

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование)

02.03.03 Математическое обеспечение и администрирование  
информационных систем

(код и наименование направления подготовки, специальности)

Мобильные и сетевые технологии

(направленность (профиль)/ специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка структуры и построение аналитических кубов для OLAP  
системы»

Студент

И.А. Вирясов

(И.О. Фамилия)

(личная подпись)

Руководитель

М.А. Тренина

(ученая степень, звание, И.О. Фамилия)

Консультант

М.В. Дайнеко

(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

## Аннотация

Тема бакалаврской работы: «Разработка структуры и построение аналитических кубов для OLAP системы».

Бакалаврская работа посвящена разработке структуры и построению аналитических кубов для OLAP системы для организации «АвтоСталь».

В ходе выполнения исследований по бакалаврской работе была разработана структура и построение аналитических кубов для системы OLAP, описано создание куба OLAP, разобрана внутренняя структура компонентов гиперкуба и построены кубические срезы. Описан метод хранения данных в виде гиперкуба.

Бакалаврской работа состоит из введения, трёх глав, заключения и списка литературы.

Во введении прописываются актуальность темы, написаны цели и задачи.

В первой главе анализируется теоретическая информация о технологиях OLAP, освещаются принципы построения куба OLAP, и описывается оперативная аналитическая обработка данных.

Во второй главе делается разработка структуры и построение аналитических кубов для системы OLAP, а так же прописаны все этапы разработки хранилища данных.

В третьей главе вся сделанная работа проходит проверку и тестирование.

В заключении прописаны выводы по всей работе. В ней присутствуют, 27 рисунков. Список литературы состоит из 22 литературных источников. Общий объем всей работы составляет 40 страницы.

## Abstract

The topic of the present graduation work is *Developing the structure and constructing analytical cubes for the OLAP system*.

The research is devoted to developing the structure and constructing the OLAP system analytical cubes for *AvtoStil* Company.

When doing a research, the analytical cubes for the OLAP system are constructed and their structure is developed. We also describe in details the process of creating an OLAP cube for *AvtoStil* Company, discuss the internal structure of the hypercube components, construct the corresponding cubic sections and explain the method of storing data in the form of a hypercube.

The graduation work consists of an introduction, three chapters, a conclusion, and a list of references. The introduction reveals the relevance of the research and gives a brief description of the work done.

The first chapter analyzes some theoretical foundations about OLAP technologies, covers the principles of constructing an OLAP cube and describes the operational analytical data processing.

The second chapter dwells on developing the structure and constructing the analytical cubes for the OLAP system.

The third chapter explains the choice of the development tool, as well as represents the stages of data store development and the main methods used to test it.

The result of this work is creating the OLAP analytical cube ordered by *AvtoStil* Company.

## Содержание

Введение.....	5
1 Теоретические сведения об OLAP–технологиях .....	6
1.1 Понятия, классификация, схемы OLAP–системы.....	6
1.2 Оперативная аналитическая обработка данных .....	10
1.3 Принципы построения OLAP–куба .....	13
1.4 Обзор существующих программных продуктов OLAP–решений .....	14
2 Построение аналитических кубов для OLAP – системы .....	20
2.1 Создание OLAP куба для организации «АвтоСталь».....	20
2.2 Реализация хранилища данных .....	32
3 Проверка созданного OLAP куба .....	36
3.1 Тестирование.....	36
Заключение .....	38
Список используемой литературы .....	39

## Введение

OLAP – это технология для обработки и анализа больших массивов данных в реальном времени. Данная технология позволяет проводить всесторонний анализ коммерческой информации. Системы построены с помощью данной технологии, дают фактически безграничные возможности по составлению отчетов, построению прогнозов и сценариев, разработки множества вариантов планов.

Актуальность темы исследования заключается в том, что в условиях современной экономики для каждого предприятия становится актуальным повысить конкурентоспособность и сохранить ее на должном уровне.

Целью выпускной квалификационной работы является разработка структуры и построение аналитических кубов для OLAP системы.

Для достижения цели в рамках выпускной квалификационной работы необходимо решить основные задачи:

1. Изучить основные понятия, классификацию, схемы OLAP–системы.
2. Разобрать принципы построения OLAP–куба.
3. Ознакомится с существующими программными продуктами OLAP–решений.
4. Создать OLAP куб для организации «АвтоСталь».
5. Реализовать хранилище данных с помощью средств языка Transact–SQL.
6. Провести анализ построения срезов куба.
7. Провести тестирование построенного куба и созданного хранилища данных.

Объектом исследования в представленной работе является процесс построения аналитических кубов для OLAP системы.

Предмет исследования: принципы, методы и алгоритмы обработки информации для разработки структуры и построения аналитических кубов для OLAP системы.

## **1 Теоретические сведения об OLAP–технологиях**

### **1.1 Понятия, классификация, схемы OLAP–системы**

Информационные системы, которые разрабатываются и используются для предприятий, содержат в себе приложения, которые предназначены для полного многомерного анализа статистических данных, их динамики развития, наличия взаимосвязей и т.д. Всесторонний анализ подобных систем дает возможность быстро принимать правильное решение для развития бизнеса. Такие системы называются системами поддержки принятия решений.

Принятие важного решения зависит от имеющейся информации, чаще количественной. Для принятия решения необходимо создание хранилищ данных (Data warehouses). Они включают в себя процесс сбора, фильтрации и предварительной обработки данных с целью предоставления результирующей информации пользователям для статистического анализа, создания аналитических отчетов.

Один из создателей концепции хранилища данных, Ральф Кимбалл, описывает хранилище данных как «место, где люди могут получить доступ к своим данным». Он сформулировал основные требования к хранилищам данных [5, с. 14]:

1. Высокая скорость получения данных из хранилища.
2. Внутренняя непротиворечивость данных.
3. Получение и сравнение срезов данных (slice and dice).
4. Удобные утилиты просмотра данных в хранилище.
5. Полнота и достоверность хранимых данных.
6. Качественный процесс пополнения данных.

Часто невозможно удовлетворить все эти требования в рамках одного и того же продукта. Поэтому для реализации хранилищ данных обычно используется несколько продуктов, некоторые из которых фактически

являются средствами хранения данных, другие—средствами их извлечения и просмотра, третьи—средствами их пополнения и т.д.

«Хранилище данных отличается от реляционной базы данных. Реляционные базы данных предназначены для того, чтобы помогать пользователям выполнять рутинную работу, а хранилища данных предназначены для принятия решений. Стоит отметить, что реляционные базы данных подвержены постоянным изменениям в ходе работы пользователей, а хранилище данных наоборот стабильно, данные в нем обычно обновляются по расписанию (например, еженедельно, ежедневно или ежечасно – в зависимости от потребностей). В идеале процесс пополнения – это просто добавление новых данных за определенный период времени без изменения предыдущей информации, уже находящейся в хранилище. Реляционные базы данных обычно являются источником данных, которые поступают в хранилище. Обратите внимание, что хранилище может пополняться из внешних источников, таких как статистические отчеты» [5, с. 21].



Рисунок 1 – Структура хранилища данных

Системы поддержки принятия решений предоставляют пользователю средства отображения агрегированных данных для выборок из исходного набора в удобном для восприятия и анализа виде. На основе групп полученных значений вычисляются агрегатные функции, которые объединяют их в одну результирующее значение.

Оси куба предполагают собой измерения, по которым откладывают параметры. Вдоль каждой оси данные могут быть организованы в иерархию, представляющую различные уровни детализации. Используя эту модель данных, пользователи могут выполнять сложные запросы, создавать отчеты и получать подмножества данных.

Функциональность OLAP может быть реализована различными способами: с помощью простейших инструментов анализа данных в офисных приложениях или с помощью распределенных аналитических систем на основе серверных продуктов.

В процессе анализа данных нередко появляется необходимость построения зависимостей между разными параметрами, количество которых может быть значительным. Измерение представляет собой последовательный набор значений одного из анализируемых параметров. Каждое измерение может быть представлено в виде иерархий.

На пересечении измерений находятся меры, количественно описываемые анализируемые измерения. На рисунке 2 изображено представление данных в виде OLAP куба.



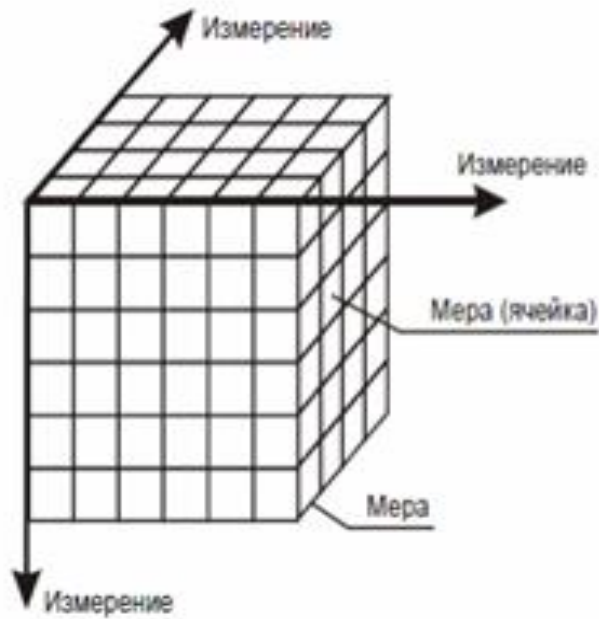


Рисунок 2 – Представление данных в виде гиперкуба

Существует три типа OLAP [6, с. 51]:

1. MOLAP – Multidimensional OLAP.
2. ROLAP – Relational OLAP.
3. HOLAP – Hybrid OLAP.

MOLAP – стандартная форма OLAP. Она использует суммирующую БД, специальный вариант процессора пространственных БД и создаёт требуемую пространственную схему данных с сохранением как базовых данных, так и агрегатов [6, с. 54].

ROLAP работает напрямую с реляционным хранилищем, факты и таблицы с измерениями хранятся в реляционных таблицах, и для хранения агрегатов создаются дополнительные реляционные таблицы [12, с. 9].

HOLAP использует реляционные таблицы для хранения базовых данных и многомерные таблицы для агрегатов [12, с. 9].

Существует несколько схем для отображения и хранения логических схем в базе данных.

Схема "звезды" имеет много синонимов: схема звездного соединения, звездная схема, звездообразная схема (от английского – star scheme) – организация реляционных таблиц, удобная для хранения многомерных показателей. Это основа реляционного OLAP.

Модель данных для такой схемы состоит из двух типов таблиц: таблицы фактов – центр " звезды "и нескольких таблиц измерений – лучи "звезды".

