МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Лекция № 10

Развертывание хранилищ данных

Методы и средства проектирования информационно-аналитических систем

(наименование дисциплины (модуля) в соответствии с учебным планом)

Уровень: специалитет

(бакалавриат, магистратура, специалитет)

Форма обучения: очная

(очная, очно-заочная, заочная)

Направление(-я) подготовки: 10.05.04 «Информационно-аналитические системы безопасности»

(код(-ы) и наименование(-я))

Институт: Институт кибербезопасности и цифровых технологий (ИКБ)

(полное и краткое наименование)

Кафедра: Информационно-аналитические системы кибербезопасности (КБ-2)

(полное и краткое наименование кафедры, реализующей дисциплину (модуль))

Используются в данной редакции с учебного года 2023/24

(учебный год цифрами)

Проверено и согласовано «____» _____20___г.

(подпись директора Института/Филиала с расшифровкой)

Москва 2024 г.

Учебные вопросы:

1. **Концепция систем складирования данных**
2. **Различия между транзакционной и аналитической обработкой данных**
3. **Логическое преобразование данных OLTP-систем и моделирование данных**
4. **Типы хранилищ данных**

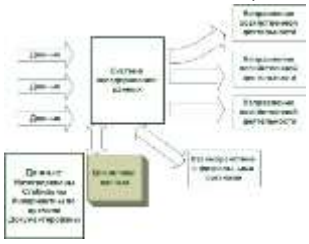
1. Концепция систем складирования данных

Информационная технология складирования данных (**data warehousing**) родилась в недрах компании IBM [1] и была окончательно сформулирована Б. Инмоном и Р. Кимбаллом в 90-х годах прошлого столетия как **метод решения информационно-аналитических задач в области принятия и поддержки решений**. Возникнув на стыке технологии баз данных (БД), систем поддержки принятия решений (СППР — DSS) и компьютерного анализа данных, в дальнейшем концепция складирования данных претерпела эволюцию, поскольку оказалась пригодной для широкого круга приложений в бизнесе, науке и технологии.

Основным посылом разработки концепции складирования данных явилось осознание руководством организаций потребности в анализе накопленных электронных массивов данных.

Во всем мире организации накапливают или уже накопили в процессе своей административно-хозяйственной деятельности большие объемы данных, в том числе и в электронном виде. Эти коллекции данных хранят в себе большие потенциальные возможности по извлечению новой аналитической информации, на основе которой можно и необходимо строить стратегию организации, выявлять тенденции развития рынка, находить новые решения, обусловливающие успешное развитие в условиях конкурентной борьбы.

Системы, построенные на основе информационной технологии складирования данных, обладают рядом характерных особенностей, которые выделяют их как **отдельный класс информационных систем** (ИС). К таким особенностям относятся предметная ориентация системы, интегрированность хранимых в ней данных, собираемых из различных источников, инвариантность этих данных во времени, относительно высокая стабильность данных, необходимость поиска компромисса в избыточности данных.



Хранилище данных (ХД — data warehouse) является местом складирования собираемых в системе данных и информационным источником для решения задач анализа данных и принятия решений. Как правило, объем информации в ХД является достаточно большим. Упрощенно можно сказать, что хранилище данных управляет данными,

которые были собраны как из операционных систем организации (**OLTP-систем — On-Line Trasactions Processing**), так и из внешних источников данных, и которые длительный период времени хранятся в системе.

Одной из главных целей создания систем складирования данных является их ориентация на анализ накопленных данных, т.е. структуризация данных в ХД должна быть выполнена таким образом, чтобы данные эффективно использовались в аналитических приложениях (analytical applications).

Заметим, что задачи анализа накопленных данных решали и до создания концепции складирования данных. В распоряжении аналитиков и сейчас имеется большой набор пакетов программ. Главным отличием использования концепции складирования данных является структуризация, систематизация, классификация, фильтрация и т. п. больших массивов электронной информации в виде, удобном для анализа, визуализации результатов анализа и производства корпоративной отчетности.

Концепция баз данных (БД) как метод представления и накопления данных в электронном виде сформировалась к середине 60-х годов прошлого века в фирме IBM. В 1969 году была создана первая СУБД для управления и манипулирования данными как самостоятельными информационными объектами. В 1970 году была предложена реляционная модель данных для БД [4], и на ее основе начали создаваться популярные ныне реляционные СУБД. В рамках реляционной модели с единых позиций были решены многие проблемы операционной (транзакционной) обработки данных.

С середины 80-х годов прошлого столетия стали интенсивно накапливаться электронные информационные массивы данных организаций, корпораций, научно-исследовательских учреждений. Так, в начале 90-х годов прошлого века только в области химических дисциплин было зарегистрировано более 7000 библиографических, фактографических и смешанных баз данных, ведущие мировые корпорации создали огромные электронные массивы конструкторской документации и документации по управлению производством. В это же время возникло четкое понимание, что сбор данных в электронном виде – не самоцель, накопленные информационные массивы могут быть полезны. Первыми осознали этот факт в области управления бизнесом и производством. В накопленных данных организации находится "информационный снимок" хронологии ее поведения на рынке. Анализ истории административно-хозяйственной деятельности организации позволил существенно увеличить эффективность ее управления, эффективно организовать взаимоотношения с клиентами, производство и сбыт.

Задачи анализа накопленных данных стали перелагаться "на плечи" компьютера и встраиваться в виде аналитических приложений в ИС с БД. Сейчас большинство исследователей сходятся к тому, что отправной точкой разработки концепции складирования данных явился ретроспективный (как иногда еще говорят, исторический) взгляд на данные, накопленные в организации как в электронном, так и в ином виде.

Отметим также, что использование технологий БД и ИС на уже разработанных моделях данных и методиках моделирования данных приводит к ряду проблем для аналитических приложений. Давайте рассмотрим, как управление анализом накопленных (и в этом смысле исторических) данных и какие факторы привели к развитию класса приложений складирования данных.

Предпосылки создания концепции складирования данных

Автоматизированная информационная система (ИС) с БД, будучи средством удовлетворения потребностей пользователей в информации как производственном ресурсе, работает с потоками информации, выраженными в потоках данных и операциях с ними. Как было указано выше, основной акцент на ранних стадиях эксплуатации ИС с БД строился на операционной концепции работы с данными. ИС, грубо говоря, должна была быстро и адекватно "переварить" поток данных для решения поставленных перед ней задач с помощью унифицированного набора операций манипулирования данными. Обработка данных сводилась к операциям вставки, удаления и обновления. Это было зафиксировано первоначально концепцией БД КОДАСИЛ [7].

Совместное действие этих операций в рамках ИС приводило к конфликтам в данных - потерям данных, ошибкам в обновлении и т.д. - так называемым аномалиям в данных. Предложив реляционную модель (которая является достаточно строго математической, а, следовательно, приемлемо контролируемой моделью), Е. Кодд в целом решил ряд проблем и задач операционной обработки данных [4,8-10]. Создание реляционных СУБД позволило достаточно грамотно (с учетом уровня компетентности разработчика) строить системы операционной (или, как ее еще называют, транзакционной) обработки данных - OLTP (On-Line Trasactions Proccessing).

На практике данные в операционных системах могут содержаться столь угодно долго, сколь в них имеется потребность. Несмотря на то, что производители жестких дисков постоянно увеличивают объемы этих дисков, хранить редко используемую информацию не имеет смысла по той простой причине, что производительность многих запросов с ростом объема данных начинает падать и совершенствование подсистем оптимизации запросов СУБД решает проблему ухудшения производительности запросов лишь отчасти. В целом с накоплением данных производительность обработки данных продолжает ухудшаться (эффект больших объемов).

Типичным организационным методом работы с редко используемыми данными является процедура архивизации. Во многих случаях процедура архивизации сводится к простому копированию данных на резервный носитель информации.

Таким образом, одной из проблем при решении задач анализа данных, помимо других скрытых проблем, в рамках операционных систем анализа данных является низкая производительность обработки запросов, которые готовят данные для последующего анализа. Такие запросы увеличивают нагрузку на процессоры ОС и в целом ухудшают обработку потока транзакций в БД, исходящего от систем операционной обработки данных.

Работа с архивом как чистой копией массива данных операционной системы обработки данных не решает проблему производительности. Отсюда простой практический ход - разделить решение задач обработки транзакций и задач анализа данных. В реляционных СУБД производительность запроса может быть улучшена за счет модификации модели данных. Архивные информационные массивы можно наделить структурой, отличной от структуры данных в несущей БД операционной ИС. Разработку таких структур данных можно связать с решением задач ретроспективного анализа данных, накопленных в системе. Это допустимо хотя бы потому, что в задачах анализа данных учитываются далеко не все функциональные зависимости, поддерживаемые в

операционных БД. Поэтому структуру данных архивов стали проектировать под задачи анализа данных, неявно породив тем самым новый класс приложений.

