**Содержание**

Введение 2

1. Основные теоретические сведения 3
2. СУБД MS SQL Server 2008 5
   1. Проектирование и создание базы 5
   2. Наполнение базы данными 9
3. Среда SQL Server Business Intelligence Development Studio 2008 10
   1. Создание проекта на основе базы 10
   2. Генерация представления BI – куба 16

Заключение 21

Список литературы 22

Введение

OLAP – куб как формат данных очень часто в современном мире используется как источник массива данных. Зачастую это объясняется высоким уровнем развития инструментов представления данных, например, web – обозревателями. Данные в веб-формате не зависят от операционных систем и от платформ. По этой причине, одним из основных способов взаимодествия с OLAP-кубами является язык ASP.NET.

Цель данной работы – продемострировать один из способов оформления данных для представления в www в виде OLAP.

Информационные системы масштаба предприятия, как правило, содержат приложения, применяемые менеджерами высшего звена и предназначенные для комплексного многомерного анализа данных, их динамики, тенденций и т.п. Такой анализ в конечном итоге призван способствовать принятию решений. Нередко такие системы так и называются — системы поддержки принятия решений. Указанные приложения обычно обладают средствами предоставления пользователю агрегатных данных для различных выборок из исходного набора в удобном для восприятия и анализа виде. Чаще всего такие агрегатные функции образуют многомерный (а следовательно, нереляционный) набор данных (нередко называемый гиперкубом или метакубом), оси которого содержат параметры, а ячейки — зависящие от них агрегатные данные. Вдоль каждой оси данные могут быть организованы в виде иерархии, отражающей различные уровни их детализации. Благодаря такой модели данных пользователи могут формулировать сложные запросы, генерировать отчеты, получать подмножества данных.

Технология комплексного многомерного анализа данных получила название OLAP (On-Line Analytical Processing). OLAP — ключевой компонент организации хранилищ данных (Data warehousing), то есть сбора, отсеивания и предварительной обработки данных с целью предоставления результирующей информации пользователям для статистического анализа (а нередко и для создания аналитических отчетов). Концепция OLAP была описана в 1993 году Э.Ф.Коддом, известным исследователем баз данных и автором реляционной модели данных.

В настоящее время поддержка OLAP реализована во многих СУБД и инструментах, так как является оптимальным решением для большого класса приложений, где пользователи сталкиваются с многомерными данными (то есть с данными, зависящими от нескольких параметров, например от времени, местоположения и других характеристик).

1. Основные теоретические сведения

Online analytical processing (OLAP) или аналитическая обработка в реальном времени - технология обработки данных, заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу. Основоположник термина OLAP — Эдгар Кодд, предложил в 1993 году «12 законов аналитической обработки в реальном времени».

OLAP-структура, созданная из рабочих данных, называется OLAP-куб. Куб создаётся из соединения таблиц с применением схемы звезды или схемы снежинки. В центре схемы звезды находится таблица фактов, которая содержит ключевые факты, по которым делаются запросы. Множественные таблицы с измерениями присоединены к таблице фактов. Эти таблицы показывают, как могут анализироваться агрегированные реляционные данные. Количество возможных агрегирований определяется количеством способов, которыми первоначальные данные могут быть иерархически отображены.

Например, все клиенты могут быть сгруппированы по городам или по регионам страны (Запад, Восток, Север и т. д.), таким образом, 50 городов, 8 регионов и 2 страны составят 3 уровня иерархии с 60 членами. Также клиенты могут быть объединены по отношению к продукции; если существуют 250 продуктов по 20 категориям, 3 группы продукции и 3 производственных подразделения, то количество агрегатов составит 16560. При добавлении измерений в схему количество возможных вариантов быстро достигает десятков миллионов и более.

OLAP-куб содержит в себе базовые данные и информацию об измерениях (агрегаты). Куб потенциально содержит всю информацию, которая может потребоваться для ответов на любые запросы. При огромном количестве агрегатов зачастую полный расчёт происходит только для некоторых измерений, для остальных же производится «по требованию».

Существуют три типа OLAP:

* многомерная OLAP (Multidimensional OLAP — MOLAP);
* реляционная OLAP (Relational OLAP — ROLAP);
* гибридная OLAP (Hybrid OLAP — HOLAP).