Схема снежинки названа так из-за ее формы, которая отображает логическую схему таблиц в многомерной базе данных. Диаграмма снежинок представлена централизованной таблицей фактов, подключенной к таблицам измерений. Отличие от схемы "звезда" заключается в том, что таблицы измерений нормализуются с рядом других связанных таблиц измерений. В схеме "звезда" таблицы измерений денормализованы. Чем выше степень нормализации таблиц измерений, тем сложнее выглядит структура схемы снежинки. Создается "эффект снежинки", который влияет только на таблицы измерений и не используется для таблиц фактов.

## **1.2 Оперативная аналитическая обработка данных**

Концепция OLAP основана на принципе многомерного представления данных. В 1993 году Э. Ф. Кодд рассмотрел недостатки реляционной модели, прежде всего, указав на невозможность "комбинировать, просматривать и анализировать данные с точки зрения нескольких измерений, то есть наиболее понятным для корпоративных аналитиков способом", и определил общие требования к системам OLAP, которые расширяют функциональность реляционных баз данных и включают многомерный анализ в качестве одной из их характеристик.

Согласно Кодду, многомерное концептуальное представление – это множественная перспектива, состоящая из нескольких независимых измерений, по которым можно анализировать определенные наборы данных.

Одновременный анализ по нескольким измерениям ориентируется как многомерный анализ. Любое измерение подключает направленности консолидации данных, которое состоит из серии поочередных значений обобщения, где каждый более высокий уровень соответствует большей степени агрегации данных для соответствующего измерения.

Таким образом, измерение исполнителя может быть определено направлением консолидации, состоящим из уровней обобщения "предприятие–подразделение–отдел–сотрудник". Измерение времени может даже включать два направления консолидации – "год–квартал–месяц–день" и "день недели", поскольку подсчет времени по месяцам и неделям несовместим. В этом случае становится возможным произвольно выбрать желаемый уровень детализации информации для каждого из измерений.

В схеме присутствуют две операции, которые показывают направления консолидации данных. Операция спуска (drilling down) соответствует движению от высших ступеней консолидации к низшим; напротив, операция подъема (rolling up) означает движение от низших уровней к высшим (рисунок 3).

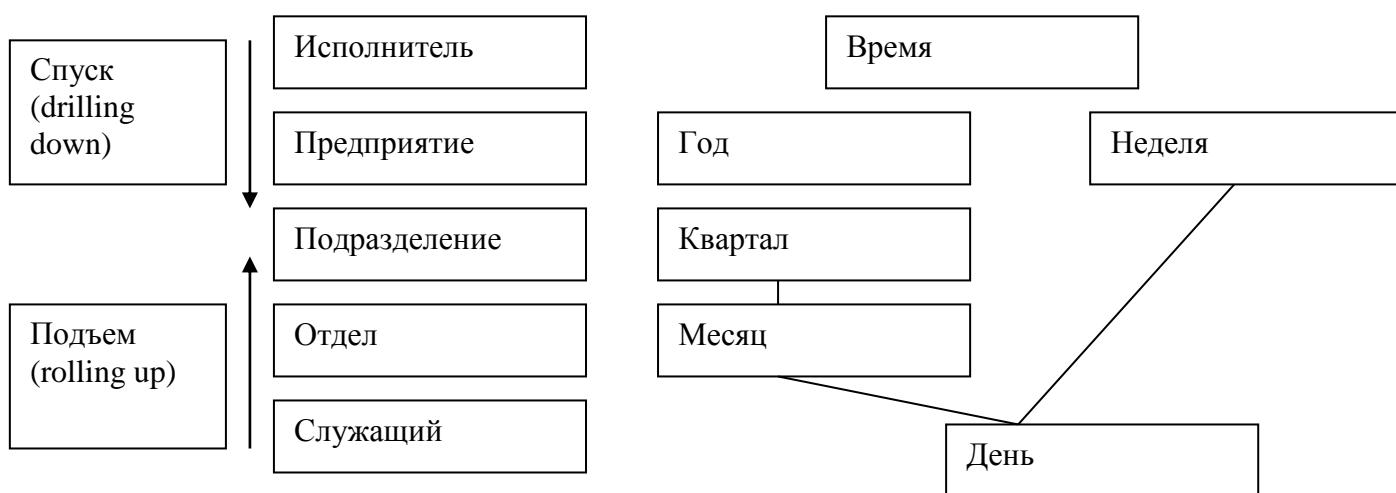


Рисунок 3 – Схема измерения и направления консолидации данных

В 1993 году было сформулировано Эдгаром Кодда 12 основных требований к средствам реализации OLAP.

Многомерное представление данных	Средства должны поддерживать многомерный на концептуальном уровне взгляд на данные.
Прозрачность	Пользователь не должен знать о том, какие конкретные средства используются для хранения и обработки данных, как данные организованы и откуда они берутся.
Доступность	Средства должны сами выбирать и связываться с наилучшим для формирования ответа на данный запрос источником данных. Средства должны обеспечивать автоматическое отображение их собственной логической схемы в различные гетерогенные источники данных.
Согласованная производительность	Производительность практически не должна зависеть от количества Измерений в запросе.
Поддержка архитектуры клиент–сервер	Средства должны работать в архитектуре клиент–сервер.
Равноправность всех измерений	Ни одно из измерений не должно быть базовым, все они должны быть равноправными (симметричными).
Динамическая обработка разреженных матриц	Неопределенные значения должны храниться и обрабатываться наиболее эффективным способом.
Поддержка многопользовательского режима работы с данными	Средства должны обеспечивать возможность работать более чем одному пользователю.
Поддержка операций на основе различных измерений	Все многомерные операции (например Агрегация) должны единообразно и согласованно применяться к любому числу любых измерений.
Простота манипулирования данными	Средства должны иметь максимально удобный, естественный и комфортный пользовательский интерфейс.
Развитые средства представления данных	Средства должны поддерживать различные способы визуализации (представления) данных.
Неограниченное число измерений и уровней агрегации данных	Не должно быть ограничений на число поддерживаемых Измерений.

Рисунок 4 – Таблица требований к средствам реализации OLAP

Изначально эти требования назывались в честь создателя Codd, но потом их переименовали в тест FASMI, эти требования нужны для приложения OLAP, чтобы обеспечивать возможность быстрого анализа совместно используемой многомерной информации.

### 1.3 Принципы построения OLAP–куба

Технологии OLAP являются одним из наиболее эффективных инструментов для анализа больших объемов данных. Общее определение этого инструмента можно дать следующим образом. OLAP (OnLine Analytical Processing – аналитическая обработка данных в реальном времени) – это подход к обработке аналитических данных, основанный на их многомерном иерархическом представлении. [12, с. 24].

С точки зрения пользователя, OLAP–системы предоставляют функции для гибкого просмотра информации, автоматического сбора агрегированных данных, выполнения аналитических операций свертки, детализации и сравнения с течением времени.

Ключевые требования системы OLAP – это скорость, которая позволяет пользователям быстро взаимодействовать с информацией, и многомерность. В этом смысле OLAP–системы отличаются, во–первых, от традиционных систем управления реляционными базами данных (далее–СУБД), результаты запросов от которых, как правило, достаточно дороги с точки зрения времени ожидания и загрузки, что приводит к снижению эффективности интерактивной работы с такими базами данных (в частности, при наличии больших объемов информации) [6, с.17]. Во–вторых, OLAP–системы отличаются от плоскофайлового представления данных, например, в виде обычных традиционных электронных таблиц, представление многомерных данных в которых сложно.

На рисунке 5 представлен пример OLAP–куба.

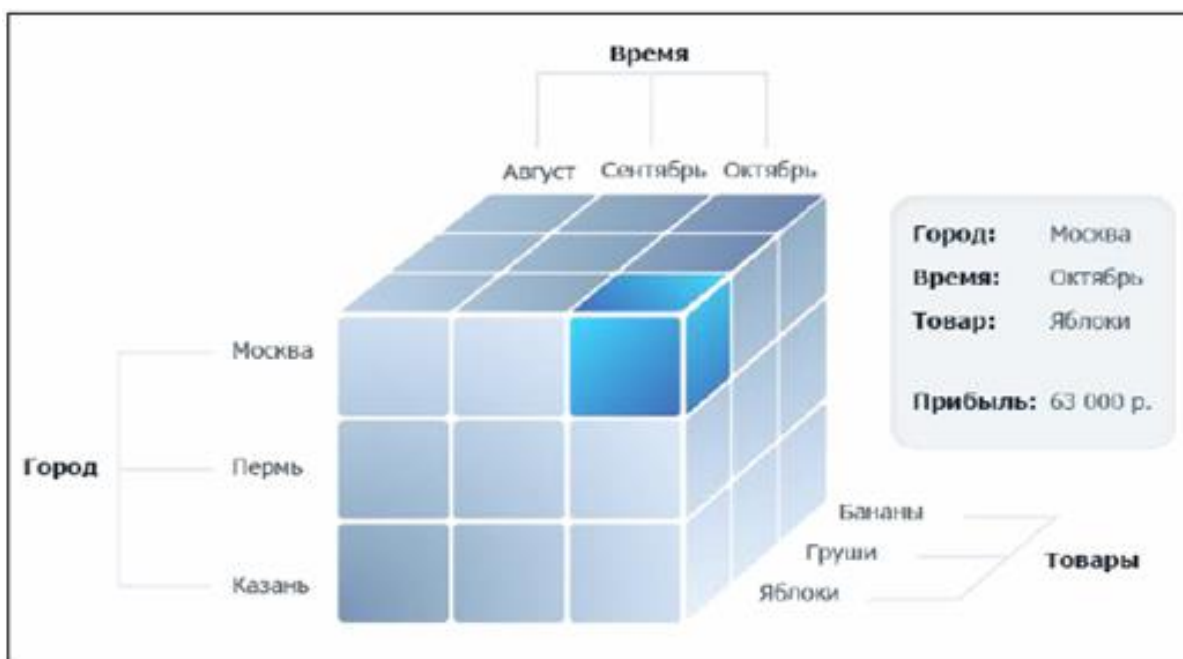


Рисунок 5– Пример OLAP куба

Как правило, кубы OLAP в большинстве случаев содержат различные бизнес–показатели, используемые для анализа и принятия любых управленческих решений, такие как: собственные и заемные средства, прибыль и рентабельность продуктов.

Такие показатели хранятся в кубах в виде разделов, которые представляют основные категории деятельности организации: товары, склады, клиенты и даты транзакций.

#### 1.4 Обзор существующих программных продуктов OLAP–решений

Разработка различных (в том числе серверных) OLAP–систем в основном проходит в несколько этапов.