Фундаментальные требования к разработке операционных систем обработки данных и систем анализа данных различны: операционным системам нужна производительность, в то время как системам анализа данных нужны гибкость и широкие возможности для получения результата. Это противоречие в целевой направленности двух классов систем обработки данных явилось одной из основных предпосылок разработки концепции складирования данных Основной побудительный мотив разработки концепции систем складирования данных, следующий из опыта решения задач анализа на данных операционных систем обработки данных

Создание новой концепции потребовало пересмотра ряда традиционных подходов к обработке данных и перестройки технологических процедур. Поскольку перестройка технологических процедур является чрезвычайно затратным мероприятием, важно отметить те причины, которые явились дополнительными побудительными мотивами применения новой концепции на практике.

Системы, доставшиеся в наследство

Одной из первых таких причин является работа с данными, доставшимися по наследству (legacy systems — система, доставшаяся по наследству). Средства вычислительной техники (ВТ) быстро эволюционировали. В 80-х годах прошлого века появились миникомпьютеры на платформах AS/400 и VAX/VMS. Конец восьмидесятых и начало девяностых сделали ОС UNIX популярной серверной платформой для повсеместного введения новой архитектуры клиент /сервер. Начиная с 90-х, быстро стало прогрессировать семейство ОС MS Windows.

В то же время, начиная с 70-х годов, большинство систем обработки данных в сфере бизнеса создавалось на маэнфреймах фирмы IBM. Несмотря на все изменения в операционных платформах, архитектуре вычислительных систем, инструментальных средствах разработки программ и информационных технологиях, значительное количество бизнес-приложений в Америке и Европе продолжало и продолжает работать на оборудовании такого класса, что, кстати, стимулировало новый виток в развитии информационных технологий на мейнфреймах.

За годы эксплуатации в этих системах накоплены огромные бизнес-знания, было зафиксировано значительное количество бизнес-правил. Этот огромный объем информации невероятно трудно перенести на новые аппаратно-программные платформы или в приложения.

Системы, обобщенно называемые системами, доставшимися по наследству (legacy systems), продолжают быть самым большим источником данных для систем анализа данных. Однако время, требуемое на получение результатов работы таких приложений, часто оказывается значительно больше того, которое может позволить себе для ожидания конечный пользователь (по большей части руководство организации) в условиях современного бизнеса.

Перенос данных из централизованного ВЦ на рабочий стол пользователя

Второй причиной стал персональный компьютер, который позволил перенести данные из централизованного вычислительного центра на рабочий стол пользователя (в частности бизнес-аналитика).

Всего за несколько лет персональный компьютер (ПК) прочно утвердился на рабочем столе руководителей бизнеса, аналитиков и финансистов. Такая популярность ПК повлекла за собой интенсивную разработку программного обеспечения, в том числе и для анализа данных бизнеса. Хорошо подготовленные пользователи могут использовать настольные базы данных, которые позволяют им хранить и работать с информацией, извлеченной из источников данных систем, доставшихся по наследству. Персональный компьютер и его программный инструментарий перенесли работу по анализу данных из больших вычислительных центров (ВЦ) на рабочий стол пользователя. Эффективность аналитической работы в особенно крупных организациях стала расти.

Вовлечение конечных пользователей для решения задач управления данными в условиях коллективного их использования не является выходом из создавшейся ситуации. Во-первых, это требует времени и усилий конечных пользователей (а следовательно, денег). Во-вторых, у них есть основная работа – анализ данных, которая им интересна и за которую им платят жалованье. Вовлечение их в работу в сфере информационных технологий совершенно точно приведет к снижению эффективности их основной работы.

Широкое применение персональных компьютеров в анализе данных привело к другой проблеме. Отсутствие общих стандартов представления данных в организации, большая свобода в выборе представления данных конечным пользователем, сбрасывание со счетов требований коллективного использования данных приводит к анархии в работе с данными, и, как следствие, появляется опасная тенденция несогласованности коллективно используемых данных, которая может сказываться на качестве принятия стратегических решений.

Системы поддержки и принятия решений и управленческие информационные системы

Еще одной причиной стало интенсивное использование систем поддержки и принятия решений (СППР — DSS) и управленческих информационных систем (ИСР — EIS, информационная система руководителя). СППР обычно фокусируются на более детальном представлении информации и ориентированы больше на менеджеров среднего уровня. ИСР обеспечивают более высокий уровень консолидации и многоаспектного (многомерного представления) взгляда на данные, поскольку руководители высокого уровня нуждаются в большем многообразии представления тех же самых данных для детального анализа.

Эти два схожих и перекрывающихся по функциям класса систем являются одной из главных предпосылок для создания концепции систем складирования данных. Отметим некоторые признаки, обычно связываемые с системами этого класса.

- В этих системах данные представлены в стандартных терминах бизнеса, а не в закодированной форме (имена полей в БД ИС). Наименования элементов данных и структуры данных в этих системах проектируются для использования конечными пользователями с невысоким уровнем подготовки в области информационных систем.
- Данные в таких системах предварительно обрабатываются в контексте стандартных бизнес-правил, таких как размещения ассигнований по продуктам, производственным единицам и рынкам.
- Допускается консолидированное представление данных по таким категориям, как продукт, производитель и рынок. Хотя в таких системах время от времени допускается

развертывание интегрированных данных, они способны обеспечить доступ ко всей детальной информации в одно и то же время.

В настоящее время системы складирования данных обеспечивают аналитические инструменты для решения таких задач, но их разработка строится не на специфических требованиях аналитиков или исполнителей, а основывается на структуре бизнеса организации. С этой точки зрения системы складирования данных дали новый виток в развитии СППР и ИСР.

Развитие технологий

Не следует забывать также о факторах, связанных с техническим прогрессом в области разработки аппаратного обеспечения ЭВМ и развитием компьютерных технологий в разработке программного обеспечения. Это обстоятельство привело к снижению цен на комплектующие с одновременным ростом их мощности, созданию дружественных интерфейсов для пользователей.

Наиболее важным фактором в развитии складирования данных стало увеличение мощности аппаратной платформы компьютеров, поскольку ХД хранят обычно очень большие объемы информации. Параллельно росла вычислительная мощность ПК и развитое программное обеспечение, которые позволили разработать и внедрить архитектуру клиент/сервер. Почти ко всем ХД можно обратиться с ПК, оснащенного развитыми инструментальными программными средствами. Эти средства изменяются от очень простых обработчиков запросов до мощных графических многомерных средств анализа данных. Создание серверных операционных систем, таких как Windows и Unix, повысило надежность в функционировании и дало мощные возможности распределенной вычислительной среде. Эти технологические факторы способствовали быстрому развитию систем складирования данных.

Создание и распространение Интернет/Интранет привело к тому, что бизнес стал перемещаться в Интернет. Сегодня одной из наиболее значительных областей компьютерной индустрии является разработка интранет-приложений. Грубо говоря, Интранет является совокупностью локальных компьютерных сетей, ориентированных на бизнес, которые основываются на стандартах сети Интернет, хотя проектируются для внутреннего использования в организации. ХД может быть доступно из любой точки сети, как локальной, так и глобальной, и стоимость доступа к нему значительно снижается по сравнению с обычной технологией. С другой стороны, использование технологии Интернет позволяет веб-серверу обеспечить обработку данных в узлах их размещения, что приводит к выполнению всех трудоемких процедур анализа до того, как его результаты представляются пользователю в его браузере.

Структурные изменения в бизнесе

Значительное влияние на формирование концепции складирования данных оказали фундаментальные изменения в организации бизнеса и изменения в его структуре в конце прошлого века. Появление ярко выраженной глобальной экономики изменило требования к информации и спрос на нее. Деятельность организаций пересекла границы своей страны и тем самым стала транснациональной.

Изменение экономических условий побудили большие корпорации к объединению (консолидации) своих усилий. Появление таких механизмов, как реинжиниринг бизнес-процессов (business process reengineering) и перестраиваемость бизнеса (downsizing), вынудило руководителей переоценить практику ведения бизнеса. Пересмотр процедур

ведения бизнеса и изменения в финансовых потоках сыграли важную роль в развитии концепции складирования данных.

Глобализация экономики выдвигает не только требования непрерывного анализа потоков экономических данных, но и определенные требования к сбору и размещению деловой информации. Теперь процесс сбора и свертывания производственных и коммерческих данных от разбросанных по всему миру производственных подразделений оказывает сильное влияние на принятие решений в корпорациях. Глобализация делает процедуры размещения данных в централизованном хранилище данных более сложными. Колебания стоимости валют или сезонные колебания в сбыте продукции в различных регионах мира добавили трудности к складированию данных и делают анализ данных более сложным.

Появление стандартов для программного обеспечения бизнеса

Еще одним важным фактором, который повлиял на развитие ХД, явилось появление специализированных поставщиков решений в автоматизации бизнеса. Фирмы-разработчики ПО SAP AG, Baan, Oracle, Microsoft, IBM и др. предлагают быстро адаптируемые к бизнес-процессам программные продукты для управления бизнесом. Разработка комплексного ПО для управления бизнесом привела к интенсификации процессов стандартизации бизнеса и стандартизации программного обеспечения. Информация в ХД поступает в унифицированном виде из всех ИС управления бизнесом, а не только из систем, доставшихся по наследству. Следует также отметить тенденцию последних лет в разработке ПО данного класса приложений — обеспечение возможности собирать информацию в ХД из данных любых внешних источников (например продукты компании SAS).

