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

ПОСТРОЕНИЕ И СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ BUSINESS INTELLIGENCE СИСТЕМ НА MICROSOFT И PENTAHO

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 273 группы
направления 01.04.02 — Прикладная математика и информатика
факультета КНиИТ
Евсеева Кирилла Дмитриевича

Научный руководитель
доцент, к. ф.-м. н. _____ А. С. Иванов

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

Саратов 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Целью настоящей выпускной квалификационной работы является построение BI-систем для многомерного анализа данных при помощи OLAP-кубов, основанных на программном обеспечении с закрытым и открытым исходным кодом, реализация хранилища данных и ETL процессов, как при помощи SSIS и Pentaho, так и в облачных сервисах. А также сравнение производительности построенных систем.

Актуальность работы заключается в необходимости работникам предприятия быстро оценить положение, анализируя огромное количество данных во всех возможных разрезах, в том числе используя интеллектуальные алгоритмы. Для малых и средних предприятий с решением этой задачи может справиться построение BI-систем на основе бесплатного программного обеспечения, так как BI-системы построенные на основе платных решений (Microsoft, Oracle) зачастую являются слишком дорогими.

Поставленные задачи:

— разработка DWH на базе MSSQL;
— разработка ETL на базе SSIS;
— построение MOLAP-системы с помощью технологий SSAS;
— разработка DWH на базе PostgreSQL;
— разработка ETL на базе Pentaho Kettle;
— построение ROLAP-системы с помощью технологий Mondrian;
— создание колоночных индексов;
— реализация секционирования в MSSQL;
— реализация секционирования в PostgreSQL;
— сравнение производительности построенных систем;
— визуализация данных при помощи Power BI.

1 Анализ технологий и средств построения BI-системы

1.1 BI и OLAP

Business intelligence (сокращённо BI) — обозначение компьютерных методов и инструментов для организаций, обеспечивающих перевод транзакционной деловой информации в человекочитаемую форму, пригодную для бизнес-анализа, а также средства для массовой работы с такой обработанной информацией [1–4].

Цель BI — интерпретировать большое количество данных, заостряя внимание лишь на ключевых факторах эффективности, моделируя исход различных вариантов действий, отслеживая результаты принятия решений.

Создание DWH является одним из необходимых условий внедрения на предприятии системы управления эффективностью бизнеса.

Технология комплексного многомерного анализа данных получила название OLAP (On-Line Analytical Processing). OLAP — это ключевой компонент организации традиционных хранилищ данных [5–8]. Концепция OLAP была описана в 1993 году Эдгаром Коддом, известным исследователем баз данных и автором реляционной модели данных. В 1995 году на основе требований, изложенных Коддом, был сформулирован так называемый тест FASMI (Fast Analysis of Shared Multidimensional Information — быстрый анализ разделяемой многомерной информации), включающий следующие требования к приложениям для многомерного анализа.

1.2 Программное обеспечение с закрытым исходным кодом

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase [9]. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

В работе используется версия Microsoft SQL Server 2014.

Microsoft SQL Server 2014 представляет собой единую BI-платформу, с помощью которой можно быстрее получить доступ к внутренним и внешним данным, ускорить выполнение анализа таких данных, а также формировать и

удалять их [10,11]. Кроме того, повышается надежность критически важных приложений, что обеспечивает компаниям быстрый поиск нужных данных и своевременное принятие решений. Платформа SQL Server 2014 поддерживает возможность масштабирования для серверов, сетевых технологий и систем хранения данных с помощью Windows Server 2012 R2 [9,12–15]. Решение SQL Server 2014 было разработано для использования в гибридной среде, охватывающей локальные и облачные вычисления. Благодаря новым инструментам, которые позволяют облегчить создание решений для резервного копирования и аварийного восстановления, платформа Microsoft SQL Server 2014 готова к установке в облачные среды.

Microsoft Службы Integration Services — это платформа для построения решений по интеграции и преобразованию данных уровня предприятия. Службы Службы Integration Services используются при решении сложных бизнес-задач путем копирования и загрузки файлов, отправки электронных сообщений в ответ на события, обновления хранилищ данных, очистки и интеллектуального анализа данных, а также управления объектами и данными SQL Server [16–20]. Пакеты могут работать отдельно или совместно с другими пакетами для решения сложных бизнес-задач. Службы Integration Services могут извлекать и преобразовывать данные из ряда таких источников, как файлы XML-данных, неструктурированные файлы и источники реляционных данных, и затем загружать эти данные в один или несколько реляционных объектов.

