Учебный курс

Хранилища данных

Лекция 1

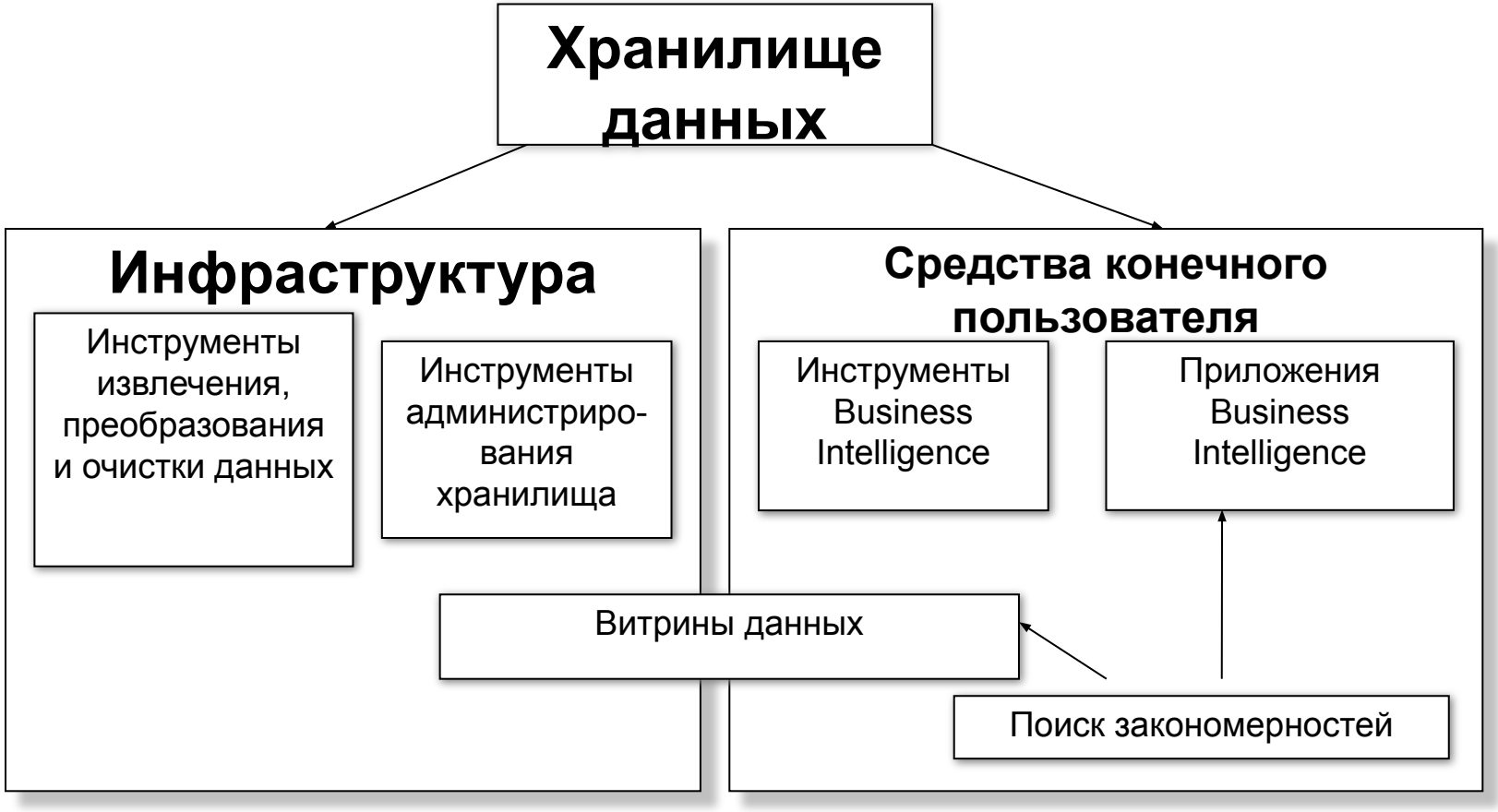
Понятия о хранилищах

Лекции читает

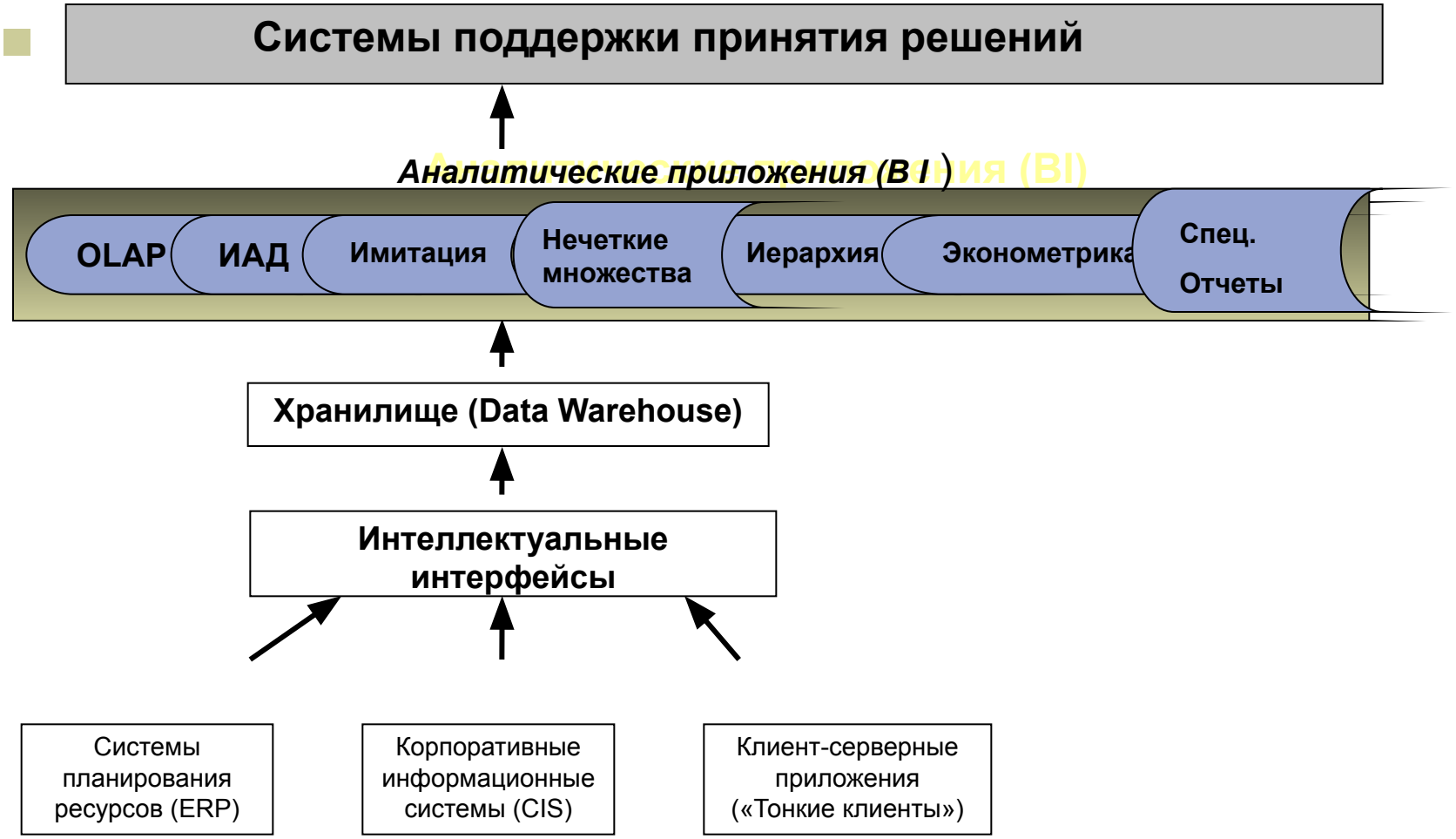
Кандидат технических наук, доцент

Перминов Геннадий Иванович

Хранилище – компонент BI



Место хранилища в информационной технологии поддержки принятия решений



Расхождения в требованиях к хранению данных в БД и ХД

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 использования в ППР. 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 удается 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 составляющие 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 не 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.

Компонента— 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

1.Денормализованные, пространственные базы 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

- 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.

