

АЛИМОВА Ф.М., КУШМАНОВА М.А., НАИМ Н.А.

BIG DATA



**BIG
DATA**



Kitobni ko'rsatilgan sanadan
kechiktirmay topshiring

--	--

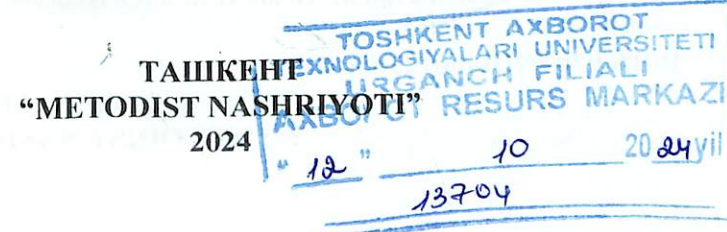
МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ, НАУКИ И
ИННОВАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ ИМЕНИ МУХАММАДА АЛ-ХОРАЗМИЙ

АЛИМОВА Ф.М., КУШМАНОВА М.А., НАИМ Н.А.

BIG DATA

*Рекомендовано в качестве учебника Министерством высшего
образования, науки и инноваций Республики Узбекистан*



УДК: 004.6(075.8)

ББК: 32.973я7

A 50

Алимова Ф.М.

Big Data/ Кушманова М.А., Наим Н.А./ Учебное пособие.

– Ташкент: “METHODIST NASHRIYOTI”, 2024. – 152 стр.

Big data nomli ўқув қўлланмада тартибланган ва тартибланмаган катта ҳажмдаги маълумотларнинг асосий тамойиллари, концепцияси ва методологияси келтирилган. Ўқув қўлланмада кўп ўлчамли OLAP, ROLAP, MOLAP, HOLAP тизимлар, уларнинг архитектуралари ҳақида маълумотлар келтирилган. Шунингдек, қарор қабул қилиш тизимларининг асосий тушунчалари, маълумотларини саралаш ва сақлаш қўриб чиқилган. Материалларни янада чуқурроқ ўрганиш, шунингдек талабалар билимларини баҳолаш учун назорат ва тест саволлари келтирилган. Ушбу ўқув қўлланма IT-service йўналишининг 4 курс талабалари учун мўлжалланган.

В учебнике “Big data” изложены основные принципы, концепции и методологии упорядоченных и неупорядоченных больших объемов данных. Учебник содержит информацию о многомерных системах OLAP, ROLAP, MOLAP, HOLAP, и их архитектуры. Здесь также предоставляется информация об основных концепциях систем принятия решений, сортировки и хранения данных. Для оценки знаний студентов предусмотрено углубленное изучение материалов, а также контрольные и тестовые вопросы. Учебник рассчитан для студентов 4 курса “IT-service”.

The Big Data tutorial outlines the basic principles, concepts and methodologies of ordered and disordered large amounts of data. The tutorial contains information about multidimensional systems OLAP, ROLAP, MOLAP, HOLAP, and architecture. It also provides information on the basic concepts of decision-making, sorting and data storage systems. To assess the knowledge of students, an in-depth study of materials is provided, as well as control and test questions. The textbook is preferred for 4th year students “IT-service”.

Учебное пособие для студентов по направлению 5330500-ИТ-сервис

Рецензенты:

Каримова В.А. - кандидат технических наук, доцент

Якубов М.С. - доктор технических наук, профессор

Публикация разрешена на основании приказа Министерства высшего и среднего специального образования Республики Узбекистан от 18 августа 2021 года № 356.

ISBN 978-9910-03-125-0

© Алимова Ф.М и другие., 2024.
© “METHODIST NASHRIYOTI”, 2024.

Введение

В современных условиях ведения бизнеса компаниям недостаточно уметь накапливать информацию о собственной операционной деятельности. Чтобы добиться действенных конкурентных преимуществ, компании вынуждены заниматься оперативным и глубоким анализом и прогнозированием, основанном на данных, хранящихся в разных информационных системах, в разных форматах не только в самой компании, но и вне ее. Технологическим решением этой проблемы является использование аналитических систем, сочетающих технологии хранилищ данных, оперативного анализа данных, интеллектуального анализа данных и современных технологий визуализации.

С целью обеспечения ускоренного цифрового развития Республики Узбекистан, формирования цифровой экономики, основанной на данных, путем создания необходимой среды для производства инновационной продукции, повышения эффективности государственного управления, оказания населению и субъектам предпринимательства соответствующих государственных услуг принят Указ Президента Республики Узбекистан «Об утверждении стратегии «Цифровой Узбекистан – 2030» и мерах по ее эффективной реализации» (№УП-6079, 05.10.2020).

«Большие данные», в течение уже более пятнадцати лет, стало словосочетанием, которое употребляется во многих выступлениях, как на крупных IT конференциях, так и фразой, которую можно услышать на различных экономических форумах. При этом отсутствует четкое толкование данного термина, как в научном сообществе, так и у тех, кто использует данную концепцию в своих целях. Не смотря на наличие в словосочетании термина «данные», отсутствует описание и классификация того, какие объемы информации будут относиться, а какие нет к данной категории.

Можно трактовать, что Big Data или большие данные — это структурированные или неструктурированные массивы данных большого объема. Большие данные помогают

анализировать текущее состояние бизнеса, строить прогнозы и автоматизировать рутинные процессы. Для работы с ними используют специальные технологии, которые позволяют быстро обрабатывать огромные массивы информации и извлекать из них пользу. Считается, что эти «умения» позволяют выявить скрытые закономерности, ускользающие от ограниченного человеческого восприятия. Это дает беспрецедентные возможности оптимизации многих сфер нашей жизни: государственного управления, медицины, телекоммуникаций, финансов, транспорта, производства и так далее.

Согласно анализу компании Gartner Group, большинство бизнес-пользователей организации в своей повседневной деятельности используют средства бизнес-аналитики; более половины крупных организаций по всему миру используют данные средства в качестве конкурентного преимущества.

Настоящее учебное пособие представляет возможность ознакомиться с основными понятиями в области аналитической обработки больших данных. Главная цель данного учебного пособия является ознакомление студентов с современными подходами к анализу «больших данных». Полученные знания и навыки по «большим данным» позволят студентам повысить эффективность принятия решений за счет грамотного сбора, структурирования и применения современных техник анализа больших объемов данных таких как Data – Information – Knowledge – Wisdom: от данных - к мудрости в принятии решений.

В учебном пособии излагаются основы сбора и обработки больших данных и интеллектуального анализа данных, современных хранилищ данных, читатель познакомится с технологиями аналитической обработки в реальном времени (OLAP).

Учебное пособие предназначено для студентов 4 курса бакалавриата, обучающихся по направлениям «IT сервис» и включает 8 глав:

- в главе 1 представлен базовый материал по технологиям больших данных, таких как определение и характеристика термина больших данных, история использования данных, метаданные и их классификации, элементы управления

метаданными, компоненты метаданных аналитической системы. А также коротко дается информация об имеющихся технологиях big data, методов и анализа обработки больших данных;

- в главе 2 представлена информация об основах системы поддержки принятия решений, о многомерных хранилищах данных и их комбинации, о системах аналитической обработки данных OLAP и их характеристики, а также говорится о транзакционных системах обработки транзакции;

- в главе 3 даны основы многомерной модели данных, архитектура хранилищ данных, а также серверы использования больших данных;

- в главе 4 представлены основы логической модели хранилищ данных, характеристики хранилищ данных, а также модели хранилищ данных по организации программной среды для практикума;

- в главе 5 представлены основы физической модели хранилищ данных, объекты физической модели данных, а также иерархия и основы объектов реляционной базы данных;

- в главе 6 содержится описание примеров использования средств бизнес-аналитики для анализа информации и поддержки принятия с помощью современных технологий;

- глава 7 посвящена методам интеллектуального анализа данных.

- в главе 8 описываются методология и процесс обнаружения в больших массивах данных, накапливающихся в информационных системах компаний, ранее неизвестных, необходимых для принятия решений в различных сферах человеческой деятельности.

В конце учебного пособия приводятся тесты, глоссарий и список использованной литературы.

ГЛАВА I. ВВЕДЕНИЕ В АНАЛИЗ БОЛЬШИХ ДАННЫХ

1.1. Определение основных понятий

Большие данные (англ. *Big data*) в информационных технологиях — совокупность подходов, инструментов и методов обработки структурированных и неструктурированных данных огромных объемов и значительного многообразия для получения воспринимаемых человеком результатов, эффективных в условиях непрерывного прироста, распределения по многочисленным узлам вычислительной сети, сформировавшихся в конце 2000-х годов, альтернативных традиционным системам управления базами данных и решениям класса Business Intelligence.

Под Big Data понимается методы обработки данных, которые позволяют распределено обрабатывать информацию. Данные методы применяются как к огромным массивам данных, так и к маленьким.

Большие данные — это совокупность технологий, которые призваны совершать следующие три операции:

1. Обработать большие по сравнению со «стандартными» сценариями объемы данных.

2. Уметь работать с быстро поступающими данными в очень больших объемах. То есть данных не просто много, а их постоянно становится все больше и больше.

3. Уметь работать со структурированными и слабо структурированными данными параллельно и в разных аспектах.

Использование больших данных позволяет раскрыть скрытые закономерности, ускользающие от ограниченного человеческого восприятия. Инструменты аналитики больших данных дают возможность оптимизации многих сфер нашей жизни: производства, медицины, транспорта, финансов, телекоммуникаций, государственного управления и так далее.

Три "V" (4, 5, 7) и три принципа работы с большими данными

В качестве определяющих характеристик для больших данных отмечают «три V»:

- объем (англ. *volume*, величина физического объема.);
- скорость (англ. *velocity* скорость прироста и необходимости быстрой обработки данных для получения результатов);
- многообразие (англ. *variety*, возможность одновременно обрабатывать различные типы данных).

Набор признаков VVV был выработан компанией MetaGroup в 2001 году с целью указать на равную значимость управления данными по всем трем аспектам. В дальнейшем появились интерпретации с четырьмя V (добавлялась *veracity* — достоверность), пятью V (*viability* — жизнеспособность и *value* — ценность), семью V (*variability* — переменчивость и *visualization* — визуализация).

Исходя из вышеприведенных определений, основные принципы работы с большими данными таковы:

1. **Горизонтальная масштабируемость.** Это — базовый принцип обработки больших данных. Как уже говорилось, больших данных с каждым днем становится все больше. Соответственно, необходимо увеличивать количество вычислительных узлов, по которым распределяются эти данные, причем обработка должна происходить без ухудшения производительности.

2. **Отказоустойчивость.** Этот принцип вытекает из предыдущего. Поскольку вычислительных узлов в кластере может быть много (иногда десятки тысяч) и их количество, не исключено, будет увеличиваться, возрастает и вероятность выхода машин из строя. Методы работы с большими данными должны учитывать возможность таких ситуаций и предусматривать превентивные меры.

3. **Локальность данных.** Так как данные распределены по большому количеству вычислительных узлов, то, если они физически находятся на одном сервере, а обрабатываются на другом, расходы на передачу данных могут стать неоправданно большими. Поэтому обработку данных желательно проводить на той же машине, на которой они хранятся.

Эти принципы отличаются от тех, которые характерны для традиционных, централизованных, вертикальных моделей хранения хорошо структурированных данных. Соответственно, для работы с большими данными разрабатывают новые подходы и технологии.

Метаданные

Понятие метаданных при сборе данных возникают метаданные, содержащие какую-либо информацию о собранных данных. Например, время создания набора данных, авторство и первоисточник, размер и кодировка данных — все это метаданные. Структурированные метаданные называют онтологией или схемой метаданных. Она включает машинно-интерпретируемые формулировки основных понятий предметной области и отношения между ними. Онтологии обычно используются в таких областях, как искусственный интеллект, семантическая паутина, системная инженерия, биомедицинская информатика, библиотечное дело, информационная архитектура и др. Все онтологии нужны для организации информации.

Метаданные можно классифицировать по

- Содержанию. Метаданные могут либо описывать сам ресурс (например, название и размер файла), либо содержимое ресурса (например, «в этом видеофайле показано как парень играет в футбол»).
- По отношению к ресурсу в целом. Метаданные могут относиться к ресурсу в целом или к его частям. Например, «Title» (название фильма) относится к фильму в целом, а «Scenedescription» (описание эпизода фильма) — отдельное для каждого эпизода фильма.
- По возможности логического вывода. Метаданные можно подразделить на три слоя: нижний слой — это «сырые» данные сами по себе; средний слой — метаданные, описывающие указанные «сырые» данные; и верхний слой — метаданные, которые позволяют делать логический вывод, используя второй слой.

Три наиболее используемых класса метаданных являются:

- Внутренние метаданные, описывающие структуру или составные части вещи, то, чем вещь является. Например, формат и размер файла.

- Административные метаданные, требующиеся для процессов обработки информации, назначение вещи. Например, информация об авторе, редакторе, дата публикации и т. п.

- Описательные метаданные, которые описывают природу вещи, её признаки. Например, набор связанных с информацией категорий, ссылки на другие вещи, связанные с данной.

Управление метаданными — процесс управления данными, связанными с информационными активами организации по обеспечению интеграции, доступа, совместного использования, анализа и пр. для достижения наилучших результатов организации.

Элементы управления метаданными

- Meta data repositories (Репозиторий метаданных) — используется для хранения, документирования, анализа и управления метаданными. Содержит в себе все данные на протяжении всех этапов жизненного цикла проекта: разработка, тестирование, ввод в промышленную эксплуатацию.

- Business glossary (Бизнес словарь) — используется для управления бизнес-терминами наряду с соответствующими определениями и отношениями между этими терминами.

- Datalineage (Происхождение данных) — определяет происхождение данных, этапы преобразования и изменения данных, направление их движения.

- Impact analysis (Анализ влияния) — предоставление обширной информации относительно зависимостей между данными с последующим влиянием на целевые показатели.

- Rules management (Управление правилами) — автоматизация соблюдения бизнес-правил, которые привязаны к элементам данных и связанным с ними метаданным.

- Semantic frameworks (Семантический фреймворк) — поддержка таксономии, ER-модели, онтологии, моделирования языков таких как RDF, OWL и UML.

- Meta data ingestion and translation (Захват и передача метаданных) — технологии и коннекторы для различных источников данных: RDBMS, BI, DIS, XML, etc.

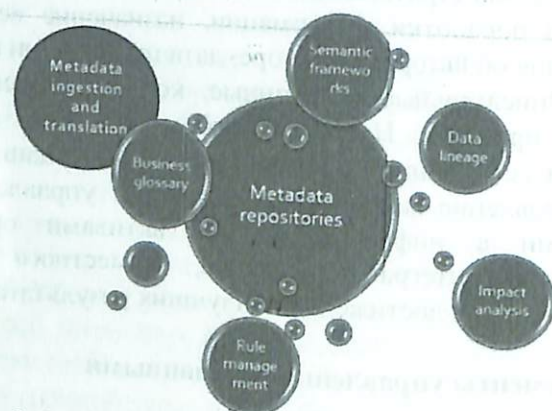


Рис1.1. Элементы управления метаданными

Примеры метаданных

В данной главе приведем примеры наиболее часто используемых метаданных.

В технологиях баз данных:

- концептуальные схемы предметных областей;
- схемы баз данных;

Описания междууровневых отображений схем в системах баз данных

В технологиях интеграции данных:

- локальные схемы источников данных;
- глобальные схемы;
- описания отображений между локальными схемами интегрируемых источников и глобальной схемой;
- онтологии локальных источников и общей онтологии;
- системы интеграции данных;

Описания отображений между онтологиями локальных источников и общей онтологией характеристики;

В технологиях текстового поиска:

- регистрации источников в посредниках, в системах виртуальной интеграции данных;
- идентификаторы текстовых документов;
- наборы значений индексирующих атрибутов документов;

поиска;

- библиографические описания документов;
- аннотации публикаций;
- каталоги коллекций документов;
- наборы ключевых слов документов;
- рубрики классификаторов для документов;
- наборы значений элементов метаданных DC;
- индексы УДК;
- индексы ISBN монографий;

В CASE-технологиях:

- UML-диаграммы проектов разрабатываемых систем;
- диаграммы IDEF;
- ER-диаграммы;

В веб-технологиях:

- гипертекстовая разметка веб-страниц;
- наборы имен и значений параметров тегов META в веб-страницах;

разметка фрагментов веб-страниц средствами микроформатов (hCard, hReview, hProduct, hRecipe и др.)

- описания типов XML-документов (DTD);
- XML-схемы для типов XML-документов;
- RDF-спецификации ресурсов;
- описания онтологий на языке OWL или OWL2;
- семантические аннотации веб-страниц или их фрагментов;

В технологии веб-сервисов:

- описания интерфейсов веб-сервисов средствами языка WSDL;

описание характеристик веб-сервисов в регистре UDDI.

Компоненты метаданных, в частности, характеризуют:

- *источники информации* в информационном хранилище (происхождение и структура системы записей);
- *преобразование данных* при передаче первоначальных данных из оперативных источников в информационное хранилище;
- *текущие описания данных* в информационном хранилище;

• *предысторию* изменения существующих данных в хранилище.



Рис. 1.2. Компоненты метаданных аналитической системы

Некоторые из представленных на рис. 1.2. компонентов являются обязательными атрибутами метаданных, другие - необязательными, но также достаточно важными. Метрика (функция расстояния между данными), или упорядочение данных, позволяет определить допустимые операции над занесенной в информационное хранилище информацией (суммирование, ранжирование, иерархический порядок и т.п.). Синонимы (ссылочные данные) позволяют установить ссылки на данные, например, при использовании их под разными именами несколькими пользователями. Модель данных служит краеугольным камнем концепции метаданных и определяет пути возможного использования информации из информационного хранилища. Регламент изменения данных обуславливает схему пополнения системы из внутренних и внешних оперативных источников.

Такая схема позволяет менеджеру и аналитику опереться на четкую конструкцию, ориентированную на их потребности. Но для этого должно быть правильно выстроено все дерево информационной поддержки бизнеса в организации, включая

оперативные информационные системы, на базе внутрифирменной модели данных. Такая модель данных должна предусматривать интегрированный взгляд на информацию. Отдельные оперативные подсистемы должны обеспечивать удобство использования информации для последующего хранения и представления в информационном хранилище. При этом должна учитываться предметная ориентация информационного хранилища, в том числе на такие существенные для бизнеса субъекты представления, как заказчики, продукты, сделки, поставщики и продавцы.

Именно метаданные служат связующим звеном информационной архитектуры: как показано на рис. 1.3, они существенны для приложений, а также для оперативного и долгосрочного информационного хранилища.



Рис. 1.3. Назначение метаданных в оперативной информационной системе

Как правило, первоисточником являются наследуемые (т.е. возникшие и накопленные до формирования информационного хранилища) базы данных. Их структура, полнота и форма представления данных практически не могут быть изменены, так как информация либо утеряна, либо требуются большие усилия для ее восстановления. Оперативные хранилища более приспособлены

для структуризации и проектирования. В них применяется технология, предусматривающая использование описателей информации. Метаданные предполагают описание системы записей, метрику и др. Важную роль в отношении к составным частям информационной архитектуры играет взаимосвязь различных компонентов метаданных.

Применительно к проблеме управленческого учета модель метаданных должна включать структуризацию плана счетов, ориентированную на управление. В частности, должна быть предусмотрена возможность учета:

- продукции и услуг;
- центров ответственности, центров прибыли и сервис-центров;
- клиентов, включая поставщиков и потребителей продукции;
- рынков сбыта и регионов;
- показателей, предназначенных для управления (планирования, контроля исполнения планов и корректировки принятых решений).

Полный цикл принятия управленческих решений охватывает все сферы современного менеджмента:

- структуризацию данных;
- доставку данных в требуемое место, в требуемое время и в требуемом объеме;
- формирование информационного поля для проведения полномасштабного анализа управленческих индикаторов и возможность оперативной настройки приложений на нестандартные виды анализа;
- проведение анализа информации;
- подготовку рекомендаций для лиц, принимающих решения, в том числе с использованием эвристических и экспертных знаний;
- принятие решения;
- доведение решения до исполнителей;
- контроль за исполнением.

1.2. Технологии и тенденции работы с Big Data

Изначально в совокупность подходов и технологий включались средства массово-параллельной обработки неопределённо структурированных данных, такие как СУБД No SQL, алгоритмы Map Reduce и средства проекта Hadoop. В дальнейшем к технологиям больших данных стали относить и другие решения, обеспечивающие сходные по характеристикам возможности по обработке сверхбольших массивов данных, а также некоторые аппаратные средства.

• **Map Reduce** — модель распределённых параллельных вычислений в компьютерных кластерах, представленная компанией Google. Согласно этой модели приложение разделяется на большое количество одинаковых элементарных заданий, выполняемых на узлах кластера и затем естественным образом сводимых в конечный результат.

• **No SQL** (от англ. Not Only SQL, не только SQL) — общий термин для различных нереляционных баз данных и хранилищ, не обозначает какую-либо одну конкретную технологию или продукт. Обычные реляционные базы данных хорошо подходят для достаточно быстрых и однотипных запросов, а на сложных и гибко построенных запросах, характерных для больших данных, нагрузка превышает разумные пределы и использование СУБД становится неэффективным.

• **Hadoop** — свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов. Считается одной из основополагающих технологий больших данных.

• **R** — язык программирования для статистической обработки данных и работы с графикой. Широко используется для анализа данных и фактически стал стандартом для статистических программ.

• **Аппаратные решения.** Корпорации Teradata, EMC и др. предлагают аппаратно-программные комплексы, предназначенные для обработки больших данных. Эти комплексы поставляются как готовые к установке телекоммуникационные шкафы, содержащие

кластер серверов и управляющее программное обеспечение для массово-параллельной обработки. Сюда также иногда относят аппаратные решения для аналитической обработки в оперативной памяти, в частности, аппаратно-программные комплексы Hana компании SAP и комплекс Exalytics компании Oracle, несмотря на то, что такая обработка изначально не является массово-параллельной, а объёмы оперативной памяти одного узла ограничиваются несколькими терабайтами.⁶ Консалтинговая компания McKinsey, кроме рассматриваемых большинством аналитиков технологий No SQL, Map Reduce, Hadoop, R, включает в контекст применимости для обработки больших данных также технологии Business Intelligence и реляционные системы управления базами данных с поддержкой языка SQL.

Таб.1.3. Технологии Big data

Наименование технологии	Краткое описание
Hadoop	ПО с открытым кодом для обработки больших данных среди различных, параллельно функционирующих серверов
Map Reduce	Архитектура, на основе которой функционирует Hadoop
Scripting Languages/Языки программирования	Языки программирования для обработки данных
Машинное обучение	По для поиска модели, которая наилучшим образом характеризует данные
Визуализация	Отображение аналитических результатов в графическом формате
NLP (Natural Language Processing)	ПО для анализа текста
In-memory analytics /Аналитика внутри памяти	Обработка данных в памяти ПК

1.3. Методы и техники анализа больших данных

Международная консалтинговая компания McKinsey, специализирующаяся на решении задач, связанных со стратегическим управлением, выделяет 11 методов и техник анализа, применимых к большим данным.

- **Методы класса Data Mining** (добыча данных, интеллектуальный анализ данных, глубинный анализ данных) — совокупность методов обнаружения в данных ранее неизвестных, нетривиальных, практически полезных знаний, необходимых для принятия решений. К таким методам, в частности, относятся обучение ассоциативным правилам (association rule learning), классификация (разбиение на категории), кластерный анализ, регрессионный анализ, обнаружение и анализ отклонений и др.

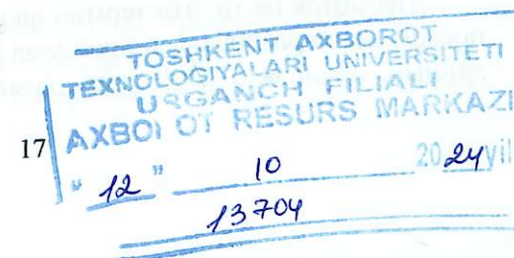
- **Краудсорсинг** — классификация и обогащение данных силами широкого, неопределённого круга лиц, выполняющих эту работу без вступления в трудовые отношения

- **Смешение и интеграция данных** (data fusion and integration) — набор техник, позволяющих интегрировать разнородные данные из разнообразных источников с целью проведения глубинного анализа (например, цифровая обработка сигналов, обработка естественного языка, включая тональный анализ, и др.)

- **Машинное обучение**, включая обучение с учителем и без учителя — использование моделей, построенных на базе статистического анализа или машинного обучения для получения комплексных прогнозов на основе базовых моделей

- **Искусственные нейронные сети**, сетевой анализ, оптимизация, в том числе генетические алгоритмы (genetic algorithm) — эвристические алгоритмы поиска, используемые для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искоемых параметров с использованием механизмов, аналогичных естественному отбору в природе)

- Распознавание образов
- Прогнозная аналитика



- **Имитационное моделирование (simulation)** — метод, позволяющий строить модели, описывающие процессы так, как они проходили бы в действительности. Имитационное моделирование можно рассматривать как разновидность экспериментальных испытаний.

- **Пространственный анализ (spatial analysis)** — класс методов, использующих топологическую, геометрическую и географическую информацию, извлекаемую из данных

- **Статистический анализ** — анализ временных рядов, A/B-тестирование (A/B testing, splittesting — метод маркетингового исследования; при его использовании контрольная группа элементов сравнивается с набором тестовых групп, в которых один или несколько показателей были изменены, для того чтобы выяснить, какие из изменений улучшают целевой показатель)

- **Визуализация аналитических данных** — представление информации в виде рисунков, диаграмм, с использованием интерактивных возможностей и анимации как для получения результатов, так и для использования в качестве исходных данных для дальнейшего анализа. Очень важный этап анализа больших данных, позволяющий представить самые важные результаты анализа в наиболее удобном для восприятия виде.

1.4. Хроника науки о данных

Введение термина «большие данные» относят к Клиффорду Линчу, редактору журнала Nature, подготовившему к 3 сентября 2008 года специальный выпуск с темой «*Как могут повлиять на будущее науки технологии, открывающие возможности работы с большими объемами данных?*», в котором были собраны материалы о феномене взрывного роста объемов и многообразия обрабатываемых данных и технологических перспективах в парадигме вероятного скачка «от количества к качеству»; термин был предложен по аналогии с расхожими в деловой англоязычной среде метафорами «*большая нефть*», «*большая руда*».

Несмотря на то, что термин вводился в академической среде и прежде всего разбиралась проблема роста и многообразия научных данных, начиная с 2009 года термин широко распространился в

деловой прессе, а к 2010 году относят появление первых продуктов и решений, относящихся исключительно и непосредственно к проблеме обработки больших данных. К 2011 году большинство крупнейших поставщиков информационных технологий для организаций в своих деловых стратегиях используют понятие о больших данных, в том числе IBM, Oracle, Microsoft, Hewlett-Packard, EMC, а основные аналитики рынка информационных технологий посвящают концепции выделенные исследования.

В 2011 году Gartner отмечает большие данные как тренд номер два в информационно-технологической инфраструктуре (после виртуализации и как более существенный, чем энергосбережение и мониторинг). Прогнозируется, что внедрение технологий больших данных наибольшее влияние окажет на информационные технологии в производстве, здравоохранении, торговле, государственном управлении, а также в сферах и отраслях, где регистрируются индивидуальные перемещения ресурсов.

С 2013 года большие данные как академический предмет изучаются в появившихся вузовских программах по науке о данных и вычислительным наукам и инженерии.

Архитектура системы обработки Больших данных Для работы с Большими данными используются сложные системы, в которых можно выделить несколько компонентов или слоёв (Layers). Обычно выделяют четыре уровня компонентов таких систем: прием, сбор, анализ данных и представление результатов (рис. 4). Это деление является в значительной мере условным так как, с одной стороны, каждый компонент в свою очередь может быть разделен под компоненты, а с другой некоторые функции компонентов могут перераспределяться в зависимости от решаемой задачи и используемого программного обеспечения, например, выделяют хранение данных в отдельный слой. Для работы с Большими данными разработчиками систем создаются модели данных, содержательно связанные с реальным миром. Разработка адекватных моделей данных представляет собой сложную аналитическую задачу, выполняемую системными архитекторами и аналитиками. Модель данных позволяет создать математическую модель взаимодействий объектов реального мира и включает в себя

описание структуры данных, методы манипуляции данными и аспекты сохранения целостности данных. Описание разработки моделей данных не является задачей настоящего руководства. Для хранения данных используются распределенные системы различных типов. Это могут быть файловые системы, базы данных, журналы, механизмы доступа к общей виртуальной памяти. Большинство систем хранения ориентированы исключительно на работу с Большими данными, они имеют крайне ограниченное число функций (например, может отсутствовать возможность 29 не только модификации, но и удаления поступивших данных) что объясняется внутренней сложностью создания высокоэффективных распределенных систем. В конце текста приведены ссылки на несколько используемых в настоящий момент систем. Для того, чтобы работа с данными происходила быстрее системы хранения и обработки данных распараллеливаются в кластере (cluster, группа компьютеров, объединенных сетью для выполнения единой задачи). Однако, согласно гипотезе Брюе ране возможно обеспечить одновременную согласованность (непротиворечивость) данных, доступность данных и устойчивость системы к отделению отдельных узлов. Гипотеза доказана для транзакций типа ACID (Atomic, Consistent, Isolated, Durable) и известна под названием CAP-теоремы (Consistency, Availability, Partition tolerance).