Основной причиной для построения многомерной модели службы Analysis Services является достижение высокой производительности нерегламентированных запросов к бизнес-данным [18,21,22]. Многомерная модель состоит из кубов и измерений, которые могут быть аннотированы и расширены для поддержки сложных конструкций запросов. Разработчики бизнес-аналитики создают кубы для поддержки малого времени ответа, а также для предоставления единого источника данных для бизнес-отчетности. При растущей важности бизнес-аналитики на всех уровнях организации единый источник аналитических данных обеспечивает минимизацию разногласий, если не полное их устранение [22–24].

В нашем случае работа в Oracle не представляется возможность в виду неоправданно больших цен для поставленных задач.

1.3 Программное обеспечение с открытым исходным кодом

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации [25].

PostgreSQL создана на основе некоммерческой СУБД Postgres, разработанной как open-source проект в Калифорнийском университете в Беркли. К разработке Postgres, начавшейся в 1986 году, имел непосредственное отношение Майкл Стоунбрейкер, руководитель более раннего проекта Ingres, на тот момент уже приобретённого компанией Computer Associates. Название расшифровывалось как «Post Ingres», и при создании Postgres были применены многие уже ранее сделанные наработки [26].

PostgreSQL может обрабатывать много данных, что является одним из ключевых преимуществ при разработке OLAP-системы:

— максимальный размер базы данных-Неограничен;
— максимальный размер таблицы-32 TB;
— максимальный размер строки-1.6 TB;
— максимальный размер поля-1 GB;
— максимальное количество строк в таблице — неограниченно;
— максимальное количество столбцов в таблице-250-1600 в зависимости от типа столбца;
— максимальное количество индексов в таблице — неограниченно;

Для сравнения, MySQL и MariaDB имеют лишь 65 535 байт для строки [25,26]. Firebird также предлагает всего лишь 64Кб. Обычно объём данных ограничивается максимальным размером файлов операционной системы. Поскольку PostgreSQL умеет хранить табличные данные в множестве файлов меньшего размера, он может обойти это ограничение. Но стоит отметить, что слишком большое количество файлов может негативно сказаться на производительности. MySQL и MariaDB поддерживают большее количество столбцов в таблице (до 4,096 в зависимости от типа данных) и большие индивидуаль-

ные размеры таблицы, чем PostgreSQL, но необходимость превысить существующие ограничения PostgreSQL возникает лишь в крайне редких случаях [27, 28].

Pentaho является оптимальным выбором для разработки OLAP-системы на свободном программном обеспечении. Среди бесплатных платформ Pentaho имеет самый широкий спектр для работы с ETL и OLAP.

Для построение многомерного куба используется Mondrian. Сервер Mondrian состоит из трёх слоёв, выделяются слой измерений, звёздный слой и слой хранения:

1. Слой измерений разбирает, проверяет и выполняет MDX-запросы [19, 20]. MDX-запрос в Mondrian выполняется в несколько этапов. Сначала вычисляются оси, затем значения ячеек на осях, для эффективности, слой измерений посылает запросы к ячейкам на уровень агрегирования партиями. Трансформатор запросов позволяет приложению управлять существующими запросами, вместо того чтобы строить MDX-выражения с нуля для каждого запроса. Метаданные описывают и собственно модель измерений, и то, как она отображается на реляционную модель.
2. Слой звезды отвечает за поддержание кэша агрегатов. Агрегат — набор измеренных значений (ячеек) в памяти, соответствующий определённому набору значений столбцов измерений [20]. Слой измерений посылает запросы для получения набора ячеек. Если требуемые ячейки не находятся в кэше, или получаются свёртыванием агрегатов в кэше, менеджер агрегатов посылает запрос на слой хранения.
3. Слой хранения обеспечивает хранение исходных данных, необходимых для получения агрегатов. Принципиально, Mondrian поддерживает любые jdbc-источники данных; в частности, заявляется о коммерческой поддержке SQL-серверов DB2, Oracle Database, Microsoft SQL Server, MySQL, PostgreSQL, колоночных хранилищ Greenplum и Infobright, аппаратнопрограммных комплексов Teradata Database, Netezza, Neoview, а также возможен доступ к неструктурированным источникам, включая некоторые NoSQL-системы, в частности, поддерживаются MongoDB и Hadoop-источники — HDFS, HBase, Hive [29–32].

2 Реализация и сранение BI-систем

2.1 Разработка Data Warehouse

Разработка BI систем имеет смысл только тогда, когда есть поставленная задача структурировать конечный объем информации. Чтобы BI был эффективен и помогал данные компании анализировать вопросы поставленные перед системой должны быть четко сформулированы. Только зная правильный вопрос, можно получить правильный ответ.