Требования пользователей

Один из наиболее важных результатов массивной инвестиции в технологию и создание высокопроизводительных ПК привел к созданию методов анализа, основанного на здравом смысле (technology-savvy business analyst). Даже если технология здравого смысла конечного пользователя не всегда выгодна для многих проектов, тенденция ее применения привела к созданию более сложных технологий анализа для сегодняшнего бизнеса. Однако именно на технологии здравого смысла была продемонстрирована выгода использования хранилищ данных и развития логических и физических моделей.

Текстовые редакторы и крупноформатные таблицы, столь популярные на первых ПК, оказали существенное влияние на представление данных в хранилищах данных.

Очень сильно влияют на тенденции развития информационных технологий требования к информации, предъявляемые средним и высшим звеном управляющего персонала. Информационные технологии стали производственным ресурсом компаний – это первый результат таких требований. Электронная почта, Интернет, мобильный телефон и карманный ПК вовлечены в процесс управления. Это также требования, приходящие от руководителей организаций и компаний. Им нужен быстрый и качественный доступ к аналитической информации в любой момент времени и по любому виду каналов связи.

Как видно из вышесказанного, потребности бизнеса в новых экономических условиях, создание мощной программно-аппаратной платформы, распространение информационных технологий создали предпосылки рождения нового класса приложений

— систем складирования данных и концепции ХД как информационного носителя для таких приложений.

Отметим, что складирование данных — это развивающаяся технология. Как и для любой развивающейся технологии, определенная доля осторожности должна присутствовать при оценке действий производителей *ПО* ХД, пытающихся позиционировать себя среди конкурентов. Например, дискуссии о размерах ХД — с какого размера *хранилище данных* можно считать собственно хранилищем? С 50 ГБ? Заметим, что в некоторых областях исследования размер анализируемого массива может быть очень небольшим. Просто нет данных. А *анализ* такого массива возможен.

Рассмотрим основные элементы концепции складирования данных.

Извлечение данных из операционных систем

Главный элемент концепции складирования данных состоит в том, что к данным, сохраняемым для анализа, может быть обеспечен наиболее эффективный доступ только при условии выделения их из операционной (транзакционной) системы, т.е. данные из операционной системы должны быть вынесены в отдельную *систему складирования данных*. Такой подход носит исторический характер. Из-за ограничений в аппаратном обеспечении и технологии, для того чтобы обеспечить производительность транзакционной системы, данные архивировались на магнитных лентах или носителях вне такой системы. Проблема доступа к ним требовала определенных технологических решений.

Нужно отметить, что с развитием концепции позиция отделения данных для анализа от данных в OLTP-системе претерпела мало изменений. Она стала более формальной и обогатилась за счет применения средств многомерного анализа данных. В настоящее время ХД можно строить и на существующей OLTP-системе, и над ней, и как самостоятельный объект. Это должно решаться руководителем ИТ-проекта в рамках выбора архитектуры ХД.

Необходимость интегрирования данных из нескольких OLTP-систем

Системы складирования данных наиболее полезны, когда данные могут быть извлечены более чем из одной OLTP-системы. Когда данные должны быть собраны от нескольких бизнес-приложений, естественно предположить, что это нужно сделать в месте, отличном от места локализации исходных приложений. Еще до создания структурированных ХД аналитики во многих случаях комбинировали данные, извлеченные из разных систем, в одну крупноформатную таблицу или базу данных. ХД может очень эффективно воедино собрать данные от конкретных приложений, таких как продажи, маркетинг, финансы, производство, с учетом их накопления, т.е. сохранить временные ряды основных показателей бизнеса — так называемые исторические данные.

Заметим, что одним из свойств данных, собранных из различных приложений и используемых аналитиками, является возможность делать перекрестные запросы к таким данным. Во многих ХД атрибут "время" является естественным критерием для фильтрования данных. Аналитиков интересует поведение временных рядов данных, характеризующих процессы бизнеса.

Целью многих *систем складирования данных* является обзор деятельности типа "год за годом". Например, можно сравнивать продажи в течение первого квартала этого года с продажами в течение первого квартала предшествующих лет. Время в ХД — фундаментальный атрибут *перекрестных запросов*. Например, аналитик может

попытаться оценить влияние новой компании маркетинга, проходящей в течение определенных периодов, рассматривая продажи в течение тех же самых периодов. Способность устанавливать и понимать корреляцию между деятельностью различных подразделений в организации часто приводится как один из самых главных аргументов о пользе *систем складирования данных*.

Система складирования данных не только может работать как эффективная платформа для консолидации данных из различных источников, но может также собирать многократные версии данных из одного приложения. Например, если организация перешла на новое программное обеспечение, то ХД сохранит необходимые данные из предыдущей системы. В этом отношении *система складирования данных* может служить средством интеграции наследуемых данных, сохраняя преемственность анализа при смене программно-аппаратной платформы OLTP-системы.

2. Различия между транзакционной и аналитической обработкой данных

Одной из наиболее важных причин отделения данных для анализа от данных OLTP-систем было потенциальное падение производительности обработки запросов при выполнении процессов анализа данных. Высокая производительность и небольшое время ответа — критические параметры OLTP-систем. Потерю производительности и объем накладных затрат, связанных с обработкой предопределенных запросов, обычно легко оценить. С другой стороны, запросы для анализа данных в ХД трудно предсказать, и следовательно, для них сложно оценить время выполнения запроса.

OLTP-системы спроектированы для оптимального выполнения предопределенных запросов в режиме работы, близком к режиму реального времени. Для таких систем обычно можно определить распределение нагрузки во времени, определить время пиковых нагрузок, оценить критические запросы и применить к ним процедуры оптимизации, поддерживаемые современными СУБД. Также относительно легко определить максимально допустимое время ответа на определенный запрос в системе. Стоимость времени ответа такого запроса может быть оценена на основе отношения стоимости выполнения операций ввода-вывода / стоимость затрат на трафик по сети. Например, для системы обработки заказов можно задать число активных менеджеров по оформлению заказов и среднее число заказов в течение каждого часа работы.

Несмотря на то, что многие из запросов и отчетов в *системе складирования данных* предопределены, почти невозможно точно предсказать поведение показателей системы (время отклика, трафик сети и т.п.) при их выполнении. Процесс исследования данных в ХД происходит зачастую непредсказуемым путем. Руководители всех рангов умеют ставить неожиданные вопросы. В процессе анализа могут возникать непредопределенные (ad hoc) запросы, которые вызваны неожиданными результатами или непониманием конечным пользователем используемой модели данных. Далее, многие из процессов анализа имеют тенденцию принимать во внимание многие аспекты деятельности организации, в то время как OLTP-системы хорошо сегментированы по видам деятельности. Пользователю может потребоваться более детальная информация, чем хранящаяся в итоговых таблицах. Это может привести к соединению двух или более огромных таблиц, что закончится созданием временной таблицы объемом, равным произведению числа строк в каждой таблице, и резко снизит производительность системы.

Данные в системах складирования данных остаются неизменными

Другое ключевое свойство данных в *системе складирования данных* состоит в том, что данные в ХД остаются неизменными. Это означает, что после того, как данные разместятся в ХД, они не могут быть изменены. Например, статус заказа не меняется, размер заказа не меняется, и т. д. Эта характеристика ХД имеет большое значение для отбора типов данных при размещении их в ХД, а также выбор момента времени, когда данные должны быть занесены в ХД. Последнее свойство называется *гранулированностью данных*.

Рассмотрим, что означает для данных быть неизменяемыми. В OLTP-системе объекты данных проходят через постоянные изменения своих атрибутов. Например, заказ может многократно изменять свой статус до того, как будет оформлен. Или, когда изделие собирается на сборочной линии, к нему применяется множество технологических операций. Вообще говоря, данные из OLTP-системы нужно загружать в ХД лишь тогда, когда обработка их в рамках бизнес-процессов будет полностью завершена. Это может означать завершение заказа или цикла производства изделия. Как только заказ закончен и отправлен, он вряд ли поменяет свой статус. Или, как только изделие собрано и сдано на склад, оно вряд ли попадет на первую стадию сборочного цикла.

Другим хорошим примером может быть размещение в ХД снимка постоянно изменяющихся данных в определенные моменты времени. Модуль управления запасами в OLTP-системе может изменять запас почти в каждой транзакции; невозможно занести все эти изменения в ХД. Вы можете определить, что такой снимок состояния запаса следует вносить в ХД каждую неделю или ежедневно, так, как это будет принято для анализа в конкретной организации. Данные такого снимка, естественно, неизменяемы.

