Московский государственный университет экономики, статистики и информатики (МЭСИ)

Курс лекций «Базы данных»

Начальник отдела НИЧ, к.э.н., доцент Д.Г. Корнеев

2009 год

Лекции 1, 2. План лекций

1. **История возникновения баз данных (БД);**
2. **Определения БД;**
3. **Определение банка данных (БнД);**
4. **Понятие и функции Системы управления базами данных (СУБД);**
5. **Примеры классификации БнД и СУБД.**
6. **Структуры данных (иерархические, сетевые, реляционные);**
7. **Основные определения, используемые в реляционной модели;**
8. **Свойства отношений;**
9. **Определение понятия ключа;**
10. **Функциональные зависимости;**

Литература:

1. К. Дейт Введение в системы баз данных. 8-е изд., М.: СПб: Вильямс.- 2005 г.
2. С.Д. Кузнецов «Базы данных: языки и модели. Учебник – М.: ООО «Бином-Пресс», 2008 г.
3. С.М. Диго Базы данных: проектирование и использование: Учебник. – М.: Финансы и статистика, 2005 г.

История возникновения БД, определения БД, БнД, СУБД

Определения:

- ***База данных*** **– поименованная совокупность взаимосвязанных данных, находящихся под управлением СУБД.**

База данных — некоторый набор перманентных (постоянно хранимых) данных, используемых прикладными программными системами какого-либо предприятия (К. Дейт);

2. ***Банк данных*** **– система БД, программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.**

Система баз данных — это компьютеризированная система хранения записей, т.е. компьютеризированная система, основное назначение которой — хранить информацию, предоставляя пользователям средства ее извлечения и модификации (К.Дейт).

3. ***СУБД*** **– совокупность программных и языковых средств, предназначенных для создания БД, поддержания их в актуальном состоянии и организации доступа к ним различных пользователей.**

Функции СУБД

Определения:

- ***Целостность БД*** **– непротиворечивость информации, хранящейся в БД.**
- ***Транзакция*** **– законченная совокупность действий над БД, которая переводит БД из одного целостного состояния в другое целостное состояние.**
- ***Привилегия пользователя*** **– права пользователя на выполнение операций с данными (запись, корректировка, чтение, удаление), а также выполнение других действий над БД.**

Основные функции СУБД:

- **Создание БД;**
- **Создание пользователей и указание привилегий;**
- ***Обеспечение работы пользователей с БД с учетом привилегий*;**
- ***Поддержание целостности данных;***
- **Поддержание механизма транзакций;**
- **Журналирование;**
- **Управление оперативной памятью (буферизация).**

Пример классификации БнД

По характеру преобладающей обработки информации:

- ***OLTP (On-Line Transaction Processing) – системы оперативной обработки информации.***
- ***OLAP (On-Line Analytical Processing) – системы для сложной аналитической обработки информации.***

Характеристика	*OLTP*	*OLAP*
Преобладающие операции	Ввод данных, поиск	Анализ данных
Характер запросов	Преобладают простые транзакции и запросы	Сложные запросы
Характеристика хранимых данных	Оперативные, детализированные	Агрегированные, охватывающие большой период времени

Пример классификации СУБД

По «*мощности*» СУБД делятся на:

- ***«Настольные» - невысокие требования к техническим средствам, ориентация на конечного пользователя («дружелюбность» интерфейса, простота создания БД и обработки информации), низкая стоимость;***

Пример: MsAccess

- ***Корпоративные* - обеспечивают работу в распределенной среде, высокую производительность, имеют развитые средства администрирования и более широкие возможности поддержания целостности. Системы сложны, дороги, требуют значительных вычислительных мощностей.**