Поэтому сначала требуется определить с какими данными необходимо работать. В свободном доступе нет внутренних данных коммерческих предприятий. Государственные организации же в свою очередь такие данные могут предоставить.

Для сравнения работы BI систем потребуется достаточно большой объем данных. Именно поэтому в качестве источника данных было выбрано FTP хранилище государственных закупок.

Измерениями будут служить адреса поставщиков и заказчиков (от улицы до региона), тип продукта, организации участвующие в тендерах, тип контракта и даты его исполнения, а также таблица стандартов ОКЕИ.

Факт таблицей будет сам заключенный контракт с его ценой и количеством товара.

Список измерений:

1. Dimension Address - данные об адресах заказчика и исполнителя
2. Dimension Contract - дополнительные данные о контракте
3. Dimension OKPD - официальные наименования товаров в РФ
4. Dimension Organization - данные о предприятиях
5. Dimension Product - дополнительные данные о продукте
6. Dimension Date - дата по годам, месяцам и т.д.
7. Dimension OKEI - системы измерения товара
8. Dimension OKPD2 - дополнительные данные о товаре, если такие пристутствуют

Схема DWH изображена на рисунке 1.

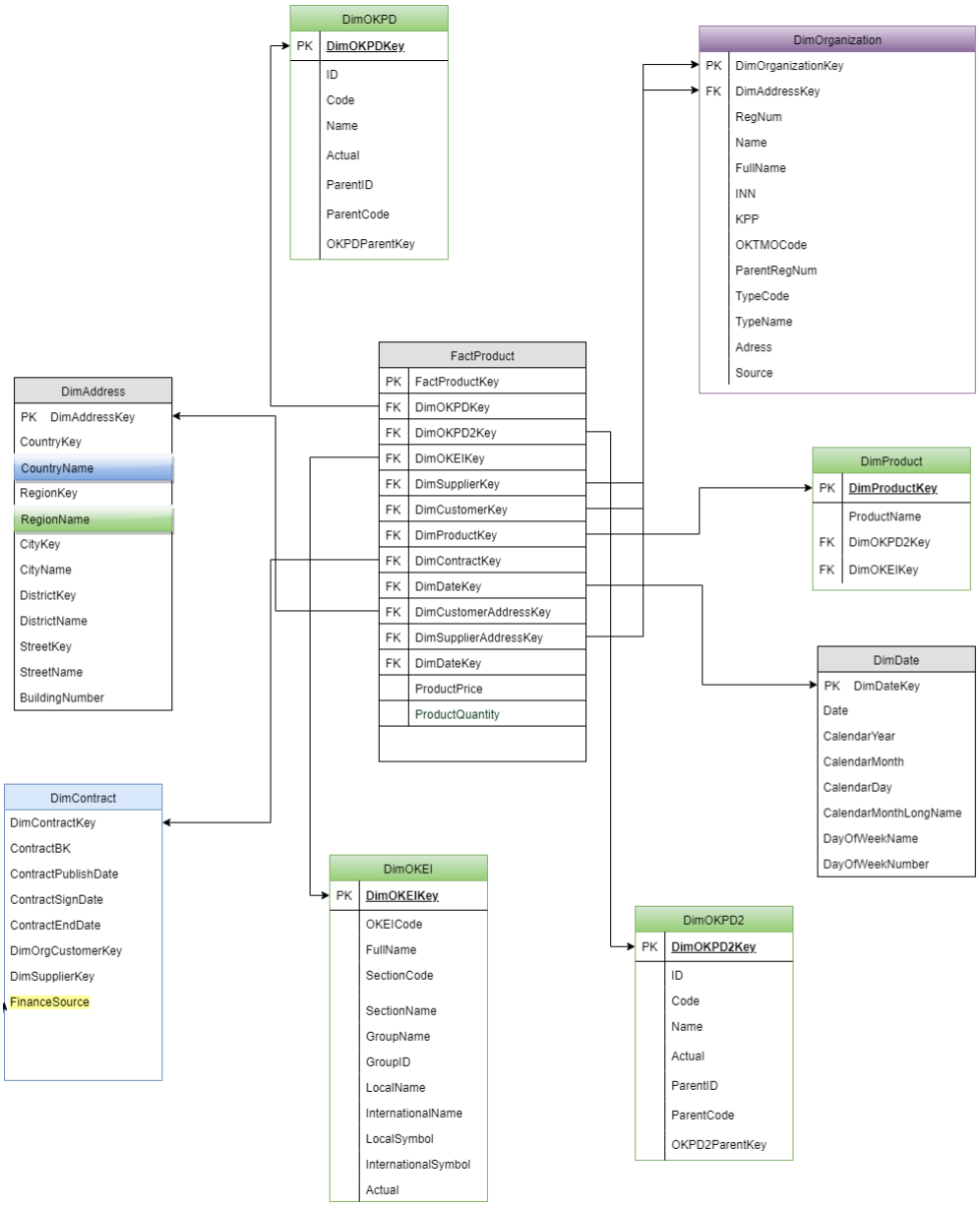

Рисунок 1 – Схема DWH

2.2 Разработка ETL

