А. А. Барсегян
М. С. Куприянов
В. В. Степаненко
И. И. Холод

Методы и модели анализа данных: OLAP и Data Mining

- *Хранилища данных*
- *OLAP – оперативный анализ*
- *Data Mining – интеллектуальный анализ*
- *Методы решения задач классификации, кластеризации и поиска ассоциативных правил*

УЧЕБНОЕ ПОСОБИЕ

bhv®

А. А. Барсегян
М. С. Куприянов
В. В. Степаненко
И. И. Холод

Методы и модели анализа данных: OLAP и Data Mining

Рекомендовано УМО вузов по университетскому политехническому образованию в качестве учебного пособия по специальности 071900 «Информационные системы и технологии» направления 654700 «Информационные системы»

Санкт-Петербург
«БХВ-Петербург»
2004

УДК 681.3.06(075.8)
ББК 32.973.26-018.2я73
Б26

Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И.

Б26 Методы и модели анализа данных: OLAP и Data Mining. — СПб.: БХВ-Петербург, 2004. — 336 с.: ил.

ISBN 5-94157-522-X

В книге освещены основные направления в области анализа данных: организация хранилища данных, оперативный (OLAP) и интеллектуальный (Data Mining) анализ данных. Приведено описание методов и алгоритмов решения основных задач анализа: классификации, кластеризации и др. Описание идеи каждого метода дополняется конкретным примером его применения. Представлены стандарты и библиотека алгоритмов Data Mining. Прилагается компакт-диск, содержащий описание стандартов Data Mining, библиотеку алгоритмов Xelopes, лабораторный практикум и необходимое для практической работы программное обеспечение.

Для студентов и специалистов в области анализа данных

УДК 681.3.06(075.8)
ББК 32.973.26-018.2я73

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Людмила Еремеевская*
Зав. редакцией	*Григорий Добин*
Редактор	*Алексей Данченко*
Компьютерная верстка	*Ольги Сергиенко*
Корректор	*Зинаида Дмитриева*
Дизайн обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.08.04.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 27,09.
Тираж 2500 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02
от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-522-X

Содержание

Предисловие авторов

Повсеместное использование компьютеров привело к пониманию важности задач, связанных с анализом накопленной информации с целью извлечения новых знаний. Возникла потребность в создании хранилищ данных и систем поддержки принятия решений, основанных в том числе на методах теории искусственного интеллекта.

Действительно, управление предприятием, банком, различные сферы бизнеса, в том числе электронного, немыслимы без процессов накопления, анализа, выявления определенных закономерностей и зависимостей, прогнозирования тенденций и рисков.

Именно давний интерес авторов к методам, алгоритмическим моделям и средствам их реализации, используемым на этапе анализа данных, явился причиной подготовки данной книги.

В книге представлены наиболее перспективные направления анализа данных: хранение информации, оперативный и интеллектуальный анализ. Подробно рассмотрены методы и алгоритмы интеллектуального анализа. Кроме описания популярных и известных методов анализа приводятся оригинальные результаты. В частности, *разд. 7.4* подготовлен С. И. Елизаровым.

Книга ориентирована на студентов и специалистов, интересующихся современными методами анализа данных. Наличие в приложениях материала, посвященного нейронным сетям и генетическим алгоритмам, делает книгу самодостаточной. Как пособие, книга в первую очередь предназначена для бакалавров и магистров, обучающихся по направлению "Информационные системы". Кроме того, книга будет полезна специалистам, занимающимся разработкой корпоративных информационных систем. Подробное описание методов и алгоритмов интеллектуального анализа позволит использовать книгу не только для ознакомления с данной областью применения информации систем, но и для разработки конкретных систем.

Первые четыре главы книги, содержащие общую информацию о современных направлениях анализа данных, будут полезны руководителям предприятий, планирующим внедрение и использование методов анализа данных.

Благодарности:

Григорию Пятецкому-Шапиро — основателю направления Data Mining за поддержку и полезные замечания.

Доктору М. Тессу — одному из руководителей немецкой компании Prudsys за исключительно содержательные консультации по структуре книги и по содержанию ее отдельных частей.

Data Mining и перегрузка информацией

В 2002 году, согласно оценке профессоров из университета Berkeley, объем информации в мире увеличился на пять миллиардов миллиардов (5,000,000,000,000,000,000) байт. Согласно другим оценкам, информация удваивается каждые 2—3 года. Этот потоп, цунами данных приходит из науки, бизнеса, Интернета и других источников. Среди самых больших баз данных в 2003 году France Telecom имела СППР (DSS system) размером 30,000 миллиардов байт, а Alexa Internet Archive содержал 500,000 миллиардов байт.

На первом семинаре, посвященном поиску знаний в данных (Knowledge Discovery in Data workshop), который я организовал в 1989 году, один мегабайт (1,000,000) считался размером для большой базы данных. На последней конференции KDD-2003 один докладчик обсуждал базу данных для астрономии размером во много терабайт и предсказывал необходимость иметь дело с петабайтами (1 терабайт = 1,000 миллиардов байт, а 1 петабайт = 1,000 терабайт).

Из-за огромного количества информации очень малая ее часть будет когда-либо увидена человеческим глазом. Наша единственная надежда понять и найти что-то полезное в этом океане информации — широкое применение методов Data Mining.

Data Mining (также называемая Knowledge Discovery In Data — обнаружение знаний в данных) изучает процесс нахождения новых, действительных и потенциально полезных знаний в базах данных. Data Mining лежит на пересечении нескольких наук, главные из которых — это системы баз данных, статистика и искусственный интеллект.