На первом этапе данные извлекаются из базы данных экономического объекта (предприятия) в так называемое хранилище SQL (база данных SQL Server) , а затем модифицируются в форму, необходимую для анализа. Часто на этом этапе возникают определенные проблемы из–за того, что многие

программные продукты (например, программная платформа 1С) используют систему автоматических имен объектов, поэтому структура может быть изменена при переносе их в базу данных SQL.

На втором этапе разрабатываются кубы OLAP на основе данных.

Третий этап включает в себя настройку инструментов доступа пользователей к аналитическим базам данных для клиентов, настройку отчетности и т.д.

Разработка OLAP-решений для различных предприятий осуществляется, чаще всего, фирмами, специализирующимися на обработке данных.

Рассмотрим существующие на данный момент аналитические платформы, которые ориентированы на комплексное решение различных задач.

Аналитическая платформа Deductor ориентирована на решение различных задач с использованием методов интеллектуального анализа данных. Этот продукт использует в качестве основы собственное программное обеспечение и аналитические разработки. Анализ OLAP не является приоритетом для этого продукта, но он также присутствует. Кроме того, важной особенностью является возможность многоступенчатого преобразования информации, которая описана в режиме проектирования. Этот продукт был изготовлен для решения задач, таких как анализ клиентской базы или прогнозирование продаж с помощью математических методов, т. е. задачи бизнес-анализа. [10, с. 74].

На данный момент платформа Deductor способна решать задачи аналитики в разнообразных областях, таких как:

1. Банки и финансы.
2. Розничная торговля.
3. Оптовая торговля.
4. Телекоммуникации.
5. Промышленность.

6. Информационные технологии.

7. Медицина.

8. Наука и образование.

Основные компоненты платформы Deductor:

1. Deductor Warehouse – хранилище данных.

2. Deductor Studio – обработка и передача данных.

3. Deductor Viewer – просмотр итоговых отчетов при помощи Deductor Studio.

4. Deductor Server – аналитическая обработка данных.

5. Deductor Client – обеспечение доступа к серверам из других приложений.

Среди главных преимуществ платформы Deductor можно выделить:

1. Встроенная интеграция со множеством различных источников данных.

2. Высокая скорость работы с данными.

3. Поддержка большинства современных технологий анализа структурированных данных.

Общая схема архитектуры аналитической платформы Deductor показана на рисунке 6.



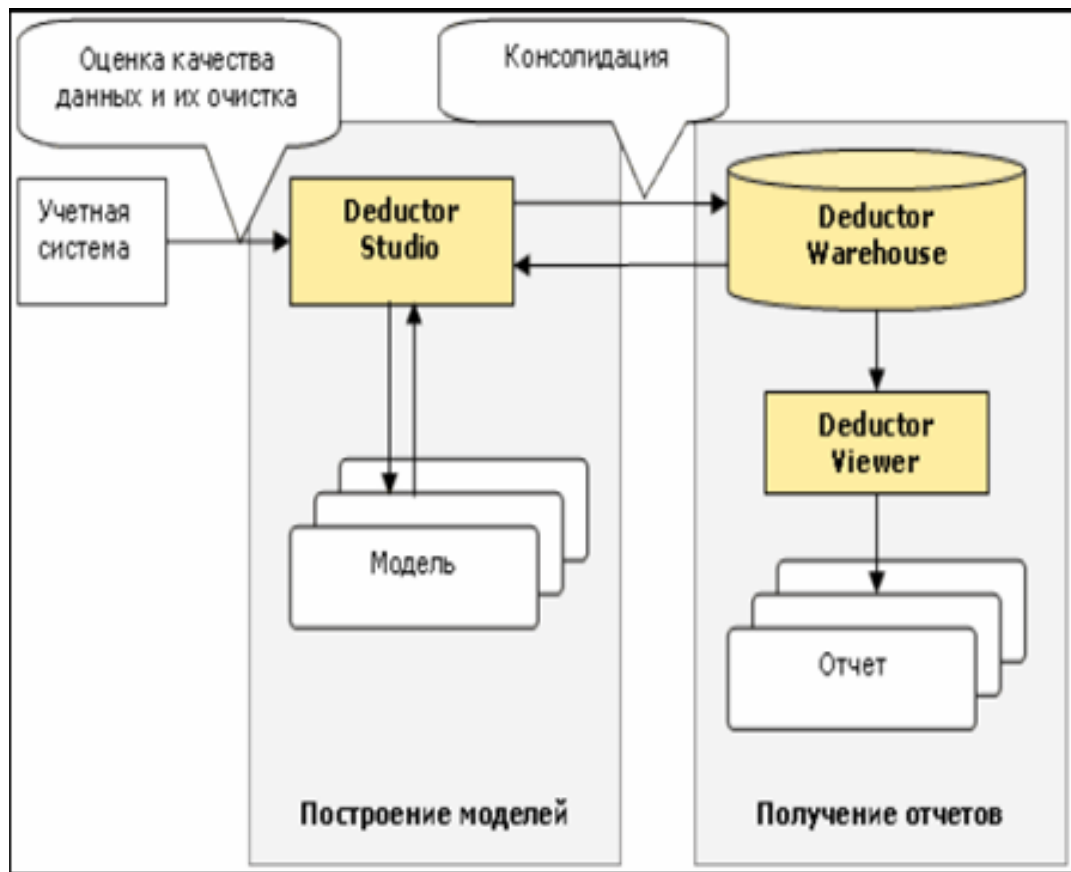


Рисунок 6 – Архитектура платформы Deductor (Base Group)

Существует много других фирм, которые делают комплексные решения. Например, фирма BIX предоставляет комплексное решение на платформе Microsoft SQL Server для анализа данных с использованием OLAP-технологий. Для работы с OLAP-кубами можно использовать программу Excel или какой-либо другой OLAP-клиент. Кроме построения кубов, также существует возможность воспользоваться средствами MS SQL Reporting Service для генерации отчетности [7, с. 32].

Основные компоненты системы BIX BI (Business Intelligence):

1. Корпоративное хранилище данных Microsoft SQL Server – данный модуль предоставляет высокопроизводительную оптимизированную базу данных, поддерживающую процедуры загрузки, очистки и преобразования данных.

2. Система корпоративной отчетности Reporting Services – данный модуль предоставляет пользователю графические средства создания новых отчетов.

3. Многомерные витрины данных Microsoft Analysis Services – данный модуль предоставляет аналитические данные, применяемые в бизнес–отчетах и клиентских приложениях, таких как Excel, отчеты служб Reporting Services.

4. Портал Sharepoint – данная платформа предоставляет различным пользователям возможность совместного доступа к данным, или возможность совместной работы. Общая схема архитектуры системы ВІХ ВІ показана на рисунке 7.

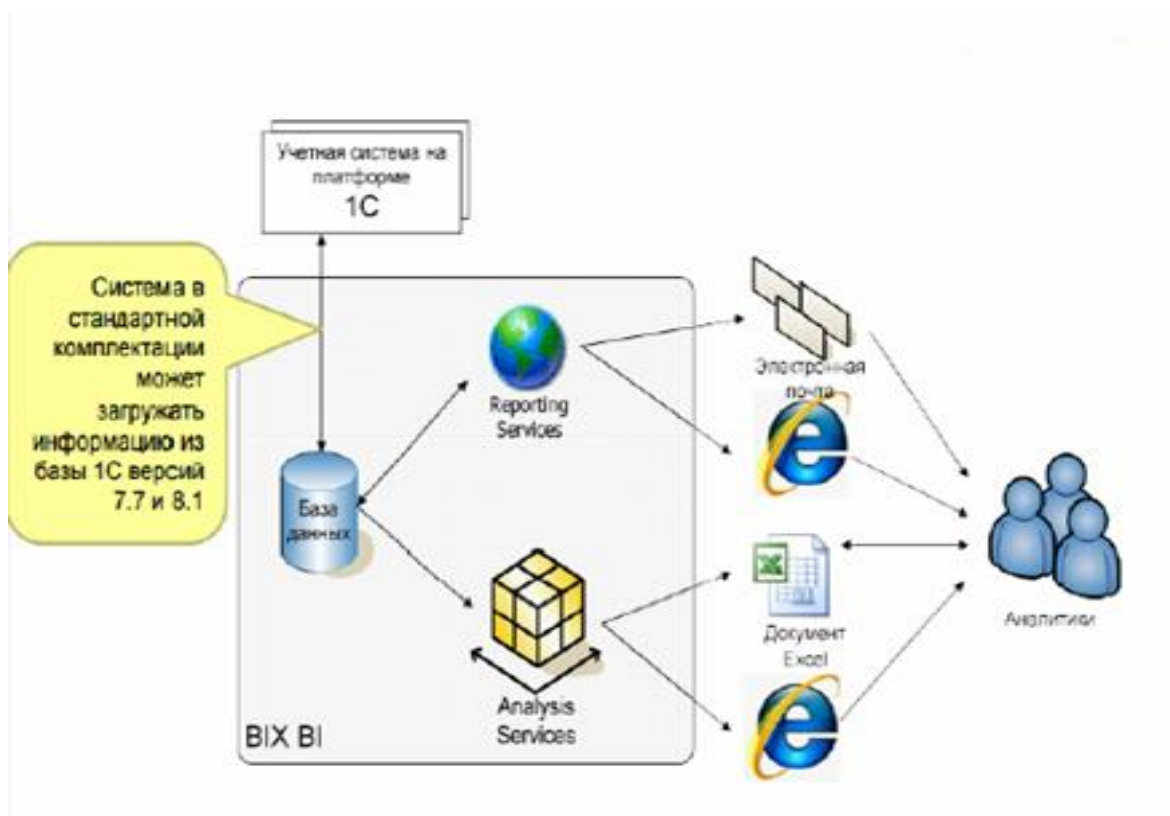


Рисунок 7 – Архитектура продукта ВІХ ВІ

Основные преимущества системы ВІХ ВІ перед другими системами анализа данных:

1. При построении различных срезов данных в созданных кубах система учета не загружается.
2. Обычно повторное сканирование кубов занимает всего несколько часов, независимо от того, когда меняется система учета.
3. Знакомый интерфейс Excel и механизм сводной таблицы и сводной диаграммы.

### **1.5 Выводы по главе**

В этой главе мы рассмотрели теоретическую информацию о технологиях OLAP, были рассмотрены принципы построения, был произведен обзор всех существующих программных продуктов для OLAP решений. Проанализировав все системы анализа данных, мы остановились на системе BIX BI (Business Intelligence), которую и будем использовать дальше, при создании OLAP куба.

Выбрали мы именно эту систему, потому что с ее помощью, можно всю необработанную информацию преобразовать в удобную и понятную аналитику.