В начале стоит сказать что компонент SSIS входит в пакет услуг предоставляемых при покупке MSSQL и поэтому является только условно платным.

Все загружаемые данные сначала загружаются через bulk insert в staging схему базы данных. Для каждого из типа загруэаемого файла в данной схеме есть таблица без каких либо ограничений. Это уровень так называемых сырых данных.

Данные государственных закупок на ETL хранятся по определенной системе и в формате zip архивов. Схема представлена на рисунке 2.

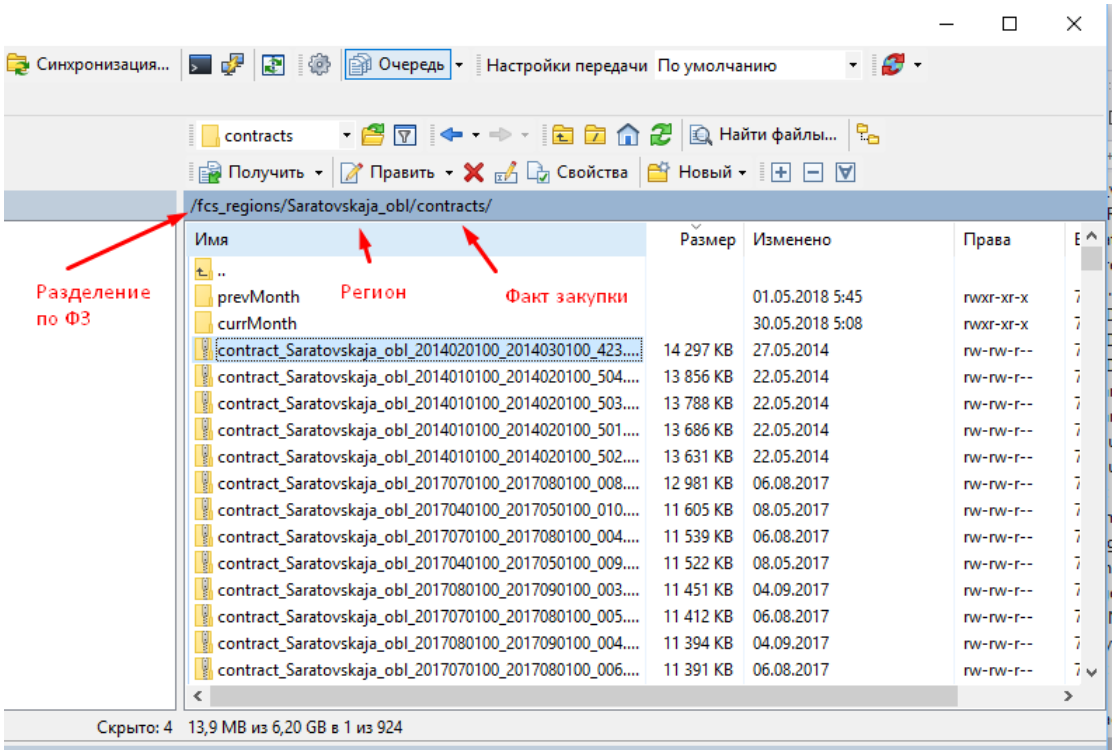

Рисунок 2 – Схема хранения файлов на FTP

С помощью распределения по областям и городам заполняется dimension address.

Для загрузки данных с FTP были разработаны 3 пакета:

— ExtractOKEIfromFTP.dtsx
— ExtractOKPDfromFTP.dtsx
— ExtractOKPD2fromFTP.dtsx
— ExtractContractFromXML.dtsx

Для преобработки данных были разработаны 3 пакета:

— ExtractOKEIfromZIP.dtsx
— ExtractOKPDfromZIP.dtsx
— ExtractOKPD2fromZIP.dtsx

Для загрузки данных в dimensions были разработаны 6 пакета:

— LoadDimOKEI.dtsx
— LoadDimOKPD.dtsx
— LoadDimOKPD2.dtsx
— LoadDimAddress.dtsx
— LoadDimOrganization.dtsx

— LoadDimProduct.dtsx

Данные пакеты используют в себе компоненты Lookup, Fuzzy Lookup, Container, For Each Container и различные компоненты для запросов в базу данных.

