	101	05	10	08:00:00	12:00:00
6	101	06	10	08:00:00	12:00:00
7	101	07	10	08:00:00	12:00:00
8	101	08	10	08:00:00	12:00:00
9	101	09	10	08:00:00	12:00:00
10	101	10	10	08:00:00	12:00:00

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

```
SELECT train_id, total
FROM (
  SELECT tr.train_id, SUM(t.price) AS total
  FROM ticket t
  JOIN trip tr ON t.trip_id = tr.trip_id
  WHERE tr.departure_date = CURRENT_DATE - 1
  GROUP BY tr.train_id
) AS revenue_by_train
ORDER BY total DESC
LIMIT 1;
```

-- поезд-лидер по выручке за вчера: суммируем price, сортируем по убыванию, берем 1 строку

	train_id integer	total numeric
1	6	9300.00

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

```

SELECT tr.number
FROM trip tp
JOIN train tr ON tr.train_id = tp.train_id
JOIN train_composition tc ON tc.trip_id = tp.trip_id
JOIN seat s ON s.train_composition_id = tc.train_composition_id
LEFT JOIN ticket t ON t.trip_id = tp.trip_id AND t.seat_id = s.seat_id AND t.status = 'Куплен'
WHERE tp.departure_date = CURRENT_DATE + 1
GROUP BY tp.trip_id, tr.number
HAVING COUNT(t.ticket_id) = COUNT(s.seat_id);

```

-- номера поездов, где завтра продано 100 % мест (count(куплено)=count(мест))

	number character varying (10) 🔒
1	770A

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

```

SELECT tr.number,c.carriage_number,s.seat_number,r.departure_time,r.arrival_time
FROM trip tp
JOIN route r ON r.route_id=tp.route_id
JOIN train tr ON tr.train_id=tp.train_id
JOIN train_composition tc ON tc.trip_id=tp.trip_id
JOIN carriage c ON c.carriage_id=tc.carriage_id AND c.carriage_type = 'Купе'
JOIN seat s ON s.train_composition_id=tc.train_composition_id AND s.status = 'Свободно'
WHERE r.arrival_station_id=1 AND tp.departure_date=CURRENT_DATE
ORDER BY tr.number,c.carriage_number,s.seat_number;

```

-- свободные купейные места до Москвы сегодня

	number character varying (10) 🔒	carriage_number character varying (10) 🔒	seat_number character varying (10) 🔒	departure_time time without time zone 🔒	arrival_time time without time zone 🔒
1	002Й	04	01	10:00:00	23:00:00
2	002Й	04	02	10:00:00	23:00:00
3	002Й	04	03	10:00:00	23:00:00
4	002Й	04	04	10:00:00	23:00:00
5	002Й	04	05	10:00:00	23:00:00
6	002Й	04	06	10:00:00	23:00:00
7	002Й	04	07	10:00:00	23:00:00
8	002Й	04	08	10:00:00	23:00:00
9	002Й	04	09	10:00:00	23:00:00
10	002Й	04	10	10:00:00	23:00:00
11	500В	09	01	14:30:00	22:00:00
12	500В	09	02	14:30:00	22:00:00
13	500В	09	03	14:30:00	22:00:00
14	500В	09	04	14:30:00	22:00:00
15	500В	09	05	14:30:00	22:00:00

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

```
SELECT SUM(price) FROM ticket
WHERE sale_date = CURRENT_DATE - 1 AND status='Куплен'
```

-- общая выручка по купленным билетам, продажи вчера

	sum numeric
1	19700.00

6. Общее количество билетов, проданных по всем направлениям в вагоны типа «CB».

```
SELECT COUNT(t.ticket_id)
FROM ticket t
JOIN seat s ON s.seat_id=t.seat_id
JOIN train_composition tc ON tc.train_composition_id=s.train_composition_id
JOIN carriage c ON c.carriage_id=tc.carriage_id
WHERE t.status='Куплен' AND c.carriage_type='CB';
```