Прием данных (Data Ingestion) Источники данных имеют различные параметры, такие как частоту поступления данных из источника, объем порции данных, скорость передачи данных, тип поступающих данных и их достоверность. Для эффективного сбора данных необходимо установить источники данных. Это могут быть хранилища данных, поставщики агрегированных данных, API каких-либо датчиков, системные журналы, с генерированный человеком контент в социальных сетях, в корпоративных информационных системах, геофизическая информация, научная информация, унаследованные данные из других систем. Источники данных определяют исходный формат данных. Например, мы можем самостоятельно проводить погодные на территории и аэропорта, использовать данные, поступающие с взлетающих и садящихся самолетов, закупить данные со спутников, пролетающих над аэропортами уместной мете о службы, а также найти их где-то

в сети в другом месте. В общем случае для каждого источника необходимо создавать собственный сборщик (Data Crawler для сбора информации в сети Data Acquisition для проведения измерений). Прием данных заключается в начальной подготовке данных от источников с целью приведения данных к общему формату представления данных. Этот единый формат выбирается в соответствии с принятой моделью данных. Выполняются преобразования систем измерения, типов (типизация), верификация. Обработка данных содержательно не затрагивает имеющуюся в данных информацию, но может изменять ее представление (например, приводить координаты к единой системе координат, а значения к единой размерности).

Сбор данных (Data Staging) Этап сбора данных характеризуется непосредственным взаимодействием с системами хранения данных. Устанавливается точка сбора, в которой собранные данные снабжаются локальными метаданными и помещаются в хранилище либо передаются для последующей обработки. Данные, по каким-либо причинам не прошедшие точку сбора, игнорируются. Для структурированных данных проводится преобразование из исходного формата по заранее заданным алгоритмам. Это наиболее эффективная процедура в случае, если структура данных известна.

Однако если данные представлены в двоичном виде, структура и связи между данными утеряны, то разработка алгоритмов и основанного на них программного обеспечения для обработки данных может оказаться крайне за труднительной. Для полуструктурированных данных требуется интерпретация поступающих данных и использование программного обеспечения, умеющего работать с используемым языком описания данных. Существенным плюсом полуструктурированных данных является то, что в них зачастую содержатся не только сами данные, но метаданные в виде информации о связях между данными и способах их получения. Разработка программного обеспечения для обработки полуструктурированных данных представляет собой достаточно сложную задачу. Однако имеется значительное количество готовых конвертеров, которые могут, например, извлечь данные из формата XML в сформированное табличное

представление. Наибольшего объема работ требует обработка неструктурированных данных. Для их перевода к заданному формату может потребоваться создание специального ПО, сложная ручная обработка, распознавание и выборочный ручной контроль. На этапе сбора проводится контроль типов данных и может выполняться базовый контроль достоверности данных. Например, координаты молекул газа, содержащихся в какой-либо области, не могут лежать за пределами этой области, а скорости — существенно превышать скорость звука. Для того, чтобы избежать ошибок типизации, необходимо проверять правильно ли заданы единицы измерения. Например, в одном наборе данных высота может измеряться в километрах, а в другом — в футах. В этом случае необходимо произвести преобразование высоты в те единицы измерения, которые приняты в используемой модели. При сборе данные систематизируются и снабжаются метаданными, хранимым и в связанных метаданных. При наличии большого количества источников данных может потребоваться управление сбором данных для того, чтобы сбалансировать объемы информации, поступающие из различных источников.

Собранные данные либо сохраняются в системах хранения, либо (в особенности, для потоковых данных) передаются для анализа в реальном времени. Анализ данных (Analysis Layer) Анализ данных, в отличие от сбора данных использует информацию, содержащуюся в самих данных. Анализ может проводиться как в реальном времени, так и в пакетном режиме. Анализ данных составляет основную по трудоемкости задачу при работе с Большими данными. Существует множество методик обработки данных: предиктивный анализ, запросы и отчетность, реконструкция по математической модели, трансляция, аналитическая обработка и другие. Методики используют специфические алгоритмы в зависимости от поставленных целей. Например, аналитическая обработка может являться анализом изображений, социальных сетей, географического местоположения, распознавания по признакам, текстовым анализом, статистической обработкой, анализом голоса, транскрибированием. Алгоритмы анализа данных также, как и алгоритмы обработки данных, опираются на модель данных. При этом при анализе может быть

использовано несколько моделей, задающих общий формат данных, но по-разному моделирующие содержательные процессы, данные о которых мы обрабатываем. При использовании при анализе методов искусственного интеллекта, в частности нейронных сетей, производится динамическое обучение моделей на различных наборах данных. При анализе данных производится идентификация сущностей, описываемых данными на основании имеющейся в данных информации и используемых моделей. Сущностью анализа является аналитических механизм, использующий аналитические алгоритмы, управление моделями и идентификацию сущностей для получения новой содержательной информации, являющейся результатом анализа. Для анализа данных также используются методы искусственного интеллекта на нейронных сетях, не рассматриваемые в данном учебном пособии. Представление результатов (Consumption Layer) Результаты анализа данных предоставляются на уровне потребления. Имеется несколько механизмов, позволяющих использовать результаты анализа больших данных. Мониторинг метаинформации. Подсистема отображения в реальном времени существенных параметров работы системы, загруженности вычислителей, распределение задач в кластере, распределение информации в хранилищах, наличие свободного места в хранилищах, поступление данных от источников, активности пользователей, отказов оборудования и т.д.

1.5. Преимущества использование больших данных

Согласно отчету компании McKinsey «Global Institute, Big data: The next frontier for innovation, competition, and productivity», данные стали таким же важным фактором производства, как трудовые ресурсы и производственные активы. За счет использования больших данных компании могут получать ощутимые конкурентные преимущества. Технологии Big Data могут быть полезными при решении следующих задач:

- прогнозирование рыночной ситуации;
- маркетинг и оптимизация продаж;
- совершенствование продукции;

- принятие управленческих решений;
- повышение производительности труда;
- эффективная логистика;
- мониторинг состояния основных фондов.

На производственных предприятиях большие данные генерируются также вследствие внедрения технологий Промышленного интернета вещей. В ходе этого процесса основные узлы и детали станков и машин снабжаются датчиками, исполнительными устройствами, контроллерами и, иногда, недорогими процессорами, способными производить граничные (туманные) вычисления. В ходе производственного процесса осуществляется постоянный сбор данных и, возможно, их предварительная обработка (например, фильтрация). Аналитические платформы обрабатывают эти массивы информации в режиме реального времени, представляют результаты в наиболее удобном для восприятия виде и сохраняют для дальнейшего использования. На основе анализа полученных данных делаются выводы о состоянии оборудования, эффективности его работы, качестве выпускаемой продукции, необходимости внесения изменений в технологические процессы и т.д.

Благодаря мониторингу информации в режиме реального времени персонал предприятия может:

- сокращать количество простоев;
- повышать производительность оборудования;
- уменьшать расходы на эксплуатацию оборудования;
- предотвращать несчастные случаи.

Последний пункт особенно важен. Например, операторы, работающие на предприятиях нефтехимической промышленности, получают в среднем около 1500 аварийных сообщений в день, то есть более одного сообщения в минуту. Это приводит к повышенной усталости операторов, которым приходится постоянно принимать мгновенные решения о том, как реагировать на тот или иной сигнал. Но аналитическая платформа может отфильтровать второстепенную информацию, и тогда операторы получают возможность сосредоточиться в первую очередь на критических ситуациях. Это позволяет им более эффективно

выявлять и предотвращать аварии и, возможно, несчастные случаи. В результате повышаются уровни надежности производства, промышленной безопасности, готовности технологического оборудования, соответствия нормативным требованиям.

Кроме того, по результатам анализа больших данных можно рассчитывать сроки окупаемости оборудования, перспективы изменения технологических режимов, сокращения или перераспределения обслуживающего персонала — т.е. принимать стратегические решения относительно дальнейшего развития предприятия.

Табл.4. Стратегия применения big data

	Исследования	Выпуск продукции
Снижение издержек	Группа IT специалистов, отвечающих за инновации	Группа IT специалистов, ответственных за операционные процессы
Ускорение принятия решений	Аналитики	Руководители
Улучшение решений	Аналитики	Руководители
Разработка продуктов	R&D	Менеджеры, отвечающие за выпуск продукции

Контрольные вопросы

1. Что означает термин «BigData» в информационных технологиях?
2. Что является основной целью обработки BigData?
3. Кто и в каком году впервые ввел термин «BigData»?
4. Какие главные характеристики BigData?
5. Какие данные занимают больше мировой памяти относительно остальных?
6. Какие понятия содержит в себе принцип трех "V"?
7. С какого года Большие данные изучаются как академический предмет в вузовских программах по науке о данных?
8. Что является примером квази-структурированных данных?
9. Как назывался первый суперкомпьютер, оснащенный вопросно-ответной системой искусственного интеллекта?
10. Чем характеризуются "Большие данные"?

ГЛАВА II. СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ. OLTP И OLAP ТЕХНОЛОГИИ

В области информационных технологий всегда существовало два взаимодополняющих друг друга направления развития:

- системы, ориентированные на оперативную (транзакционную или операционную) обработку данных;
- системы, ориентированные на анализ данных, — системы поддержки принятия решений.

Они предназначены для автоматизации оперативной деятельности предприятий и называются системами On-Line Transaction Processing — (OLTP-системы). Их основная задача — обеспечить регистрацию некоторых фактов, их непродолжительное хранение с последующим размещением в архивах.

Принятие решения есть особый вид деятельности, который состоит в формировании вариантов решения (альтернатив) с последующей оценкой их относительной эффективности и распределением согласно с этим ресурсов между вариантами. Более простыми типами решений есть принятие или отклонение альтернатив, выбор самой лучшей альтернативы, ранжирование альтернатив.

Для принятия обоснованного решения нужно учитывать много (десятки и сотни) факторов, которые сложным образом взаимодействуют между собой. В то же время человек может синхронно оперировать не более чем 7-9 объектами. Для преодоления этого противоречия применяют системы поддержки принятия решений (СППР).

2.1. История создания системы поддержки принятия решений.

Современные системы поддержки принятия решения (СППР), возникшие как естественное развитие и продолжение управленческих информационных систем и систем управления базами данных, представляют собой системы, максимально приспособленные к решению задач повседневной управленческой деятельности, являются инструментом, призванным оказать помощь лицам, принимающим решения (ЛПР).

С помощью СППР могут решаться неструктурированные и слабоструктурированные многокритериальные задачи. СППР, как правило, являются результатом мультидисциплинарного исследования, включающего теории баз данных, искусственного интеллекта, интерактивных компьютерных систем, методов имитационного моделирования.

СППР — в большинстве случаев — это интерактивная автоматизированная система, которая помогает пользователю (ЛПР) использовать данные и модели для идентификации и решения задач и принятия решений. Система должна обладать возможностью работать с интерактивными запросами с достаточно простым для изучения языком запросов.

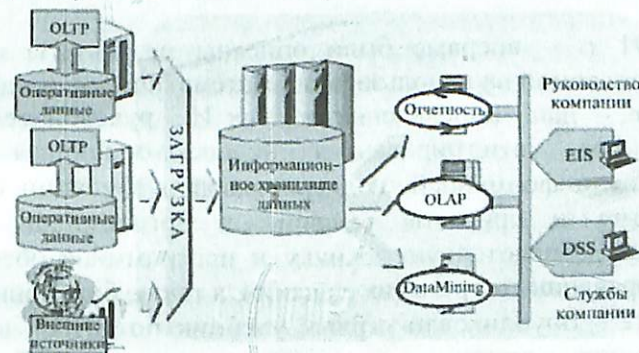


Рис. 2.1. Система поддержки принятия решения как составная часть КИС

СППР обладает следующими четырьмя основными характеристиками:

1. использует и данные, и модели;
2. предназначены для помощи руководителям в принятии решений для слабоструктурированных и неструктурированных задач;
3. поддерживают, а не заменяют, выработку решений руководителями;
4. улучшают эффективность решений.

До середины 60-х годов прошлого века создание больших информационных систем было чрезвычайно дорогостоящим, поэтому первые ИС руководители (так называемые Management

Information Systems — MIS) были созданы в эти годы лишь в достаточно больших компаниях. MIS предназначались для подготовки периодических структурированных отчетов для менеджеров.

В конце 60-х годов появляется новый тип ИС — модель-ориентированные СППР (Model-oriented Decision Support Systems — DSS) или системы управленческих решений (Management Decision Systems — MDS).

По мнению первооткрывателей СППР Keen P. G. W., Scott Morton M. (1978), концепция поддержки решений была развита на основе теоретических исследований в области принятия решений и технических работ по созданию интерактивных компьютерных систем.

В 1971 г. — впервые были описаны результаты внедрения СППР, основанной на использовании математических моделей.

1974 г. — дано определение первых ИС руководителя — MIS: «MIS — это интегрированная человеко-машинная система обеспечения информацией, поддерживающая функции операций, менеджмента и принятия решений в организации. Системы используют компьютерную технику и программное обеспечение, модели управления и принятия решений, а также базу данных».

1978 г. — опубликован первый учебник по СППР, в котором исчерпывающе описаны аспекты создания СППР: анализ, проектирование, внедрение, оценка и разработка.

1980 г. — сформулированы основы классификации СППР.

1981 г. — созданы теоретические основы проектирования СППР. Выделены 4 необходимых компонента, присущих всем СППР: 1) языковая система (LanguageSystem — LS) — СППР может принимать все сообщения; 2) система презентаций (PresentationSystem (PS)) (СППР может выдавать свои сообщения); 3) система знаний (Knowledge System — KS) — все знания СППР сохраняет; 4) система обработки задач (Problem-Processing System (PPS)) — программный «механизм», который пытается распознать и решить задачу во время работы СППР.

1981 г. — разработана информационная система руководителя (Executive Information System (EIS)) — компьютерная система,

предназначенная для обеспечения текущей адекватной информации для поддержки принятия управленческих решений менеджером.

Начиная с 1990-х разрабатываются так называемые Data Warehouses — хранилища данных. Хранилище данных — это очень большая предметно-ориентированная информационная корпоративная база данных, предназначенная для подготовки отчетов, анализа бизнес-процессов и поддержки принятия решений. Строится на базе клиент-серверной архитектуры, реляционной СУБД и утилит поддержки принятия решений. Данные, поступающие в хранилище данных, становятся доступны только для чтения.

В 1993 г. Появился термин OLAP (Online Analytical Processing) — оперативный анализ данных, онлайн-аналитическая обработка данных для поддержки принятия важных решений. Исходные данные для анализа представлены в виде многомерного куба, по которому можно получать нужные разрезы — отчеты.

В начале нового тысячелетия была создана СППР на основе Web.

Для СППР отсутствует не только единое общепринятое определение, но и исчерпывающая классификация. Разные авторы предлагают разные классификации.

На уровне пользователя СППР разделяются на пассивные, активные и кооперативные СППР. Пассивной СППР называется система, которая помогает процессу принятия решения, но не может вынести предложение, какое решение принять. Активная СППР может сделать предложение, какое решение следует выбрать. Кооперативная позволяет ЛПР изменять, пополнять или улучшать решения, предлагаемые системой, посылая затем эти изменения в систему для проверки. Система изменяет, пополняет или улучшает эти решения и посылает их опять пользователю. Процесс продолжается до получения согласованного решения.

На концептуальном уровне могут быть СППР, управляемые сообщениями (Communication-Driven DSS), СППР, управляемые данными (Data-Driven DSS), СППР, управляемые документами (Document-Driven DSS), СППР, управляемые знаниями (Knowledge-Driven DSS) и СППР, управляемые моделями (Model-Driven DSS). СППР, управляемые моделями, характеризуются в основном

доступ и манипуляции с математическими моделями (статистическими, финансовыми, оптимизационными, имитационными). Отметим, что некоторые OLAP-системы, позволяющие осуществлять сложный анализ данных, могут быть отнесены к гибридным СППР, которые обеспечивают моделирование, поиск и обработку данных.

Управляемая сообщениями (Communication-Driven DSS) (ранее групповая СППР - GDSS) СППР поддерживает группу пользователей, работающих над выполнением общей задачи.

СППР, управляемые данными (Data-Driven DSS) или СППР, ориентированные на работу с данными (Data-oriented DSS) в основном ориентируются на доступ и манипуляции с данными. СППР, управляемые документами (Document-Driven DSS), управляют, осуществляют поиск и манипулируют неструктурированной информацией, заданной в различных форматах. Наконец, СППР, управляемые знаниями (Knowledge-Driven DSS) обеспечивают решение задач в виде фактов, правил, процедур.

На техническом уровне различает СППР всего предприятия и настольную СППР. СППР всего предприятия подключена к большому хранилищу информации и обслуживает многих менеджеров предприятия. Настольная СППР – это малая система, обслуживающая лишь один компьютер пользователя.

При создании СППР можно использовать Web-технологии. В настоящее время СППР на основе Web-технологий для ряда компаний являются синонимами СППР предприятия.

С точки зрения функциональных возможностей и областей использования можно выделить СППР трех классов.

СППР первого класса – системы индивидуального пользования, базы знаний которых формируются непосредственным пользователем. В них используются многокритериальное оценивание альтернатив. Они предназначены для использования государственными служащими среднего ранга.

СППР второго класса есть системы индивидуального пользования, которые адаптируются к опыту пользователя. Они предназначены для поддержки решений в ситуациях, которые часто встречаются (выбор субъекта кредитования, выбор исполнителя

работы, назначение на должность и т.д.) Такие системы также используют оценивание альтернатив за несколькими критериями и обеспечивают поддержку решения в имеющейся ситуации на основе результатов практического использования результатов, которые приняты в прошлом.

СППР третьего класса могут использоваться даже тогда, когда СППР первых двух типов нельзя применить из-за отсутствия единых критериев для оценки каждой альтернативы. Они имеют наибольшие функциональные возможности, предназначенные для использования в органах государственного управления высшего уровня (Администрация Президента, Верховный Совет, Кабинет Министров, министерства, областные госадминистрации) и крупных бизнес-структурах.

Архитектура СППР

Архитектура СППР представляется авторами по-разному. Приведем пример. G. Marakas предложил обобщенную архитектуру, состоящую из пяти различных частей:

- система управления данными (the data management system);
- система управления моделями (the model management system);
- машина знаний (the knowledge engine);
- интерфейс пользователя (the user interface);
- пользователи (the user).

На сегодняшний день можно выделить четыре наиболее популярных типа архитектур СППР:

- функциональная СППР;
- независимые витрины данных;
- двухуровневое хранилище данных;
- трехуровневое хранилище данных.

Функциональная СППР является наиболее простой с архитектурной точки зрения (рис. 2.2). Такие системы часто встречаются на практике, в особенности в организациях с невысоким уровнем аналитической культуры и недостаточно развитой информационной инфраструктурой.

Характерной чертой функциональной СППР является то, что анализ осуществляется с использованием данных из оперативных систем.

Преимущества: быстрое внедрение за счет отсутствия этапа перегрузки данных в специализированную систему; минимальные затраты за счет использования одной платформы.



Рис. 2.2. Функциональная СППР

Недостатки: единственный источник данных, потенциально сужающий круг вопросов, на которые может ответить система; очень низкое качество данных оперативных систем с точки зрения их роли в поддержке принятия стратегических решений. В силу отсутствия этапа очистки данных функциональной СППР, как правило, обладают невысоким качеством; большая нагрузка на оперативную систему. Сложные запросы могут привести к остановке работы оперативной системы, что весьма нежелательно.

Независимые витрины данных (рис. 2.3) часто появляются в организации исторически и встречаются в крупных организациях с большим числом независимых подразделений, зачастую имеющих свои собственные отделы информационных технологий.

Витрины проектируются для ответов на конкретный ряд вопросов; данные в витрине оптимизированы для использования определенными группами пользователей, что облегчает процедуры их наполнения, а также способствует повышению производительности.

Недостатки: данные хранятся многократно в различных витринах данных (это приводит к дублированию данных, а следовательно, к увеличению расходов на хранение и потенциальным проблемам, связанным с необходимостью поддержания непротиворечивости данных); процесс наполнения витрин данных при большом количестве источников данных

потенциально очень сложен; данные не консолидируются на уровне предприятия, таким образом, отсутствует единая картина бизнеса.



Рис. 2.3. Независимые витрины данных

Преимущества: витрины данных можно внедрять достаточно быстро. **Двухуровневое хранилище данных** (см. рис. 2.4) строится централизованно для представления информации в рамках компании. Для поддержки такой архитектуры необходима выделенная команда профессионалов в области хранилищ данных. Это означает, что вся организация должна согласовать все определения и процессы преобразования данных.

Преимущества: данные хранятся в единственном экземпляре; затраты на хранение данных минимальны; отсутствуют проблемы, связанные с синхронизацией нескольких копий данных; данные консолидируются на уровне предприятия, что позволяет иметь единую картину бизнеса.

Недостатки: данные не структурируются для поддержки потребностей отдельных пользователей или групп пользователей; возможны проблемы с производительностью системы; имеются трудности с разграничением прав пользователей на доступ к данным.



Рис. 2.4. Двухуровневое хранилище данных

Трехуровневое хранилище данных представляет собой единый централизованный источник корпоративной информации (рис. 2.5). Витрины данных представляют собой подмножества данных из хранилища, организованные для решения задач отдельных подразделений компании. Конечные пользователи имеют возможность доступа к детальным данным хранилища в случае, если данных в витрине недостаточно, а также для получения более полной картины состояния бизнеса.

Преимущества: создание и наполнение витрин данных упрощено, поскольку наполнение происходит из единого стандартизованного надежного источника очищенных нормализованных данных; витрины данных синхронизированы и совместимы с корпоративным представлением (имеется корпоративная модель данных, существует возможность сравнительно легкого расширения хранилища и добавления новых витрин данных); гарантированная производительность.

Недостатки: существует избыточность данных, ведущая к росту требований на хранение данных; требуется согласованность с принятой архитектурой многих областей с потенциально различными требованиями (например, скорость внедрения иногда конкурирует с требованиями следовать архитектурному подходу).

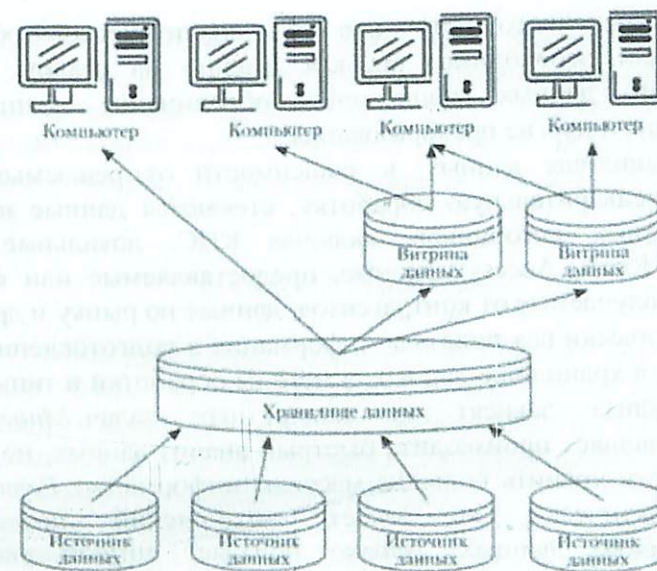


Рис. 2.5. Трехуровневое хранилище данных

Выше приведены основные варианты архитектур СППР. Выбор конкретного варианта зависит от условий, в которые поставлена проектная группа. На выбор архитектуры влияют ответы на такие вопросы, как: нужен ли быстрый возврат от инвестиций; является ли проектная группа профессиональной; существует ли формализованная методология.

Многомерные хранилища данных

Хранилище данных является неотъемлемым элементом большинства корпоративных систем, основанных на технологии Business Intelligence (BI). Как правило, даже небольшие компании используют несколько ИС для автоматизации различных сфер деятельности. Кроме того, большинство компаний использует отдельные системы в филиалах и региональных офисах. Данные, получаемые от различных структурных элементов компании, не унифицированы, часто противоречивы, и показатели, используемые для анализа и управления, не могут быть из них получены напрямую. Использование хранилища данных как

специализированного источника для аналитической обработки информации необходимо, так как именно на стадии сбора и интеграции данных происходит объединение данных, их унификация и другие преобразования.

В хранилище данных, в зависимости от решаемых задач, пройдя предварительную обработку, стекаются данные из самых разнообразных источников, включая КИС, локальные файлы (таблицы Excel, Access), данные, предоставляемые или каким-то образом получаемые от контрагентов, данные по рынку и др.

Практически вся полезная информация в подготовленном виде находится в хранилище данных, а методы обработки и типы систем бизнес-анализа зависят от конкретных задач. Многомерная модель позволяет производить быстрый анализ данных, но не дает возможности хранить большие массивы информации. Реляционная модель практически не имеет ограничений по объему накапливаемых данных, однако обладает низкой скоростью выполнения аналитических запросов.

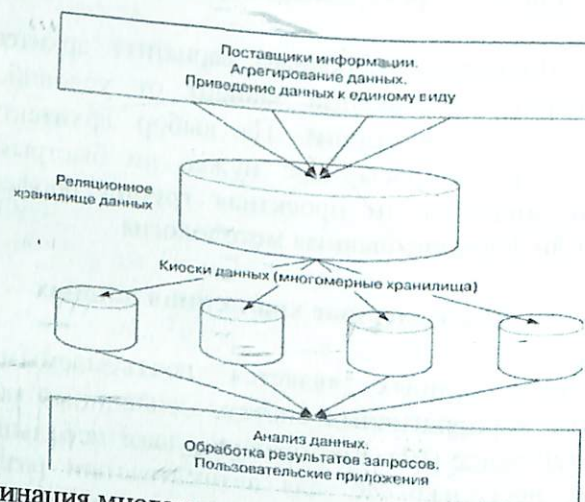


Рис 2.6. Комбинация многомерного и реляционного подхода: киоск данных

Киоск данных — это специализированное многомерное хранилище, обслуживающее одно из направлений деятельности предприятия.

Системы, использующие хранилища данных, строятся на основе архитектуры «клиент-сервер». Хранилище размещается на специализированном сервере, которому требуются мощные многопроцессорные вычислительные системы. В качестве СУБД используют СУБД, поддерживающие параллельную обработку запросов.

2.2. OLAP технологии

OLAP (Online Analytical Processing) - это система аналитической обработки данных. Она предназначена для подготовки отчетов, построения прогностических сценариев и выполнения статистических расчетов на базе больших информационных массивов, имеющих сложную структуру.

Основное назначение OLAP-систем: динамический многомерный анализ исторических и текущих данных, стабильных во времени; анализ тенденций; моделирование и прогнозирование будущего. Такие системы, как правило, ориентированы на обработку произвольных, заранее не регламентированных запросов. В качестве основных характеристик этих систем можно отметить следующие:

- поддержка многомерного представления данных, равноправие всех измерений, независимость производительности от количества измерений;
- прозрачность для пользователя структуры, способов хранения и обработки данных;
- автоматическое отображение логической структуры данных во внешние системы;
- динамическая обработка разреженных матриц эффективным способом.

Термин OLAP часто отождествляют с системами поддержки принятия решений DSS (Decision Support Systems). А в качестве синонима термина «решения» используют Data Warehousing - «хранилища (склады) данных». Под этим понимается набор организационных решений, программных и аппаратных средств для обеспечения аналитиков информацией на основе данных из

систем обработки транзакций нижнего уровня и других источников.

«Склады данных» позволяют обрабатывать данные, накопленные за длительные периоды времени. Эти данные являются разнородными (и не обязательно структурированными). Для «складов данных» присущ многомерный характер запросов. Огромные объемы данных, сложность структуры как данных, так и запросов - все это требует использования специальных методов доступа к информации.

В других источниках понятие Системы Поддержки Принятия Решений (СППР) считается более широким. Хранилища данных и средства оперативной аналитической обработки могут служить одними из компонентов архитектуры СППР.

OLAP всегда включает в себя интерактивную обработку запросов и последующий многопроходный анализ информации, который позволяет выявить разнообразные, не всегда очевидные тенденции, наблюдающиеся в предметной области.

Иногда различают OLAP в узком смысле - как системы, которые обеспечивают только выборку данных в различных разрезах, и OLAP в широком смысле, или просто OLAP, включающие в себя:

- поддержку нескольких пользователей, редактирующих БД.
- функции моделирования, в том числе вычислительные механизмы получения производных результатов, а также агрегирования и объединения данных;
- прогнозирование, выявление тенденций и статистический анализ.

Каждый из этих типов систем требует специфической организации данных, а также специальных программных средств, обеспечивающих эффективное выполнение стоящих задач.

OLAP-средства обеспечивают проведение анализа деловой информации по множеству параметров, таких как вид товара, географическое положение покупателя, время оформления сделки и продавец, каждый из которых допускает создание иерархии представлений. Так, для времени можно пользоваться годовыми, квартальными, месячными и даже недельными и дневными промежутками; географическое разбиение может проводиться по

городам, штатам, регионам, странам или, если потребуется, по целым полушариям.

OLAP-системы можно разбить на три класса.

1 класс. Наиболее сложными и дорогими из них являются основанные на патентованных технологиях серверы многомерных БД. Эти системы обеспечивают полный цикл OLAP-обработки и либо включают в себя, помимо серверного компонента, собственный интегрированный клиентский интерфейс, либо используют для анализа данных внешние программы работы с электронными таблицами. Продукты этого класса в наибольшей степени соответствуют условиям применения в рамках крупных информационных хранилищ. Для их обслуживания требуется целый штат сотрудников, занимающихся как установкой и сопровождением системы, так и формированием представлений данных для конечных пользователей. Обычно подобные пакеты довольно дороги. В качестве примеров продуктов этого класса можно привести систему Essbase корпорации ArborSoftware, Express фирмы IRI (входящей теперь в состав Oracle), Lightship производства компании PilotSoftware и др.