После того, как данные занесены в ХД, их модификация возможна в крайне редких случаях. Очень трудно (хотя такие попытки есть) поддерживать динамические данные в ХД. *Задача синхронизации* часто изменяемых данных в OLTP-системах и *системах складирования данных* еще далека от приемлемого решения. Здесь следует также упомянуть, что размещение динамично меняющихся данных в ХД в настоящее время является предметом интенсивных исследований. Например, разработка процедур поддержки в ХД медленно меняющихся таблиц измерений является задачей, которая уже находит свое решение на уровне ПО производителей решений в области ХД.

Данные в хранилище данных хранятся значительно более длительное время, чем в OLTP-системах

Данные в большинстве OLTP-систем архивируются сразу после того, как они становятся неактивными. Например, заказ может стать неактивным после того, как он выполнен; банковский счет может стать неактивным после того, как он был закрыт. Главная причина для архивирования неактивных данных — это производительность OLTP-системы (зачем хранить данные, если к ним не обращаются). Большие объемы таких данных могут заметно ухудшить производительность выполнения запросов в предположении, что обрабатываются только активные данные. Для обработки таких данных в СУБД предлагаются различные процедуры разбиения базовых таблиц на секции. С другой стороны, поскольку ХД предназначены, в частности, быть архивом для OLTP-данных, данные в них хранятся в течение очень длительного периода.

Фактически, проект *системы складирования данных* может начинаться и без любого определенного плана архивирования данных из ХД. Стоимость сопровождения данных

после их загрузки в хранилище невысока. Наибольшие затраты при создании хранилища выпадают на *трансформацию данных* (*data transfer*) и их очистку (*data scrubbing*). Хранение данных в течение пяти и более лет типично для *систем складирования данных*. Поэтому процедурам архивизации данных из ХД на стадиях их создания и эксплуатации в начале периода можно не уделять много времени. Особенно если учесть снижение цен на аппаратные средства ЭВМ.

Иначе говоря, отделение данных OLTP-систем от данных систем анализа является фундаментальной концепцией складирования данных. Сейчас бизнес невозможен без принятия обоснованных решений. Такие решения могут быть построены на основе всестороннего анализа результатов выполнения бизнес-процессов в организации и деятельности организации на рынке товаров и услуг. Время принятия решений в современных условиях и потоках информации сокращается. Роль создания и поддержки *систем анализа данных* на основе новых информационных технологий возрастает. ХД является одним из основных звеньев применения таких технологий.

Можно выделить следующие причины для разделения данных *систем складирования данных* и *систем операционной обработки данных* (рис. 1.5).

- Различие целевых требований к *системам складирования данных* и OLTP-системам.
- Необходимость собирать данные в ХД из различных информационных источников, т.е. если данные генерируются в самой OLTP-системе, то для *системы складирования данных* в большинстве случаев данные генерируются вне ее.
- Данные, попадая в ХД, остаются в большинстве случаев неизменными.
- Данные в ХД сохраняются длительное время.

3. Логическое преобразование данных OLTP-систем и моделирование данных

Данные в ХД логически являются преобразованными, когда они перенесены в него из OLTP-системы или другого внешнего источника. Проблемы, связанные с логическим преобразованием данных при переносе их в ХД, могут потребовать значительного анализа и усилий проектировщиков. Архитектура *системы складирования данных* и модели ХД имеют огромное значение для успеха таких проектов. Ниже будут рассмотрены некоторые фундаментальные понятия реляционной теории БД, которые полностью не применимы к *системам складирования данных*. Даже при том, что большинство ХД развернуто на реляционных БД, некоторые основные принципы реляционных БД сознательно нарушаются при создании логической и физической модели ХД.

Модель данных ХД определяет его логическую и физическую структуру. В отличие от просто архивированных данных, в данном случае невозможно обойтись без процедур детального моделирования. Такое моделирование на ранних стадиях проекта системы складирования необходимо для создания эффективной системы, охватывающей данные всех бизнес-процессов и процедур организации.

Процесс моделирования данных должен структурировать данные в ХД в виде, не зависимом от реляционной модели данных системы, которая поставляет эти данные. Как будет показано ниже, модель ХД, вероятно, будет менее нормализована, чем модель OLTP-системы – источника данных.

В OLTP-системах данные по разным подсистемам могут значительно перекрываться. Например, информация относительно разрабатываемых изделий используется в различных формах во многих подсистемах OLTP-системы. *Система складирования данных* должна объединить все такие данные в одной системе. Некоторые атрибуты объектов, которые являются существенными для OLTP-системы, окажутся ненужными для ХД. Могут появиться новые атрибуты, так как сущность (entity) в ХД изменяет свое качество. Основное требование — все данные в ХД должны участвовать в процессе анализа.

Модель данных ХД должна быть расширена и структурирована таким образом, чтобы данные от различных приложений могли быть добавлены. Проект ХД в большинстве случаев не может включать данные от всех возможных бизнес-приложений организации. Обычно объем данных в ХД увеличивается по принципу инкремента: данные экстрагируются из OLTP-систем и добавляются в ХД определенными порциями. Начинают с сохранения особенно существенных данных, затем планомерным образом наращивают по мере необходимости их объем.

Модель хранилища данных подстраивается под структуру бизнеса

Следующий важный момент состоит в том, что логическая модель ХД настраивается на структуру бизнеса (ориентирована на предметную область), а не на агрегацию логических моделей конкретных приложений. Сущности (объекты), поддерживаемые в ХД, аналогичны сущностям (объектам) бизнеса — таким как клиенты, продукция (товар), заказы и дистрибьютеры. В рамках конкретных подразделений организации может быть очень узкое представление об объектах бизнеса организации, например, о клиентах. Так, группа обслуживания ссуд в банке может знать что-либо о клиенте только в контексте одной или нескольких выданных ссуд. Другое подразделение того же банка может знать о том же клиенте в контексте *депозитного счета*. Представление данных о клиенте в ХД намного превышает аналогичное представление конкретного подразделения банка. Клиент в ХД представляет клиента банка во всех его взаимоотношениях с банком. С точки зрения реляционной теории меняется базисный набор функциональных зависимостей, поддерживаемых в БД.

ХД следует строить на атрибутах сущностей бизнеса (предметно ориентированно), собирая данные об этих сущностях из различных источников. Структура данных любого отдельного источника данных, вероятно, будет неадекватна для ХД. На структуру данных конкретного приложения оказывают влияние такие факторы, как:

- вид конкретного бизнес-процесса. Так, в автоматизированной подсистеме закупок структура данных может быть продиктована характером бизнес-процедур закупок на данном секторе рынка;
- влияние модели действующих систем. Например, исходное приложение может быть достаточно старым и учитывать развитие модели данных за счет изменения структуры БД приложения. Многие такие изменения могут быть плохо документированы;
- ограничения программно-аппаратной платформы. Логическая структура данных может не поддерживать некоторые логические взаимоотношения между данными или иметь ограничения, связанные с ограничениями программно-аппаратной платформы.

Модель ХД не связана с ограничениями моделей данных источников. Для нее должна быть разработана модель, которая отражает структуру бизнеса организации, а не структуру бизнес-процесса. Такая *расширенная модель* данных должна быть понятна как аналитикам, так и менеджерам. Таким образом, проектировщик ХД должен выполнить настройку объектов ХД к структуре бизнеса организации, с учетом ее бизнес-процессов и бизнес-процедур.

Преобразование информации, описывающей состояние объектов в OLTP-системе

Следующий важный момент состоит в том, что перед размещением данных в ХД они должны быть преобразованы. Большинство данных из OLTP-системы или иного внешнего источника не могут поддерживаться в ХД. Многие из атрибутов объектов в OLTP-системе очень динамичны и постоянно изменяются. Многие из этих атрибутов не загружаются в ХД, другие же атрибуты являются статичными во времени и загружаются в ХД. ХД вообще не должно содержать информации об объектах, которые являются динамическими и постоянно находятся в состоянии модификации.

Чтобы понять, что означает потеря информации, описывающей текущее состояние объекта, рассмотрим пример системы управления заказами, которая отслеживает состояния запасов при заполнении заказа. Сначала рассмотрим сущность "Заказ" в OLTP-системе. Заказ может пройти множество различных статусов или состояний, прежде чем он будет выполнен и обретет *статус завершенного*. Статус заказа может указывать, что он готов к заполнению, что заказ заполняется, возвращен обратно на доработку, готов к отгрузке и т.д. Конкретный заказ может пройти много состояний, которые отражаются в статусе заказа и определяются бизнес-процессами, которые применялись к нему. Практически невозможно перенести все атрибуты такого объекта в ХД. *Система складирования данных*, вероятно, должна содержать только один конечный снимок такого объекта, как заказ. Таким образом, объект "Заказ" должен быть преобразован для размещения в ХД. Тогда в ХД может быть собрана информация о многих объектах типа "заказ" и построен окончательный объект ХД – "Заказ".