-- сколько купленных билетов в вагонах типа «CB»

	count bigint
1	5

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

```
SELECT tr.number, tr.name
FROM trip tp
JOIN train tr ON tr.train_id=tp.train_id
JOIN train_composition tc ON tc.trip_id=tp.trip_id
JOIN seat s ON s.train_composition_id=tc.train_composition_id
LEFT JOIN ticket t ON t.trip_id=tp.trip_id AND t.seat_id=s.seat_id AND t.status='Куплен'
WHERE tp.departure_date=CURRENT_DATE-1
GROUP BY tr.train_id, tr.number, tr.name
HAVING COUNT(t.ticket_id)*2<COUNT(s.seat_id);
# если купленных мест вдвое меньше чем всех
```

-- поезда, которые вчера были заполнены < 50 %

	number character varying (10)	name character varying (100)
1	770A	Сапсан
2	100M	Сибирь
3	150Y	Урал
4	300A	Восток

#### 4. Представления:

- 1) Представление для пассажиров о наличии свободных мест на заданный рейс;

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

– Пример использования SELECT \* FROM available\_seats WHERE trip\_id=2;

	trip_id integer	number character varying (10)	carriage_number character varying (10)	carriage_type character varying (50)	seat_number character varying (10)	seat_type character varying (50)
1	2	821Ч	03	CB	01	Нижнее
2	2	821Ч	03	CB	02	Верхнее
3	2	821Ч	03	CB	03	Нижнее
4	2	821Ч	03	CB	04	Верхнее
5	2	821Ч	03	CB	05	Нижнее
6	2	821Ч	03	CB	06	Верхнее
7	2	821Ч	03	CB	07	Нижнее
8	2	821Ч	03	CB	08	Верхнее
9	2	821Ч	03	CB	09	Нижнее
10	2	821Ч	03	CB	10	Верхнее

- 2) Представления для нахождения количества непроданных билетов на все поезда, формирующиеся за прошедшие сутки (номер поезда, тип вагона, количество).

```
CREATE OR REPLACE VIEW unsold_prev_day AS
SELECT tr.number,c.carriage_type,COUNT(*) AS free_seats
FROM seat s
JOIN train_composition tc ON tc.train_composition_id=s.train_composition_id
JOIN carriage c ON c.carriage_id=tc.carriage_id
JOIN trip tp ON tp.trip_id=tc.trip_id
JOIN train tr ON tr.train_id=tp.train_id
LEFT JOIN ticket t ON t.trip_id=tp.trip_id AND t.seat_id=s.seat_id AND t.status='Куплен'
WHERE tp.departure_date=CURRENT_DATE-1 AND t.ticket_id IS NULL
GROUP BY tr.number,c.carriage_type;
```

-- Пример использования SELECT \* FROM unsold\_prev\_day;

	number character varying (10) 🔒	carriage_type character varying (50) 🔒	free_seats bigint 🔒
1	100M	CB	10
2	150Y	Купе	10
3	300A	CB	10
4	770A	Купе	7
5	770A	Плацкарт	10

- Первое представление показывает все свободные места на рейс: выборка из seat, соединенная с вагонами, рейсом и поездом; берём только строки, где status='Свободно'. Пользователь затем фильтрует по нужному trip\_id.
- Второе представление считает непроданные места за вчера: для рейсов вчерашней даты оставляем места, к которым нет билета со статусом «Куплен», группируем по номеру поезда и типу вагона и выводим их количество.

##### 5. Создание запросов на модификацию данных с подзапросами:

```
INSERT INTO ticket(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 2000,current_date,'Онлайн','Куплен',3,
       r.departure_station_id,r.arrival_station_id,
       s.seat_id,1,1,1
FROM seat s
JOIN train_composition tc ON tc.train_composition_id=s.train_composition_id
JOIN trip tp ON tp.trip_id=tc.trip_id
JOIN route r ON r.route_id=tp.route_id
LEFT JOIN ticket t ON t.seat_id=s.seat_id AND t.status='Куплен'
WHERE tp.trip_id=3 AND t.ticket_id IS NULL;
```

Суть: добавляет билеты «Куплен» на все еще свободные места рейса 3, делая поезд полностью распроданным.