2 класс OLAP-систем - реляционные OLAP-системы (ROLAP). Здесь для хранения данных используются старые реляционные СУБД, а между БД и клиентским интерфейсом организуется определяемый администратором системы слой метаданных. Через этот промежуточный слой клиентский компонент может взаимодействовать с реляционной БД как с многомерной. Подобно средствам первого класса, ROLAP-системы хорошо приспособлены для работы с крупными информационными хранилищами, требуют значительных затрат на обслуживание специалистами информационных подразделений и предусматривают работу в многопользовательском режиме. Среди продуктов этого типа - IQ/Vision корпорации IQ Software, DSS/Server и DSS/Agent фирмы MicroStrategy и DecisionSuite компании Information Advantage.

ROLAP-средства реализуют функции поддержки принятия решений в надстройке над реляционным процессором БД. Такие программные продукты должны отвечать ряду требований, в частности:

- иметь мощный оптимизированный для OLAP генератор SQL-выражений, позволяющий применять многопроходные SQL-операторы SELECT и/или коррелированные подзапросы;

- обладать достаточно развитыми средствами для проведения нетривиальной обработки, обеспечивающей ранжирование, сравнительный анализ и вычисление процентных соотношений в рамках класса;

- генерировать SQL-выражения, оптимизированные для целевой реляционной СУБД, включая поддержку доступных в ней расширений этого языка;

- предоставлять механизмы описания модели данных с помощью метаданных и давать возможность использовать эти метаданные для построения запросов в реальном масштабе времени;

- включать в себя механизм, позволяющий оценивать качество построения сводных таблиц с точки зрения скорости вычисления, желательно с накоплением статистики по их использованию.

3 класс OLAP-систем - инструменты генерации запросов и отчетов для настольных ПК, дополненные OLAP-функциями или интегрированные с внешними средствами, выполняющими такие функции. Эти весьма развитые системы осуществляют выборку данных из исходных источников, преобразуют их и помещают в динамическую многомерную БД, функционирующую на ПК конечного пользователя. Указанный подход, позволяющий обойтись как без дорогостоящего сервера многомерной БД, так и без сложного промежуточного слоя метаданных, необходимого для ROLAP-средств, обеспечивает в то же время достаточную эффективность анализа. Эти средства для настольных ПК лучше всего подходят для работы с небольшими, просто организованными БД. Потребность в квалифицированном обслуживании для них ниже, чем для других OLAP-систем, и примерно соответствует уровню обычных сред обработки запросов. В числе основных участников этого сектора рынка - компания Brio Technology со своей системой Brio Query Enterprise, Business Objects с одноименным продуктом и Cognos с PowerPlay.

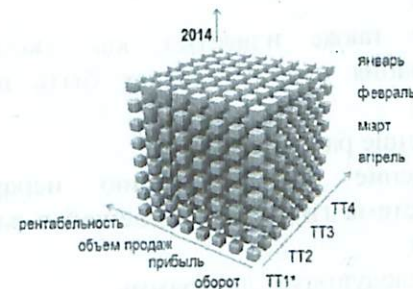


Рис. 2.7. OLAP куб

В основе концепции OLAP лежит куб OLAP. OLAP-куб — это структура данных, оптимизированная для очень быстрого анализа данных.

Куб OLAP состоит из числовых фактов, называемых мерами, которые классифицируются по измерениям. OLAP Cube также называют гиперкубом.

Обычно операции с данными и анализ выполняются с использованием простой электронной таблицы, где значения данных располагаются в формате строк и столбцов. Это идеально подходит для двумерных данных. Однако OLAP содержит многомерные данные, причем данные обычно получают из другого и несвязанного источника. Использование электронной таблицы не является оптимальным вариантом. Куб может хранить и анализировать многомерные данные в логической и упорядоченной форме.

2.3. Основные аналитические операции OLAP

Четыре типа аналитических операций в OLAP:

1. Свернуть
2. Дрель-вниз
3. Ломтик и кости
4. Поворот (поворот)

1. Свернуть:

Свертывание также известно, как «консолидация» или «агрегация». Операция свертки может быть выполнена двумя способами (рис 2.7).

1. Уменьшение размеров
2. Восхождение на концепцию иерархии. Иерархия понятий — это система группировки вещей в зависимости от их порядка или уровня.

Рассмотрим следующую диаграмму

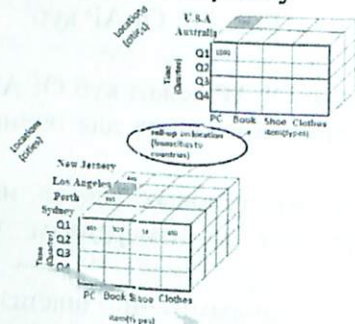


Рис.2.8. Свертывание куба

• В этом примере города Нью-Джерси и Потерянные Ангелы свернуты в страну США.

• Показатели продаж в Нью-Джерси и Лос-Анджелесе составляют 440 и 1560 соответственно. Они становятся 2000 после свертывания

• В этом процессе агрегирования данные перемещаются вверх по иерархии от города к стране.

• В процессе свертки необходимо удалить хотя бы одно или несколько измерений. В этом примере измерение Quarter удалено.

2. Развертка:

При детализации данные разбиты на более мелкие части. Это противоположность процесса накопления (рис 2.8). Это можно сделать через

- Двигаясь вниз по иерархии понятий
- Увеличение размера

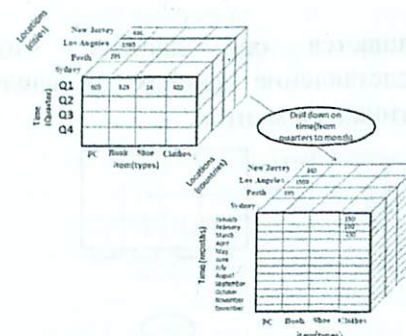


Рис.2.9. Развертка куба

Рассмотрим схему выше

1. Квартал Q1 пробурен до месяцев января, февраля и марта. Соответствующие продажи также являются регистрами.
2. В этом примере добавляются месяцы измерений.

3. Срез:

Здесь выбрано одно измерение и создан новый вложенный куб.

Следующая диаграмма объясняет, как выполнялась операция среза:

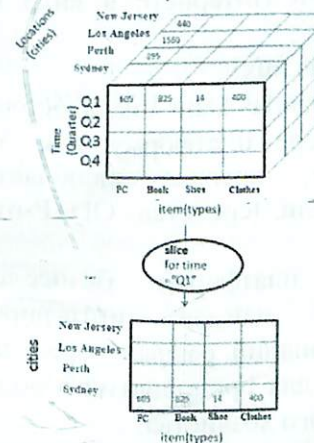


Рис. 2.10. Срез куба

- Время измерения нарезается с Q1 в качестве фильтра.
- Новый куб создан в целом.

4. Пивот

В Pivot вращается оси данных, чтобы обеспечить альтернативное представление данных. В следующем примере сводка основана на типах элементов.

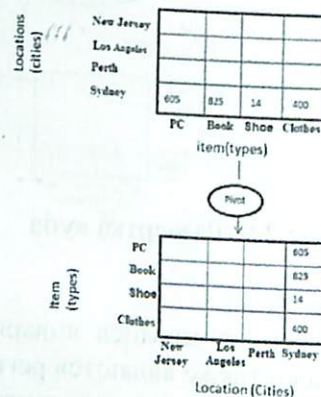


Рис. 2.11. Пивот

Инструменты OLAP

Инструменты для онлайн-аналитической обработки очень хорошо представлены в Интернете в виде как платных, так и бесплатных версий.

Наиболее популярные из них:

1. Dundas BI из Dundas Data Visualization представляет собой основанную на браузере платформу для бизнес-аналитиков и визуализации данных, которая включает интегрированные информационные панели, средства OLAP-отчетов и аналитику данных.

2. Yellowfin - платформа бизнес-аналитики, которая представляет собой единое интегрированное решение, разработанное для компаний разных отраслей и масштабов. Эта система настраивается для предприятий в области бухгалтерского учета, рекламы, сельского хозяйства.

3. ClicData - это решение для бизнес-аналитиков (BI), предназначенное для использования в основном предприятиями малого и среднего бизнеса. Инструмент позволяет конечным пользователям создавать отчеты и информационные панели. Board создан для объединения бизнес-аналитики, управления

корпоративной эффективностью и представляет собой полнофункциональную систему, которая обслуживает компании среднего и корпоративного уровня.

4. Domo - это облачный пакет управления бизнесом, который объединяется с несколькими источниками данных, включая электронные таблицы, базы данных, социальные сети и любое существующее облачное или локальное программное решение.

5. InetSoft Style Intelligence - это программная платформа для бизнес-аналитиков, которая позволяет пользователям создавать информационные панели, визуальную технологию анализа OLAP и отчеты с помощью механизма mashup.

6. Birst от Infor Company представляет собой сетевое решение для бизнес-аналитиков и анализа, который объединяет идеи различных команд и помогает принимать обоснованные решения. Инструмент позволяет децентрализованным пользователям увеличить модель корпоративных команд.

7. Halo - это комплексная система управления цепочками поставок и бизнес-аналитики, которая помогает в планировании бизнеса и прогнозировании запасов для управления цепочками поставок. Система использует данные из всех источников - больших, малых и промежуточных.

8. Chartio - это облачное решение для бизнес-аналитиков, которое предоставляет учредителям, бизнес-группам, аналитикам данных и группам продуктов инструменты организации для повседневной работы.

9. Exago BI - это веб-решение, предназначенное для внедрения в веб-приложения. Внедрение Exago BI позволяет компаниям всех размеров предоставлять своим клиентам специальную, оперативную и интерактивную отчетность.

2.4. OLTP-системы

OLTP, транзакционная система — обработка транзакций в реальном времени. Способ организации БД, при котором система работает с небольшими по размерам транзакциями, но идущими большим потоком, и при этом клиенту требуется от системы

минимальное время отклика. Термин OLTP применяют также к системам.

Системы БД, используемые в рамках локальных компьютерных сетей маркетинговых подразделений, оперируют большими объемами актуальной детализированной информации с помощью *технологии оперативной обработки транзакций* — OLTP (On-line Transaction Processing — обработка транзакций в реальном времени).

Транзакция представляет собой неделимую в отношении воздействия на БД последовательность операций манипулирования данными БД (*чтение, удаление, вставка, модификация значений данных*), которая реализует некоторое осмысленное, с точки зрения пользователя, действие (например, оформление заказа на товар, перевод денег со счета на счет). Транзакция выполняется по принципу «*все или ничего*», т. е. либо выполняются полностью все операции транзакции (без пропусков), либо она не выполняется совсем. Управление выполнением транзакций осуществляют СУБД.

Как правило, транзакцию составляют операции над данными, логически связанными друг с другом. Если транзакция выполняется целиком и успешно, то в БД сохраняются все произведенные изменения в данных, т. е. БД актуализируется, переходя из одного *целостного состояния* в другое *целостное состояние*. Если же по каким-либо причинам одно из действий транзакции невыполнимо или произошло какое-либо нарушение работы системы, БД возвращается в исходное состояние, которое было до начала транзакции (говорят, что происходит *откат транзакции*). Таким образом, при любом исходе выполнения транзакции целостность данных БД не нарушается.

Пример

В БД системы продаж товаров по заказам при оформлении каждого заказа должны выполняться три операции над данными, хранящимися в различных частях БД:

- должна быть произведена регистрация заказа;
- должно произойти резервирование указанного количества товара;

• должно быть уменьшено соответствующим образом количество товара на складе, предлагающегося к дальнейшим продажам.

На любом этапе изменения данных возможен технический или программный сбой. В таком случае целостность БД будет нарушена: с одной стороны, продано определенное количество товара, с другой — к продаже предлагается такое же количество товара, как и до оформления заказа. Для предотвращения подобных нарушений вводится транзакция «*Оформление заказа*» (рис. 6), в которой над БД либо должны произвестись все необходимые операции (констатирующие, что товар продан, уменьшен его запас на складе), либо должен произойти откат к исходному состоянию (товар не продан, его количество на складе не изменилось).

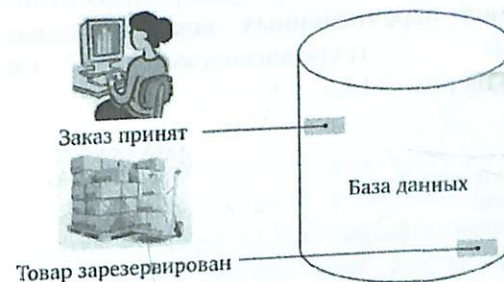


Рис. 2.12. Транзакция «Оформление заказа»

В случае одновременного обращения пользователей (маркетологов) к некоторой БД транзакции, инициируемые разными пользователями со своих компьютеров, выполняются не параллельно (что невозможно для одной БД), а в соответствии с некоторым алгоритмом ставятся в очередь и выполняются поочередно. При этом пользователям, одновременно работающим с одной и той же БД, кажется (в силу быстродействия процессора), что они работают как бы в однопользовательской системе и не мешают друг другу.

Таким образом, основная функция OLTP-технологий заключается в одновременном выполнении (в режиме реального времени) большого количества коротких транзакций, идущих интенсивным потоком от большого числа пользователей, а также

регистрации в БД всех событий, связанных с деятельностью предприятия.

Для реализации OLTP-технологий используются, как правило, компьютерные сети с архитектурой «клиент — сервер» (см. 6).

Системы обработки данных на базе OLTP-технологий называют *OLTP-системами* или *транзакционными системами*. Они присутствуют практически на всех предприятиях и используются для оперативного ввода, структурированного хранения данных, выполнения ежедневно повторяющихся задач обработки данных (счетов, актов, накладных, прайс-листов и других документов и сообщений), а также для создания различных результирующих документов — отчетов, сводок, ведомостей. При этом формы входных и выходных документов, схемы документооборота должны быть жестко регламентированы.

Для решения перечисленных задач выполняются типовые информационные технологические OLTP-операции, представленные на рис. 2.13.

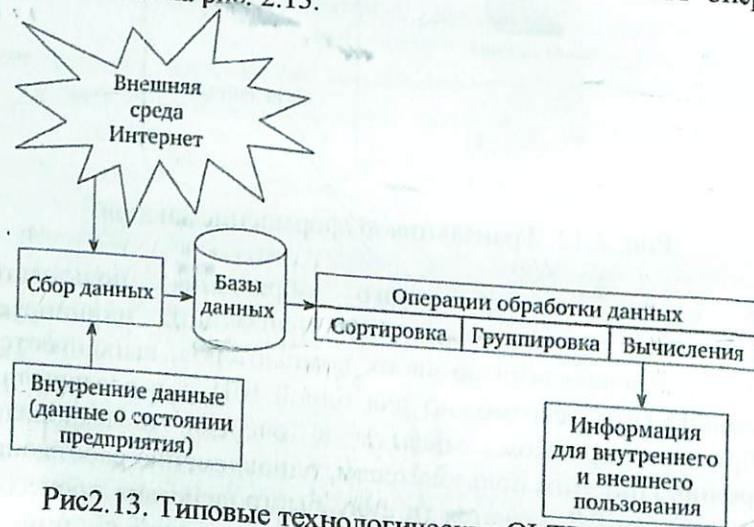


Рис. 2.13. Типовые технологические OLTP-операции

Предоставляя данные из постоянно обновляемых БД, OLTP-системы не отслеживают динамику изменения процессов на больших временных промежутках, практически не производят сложную обработку данных и, что самое важное, не формируют

выводы по имеющимся данным, оставляя эту функцию лицу, принимающему решение.

В системах автоматизации маркетинга OLTP-системы выполняют операции сбора, регистрации и ввода исходных данных, первичной обработки данных, их хранения, адекватной визуализации, поиска, выдачи рекомендаций, отчетных материалов.

К особенностям транзакционных систем автоматизации маркетинга предприятия можно отнести следующие факторы:

- использование фиксированного набора надежных и безопасных методов ввода;
- относительная простота алгоритмов производимых расчетов;
- оперативность изменений в номенклатуре и структуре обрабатываемой маркетинговой информации;
- массовость и территориальная распространенность мест сбора исходных маркетинговых данных;
- поддержка большого числа пользователей;
- стабильность структуры БД в течение долгого времени (5—7 лет), что необходимо для эффективного функционирования маркетинговых прикладных программ в данной системе;
- относительно простая структура запросов к БД, причем обычно запросы носят стандартный характер, а система стандартных запросов фиксирована;
- небольшое время отклика на запрос.

Аналитические возможности OLTP-систем, используемых в маркетинге, сильно ограничены (либо вообще отсутствуют). Объясняется это тем, что в настоящее время OLTP-системы реализуются на основе табличного (реляционного) представления данных БД, причем структура БД обычно довольно сложна — она может содержать многие десятки и даже сотни таблиц, ссылающихся друг на друга. Запросы на некую комплексную информацию, характерные для аналитических маркетинговых задач (к примеру, запрос данных о поквартальной динамике объемов продаж по определенному виду товара в определенном филиале), породят сложные соединения таблиц и просмотр таблиц целиком. На один такой запрос уйдет масса времени и компьютерных ресурсов, что затормозит обработку текущих транзакций.

Контрольные вопросы

1. Как связаны OLAP, OLTP, хранилища данных, аналитика, анализ и интеллектуальный анализ данных?
2. Что такое OLAP системы?
3. Что такое OLTP системы?
4. Различия между базами данных OLAP и OLTP?
5. Типы аналитических операций в OLAP?
6. Какие классы имеются в OLAP-системах?
7. Какие типы задач решают системы бизнес-аналитики?
8. Что представляет собой технология OLAP и каково ее назначение?
9. Какие требования предъявляются к OLAP-системам?
10. В чем состоят особенности и принципиальные отличия OLAP от OLTP?
11. Какие существуют разновидности OLAP?
12. Какими возможностями обладают службы бизнес-аналитики в составе Microsoft SQL Server?

ГЛАВА III. МНОГОМЕРНАЯ МОДЕЛЬ ДАННЫХ. OLAP СИСТЕМЫ

В процессе принятия решений пользователь генерирует некоторые гипотезы. Для превращения их в законченные решения эти гипотезы должны быть проверены. Проверка гипотез осуществляется на основании информации об анализируемой предметной области. Как правило, наиболее удобным способом представления такой информации для человека является зависимость между некоторыми параметрами. Например, зависимость объемов продаж от региона, времени, категории товара и т. п. Другим примером может служить зависимость количества выздоравливающих пациентов от применяемых средств лечения, возраст, а ит. п. В процессе анализа данных, поиска решений часто возникает необходимость в построении зависимостей между различными параметрами. Кроме того, число таких параметров может варьироваться в широких пределах. Как уже отмечалось ранее, традиционные средства анализа, оперирующие данными, которые представлены в виде таблиц реляционной БД, не могут в полной мере удовлетворять таким требованиям.

3.1. Что такое многомерная модель данных

В многомерной модели данные представляются в виде гиперкубов, используя которые, можно получать различные срезы при аналитической обработке данных. Оси гиперкуба содержат параметры, а ячейки — зависящие от них агрегатные данные. Вдоль каждой оси данные могут быть организованы в виде иерархии, представляющей различные уровни их детализации. Благодаря такой модели данных пользователи могут формулировать сложные запросы, генерировать отчеты, получать подмножества данных. Пример гиперкуба для трех измерений.

Осями многомерной системы координат служат основные атрибуты анализируемого бизнес-процесса. Например, для продаж это могут быть товар, регион, тип покупателя. В качестве одного из измерений используется время. На пересечениях осей измерений

находятся данные, количественно характеризующие процесс, — меры. Это могут быть объемы продаж в штуках или в денежном выражении, остатки на складе, издержки и т.п. В качестве мер в трехмерном кубе, изображенном.

Многомерная модель называется *полукубической*, если предметная область может быть отображена в виде нескольких гиперкубов с различной размерностью и разными измерениями в качестве граней. Если же модель может состоять из одного гиперкуба или нескольких гиперкубов с одинаковой размерностью и совпадающими измерениями, то она называется *гиперкубической*.

	Самара			
СПб				
Москва				
	Октябрь	Ноябрь	Декабрь	
Иванов	120 000	140 000	150 000	
Петров	110 000	150 000	200 000	

Рис. 3.1. Пример гиперкуба

Достоинством такой модели является удобство и эффективность аналитической обработки данных, изменяющихся во времени. Однако она является громоздкой для простых задач оперативной обработки данных. Типичным представителем баз данных, реализующих многомерную модель, является Oracle Express Server.

Объектно-ориентированная модель данных. Объектно-ориентированная модель данных является расширением положений объектно-ориентированного программирования (в то время как реляционная модель возникла на основе теории множеств именно как модель данных). Группой управления Объектно-ориентированных БД разработан стандарт ODMG-93 (Object Data Base Management Group).

Структура объектно-ориентированной БД графически представима в виде дерева, узлами которого являются объекты.

Видимая структура объекта определяется свойствами его класса. *Класс* включает в себя объекты, при этом структура и поведение объектов одного класса одинаковы. Каждый объект, а именно экземпляр класса считается потомком объекта, в котором он определен как свойство. *Свойства объектов* — либо стандартный тип, например *string*, либо конструируемый пользователем тип *class*. Поведение объектов задается с помощью методов. *Метод* — это некая операция, которую можно применить к объекту.

В качестве примера рассмотрим БД «Сотрудник» (рис. 3.6). Для каждого объекта определены свойства, типы данных и значения. В этой БД «Сотрудник» — родитель (предок) для «Аттестация» и «Повышение квалификации»; «Повышение квалификации» — родитель для «Обучающая организация».

Логическая структура объектно-ориентированной БД похожа на иерархическую, основное отличие — в методах манипулирования данными. Над БД можно производить такие действия, как логические операции, усиленные объектно-ориентированными методами инкапсуляции, наследования и полиморфизма.

Инкапсуляция ограничивает область видимости имени свойства пределами того объекта, в котором это свойство определено. Так, если в «Сотрудник» добавлено свойство «телефон», то это будет телефон сотрудника; свойство с таким же названием в «Обучающая организация» будет телефоном организации. Иными словами, смысл свойства будет определяться тем объектом, в который оно инкапсулировано.

Наследование, наоборот, распространяет область видимости свойства на всех потомков объекта. Так, всем объектам типа «Повышение квалификации», являющимся потомками объекта «Сотрудник», можно приписать свойства родителя (например, фамилия, имя, район проживания).

Достоинствами объектно-ориентированной модели по сравнению с реляционной является возможность отображения информации о сложных взаимосвязях объектов, отсутствие ограниченности в структурах хранения данных. В объектно-ориентированной БД могут храниться не только структура, но и

поведенческие аспекты данных. С использованием объектно-ориентированного подхода можно создавать и БД с большими объемами семантической информации, такие как мультимедийные и специализированные для конкретных предметных областей (географические, проектные и др.).

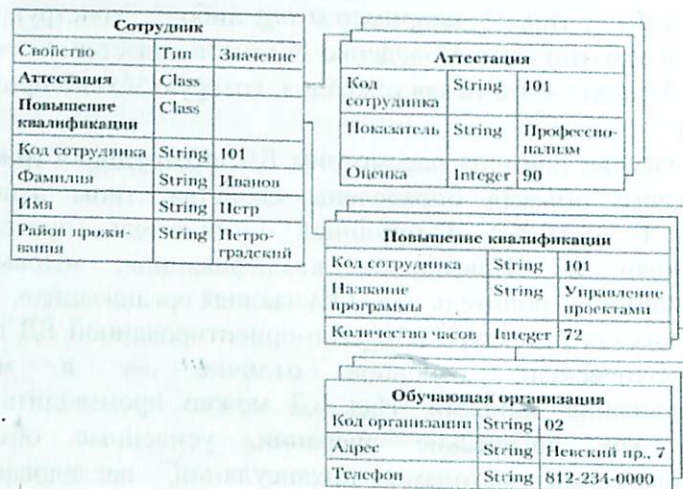


Рис.3.2 Объектно-ориентированная база данных «Сотрудник»

Полиморфизм означает способность одного и того же программного кода работать с разнотипными данными. Иными словами, он означает допустимость в объектах разных типов иметь методы — процедуры и функции — с одинаковыми именами. Во время выполнения объектной программы одни и те же методы оперируют с разными объектами в зависимости от типа аргумента.

Полиморфизм означает способность одного и того же программного кода работать с разнотипными данными. Иными словами, он означает допустимость в объектах разных типов иметь методы — процедуры и функции — с одинаковыми именами. Во время выполнения объектной программы одни и те же методы оперируют с разными объектами в зависимости от типа аргумента. Хранилища данных, как правило, содержат следующие три типа фактов:

- **события (event)**, по крайней мере, на уровне самой большой гранулярности, как правило, моделируют события

реального мира, при этом каждый факт представляет определенный экземпляр изучаемого явления. Примерами могут служить продажи, щелчки мышью на Web-странице или движение товаров на складе;

- **мгновенные снимки (snapshot)** моделируют состояние объекта в данный момент времени, такие как уровни наличия товаров в магазине или на складе и число пользователей Web-сайта. Один и тот же экземпляр явления реального мира, например, конкретная банка бобов, может возникать в нескольких фактах;

- **совокупные мгновенные снимки (cumulative snapshot)** содержат информацию о деятельности организации за определенный отрезок времени. Например, совокупный объем продаж за предыдущий период, включая текущий месяц, можно легко сравнить с показателями за соответствующие месяцы прошлого года.

Хранилище данных часто содержит все три типа фактов. Одни и те же исходные данные, например, движение товаров на складе, могут содержаться в трех различных типах кубов: поток товаров на складе, список товаров и поток за год к текущей дате.

Параметры состоят из двух компонентов:

- численная характеристика факта, например, цена или доход от продаж;
- формула, обычно простая агрегативная функция, скажем, сумма, которая может объединять несколько значений параметров в одно.

В многомерной базе данных параметры, как правило, представляют свойства факта, который пользователь хочет изучить. Параметры принимают различные значения для разных комбинаций измерений. Свойство и формула выбираются таким образом, чтобы представлять осмысленную величину для всех комбинаций уровней агрегирования. Поскольку метаданные определяют формулу, данные, в отличие от случая электронных таблиц, не тиражируются.

При вычислениях три различных класса параметров ведут себя совершенно по-разному.

- **Аддитивные параметры** могут содержательным образом комбинироваться в любом измерении. Например, имеет смысл

суммировать общий объем продаж для продукта, местоположения и времени, поскольку это не вызывает наложения среди явлений реального мира, которые генерируют каждое из этих значений.

- **Полуаддитивные параметры**, которые не могут комбинироваться в одном или нескольких измерениях. Например, суммирование запасов по разным товарам и складам имеет смысл, но суммирование запасов товаров в разное время бессмысленно, поскольку одно и то же физическое явление может учитываться несколько раз.

- **Неаддитивные параметры** не комбинируются в любом измерении, обычно потому, что выбранная формула не позволяет объединить средние значения низкого уровня в среднем значении более высокого уровня.

Аддитивные и неаддитивные параметры могут описывать факты любого рода, в то время как полуаддитивные параметры, как правило, используются с мгновенными снимками или совокупными мгновенными снимками.

Запросы

Многомерная база данных естественным образом предназначена для определенных типов запросов.

- **Запросы вида slice-and-dice** осуществляют выбор, сокращающий куб. К примеру, можно рассмотреть сечение куба на рис. 1, приняв во внимание только те ячейки, которые касаются хлеба, а затем еще больше сократить его, оставив ячейки, относящиеся только к 2000 году. Фиксация значения измерения сокращает размерность куба, но при этом возможны и более общие операции выбора.

- **Запросы вида drill-down и roll-up** — взаимобратные операции, которые используют иерархию измерений и параметры для агрегирования. Обобщение до высших значений соответствует исключению размерности. Например, свертка от уровня «Город» до уровня «Страна» на рис. 2 агрегирует значения для Аалборга и Копенгагена в одно значение — Дания.

- **Запросы вида drill-across** комбинируют кубы, которые имеют одно или несколько общих измерений. С точки зрения реляционной алгебры такая операция выполняет слияние (join).

- **Запросы вида ranking** [6] возвращает только те ячейки, которые появляются в верхней или нижней части упорядоченного определенным образом списка, например, 10 самых продаваемых продуктов в Копенгагене в 2000 году.

- **Поворот (rotating)** куба дает пользователям возможность увидеть данные, сгруппированные по другим измерениям.

Реализация

Многомерные базы данных реализуют в двух основных формах.

- Системы многомерной оперативной аналитической обработки (MOLAP) хранят данные в специализированных многомерных структурах. Системы MOLAP, как правило, содержат средства для обработки разреженных массивов и применяют усовершенствованную индексацию и хеширование для поиска данных при выполнении запросов [6].

- Реляционные системы OLAP (ROLAP) [5] для хранения данных используют реляционные базы данных, а также применяют специализированные индексные структуры, такие как битовые карты, чтобы добиться высокой скорости выполнения запросов.

Системы MOLAP, как правило, позволяют добиться более эффективного использования дискового пространства, а также меньшего времени ответов при обработке запросов.

Врезка «Сокращение времени ответа при обработке запросов» описывает некоторые из применяемых с этой целью методик. Системы ROLAP, как правило, лучше масштабируются с ростом числа фактов, которые они могут хранить (хотя некоторые инструментальные средства MOLAP теперь становятся столь же масштабируемыми), более гибкими в том, что касается переопределения кубов, и лучше поддерживают частые обновления.

Достоинства двух подходов объединены в гибридном решении, при котором для хранения сводных данных более высокого уровня используется технология MOLAP, а в системах ROLAP размещаются детальные данные.

В ROLAP, как правило, используются схемы «звезда» и «снежинка», при которых данные хранятся в таблицах фактов и таблицах измерений. Таблица фактов содержит одну строку для

каждого факта в кубе. Для каждого измерения отводится отдельный столбец, содержащий значение параметра для конкретного факта, а также столбец для каждого измерения, которое содержит внешний ключ, ссылающийся на таблицу измерений для конкретного измерения.