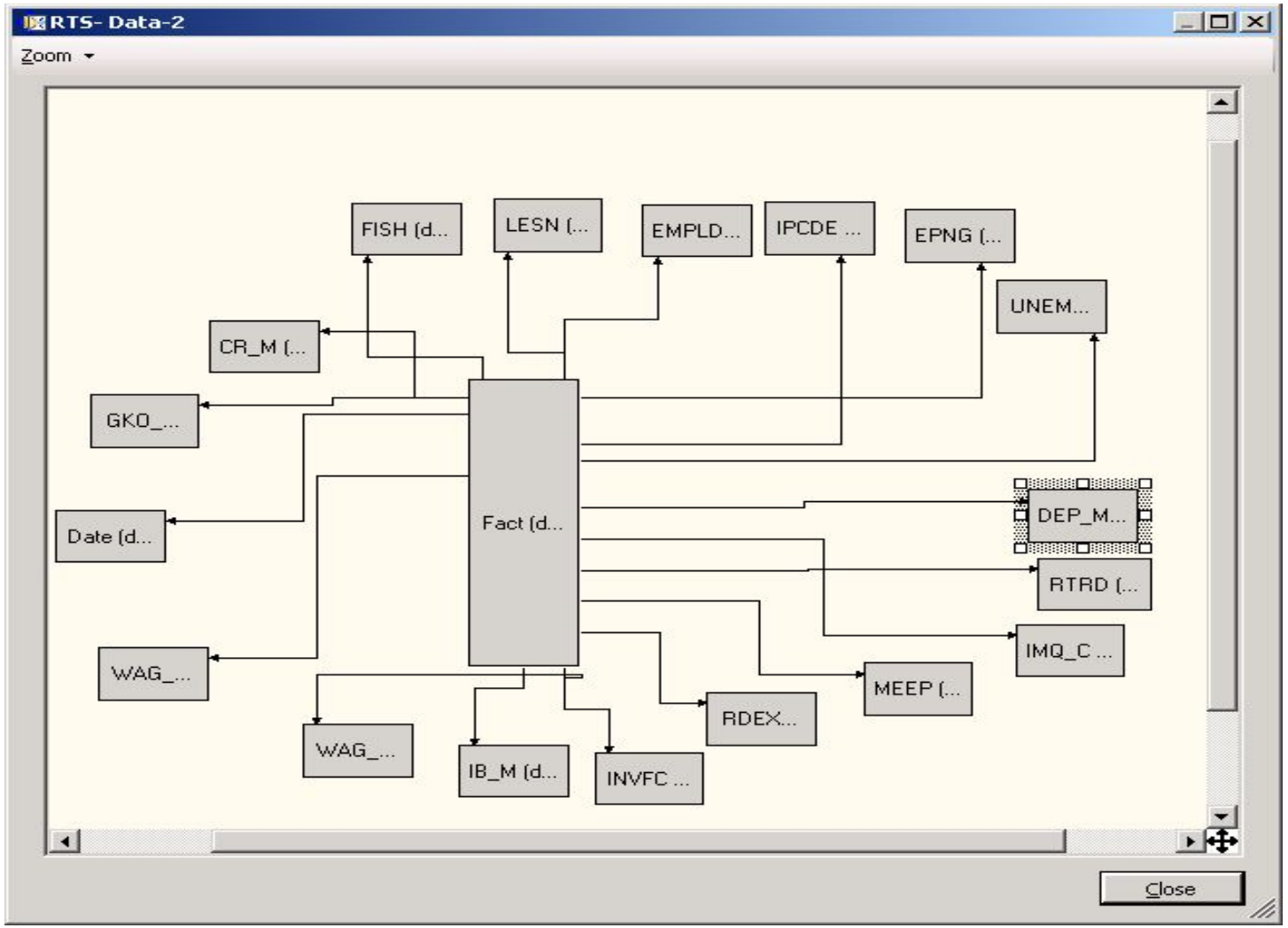
О схеме звезда

- 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.

