УДК 004.651

ОСНОВНЫЕ АРХИТЕКТУРНЫЕ РЕШЕНИЯ ДЛЯ ОРГАНИЗАЦИИ ХРАНЕНИЯ ДАННЫХ СУБД

Кузмин И.А.

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь

Научный руководитель: Пискун Г.А. – канд.техн.наук, доцент, доцент кафедры ПИКС

Аннотация. В статье рассмотрены основные принципы структурирования данных при работе с СУБД. Обозначены основные типы архитектур хранилища данных.

Ключевые слова: *БД, Базы данных, СУБД.*

Введение. База данных представляет собой единую и стандартизированную совокупность данных, которую использует персонал или жители группы, предприятия, региона, страны или мира. Задача базы данных заключается в том, чтобы хранить все данные, которые являются интересными, в одном или нескольких местах таким образом, чтобы не было ненужной избыточности. Создание баз данных направлено на две основные цели: снижение избыточности данных и повышение их надежности.

В работе будут рассмотрены принципы структурирования данных в известных СУБД.

Традиционная архитектура хранилища данных. Трехуровневая архитектура. В традиционной архитектуре хранилища данных часто используется трехуровневая структура.

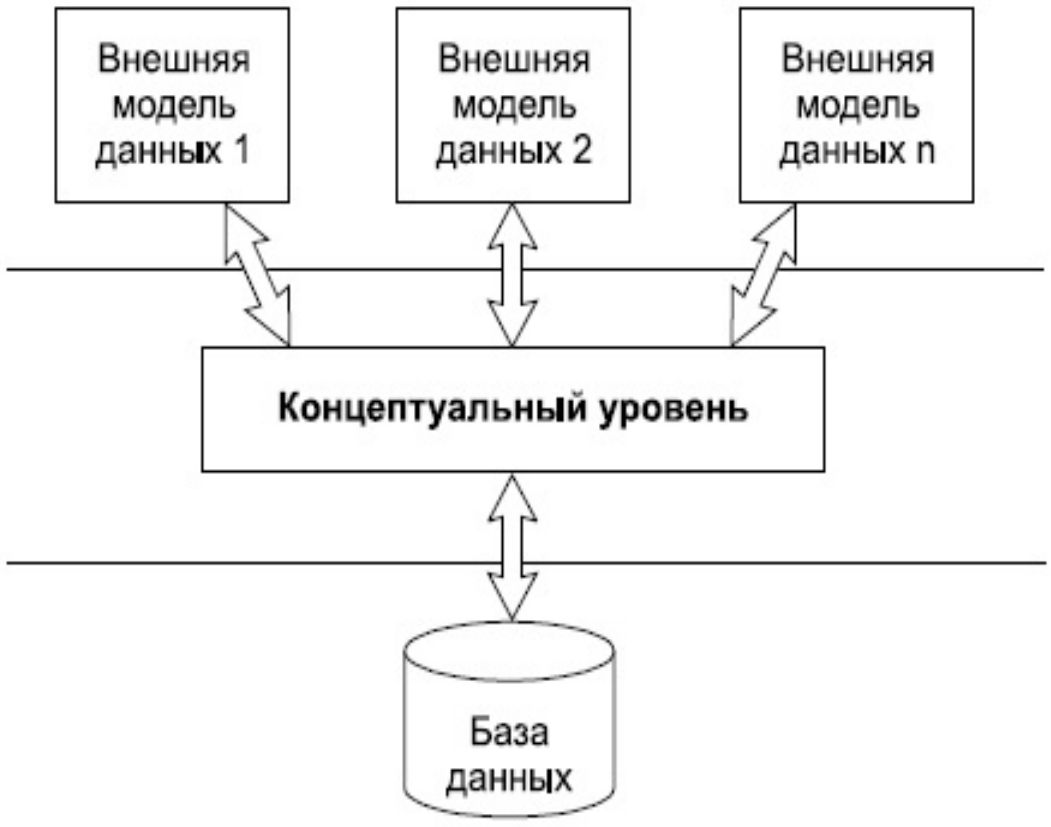

Рисунок 1 – трёхуровневая архитектура БД [1]