## 2 Построение аналитических кубов для OLAP – системы

### 2.1 Создание OLAP куба для организации «АвтоСталь»

Для создания куба, мы будем использовать базу данных NWTraders, потому что база очень удобна в использование и имеет небольшой объем.

Чтобы создать OLAP куб нам необходимо войти в Microsoft Visual Studio 2010. После входа, нужно выбрать новый проект, а в качестве шаблона выбрать аналитические сервисы. После этого мы оказываемся в Solution Explorer, для создания проекта, необходимо выбрать источник данных (Data Sources), на основе которых будет строиться наш куб. После того, как мы создали источник данных, нужно создать представления данных (Data Sources Views), представления данных – это таблицы или запроса, с помощью которой будут создаваться измерения, которые являются основами для построения куба.

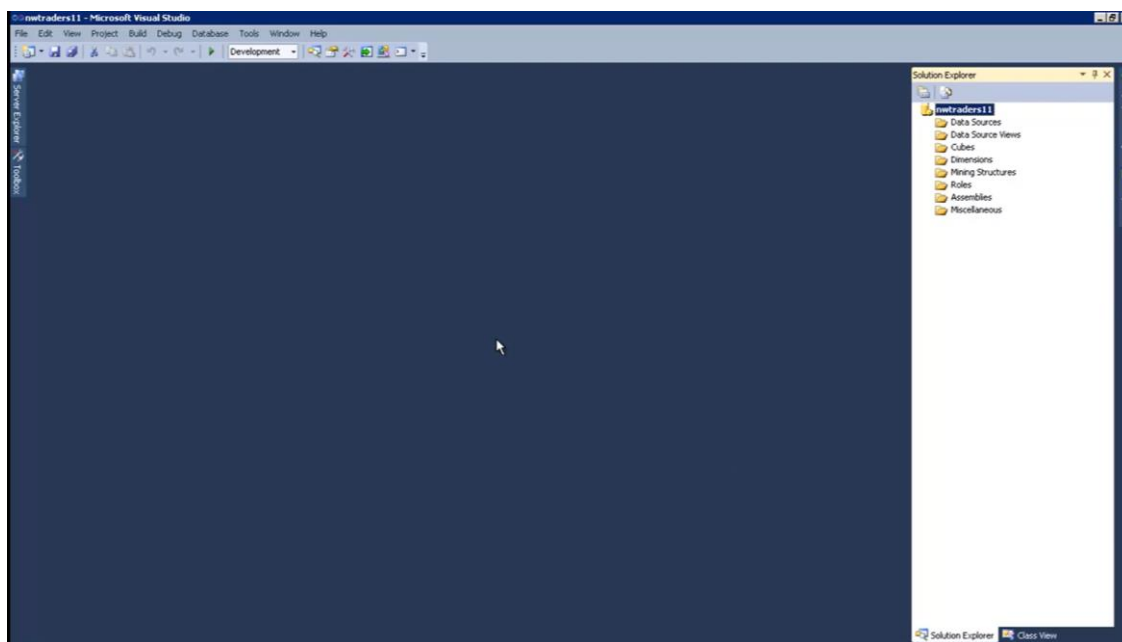


Рисунок 8 – Переход в Solution Explorer

Дальше мы создаем источник, в источнике нужно указать таблицы, которые мы будем использовать в кубе. Задействуем таблицы: категории товаров, товары, заказчик, заказы, подробности заказов. В дальнейшем мы настроим связи между этими таблицами.

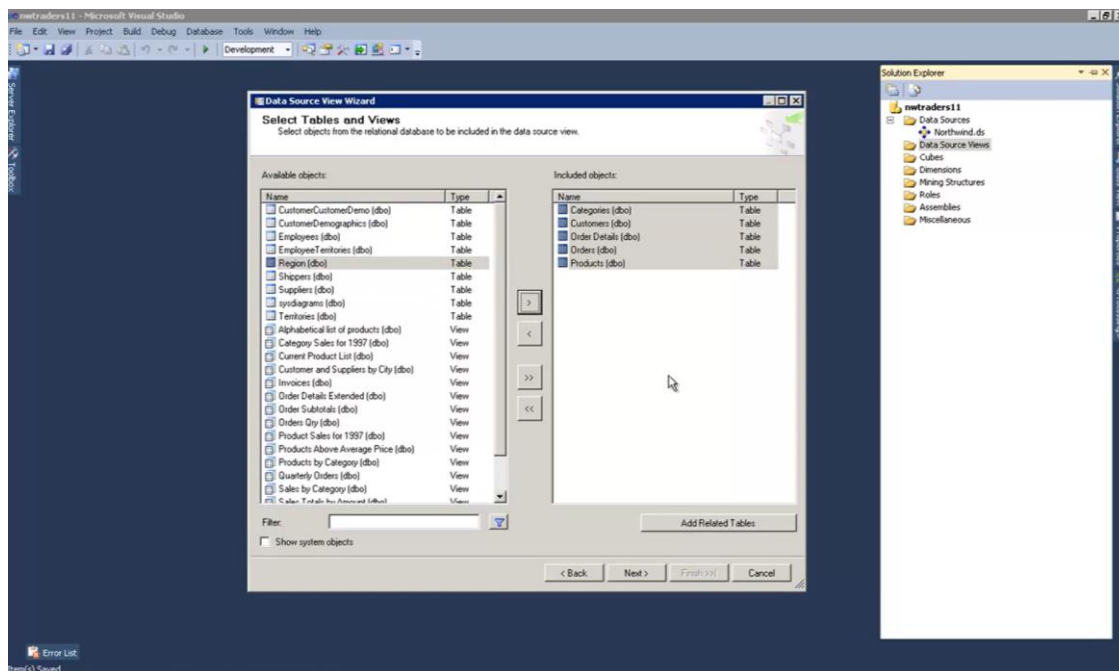


Рисунок 9 – Выбор таблиц для создания источника

На схеме данных мы можем увидеть, что:

- У нас есть товары, которые сгруппированы по категориям;
- Есть заказы, которые заказывают заказчики;
- И чтобы узнать какие товары, в каких заказах содержатся, для этого у нас есть таблица Order Details.

Из таблиц категории и продукты, мы будем делать измерение товаров.

Из таблицы заказчиков, мы сделаем измерение заказчиков.

А таблицы Order Details и Orders, будут использоваться в качестве фактов. Фактом у нас будет, факт продаж, у каждого факта есть атрибуты, а точнее – это дата заказа, заказчик и товар, который мы продаем заказчику.

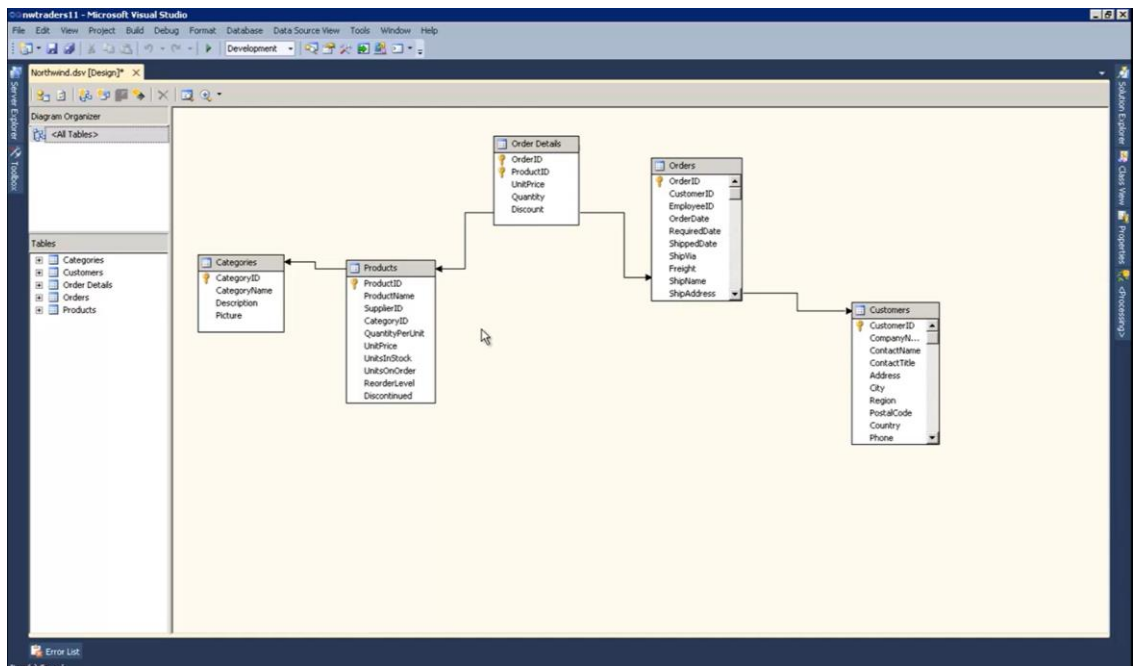


Рисунок 10 – Связи в схеме данных

При помощи измерений мы сможем делать разрезы или срезы выборок по определенным товарам или по определенным категориям. Для этого нужно в разделе Dimensions создать новое измерение. Чтобы создать новое измерение, необходимо будет указать ту таблицу, которая будет иметь первичный ключ. В нашем случае первичный ключ, это ключ товара, с помощью которого будем анализировать свои заказы. На следующем шаге, нам нужно добавить те столбцы, которые мы будем использовать в созданных измерениях, поэтому добавим название товара и название категорий.