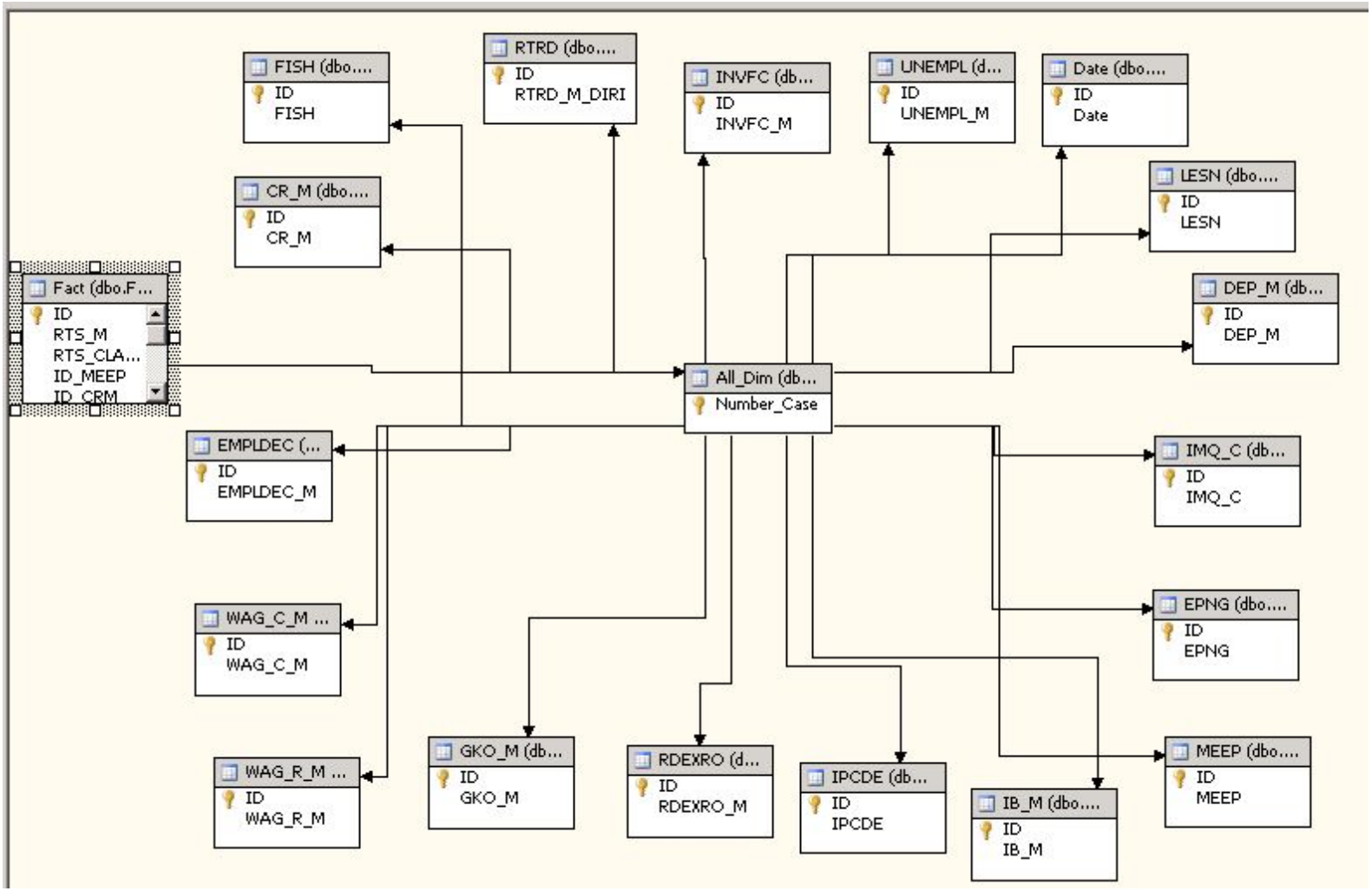
Основные составляющие структуры хранилищ данных

- 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)

