МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра математической кибернетики и компьютерных наук

СОЗДАНИЕ ХРАНИЛИЩ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ OLAP-КУБОВ ДЛЯ СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ POSTGRESQL

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Соколовой Дарьи Николаевны

Научный руководитель
зав. каф. техн. прог. , к. ф.-м. н.,
доцент _____ И. А. Батраева

Заведующий кафедрой
к. ф.-м. н., доцент _____ С. В. Миронов

Саратов 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

За недели, месяцы и даже года своей деятельности организации накапливают огромные массивы данных, например, данные о совершенных сделках, банковских транзакциях, биржевые изменения, данные о телефонных звонках пользователей и подключенных услугах. При планировании стратегии деятельности компании необходимо анализировать собранные данные. Но из-за своих объемов простые запросы к базам данных начинают занимать больше времени и требовать мощных технических ресурсов, соответственно требуется оптимизация. Для работы с большим массивом данных существуют два популярных подхода — Online Transaction Processing (OLTP) и Online Analytical Processing (OLAP).

Для анализа продаж будет необходимо использовать подход OLAP. Реализовать его можно на разных базах данных, например, SQL Server, облачные базы данных Amazon Web Services (AWS). Многие базы данных, например, SQL Server, предлагают свои технологии по созданию и развертыванию OLAP-куба — SQL Server Analysis Services (SSAS), облачные базы данных AWS — Amazon Redshift, предоставляя пользователю удобный интерфейс с визуализацией. Но предоставляемый бесплатный функционал ограничен, для использования в коммерческом проекте необходима платная подписка. В то время как PostgreSQL является альтернативой коммерческим базам данных с широким и удобным функционалом, поэтому с точки зрения выгоды при выборе базы данных выигрывает PostgreSQL. Многие российские компании и команды разработчиков используют или переводят свои проекты на СУБД PostgreSQL, который помимо реляционной модели может выступать и как объектная. Из минусов PostgreSQL можно выделить то, что в нем нет встроенной технологии для работы с подходом OLAP, возможно использование платных сервисов или дополнений, которые созданы как отдельные коммерческие проекты (такие как olapcube.com и cdata.com) и не относятся к официальному программному обеспечению, предоставляемым системой управления большими данными. Из этого вытекает гипотеза, что можно реализовать свой OLAP-куб на PostgreSQL с помощью библиотек и средств разработки.

Целью настоящей работы является реализация технологии OLAP на базе данных PostgreSQL.Для выполнения указанной цели были поставлены следующие задачи:

— Изучить материал об OLAP-подходе, концепциях построения и разнице с OLTP;
— Провести исследование среди возможных способов реализации и наметить стратегию выполнения цели;
— Выбрать и подготовить соответствующий набор данных, размером не менее 100 тысяч строк. Создать связанные таблицы и заполнить данными;
— Реализовать OLAP-сервер средствами библиотеки cubes языка Python;
— Провести тестовые запуски при прямых SQL-запросах к базе данных с помощью psycopg2 и OLAP-серверу с помощью библиотеки requests, сравнить полученные результаты и затраченное на выполнение запросов время, сделать вывод о работе созданного сервиса.

Бакалаврская работа состоит из введения, трех глав, заключения, списка использованной литературы и приложения. Объем работы 62 страницы. Список литературы включает в себя 25 источников. Первая глава посвящена описанию технологии Online Analytical Processing (OLAP). Описываются история бизнес-аналитики (BI), OLAP как часть BI, приводится сравнение с технологией OLTP (Online Transaction Processing), концепции построения OLAP-куба. Вторая глава посвящена описанию используемых технологий, актуальности выбора СУБД PostgreSQL, примеров работы с библиотекой языка Python cubes. Третья глава содержит в себе описание процесса реализации куба, подготовки данных из датасета, созданию таблиц и их заполнению, тестированию созданного куба.

