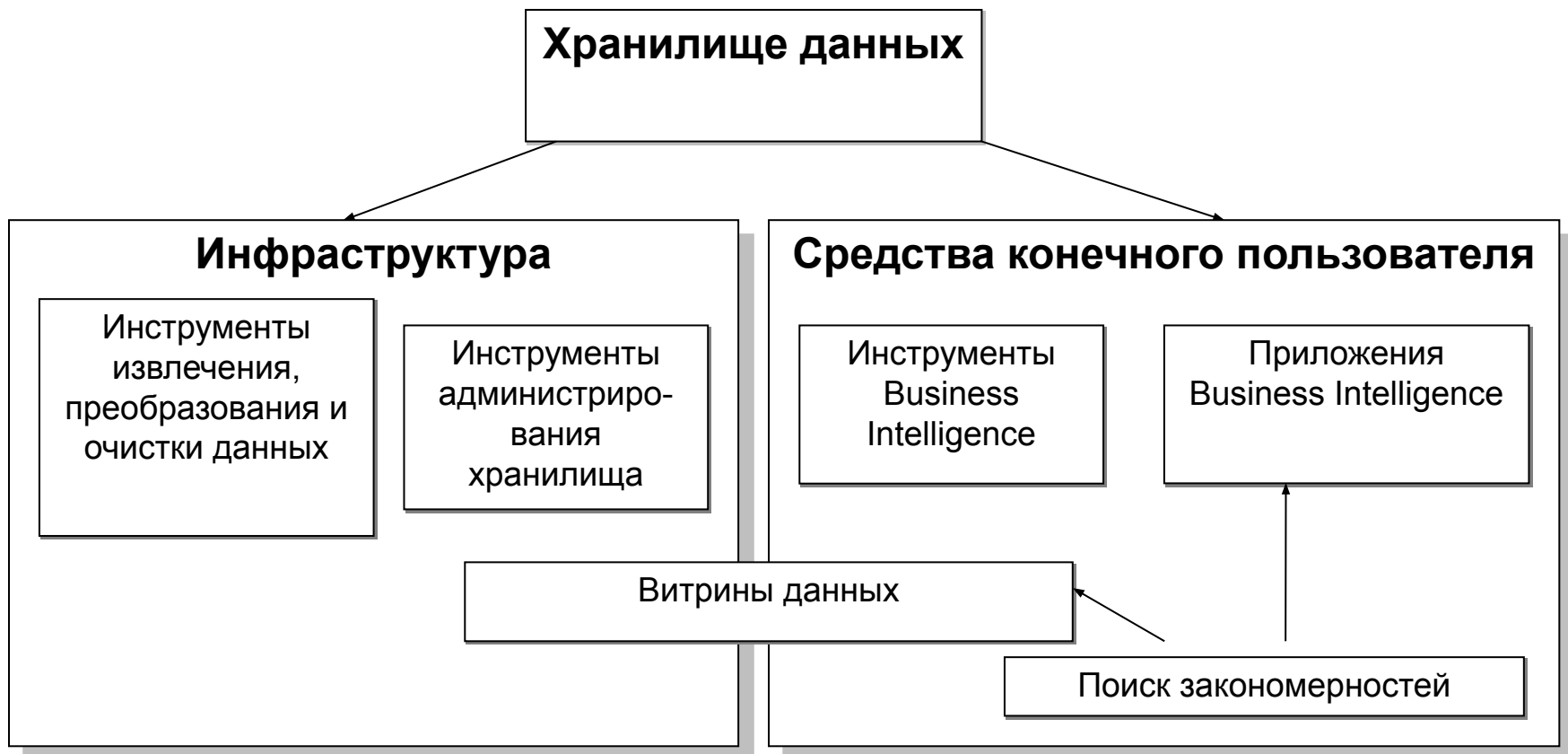


Организация хранилищ данных

Салмин Павел Сергеевич

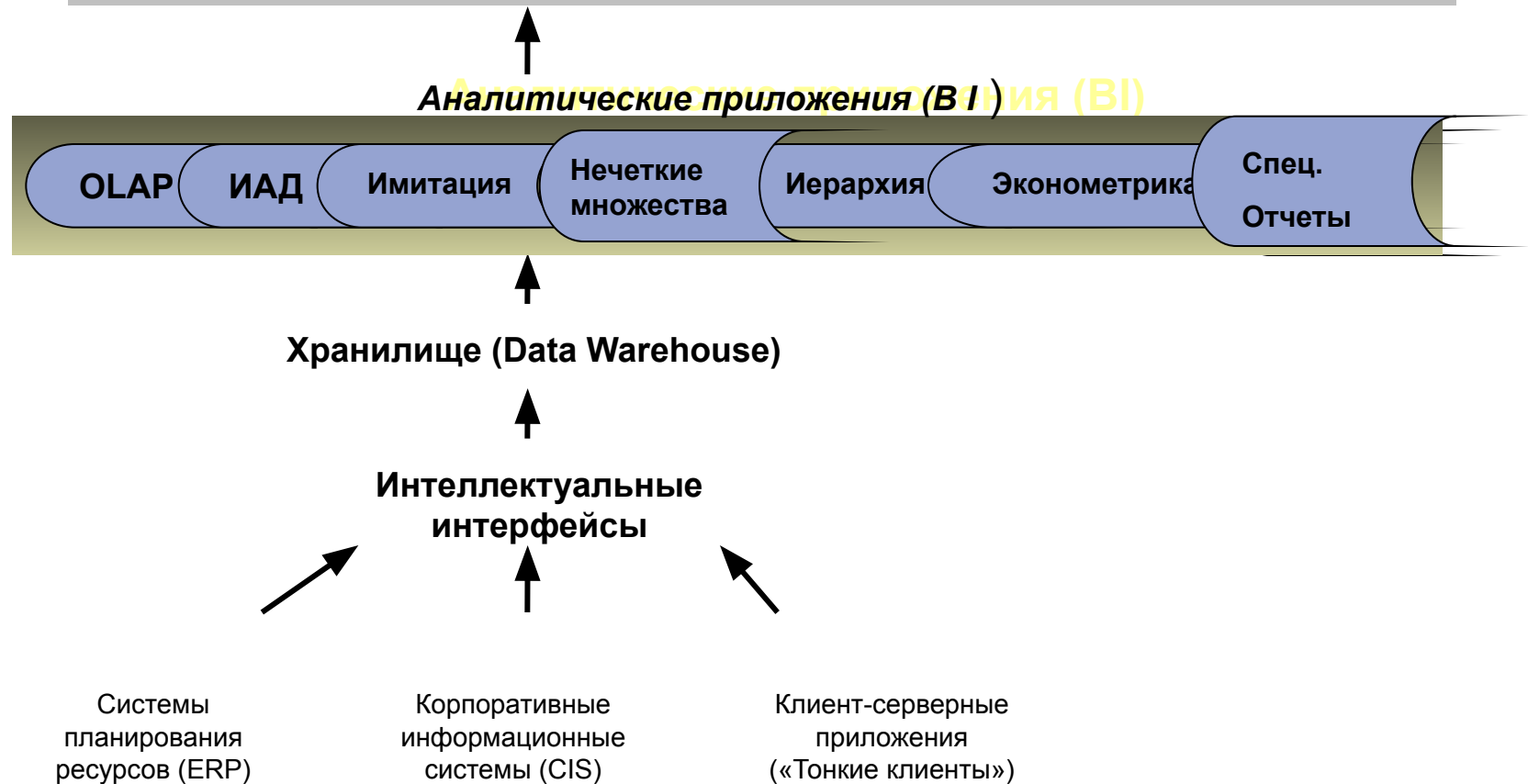
salmin@bk.ru

Хранилище - компонент BI



Место хранилища в информационной технологии поддержки принятия решений

- Системы поддержки принятия решений



Традиционные данные, хранимые в БД	Данные для принятия решений
Детализированы	Обобщены либо очищены
Точны в момент доступа	Представляют значения на указанное время
Могут корректироваться	Не корректируются, если введены в Хранилище
Требования к способам дальнейшей обработки выясняются заранее	Требования к способам дальнейшей обработки не имеют первостепенного значения
Строятся на основе обычного цикла разработки систем	Совершенно иной цикл разработки систем
Чувствительны к производительности БД и поэтому предъявляют к ним жесткие требования	Мягкие требования к производительности БД

Продолжение таблицы

Обрабатывается один элемент данных за один запрос	Обрабатывается множество элементов данных за один запрос
Управляются транзакциями	Управляются аналитическими запросами
Ориентированы на приложения	Ориентированы на анализ
Высокая степень доступности	Относительная доступность
Контролируется целостность всех данных	Контролируется целостность подмножества данных
Данные не избыточны	Данные избыточны
Статическая структура, произвольное содержание	Гибкая структура
Массивы данных редко используются в процессе обработки	Массивы данных широко используются в процессе обработки
Поддерживают ежедневные операции	Поддерживают периодический анализ

Появление хранилищ вызвано, двумя причинами:

- аналитическая работа с данными в ХД (специализированных БД) не сказывается на производительности основных БД;
- аналитики и работники управления могут полностью ориентироваться на специализированные хранилища в режиме "Что, если...".