MOLAP — это классическая форма OLAP, так что её часто называют просто OLAP. Она использует суммирующую БД, специальный вариант процессора пространственных БД и создаёт требуемую пространственную схему данных с сохранением как базовых данных, так и агрегатов.

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

HOLAP использует реляционные таблицы для хранения базовых данных и многомерные таблицы для агрегатов.

В данной работе используется форма MOLAP.

Режимы хранения:

Microsoft Analysis Services, используемый в данной работе, занимает нейтральную позицию в споре MOLAP против ROLAP, разгорающемся вокруг OLAP-продуктов. Благодаря этому можно использовать все виды MOLAP, ROLAP и HOLAP внутри одной модели.

Режимы раздельного хранения:

MOLAP - Multidimensional OLAP (многомерный OLAP) - Обрабатываются сами данные и их обработки, сохраняются и индексируются при помощи специального формата, оптимизированного под многомерные данные.

ROLAP - Relational OLAP (реляционный OLAP) - Данные и их обработки остаются в реляционном источнике данных, устраняется необходимость в специальной обработке.

HOLAP - Hybrid OLAP (гибридный OLAP) - этот режим использует реляционный источник данных для хранения самих данных, но предварительные обработки и индексы хранятся в специальном формате, оптимизированном для многомерных данных.

Режимы размерного хранения:

MOLAP - атрибуты размерности и иерархии обрабатываются и хранятся в специальном формате

ROLAP - атрибуты размерности не обрабатываются и остаются в реляционном источнике данных. Разделы, измеряемые размерностью ROLAP, также должны быть в режиме ROLAP.

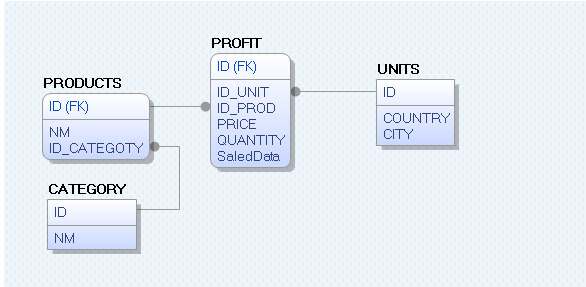
1. СУБД MS SQL Server 2008

2.1 Проектирование и создание базы

По условию задачи необходимо реализовать справочники данных:

* Товары
* Представительства компании
* Время

Для данных целей необходимы таблицы



CATEGORY – категории товаров

PRODUCTS – наименования товаров

UNITS – территории

PROFIT – «связка» территорий, товаров, цены, количества проданного, даты продажи.

Для создания данных таблиц используем следующие скрипты:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

USE [tcorp]

GO

/\*\*\*\*\*\* Object: Table [dbo].[PRODUCTS] Script Date: 12/02/2014 14:08:13 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[PRODUCTS](

[ID] [int] IDENTITY(1,1) NOT NULL,

[NM] [varchar](100) NOT NULL,

[ID\_CATEGORY] [int] NOT NULL,

CONSTRAINT [PK\_PRODUCTS] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

ALTER TABLE [dbo].[PRODUCTS] WITH CHECK ADD CONSTRAINT [FK\_PRODUCTS\_CATEGORY] FOREIGN KEY([ID\_CATEGORY])

REFERENCES [dbo].[CATEGORY] ([ID])

GO

ALTER TABLE [dbo].[PRODUCTS] CHECK CONSTRAINT [FK\_PRODUCTS\_CATEGORY]

GO

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

USE [tcorp]

GO

/\*\*\*\*\*\* Object: Table [dbo].[CATEGORY] Script Date: 12/02/2014 14:08:40 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[CATEGORY](

[ID] [int] IDENTITY(1,1) NOT NULL,

[NM] [varchar](100) NOT NULL,

CONSTRAINT [PK\_CATEGORY] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

USE [tcorp]

GO

/\*\*\*\*\*\* Object: Table [dbo].[UNITS] Script Date: 12/02/2014 14:09:54 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[UNITS](