1 Описание технологии Online Analytical Processing (OLAP)

1.1 Введение в бизнес-аналитику

Online Analytical Processing (OLAP) и Business Intelligence (BI) — две тесно связанные концепции. OLAP — это технология, которая позволяет пользователям быстро и интерактивно анализировать большие объемы данных с разных точек зрения [1]. С другой стороны, бизнес-аналитика (BI) — это общий термин, который относится к процессам, технологиям и инструментам, используемым для преобразования необработанных данных в значимые идеи, которые могут использоваться для принятия бизнес-решений. BI обычно включает в себя несколько этапов, включая извлечение, преобразование и загрузку данных (ETL — Extract, Transform, Load, дословно «извлечение, преобразование, загрузка»), хранение данных, анализ данных и создание отчетов. OLAP используется на этапах анализа данных и составления отчетов, где он позволяет пользователям выполнять специальный анализ, создавать отчеты и визуализировать данные осмысленным и простым для понимания способом [2].

OLAP позволяет делать сложные запросы и анализировать данные в режиме реального времени без необходимости предварительной агрегации данных или сложных SQL-запросов. OLAP-базы данных хранят данные в формате многомерного куба, что позволяет быстро получать доступ к данным и эффективно обрабатывать запросы.

1.2 Разница между OLTP и OLAP

Часто OLAP ошибочно путают с OLTP, но это совершенно разные подходы, имеющие разные применения. OLTP (Online Transaction Processing) представляет собой систему для обработки транзакций в реальном времени, а OLAP (Online Analytical Processing) — систему аналитической обработки данных.

1.3 Концепции построения OLAP-куба

Построение OLAP-куба включает в себя несколько ключевых концепций:

Измерения (Dimensions): Измерения представляют собой атрибуты или характеристики данных, такие как год, месяц, продукт, регион и т.д. Измерения обычно представляются в виде иерархий, которые позволяют пользователям анализировать данные на разных уровнях детализации [**?**].

Факты (Facts): Факты представляют собой числовые значения, которые анализируются в OLAP-кубе, такие как продажи, выручка, количество товаров

и т.д.

Куб (Cube): Куб представляет собой многомерную структуру данных, которая содержит измерения и факты. Куб позволяет пользователям анализировать данные в разных измерениях и на разных уровнях детализации.

Агрегация данных (Data Aggregation): Агрегация данных представляет собой процесс суммирования или сгруппирования данных в OLAP-кубе по измерениям и фактам. Агрегация данных позволяет пользователям анализировать данные в разных уровнях детализации и получать общую картину бизнес-операций.

Распределение (Slicing and Dicing): Распределение позволяет пользователям анализировать данные в OLAP-кубе по различным измерениям и фактам. Пользователи могут срезать (Slicing) данные по определенным измерениям или детализировать (Dicing) данные, чтобы получить более глубокий уровень анализа.

Бурение (Drilling): Бурение позволяет пользователям просматривать данные в OLAP-кубе на более глубоком уровне детализации. Пользователи могут проходить от более крупных уровней данных к более мелким, чтобы получить более детальную информацию [3, 4].

2 Используемые для реализации технологии

Для реализации OLAP-куба для PostgreSQl использовался язык программирования Python версии 3.9, средства jupiter notebook, в том числе библиотека pandas. Для работы с PostgreSQl средствами python была использована библиотека psycopg2, для заполнения таблиц из датафреймов — SQLAlchemy, для создания куба была использована библиотека cubes, к ней был нужен фреймфорк Flask. Запросы к кубу осуществлялись с помощью requests. А для просмотра содержимого PostgreSQl будет использовано приложение DBeawer.

2.1 Актуальность выбора СУБД PostgreSQL и используемых средств

PostgreSQL — это объектно-реляционная система управления базами данных (ORDBMS), которая предоставляет расширенные функциональные возможности по сравнению с другими РСУБД, такими как MySQL и SQLite. PostgreSQL является бесплатной и открытой, что означает, что любой желающий может использовать ее исходный код в своем проекте. PostgreSQL поддерживает полный стандарт SQL, включая оконные функции, группировку множеств, подзапросы и full-text search [5].

psycopg2 — это библиотека для языка Python, которая позволяет взаимодействовать с базами данных PostgreSQL. Она предоставляет удобный и эффективный способ работы с базами данных PostgreSQL из Python-скриптов. psycopg2 полностью поддерживает все функции PostgreSQL, включая транзакции, представления, хранимые процедуры, позволяет работать с двоичными данными поддерживает расширения PostgreSQL, такие как PostGIS, что позволяет работать с геоданными в базах данных PostgreSQL. psycopg2 поддерживает многопоточность, что позволяет работать с базами данных PostgreSQL из нескольких потоков одновременно [5, 6].