Почему нельзя использовать традиционные БД в процессе принятия решений?

- недостоверность данных;
- низкая производительность при нестандартных запросах;
- невозможность преобразования разнородных данных, так как они часто не имеют меток времени.

Проблемы при подготовке отчетов возникают из-за того, что:

- трудно понять, где находятся данные, необходимые для анализа и принятия решения;
- большинство БД ориентировано только на стандартные запросы;
- нужно привлекать программистов для выполнения нестандартных запросов.

Опыт использования БД

- Подводя итоги, можно отметить, что, несмотря на обилие данных, возможностей их сбора и хранения, организации до сих пор испытывают серьезный недостаток в информации, необходимой для принятия решений.
- Существующие системы сбора и обработки корпоративных данных в принципе не пригодны для принятия решений. Данные разнотипны и распределены как внутри организации, так и за ее пределами. Лицам, принимающим решения (ЛПР) и аналитикам приходится принимать решения не только в условиях неполной, но и зачастую недостоверной и противоречивой информации. К тому же не всегда удастся получить требуемую информацию во время и в наглядном виде. В результате - неудачные решения.

Вывод из опыта использования БД

- Возникает необходимость в технологиях, позволяющих автоматически собирать данные из различных баз данных, систем обработки данных, согласовывать и объединять в предметно-ориентированный формат, который нужен аналитикам.

Требования к Хранилищам данных для руководящего состава и аналитиков

- ХД должно быть предметно-ориентированным, интегрированным, предназначенным для поддержки принятия решений.
- Хранилище представляет собой такую среду накопления данных, которая оптимизирована для выполнения сложных аналитических запросов управленческого персонала.
- Эти запросы могут быть достаточно индивидуальны для каждой организации, каждого подразделения и даже отдельного аналитика.

Основные свойства Хранилища данных:

- предметная ориентированность;
- интегрированность (целостность и внутренняя взаимосвязь);
- временная привязка;
- неразрушаемая совокупность данных.

Предметная ориентированность:

- Локальные базы данных содержат мегабайты информации, абсолютно не нужной для анализа (адреса, почтовые индексы, идентификаторы записей и др.). Подобная информация не заносится в хранилище, что ограничивает спектр рассматриваемых данных при принятии решения до минимума.

Интегрированность (целостность и внутренняя взаимосвязь):

- Несмотря на то что данные погружаются из различных источников, но они объединены едиными законами именования, способами измерения атрибутов и др. Это имеет большое значение для корпоративных организаций, в которых одновременно могут эксплуатироваться различные по своей архитектуре вычислительные системы, представляющие одинаковые данные по-разному. Например, могут использоваться несколько различных форматов представления дат или один и тот же показатель может называться по-разному, например, "вероятность доведения информации" и "вероятность получения информации". В процессе погружения подобные несоответствия устраняются автоматически;

Временная привязка:

- Оперативные системы охватывают небольшой интервал времени, что достигается за счет периодического архивирования данных. DW, напротив, содержит исторические данные, накопленные за большой интервал времени (пять—семь лет);

Неразрушаемая совокупность данных :

- Модификация данных не производится, поскольку может привести к нарушению их целостности.
- Поскольку требуется минимизировать время погружения, то структура хранилища может быть оптимизирована для обработки определенных запросов, что достигается за счет **денормализации** реляционной схемы, предварительного агрегирования и построения соответствующих индексов.

Особенности хранилищ данных:

- Хранилища данных содержат информацию, собранную из нескольких оперативных баз данных. Хранилища, как правило, на порядок больше оперативных баз, зачастую имея объем от сотен гигабайт до нескольких терабайт.
- Как правило, хранилище данных поддерживается независимо от оперативных баз данных организации, поскольку требования к функциональности и производительности аналитических приложений отличаются от требований к транзакционным системам.
- Хранилища данных создаются специально для приложений поддержки принятия решений и предоставляют накопленные за определенное время, сводные и консолидированные данные, которые более приемлемы для анализа, чем детальные индивидуальные записи. Рабочая нагрузка состоит из нестандартных, сложных запросов, которые обращаются к миллионам записей и выполняют огромное количество операций сканирования, соединения и агрегирования. Время ответа на запрос в данном случае важнее, чем пропускная способность.

Разновидности хранилищ - витрины данных:

- Поскольку конструирование хранилища данных — сложный процесс, который может занять несколько лет, некоторые организации вместо этого строят витрины данных (data mart), содержащие информацию для конкретных подразделений. Например, маркетинговая витрина данных может содержать только информацию о клиентах, продуктах и продажах и не включать в себя планы поставок.
- Несколько витрин данных для подразделений могут сосуществовать с основным хранилищем данных, давая частичное представление о содержании хранилища. Витрины данных строятся значительно быстрее, чем хранилище, но впоследствии могут возникнуть серьезные проблемы с интеграцией, если первоначальное планирование проводилось без учета полной бизнес-модели.

Средства извлечения, преобразования и загрузки данных:

- этап извлечения и преобразования;
- этап очистки данных;
- этап загрузки;
- этап обновления;
- управление метаданными.

Этап извлечения и преобразования

- Цель этапа извлечения данных — перенести данные из разнородных источников в базу данных, где их можно модифицировать и добавить в хранилище. Цель последующего этапа преобразования данных — устранить несоответствия в схеме и соглашениях о значениях атрибутов. Набор правил и скриптов, как правило, выполняет преобразование данных из исходной схемы в итоговую схему.
- К примеру, дистрибьютор может разделить имя каждого клиента на три части: имя, отчество (или инициалы) и фамилия.

Этап очистки данных

- Ошибки при вводе данных и различия в схемах могут привести к тому, что таблица измерений «Клиент» будет иметь несколько соответствующих кортежей для одного клиента, что приводит к неточным ответам на запросы и некорректным моделям добычи данных.
- Инструменты, которые помогают определить и исправить аномалии данных, должны иметь высокую отдачу.

Этап загрузки

- После того, как данные извлечены и преобразованы, возможно, что их еще необходимо дополнительно обработать перед тем, как добавить в хранилище.
- Как правило, утилиты фоновой загрузки поддерживают такие функции, как проверка ограничений целостности; сортировка; суммирование, агрегирование и выполнение других вычислений для создания производных таблиц, размещаемых в хранилище; создание индексов и других способов доступа.
- Помимо наполнения хранилища, утилита загрузки должна позволять системным администраторам проверять статус; отменять, приостанавливать и возобновлять загрузку; возобновлять работу после ошибки без потери целостности данных.

Этап обновления

Должны быть рассмотрены два вопроса: когда обновлять и как обновлять:

1. Обычно хранилища данных обновляются периодически в соответствии с заранее установленным расписанием, например, ежедневно или еженедельно.
2. Администраторы хранилища данных определяют правила обновления в зависимости от требований пользователей и трафика. Расписание обновлений может быть различным для разных источников данных.

Управление метаданными

Метаданные — информация любого рода, которая требуется для управления хранилищем данных, а управление метаданными — существенный компонент архитектуры хранения.