[ID] [int] IDENTITY(1,1) NOT NULL,

[COUNTRY] [varchar](100) NOT NULL,

[CITY] [varchar](100) NOT NULL,

CONSTRAINT [PK\_UNITS] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

USE [tcorp]

GO

/\*\*\*\*\*\* Object: Table [dbo].[PROFIT] Script Date: 12/02/2014 14:10:07 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[PROFIT](

[ID] [int] IDENTITY(1,1) NOT NULL,

[ID\_UNIT] [int] NOT NULL,

[ID\_PROD] [int] NOT NULL,

[PRICE] [int] NOT NULL,

[QUANTITY] [int] NOT NULL,

[SaledDate] [date] NULL,

CONSTRAINT [PK\_PROFIT] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[PROFIT] WITH CHECK ADD CONSTRAINT [FK\_PROFIT\_PRODUCTS] FOREIGN KEY([ID\_PROD])

REFERENCES [dbo].[PRODUCTS] ([ID])

GO

ALTER TABLE [dbo].[PROFIT] CHECK CONSTRAINT [FK\_PROFIT\_PRODUCTS]

GO

ALTER TABLE [dbo].[PROFIT] WITH CHECK ADD CONSTRAINT [FK\_PROFIT\_UNITS] FOREIGN KEY([ID\_UNIT])

REFERENCES [dbo].[UNITS] ([ID])

GO

ALTER TABLE [dbo].[PROFIT] CHECK CONSTRAINT [FK\_PROFIT\_UNITS]

GO

* 1. Наполнение базы данными

Ниже приведены примеры данных, содержащихся в указанных таблицах

**PRODUCTS:**

SELECT [ID]

,[NM]

,[ID\_CATEGORY]

FROM [tcorp].[dbo].[**PRODUCTS**]

|  |  |  |
| --- | --- | --- |
| **ID** | **NM** | **ID\_CATEGORY** |
| 1 | молоко буренка | 1 |
| 2 | кефир ряженка | 1 |
| 3 | йогурт Мданон | 1 |
| 4 | молоко зеленый клевер | 1 |
| 5 | конфеты Маска | 2 |
| 6 | леденцы Петя | 2 |
| 7 | карамель янтарь | 2 |
| 8 | зефир в шоколаде | 2 |
| 9 | спагетти | 3 |
| 10 | рожки | 3 |

……….

**CATEGORY:**

SELECT [ID]

,[NM]

FROM [tcorp].[dbo].[CATEGORY]

|  |  |
| --- | --- |
| **ID** | **NM** |
| 1 | молочная продукция |
| 2 | кондитерские изделия |
| 3 | макаронные изделия |
| 4 | хлебобулочная продукция |
| 5 | напитки |
| 6 | жвачка |
| 7 | мороженое |
| 8 | овощи |

**UNITS:**

SELECT [ID]

,[COUNTRY]

,[CITY]

FROM [tcorp].[dbo].[UNITS]

|  |  |  |
| --- | --- | --- |
| **ID** | **COUNTRY** | **CITY** |
| 1 | Россия | Санкт-Петербург |
| 2 | Россия | Уфа |
| 3 | Россия | Пермь |
| 4 | Беларусь | Минск |
| 5 | Украина | Киев |
| 6 | Казахстан | Астана |
| 7 | Россия | Улан-Удэ |
| 8 | Россия | Владивосток |
| 9 | Украина | Ровно |
| 10 | Россия | Екатеринбург |

**PROFIT:**

SELECT [ID]

,[ID\_UNIT]

,[ID\_PROD]

,[PRICE]

,[QUANTITY]

,[SaledDate]

FROM [tcorp].[dbo].[PROFIT]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **ID\_UNIT** | **ID\_PROD** | **PRICE** | **QUANTITY** | **SaledDate** |
| 1 | 1 | 1 | 45 | 70 | 01.04.2008 |
| 2 | 2 | 1 | 46 | 70 | 01.10.2000 |
| 3 | 3 | 1 | 52 | 70 | 01.12.2001 |
| 4 | 4 | 1 | 45 | 70 | 01.02.2001 |
| 5 | 5 | 1 | 50 | 70 | 01.06.2002 |
| 6 | 6 | 1 | 49 | 70 | 01.11.2000 |
| 7 | 7 | 1 | 48 | 70 | 01.02.2004 |

1. Среда SQL Server Business Intelligence Development Studio 2008
   1. Создание проекта на основе базы

На основе приведенной модели согласно плану задания требуется осуществить построение куба с выделением 3-х измерений. Мерами для данных измерения будут являться *количество* и *цена* товаров.