Область Data Mining выросла из одного семинара в 1989 году до десятков международных конференций в 2003 году с тысячами исследователей во многих странах мира. Data Mining широко используется во многих областях с большим объемом данных. В науке — астрономии, биологии, биоинформатике, медицине, физике и других областях. В бизнесе — торгов-

ле, телекоммуникациях, банковском деле, промышленном производстве и т. д. Благодаря сети Интернет Data Mining используется каждый день тысячи раз в секунду — каждый раз, когда кто-то использует Гугл (Google) или другие поисковые системы (search engines) на просторах Интернета.

Виды информации, с которыми работают исследователи, включают не только цифровые данные, но и все более текст, изображение, видео, звук и т. д. Одна новая и быстро растущая часть Data Mining — это анализ связей между данными (link analysis), которая имеет приложения в таких разных областях, как биоинформатика, цифровые библиотеки и защита против терроризма.

Математический и статистический подходы являются основой для Data Mining. Как уроженцу Москвы и ученику известной в 1970-е годы 2-й математической школы, мне особенно приятно писать предисловие к первой книге на русском языке, покрывающей эту важную и интересную область.

Эта книга дает читателю обзор технологий и алгоритмов для хранения и организации данных, включая ХД и OLAP, а затем переходит к методам и алгоритмам реализации Data Mining.

Авторы приводят обзор наиболее распространенных областей применения Data Mining и объясняют процесс обнаружения знаний. Ряд глав рассматривают основные методы Data Mining, включая классификацию и регрессию, поиск ассоциативных правил и кластеризацию. Книга также обсуждает главные стандарты в области Data Mining.

Важная часть книги — это обзор библиотеки Xelopes компании Prudsys, содержащей многие важные алгоритмы для Data Mining. В заключение дается более детальный анализ продвинутых на сегодняшний день методов — самоорганизующихся, нейронечетких систем и генетических алгоритмов.

Я надеюсь, что эта книга найдет много читателей и заинтересует их важной и актуальной областью Data Mining и поиска знаний.

Григорий Пятецкий-Шапиро, KDnuggets

Закоровье, Нью Гемпшир, США, Январь 2004

ГЛАВА 1

Системы поддержки принятия решений

1.1. Задачи систем поддержки принятия решений

С появлением первых ЭВМ наступил этап информатизации разных сторон человеческой деятельности. Если раньше человек основное внимание уделял веществу, затем энергии (рис. 1.1), то сегодня можно без преувеличения сказать, что наступил этап осознания процессов, связанных с информацией. Вычислительная техника создавалась прежде всего для обработки данных. В настоящее время современные вычислительные системы и компьютерные сети позволяют накапливать большие массивы данных для решения задач обработки и анализа. К сожалению, сама по себе машинная форма представления данных содержит информацию, необходимую человеку, в скрытом виде, и для ее извлечения нужно использовать специальные методы анализа данных.

Большой объем информации, с одной стороны, позволяет получить более точные расчеты и анализ, с другой — превращает поиск решений в сложную задачу. Неудивительно, что первичный анализ данных был переложен на компьютер. В результате появился целый класс программных систем, призванных облегчить работу людей, выполняющих анализ (аналитиков). Такие системы принято называть *системами поддержки принятия решений* — СППР (DSS, Decision Support System).

Для выполнения анализа СППР должна накапливать информацию, обладая средствами ее ввода и хранения. Таким образом, можно выделить три основные задачи, решаемые в СППР:

- ввод данных;
- хранение данных;
- анализ данных.

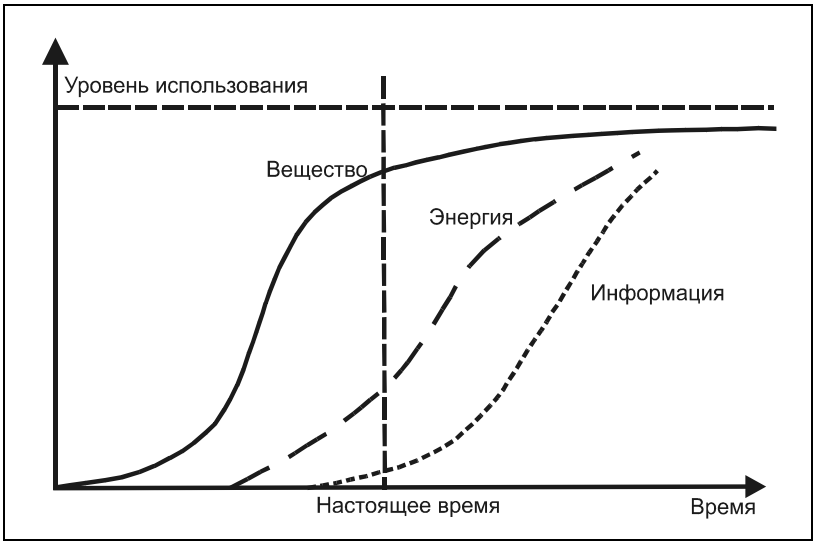

Рис. 1.1. Уровень использования человеком различных объектов материального мира

Таким образом, СППР — это системы, обладающие средствами ввода, хранения и анализа данных, относящихся к определенной предметной области, с целью поиска решений.

Ввод данных в СППР осуществляется либо автоматически от датчиков, характеризующих состояние среды или процесса, либо человеком-оператором. В первом случае данные накапливаются путем циклического опроса, либо по сигналу готовности, возникающему при появлении информации. Во втором случае СППР должны предоставлять пользователям удобные средства ввода данных, контролирующие корректность вводимых данных и выполняющие сопутствующие вычисления. Если ввод осуществляется одновременно несколькими операторами, то система должна решать проблемы параллельного доступа и модификации одних и тех же данных.