2.3 Разработка OLAP системы

OLAP системы от Microsoft также идут в комплекте к продукту MSSQL. В качестве основной цели для анализа были выбраны контракты. Схема куба представлена на рисунке 3.

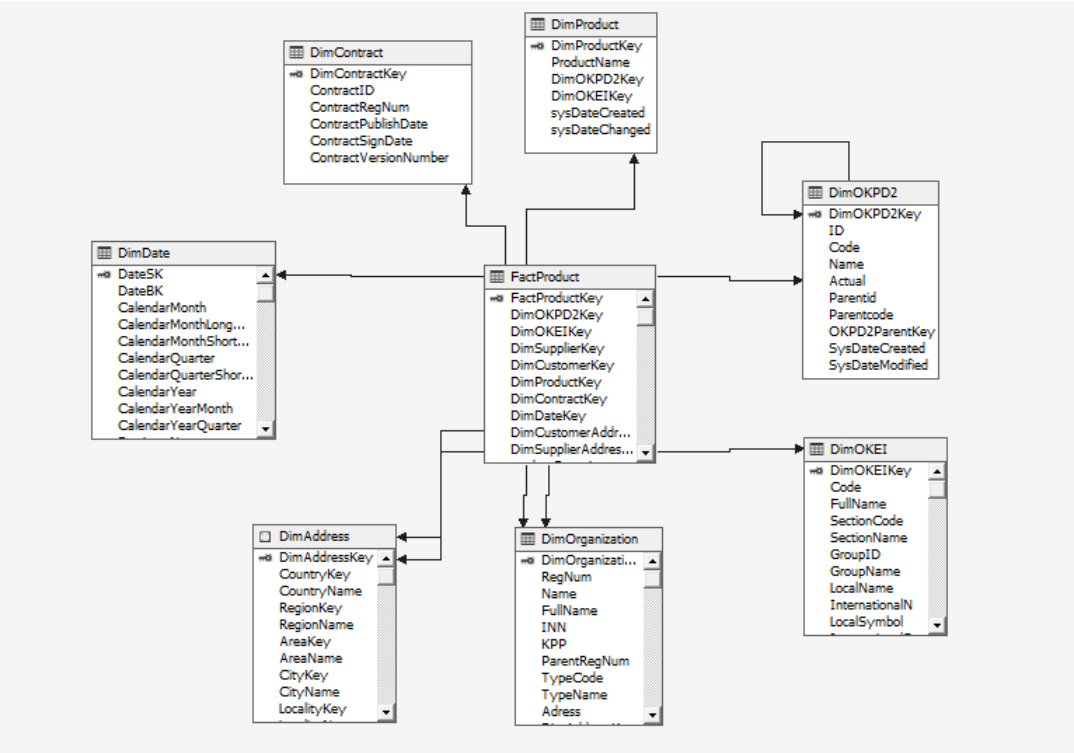

Рисунок 3 – Схема куба

Как было описано в схеме DWN были созданы соответсnвующие связи в кубе (рисунок 4).

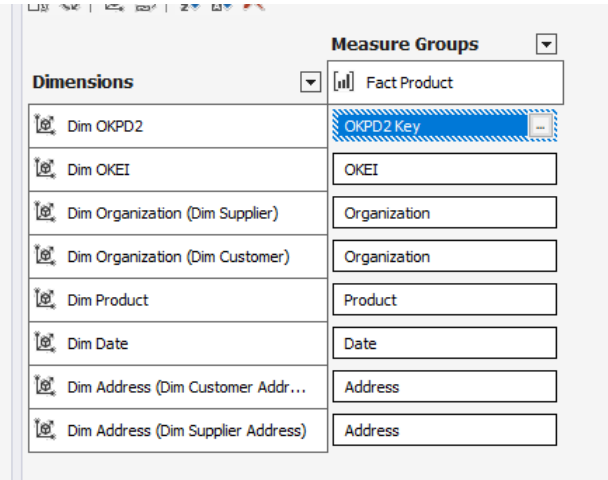

Рисунок 4 – Связи куба

MOLAP куб создается на отдельной базе Analysis сервера, что позволяет освободить базу DWH от запросов, однако занимает возможно излишнее место. Так же одним из основных преимуществ является хранение в этой базе уже заранее сформированных разрезов по измерениям и соответственно значений разных фактов.