Нижний уровень включает сервер базы данных, который извлекает данные из разных источников, таких как транзакционные базы данных, используемые для интерфейсных при-

ложений. Средний уровень содержит сервер OLAP, который преобразует данные в структуру, наиболее подходящую для анализа и сложных запросов. Сервер OLAP может работать двумя способами: в качестве расширенной системы управления реляционными базами данных, которая отображает операции над многомерными данными в стандартные реляционные операции (Relational OLAP), или с использованием многомерной модели OLAP. Верхний уровень – это уровень клиента, содержащий инструменты для высокоуровневого анализа данных, создания отчетов и анализа данных [1].

Билл Инмон и Ральф Кимбалл - два известных эксперта в области хранилищ данных, которые предлагают разные подходы к проектированию. Кимбалл основывается на создании витрин данных, которые представляют отчеты и аналитические данные для конкретных направлений бизнеса, а его проект хранилища данных основан на подходе "снизу-вверх". Хранилище данных – это просто сочетание различных витрин данных, которые облегчают отчетность и анализ. С другой стороны, Инмон предлагает централизованное хранилище данных для всех корпоративных данных, которое начинается с нормализованной модели хранилища данных и использует подход "сверху вниз". При таком подходе организация сначала создает нормализованную модель хранилища данных. Затем создаются витрины размерных данных на основе модели хранилища.

Введение трехуровневой архитектуры баз данных позволило отделить пользовательское представление от физического представления БД. В соответствии с классической теорией баз данных, модели данных состоят из трех основных компонентов: структуры представления данных, операций манипулирования данными и ограничений данных [2-3].

Отделение пользовательского представления БД от ее физического представления необходимо по нескольким причинам. Во-первых, каждый пользователь должен иметь доступ к тем же данным и иметь возможность изменять их представление без влияния на других пользователей. Во-вторых, обращение к БД не должно зависеть от специфических деталей хранения данных. В-третьих, администратор БД должен иметь возможность изменять структуру хранения данных без влияния на пользовательские представления. И, наконец, внутренняя структура хранения данных не должна зависеть от изменения физических устройств хранения информации. Эти причины были подробно описаны в приложениях к теории баз данных.

Приложения ИС и пользовательские представления [2].

Традиционные модели архитектуры хранилища данных. Большинство хранилищ данных полагаются на одну из трех различных моделей:

Традиционные модели архитектуры хранилища данных включают три различных типа хранилищ: виртуальное, витрина и корпоративное. Виртуальное хранилище данных является центром активов данных организации, объединяющим данные из всех направлений бизнеса для общего доступа. Витрина данных, в свою очередь, фокусируется на данных отдельных бизнес-единиц для аналитики и отчетности. Наконец, корпоративное хранилище данных использует распределенный подход для доступа к различным базам данных через единый запрос.

Традиционные локальные хранилища данных могут включать несколько структурных компонентов, таких как пользовательский уровень для анализа данных и интеллектуального анализа данных. Если источники данных содержат те же типы данных, их можно интегрировать в структуру хранилища данных и анализировать через пользовательский уровень.

Для обработки данных, содержащихся в источниках различных структур, форматов и моделей, необходима структура промежуточной области, которая преобразует эти данные в обобщенный формат, позволяющий проводить комплексный анализ на уровне пользователя. Однако, если компания анализирует только данные схожих типов, то использование промежуточной области может быть необязательным. Промежуточная область может поддерживаться с помощью добавления других структур, включая витрины данных. Витрины данных

полезны для хранения сводных данных по конкретным бизнес-направлениям для проведения детального анализа по запросам конкретных команд, например, отдела продаж.