До:

Showing rows: 1 to 35 Page No: 1 of 1										
	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
1	1	1500.00	2025-05-16	Онлайн	Куплен	1	2	1	1	
2	2	1600.00	2025-05-16	Обычная продажа	Куплен	1	2	1	2	
3	3	1700.00	2025-05-16	Предварительная	Куплен	1	2	1	3	
4	4	1800.00	2025-05-17	Онлайн	Куплен	2	2	1	4	
5	5	1900.00	2025-05-17	Обычная продажа	Куплен	2	2	1	5	
6	6	2000.00	2025-05-17	Предварительная	Куплен	2	2	1	6	
7	7	2100.00	2025-05-18	Онлайн	Куплен	3	2	1	7	
8	8	2200.00	2025-05-18	Обычная продажа	Куплен	3	2	1	8	
9	9	2300.00	2025-05-18	Предварительная	Куплен	3	2	1	9	
10	10	2400.00	2025-05-16	Онлайн	Куплен	4	2	1	10	
11	11	2500.00	2025-05-16	Обычная продажа	Куплен	4	2	1	11	
12	12	2600.00	2025-05-17	Предварительная	Куплен	4	2	1	12	
13	13	2700.00	2025-05-17	Онлайн	Куплен	5	2	1	13	
14	14	2800.00	2025-05-17	Обычная продажа	Куплен	5	2	1	14	
15	15	2900.00	2025-05-18	Предварительная	Куплен	5	2	1	15	
Total rows: 35 Query complete 00:00:00.048 LF Ln 1, Col 21										

После:



Showing rows: 1 to 45 Page No: 1 of 1										
	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	price integer
31	31	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	96	
32	32	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	97	
33	33	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	98	
34	34	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	99	
35	35	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	100	
36	36	2000.00	2025-05-17	Онлайн	Куплен	3	4	1	39	
37	37	2000.00	2025-05-17	Онлайн	Куплен	3	4	1	33	
38	38	2000.00	2025-05-17	Онлайн	Куплен	3	4	1	31	
39	39	2000.00	2025-05-17	Онлайн	Куплен	3	4	1	34	
40	40	2000.00	2025-05-17	Онлайн	Куплен	3	4	1	40	
41	41	2000.00	2025-05-17	Онлайн	Куплен	3	4	1	37	
42	42	2000.00	2025-05-17	Онлайн	Куплен	3	4	1	32	
43	43	2000.00	2025-05-17	Онлайн	Куплен	3	4	1	38	
44	44	2000.00	2025-05-17	Онлайн	Куплен	3	4	1	36	
45	45	2000.00	2025-05-17	Онлайн	Куплен	3	4	1	35	

Total rows: 45 Query complete 00:00:00.065 LF Ln 1, Col 21

```
UPDATE ticket
SET status = 'Возврат'
WHERE trip_id = 3
AND status = 'Куплен';
```

```
UPDATE seat
SET status = 'Свободно'
WHERE seat_id IN (
  SELECT seat_id
  FROM ticket
  WHERE trip_id = 3
  AND status = 'Возврат'
);
```

Суть: переводит билеты этого рейса в статус «Возврат» и снова отмечает соответствующие места как свободные.