Для дальнейшего использования была создана калькуляция (рисунок 5).

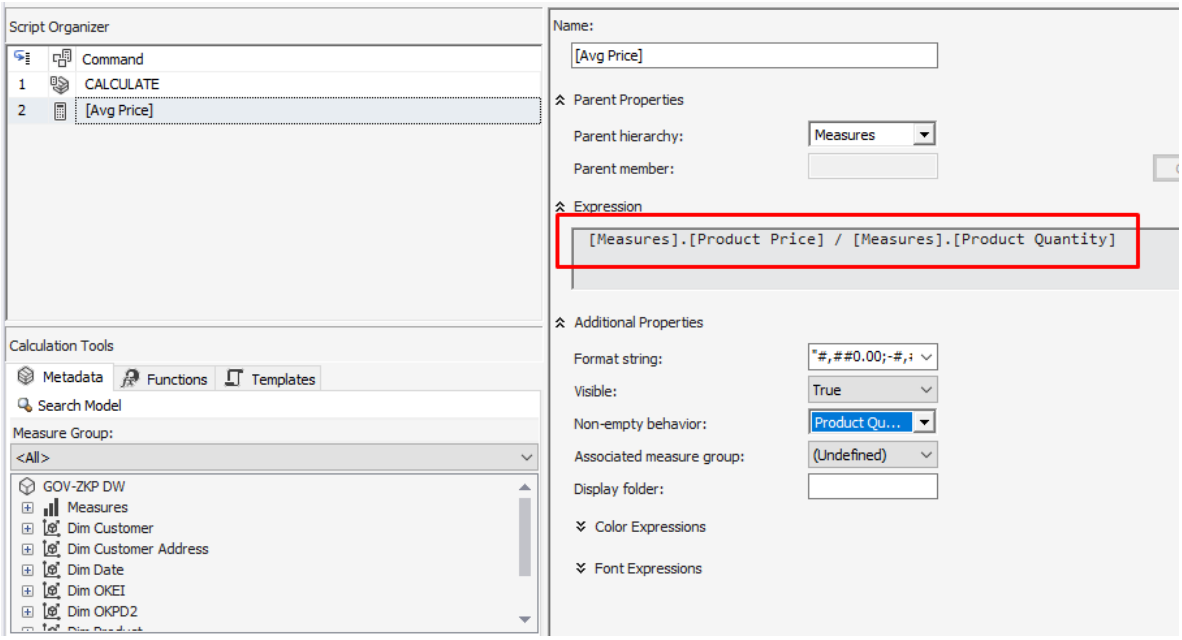

Рисунок 5 – Калькуляция в кубе

Подобных подготовленных калькуляций в ROLAP решении не предоставлено, поэтому запросы к базе занимают дополнительное время.

Общая схема работы ROLAP системы представлена на рисунке 6).

Mondrian поддерживает язык запросов MDX, который и будет исполь-

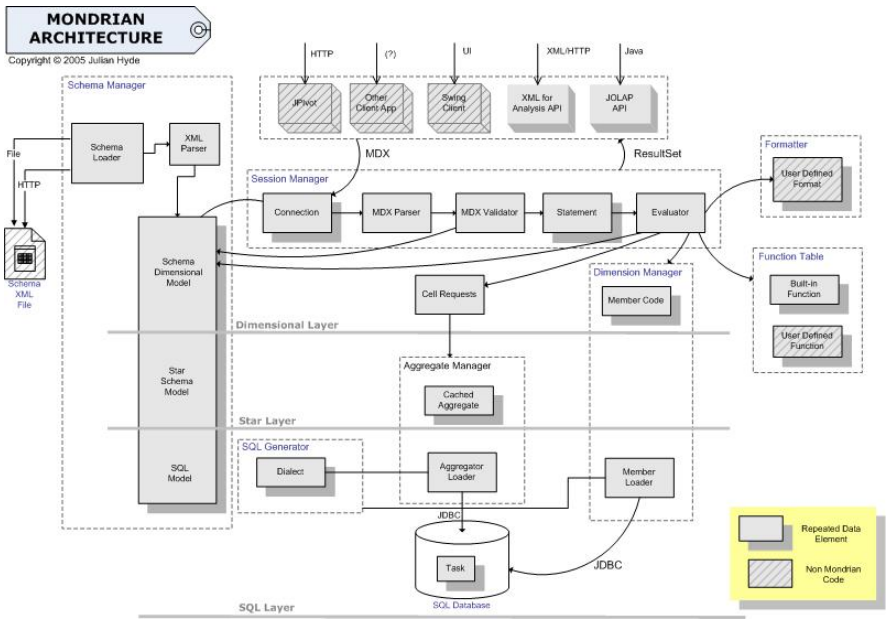

Рисунок 6 – Архитектура Mondrian

зоваться как основным при агрегации различных данных, а также для последующего сравнения скорости работы.