Источником данных является созданная база данных

Представления – все таблицы из базы данных и «именованный запрос» (named query) для создания измерения времени (календарный справочник).

Запрос:

SELECT DISTINCT

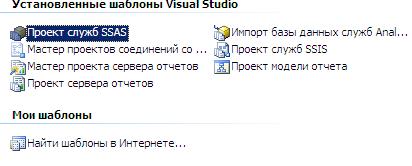
SaledDate

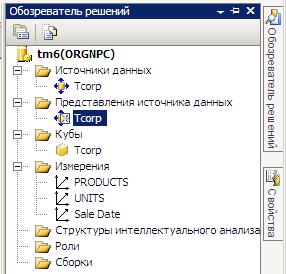
, YEAR(SaledDate) AS y

, MONTH(SaledDate) AS m

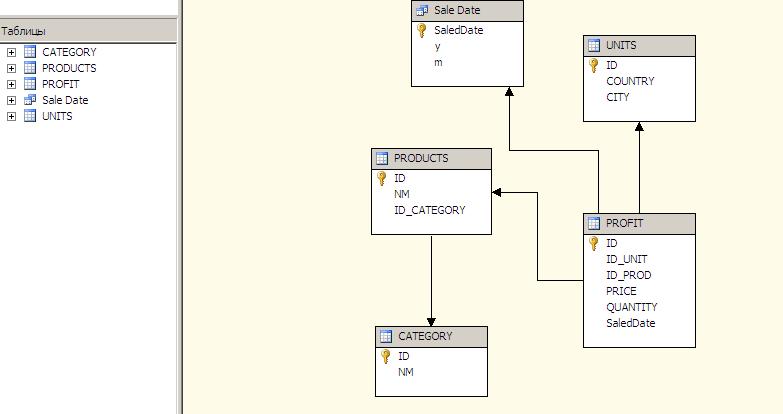
FROM PROFIT

Для построения куба необходимо в выбранной среде организовать измерения. Для этого вначале откроем BI – среду, создадим у ней указанной ниже проект



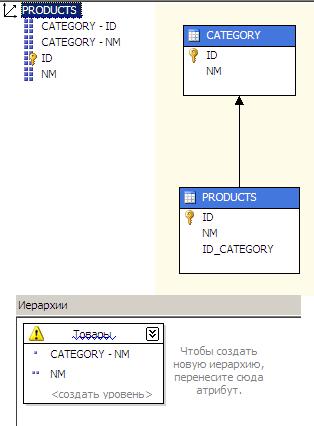


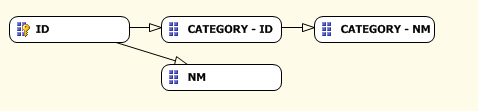
Затем, подключим нашу базу данный, созданную ранее в СУБД MS SQL Server 2008, в итоге получим подключенный к проекту источник данных и получим представление источника данных в виде диаграммы, приведенной ниже

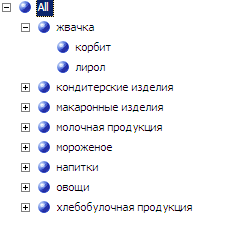


После подключения источника данных и получения его представления необходимо создать измерения. Для более наглядного представления была добавлена иерархия по виду товаров.