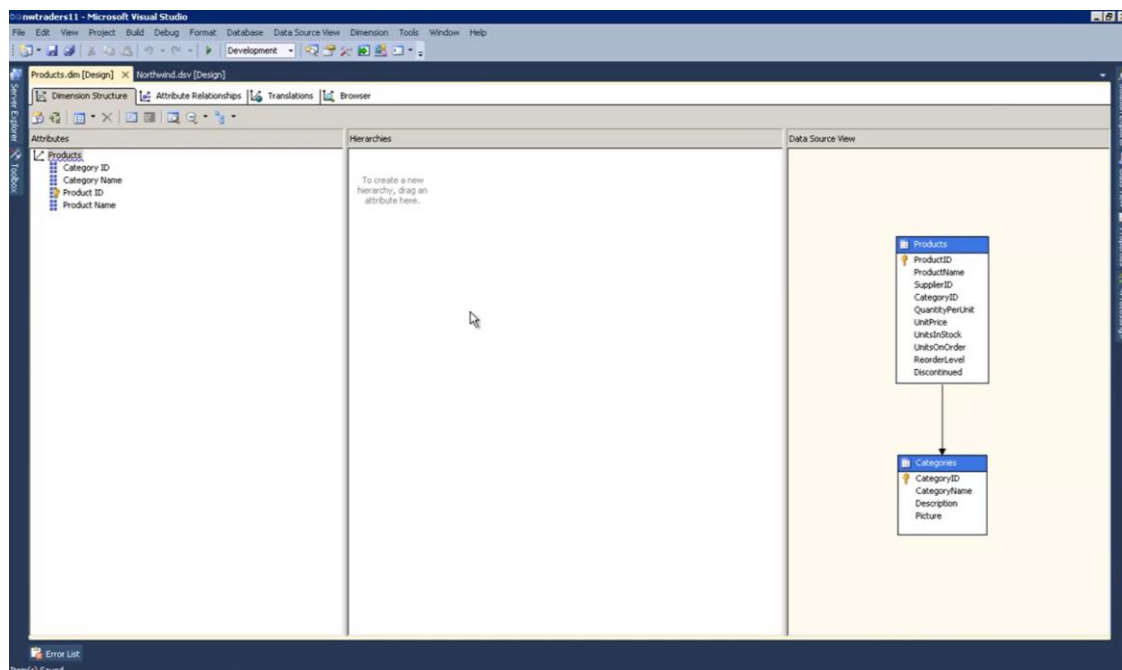


Рисунок 11 – Готовое измерение

В этом окне слева находятся атрибуты, которые можно использовать для построения измерения, в правой части отображается структура таблиц, а в середине находится иерархия.

Когда мы создавали измерения, мы не делали иерархию, поэтому нужно ее создать. При построении будем использовать двухуровневую структуру. Верхним уровнем этой структуры будут являться названия категорий, вторым уровнем структуры будут являться названия товара. Выбор подобной иерархии обусловлен тем, чтобы при анализе продаж была возможность сначала выбрать категорию и проанализировать данные продаж по всей категории или по нескольким. А затем так же можно спуститься на более низкий уровень и выполнить анализ по отдельным продуктам. Такая детализация называется *drilling down*, потому что мы понижаем уровень анализа.

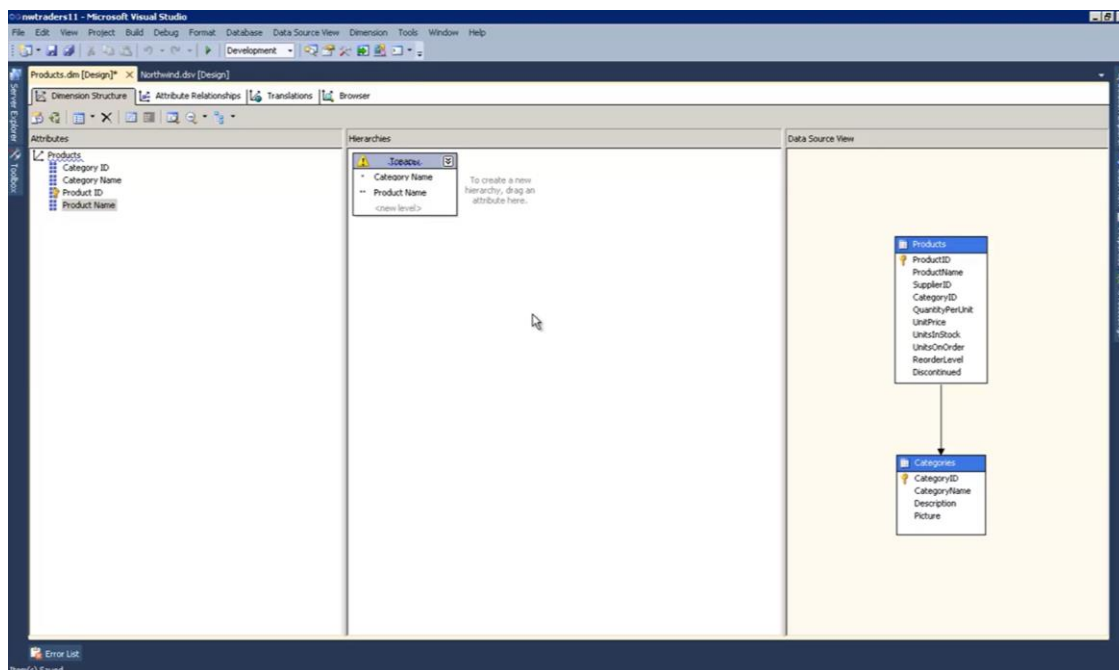


Рисунок 12 – Создание иерархии

Теперь с помощью браузера мы можем сделать предварительный просмотр измерения, как измерение будет выглядеть для заказчика.

Т.к. у нас сделано только одно измерение, значит уже можно создать куб. Чтобы создать куб, нужно в Solution Explorer открыть папку Cubes, после этого указывается таблица с группами мер, которая будет в дальнейшем подвергнута анализу. Поэтому выбираем таблицу Order Details, которую в дальнейшем и будем анализировать. В следующем окне, мы выбираем столбцы: цена и количество товаров, которые присутствуют в заказе, для анализа объема продаж. Дальше нужно выбрать то измерение, которое будем использовать, у нас выбора нет, потому что мы создали только одно измерение. В следующем окне, если нужно, можно создать еще одно измерение.



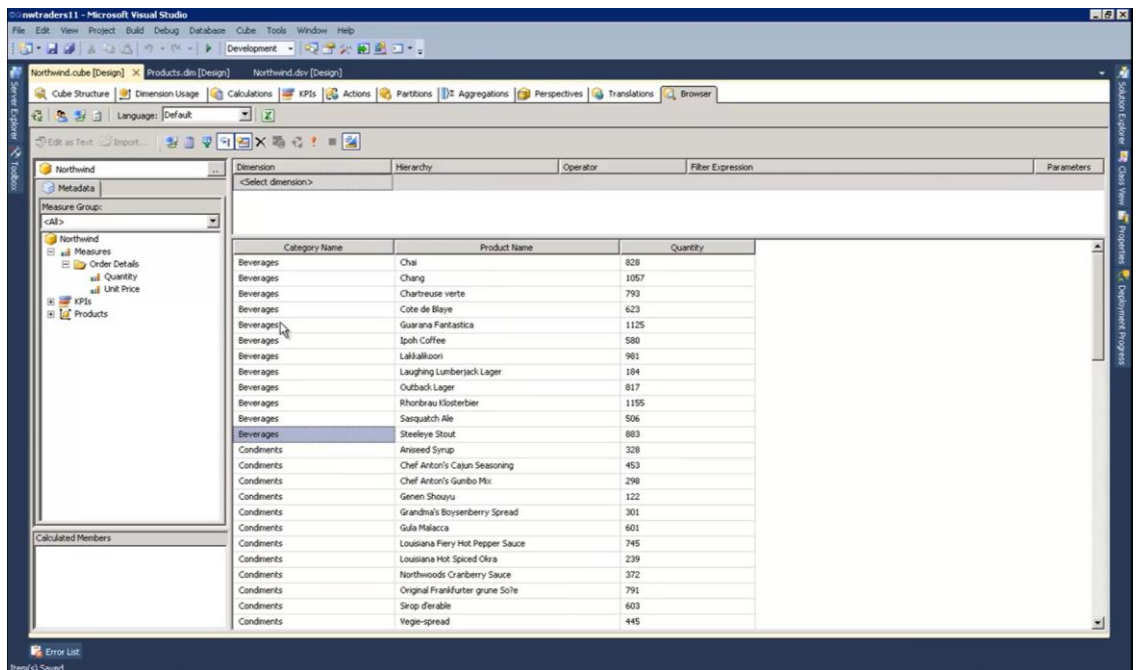


Рисунок 13 – Развернутый куб

В результате получился куб, где первый столбик – категории, второй – название измерения и третий столбик – суммарное количество проданных товаров.

Для пользователей Microsoft предлагает использовать Microsoft Excel для анализа. Excel можно открыть сразу же из браузера.

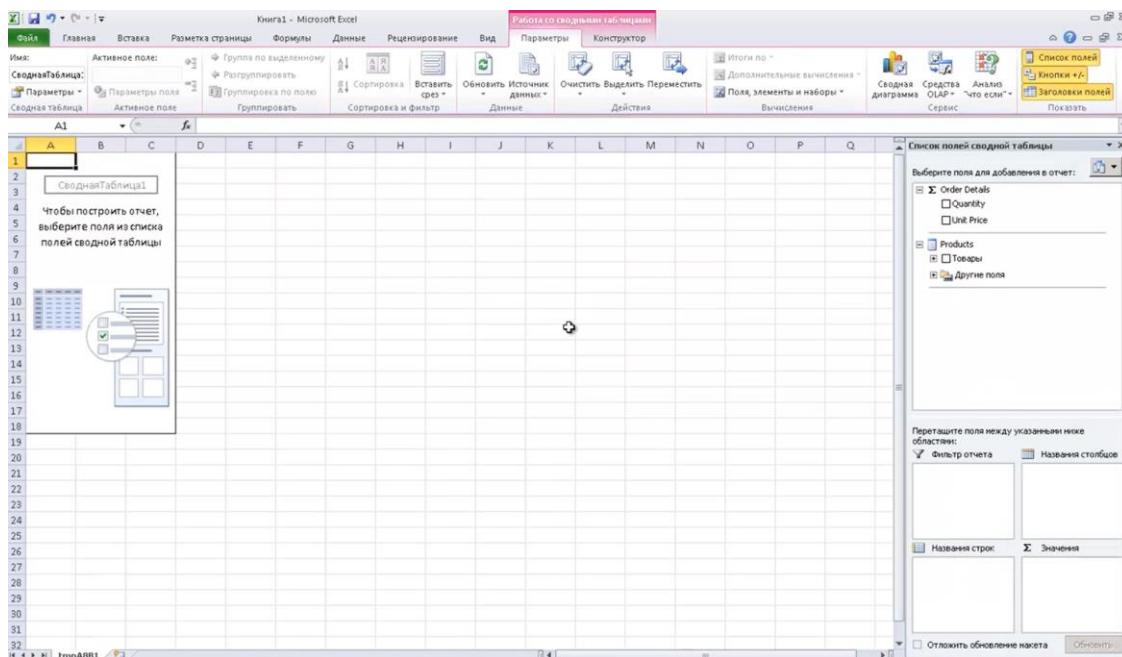


Рисунок 14 – Переход в Microsoft Excel

После перехода в Microsoft Excel мы добавляем два столбца, первый – группа товаров, второй – количество. Благодаря этим двум столбцам мы сможем проанализировать, какое количество каждого товара было продано. Чтобы тщательнее проанализировать каждый товар, необходимо выбрать нужную категорию, нажав на которую появится развернутая таблица, по которой мы сможем произвести анализ. В каждой группе можно сделать сортировку по названию товара, а так же по количеству проданных товаров, с помощью сортировки можно произвести анализ проданных товаров.