Постоянное накопление данных приводит к непрерывному росту их объема. В связи с этим на СППР ложится задача обеспечить надежное хранение больших объемов данных. На СППР также могут быть возложены задачи предотвращения несанкционированного доступа, резервного хранения данных, архивирования и т. п.

Основная задача СППР — предоставить аналитикам инструмент для выполнения анализа данных. Необходимо отметить, что для эффективного использования СППР ее пользователь — аналитик должен обладать соответствующей квалификацией. Система не генерирует правильные решения, а только предоставляет аналитику данные в соответствующем виде (отчеты, таблицы,

графики и т. п.) для изучения и анализа, именно поэтому такие системы обеспечивают выполнение функции поддержки принятия решений. Очевидно, что, с одной стороны, качество принятых решений зависит от квалификации аналитика. С другой — рост объемов анализируемых данных, высокая скорость обработки и анализа, а также сложность использования машинной формы представления данных стимулируют исследования и разработку интеллектуальных СППР. Для таких СППР характерно наличие функций, реализующих отдельные умственные возможности человека.

По степени "интеллектуальности" обработки данных при анализе выделяют три класса задач анализа:

- *информационно-поисковый* — СППР осуществляет поиск необходимых данных. Характерной чертой такого анализа является выполнение заранее определенных запросов;
- *оперативно-аналитический* — СППР производит группирование и обобщение данных в любом виде, необходимом аналитику. В отличие от информационно-поискового анализа в данном случае невозможно заранее предсказать необходимые аналитику запросы;
- *интеллектуальный* — СППР осуществляет поиск функциональных и логических закономерностей в накопленных данных, построение моделей и правил, которые объясняют найденные закономерности и/или (с определенной вероятностью) прогнозируют развитие некоторых процессов.

Таким образом, обобщенная архитектура СППР может быть представлена следующим образом (рис. 1.2).

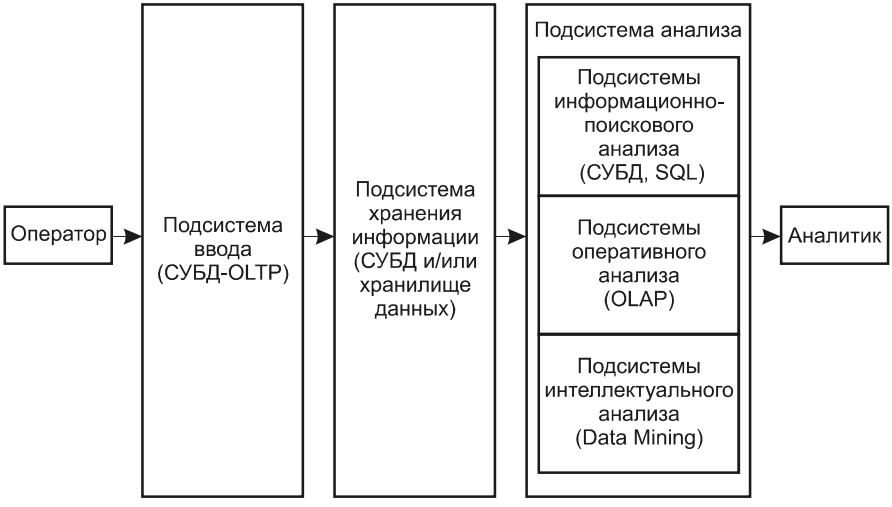

Рис. 1.2. Обобщенная архитектура системы поддержки принятия решений

Рассмотрим отдельные подсистемы более подробно.

Подсистема ввода данных. В таких подсистемах, называемых OLTP (On-line transaction processing), реализуется операционная (транзакционная) обработка данных. Для их реализации используют обычные системы управления базами данных (СУБД).

Подсистема хранения. Для реализации данной подсистемы используют современные СУБД и концепцию хранилищ данных.

Подсистема анализа. Данная подсистема может быть построена на основе:

- подсистемы информационно-поискового анализа на базе реляционных СУБД и статических запросов с использованием языка SQL (Structured Query Language);
- подсистемы оперативного анализа. Для реализации таких подсистем применяется технология оперативной аналитической обработки данных OLAP (On-line analytical processing), использующая концепцию многомерного представления данных;
- подсистемы интеллектуального анализа. Данная подсистема реализует методы и алгоритмы Data Mining ("добыча данных").

1.2. Базы данных — основа СППР

Ранее было отмечено, что для решения задач анализа данных и поиска решений необходимо накопление и хранение достаточно больших объемов данных. Этим целям служат базы данных (БД).

Внимание! База данных является моделью некоторой предметной области, состоящей из связанных между собой данных об объектах, их свойствах и характеристиках.

Системы, предоставляющие средства работы с БД, называются СУБД. Не решая непосредственно никаких прикладных задач, СУБД является инструментом для разработки прикладных программ, использующих БД.

Чтобы сохранять данные согласно какой-либо модели предметной области, структура БД должна максимально соответствовать этой модели. Первой такой структурой, используемой в СУБД, была иерархическая структура, появившаяся в начале 60-х годов прошлого века.

Иерархическая структура предполагала хранение данных в виде дерева. Это значительно упрощало создание и поддержку таких БД. Однако невозможность представить многие объекты реального мира в виде иерархии привела к использованию таких БД в сильно специализированных областях. Типичным

представителем (наиболее известным и распространенным) иерархической СУБД является Information Management System (IMS) фирмы IBM. Первая версия этого продукта появилась в 1968 году.