Промежуточная область может поддерживаться добавлением другой структуры, и витрин данных. Витрины данных полезны для хранения сводных данных по конкретному бизнес-направлению для очень конкретных запросов. Например, команды отдела продаж могут получить доступ к этой структуре данных для подробной прогнозной аналитики продаж в разных местах.

Новые архитектуры хранилищ данных. В последние годы хранилища данных переносятся в облако, что приводит к разработке новых архитектур хранилищ данных, которые не придерживаются традиционной модели. Каждое облачное хранилище предлагает свою уникальную архитектуру. В этом разделе будет кратко описаны архитектуры двух наиболее популярных облачных хранилищ данных: Amazon Redshift и Google BigQuery.

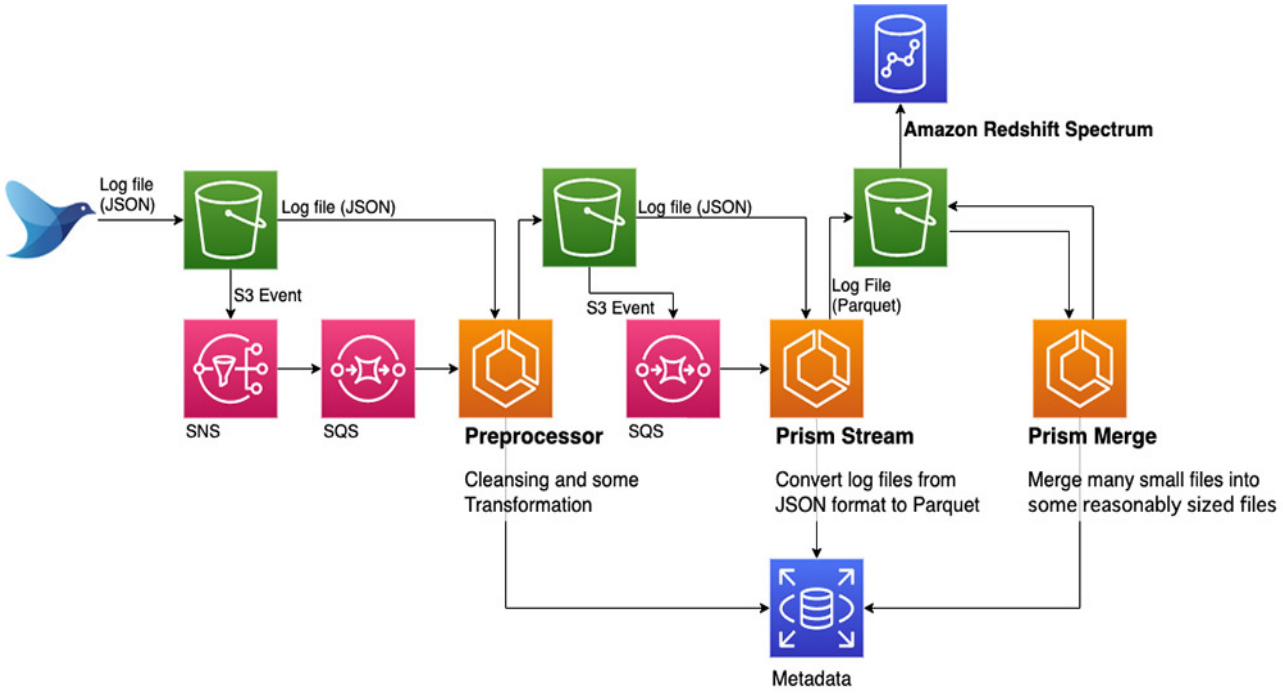

Рисунок 2 – Amazon Redshift [4]

Amazon Redshift требует, чтобы вычислительные ресурсы были настроены в виде кластеров, содержащих один или несколько узлов, каждый из которых имеет свой собственный процессор, память и оперативную память. Leader Node компилирует запросы и передает их вычислительным узлам, которые выполняют запросы.

