

**УЗБЕКСКОЕ АГЕНТСТВО СВЯЗИ И ИНФОРМАТИЗАЦИИ  
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ**

На правах рукописи

**Хамидов Умиджон Хамзаевич**

**РАЗРАБОТКА И ВЕДЕНИЕ ХРАНИЛИЩА ДАННЫХ ДЛЯ СИСТЕМ  
ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ**

Специальность: 5А521902 - «Управление и обработка информации»

ДИССЕРТАЦИЯ  
на соискание степени магистра информационных технологий

Работа рассмотрена  
и допускается к защите.  
Зав. кафедрой ИТ

\_\_\_\_\_

Научный руководитель:

д.т.н. Гаипназаров Т.С. \_\_\_\_\_

«    » \_\_\_\_\_ 2012 г

«    » \_\_\_\_\_ 2012 г

Ташкент 2012

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ГЛАВА 1. АНАЛИЗ ОБЪЕКТА ИССЛЕДОВАНИЯ	9
1.1. Базовые понятия информационно-аналитических систем	9
1.2. Data Warehouse – хранилище данных – ХД - систем обработки данных	13
1.3. Подходы к архитектуре хранилищ данных – ХД	26
1.4. Анализ процесса принятий решений в системе высшего образования	34
1.5. Постановка задачи	36
1.6. Выводы по первой главе	37
ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА	39
ХРАНИЛИЩЕ ДАННЫХ	
2.1. Определение требований	39
2.2. Анализ вопросов	39
2.3. Определение структуры ХД	40
2.4. OLAP куб	49
2.5. Выводы ко второй главе	56
ГЛАВА 3. РЕАЛИЗАЦИЯ ПОДСИСТЕМЫ МНОГОМЕРНОГО АНАЛИЗА ДАННЫХ В СППР ДЛЯ УПРАВЛЕНИЯ ВНЕДРЕНИЯ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ В УЧЕБНЫЙ ПРОЦЕСС МВССО	57
3.1. Разработка хранилище данных для многомерного анализа	57
3.2. BI-решения Microsoft Analysis Services	72
3.3. Выводы по третьей главе	84
ЗАКЛЮЧЕНИЕ	85
ЛИТЕРАТУРА	86
ПРИЛОЖЕНИЕ	87

## Аннотация

Диссертационная работа посвящена разработке и ведению хранилища данных для систем поддержке принятий решений на примере образование. Проведено исследование технологий хранилище данных. Рассмотрен анализ процесса принятий решений в системе высшего образования. Объектом исследования в диссертационной работ является процесс принятия решения в Министерстве высшего и среднего специального образования на основе анализа данных полученных от подведомственных организации, а предметом исследования - модели систем поддержки принятия решений на основе многомерных моделей баз и хранилищ данных. В работе для решения задачи использованы методы основанные на методологии структурного анализа и проектирования (SADT - Structured Analysis & Design Technique), теории реляционных баз данных, технологии хранилищ данных и объектно-ориентированных методах построения программных систем.

Проведен обзор и анализ информационно-аналитических систем. Изучена технология хранилище данных. Приведены подходы к архитектуре хранилищ данных. Рассмотрено процесса принятий решений в системе высшего образования. Разработана постановка задачи создание и ведение хранилище данных для министерства высшего и среднего специального образования. Определено требования к разрабатываемому хранилищу данных. Проведен анализ вопросов и определено перечень данных, которые могут потребоваться для ответа на запрос пользователей. Предложено структура хранилища данных для анализа материально техническую базу высшего образовательного учреждений. Разработано хранилище данных для многомерного анализа. Проведен сравнительный анализ OLAP серверов. Для разработки хранилища данных было создан многомерный куб «Анализ показателей материально технической базы ВОУ» в Microsoft Analysis Services.

## **Abstract**

Dissertational work is devoted development and maintaining data warehouse given for systems to support of decision-making on an example education. Research of technologies data warehouse is carried out. The analysis of process of decision-making in system of the higher education is considered. Object of research in dissertational works is decision-making process in the Ministry of the higher and secondary vocational education on the basis of the analysis of data received from subordinated the organization, and an object of research - models of systems of support of decision-making on the basis of multidimensional models of data bases and data warehouse. In work for the solution of a task methods based on methodology of the structural analysis and design (by SADT - Structured Analysis & Design Technique), theories of relational databases, technologies of storages of data and object-oriented methods of creation of program systems are used.

The review and the analysis of information and analytical systems is carried out. The technology data warehouse is studied. Approaches are brought to architecture of data warehouse. It is considered process of decision-making in system of the higher education. Task statement creation and maintaining data warehouse is developed for the Ministry of Higher and Secondary Special Education. It is defined requirements to developed data warehouse. The analysis of questions is carried out and is defined the list of data which can be demanded for the answer to inquiry of users. It is offered structure of data warehouse for the analysis financially technical base of high educational universities. The data warehouse is developed for the multidimensional analysis. The comparative analysis of OLAP of servers is carried out. For development of data warehouse was the multidimensional cube «Analysis of the material and technical base of HEU » in Microsoft Analysis Services is created.

## **Mazmunnoma**

Dissertatsiya ishi qaror qabul qilish qo'llab quvvatlovchi tizimlar uchun ma'lumotlar omborini ishlab chiqish va joriy etish masalasiga bag'ishlangan. Ma'lumotlar ombori texnologiyasi tadqiq qilingan. Oliy ta'lim tizimida qaror qabul qilish jarayoni tahlili kurib chiqilgan. Dissertatsiya ishining tadqiqot ob'ekti Oliy va o'rta maxsus ta'lim vazirligida qaror qabul qilish jarayoni, tadqiqot predmeti ma'lumotlar ombori va bazasining ko'p ulchovli modellari asosida qaror qabul qilish tizimi modeli. Masalani yechishda strukturaviy tahlil va loyihalashtirish metodologiyasi (SADT - Structured Analysis & Design Technique), relyatsion ma'lumotlar bazasi nazariyasi, ma'lumotlar ombori texnologiyasi hamda ob'ektga yo'naltirilgan usullar dasturiy tizimlarni yaratish metodlaridan foydalanilgan.

Tahliliy axborot tizimlari tahlil amalga oshirilgan. Ma'lumotlar ombori texnologiyasi o'rganib chiqilgan. Ma'lumotlar ombori arxitekturasi turlari ko'rib chiqilgan. Oliy ta'lim vazirligida tahlil olib borish hamda qaror qabul qilish jarayoni o'rganilgan va mavjud holatda kamchiliklarni bartaraf etish hamda tahlil jarayonini yanada takomillashtirish bo'yicha takliflar ishlab chiqilgan. Ma'lumotlar omborini loyihalashtirishda foydalanuvchilar tomonidan qo'yiladigan talablar aniqlangan va belgilab olingan. Shuningdek ma'lumotlar omborida foydalanuvchilar so'roviga javob berish uchun kerak bo'ladigan ma'lumotlar ro'yhati ishlab chiqilgan, va tahlili amalga oshirilgan. Oliy ta'lim muassasalarining moddiy texnik bazasini tahlil qilish mo'ljallangan ma'lumotlar ombori strukturasi loyihalashtirilgan. Ushbu loyiha asosida ko'p ulchovli tahlil uchun ma'lumotlar ombori yaratilgan. OLAP serverlarning tahlili o'tkazilgan va tahlil natijalari asosida Microsoft Analysis Services dasturi tanlab olingan. Ushbu dasturiy muhitda "Oliy ta'lim muassasasining moddiy texnik bazasining tahliliy ko'rsatkichlari" ko'p ulchovli kub ishlab chiqilgan.

## ВВЕДЕНИЕ

**Актуальность темы.** Важнейшие задачи развития и внедрения современных систем компьютеризации и информационно-коммуникационных технологий четко определены государством, изложены в Указе Президента Республики Узбекистан от 30 мая 2002 года «О дальнейшем развитии компьютеризации и внедрении информационно-коммуникационных технологий». Использование ИКТ в обществе является главным составляющим Концепции информатизации, разработанной в соответствии с Указом Президента Республики Узбекистан от 30.05.02 года. «Сегодня нет необходимости доказывать, что XXI век, по общему признанию, становится веком глобализации и стирания границ, информационно-коммуникационных технологий и Интернета, веком все более растущей конкуренции на мировом пространстве и мировом рынке.» отметил при вступительном речи на открытие международной конференции «Подготовка образованного и интеллектуально развитого поколения – как важнейшее условие устойчивого развития и модернизации страны» Президент Узбекистана.

В настоящее время базы данных широко используются в нашей повседневной жизни. Любая, успешно работающая организация, хранит свои данные и обрабатывает их с использованием современных систем управления базами данных. Они повсеместно используются для получения сведений о сотрудниках, о товарах, о продажах, бухгалтерских данных, данных бизнеса и т.д. Но информация сама по себе без обработки не представляет интерес, поэтому работа с базами данных всегда требует совершенствования способов хранения данных, а также сокращения времени выборки данных, необходимых для получения своевременной и необходимой информации.

Важным фактором в современных рыночных условиях является оперативное принятие деловых решений. Однако многие предприятия сталкиваются с таким препятствием, как большой объём и высокая сложность

данных. Решением данного вопроса может стать создание системы поддержки принятия решений (СППР) на основе хранилищ данных (ХД). Хранилище данных представляет собой накопитель информации из других систем (баз данных, файлов, электронных таблиц и др.), на основе которых строятся процессы принятия решений и анализа данных.

Системы поддержки принятия решений (СППР) - это системы, обладающие средствами ввода, хранения и анализа данных, относящихся к определённой предметной области, с целью поиска решений.

Основная задача СППР - предоставление аналитикам инструмента для выполнения анализа данных. СППР не гарантирует правильных решений, а только предоставляет аналитику данные в виде таблиц, отчётов, графиков и т.п. для изучения и анализа.

2011 году 10 мая принят Постановление Президента Республики Узбекистан «О мерах по укреплению материально-технической базы высших образовательных учреждений и кардинальному улучшению качества подготовки высококвалифицированных специалистов». Постановлением главы государства одобрена Программа модернизации материально-технической базы вузов и улучшения качества подготовки специалистов на 2011-2016 годы. Министерстве высшего и среднего специального образования возложена утверждения адресные списки оснащения зданий и сооружений высших образовательных учреждений, с учетом их фактического технического состояния, а также материально-технической обеспеченности каждого высшего образовательного учреждения. Поэтому на сегодняшний день является особо актуальным реализация подсистемы многомерного анализа данных в СППР дающий возможность анализировать показателей материально-технической обеспеченности высших образовательных учреждений.

Подсистема многомерного анализа данных разработана на основе проекта кафедры «Теоретико-методологическое основы построения интеллектуальных программно-технических систем представления и обработки данных для

поддержки принятия решений в слабо формализуемых информационных структурах».

**Степень изученности проблемы.** Исследованию СППР на основе ХД посвящены работы Э.Спирли, Р.Кимбала, А.А.Барсегяна, И.А.Чубуковой, М.С. 11.Аҫга\ya1, Р.УаззШасПя, С.Хайкина, И.С.Ризаева, А.Н.Кузьмина, Л.Ю.Емалетдиновой, Н.М.Вдовичева и др.

**Цель работы.** Целью работы является повышение эффективности обработки и хранения больших объемов информации за счет использования технологии хранилищ данных.

**Задачами исследования являются:**

1. Проанализировать и исследовать существующие способы хранения и обработки информации.

2. Создать методику разработки концептуальной модели многомерного представления данных для эффективного хранения и быстрого выполнения запросов при хранении объектной информации.

3. Провести экспериментальные исследования моделей и алгоритмов с помощью разработанных программ интеллектуального анализа данных и системы поддержки принятия решений в среде СУБД SQL SERVER 2008 R2 на основе концепции хранилищ данных

**Объект исследования.** Объектом исследования является процесс принятия решения в Министерстве высшего и среднего специального образования на основе анализа данных полученных от подведомственных организации.

**Методы исследования.** Методы исследования, применяемые в работе, основаны на методологии структурного анализа и проектирования (SADT - Structured Analysis & Design Technique), теории реляционных баз данных, технологии хранилищ данных и объектно-ориентированных методах построения программных систем.

**Научная новизна работы.** Разработана многомерной модели хранилища данных в сфере высшего образования, а также дано формализованное описание гиперкуба и возможных операций над кубом данных.

**Предмет исследования.** Модели систем поддержки принятия решений на основе многомерных моделей баз и хранилищ данных.

**Научная задача.** Разработка новых аналитических моделей и алгоритмов интеллектуального анализа данных и программного комплекса системы поддержки принятия решений на основе многомерных моделей хранилищ данных.

**Научная и практическая значимость работы** заключается в реализации предложенного подхода в СППР для управления Внедрения информационно коммуникационных технологий в учебный процесс Министерства высшего и среднего специального образования РУз.

**Практическая ценность** диссертации состоит в следующем:

- разработан хранилище данных на языке SQL в среде СУБД MSSQL 2008 R2;
- разработаны алгоритмы и комплексы программ в среде SQL Server Business Development Studio для решения задач загрузке, очистке данных из внешних источников данных.

**Апробация работы.** Основные результаты работы докладывались и обсуждались на:

- научных семинарах кафедры «Информационные технологии» Ташкентского университета информационных технологий
- международная конференция ИГРА 2011 ТУИТ (сентябрь, 2011 г.) на тему «Развитие информационных технологий в Азии»;
- Научно-практической конференции аспирантов, магистрантов и одаренных студентов Каршинский Инженерно-Экономический Институт (КИЭИ, Карши, май, 2012 г.) на тему «Иқтисодийни модернизация қилиш

шароитида ишлаб чиқариш соҳаларини инновацион ривожлантириш муаммолари».

**Опубликованность результатов.** По теме диссертации опубликовано 2 статьи.

**Структура и объем работы.** Выпускная диссертационная работа состоит из введения, трёх глав, заключения, списка использованной литературы, списка Интернет ресурсов и приложения.

**Во введении** обоснована актуальность работы, поставлены цели и задачи, научная новизна диссертационной работы.

**В первой главе** – «Анализ объекта исследования» приведено описание концепция хранилище данных и OLAP, анализ процесса принятий решения в MBCCO.

**Во второй главе** – «Проектирование и разработка хранилище данных» рассматриваются вопросы проектирования: разработка хранилище данных на основы поставленной задаче.

**В третьей главе** – «Реализация подсистемы многомерного анализа данных в СППР для департамента ИКТ MBCCO» описывается процесс сравнение OLAP серверов, и на основы выбранного OLAP сервер создается многомерный куб для анализа.

**В заключении** приведены основные полученные результаты исследования.

Содержание работы изложено в 105 страницах текста, а также включает в себя 39 рисунков и 5 таблицу.

Автор выражает свою искреннюю благодарность профессору Зайниддинову Х. Н., доценту кафедры «ИТ» Гаипназарову Т. С. за поддержку, консультации, адекватное оценивание и за оказанное внимание в течение всего времени работы над диссертацией.

## ГЛАВА 1. АНАЛИЗ ОБЪЕКТА ИССЛЕДОВАНИЯ

### 1.1. Базовые понятия информационно-аналитических систем

Современный этап развития рыночных отношений в узбекской экономике (первое десятилетие XXI века) характеризуется началом экономического подъема. Период времени быстрых, в значительной мере интуитивных, импровизационных, а зачастую и силовых решений меняется на зону продуманных, просчитанных выводов и решений — оперативных, инвестиционных.

Необходимо также принимать во внимание открытость экономики Узбекистана и связанной с ней конкуренции с высокоразвитыми экономическими субъектами. В регионах мира со сложившейся развитой рыночной экономикой достижение заметного повышения прибыли (от долей процента) связано со сложной аналитической работой с использованием новейших достижений науки: математики всех направлений, информационных технологий (ИТ), которые питают и подкрепляют экономические науки, менеджмент, маркетинг, образование, социологию, юриспруденцию и т.д. Начинают приобретать определяющее значение знания о протекающих хозяйственных процессах.

На успех ведения дела влияют как объективные, так и субъективные факторы.

К объективным факторам можно отнести:

- закономерности протекания хозяйственных процессов,
- правовую среду,
- неписаные правила и традиции ведения дел,
- экономическую конъюнктуру и т.д.

Большое значение имеет субъективный фактор, под которым будем понимать влияние на ход бизнес-процессов работников предприятия и в особенности лиц, принимающих решения (ЛПР).

Для выработки и принятия соответствующих складывающейся обстановке решений необходимы информация и знания, которые должны удовлетворять требованиям полноты, достоверности, своевременности (актуальности), полезности.

Основополагающую роль в подготовке принятия решений играет его обоснование имеющейся у ЛПР информации. Ее, как правило, получают из различных внутренних и внешних источников. В интересах выработки адекватного решения используются внутренние информационные ресурсы, которые складываются из отражения деятельности (функционирования) объекта в документах, других видах и способах сбора, обработки, хранения информации, а также внешние по отношению к объекту информационные ресурсы, например (если это предприятие) — корпорации, отрасли, региона, а также глобальные — из средств массовой информации, специальной литературы, всемирной информационной сети Internet и т.д.

Таким образом, границы информационного пространства как отображения деятельности предприятия и его взаимодействия с внешней средой, в рамках которого принимаются решения, выходят далеко за пределы предприятия. Одной из первостепенных задач при подготовке и принятии решений является анализ имеющейся в распоряжении ЛПР информации, который является фундаментом обоснования решения.

Объемы информации, необходимой и используемой при принятии решений, достигают десятков и сотен мегабайт, а в крупных корпоративных и общегосударственных системах и терабайт (10<sup>12</sup> Гигабайт). Информация характеризуется многоплановостью, сложностью отображаемых объектов и систем, а также связей между объектами, явлениями и процессами, скрытостью закономерностей.

Эти обстоятельства вынуждают использовать имеющиеся в настоящее время весьма развитые программно-технические средства. Широкое и эффективное применение этих средств стало одним из факторов выживаемости

и успеха предприятия в условиях острой конкурентной борьбы. Получили широкое распространение автоматизированные информационные системы, которые в последние годы чаще называют информационные системы, подразумевая, что без автоматизации их просто невозможно представить.

Проблема анализа исходной информации для принятия решений оказалась настолько серьезной, что появилось отдельное направление или вид информационных систем — информационно-аналитические системы (ИАС), под которыми понимают комплекс аппаратных, программных средств, информационных ресурсов, методик, которые используются для обеспечения автоматизации аналитических работ в целях обоснования принятия управленческих решений и других возможных применений.

Вся проблема аналитической подготовки принятия решений имеет следующие аспекты:

- извлечение из многих источников разнородных данных, представленных в различных форматах и приведение их к единому формату и единой структуре;
- организация хранения и предоставления пользователям необходимой для принятия решений информации;
- собственно анализ, в том числе оперативный и интеллектуальный, и подготовка плановой или регулярной оценки состояния управляемого объекта в виде бумажных документов или экранных форм;
- подготовка результатов оперативного и интеллектуального анализа для эффективного их восприятия потребителями и принятия на основе адекватных решений.

Аспект, касающийся сбора и хранения информации с сопутствующей доработкой, оформился в концепцию информационных хранилищ (Data Warehouse). Эта концепция состоит в том, что сведения о деятельности предприятия или иного объекта хозяйственной или иной деятельности накапливаются в течение длительного периода времени (годы) в информационном хранилище по определенным правилам. Накопленные данные

используются в различных временных режимах для анализа, как источник данных для разного рода отчетности и работы с партнерами (Reporting) и обоснования управленческих решений.

В связи с большим объемом и сложностью аспект проблемы собственно анализа имеет два направления — оперативный анализ данных (информации), широко распространена аббревиатура англоязычного названия — On-Line Analytical Processing — OLAP. Основной задачей оперативного или OLAP-анализа является быстрое (в пределах секунд) извлечение необходимой аналитики или ЛПП для обоснования или принятия решения информации.

Интеллектуальный анализ информации — имеет также широко распространенное в русской специальной литературе англоязычное название Data mining. Предназначен для фундаментального исследования проблем в той или иной предметной области. Требования по времени менее жестки, но используются более сложные методики. Ставятся, как правило, задачи и получают результаты стратегического значения. При решении сложных задач в режиме Data mining приходится использовать весьма мощные специальные программные средства или, как говорят, инструменты.

Аспекты проблемы анализа и необходимые для их разрешения функции нашли выражение в соответствующих программных продуктах. Соответственно средства автоматизации анализа представлены в различных видах. Имеются комплексные информационно-аналитические системы, выполняющие в той или иной степени функции в соответствии с рассмотренными аспектами. Представлены на рынке программных продуктов и целевые программные системы, выполняющие в увеличенном объеме, расширенном составе и повышенной сложности какие либо функции, например оперативного или интеллектуального анализа. ИАС информационно подпитывают системы поддержки принятия решений (СППР), в литературе также применяют аббревиатуру DSS (Decisin Support Sistem).

В целом сложился рынок инструментальных средств создания и поддержки OLAP-систем, информационных хранилищ (DWH), СППР (DSS), интеллектуального анализа Data mining (DMg), который получил обобщенное название — Business intelligence (BI), которому пока не подобран соответствующий русскоязычный термин.

Как правило, все инструментальные средства, предназначенные для автоматизации аналитических работ, приспособлены для обработки многомерных массивов информации; имеют также возможность импорта/экспорта данных в другие операционные среды, развитые средства визуального двумерного (2D) и трехмерного (3D) представления информации.

## **1.2. Data Warehouse – хранилище данных – ХД - систем обработки данных**

Автоматизация или компьютеризация предприятий, как правило, происходит исторически и «снизу вверх» (исключая случаи полной реорганизации). Определенные подразделения, которые заняты массовыми операциями по обработке данных (выписка счетов, учет производства, складской учет, работа с поставщиками и пр.), подвергаются процессу целевой автоматизации. Это означает, что массовые задачи будут производиться с помощью компьютеров, которые заменят работу с бумажными носителями информации (документы, бланки) на режим, в котором действия отражаются в БД, или (что чаще всего и происходит) информация будет дублироваться как на бумажных носителях, так и в БД.

В силу того, что компьютеризации (или отражению в БД) подвергается каждый раз небольшой фрагмент процессов предприятия, и это происходит не всегда согласованно, большинство БД плохо согласованы друг с другом, т.е. в различных БД одинаковые объекты реального мира могут быть отражены по-разному (например, Иванов И. И. при начислении заработной платы и Иванов

Иван Иванович при получении материальных ценностей в отчет). Эта проблема может решаться введением централизованных справочников и словарей уровня предприятия, что само по себе является нетривиальным техническим процессом.

Хранилище Данных (Data Warehouse , DW , ХД, информационное хранилище, склад данных):

- это централизованный фонд структурированного хранения и быстрого поиска информации при Off-Line анализе информации;

- это предметно-ориентированный, интегрированный, неизменяемый, поддерживающий хронологию и неизменяемый набор данных, предназначенный для поддержки принятия решений, и его пользователи — это высший и средний менеджмент, аналитики, представители подразделений бизнес-анализа и маркетинга;

- это среда, состоящая из одной или более БД, спроектированная для доставки соответствующего и согласованного бизнес-анализа (Business Intelligence, BI) во все бизнес-подразделения организации;

- это инструмент управления, благодаря которому руководство получает доступ к информации, необходимой для принятия обоснованных решений;

- это средство, которое помогает руководству принимать стратегические решения, производить анализ огромных объемов данных, открывать скрытые возможности и улучшать конкурентоспособность своего предприятия.

