

## ИСПОЛЬЗОВАНИЕ ДИАПАЗОННОГО ТИПА ДАННЫХ ДЛЯ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ БАЗЫ ДАННЫХ БИЛЛИНГОВОЙ СИСТЕМЫ

Лысенко А.В. (ИТМО), Войтюк Т.Е.(ИТМО)

Научный руководитель – кандидат технических наук, доцент Войтюк Т.Е. (ИТМО)

**Введение.** Необходимость хранения и обработки данных - одна из ключевых особенностей в эпоху цифровизации. Цена информации возросла, а скорость доступа к ней стал решающим фактором [1-2]. Производительность ресурсоёмких биллинговых систем, обслуживающих миллионы абонентов и агентов, напрямую зависит от скорости работы используемой системы управления базами данных (СУБД) и физической модели данных. Обслуживание множества префиксных зон, диапазонов и масок номеров, хранение данных об изменениях в тарифных планах в биллинговых системах обеспечивается множеством таблиц, содержащих диапазоны значений (начальный и конечный, минимальный и максимальный). В данной работе рассматриваются ведущие СУБД на предмет наличия встроенных диапазонных типов данных, и их влияние на повышение производительности базы данных (БД).

**Основная часть.** Анализ ведущих реляционных СУБД на предмет наличия встроенных типов, поддерживающих хранение диапазона значений, позволил определить, что их реализация существует только в PostgreSQL. СУБД для последующего исследования были определены из числа совместимых с PostgreSQL на уровне протокола (дистрибутивов) [3-6]. На данный момент распространение проприетарного программного обеспечения (ПО) регулируется политикой экспортных ограничений, что является негативным фактором при выборе таких продуктов в качестве СУБД для бизнеса. Для минимизации рисков к рассмотрению приняты отечественные дистрибутивы PostgreSQL, развитию которых способствовало импортозамещение ПО, разработанного ушедшими с российского рынка компаниями [7-10].

Реализация диапазонных типов включает ряд встроенных диапазонных типов и возможность создания пользовательских диапазонных типов [11]. Диапазон может быть задан как с включением граничных значений, так и с их исключением. Можно задать в качестве границы диапазона значение бесконечности или минус бесконечности.

PostgreSQL поддерживает десяток встроенных булевых и логических операций над диапазонами, включая проверку на входжение элемента в диапазон, расположение диапазонов друг относительно друга, определение пересечений, вычитаний и включений диапазонов [12]. Возможность добавления индексов на поле диапазонного типа, таких как B-дерево, хеш и GiST позволяет оптимизировать построение плана выполнения SQL запроса, что позволяет повысить скорость его выполнения [13].

Тестирование изменения скорости выполнения запросов на выборку, вставку, обновление и удаление полей в базе данных производилось на основании данных крупного мобильного оператора, обслуживаемого отечественной биллинговой компанией. Использование диапазонного типа данных и индекса позволило получить значительное увеличение скорости выполнения запросов по сравнению с физической моделью данных, в рамках которой границы диапазонов значений хранились в двух полях таблицы.

**Выводы.** Концепция диапазонного типа данных имеет реализацию лишь в СУБД PostgreSQL и её дистрибутивах. Ключевыми преимуществами использования диапазонных типов являются: перекладывание ответственности за операции над диапазонами на СУБД, оптимизация плана построения запросов за счет индексов. Результат проведенных экспериментов показал, что использование диапазонного типа данных в комбинации с индексом даёт существенное повышение производительности БД биллинговой системы.

## Список использованных источников:

1. Рейтинг систем управления базами данных DB-engines [Электронный ресурс] – Режим доступа: <https://db-engines.com/en/ranking> (дата обращения: 16.12.2024).
2. Jim Gray. The Transaction Concept: Virtues and Limitations // Tandem TR 81.3 –1981 [Электронный ресурс] – Режим доступа: <https://jimgray.azurewebsites.net/papers/thetransactionconcept.pdf> (дата обращения: 16.12.2024).
3. Документация CockroachDB [Электронный ресурс] – Режим доступа: <https://www.cockroachlabs.com/docs/> (дата обращения: 20.12.2024).
4. Документация TimescaleDB [Электронный ресурс] – Режим доступа: <https://docs.timescale.com/> (дата обращения: 20.12.2024).
5. Документация Azure Database for PostgreSQL [Электронный ресурс] – Режим доступа: <https://azure.microsoft.com/en-us/products/postgresql> (дата обращения: 20.12.2024).
6. Документация Greenplum [Электронный ресурс] – Режим доступа: <https://greenplum.org/> (дата обращения: 20.12.2024).
7. Документация Postgres Pro [Электронный ресурс] – Режим доступа: <https://postgrespro.ru/products/postgrespro> (дата обращения: 20.10.2024).
8. Документация Platform V Pangolin [Электронный ресурс] – Режим доступа: <https://pangolin.sbertech.ru/> (дата обращения: 20.12.2024).
9. Документация Tantor [Электронный ресурс] – Режим доступа: <https://tantorlabs.ru/> (дата обращения: 20.12.2024).
10. Документация Nexign Nord [Электронный ресурс] – Режим доступа: <https://nexign.com/ru/products/nexign-nord> (дата обращения: 20.12.2024).
11. Документация версии PostgreSQL 9.2 [Электронный ресурс] – Режим доступа: <https://www.postgresql.org/docs/9.2/rangetypes.html> (дата обращения: 20.12.2024).
12. Range Types in PostgreSQL and GiST Indexes Increases Speed of MySQL Index Combine Queries) [Электронный ресурс] – Режим доступа: <https://medium.com/dataseries/range-types-in-postgresql-and-gist-indexes-788db23346c5> (дата обращения: 16.12.2024).
13. Поддерживаемые операциями над диапазонными типами данных [Электронный ресурс] – Режим доступа: <https://www.postgresql.org/docs/current/functions-range.html> (дата обращения: 20.12.2024).