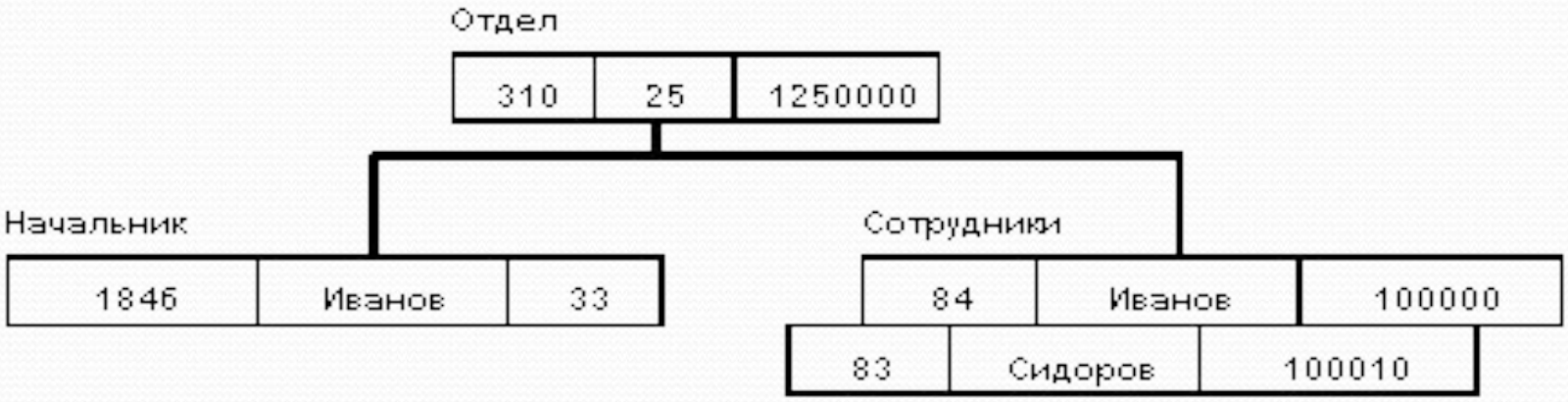
Примеры: Oracle, DB2, Sybase, Ms SQL Server, Progress

Иерархические структуры данных.

Иерархическую структуру данных можно представить в виде набора графов «древовидной структуры».

Основное правило: никакой потомок не может существовать без своего родителя.

Пример:



Иерархические СУБД.

Примерами типичных операций манипулирования иерархически организованными данными могут быть следующие:

- *найти указанный экземпляр типа дерева БД (например, отдел 310);*
- *перейти от одного экземпляра типа дерева к другому;*
- *перейти от экземпляра одного типа записи к экземпляру другого типа записи внутри дерева (например, перейти от отдела к первому сотруднику);*
- *перейти от одной записи к другой в порядке обхода иерархии;*
- *вставить новую запись в указанную позицию;*
- *удалить текущую запись.*

Типичным представителем (наиболее известным и распространенным) является СУБД IMS (Information Management System) компании IBM. Первая версия системы появилась в 1968 г.

Сетевые СУБД

Типичным представителем является Integrated Database Management System (IDMS) компании Cullinet Software, Inc., предназначенная для использования на машинах основного класса фирмы IBM под управлением большинства операционных систем. Архитектура системы основана на предложениях Data Base Task Group (DBTG) Комитета по языкам программирования Conference on Data Systems Languages (CODASYL), организации, ответственной за определение языка программирования Кобол. Отчет DBTG был опубликован в 1971 г., а в 70-х годах появилось несколько систем, среди которых IDMS.

Сетевые СУБД

Сетевые СУБД

Тип связи определяется для двух типов записи: предка и потомка. Экземпляр типа связи состоит из одного экземпляра типа записи предка и упорядоченного набора экземпляров типа записи потомка. Для данного типа связи L с типом записи предка P и типом записи потомка C должны выполняться следующие два условия:

- Каждый экземпляр типа P является предком только в одном экземпляре L;
- Каждый экземпляр C является потомком не более, чем в одном экземпляре L.

Сетевые СУБД

Примерный набор операций может быть следующим:

- Найти конкретную запись в наборе однотипных записей (инженера Сидорова);
- Перейти от предка к первому потомку по некоторой связи (к первому сотруднику отдела 310);
- Перейти к следующему потомку в некоторой связи (от Сидорова к Иванову);
- Перейти от потомка к предку по некоторой связи (найти отдел Сидорова);
- Создать новую запись;
- Уничтожить запись;
- Модифицировать запись;
- Включить в связь;
- Исключить из связи;
- Переставить в другую связь и т.д.

Достоинства и недостатки «ранних» СУБД

Сильные места ранних СУБД:

- Развитые средства управления данными во внешней памяти на низком уровне;
- Возможность построения вручную эффективных прикладных систем;

Недостатки:

- Слишком сложно пользоваться;
- Фактически необходимы знания о физической организации;
- Прикладные системы зависят от этой организации;
- Их логика перегружена деталями организации доступа к БД.

Реляционные СУБД

- Отношение → двухмерная таблица не содержащая строк - дубликатов;

Определение:

Реляционная база данных (от англ. Relation – отношение) – совокупность взаимосвязанных отношений.

- Запись – строка (ряд, запись,row, кортеж) таблицы;
- *Отношение – **множество** кортежей;*
- Атрибут (столбец);
- Домен – множество значений атрибута;

Свойства отношений

Отсутствие кортежей-дубликатов

- То свойство, что отношения не содержат кортежей-дубликатов, следует из определения отношения как *множества* кортежей. В классической теории множеств по определению каждое множество состоит из различных элементов.