**ХД собирает информацию** из различных источников и рисует исчерпывающую картину бизнес деятельности организации. Такая система преобразует данные в согласованный, легко доступный формат и распределяет их среди тех, кому они необходимы.

Необходимость создания ХД на крупных предприятиях чаще всего определяется несовершенством существующих систем обработки транзакций.

Назначение ХД:

- выполнения серверных/дисковых задач, связанных с созданием запросов и отчетов на серверах/дисках, не используемых в системах обработки транзакций OLTP. Большинство фирм стремятся настроить системы обработки транзакций так, чтобы все операции выполнялись за приемлемое время. В связи с этим рекомендуется реализовывать архитектуру ХД, использующую отдельные серверы/диски для создания запросов/отчетов, что позволит добиться приемлемого времени обработки транзакций и будет разумно и с финансовой и с организационной точки зрения;

- использования моделей данных и/или серверных технологий, ускоряющих создание запросов и отчетов, но не предназначенных для обработки транзакций;

- создания комфортной среды, в которой написание и поддержка запросов и отчетов не требует больших знаний в области технологий БД. А также для обеспечения средств, позволяющих техническим специалистам ускорить процесс написания и поддержки запросов и отчетов, что сокращает количество бюрократических процедур;

- создания репозитория "очищенных" данных системы обработки транзакций и последующего получения отчетов из этих данных без изменения самой OLTP-системы. ХД дает возможность очистки данных без изменения систем обработки транзакций;

- упрощения формирования запросов и отчетов по данным из нескольких систем обработки транзакций, а также из внешних источников данных и/или по данным, которые хранятся только для отчетности. Долгое время для составления отчетов по данным из нескольких систем организации были вынуждены писать специальные процедуры извлечения данных и выполнять операции сортировки и объединения, а затем уже составлять отчеты по отсортированным (и/или объединенным) выборкам данных;

- создания репозитория данных для OLTP-системы, содержащего долговременную информацию, хранение которой в системе обработки

транзакций не эффективно. Чтобы не замедлять выполнение операций, старые данные часто удаляются из систем обработки транзакций. Однако для отчетности и составления запросов есть смысл держать эту информацию в ХД, где время отклика не так критично;

- ограничения доступа к БД системы и программной логике ее управления лицам, использующим данные OLTP-систем исключительно для составления отчетов и запросов. В этом случае основная цель - защита информации;

- развития систем управления взаимоотношениями с клиентами (CRM). Процесс выстраивания взаимоотношений с клиентами нацелен на сохранение старых и привлечение новых клиентов, что трудно осуществить без автоматизации продаж, маркетинга и совершенствования обслуживания;

- ликвидации разрозненности данных. Несмотря на то, что предприятия склонны к централизации всех систем автоматизации в рамках КИС, достичь этого удастся далеко не всегда, поскольку неизбежно присутствуют разнородные источники информации.

Кроме определенной выше глобальной цели – создание единой модели данных, можно выделить т.н. «краткосрочные цели», которые дают немедленный выигрыш пользователям на каждом этапе реализации ХД. Вот несколько примеров краткосрочных целей:

1. Улучшение качества данных. Поскольку обычным недостатком СППР являются "грязные данные", пользователь должен уделять внимание качеству своих данных на каждом этапе реализации ХД. Очистка данных представляет собой проблему в организации Хранилищ: с одной стороны, предполагается, что ХД обеспечит чистые, интегрированные, соответствующие и согласованные данные, извлеченные из множества источников; с другой стороны, есть расписание разработки, составленное в расчете на 6-12 месяцев. Практически невозможно достичь обеих целей одновременно, не идя на какие-либо компромиссы. Трудность в том, чтобы определиться с существом этих компромиссов.

2. Подготовка данных для СППР. Создание ХД позволяет произвести следующий этап автоматизации деятельности предприятия — подготовить данные для систем поддержки принятия решений. Основным отличием деятельности по принятию решений от исполнительской деятельности, с точки зрения используемых данных, является потребность во всестороннем видении процессов во всем многообразии параметров, от которых они зависят, за различные временные промежутки. Для создания систем поддержки принятия решений необходима полная, непротиворечивая, информация за различные временные промежутки, которая может быть как обобщена (просуммирована или агрегирована другим способом), так и детализирована.

3. Минимизация количества несовместимых отчетов. Другой распространенной проблемой сегодняшних сред СППР является несовместимость отчетов. Несовместимые отчеты в основном происходят от неправильного использования данных, и первопричиной этого является разногласия или непонимание значения или содержимого данных.

4. Захват и доступ к метаданным. Метаданные необходимы для совместного доступа к данным и навигации по ним. Сегодня к большинству данных нет совместного доступа по многим причинам, одной из которых является непонимание данных, а другой - недоверие к содержимому данных. Как только исходные данные очищены, преобразованы, агрегированы, просуммированы и рассечены несколькими различными способами, пользователи никогда снова не найдут их в ХД без помощи метаданных. Захват метаданных (то есть, определений данных, доменов, алгоритмов преобразования исходных данных, столбцы и таблицы, в которых находятся результирующие данные, и все другие технические компоненты) представляет собой возможное решение

5. Обеспечение возможности совместного доступа к данным. Если совместный доступ к данным является одной из задач ХД, то необходимо включить туда некоторую очистку данных, урегулирование разногласий по

данным и компоненты средств доступа, основанные на метаданных в качестве инструментов достижения этой цели. Эти компоненты представляют собой предварительные условия совместного доступа к данным. Двумя другими существенными компонентами являются проектирование БД и организация доступа к ней.

6. Проектирование БД - Базы Данных. После того, как проанализированы требования, необходимые данные логически смоделированы и относящиеся к ним метаданные захвачены в репозиторий, следующим шагом является проект БД. Проектирование самостоятельной БД для одного бизнес-подразделения отличается от проектирования БД совместного доступа для множества бизнес-подразделений.

7. Интеграция данных из множества источников. Это существенная задача для всех ХД, поскольку это первостепенная проблема в различных СППР. Самостоятельные системы, имеющие одни и те же данные, идентифицируемые различными ключами, представляют собой одну из причин того, почему в большинстве компаний отсутствует интеграция данных. Некоторые другие причины заключаются в том, что содержимое данных в одном файле находится на отличном от другого файла уровне детализации или в том, что одни и те же данные модернизируются с разной периодичностью в различных файлах.

8. Соединение исторических данных с текущими данными. Типичной задачей ХД является сохранение истории. Эта задача сопровождается своими проблемами. Исторические данные редко хранятся в операционных системах, и даже если они там хранятся, трудно найти трехлетнюю или пятилетнюю историю в рамках одного файла.

**Концепция** ХД определяет процесс сбора, отсеивания, предварительной обработки и накопления данных с целью долговременного хранения данных и предоставления результирующей информации пользователям в удобной форме для статистического анализа и создания аналитических отчетов. В основе

концепции ХД для интеграции неоднородных источников данных принципиально отличается от подхода динамической интеграции разнородных БД, лежат две основополагающие идеи:

Интеграция ранее разъединенных детализированных (описывающих некоторые конкретные факты, свойства, события и т.д.) данных в едином ХД : исторические архивы, данные из традиционных СОД, данные из внешних источников в едином ХД, их согласование и возможно агрегация. Интегрированность означает, что, например, данные, полученные из различных источников, хранятся согласованно и централизованно.

Разделение наборов данных используемых для операционной обработки и наборов данных используемых для решения задач анализа, применяемых в системах поддержки принятия решений. Такое разделение возможно путем интеграции источников ранее разъединенных детализированных данных в едином ХД, их согласования и, возможно, агрегации. Организация информационного процесса при построении ХД представлена на рисунке 1.1..



Рис. 1.1. Организация информационного процесс

Цель концепции ХД - прояснить отличия в характеристиках данных в операционных и аналитических системах, определить требования к данным помещаемым в целевую БД ХД, определить общие принципы и этапы её построения, основные источники данных, дать рекомендации по решению потенциальных проблем возникающих при их выгрузке, очистке, согласовании, транспортировке и загрузке в целевую БД.

Для правильного понимания данной концепции необходимо понимание следующих принципиальных моментов, заключающихся в том, что концепция ХД:

- это не концепция анализа данных, скорее это концепция подготовки данных для анализа.

- не предопределяет архитектуру целевой аналитической системы. Она говорит о том, какие процессы должны выполняться в системе, но не о том, где конкретно и как эти процессы должны выполняться.

- предполагает не просто единый логический взгляд на данные организации (как иногда это трактуется). Она предполагает реализацию единого интегрированного источника данных.

Вопрос реализации единого интегрированного источника данных достаточно принципиален. Концепция ХД предполагает не просто единый логический взгляд на данные организации, а действительную реализацию единого интегрированного источника данных для систем обработки данных (СОД).

Сегодня, достаточно популярны решения, предполагающие интеграцию различных СОД на основе единого справочника метаданных ( поддерживающего единый логический взгляд данные организации ), но не единого интегрированного источника данных. При этом предполагается динамическая выгрузка, по каждому новому запросу, данных из различных операционных источников (СОД) их динамическое согласование, агрегация и транспортировка к пользователю. Очевидно, что для определённых классов приложений, это решение вполне корректно. Но следует заранее понимать все ограничения им накладываемые.

Кроме единого справочника метаданных, средств выгрузки, агрегации и согласования данных, концепция ХД, как отмечалось ранее, подразумевает: интегрированность, не изменчивость, поддержку хронологии и согласованность данных. И если, два первых свойства ( интегрированность и не изменчивость )

вливают на режимы анализа данных (как будет показано ниже, без интегрированной БД, в которой используются специализированные методы хранения и доступа, по крайней мере, сегодня, трудно говорить о реализации интерактивного динамического анализа), то последние два (поддержка хронологии и согласованность), существенно сужают список решаемых аналитических задач.

Компоненты типичного ХД. Компоненты, входящие в типичное ХД, представлены на рисунке 1.2..

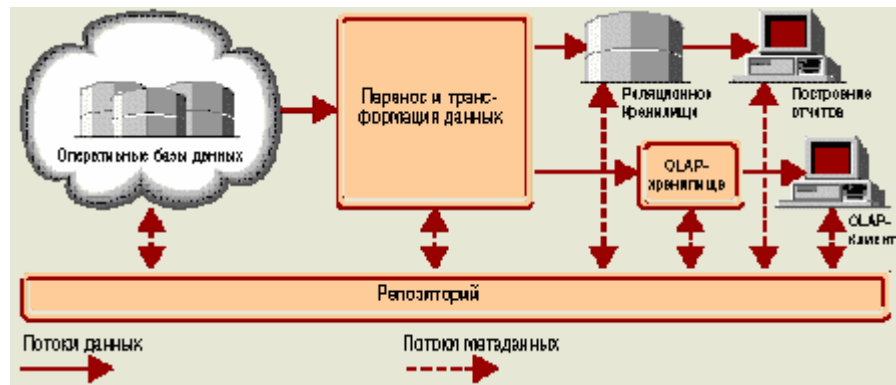


Рис. 1.2. Структура хранилище данных

Оперативные данные собираются из различных источников, очищаются, интегрируются и складываются в реляционное ХД. При этом они уже доступны для анализа при помощи различных средств построения отчетов. Затем данные (полностью или частично) подготавливаются для OLAP-анализа. Они могут быть загружены в специальную БД OLAP или оставлены в реляционном ХД. Важнейшим его элементом являются метаданные, т. е. информация о структуре, размещении и трансформации данных. Благодаря им обеспечивается эффективное взаимодействие различных компонентов хранилища.

Без поддержки хронологии (наличия исторических данных) нельзя говорить о решении задач прогнозирования и анализа тенденций. Но наиболее критичными и болезненными, оказываются вопросы, связанные с согласованием данных.

Основным требованием аналитика, является даже не столько оперативность, сколько достоверность ответа. Но достоверность, в конечном

счете, и определяется согласованностью. Пока не проведена работа по взаимному согласованию значений данных из различных источников, сложно говорить об их достоверности.

Практически в любой организации, вопрос о согласованности данных в различных информационных системах стоит чрезвычайно остро. И, нередко, менеджер сталкивается с ситуацией, когда на один и тот же вопрос, различные системы могут дать и обычно дают различный ответ. Это может быть связано как с не синхронностью моментов модификации данных, отличиями в трактовке одних и тех же событий, понятий и данных, изменением семантики данных в процессе развития предметной области, элементарными ошибками при вводе и обработке, частичной утратой отдельных фрагментов архивов и т.д. Очевидно, что учесть и заранее определить алгоритмы разрешения всех возможных коллизий мало реально. Тем более, это нереально сделать в оперативном режиме, динамически, непосредственно в процессе формирования ответа на запрос.

Примерная структура ИАС, построенной на основе ХД, показана на рисунке 1.3..



Рис. 1.3. Примерная структура информационно-аналитической системы, построенной на основе Хранилища данных

В конкретных реализациях отдельные компоненты этой схемы могут отсутствовать. При такой организации ИАС ХД функционирует по следующему сценарию: по заданному регламенту в него собираются данные из различных источников – БД систем оперативной обработки. В ХД поддерживается хронология: наравне с текущими хранятся исторические данные с указанием времени, к которому они относятся. В результате необходимые доступные данные об объекте управления собираются в одном месте, приводятся к единому формату, согласовываются и, в ряде случаев, агрегируются до минимально требуемого уровня обобщения.

Несмотря на то, что ХД содержат заведомо избыточную информацию, которая и так имеется в базах или файлах оперативных систем, появление концепции ХД вызвано тем, анализировать данные оперативных систем напрямую невозможно или очень затруднительно. Это объясняется:

- разрозненностью данных (OLTP-системы, текстовые отчеты, xls-файлы);
- хранением их в форматах различных СУБД и в разных узлах корпоративной сети. Но даже если на предприятии все данные хранятся на центральном сервере БД (что бывает крайне редко), аналитик почти наверняка не разберется в их сложных, подчас запутанных структурах
- сложные аналитические запросы к оперативной информации тормозят текущую работу компании, надолго блокируя таблицы и захватывая ресурсы сервера.

Можно констатировать, что практически в любой организации сложилась парадоксальная ситуация: - информация вроде бы, где-то и есть, её даже слишком много, но она неструктурирована, несогласованна, разрознена, не всегда достоверна, её практически невозможно найти и получить. В результате можно говорить об отсутствии информации при ее наличии и даже избыточности.

Для того, чтобы извлекать полезную информацию из данных, они должны быть организованы способом, отличным от принятого в OLTP-системах потому что:

- в OLTP-системах используются нормализованные таблицы БД. Нормализация эффективна, если отношения часто перестраиваются (вставка), но дает отрицательный эффект в случае операции выборки (особенно в случае сложных запросов). А в DSS-системах только операции выборки, и данные редко меняются, поэтому данные целесообразно хранить в виде слабо нормализованных отношений, содержащих заранее вычисленные основные итоговые данные. Большая избыточность и связанные с ней проблемы тут не страшны, т.к. обновление происходит только в момент загрузки новой порции данных. При этом происходит как добавление новых данных, так и пересчет итогов.

- выполнение некоторых аналитических запросов требует хронологической упорядоченности данных. Реляционная модель не предполагает существования порядка записей в таблицах.

- в случае аналитических запросов чаще используются не детальные, а обобщенные (агрегированные данные).

Организация потоков данных в ХД показана на рисунке

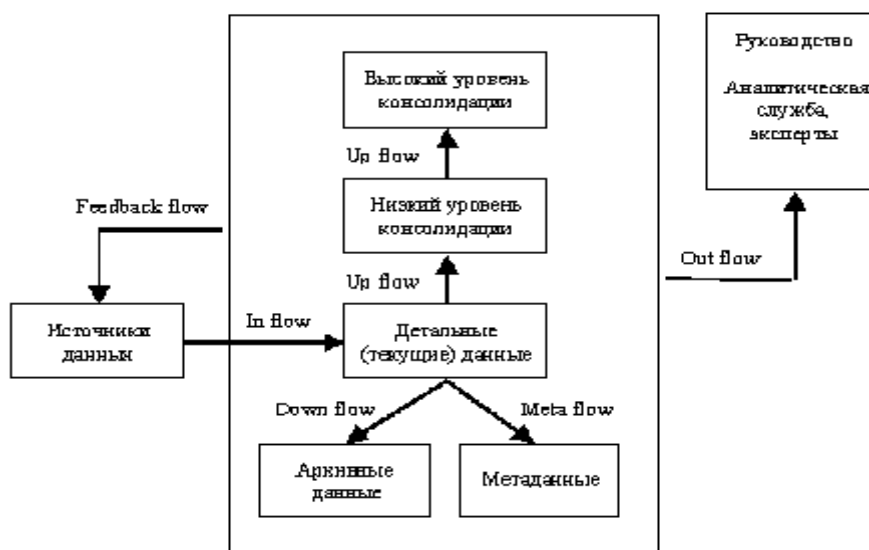


Рис. 1.4. Организация потоков данных в Хранилище Данных

Так как ХД носит предметно-ориентированный характер, его организация нацелена на содержательный анализ информации, а не на автоматизацию бизнес-процессов. Это свойство определяет архитектуру построения хранилища и принципы проектирования модели данных, отличные от тех, что применяются в оперативных системах. Методика построения ХД из простой теоретической дисциплины превратилась в сложную науку, полную вариаций и направлений.

Data Mart - Витрины данных. Облегченным вариантом корпоративного ХД является Витрина Данных (киоск, секция, лавка, рынок, ВД) - это набор тематически связанных БД, содержащие информацию, относящуюся к отдельным аспектам деятельности организации. По сути дела, ВД - это облегченный вариант ХД, существенно меньше по объему, чем корпоративный ХД, и для его реализации не требуется особо мощная вычислительная техника.

Концепция ВД была предложена Forrester Research в 1991 году. При этом главная идея заключалась в том, что ВД максимально приближены к конечному пользователю и содержат только тематические подмножества заранее агрегированных данных, по размерам гораздо меньшие, чем общекорпоративное ХД, и, следовательно, требующие менее производительной техники для поддержания. Концепция ВД ориентирована исключительно на хранение, а не на обработку корпоративных данных. Она не предопределяет архитектуру целевых аналитических систем, а только создает поле деятельности для их функционирования, концентрируясь на требованиях к данным. Таким образом, она оставляет свободу выбора во всем, что касается способов представления данных в целевом ХД (реляционный, многомерный) и режимов анализа данных хранилища.

В большинстве случаев ВД (витрина данных) - это аналитическая структура, которая обычно поддерживает область работы одного приложения, бизнес-процесса или отдела. Сотрудники отдела обобщают требования к информации и приспособливают каждую витрину к своим нуждам. Затем они

обеспечивают персонал, работающий с информацией, средствами интерактивной отчетности (например, инструментами OLAP, средствами формирования незапланированных запросов или параметризованных отчетов). Эти средства позволяют сотрудникам углубляться в данные и исследовать их пространственную структуру "вдоль и поперек", чтобы выявить тренды и получить более детальную картину событий, являющихся движущими силами тех процессов или задач, которыми эти сотрудники управляют.

### 1.3. Подходы к архитектуре хранилищ данных – ХД

На сегодняшний день существуют следующие подходы к архитектуре ХД.

- корпоративная информационная фабрика ( CIF , Corporate Information Factory ) Билла Инмона;
- ХД с архитектурой шины ( Data Warehouse Bus , BUS) Ральфа Кимболла (Ralph Kimball);
- объединенное ХД (Federated Data Warehouse/Federated Data Mart, FDW\FDM).

Выбрать оптимальную систему из этого набора архитектурных решений не такая простая задача.

**Corporate Information Factory, Корпоративное хранилище данных.** Когда-то этот подход был известен под названием классического ХД ( Enterprise Data Warehouse , EDW). Корпоративное ХД является широко распространённым и уникальным репозиторием информации предприятия. Среда Хранилища предназначена только для чтения и состоит из детальных и агрегированных данных, которые полностью очищены и интегрированы; кроме того, в нем хранится обширная и детальная история данных на уровне транзакций. С точки зрения этого архитектурного решения ХД реализует свои функции, прежде всего, через подмножество зависимых Витрин данных.

Корпоративное ХД - хранилище данных - это:

- проект корпоративного масштаба, охватывающий все отделы и обслуживающий нужды всех пользователей корпорации
- не механическая коллекция Витрин Данных, а физически целостный объект.

Работа такого Хранилища начинается со скоординированного извлечения данных из источников. После этого загружается реляционная база данных с третьей нормальной формой, содержащая атомарные данные. Получившееся нормализованное ХД используется для того, чтобы наполнить информацией дополнительные репозитории презентационных данных, т.е. данных, подготовленных для анализа. Эти репозитории, в частности, включают специализированные Хранилища для изучения и "добычи" данных ( Data Mining ), а также Витрины Данных.

Реляционная база данных. Реляционная база данных - это совокупность отношений, содержащих всю информацию, которая должна храниться в базе. Физически это выражается в том, что информация хранится в виде двумерных таблиц, связанных по ключевым полям. В основе этих БД лежит реляционная модель, разработанная англо-американским ученым Эдгаром Коддом в 1960-70 гг.

При таком сценарии конечные Витрины данных создаются для обслуживания бизнес-отделов или для реализации бизнес-функций и используют пространственную модель для структурирования суммарных данных. Атомарные данные остаются доступными через нормализованное ХД. Очевидно, что структура атомарных и суммарных данных при таком подходе существенно различается.

Пространственная модель - dimensional model. Пространственная модель - это одна из моделей ХД, в которой данные организованы не по третьей нормальной форме, а в виде тематических таблиц, каждая из которых содержит характеристику отдельных категорий информации ( dimensions ). Основная цель пространственной модели - минимизировать время выполнения запроса,

поэтому допускается денормализация данных. С этой же целью данные группируются вокруг центральной задачи (или вопроса), которую придется выполнять наиболее часто. Центральная таблица связана со всеми описательными таблицами, но последние напрямую не связаны между собой (так называемая архитектура "звезда").

Отличительные характеристики подхода Билла Инмона к архитектуре корпоративного ХД - хранилища данных :

- использование реляционной модели организации атомарных данных и пространственной - для организации суммарных данных

- использование итеративного или "спирального" подхода при создании больших ХД, т.е. "строительство" не сразу, а по частям. Это позволяет при необходимости вносить изменения в небольшие блоки данных или программных кодов и избавляет от необходимости перепрограммировать значительные объемы данных в ХД. То же самое можно сказать и о потенциальных ошибках: они также будут локализованы в пределах сравнительно небольшого массива без риска испортить все ХД

- использование третьей нормальной формы для организации атомарных данных, что обеспечивает высокую степень детальности интегрированных данных и, соответственно, предоставляет корпорациям широкие возможности для манипулирования ими и изменения формата и способа представления данных по мере необходимости

Достоинства архитектуры корпоративного хранилища данных:

- непротиворечивость информации.
- один набор процессов извлечения и бизнес-правил.
- общая семантика.
- централизованная, управляемая среда.
- легко создаваемые и наполняемые витрины данных.
- единый репозиторий метаданных.