- К административным метаданным относится вся информация, которая требуется для настройки и использования хранилища данных.
- Бизнес-метаданные включают в себя бизнес-термины и определения, принадлежность данных и правила оплаты услуг хранилища.
- Оперативные метаданные — это информация, собранная во время работы хранилища данных, такая как происхождение перенесенных и преобразованных данных; статус использования данных (активные, архивированные или удаленные); данные мониторинга, такие как статистика использования, сообщения об ошибках и результаты аудита.
- Метаданные хранилища часто размещаются в репозитории, который позволяет совместно использовать метаданные различным инструментам и процессам при проектировании, установке, использовании, эксплуатации и администрировании хранилища.

Технологии хранения данных

Денормализованные, пространственные базы данных

- Одним из направлений развития РБД в интересах систем принятия решений является разработка таблиц с денормализованной формой (модификации схемы организации данных типа звезда).
- Структура такой базы данных не будет реляционной - это будет пространственная база данных с целью анализа данных, а не выполнения транзакций.

Методология Dimensional

- Нормализация данных в реляционных СУБД приводит к созданию множества связанных между собой таблиц. В результате, выполнение сложных запросов неизбежно приводит к объединению многих таблиц, что существенно увеличивает время отклика.
- Создание хранилища данных подразумевает создание денормализованной структуры данных (допускается избыточность данных и возможность возникновения аномалий при манипулировании данными), ориентированной в первую очередь на высокую производительность при выполнении аналитических запросов.
- Нормализация делает модель хранилища слишком сложной, затрудняет ее понимание и ухудшает эффективность выполнения запроса.

Как проектировать ненормализованную БД?

- Большинство Case – средств проектирования БД поддерживает методологию моделирования хранилищ благодаря использованию специальной нотации для физической модели – Dimensional.

Особенности проектирования Dimensional

- Моделирование Dimensional сходно с моделированием связей и сущностей для реляционной модели, но отличаются целями.
- Реляционная модель акцентируется на целостности и эффективности ввода данных.
- Размерная (Dimensional) модель ориентирована в первую очередь на выполнение сложных запросов к БД.

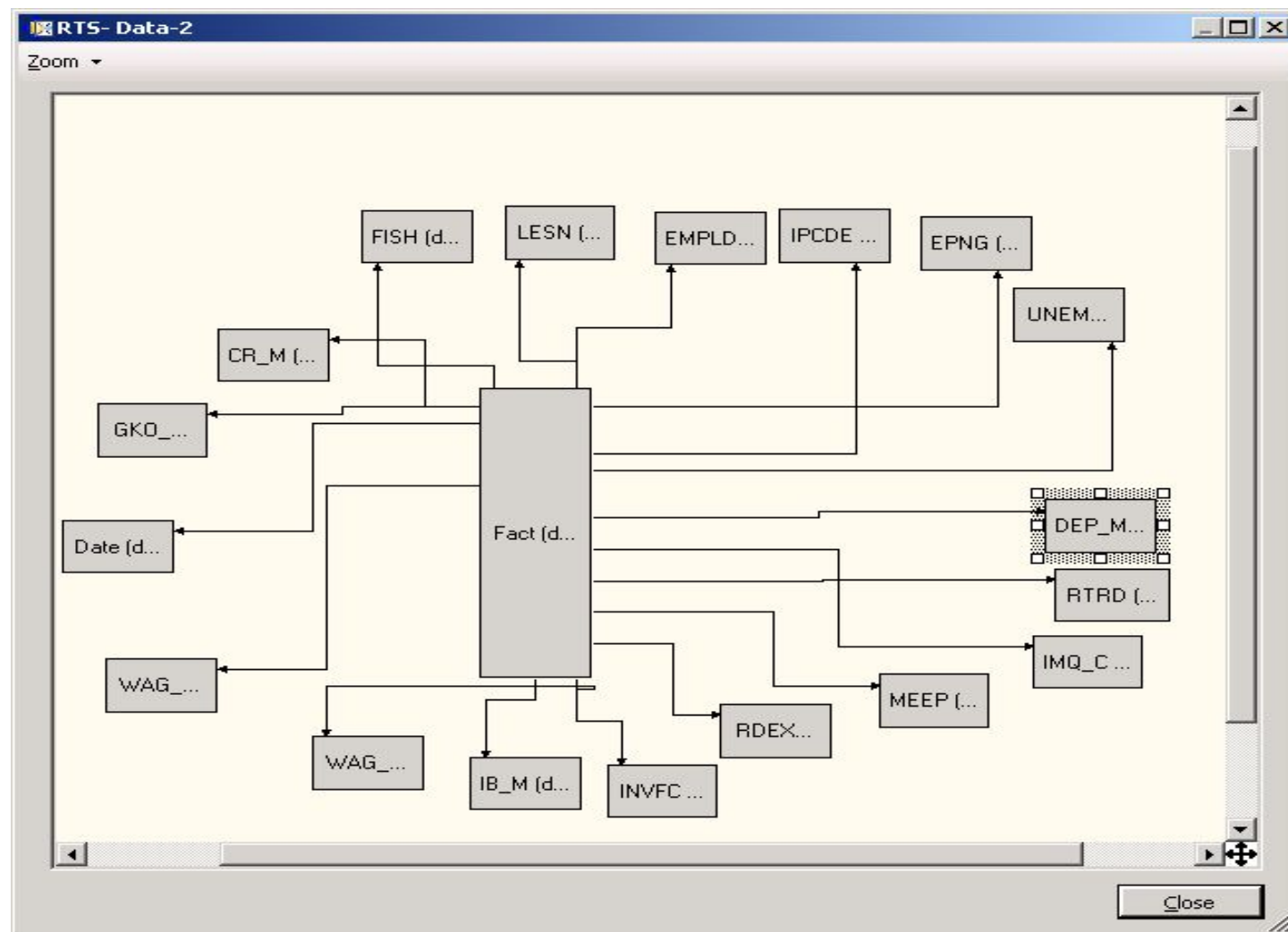
О схеме звезда

- В размерном моделировании принят стандарт модели, называемый **схемой звезда (star schema)**, которая обеспечивает высокую скорость выполнения запроса посредством денормализации и разделения данных.
- Невозможно создать универсальную денормализованную структуру данных, обеспечивающую высокую производительность при выполнении любого аналитического запроса. Поэтому схема звезда строится так, чтобы обеспечить наивысшую производительность при выполнении одного самого важного запроса, либо для группы похожих запросов.