Сложные многомерные данные

Традиционные многомерные модели данных и методы их реализации предполагают, что:

- все факты напрямую отображаются на значения измерений более низкого уровня, причем ровно на одно значение в каждом измерении;
- иерархии измерений представляют собой сбалансированные деревья.

Если эти предположения не выполняются, то стандартные модели и системы оказываются неадекватными. Особенно серьезные проблемы вызывают комплексные многомерные данные, поскольку они не являются суммируемыми (summarizable) — агрегированные результаты более высокого уровня нельзя получить из агрегированных результатов более низкого уровня. Запросы по результатам более низкого уровня будут приносить неверные данные, или предварительные вычисления, сохранение и последующее использование их результатов в данном случае невозможны. Вместо этого агрегированные результаты должны вычисляться непосредственно из базовых данных, что значительно увеличивает затраты на вычисления.

Суммирование требует применения распределенных агрегированных функций и значений иерархии измерений. Неформально иерархия измерений является «строгой», если ни одно из значений измерений не имеет более одного прямого родителя, «скьюррективной» (onto), если иерархия сбалансирована, и «покрывающей» (covering), если ни один локальный путь не «перескакивает» через уровень. Интуитивно это значит, что иерархии измерений должны быть сбалансированными деревьями. В случае нерегулярных иерархий измерений некоторые значения более низкого уровня при повторном использовании промежуточных результатов запросов будут либо посчитаны дважды, либо ни разу.

Нерегулярные иерархии возникают в разных приложениях, в том числе в иерархии административных структур, иерархии медицинских диагнозов и иерархии концепций для Web-порталов, подобных Yahoo. Одно из решений — нормализовать нерегулярные иерархии, процесс, который предусматривает пополнение несьюррективных и непокрывающих иерархий фиктивными значениями измерений, и перестраивает наборы родителей, для того чтобы решить проблемы нестрогих иерархий. Это преобразование может выполняться прозрачным для пользователя образом.

За 30 лет с момента своего возникновения технология многомерных баз данных прошла серьезную эволюцию. С недавних пор она стала реализовываться в решениях, предназначенных для массового рынка, а ведущие производители теперь выпускают многомерные ядра вместе со своими реляционными базами данных, причем часто без дополнительной оплаты. Многомерная технология стала значительно более масштабируемой и зрелой.

Это порождает несколько важных тенденций. Данные, которые необходимо анализировать, становятся все более распределенными. К примеру, это часто необходимо для выполнения анализа, при котором используются данные в формате XML, получаемые с определенных Web-сайтов. Растущая распределенность данных, в свою очередь, требует применения методов, которые позволяют легко добавлять новые данные в многомерные базы данных, тем самым, упрощая задачу создания интегрированного хранилища данных. Среди примеров — автоматическая генерация измерений и кубов из новых источников данных и методы простой и динамической очистки данных.

Технология многомерных баз данных также применяется к новым типам данных, которые современные технологии зачастую не в состоянии адекватно анализировать. К примеру, классические методики, такие как предагрегирование не могут гарантировать небольшое время ответа на запросы, если данные постоянно меняются, как это происходит, например, когда информация поступает с датчиков или от движущихся объектов, таких как автомобили, оснащенные средствами глобального позиционирования.

Наконец, технология многомерных баз данных все больше будет применяться там, где результаты анализа напрямую передаются в другие системы, тем самым, исключая участие человека в этом процессе. Этот контекст в совокупности с необходимостью постоянного обновления предъявляет более жесткие требования к производительности, которым не удовлетворяет современная технология.

3.2. Архитектура OLAP-систем: ROLAP, MOLAP, HOLAP

В предыдущей главе рассказано об основе концепции OLAP оперативной аналитической обработки данных (On-Line Analytical Processing).

Термин OLAP введен Коддом (E. F. Codd) в 1993 году. Главная идея данной системы заключается в построении многомерных таблиц, которые могут быть доступны для запросов пользователей. Эти многомерные таблицы или так называемые многомерные кубы строятся на основе исходных и агрегированных данных. И исходные, и агрегированные данные для многомерных таблиц могут храниться как в реляционных, так и в многомерных базах данных. Взаимодействуя с OLAP -системой, пользователь может осуществлять гибкий просмотр информации, получать различные срезы данных, выполнять аналитические операции детализации, свертки, сквозного распределения, сравнения во времени. Вся работа с OLAP -системой происходит в терминах предметной области.

OLAP-система включает в себя два основных компонента:

- OLAP-сервер—обеспечивает хранение данных, выполнение над ними необходимых операций и формирование многомерной модели на концептуальном уровне. В настоящее время OLAP-серверы объединяют с ХД или ВД;
- OLAP-клиент—представляет пользователю интерфейс к многомерной модели данных, обеспечивая его возможностью удобно манипулировать данными для выполнения задач анализа. OLAP-серверы скрывают от конечного пользователя способ реализации многомерной модели. Они формируют гиперкуб, с которым пользователи

посредством OLAP-клиента выполняют все необходимые манипуляции, анализируя данные. Между тем способ реализации очень важен, т. к. от него зависят такие характеристики, как производительность и занимаемые ресурсы.

Выделяют три основных способа реализации:

MOLAP—многомерный (multivariate) OLAP. Для реализации многомерной модели используют многомерные БД;

ROLAP— реляционный (relational) OLAP. Для реализации многомерной модели используют реляционные БД;

HOLAP — гибридный (hybrid) OLAP. Для реализации многомерной модели используют и многомерные, и реляционные БД;

DOLAP — настольный (desktop) OLAP. Является недорогой и простой в использовании OLAP-системой, предназначенной для локального анализа и представления данных, которые загружаются из реляционной или многомерной БД на машину клиента;

JOLAP — новая, основанная на Java коллективная OLAP-API-инициатива, предназначенная для создания и управления данными и метаданными на серверах OLAP.

Можно выделить следующие преимущества использования многомерных БД в OLAP-системах:

- поиски выборка данных осуществляются значительно быстрее, чем при многомерном концептуальном взгляде на реляционную БД, т. к. многомерная база данных денормализована и содержит заранее агрегированные показатели, обеспечивая оптимизированный доступ к запрашиваемым ячейкам и не требуя дополнительных преобразований при переходе от множества связанных таблиц к многомерной модели;
- многомерные БД легко справляются с задачами включения в информационную модель разнообразных встроженных функций, тогда как объективно существующие ограничения языка SQL делают выполнение этих задач на основе реляционных БД достаточно сложным, а иногда и невозможным.

С другой стороны, имеются так же существенные недостатки многомерных БД. За счет денормализации и предварительно выполненной агрегации объем данных в многомерной БД, как

правило, соответствует (по оценке Кодда) в $2,5 \div 100$ раз меньшему объему исходных детализированных данных; в подавляющем большинстве случаев информационный гиперкуб является сильно разреженным, а поскольку данные хранятся в упорядоченном виде, не определенные значения удастся удалить только за счет выбора оптимального порядка сортировки, позволяющего организовать данные в максимально большие непрерывные группы. Но даже в этом случае проблема решается только частично. Кроме того, оптимальный с точки зрения хранения разреженных данных порядок сортировки, скорее всего, не будет совпадать с порядком, который чаще других используется в запросах.

Так при добавлении нового измерения приходится изменять структуру всей БД, что влечет за собой большие затраты времени. На основании анализа достоинств и недостатков многомерных БД можно выделить следующие условия, при которых их использование является эффективным:

- объем исходных данных для анализа не слишком велик (не более нескольких гигабайт), т. е. уровень агрегации данных достаточно высок;
- набор информационных измерений стабилен;
- время ответа системы на нерегламентированные запросы является наиболее критичным параметром;
- требуется широкое использование сложных встроенных функций для выполнения кросс мерных вычислений над ячейками гиперкуба, в том числе необходима возможность написания пользовательских функций.

Любое конечное решение содержит OLAP-компоненту, которая является интерфейсом пользователя. Эти компоненты похожи друг на друга. Их визуальная часть состоит из элементов управления и элементов отображения данных. Как правило, это таблица, в полях которой содержатся данные, а колонки и строки являются элементами управления.

Подавляющее большинство поставщиков OLAP, а их в мире насчитывается около 140, не продают свои компоненты. Нам известно только три компонента, которые можно купить для собственной разработки. Это Decision Cube компании Inprise в составе компиляторов Delphi и C++ Builder, Pivot Table корпорации

Microsoft в составе MS Office, и Dynamic Cube компании Data Dynamic, специализирующейся на разработке OLAP-компонент.

Decision Cube компании Inprise поставляется как VCL-компонента. По нашей классификации относится к ROLAP-компонентам, то есть содержит в своем составе OLAP-машину и предназначен только для работы с реляционными СУБД или локальными таблицами. Он отличается весьма скромными возможностями. Например, в нем нельзя открыть один элемент измерения, или установить фильтр по нескольким измерениям, отобразить несколько фактов одновременно. Производительность компоненты невысока. Пределом является около 4000 записей при 5 измерениях. Компонента отображает в таблице одновременно только один факт. Неприятной особенностью является наличие в исходных текстах нескольких ошибок, в результате чего только высококвалифицированные программисты после исправления этих ошибок могут использовать компоненту в своих разработках. К достоинствам можно отнести простоту применения и освоения компоненты. При правильном использовании и небольших объемах данных продукты на базе этой компоненты могут оказаться полезными и приемлемыми по быстродействию.

Pivot Table корпорации Microsoft поставляется в двух вариантах: как составная часть MS Excel и как Web-компонента. Web-компонента (ActiveX) может быть использована как в браузере, так и собственном Windows-приложении. Pivot Table является одновременно и MOLAP- и ROLAP-компонентой. По протоколу OLE DB for OLAP он может взаимодействовать с многомерной СУБД MS OLAP Server, или другими 70-ю многомерными СУБД, разработчики которых поддержали этот протокол. По протоколу OLE DB Pivot Table может получать данные от реляционной СУБД и выполнять вычисления кубов в памяти. И конечно данные могут быть получены из заданной области таблицы MS Excel. В этом случае его производительность не отличается от производительности Decision Cube. Компонента отображает в таблице одновременно только один факт. Однако инструментарий компоненты шире, чем у Decision Cube - реализована произвольная фильтрация и раскрытие одного элемента измерения. Основным назначением компоненты является

создание интерфейсов к OLAP-серверу в рамках концепции Business Intelligent корпорации Microsoft.

Dynamic Cube компании Data Dynamic является классической ROLAP-компонентой. Он поставляется как VCL для программистов Delphi и C++ Builder и как COM для приверженцев компонентной модели. OLAP-машина компоненты весьма мощна. Она с легкостью обрабатывает десятки и чуть медленнее даже сотни тысяч записей. Есть множественная фильтрация, открытие элемента одного измерения, некоторые дополнительные функции. Компонента позволяет отображать в таблице одновременно несколько фактов. Однако эта компонента довольно дорога, особенно впечатляет ее стоимость для профессиональных разработчиков.

Все три описанные выше компоненты по сравнению с готовыми продуктами многих поставщиков имеют весьма скудную функциональность, ограничивающуюся классическими функциями OLAP: drill down, move, rotate и пр. В то же время в некоторых готовых продуктах часто встречается инструментальная панель, наполненная кнопками дополнительных удобных функций. Таких как <убрать нули>, <поменять строки со столбцами> и даже кнопками, выполняющими популярные аналитические задачи, например классический маркетинговый анализ 20/80.

Настольные OLAP-программы

Еще недавно поставщики OLAP-серверов продавали свои продукты по таким ценам, что их покупатели должны были быть богаты как арабские шейхи. Так, приобретение Oracle Express обошлось бы в \$100 000 за рабочие места двух аналитиков и двух администраторов. Но, даже после выхода на рынок компании Microsoft, которая обрушила цены, предоставив OLAP-сервер бесплатно в составе MS SQL Server, создание Хранилищ данных или витрин данных остается серьезным мероприятием, требующим привлечения профессионального разработчика, администрирования в процессе эксплуатации и других расходов.

Поэтому на рынке появился особый класс продуктов - DOLAP (Desktop OLAP) - настольный OLAP. Это программа, которая устанавливается на каждый персональный компьютер. Она не требует сервера, имеет "нулевое администрирование". Программа

позволяет пользователю настроиться на существующие у него базы данных; как правило, при этом создается словарь, скрывающий физическую структуру данных за ее предметным описанием, понятным специалисту. После этого программа выполняет произвольные запросы и результаты их отображает в OLAP-таблице. В этой таблице, в свою очередь, пользователь может манипулировать данными и получать на экране или на бумаге сотни различных отчетов.

По способу получения данных такие программы можно разделить на локальные и корпоративные:

- Локальные манипулируют данными таблицы MS Excel или небольших баз данных типа Access, DBF, Paradox.
- Корпоративные DOLAP имеют доступ к SQL-серверам или многомерным базам данных и, таким образом, тоже делятся на две категории.

Корпоративные DOLAP, предназначенные для анализа данных SQL-серверов позволяют анализировать уже имеющиеся в корпорации данные, хранящиеся в OLTP-системах. Однако вторым их назначением может быть быстрое и дешевое создание Хранилищ или витрин данных, когда программистам организации требуется лишь создать совокупности таблиц типа "звезда" и процедуры загрузки данных. Наиболее трудоемкая часть работы - разработка интерфейсов с многочисленными вариантами пользовательских запросов, интерфейсов и отчетов становится ненужной. Это буквально за несколько часов реализуется в DOLAP-программе. Освоение же такой программы конечным пользователем требует 30 минут.

DOLAP программы поставляются самими разработчиками баз данных, многомерных и реляционных. Это SAS Corporate Reporter, являющийся почти эталонным по удобству и красоте продуктом, Oracle Discover, комплекс программ MS Pivot Services и Pivot Table и другие. Эти продукты, за исключением программ Microsoft, стоят недешево. Так SAS Corporate Reporter обойдется в \$2000 на одного пользователя.

Большая группа программ поставляется в рамках компании "OLAP в массы", которую проводит корпорация Microsoft. Эти программы предназначены для работы с MS OLAP Services. Как

правило, они являются улучшенными вариантами Pivot Table и предназначены для использования в рамках MS Office или Web. Это Matryx, Knosys и т.д.

Благодаря простоте, дешевизне и огромной эффективности этот класс продуктов приобрел огромную популярность на Западе. Большие корпорации строят свои Хранилища с распределенным доступом на основе таких программ.

OLAP-продукты компании "Intersoft Lab"

Контур Стандарт

Основным продуктом компании "Intersoft Lab" является большая информационно-управленческая система "Контур Корпорация", построенная по принципам Хранилища данных. Однако в процессе общения с клиентами компании осознала, что далеко не все готовы на инвестиции и организационные мероприятия, связанные с построением серьезного Хранилища данных. Первым шагом на этом пути для многих банков и предприятий мог бы стать OLAP-анализ данных из имеющихся OLTP-систем и собственных аналитических баз данных.

Для этих целей был создан DOLAP-продукт "Контур Стандарт".

Контур Стандарт 1.0 Первая версия системы относилась к классу локальных DOLAP. Средства программы позволяли организовать прямой доступ к dbf- и paradox-файлам. Кроме того, в состав дистрибутивного пакета входил мигратор данных, который помогал собрать в локальные таблицы данные из имеющихся у организации систем.

Контур Стандарт 2.0 В дальнейшем, для расширения мощности продукта в системе "Контур Стандарт" 2.0 был обеспечен и доступ к произвольным SQL-серверам на уровне таблиц и, что не встречается в зарубежных аналогах, хранимых процедур. Это превратило программу в корпоративную информационно-аналитическую систему. Отдельно был реализован интерфейс к системе "Контур Корпорация".

Одновременно для удобства администрирования программа была разделена на две редакции. Редакция "Developer" позволяет IT-специалисту описать источники данных и выборки. При этом создаются семантические словари, которые скрывают от конечного

пользователя физический слой и переводят данные на язык предметной области. Редакция "Run-Time" позволяет анализировать данные и выпускать отчеты. Основным способом манипуляции данными является OLAP-компонента, которая позволяет без программирования и специальных навыков создавать необходимые отчеты. Одновременно были созданы и новые виды удобных аналитических инструментов, которые формально не являются OLAP-таблицами, но являются OLAP-средствами по духу, т.е. реализуют on-line анализ, но в другой форме представления данных.

В первых двух версиях применялась ROLAP-компонента Decision Cube компании Inprise. Однако ее невысокая мощность и функциональная упрощенность сдерживала применение программы в банках и организациях для анализа больших объемов данных. Поэтому было принято решение о ее замене. Маркетинговый анализ и ревизия интеллектуальных и производственных мощностей самой компании привели к решению о создании собственной OLAP-компоненты. В результате разработки компоненты, которую назвали Contour Cube, появилась следующая версия программы - "Контур Стандарт" 3.0, которая позволяет обрабатывать выборки данных до миллиона записей и обладает расширенной аналитической функциональностью.

Contour Cube

Компонента Contour Cube компании "Intersoft Lab" является представителем ROLAP-компонент. Она состоит из OLAP-машины, интерфейса доступа к данным, находящимся в SQL-серверах и других источниках, и визуальной части.

Компонента будет реализована в нескольких версиях для различных применений.

Версия VCL для использования в средах Delphi и C++ Builder компании Inprise. В этом случае данные поставляются через стандартный Data Set этих компиляторов. Доступ к источникам обеспечивается как при помощи BDE, так и ADO, поддерживаемой в последних версиях этих сред.

Версия COM предназначена для разработчиков на Visual Basic, Visual C++ и т.д. Она обеспечивает доступ к данным при

помощи ADO. В будущих версиях будет поддержан и доступ к OLAP-серверам через интерфейс OLE DB for OLAP.

Версия ActiveX является Web-компонентой для создания аналитических Интернет-интерфейсов в стиле, предложенном Microsoft.

Версия DHTML состоит из сервера и DHTML-страниц. Она предназначена для создания аналитических Интернет-интерфейсов в среде UNIX, а также для бурно развивающегося рынка мобильных Интернет-устройств.

Основными достоинствами компоненты являются:

- Обработка больших объемов данных.
- Минимальные требования к памяти.
- Расширенная функциональность.

Высокие характеристики компоненты достигнуты за счет уникальной математической модели, созданной специалистами компании.

Создание множества версий компоненты стало возможно благодаря ее многослойной архитектуре. Слой OLAP Engine является относительно независимой частью компоненты. Он реализован как кросс-платформенная библиотека, имеющая API для различных слоев визуализации. Этот API обладает функциями загрузки данных, вычисления срезов многомерного куба и выполнения аналитических и сервисных функций. Сам слой OLAP Engine состоит из машины вычислений и абстрактного многомерного Хранилища данных, которое может сохраняться в виде файла для передачи другим пользователям или длительного использования.

Обработка больших объемов данных

Тесты на персональном компьютере с процессором Intel Celeron 400 и оперативной памятью 64 Мб дали следующие результаты. Загрузка 60 000 записей с 6-ю измерениями занимает 5 секунд; дальнейшие манипуляции, такие как полный поворот таблицы, drill down и drill up выполняются за десятые доли секунды.

Загрузка 500 000 записей с 7-ю измерениями занимает около 30 секунд, поворот такого куба выполняется в среднем за 2.5 секунды.

Это лучшие по порядку величины (sic!) результаты из известных нам OLAP-компонент. Так, Decision Cube и Pivot Table (без

использования OLAP Services) требуют десятки секунд для загрузки и поворота таблицы объемом в 4000 записей и 6-ю измерениями. Скорость работы Dynamic Cube ниже, чем у Contour Cube, в среднем на 30% на средних объемах данных и в разы на предельных объемах.

Таким образом, во многих случаях благодаря своей мощности компонента делает необязательным использование OLAP-сервера. Это значительно упрощает процессы внедрения и администрирования корпоративной системы.

Минимальные требования к памяти

В момент работы с данными компонента занимает наименьший объем оперативной памяти по сравнению с одноклассниками. Так при загрузке 40 000 записей Contour Cube потребляет 7 МБ, Decision Cube 15 МБ.

Расширенная функциональность

В компоненте объединены функции лучших OLAP-компонент:

- Множественный фильтр по измерениям.
- Генерация как стандартных временных периодов ("Год", "Квартал", "Месяц", "Декада", "Неделя", etc.), так и задаваемых пользователем ("Финансовый год", "Сезон", "Время суток") по измерению типа "дата".
- Сортировка по измерениям.
- Сортировка по фактам.
- Открытие одного значения измерения (ветви).
- Автоматическое управление диаграммой.
- Ручная настройка диаграммы.
- Множество фактов.
- Множество стандартных алгоритмов агрегации фактов.
- Алгоритм агрегации "Остаток счета".

Уникальное свойство компоненты - алгоритм агрегации "Остаток счета". В связи с тем, что в основном OLAP-компоненты предназначены для анализа продаж и других суммирующих видов анализов, они агрегируют по времени и остатки счетов. Это является ошибкой - остаток счета за квартал не является суммой остатков счета за день, а является остатком за последний день квартала. Реализация этого алгоритма позволяет использовать

компоненту для анализа балансов и делает ее полезной не только для экономистов и маркетологов, но и для бухгалтеров.

Для того чтобы при использовании компоненты за минимальное время создавались мощные законченные продукты, в нее встроено много часто встречающихся в реальной работе аналитических функций. Каждая из этих функций реализована как кнопка в инструментальной панели визуальной части компоненты. Вот перечень этих функций:

- Удалить нулевые колонки, удалить нулевые строки, удалить нулевые колонки и строки. Применяется для сжатия разреженных таблиц.

- Полный поворот. При этом колонки и строки таблицы меняются местами. Применяется для улучшения восприятия таблиц аналитиком, для подбора лучшей печатной формы.

- Фильтр по факту. Позволяет задать абсолютные граничные значения факта или количество наибольших или наименьших элементов. Является одним из инструментов факторного анализа.

- Кластерный анализ. Разбиение данных на заданное количество групп по предельным значениям факта. Например, разбиение клиентов на крупных, средних и мелких по объемам полученных от них доходов.

- 80/20. Популярная на Западе разновидность кластерного анализа в маркетинге. Пример ее применения: показать 20% клиентов, которые приносят 80% прибыли.

- Ранжирование. Генерация нового измерения "место в списке" по значению заданного факта и сортировка по нему. Полезно для анализа избирательных компаний, сравнения банков, предприятий, филиалов по заданному показателю.

- Отображение одновременно нескольких статистических итогов, таких как среднее, среднее квадратическое отклонение и т.д. Эта функция понравится продвинутому специалисту, особенно в области финансового, фондового анализа.

- Выгрузка в форматы MS Excel, MS Word, html. Позволяют продолжить анализ привычными средствами MS Excel, создать отчет произвольной формы, опубликовать отчет в Интернет.

В связи с невозможностью защиты авторских прав в России на программные продукты, физическая защита которых

принципиально не реализуема, компонента как коммерческий продукт будет поставляться только на Западный рынок. Однако российские потребители могут воспользоваться ее достоинствами для развития собственного бизнеса в продуктах "Контур Стандарт" и "Контур Корпорация"

3.3. ROLAP-серверы

ROLAP-серверы используют реляционные БД. По словам Кодда, "реляционные БД были, есть и будут наиболее подходящей технологией для хранения данных."

Необходимость существует не в новой технологии БД, а скорее в средствах анализа, дополняющих функции существующих СУБД, и достаточно гибких, чтобы предусмотреть и автоматизировать разные виды интеллектуального анализа, присущие OLAP".

В настоящее время распространены две основные схемы реализации многомерного представления данных с помощью реляционных таблиц: схема «звезда» (рис.3.3) и схема «снежинка» (рис. 3.4).

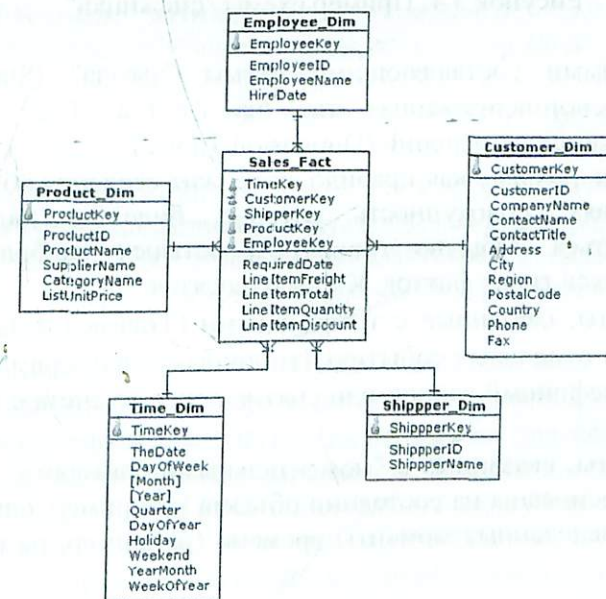


Рисунок 3.3. Пример схемы «звезда»

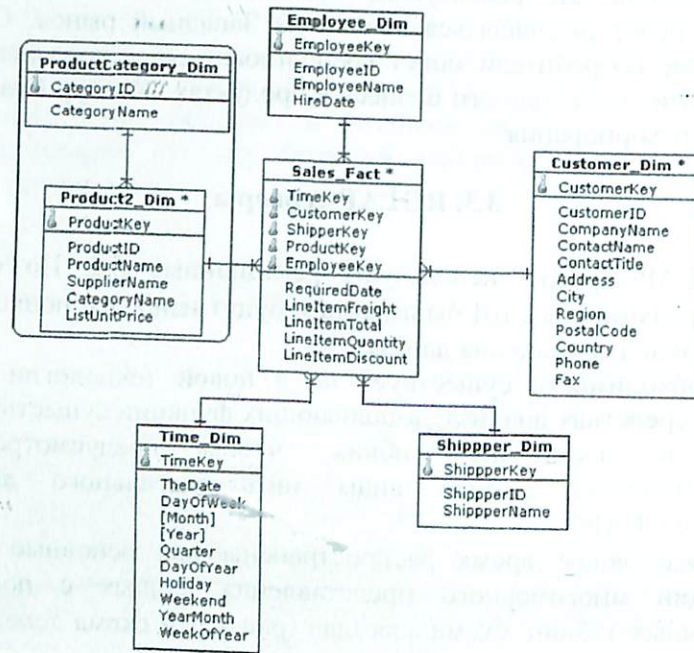


Рисунок 3.4. Пример схемы "снежинка"

Основными составляющими схемы "звезда" (StarSchema) являются денормализованная таблица фактов (Fact Table) и множество таблиц измерений (Dimension Tables).

Таблица фактов, как правило, содержит сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться. Обычно говорят о четырех наиболее часто встречающихся типах фактов. К ним относятся:

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

- факты, связанные с "моментальными снимками" (Snapshot facts). Они основаны на состоянии объекта (например, банковского счета) в определенные моменты времени (например, на конец дня

или месяца). Типичными примерами таких фактов является объем продаж за день или дневная выручка;

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

- факты, связанные с событиями или состоянием объекта (Event or state facts).

Они представляют возникновение события без подробностей о нем (например, просто факт продажи или факт отсутствия таковой без иных подробностей).

Таблица фактов, как правило, содержит уникальный составной ключ, объединяющий первичные ключи таблиц измерений. При этом как ключевые, так и некоторые не ключевые поля должны соответствовать измерениям гиперкуба. Помимо этого таблица фактов содержит одно или несколько числовых полей, на основании которых в дальнейшем будут получены агрегатные данные.

Для многомерного анализа пригодны таблицы фактов, содержащие как можно более подробные данные, т. е. соответствующие членам нижних уровней иерархии соответствующих измерений. 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, используемые в дальнейшем для построения иерархий в измерениях куба, содержатся в таблицах измерений.

Таблицы измерений содержат неизменяемые или редко изменяемые данные. В подавляющем большинстве случаев эти данные представляют собой по одной записи для каждого члена нижнего уровня иерархии в измерении. Таблицы измерений также содержат как минимум одно описательное поле (обычно с именем члена измерения) и, как правило, целочисленное ключевое поле (обычно это суррогатный ключ) для однозначной идентификации

члена измерения. Если измерение, соответствующее таблице, содержит иерархию, то такая таблица также может содержать поля, указывающие на "родителя" данного члена в этой иерархии. Каждая таблица измерений должна находиться в отношении «один-ко-многим» с таблицей фактов.

Скорость роста таблиц измерений должна быть незначительной по сравнению со скоростью роста таблицы фактов. Например, новая запись в таблицу измерений, характеризующую товары, добавляется только при появлении нового товара, не продававшегося ранее.

В сложных задачах с иерархическими измерениями имеет смысл обратиться к расширенной схеме "снежинка" (Snowflake Schema). В этих случаях отдельные таблицы фактов создаются для возможных сочетаний уровней обобщения различных измерений (см. рис. 3.3). Это позволяет добиться лучшей производительности, но часто приводит к избыточности данных и к значительным усложнениям в структуре базы данных, в которой оказывается огромное количество таблиц фактов.

Увеличение числа таблиц фактов в базе данных определяется не только множественностью уровней различных измерений, но и тем обстоятельством, что в общем случае факты имеют разные множества измерений. При абстрагировании от отдельных измерений пользователь должен получать проекцию максимально полного гиперкуба, причем далеко не всегда значения показателей в ней должны являться результатом элементарного суммирования. Таким образом, при большом числе независимых измерений необходимо поддерживать множество таблиц фактов, соответствующих каждому возможному сочетанию выбранных в запросе измерений, что также приводит к неэкономному использованию внешней памяти, увеличению времени загрузки данных в БД схемы «звезды» из внешних источников и сложностям администрирования.