Недостатки такого архитектурного решения ХД:

- реализация требует больших затрат.
- высокая ресурсоемкость.
- потребность в системах и ресурсах в масштабе всего предприятия.
- рискованный сценарий ("все поставлено на карту").

**Data Warehouse Bus - ХД с архитектурой шины.** Альтернативный подход к архитектуре ХД, известен как ХД с архитектурой шины или подход Ральфа Кимболла.

В этой модели первичные данные преобразуются в информацию, пригодную для использования, на этапе подготовки данных. При этом обязательно принимаются во внимание требования к скорости обработки информации и качеству данных. Как и в модели Билла Инмона, подготовка данных начинается со скоординированного извлечения данных из источников. Ряд операций совершается централизованно, например, поддержание и хранение общих справочных данных, другие действия могут быть распределенными.

Область представления пространственно структурирована, при этом она может быть централизованной или распределенной. Пространственная модель ХД содержит ту же атомарную информацию, что и нормализованная модель, но информация структурирована по-другому, чтобы облегчить ее использование и выполнение запросов. Эта модель включает как атомарные данные, так и обобщающую информацию (агрегаты в связанных таблицах или многомерных кубах) в соответствии с требованиями производительности или пространственного распределения данных. Запросы в процессе выполнения обращаются к все более низкому уровню детализации без дополнительного перепрограммирования со стороны пользователей или разработчиков приложения.

В отличие от подхода Билла Инмона, пространственные модели строятся для обслуживания бизнес-процессов (которые, в свою очередь, связаны с бизнес-показателями или бизнес-событиями), а не бизнес-отделов. Например,

данные о заказах, которые должны быть доступны для общекорпоративного использования, вносятся в пространственное ХД только один раз, в отличие от СIF-подхода, в котором их пришлось бы трижды копировать в витрины данных отделов маркетинга, продаж и финансов. После того, как в ХД появляется информация об основных бизнес-процессах, консолидированные пространственные модели могут выдавать их перекрестные характеристики. Матрица корпоративного ХД с архитектурой шины выявляет и усиливает связи между показателями бизнес-процессов (фактами) и описательными атрибутами (измерениями).

Типичные черты подхода Ральфа Кимболла:

- использование пространственной модели организации данных с архитектурой "звезда" ( star scheme ).
- использование двухуровневой архитектуры, которая включает стадию подготовки данных, недоступную для конечных пользователей, и ХД с архитектурой шины как таковое. В состав последнего входят несколько витрин атомарных данных, несколько витрин агрегированных данных и персональная витрина данных, но оно не содержит одного физически целостного или централизованного ХД.

ХД с архитектурой шины обладает следующими характеристиками:

- оно пространственное;
- оно включает как данные о транзакциях, так и суммарные данные;
- оно включает витрины данных, посвященные только одной предметной области или имеющие только одну таблицу фактов ( fact table );
- оно может содержать множество витрин данных в пределах одной БД.

ХД не является единым физическим репозиторием (в отличие от подхода Билла Инмона). Это "виртуальное" ХД. Это коллекция витрин данных, каждая из которых имеет архитектуру типа "звезда".

**Объединенное (федеративное) ХД.** В наше время уже нет необходимости доказывать, как важно для многофилиальных корпораций наличие

согласованной управленческой информации, необходимой для четкого понимания того, как функционирует бизнес. К сожалению, сегодня очень немногим компаниям удалось достичь высокого уровня информационного обеспечения.

Во многих организациях сложилась практика реализации многочисленных ХД. Хотя, по определению, существует только одно ХД, а все остальные объекты являются его подмножеством или постепенно развиваемыми витринами данных, не все организации придерживаются этого правила. Таким образом, во многих компаниях существует два, три, десятка и даже более систем ХД.

Обычный подход к улучшению информированности о бизнес-операциях - проведение стандартизации "сверху вниз" как структуры отчетности, так и модели данных. Однако с практической точки зрения стандартизация бизнес-структур оказывается для большинства организаций исключительно нецелесообразной - требуется слишком много средств, времени. Кроме того, данный подход может быть просто нереализуем, поскольку в любом бизнесе присутствует чрезвычайно мало простых и однообразных областей, которые не потребовали бы учета местной специфики. Если, например, можно стандартизировать коды валют по всей корпорации, введение полностью стандартизированной линейки продуктов редко оправданно, а в некоторых случаях - даже нежелательно.

Поэтому даже если стандартизацию и удалось бы полностью реализовать на практике, необходимость обеспечения согласованности лишала бы филиалы компании возможности приспосабливаться к условиям местного бизнеса и происходящим на нем изменениям. Такая ситуация чревата возникновением противоречий между центральным офисом и местными отделениями - центральная система неспособна отвечать потребностям пользователей, а сотрудники филиалов не стремятся тщательно проверять качество данных, и

поэтому возникает опасность, что данные, отправляемые в головной офис, могут оказаться ошибочными.

Выше описанные проблемы могут быть решены на основе подхода, в основе которого лежит создание Федеративного ХД. В соответствии с данным подходом, с помощью иерархии связанных ХД можно обмениваться данными, бизнес моделями и структурами отчетности, благодаря чему можно, с одной стороны, осуществлять общий контроль и предусмотреть определенную степень стандартизации, а, с другой - позволить региональным отделениям сохранить автономность и учесть местную специфику.

Система объединенных ХД характеризуется совместным использованием общих информационных точек, устраняя, таким образом, избыточность и гарантируя достоверность информации по всей организации (см. рисунок).

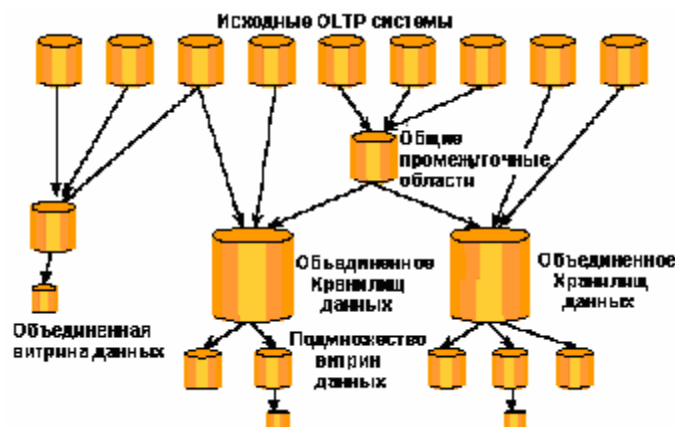


Рис. 1.5. Системы объединенных Хранилищ данных

Федеративное ХД состоит из ряда экземпляров ХД, которые функционируют на полуавтономной основе и, как правило, организационно или географически разнесены, однако могут рассматриваться и управляться как одно большое ХД. Поскольку построение федеративного ХД можно осуществлять постепенно - "шаг за шагом", при его создании разумно воспользоваться методом "начинай с малого, планируй в глобальном аспекте". Такой подход существенно снижает риск неудачи при глобальном развертывании системы, поскольку каждое локальное ХД меньше по масштабу,

незамедлительно отвечает местным требованиям и может управляться сотрудниками регионального бизнес-подразделения.

Каждый из экземпляров ФХД хранит копию базовой бизнес-модели и общие основные данные ( common master data ), причем каждое ХД более высокого уровня содержит итоговые транзакционные данные более низкого уровня. Общие основные данные - например, схема организационной структуры компании - отправляется «вниз», т.е. из корпоративного (глобального) ХД, а суммарные данные о транзакциях отправляются "верх", т.е. из локального ХД. Таким образом "федерация" ХД может предоставить местным отделениям необходимую гибкость, а также обеспечить общий контроль и согласованность; при этом каждое ХД функционирует независимо от всех других остальных.

Данный подход выгоден не только с точки зрения ежедневных операций, но и при процессе внедрения - при построении "федерации" ХД, которые позволяют вносить изменения, предприятия могут начать с одного единственного проекта, а затем построить корпоративную систему, добавляя новые ХД в соответствии с приоритетами бизнеса. Таким образом, возможность настройки используемых ХД с учетом изменений, снимает необходимость заранее устанавливать окончательную архитектуру, другими словами рискованный монолитный проект может быть разбит на многочисленные, менее крупные и рискованные, но более доходные проекты.

Достоинства системы объединенных ХД

- общая семантика и бизнес-правила.
- один набор процессов извлечения и бизнес-правил.
- децентрализованные ресурсы и управление.
- параллельная разработка.

Недостатки такого архитектурного решения

- необходимость в координировании работ.

- сложности в преодолении "политических" моментов и решении вопросов авторских прав.
- требуется согласованность среди различных отделов по вопросам архитектуры, бизнес правил и семантики.
- сложнейшая техническая среда.
- очень часто наличие многочисленных репозиторий метаданных.

#### **1.4. Анализ процесса принятий решений в системе высшего образования**

Указом Президента от 7 января 1990 года также Постановлением Кабинета Министров Республики Узбекистан " О совершенствовании деятельности Министерства высшего и среднего специального образования Республики Узбекистан " от «20» июля 2004 года № 341 утверждено Положение о Министерстве высшего и среднего специального образования Республики Узбекистан. В положение сказано: «Министерство высшего и среднего специального образования Республики Узбекистан является органом государственного управления, осуществляющим руководство высшим и средним специальным, профессиональным образованием в республике. Министерство в своей деятельности подотчетно Кабинету Министров Республики Узбекистан.». Первым и основным функциональным обязанности министерства является анализ экономического, социального и т.д. состояний высших образовательных учреждений и на основе анализе принимать стратегический решения для улучшения и развития образований в республике. Количественные характеристики Министерства:

1. Количество высших образовательных учреждений (ВОУ) 70;
2. Количество студентов республики 66723;
3. Общая количество педагогического персонала 44234;
4. В 2011 году расходная часть госбюджета направленную на образование составила 2,2 триллиона сумов;

5. Управленческий персонал в самом министерства около 60 штатных единиц.

Как видно из показателей министерства имеет очень большой потенциал. Процесс сбора данных для анализа на основе нижеследующего алгоритма:

1. Создаётся поручения для ректоров ВОУ и шаблон документа для заполнения данными (например Excel).

2. После утверждения поручения начальником управления а также соответствующим (координирующим) заместителем министра электронный вариант документ отправляется ВОУ через программу Performance.

3. Обрато данные принимаются через электронную почту и на основе этих данных создаётся сводная таблица.

4. После сбора данных со всех ВУО специалистами ведется анализ данных.

Сбор данных может продлиться с 3 до 7 дней. Это конечно приводит к временному опозданию принятие важных решений.

В 2006 году вместе с чиновниками Кабинета Министров и Центральным аппаратом Министерства был проведен совещание на тему «Повышение эффективности информационно коммуникационных технологий в сфере образование». В конце совещание было принято решение создать автоматизированную систему управления ВОУ. В этом же году был утвержден проект «Единое информационное пространство ВОУ» при Национальном университете. В конце 2010 года на основе этого проекта было разработана 6 автоматизированных информационных систем (АИС), работающих на основе единой базе данных ВОУ:

1. АИС Студент
2. АИС Отдел Кадры
3. АИС Наука
4. АИС Информационно коммуникационных технологий (ИКТ)
5. Документооборот
6. Новый веб-портал Министерства

Достоинства этих систем:

1. Логика программного комплекса был разработан на основе единой базы данных;

2. Программный продукт работает на основе веб-технологий, это дает пользователям системы доступа в любом месте и в любое время;

3. Пользователи АИС распределены на несколько ролей которое дает возможность пользователям работать со своими виртуальными рабочими столами. А также дано доступ создание ролей на основе выбора функции систему.

4. В комплекс программ было разработана детальная информация каждого объекта и процесса;

5. На основе предложений руководителей ВОУ подготавливается новый проект, в котором будет совершенствовать комплекс АИ система и специализироваться для некоторых ВОУ;

6. Установка и поддержка системе для ВОУ бесплатно.

Недостатки:

1. Система работает с транзакциями ежеминутно, которое не дают руководителям ВОУ, специалистам министерства вести анализ данных по времени;

2. Для анализа данных не предусмотрено таблице агрегации данных.

С 2011 года ведется внедрения АИС по приказу министерства. Такие же системы были разработаны в ташкентском университете информационных технологий, национальном университете. Но продукты были разработаны специализированы для самых вузов.

### **1.5. Постановка задачи**

Каждый год для высших образовательных распределяться определённая сумма для закупки новый орг. техники (компьютеры, сканеры), мебели, ремонта и т.д.. В начале года все управления ведут анализ состояний ВОУ.

Например управления Внедрения информационно-коммуникационных технологий в учебный процесс ведут сбор данных о состоянии информационных коммуникационных технологий например компьютеры, сканеры, принтеры или программное обеспечения. На основе данных принимается решения каким ВОУ нужна покупать новые компьютеры, принтеры, электронные доски. Предлагается разработка хранилище данных для анализа состояния материально-технической базы высших образовательных учреждений.

Задачи:

1. Создание хранилище данных;
2. Разработка гиперкуба на основе созданного хранилища данных для многомерного анализа информации

### **1.6. Выводы по первой главе**

1. Проблема анализа исходной информации для принятия решений является отдельном направлением результаты которого используются для обеспечения автоматизации аналитических работ в целях обоснования принятия управленческих решений и других возможных применений.

2. Концепция ХД обеспечивает процесс сбора, отсеивания, предварительной обработки и накопления данных с целью долговременного хранения данных и предоставления результирующей информации пользователям в удобной форме для статистического анализа и создания аналитических отчетов что является достаточно актуальной.

3. В настоящее время проблема использования большого объема накопленной информации является ключевой для руководства Министерства высшего и среднего специального образования (далее МВССО). Данные ведутся в Управлениях МВССО в различных информационных приложениях с использованием разных программных комплексов и фактически используются только сотрудниками конкретного Управления для контроля и формирования

текущей бухгалтерской, экономической, аналитической и статистической отчетности. Руководители из-за масштабов и отсутствия механизма агрегирования и оперативного доступа к банку данных Управлений вынуждены интуитивно принимать решения по управлению ресурсами отрасли. На основе анализа было принято решения разработать хранилище данных для анализа состояния материально-технической базы высших образовательных учреждений.

## **ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ХРАНИЛИЩЕ ДАнных**

Методика проектирования будет рассмотрена на примере проектирования ХД для анализа состояния материально-технической базы высших образовательных учреждений.

### **2.1. Определение требований**

Первым шагом проектирования хранилище данных является анализ запросов и требований, предъявляемых пользователями к ХД. Необходимо проконсультироваться со специалистами, занимающимися анализом в данной области и выяснить вопросы, с которыми конечные пользователи хотели бы обратиться к ХД. На этом этапе интерес представляют не столько тексты вопросов, сколько понимание того, о каких фактах, событиях и объектах в них спрашивается.

Пример:

1. На сколько процентов изменилось материально техническая база ВУЗов последние 2 года?
2. В каких ВОУ наибольшая соотношения компьютеров к студентом?
3. В каких регионах наибольшая потребность к новым информационно коммуникационным технологиям?
4. Насколько процентов увеличилось среднее число соотношения компьютеров к студентам в республике?

### **2.2. Анализ вопросов**

После этого необходимо провести первичный анализ вопросов и определить перечень данных, которые могут потребоваться для ответа на запрос пользователей.

Пример:

1. Количество продуктов по типам входящих в материально техническую базу ВОУ;
2. Количество студентов в ВОУ.

### 2.3. Определение структуры ХД

Теперь необходимо определить структуру ХД, т.е. определить конкретные Измерения и Факты, их взаимосвязи и уровни агрегации хранимых данных.

**Таблица фактов.** Таблица фактов является основной таблицей хранилища данных. Как правило, она содержит сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться. Обычно говорят о четырех наиболее часто встречающихся типах фактов. К ним относятся:

- факты, связанные с транзакциями (Transaction facts). Они основаны на отдельных событиях (типичными примерами которых являются телефонный звонок или снятие денег со счета с помощью банкомата);

- факты, связанные с «моментальными снимками» (Snapshot facts). Основаны на состоянии объекта (например, банковского счета) в определенные моменты времени, например на конец дня, или месяца. Типичными примерами таких фактов являются объем продаж за день или дневная выручка;

- факты, связанные с элементами документа (Line-item facts). Основаны на том или ином документе (например, счете за товар или услуги) и содержат подробную информацию об элементах этого документа (например, количестве, цене, проценте скидки);

- факты, связанные с событиями или состоянием объекта (Event or state facts). Представляют возникновение события без подробностей о нем (например, просто факт продажи или факт отсутствия таковой без иных подробностей).


Для примера рассмотрим факты, связанные с элементами документа (в данном случае счета, выставленного за товар).

Таблица фактов, как правило, содержит уникальный составной ключ, объединяющий первичные ключи таблиц измерений. Чаще всего это целочисленные значения либо значения типа «дата/время» — ведь таблица фактов может содержать сотни тысяч или даже миллионы записей, и хранить в ней повторяющиеся текстовые описания, как правило, невыгодно — лучше поместить их в меньшие по объему таблицы измерений. При этом как ключевые, так и некоторые неключевые поля должны соответствовать будущим измерениям OLAP-куба. Помимо этого таблица фактов содержит одно или несколько числовых полей, на основании которых в дальнейшем будут получены агрегатные данные.

Пример таблицы фактов, которая может быть построена на основе базы данных VSource, приведен на рис. 2.1 и рис 2.2..

FactDateID	FactYearID	FactMonthID	UniversityID	ProductCount	ProductCount@1	ProductCountWorkPn	StudentsCount
1	1	1	1	282	141	212	358
2	2	1	2	270	150	227	527
3	3	1	3	129	176	125	481
4	4	1	4	286	158	178	589
5	5	1	5	63	186	106	388
6	6	1	6	179	191	137	55
7	7	1	7	280	198	183	236
8	8	1	8	289	73	150	69
9	9	1	9	244	7	35	106
10	10	1	10	291	86	72	537
11	11	1	11	5	218	180	388
12	12	1	12	54	38	33	53
13	13	1	13	79	32	231	184
14	14	1	14	240	207	177	90
15	15	1	15	24	156	85	6
16	16	1	1	2	14	80	264
17	17	1	2	80	83	48	147
18	18	1	3	285	126	127	227
19	19	1	4	181	94	130	564
20	20	1	5	138	73	51	81
21	21	1	6	3	214	212	90
22	22	1	7	31	33	112	132
23	23	1	8	110	35	151	444
24	24	1	9	210	40	138	188
25	25	1	10	141	60	62	301
26	26	1	11	196	213	189	121
27	27	1	12	222	33	94	279
28	28	1	13	133	114	21	486

Рис. 2.1. Таблица фактов с данными

FactEdu	
	FactEduID
	TimeID
	ProductID
	UniversityID
	ProductsCount
	ProductsCountInEdu
	ProductsCountWorkPro
	StudentsCount

*Рис. 2.2. Структура таблицы фактов*

В данном примере измерениям будущего куба соответствуют первые шесть полей, а агрегатным данным — последние четыре.

Отметим, что для многомерного анализа пригодны таблицы фактов, содержащие как можно более подробные данные (то есть соответствующие членам нижних уровней иерархии соответствующих измерений). В данном случае предпочтительнее взять за основу факты продажи товаров отдельным заказчикам, а не суммы продаж для разных стран — последние все равно будут вычислены OLAP-средством. Исключение можно сделать, пожалуй, только для клиентских OLAP-средств (о них мы поговорим чуть позже), поскольку в силу ряда ограничений они не могут манипулировать большими объемами данных.

Отметим, что в таблице фактов нет никаких сведений о том, как группировать записи при вычислении агрегатных данных. Например, в ней есть идентификаторы продуктов или клиентов, но отсутствует информация о том, к какой категории относится данный продукт или в каком городе находится данный клиент. Эти сведения, в дальнейшем используемые для построения иерархий в измерениях куба, содержатся в таблицах измерений.

Пример:

Факты (меры) ХД для анализа состояния материально-технической базы высших образовательных учреждений:

- Количество продуктов по типам входящих в материально техническую базу ВОУ;

- Средняя число соотношения продуктов к числу студентов ВОУ.

**Таблицы измерений.** Таблицы измерений содержат неизменяемые либо редко изменяемые данные. В подавляющем большинстве случаев эти данные представляют собой по одной записи для каждого члена нижнего уровня иерархии в измерении. Таблицы измерений также содержат как минимум одно описательное поле (обычно с именем члена измерения) и, как правило, целочисленное ключевое поле (обычно это суррогатный ключ) для однозначной идентификации члена измерения. Если будущее измерение, основанное на данной таблице измерений, содержит иерархию, то таблица измерений также может содержать поля, указывающие на «родителя» данного члена в этой иерархии. Нередко (но не всегда) таблица измерений может содержать и поля, указывающие на «прародителей», и иных «предков» в данной иерархии (это обычно характерно для сбалансированных иерархий), а также дополнительные атрибуты членов измерений, содержащиеся в исходной оперативной базе данных (например, адреса и телефоны клиентов).

Каждая таблица измерений должна находиться в отношении «один ко многим» с таблицей фактов.

Отметим, что скорость роста таблиц измерений должна быть незначительной по сравнению со скоростью роста таблицы фактов; например, добавление новой записи в таблицу измерений, характеризующую товары, производится только при появлении нового товара, не продававшегося ранее.

Пример:

Измерения ХД для анализа состояния материально-технической базы высших образовательных учреждений:

- Время
- ВОУ
- Продукты (относящийся к материально технической базе ВОУ)

Пример таблицы измерений приведен на рис. 2.3. и 2.4.


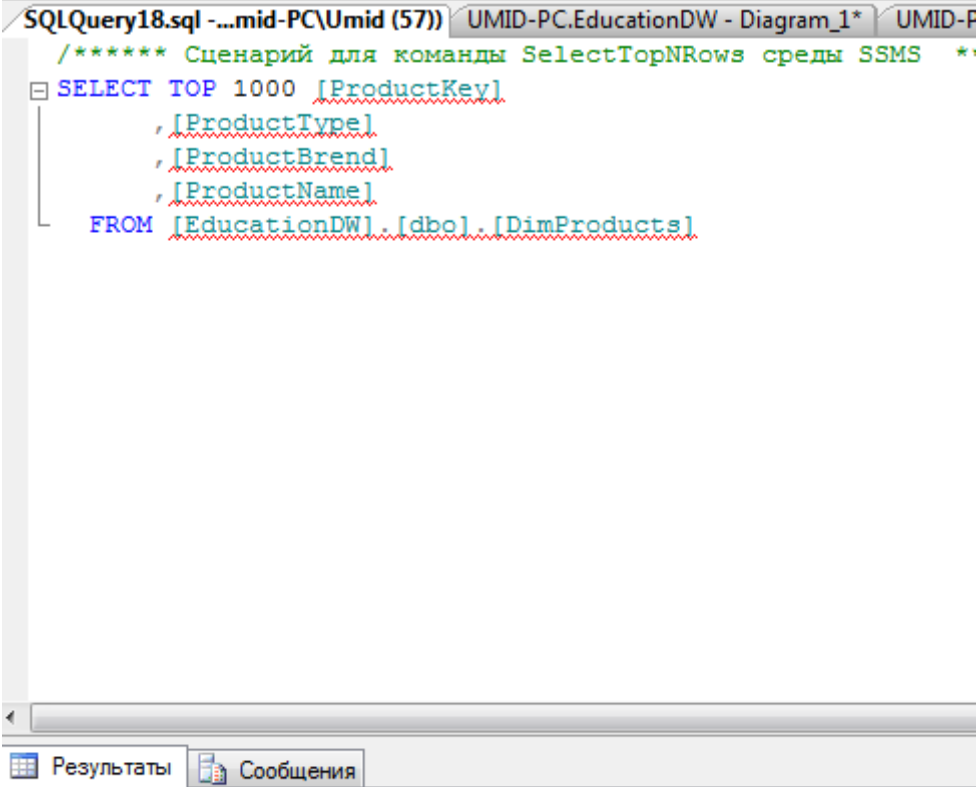
DimProducts	
	ProductKey
	ProductType
	ProductBrend
	ProductName

Рис. 2.3. Примерная структура таблицы измерений



```

SQLQuery18.sql - ...mid-PC\Umid (57)  UMID-PC.EducationDW - Diagram_1*  UMID-F
/***** Сценарий для команды SelectTopNRows среды SSMS ***/
SELECT TOP 1000 [ProductKey]
, [ProductType]
, [ProductBrend]
, [ProductName]
FROM [EducationDW].[dbo].[DimProducts]

```

	ProductKey	Product Type	ProductBrend	ProductName
1	1	Компьютер	Dell	Dell XPS502
2	2	Компьютер	Acer	Acer 1223
3	3	Компьютер	HP	HP23
4	4	Компьютер	Dell	Del34
5	5	Компьютер	Toshiba	Toshiba34

Рис. 2.4. Таблица измерений с данными

Следующий шаг - определение структурной единицы (зерна). Зерно (grain) - нижний уровень детализации данных, хранящихся в ХД.