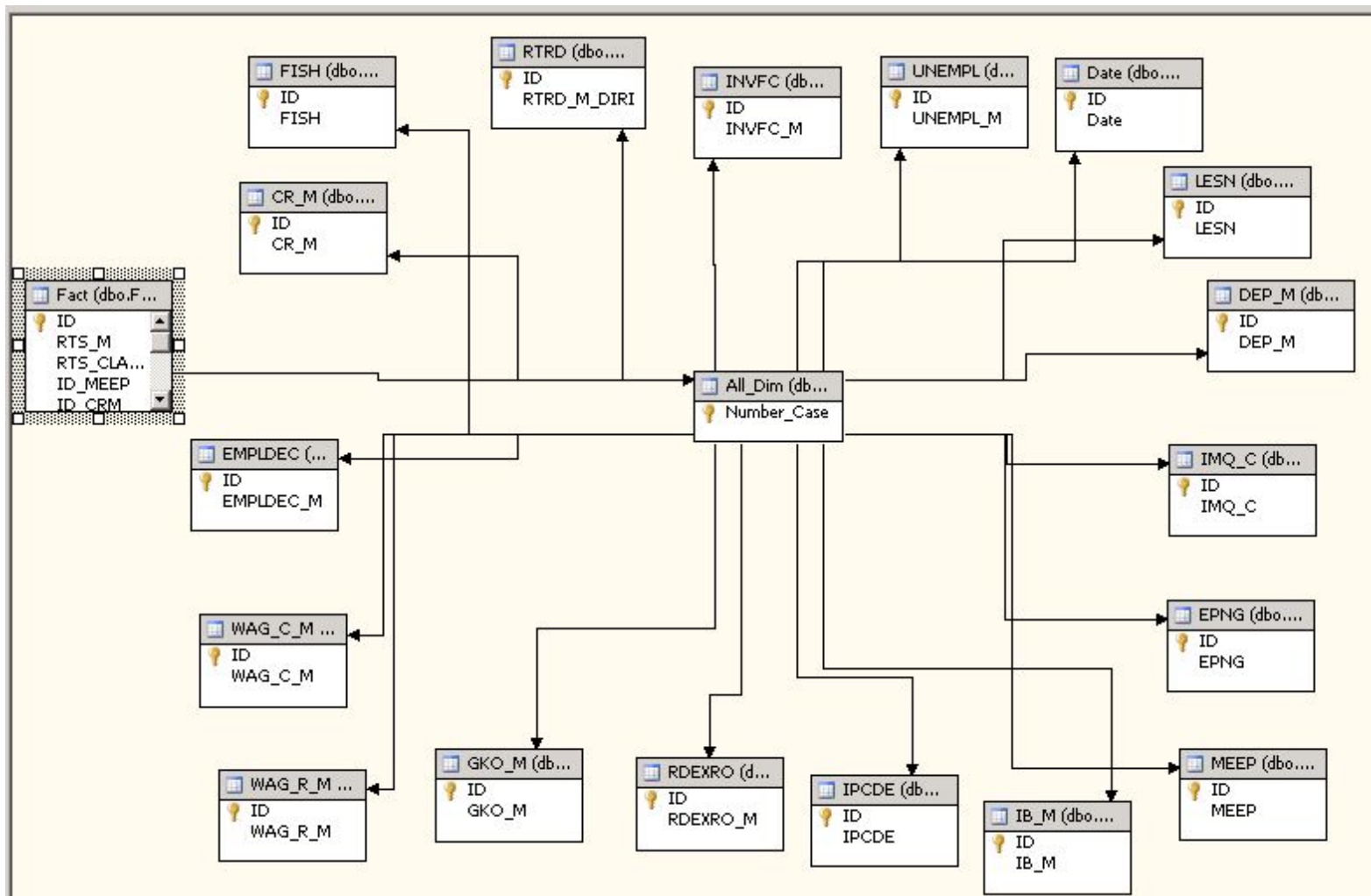
Основные составляющие структуры хранилищ данных

- Схема звезда обычно содержит одну большую таблицу, называемую таблицей факта (*fact table*), помещенную в центр, и окружающие ее меньшие таблицы, называемые таблицами размерности (*dimensional table*), соединенные с таблицей факта в виде звезды радиальными связями. В этих связях таблицы размерности являются родительскими, таблица факта - дочерней.
- Схема звезда может иметь также консольные таблицы (*outrigger table*), присоединенные к таблице размерности. Консольные таблицы являются родительскими, таблицы размерности - дочерними.

Структура ХД - звезда



Структура ХД - снежинка



Обозначения таблиц в схеме “звезда”



Таблица факта (fact table)



Таблица размерности (dimensional table)



Консольная таблица (outrigger table)

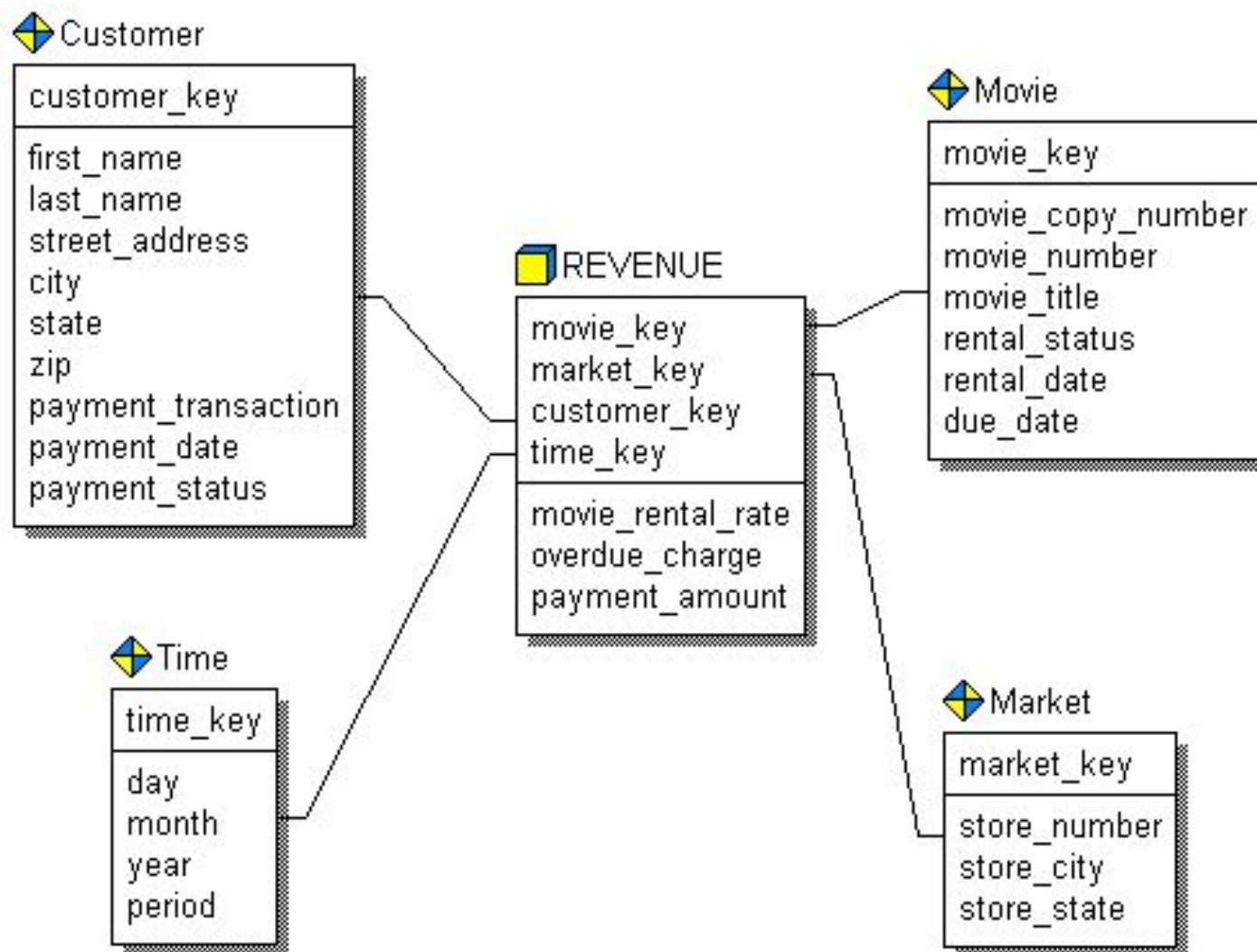
Таблица(ы) фактов

- Прежде чем создать DW со схемой типа звезда, необходимо проанализировать бизнес-правила предметной области с целью выяснения центрального вопроса, ответ на который наиболее важен. Все прочие вопросы должны быть объединены вокруг этого основного вопроса и моделирование должно начинаться с него. Данные, необходимые для ответа на этот вопрос, должны быть помещены в центральную таблицу модели - таблицу факта

О связи таблицы фактов с таблицами измерений

- Таблица факта является центральной таблицей в схеме звезда. Она может состоять из миллионов строк и содержать суммирующие или фактические данные, которые могут помочь ответить на требуемые вопросы. Она соединяет данные, которые хранились бы во многих таблицах традиционных реляционных базах данных. Таблица факта и таблицы размерности связаны идентифицирующими связями, при этом первичные ключи таблицы размерности мигрируют в таблицу факта в качестве внешних ключей. В размерной модели направления связей явно не показываются – они определяются типом таблиц. 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.

Первичный ключ (таблица факта “REVENUE”)
составлен из четырех внешних ключей: movie_key,
market_key, customer_key и time_key



Наиболее часто встречающихся типы фактов

- 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).

О детализации фактов

- 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.

Правила агрегации данных

- 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.

Таблицы измерений

- Таблицы измерений содержат неизменяемые либо редко изменяемые данные (типа справочник). 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.