Использование реляционных БД в OLAP-системах имеет следующие достоинства:

- в большинстве случаев корпоративные хранилища данных реализуются

средствами реляционных СУБД, и инструменты ROLAP позволяют производить анализ не посредственно над ними. При этом размер хранилища не является таким критичным параметром, как в случае MOLAP;

- в случае переменной размерности задачи, когда изменения в структуру

измерений приходится вносить достаточно часто, ROLAP-системы с динамическим представлением размерности являются оптимальным решением, т. к. в них такие модификации не требуют физической реорганизации БД;

- реляционные СУБД обеспечивают значительно более высокий уровень

защиты данных и хорошие возможности разграничения прав доступа.

Главный недостаток ROLAP по сравнению с многомерными СУБД — меньшая производительность. Для обеспечения производительности, сравнимой с MOLAP, реляционные системы требуют тщательной проработки схемы базы данных и настройки индексов, т. е. больших усилий со стороны администраторов БД. Только при использовании схем типа "звезда" производительность хорошо настроенных реляционных систем может быть приближена к производительности систем на основе многомерных баз данных.

3.4. HOLAP-серверы

HOLAP-серверы используют гибридную архитектуру, которая объединяет технологии ROLAP и MOLAP. В отличие от MOLAP, которая работает лучше, когда данные более-менее плотные, серверы ROLAP показывают лучшие параметры в тех случаях, когда данные сильно разрежены. Серверы HOLAP применяют подход ROLAP для разреженных областей многомерного пространства и подход MOLAP для плотных областей. Серверы HOLAP разделяют запрос на несколько подзапросов, направляют их к соответствующим фрагментам данных, комбинируют результаты, а затем предоставляют результат пользователю.

Достоинствами MOLAP являются высокая производительность и простота использования встроенных функций.

Достоинствами ROLAP являются возможность работы с существующими реляционными БД, более экономичное использование ресурсов и большая гибкость при добавлении новых измерений.

3.5. Применение OLAP систем

На данное время разработан довольно много аналитических систем, сконструированных с использованием OLAP-технологии (Hyperion OLAP, Elite OLAP, Oracle Express и много других). Рынок программных OLAP-продуктов постоянно расширяется. Современные системы оперативной аналитической обработки дают пользователям возможность решать ключевые задачи управления бизнесом-процессом, в частности прикладные программы Hyperion OLAP разрешают выполнять анализ прибыльности; анализ направлений развития продукции; анализ продажи; анализ положения на рынке; анализ ассортимента продуктов; анализ риска; анализ конкурентоспособности; складывания отчетов из производительности; моделирования сценария; анализ бюджета и прогнозов и т.п.

Следует отметить, что в соответствии с современными взглядами на создание информационных систем OLAP-системы должны базироваться на специальной базе данных — ХД.

В OLAP реализуется сложный интеллектуальный анализ данных. Возможность использования хорошо зарекомендовавших себя методов математической статистики, нейронных сетей, машинного обучения, визуализации данных, индукции правил, нечеткой логики, генетических алгоритмов и др. для решения задач подобного рода открыло новые возможности перед аналитиками, исследователями, а также теми, кто принимает решения — менеджерами и руководителями компаний. Сложность и разнообразие методов обработки данных требуют создания специализированных средств конечного пользователя для решения типовых задач анализа информации в конкретных областях.

Поскольку эти средства используются в составе сложных многофункциональных систем поддержки принятия решений, они должны легко интегрироваться в подобные системы. Другими словами, вся эта сложная математика заботливо скрыта от конечного пользователя дружелюбным интерфейсом.

В OLAP-системах основное внимание уделено гибкости доступа и манипулирования информацией. На сегодняшний день большинство разработок в этой области относится к финансовой сфере, хотя возможно применение таких систем в различных областях деятельности человека.

Одним из первых пользователей технологии интеллектуального анализа данных стало федеральное правительство США. Оно применяет с 1996 года специализированное программное обеспечение, для выявления случаев уклонения от уплаты налогов и для обработки материалов перехвата информационного обмена других государств.

По оценкам специалистов, очень перспективно применение подобных систем и для решения задач защиты информации. Например, анализ подлинности электронных платежей, выявление случаев неоправданной «активности» легальных пользователей в сети до совершения ими нарушений, анализ действенности принятой политики безопасности и т.д.

OLAP в отличие от прочих способов автоматизации бизнес-деятельности дает возможность получить пользователю «на выходе» не готовое четко структурированное решение, выдаваемое после включения ранее настроенного мастера обработки форм, а своеобразный материал для творческой оценки существующей ситуации. Поэтому сфера применения OLAP-анализа обычно ограничивается менеджерским составом предприятий разных размеров, которому приходится часто заниматься тактическими и стратегическими задачами вроде анализа ключевых показателей деятельности и сценариев развития, маркетинговым и финансово-экономическим анализом групп товаров или услуг, а также долгосрочным прогнозированием работы предприятия или его подразделений.

Для этого пользователь OLAP-систем получает в руки мощный и главное очень гибкий инструмент создания различных

отчетов по выбираемым им же разрезам и направлениям. При этом методики OLAP куда совершеннее привычных электронных таблиц, ведь помимо простых функций создания таблиц, графиков и диаграмм, OLAP-системы дают возможность получить обобщенные данные по самостоятельно выбранным критериям, моментально углубиться в детали выбранных направлений, отфильтровать, сортировать или отбросить ненужные цифры или показатели.

Пример.

Если менеджеру продаж компании требуется получить сезонные сводки динамики продаж выбранной категории товаров, система предложит ему всевозможные данные о продажах за месяц, квартал, год, а также найдет и проанализирует их зависимость от означенных факторов, скажем, времени проведения маркетинговых акций.

Кроме того, базируясь на одной лишь статистике продаж, OLAP-система может выявить эффективность работы разных подразделений компании, в том числе и в разрезе географической иерархии их взаимодействия. При этом параметры, характеризующие успешность подразделений, выбираются менеджером самостоятельно и в ряде случаев могут стать инструментом мотивации успешного персонала.

Ниже перечислены наиболее важные сферы применения OLAP-технологий.

Сферы применения OLAP-технологий:

Продажи

Ключевой вопрос отдела сбыта любого предприятия: "Какое количество изделий продано?", "На какую сумму реализовано?" расширяются по мере усложнения бизнеса и накопления исторических данных до некоторого множества факторов, или разрезов: регион поставок, в прошлом месяце, квартале, по сравнению с нынешним, через канал сбыта А, по сравнению с каналом Б и т.д. Ответы на подобные вопросы необходимы для принятия управленческих решений: об изменении ассортимента, цен, закрытии и открытии филиалов, расторжении и подписании договоров с дилерами, проведения или прекращения рекламных кампаний и т.д.

Закупки

Задача обратно противоположная анализу продаж. Многие предприятия закупают комплектующие и материалы у поставщиков. Торговые предприятия закупают товары для перепродажи. Возможных задач при анализе закупок множество, от планирования денежных средств на основе прошлого опыта, до контроля размеров поставок различных поставщиков.

Маркетинг

Под маркетинговым анализом имеется ввиду только область анализа покупателей или клиентов-потребителей услуг. Например, для розничной торговли задачей анализа является правильное позиционирование товара, выявление групп покупателей для целевой рекламы, оптимизация ассортимента. Например, если выясняется, что телефонами темно-серого цвета стоимостью более \$500 пользуются исключительно мужчины старше 25 лет, то стоит изобразить в рекламе таких телефонов вместо девушек одного преуспевающего бизнесмена. Это очень грубый пример, но известно, что маркетинговый анализ находится на грани между сложной наукой и малообъяснимым искусством. Поэтому задача OLAP в данном случае - дать пользователю инструмент быстрого получения ответов на вопросы, интуитивно возникающие по ходу анализа данных.

Движение денежных средств

Могут анализироваться денежные обороты безналичных и наличных средств в разрезе бизнес-операций, контрагентов, валют и времени с целью оптимизации потоков, обеспечения ликвидности, и т.д. Состав измерений сильно зависит от особенностей бизнеса, отрасли и т.д.

Бюджет

Одна из самых перспективных областей применения OLAP-технологий - ни одна современная система бюджетирования не считается завершенной без наличия в ее составе OLAP-инструментария для анализа бюджета. Большинство бюджетных отчетов легко строятся на основе OLAP-систем. При этом отчеты отвечают на очень широкую гамму вопросов: анализ структуры расходов и доходов, сравнение расходов по определенным статьям

у разных подразделений, анализ динамики и тенденций расходов на определенные статьи, анализ себестоимости и прибыли.

Финансовая отчетность

Технологично построенная система отчетности есть ни что иное, как набор именованных показателей со значениями на дату, которые требуется сгруппировать и просуммировать в различных разрезах для получения конкретных отчетов. Когда это так, то отображение и печать отчетов наиболее просто и дешево реализуются в OLAP-системах. Некоторые страны уже перешли на такую технологию сбора данных. В некоторых отечественных контролирующих органах существуют планы перехода от ГОСТовских стандартов отчетов с многостажными шапками и алгоритмами типа "Итого, исключая строку 234 и включая строку 598 из отчета №987" к системе к сбору показателей и выпуску отчетов по OLAP-технологии.

Результаты социологических опросов

Возможно, что в этой области существуют тонкие нюансы, которые следует знать при решении конкретных задач, но в первом приближении кажется, что OLAP исключительно хорошо подходит для представления и анализа результатов социологических опросов.

Объемы производства

Это еще один пример статистического анализа. Таким образом, можно анализировать объемы выращенного картофеля, выплавленной стали, сваренного пива и пр.

Потребление расходных материалов

Например, имеется завод, состоящий из десятков цехов, в которых расходуются охлаждающие, промывочные жидкости, масла, ветошь, наждачная бумага - сотни наименований расходных материалов. Для точного планирования, оптимизации издержек требуется тщательный анализ фактического потребления расходных материалов.

Заработная плата

Анализ расходов на зарплату, сравнение расходов по специальностям, филиалам, людям, динамика фонда ЗП.

Текучесть кадров на предприятии

Анализ текучести кадров в отделе, цехе, филиале по предприятию в целом в разрезе профессий, уровня образования, пола, возраста, времени.

Пассажирские перевозки

Анализ количества проданных билетов и сумм в разрезе сезонов, направлений, видов вагонов (классов), типов поездов (самолетов).

Грузовые перевозки

Анализ объемов перевозок, платы в разрезе сезонов, направлений, видов вагонов, грузов, грузоотправителей, грузополучателей, станций отправления, станций получения.

Простои транспорта (вагонов, самолетов, паромов, грузовиков)

Анализ времени простоя (полученных штрафов) в разрезе причин (ремонт, отказ от погрузки-разгрузки), клиентов, исполнителей, железнодорожных станций (вокзалов, гаражей, аэропортов), типов грузов.

Заболеваемость персонала (учащихся, трудящихся)

Измерения - категории сотрудников (граждан), регион, отдел, профессия, возраст, дата. Факт - количество дней в году, в течение которых люди были не-трудоспособны.

Выбор недвижимости (офисов, складов, квартир)

Измерения - обычные для этого рынка. Город, Район, Количество комнат, Расстояние до метро, Этаж, Тип дома, Дата и т.д. Фактов три - средняя цена, максимальная цена, минимальная цена. Манипулируя измерениями, покупатель может определиться со своими возможностями, а продавец проанализировать зависимости цен, динамику цен и назначить правильную цену.

Урожайности агрокультур

Измерения - Сорт, Регион, Почва, Удобрение, Средняя температура воздуха, уровень осадков, Год, другие факторы, влияющие на урожайность. Факт - количество центнеров с гектара. Анализ позволит выявить лучшие сорта для данного региона, лучшие удобрения для данной почвы и прочие зависимости.

Контрольные вопросы

1. Какие данные называются многомерными данными?
2. Основные компоненты OLAP систем?
3. Архитектура OLAP систем?
4. Способы реализации многомерной модели?
5. Преимущества использования многомерных БД в OLAP-системах?
6. Перечислите условия использования многомерных БД является эффективным?
7. Инструменты ROLAP серверов?
8. Какую архитектуру используют HOLAP-серверы?
9. Архитектура MOLAP систем?
10. Архитектура JOLAP систем?

ГЛАВА IV. ЛОГИЧЕСКИЕ МОДЕЛИ ХРАНИЛИЩ ДАННЫХ

Ядром любого ХД является его модель данных. Без модели данных будет очень сложно организовать данные в ХД. Поэтому разработчики ХД должны потратить время и силы на разработку такой модели. Разработка модели ХД ложится на плечи проектировщика ХД.

По сравнению с проектированием OLTP-систем, методика проектирования ХД имеет ряд отличительных особенностей, связанных с ориентацией структур данных хранилища на решение задач анализа и информационной поддержки процесса принятия решений. Модель данных ХД должна обеспечивать эффективное решение именно этих задач.

Отправной точкой в проектировании ХД может служить так называемая корпоративная модель данных (corporate data model или enterprise data model, EDM), которая создается в процессе проектирования OLTP-систем организации. При проектировании корпоративной модели данных обычно предпринимается попытка создать на основе бизнес-операций такую структуру данных, которая бы собрала и синтезировала в себе все информационные потребности организации.

Таким образом, корпоративная модель данных содержит в себе необходимую информацию для построения модели ХД. Поэтому на первом этапе, если такая модель есть в организации, проектировщик ХД может начать проектирование ХД с решения задачи преобразования корпоративной модели данных в модель ХД.

4.1. Хранилище данных

Хранилище данных — это разновидность системы управления данными, которая обеспечивает поддержку бизнес-аналитики. Хранилища данных предназначены только для выполнения запросов и анализа и обычно содержат большие объемы исторических данных. Данные обычно поступают в хранилище из самых различных источников, таких как журналы приложений и приложения транзакций.

Хранилище данных служит для централизации и консолидации больших объемов данных из различных источников. Аналитические инструменты дают возможность организациям извлекать из собственных данных ценные для бизнеса сведения и повышать эффективность принятых решений. Со временем в хранилище накапливаются записи за прошедшие периоды, которые представляют большую ценность для специалистов по изучению данных и бизнес-аналитиков. Эти возможности делают хранилища данных "единым источником проверенных данных компании."

Обычно хранилище данных включает в себя следующие компоненты:

- реляционную базу данных для хранения данных и управления ими;
- решение для извлечения, загрузки и преобразования данных, которое служит для подготовки данных к анализу;
- средства статистического анализа, отчетности и глубинного анализа данных;
- инструменты анализа для визуализации данных и их представления для корпоративных пользователей;
- другие, более сложные приложения для обработки сведений и разработки рекомендаций.

Хранилища данных обеспечивают для компаний обширные преимущества, так как дают возможность анализировать большие объемы разнообразных данных, извлекать из них значительную ценность, а также хранить записи за прошедшие периоды.

В основе создания хранилища данных лежат две основные идеи.

1. Сбор (интеграция) поступающих из оперативных баз данных, внешних источников, архивов корпорации детализированных данных в едином хранилище с их согласованием и агрегацией.
2. Разделение наборов данных и приложений, используемых для оперативной обработки и применяемых для решения задач анализа.

Существует два типа ХД: виртуальное и физическое.

Создание ХД из независимых источников представляет собой многоэтапный процесс, включающий извлечение, преобразование и очистку, загрузку.

Извлечение данных представляет собой перенесение данных из разнородных источников в базу данных, где их можно модифицировать и добавить в хранилище.

Преобразование – устранение несоответствий в схеме данных и в соглашениях о значениях атрибутов. Например, в оперативной базе данных имя клиента хранится в трех полях: Фамилия, Имя, Отчество, а в хранилище – в одном (Имя клиента).

Очистка данных необходима для устранения аномалий и дублирования.

Преобразованные и очищенные данные при загрузке подвергаются дополнительной обработке: проверке на целостность, сортировке, агрегированию и индексированию.

Эффективное взаимодействие различных компонентов ХД реализуется с помощью *метаданных* – информации о структуре, размещении и трансформации данных. Различают *административные* метаданные (информация, требуемая для настройки и использования ХД), *бизнес-метаданные* (бизнес-термины и определения), а также *оперативные* метаданные (информация о происхождении перенесенных и преобразованных данных, статус и статистика использования данных и др.).

Таким образом, данные в ХД – это надежный источник, на основе которого строятся аналитические информационные системы.

Основными преимуществами такой организации ХД можно считать простоту, невысокую стоимость реализации, единую платформу с источником информации, отсутствие сетевых соединений между источником информации и ХД. А серьезными проблемами является значительное время обработки запросов, интеграция данных с другими источниками, отсутствие истории, чистота данных, зависимость от доступности и структуры основной базы данных.

4.2. Характеристики хранилищ данных

Эти уникальные преимущества доступны благодаря четырем отличительным особенностям хранилищ данных, которые описал

специалист по вычислительным системам Уильям Инмон (William Inmon). Согласно данному им определению, хранилища данных имеют следующие характеристики.

• **Субъектно-ориентированность.** Хранилища можно использовать для анализа данных, которые относятся к одной теме или функциональной области (например, продажи).

• **Единообразие.** Хранилища данных обеспечивают целостность данных различных типов, полученных из разных источников.

• **Неизменность.** Элементы данных, помещенные в хранилище данных, не подвергаются изменениям.

• **Изменения во времени.** Анализ данных, помещенных в хранилище данных, предназначен для выявления изменений в закономерностях, возникающих со временем.

Хорошо спроектированное хранилище данных обеспечивает быстрое выполнение запросов, эффективное прохождение больших объемов данных и достаточный уровень гибкости, чтобы конечные пользователи могли формировать «продольные и поперечные» срезы данных или уменьшать их объем для более подробного изучения, то есть обеспечивает соответствие самым различным — потребностям в изучении данных как на высшем, так и на самом низовом уровне. Хранилища данных служат функциональной основой для промежуточных сред бизнес-аналитики, которые предоставляют конечным пользователям доступ к отчетам, панелям мониторинга и прочим элементам интерфейса.

Согласно классическому определению Б. Инмона, хранилище данных представляет предметно-ориентированный, интегрированный, привязанный ко времени и неизменяемый набор данных, предназначенный для поддержки принятия решений.

Перечисленные характеристики принципиально отличают хранилище данных от базы данных OLTP-системы. Рассмотрим эти характеристики подробнее.

Предметная ориентированность предполагает, что данные в хранилище данных объединяются в категории и хранятся в соответствии с областями, которые они описывают (продажи, бюджетные поступления и т.п.). Реализуется предметная ориентированность через использование особых схем организации

данных. Это позволяет упростить создание аналитических запросов, а также увеличить скорость их выполнения. Предметная ориентированность данных в хранилище данных отличается от организации данных в базе данных OLTP-систем, группирующихся в соответствии с бизнес-операциями (выписка счетов, отгрузка товара и т.п.), оптимизированных на интенсивное оперативное выполнение операций вставки/обновления/удаления небольших порций данных; имеющих, как правило, нормализованную реляционную структуру баз данных; удовлетворяющих жестким требованиям ссылочной целостности данных. Интегрированность означает, что данные для анализа не берутся напрямую из источников, в частности, баз данных OLTP-систем предприятия. Исходные данные извлекаются, проверяются, очищаются, унифицируются и т.п., чтобы удовлетворять требованиям аналитика, исследующего не отдельную бизнес-функцию, а деятельность всего предприятия.

Процесс ETL позволяет повысить качество данных аналитической системы и скорость выполнения аналитических запросов к хранилищу данных. Привязка ко времени подразумевает, что данные в хранилище всегда жестко «привязаны» к определенному периоду времени. Данные, выбранные из баз данных OLTP-систем других источников, накапливаются в хранилище в виде исторических слоев, каждый из которых относится к конкретному периоду времени. Это позволяет анализировать тенденции в развитии бизнеса. Не изменимость хранилища, данные уже никогда не будут изменены. Стабильность данных способствует достоверности результатов аналитических запросов.

Архитектура хранилища данных зависит от потребностей организации. Наиболее распространенными типами архитектур являются следующие.

• **Простая.** Все хранилища данных имеют общую конструкцию, где метаданные, сводные данные и необработанные данные хранятся в центральном репозитории хранилища. В репозиторий поступают данные из источников, к которым затем

получают доступ конечные пользователи для выполнения, анализа, составления отчетов и изучения.

- **Простая архитектура с областью подготовки.** Перед помещением в хранилище операционные данные должны пройти процедуру очистки и обработки. Это можно сделать программным способом, однако во многих хранилищах данных есть специальная область, где данные проходят обработку перед поступлением непосредственно в хранилище.

- **Основное и дополнительные хранилища.** Добавление витрин данных между центральным репозиторием и конечными пользователями дает возможность организациям использовать хранилища данных для обслуживания различных направлений бизнеса. Когда данные готовы к использованию, их помещают в соответствующую витрину.

- **«Песочницы».** «Песочницы» представляют собой безопасные частные и защищенные области, в которых компании могут быстро изучать новые наборы данных или способы анализа без необходимости обеспечивать соответствие формальным правилам и протоколам хранилища данных.

Это также отличает хранилища данных от баз данных OLTP-систем. В OLTP-системах данные все время меняются, и один и тот же аналитический запрос, выполненный дважды с интервалом в 10 минут, может дать разные результаты, приводя к несогласованным управленческим решениям. Таким образом, хранилище можно охарактеризовать как особую базу данных, содержащую данные—масштаба организации;—собранные из различных источников за длительный период времени, очищенные и консолидированные;—привязанные ко времени;—структурированные в целях упрощения выполнения аналитических запросов.

4.3. Модель данных хранилища

Модель данных хранилища может быть представлена на трех уровнях:

- концептуальной модели данных;
- логической модели данных;

- физической модели данных.

Концептуальная модель является отражением предметной области, в рамках которой планируется построение хранилища данных, а также спектра аналитических задач, для решения которых разрабатывается хранилище.

Логическая модель расширяет концептуальную, уточняя состав таблицы взаимосвязи между ними, добавляя колонки, а также определения для таблицы колонок.

Физическая модель описывает реализацию объектов логической модели на уровне объектов конкретной базы данных. Концептуальная модель данных хранилища В основе концептуальной модели хранилища лежит многомерная модель данных. Основными понятиями многомерной модели данных являются: куб (гиперкуб) данных, факты, показатели(меры), измерения, иерархии, агрегаты, срез. Физическое ХД является репозиторием информации из оперативных источников данных. Позволяет только читать детальные и агрегированные данные, которые полностью очищены и интегрированы.

Поскольку многомерная модель данных предназначена для предоставления BI-информации, все показатели, измерения, иерархии должны иметь названия, доступные для понимания лицам, принимающим решения в организации. Кроме того, модель предусматривает возможность сопровождать названия объектов развернутыми описаниями. Это позволяет ответственному за принятие решений, удостовериться, что он работает с требуемой аналитической информацией.

4.4. Логическая хранилище данных

Концептуальная модель хранилища (гиперкуб) на логическом уровне представляется с помощью схемы «Звезда» или ее варианта —схемы «Снежинка».

Схема «звезда» состоит только из таблиц размерности и таблицы факта. Она представлена на рис. 4.1. и 4.2. Если каждое измерение содержится в одной таблице, такая схема хранилища данных носит название «звезда» (starschema). Развитием схемы «звезда» является схема «снежинка». Ее отличает от первой схемы

большое количество консольных таблиц, они имеются практически на каждой таблице размерности и могут иметь несколько уровней иерархии, как показано на рис. 4.2. и 4.6.

Говоря об измерениях, следует упомянуть о том, что значения, могут иметь различные уровни детализации. Например, нас может интересовать суммарная стоимость заказов, сделанных клиентами в разных странах, либо суммарная стоимость заказов, сделанных иногородними клиентами или даже отдельными клиентами. Естественно, результирующий набор агрегатных данных во втором и третьем случаях будет более детальным, чем в первом. Заметим, что возможность получения агрегатных данных с различной степенью детализации соответствует одному из требований, предъявляемых к хранилищам данных, — требованию доступности различных срезов данных для сравнения и анализа.

Схема «созвездие» получается из нескольких таблиц фактов. В этом варианте многомерной модели через консольные или таблицы размерности сообщаются несколько таблиц фактов, отображающих несколько объектов с общими атрибутами.

В схемах «снежинка» и «созвездие» применение консольных таблиц приводит к дополнительным затратам времени на реализацию запроса. При проектировании этот фактор должен учитываться. При создании многомерных моделей на основе реляционной базы данных рекомендуют "создавать длинные и узкие таблицы фактов и сравнительно небольшие и широкие таблицы размерности (измерений).

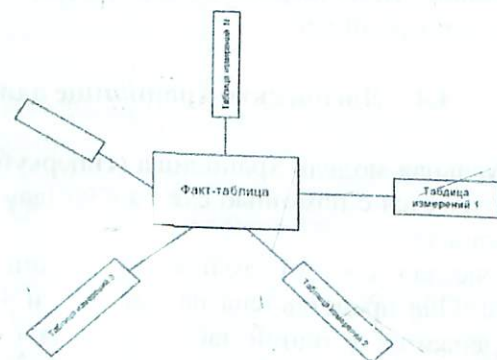


Рис.4.1. Модель данных по схеме «звезда»

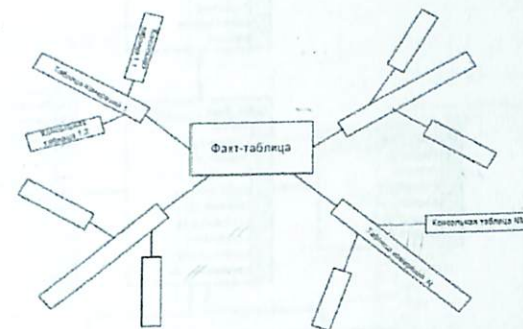


Рис.4.2. Модель данных по схеме «снежинка»

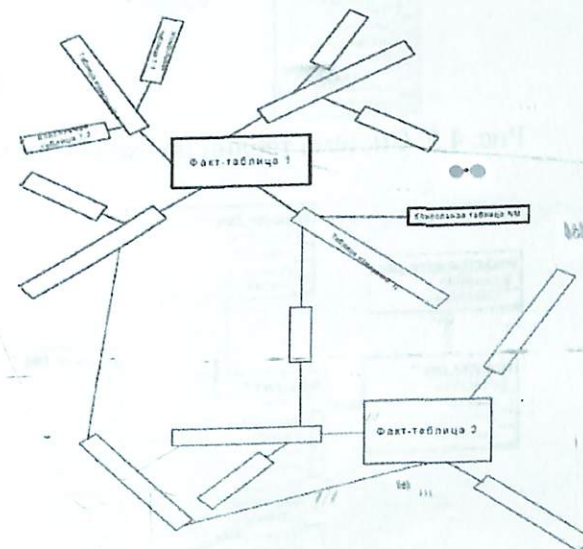


Рис.4.3. Модель данных по схеме «созвездие»

Далее на рисунках 4.5 приведены примеры реализации названных моделей в среде СУБД MS SQL Server.

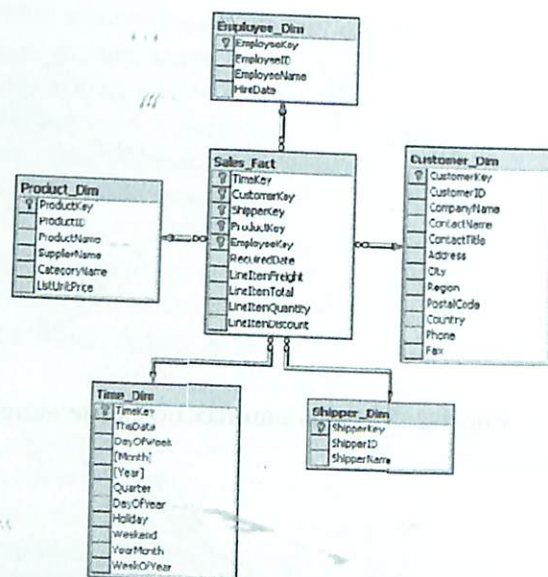


Рис. 4.5. Система таблиц по схеме «звезда»

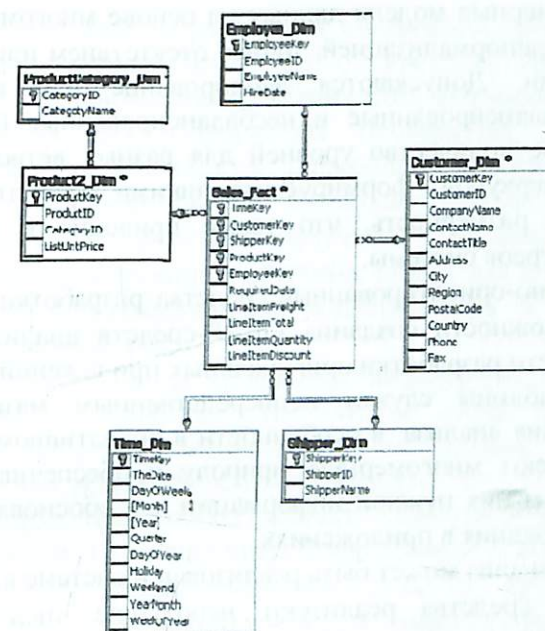
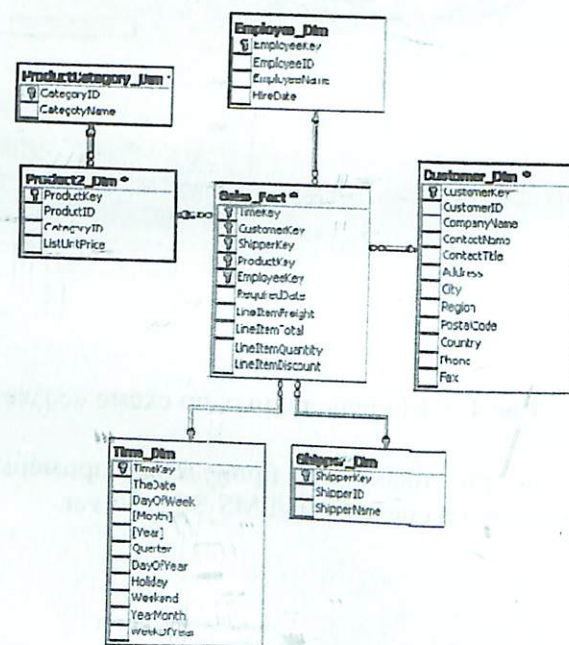


Рис. 4.6. Система таблиц по схеме «Снежинка»

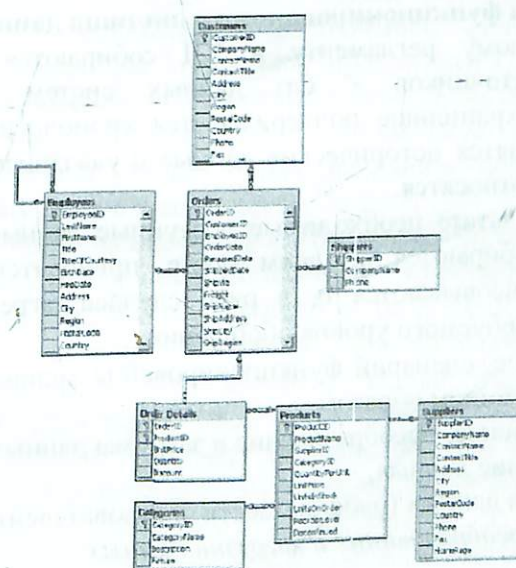


Рис. 4.7. Система таблиц по схеме «созвездие»

Многомерные модели данных на основе многомерных СУБД отличаются денормализацией, точнее отсутствием или неполнотой нормализации. Допускаются дублирование или избыточность данных, сбалансированные и несбалансированные (включающие неодинаковое количество уровней для разных ветвей) иерархии. Ячейки гиперкубов, формируемые такими средствами, имеют одинаковую размерность, что также приводит к избыточному расходу ресурсов системы.