Информация может суммироваться на различных уровнях в соответствии с иерархической структурой. Ежедневные покупки техники или канцелярий могут суммироваться по неделям, месяцам или годам; подкупленные товары

можно суммировать по продуктам, категориям продуктов или линейкам продуктов.

Пользователи ХД могут работать с различными уровнями суммируемых данных. Вопрос заключается в следующем: в какой степени детализации нуждается пользователь?

Всегда существует вероятность того, что какой-то пользователь в своем поиске пожелает дойти до базовой транзакции, и на основании этого можно прийти к выводу, что разработчик ХД всегда должен обеспечить самый низкий уровень детализации, но обычно в этом нет необходимости, а на практике зачастую неудобно. Это требует хранения в хранилище данных огромных объемов информации. Необходимо внимательно обдумать те вопросы относительно деятельности вашей фирмы, на которые вы хотите получать ответы, и соответственно выберите степень детализации данных. Сотрудникам управления министерства, желающим отследить изменения материально-технической базе ВОУ, нет нужды сохранять в ХД данные о каждой покупке техники и т.д.; покупки можно суммировать на уровне продуктов. Но если необходимо проанализировать поведение потребителей, то потребуется информация об отдельных покупках. Иногда выбор степени детализации определяется интенсивностью изменений: для отслеживания заработной платы служащих в течение длительного периода времени не имеет смысла использовать в качестве "зерна" день или неделю, поскольку оплата труда сотрудников пересматривается один раз в год.

Определив степень детализации, можно установить остальные атрибуты таблицы измерений. В ХД исторических данных в качестве одного из измерений обязательно присутствует время, и в таблицу измерений необходимо поместить информацию о единицах времени, подходящих для данного ХД. Дни, недели, месяцы и годы, очевидно, необходимы, если степень детализации соответствует уровню ежедневных транзакций. Если требуются итоговые

цифры за месяц или год, то, вероятно, для организации таблицы потребуются лишь такие единицы времени, как годы и месяцы.

Пример:

*Таблица 2.1. Степень детализации ХД для анализа состояния материальной технической базы ВУЗов:*

<b>Время</b>	<b>Продукты</b>	<b>ВОУ</b>
Год	Тип	Название
Полугодие	Бренд	Регион
Квартал	Название	

Иерархия строится по принципу убывания сверху вниз: вверху более общие определения, внизу более подробные. В измерении Время я выбрал 3 уровня иерархии: Год, Полгодие, Квартал. Это обусловлено тем, что ВОУ приобретают продукты не очень часто, нет необходимости отслеживать еженедельные или ежемесячные изменения. В измерении Продукты я выбрал 3 уровня иерархии чтобы можно было отбирать данные по Типу продуктов (принтеры, сканеры, компьютеры), По бренду (Acer, HP, Dell, Toshiba), по Названию. В измерении высшего образовательного учреждения я выбрал два уровня иерархии: название (ТГТУ, НУУ и тд), Регион (Ташкент, Самарканд).

Таким образом, общая структура измерений выглядит следующим образом (Таблица 2.2.).

*Таблица 2.2. Общая структура измерений*

<b>Время</b>	<b>Продукты</b>	<b>ВОУ</b>
Год	Тип	Название
Полгодие	Бренд	Регион
Квартал	Название	Адрес
	Общая Количество	Телефон
	Общая количество технике в учебным	Факс

	процессе	
	Количество Технике в пригодном состояний	

Одно измерение куба может содержаться как в одной таблице (в том числе и при наличии нескольких уровней иерархии), так и в нескольких связанных таблицах, соответствующих различным уровням иерархии в измерении. Если каждое измерение содержится в одной таблице, такая схема хранилища данных носит название «звезда» (star schema). Пример такой схемы приведен на рис. 2.5.

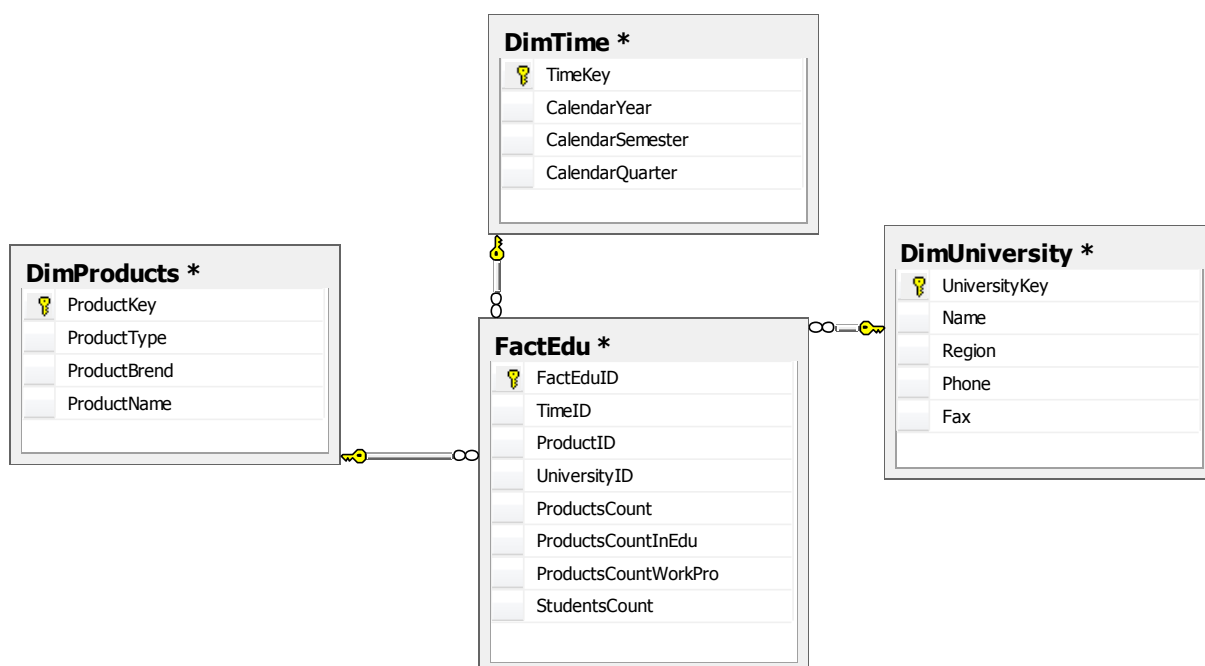


Рисунок 2.5. Пример схемы "звезда"

Если же хотя бы одно измерение содержится в нескольких связанных таблицах, такая схема хранилища данных носит название «снежинка» (snowflake schema). Дополнительные таблицы измерений в такой схеме, обычно соответствующие верхним уровням иерархии измерения и находящиеся в соотношении «один ко многим» в главной таблице измерений,

соответствующей нижнему уровню иерархии, иногда называют консольными таблицами (outrigger table). Пример схемы «снежинка» приведен на рис. 2.6.

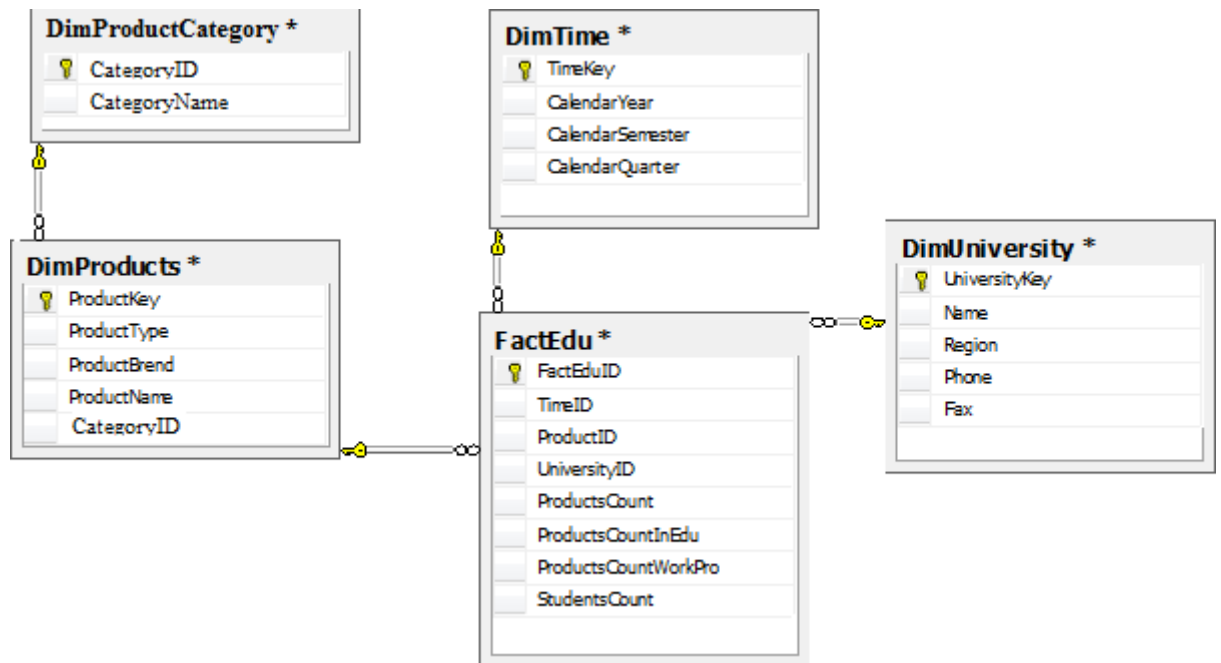


Рисунок 2.6. Пример схемы "снежинка"

Отметим, что даже при наличии иерархических измерений с целью повышения скорости выполнения запросов к хранилищу данных нередко предпочтение отдается схеме «звезда».

Однако не все хранилища данных проектируются по двум приведенным выше схемам. Так, довольно часто вместо ключевого поля для измерения, содержащего данные типа «дата», и соответствующей таблицы измерений сама таблица фактов может содержать ключевое поле типа «дата». В этом случае соответствующая таблица измерений просто отсутствует.

Пример отступления от правил — наличие нескольких разных иерархий для одного и того же измерения. Типичные примеры таких иерархий — иерархии для календарного и финансового года (при условии, что финансовый год начинается не с 1 января), или с различными способами группировки членов измерения (например, группировать товары можно по категориям, а можно и по компаниям-поставщикам). В этом случае таблица измерений

содержит поля для всех возможных иерархий с одними и теми же членами нижнего уровня, но с разными членами верхних уровней.

Как мы уже отмечали выше, таблица измерений может содержать поля, не имеющие отношения к иерархиям и представляющие собой просто дополнительные атрибуты членов измерений (*member properties*). Иногда такие атрибуты могут быть использованы при анализе данных.

Следует сказать, что для создания реляционных хранилищ данных нередко применяются специализированные СУБД, хранение данных в которых оптимизировано с точки зрения скорости выполнения запросов. Примером такого продукта является Sybase Adaptive Server IQ, реализующий нетрадиционный способ хранения данных в таблицах (не по строкам, а по столбцам). Однако создавать хранилища можно и в обычных реляционных СУБД.

## 2.4. OLAP куб

Технология комплексного многомерного анализа данных получила название OLAP (On-Line Analytical Processing). OLAP — это ключевой компонент организации хранилищ данных. Концепция OLAP была описана в 1993 году Эдгаром Коддом, известным исследователем баз данных и автором реляционной модели данных (см. E.F. Codd, S.B. Codd, and C.T.Salley, Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. Technical report, 1993). 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:

- Fast (Быстрый) - предоставление пользователю результатов анализа за приемлемое время (обычно не более 5 с), пусть даже ценой менее детального анализа;

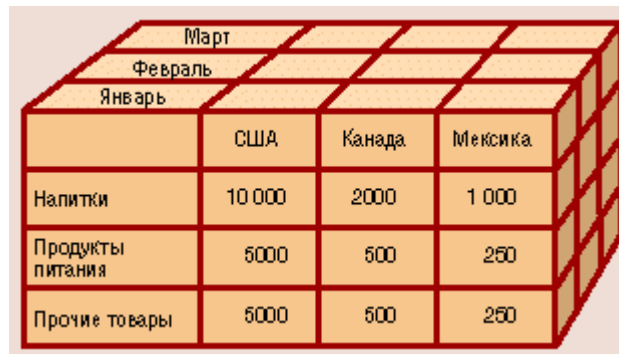
- **Analysis (Анализ)** - возможность осуществления любого логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для конечного пользователя виде;
- **Shared (Разделяемой)** - многопользовательский доступ к данным с поддержкой соответствующих механизмов блокировок и средств авторизованного доступа;
- **Multidimensional (Многомерной)** - многомерное концептуальное представление данных, включая полную поддержку для иерархий и множественных иерархий (это — ключевое требование OLAP);
- **Information (Информации)** - приложение должно иметь возможность обращаться к любой нужной информации, независимо от ее объема и места хранения.

Следует отметить, что OLAP-функциональность может быть реализована различными способами, начиная с простейших средств анализа данных в офисных приложениях и заканчивая распределенными аналитическими системами, основанными на серверных продуктах.

**Многомерное представление информации. Кубы.** OLAP предоставляет удобные быстродействующие средства доступа, просмотра и анализа деловой информации. Пользователь получает естественную, интуитивно понятную модель данных, организуя их в виде многомерных кубов (Cubes). Осями многомерной системы координат служат основные атрибуты анализируемого бизнес-процесса. Например, для продаж это могут быть товар, регион, тип покупателя. В качестве одного из измерений используется время. На пересечениях осей - измерений (Dimensions) - находятся данные, количественно характеризующие процесс - меры (Measures). Это могут быть объемы продаж в штуках или в денежном выражении, остатки на складе, издержки и т. п. Пользователь, анализирующий информацию, может “разрезать” куб по разным направлениям, получать сводные (например, по годам) или, наоборот,

детальные (по неделям) сведения и осуществлять прочие манипуляции, которые ему придут в голову в процессе анализа.

В качестве мер в трехмерном кубе, изображенном на рис. 2.7., использованы суммы продаж, а в качестве измерений - время, товар и магазин. Измерения представлены на определенных уровнях группировки: товары группируются по категориям, магазины - по странам, а данные о времени совершения операций - по месяцам.



	США	Канада	Мексика
Напитки	10 000	2000	1 000
Продукты питания	5000	500	250
Прочие товары	5000	500	250

*Рис. 2.7. Пример куба*

“Разрезание” куба. Даже трехмерный куб сложно отобразить на экране компьютера так, чтобы были видны значения интересующих мер. Что уж говорить о кубах с количеством измерений, большим трех? Для визуализации данных, хранящихся в кубе, применяются, как правило, привычные двумерные, т. е. табличные, представления, имеющие сложные иерархические заголовки строк и столбцов.

Двумерное представление куба можно получить, “разрезав” его поперек одной или нескольких осей (измерений): мы фиксируем значения всех измерений, кроме двух, - и получаем обычную двумерную таблицу. В горизонтальной оси таблицы (заголовки столбцов) представлено одно измерение, в вертикальной (заголовки строк) - другое, а в ячейках таблицы - значения мер. При этом набор мер фактически рассматривается как одно из измерений - мы либо выбираем для показа одну меру (и тогда можем разместить в заголовках строк и столбцов два измерения), либо показываем

несколько мер (и тогда одну из осей таблицы займут названия мер, а другую - значения единственного “неразрезанного” измерения).

Взгляните на рис. 2.8. - здесь изображен двумерный срез куба для одной меры - Unit Sales (продано штук) и двух “неразрезанных” измерений - Store (Магазин) и Время (Time).

	США	Канада	Мексика
Январь	20 000	4000	2000
Февраль	30 000	6000	3000
Март	50 000	10 000	5000

*Рис. 2.8. Двумерный срез куба для одной меры*

На рис. 2.9. представлено лишь одно “неразрезанное” измерение - Store, но зато здесь отображаются значения нескольких мер - Unit Sales (продано штук), Store Sales (сумма продажи) и Store Cost (расходы магазина).

	США	Канада	Мексика
Unit Sales	2000	400	200
Store Sales	30 000	6000	3000
Store Cost	10 000	2000	1000

*Рис. 2.9. Двумерный срез куба для нескольких мер*

Двумерное представление куба возможно и тогда, когда “неразрезанными” остаются и более двух измерений. При этом на осях среза (строках и столбцах) будут размещены два или более измерений “разрезаемого” куба - см. рис. 2.10.

	Январь			Февраль		
	США	Канада	Мексика	США	Канада	Мексика
Unit Sales	500	100	50	500	100	50
Store Sales	7500	1500	750	7500	1500	750
Store Cost	2500	500	250	2500	500	250

*Рис 2.10. Двумерный срез куба с несколькими измерениями на одной оси*

Значения, “откладываемые” вдоль измерений, называются членами или метками (members). Метки используются как для “разрезания” куба, так и для

ограничения (фильтрации) выбираемых данных - когда в измерении, остающемся “неразрезанным”, нас интересуют не все значения, а их подмножество, например три города из нескольких десятков. Значения меток отображаются в двумерном представлении куба как заголовки строк и столбцов.

Метки могут объединяться в иерархии, состоящие из одного или нескольких уровней (levels). Например, метки измерения “Магазин” (Store) естественно объединяются в иерархию с уровнями:

All (Мир)

Country (Страна)

State (Штат)

City (Город)

Store (Магазин).

В соответствии с уровнями иерархии вычисляются агрегатные значения, например объем продаж для USA (уровень “Country”) или для штата California (уровень “State”). В одном измерении можно реализовать более одной иерархии - скажем, для времени: {Год, Квартал, Месяц, День} и {Год, Неделя, День}.

Отметим, что иерархии могут быть сбалансированными (balanced), как, например, иерархия, представленная на рис. 2.11., а также иерархии, основанные на данных типа "дата—время", и несбалансированными (unbalanced). Типичный пример несбалансированной иерархии — иерархия типа "начальник—подчиненный" (ее можно построить, например, используя значения поля Salesperson исходного набора данных из рассмотренного выше примера), представлен на рис. 2.12. Иногда для таких иерархий используется термин Parent-child hierarchy.

Существуют также иерархии, занимающие промежуточное положение между сбалансированными и несбалансированными (они обозначаются термином ragged — "неровный").

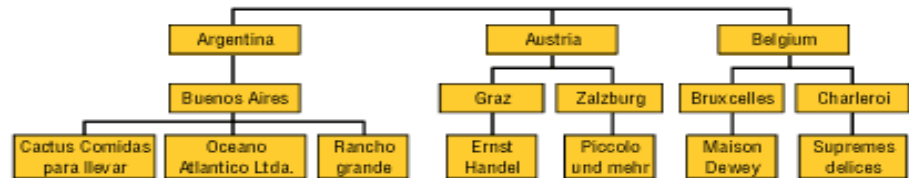


Рисунок 2.11. Иерархия в измерении, связанном с географическим положением клиентов

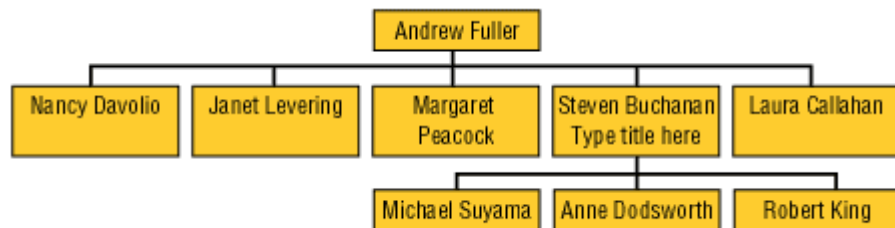


Рис 2.12. Несбалансированная иерархия

Обычно они содержат такие члены, логические "родители" которых находятся не на непосредственно вышестоящем уровне (например, в географической иерархии есть уровни Country, City и State, но при этом в наборе данных имеются страны, не имеющие штатов или регионов между уровнями Country и City; рис. 2.13.).

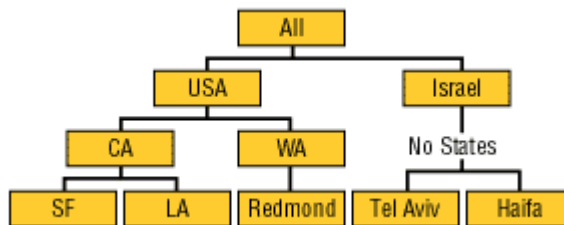


Рисунок 2.13. "Неровная" иерархия

Отметим, что несбалансированные и "неровные" иерархии поддерживаются далеко не всеми OLAP-средствами. Например, в Microsoft Analysis Services 2000 поддерживаются оба типа иерархии, а в Microsoft OLAP Services 7.0 — только сбалансированные. Различным в разных OLAP-средствах может быть и число уровней иерархии, и максимально допустимое число членов одного уровня, и максимально возможное число самих измерений.

**Архитектура OLAP приложений.** Все, что говорилось выше про OLAP, по сути, относилось к многомерному представлению данных. То, как данные

хранятся, грубо говоря, не волнует ни конечного пользователя, ни разработчиков инструмента, которым клиент пользуется.

Многомерность в OLAP-приложениях может быть разделена на три уровня:

- Многомерное представление данных - средства конечного пользователя, обеспечивающие многомерную визуализацию и манипулирование данными; слой многомерного представления абстрагирован от физической структуры данных и воспринимает данные как многомерные.

- Многомерная обработка - средство (язык) формулирования многомерных запросов (традиционный реляционный язык SQL здесь оказывается непригодным) и процессор, умеющий обработать и выполнить такой запрос.

- Многомерное хранение - средства физической организации данных, обеспечивающие эффективное выполнение многомерных запросов.

Первые два уровня в обязательном порядке присутствуют во всех OLAP-средствах. Третий уровень, хотя и является широко распространенным, не обязателен, так как данные для многомерного представления могут извлекаться и из обычных реляционных структур; процессор многомерных запросов в этом случае транслирует многомерные запросы в SQL-запросы, которые выполняются реляционной СУБД.