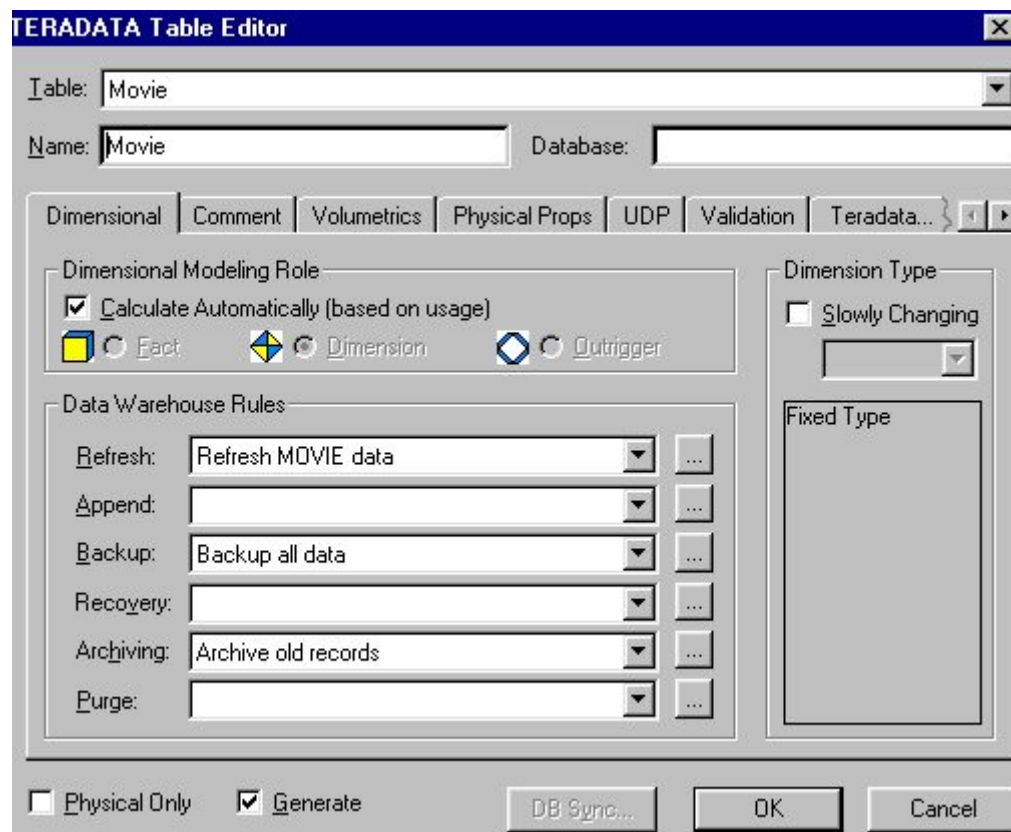
Отличие от схемы «звезда»

- Если 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).

СВЯЗИ КОНСОЛЬНЫХ ТАБЛИЦ

- Консольные таблицы могут быть связаны только таблицами размерности, причем консольная таблица в этой связи родительская, а таблица размерности - дочерняя. Связь может быть идентифицирующей или неидентифицирующей.
- Консольная таблица не может быть связана таблицей факта. Она используется для нормализации данных в таблицах размерности. Нормализация данных полезна при моделировании реляционной структуры, но она уменьшает эффективность выполнения запросов к хранилищу данных. В размерной модели главной целью является обеспечение высокой эффективности просмотра данных и выполнения сложных запросов. Схема снежинка обычно препятствует эффективности, потому что требует объединения многих таблиц для построения результирующего набора данных, что увеличивает время выполнения запроса. Поэтому при проектировании не следует злоупотреблять созданием множества консольных таблиц.

Закладка Dimensional диалога Table Editor



- В диалоге описания свойств таблицы Table Editor имеется закладка Dimensional, в которой задаются специфические свойства таблицы в размерной модели, роль таблицы в схеме (Dimensional Modeling Role)

Правила хранения данных (Data Warehouse Rules)

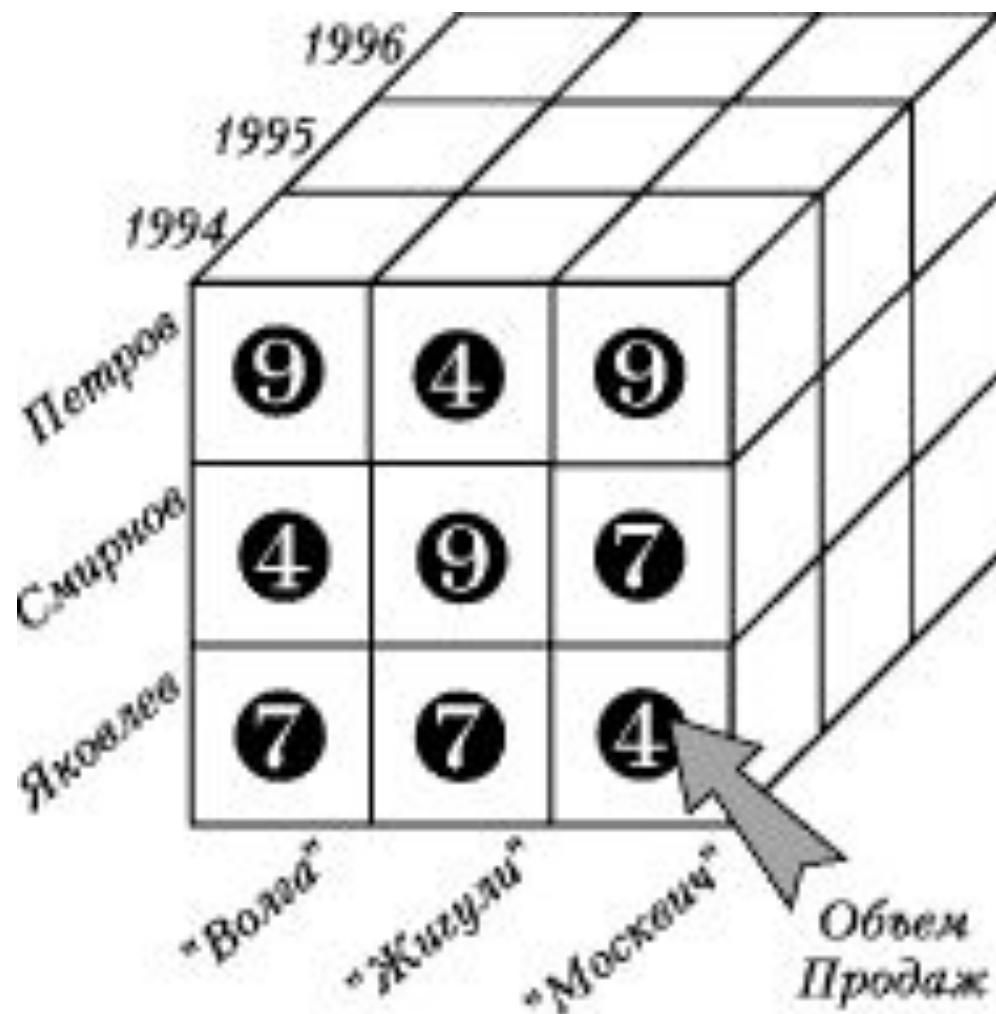
- Для каждой таблицы можно задать шесть типов правил манипулирования данными: **обновление (Refresh)**, **дополнение (Append)**, **резервное копирование (Backup)**, **восстановление (Recovery)**, **архивирование (Archiving)** и **очистка (Purge)**.
- Для задания правила следует выбрать имя правила из соответствующего списка выбора. Каждое правило должно быть предварительно описано в диалоге Data Warehouse Rule Editor (меню Edit / Data Warehouse Rule).
- Для каждого правила должно быть задано имя, тип, определение.
- Например, определение правила дополнения данных может включать частоту и время дополнения (ежедневно, в конце рабочего дня), продолжительность операции и т.д. Связать правила с определенной таблицей можно с помощью диалога Table Editor.