Попыткой улучшить иерархическую структуру была сетевая структура БД, которая предполагает представление данных в виде сети. Она основана на предложениях группы Data Base Task Group (DBTG) Комитета по языкам программирования Conference on Data Systems Languages (CODASYL). Отчет DBTG был опубликован в 1971 году.

Работа с сетевыми БД представляет гораздо более сложный процесс, чем работа с иерархической БД, поэтому данная структура не нашла широкого применения на практике. Типичным представителем сетевых СУБД является Integrated Database Management System (IDMS) компании Cullinet Software, Inc.

Наиболее распространены в настоящее время реляционные БД. Термин "реляционный" произошел от латинского слова *relatio* — отношение. Такая структура хранения данных построена на взаимоотношении составляющих ее частей. Реляционный подход стал широко известен благодаря работам Е. Кодда, которые впервые были опубликованы в 1970 году. В них Кодд сформулировал следующие 12 правил для реляционной БД.

1. **Данные представляются в виде таблиц** — БД представляет собой набор таблиц. Таблицы хранят данные, сгруппированные в виде рядов и колонок. Ряд представляет собой набор значений, относящихся только к одному объекту, хранящемуся в таблице, и называется записью. Колонка представляет собой одну характеристику для всех объектов, хранящихся в таблице, и называется полем. Ячейка на пересечении ряда и колонки представляет собой значение характеристики, соответствующей колонке для элемента соответствующего ряда.
2. **Данные доступны логически** — реляционная модель не позволяет обращаться к данным физически, адресуя ячейки по номерам колонки и ряда (нет возможности получить значение в ячейке колонка 2, ряд 3). Доступ к данным возможен только через идентификаторы таблицы, колонки и ряда. Идентификаторами таблицы и колонки являются их имена. Они должны быть уникальны в пределах, соответственно, БД и таблицы. Идентификатором ряда является первичный ключ — значения одной или нескольких колонок, однозначно идентифицирующих ряды. Каждое значение первичного ключа в пределах таблицы должно быть уникальным. Если идентификация ряда осуществляется на основании значений нескольких колонок, то ключ называется составным.
3. **NULL трактуется как неизвестное значение** — если в ячейку таблицы значение не введено, то записывается NULL. Его нельзя путать с пустой строкой или со значением 0.

4. **БД должна включать в себя метаданные** — БД хранит два вида таблиц: пользовательские таблицы и системные таблицы. В пользовательских таблицах хранятся данные, введенные пользователем. В системных таблицах хранятся метаданные: описание таблиц (название, типы и размеры колонок), индексы, хранимые процедуры и др. Системные таблицы тоже доступны, т. е. пользователь может получить информацию о метаданных БД.
5. **Должен использоваться единый язык для взаимодействия с СУБД** — для управления реляционной БД должен использоваться единый язык. В настоящее время таким инструментом стал язык структурных запросов SQL.
6. **СУБД должна обеспечивать альтернативный вид отображения данных** — СУБД не должна ограничивать пользователя только отображением таблиц, которые существуют. Пользователь должен иметь возможность строить виртуальные таблицы — представления (View). Представления являются динамическим объединением нескольких таблиц. Изменения данных в представлении должны автоматически переноситься на исходные таблицы (за исключением нередактируемых полей в представлении, например вычисляемых полей).
7. **Должны поддерживаться операции реляционной алгебры** — записи реляционной БД трактуются как элементы множества, на котором определены операции реляционной алгебры. СУБД должна обеспечивать выполнение этих операций. В настоящее время выполнение этого правила обеспечивает язык SQL.
8. **Должна обеспечиваться независимость от физической организации данных** — приложения, оперирующие с данными реляционных БД, не должны зависеть от физического хранения данных (от способа хранения, формата хранения и др.).
9. **Должна обеспечиваться независимость от логической организации данных** — приложения, оперирующие с данными реляционных БД, не должны зависеть от организации связей между таблицами (логической организации). При изменении связей между таблицами не должны меняться ни сами таблицы, ни запросы к ним.
10. **За целостность данных отвечает СУБД** — под целостностью данных в общем случае понимается готовность БД к работе. Различают следующие типы целостности:
    - *физическая целостность* — сохранность информации на носителях и корректность форматов хранения данных;
    - *логическая целостность* — непротиворечивость и актуальность данных, хранящихся в БД.

Потеря целостности базы данных может произойти от сбоев аппаратуры ЭВМ, ошибок в программном обеспечении, неверной технологии ввода и корректировки данных, низкой достоверности самих данных и т. д.

За сохранение целостности данных должна отвечать СУБД, а не приложение, оперирующее ими. Различают два способа обеспечения целостности: *декларативный* и *процедурный*. При декларативном способе целостность достигается наложением ограничений на таблицы, при процедурном — обеспечивается с помощью хранимых в БД процедур.

11. **Целостность данных не может быть нарушена** — СУБД должна обеспечивать целостность данных при любых манипуляциях, производимых с ними.
12. **Должны поддерживать распределенные операции** — реляционная БД может размещаться как на одном компьютере, так и на нескольких — распределенно. Пользователь должен иметь возможность связывать данные, находящиеся в разных таблицах и на разных узлах компьютерной сети. Целостность БД должна обеспечиваться независимо от мест хранения данных.