Объектно-ориентированные средства разработки приложений имеют возможности создания OLAP-средств анализа в случаях необходимости разработки оригинальных приложений.

Факт-таблицы служат непосредственным материалом для осуществления анализа, в особенности в оперативном режиме, так как они имеют многомерную природу и обеспечивают быстрое извлечение из них нужной информации для обоснования решений или использования в приложениях.

OLAP-анализ может быть реализован в системе клиент-сервер. Клиентские средства реализуют небольшое число измерений. Возможны различные варианты распределения функций между серверной и клиентской частями ИАС в OLAP-анализе.

Сценарий функционирования хранилища данных

По заданному регламенту в ХД собираются данные из различных источников — баз данных систем оперативной обработки. В хранилище поддерживается хронология: наравне с текущими хранятся исторические данные с указанием времени, к которому они относятся.

В результате необходимые доступные данные об объекте управления собираются в одном месте, приводятся к единому формату, согласовываются и, в ряде случаев, агрегируются до минимально требуемого уровня обобщения.

В общем виде, сценарий функционирования хранилища данных выглядит следующим образом.

1. Извлечение, преобразование и загрузка данных.
2. Хранение данных.
3. Анализ данных (рабочие места пользователей).

Извлечение, преобразование и загрузка данных

Данные поступают из различных внутренних транзакционных систем, от подчиненных структур, от внешних организаций в соответствии с установленным регламентом, формами и макетами отчетности. Вся эта информация проверяется, согласуется, преобразуется и помещается в хранилище и витрины данных. После этого пользователи с помощью специализированных инструментальных средств получают необходимую им информацию для построения различных табличных и графических представлений, прогнозирования, моделирования и выполнения других аналитических задач.

В ходе наполнения хранилища информация подвергается многоступенчатой обработке. На первом этапе производится буферизация сведений, поступающих из разных источников информации, что позволяет стабилизировать процесс загрузки и выполнять сверку присланных данных. Первую буферную зону называют шлюзом. На следующем этапе данные преобразуются и перемещаются из одной буферной зоны в другую. Здесь устраняется структурная неоднородность данных (унифицируются длины полей, преобразуются форматы и пр.), проверяется их целостность, служебным полям хранилища присваиваются соответствующие значения, решаются задачи интеграции данных. На этом этапе производится также очистка информации. Сведения, не прошедшие процедуры преобразования, очистки или проверки, попадают в специальную зону, где их в полуавтоматическом режиме можно исправить, дополнить, после чего заново загрузить в хранилище.

Из источников данных информация перемещается на основе некоторого регламента в централизованное хранилище (более того, отдельные таблицы, могут обновляться по разному расписанию: ежедневно, еженедельно, ежемесячно, либо не обновляться вовсе). Часто, необходимые для анализа данные не хранятся в окончательном виде ни в одной из транзакционных систем. Эти данные либо можно получить из исходных баз данных путем специальных преобразований, вычислений и агрегирования, либо они могут быть загружены в хранилище в ручном или полуавтоматическом режиме. Таким образом процесс настройки хранилища данных подразумевает выполнение следующих шагов:

внешней памяти, называемые чаще страничными системами хранения информации. Поэтому наш раздел, посвященный физическим моделям данных, мы начнем с обзора файлов и файловых структур, используемых для организации физических моделей, применяемых в базах данных, а в конце ознакомимся с механизмами организации данных во внешней памяти, использующими страничный принцип организации.

Физические модели данных описывают то, как данные хранятся в компьютере, представляя информацию о структуре записей, их упорядоченности и существующих путях доступа. Физических моделей данных не так много, как логических, а самыми популярными среди них являются обобщающая модель (unifying model) и модель памяти кадров (frame memory).

Иерархическая модель является ограниченным подтипом сетевой модели. В ней данные также представлены как коллекции записей, а связи — как наборы. Однако в иерархической модели узел может иметь только одного родителя. Иерархическая модель может быть представлена как древовидный граф с записями в виде узлов (которые также называются сегментами) и множествами в виде ребер.

Сетевая модель данных — модель, состоящая из записей, элементов данных и связей типа “один ко многим” (1:M), установленных между записями.

В сетевой модели данные представлены в виде коллекций записей, а связи — в виде наборов. В отличие от реляционной модели, связи здесь явным образом моделируются наборами, которые реализуются с помощью указателей. Сетевую модель можно представить как граф с записями в виде узлов графа и наборами в виде его ребер.

Объектно-ориентированная база данных — база данных, в которой данные оформлены в виде моделей объектов, включающих прикладные программы, которые управляются внешними событиями. Результатом совмещения возможностей (особенностей) баз данных и возможностей объектно-ориентированных языков программирования являются Объектно-ориентированные системы управления базами данных (ООСУБД). ООСУБД позволяет работать с объектами баз данных также, как с объектами в

программировании на ООЯП. ООСУБД расширяет языки программирования, прозрачно вводя долговременные данные, управление параллелизмом, восстановление данных, ассоциированные запросы и другие возможности.

Постреляционная модель данных представляет собой расширенную реляционную модель, снимающую ограничение неделимости данных, хранящихся в записях таблиц. Постреляционная модель данных допускает многозначные поля — поля, значения которых состоят из подзначений. Набор значений многозначных полей считается самостоятельной таблицей, встроенной в основную таблицу.

Помимо обеспечения вложенности полей постреляционная модель поддерживает ассоциированные многозначные поля (множественные группы). Совокупность ассоциированных полей называется ассоциацией. При этом в строке первое значение одного столбца ассоциации соответствует первым значениям всех других столбцов ассоциации. Аналогичным образом связаны все вторые значения столбцов и т.д.

На длину полей и количество полей в записях таблицы не накладывается требование постоянства. Это означает, что структура данных и таблиц имеет большую гибкость.

Поскольку постреляционная модель допускает хранение в таблицах ненормализованных данных, возникает проблема обеспечения целостности и непротиворечивости данных. Эта проблема решается включением в СУБД механизмов, подобных хранимым процедурам в клиент-серверных системах.

Для описания функций контроля значений в полях имеется возможность создавать процедуры (коды конверсии и коды корреляции), автоматически вызываемые до или после обращения к данным. Коды корреляции выполняются сразу после чтения данных, перед их обработкой. Коды конверсии, наоборот, выполняются после обработки данных.

Достоинством постреляционной модели является возможность представления совокупности связанных реляционных таблиц одной постреляционной таблицей. Это обеспечивает высокую наглядность представления информации и повышение эффективности ее обработки.

Недостатком постреляционной модели является сложность решения проблемы обеспечения целостности и непротиворечивости хранимых данных.

Объектная модель данных разрабатывается в виде модели сущностных классов – концептуальный уровень.

Модели данных:

- реляционная;
- иерархическая;
- сетевая.

Физическая модель данных ориентирована на СУБД (Physical Data Model).

Объекты – классы;

Реляционная модель – таблицы (логические);

Физическая модель – Data definition language (DDL).

При проектировании происходит трансформация моделей: $a \rightarrow b \rightarrow c$.

Переход к реляционной модели данных называется трансформацией объектной модели в реляционную модель данных.

Это соответствует переходу к более детальному рассмотрению структуры базы данных и к формированию физической модели базы данных.

Сама по себе база данных рассматривается как компонента. Физически она реализуется в виде файла СУБД, либо в виде файла DDL. Эта модель называется компонентной моделью данных.

Физическая модель данных, напротив, зависит от конкретной СУБД, фактически являясь отображением системного каталога. В физической модели содержится информация о всех объектах БД. Поскольку стандартов на объекты БД не существует (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей. Если в логической модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах – таблицах, колонках, индексах, процедурах и т. д. Разделение модели данных на логические и физические позволяет решить несколько важных задач.

Документирование модели. Многие СУБД имеют ограничение на именование объектов (например, ограничение на длину имени таблицы или запрет использования специальных символов – пробела и т. п.). Зачастую разработчики ИС имеют дело с нелокализованными версиями СУБД. Это означает, что объекты БД могут называться короткими словами, только латинскими символами и без использования специальных символов (т. е. нельзя назвать таблицу предложением – только одним словом). Кроме того, проектировщики БД нередко злоупотребляют "техническими" наименованиями, в результате таблица и колонки получают наименования типа *RTD_324* или *CUST_A12* и т. д. Полученную в результате структуру могут понять только специалисты (а чаще всего только авторы модели), ее невозможно обсуждать с экспертами предметной области. Разделение модели на логическую и физическую позволяет решить эту проблему. На физическом уровне объекты БД могут называться так, как того требуют ограничения СУБД. На логическом уровне можно этим объектам дать синонимы – имена более понятные неспециалистам, в том числе на кириллице и с использованием специальных символов. Например, таблице *CUST_A12* может соответствовать сущность *Постоянный клиент*. Такое соответствие позволяет лучше задокументировать модель и дает возможность обсуждать структуру данных с экспертами предметной области.

Масштабирование. Создание модели данных, как правило, начинается с создания логической модели. После описания логической модели, проектировщик может выбрать необходимую СУБД и ERwin автоматически создаст соответствующую физическую модель. На основе физической модели ERwin может сгенерировать системный каталог СУБД или соответствующий SQL-скрипт. Этот процесс называется прямым проектированием (Forward Engineering). Тем самым достигается масштабируемость – создав одну логическую модель данных, можно сгенерировать физические модели под любую поддерживаемую ERwin СУБД. С другой стороны, ERwin способен по содержимому системного каталога или SQL-скрипту воссоздать физическую и логическую модель данных (Reverse Engineering). На основе полученной логической модели данных можно сгенерировать физическую

модель для другой СУБД и затем сгенерировать ее системный каталог. Следовательно, ERwin позволяет решить задачу по переносу структуры данных с одного сервера на другой. Например, можно перенести структуру данных с Oracle на Informix (или наоборот) или перенести структуру dbf-файлов в реляционную СУБД, тем самым облегчив решение по переходу от файл-серверной к клиент-серверной ИС. Заметим, однако, что формальный перенос структуры "плоских" таблиц на реляционную СУБД обычно неэффективен. Для того чтобы извлечь выгоды от перехода на клиент-серверную технологию, структуру данных следует модифицировать. Процессы прямого и обратного проектирования будут рассмотрены ниже.

Основными объектами логической модели данных являются сущности, атрибуты и взаимосвязи. Физическая модель данных, как правило, создается на основе логической, поэтому каждому объекту логической модели соответствует объект физической модели (хотя соответствие может быть неоднозначным). В физической модели данных сущности логической модели данных соответствует таблица, экземпляру сущности — строка в таблице, а атрибуту — колонка таблицы. Кроме перечисленных выше объектов, физическая модель может содержать объекты, тип которых зависит от СУБД: индексы, представления, последовательности, триггеры, процедуры и т.п. Если в логической модели данных не имеет большого значения, какой конкретно тип данных у атрибута, то в физической важно описать всю информацию о конкретных объектах.

Далее, при изложении материала, мы будем предполагать, что имеем дело с реляционными или объектно-реляционными СУБД и соответствующими им диалектами SQL.

Можно выделить два этапа создания физической модели данных. Основными целями первого этапа являются:

- удовлетворение потребности в хранении данных предметной области в рамках реляционной модели данных, т.е. должны быть созданы базовые таблицы для хранения информации обо всех сущностях предметной области;
- удовлетворение требования целостности данных, т.е. должны быть определены типы колонок и наложены ограничения

на значения колонок базовых таблиц, которые бы следовали из бизнес-правил предметной области;

- удовлетворение требования **ссылочной целостности** (referential integrity, RI), т.е. в случае принятия решения о поддержке ссылочной целостности встроенными средствами СУБД должны быть наложены ограничения ссылочной целостности на таблицы, исходя из бизнес-правил ссылочной целостности предметной области;

- удовлетворение (частично) требования независимости представления данных для конечного пользователя от характера физического хранения данных.

На первом этапе в рамках требований реляционной модели создаются объекты хранения данных, соответствующие сущностям и взаимосвязям логической модели данных — таблицы, индексы, представления и т.д.

Главной целью второго этапа является обеспечение требуемого уровня производительности. Для достижения этой цели необходимо учитывать как особенности реализации СУБД, для которой создается физическая модель, так и особенности функционирования будущей информационной системы в целом. Обычно производительность БД измеряется в терминах **производительности транзакций** (transaction performance).

Для повышения производительности транзакций могут быть модифицированы объекты, созданные на первом этапе, или созданы новые объекты БД. Другими словами, разработка физической модели представляет собой итерационный процесс, причем итераций (создание объектов — анализ транзакций — модификация объектов — анализ транзакций) может быть несколько. Вопросы повышения производительности транзакций будут рассмотрены в одной из следующих лекций.

5.2. Иерархия объектов реляционной базы данных

Одной из главных задач, которые обязан решить проектировщик на стадии проектирования физической модели ХД, является задача превращения объектов логической модели данных в объекты реляционной БД. Для решения этой задачи

проектировщику необходимо знать: а) какими объектами располагает реляционная база данных в принципе; б) какие объекты поддерживает конкретная СУБД, которая выбрана для реализации базы данных.

Таким образом, мы предполагаем, что решение о выборе СУБД уже принято руководителем ИТ-проекта и согласовано с заказчиком базы данных, т.е. СУБД задана. Проектировщик ХД должен ознакомиться с документацией, в которой описан диалект SQL, поддерживаемый выбранной СУБД. В настоящей лекции предполагается, что была выбрана СУБД семейства MS SQL Server компании Microsoft, хотя подавляющая часть материала охватывает объекты в любой промышленной реляционной СУБД.

Иерархия объектов реляционной БД прописана в стандартах по SQL, в частности, в стандарте SQL-92, на который мы будем ориентироваться при изложении материала настоящей лекции. Этот стандарт поддерживается практически всеми современными СУБД. Иерархия объектов БД показана на рис. 5.1.

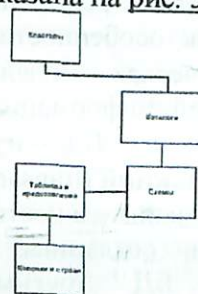


Рис. 5.1. Иерархия объектов реляционной базы данных, соответствующая стандарту SQL-92.

На самом нижнем уровне находятся объекты, с которыми работает реляционная БД — столбцы (колонки) и строки. Они, в свою очередь, группируются в таблицы и представления. Заметим, что в контексте лекции атрибуты, колонки, столбцы и поля считаются синонимами. То же относится и к терминам «строка», «запись» и «кортеж». Таблицы и представления, которые представляют физическое отражение логической структуры БД, собираются в схему. Несколько схем собираются в каталоги, которые затем могут быть сгруппированы в кластеры. Следует

отметить, что ни одна из групп объектов стандарта SQL-92 не связана со структурами физического хранения информации в памяти компьютеров. Помимо указанных на рисунке объектов в реляционной базе данных могут быть созданы индексы, триггеры, события, хранимые команды, хранимые процедуры и ряд других. Теперь перейдем к определению объектов БД.

5.3. Основные объекты реляционной базы данных

Кластеры, каталоги и схемы не являются обязательными элементами стандарта и, следовательно, программной среды БД.

Под кластером понимается группа каталогов, к которым можно обращаться через одно соединение с сервером базы данных (программная компонента СУБД). Обычно процедура создания каталога определяется реализацией СУБД на конкретной операционной платформе. Под каталогом понимается группа схем. На практике каталог часто ассоциируется с физической базой данных как набором физических файлов операционной системы, которые идентифицируются ее именем.

Для проектировщика БД схема — это общее логическое представление отношений законченной базы данных. С точки зрения SQL, схема — это контейнер для таблиц, представлений и других структурных элементов реляционной базы данных. Принцип размещения элементов БД в каждой схеме полностью определяется проектировщиком. На практике схема часто ассоциируется с объектами определенного пользователя — владельца физической БД.

Основными компонентами (объектами) СУБД являются таблицы, запросы, формы, отчеты, макросы и модули.

Таблица — фундаментальная структура системы управления реляционными базами данных. Таблица — это объект, предназначенный для хранения данных в виде записей (строк) и полей (столбцов). При этом каждое поле содержит отдельную часть записи (например, фамилию, должность и т.д.). Обычно каждая таблица используется для хранения сведений по одному направлению описания предметной области (например, в сотрудниках).

Запрос – вопрос о данных, хранящихся в таблицах, или инструкция на отбор записей, подлежащих выборке, обновлению, удалению. СУБД Access позволяет создавать статические и параметрические запросы, определяющие условия отбора во время генерации; запросы могут конструироваться или программироваться на языке SQL.

Форма – это объект СУБД, предназначенный для ввода, отображения и изменения данных в полях таблиц. На формах могут размещаться элементы управления.

Отчет – это объект СУБД, который позволяет представлять определенную пользователем информацию в требуемом виде, просматривать, распечатывать ее.

Макрос – одна или несколько макрокоманд, которые можно использовать для автоматизации конкретной задачи. Макрокоманда – основной строительный блок макроса; самостоятельная инструкция, которая может быть объединена с другими макрокомандами для автоматизации выполнения задачи.

Модуль – набор описаний, инструкций и процедур, сохраненных под одним именем. В Microsoft Access имеется три типа модулей: модуль формы, модуль отчета и общий. Модули форм и отчетов содержат локальную программу для форм или отчетов. Если процедуры общего модуля явным образом не объявлены личными для модуля, в котором они размещены, то они распознаются и могут вызываться процедурами из других модулей.

Страница – специальный тип веб-страниц, предназначенный для просмотра и работы с данными, хранящимися в базе данных через Интернет или корпоративную сеть. Страница доступа к данным может также включать данные из других источников, таких как электронные таблицы.

Для упрощения идентификации и именования объектов в базе данных поддерживаются такие объекты, как синоним, последовательность и определенные пользователем типы данных.

Синоним (Synonym) – это альтернативное имя объекта (псевдоним) реляционной базы данных, которое позволяет иметь доступ к данному объекту. Синоним может быть общим и частным. Общий синоним позволяет всем пользователям базы данных обращаться к соответствующему объекту по его псевдониму.

Синоним позволяет скрыть от конечных пользователей полную квалификацию объекта в базе данных.

Последовательность (Sequence) – это объект базы данных, который позволяет генерировать последовательность уникальных чисел (номеров) в условиях многопользовательского асинхронного доступа. Обычно элементы последовательности используются для уникальной нумерации элементов таблиц (строк) в операциях модификации данных.

Определенные пользователем типы данных (User-defined data types) представляют собой определенные пользователем типы атрибутов (домены), которые отличаются от поддерживаемых (встроенных) СУБД типов. Они определяются на основе встроенных типов. Определенные пользователем типы данных образуют ту часть среды СУБД, которая организована в соответствии с объектно-ориентированной парадигмой.

Для обеспечения эффективного доступа к данным в реляционных СУБД поддерживаются ряд других объектов: индекс, табличная область, кластер, секция.

Индекс (Index) – это объект базы данных, создаваемый для повышения производительности выборки данных и контроля уникальности первичного ключа (если он задан для таблицы). Полностью индексные таблицы (index-organized tables) исполняют роль таблицы и индекса одновременно.

Табличное пространство или область (Tablespace) – это именованная часть базы данных, используемая для распределения памяти для таблиц и индексов. В Oracle 9i – это логическое имя физических файлов операционной системы. Все объекты базы данных, в которых хранятся данные, соответствуют некоторым табличным пространствам. Большинство объектов базы данных, в которых данные не хранятся, находятся в словаре данных, расположенном в табличном пространстве SYSTEM.

Кластер (Cluster) – это объект, задающий способ совместного хранения данных в нескольких или одной таблице. Одним из критериев использования кластера является наличие общих ключевых полей в нескольких таблицах, которые используются в одной и той же команде SQL. Обычно кластеризованные столбцы

или таблицы хранятся в базе данных в виде таблиц хэширования (т.е. специальным образом).

Секция (Partition) - это объект базы данных, который позволяет представить объект с данными в виде совокупности подобъектов, отнесенных к различным табличным пространствам. Таким образом, секционирование позволяет распределять очень большие таблицы на нескольких жестких дисках.

Для обработки данных специальным образом или для реализации поддержки ссылочной целостности базы данных используются объекты: хранимая процедура, функция, команда, триггер, таймер и пакет (Oracle). С помощью этих объектов базы данных можно выполнять так называемую построчную обработку (record processing) данных. С точки зрения приложений баз данных построчная обработка - это последовательная выборка данных по одной строке, ее обработка и переход к обработке следующей строки.

Данные объекты реляционной базы данных представляют собой программы, т.е. исполняемый код. Этого код обычно называют серверным кодом (server-side code), поскольку он выполняется компьютером, на котором установлено ядро реляционной СУБД. Планирование и разработка такого кода является одной из задач проектировщика реляционной базы данных.

Хранимая процедура (Stored procedure) - это объект базы данных, представляющий поименованный набор команд SQL и/или операторов специализированных языков обработки программирования базы данных (например, SQLWindows или PL/SQL).

Функция (Function) - это объект базы данных, представляющий поименованный набор команд SQL и/или операторов специализированных языков обработки программирования базы данных, который при выполнении возвращает значение - результат вычислений.

Команда (Command) - это поименованный оператор SQL, который заранее откомпилирован и сохраняется в базе данных. Скорость обработки команды выше, чем у соответствующего ему

оператора SQL, т.к. при этом не выполняются фазы синтаксического разбора и компиляции.

Триггер (Trigger) - это объект базы данных, который представляет собой специальную хранимую процедуру. Эта процедура запускается автоматически, когда происходит связанное с триггером событие (например, до вставки строки в таблицу).

Таймер (Timer) отличается от триггера тем, что запускающим событием для хранимой процедуры является событие таймера.

Пакет (Package) - это объект базы данных, который состоит из поименованного структурированного набора переменных, процедур и функций.

В распределенных реляционных СУБД имеются специальные объекты: снимок и связь базы данных.

Снимок (Snapshot) - локальная копия таблицы удаленной базы данных, которая используется для тиражирования (репликации) таблицы или результата запроса. Снимки могут быть модифицируемыми или предназначенными только для чтения.

Связь базы данных (Database Link) или связь с удаленной базой данных - это объект базы данных, который позволяет обратиться к объектам удаленной базы данных. Имя связи базы данных, грубо говоря, можно представить как ссылку на параметры доступа к удаленной базе данных.

Для эффективного управления разграничением доступа к данным в Oracle поддерживает объект роль.

Роль (Role) - объект базы данных, представляющий собой поименованную совокупность привилегий, которые могут назначаться пользователям, категориям пользователей или другим ролям.

Контрольные вопросы

1. Основные объекты логической модели данных?
2. Перечислите этапы создания физической модели данных?
3. Дайте характеристику к иерархии объектов реляционной базы данных?
4. Какие основные объекты реляционной базы данных вы знаете?
5. Как проектируется схема БД?

ГЛАВА VI. СОВРЕМЕННЫЕ ПРОГРАММНЫЕ СРЕДСТВА ДЛЯ АНАЛИЗА БОЛЬШИХ ОБЪЕМОВ ИНФОРМАЦИИ

Изначально в совокупность подходов и технологий включались средства массово-параллельной обработки неопределённо структурированных данных, такие как СУБД No SQL, алгоритмы Map Reduce и средства проекта Hadoop. В дальнейшем к технологиям больших данных стали относить и другие решения, обеспечивающие сходные по характеристикам возможности по обработке сверхбольших массивов данных, а также некоторые аппаратные средства.

6.1. Технологии и тенденции работы с Big data

Под большими данными понимается широкое разнообразие массивов данных, которые не могут быть надлежащим образом обработаны традиционными приложениями из-за своего огромного объема или сложного состава. Сложность анализа больших данных заключается в специфике их сбора, курирования, разделения, хранения, передачи, визуализации и сохранении конфиденциальности информации.

Рассмотрим, как используются технологии обработки больших данных в научных исследованиях. При помощи Большого адронного коллайдера со 150 миллионов датчиков снимались данные 40 миллионов раз в секунду. Это около 600 миллионов столкновений в секунду. После фильтрации и удаления более 99,99995% из них осталось 100 столкновений в секунду, которые были изучены исследователями.

В результате, после изучения менее чем 0,001% данных, полученных от датчиков, объем данных со всех четырех экспериментов с использованием Большого адронного коллайдера составил 25 петабайт в год до репликации

(по данным на 2012 год) или почти 200 петабайт после репликации.

Если бы фиксировались все данные, полученные от датчиков, установленных в Большом адронном коллайдере, то такой объем данных было бы чрезвычайно сложно обработать. Объем потока данных превысил 150 миллионов петабайт в год или 500 экзабайт в

день до репликации. Это эквивалентно 500 квинтильонам (5×10^{20}) байт в день, что почти в 200 раз превышает объем данных, получаемых от всех других ресурсов в мире.

Под анализом больших данных понимается как анализ массивов данных в рамках возможностей персонального компьютера, так и в рамках возможностей систем управления реляционными базами данных, при этом как в первом, так и во втором случае при формировании и статистики, и визуализации возникают определенные трудности, которые заключаются в необходимости обеспечения скоординированной работы компьютерных программ на десятках, сотнях или даже тысячах серверов.

Анализ больших данных может быть охарактеризован по следующим параметрам:

Объем, т.е. количество генерируемых данных. От этого показателя зависит, может ли определенный массив данных считаться большими данными или нет. Данных хранятся SQL-серверах в облачной среде.

1. Многообразие, т.е. категория, к которой принадлежат большие данные. Знание такой принадлежности позволяет аналитикам наиболее эффективно работать с информацией.

2. Скорость, т.е. скорость генерирования или обработки данных с целью осуществления поставленных целей.

3. Изменчивость, т.е. нестабильность данных во времени.

4. Достоверность, т.е. качество собранных данных, от которого зависит точность анализа.

5. Сложность, т.е. трудоемкость процесса корреляции и построения взаимосвязей между данными.

В 2004 году Google опубликовал статью, посвященную процессу MapReduce. Каркас MapReduce использует модель параллельной обработки очень большого объема данных. В рамках MapReduce запросы разделяются и распределяются по параллельным узлам, при этом они обрабатываются параллельно (этап распределения (Map)). Затем осуществляется сбор и доставка результатов (этап предварительного преобразования (Reduce)). Этот каркас был весьма эффективен, поэтому другие компании также захотели скопировать алгоритм. Каркас MapReduce был использован в

проекте Apache с открытым исходным кодом, носящем название Hadoop.

Последние исследования показали, что многоуровневая архитектура может успешно использоваться для анализа больших данных. Архитектура распределенной параллельной обработки распределяет данные по многочисленным узлам, которые, производя, соответственно, параллельную обработку, генерируют результат гораздо быстрее, чем другие системы. В рамках такой архитектуры данные передаются в параллельные СУБД с использованием каркасов MapReduce и Hadoop. При этом задействуются фронтальные серверы приложений.

Краткий обзор инструментов big data. Учитывая огромные объемы информации, которые необходимо хранить, обрабатывать в процессе работы, следует заметить, что подобные манипуляции не могут выполняться на простых жестких дисках. А программное обеспечение, которое структурирует и анализирует накапливаемые данные — это отдельная интеллектуальная собственность и в каждом отдельном случае является авторской разработкой. При этом можно отметить наиболее популярные на сегодняшний день инструменты, на основе которых создаются такие решения: — Hadoop & MapReduce; — NoSQL базы данных; — Инструменты класса Data Discovery. Анализу особенностей и отличий перечисленных инструментов, а также обзору решений, которые могут предлагаться на основе данных инструментариев, может быть посвящена отдельная статья. Но, в качестве примера, хотелось бы привести модель, которая, пожалуй, является на сегодняшний день одним из лидеров на рынке — это Oracle Big Data Appliance X5-2. Ориентировочная стоимость такой системы в максимальной комплектации может достигать 30 миллионов рублей за 1 стойку. Конечно, речь идет о промышленных системах премиум класса. Тем не менее, приведенный пример позволяет оценить порядок расходов, которые потребуются на реализацию подобных решений в компании. И это еще без учета узкоспециализированных специалистов и дополнительной ИТ инфраструктуры. Поэтому говорить о применении больших данных, например, в малом бизнесе не приходится.