Конкретные OLAP-продукты, как правило, представляют собой либо средство многомерного представления данных, OLAP-клиент (например, Pivot Tables в Excel 2000 фирмы Microsoft или ProClarity фирмы Knosys), либо многомерную серверную СУБД, OLAP-сервер (например, Oracle Express Server или Microsoft OLAP Services).

Слой многомерной обработки обычно бывает встроен в OLAP-клиент и/или в OLAP-сервер, но может быть выделен в чистом виде, как, например, компонент Pivot Table Service фирмы Microsoft.

## 2.5. Выводы ко второй главе

1. Проведен первичный анализ вопросов и определен перечень данных, которые могут потребоваться для ответа на запрос пользователей.
2. Разработана структура ХД, содержащая три таблицы измерений и одну таблицу фактов связанных между собой по схеме звезда.
3. OLAP (On-Line Analytical Processing) является основным компонентом построения и применения хранилищ данных. Эта технология основана на построении многомерных наборов данных — OLAP-кубов, оси которого содержат параметры, а ячейки — зависящие от них агрегатные данные.

### **ГЛАВА 3. РЕАЛИЗАЦИЯ ПОДСИСТЕМЫ МНОГОМЕРНОГО АНАЛИЗА ДАННЫХ В СППР ДЛЯ УПРАВЛЕНИЯ ВНЕДРЕНИЯ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ В УЧЕБНЫЙ ПРОЦЕСС МВССО**

#### **3.1. Разработка хранилище данных для многомерного анализа**

Для разработки хранилища данных было принято решение создать многомерный куб «Анализ показателей материально технической базы ВОУ». Куб «Анализ показателей материально технической базы ВОУ», показывает параметры материально техническую обеспеченность ВОУ. Данный куб использует небольшое число измерений и использует пользовательских агрегирующих функций.

Можно выделить следующий алгоритм создания куба:

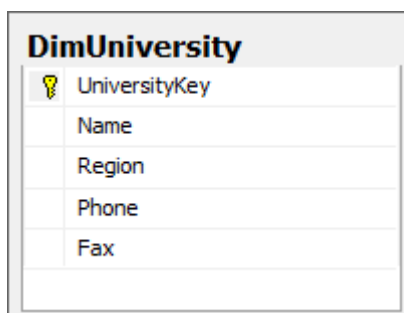
1. Спроектировать многомерный куб – выделить необходимые измерения, иерархии и меры;
2. На основании предыдущего пункта создать в хранилище данных соответствующие таблицы измерений и фактов;
4. Добавить созданный куб в OLAP-сервер.

Нам требуется создать куб, хранящий информацию о материально-технической базе ВОУ. У нашего куба должно быть меры: Количество продуктов (орг. техника, мебель) входящих в состав мат. Базу ВОУ, Количество продуктов используемых в учебном процессе, Количество продуктов в состоянии исправности, Количество студентов (таблица «FactEdu»). В качестве измерений выбираем Время (таблица «DimTime»), Продукты (таблицы «DimProduct»), ВОУ(таблица «DimUniversity»).

Для создание хранилище данных с среде MSSQL создаем базу данных EducationDW. Хранилище данных будет служить для физического хранения очищенных и согласованных данных из разных источников. Для анализа

состоянии материально технической базе ВОУ создаем нижеследующие таблицы:

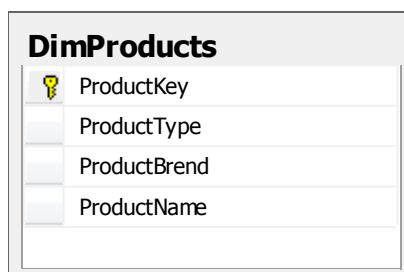
1. DimUniversity – таблица измерений. В этой таблице будут храниться обобщенные данные о высших образовательных учреждениях. Также создается столбец Регион для фильтрации данных



DimUniversity	
🔑	UniversityKey
	Name
	Region
	Phone
	Fax

*Рис 3.1. Таблица измерений DimUniversity*

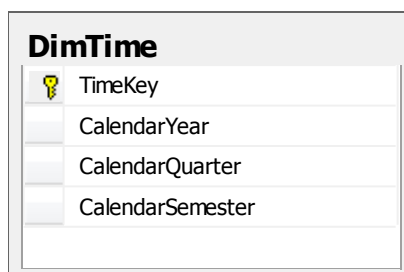
2. DimProducts-таблица измерения. В этой таблице будет храниться данные об объектах входящих в состав материально технической базе ВОУ. Например: компьютеры, принтеры, парты, маршрутизаторы, Телевизоры, стулья и т.д. . Объекты будут группироваться по типу (ProductType).



DimProducts	
🔑	ProductKey
	ProductType
	ProductBrend
	ProductName

*Рис 3.2. Таблица измерений DimProducts*

3. DimTime –таблица измерения времени. Нижний границе детализации выбран квартал года.



DimTime	
🔑	TimeKey
	CalendarYear
	CalendarQuarter
	CalendarSemester

*Рис 3.3. Таблица измерений DimTime*

4. FactEdu –таблица фактов. В таблице хранятся факты об объектах входящих в состав материальную техническую база ВОУ.


<b>FactEdu</b>	
	FactEduID
	TimeID
	ProductID
	UniversityID
	ProductsCount
	ProductsCountInEdu
	ProductsCountWorkPro
	StudentsCount

Рис 3.4. Таблица фактов FactEdu

После этого создаем связь между таблицами изменение и таблицей фактов.

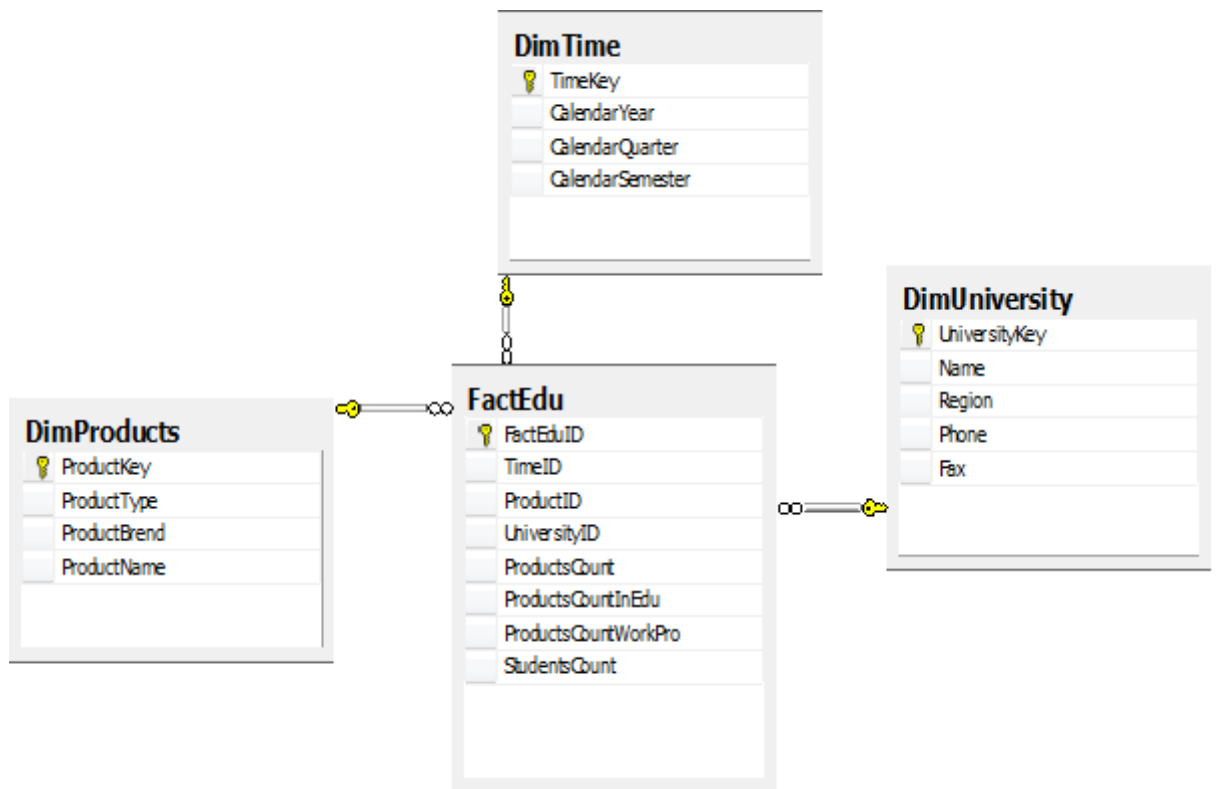


Рис 3.5. Диаграмма таблиц

Для создания куба был проведен сравнительный анализ OLAP-серверов.

### *3.1.1. Выбор OLAP-сервера*

Основные требования, предъявляемые к OLAP-серверу:

- возможность создания пользовательских функций для вычисления значений мер (в частности, создание пользовательских агрегирующих функций);
- доступ OLAP-клиентов по унифицированному протоколу;
- возможность добавления новых кубов;
- широкий спектр поддерживаемых СУБД.

На сегодняшний день существует множество поставщиков OLAP-серверов. Все крупные поставщики СУБД, такие как Oracle, Microsoft, IBM, Sybase, выпускают также и OLAP-сервера к своим СУБД. Помимо крупнейших производителей СУБД на рынке OLAP-продуктов фигурируют и другие компании, работающие в области Business Intelligence, такие как MicroStrategy, Pentaho, SAS Institute, Jedox.

Все современные OLAP-серверы удовлетворяют принципам аналитической обработки в реальном времени, изложенным Коддом в 1993 году, а позднее дополненными в 1995 году.

Основными проблемами использования OLAP-серверов являются проблемы производительности, разреженности данных и «взрыва» данных.

Проблема производительности связана к требованию к OLAP-продукту – времени отклика сервера (которое должно быть не более 5 секунд по тесту FASMI). Объем анализируемых данных часто превышает миллионы записей, поэтому такое время сложно достичь. Данная проблема производителями решается по-разному, но в основном, решения основываются на кешировании сосчитанных агрегатных (суммарных) значениях, либо на предварительном их вычислении и хранении в базе данных.

Предварительное вычисление и хранения в БД агрегатных значений порождает новую проблему – проблему эффекта «взрыва данных», т.е. к

резкому увеличению объёма хранимых данных. Это вызвано тем, что количество хранимых агрегатов растёт в экспоненциальной зависимости от числа измерений в кубе. Синдром взрыва может приводить к еще большим проблемам при разреженном распределении исходных данных по многомерному кубу. Отсутствующие или недопустимые значения данных создают разрежение в модели данных OLAP. Наиболее удачные OLAP-серверы борются с этой проблемой, не сохраняя пустые значения, таким образом, даже плохо заполненные кубы «не раздуваются» в объеме.

Все современные OLAP-серверы могут работать одновременно с несколькими кубами, поддерживать работу с большим числом измерений, мер и иерархий. Некоторые продукты поддерживают несколько иерархий для измерения, несбалансированные иерархии, полуаддитивные меры и некоторые другие функции, такие как «write-back», позволяющую аналитику изменить любое значение в таблице-отчёте, посмотреть к каким последствиям оно приведёт и при необходимости сохранить новое значение в хранилище данных.

У каждого сервера есть свои протоколы обмена информацией и языки запросов к многомерным данным – в области OLAP на данный момент не существует каких-либо стандартизованных протоколов обмена информацией. Но при этом почти каждый сервер поддерживает стандарты «де факто» обмена информацией – XMLA (XML for Analysis) и языка запросов к многомерным данным фирмы Microsoft – MDX (Multidimensional Expressions).

Для выбора OLAP-сервера, удовлетворяющего заявленным требованиям, было принято решение провести сравнительный анализ наиболее известных серверов.

### *3.1.2. Сравнительный анализ существующих OLAP-серверов*

В виду достаточно большого количества производителей OLAP-серверов было принято решение рассмотреть только наиболее известные и долго

работающие на рынке фирмы, а также ряд бесплатных продуктов. Таким образом, были выбраны следующие продукты (табл. 3.1):

Таблица 3.1. Исследуемые OLAP-сервера

	Фирма-производитель	Название OLAP-сервера
	Microsoft	Analysis Services
	Oracle	Essbase
	IBM	Cognos TM1
	Pentaho	Mondrian
	Jedox	Palo

Для исследования возможностей и сравнения OLAP-серверов необходимо определить критерии, по которым данные продукты можно сравнивать. Выбор критериев сравнения серверов происходил на основании требований, предъявляемых к OLAP-серверу, а также некоторых характеристик, влияющих на быстродействие сервера, гетерогенность и безопасность среды, в которой работает сервер.

Таким образом, были выбраны следующие критерии:

1. Список поддерживаемых СУБД.
2. Список поддерживаемых форм хранения данных (MOLAP/ROLAP/HOLAP).

3. Возможность создания вычисляемых меток. С помощью вычисляемых меток можно включать в многомерную базу меры, которых нет в исходных данных, но которые можно вычислить по формуле.

4. Возможность создания пользовательских агрегирующих функций. Агрегирующие функции – функции, вычисляющие суммарный результат, например сумма значений, среднее значение, количество элементов и др. Помимо стандартных агрегирующих функций, необходимо предусмотреть возможность создания пользовательских функций, вычисляющих агрегирующее значение по определённой формуле.

5. Оптимизация схемы агрегирования, т.е. возможность изменения количества вычисленных заранее агрегирующих значений (агрегатов). Чем больше агрегатов хранится в готовом виде, тем выше производительность системы и тем меньше среднее время ответа на запрос. В то же время, увеличение количества хранимых агрегатов приводит к увеличению объема хранимых данных. Возможность сохранять только те агрегаты, которые наиболее часто используются в запросах пользователей, позволяет получить наименьший объем хранилища данных при максимальной производительности для наиболее популярных запросов к БД. Поэтому данный критерий достаточно важен при выборе OLAP-сервера.

6. Масштабируемость и расширяемость. Масштабируемость – возможность работы с различными объемами данных: от очень маленьких (менее 10 тыс. записей) до очень больших (более 1 млн. записей) без изменения состава программного продукта. Расширяемость – возможность добавления новых многомерных кубов данных без изменения состава программного продукта.

7. Удобные средства создания кубов и администрирования.

8. Решение проблемы «взрыва данных».

9. Решение проблемы разреженности данных.

10. Список поддерживаемых API и языков запросов. Необходимо, чтобы API и язык запросов, используемые OLAP-сервером, поддерживались OLAP-клиентом.

11. Поддержка Unix-подобных операционных систем.

12. Документирование. OLAP-сервер должен иметь хорошую документацию для быстрого освоения продукта.

13. Цена продукта на минимальную конфигурацию.

Проанализируем каждый продукт по предложенным критериям:

#### **1) OLAP-сервер Microsoft Analysis Services**

Microsoft Analysis Services (MSAS) - часть Microsoft SQL Server, включающий в себя набор средств для работы с OLAP и интеллектуальным анализом данных. Последняя версия - Microsoft Analysis Services 2008.

1. MSAS поддерживает только одну СУБД - Microsoft SQL Server.
2. MSAS поддерживает все три вида хранения данных: MOLAP, ROLAP и HOLAP.
3. MSAS поддерживает создание вычисляемых меток. Следует отметить, что для этого предусмотрен удобный мастер, облегчающий создание вычисляемой метки и редактирования её формулы.
4. MSAS поддерживает создание пользовательских агрегирующих функций.
5. MSAS предусматривает оптимизацию схемы агрегирования. Используя Storage Design Wizard можно найти компромисс между быстродействием системы и размером хранимых на диске агрегатов.
6. MSAS поддерживает гибкую модель масштабирования, включающую оптимизацию хранения, компрессию данных и распределённые вычисления (т.е. часть вычислений можно перенести на клиент).
7. MSAS содержит множество различных «мастеров» и «дизайнеров» по созданию кубов, измерений, иерархий и администрированию.
8. Проблема «взрыва данных» в MSAS решена.
9. Проблема разреженности данных в MSAS также решена (посредством компрессии данных).
10. MSAS поддерживает спецификации OLE DB, ADO DB, XMLA. Язык запросов к многомерным данным – MDX.
11. MSAS может быть установлен только на ОС Windows совместно с Microsoft SQL Server.
12. MSAS очень подробно документирован. Документация на русском языке.
13. Стоимость SQL Server 2008 Workgroup Edition - 4000\$.

Следует отметить, что данный продукт поддерживает все типы иерархий и мер, а также функцию «write-back».

## **2) OLAP-сервер Oracle Essbase**

Oracle Essbase (в дальнейшем Essbase) – первый в мире OLAP-сервер. Данный сервер изначально создавался фирмой Hyperion как расширение электронных таблиц Lotus 1-2-3 и Microsoft Excel, но впоследствии в 2007 году Oracle купил Hyperion и переименовал продукт в Oracle Essbase. Последняя версия - Oracle Essbase V.11.1.2

1. Essbase поддерживает только свою СУБД – Oracle.
2. Essbase поддерживает 2 вида хранения данных: MOLAP и HOLAP.
3. Essbase поддерживает создание вычисляемых меток. Стоит отметить, что помимо обычных математических формул можно использовать процедурный язык программирования - PL/SQL.
4. Essbase поддерживает создание пользовательских агрегирующих функций, используя язык PL/SQL.
5. Essbase, так же как и MSAS, предусматривает гибкую оптимизацию схемы агрегирования.
6. Essbase поддерживает гибкую модель масштабирования, включающую кластеризацию, функции отказоустойчивости, оптимизацию хранения и компрессию данных.
7. Essbase содержит удобный графический интерфейс, напоминающий интерфейс Microsoft Office, поэтому освоить такой интерфейс не будет проблемой даже для новичка.
8. Проблема «взрыва данных» в Essbase решена.
9. Проблема разреженности данных в Essbase также решена.
10. Essbase поддерживает API для языков Java, C, Visual Basic и спецификацию XMLA. Язык запросов к многомерным данным – MDX.
11. Essbase может быть установлен как на ОС Windows, так и на Unix-подобную ОС.

12. Essbase, так же как и MSAS, очень подробно документирован. Документация на русском языке.

13. Стоимость Oracle Essbase V.11.1.2– 40 000 \$

Стоит отметить, что данный продукт поддерживает все типы иерархий и мер, а также функцию «write-back».

### **3) OLAP-сервер IBM Cognos TM1**

Изначально продукт разрабатывался фирмой Cognos, но в 2009 году Cognos была поглощена компанией IBM и продукт был переименован в IBM Cognos TM1. Основное отличие TM1 от предыдущих двух серверов заключается в том, что это запатентованный 64-разрядный OLAP-сервер, использующий подход In-memory OLAP. Данный подход состоит в кешировании исходных данных и агрегатов в оперативной памяти, тем самым увеличивается производительность сервера. Последняя версия - Cognos TM1 9.5

1. Сервер хранит данные в своих собственных многомерных структурах данных. Для импорта данных в свою БД используется дополнительный инструмент Turbo Integrator, поддерживающий большинство существующих СУБД. Подключение в данном случае происходит через ODBC-драйвер.

2. TM1 поддерживает один вид хранения данных – MOLAP.

3. TM1 поддерживает создание вычисляемых меток.

4. TM1 поддерживает создание пользовательских агрегирующих функций.

5. Помимо кеширования TM1 позволяет сохранять подсчитанные агрегаты в БД, тем самым предусматривается гибкая оптимизация схемы агрегирования.

6. TM1 поддерживает гибкую модель масштабирования, включающую кластеризацию, функции отказоустойчивости, оптимизацию хранения и компрессию данных.

7. TM1 содержит удобный графический интерфейс, встраиваемый в Microsoft Excel, а также собственный Web-интерфейс.

8. Проблема «взрыва данных» в TM1 решена.

9. Проблема разреженности данных в TM1 также решена.

10. TM1 поддерживает спецификации OLE DB, XMLA. Язык запросов к многомерным данным – MDX.

11. TM1 может быть установлен как на ОС Windows, так и на Unix-подобную ОС.

12. TM1, так же как Essbase и MSAS, очень подробно документирован. Документация на русском языке.

13. Стоимость IBM TM1 – 50 000 \$

Стоит отметить, что данный продукт поддерживает все типы иерархий и мер, а также функцию «write-back».

#### 4) **OLAP-сервер Pentaho Mondrian**

В отличие от вышеперечисленных OLAP-серверов данный продукт является открытым и распространяется под лицензией EPL. Mondrian написан на Java, что с одной стороны уменьшает быстродействие (ввиду java-машины), но с другой – добавляет свойство гетерогенности среды, в которой будет использоваться сервер. Так же, как и IBM TM1, данный сервер использует принцип In-Memory (кеширование агрегатов). Данный сервер используется в коммерческом продукте Pentaho Analysis. Последняя версия сервера - Mondrian 3.0.4

1. Для подключения к хранилищам данных Mondrian используется JDBC-драйвер. Таким образом, сервер поддерживает СУБД, к которым можно подключаться через JDBC.

2. Mondrian поддерживает только один вид хранения данных – ROLAP.

3. Mondrian поддерживает создание вычисляемых меток. Выражения для меток строятся на основе математических формул и языков SQL/MDX.

4. Mondrian не поддерживает создание пользовательских агрегирующих функций.

5. Mondrian изначально хранит все агрегаты в оперативной памяти, но существует возможность создавать для агрегатов свои таблицы в хранилище данных.

6. Mondrian масштабируем только с точки зрения увеличения исходных данных. Функций кластеризации и отказоустойчивости у сервера нет.

7. Mondrian содержит достаточно простой, но с другой стороны, удобный графический интерфейс для создания кубов. Графического интерфейса для администрирования сервера нет – всё администрирование сводится к изменениям параметров в XML-файлах.

8. Проблема «взрыва данных» в Mondrian решена.

9. Проблема разреженности данных в Mondrian также решена.

10. Mondrian поддерживает API для языков Java (OPAL4J) и спецификации XMLA и OLE DB. Язык запросов к многомерным данным – MDX.

11. Mondrian может быть установлен любую машину, для которой существует Java-машина. В частности, на Windows и Unix-подобную ОС.

12. Сервер документирован достаточно хорошо. Документация на английском языке.

13. Продукт предоставляется бесплатно. Кроме того доступны исходные коды продукта.

Следует отметить, что данный продукт поддерживает все типы иерархий, но не поддерживает полуаддитивные меры и функцию «write-back».

## **5) Jedox Palo**

Данный продукт, так же как Mondrian является открытым и распространяется под лицензией GPL. Так же, как IBM TM1 и Mondrian, данный сервер использует принцип In-Memory (кеширование агрегатов).

Данный сервер используется в коммерческом продукте Palo Suite. Последняя версия сервера - Palo 3.1

1. Сервер хранит данные в своих собственных многомерных структурах данных. Для импорта данных в свою БД используется продукт Palo ETL Server, поддерживающий большинство существующих СУБД.

2. Palo поддерживает только один вид хранения данных – MOLAP.

3. Palo поддерживает создание вычисляемых меток. Выражения для меток строятся на основе математических формул и языков SQL/MDX.