Рассмотрим более сложный пример *трансформации данных* при управлении запасом товара в OLTP-системе. Запас может изменяться в каждой транзакции. Количество конкретного товара на складе может быть уменьшено транзакцией подсистемы заполнения заказа или увеличено при поступлении купленного товара. Если система обработки заказа выполняет десятки тысяч транзакций в день, то, вероятно, фактический уровень запаса в БД будет иметь много состояний и зафиксируется во многих снимках в течение этого дня. Невозможно зафиксировать все эти постоянные изменения в БД и перенести их в ХД. Отображение такого поведения объекта в системе – источнике данных по-прежнему является одной из нерешенных задач в *системах складирования данных*. Есть ряд подходов к решению этой проблемы. Например, можно периодически фиксировать снимки уровня запаса в ХД.

Этот подход может быть применим к очень большой части данных в OLTP-системах. В свою очередь, такое решение повлечет за собой ряд задач, связанных с выбором периода времени, объема снимаемых данных и т.д. Таким образом, большая часть данных о состоянии объектов в OLTP-системе не может быть непосредственно перенесена в ХД. Они должны быть преобразованы на логическом уровне.

Денормализация модели данных

Следующий момент в проектировании реляционных ХД состоит в решении вопроса о том, насколько важно в ХД соблюдать принципы реляционной теории, а именно: разрешить *денормализацию* модели, в частности, для увеличения производительности запросов. Прежде чем мы рассмотрим *денормализацию* модели данных в контексте складирования данных, давайте кратко вспомним основные моменты теории реляционных БД и *процесса нормализации*. Е.Ф. Кодд разработал реляционную теорию БД в конце 60-х прошлого века, когда он работал в исследовательском центре IBM. Сегодня большинство популярных платформ БД полностью следует этой модели. Реляционная модель БД — коллекция двухмерных таблиц, состоящих из рядов и колонок. В терминологии реляционной модели эти таблицы, строки и колонки соответственно называются отношениями или сущностями, кортежами, атрибутами (attribute) и доменами (domain). Модель идентифицирует уникальные ключи (Key) для всех таблиц и описывает отношения между таблицами через значения атрибутов (ключей).

Нормализация (Normalization) является процессом моделирования реляционной БД, где отношения или таблицы разбиваются до тех пор, пока все атрибуты в отношении полностью не будут определяться его первичным ключом. Большинство проектировщиков пытаются достичь третьей нормальной формы (3НФ) на всех отношениях до того, как они будут денормализоваться по тем или иным причинам. Три последовательных этапа нормализации реляционных БД кратко описаны ниже.

- Первая нормальная форма *1NF* (*1НФ*). Говорят, что отношение находится в первой нормальной форме, если оно описывает одну единственную сущность и не содержит в качестве атрибутов массивов или повторяющихся групп значений. Например, таблица заказов, включающая в себя позиции заказа, не будет находиться в первой нормальной форме, поскольку содержит повторяющиеся атрибуты для каждой позиции заказа.
- *Вторая нормальная форма 2NF* (2НФ). Говорят, что отношение находится во *второй нормальной форме*, если оно находится в первой нормальной форме и все неключевые атрибуты функционально полно зависят от *первичного ключа отношения*.
- *Третья нормальная форма 3NF* (3НФ). Говорят, что отношение находится в *третьей нормальной форме*, если оно находится во *второй нормальной форме* и его неключевые атрибуты полностью независимы друг от друга.

Процесс нормализации приводит к разбиению исходного отношения на несколько независимых отношений. Каждому отношению в БД отвечает по крайней мере одна таблица. Несмотря на большую гибкость реляционной модели в представлении данных, она может быть более сложной и трудной для понимания. Кроме того, полностью нормализованная модель может быть очень неэффективной при реализации. Поэтому проектировщики БД при преобразовании нормализованной логической модели в физическую модель допускают значительную *денормализацию*. Основное назначение *денормализации* состоит в ограничении межтабличных соединений в запросах.

Еще одной из причин *денормализации* модели ХД является, так же, как и для операционных систем, производительность и простота. Каждый запрос в реляционных БД имеет свою стоимость выполнения (cost performance). Стоимость выполнения запросов очень высока в ХД из-за количества обрабатываемых данных в запросе (и межтабличных

соединений, число которых растет пропорционально размерности модели). Соединение трех маленьких таблиц в OLTP-системе может иметь приемлемую стоимость выполнения запроса, но в *системе складирования данных* выполнение такого соединения может занять очень много времени.

Статичность взаимосвязей в исторических данных

Денормализация является важным процессом в моделировании ХД: взаимосвязь между атрибутами не изменяется для исторических данных. Например, в OLTP-системах товар может быть частью другого товара группы "А" в этом месяце и частью товара группы "В" в следующем месяце. В нормализованной модели данных для отображения этого факта необходимо включить атрибут "группа товаров" в отношение (сущность) "товар", но не в отношение (сущность) "заказ", которая формирует заказы на этот товар. В сущность "заказ" включается только идентификатор товара. Реляционная теория будет требовать соединения между таблицами "Заказ" и "Товар" для определения группы товаров и других атрибутов этого продукта. Этот факт (функциональная зависимость) не имеет значения для ХД, поскольку сохраняемые данные относятся к уже выполненным заказам, т.е. принадлежность товара группе уже зафиксирована (фактически указанная функциональная зависимость не поддерживается). Даже если товар принадлежал различным группам в разное время, взаимосвязь между группой товаров и товаром каждого отдельного заказа статична. Таким образом, это не является денормализацией для ХД. В данном случае функциональная зависимость OLTP-системы не используется в ХД.

В качестве другого примера возьмем цену товара. Цены в OLTP-системе могут изменяться постоянно. Некоторые изменения этих цен могут быть перенесены в ХД, как периодические снимки таблицы "Цена товара". В ХД история прайс-листа товара зафиксирована и уже привязана к заказам, т.е. не нужно динамически определять прайс-лист при обработке заказа, поскольку он уже был применен к сохраненному заказу. В реляционных БД проще поддерживать динамические взаимоотношения между сущностями бизнеса, в то время как ХД содержит взаимосвязи между *сущностями предметной области* в заданное время.

Концепция логического преобразования данных приложений-источников, рассмотренная выше, требует определенных усилий при реализации и очень полезна при разработке ХД.

Физическое преобразование данных приложений источников

Важным моментом в *системах складирования данных* является физическое преобразование данных. Эти процедуры в складировании данных известны как процессы *очистки данных* ("*data scrubbing*", "data staging" или "data *purge*"). Процесс *очистки данных* является наиболее интенсивным и трудоемким в любом проекте создания ХД. Физическое преобразование включает использование стандартных терминов предметной области ХД и стандартов данных. В течение процесса физического преобразования данные находятся в некотором промежуточном файле до того, как будут занесены в ХД. Когда данные собираются из многих приложений, их целостность может быть проверена в течение процесса формирования преобразованных данных до загрузки в ХД.

Термины и имена *атрибутов сущностей*, используемые в OLTP-системах, в процессе преобразования данных для ХД преобразуются в универсальные, стандартные термины, принятые для данной сферы бизнеса. Приложения могут использовать

сокращения или трудные для понимания термины по множеству различных причин. Программно-аппаратная платформа может ограничивать длину и формат имен, а бизнес-приложения могут применять в разных предметных областях общие термины. В ХД необходимо пользоваться стандартными бизнес-терминами, которые понятны сами по себе большинству пользователей.

Идентификатор клиента (покупателя) в OLTP-системе может быть назван "Покуп.", "покуп_ид" или "покуп_но". Далее, различные приложения таких систем могут использовать различные имена (синонимы) при ссылке к одному и тому же атрибуту сущности. Проектировщик ХД выбирает простой стандартный бизнес-термин, такой, как "Идентификатор клиента". Таким образом, имена *атрибутов сущностей* из подающих систем должны быть унифицированы для использования в ХД.

Различные подсистемы OLTP-систем и внешних источников данных могут использовать различное определение доменов атрибутов на физическом *уровне представления данных*. Так, атрибут типа "идентификатор продукта" в одной системе имеет длину от 12 символов, а в другой — 18 символов. С другой стороны, ПО одних существующих систем может иметь ограничения на определение длин имен атрибутов и бедный набор типов для определения доменов, а в других такие ограничения могут отсутствовать и может предоставляться широкий выбор типов атрибутов.

При определении атрибутов в физической модели ХД необходимо использовать такие длины и типы данных в определении домена атрибута, которые позволили бы учесть как требования предметной области, так и возможности систем — источников данных. Определение стандартов доменов для ХД является одной из важных задач проектировщиков ХД. Правила преобразования доменов атрибутов систем — источников данных в домены атрибутов ХД следует фиксировать в метаданных ХД.