2.4 Реализация ETL в облаке

Задачей является реализовать ETL процесс для загрузки данных из FTP хранилища государственных закупок в хранилище данных, как это могло быть реализовано с помощью служб SSIS.

Схема потока данных изображена на рисунке 7:

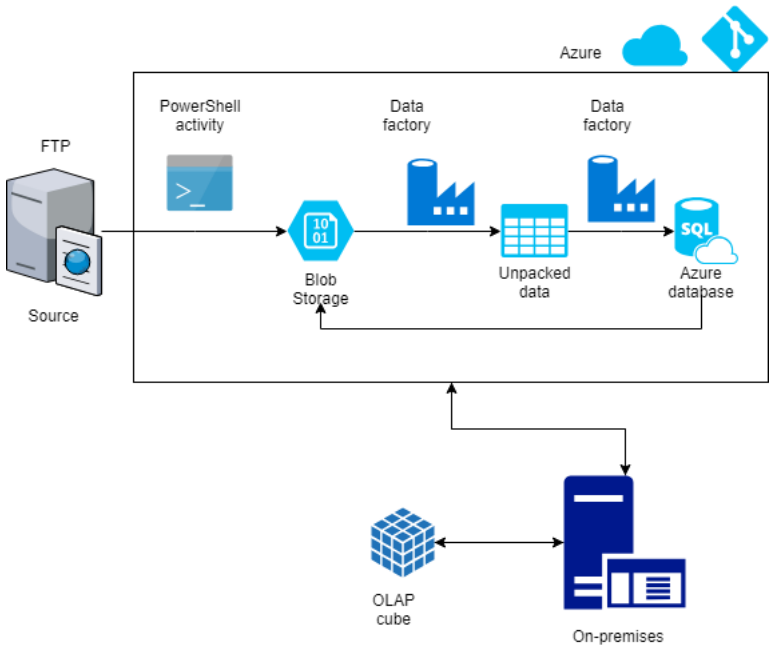

Рисунок 7 – Azure Data Flow

ЗАКЛЮЧЕНИЕ

В ходе данной работы были реализованы все поставленные задачи такие как:

- разработка DWH на базе MSSQL;
- разработка ETL на базе SSIS;
- построение MOLAP-системы с помощью технологий SSAS;
- разработка DWH на базе PostgreSQL;
- разработка ETL на базе Pentaho Kettle;
- построение ROLAP-системы с помощью технологий Mondrian;
- создание колоночных индексов;
- реализация секционирования в MSSQL;
- реализация секционирования в PostgreSQL;
- сравнение производительности построенных систем;
- визуализация данных при помощи PowerBI.

В данной работе были проанализированы все необходимые теоретические материалы, даны понятия всем основным терминам, которые используются в данной работе.

Было реализовано хранилище данных, а также ETL процесс для его заполнения. Также было реализовано секционирование в MSSQL и PostgreSQL. Проведен сравнительный анализ скорости выполнения запросов при создании колоночных индексов, спроектированы и созданы OLAP системы.

После этого были построены ETL процессы в облачных ресурсах, таких как Azure и AWS

Решения ETL оказались сопоставимыми как по функционалу, так и по скорости, чего нельзя сказать о построенных OLAP системах, где MOLAP выигрывает в производительности. Однако если система не подразумевает нагрузки большого количества пользователей, то систему ROLAP можно признать удачной, так как она справляется со своими задачами.

Решения ETL в облаке также оказались похожи, однако Azure предоставляет большую гибкость и простоту разработки. В свою очередь AWS при правильном подходе к архитектуре, может обойтись гораздо дешевле, чем Azure. Данные результаты также были опубликованы в статье в научном журнале [33].

Визуализация данных работала одинаково быстро, как на PowerBI, так

и на PantahoBI.

Несмотря на особые принципы секционирования PostgreSQL для сравнительно небольших систем предоставляет больше возможностей, за счет наследования таблиц для каждой секции. MSSQL за счет более детального подхода может быть предпочтительнее только в больших системах.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Ben-Gan, I.* Training Kit 70-461 Querying Microsoft SQL Server / I. Ben-Gan. — Microsoft, 2012.