Данные хранятся на каждом узле в блоках, называемых срезами, используя колоночное хранение. Архитектура MPP (Massively Parallel Processing), используемая в Redshift, разбивает большие наборы данных на куски, которые назначаются слайсам в каждом узле. Это позволяет узлам обрабатывать запросы одновременно, что приводит к более быстрому выполнению запросов. Узел Leader Node объединяет результаты и возвращает их клиентскому приложению.

Клиентские приложения, такие как BI и аналитические инструменты, имеют возможность непосредственного подключения к Redshift с помощью драйверов PostgreSQL JDBC и ODBC с открытым исходным кодом. Это позволяет аналитикам выполнять свою работу непосредственно на данных Redshift. Redshift способен загружать только структурированные данные. Для загрузки данных в Redshift можно использовать заранее интегрированные системы, такие как Amazon S3 и DynamoDB, передав данные с любого локального хоста с подключением SSH, или интегрировать другие источники данных через API Redshift.

Архитектура BigQuery. Google BigQuery - это корпоративное хранилище данных в облаке, которое было представлено в 2011 году. Благодаря безсерверной архитектуре, BigQuery может масштабироваться и работать с большими объемами данных очень быстро. За годы работы над платформой было добавлено множество новых функций для повышения безопасности и надежности, а также удобства использования для пользователей.

Google BigQuery основан на безсерверной архитектуре, в которой поставщик использует разные машины для управления ресурсами. Следовательно, клиенты не участвуют в процессе управления ресурсами, так как за них это делает сам поставщик услуг. Архитектура BigQuery поддерживает как традиционную загрузку данных, так и потоковую передачу данных, последняя предназначена для приема данных в режиме реального времени.

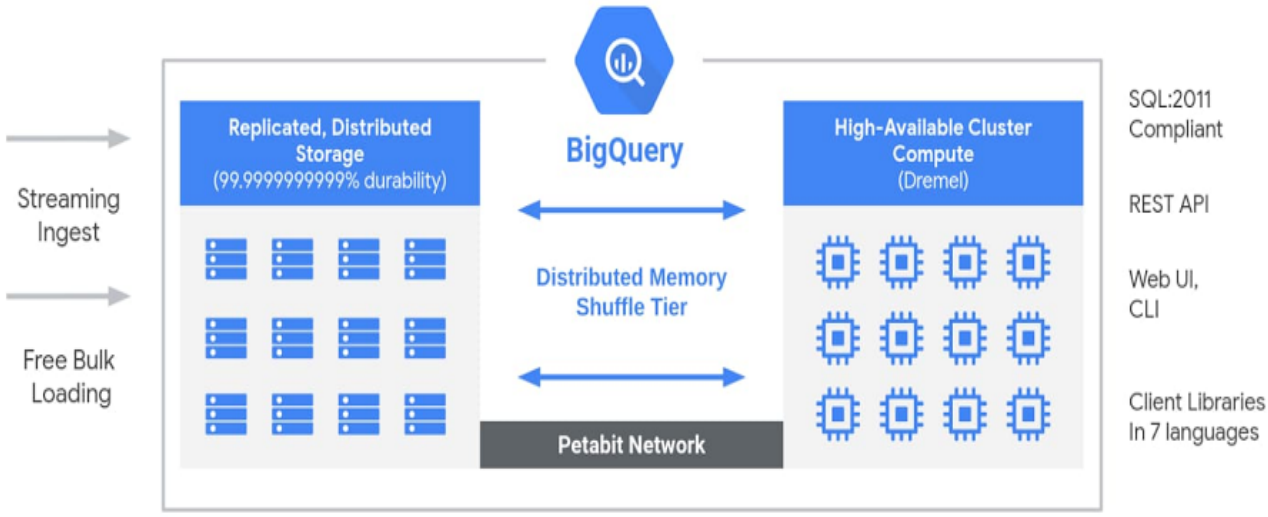

Рисунок 3 – Архитектура BigQuery [5]

Dremel является главным компонентом архитектуры облачного хранилища данных, который представляет собой механизм массовых параллельных запросов, способный читать сотни миллионов строк за секунду. Данные хранятся в системе управления файлами под названием Colossus в столбцовом формате, который распространяется в объеме 64 мегабайта, и помещаются в кластеры, состоящие из разных узлов. Запросы отправляются между разными машинами с помощью древовидной архитектуры для уменьшения времени отклика.