• **MapReduce** — модель распределенных параллельных вычислений в компьютерных кластерах, представленная компанией Google. Согласно этой модели приложение разделяется на большое количество одинаковых элементарных заданий, выполняемых на узлах кластера и затем естественным образом сводимых в конечный результат.

• **NoSQL** (от англ. Not Only SQL, не только SQL) — общий термин для различных нереляционных баз данных и хранилищ, не обозначает какую-либо одну конкретную технологию или продукт. Обычные реляционные базы данных хорошо подходят для достаточно быстрых и однотипных запросов, а на сложных и гибко построенных запросах, характерных для больших данных, нагрузка превышает разумные пределы и использование СУБД становится неэффективным.

• **Hadoop** — свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределенных программ, работающих на кластерах из сотен и тысяч узлов. Считается одной из основополагающих технологий больших данных.

• **R** — язык программирования для статистической обработки данных и работы с графикой. Широко используется для анализа данных и фактически стал стандартом для статистических программ.

• **Аппаратные решения.** Корпорации Teradata, EMC и др. предлагают аппаратно-программные комплексы, предназначенные для обработки больших данных. Эти комплексы поставляются как готовые к установке телекоммуникационные шкафы, содержащие кластер серверов и управляющее программное обеспечение для массово-параллельной обработки. Сюда также иногда относят аппаратные решения для аналитической обработки в оперативной памяти, в частности, аппаратно-программные комплексы Hana компании SAP и комплекс Exalytics компании Oracle, несмотря на то, что такая обработка изначально не является массово-параллельной, а объемы оперативной памяти одного узла ограничиваются несколькими терабайтами.

6.2. Map Reduce программа

Программы, использующие Map Reduce, автоматически распараллеливаются и исполняются на распределенных узлах кластера, при этом исполнительная система сама заботится о деталях реализации (разбиение входных данных на части, разделение задач по узлам кластера, обработка сбоев и сообщение между распределенными компьютерами). Благодаря этому программисты могут легко и эффективно использовать ресурсы распределённых Big Data систем.

Технология практически универсальна: она может использоваться для индексации веб-контента, подсчета слов в большом файле, счётчиков частоты обращений к заданному адресу, вычисления объём всех веб-страниц с каждого URL-адреса конкретного хост-узла, создания списка всех адресов с необходимыми данными и прочих задач обработки огромных массивов распределенной информации. Также к областям применения Map Reduce относится распределённый поиск и сортировка данных, обращение графа веб-ссылок, обработка статистики логов сети, построение инвертированных индексов, кластеризация документов, машинное обучение и статистический машинный перевод. Также Map Reduce адаптирована под многопроцессорные системы, добровольные вычислительные, динамические облачные и мобильные среды.

История развития главной технологии BigData

Авторами этой вычислительной модели считаются сотрудники Google Джеффри Дин (Jeffrey Dean) и Санджай Гемават (Sanjay Ghemawat), взявшие за основу две процедуры функционального программирования: map, применяющая нужную функцию к каждому элементу списка, и reduce, объединяющая результаты работы map. В процессе вычисления множество входных пар ключ/значение преобразуется в множество выходных пар ключ/значение.

Как устроен Map Reduce: принцип работы

Прежде всего, еще раз поясним смысл основополагающих функций вычислительной модели:

- **Map** принимает на вход список значений и некую функцию, которую затем применяет к каждому элементу списка и возвращает новый список;
- **reduce** (свёртка) – преобразует список к единственному атомарному значению при помощи заданной функции, которой на каждой итерации передаются новый элемент списка и промежуточный результат.

Для обработки данных в соответствии с вычислительной моделью Map Reduce следует определить обе эти функции, указать имена входных и выходных файлов, а также параметры обработки.

Сама вычислительная модель состоит из 3-хшаговой комбинации вышеприведенных функций [2]:

1. **Map** – предварительная обработка входных данных в виде большого списка значений. При этом главный узел кластера (masternode) получает этот список, делит его на части и передает рабочим узлам; (workernode). Далее каждый рабочий узел применяет функцию Map к локальным данным и записывает результат в формате «ключ-значение» во временное хранилище.

2. **Shuffle**, когда рабочие узлы перераспределяют данные на основе ключей, ранее созданных функцией Map, таким образом, чтобы все данные одного ключа лежали на одном рабочем узле.

3. **Reduce** – параллельная обработка каждым рабочим узлом каждой группы данных по порядку следования ключей и «склейка» результатов на master node. Главный узел получает промежуточные ответы от рабочих узлов и передаёт их на свободные узлы для выполнения следующего шага. Получившийся после прохождения всех необходимых шагов результат – это и есть решение исходной задачи.

Hadoop – это свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов. Эта основополагающая технология хранения и обработки больших данных (Big Data) является проектом верхнего уровня фонда Apache Software Foundation.

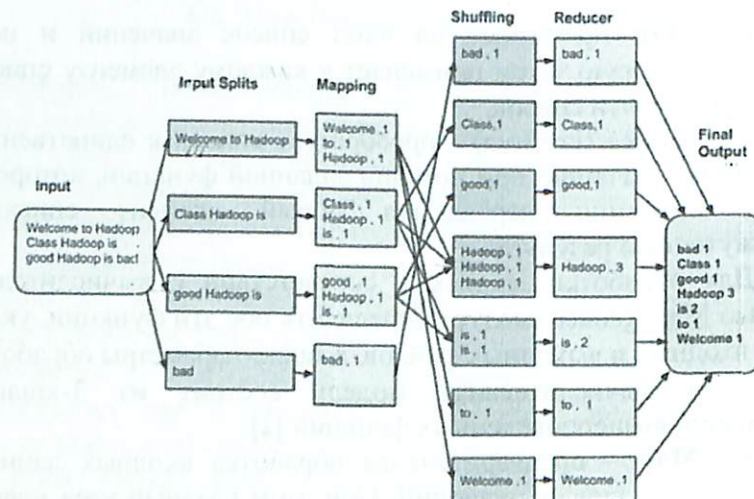


Рис. 6.1. Принцип работы Map Reduce

6.3. Концептуальная архитектура Hadoop

Изначально проект разработан на Java в рамках вычислительной парадигмы Map Reduce, когда приложение разделяется на большое количество одинаковых элементарных заданий, которые выполняются на распределенных компьютерах (узлах) кластера и сводятся в единый результат.

Проект состоит из основных 4-х модулей:

1. **Hadoop Common** – набор инфраструктурных программных библиотек и утилит, которые используются в других решениях и родственных проектах, в частности, для управления распределенными файлами и создания необходимой инфраструктуры;

2. **HDFS** – распределенная файловая система, Hadoop Distributed FileSystem – технология хранения файлов на различных серверах данных (узлах, Data Nodes), адреса которых находятся на специальном сервере имен (мастере, Name Node). За счет дублирования (репликации) информационных блоков, **HDFS** обеспечивает надежное хранение файлов больших размеров,

поблочно распределенных между узлами вычислительного кластера;

3. **YARN** – система планирования заданий и управления кластером (Yet Another Resource Negotiator), которую также называют Map Reduce 2.0 (MRv2) – набор системных программ (демонов), обеспечивающих совместное использование, масштабирование и надежность работы распределенных приложений. Фактически, YARN является интерфейсом между аппаратными ресурсами кластера и приложениями, использующих его мощности для вычислений и обработки данных;

4. **Hadoop Map Reduce** – платформа программирования и выполнения распределенных Map Reduce-вычислений, с использованием большого количества компьютеров (узлов, nodes), образующих кластер.

Сегодня вокруг Hadoop существует целая экосистема связанных проектов и технологий, которые используются для интеллектуального анализа больших данных (Data Mining), в том числе с помощью машинного обучения (Machine Learning).

6.4. Решение практических задач с помощью технологии обработки больших данных

Обработка больших данных сейчас предполагает, как правило, внедрение специальных программных комплексов, таких как, Hadoop, позволяющих производить обработку больших объемов данных на основании концепции Map-Reduce.

Hadoop на данный момент является «де-факто» стандартом обработки больших данных. Hadoop представляет собой фреймворк, на основе которого разрабатываются приложения для анализа и визуализации больших данных.

Хранение данных в данном фреймворке осуществляется с помощью специальной распределенной файловой системы HDFS (Hadoop Distributed File System), которая лежит в основе Hadoop и позволяет хранить и предоставлять доступ к данным сразу на нескольких узлах кластера. Таким образом, если один или несколько узлов кластера выходят из строя, то риск потери

информации сводится к минимуму и кластер продолжает работу в штатном режиме.

Процесс обработки данных в Hadoop-приложении

Для обработки больших данных используется алгоритм Map-Reduce, при этом все стадии Map должны завершить свою работу до начала Reduce. Также входные данные требуют предварительной обработки. Одной из самых актуальных задач современных информационных технологий является задача быстрой обработки больших объемов данных. Эффективное решение данной задачи, позволяет быстрее принимать решения на основе данных, полученных в прошлом.

Контрольные вопросы

1. Какие современные программные средства для анализа больших объемов информации знаете?
2. Как работает Map Reduce?
3. Принцип работы Hadoop?
4. Какие основные модули имеет Map Reduce?
5. Архитектурная концепция экосистемы Hadoop?
6. Что знаете о системе YARN?
7. Как работает распределённая файловая система?
8. Что такое Hadoop Common?

ГЛАВА VII. DATA MINING. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ

За последние годы, когда, стремясь к повышению эффективности и прибыльности бизнеса, при создании БД все стали пользоваться средствами обработки цифровой информации, появился и побочный продукт этой активности - горы собранных данных: И вот все больше распространяется идея о том, что эти горы полны золота. В прошлом процесс добычи золота в горной промышленности состоял из выбора участка земли и дальнейшего ее просеивания большое количество раз. Иногда искатель находил несколько ценных самородков или мог натолкнуться на золотonosную жилу, но в большинстве случаев он вообще ничего не находил и шел дальше к другому многообещающему месту или же вовсе бросал добывать золото, считая это занятие напрасной тратой времени.

7.1. Общие понятия о Data Mining

Сегодня появились новые научные методы и специализированные инструменты, сделавшие горную промышленность намного более точной и производительной. Data Mining для данных развилась почти таким же способом. Старые методы, применявшиеся математиками и статистиками, отнимали много времени, чтобы в результате получить конструктивную и полезную информацию. Сегодня на рынке представлено множество инструментов, включающих различные методы, которые делают Data Mining прибыльным делом, все более доступным для большинства компаний.

Часто интеллектуальный анализ данных рассматривают как трансформацию термина Data Mining (раскопка, разработка данных) и считают два этих понятия синонимами. В широком смысле это вполне допустимо. Однако считать их абсолютно идентичными было бы неверно, поскольку методы Data Mining сосредотачиваются главным образом на процессах анализа данных и интеллектуальном моделировании. В то же время считается, что

интеллектуальный анализ данных рассматривает весь спектр проблем, связанных с процессом извлечения знаний из баз данных.

В основе интеллектуального анализа данных, как и в Data Mining, лежит идея активного применения математических методов, таких как оптимизация, генетические алгоритмы, распознавание образов, статистика, Data Mining и т.д., а также использующих визуальное представление информации.

Однако интеллектуальный анализ данных также включает вопросы извлечения данных из различных источников, их консолидацию, профайлинг, трансформацию, предобработку, очистку и обогащение.

Модели интеллектуального анализа данных могут применяться в конкретных бизнес-сценариях, а именно:

- Прогнозирование: оценка продаж, прогнозирование нагрузки сервера или времени простоя сервера;
- Риск и вероятность: выбор наиболее подходящих заказчиков для целевой рассылки, определение точки равновесия для рискованных сценариев, назначение вероятностей диагнозам или другим результатам;
- Рекомендации: определение продуктов, которые с высокой долей вероятности могут быть проданы вместе, создание кросс-пакетов;
- Поиск последовательностей: анализ выбора заказчиков во время совершения покупок, прогнозирование следующего возможного события;
- Группировка: разделение заказчиков или событий на кластеры связанных элементов, анализ и прогнозирование общих черт.

Термин Data Mining получил свое название из двух понятий: поиска ценной информации в большой базе данных (data) и добычи горной руды (mining). Оба процесса требуют или просеивания огромного количества сырого материала, или разумного исследования и поиска искомых ценностей.

Термин Data Mining часто переводится как добыча данных, извлечение информации, раскопка данных, интеллектуальный анализ данных, средства поиска закономерностей, извлечение знаний, анализ шаблонов, "извлечение зерен знаний из гор

данных", раскопка знаний в базах данных, информационная проходка данных, "промывание" данных.

Понятие Data Mining, появившееся в 1978 году, приобрело высокую популярность в современной трактовке примерно с первой половины 1990-х годов. До этого времени обработка и анализ данных осуществлялся в рамках прикладной статистики, при этом в основном решались задачи обработки небольших баз данных.

Data Mining - мультидисциплинарная область, возникшая и развивающаяся на базе таких наук как прикладная статистика, распознавание образов, искусственный интеллект, теория баз данных и др.

Data Mining - это процесс поддержки принятия решений, основанный на поиске в данных скрытых закономерностей (шаблонов информации).

Технологию Data Mining достаточно точно определяет Григорий Пиатецкий-Шапиро (Gregory Piatetsky-Shapiro) - один из основателей этого направления:

Data Mining - это процесс обнаружения в сырых данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности.

Суть и цель технологии Data Mining можно охарактеризовать так: это технология, которая предназначена для поиска в больших объемах данных неочевидных, объективных и полезных на практике закономерностей.

Неочевидных - это значит, что найденные закономерности не обнаруживаются стандартными методами обработки информации или экспертным путем.

Объективных - это значит, что обнаруженные закономерности будут полностью соответствовать действительности, в отличие от экспертного мнения, которое всегда является субъективным.

Практически полезных - это значит, что выводы имеют конкретное значение, которому можно найти практическое применение.

Знания - совокупность сведений, которая образует целостное описание, соответствующее некоторому уровню осведомленности об описываемом вопросе, предмете, проблеме и т.д.

Приведем еще несколько определений понятия Data Mining.

Data Mining - это процесс выделения из данных неявной и неструктурированной информации и представления ее в виде, пригодном для использования.

Data Mining - это процесс выделения, исследования и моделирования больших объемов данных для обнаружения неизвестных до этих структур (patterns) с целью достижения преимуществ в бизнесе (определение SASInstitute).

Data Mining - это процесс, цель которого - обнаружить новые значимые корреляции, образцы и тенденции в результате просеивания большого объема хранимых данных с использованием методик распознавания образов плюс применение статистических и математических методов (определение Gartner Group).

7.2. Обзор и классификация методов Data Mining

С помощью методов интеллектуального анализа выполняется систематизация данных по критериям количества и качества. При обработке массивов информации наиболее часто задействуются следующие способы:

- кластерный анализ (иерархический и неиерархический);
- байесовские сети;
- поиск явных и неявных ассоциаций;
- линейная регрессия;
- представление сведений в визуальной форме;
- эволюционное программирование;
- генетические алгоритмы.

При этом каждый из перечисленных методов применяется для выборки конкретных данных, построения прогностической модели или очистки БД от ошибочных сведений.

Интеграция OLAP и ИАД.

Оперативная аналитическая обработка и интеллектуальный анализ данных - две составные части процесса поддержки принятия решений. Но сегодня большинство систем OLAP заостряет

внимание только на обеспечении доступа к многомерным данным, а большинство средств ИАД, работающих в сфере закономерностей, имеют дело с одномерными перспективами данных. Эти два вида анализа должны быть тесно объединены, то есть системы OLAP должны фокусироваться не только на доступе, но и на поиске закономерностей. Как заметил N. Raden, "многие компании создали прекрасные хранилища данных, идеально разложив по полочкам горы неиспользуемой информации, которая сама по себе не обеспечивает ни быстрой, ни достаточно грамотной реакции на рыночные события".

K. Parsaye вводит составной термин "OLAP Data Mining" (многомерный интеллектуальный анализ) для обозначения такого объединения. J.Han предлагает еще более простое название - "OLAP Mining", и предлагает несколько вариантов интеграции двух технологий.

1. "Cubing then mining". Возможность выполнения интеллектуального анализа должна обеспечиваться над любым результатом запроса к многомерному концептуальному представлению, то есть над любым фрагментом любой проекции гиперкуба показателей.

2. "Mining then cubing". Подобно данным, извлеченным из хранилища, результаты интеллектуального анализа должны представляться в гиперкубической форме для последующего многомерного анализа.

3. "Cubing while mining". Этот гибкий способ интеграции позволяет автоматически активизировать однотипные механизмы интеллектуальной обработки над результатом каждого шага многомерного анализа (перехода между уровнями обобщения, извлечения нового фрагмента гиперкуба и т. д.).

К сожалению, очень немногие производители предоставляют сегодня достаточно мощные средства интеллектуального анализа многомерных данных в рамках систем OLAP. Проблема также заключается в том, что некоторые методы ИАД (байесовские сети, метод k-ближайшего соседа) неприменимы для задач многомерного интеллектуального анализа, так как основаны на определении сходства детализированных примеров и не способны работать с агрегированными данными.

Классификация методов Data Mining

Классификация по принципу работы с данными разбивает методы Data Mining на две крупные категории:

- методы, связанные с непосредственным использованием (сохранением) данных. Данные в ходе обработки детализируются при построении прогностической модели или в ходе анализа исключений. Однако такие методы малоэффективны при работе с крупными массивами данных. Методики этой категории применяются в формах кластерного анализа, метода ближайшего соседа, метода k-ближайшего соседа, рассуждений по аналогии.

- дистилляция шаблонов - формирование и применение закономерностей, имеющих упорядоченный вид, то есть извлечение информации из изначальных данных с ее преобразованием в определенную систематизированную конструкцию. Технологии этой группы представлены логическими, визуализирующими, кросс-табуляционными и базирующимися на уравнениях методами. Задействование этих методов обеспечивает эффективное применение полученных в ходе свободного поиска результатов (они более компактны по сравнению с базами данных) и преобразование этих сведений в понятные для пользователей закономерности.

В свою очередь, способы логической аналитики делятся на подклассы, к которым относятся постановка нечетких запросов, использование символьных правил, деревьев решений и генетических алгоритмов. Технологии кросс-табуляции основаны на применении так называемых агентов, байесовских сетей и визуальных кросс-таблиц. Статметоды и нейронные сети объединяются в методы на основе уравнений.

Существует еще одна разбивка методов Data Mining - по принципам применения математических моделей в обучении. Здесь выделяются две группы:

- статистические методы, в которых используется усредненный опыт по данным, накопившимся в БД за длительный период. При использовании статметодов предварительно анализируется природа статистических данных, выявляются связи и закономерности, осуществляется многомерный статистический

анализ, строятся динамические модели и прогноз на основе временных рядов;

- кибернетические методы, в которых используются основы компьютерной математики и технологии искусственного интеллекта. В число таких методов входят: эволюционное программирование, нейросети, системы обработки экспертных знаний.

К кибернетическим методам также относятся ассоциативные правила, деревья решений, нечеткая логика, генетические алгоритмы.

В основу технологии Data Mining положена концепция шаблонов (patterns), которые представляют собой закономерности, свойственные под выборкам данных, кои могут быть выражены в форме, понятной человеку. "Mining" по-английски означает "добыча полезных ископаемых", а поиск закономерностей в огромном количестве данных действительно сродни этому процессу.

Цель поиска закономерностей - представление данных в виде, отражающем искомые процессы. Построение моделей прогнозирования также является целью поиска закономерностей.

7.3 Перспективы технологии Data Mining

Традиционные методы анализа данных (статистические методы) и OLAP в основном ориентированы на проверку заранее сформулированных гипотез (verification-driven Data Mining) и на "грубый" разведочный анализ, составляющий основу оперативной аналитической обработки данных (OnLine Analytical Processing, OLAP), в то время как одно из основных положений Data Mining - поиск неочевидных закономерностей. Инструменты Data Mining могут находить такие закономерности самостоятельно и также самостоятельно строить гипотезы о взаимосвязях. Поскольку именно формулировка гипотезы относительно зависимостей является самой сложной задачей, преимущество Data Mining по сравнению с другими методами анализа методов для выявления

Большинство статистических методов для выявления взаимосвязей в данных используют концепцию усреднения по

выборке, приводящую к операциям над несуществующими величинами, тогда как Data Mining оперирует реальными значениями.

OLAP больше подходит для понимания ретроспективных данных, Data Mining опирается на ретроспективные данные для получения ответов на вопросы о будущем.

Потенциал Data Mining дает "зеленый свет" для расширения границ применения технологии. Относительно перспектив Data Mining возможны следующие направления развития:

- выделение типов предметных областей с соответствующими им эвристиками, формализация которых облегчит решение соответствующих задач Data Mining, относящихся к этим областям;

- создание формальных языков и логических средств, с помощью которых будут формализованы рассуждения и автоматизация которых станет инструментом решения задач Data Mining в конкретных предметных областях;

- создание методов Data Mining, способных не только извлекать из данных закономерности, но и формировать некие теории, опирающиеся на эмпирические данные;

- преодоление существенного отставания возможностей инструментальных средств Data Mining от теоретических достижений в этой области.

Если рассматривать будущее Data Mining в краткосрочной перспективе, то очевидно, что развитие этой технологии наиболее направлено к областям, связанным с бизнесом.

В краткосрочной перспективе продукты Data Mining могут стать такими же обычными и необходимыми, как электронная почта, и, например, использоваться пользователями для поиска самых низких цен на определенный товар или наиболее дешевых билетов.

В долгосрочной перспективе будущее Data Mining является действительно захватывающим - это может быть поиск интеллектуальными агентами как новых видов лечения различных заболеваний, так и нового понимания природы вселенной.

Однако Data Mining таит в себе и потенциальную опасность - ведь все большее количество информации становится доступным

через всемирную сеть, в том числе и сведения частного характера, и все больше знаний возможно добыть из нее:

Не так давно крупнейший онлайн-магазин "Amazon" оказался в центре скандала по поводу полученного им патента "Методы и системы помощи пользователям при покупке товаров", который представляет собой не что иное как очередной продукт Data Mining, предназначенный для сбора персональных данных о посетителях магазина. Новая методика позволяет прогнозировать будущие запросы на основании фактов покупок, а также делать выводы об их назначении. Цель данной методики - то, о чем говорилось выше - получение как можно большего количества информации о клиентах, в том числе и частного характера (пол, возраст, предпочтения и т.д.). Таким образом, собираются данные о частной жизни покупателей магазина, а также членах их семей, включая детей. Последнее запрещено законодательством многих стран - сбор информации о несовершеннолетних возможен там только с разрешения родителей.

Исследования отмечают, что существуют как успешные решения, использующие Data Mining, так и неудачный опыт применения этой технологии. Области, где применения технологии Data Mining, скорее всего, будут успешными, имеют такие особенности:

- требуют решений, основанных на знаниях;
- имеют изменяющуюся окружающую среду;
- имеют доступные, достаточные и значимые данные;
- обеспечивают высокие дивиденды от правильных решений.

7.4. Отличительные особенности и проблемы

Существующие традиционные методы анализа данных (статистические методы) в основном ориентированы на проверку заранее сформулированных гипотез и на «грубый» разведочный анализ, составляющий основу оперативной аналитической обработки данных (OnLine Analytical Processing, OLAP), в то время как одно из основных положений Data Mining, смотрящееся наиболее выигрышно, - поиск неочевидных закономерностей. Инструменты Data Mining могут находить такие закономерности

самостоятельно и также самостоятельно строить гипотезы о взаимосвязях. Поскольку именно формулировка гипотезы относительно зависимостей является самой сложной задачей, казавшейся до недавнего времени неразрешимой, преимущество Data Mining по сравнению с другими методами анализа является очевидным. Большинство статистических методов для выявления взаимосвязей в данных используют концепцию усреднения по выборке, приводящую к операциям над несуществующими величинами, тогда как Data Mining оперирует реальными значениями. OLAP больше подходит для понимания ретроспективных данных, Data Mining опирается на ретроспективные данные для получения ответов на вопросы о будущем.

Прежде чем использовать технологию Data Mining, необходимо тщательно проанализировать ее проблемы, ограничения и критические вопросы, с ней связанные, а также понять, чего эта технология не может.

Data Mining не может заменить аналитика. Технология не может дать ответы на те вопросы, которые не были заданы. Она не может заменить аналитика, а всего лишь дает ему мощный инструмент для облегчения и улучшения его работы.

Сложность разработки и эксплуатации приложений Data Mining. Поскольку данная технология является мультидисциплинарной областью, для разработки приложения, включающего Data Mining, необходимо задействовать специалистов из разных областей, а также обеспечить их качественное взаимодействие.

Квалификация пользователя. Различные инструменты Data Mining имеют различную степень «дружелюбности» интерфейса и требуют определенной квалификации пользователя. Поэтому программное обеспечение должно соответствовать уровню подготовки пользователя. Использование Data Mining должно быть неразрывно связано с повышением квалификации пользователя.

Извлечение полезных сведений невозможно без хорошего понимания сути данных. Необходим тщательный выбор модели и интерпретация зависимостей или шаблонов, которые обнаружены. Поэтому работа с такими средствами требует тесного

сотрудничества между экспертом в предметной области и специалистом по инструментам Data Mining. Построенные модели должны быть грамотно интегрированы в бизнес-процессы для возможности оценки и обновления моделей. В последнее время системы Data Mining поставляются как часть технологии хранилищ данных.

Сложность подготовки данных. Успешный анализ требует качественной предобработки данных. По утверждению аналитиков и пользователей баз данных, процесс предобработки может занять до 80% процентов всего Data Mining-процесса. Таким образом, чтобы технология работала на себя, потребуется много усилий и времени, которые уходят на предварительный анализ данных, выбор модели и ее корректировку.

Большой процент ложных, недостоверных или бессмысленных результатов. С помощью Data Mining можно отыскивать действительно очень ценную информацию, которая вскоре даст большие дивиденды, в частности в виде финансовой и конкурентной выгоды. Однако Data Mining достаточно часто делает множество ложных и не имеющих смысла открытий. Многие специалисты утверждают, что Data Mining-средства могут выдавать огромное количество статистически недостоверных результатов. Чтобы этого избежать, необходима проверка адекватности полученных моделей на тестовых данных.

Наличие достаточного количества репрезентативных данных. Средства Data Mining, в отличие от статистических, теоретически не требуют наличия строго определенного количества ретроспективных данных. Эта особенность может стать причиной обнаружения недостоверных, ложных моделей и, как результат, принятия на их основе неверных решений. Необходимо осуществлять контроль статистической значимости обнаруженных знаний.

Исследования отмечают, что существуют как успешные решения, использующие Data Mining, так и неудачный опыт применения этой технологии [1]. Области, где применения технологии Data Mining, скорее всего, будут успешными, имеют такие особенности:

– требуют решений, основанных на знаниях;

- имеют изменяющуюся окружающую среду;
- имеют доступные, достаточные и значимые данные;
- обеспечивают высокие дивиденды от правильных решений

Направления применения

Исследование областей применения. Приложения интеллектуального анализа данных уже сильно помогают предпринимательской деятельности получать конкурентные преимущества. Область применимости ИАД для бизнеса продолжает расширяться, так как в розничной торговле стали широко применяться электронная коммерция и электронный маркетинг. Data Mining начинает использоваться в других областях, таких как анализ текстов веб-ресурсов, финансовый анализ, промышленность, государственный сектор, биология, медицина и естественные науки. Новейшие области применения включают в себя интеллектуальный анализ данных для борьбы с терроризмом и мобильный (беспроводной) DM. Поскольку универсальные Data Mining-системы могут иметь ограничения в решении конкретных прикладных задач, очевидна тенденция развития более проблемно-ориентированных систем и инструментов, а также скрытых функций ИАД, встроенных в различные сервисы.

Масштабируемые и интерактивные методы интеллектуального анализа данных.

В отличие от традиционных методов анализа данных, DM должен быть в состоянии обрабатывать огромные объемы данных эффективно и, если это возможно, в интерактивном режиме. Поскольку объемы собираемых данных продолжают неумолимо расти, становятся необходимы хорошо масштабируемые алгоритмы. Важным направлением в сторону улучшения общей эффективности процесса анализа данных, при одновременном повышении уровня взаимодействия с пользователем, является constraint-based mining, интеллектуальный анализ данных с комплексной увязкой параметров. Это предоставляет пользователям дополнительные возможности управления, позволяя устанавливать и использовать некоторые ограничения, чтобы руководить системами ИАД в поисках интересных паттернов и знаний.

Интеграция ИАД с системами поиска, баз данных, хранилищ данных и облачных вычислений. Вышеуказанные системы являются господствующими тенденциями в области систем обработки информации и вычислительных систем. Важно гарантировать, чтобы Data Mining служил неотъемлемым компонентом анализа данных, который можно было бы легко интегрировать в такие среды обработки информации. Подсистема интеллектуального анализа данных должна быть реализована в виде легко интегрируемого унифицированного фреймворка или в виде скрытой функция. Это позволит обеспечить доступность данных, переносимость методов ИАД, масштабируемость, высокую производительность и единую среду обработки информации для многомерного анализа и исследований данных.

Интеллектуальный анализ социальных и информационных сетей. Данная задача является критичной, потому что такие сети уже используются повсеместно и продолжают бурно эволюционировать. Остро стоит задача мониторинга сетей и поиска скрытых зависимостей в собранных данных. Развитие масштабируемых и эффективных методов и приложений обнаружения знаний для большого количества сетевых данных имеет очень важное значение. Наиболее активные разработки ведутся в следующих сферах: очистка, интеграция, и валидация данных с помощью анализа информационных сетей; контролируемая кластеризация и классификация гомогенных и гетерогенных сетей; выявление ролей, предсказание связей, установление сходств в информационных сетях; установление эволюционных алгоритмов, поиск аномалий.