2. Кубы данных (многомерная модель данных)

Понятие о кубах

- Куб OLAP - это структура, в которой хранятся совокупности данных, полученные из базы данных OLAP путем всех возможных сочетаний измерений с фактами продаж в таблице фактов.
- Исходя из этого, создание окончательного отчета выполняется гораздо эффективнее, поскольку не требует выполнения никакого сложного запроса.

Вид трехмерного куба



Основными понятиями многомерной модели данных являются:

- **Показатель** - это величина (обычно числового типа), которая собственно и является предметом анализа. Один OLAP-куб может обладать одним или несколькими показателями.
- **Измерение** (dimension) - это множество объектов одного или нескольких типов, организованных в виде иерархической структуры и обеспечивающих информационный контекст числового показателя. Измерение принято визуализировать в виде ребра многомерного куба.
- Объекты, совокупность которых и образует измерение, называются **членами измерений** (members). Члены измерений визуализируют как точки или участки, откладываемые на осях гиперкуба. Например, временное измерение: Дни, Месяцы, Кварталы, Годы - наиболее часто используемые в анализе, могут содержать следующие члены: 8 мая 2002 года, май 2002 года, 2-ой квартал 2002 года и 2002 год. Как уже было сказано, объекты в измерениях могут быть различного типа, например "производители" - "марки автомобиля" или "годы" - "кварталы". Эти объекты должны быть организованы в иерархическую структуру так, чтобы объекты одного типа принадлежали только одному уровню иерархии.
- **Ячейка** (cell) - атомарная структура куба, соответствующая конкретному значению некоторого показателя. Ячейки при визуализации располагаются внутри куба и здесь же принято отображать соответствующее значение показателя.

Роль измерений в кубе

- Измерения играют роль индексов, используемых для идентификации значений показателей, находящихся в ячейках гиперкуба. Комбинация членов различных измерений играют роль координат, которые определяют значение определенного показателя. Поскольку для куба может быть определено несколько показателей, то комбинация членов всех измерения будет определять несколько ячеек со значениями каждого из показателей. Поэтому для однозначной идентификации ячейки необходимо указать комбинацию членов всех измерений и показатель.

Иерархии в измерениях необходимы для возможности агрегации и детализации значений показателей

Существуют следующие типы иерархий:

- **сбалансированные;**
- **несбалансированные;**
- **неровные.**

Сбалансированные иерархии

- Это - иерархии, в которых число уровней определено её структурой и неизменно, и каждая ветвь иерархического дерева содержит объекты каждого из уровней. Каждому производителю автомобилей может соответствовать несколько марок автомобилей, а каждой марке - несколько моделей автомобилей, поэтому можно говорить о трёхуровневой иерархии этих объектов. В этом случае на первом уровне иерархии располагаются производители, на втором - марки, а на третьем - модели.
- Как видно, для формирования сбалансированной иерархии необходимо наличие связи "один-ко-многим" между объектами менее детального уровня по отношению к объектам более детального уровня. В принципе каждый уровень сбалансированной иерархии можно представить как отдельное простое измерение, но тогда эти измерения окажутся зависимыми, в значит неизбежно повышение разреженности куба.

Несбалансированные иерархии

- Это - иерархии, в которых число уровней может быть изменено, и каждая ветвь иерархического дерева может содержать объекты, принадлежащие не всем уровням, только нескольким первым.
- Необходимо заметить, что все объекты несбалансированной иерархии принадлежат одному типу.
- Типичный пример несбалансированной иерархии - иерархия типа "начальник-подчиненный", где все объекты имеют один и тот же тип - "Сотрудник".

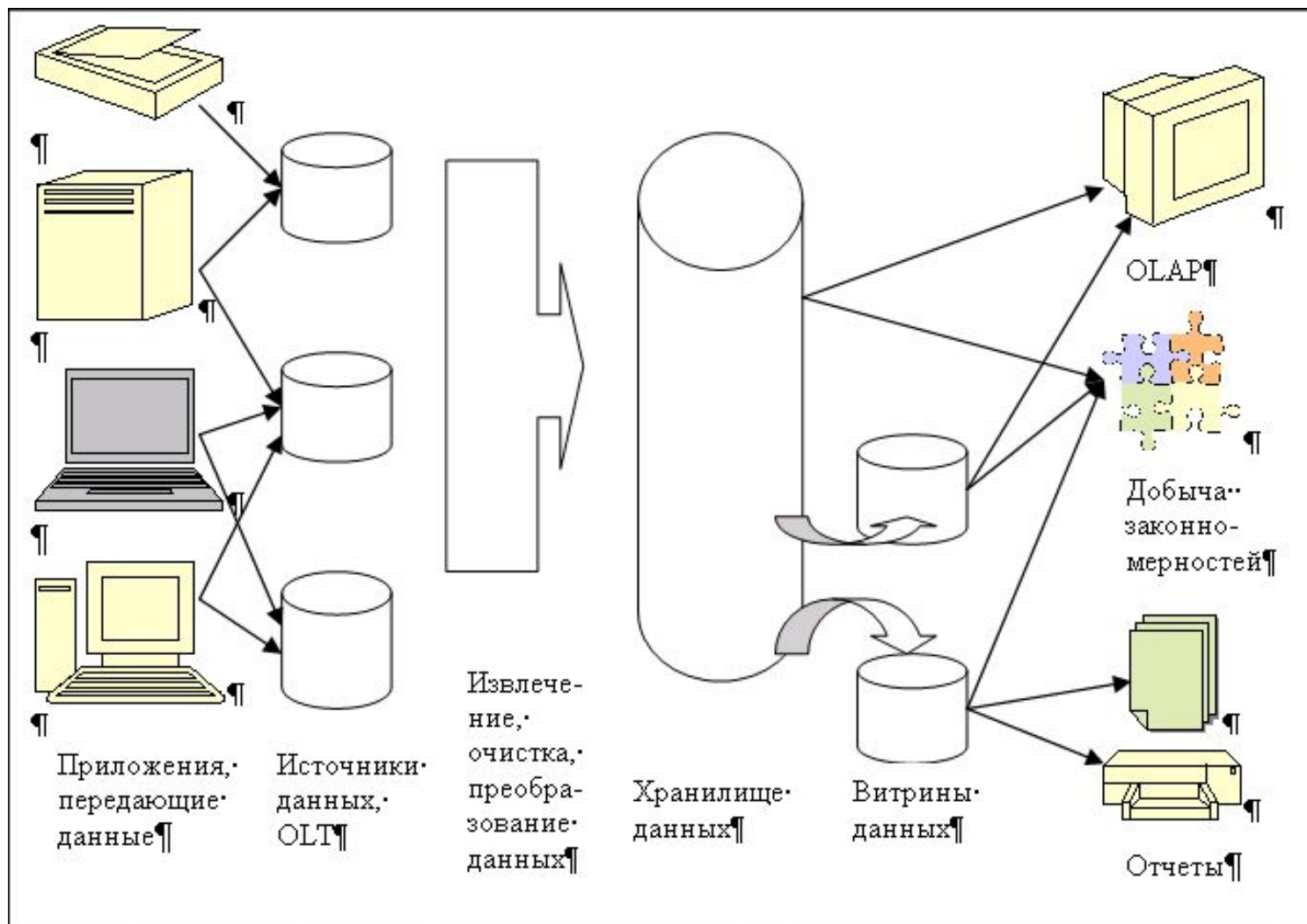
Неровные иерархии

- Это- иерархии, в которых число уровней определено её структурой и постоянно, однако в отличие от сбалансированной иерархии некоторые ветви иерархического дерева могут не содержать объекты какого-либо уровня.
- Иерархии такого вида содержат такие члены, логические "родители" которых не находятся на непосредственно вышестоящем уровне.
- Типичным примером является географическая иерархия, в которой есть уровни "Страны", "Штаты" и "Города", но при этом в наборе данных имеются страны, не имеющие штатов или регионов между уровнями "Страны" и "Города".