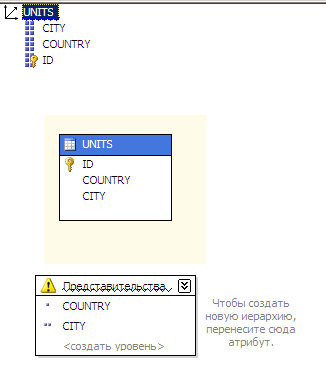
Измерение ТОВАРОВ

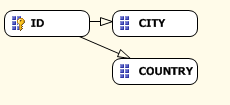


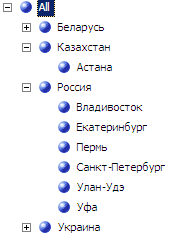




Измерение ПРЕДСТАВИТЕЛЬСТВ

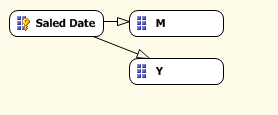


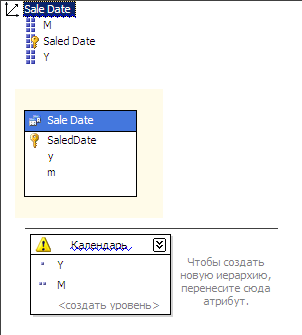




Измерение ВРЕМЕНИ

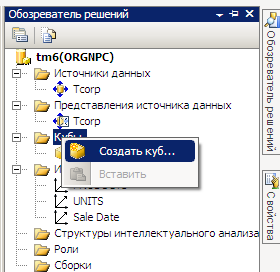






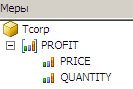
* 1. Генерация представления BI – куба

Операцией создания

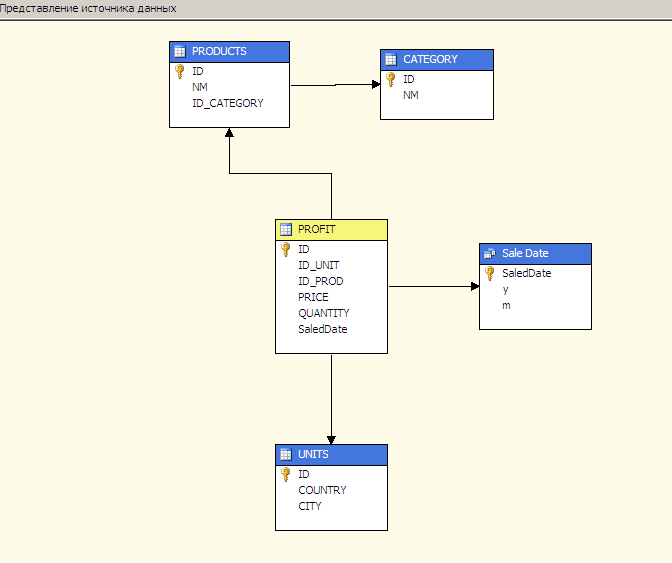
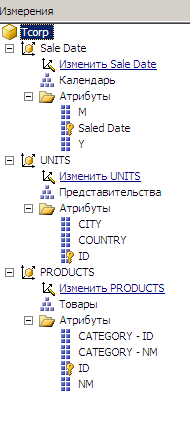


Создадим куб из полученных измерений пользуясь мастером создания

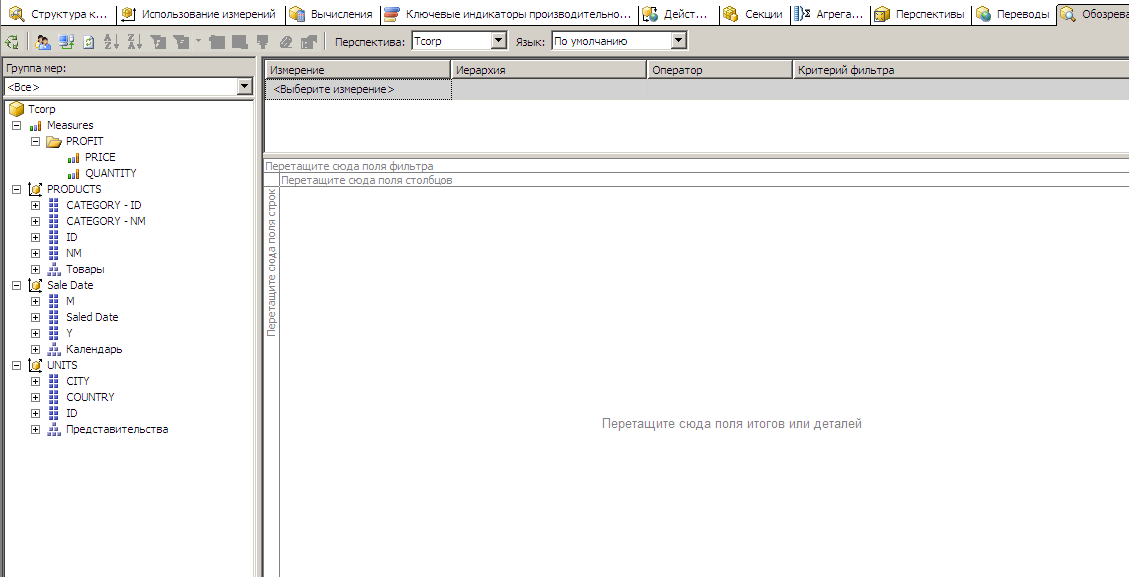
В качестве мер мы получим меры количества и цены. Данные меры изображены ниже