После:

32	35	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	100	
33	7	2100.00	2025-05-18	Онлайн	Возврат	3	2	1	7	
34	8	2200.00	2025-05-18	Обычная продажа	Возврат	3	2	1	8	
35	9	2300.00	2025-05-18	Предварительная	Возврат	3	2	1	9	
36	36	2000.00	2025-05-17	Онлайн	Возврат	3	4	1	39	
37	37	2000.00	2025-05-17	Онлайн	Возврат	3	4	1	33	
38	38	2000.00	2025-05-17	Онлайн	Возврат	3	4	1	31	
39	39	2000.00	2025-05-17	Онлайн	Возврат	3	4	1	34	
40	40	2000.00	2025-05-17	Онлайн	Возврат	3	4	1	40	
41	41	2000.00	2025-05-17	Онлайн	Возврат	3	4	1	37	
42	42	2000.00	2025-05-17	Онлайн	Возврат	3	4	1	32	
43	43	2000.00	2025-05-17	Онлайн	Возврат	3	4	1	38	
44	44	2000.00	2025-05-17	Онлайн	Возврат	3	4	1	36	
45	45	2000.00	2025-05-17	Онлайн	Возврат	3	4	1	35	

Total rows: 45 Query complete 00:00:00.034 LF Ln 1, Col 21

```
DELETE FROM ticket
WHERE ticket_id IN (
    SELECT ticket_id
    FROM ticket
    WHERE trip_id = 3
    AND status = 'Возврат'
);
```

Суть: удаляет из базы все билеты со статусом «Возврат» для рейса 3.

После:

Showing rows: 1 to 32 Page No: 1 of 1										
	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	pur int
18	21	3500.00	2025-05-18	Предварительная	Куплен	7	2	1	21	
19	22	3600.00	2025-05-18	Онлайн	Куплен	8	2	1	22	
20	23	3700.00	2025-05-16	Обычная продажа	Куплен	8	2	1	23	
21	24	3800.00	2025-05-17	Предварительная	Куплен	8	2	1	24	
22	25	3900.00	2025-05-17	Онлайн	Куплен	8	2	1	25	
23	26	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	91	
24	27	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	92	
25	28	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	93	
26	29	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	94	
27	30	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	95	
28	31	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	96	
29	32	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	97	
30	33	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	98	
31	34	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	99	
32	35	2500.00	2025-05-17	Онлайн	Куплен	10	2	1	100	

Total rows: 32 Query complete 00:00:00.060 LF Ln 1, Col 21

## 6. Работа с индексами:

1. Заполняем TICKET большим объёмом данных:

```
INSERT INTO ticket(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 (RANDOM()*4000+1000)::NUMERIC(10,2),
       DATE '2025-05-17' + (RANDOM()*30)::INT,
       'Онлайн','Куплен',
       1,2,1,1,1,1,1
FROM GENERATE_SERIES(1,1000000);
```

2. Посчитаем время БЕЗ индексов:

```
EXPLAIN ANALYZE
SELECT COUNT(*),SUM(price)
FROM ticket
WHERE sale_date='2025-05-17';
```

```
EXPLAIN ANALYZE
SELECT COUNT(*),SUM(price)
FROM ticket
WHERE sale_date='2025-05-17' AND status='Куплен';
```

Planning Time: 0.053 ms

Execution Time: 23.664 ms

3. Создаём два простых индекса:

```
CREATE INDEX idx_ticket_date  ON ticket(sale_date);
CREATE INDEX idx_ticket_status ON ticket(status);
```

4. Время с простыми индексами

Первый запуск был ~73 ms, второй:

Planning Time: 0.072 ms

Execution Time: 13.212 ms

5. Создаем составной индекс, один простой убираем:

```
DROP INDEX idx_ticket_status;  
CREATE INDEX idx_ticket_date_status ON ticket(sale_date,status);
```

6. Время с составным индексом

Planning Time: 0.356 ms

Execution Time: 18.692 ms

7. Удалим индексы

```
DROP INDEX idx_ticket_date_status;  
DROP INDEX idx_ticket_date;
```

8. Вывод по работе с индексами: простой индекс ускоряет фильтр по одной колонке; составной дает максимум пользы, когда запрос использует обе колонки одновременно.

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