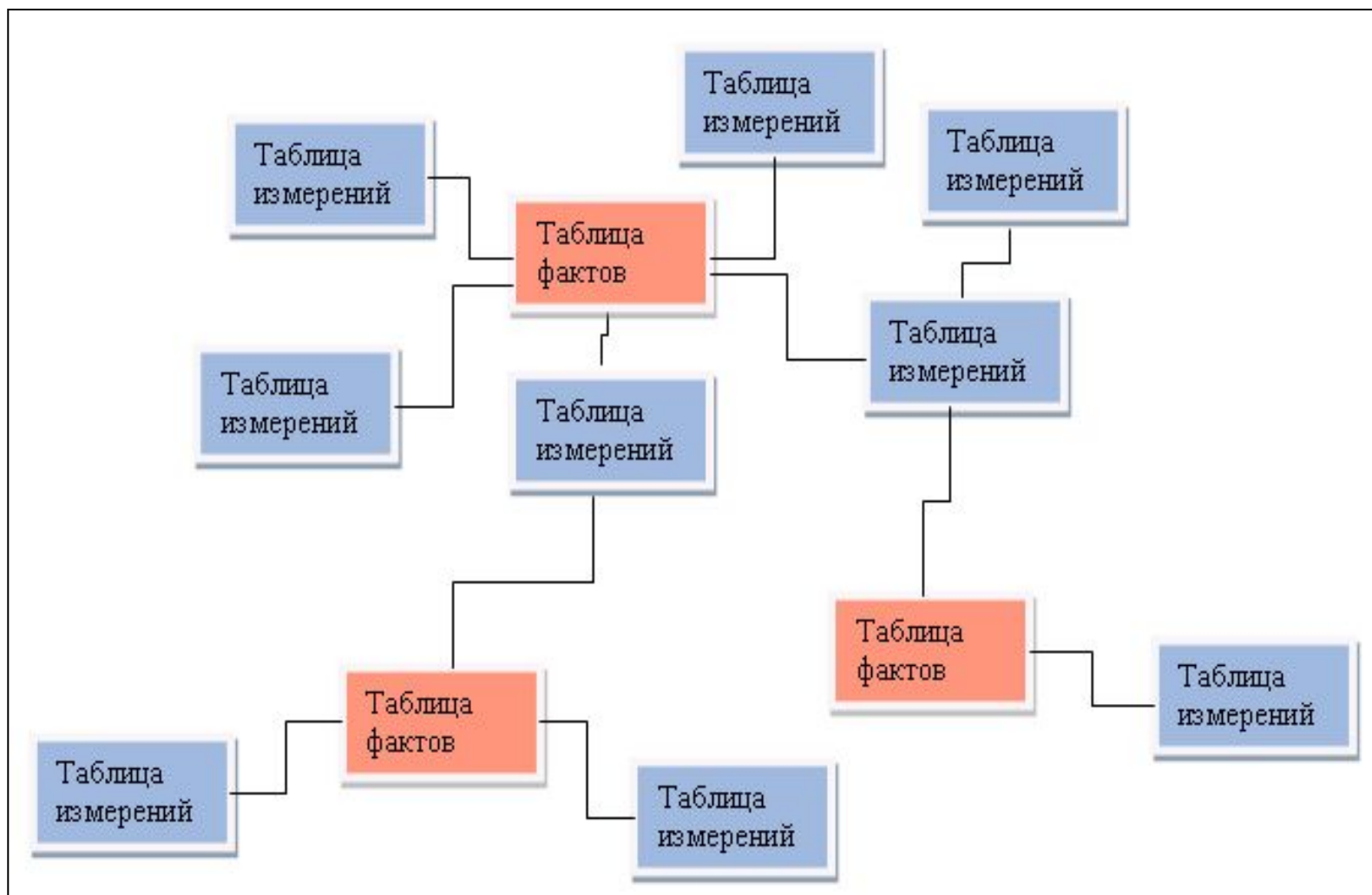
Агрегаты

- Агрегатами называют агрегированные по определенным условиям исходные значения показателей. Обычно под агрегацией понимается любая процедура формирования меньшего количества значений (агрегатов) на основании большего количества исходных значений. В дальнейшем под терминами агрегирование и агрегация будем понимать исключительно процесс суммирования данных.
- Заблаговременное формирование и сохранение агрегатов с целью уменьшения времени отклика на пользовательский запрос является основным свойством систем поддержки оперативного анализа.

DW с витринами данных



Многомерный куб с несколькими таблицами фактов



Варианты реализации хранилищ данных:

- Виртуальное хранилище данных
- Концепция CIF
- Концепция Data Warehouse Bus
- Гибридная многоуровневая архитектура хранилища данных

Виртуальное хранилище данных

- В данном случае в отличие от классического (физического) ХД данные из оперативных источников данных (ОИД) не копируются в единое хранилище.
- Они извлекаются, преобразуются и интегрируются непосредственно при выполнении аналитических запросов в оперативной памяти компьютера. Фактически такие запросы напрямую адресуются к ОИД

Основными достоинствами виртуального ХД являются:

- минимизация объема памяти, занимаемой на носителе информации;
- работа с текущими, детализированными данными.

Недостатки технологии виртуального хранилища

- Время обработки запросов к виртуальному ХД значительно превышает соответствующие показатели для физического хранилища.
- Интегрированный взгляд на виртуальное хранилище возможен только при выполнении условия постоянной доступности всех ОИД. Таким образом, временная недоступность хотя бы одного из источников может привести либо к невыполнению аналитических запросов, либо к неверным результатам.
- Различные ОИД могут поддерживать разные форматы и кодировки данных. Часто на один и тот же вопрос может быть получено несколько вариантов ответа. Это может быть связано с несинхронностью моментов обновления данных в разных ОИД, отличиями в описании одинаковых объектов и событий предметной области, ошибками при вводе, утерей фрагментов архивов и т. д.
- Главным же недостатком виртуального хранилища следует признать практическую невозможность получения данных за долгий период времени. При отсутствии физического хранилища доступны только те данные, которые на момент запроса есть в ОИД. Основное назначение OLTP-систем — оперативная обработка текущих данных, поэтому они не ориентированы на хранение данных за длительный период времени. По мере устаревания данные выгружаются в архив и удаляются из оперативной БД.

Концепция Corporate Information Factory, (сокр. CIF) Билла Инмона

- Концепция CIF объединила оперативные приложения, накопители оперативных данных (Operational Data Store, ODS, OLTP-системы), центральное хранилище данных (DW), витрины данных (Data Mart) и системы интеллектуального анализа данных (Data Mining) в единый процесс выработки и потребления информации на предприятии.
- В CIF оперативные приложения служат для управления частными процессами. ODS накапливают в себе временные срезы различных процессов, происходящих на предприятии, и согласуют их между собой. ODS часто используется как оперативный источник информации. Как правило, ODS хранят значительно более детализированную информацию, чем хранилище, но за меньший период времени — от полугода до года, так как для доступа к данным в нем не используются предварительно рассчитываемые агрегаты.

Работа Хранилища CIF состоит из следующих этапов:

- скоординированное извлечение данных из источников.
- загрузка реляционной базы данных, состоящей из таблиц в третьей нормальной форме, содержащей атомарные данные.
- получившееся нормализованное Хранилище 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.

