

## ОПТИМИЗАЦИЯ SQL-ЗАПРОСОВ: ИНДЕКСЫ, АНАЛИЗ ПЛАНОВ ВЫПОЛНЕНИЯ И ДЕНОРМАЛИЗАЦИЯ

КОЛОБОВА Дарья Алексеевна

ДРАГУН Виктория Александровна

Научный руководитель: КОБЫЛКИН Дмитрий Сергеевич

кандидат технических наук, доцент

Оренбургский государственный университет

г. Оренбург, Россия

*Целью данной статьи является анализ методов оптимизации SQL-запросов, включая применение индексов, их влияние на производительность запросов, денормализацию данных и анализ планов выполнения с помощью команды EXPLAIN.*

**Ключевые слова:** PostgreSQL, оптимизация запросов, индексы, EXPLAIN, денормализация, производительность БД.

**В** условиях роста объемов данных и увеличения нагрузки на базы данных эффективность выполнения SQL-запросов становится критически важной. Оптимизация SQL-запросов представляет собой комплекс мер, направленных на повышение эффективности выполнения запросов к базе данных. Она позволяет сократить время отклика, уменьшить нагрузку на сервер и обеспечить стабильную работу приложения даже при больших объемах данных [3].

СУБД PostgreSQL предлагает широкий набор инструментов для анализа и улучшения производительности запросов. Основные методы оптимизации SQL-запросов включают использование индексов, анализ планов выполнения, денормализацию данных и пересмотр структуры запросов.

Индексы – это специальные структуры данных, которые ускоряют поиск информации в таблицах [2]. При выполнении запроса с условием WHERE СУБД может использовать индекс для быстрого нахождения нуж-

ных строк без полного сканирования таблицы (Seq Scan). Однако индексы требуют дополнительного дискового пространства и могут замедлять операции вставки и обновления данных. Оптимальная стратегия предполагает создание индексов только для часто фильтруемых полей, чтобы минимизировать накладные расходы.

Команда EXPLAIN позволяет увидеть план выполнения запроса, что помогает выявить узкие места в производительности [4].

Основные операции: Seq Scan (полное сканирование таблицы), Index Scan (сканирование с использованием индекса), Bitmap Index Scan (комбинированное использование индексов).

EXPLAIN ANALYZE дополнительно выполняет запрос и предоставляет реальные метрики времени и количества обработанных строк.

Для проведения комплексного исследования была создана база данных с таблицей employees, содержащей 1 миллион записей (рисунок 1).

	id [PK] integer	name character varying (100)	department character varying (50)	salary integer
1	1	Employee_1	IT	6603
2	2	Employee_2	Sales	9674
3	3	Employee_3	Marketing	7873
4	4	Employee_4	Finance	8583
5	5	Employee_5	HR	9941
6	6	Employee_6	IT	347
7	7	Employee_7	Sales	474
8	8	Employee_8	Marketing	1296
9	9	Employee_9	Finance	6664
10	10	Employee_10	HR	8269
11	11	Employee_11	IT	202
12	12	Employee_12	Sales	3177
13	13	Employee_13	Marketing	5799

Рисунок 1. Фрагмент записей таблицы employees с демонстрацией структуры и значений

Выполним запрос поиска сотрудников отдела IT без индекса:  
**EXPLAIN ANALYZE**  
**SELECT \* FROM employees WHERE depart-**

**ment = 'IT';**  
 Результат выполнения запроса изображен на рисунке 2.

	QUERY PLAN text
1	Seq Scan on employees (cost=0.00..20074.00 rows=203433 width=28) (actual time=0.011..120.044 rows=200000 loops...
2	Filter: ((department)::text = 'IT)::text)
3	Rows Removed by Filter: 800000
4	Planning Time: 0.285 ms
5	Execution Time: 128.509 ms

Рисунок 2. Результат выполнения запроса без индекса

Создадим индекс и повторим запрос (рисунок 3):  
**CREATE INDEX idx\_department ON employees(department);**

**EXPLAIN ANALYZE**  
**SELECT \* FROM employees WHERE depart-**  
**ment = 'IT';**

	QUERY PLAN text
1	Bitmap Heap Scan on employees (cost=2273.03..12389.94 rows=203433 width=28) (actual time=7.172..26.012 rows=200000 loops...
2	Recheck Cond: ((department)::text = 'IT'::text)
3	Heap Blocks: exact=7574
4	-> Bitmap Index Scan on idx_department (cost=0.00..2222.17 rows=203433 width=0) (actual time=6.053..6.053 rows=200000 loop...
5	Index Cond: ((department)::text = 'IT'::text)
6	Planning Time: 0.175 ms
7	Execution Time: 31.195 ms

Рисунок 3. Результат выполнения запроса с индексом

В таблице 1 приведено сравнение методов выполнения запросов.

Таблица 1

**СРАВНЕНИЕ МЕТОДОВ ВЫПОЛНЕНИЯ ЗАПРОСА**

Метод	Тип сканирования	Время выполнения	Кол-во обработанных строк
Без индекса	Seq Scan	128.509 мс	1 000 000
С индексом	Bitmap Index Scan	31.195 мс	200 000

Использование индексов позволило сократить время выполнения запросов до 4 раз за счет уменьшения количества обрабатываемых строк.

Денормализация – это намеренное дублирование данных для ускорения запросов в ущерб нормальной форме [1]. Этот подход жертвует идеальной структурой в пользу скорости чтения, что особенно важно для аналитических систем и высоконагруженных приложений.

Таким образом, для максимальной эффективности PostgreSQL важно грамотно сочетать индексацию, анализ запросов и умеренную денормализацию. Оптимальная производительность достигается, когда ускорение чтения не приводит к критическому замедлению операций обновления. Рациональное применение этих методов позволяет поддерживать высокую скорость работы даже в ресурсоемких системах.

**СПИСОК ЛИТЕРАТУРЫ**

1. Бандель Е.А. Методика повышения скорости выполнения запросов в СУБД PostgreSQL / Е.А. Бандель, Н.С. Нестеров // Системный анализ в проектировании и управлении. – 2024. – С. 479-484.
2. Домбровская Г. Оптимизация запросов в PostgreSQL / Г. Домбровская, Б. Новиков, А. Бейликова, пер. с англ. Д.А. Беликова // ДМК Пресс. –2022. – С. 279.
3. Каримов Т.Ш. Оптимизация запросов в реляционных базах данных: подходы и инструменты / Т.Ш. Каримов// Вестник науки. – 2025. – № 4 (85) – С. 780-786.
4. Колобов А.Н. Информационно-образовательные технологии / А. Н. Колобов // Университетский комплекс как региональный центр образования, науки и культуры: сб. материалов Всерос. науч.-метод. конф., – 2023. – С. 1309-1315.

## **OPTIMIZATION OF SQL QUERIES: INDEXES, EXECUTION PLAN ANALYSIS, AND DENORMALIZATION**

**KOLOBOVA Darya Alekseevna**

**DRAGUN Victoriya Alexandrovna**

*Scientific Supervisor:* **KOBYLKIN Dmitry Sergeevich**  
Candidate of Sciences in Technology, Associate Professor  
Orenburg State University  
Orenburg, Russia

---

*The purpose of this article is to analyze SQL query optimization methods, including the use of indexes, their impact on query performance, data denormalization, and execution plan analysis using the EXPLAIN command.*

**Keywords:** PostgreSQL, query optimization, indexes, EXPLAIN, denormalization, database performance.

---