Свойства отношений

Отсутствие упорядоченности кортежей

- Свойство отсутствия упорядоченности кортежей отношения также является следствием определения отношения-экземпляра как множества кортежей. Отсутствие требования к поддержанию порядка на множестве кортежей отношения дает дополнительную гибкость СУБД при хранении баз данных во внешней памяти и при выполнении запросов к базе данных. Это не противоречит тому, что при формулировании запроса к БД, например, на языке SQL можно потребовать сортировки результирующей таблицы в соответствии со значениями некоторых столбцов. Такой результат, вообще говоря, не отношение, а некоторый упорядоченный список кортежей.

Свойства отношений

Атомарность значений атрибутов

- **Значения всех атрибутов являются атомарными. Это следует из определения домена как потенциального множества значений простого типа данных, т.е. среди значений домена не могут содержаться множества значений (отношения). Принято говорить, что в реляционных базах данных допускаются только *нормализованные отношения или отношения*, представленные в *первой нормальной форме*. Потенциальным примером ненормализованного отношения является следующее:**

Свойства отношений

НОМЕР_ОТДЕЛА	ОТДЕЛ		
	СОТР_НОМЕР	СОТР_ИМЯ	СОТР_ЗАРП
310	2934	Иванов	112,000
	2935	Петров	112,500
313	2937	Федоров	110,000
315	2938	Иванова	112,000

Свойства отношений

СОТР_НОМЕР	СОТР_ИМЯ	СОТР_ЗАРП	СОТР_ОТД_НОМЕР
2934	Иванов	112,000	310
2935	Петров	144,000	310
2936	Сидоров	92,000	313
2937	Федоров	110,000	310

Функциональные зависимости

SCP

S#	CITY	P#	QTY
S1	London	P1	100
S1	London	P2	100
S2	Paris	P1	200
S2	Paris	P2	200
S3	Paris	P2	300
S4	London	P2	400
S4	London	P4	400
S4	London	P5	400

Функциональные зависимости

- Y **функционально зависимо** от X, что в символическом виде записывается как $X \rightarrow Y$

  (читается либо как "X **функционально определяет** Y", либо как "X стрелка Y") тогда и только тогда, когда каждое значение множества X отношения R связано точно с одним значением множества Y отношения R.

  Иначе говоря, если два кортежа отношения R совпадают по значению X, они совпадают и по значению Y.

Функциональные зависимости

- S#→ CITY; {S#,P#}→{QTY} ; {S#,P#}→{CITY};
- {S#,P#}→{CITY,QTY}; {S#,P#}→{S#}
- {S#,P#}→{S#,P#,CITY,QTY}; {S#} →{QTY}; {S#,P#}→{S#}; QTY → S#

SCP

S#	CITY	P#	QTY
S1	London	P1	100
S1	London	P2	100
S2	Paris	P1	200
S2	Paris	P2	200
S3	Paris	P2	300
S4	London	P2	400
S4	London	P4	400
S4	London	P5	400

Функциональные зависимости

- **Аксиомамы Армстронга.**

  Эти правила вывода могут формулироваться разными способами, из которых самым простым является следующий. Пусть A, B и C — произвольные подмножества множества атрибутов заданной переменной отношения R. Условимся также, что символическая запись AB означает объединение множеств A и B. Тогда правила вывода определяются следующим образом.

- 1. Правило **рефлексивности.** Если множество B является подмножеством множества A, то **A → B.**
- 2. Правило **дополнения.** Если A → B, то AC → BC.
- 3. Правило **транзитивности.** Если A → B и B→C, то A → C.

Декомпозиция

- **Декомпозиция без потерь.**

  Как уже упоминалось, процедура нормализации предусматривает разбиение, или *декомпозицию,* отношения на другие переменные отношения, причем декомпозиция должна быть *обратимой,* т.е. выполняться без потерь информации.

Пример декомпозиции

S

S#	STATUS	CITY
S3	30	Paris
S5	30	Athens

а)

SST

S#	STATUS
S3	30
S5	30

SC

S#	CITY
S3	Paris
S5	Athens

б)

SST

S#	STATUS
S3	30
S5	30

STC

STATUS	CITY
30	Paris
30	Athens

Понятие ключа

- Ключ - атрибут или совокупность атрибутов однозначно идентифицирующих строку отношения;
- Ключ, состоящий из одного атрибута, называется простым.
- Ключ, состоящий из нескольких атрибутов, называется составным.

Свойства ключа:

- **Уникальность;**
- **Неизбыточность;**
- **Не может содержать пустых значений.**