На практике в дополнение к перечисленным правилам существует требование минимизации объемов памяти, занимаемых БД. Это достигается проектированием такой структуры БД, при которой дублирование (избыточность) информации было бы минимальным. Для выполнения этого требования была разработана *теория нормализации*. Она предполагает несколько уровней нормализации БД, каждый из которых базируется на предыдущем. Каждому уровню нормализации соответствует определенная нормальная форма (НФ). В зависимости от условий, которым удовлетворяет БД, говорят, что она имеет соответствующую нормальную форму. Например:

- БД имеет 1-ю НФ, если каждое значение, хранящееся в ней, неразделимо на более примитивные (неразложимость значений);
- БД имеет 2-ю НФ, если она имеет 1-ю НФ, и при этом каждое значение целиком и полностью зависит от ключа (функционально независимые значения);
- БД имеет 3-ю НФ, если она имеет 2-ю НФ, и при этом ни одно из значений не предоставляет никаких сведений о другом значении (взаимно независимые значения) и т. д.

В заключение описания реляционной модели необходимо заметить, что она имеет существенный недостаток. Дело в том, что не каждый тип информации можно представить в табличной форме, например изображения, музыку и др. Правда, в настоящее время для хранения такой информации в реляционных СУБД сделана попытка использовать специальные типы полей — BLOB

(Binary Large OBjects). В них хранятся ссылки на соответствующую информацию, которая не включается в БД. Однако такой подход не позволяет оперировать информацией, не помещенной в базу данных, что ограничивает возможности по ее использованию.

Для хранения такого вида информации предлагается использовать постреляционные модели в виде объектно-ориентированных структур хранения данных. Общий подход заключается в хранении любой информации в виде объектов. При этом сами объекты могут быть организованы в рамках иерархической модели. К сожалению, такой подход, в отличие от реляционной структуры, которая опирается на реляционную алгебру, недостаточно формализован, что не позволяет широко использовать его на практике.

В соответствии с правилами Кодда СУБД должна обеспечивать выполнение операций над БД, предоставляя при этом возможность одновременной работы нескольким пользователям (с нескольких компьютеров) и гарантируя целостность данных. Для выполнения этих правил в СУБД используется механизм управления транзакциями.

Внимание! Транзакция — это последовательность операций над БД, рассматриваемых СУБД как единое целое. Транзакция переводит БД из одного целостного состояния в другое.

Как правило, транзакцию составляют операции, манипулирующие с данными, принадлежащими разным таблицам и логически связанными друг с другом. Если при выполнении транзакции будут выполнены операции, модифицирующие только часть данных, а остальные данные не будут изменены, то будет нарушена целостность. Следовательно, все операции, включенные в транзакцию, должны быть выполненными, либо не выполнена ни одна из них. Процесс отмены выполнения транзакции называется откатом транзакции (ROLLBACK). Сохранение изменений, производимых в результате выполнения операций транзакции, называется фиксацией транзакции (COMMIT).

Свойство транзакции переводить БД из одного целостного состояния в другое позволяет использовать понятие транзакции как единицу активности пользователя. В случае одновременного обращения пользователей к БД транзакции, инициируемые разными пользователями, выполняются не параллельно (что невозможно для одной БД), а в соответствии с некоторым планом ставятся в очередь и выполняются последовательно. Таким образом, для пользователя, по инициативе которого образована транзакция, присутствие транзакций других пользователей будет незаметно, если не считать некоторого замедления работы по сравнению с однопользовательским режимом.

Существует несколько базовых алгоритмов планирования очередности транзакций. В централизованных СУБД наиболее распространены алгоритмы,

основанные на синхронизированных захватах объектов БД. При использовании любого алгоритма возможны ситуации конфликтов между двумя или более транзакциями по доступу к объектам БД. В этом случае для поддержания плана необходимо выполнять откат одной или более транзакций. Это один из случаев, когда пользователь многопользовательской СУБД может реально ощутить присутствие в системе транзакций других пользователей.

История развития СУБД тесно связана с совершенствованием подходов к решению задач хранения данных и управления транзакциями. Развитый механизм управления транзакциями в современных СУБД сделал их основным средством построения OLTP-систем, основной задачей которых является обеспечение выполнения операций с БД.

OLTP-системы оперативной обработки транзакций характеризуются большим количеством изменений, одновременным обращением множества пользователей к одним и тем же данным для выполнения разнообразных операций — чтения, записи, удаления или модификации данных. Для нормальной работы множества пользователей применяются блокировки и транзакции. Эффективная обработка транзакций и поддержка блокировок входят в число важнейших требований к системам оперативной обработки транзакций.

К этому классу систем относятся, кстати, первые СППР — информационные системы руководства (ИСР, Executive Information Systems). Такие системы, как правило, строятся на основе реляционных СУБД, включают в себя подсистемы сбора, хранения и информационно-поискового анализа информации, а также содержат в себе предопределенное множество запросов для повседневной работы. Каждый новый запрос, непредусмотренный при проектировании такой системы, должен быть сначала формально описан, закодирован программистом и только затем выполнен. Время ожидания в таком случае может составлять часы и дни, что неприемлемо для оперативного принятия решений.

1.3. Неэффективность использования OLTP-систем для анализа данных

Практика использования OLTP-систем показала неэффективность их применения для полноценного анализа информации. Такие системы достаточно успешно решают задачи сбора, хранения и поиска информации, но они не удовлетворяют требованиям, предъявляемым к современным СППР. Подходы, связанные с наращиванием функциональности OLTP-систем, не дали удовлетворительных результатов. Основной причиной неудачи является противоречивость требований, предъявляемых к системам OLTP и СППР. Перечень основных противоречий между этими системами приведен в табл. 1.1.