Название строки	Quantity
<b>Beverages</b>	<b>9532</b>
Chai	828
Chang	1057
Chartreuse verte	793
Cote de Blaye	623
Guarana Fantastica	1125
Iphoh Coffee	580
Lakkalikoori	981
Laughing Lumberjack Lager	184
Outback Lager	817
Rhonbrau Klosterbier	1155
Sasquatch Ale	506
Steeleye Stout	883
<b>Condiments</b>	<b>5298</b>
<b>Confections</b>	<b>7906</b>
<b>Dairy Products</b>	<b>9149</b>
<b>Grains/Cereals</b>	<b>4562</b>
<b>Meat/Poultry</b>	<b>4199</b>
<b>Produce</b>	<b>2990</b>
<b>Seafood</b>	<b>7681</b>
<b>Общий итог</b>	<b>51317</b>

Рисунок 15 – Анализ проданных товаров

Одномерный куб у нас готов, теперь можно добавить еще одно измерение. Например, можно добавить измерение даты. Чтобы добавить дату, необходимо вернуться в схему данных, где создавались связи и там выбрать пункт в таблице, который далее будем использовать как дату. Но есть некоторые особенности, которые не позволяют сделать стандартное измерение «дата», которые предлагают Microsoft. Поэтому можно сделать немного иначе, в созданной ранее таблице Orders, есть пункт OrderDate, то есть дата заказа. Мы и будем использовать это поле для измерения «дата», формат которого содержит год, месяц и день.

Для этого нам нужно сделать небольшой запрос, благодаря которому, мы сможем создать новую таблицу.

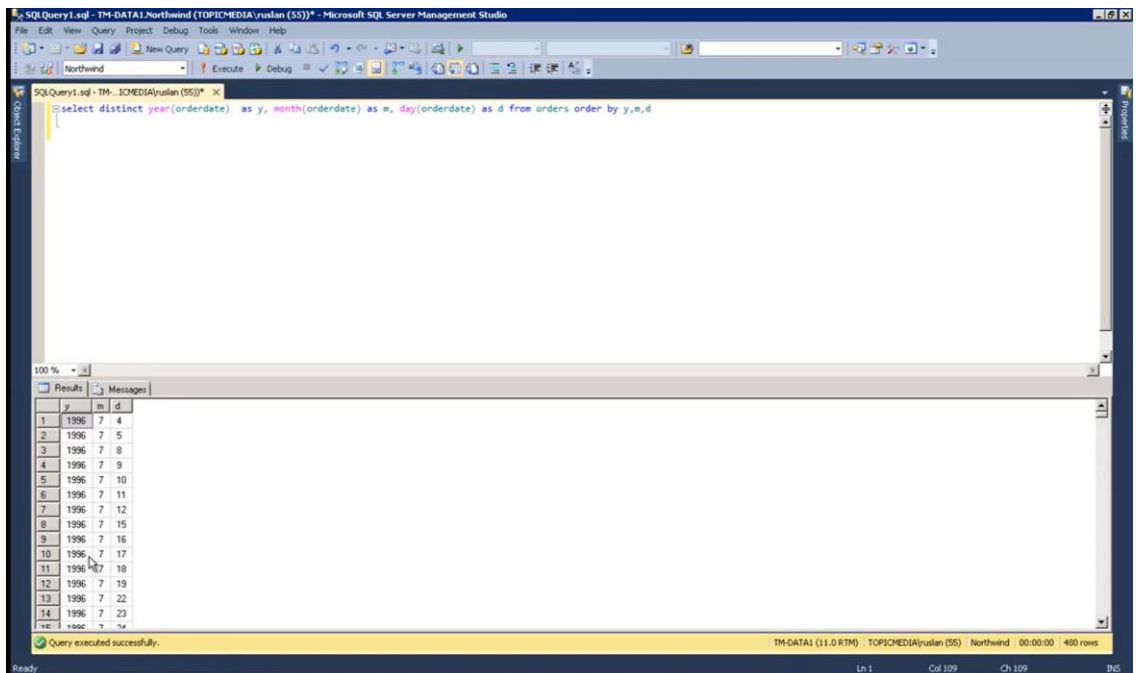


Рисунок 16 – Запрос для создания новой таблице

Теперь этот запрос нужно добавить в строку именованный запрос.

В результате появилась новая таблица OrderDate, где присутствуют поля: y(год), m(месяц), d(день) и OrderDate. Поле OrderDate создано, чтобы появилась связь между таблицами Orders и OrderDate. Связь между этими двумя таблица будет происходить по ключу OrderDate, поэтому их надо связать друг с другом. В итоге в таблице OrderDate, в поле OrderDate появился первичный ключ.

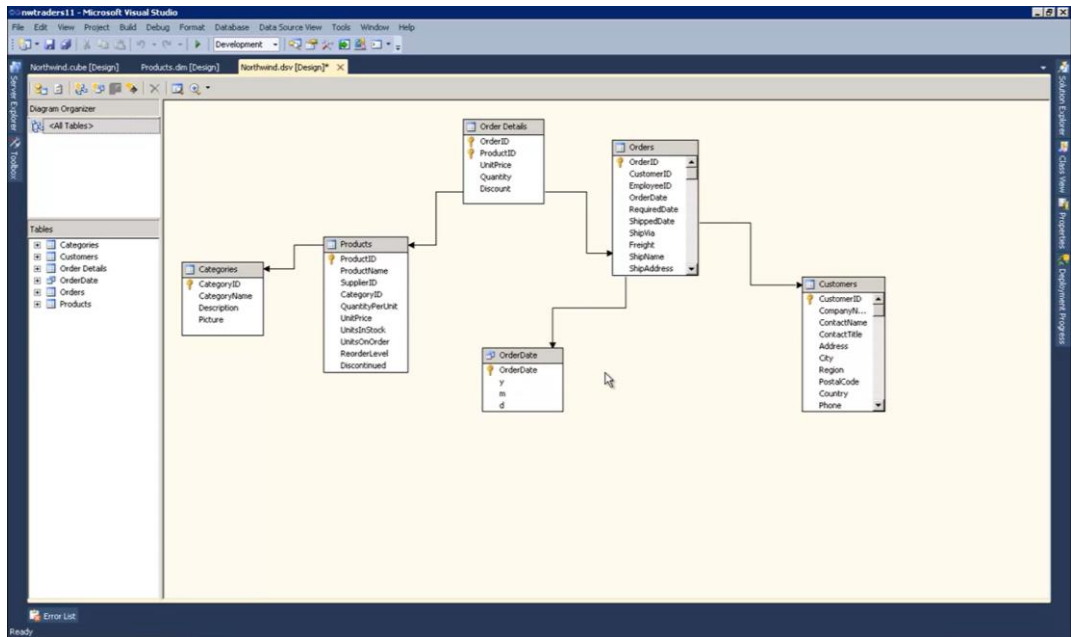


Рисунок 17 – Получение связи между таблицами Orders и OrderDate

Дальше создадим еще одно измерение, где укажем, что оно будет на основе существующей таблицы OrderDate. В таблице OrderDate, в ключевой строке нужно выбрать OrderDate, а так же в этом измерение необходимо включить строки: год, месяц и день. После этого мы строим иерархию, где у нас идет сверху вниз: год, месяц, день.

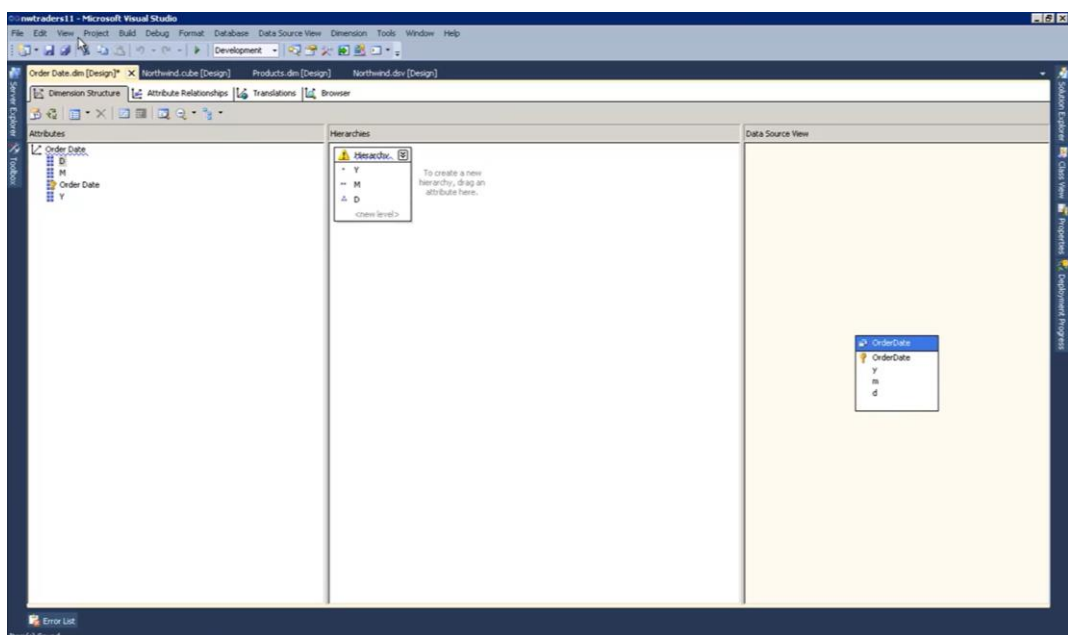


Рисунок 18 – Построение иерархии для таблицы OrderDate

Дальше нужно привязать созданное измерение к кубу, как же это сделать? Нужно добавить новое измерение к кубу, но измерение не связывается, потому как, факты лежат в таблице Order Details, а измерение связано с таблицей Orders. Чтобы решить этот вопрос, нужно добавить новое измерение, которое будет использовать только таблицу Orders, где ключом этого измерения будет OrderID, то есть по этому полю измерения связывается с таблицей фактов. В качестве дополнительного поля, по которому будет выполняться поиск, в этом измерение будет являться поле OrderDate. В созданном нами измерение не надо делать иерархию, так как пользователи не будут пользоваться этим измерением.

Теперь мы снова заходим в куб и добавляем еще одно измерение, и автоматически настраивается связь между всеми таблицами.