4. Palo не поддерживает создание пользовательских агрегирующих функций.

5. Palo хранит все агрегаты в оперативной памяти. Не существует возможности сохранять агрегаты на жёстком диске. Стоит отметить, что сервер использует процессор видеоадаптера (GPU), что способствует уменьшению времени вычисления агрегатов.

6. Palo масштабируем с точки зрения увеличения исходных данных. Palo очень эффективно использует ресурсы процессоров (CPU и GPU), поэтому сервер хорошо масштабируем в сторону увеличения количества процессов (ядер процессоров). Функций кластеризации и отказоустойчивости у сервера нет.

7. Palo содержит достаточно удобный графический интерфейс для создания кубов и администрирования сервера. Palo может быть встроен в Microsoft Excel и в Open Office, поэтому освоить такой интерфейс не будет проблемой даже для новичка.

8. Проблема «взрыва данных» в Palo решена.

9. Проблема разреженности данных в Palo также решена.

10. Palo поддерживает API для языков Java, C/C++, PHP, .NET и спецификации OLE DB и XMLA. Язык запросов к многомерным данным – MDX.

11. Palo может быть установлен как на ОС Windows, так и на Unix-подобную ОС.

12. Сервер документирован достаточно хорошо, но только на английском языке.

13. Продукт предоставляется бесплатно. Кроме того доступны исходные коды продукта.

Стоит отметить, что данный продукт поддерживает все типы иерархий и мер, а также функцию «write-back».

### 3.1.3. Результаты сравнительного анализа

Объединим результаты сравнения OLAP-серверов в одну сводную таблицу:

Таблица 3.2. Сравнительный анализ OLAP-серверов

Критерий/OLAP-сервер	MSAS	Essbase	TM1	Mondrian	Palo
Поддержка PostgreSQL	-	-	+	+	+
Поддерживаемые формы хранения данных (MOLAP/ROLAP/HOLAP)	(+ / + / +)	(+ / - / -)	(+ / - / -)	(- / + / -)	(+ / - / -)
Создание вычисляемых меток	+	+	+	+	+
Создание пользовательских агрегирующих функций	+	+	+	-	-
Оптимизация схемы агрегирования	+	+	+	+	-
Масштабируемость	+	+	+	+	+

Удобные средства создания кубов и администрирования	+	+	+	+	+
Решение проблемы «взрыва данных»	+	+	+	+	+
Решение проблемы разреженности данных	+	+	+	+	+
Поддержка XMLA и MDX	+	+	+	+	+
Поддержка Unix-подобных операционных систем	-	+	+	+	+
Подробное документирование на русском языке	+	+	+	-	-
Цена продукта на минимальную конфигурацию	4000\$	40000\$	50000\$	беспл.	беспл.

В ходе проведённого анализа можно сделать следующие выводы:

- все рассматриваемые продукты полностью соответствуют принципам аналитической обработки в реальном времени, изложенным Коддом, и тесту FASMI ;
- у платных продуктов очень гибкая схема масштабирования, включающая кластеризацию, функции отказоустойчивости;
- платные продукты поддерживают все типы иерархий и мер, а также функцию «write-back»;
- платные продукты очень подробно документированы (на русском языке);
- все рассматриваемые продукты поддерживают создание вычисляемых меток;

- бесплатные продукты не поддерживают создание пользовательских агрегирующих функций;
- во всех продуктах решена проблема «взрыва данных» и разреженности данных;
- все рассматриваемые продукты поддерживают спецификацию XMLA и язык запросов к многомерным данным - MDX;
- TM1, Mondrian, Palo используют принцип In-Memory OLAP, что увеличивает быстродействие OLAP-сервера;
- Mondrian и Palo имеют открытый исходный код.

Для предлагаемого подхода было принято решение использовать OLAP-сервер Microsoft Analysis Services, исходя из следующих соображений:

- Министерством уже куплено Microsoft Sql Server R8 2008;
- Microsoft Analysis Services обеспечивают высокую производительность работы приложений и масштабируемость на уровне миллионов записей и тысяч пользователей;
- Графический интерфейс создания и администрирования кубов у Microsoft Analysis Services очень удобный.

### 3.2. BI-решения Microsoft Analysis Services

BI-решение компании Microsoft основывается на масштабируемой платформе, предназначенной для интеграции данных, построения хранилищ данных, анализа накопленных данных и построения отчетов. Ядром BI-решения Microsoft является платформа SQL Server. Входящие в нее компоненты, специфические для BI-решения, приведены в таблица 3.3.

*Таблица 3.3. Компоненты BI-решения Microsoft*

Компонент	Описание
SQL Server Database	Масштабируемая,

Engine	высокопроизводительная СУБД, способная хранить большие объемы данных, образующиеся в результате консолидации данных в единое хранилище для анализа и построения отчетов.
SQL Server Integration Services	Платформа для выполнения операций извлечения, преобразования и загрузки, которая обеспечивает заполнение ХД и его синхронизацию с данными из различных источников, которые используются бизнес-приложениями организации.
SQL Server Analysis Services	Обеспечивает возможность построения OLAP-решений, включая возможность расчета ключевых индикаторов производительности (KPI). Применяется также для построения data-mining-решений, которые используют специализированные алгоритмы для выявления трендов и зависимостей в бизнес-данных.
SQL Server Reporting Services	Инструментарий построения отчетов, предназначенный для создания, публикации и распространения детализированных бизнес-отчетов, как для внутренних, так и для внешних целей.

**Этапы создания многомерного куба. Для создания куба нужно:**

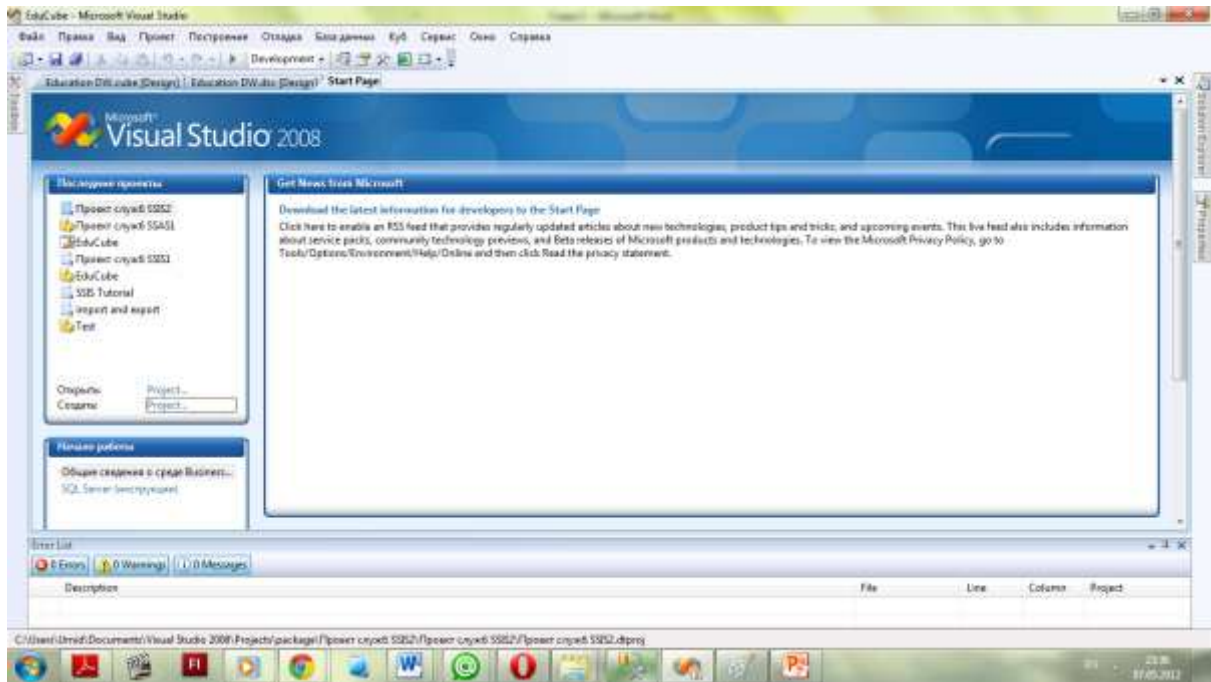
1. Создать источник данных;

2. Создать представления источника данных;
3. Создать измерений для куба и отредактировать;
4. Выбрать в источнике данных таблицу фактов, а в ней указать, какие меры будут использованы в кубе;
5. Выбрать уже созданные измерения куба;
6. Сохранить куб;
7. Отредактировать схему куба;
8. Создать пользовательские меры;
9. Загрузить данные в куб.

OLAP manager предлагает для администрирования OLAP Services набор мастеров и специальных инструментов. Мастера позволяют выполнить типовые действия самым простым способом и в нужной последовательности. Специальные инструменты имеют большую гибкость, но в свою очередь требуют от большей квалификации. Ряд административных действий можно выполнить минимум двумя путями: при помощи мастера или специального инструмента. Зачастую одно средство вызывается из другого — например, добавить измерение в куб можно непосредственно в Cube Editor либо вызвать Dimension Manager, который в свою очередь может вызвать Dimension Wizard.

Все административные средства оборудованы интуитивно понятным графическим интерфейсом, позволяющим легко добиться желаемого результата.

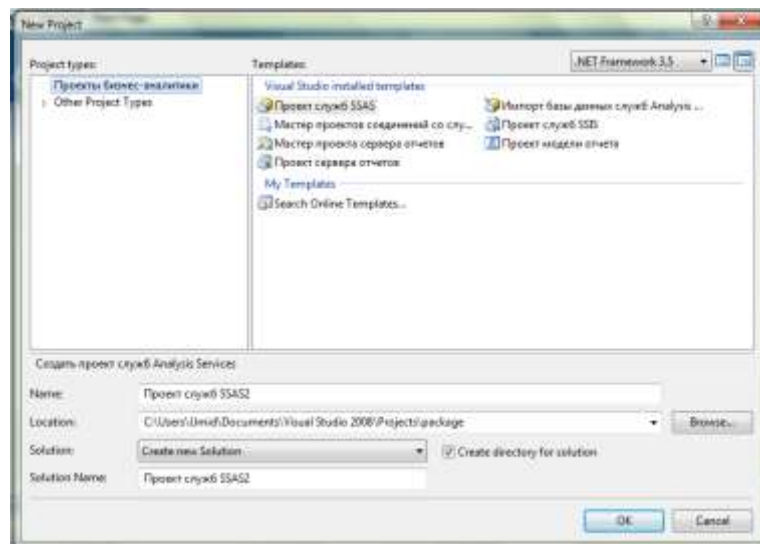
В меню «Пуск» последовательно выбираются пункты «Все программы», «Microsoft SQL Server 2008», а затем выбирается «Среда SQL Server Business Intelligence Development Studio». Открывается среда разработки Microsoft Visual Studio (рис. 3.6.).



*Рис. 3.6. Среда разработки Visual Studio*

В меню «Файл» Visual Studio указывается команду «Создать», затем выбирается пункт «Проект».

В диалоговом окне «Новый проект» на панели «Типы проектов» выбирается значение «Проекты бизнес-аналитики», а на панели «Шаблоны» указывается «Проект служб SSAS» (рис 3.7).



*Рис. 3.7. Выбор типа создаваемого проекта*

Обратите внимание, что в нижней части этого диалогового окна отображаются установленные по умолчанию имя проекта, имя решения и путь к проекту. По умолчанию для решения создается новый каталог.

Изменяем имя проекта на EduCube (при этом изменится и имя решения) и нажмите кнопку ОК(рис 3.8.).

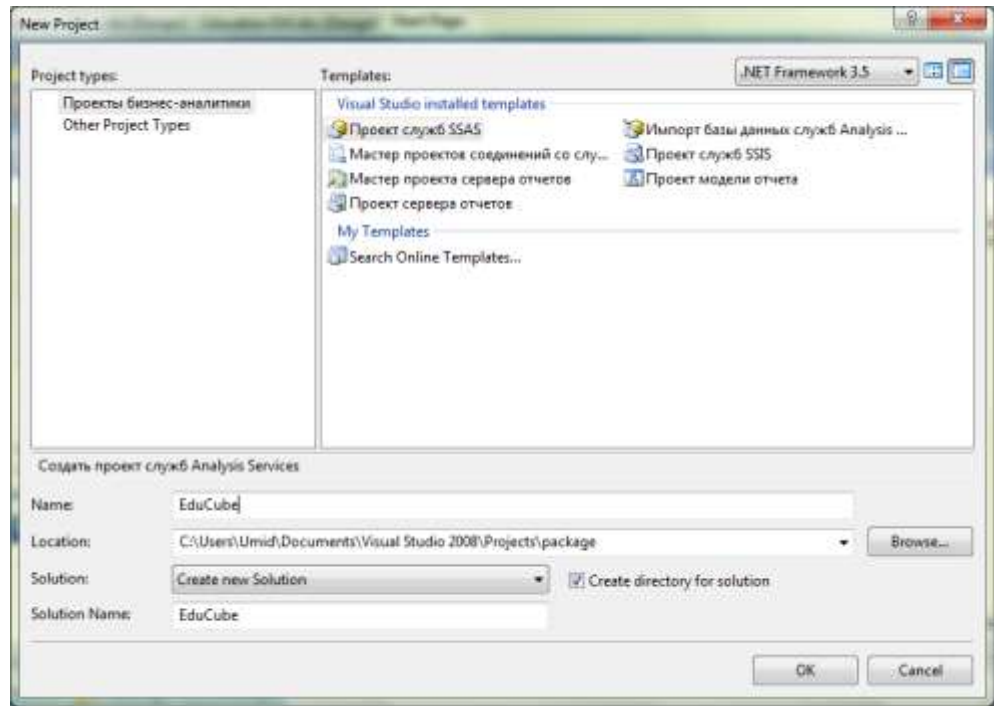


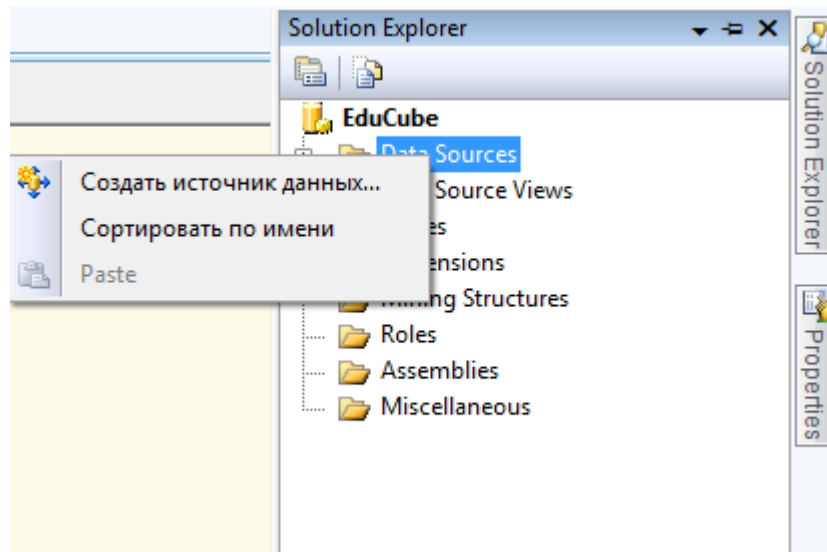
Рис. 3.8. Указание имени проекта

Проект EduCube, основанный на шаблоне проекта Analysis Services, будет создан в рамках нового решения, которое также называется EduCube.

После создания проекта служб Analysis Services работа с проектом обычно начинается с определения одного или нескольких источников данных, которые будут использоваться в этом проекте. Для определения источника данных нужно задать строку соединения, которая будет использована для подключения к этому источнику данных.

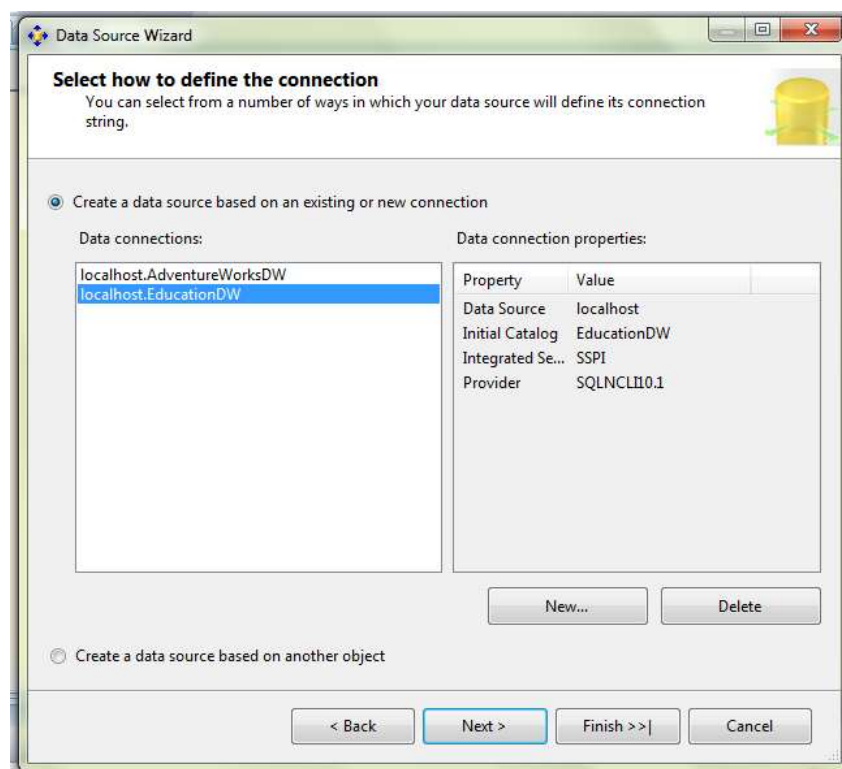
Далее в рамках выполнения работы необходимо указать образец базы данных EducationDW в качестве источника данных для проекта служб Analysis Services Tutorial. Хотя эта база данных в данном случае расположена на локальном компьютере, часто исходные базы данных размещаются на одном или нескольких удаленных компьютерах.

Чтобы запустить мастер создание источника данных нужно щелкнуть правой кнопкой мыши в окне обозревателе решений элемент «Источники данных» и выбрать команду «Создать источник данных» (рис 3.9.).



*Рис. 3.9. Контекстное меню элемента "Источники данных" в обозревателе решений*

На странице «Мастер источников данных» нажимается кнопка «Далее», потом открывается страница «Выбор метода определения соединения»(3.10.).



*Рис. 3.10. Выбор метода определения соединения*

На странице «Выбор метода определения соединения» можно определить источник данных на основе нового соединения, существующего соединения или предварительно определенного объекта источника данных. В данной работе будет определен источник данных на основе нового соединения, для этого нажимается кнопка «Создать».

В диалоговом окне Диспетчер соединений определяются свойства соединения для источника данных. В текстовое поле Имя сервера вводится localhost (рис. 3.11.).

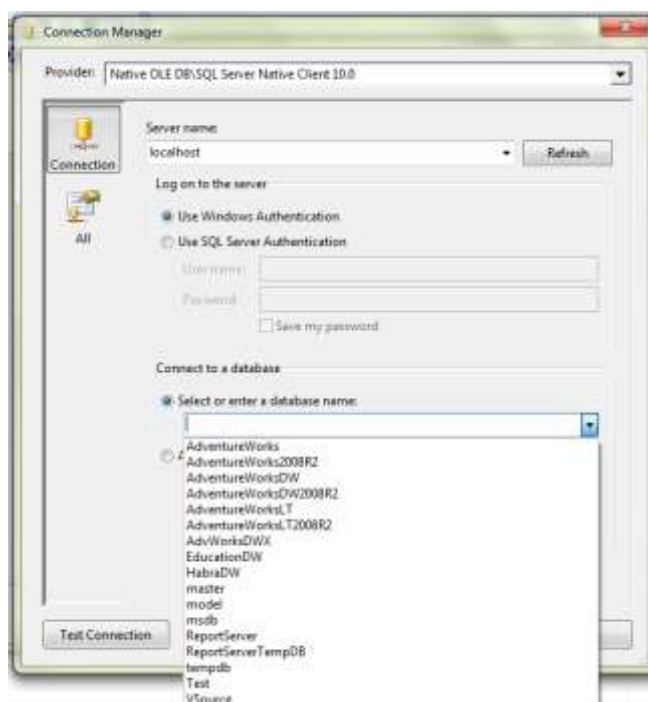


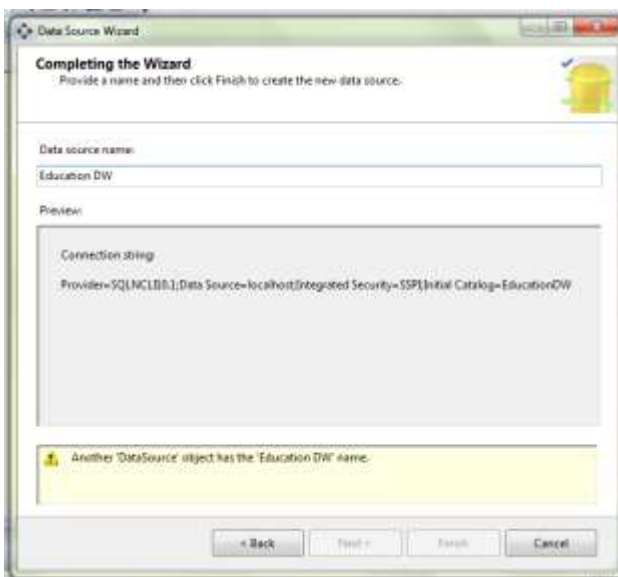
Рис. 3.11. Соединение с хранилищем данных EducationDW

В раскрывающемся списке «Выберите или введите имя базы данных» выбирается «EducationDW». На странице мастера «Сведения об олицетворении» определяются учетные данные безопасности, которые будут использованы в службах Analysis Services для подключения к источнику данных. Олицетворение влияет на учетную запись Windows, используемую для подключения к источнику данных, если выбран метод проверки подлинности Windows. Службы Analysis Services не поддерживают олицетворения при работе с объектами OLAP. Выберется параметр «Использовать учетную запись службы» и нажимается кнопка «Далее» (рис 3.12.).

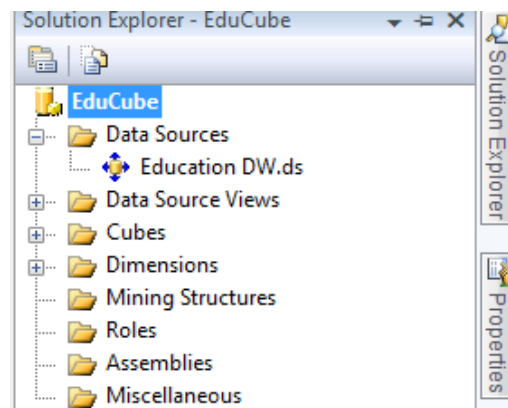


*Рис. 3.12. Сведения об олицетворении*

На странице «Завершение работы мастера» вводится имя EducationDW, затем нажимается кнопка «Готово» (Рисунок 3.13.), чтобы создать новый источник данных (Рисунок 3.14.).