Все атрибуты в ХД должны согласованно использовать предопределенные значения. В различных приложениях могут быть приняты различные соглашения по предопределенным значениям атрибутов. К таким предопределенным значениям относятся значения по умолчанию, значения, заменяющие null-значения, и т. п. Например, признак пола в различных системах может иметь различные значения: в одних это символьные значения "М" и "Ж", в других — цифровые значения 0 и 1. Более неприятным примером является случай, когда одно значение данных используется в приложении в нескольких целях, т.е. атрибут на самом деле представляет множественное значение. Например, когда в атрибуте "тип метода измерения" две первые цифры означают метод измерения, а две вторые — метод физического контроля измерения. Такие различные значения перед загрузкой в ХД должны быть преобразованы к принятому в ХД предопределенному значению.

В некоторых системах — источниках данных могут отсутствовать значения (проблема пропущенных значений, "missing data") или преобразование для них не может быть выполнено ("*corrupt data*" — данные, для которых преобразование не может быть выполнено). Важно, чтобы в процессе преобразования такие данные принимали в ХД значения, которые позволяли бы пользователям интерпретировать их правильно. Одним атрибутам можно просто назначить разумное значение по умолчанию в случае отсутствия значения или конфликтов при преобразовании, а другим атрибутам — определить значения из значений прочих атрибутов. Например, пусть в сущности "Заказ" значение атрибута единицы измерения товара пропущено. Это значение может быть получено из

соответствующего атрибута сущности "Товар" этой системы-источника. Для некоторых атрибутов не существует подходящих значений по умолчанию в случае, когда их значения отсутствуют. Для таких пропущенных значений в ХД следует также определять значение по умолчанию, например, как null-значение.

Таким образом, в процессе преобразования данных проектировщик ХД должен привести данные систем-источников к определенным стандартам (рис. 1.6), а именно:

- стандартизовать наименования атрибутов в ХД;
- определить одинаковые домены для одних и тех же атрибутов различных систем-источников;
- принять соглашения о значениях по умолчанию для пропущенных данных;
- принять соглашения о предопределенных значениях атрибутов.

В табл. 1.1 приведены основные отличия использования данных в *системах операционной обработки данных* и *системах анализа данных*.

Таблица 1.1.

	Операционные системы обработки данных	***Системы складирования данных***
Частота обновления данных	режим реального времени	периодически
Данные структурируются с целью	обеспечения целостности данных	обеспечения простоты выполнения запросов
Оптимизируются для обеспечения	процесса выполнения транзакций	процесса выполнения выборки данных

Агрегация и суммирование конкретных данных

При создании *системы складирования данных* важным моментом является сбор и определение требований пользователей. Как правило, такие требования позволяют оценить число вопросов, на которые система должна давать ответы. Большинство вопросов носят аналитический характер. Аналитический характер вопроса подразумевает определенный уровень агрегации и суммирования данных перед получением конечного результата. Во многих современных *системах складирования данных* при их создании закладывается определенный набор предопределенных и автоматически генерируемых итоговых отчетов и справок. Например, руководителям организации необходимо знать картину продаж производимой продукции, что предполагает суммирование продаж как в денежном, так и в товарном выражении за неделю, месяц, квартал, год. Подведение итогов деятельности организации в такой форме обычно делается по товарам, клиентам и каналам сбыта.

Агрегацию и суммирование конкретных данных можно выполнять и средствами СУБД в OLTP-системах. В реляционных ХД эти операции выполняются на таблицах, которые поддерживаются независимо от таблиц OLTP-систем. С этой точки зрения ХД предоставляет более широкие возможности в построении итоговых отчетов и запросов.

Первой такой возможностью является применение широкого спектра бизнес-правил для агрегации детальных данных. Эти правила могут быть реализованы в виде набора фильтров, которые применяются к данным. Данные в ХД могут анализироваться в соответствии с этими правилами с различных точек зрения. Применение этих правил приводит к использованию в ХД большого числа многотабличных соединений, самой

трудоемкой *реляционной операции*. Размещение и обработка детальных данных в ХД может быть выполнена более эффективно, чем в соответствующей OLTP-системе.

Каждая точка зрения на детальные данные определяет аспект их анализа. Таким образом, детальные данные как бы становятся точками многомерного пространства, в котором оси определяются точкой зрения на данные. Такие оси называются измерениями. Проектировщику ХД необходимо определить предопределенные измерения детальных данных, размещаемых в ХД, для проведения многомерного анализа этих данных пользователями.

Например, для данных о продажах можно рассмотреть четыре точки зрения на их анализ: по товарам, по покупателям, по каналам сбыта, по регионам. Таким образом будет получено четыре измерения для детальных данных о продажах. В рамках каждого измерения может быть введена иерархия. Регион может рассматриваться как страна, область, район, населенный пункт. При проектировании ХД проектировщик должен учесть такие аспекты анализа и представления детальных данных.

Определение хранилища данных

После проведенного выше обсуждения причин и факторов, повлиявших на рождение концепции ХД, подведем итоги и дадим определения.

Концепция ХД была предложена в начале 90-х годов прошлого столетия как основа методологии организации данных в системах поддержки и принятия решений. Согласно классическому определению Б. Инмона, **хранилище данных** [2] **есть предметно-ориентированная, интегрированная, неизменяемая и поддерживающая хронологию электронная коллекция данных для обеспечения процесса принятия решений**.

Предметная ориентированность. Информация в ХД организована в соответствии с основными аспектами деятельности предприятия (заказчики, продажи, склад и т.п.), т.е. бизнес-процессами. Это является принципиальным отличием ХД от оперативной БД, где данные организованы в соответствии с операциями (выписка счетов, *отгрузка товара* и т.п.), т.е. бизнес-операциями. Предметная организация данных в ХД способствует как значительному упрощению анализа, так и повышению скорости выполнения аналитических запросов. Выражается она, в частности, в использовании иных, чем в операционных системах, схемах организации данных.

Интегрированность. Исходные данные извлекаются из операционных БД, проверяются, очищаются, приводятся к единому виду, в нужной степени агрегируются (то есть вычисляются суммарные показатели) и загружаются в ХД. Такие интегрированные данные намного проще анализировать.

Привязка ко времени. Данные в хранилище всегда напрямую связаны с определенным периодом времени. Данные, выбранные из операционных БД, накапливаются в хранилище в виде исторических слоев, каждый из которых относится к конкретному периоду времени. Это позволяет анализировать тенденции в развитии бизнеса.

Неизменяемость. Попав в определенный исторический слой ХД, данные уже никогда не будут изменены. Это также отличает ХД от операционных БД, в которой данные все время меняются и один и тот же запрос, выполненный дважды с интервалом в 10 минут, может дать разные результаты. Стабильность данных также облегчает их анализ.

Концепция ХД оказалась пригодной для решения задач анализа данных не только в бизнесе, но и в науке и технологии. Следует отметить, что в определении соединены две различные функции:

- сбор, организация и подготовка данных для анализа в виде постоянно наращиваемого набора данных;
- собственно анализ как элемент подготовки и принятия решений.

Использование термина "поддержка и принятие решений" в качестве сферы применения ХД существенно сужает как определение, так и возможность применения концепции в других сферах. Если в определении в качестве области применения оставить лишь анализ и воспроизводство новых данных (как элемент обработки информации в научных, технологических и экологических системах), круг использования данной концепции может быть значительно расширен. Таким образом, можно дать и такое определение [11]:

ХД есть организация и поддержка предметно-ориентированной, интегрированной, слабо изменяемой по внутренней структуре и поддерживающей хронологию электронной коллекции данных для обработки с целью извлечения новых данных или обобщения имеющихся.

Очень важен основной принцип действия ХД: единожды занесенные в ХД данные затем многократно извлекаются из него и используются для анализа. Отсюда вытекает одно из основных преимуществ использования этой технологии: контроль информации, полученной из различных источников, предварительно согласованной и размещенной в ХД. Отметим, что отсюда следует и наиболее уязвимое место ХД — корректность его данных, полученных из разных источников. Данные перед загрузкой должны быть либо "очищены от шума", либо обработаны методами нечеткой логики, допускающей наличие противоречивых фактов, чтобы противоречия в данных были по возможности устранены. Заметим также, что интеграция в определении ХД понимается не только как интеграция информации по всем источникам, но и в смысле согласованного представления данных из разных источников по их типу, размерности и содержательному описанию.

С точки зрения применения концепции в бизнесе, производстве и технологиях следует придерживаться следующего определения [12]:

ХД — структурно расширяемая вычислительная среда, спроектированная для анализа неизменяемых во времени данных, которые логически и физически преобразованы из различных источников, соответствующая направлениям бизнеса, обновляемая и поддерживаемая длительный период времени, выраженная в простых бизнес-терминах и обобщенная (суммированная) для быстрого анализа.

На практике для реализации ХД используются СУБД, поддерживающие определенную модель данных. Поэтому с точки зрения реализации ХД следует считать БД специальной структуры. Предметом настоящей книги является изучение вопросов, связанных с проектированием реляционных ХД.

4. Типы хранилищ данных

Концепция ХД развивалась *по* мере расширения сферы применения. Вначале под ХД понимался набор предметно-ориентированных, интегрированных, не меняющихся во времени исторических данных, предназначенных для *принятия решений* руководством.