В панели измерений представлены ранее созданные измерения с атрибутами, расположенными иерархически



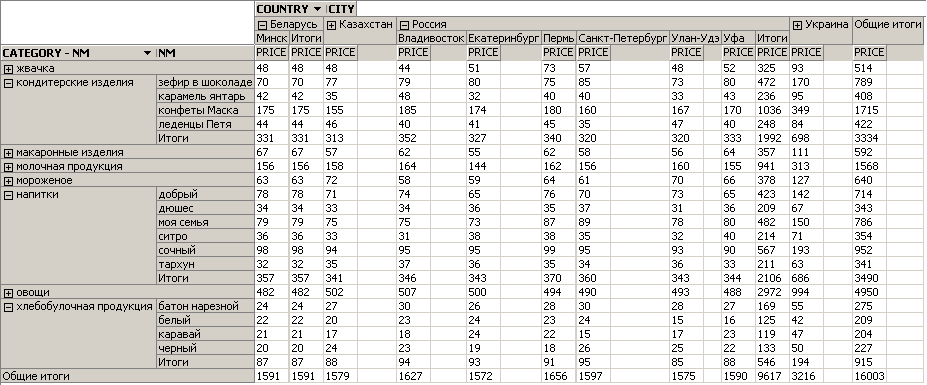
Итак, сгенерируем куб, путем переноса мер и измерений на необходимые строки и области



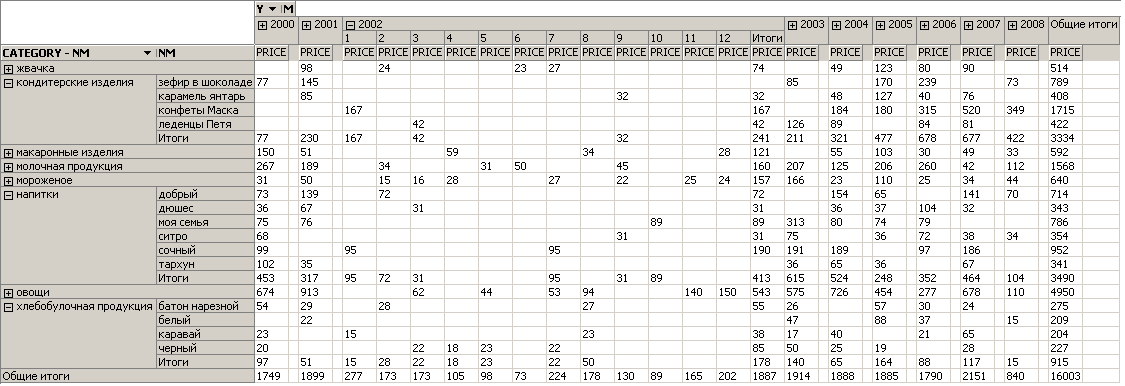
Полученный результат представлен ниже

Куб:

Измерения: Товары, Представительства. Мера: стоимость.



Измерения: Товары, Календарь. Мера: стоимость.



Измерение: Товары, Календарь, Представительства. Мера: стоимость, цена.



Заключение

OLAP - модель для более удобного представления в реально работающей организации чаще всего используют как веб-приложение, извлекая данные с помощью MDX запросов.

Полученный опыт работы с кубом позволил мне ближе подойти к реализации реально работающего примера (с целью демонстрации своих навыков) на ASP.NET.

Список литературы

1. http://ru.wikipedia.org/
2. http://habrahabr.ru/