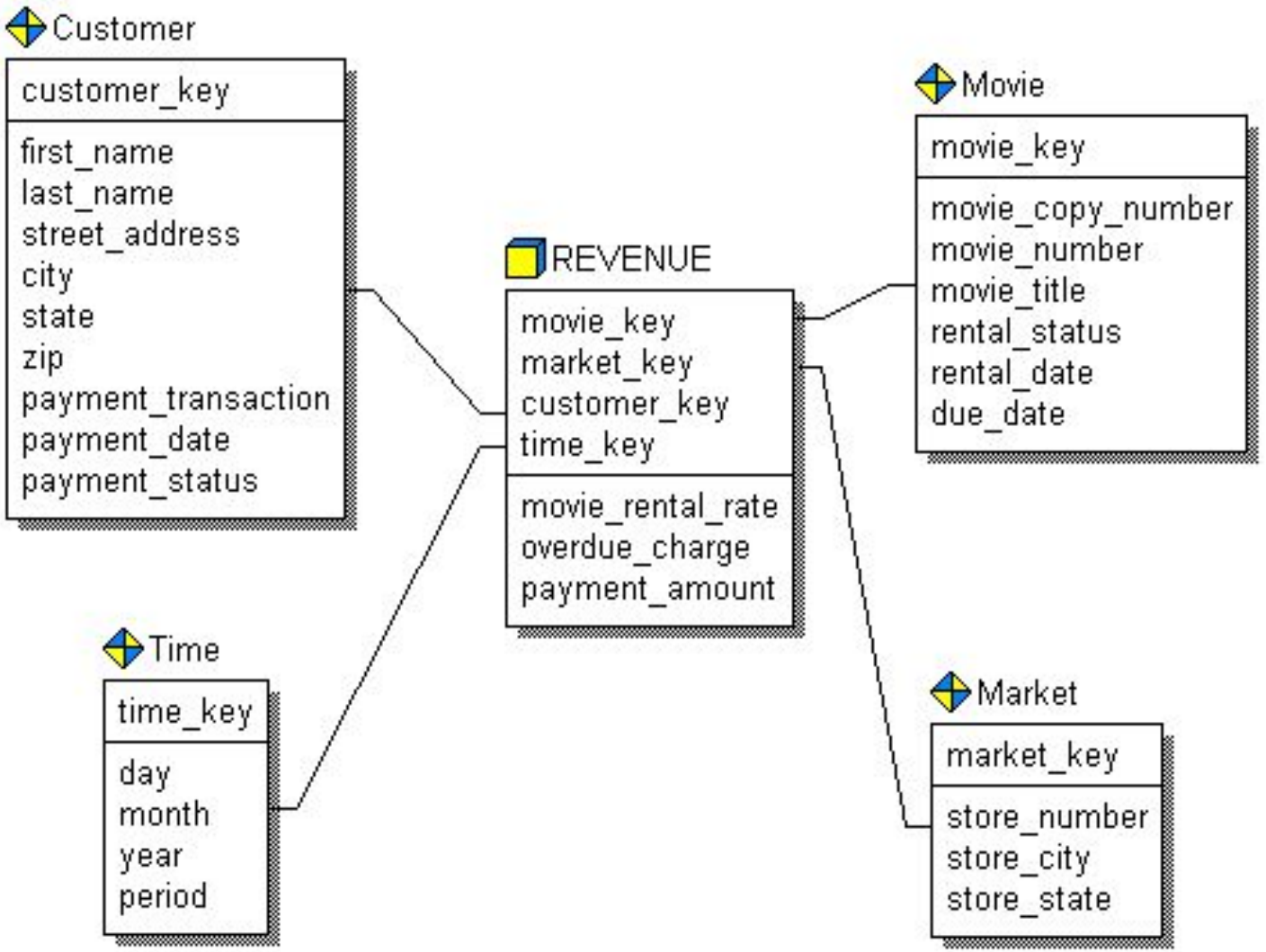
Таблица(ы) фактов

- 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.

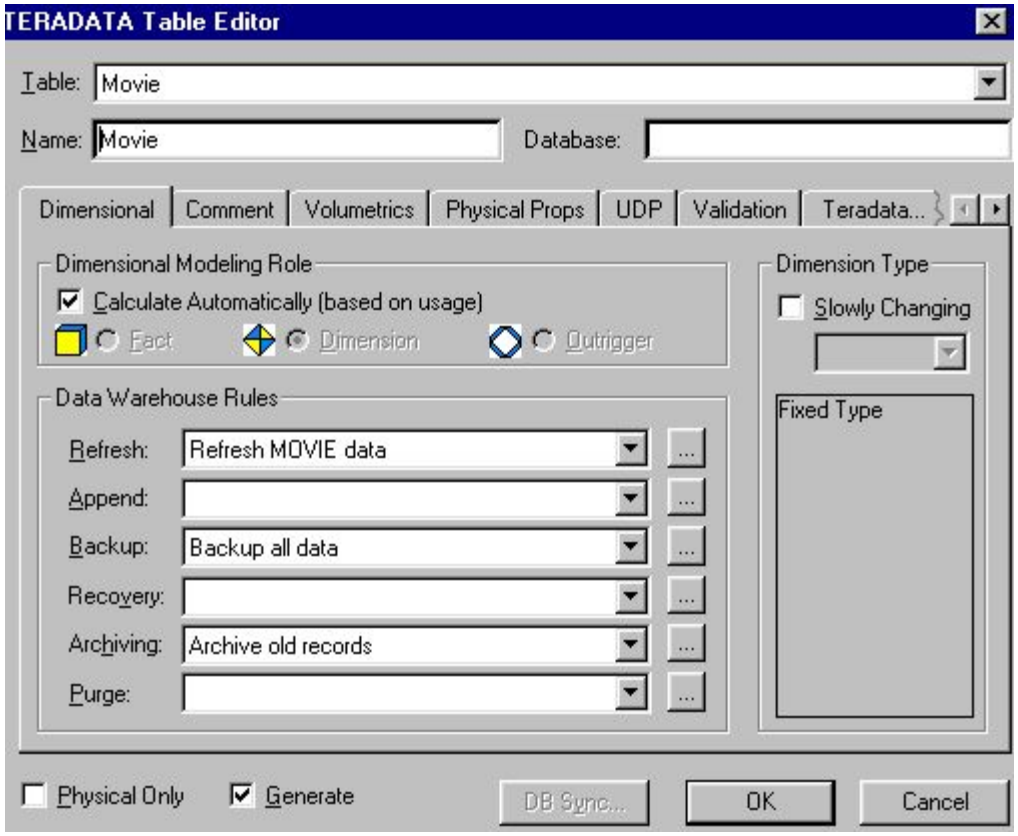
Отличие от схемы «звезда»

- 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)