Интеллектуальный анализ пространственно-временных данных и движущихся объектов. Данное направление быстро развивается из-за широкого использования сотовых телефонов, GPS-устройств, датчиков, сенсоров и прочего оборудования беспроводной связи. Естественно, что существует множество сложных вопросов реализации эффективного и своевременного обнаружения знаний в таких данных, поскольку очевидно широкое внедрение сенсорных сетей в ближайшем будущем во многие аспекты жизнедеятельности человека.

Интеллектуальный анализ биологических и биомедицинских данных.

Уникальное сочетание сложности, богатства, размера и значения биологических и биомедицинских данных требует особого внимания. Анализ белковых и ДНК-последовательностей, биологических путей метаболизма и биологических сетей – это только некоторые перспективные области. Другие области биологических применений ИАД включают в себя добычу знаний из гор специализированной литературы, анализ связей в гетерогенных биологических данных, а также интеграция и консолидация биологической информации.

Data Mining, связанный с разработкой программного обеспечения и проектированием систем. Компьютерные программы и большие информационные системы становятся всё более сложными и громоздкими, а также, как правило, всё больше стремятся к архитектуре интеграции множества компонентов, разработанных разными командами. Эта тенденция сделала намного более сложной задачу обеспечения надёжности и отказоустойчивости ПО. Анализ программ, содержащих ошибки, по существу, является процессом интеллектуального анализа данных. Программы во время своего исполнения генерируют большие объёмы данных о своём состоянии и происходящих процессах (логи). Эти данные, сопровождающие работу программ, необходимо всесторонне анализировать и отслеживать, проводить тесты – это может открыть важные скрытые паттерны и аномалии; обнаружение которых обеспечит повышение надёжности и отказоустойчивости, и, возможно, приблизит полностью автоматическое обнаружение ошибок (багов) в ПО. Дальнейшее развитие методологии интеллектуального анализа данных для отладки ПО и информационных систем безусловно необходимо для улучшения их эксплуатационных характеристик.

Визуальный и звуковой анализ данных. Данная сфера представляется эффективным способом интеграции с визуальными и слуховыми системами людей для обнаружения знаний в огромных объёмах данных. Планомерное развитие подобных технологий будет способствовать участию человека в повышении эффективности анализа данных.

Распределенный ИАД, и интеллектуальный анализ потоков данных в реальном времени. Традиционные методы ИАД, предназначенные для работы в централизованном размещении, плохо работают в существующих на данный момент распределенных вычислительных средах (интернет, локальные сети, высокоскоростные беспроводные сети, сенсорные сети и облачные среды). Ожидаются серьёзные подвижки в области распределенных методов Data Mining. Кроме того, многие области приложений, использующие потоки данных (электронная коммерция, анализ веб-ресурсов, биржевой анализ, системы обнаружения и предотвращения вторжений, мобильный Data Mining), требуют, чтобы модели анализа данных строились в реальном времени. В этом направлении также необходимы дополнительные исследования.

Контрольные вопросы

1. Что знаете о Data Mining?
2. Какие способы имеются при обработке массивов информации?
3. Какие категории классификация по принципу работы с данными разбивает методы Data Mining?
4. Какие инструменты Data Mining знаете?
5. Отличия Data Mining от других методов анализа данных?

ГЛАВА VIII. ЗАДАЧИ DATA MINING

Основу методов data mining составляют всевозможные методы классификации, моделирования и прогнозирования, основанные на применении деревьев решений, искусственных нейронных сетей, генетических алгоритмов, эволюционного программирования, ассоциативной памяти, нечёткой логики. К методам data mining нередко относят *статистические методы* (дескриптивный анализ, корреляционный и регрессионный анализ, факторный анализ, дисперсионный анализ, компонентный анализ, дискриминантный анализ, анализ временных рядов, анализ выживаемости, анализ связей). Такие методы, однако, предполагают некоторые априорные представления об анализируемых данных, что несколько расходится с целями data mining (обнаружение ранее неизвестных нетривиальных и практически полезных знаний).

Одно из важнейших назначений методов data mining состоит в наглядном представлении результатов вычислений (визуализация), что позволяет использовать инструментарий data mining людьми, не имеющими специальной математической подготовки.

Применение статистических методов анализа данных требует хорошего владения теорией вероятностей и математической статистикой.

8.1. Обзор методов Data mining

С помощью методов интеллектуального анализа выполняется систематизация данных по критериям количества и качества. При обработке массивов информации наиболее часто задействуются следующие способы:

- кластерный анализ (иерархический и неиерархический);
- байесовские сети;
- поиск явных и неявных ассоциаций;
- линейная регрессия;
- представление сведений в визуальной форме;
- эволюционное программирование;
- генетические алгоритмы.

При этом каждый из перечисленных методов применяется для выборки конкретных данных, построения прогностической модели или очистки БД от ошибочных сведений.

Все методы Data Mining можно разделить на две большие группы по принципу работы с исходными обучающими данными. В этой классификации верхний уровень определяется на основании того, сохраняются ли данные после Data Mining либо они дистиллируются для последующего использования.

Непосредственное использование данных, или сохранение данных.

В этом случае исходные данные хранятся в явном детализированном виде и непосредственно используются на стадиях прогностического моделирования и/или анализа исключений. Проблема этой группы методов – при их использовании могут возникнуть сложности анализа сверхбольших баз данных.

Логические методы, или методы логической индукции, включают: нечеткие запросы и анализы; символьные правила; деревья решений; генетические алгоритмы. Методы этой группы являются, пожалуй, наиболее интерпретируемыми – они оформляют найденные закономерности, в большинстве случаев, в достаточно прозрачном виде с точки зрения пользователя. Полученные правила могут включать непрерывные и дискретные переменные. Следует заметить, что деревья решений могут быть легко преобразованы в наборы символьных правил путем генерации одного правила по пути от корня дерева до его терминальной вершины. Деревья решений и правила фактически являются разными способами решения одной задачи и отличаются лишь по своим возможностям. Кроме того, реализация правил осуществляется более медленными алгоритмами, чем индукция деревьев решений.

Методы кросс-табуляции: агенты, байесовские (доверительные) сети, кросс-табличная визуализация. Последний метод не совсем отвечает одному из свойств Data Mining – самостоятельному поиску закономерностей аналитической системой. Однако предоставление информации в виде кросс-таблиц обеспечивает реализацию основной задачи Data Mining – поиск шаблонов,

поэтому этот метод можно также считать одним из методов Data Mining.

Методы на основе уравнений. Методы этой группы выражают выявленные закономерности в виде математических выражений — уравнений. Следовательно, они могут работать лишь с численными переменными, и переменные других типов должны быть закодированы соответствующим образом. Это несколько ограничивает применение методов данной группы, тем не менее, они широко используются при решении различных задач, особенно задач прогнозирования. Данная классификация разделяет все многообразие методов Data Mining на две группы: статистические и кибернетические методы.

Статистические методы Data Mining

В эти методы представляют собой четыре взаимосвязанных раздела:

- предварительный анализ природы статистических данных (проверка гипотез стационарности, нормальности, независимости, однородности, оценка вида функции распределения, ее параметров и т.п.);
- выявление связей и закономерностей (линейный и нелинейный регрессионный анализ, корреляционный анализ и др.);
- многомерный статистический анализ (линейный и нелинейный дискриминантный анализ, кластерный анализ, компонентный анализ, факторный анализ и др.);
- динамические модели и прогноз на основе временных рядов.

Классификация по задачам

Методы Data Mining также можно классифицировать по задачам Data Mining. В соответствии с такой классификацией выделяем две группы. Первая из них — это подразделение методов Data Mining на решающие задачи сегментации (т.е. задачи классификации и кластеризации) и задачи прогнозирования. В соответствии со второй классификацией по задачам методы Data Mining могут быть направлены на получение описательных и прогнозирующих результатов.

Описательные методы

Описательные методы служат для нахождения шаблонов или образцов, описывающих данные, которые поддаются интерпретации с точки зрения аналитика.

Прогнозирующие методы

Прогнозирующие методы используют значения одних переменных для предсказания/прогнозирования неизвестных (пропущенных) или будущих значений других (целевых) переменных.

8.2. Кластерный анализ данных. Задачи кластерного анализа

Кластерный анализ (англ. clusteranalysis) — многомерная статистическая процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы. Задача кластеризации относится к статистической обработке, а также к широкому классу задач обучения без учителя.

Большинство исследователей склоняются к тому, что впервые термин «кластерный анализ» (англ. cluster — гроздь, сгусток, пучок) был предложен математиком Р. Трионом. Впоследствии возник ряд терминов, которые в настоящее время принято считать синонимами термина «кластерный анализ»: автоматическая классификация, ботриология.

Спектр применений кластерного анализа очень широк: его используют в археологии, медицине, психологии, химии, биологии, государственном управлении, — филологии, антропологии, маркетинге, социологии, геологии и других дисциплинах. Однако универсальность применения привела к появлению большого количества несовместимых терминов, методов и подходов, затрудняющих однозначное использование и непротиворечивую интерпретацию кластерного анализа.

Кластерный анализ выполняет следующие основные задачи:

- разработка типологии или классификации;
- исследование полезных концептуальных схем группирования объектов;
- порождение гипотез на основе исследования данных;

- проверка гипотез или исследования для определения, действительно ли типы (группы), выделенные тем или иным способом, присутствуют в имеющихся данных.

Независимо от предмета изучения применение кластерного анализа предполагает следующие этапы:

- отбор выборки для кластеризации. Подразумевается, что имеет смысл кластеризовать только количественные данные;
- определение множества переменных, по которым будут оцениваться объекты в выборке, то есть признакового пространства;
- вычисление значений той или иной меры сходства (или различия) между объектами;
- применение метода кластерного анализа для создания групп сходных объектов;
- проверка достоверности результатов кластерного решения.

Байесовская сеть (или байесова сеть, байесовская сеть доверия, англ. Bayesiannetwork, beliefnetwork) — графовая вероятностная модель, представляющая собой множество переменных и их вероятностных зависимостей по Байесу. Например, байесовская сеть может быть использована для вычисления вероятности того, чем болен пациент, по наличию или отсутствию ряда симптомов, основываясь на данных о зависимости между симптомами и болезнями. Математический аппарат байесовых сетей создан американским учёным Джудой Перлом, лауреатом Премии Тьюринга.

Формально, байесовская сеть — это ориентированный ациклический граф, каждой вершине которого соответствует случайная переменная, а дуги графа кодируют отношения условной независимости между этими переменными. Вершины могут представлять переменные любых типов, быть взвешенными параметрами, скрытыми переменными или гипотезами. Существуют эффективные методы, которые используются для вычислений и обучения байесовских сетей. Если переменные байесовской сети являются дискретными случайными величинами, то такая сеть называется дискретной байесовской сетью. Байесовские сети, которые моделируют последовательности переменных, называют динамическими байесовскими сетями.

Байесовские сети, в которых могут присутствовать как дискретные переменные, так и непрерывные, называются гибридными байесовскими сетями. Байесовская сеть, в которой дуги помимо отношений условной независимости кодируют также отношения причинности, называют причинно-следственными байесовскими сетями (англ. causalbayesiannetworks)).

Линейная регрессия (англ. Linearregression) — используемая в статистике регрессионная модель зависимости одной (объясняемой, зависимой) переменной y от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) x с линейной функцией зависимости.

Модель линейной регрессии является часто используемой и наиболее изученной в эконометрике. А именно изучены свойства оценок параметров, получаемых различными методами при предположениях о вероятностных характеристиках факторов, и случайных ошибок модели. Предельные (асимптотические) свойства оценок нелинейных моделей также выводятся исходя из аппроксимации последних линейными моделями. Необходимо отметить, что с эконометрической точки зрения более важное значение имеет линейность по параметрам, чем линейность по факторам модели.

Визуализация информации — это (интерактивное) изучение визуальных представлений абстрактных данных для усиления человеческого восприятия. Абстрактные данные включают как числовые, так и нечисловые данные, такие как текст и географическая информация. Однако визуализация информации отличается от научной визуализации — «это визуализация информации, при которой пространственное представление выбирается, а в научной визуализации пространственное представление задано».

Эволюционное моделирование (англ. Evolutionarycomputation) использует признаки теории Дарвина для построения интеллектуальных систем (методы группового учёта, генетические алгоритмы). Является частью более обширной области искусственного интеллекта — вычислительного интеллекта.

Эволюционное моделирование это уже достаточно сложившаяся область, в которой можно выделить:

- модели возникновения молекулярно-генетических информационных систем;

- моделирование общих закономерностей эволюции (Эволюционные алгоритмы). Это системы, которые используют только эволюционные принципы. Они успешно использовались для задач типа функциональной оптимизации и могут легко быть описаны на математическом языке. К ним относятся эволюционные алгоритмы, такие как Эволюционное программирование, Генетические алгоритмы, Эволюционные стратегии, Генетическое программирование;

- эволюционные модели. Это системы, которые являются биологически более реалистичными, чем эволюционные алгоритмы, но которые не оказались полезными в прикладном смысле. Они больше похожи на биологические системы и менее направлены на решение технических задач. Они обладают сложным и интересным поведением, и, видимо, вскоре получат практическое применение. К этим системам относят так называемую искусственную жизнь.

- прикладное эволюционное моделирование.

Генетические алгоритмы предназначены для решения задач оптимизации. Примером подобной задачи может служить обучение нейросети, то есть подбора таких значений весов, при которых достигается минимальная ошибка. При этом в основе генетического алгоритма лежит метод случайного поиска. Основным недостатком случайного поиска является то, что нам неизвестно сколько понадобится времени для решения задачи.

Для того, чтобы избежать таких расходов времени при решении задачи, применяются методы, открытые при изучении эволюции и происхождения видов. Как известно, в процессе эволюции выживают наиболее приспособленные особи. Это приводит к тому, что приспособленность популяции возрастает, позволяя ей лучше выживать в изменяющихся условиях.

Согласно классификации по стратегиям, задачи Data Mining подразделяются на следующие группы:

- обучение с учителем;
- обучение без учителя;
- другие.

Категория обучение с учителем представлена следующими задачами Data Mining: классификация, оценка, прогнозирование.

Категория обучение без учителя представлена задачей кластеризации.

В категорию другие входят задачи, не включенные в предыдущие две стратегии.

Задачи Data Mining, в зависимости от используемых моделей, могут быть дескриптивными и прогнозирующими. Эти типы моделей будут подробно описаны в лекции, посвященной процессу Data Mining.

В соответствии с этой классификацией, задачи Data Mining представлены группами описательных и прогнозирующих задач.

В результате решения описательных (descriptive) задач аналитик получает шаблоны, описывающие данные, которые поддаются интерпретации.

Эти задачи описывают общую концепцию анализируемых данных, определяют информативные, итоговые, отличительные особенности данных. Концепция описательных задач подразумевает характеристику и сравнение наборов данных. Характеристика набора данных обеспечивает краткое и сжатое описание некоторого набора данных.

Сравнение обеспечивает сравнительное описание двух или более наборов данных.

Прогнозирующие (predictive) основываются на анализе данных, создании модели, предсказании тенденций или свойств новых или неизвестных данных.

Достаточно близким к вышеупомянутой классификации является подразделение задач Data Mining на следующие: исследования и открытия, прогнозирования и классификации, объяснения и описания.

8.3. Ведущие производители средств Data Mining

Средства Data Mining, как и большинство средств BusinessIntelligence, традиционно относятся к дорогостоящим программным инструментам - цена некоторых из них доходит до нескольких десятков тысяч долларов. Поэтому до недавнего

времени основными потребителями этой технологии были банки, финансовые и страховые компании, крупные торговые предприятия, а основными задачами, требующими применения Data Mining, считались оценка кредитных и страховых рисков и выработка маркетинговой политики, тарифных планов и иных принципов работы с клиентами. В последние годы ситуация претерпела определенные изменения: на рынке программного обеспечения появились относительно недорогие инструменты Data Mining от нескольких производителей, что сделало доступной эту технологию для предприятий малого и среднего бизнеса, ранее о ней и не помышлявших.

К современным средствам Business Intelligence относятся генераторы отчетов, средства аналитической обработки данных, средства разработки BI-решений (BI Platforms) и так называемые Enterprise BI Suites - средства анализа и обработки данных масштаба предприятия, которые позволяют осуществлять комплекс действий, связанных с анализом данных и с созданием отчетов, и нередко включают интегрированный набор BI-инструментов и средства разработки BI-приложений. Последние, как правило, содержат в своем составе и средства построения отчетов, и OLAP-средства, а нередко - и Data Mining-средства.

По данным аналитиков Gartner Group, лидерами на рынке средств анализа и обработки данных масштаба предприятия являются компании Business Objects, Cognos, Information Builders, а претендуют на лидерство также Microsoft и Oracle (рис. 1). Что касается средств разработки BI-решений, то основными претендентами на лидерство в этой области являются компании Microsoft и SAS Institute.

Отметим, что средства Business Intelligence компании Microsoft относятся к сравнительно недорогим продуктам, доступным широкому кругу компаний. Именно поэтому мы и собираемся рассмотреть некоторые практические аспекты применения Data Mining на примере продуктов этой компании в последующих частях данной статьи.

8.4. Большие данные и глубокое машинное обучение в искусственных нейронных сетях

В последнее время сформировалось отличие между обычными структурированными базами данных с четкими и понятными обработками и большими данными, которые могут не иметь большой объем информации на начальном этапе, но обладают главной отличительной чертой - умение приспосабливаться под задачу в результате использования «машинного обучения». Вообще определяющими характеристиками больших данных являются объем, вариативность, скорость и ценность.

Вариативность данных позволяет выявить зависимости там, где на первый взгляд их не стоит искать. Например, зависимость активности покупателей от погоды или зависимость продолжительности сна от потребления лекарств. Скорость обработки информации в больших данных близка к реальному времени. Широкое распространение использования больших данных получили Интернет-ресурсы, занимающиеся продажей товаров и услуг. Прежде всего, это вызвано возможностью получения большого объема информации о действиях пользователей на этих ресурсах. Рекомендованные к просмотру фильмы, к прослушиванию музыкальные композиции, к чтению книги и статьи, контекстная реклама и СПАМ-фильтры в почте, сайты знакомств. Данная технология используется в медицине, банковской и военной сферах, полиции, экономике государств, изучении социальных явлений и для управления большими и малыми социальными группами. На мой взгляд, большие данные являются относительно новым витком в информационных технологиях и в ближайшее время проникнут во все сферы деятельности человека, вытесняя обычные базы данных с их бизнес-аналитиками. Прежде всего, данные собираются на сетевых хранилищах, которые могут быть как разрозненными, так и объединены в единую систему. Информация в обязательном порядке дублируется для исключения возможных потерь и для неё характерно отсутствие структуры, т.е. это может быть текст, изображения в различных форматах, голос, музыка и т.п. В

дальнейшем данные обрабатываются алгоритмом, написанным программистами для получения информации в удобном для человека виде. После работы алгоритма человек или группа людей делает аналитику в области обработанных данных и принимают решения о дальнейших действиях. Всё больше набирает популярность «черных ящиков» в технологии «машинного обучения», когда исключается человек при анализе данных и принятии решений о необходимых действиях. Наука о данных изучающая вопросы получения, обработки, анализа информации и предоставления этих данных в понятной форме набирает популярность. Одним из плодов развития науки о данных получила перспективная технология «машинного обучения», так называемые чёрные ящики. Существует множество разнообразных методик анализа массивов данных, в основе которых лежит инструментарий, заимствованный из статистики и информатики. Глубинное или глубокое обучение—набор алгоритмов машинного обучения, основанный на нейросетях, которые пытаются моделировать высокоуровневые абстракции в данных, используя архитектуры, состоящие из множества не линейных преобразований. Глубокое обучение является частью более широкого семейства методов машинного обучения, которые подбирают представление данных. Наблюдение (например, изображение) может быть представлено многими способами, такими как вектор интенсивности значений на пиксель, или (в более абстрактной форме) как множество примитивов, областей определенной формы, и т. д. Некоторые представления позволяют легче решать поставленные задачи (например, распознавание лица или распознавание выражения лица). Применение глубокого обучения автоматизирует сам процесс выбора и настройки признаков, проводя обучение признаков без учителя или с частичным привлечением учителя, используя для этого эффективные алгоритмы и иерархическое извлечение признаков. Искусственная нейронная сеть (ИНС) —математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей —сетей нервных клеток живого организма. Также есть мнение, что ИНС и нейронная сеть живых организмов имеют

гораздо меньше общего, чем об этом принято считать. Тем не менее, наблюдается увеличение сферы применения искусственных сетей. Метод глубокого машинного обучения получает всё большее распространение в сравнении с другими методами машинного обучения. Такая ситуация связана с тем, что в «традиционном» машинном обучении человеку необходимо самостоятельно выбрать из всего объёма информации только необходимые данные для решения конкретной задачи. Такие данные называются признаками. Если такие признаки выявить не удаётся, то машинное обучение работает плохо. Нейронные сети автоматически отделяют нужные данные от ненужных и умеют определять правильные признаки. К минусам можно отнести потребность в больших вычислительных мощностях в связи с необходимостью обрабатывать большой объём данных. Необходимо отметить, что для успешной работы нейронных сетей является необходимость их обучения на большом объеме данных. На сегодняшний день глубокое обучение нейронных сетей получает всё большее распространение практически во всех областях деятельности человека. Скорость обработки информации практически достигла реального времени и появилась возможность работы не со статичными данными, а с постоянно прибывающим потоком. Это позволяет использовать нейронные сети там, где по естественным причинам внимание человека может быть притуплено с течением времени по естественным биологическим причинам. Также это даёт возможность исключить массу человеческих ошибок в типичных ситуациях в таких областях как медицина, различные виды транспорта, систем безопасности, экономика и финансы, социология, педагогическая деятельность и т.п. Текущий уровень развития не даёт возможности целиком и полностью автоматизировать управление многими процессами только искусственными нейронными сетями, тем более, когда речь идёт о нестандартных ситуациях, хотя, ситуация является нестандартной, пока человек или ИНС не научатся в ней действовать. Не смотря на серьёзные продвижения в области совершенствования искусственных нейронных сетей, человечество находится только в начале этого пути. Скорее всего, будут разрабатываться новые архитектуры и способы обучения ИНС более близких к работе

биологических нейронных сетей. Уже сейчас появляются отдельные мысли и работы о необходимости пересмотра имеющейся концепции искусственных сетей. Научное и околонуучное сообщество осознаёт, что существующие искусственные нейронные сети очень далеки от биологических нейронных сетей даже по архитектуре. Новые исследования в области изучения мозга дают представление о векторе развития ИНС. Одна из моделей принципа работы нейронной сети мозга, которая мне показалась очень перспективной для рассмотрения возможности применения в искусственных сетях, была представлена Алексеем Редозубовым. Согласно этой модели работа мозга ближе к принципу работы компьютера нежели к искусственной нейронной сети. Активность нейронов формирует код, с которым в дальнейшем работает всё нейронное пространство. Нет привязки активности конкретного нейрона при реакции на конкретную информацию, т.к. один и тот же нейрон может реагировать как на похожую информацию, так и на совершенно разную. Реакция биологических нейронов сравнивается с реакцией элементов компьютера, которые формируют код. Рассматривается понятие клеточных автоматов формирующие уникальные узоры из волны активных нейронов по всей сети и передача информации такими узорами. При этом каждый нейрон имеет свой «список» комбинаций соседних активных нейронов для «понимания» в каких случаях он сам должен активироваться. Предполагается, что мозг работает не с аналоговыми, а с дискретными сигналами, таким образом, мозг (кора мозга) работает с конечным числом дискретных понятий. Вводится понятие интерференции информационной волны, способствующей формированию ключа воспоминаний (время, пространство, обстоятельства и пр.) для работы памяти с информацией, которую необходимо запомнить. При этом сохранение информации происходит не в конкретном месте, а распределено по всему мозгу. Причём подразумевается распределение не частей информации по разным местам, а сохранение всего жизненного объёма информации во множестве мест в головном мозге в кортикальных столбах. Одним из фактов, в пользу данной концепции являются эксперименты Карла Лешли в

течение тридцати лет над обученными выходить из лабиринта крысами, которым последовательно удаляли части мозга в поиске места хранения информации с данным навыком. Эксперименты показали, что не существует отдельных частей мозга, где хранились бы конкретные воспоминания, вся память дублирована в различных частях мозга. Повторением экспериментов другими учёными также не удалось обнаружить, что целостная память находится в определённом месте мозга. Становится очевидно, что эти сети будут иметь всё большее значение и участвовать в большем количестве процессов с нарастающей долей. И при дальнейшей эволюции искусственных нейронных сетей мы рано или поздно придём к созданию искусственного интеллекта.

Контрольные вопросы

1. Что такое data mining, компоненты data mining?
2. Цель применения сантмент анализа (три типа моделей, привести пример по каждой модели) ?
3. Что такое препроцесинг документов для целей тематического моделирования и сантмент анализа?
4. Цели и задачи кластерного анализа?
5. Направления в кластерном анализе?
6. Меры близости в кластерном анализе?
7. Алгоритм K-means?
8. Алгоритм иерархической кластеризации?
9. Достоинство и недостатки кластерного подхода?
10. Особенности тематического моделирования. Цель применения тематического моделирования?

Список литературы

1. J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann (2000).
2. Дивакар Майсор, Шрикант Кхупат, и Швета Джайн. Архитектура и шаблоны больших данных. [Электронный ресурс] Режим доступа: <https://www.ibm.com/developerworks/ru/library/bdarchpatterns1/index.html>
3. Davy Cielen, Arno D. B. Meysman, and Mohamed Ali. Introducing Data Science. Big data, machine learning, and more, using Python tools <https://www.manning.com/books/introducing-data-science>
4. Hilbert, M. (2016). Big Data for Development: A Review of Promises and Challenges. Development Policy Review, 34(1), 135–174. <http://doi.org/10.1111/dpr.12142>
5. Environmental Protection Agency. Subpart A—General Provisions. <https://www.gpo.gov/fdsys/pkg/CFR-2011-title40-vol1/pdf/CFR-2011-title40-vol1-part3-subpartA.pdf>
6. Seth Gilbert and Nancy Lynch. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. <https://doi.org/10.1145/564585.564601>
7. Steven J. Plimpton and Karen D. Devine (2011), MapReduce in MPI for Largescale Graph Algorithms, <https://doi.org/10.1016/j.parco.2011.02.004>
8. Образовательный портал: www.ziyounet.uz.
9. Каталог национальных интернет ресурсов: www.uz.
10. Infocom.uz электрон журнал: www.infocom.uz.
11. «O'zbektelekom» АК: www.uztelecom.uz
12. www.intuit.ru
13. <http://www.db-book.com/> Database System Concepts
14. <http://www.amazon.com/Database-System-Concepts-Abraham-Silberschatz/dp/0073523321>

ОГЛАВЛЕНИЕ

Введение	3
Глава I. Введение в анализ больших данных.	6
1.1. Определение основных понятий.	6
1.2. Технологии и тенденции работы с BigData.	15
1.3. Методы и техники анализа больших данных.	17
1.4. Хроника науки о данных.	18
1.5. Преимущества использования больших данных.	23
Глава II. Системы поддержки принятия решений. OLTP и OLAP технологии.	26
2.1. История создания СППР	26
2.2. OLAP технологии.	37
2.3. Основные аналитические операции OLAP.	42
2.4. OLTP технологии.	45
Глава III. Многомерная модель данных. OLAP системы.	51
3.1. Что такое многомерная модель данных	51
3.2. Архитектура OLAP-систем: ROLAP, MOLAP, HOLAP	60
3.3. ROLAP-серверы	71
3.4. HOLAP-серверы	75
3.5. Применение OLAP систем	76
Глава IV. Логические модели хранилищ данных.	83
4.1. Хранилище данных	83
4.2. Характеристики хранилищ данных	85
4.3. Модели хранилище данных	88
4.4. Логической модели данных	89
Глава V. Физические модели хранилищ данных.	97
5.1. Объекты физической модели данных	97
5.2. Иерархия объектов реляционной базы данных	103
5.3. Основные объекты реляционной базы данных	105
Глава VI. Современные программные средства для анализа больших объемов информации.	110
6.1. Технологии и тенденции работы с Bigdata	110
6.2. Map Reduce платформа	114

6.3. Концептуальная архитектура Hadoop	116
6.4. Решение практических задач с помощью технологии обработки больших данных	117
Глава VII. Data Mining. Интеллектуальный анализ данных.	119
7.1. Общие понятие о Data Mining	119
7.2. Обзор и классификация методов Data Mining	122
7.3. Перспективы технологии Data Mining	125
7.4. Существующие подходы к анализу	127
Глава VIII. Задачи Data Mining и машинное обучение в искусственных нейронных сетях	134
8.1. Обзор методов Data mining	134
8.2. Кластерный анализ данных. Задачи кластерного анализа.	137
8.3. Ведущие производители средств Data Mining	141
8.4. Большие данные и глубокое машинное обучение в искусственных нейронных сетях	143
Список литературы	148

АЛИМОВА Ф.М., КУШМАНОВА М.А., НАИМ Н.А.

BIG DATA

УЧЕБНОЕ ПОСОБИЕ

Ташкент - "METHODIST NASHRIYOTI" - 2024

Редактор: Бакиров Нурмухаммад

Разрешено в печать 1.04.2024.
 Формат 60х90. Гарнитура "Times New Roman" garnituras.
 Офсетная печать.
 Усл.печ.л. 10. Изд.печ.л 9,5.
 Тираж 300.

Опубликовано в типографии ООО "METHODIST NASHRIYOTI".
 Адрес: город Ташкент, улица Шота Руставели 2-ваторный, дом 1.



+99893 552-11-21

Воспроизведение запрещено без разрешения издателя.

89303, 4

ISBN 978-9910-03-125-0