Таблица 1.1

Характеристика	Требования к OLTP-системе	Требования к системе анализа
Степень детализации хранимых данных	Хранение только детализированных данных	Хранение как детализированных, так и обобщенных данных
Качество данных	Допускаются неверные данные из-за ошибок ввода	Не допускаются ошибки в данных
Формат хранения данных	Может содержать данные в разных форматах в зависимости от приложений	Единый согласованный формат хранения данных
Допущение избыточных данных	Должна обеспечиваться максимальная нормализация	Допускается контролируемая денормализация (избыточность) для эффективного извлечения данных
Управление данными	Должна быть возможность в любое время добавлять, удалять и изменять данные	Должна быть возможность периодически добавлять данные
Количество хранимых данных	Должны быть доступны все оперативные данные, требующиеся в данный момент	Должны быть доступны все данные, накопленные в течение продолжительного интервала времени
Характер запросов к данным	Доступ к данным пользователей осуществляется по заранее составленным запросам	Запросы к данным могут быть произвольные и заранее не оформлены
Время обработки обращений к данным	Время отклика системы измеряется в секундах	Время отклика системы может составлять несколько минут
Характер вычислительной нагрузки на систему	Постоянно средняя загрузка процессора	Загрузка процессора формируется только при выполнении запроса, но на 100 %
Приоритетность характеристик системы	Основными приоритетами являются высокая производительность и доступность	Приоритетными являются обеспечение гибкости системы и независимости работы пользователей

Рассмотрим требования, предъявляемые к системам OLTP и СППР более подробно.

Степень детализации хранимых данных — типичный запрос в OLTP-системе, как правило, выборочно затрагивает отдельные записи в таблицах, которые эффективно извлекаются с помощью индексов. В системах анализа, наоборот, требуется выполнять запросы сразу над большим количеством данных с широким применением группировок и обобщений (суммирования, агрегирования и т. п.).

Например, в стандартных системах складского учета наиболее часто выполняются операции вычисления текущего количества определенного товара на складе, продажи и оплаты товаров покупателями и т. д. В системах анализа выполняются запросы, связанные с определением общей стоимости товаров, хранящихся на складе, категорий товаров, пользующихся наибольшим и наименьшим спросом, обобщение по категориям и суммирование по всем продажам товаров и т. д.

Качество данных — OLTP-системы, как правило, хранят информацию, вводимую непосредственно пользователями систем (операторами ЭВМ). Присутствие "человеческого фактора" при вводе повышает вероятность ошибочных данных и может создать локальные проблемы в системе. При анализе ошибочные данные могут привести к неправильным выводам и принятию неверных стратегических решений.

Формат хранения данных — OLTP-системы, обслуживающие различные участки работы, не связаны между собой. Они часто реализуются на разных программно-аппаратных платформах. Одни и те же данные в разных базах могут быть представлены в различном виде и могут не совпадать (например, данные о клиенте, который взаимодействовал с разными отделами компании, могут не совпадать в базах данных этих отделов). В процессе анализа такое различие форматов чрезвычайно затрудняет совместный анализ этих данных. Поэтому к системам анализа предъявляется требование единого формата. Как правило, необходимо, чтобы этот формат был оптимизирован для анализа данных (нередко за счет их избыточности).

Допущение избыточных данных — структура базы данных, обслуживающей OLTP-систему, обычно довольно сложна. Она может содержать многие десятки и даже сотни таблиц, ссылающихся друг на друга. Данные в такой БД сильно нормализованы для оптимизации занимаемых ресурсов. Аналитические запросы к БД очень трудно формулируются и крайне неэффективно выполняются, поскольку содержат в себе представления (view), объединяющие большое количество таблиц. При проектировании систем анализа стараются максимально упростить схему БД и уменьшить количество таблиц, участвующих в запросе. С этой целью часто допускают денормализацию (избыточность данных) БД.

Управление данными — основное требование к OLTP-системам — обеспечить выполнение операций модификации над БД. При этом предполагается, что они должны выполняться в реальном режиме, и часто очень интенсивно. Например, при оформлении розничных продаж в систему вводятся соответствующие документы. Очевидно, что интенсивность ввода зависит от интенсивности покупок и в случае ажиотажа будет очень высокой, а любое промедление ведет к потере клиента. В отличие от OLTP-систем данные в системах анализа меняются редко. Единожды попав в систему, данные уже практически не изменяются. Ввод новых данных, как правило, носит эпизодический характер и выполняется в периоды низкой активности системы (например, раз в неделю на выходных).

Количество хранимых данных — как правило, системы анализа предназначены для анализа временных зависимостей, в то время как OLTP-системы обычно имеют дело с текущими значениями каких-либо параметров. Например, типичное складское приложение OLTP оперирует с текущими остатками товара на складе, в то время как в системе анализа может потребоваться анализ динамики продаж товара. По этой причине в OLTP-системах допускается хранение данных за небольшой период времени (например, за последний квартал). Для анализа данных, наоборот, необходимы сведения за максимально большой интервал времени.

Характер запросов к данным — в OLTP-системах из-за нормализации БД составление запросов является достаточно сложной работой и требует необходимой квалификации. Поэтому для таких систем заранее составляется некоторый ограниченный набор статических запросов к БД, необходимый для работы с системой (например, наличие товара на складе, размер задолженности покупателей и т. п.). Для СППР невозможно заранее определить необходимые запросы, поэтому к ним предъявляется требование обеспечить формирование произвольных запросов к БД аналитиками.

Время обработки обращений к данным — OLTP-системы, как правило, работают в режиме реального времени, поэтому к ним предъявляются жесткие требования по обработке данных. Например, время ввода документов продажи товаров (расходных, накладных) и проверки наличия продаваемого товара на складе должно быть минимально, т. к. от этого зависит время обслуживания клиента. В системах анализа, по сравнению с OLTP, обычно выдвигают значительно менее жесткие требования ко времени выполнения запроса. При анализе данных аналитик может потратить больше времени для проверки своих гипотез. Его запросы могут выполняться в диапазоне от нескольких минут до нескольких часов.