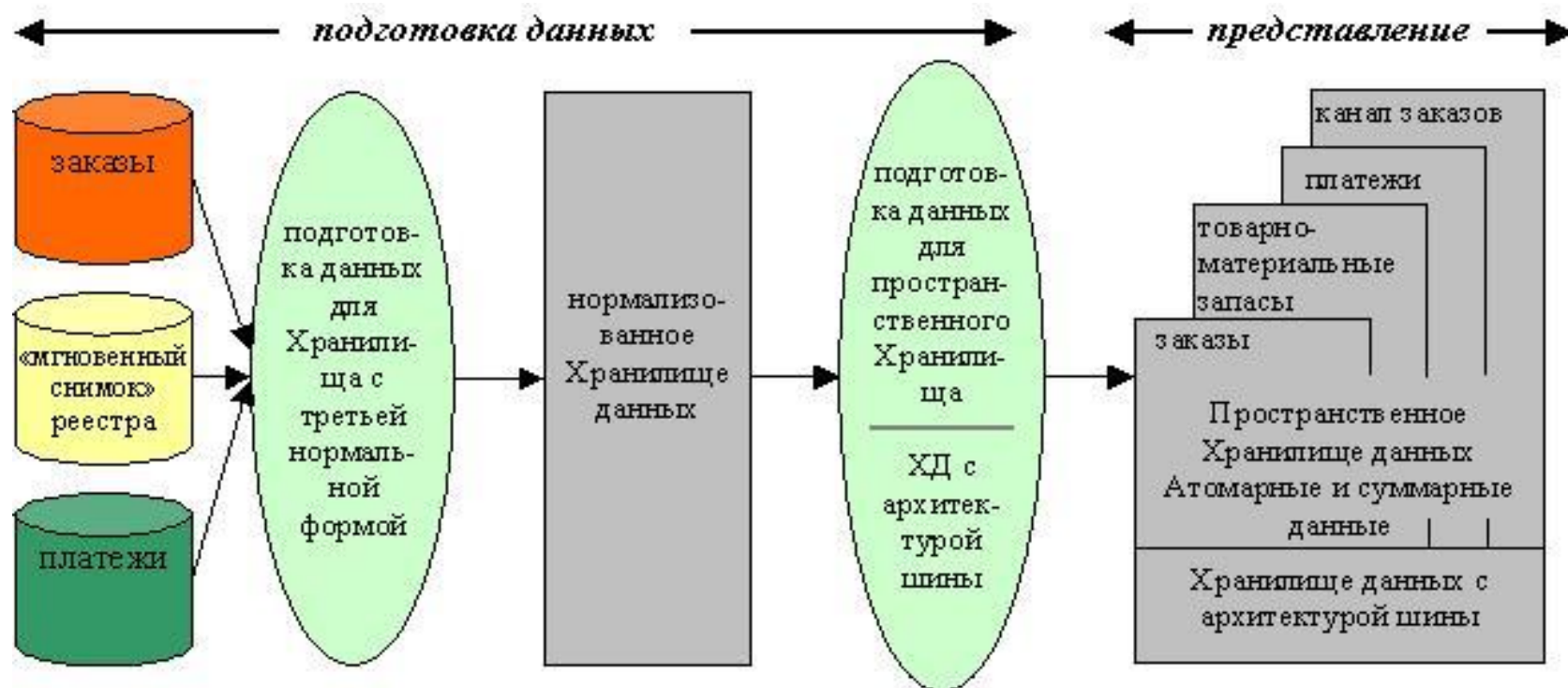
Концепция Data Warehouse Bus

- Использование пространственной модели организации данных с архитектурой "звезда" (star scheme).
- Использование двухуровневой архитектуры, которая включает стадию подготовки данных, недоступную для конечных пользователей, и Хранилище.
- 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 Хранилища данных.
- Хранилище Кимболла - скорее "виртуальный" объект. Это коллекция витрин данных, которые могут быть пространственно разобобщенными.

Гибридное хранилище данных

- В последнее время все более популярной становится идея совместить концепции хранилища и витрины данных в одной реализации и использовать хранилище данных в качестве единственного источника интегрированных данных для всех витрин данных.
- Тогда естественной становится трехуровневая архитектура системы.

Гибрид нормализованного и пространственного Хранилищ данных



Первый уровень гибридного хранилища

- На **первом уровне** реализуется корпоративное хранилище данных на основе одной из развитых современных реляционных СУБД. Это хранилище интегрированных в основном детализированных данных. Реляционные СУБД обеспечивают эффективное хранение и управление данными очень большого объема, но не слишком хорошо соответствуют потребностям OLAP-систем, в частности, в связи с требованием многомерного представления данных.

Второй уровень гибридного хранилища

- На **втором уровне** поддерживаются витрины данных на основе многомерной системы управления базами данных (примером такой системы является Oracle Express Server). Такие СУБД почти идеально подходят для целей разработки OLAP-систем, но пока не позволяют хранить сверхбольшие объемы данных (предельный размер многомерной базы данных составляет 10-40 Гбайт). В данном случае это и не требуется, поскольку речь идет о витринах данных.
- Витрина данных не обязательно должна быть полностью сформирована. Она может содержать ссылки на хранилище данных и добирать оттуда информацию по мере поступления запросов. Конечно, это несколько увеличивает время отклика, но зато снимает проблему ограниченного объема многомерной базы данных.

Третий уровень гибридного хранилища

- На **третьем уровне** находятся клиентские рабочие места конечных пользователей, на которых устанавливаются средства оперативного анализа данных.

Форматы хранения данных в OLAP кубах

Данные форматы различаются методами хранения кубов данных

- многомерный OLAP-формат (Multi-dimensional OLAP - MOLAP);
- реляционный OLAP-формат (Relational OLAP - ROLAP);
- гибридный OLAP-формат (Hybrid OLAP - HOLAP).

MOLAP

- MOLAP является многомерным форматом хранения данных, который отличается высоким быстродействием. Помимо поддержки OLAP самих кубов данных при выборе данного формата данные будут храниться в многомерных структурах на OLAP-сервере (OLAP-структуры).
- MOLAP обеспечивает наилучшее быстродействие выполнения запросов, поскольку этот формат специально оптимизирован для многомерных запросов к данным.

Преимущества и недостатки MOLAP

- Поскольку MOLAP требует копирования и преобразования всех данных в надлежащий формат для многомерной структуры хранилища данных, MOLAP можно применять для небольших или средних объемов данных.
- Основное преимущество MOLAP заключается в превосходных свойствах индексации; ее недостаток — низкий коэффициент использования дискового пространства, особенно в случае разреженных данных.

Область применения MOLAP

- объем исходных данных для анализа не слишком велик (не более нескольких гигабайт), т.е. уровень агрегации данных достаточно высок;
- набор информационных измерений стабилен (поскольку любое изменение в их структуре почти всегда требует полной перестройки гиперкуба);
- время ответа системы на нерегламентированные запросы является наиболее критичным параметром;
- широкое использование сложных встроенных функций требуется для выполнения кроссмерных вычислений над ячейками гиперкуба, в том числе возможности написания пользовательских функций.

ROLAP

- Реляционные хранилища OLAP содержат данные, передаваемые в кубы данных, вместе с агрегациями данных куба, причем данные хранятся в реляционных таблицах, размещенных в реляционном ХД.

Преимущества ROLAP:

- в большинстве случаев корпоративные хранилища данных реализуются средствами реляционных СУБД, и инструменты ROLAP позволяют производить анализ непосредственно над ними. При этом размер хранилища не является таким критичным параметром, как в MOLAP;
- при переменной размерности задачи, когда изменения в структуру измерений приходится вносить достаточно часто, ROLAP-системы с динамическим представлением размерности являются оптимальным решением, так как в них такие модификации не требуют физической реорганизации БД;
- реляционные СУБД обеспечивают значительно более высокий уровень защиты данных и хорошие возможности разграничения прав доступа.

Недостатки ROLAP

- Главный недостаток ROLAP по сравнению с MOLAP — меньшая производительность.
- Для обеспечения производительности, сравнимой с многомерными базами данных, необходимо использовать звездообразные схемы. В этом случае производительность реляционных систем может быть приближена к производительности систем на основе MOLAP.

HOLAP

- Гибридная архитектура, которая объединяет технологии ROLAP и MOLAP. В отличие от MOLAP, которая работает лучше, когда данные более плотные, серверы ROLAP лучше в тех случаях, когда данные довольно разрежены.
- Серверы HOLAP применяют подход ROLAP для разреженных областей многомерного пространства и подход MOLAP — для плотных областей.
- Серверы HOLAP разделяют запрос на несколько подзапросов, направляют их к соответствующим фрагментам данных, комбинируют результаты, а затем предоставляют результат пользователю.
- При использовании данного формата OLAP-данные, передаваемые в куб данных, хранятся в реляционных базах данных подобно ROLAP. А агрегации данных (данные куба) записываются и представляются в многомерном формате.

Преимущества и недостатки HOLAP

- Преимуществом данной системы является обеспечение возможности связи с огромными наборами данных в реляционных таблицах и прирост производительности за счет использования многомерных хранилищ.
- Недостаток состоит том, что количество проводимых преобразований между ROLAP и MOLAP системами может существенно влиять на общую эффективность.