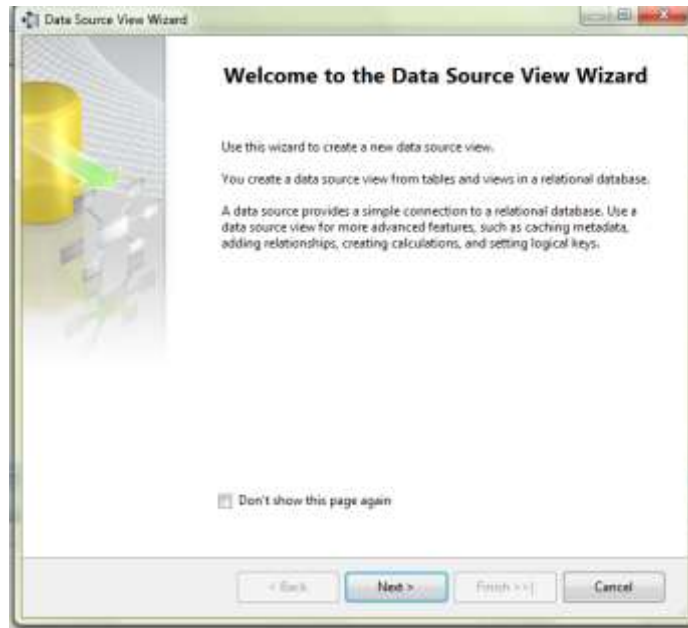


*Рис. 3.13. Имя источника данных*



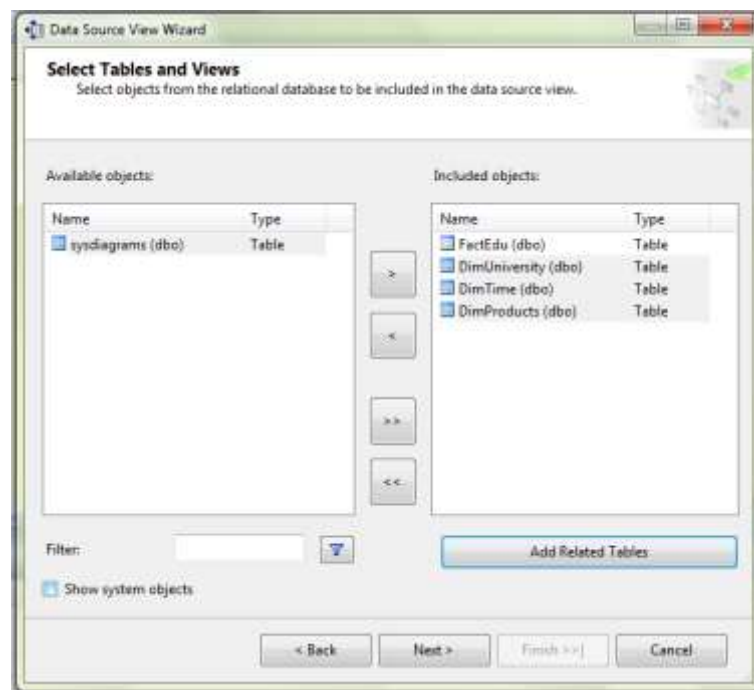
*Рис. 3.14. Созданный источник данных*

После этого подключается представления источника данных. Для этого запускается мастер «Определения представления источника данных»(рис 3.14.)



*Рис. 3.14. Мастер «Определения представления источника данных»*

С помощью мастера подключается Хранилище данных, выбираются таблицы измерений и фактов, которые нужны для создания гиперкуба (рис 3.15.). В конце задается имя для представления источника данных.



*Рис. 3.15. Окно для подключения таблиц измерений и фактов*

Если нажать на созданную представления источника данных (рис. 3.16.) увидим наши таблицы и автоматически сгенерированную диаграмму,

показывающую таблицы и связи между ними. С помощью диаграмме можно создать недостающие связи между таблицами.

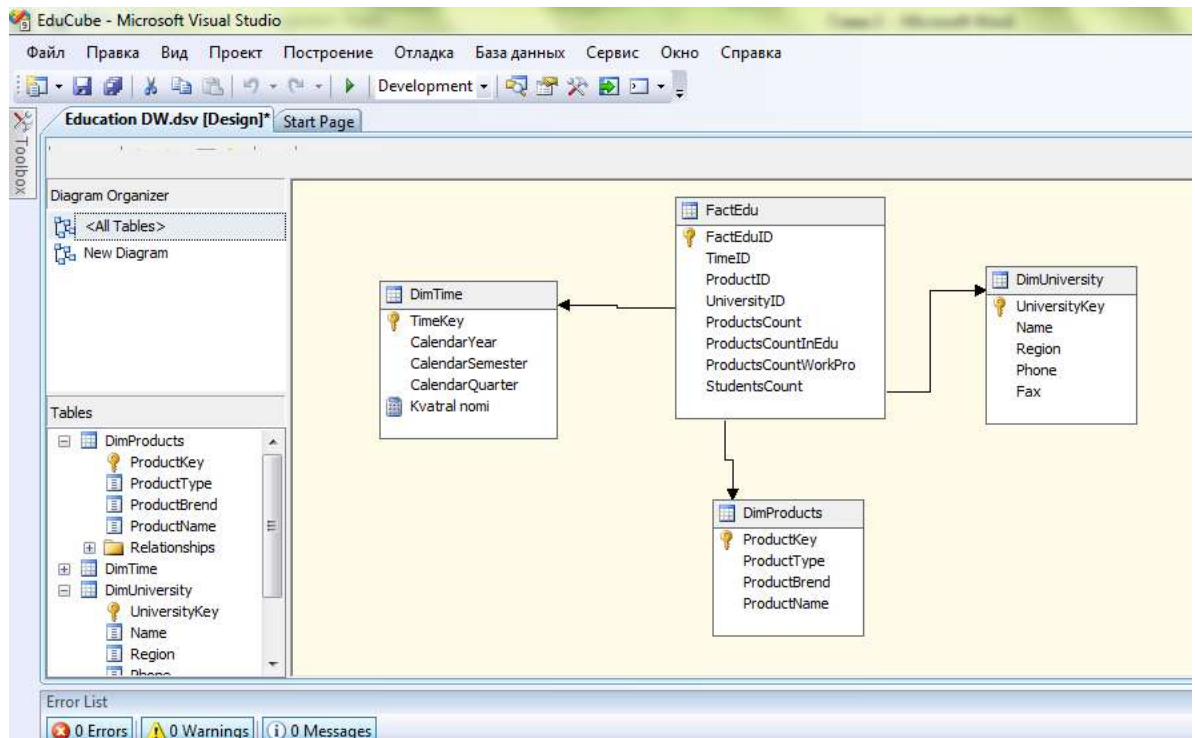


Рис. 3.16. Окно представления источника данных

Теперь создается измерения для многомерного куба, для этого запускаем мастер измерений. Первым создается измерения время на основе таблиц «DimTime». Выбирается нужные столбцы из таблицы (рис. 3.17.), которые будут использованы в кубе. Задается имя для измерения и завершается процесс.

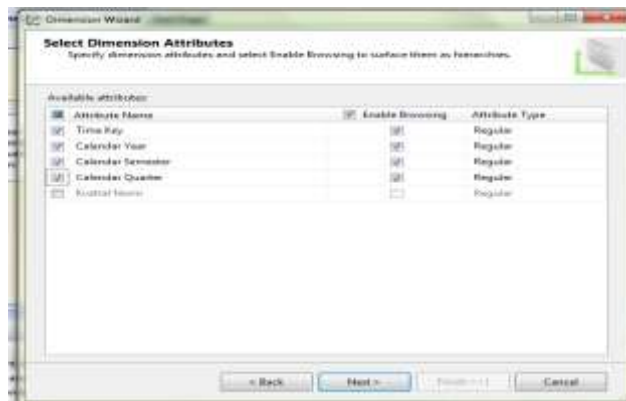


Рис. 3.17. Выбор столбцов из таблицы

После этого точно также создается другие измерения, в нашем случае измерение продукты, высшие образовательные учреждения. Если нажать на измерения откроется окна структуре измерения, показывающее исходную таблицу, а также элементы измерения (рис. 3.18.).

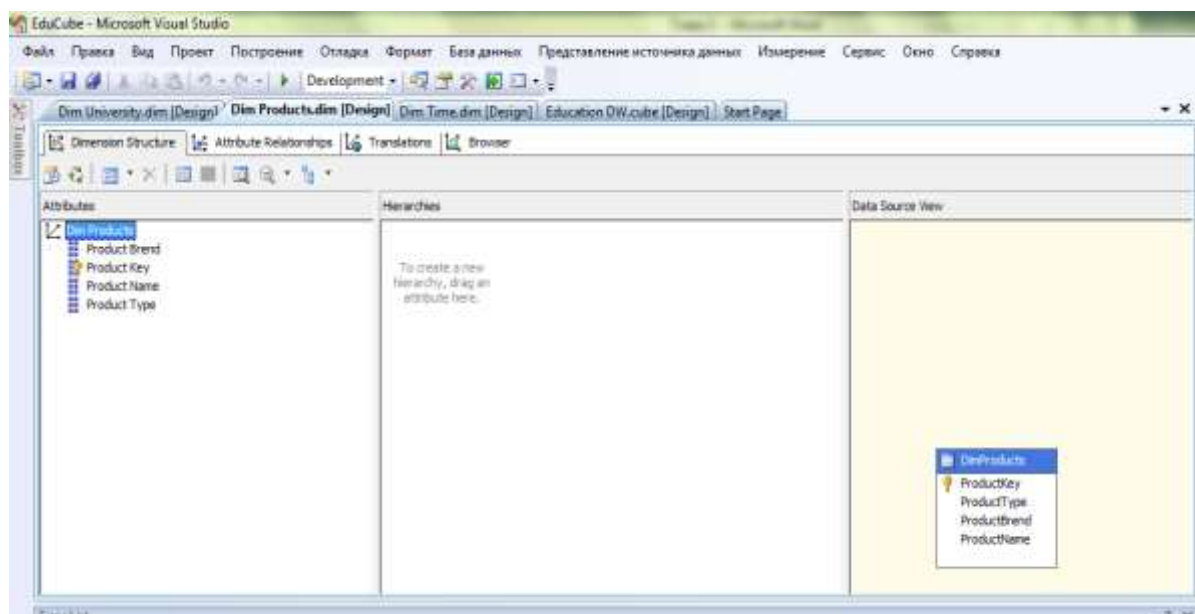


Рис. 3.18. Окно измерений

После нужных настроек измерений создается многомерный куб. Запускается мастер создания куба, выбирается таблица фактов (рис 3.19.) , и измерений, созданные ранние. Задается имя кубу и завершается процесс определение куба.



Рис. 3.19. Выбирается таблица фактов для куба

После этого разворачивается куб, и начинается процесс получения данных с ХД. Если системой не будет найдено ошибки, служба Analysis Services откроет окно куба (рис. 3.20.). Также выбрав нужные измерения и нужные меры можно увидеть срез куба (рис. 3.21.).

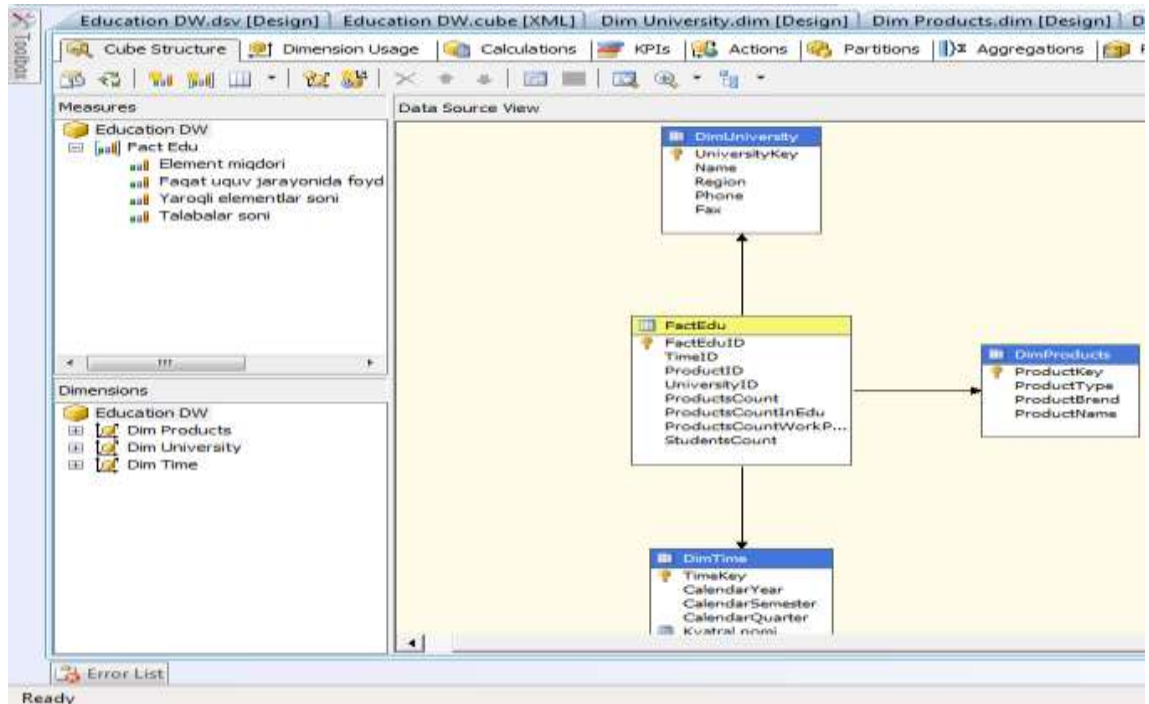


Рис. 3.20. Вкладка «Структура» в конструкторе кубов

Перетащите сюда поля фильтра				Product Brand ▾			
Calendar Year ▾	Calendar Semester ▾	Region ▾	Univer_Name	Acer	Faqat uquv j	Canon	Faqat uk
2007				4 044	2902	3 868	2961
2008				3 427	2660	4 048	2349
2009				2 914	2982	3 561	2539
2010				3 607	2511	3 821	2156
2011	1	Jizzax viloyati	Jizzax Pedagogika Instituti	575	357	263	92
			Jizzax Politehnika Instituti	43	273	278	217
			Итоги	618	630	541	309
		Sirdaryo viloyati	Guliston Davlat Universiteti	378	260	341	331
			Итоги	378	260	341	331
		Toshkent shahri		884	908	612	689
			Итоги	1 880	1798	1 494	1329
	2	Jizzax viloyati	Jizzax Pedagogika Instituti	260	276	231	138
			Jizzax Politehnika Instituti	448	280	300	169
			Итоги	708	556	531	307
		Sirdaryo viloyati	Guliston Davlat Universiteti	444	300	215	209
			Итоги	444	300	215	209
		Toshkent shahri	Jizzax Politehnika Instituti (Univer_Name)†	1 060	792	1 230	377
			Итоги	2 212	1648	1 976	893
			Итоги	4 092	3446	3 470	2222
2012				1 993	1152	2 071	1382
Общие итоги				20 077	15653	20 839	13609

Рис. 3.21. Вкладка «Обозреватель» в конструкторе кубов

Теперь нужно создать пользовательскую меру, определяющую соотношения количество студентов к количеству компьютеров. Для этого во

вкладке «Калькуляция» конструктора кубов создается вычисляемый элемент, и для этого элемента задается выражения:

[Measures].[Element miqdori]/[Measures].[Talabalar soni]

### 3.3. Выводы по третьей главе

1. На основе спроектированной ХД реализована реляционная база данных в Microsoft SQL Server 2008 R2. Разработан многомерный куб «Анализ показателей материально технической базы ВОУ». Произведен сравнительный анализ OLAP серверов, и на основы результатов анализа выбран Microsoft Analysis Services.

2. На основе выбранного OLAP сервера поэтапно создается многомерный куб. В начальном этапе создан источник данных, представления источника данных, измерения для многомерного куба. Для удобного использование отредактирована измерение куба. Из созданного источника данных выбран таблица фактов, и на основе его указаны меры, которые будут использованы в кубе. Из созданных измерений разработан многомерный куб. Для анализа и представления данных отредактирован схема куба. Также создан пользовательский мера «Соотношение объектов входящий материальную техническую базу данных ВОУ на контингент студентов».

## ЗАКЛЮЧЕНИЕ

В ходе проведенных работ и исследований автором получены следующие научно-практические результаты:

1. В результате анализа предметной области показана актуальность задачи построения отраслевого центра накопления информации на основе технологии хранилищ данных. Выявлены недостатки существующего инструментария, связанные с особенностями области управления образованием.

2. Обоснована необходимость разработки программного обеспечения для управления централизованным хранилищем информации.

3. Предложена функционально-информационная модель централизованного хранилища данных. Хранилище данных спроектировано для анализа материально технической базе высших образовательных учреждений.

4. В среде SQL Server 2008 R2 на основе созданной Хранилищ данных разработан многомерный куб для анализа данных. В измерениях многомерного куба представлена высшие образовательные учреждения, время, также объекты входящие материальную техническую базу ВОУ.

## ЛИТЕРАТУРА

1. Указ Президента Республики Узбекистан от 30 мая 2002 года «О дальнейшем развитии компьютеризации и внедрении информационно-коммуникационных технологий».

2. Вступительный речь Президента Республики Узбекистан Ислама Каримова на открытии международной конференции «Подготовка образованного и интеллектуально развитого поколения – как важнейшее условие устойчивого развития и модернизации страны» 16.02.2012. <http://uza.uz/ru/politics/18080/>.

3. Постановление Президента Республики Узбекистан 10 мая 2011 года «О мерах по укреплению материально-технической базы высших образовательных учреждений и кардинальному улучшению качества подготовки высококвалифицированных специалистов».

4. Архипенков С. Хранилища данных. / Д. Голубев, О. Максименко – М.: Диалог-МИФИ, 2002.

5. Барсегян А. Методы и модели анализа данных: OLAP и Data Mining СПб.: БХВ-Петербург, 2004.

6. Бергер А. Microsoft SQL Server 2005 Analysis Services. OLAP и многомерный анализ данных. – СПб.: БХВ-Петербург, 2007.

7. Спирли Э. Корпоративные хранилища данных. Планирование, разработка и реализация. – М.: Вильямс, 2001.

8. Харинатх С. SQL Server 2005 Analysis Services и MDX для профессионалов. / С. Куинн – М.: Диалектика, 2008.

9. Козлов В.А. Открытые информационные системы. — М., 1999.

10. Бойченко А.В., Кондратьев В.К., Филинов Е.Н. Основы открытых информационных систем. Учебное пособие. — М., 2001.

11. Дубров А.М., Мхитарян В.С. Многомерные статистические методы и основы эконометрики. — М., 1998.

12. Андрейчиков А.В., Андрейчикова О.Н. Анализ, синтез, планирование решений в экономике. — М., 2000.
13. Бритов П.А., Липчинский Е.А. Практика построения хранилищ данных: Система SAS. Ж. Системы управления базами данных. 04-05/1998/
14. Business Objects Руководство пользователя. Материалы фирмы Терн, 1996.
15. Калянов Г.Н. Консалтинг при автоматизации предприятий: научно-методическое издание. — М.: СИНТЕГ, 1997.
16. Кузьмичева Т.С. Контур Стандарт. Технология работы с системой. Руководство пользователя. ISL. — М., 1999.
17. Ивантер А. Деловой барометр показывает «Ясно» ж. «Эксперт» № 29, авг. 2001.
18. Корнев В.В., Гареев А.Ф., Васютин С.В., Райх В.В. Базы данных. Интеллектуальная обработка информации. — М., 2000.
19. Маклаков С.В. Erwin. CASE-средства разработки информационных систем. — М., 2000.
20. Харрингтон Д. Проектирование объектно-ориентированных баз данных / Пер. с англ. — М.: ДМК Пресс, 2001.
21. Кирстен В., Ирингер М., Рериг Б., Шульте П. СУБД Cache: объектно-ориентированная разработка приложений. Учебный курс. — СПб.: Питер, 2001.
22. Пендс Н. Перевод Абушаева Ш. Что относится к OLAP? OLAPReport.com Last updated on Jan 1999.
23. Щавелев Л.В. Способы аналитической обработки данных для поддержки принятия решений. — Ж. СУБД 04-05/1998/
24. Белов В.С. Информационно-аналитические системы. Учебное пособие МЭСИ. — М., 2001.
25. Шрек М. Перевод Intersoft Lab Технология Хранилищ данных: последние 10 лет, несомненно были впечатляющими <http://www.iso.ru> 08/01.

26. Косов Д.В. Хранилища данных и семантические разрывы. Olap.ru. 2000.
27. Коломиец С.В. Проектирование процессов перегрузки данных. Olap.ru. 2000.
28. Федечкин С. Хранилище данных: вопросы и ответы. PCWeek, 31, 26.08.2003.
29. Строддер Д., Бурьесси Ж., Янг М. Лучшие ИТ-поставщики на 2004 год. Intelligent enterprise. Деловой ИТ-журнал. 26.01.2004.
30. Федоров А.Г., Елманова Н.З. Введение в OLAP-технологии Microsoft. Учебно-справочное издание. — М.: Диалог-МИФИ, 2002.
31. Белов В.С. Информационно-аналитические системы. Основы проектирования и применения. Учебное пособие. — М: МЭСИ, 2004.
32. Хамидов У.Х., Маткаримов С. Разработка и внедрение информационно-аналитическую интегрированную систему в сфере высшего образования для повышения эффективность электронного документооборота. Научный журнал. «Иқтисодиётни модернизация қилиш шароитида ишлаб чиқариш соҳаларини инновацион ривожлантириш муаммолари» Тош. ,2012 г.
33. Хамидов У.Х., Гоипназаров Т.С. «Ўқув жараёнини бошқариш масалаларини ечиш учун маълумотлар омборини ишлаб чиқиш» Научный журнал. Тош. ,2012 г.