Характер вычислительной нагрузки на систему — как уже отмечалось, работа с OLTP-системами, как правило, выполняется в режиме реального

времени. В связи с этим такие системы нагружены равномерно в течение всего интервала времени работы с ними. Документы продажи или прихода товара оформляются в общем случае постоянно в течение всего рабочего дня. Аналитик при работе с системой анализа обращается к ней для проверки некоторых своих гипотез и получения отчетов, графиков, диаграмм и т. п. При выполнении запросов степень загрузки системы высокая, т. к. обрабатывается большое количество данных, выполняются операции суммирования, группирования и т. п. Таким образом, характер загрузки систем анализа является пиковым. На рис. 1.3 приведены данные фирмы Oracle для систем OLTP, на рис. 1.4 — для систем анализа, отражающие загрузку процессора в течение дня.

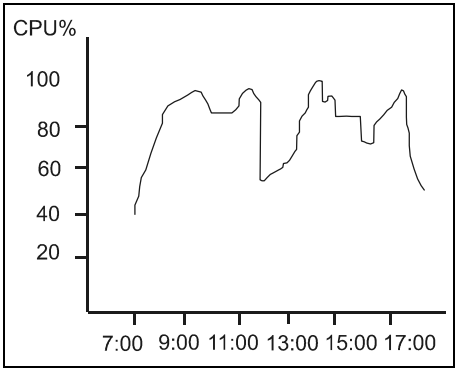

Рис. 1.3. Загрузка процессора для систем OLTP

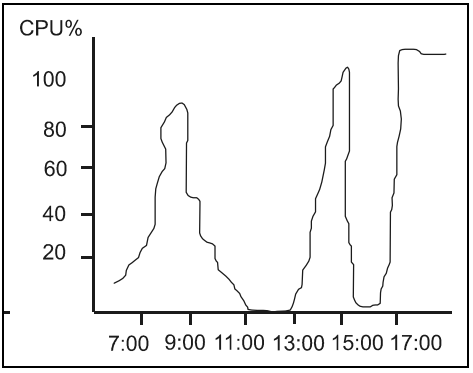

Рис. 1.4. Загрузка процессора для систем анализа

Приоритетность характеристик системы — для OLTP-систем приоритетным является высокая производительность и доступность данных, т. к. работа с ними ведется в режиме реального времени. Для систем анализа более приоритетными являются задачи обеспечения гибкости системы и независимости работы пользователей, другими словами то, что необходимо аналитикам для анализа данных.

Противоречивость требований к OLTP-системам и системам, ориентированным на глубокий анализ информации, усложняет задачу интеграции их как подсистем единой СППР. В настоящее время наиболее популярным решением этой проблемы является подход, ориентированный на использование концепции хранилищ данных.

Общая идея хранилищ данных заключается в разделении БД для OLTP-систем и БД для выполнения анализа и последующем их проектировании с учетом соответствующих требований.

Выводы

Из материала, изложенного в данной главе, можно сделать следующие выводы.

- СППР решают три основные задачи: сбор, хранение и анализ хранимой информации. Задача анализа разделяется на информационно-поисковый, оперативно-аналитический и интеллектуальный классы.
- Подсистемы сбора, хранения информации и решения задач информационно-поискового анализа в настоящее время успешно реализуются в рамках ИСР средствами СУБД. Для реализации подсистем, выполняющих оперативно-аналитический анализ, используется концепция многомерного представления данных (OLAP). Подсистема интеллектуального анализа данных реализует методы и алгоритмы Data Mining.
- Исторически выделяют три основные структуры БД: иерархическую, сетевую и реляционную. Первые две не нашли широкого применения на практике. В настоящее время подавляющее большинство БД реализует реляционную структуру представления данных.
- Основной недостаток реляционных БД заключается в невозможности обработки информации, которую нельзя представить в табличном виде. В связи с этим предлагается использовать постреляционные модели, например объектно-ориентированные.
- Для упрощения разработки прикладных программ, использующих БД, создаются системы управления базами данных (СУБД) — программное обеспечение для управления данными, их хранения и безопасности данных.
- В СУБД развит механизм управления транзакциями, что сделало их основным средством создания систем оперативной обработки транзакций (OLTP-систем). К таким системам относятся первые СППР, решающие задачи информационно-поискового анализа — ИСР.
- OLTP-системы не могут эффективно использоваться для решения задач оперативно-аналитического и интеллектуального анализа информации. Основная причина заключается в противоречивости требований к OLTP-системе СППР.
- В настоящее время для объединения в рамках одной системы OLTP подсистем и подсистем анализа используется концепция хранилищ данных. Общая идея заключается в разделении БД для OLTP-систем и БД для выполнения анализа.

ГЛАВА 2

Хранилище данных

2.1. Концепция хранилища данных

Стремление объединить в одной архитектуре СППР возможности OLTP-систем и систем анализа, требования к которым во многом, как следует из табл. 1.1, противоречивы, привело к появлению концепции *хранилищ данных* (ХД).

Концепция ХД так или иначе обсуждалась специалистами в области информационных систем достаточно давно. Первые статьи, посвященные именно ХД, появились в 1988 г., их авторами были Девлин и Мэрфи. В 1992 г. Уильман Г. Инмон подробно описал данную концепцию в своей монографии "Построение хранилищ данных".

В основе концепции ХД лежит идея разделения данных, используемых для оперативной обработки и для решения задач анализа. Это позволяет применять структуры данных, которые удовлетворяют требованиям их хранения с учетом использования в OLTP-системах и системах анализа. Такое разделение позволяет оптимизировать как структуры данных оперативного хранения (оперативные БД, файлы, электронные таблицы и т. п.) для выполнения операций ввода, модификации, удаления и поиска, так и структуры данных, используемые для анализа (для выполнения аналитических запросов). В СППР эти два типа данных называются соответственно *оперативными источниками данных* (ОИД) и хранилищем данных.