Потом стало очевидным, что ХД обладают определенной внутренней структурой. Они содержат базовые данные, которые образуют единый источник для обработки данных во всех системах поддержки *принятия решений* (*DSS*). С помощью ХД можно выполнить согласование данных, несмотря на разногласие данных-источников. А элементарные данные, присутствующие в ХД, могут быть представлены в различной форме, отвечая не только известным требованиям, но и еще неизвестным.

ХД обычно имеют очень большой объем данных, поскольку в них содержатся исторические и детализированные данные, от нескольких терабайт и больше. *По* частоте использования данные в ХД подразделяются на два класса: активно и неактивно используемые данные. Большой объем неактивно используемых данных может значительно снизить *производительность* обработки запросов к ХД.

ХД содержат интегрированные данные. Они интегрированы на множестве уровней: на уровне ключа, атрибута, на описательном, структурном уровне и так далее. Общие данные и общая обработка данных консолидированы и являются единообразными для всех данных, которые обладают структурным сходством.

Несмотря на то, что указанные выше характеристики являются общими для всех ХД, в настоящее время довольно трудно типизировать и классифицировать всевозможные ХД. Можно предложить некоторую классификацию ХД в зависимости от характеристик *предметной области*, которые придают ХД индивидуальные особенности. Классификация архитектурных программно-аппаратных решений будет дана в следующей лекции.

Далее будут кратко описаны типы ХД *по* Б. Инмону. В основу его классификации положен отраслевой принцип применения ХД.

Финансовые хранилища данных

В большинстве случаев финансовые ХД организации строят в первую очередь. Создание **финансового ХД** — необходимый компонент финансовой инфраструктуры любой организации.

- Финансовые данные всегда находятся в центре внимания руководства организации. Поэтому привлечь интерес к созданию такой информационной системы данных очень легко.
- Финансовая активность большинства организаций (за исключением финансово-кредитных учреждений) невелика, поэтому объемы финансовых данных не очень большие, скорость поступления данных также невелика. Финансовые данные хорошо структурированы. Поэтому имеющиеся программно-аппаратные средства позволяют создать и поддерживать компактные финансовые ХД.
- Финансы охватывают все аспекты функционирования организации и имеют один общий знаменатель — деньги.
- Финансовые данные по своей природе имеют структуру, на которую напрямую влияет повседневная практика обработки финансовой информации.

По этим причинам финансы становятся самой предпочтительной областью построения корпоративного ХД. Однако финансовые ХД имеют серьезные, присущие только этому типу проблемы. Первая проблема заключается в следующем. Руководство организации ожидает, что сведения из финансовых ХД будут с точностью до одной копейки совпадать с данными существующей финансовой среды. Ожидание того, что информация в финансовом ХД должна точь-в-точь совпасть с цифрами из текущего

финансового отчета, является глубоко ошибочным. Люди (то есть финансовые работники), которые так думают, просто не понимают, что, когда данные переходят из операционной среды в финансовое ХД, происходит их трансформация. А когда данные перетекают из мира приложений в реальный мир организации, их рассматривают в другом измерении. При такой трансформации происходит следующее.

- Меняются отчетные периоды. В операционной среде отчетный период завершается в конце месяца, а в ХД отчетный период заканчивается на корпоративном календаре, например, 15-го числа месяца.
- Меняются схемы группировки и кодирования счетов. В операционной среде данные рассчитываются в соответствии с планом бухгалтерских счетов, а в финансовой среде всей организации может быть совершенно другой набор схемы группировки и кодирования.
- Меняются классификации данных.
- Меняются валюты. Операционные денежные средства соответствуют той валюте, в которой они обращаются: рубли, доллары, евро, фунты и так далее. В глобальной среде деньги преобразуются к одной общей валюте.

Как видно, существует много причин, почему данные, находящиеся в ХД, отличаются от данных операционных систем. Финансовые работники думают иначе, и поэтому необходимо им разъяснять, что такое трансформация и что означают различные измерения данных.

Хранилища данных в области страхования

ХД в области страхования, за некоторыми небольшими исключениями, похожи на все другие. Первое исключение (характерное для западных компаний) заключается в том, что продолжительность существования имеющихся ХД очень велика. Такие ХД содержат данные, которые являются очень старыми (до начала XX века)

Второе отличие этих ХД определяется датами, сведения о которых хранятся в этой сфере деятельности. Среда страхования — по каким бы то ни было причинам — отличается наличием огромного числа дат, связанных с бизнесом, большим, чем в каком-либо другом виде деятельности. Так, в сфере розничной торговли имеется несколько важных дат: дата продажи, дата появления на складе, возможно, дата производства. В банковском деле существенна дата транзакции. В телекоммуникации — дата телефонного звонка. В страховании же присутствуют даты всевозможных типов.

Наконец, третье отличие заключается в том, что эти ХД используют свой *рабочий цикл* деловой активности. Большинство организаций имеет весьма ограниченный и короткий экономический цикл. Так, в банках это — обналичивание чека. В торговле — покупка изделия. В телефонной компании — звонок. В страховании им может быть заявка на страховое возмещение, которая может быть удовлетворена спустя пять лет, или закрытие полиса может сопровождаться двухмесячной отсрочкой. В итоге скорость, с которой функционирует страхование, отличается от скорости, характерной для других отраслей.

Эта разница в скорости отражается в ХД, как в зеркале. В других ХД транзакции просто собираются и обрабатываются. В области страхования транзакция может откладываться на неопределенный срок, а ее различные части могут отражаться в ХД, т.е. существенным становится не только сама операция, но и ее состояние. Результатом этого является совершенно особый подход при проектировании и внедрении таких ХД.

Хранилища данных для управления персоналом

ХД для управления людскими ресурсами имеют весьма существенные отличия от других ХД. Первое отличие — число предметных областей. Такое ХД неизбежно имеет одну важную предметную область — это работник. Практически все остальное подчинено этой области или занимает второстепенное положение. Большинство же других ХД имеют несколько базовых предметных областей.

Основное отличие ХД для управления людскими ресурсами состоит в том, что они используют очень мало транзакций. Так, имеется дата, когда субъект становится работником; дата, когда человек увольняется; годовые прибавки и повышения. Но, кроме транзакций фонда заработной платы и прочих редких, сгенерированных работником, транзакций, в таком ХД практически больше ничего и нет. Сравните сферу управления людскими ресурсами с коммуникацией или банковской средой, и разница в числе транзакций станет очевидной.

Эта разница в числе транзакций является причиной возникновения определенной сложности, которая заключается в том, что в области управления человеческими ресурсами наблюдается тенденция к объединению операционной обработки людских ресурсов и обработки людских ресурсов для систем принятия решения в одну среду. В других же отраслях соблазн совершить такую архитектурную ошибку весьма невелик.

Глобальные хранилища данных

Глобальные хранилища данных предназначены для глобального представления деятельности организации. Различают три типа таких ХД.

- Географически превалирующая обработка данных. Например, необходимо интегрировать бизнес в Гонконге с бизнесом в Париже, который, в свою очередь, следует интегрировать с Москвой, а тот — с Владивостоком.
- Функционально превалирующая обработка данных. Производственная деятельность должна быть интегрирована с поставками, которые необходимо интегрировать с продажами, а те — с исследованиями и так далее.
- Отраслевая превалирующая обработка данных. Например, требуется интегрировать печатное дело с консалтингом, который подлежит интеграции с бизнесом в сфере медицинского оборудования, а тот — со специализацией в области программного обеспечения.

Особенность глобального ХД заключается в том, что на глобальном уровне зачастую очень мало общих измерений. Единственное общее измерение — это деньги. И интеграция бизнеса может быть достигнута только с его помощью. Другие же измерения могут иметь или не иметь смысл на глобальном уровне. Так, клиент, продукт, поставщик, транзакция — все эти классические предметные области могут как присутствовать, так и отсутствовать в глобальной интегрированной сфере — глобальном ХД.

Помимо этого, глобальное ХД должно непрерывно реагировать на возможные изменения в бизнес-данных. В этом случае такие изменения, как правило, носят постоянный характер. Поэтому структура и технология, используемая для размещения и обслуживания глобального ХД, должна позволять поддерживать эти непрерывные перемены.

Хранилища данных с возможностями обнаружения новых данных (Data Mining)

ХД, поддерживающие технологию обнаружения новых данных (Data Mining), являются гибридом классических ХД. Они используются для выполнения мощной статистической обработки данных. Эти ХД являются:

- очень детальными;
- глубоко историческими;
- оптимизированными для статистического анализа.

Кроме того, для таких ХД характерна ориентация на какой-либо проект. Это означает, что, в отличие от всех других типов ХД, в большинстве случаев их перестают использовать сразу по завершении анализа, ради которого они создавались.

Еще одно важное отличие ХД с возможностями анализа заключается в том, что оно очень часто включает внешние данные. Такие данные очень полезны с точки зрения прогнозирования изменения бизнес-данных, которое не так легко увидеть без их участия.