Google BigQuery – это часть комплексной платформы Google Cloud для анализа данных, которая охватывает всю цепочку создания ценности аналитики, включая прием, обработку и хранение данных, а также расширенную аналитику и совместную работу. BigQuery интегрирован с предложениями GCP по анализу и обработке данных, что позволяет клиентам создать корпоративное облачное хранилище данных [5].

Для выполнения запросов к данным используются простые команды SQL.

Архитектура Panoply. Panoply обеспечивает комплексное управление данными как услуга. Его уникальная самооптимизирующаяся архитектура использует машинное обучение и обработку естественного языка (NLP) для моделирования и рационализации передачи данных от источника к анализу, сокращая время от данных до значения как можно ближе к нулю.

Интеллектуальная инфраструктура данных Panoply включает в себя следующие функции

1 Анализ запросов и данных – определение наилучшей конфигурации для каждого варианта использования, корректировка ее с течением времени и создание индексов, сортировочных ключей, дисковых ключей, типов данных, вакуумирование и разбиение.

2 Идентификация запросов, которые не следуют передовым методам – например, те, которые включают вложенные циклы или неявное приведение – и переписывает их в эквивалентный запрос, требующий доли времени выполнения или ресурсов.

3 Оптимизация конфигурации сервера с течением времени на основе шаблонов запросов и изучения того, какая настройка сервера работает лучше всего. Платформа плавно переключает типы серверов и измеряет итоговую производительность.

Заключение. Приведен пример возможных решений по принципам структуризации Баз Данных. На текущий момент существуют несколько вариантов развития структурирования данных для последующей обработки данных. Облачные хранилища данных – это большой шаг вперед по сравнению с традиционными подходами к архитектуре. Однако пользователи по-прежнему сталкиваются с рядом проблем при их настройке. Например, обновления, вставки и удаления могут быть сложными и должны выполняться осторожно, чтобы не допустить снижения производительности запросов, а также трудно иметь дело с полуструктурированными данными – их необходимо нормализовать в формате реляционной базы данных, что требует автоматизации больших потоков данных

Список литературы

[5] Архитектура хранилищ данных: традиционная и облачная [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/441538/. – Дата доступа: 23.02.2023.

[6] Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы. - М.: Издательский дом “Вильямс”, 2007. - 400с.

[7] Кузнецов С.Д. Базы данных: языки и модели. - М.: ООО “Бином-Пресс”, 2008. - 720 с.

[8] How Cookpad scaled its Amazon Redshift cluster while controlling costs with usage limits [Electronic Resource]. – Mode of access: https://aws.amazon.com/ru/blogs/big-data/how-cookpad-scaled-its-amazon-redshift-cluster-while-controlling-costs-with-usage-limits/. – Date of access: 21.02.2023.

[9] BigQuery explained: An overview of BigQuery's architecture [Electronic Resource]. – Mode of access https://cloud.google.com/blog/products/data-analytics/new-blog-series-bigquery-explained-overview. – Date of access: 21.02.2023.

UDC 004.921

STORAGE FEATURES

Kuzmin I.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Piskun G.A. – PhD, associate professor, associate professor of the Department of ICSD

Annotation. The article discusses the basic principles of data structuring when working with a DBMS. The main types of data warehouse architectures are indicated.

Keywords: *DB, Databases, DBMS.*