В своей работе Инмон дал следующее определение ХД.

Внимание! Хранилище данных — предметно-ориентированный, интегрированный, неизменчивый, поддерживающий хронологию набор данных, организованный для целей поддержки принятия решений.

Рассмотрим свойства ХД более подробно.

Предметная ориентация — является фундаментальным отличием ХД от ОИД. Разные ОИД могут содержать данные, описывающие одну и ту же предметную область с разных точек зрения (например, с точки зрения бухгалтерского учета, складского учета, планового отдела и т. п.). Решение, принятое на основе только одной точки зрения, может быть неэффективным или даже неверным. ХД позволяют интегрировать информацию, отражающую разные точки зрения на одну предметную область.

Предметная ориентация позволяет также хранить в ХД только те данные, которые нужны для их анализа (например, для анализа нет необходимости хранить информацию о номерах документов купли-продажи, в то время как их содержимое — количество, цена проданного товара — необходимо). Это существенно сокращает затраты на носители информации и повышает безопасность доступа к данным.

Интеграция — ОИД, как правило, разрабатываются в разное время несколькими коллективами с собственным инструментарием. Это приводит к тому, что данные, отражающие один и тот же объект реального мира в разных системах, описывают его по-разному. Обязательная интеграция данных в ХД позволяет решить эту проблему, приведя данные к единому формату.

Поддержка хронологии — данные в ОИД необходимы для выполнения над ними операций в текущий момент времени. Поэтому они могут не иметь привязки ко времени. Для анализа данных часто важно иметь возможность отслеживать хронологию изменений показателей предметной области. Поэтому все данные, хранящиеся в ХД, должны соответствовать последовательным интервалам времени.

Неизменяемость — требования к ОИД накладывают ограничения на время хранения в них данных. Те данные, которые не нужны для оперативной обработки, как правило, удаляются из ОИД для уменьшения занимаемых ресурсов. Для анализа, наоборот, требуются данные за максимально больший период времени. Поэтому, в отличие от ОИД, данные в ХД после загрузки только читаются. Это позволяет существенно повысить скорость доступа к данным как за счет возможной избыточности хранящейся информации, так и за счет исключения операций модификации. При реализации в СППР концепции ХД данные из разных ОИД копируются в единое хранилище. Собранные данные приводятся к единому формату, согласовываются и обобщаются. Аналитические запросы адресуются к ХД (рис. 2.1).

Такая модель неизбежно приводит к дублированию информации в ОИД и в ХД. Однако Инмон в своей работе утверждает, что избыточность данных,

хранящихся в СППР, не превышает 1 % ! Это можно объяснить следующими причинами.

При загрузке информации из ОИД в ХД данные фильтруются. Многие из них не попадают в ХД, поскольку лишены смысла с точки зрения использования в процедурах анализа.

Информация в ОИД носит, как правило, оперативный характер, и данные, потеряв актуальность, удаляются. В ХД, напротив, хранится историческая информация. С этой точки зрения дублирование содержимого ХД данными ОИД оказывается весьма незначительным.

В ХД хранится обобщенная информация, которая в ОИД отсутствует.

Во время загрузки в ХД данные очищаются (удаляется ненужная информация) и приводятся к единому формату. После такой обработки данные занимают гораздо меньший объем.

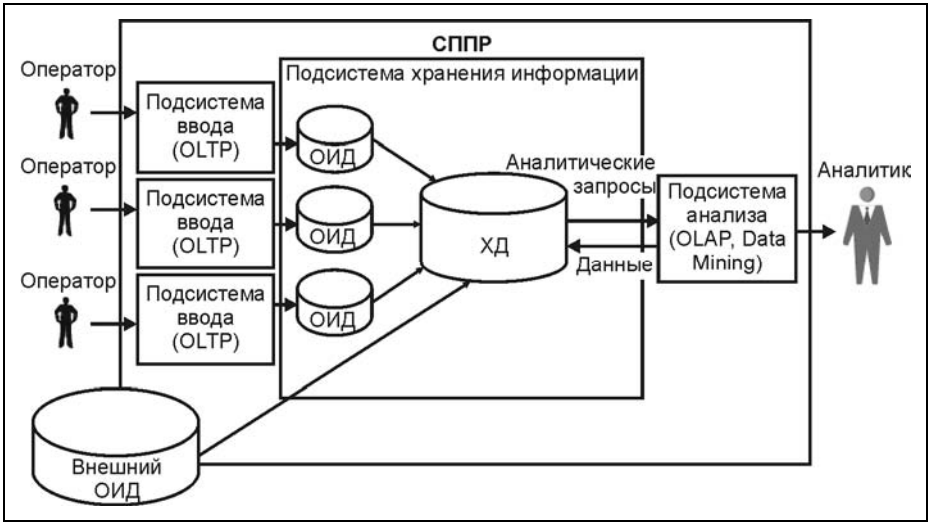

Рис. 2.1. Структура СППР с физическим ХД

Избыточность информации можно свести к нулю, используя виртуальное ХД. В данном случае в отличие от классического (физического) ХД данные из ОИД не копируются в единое хранилище. Они извлекаются, преобразуются и интегрируются непосредственно при выполнении аналитических запросов в оперативной памяти компьютера. Фактически такие запросы напрямую адресуются к ОИД (рис. 2.2). Основными достоинствами виртуального ХД являются:

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