Хранилища данных в области телекоммуникаций

Отличительная особенность этих ХД состоит в том, что они в значительной степени определяются данными, касающимися факта телефонных разговоров. Разумеется, в отрасли телекоммуникации присутствует множество других типов данных. Но ни одна другая область ХД не предопределяется в такой степени размером одной предметной области — деталями на уровне разговора.

Существуют несколько способов хранения подробностей на уровне телефонного разговора:

- хранение деталей на уровне разговора только за несколько месяцев;
- хранение множества деталей на уровне разговора, размещенных на различных носителях;
- резюмирование или агрегирование деталей на уровне разговора;
- хранение только отобранных деталей на уровне разговора, и так далее.

К сожалению, несмотря на разнообразие методов обработки, для данного ХД обработка может быть выполнена только над деталями на уровне разговора. А работа на итоговом или агрегированном уровне просто невозможна.

Характерные особенности различных типов ХД и доводы в пользу их внедрения

ХД — это логически интегрированный источник данных для систем принятия решений, информационных систем руководителей, *систем анализа данных* и систем обнаружения новых данных (Data Mining). ХД предназначено для информационной поддержки анализа данных, принятия решений, т.е. информационной поддержки деятельности, а не собственно поддержки каждодневных бизнес-процедур организации, и поэтому многие принципы технологии БД утрачивают в ХД свое значение.

ХД ориентируется на определенную предметную область и организуется так, чтобы решать конкретные задачи анализа и информационной поддержки деятельности организации. Данные различных источников агрегируются ХД, приобретая при этом статус неизменчивых. Для ХД характерно массовое добавление данных и фактическое отсутствие операций обновления. Процесс пополнения данных включает в себя сложные процедуры *очистки данных*: устранения несоответствия типов, размеров и других свойств данных.

Основные отличия различных типов ХД состоят в следующем.

- Данные финансовых ХД — а именно их обычно создают в первую очередь — не будут с точностью до одной копейки совпадать с информацией в существующей финансовой среде.
- ХД в области страхования отличаются от других продолжительностью существования, а также разнообразием дат и продолжительностью экономического цикла.
- Для ХД управления человеческими ресурсами характерна только одна основная предметная область.
- ХД с возможностями обнаружения новых данных (Data Mining) и исследования данных (Exploration *Data Warehouse*), которые используются для выполнения мощной статистической обработки данных, являются гибридом классических ХД.
- Отличительная особенность ХД в области телекоммуникаций состоит в том, что они в значительной степени определяются данными, сгенерированными в одной предметной области.

Вне всяких сомнений, будут появляться и другие типы ХД, каждому из которых присущи свои отличительные особенности.

Технология хранения данных обеспечивает адекватную основу для информационной поддержки деятельности руководителей организаций в области принятия решений и дает преимущества в тех областях деятельности, которые связаны с управлением и использованием долговременно хранимой информации, а именно:

- организация получает взгляд на данные как на единое целое. Например, это дает ответы на такие вопросы, как:
  - Сколько продуктов реально производится?
  - Что влияет на изменение спроса?
  - Какие товары или услуги приносят наибольший доход?
  - Каковы особенности и предпочтения ваших клиентов?
- среднее суммарное значение за три года возврата инвестиций, вложенных в создание ХД, составляет 400% (по результатам трехлетнего исследования опыта 62 корпораций, проведенного компанией IDC). При этом средняя стоимость организации *хранилища данных* составляет 2,2 млн долл., а среднее время самоокупаемости - 2,3 года. По оценкам, 90% участвовавших в этом проекте компаний получили более 40% прибыли на инвестированный капитал, а у 50% этот показатель составил более 160%;
- возрастает надежность данных для принятия решений. Данные, загружаемые в ХД, подвергаются очистке - согласуются, проверяются, уточняются;
- появляется возможность эффективного геопространственного анализа данных. Анализ такой информации имеет решающее значение в принятии решений по всем вопросам, связанным с географией бизнеса;
- исследование трендов и колебаний в бизнес-данных с помощью ХД позволяет достаточно надежно прогнозировать развитие бизнес-процессов организации во времени.

Резюме

Концепция ХД была предложена в начале 90-х годов прошлого столетия как основа методологии организации данных в системах поддержки и *принятия решений*. Согласно классическому определению В. Инмона, *хранилище данных* есть предметно-

ориентированная, интегрированная, неизменяемая и поддерживающая хронологию электронная коллекция данных для обеспечения процесса *принятия решений*.

Данные поступают в ХД из внешних источников. Методика построения ХД предполагает выполнение ряда процедур преобразования и *очистки данных* внешних источников.

Использование концепции ХД предполагает использование иных, чем в операционных системах обработки данных, методов построения модели данных.

Таким образом, в ХД хранятся:

- данные масштаба организации;
- интегрированные наборы исторических данных из различных источников данных;
- предметно-ориентированные, согласованные и консолидированные данные;
- данные, структурированные с целью упростить выполнение запросов.

Использование информационных технологий на основе ХД предполагает применение систематизированного позадачного подхода. ХД создается для решения конкретных, строго определенных задач анализа и воспроизводства данных. Таким образом, определяющим моментом в его построении являются задачи обработки данных. Именно это обстоятельство определяет и подходы к проектированию ХД.

На практике для реализации ХД используются *СУБД*, поддерживающие определенную *модель данных*. Поэтому с точки зрения реализации ХД следует считать *БД* специальной структуры.

Каталог данных — это набор метаданных вместе с инструментами управления данными и средствами поиска, который помогает пользователям находить нужную информацию. Он предоставляет информацию для оценки пригодности данных в использовании, обеспечивая классификацию информации и быстрый доступ к ней. Каталог данных в DWH помогает искать нужные и просматривать все доступные наборы данных, а также оценивать и анализировать их.

Каталог данных позволяет понять представления данных, хранящихся в источниках данных, с технической и бизнесовой точек зрения. Автоматическое обнаружение наборов данных требуется не только для постоянного обнаружения новых наборов, но и для первоначального построения каталога. Кроме того, каталог данных позволяет узнать, кто является владельцем данных в DWH и кто отвечает за их отправку в хранилище.

Для этого каталог данных должен реализовывать следующие функции:

- представлять собой единый источник достоверной информации обо всех загруженных корпоративных данных;
- быть способным обнаруживать новые источники данных и следить за тем, как они повышают ценность бизнеса;
- вести мониторинг данных и обеспечивать помощь в создании линии их передачи;
- иметь элементы управления для обеспечения доступа и безопасности на основе ролей;
- отмечать проблемы с качеством данных;
- иметь возможность интеграции с внешними инструментами и технологиями;
- предоставлять доступ к данным для анализа и передачи к месту назначения.

Таким образом, каталог данных помогает повысить их эффективность, предоставляя полный контекст, повышая скорость поиска информации и ее надежность. Это позволяет бизнесу принимать прозрачные и комплексные решения, основанные на данных согласно data-driven управлению. Наличие каталога данных улучшает сотрудничество между техническими специалистами и бизнес-группами.

Чтобы реализовать эффективный каталог данных в DWH, следует учитывать, кто является его конечным пользователем, какой должна быть стратегия развертывания, каковы особенности рабочих процессов и сценариев использования корпоративных витрин и хранилищ. Сегодня на рынке есть набор готовых решений, которые можно применить для создания каталога данных в Data Warehouse. Самыми популярными из них считаются следующие:

- *Aginity (coginiti)* – имеет отличную экосистему и поддерживает SQL, каталогизирует все данные компании и математику для аналитических вычислений;
- *Apache Atlas* — предоставляет открытые возможности управления метаданными для создания каталога активов данных, классификации и управления ими, а также средства совместной работы;
- *world* — облачная платформа, которая сочетает наглядный GUI с мощным графом знаний для обеспечения расширенного обнаружения данных и управления ими;
- *LinkedIn DataHub* — платформа управления метаданными с открытым исходным кодом для современного стека данных, которая обеспечивает обнаружение данных, их наблюдаемость и федеративное управление;
- *Alation* — корпоративный каталог данных, который повышает производительность и точность аналитики, позволяя быстро находить, понимать и управлять нужными данными;
- *Collibra* – помогает унифицировать данные отдельных команд, людей, компаний и систем благодаря отличным возможностям каталогизации со настройками управления и конфиденциальности;
- *каталог данных AWS Glue* – постоянное хранилище технических метаданных от Amazon, управляемый сервис, который можно использовать для хранения, комментирования и обмена метаданными в облаке AWS;
- *каталог данных Azure* — полностью управляемая облачная служба, которая позволяет пользователям находить нужные им источники данных и анализировать их;
- *каталог Google Dataplex* —полностью управляемая, масштабируемая служба управления метаданными в рамках GCP.

Узнайте больше подробностей по проектированию и поддержке современных дата-архитектур в проектах аналитики больших данных на специализированных курсах в нашем лицензированном учебном центре обучения и повышения квалификации для разработчиков, менеджеров, архитекторов, инженеров, администраторов, Data Scientist’ов и аналитиков Big Data в Москве: