**Содержание**

Введение

Глава 1. Оперативная аналитическая обработка данных (OLAP)

1.1 Определение и требования OLAP-систем

1.2 Классификация OLAP - систем по типу доступа к БД

1.3 Сравнение OLAP - серверов

Глава 2. Обработка файлов копии базы данных stackoverflow.com.

2.1 Выбор системы управления базами данных

2.2 Подготовка данных из копии базы данных stackoverflow.com для вставки в СУБД

**Введение**

“Кто владеет информацией, тот владеет миром”. Но сегодня информация это не просто набор данных - это результат анализа.

Актуальность дипломной работы обуславливается тем, что современная корпоративная база данных любого предприятия в современном мире содержит набор таблиц, хранящих записи об объектах либо фактах (например, о пользователях, записях пользователей). Как правило, каждая запись в подобной таблице описывает какой-то конкретный факт или объект. Совокупность большого количества таких записей, накопленных за несколько лет, может стать источником дополнительной, гораздо более ценной информации, которую нельзя получить на основе одной конкретной записи, а именно, сведений о закономерностях, тенденциях или взаимозависимостях между данными. Объемы данных настолько внушительны, что человеку просто не по силам проанализировать их самостоятельно, хотя необходимость проведения такого анализа вполне очевидна, ведь в этих "сырых данных" заключены знания, которые могут быть использованы при принятии решений. Для того чтобы провести анализ данных, используются специальные средства, которые помогают выявить скрытые тенденции, строить стратегию развития, находить новые пути развития.

Такой анализ нужен прежде всего, аналитикам, руководителям компаний и лицам, принимающим ключевые бизнес решения.

Кроме централизации и структурирования хранимых данных предприятия требуется инструмент для просмотра и визуализации хранимой информации. Традиционные отчеты лишены гибкости. К ним нельзя применить фильтрацию, чтобы получить желаемое представление данных. Конечно, можно вызвать программиста, и он сделает новый отчет достаточно быстро - скажем, в течение часа. Но в современных условиях ведения бизнеса этого недостаточно. Оперативность в данном случае — один из факторов успеха. Нужен такой инструмент, который позволил бы визуализировать данные быстро, просто и удобно. В качестве такого инструмента и выступает OLAP.

На основе вышеизложенного целью дипломного проекта следует определить разработку аналитического сервиса.

Для решения поставленной цели необходимо рассмотреть ряд сопутствующих задач:

1. Рассмотреть оперативную аналитическую обработку данных (OLAP).
2. Произвести обработку файлов копии базы данных stackoverflow.com.
3. Произвести анализ базы данных stackoverflow.com.
4. Разработать веб-приложение для отображения результатов анализа данных.

Объектом исследования является база данных stackoverflow.com.

Предметом исследования является аналитический сервис для базы данных stackoverflow.com.

**Глава 1. Оперативная аналитическая обработка данных (OLAP)**

### 1.1 Определение и требования OLAP-систем

OLAP (On-Line Analytical Processing) - технология оперативной аналитической обработки данных, использующая методы и средства для сбора, хранения и анализа многомерных данных в целях поддержки процессов принятия решений.

Краткая формулировка требований к OLAP заключена в так называемом тесте FASMI – Fast Analisys of Shared Multidimentional Information, который был сформулирован в начале 1995 года:

 Fast (Быстрый) - предоставление пользователю результатов анализа за приемлемое время, пусть даже ценой менее детального анализа;

 Analysis (Анализ) - возможность осуществления любого логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для конечного пользователя виде;

 Shared (Разделяемый) - многопользовательский доступ к данным с поддержкой соответствующих механизмов блокировок и средств авторизованного доступа;

 Multidimensional (Многомерный) - многомерное концептуальное представление данных, включая полную поддержку для иерархий и множественных иерархий (это ключевое требование OLAP);

 Information (Информация) - приложение должно иметь возможность обращаться к любой нужной информации, независимо от ее объема и места хранения.

**1.2 Классификация OLAP - систем по типу доступа к БД**

На данный момент существует три основных модели OLAP отличающихся по способу организации база дынных:

* многомерный OLAP (MOLAP, Multidimensional OLAP);
* реляционный OLAP (ROLAP, Relational OLAP);
* гибридный OLAP (HOLAP, Hybrid OLAP).

Многомерный OLAP

В многомерных OLAP-системах структура куба хранится в многомерной базе данных. В той же базе данных хранятся предварительно обработанные агрегаты и копии листовых значений. В связи с этим все запросы к данным удовлетворяются многомерной системой баз данных, что делает MOLAP-системы исключительно быстрыми.

Архитектура MOLAP требует большего объема дискового пространства из-за хранения в многомерной базе копий листовых данных.

Достоинства MOLAP-систем:

* все данные хранятся в многомерных структурах, что существенно повышает скорость обработки запросов;
* доступны расширенные библиотеки для сложных функций оперативного анализа;
* обработка разреженных данных выполняется лучше, чем в ROLAP.

Недостатки:

* необходимы специальные инструменты для формирования кубов и их пересчёта в случае изменения базовых значений;
* сложно изменять измерения без повторной агрегации.

Реляционный OLAP

В реляционных OLAP-системах структура куба данных хранится в реляционной базе данных. Меры самого нижнего уровня остаются в реляционной витрине данных, служащей источником данных для куба. Предварительно обработанные агрегаты также хранятся в реляционной таблице.

Благодаря реляционным таблицам, архитектура ROLAP позволяет хранить большие объемы данных.

Достоинства этого класса систем:

* возможность манипулирования большими объемами данных; объем данных могут ограничивать только лежащие в основе ROLAP системы реляционных баз данных, подход ROLAP сам по себе не ограничивает объем данных;
* безопасность и администрирование обеспечивается реляционными СУБД.

Недостатки:

* Меньшая производительность, чем у MOLAP и HOLAP. Для обеспечения сравнимой с MOLAP производительности реляционные системы требуют тщательной проработки схемы БД и специальной настройки индексов;
* функциональность систем ограничивается возможностями SQL, так как аналитические запросы пользователя транслируются в SQL-операторы выборки;
* сложно пересчитывать агрегированные значения при изменениях начальных данных.

Гибридный OLAP

В гибридных OLAP сочетаются черты ROLAP и MOLAP, отсюда и название – гибридный. В моделях HOLAP используются преимущества и минимизируются недостатки обеих архитектур.

В HOLAP-системах структура куба и предварительно обработанные агрегаты хранятся в многомерной базе данных. Это позволяет обеспечить быстрое извлечение агрегатов из структур MOLAP. Значения нижнего уровня иерархии в HOLAP остаются в реляционной витрине данных, которая служит источником данных для куба.

К достоинствам подхода можно отнести комбинирование технологии ROLAP для разреженных данных и MOLAP для плотных областей, а к недостаткам – необходимость поддерживания MOLAP и ROLAP.

Так как анализ будет производиться над данными из реляционной БД, то следует выбрать реляционный OLAP (ROLAP).

**1.3 Сравнение OLAP - серверов**

В приведенных таблицах сравниваются основные и технические характеристики основных OLAP-серверов.

## Таблица 1 - Общая информация

|  |  |  |
| --- | --- | --- |
| OLAP - сервер | Производитель | Лицензия |
| Microsoft Analysis Services | Microsoft | - |
| Essbase | Oracle | - |
| TM1 | IBM | - |
| Mondrian OLAP server | Pentaho | свободное программное обеспечение |
| Palo | Jedox | - |
| Oracle OLAP Option | Oracle | - |
| Microstrategy OLAP Services | Microstrategy | - |
| SAS OLAP Server | SAS Institute | - |
| icCube OLAP Server | MISConsulting SA | свободное программное обеспечение |

Таблица 2 - Модель хранения данных

|  |  |  |  |
| --- | --- | --- | --- |
| OLAP - сервер | MOLAP | ROLAP | HOLAP |
| Microsoft Analysis Services | Да | Да | Да |
| Essbase | Да | Да | Да |
| TM1 | Да | Нет | Нет |
| Mondrian OLAP server | Нет | Да | Нет |
| Palo | Да | Нет | Нет |
| Oracle OLAP Option | Да | Да | Да |
| Microstrategy OLAP Services | Да | Да | Нет |
| SAS OLAP Server | Да | Да | Да |
| icCube OLAP Server | Да | Нет | Нет |

Таблица 3 - API и язык запросов

|  |  |  |  |
| --- | --- | --- | --- |
| OLAP - сервер | XML для анализа | MDX | Пользовательские функции |
| Microsoft Analysis Services | Да | Да | Да |
| Essbase | Да | Да | Да |
| TM1 | Да | Да | Да |
| Mondrian OLAP server | Да | Да | Да |
| Palo | Да | Да | Да |
| Oracle OLAP Option | Нет | Да | Да |
| Microstrategy OLAP Services | Да | Да | Да |
| SAS OLAP Server | Да | Да | Нет |
| icCube OLAP Server | Да | Да | Да |

Таблица 4 - Операционные системы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OLAP - сервер | Windows | Linux | Unix | z/OS |
| Microsoft Analysis Services | Да | Нет | Нет | Нет |
| Essbase | Да | Да | Да | Нет |
| TM1 | Да | Да | Да | Нет |
| Mondrian OLAP server | Да | Да | Да | Да |
| Palo | Да | Да | Да | Нет |
| Oracle OLAP Option | Да | Да | Да | Да |
| Microstrategy OLAP Services | Да | Да | Да | Нет |
| SAS OLAP Server | Да | Да | Да | Да |
| icCube OLAP Server | Да | Да | Да | Да |

Для выполнения аналитической обработки данных был выбран Mondrian OLAP server, по следующим критериям:

* поддерживает модель хранения данных ROLAP;
* бесплатно распространяемое программное обеспечение.

В данной главе мы ознакомились с основами OLAP.

Мы узнали следующее:

* OLAP - технология обработки данных, заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу.
* Приложения с OLAP-функциональностью должны предоставлять пользователю результаты анализа за приемлемое время, осуществлять логический и статистический анализ, поддерживать многопользовательский доступ к данным, осуществлять многомерное концептуальное представление данных и иметь возможность обращаться к любой нужной информации.

Кроме того, мы выбрали Mondrian OLAP server для аналитической обработки данных, который позволяет анализировать большие и сложные объемы данных в режиме реального времени. Поддерживает язык запросов MDX, а также спецификации XML for Analysis. Для хранения данных может использовать любую систему управления базами данных с поддержкой SQL.

**Глава 2. Обработка файлов копии базы данных stackoverflow.com.**

**2.1 Выбор системы управления базами данных**

Рассмотрим две основных свободно распространяемых СУБД:

* MySQL - самая распространённая СУБД
* PostgreSQL - наиболее продвинутая СУБД

MySQL

MySQL - это полноценная серверная СУБД. MySQL очень функциональная, свободно распространяемая СУБД, которая успешно работает с различными сайтами и веб-приложениями. Обучиться использованию этой СУБД довольно просто, так как на просторах интернета вы легко найдете большее количество информации.

Несмотря на то, что в ней не реализован весь SQL функционал, MySQL предлагает довольно много инструментов для разработки приложений.

Преимущества MySQL

* Простота в работе - установить MySQL довольно просто.
* Богатый функционал - MySQL поддерживает большинство функционала SQL.
* Масштабируемость - MySQL легко работает с большими объемами данных и легко масштабируется.
* Скорость - упрощение некоторых стандартов позволяет MySQL значительно увеличить производительность.

Недостатки MySQL

* Известные ограничения - по задумке в MySQL заложены некоторые ограничения функционала, которые иногда необходимы в особо требовательных приложениях.
* Проблемы с надежностью - из-за некоторых способов обработки данных MySQL (связи, транзакции, аудиты) иногда уступает другим СУБД по надежности.
* Многопоточность - параллельные операции чтения-записи могут создать проблемы.

PostgreSQL

PostgreSQL является наиболее профессиональным СУБД, чем MySQL. Она свободно распространяемая и максимально соответствует стандартам SQL.

От других СУБД PostgreSQL отличается поддержкой востребованного объектно-ориентированного и/или реляционного подхода к базам данных. Например, полная поддержка надежных транзакций, т.е. атомарность, последовательность, изоляционность, прочность (Atomicity, Consistency, Isolation, Durability (ACID).) Благодаря мощным технологиям Postgre очень производительна.

Хотя PostgreSQL и не может похвастаться большой популярностью в отличии от MySQL, существует довольно большое число приложений облегчающих работу с PostgreSQL. Сейчас довольно легко установить эту СУБД используя стандартные менеджеры пакетов операционных систем.

Достоинства PostgreSQL

* Открытое ПО соответствующее стандарту SQL - PostgreSQL - бесплатное ПО с открытым исходным кодом. Эта СУБД является очень мощной системой.
* Надежное и защищенное хранение данных, которыми она управляет.
* Большое сообщество - существует довольно большое сообщество в котором вы запросто найдёте ответы на свои вопросы
* Большое количество дополнений - несмотря на огромное количество встроенных функций, существует очень много дополнений, позволяющих разрабатывать данные для этой СУБД и управлять ими.
* Расширения - существует возможность расширения функционала за счет сохранения своих процедур.
* PostgreSQL предлагает 4-ре типа индексов. Каждый тип индекса имеет свой алгоритм реализации, что позволяет существенно увеличить быстродействие.

Недостатки PostgreSQL

* Популярность - по своей природе, популярностью эта СУБД похвастаться не может, хотя и присутствует довольно большое сообщество.

На основании выше изложенного в качестве СУБД выбран PostgreSQL.

**2.2 Подготовка данных из копии базы данных stackoverflow.com для вставки в СУБД**

Для обработки данных использовался Apache Spark.

Apache Spark — это фреймворк для анализа больших массивов данных.

Благодаря представлению информации в виде так называемых упругих распределенных наборов данных (RDD - Resilient Distributed Dataset), осуществляется быстрый и простой доступ к данным через прикладные интерфейсы на Python, Scala и Java. Кроме того, такая технология позволяет запрашивать данные и одновременно запускать сложные алгоритмы их анализа.

Большинство операций над RDD не приводит к каким-либо вычислениям, а только создаёт очередную обёртку, обещая выполнить операции только тогда, когда они понадобятся. Ниже показан код для обработки данных из файла типа xml:

val conf = new SparkConf().setMaster("local[4]").setAppName("test")

val sc = new SparkContext(conf)

val rdd = sc.textFile("D:\\stackexchange\\Users.xml")

rdd.map(line => {

val map = scala.collection.mutable.Map[String,String]()

val regex = """([A-z]\*?)=(".\*?")""".r

for (p <- regex.findAllIn(line)) p match {

case regex(key, value) => map += key -> value.replace("\"", "")

}

val id = map.getOrElse("Id","null")

val reputation = map.getOrElse("Reputation","null")

val creationDate = if(map.getOrElse("CreationDate","null").equals("null")) "null"

else "'%s'".format(map.get("CreationDate").get.replace("T", " ").

replaceAll("\\.\\d{3}", ""))

val displayName = map.getOrElse("DisplayName","null")

val lastAccessDate = if(map.getOrElse("LastAccessDate","null").equals("null")) "null"

else "'%s'".format(map.get("LastAccessDate").get.replace("T", " ").

replaceAll("\\.\\d{3}", ""))

val websiteUrl = map.getOrElse("WebsiteUrl","null")

val location = map.getOrElse("Location","null")

val aboutMe = map.getOrElse("AboutMe","null").replace("'", "")

val views = map.getOrElse("Views","null")

val upVotes = map.getOrElse("UpVotes","null")

val downVotes = map.getOrElse("DownVotes","null")

val profileImageUrl = map.getOrElse("ProfileImageUrl","null")

val emailHash = map.getOrElse("EmailHash","null")

val age = map.getOrElse("Age","null")

val accountId = map.getOrElse("AccountId","null")

"INSERT INTO users values (%s,%s,%s,'%s',%s,'%s','%s','%s',%s,%s,%s,'%s','%s',%s,%s);".

format(id, reputation, creationDate, displayName, lastAccessDate, websiteUrl, location, aboutMe, views, upVotes, downVotes, profileImageUrl, emailHash, age, accountId)

}).saveAsTextFile("D:\\stackexchange\\users")

}

Вычисления начинаются только на последней строчке: Spark видит, что нужно материализовать результаты, и для этого начинает применять операции к данным.

Каждая строчка считывается из файла и обрабатывается следующим образом: из строки выделяются подстроки подходящие под следующее регулярное выражение: ([A-z]\*?)=(".\*?"). Полученные подстроки попадают в коллекцию “Ключ-Значение”, после чего в зависимости от колонок таблицы вытаскиваются нужные значения по ключу, если в коллекции не оказалось такого ключа записывается null. Все полученные значения записываются в строку шаблон для вставки в СУБД , после чего эта строка добавляется в файл.