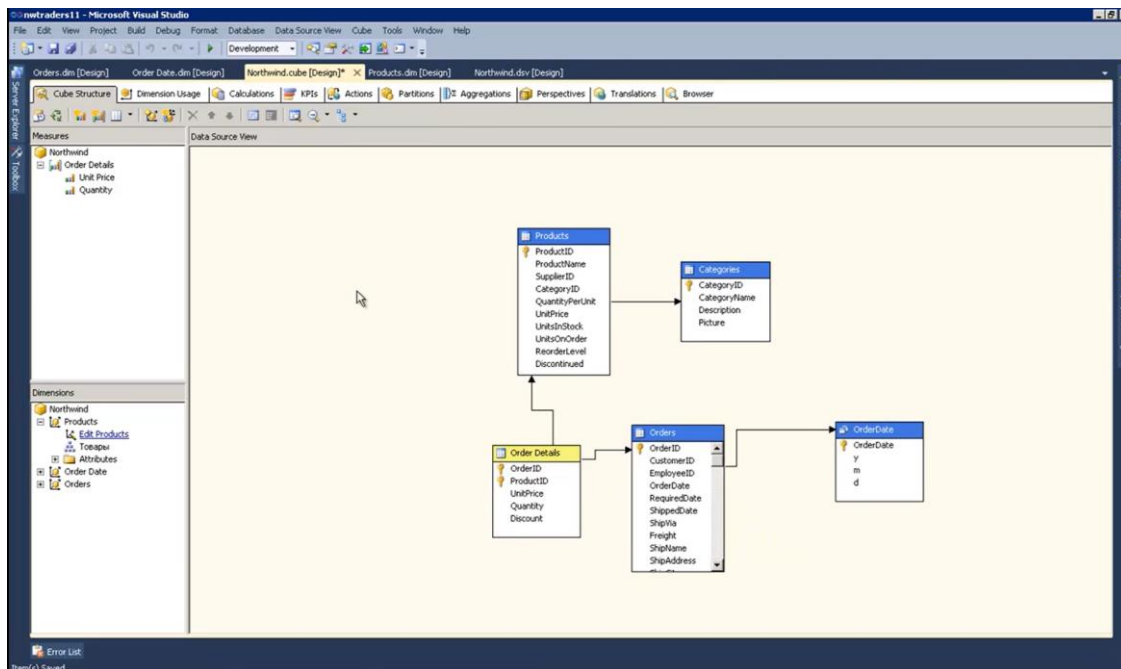


Рисунок 19 – Появление связей между таблиц в кубе

Чтобы проверить, корректно ли новое измерение привязалось к фактам, необходимо зайти на вкладку Dimension Usage и по этой вкладке можно увидеть, что измерение OrderDate никак не связано каким-нибудь условием с

группой мер. В пустом поле, где нет никаких связей, нужно указать, что эта связь настроена вручную и нужно указать тип отношений Referenced. Тип отношений Referenced – это означает, что мы будем связываться через промежуточное измерение. Промежуточное измерение у нас будет являться Orders, атрибут измерения OrderDate и атрибут промежуточного измерения тоже будет OrderDate. Именно по этим полям будет сопоставляться два этих измерения.

Теперь мы можем построить итоговый куб. Для этого нам нужно перейти в браузер, обновить все добавленные измерения и после, мы можем зайти в Excel и собрать куб. В название строк мы переносим товар, в название столбцов – даты, а в значение переносим количество. И после всех переносов, мы получаем готовый куб.

Quantity	1996					1997					1998					Итого	Общий итог
Название строк																	
Beverages	1042	3996	622	834	925	1092	221	3694	9532								
Chai	125	304	84	90	81	104	40	399	828								
Chang	226	495	62	61	10	201	62	396	1057								
Chartreuse d'arte	266	283	59	133	30	22	244	793									
Cote de Blaye	140	223	70	100	65	25	260	623									
Guarana Fantastica	158	421	40	146	200	140	20	546	1125								
Ipho Coffee	136	258	41	49	60	36	186	580									
Lakkalikoori	146	447	91	10	139	146	2	388	981								
Laughing Lumberjack Lager	5	65	30	64	20		114	184									
Outback Lager	156	413	96	83	69		248	817									
Rhonbrau Klosterbier	120	630	20	134	217	30	4	405	1155								
Sasquatch Ale	90	171	10	20	180	35	245	506									
Steeleye Stout	274	346	59	37	50	117	263	883									
Condiments	962	2895	269	289	378	446	59	1441	5298								
Confections	1357	4137	483	432	880	438	179	2412	7906								
Dairy Products	2086	4374	622	446	581	1005	35	2689	9149								
Grains/Cereals	549	2636	544	211	189	297	136	1377	4562								
Meat/Poultry	950	2189	133	302	92	405	128	1060	4199								
Produce	549	1583	58	37	351	367	45	858	2990								
Seafood	1286	3679	735	564	669	630	118	2716	7681								
Общий итог	9581	25489	3466	3115	4065	4680	921	16247	51317								

Рисунок 20 – Готовый куб

После построения куба, мы можем посмотреть и провести требуемый анализ для предприятия.



## 2.2 Реализация хранилища данных

Чтобы мы смогли получить доступ к нашим данным, нам необходимо создать хранилище данных.

Для создания хранилища данных, необходимо реализовать физическую модель данных с помощью средств языка Transact-SQL. Так как нет необходимости обрабатывать данные в реальном времени, поэтому не требуется создавать журнал транзакций. Можно использовать простой режим восстановления, этот режим не требует больших требований для хранения данных в журнале, на рисунке 21, показан код создания хранилища данных.

```
CREATE DATABASE EkranDW
ON PRIMARY
(
  NAME = N'EkranDW_fg1', FILENAME = N'C:\sqlserver\disk\data1\EkranDW_fg1.mdf',
  SIZE = 5 GB, FILEGROWTH= 1 GB),
FILEGROUP EkranDW_fg2
(NAME='EkranDW_fg2', FILENAME = N'C:\sqlserver\disk\data2\EkranDW_fg2.ndf',
SIZE = 5 GB, FILEGROWTH= 1 GB),
FILEGROUP EkranDW_fg3
(NAME='EkranDW_fg3', FILENAME = N'C:\sqlserver\disk\data3\EkranDW_fg3.ndf',
SIZE = 5 GB, FILEGROWTH= 1 GB)
LOG ON (name = 'EkranDW_log'
, filename = N'C:\sqlserver\disk\data1\EkranDW_log.ldf'
, size = 1 GB, filegrowth = 512 MB)
go
alter database EkranDW set recovery simple
go
alter database EkranDW set auto_shrink off
go
alter database EkranDW set auto_create_statistics on
go
alter database EkranDW set auto_update_statistics on
go
```

Рисунок 21 – Код создания базы хранилища

После написания части кода, у нас создалась база хранилища данных, теперь необходимо создать таблицы измерений и фактов, которые в итоге будут использоваться в кубе. На рисунке 22, написан код по созданию измерений.



```
SQLQuery2.sql - D...PC\Darkoomen (52))* X
CREATE TABLE dim_date
(
  DateKey int not null,
  DateAltKey datetime not null,
  [DayOfMonth] int not null,
  [DayOfWeek] int not null,
  [DayName] nvarchar(15) not null,
  MonthOfYear int not null,
  [MonthName] nvarchar(15) not null,
  CalendarYear int not null,
  CalendarQuarter int not null,
  FiscalYear int not null,
  FiscalQuarter int not null,
  CONSTRAINT PK_DateKey PRIMARY KEY CLUSTERED (DateKey)
  ON EkranDW_fg2
) ON EkranDW_fg2
GO
CREATE UNIQUE INDEX DateAltKey
ON dim_date (DateAltKey)
ON EkranDW_fg2
go
```

Рисунок 22 – Код Создания измерений

В этом коде можно выделить несколько моментов:

1. Когда создавался первичный ключ (DateKey), был указан кластерный индекс ключа. Это сделано, для того чтобы производительность запросов была ускорена.
2. Были использованы файловые группы, которые позволили определить, где находится физическая таблица и хранится кластеризованный индекс первичного ключа.

Точно так же мы создаем и все оставшиеся таблицы измерений и фактов, которые нам пригодятся в кубе.

Для того чтобы передавать загрузочные данные, необходимо создать еще одно хранилище. Новое хранилище мы создаем точно так же, как показано на рисунке 21.

После создания пустого хранилища, туда нужно загрузить все исходные данные из старого хранилища, это можно сделать 2 способами:

ручным и автоматическим. Если делать через ручную загрузку, то используется код языка T-SQL.

```
SQLQuery1.sql - D...PC\Darkoomen (52))* ×
INSERT INTO dbo.Products
(ProductKey, ProductAltKey, ProductName, ProductDescription, ProductCategoryName)
SELECT P.ProductKey, P.ProductAltKey, P.ProductName, P.ProductDescription, P.ProductCategoryName
FROM EKRAN.dbo.Product AS P
```

Рисунок 23 – Ручная загрузка данных

После того, как программа была реализована, переходим в рабочее окно SSIS, где в дальнейшем мы и будем работать.

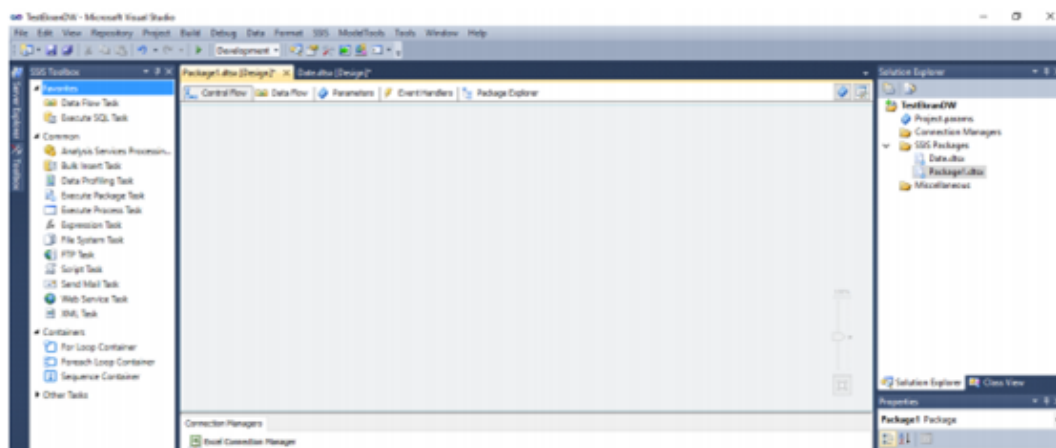


Рисунок 24 – Окно SSIS

В исходных данных есть «плоские» файлы из Excel, которые нужны нам для создания хранилища данных. Благодаря средствам SSIS, мы сможем перенести эти данные в созданное и готовое хранилище данных. На рисунке 25 показан готовый результат передачи данных в хранилище.