DBeawer — бесплатный многоплатформенный инструмент для работы с базами данных для разработчиков, администраторов баз данных, аналитиков и всех, кому необходимо работать с базами данных. Поддерживает все популярные базы данных: MySQL, PostgreSQL, SQLite, Oracle, DB2, SQL Server, Sybase, MS Access, Teradata, Firebird, Apache Hive, Phoenix, Presto и др [7].

SQLAlchemy — это библиотека для работы с базами данных в языке Python. Она предоставляет удобные и гибкие инструменты для работы с различными СУБД, включая PostgreSQL, MySQL, SQLite и другие. SQLAlchemy

предоставляет объектно-реляционную модель (ORM), которая позволяет работать с базами данных в терминах объектов Python. Это упрощает разработку приложений и увеличивает их гибкость [8].

requests — это библиотека Python для отправки HTTP-запросов. Она позволяет выполнить запросы GET, POST, PUT, DELETE и другие, а также работать с заголовками, параметрами и телом запроса. Вы можете использовать requests для взаимодействия с API и получения данных во Flask-приложении.

Flask — это легковесный веб-фреймворк для языка Python, который позволяет быстро создавать веб-приложения. Flask позволяет определять маршруты и функции-обработчики для этих маршрутов, а также работать с шаблонами и формами HTML. Можно использовать Flask для создания веб-приложений и requests для взаимодействия с API [9].

2.2 Библиотека Cubes

Cubes — облегченная платформа Python и HTTP-сервер OLAP для простой разработки приложений для создания отчетов и совокупного просмотра многомерных смоделированных данных, как пишут на официальном сайте библиотеки. Она позволяет строить модели, где реализованы измерения с несколькими иерархиями, ориентированные на пользователя метаданные, шаблоны размеров — определение сложных размеров, возможна локализация модели и данных. Обеспечивает совокупный просмотр в нужных разрезах. Одним из бэкендов, поставляемых с фреймворком, является SQL. Он основан на SQLAlchemy. Cubes требует дополнительной установки фреймворка Flask, для которого в свою очередь будет полезен requests. Данные о созданном сервере хранятся в виде JSON-объекта [10].

3 Реализация OLAP на PostgreSQL

3.1 Подготовка данных

Для наглядной визуализации разницы OLAP и обычных SQL-запросов будет использован открытый датасет с Kaggle «E-Commerce Data». Согласно описанию к датасету, это транснациональный набор данных, который содержит все транзакции, имевшие место в период с 12.01.2010 по 12.09.2011 для британской и зарегистрированной немагазинной онлайн-розничной торговли. Компания в основном продает уникальные подарки на все случаи жизни [11].

Обозначение столбцов следующее:

1. InvoiceNo — номер заказа;
2. StockCode — номер склада;
3. Description — описание купленного объекта;
4. Quantity — количество;
5. InvoiceDate — дата заказа;
6. UnitPrice — цена за единицу товара;
7. CustomerID — идентификатор покупателя;
8. Country — страна покупателя.

В процессе обработки из датасета будут удалены строки, содержащие пустые значения, столбцы будут переименованы в удобный формат, дата будет разбита на три столбца (день, месяц, год). Это будет необходимо для занесения информации в базу данных, а далее для отбора строк для деятельности созданного OLAP-куба.

3.2 Вынесение данных в новые датафреймы

Этот раздел посвящен преобразованию очищенного датафрейма в три датафрейма, соответствующие будущим таблицам sales, products, customers. Sales хранит в себе информацию о совершенных продажах, products является датасетом, содержащим соотношение идентификатора (stock_code) и наименования продаваемой продукции (description), цену за единицу товара (unit_price), customers — идентификатора покупателя (customer) и его страны (country).

3.3 Перенос датафреймов в базу данных

В данном разделе подключаются необходимые библиотеки, описанные в requirements.txt, представляется краткое описание менеджера пакетов Python pip, описывается способ подключения к СУБД PostgreSQL на языке Python

(connection_config.py). Далее с помощью psycopg2 выполняются запросы к базе данных по удалению (если присутствуют), созданию пустых и связанных между собой таблиц sales, products, customers. Далее с помощью sqlalchemy данными из созданных ранее датафреймов pandas заполняются таблицы.