Интернет-ресурсы:

34. <http://ru.wikipedia.org/wiki> - свободная Интернет-энциклопедия.
35. <http://www.fpm.com/refer/codd.html> - статья Е.Ф. Кодда, С.Б. Кодда и С.Т. Солли "OLAP для пользователей-аналитиков: информационно-технологический мандат".
36. <http://www.olap.ru> – сайт о бизнес-аналитике.
37. <http://www.microsoft.com/msdn> - Microsoft Developer Network

## ПРИЛОЖЕНИЕ

### Приложение 1. Исходный код в виде XMLa многомерного куба.

```

<Cube xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ddl2="http://schemas.microsoft.com/analysisservices/2003/engine/2"
xmlns:ddl2_2="http://schemas.microsoft.com/analysisservices/2003/engine/2/2"
xmlns:ddl100_100="http://schemas.microsoft.com/analysisservices/2008/engine/100/
100"
xmlns:ddl200="http://schemas.microsoft.com/analysisservices/2010/engine/200"
xmlns:ddl200_200="http://schemas.microsoft.com/analysisservices/2010/engine/200/
200" xmlns:dwd="http://schemas.microsoft.com/DataWarehouse/Designer/1.0"
dwd:design-time-name="e6c974d7-f915-4829-8f82-b177f58813cb"
xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
  <ID>Education DW</ID>
  <Name>Education DW</Name>
  <CreatedTimestamp>0001-01-01T00:00:00Z</CreatedTimestamp>
  <LastSchemaUpdate>0001-01-01T00:00:00Z</LastSchemaUpdate>
  <Annotations>
    <Annotation>

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramLayout
</Name>
  <Value>
    <dds xmlns="">
      <diagram fontclsid="{0BE35203-8F91-11CE-9DE3-00AA004BB851}"
mouseiconclsid="{0BE35204-8F91-11CE-9DE3-00AA004BB851}"
defaultlayout="MSDDS.Rectilinear" defaultlineroute="MSDDS.Rectilinear"

```

```

version="7"      nextobject="16"      scale="100"      pagebreakanchorx="0"
pagebreakanchory="0" pagebreaksizex="0" pagebreaksizey="0" scrollleft="-2581"
scrolltop="4282" gridx="150" gridy="150" marginx="5000" marginy="5000"
zoom="84" x="22913" y="8493" bgcolor="15334399" defaultpersistence="2"
PrintPageNumbersMode="3" PrintMarginTop="0" PrintMarginBottom="635"
PrintMarginLeft="0" PrintMarginRight="0" marqueeselectionmode="0"
mousepointer="0" snaptogrid="0" autotypeannotation="1" showscrollbars="1"
viewpagebreaks="0" donotforceconnectorsbehindshapes="1"
backpictureclsid="{00000000-0000-0000-0000-000000000000}">

```

```
<font>
```

```
<ddsxmobjectstreamwrapper
```

```
binary="01cc0000900144420100065461686f6d61" />
```

```
</font>
```

```
<mouseicon>
```

```
<ddsxmobjectstreamwrapper binary="6c74000000000000" />
```

```
</mouseicon>
```

```
</diagram>
```

```
<layoutmanager>
```

```
<ddsxmlobj />
```

```
</layoutmanager>
```

```
<ddscontrol      controlprogid="DdsShapes.DdsObjectManagedBridge.2"
```

```
tooltip="FactEdu" left="20" top="5549" logicalid="8" controlid="1" masterid="0"
```

```
hint1="0" hint2="0" width="4180" height="4419" noresize="0" nomove="0"
```

```
nodefaultattachpoints="0" autodrag="1" usedefaultiddshape="1" selectable="1"
```

```
showselectionhandles="1" allownudging="1" isannotation="0" dontautolayout="0"
```

```
groupcollapsed="0" tabstop="1" visible="1" snaptogrid="0">
```

```
<control>
```

```
<ddsxmobjectstreaminitwrapper binary="000800005410000043110000" />
```

```
</control>
```

```

<layoutobject>
  <ddsxmlobj>
    <property name="LogicalObject" value="dbo_FactEdu" vartype="8" />
  </ddsxmlobj>
</layoutobject>
<shape groupshapeid="0" groupnode="0" />
</ddscontrol>
<ddscontrol      controlprogid="DdsShapes.DdsObjectManagedBridge.2"
tooltip="DimTime"  left="363"  top="12368"  logicalid="9"  controlid="2"
masterid="0"  hint1="0"  hint2="0"  width="3493"  height="2725"  noresize="0"
nomove="0"  nodefaultattachpoints="0"  autodrag="1"  usedefaultiddshape="1"
selectable="1"  showselectionhandles="1"  allownudging="1"  isannotation="0"
dontauiolayout="0"  groupcollapsed="0"  tabstop="1"  visible="1"  snaptogrid="0">
  <control>
    <ddsxmlobjectstreaminitwrapper binary="00080000a50d0000a50a0000" />
  </control>
</layoutobject>
<ddsxmlobj>
  <property name="LogicalObject" value="dbo_DimTime" vartype="8" />
</ddsxmlobj>
</layoutobject>
<shape groupshapeid="0" groupnode="0" />
</ddscontrol>
<ddscontrol      controlprogid="DdsShapes.DdsObjectManagedBridge.2"
tooltip="DimProducts"  left="6600"  top="6396"  logicalid="10"  controlid="3"
masterid="0"  hint1="0"  hint2="0"  width="3000"  height="2725"  noresize="0"
nomove="0"  nodefaultattachpoints="0"  autodrag="1"  usedefaultiddshape="1"
selectable="1"  showselectionhandles="1"  allownudging="1"  isannotation="0"
dontauiolayout="0"  groupcollapsed="0"  tabstop="1"  visible="1"  snaptogrid="0">

```

```

<control>
  <ddsxmlobjectstreaminitwrapper binary="00080000b80b0000a50a0000" />
</control>
<layoutobject>
  <ddsxmlobj>
    <property name="LogicalObject" value="dbo_DimProducts" vartype="8"
/>

  </ddsxmlobj>
</layoutobject>
<shape groupshapeid="0" groupnode="0" />
</ddscontrol>
<ddscontrol      controlprogid="DdsShapes.DdsObjectManagedBridge.2"
tooltip="DimUniversity" left="610" top="0" logicalid="11" controlid="4"
masterid="0" hint1="0" hint2="0" width="3000" height="3149" noresize="0"
nomove="0" nodefaultattachpoints="0" autodrag="1" usedefaultiddshape="1"
selectable="1" showselectionhandles="1" allownudging="1" isannotation="0"
dontautolayout="0" groupcollapsed="0" tabstop="1" visible="1" snaptogrid="0">
  <control>
    <ddsxmlobjectstreaminitwrapper binary="00080000b80b00004d0c0000" />
  </control>
  <layoutobject>
    <ddsxmlobj>
      <property name="LogicalObject" value="dbo_DimUniversity" vartype="8"
/>

    </ddsxmlobj>
  </layoutobject>
  <shape groupshapeid="0" groupnode="0" />
</ddscontrol>

```

```

<ddscontrol controlprogid="MSDDS.Polyline" left="3900" top="7359"
logicalid="12" controlid="5" masterid="0" hint1="0" hint2="0" width="3000"
height="799" noresize="0" nomove="0" nodefaultattachpoints="1" autodrag="0"
usedefaultidshape="0" selectable="1" showselectionhandles="0" allownudging="1"
isannotation="0" dontautolayout="0" groupcollapsed="0" tabstop="1" visible="1"
snaptogrid="0">
  <control>
    <ddsxmlobj>
      <polyline endtypedst="6" endtypesrc="3" usercolor="0" linestyle="0"
linerender="1" customendtypedstid="0" customendtypesrcid="0" adornsvisible="1"
/>
    </ddsxmlobj>
  </control>
  <layoutobject>
    <ddsxmlobj>
      <property name="LogicalObject"
value="dataSet.Relations[FK_FactEdu_DimProducts]" vartype="8" />
      <property name="Virtual" value="0" vartype="11" />
      <property name="VisibleAP" value="0" vartype="3" />
    </ddsxmlobj>
  </layoutobject>
  <connector lineroutestyle="MSDDS.Rectilinear" sourceid="3" destid="1"
sourceattachpoint="14" destattachpoint="21" segmenteditmode="0"
bendpointeditmode="0" bendpointvisibility="0" relatedid="0" virtual="0">
    <point x="6600" y="7758" />
    <point x="4200" y="7758" />
  </connector>
</ddscontrol>

```

```

<ddscontrol controlprogid="MSDDS.Polyline" left="1711" top="2650"
logicalid="13" controlid="6" masterid="0" hint1="0" hint2="0" width="799"
height="3199" noresize="0" nomove="0" nodefaultattachpoints="1" autodrag="0"
usedefaultiddshape="0" selectable="1" showselectionhandles="0" allownudging="1"
isannotation="0" dontautolayout="0" groupcollapsed="0" tabstop="1" visible="1"
snaptogrid="0">
  <control>
    <ddsxmlobj>
      <polyline endtypedst="6" endtypesrc="3" usercolor="0" linestyle="0"
linerender="1" customendtypedstid="0" customendtypesrcid="0" adornsvisible="1"
/>
    </ddsxmlobj>
  </control>
  <layoutobject>
    <ddsxmlobj>
      <property name="LogicalObject"
value="dataSet.Relations[FK_FactEdu_DimUniversity]" vartype="8" />
      <property name="Virtual" value="0" vartype="11" />
      <property name="VisibleAP" value="0" vartype="3" />
    </ddsxmlobj>
  </layoutobject>
  <connector lineroutestyle="MSDDS.Rectilinear" sourceid="4" destid="1"
sourceattachpoint="5" destattachpoint="6" segmenteditmode="0"
bendpointeditmode="0" bendpointvisibility="0" relatedid="0" virtual="0">
    <point x="2110" y="3149" />
    <point x="2110" y="5549" />
  </connector>
</ddscontrol>

```

```

<ddscontrol controlprogid="MSDDS.Polyline" left="1710" top="9668"
logicalid="14" controlid="7" masterid="0" hint1="0" hint2="0" width="799"
height="3200" noresize="0" nomove="0" nodefaultattachpoints="1" autodrag="0"
usedefaultiddshape="0" selectable="1" showselectionhandles="0" allownudging="1"
isannotation="0" dontautolayout="0" groupcollapsed="0" tabstop="1" visible="1"
snaptogrid="0">

```

```

  <control>

```

```

    <ddsxmlobj>

```

```

      <polyline endtypedst="6" endtypesrc="3" usercolor="0" linestyle="0"
linerender="1" customendtypedstid="0" customendtypesrcid="0" adornsvisible="1"
/>

```

```

    </ddsxmlobj>

```

```

  </control>

```

```

</layoutobject>

```

```

  <ddsxmlobj>

```

```

    <property name="LogicalObject"
value="dataSet.Relations[FK_FactEdu_DimTime]" vartype="8" />

```

```

      <property name="Virtual" value="0" vartype="11" />

```

```

      <property name="VisibleAP" value="0" vartype="3" />

```

```

    </ddsxmlobj>

```

```

</layoutobject>

```

```

  <connector lineroutestyle="MSDDS.Rectilinear" sourceid="2" destid="1"
sourceattachpoint="4" destattachpoint="7" segmenteditmode="0"
bendpointeditmode="0" bendpointvisibility="0" relatedid="0" virtual="0">

```

```

    <point x="2109" y="12368" />

```

```

    <point x="2109" y="11168" />

```

```

    <point x="2110" y="11168" />

```

```

    <point x="2110" y="9968" />

```

```

  </connector>

```

</ddscontrol>

</dds>

</Value>

</Annotation>

<Annotation>

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:ShowFriendlyNames</Name>

<Value>>true</Value>

</Annotation>

<Annotation>

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:ShowRelationshipNames</Name>

<Value>>false</Value>

</Annotation>

<Annotation>

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:UseDiagramDefaultLayout</Name>

<Value>>true</Value>

</Annotation>

<Annotation>

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramViewPortLeft</Name>

<Value>-2581</Value>

</Annotation>

<Annotation>

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramViewPortTop</Name>

<Value>4282</Value>

</Annotation>

<Annotation>

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramBoundingLeft</Name>

<Value>20</Value>

</Annotation>

<Annotation>

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramBoundingTop</Name>

<Value>0</Value>

</Annotation>

<Annotation>

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramZoom</Name>

<Value>84</Value>

</Annotation>

</Annotations>

<LastProcessed>0001-01-01T00:00:00Z</LastProcessed>

<State>Unprocessed</State>

<Dimensions>

<Dimension dwd:design-time-name="8b810038-5a61-40df-8a65-069f70bd651b">

<ID>Dim Products</ID>

```

<Name>Dim Products</Name>
<DimensionID>Dim Products</DimensionID>
<Attributes>
  <Attribute dwd:design-time-name="71bd2850-b461-41cb-a481-f3ab51ffbf7">
    <AttributeID>Product Key</AttributeID>
  </Attribute>
  <Attribute dwd:design-time-name="822798c7-1374-43dd-9eb1-a4fe810a408c">
    <AttributeID>Product Type</AttributeID>
  </Attribute>
  <Attribute dwd:design-time-name="b0a169e0-210c-48b3-bf87-1f217995b7c9">
    <AttributeID>Product Brend</AttributeID>
  </Attribute>
  <Attribute dwd:design-time-name="d2f3a6d6-9577-4d5e-95b5-896ecee916a3">
    <AttributeID>Product Name</AttributeID>
  </Attribute>
</Attributes>
</Dimension>
<Dimension dwd:design-time-name="aa0c9cc2-f0cf-4468-a02b-edafaa0d33b0">
  <ID>Dim University</ID>
  <Name>Dim University</Name>
  <DimensionID>Dim University</DimensionID>
  <Attributes>
    <Attribute dwd:design-time-name="fd3aaf75-8b83-4540-a462-176622fbe77c">
      <AttributeID>University Key</AttributeID>
    </Attribute>
    <Attribute dwd:design-time-name="c3273971-2c93-4a42-b21f-2c34112cc0b8">
      <AttributeID>Name</AttributeID>
    </Attribute>
    <Attribute dwd:design-time-name="58378a8a-0df0-4030-b0f9-05d8f40c67a2">

```

```

    <AttributeID>Region</AttributeID>
  </Attribute>
</Attributes>
<Hierarchies>
  <Hierarchy          dwd:design-time-name="54ebe3a3-6957-42be-aac5-
a21ed0ebb07f">
    <HierarchyID>Hierarchy</HierarchyID>
  </Hierarchy>
</Hierarchies>
</Dimension>
<Dimension dwd:design-time-name="0e1532b7-eb8f-4adc-8fe5-ace5e0a752d9">
  <ID>Dim Time</ID>
  <Name>Dim Time</Name>
  <DimensionID>Dim Time</DimensionID>
  <Attributes>
    <Attribute          dwd:design-time-name="73b8e7d9-5f40-418c-870d-
9c49bc23db1b">
      <AttributeID>Time Key</AttributeID>
    </Attribute>
    <Attribute          dwd:design-time-name="bdce1e21-9dd6-450c-ba04-
490d4092d78a">
      <AttributeID>Calendar Year</AttributeID>
    </Attribute>
    <Attribute dwd:design-time-name="ac2aaa36-f4a6-45ef-8e33-c5da14cfb9f3">
      <AttributeID>Calendar Semester</AttributeID>
    </Attribute>
    <Attribute dwd:design-time-name="4af75cf4-227d-457f-818b-0fb2cbd8ffb1">
      <AttributeID>Calendar Quarter</AttributeID>
    </Attribute>
  </Attributes>
</Dimension>

```

```

<Attribute dwd:design-time-name="a68be3ba-df27-4735-a2b7-c04c2fd591bb">
  <AttributeID>Kvatral Nomi</AttributeID>
</Attribute>
</Attributes>
</Dimension>
</Dimensions>
<MeasureGroups>
  <MeasureGroup dwd:design-time-name="95884d89-ccc3-4998-9488-
bba333ec0f15">
    <ID>Fact Edu</ID>
    <Name>Fact Edu</Name>
    <CreatedTimestamp>0001-01-01T00:00:00Z</CreatedTimestamp>
    <LastSchemaUpdate>0001-01-01T00:00:00Z</LastSchemaUpdate>
    <LastProcessed>0001-01-01T00:00:00Z</LastProcessed>
    <State>Unprocessed</State>
    <Measures>
      <Measure dwd:design-time-name="f853c3dc-5906-4e60-8493-887c10629e58">
        <ID>Products Count</ID>
        <Name>Element miqdori</Name>
        <Source dwd:design-time-name="77860164-6495-4a0b-a544-0041a1a2f52b">
          <DataType>Integer</DataType>
          <Source xsi:type="ColumnBinding" dwd:design-time-name="484de6c7-
dbac-4bb6-979c-306dd976e1f7">
            <TableID>dbo_FactEdu</TableID>
            <ColumnID>ProductsCount</ColumnID>
          </Source>
        </Source>
        <FormatString>#,#</FormatString>
      </Measure>
    </Measures>
  </MeasureGroup>
</MeasureGroups>

```

```

<Measure dwd:design-time-name="c6d6126c-a914-49ee-a54f-b740b5b521e1">
  <ID>Products Count In Edu</ID>
  <Name>Faqat uquv jarayonida foydalaniladigan elementlar soni</Name>
  <Source dwd:design-time-name="d60babab-58ca-49e3-8c54-1664b499ee35">
    <DataType>Integer</DataType>
    <Source xsi:type="ColumnBinding" dwd:design-time-name="407746e7-
86b4-4cd7-8b98-5558a211d437">
      <TableID>dbo_FactEdu</TableID>
      <ColumnID>ProductsCountInEdu</ColumnID>
    </Source>
  </Source>
</Measure>
<Measure dwd:design-time-name="b5fd6c7d-e193-4fff-b8b5-f33de8d10fff">
  <ID>Products Count Work Pro</ID>
  <Name>Yaroqli elementlar soni</Name>
  <Source dwd:design-time-name="19b7571c-3543-46af-b25c-b2c9a6c06916">
    <DataType>Integer</DataType>
    <Source xsi:type="ColumnBinding" dwd:design-time-name="3d236088-
39e5-4cdf-9ea7-a37af17fabb0">
      <TableID>dbo_FactEdu</TableID>
      <ColumnID>ProductsCountWorkPro</ColumnID>
    </Source>
  </Source>
</Measure>
<Measure dwd:design-time-name="b47383d2-56f5-4331-b3f7-6819c2966801">
  <ID>Students Count</ID>
  <Name>Talabalar soni</Name>
  <Source dwd:design-time-name="416c3641-789b-43ea-a928-97bb822babc0">
    <DataType>Integer</DataType>

```

```

    <Source xsi:type="ColumnBinding" dwd:design-time-name="09e3a583-
ad56-4ff0-b20d-1b9d28f5fb07">
      <TableID>dbo_FactEdu</TableID>
      <ColumnID>StudentsCount</ColumnID>
    </Source>
  </Source>
</Measure>
</Measures>
<StorageMode>Molap</StorageMode>
<ProcessingMode>Regular</ProcessingMode>
<Dimensions>
  <Dimension xsi:type="RegularMeasureGroupDimension" dwd:design-time-
name="b1da72de-787f-4cb8-b243-feebe25c9b1c">
    <CubeDimensionID>Dim University</CubeDimensionID>
    <Attributes>
      <Attribute dwd:design-time-name="817a26ff-ecfd-49ee-80c3-
f0c24819b1d5">
        <AttributeID>University Key</AttributeID>
        <KeyColumns>
          <KeyColumn dwd:design-time-name="17d10c47-aea9-4929-b57f-
456761e07977">
            <DataType>Integer</DataType>
            <Source xsi:type="ColumnBinding" dwd:design-time-name="ee4aa3f4-
bd2e-465f-8710-2737faa69240">
              <TableID>dbo_FactEdu</TableID>
              <ColumnID>UniversityID</ColumnID>
            </Source>
          </KeyColumn>
        </KeyColumns>
      </Attribute>
    </Attributes>
  </Dimension>
</Dimensions>

```

```

    <Type>Granularity</Type>
  </Attribute>
</Attributes>
</Dimension>
<Dimension xsi:type="RegularMeasureGroupDimension" dwd:design-time-
name="8ac79551-5301-4c34-8801-c6a98af359bf">
  <CubeDimensionID>Dim Time</CubeDimensionID>
  <Attributes>
    <Attribute dwd:design-time-name="1ca42325-ab94-471f-a213-
62af8dfc132d">
      <AttributeID>Time Key</AttributeID>
      <KeyColumns>
        <KeyColumn dwd:design-time-name="335f0535-7bf9-41b2-a767-
7feaa981b71a">
          <DataType>Integer</DataType>
          <Source xsi:type="ColumnBinding" dwd:design-time-name="1772e366-
a9f3-4fbf-8768-9079fb52933f">
            <TableID>dbo_FactEdu</TableID>
            <ColumnID>TimeID</ColumnID>
          </Source>
        </KeyColumn>
      </KeyColumns>
      <Type>Granularity</Type>
    </Attribute>
  </Attributes>
</Dimension>
<Dimension xsi:type="RegularMeasureGroupDimension" dwd:design-time-
name="1971dc95-ea78-4280-b2c6-6f56463fa71b">
  <CubeDimensionID>Dim Products</CubeDimensionID>

```

```

<Attributes>
  <Attribute dwd:design-time-name="71d720b1-558e-4bb4-aa6c-
f4ab30df2eb9">
    <AttributeID>Product Key</AttributeID>
    <KeyColumns>
      <KeyColumn dwd:design-time-name="11267c78-9d98-4262-bc7b-
65b3457eb98f">
        <DataType>Integer</DataType>
        <Source xsi:type="ColumnBinding" dwd:design-time-name="926c55ac-
4ddd-4c63-a6b6-cf04fad1597d">
          <TableID>dbo_FactEdu</TableID>
          <ColumnID>ProductID</ColumnID>
        </Source>
      </KeyColumn>
    </KeyColumns>
    <Type>Granularity</Type>
  </Attribute>
</Attributes>
</Dimension>
</Dimensions>
<Partitions />
</MeasureGroup>
</MeasureGroups>
<Source dwd:design-time-name="dc17e364-4520-49c5-938e-c6bb4b4d9594">
  <DataSourceViewID>Education DW</DataSourceViewID>
</Source>
<MdxScripts>
  <MdxScript dwd:design-time-name="c59515de-2e00-422e-b92e-2d269905216c">
    <ID>MdxScript</ID>

```

```

<Name>MdxScript</Name>
<CreatedTimestamp>0001-01-01T00:00:00Z</CreatedTimestamp>
<LastSchemaUpdate>0001-01-01T00:00:00Z</LastSchemaUpdate>
<Commands>
  <Command>
    <Text>/*

```

The CALCULATE command controls the aggregation of leaf cells in the cube.

If the CALCULATE command is deleted or modified, the data within the cube is affected.

You should edit this command only if you manually specify how the cube is aggregated.

```
*/
```

```
CALCULATE;
```

```
CREATE MEMBER CURRENTCUBE.[Measures].[Moddiy texnik bazaning
talabalar soniga nisbati (protsent kursatkichida)]
```

```
AS [Measures].[Element miqdori]/[Measures].[Talabalar soni],
```

```
FORMAT_STRING = "Percent",
```

```
NON_EMPTY_BEHAVIOR = { [Element miqdori], [Talabalar soni] },
```

```
VISIBLE = 1 , ASSOCIATED_MEASURE_GROUP = 'Fact Edu' ; </Text>
```

```
</Command>
```

```
</Commands>
```

```
</MdxScript>
```

```
</MdxScripts>
```

```
<Kpis>
```

```
<Kpi dwid:design-time-name="78b00242-6c2d-4394-89e4-fda61dc21a36">
```

```
<ID>KPI</ID>
```

```
<Name>Kompyuter bilan taminlanganlik darajasi</Name>
```

```
<AssociatedMeasureGroupID>Fact Edu</AssociatedMeasureGroupID>
```

```
<Value>[Measures].[Element miqdori]</Value>
```

<Goal>[Measures].[Talabalar soni]/8</Goal>

<Status>Case

When

KpiValue("Kompyuter bilan taminlanganlik darajasi")/KpiGoal("Kompyuter bilan taminlanganlik darajasi")>=.95

Then 1

When

KpiValue("Kompyuter bilan taminlanganlik darajasi")/KpiGoal("Kompyuter bilan taminlanganlik darajasi")<.95

And

KpiValue("Kompyuter bilan taminlanganlik darajasi")/KpiGoal("Kompyuter bilan taminlanganlik darajasi")>=.85

Then 0

Else-1

End

</Status>

<TrendGraphic>Standard Arrow</TrendGraphic>

<StatusGraphic>Gauge - Ascending</StatusGraphic>

</Kpi>

</Kpis>

</Cube>