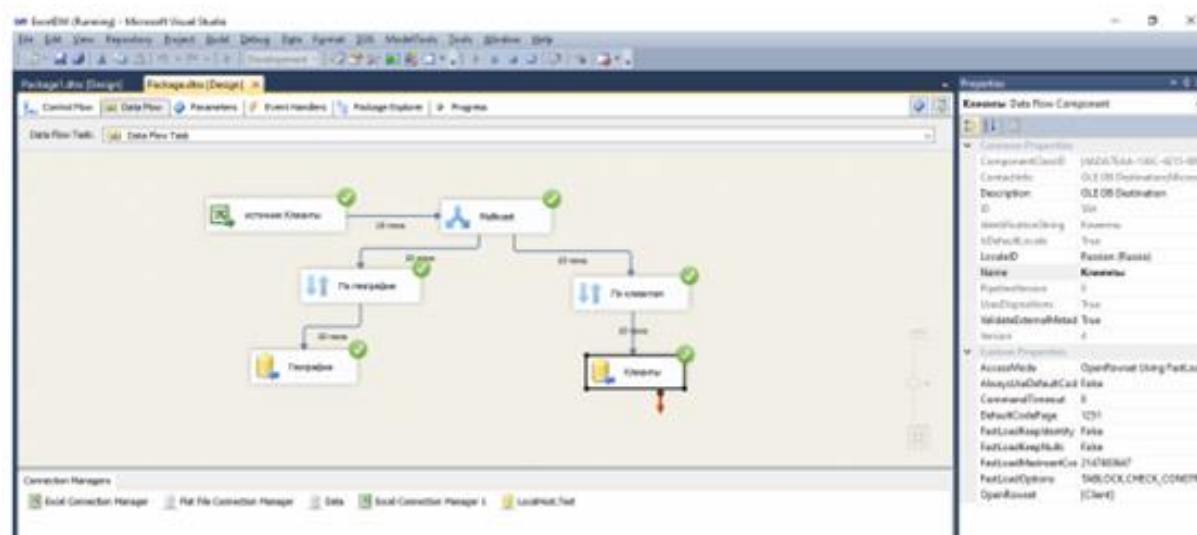


Рисунок 25 – Успешный результат передачи данных

### 2.3 Выводы по главе

Во второй главе был построен аналитический куб для системы OLAP, организация «АвтоСталь». В данном кубе сделано 3 измерения, двухуровневая иерархия и в итоге получился развернутый куб, благодаря которому можно произвести анализ движения товаров и увидеть в какие дни был пик продаж, и проанализировать причины увеличения продаж, а так же использовать это в дальнейшей деятельности предприятия. Для построения куба были сделаны таблицы, настроены связи между ними, добавили новое измерение, чтобы связывать имеющую таблицу и вновь созданную между собой.

В процессе работы был написан код, в котором было создано хранилище данных, описано создание измерения. Произведена ручная загрузка данных, после которой мы перешли в SSIS, где смогли перенести данные в созданное и готовое хранилище данных.

Реализация хранилища данных, была сделана при помощи языка Transact-SQL.

### 3 Проверка созданного OLAP куба

#### 3.1 Тестирование

Тестирование – это процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта [17, с. 43].

После того, как было создано и наполнено хранилище данных необходимо проверить работоспособность, а это значит, необходимо провести тестирование.

Тестирование хранилища данных проведем в два этапа. Первый этап проверки будет состоять из SQL-запросов. Второй этап будет состоять из создания тестового куба. Тестовый куб создадим с помощью средств MS SQL Server Analysis Services и проведем тестирование этого куба в клиентских приложениях. Во время тестирования необходимо проверить:

- Совместимость;
- Производительность запросов;
- Работоспособность средств создания отчетов;
- Проверка аппаратного устройства.

Посмотреть и проанализировать результат тестирования можно на рисунке 26.

Название теста	Проверяемый случай	Результаты теста
Проверка аппаратного устройства	Взаимодействие с пакетами прикладных программ	Взаимодействие осуществляется
Проверка совместимости	<u>Windows 7</u>	Совместима
	<u>Windows 8</u>	Совместима
Проверка работоспособности	<u>Windows 7</u>	Работоспособна
	<u>Windows 8</u>	Работоспособна
Проверка производительности	Скорость загрузки сформированного отчета в браузере	Время загрузки – 2 сек. (SQL Server Profiler)

Рисунок 26 – Таблица результата тестирования

Чтобы написать аналитические требования к кубу данных, можно использовать специализированный язык запросов. MDX – это язык запросов к многомерным структурам данных, этот язык используется IT-специалистами при создании и работе с отчетами и кубами.

Сделаем соответствующий запрос, чтобы узнать сколько товаров определенного типа было реализовано.

Product Name	Quantity
LG	281
Panasonic	639
Sony	631

Рисунок 27 – Выполнение MDX-запроса

В результате этого запроса, мы увидели, какое количество определенных товаров было реализовано. После запроса, мы сможем развернуть проект в специальном браузере. И так же сделать Webотчеты.

### 3.2 Выводы по главе

В последней главе мы сделали тестирование разработанного куба и хранилища данных. Во время проведения тестирования мы проверяли их на: совместимость, производительность запросов, работоспособность средств создания отчетов и делали проверку аппаратного устройства.

Был произведен MDX-запрос. В результате этого запроса, мы увидели, какое количество определенных товаров было реализовано. Чтобы продемонстрировать реализацию и сделать Webотчеты, необходимо будет развернуть на сервере.

## Заключение

В работе была рассмотрена разработка структуры и построение аналитических кубов для системы OLAP, описано создание куба OLAP для организации «АвтоСталь», разобрана внутренняя структура компонентов гиперкуба.

В данном кубе сделано 3 измерения, двухуровневая иерархия и в итоге получился развернутый куб, благодаря которому можно произвести анализ движения товаров и увидеть в какие дни был пик продаж, и проанализировать причины увеличения продаж, а так же использовать это в дальнейшей деятельности предприятия. Для построения куба были сделаны таблицы, настроены связи между ними, добавили новое измерение, чтобы связывать имеющую таблицу и вновь созданную между собой. В процессе работы был написан код, в котором было создано хранилище данных, описано создание измерения. Произведена ручная загрузка данных, после которой мы перешли в SSIS, где смогли перенести данные в созданное и готовое хранилище данных.

OLAP система одна из лучших технологий для организации, с ее помощью можно обрабатывать огромное количество информации и данных в реальном времени. Главные плюсы это, конечно же, высокая скорость и большой объем данных, а так же открытый доступ для всех пользователей.

Все пользователи могут выполнить комплексный анализ данных, так же пользователям могут помочь в решение аналитических задач, для этого сделаны аналитические приложения.

Для многих OLAP система будет полезна, особенно в областях: финансов, маркетинга, банков, медицина, наука, образование, розничной и оптовой торговли.

## Список используемой литературы

1. Аналитическая платформа Deductor Электронный ресурс. URL: <http://www.basegroup.ru/deductor/> (дата обращения: 25.05.2021)
2. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP. Спб.: БХВ–Петербург, 2017. – 384 с.
3. Гладченко А., Щербинин В. Репликация Microsoft SQL Server 2005/2008. –М.: ЭКОМ Паблишерз, 2019. – 288 с.
4. Дейт К.Дж. Введение в системы баз данных. – 8–е изд. – М.: Вильямс, 2018. – 1328 с.
5. Дорожкин А.К. Оценка объемов многомерного куба в OLAP системах // Научно–технический вестник Санкт–Петербургского государственного университета информационных технологий, механики и оптики, 2018. –111 с.
6. Зыкин С.В. Реализация OLAP–технологии на основе межмодельных преобразований данных // Информационные технологии моделирования и управления. 2018. – № 1 (выпуск 44). – 770 с.
7. Каширин И.Ю., Семченков С.Ю. Интерактивная аналитическая обработка данных в современных OLAP–системах // Журнал «Бизнес–информатика». Москва, 2019. № 8(02). – 148 с.
8. Макаров И.М., Рахманкулов В.З., Ахрем А.А., Ровкин И.О. Построение СППР на основе OLAP–технологии // Информационные технологии и вычислительные системы. 2019. – № 1. – 144 с.
9. Мейер Б. Объектно–ориентированное конструирование программных систем / Пер. с англ. М.: Издательско–торговый дом «Русская редакция», 2018. – 1232 с.
10. Плоткин Б.И. Универсальная алгебра, алгебраическая логика и базы данных–М.:Наука, 2018. – 448 с.

11. Послед Б.С. Vorland C++ Builder 6. Разработка приложений баз данных. –Спб.:ДиасофтЮП, 2019. – 320 с.
12. Семченков С.Ю. Вопросы организации промежуточной области хранения для OLAP систем // Математическое и программное обеспечение вычислительных систем. 2018.– 358 с.
13. Семченков С.Ю. Особенности применения OLAP систем: проблемы и актуальные подходы // Математическое и программное обеспечение вычислительных систем. 2018. – 897 с.
14. Семченков С.Ю. Принципы реализации иерархической структуры измерений в OLAP системах // Математическое и программное обеспечение вычислительных систем. 2017. – 471 с.
15. Харинатх С., Куинн С. SQL Server 2005 Analysis Services и MDX для профессионалов. – М.: Диалектика, 2018. – 848 с.
16. Чубукова И.А. Data Mining: учебное пособие. М.: БИНОМ, 2018. – 382 с.
17. Шилдт Г. Swing. Руководство для начинающих. М.: Вильямс, 2017. – 704 с.
18. Alkharouf N.W, Matthews B.F. The soybean genomics and microarray database. *Nucleic Acids Research*. 2004; 32:398–400.
19. Eastmond P.J, Li Y, Graham I.A. Is trehalose-6-phosphate a regulator of sugar metabolism in plants? *Exp Bot*. 2003; 54(582):533–537.
20. Jukic, N., “Data Modeling strategies and Alternatives for Data Warehousing Projects,” *Communications of the ACM*. 2006; 49(4):83-88.
21. Khan R, Alkharouf N, Beard H, et al. Resistance mechanisms in soybean: gene expression profile at an early stage of soybean cyst nematode invasion. *Nematology*. 2004; 36(3):241–248.
22. Puthoff D.P, Nettleton D, Rodermeel S.R, Baum T.J. *Arabidopsis* gene expression changes during cyst nematode parasitism revealed by statistical analyses of microarray expression profiles. *Plant*. 2003; 33(5):911–921.