3.4 Модель OLAP-куба

Здесь описан процесс конфигурации модели будущего OLAP-куба. Параметры сервера и подключение к СУБД указаны в slicer.ini. Модель OLAP-куба реализована в виде JSON-объекта, содержащем измерения, кубы, функции агрегации.

3.5 Запуск и тестирование OLAP-куба

Этот раздел посвящен запуску сервера и использованию созданного OLAP-куба. Показаны изображения отображаемого содержимого созданной модели. Также качество и время обращения сравниваются с прямыми запросами к базе данных, созданный куб выдает аналогичные результаты, что и при прямом запросе, а затраченное на обращение к серверу время отстает от прямых запросов к базе данных менее, чем на секунду.

ЗАКЛЮЧЕНИЕ

В результате проведенного исследования на языке программирования Python версии 3.9 и библиотек cubes, flask, requests и фреймворков и средств для работы с базами данных был реализован OLAP-куб, который находится на собственном локальном сервере. По результатами сравнительных запусков с прямыми SQL-запросами в СУБД OLAP-куб выдает идентичные результаты. В сравнении времени выполнения, созданный OLAP-куб выдает ответ на запрос медленнее, но отставание не превышает и двух секунд. В дальнейшем исследовании стоит проверить результаты на еще большем массиве данных (в исследовании использовался обработанный датасет) и более сложных запросах, возможно реализованный подход окажется эффективнее. Гипотезу о том, что можно создать свой OLAP-сервис для проектов на PostgreSQL (где нет встроенной реализации OLAP) можно считать подтвержденной.

Для выполнения цели реализации OLAP-куба на СУБД PostgreSQL были выполнены следующие задачи:

- Изучен материал об OLAP-подходе, концепциях построения и разнице с OLTP;
- Проведено исследование среди возможных способов реализации и наметить стратегию выполнения цели;
- Выбран и подготовлен соответствующий набор данных с Kaggle о продажах за 2010-2011 годы некоторой компании размером 400 тысяч строк. Созданы связанные таблицы sales, products, customers и заполнены данными;
- Реализован OLAP-сервер средствами библиотеки cubes языка Python;
- Проведены тестовые запуски при прямых SQL-запросах к базе данных с помощью psycopg2 и OLAP-серверу с помощью библиотеки requests, сделан вывод о качестве на основе сравнения полученных результатов и затраченного на выполнение запросов время.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Авторов, К.* DAMA-DMBOK. Свод знаний по управлению данными / К. Авторов. — Олимп-Бизнес, 2020.

2 *Сарсимбаева, С.* Преимущества olap технологии в экономическом анализе / С. Сарсимбаева, А. Балтабаева // *Актуальные научные исследования в современном мире*. — 2021. — P. 98–101.

3 *Замятин, А.* Введение в интеллектуальный анализ данных / А. Замятин. — Томский государственный университет, 2022.

4 *Уханов, А.* Система olap / А. Уханов // *Экономика и менеджмент в XXI веке: информационные технологии, биотехнологии, физкультура и спорт*. — 2020. — P. 74–75.

5 *Зыкин, В.* Обновление многотабличных представлений на основе коммутативных преобразований базы данных / В. Зыкин, М. Цымблер // *Вестник Южно-Уральского Государственного Университета. Серия: вычислительная математика и информатика*. — Vol. 8.

6 Psycopg.org [Электронный ресурс]. — URL: https://www.psycopg.org/docs/ (Дата обращения 21.05.2023). Загл. с экр. Яз. рус.

7 *Уилсон, Д.* Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL / Д. Уилсон, Э. Редмонд. — ДМК Пресс, 2022.

8 SQLAlchemy [Электронный ресурс]. — URL: https://www.sqlalchemy.org/ (Дата обращения 21.05.2023). Загл. с экр. Яз. рус.

9 *Джульен, Д.* Путь Python. Черный пояс по разработке, масштабированию, тестированию и развертыванию / Д. Джульен. — «Издательский дом «Питер»», 2019.

10 Pythonhosted.org [Электронный ресурс]. — URL: https://pythonhosted.org/cubes/slicer.html (Дата обращения 21.05.2023). Загл. с экр. Яз. рус.

11 Kaggle.com [Электронный ресурс]. — URL: https://www.kaggle.com/datasets/carrie1/ecommerce-data (Дата обращения 21.05.2023). Загл. с экр. Яз. рус.