2 *Ben-Gan, I.* Training Kit 70-462 Implementing a Data Warehouse / I. Ben-Gan. — Microsoft, 2012.

3 *Ben-Gan, I.* Training Kit 70-463 Administering Microsoft SQL Server / I. Ben-Gan. — Microsoft, 2012.

4 *Ben-Gan, I.* Training Kit 70-465 Designing Database Solutions / I. Ben-Gan. — Microsoft, 2012.

5 *Eckerson, W.* Performance dashboards / W. Eckerson. — Wiley, 2010.

6 *Simon, P.* Too big to ignore / P. Simon. — Wiley, 2013.

7 *Howson, C.* Successful Business Intelligence / C. Howson. — Mc Graw Hill, 2013.

8 *Sherman, R.* Business Intelligence Guidebook / R. Sherman. — Wiley, 2015.

9 *Кузнецов, С. Д.* Основы баз данных / С. Д. Кузнецов. — Москва: Лаборатория знаний, 2007.

10 *Гроффф, Дж.* Энциклопедия SQL / Дж. Грофф. — Москва: Лори, 2003.

11 *Оти, М.* Будущее sql server: прогноз Теда Каммерта / М. Оти // *Открытые системы.* — 2011. — № 01.

12 *Петкович, Д.* Microsoft SQL Server 2012. Руководство для начинающих / Д. Петкович. — СПб: БХВ-Петербург, 2013.

13 *Casters, M.* The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data / M. Casters. — Wiley, 2004.

14 *Casters, M.* The Data Warehouse Lifecycle Toolkit: Practical Techniques for Building Data Warehouse and Business Intelligence Systems / M. Casters. — Wiley, 2004.

15 *Casters, M.* The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling / M. Casters. — Wiley, 2004.

16 *Reeve, A.* Managing Data in Motion: Data Integration Best Practice Techniques and Technologies / A. Reeve. — MK, 2014.

17 *Doan, A.* Principles of Data Integration / A. Doan. — MK, 2012.

18 *Dyche, J.* Customer Data Integration: Reaching a Single Version of the Truth / J. Dyche. — Wiley, 2011.

19 *Meadows, A.* Pentaho Data Integration Cookbook / A. Meadows. — PACKT, 2013.

20 *Doan, A.* Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration / A. Doan. — Wiley, 2016.

21 Официальный сайт Microsoft [Электронный ресурс] / Microsoft. — 2017. — URL: https://www.microsoft.com/ru-ru/ (Дата обращения 21.04.2019). Загл. с экр. Яз. русс.

22 Официальный сайт Pentaho [Электронный ресурс] / Pentaho. — 2017. — URL: http://www.pentaho.com/ (Дата обращения 29.04.2019). Загл. с экр. Яз. русс.

23 *Feuerstein, S.* Oracle PL/SQL Programming / S. Feuerstein. — O'REILLY, 1995.

24 *Savas, G.* The Oracle Book: Answers to Life's Questions / G. Savas. — O'REILLY, 2001.

25 Официальный сайт MySQL [Электронный ресурс] / MySQL. — 2017. — URL: https://www.mysql.com/ (Дата обращения 27.04.2019). Загл. с экр. Яз. русс.

26 Официальный сайт PostgreSQL [Электронный ресурс] / PostgreSQL. — 2017. — URL: https://www.postgresql.org/ (Дата обращения 10.05.2019). Загл. с экр. Яз. русс.

27 *Wong, G.* AWS Basics: Beginners Guide / G. Wong. — Wiley, 2015.

28 *Sarkar, A.* Learning AWS / A. Sarkar. — PACKT, 2015.

29 *Schahan, R.* Microsoft Azure Essentials - Fundamentals of Azure / R. Schahan. — Microsoft, 2015.

30 *Schinder, D.* Microsoft Azure Security Infrastructure / D. Schinder. — Microsoft, 2016.

31 *Savill, J.* Mastering Microsoft Azure Infrastructure Services / J. Savill. — Microsoft, 2015.

32 *Barns, J.* Microsoft Azure Essentials Azure Machine Learning / J. Barns. — Microsoft, 2015.

33 *Евсеев, К. Д.* Сравнение проприетарных и свободных bi-систем / К. Д. Евсеев, А. С. Иванов // *Интернаука: CXVII Международный научно-практическая конференция «Молодой исследователь: вызовы и перспективы»*. — 2019. — Vol. 17(117). — Pp. 177–181.