ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Лекция 7. Базы данных и системы управления ими. Транзакции. Методы и этапы проектирования ИС

Курс лекций

Базы данных

База данных (БД) - совокупность определенным образом организованной информации на какую-то тему. **Она составляет важнейший компонент ИС.**

Предметная область БД - ее тематика.

Примеры:

- база данных книжного фонда библиотеки;
- база данных кадрового состава учреждения;
- база данных законодательных актов в области права (Консультант плюс);
- база данных современной эстрадной песни.

Базы данных бывают фактографическими и документальными.

Фактографические БД содержат краткие сведения об описываемых объектах, представленные в строго определенном формате (первые 2 примера).

Документальные БД содержат информацию самого разного типа: текстовую, графическую, звуковую, мультимедийную (вторые 2 примера).

Границы между обоими типами БД размыты.

Компьютерная БД — это организованная совокупность данных, предназначенная для длительного хранения во внешней памяти ЭВМ и постоянного применения. Для хранения БД может использоваться как один компьютер, так и множество взаимосвязанных компьютеров. В последнем случае БД называется распределенной.

Вообще говоря, данные могут быть структурированными и неструктурированными.

Пример неструктурированных данных приведен на рисунке.

Компьютерные БД — это, как правило, структурированные данные (Кроме Больших данных).

Три типа структур:

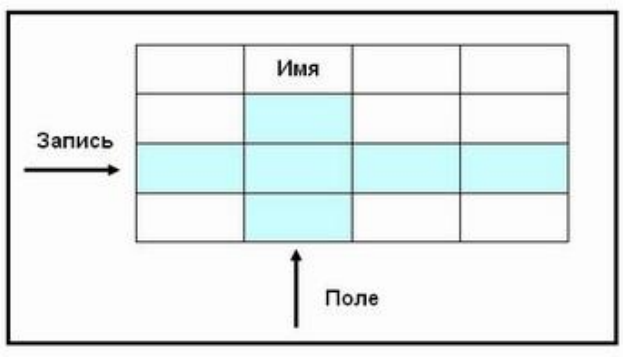
1. Реляционные БД – система связанных таблиц.
2. Иерархические БД - один тип объекта является главным, все нижележащие – подчиненными.
3. Сетевые БД - любой тип данных одновременно может быть главным и подчиненным.

Примеры структур БД

Реляционная:

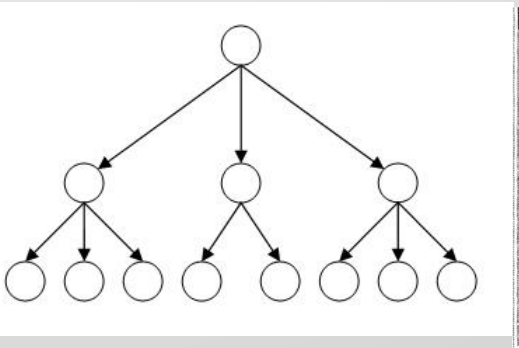
Строка – запись, столбец – поле. Поля – атрибуты объекта. Главный ключ в БД - поле или совокупность полей, значение которого не повторяется у разных записей.

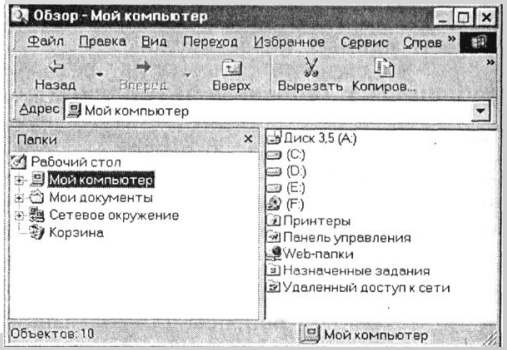
Пример – расписание, телефонный справочник.



Иерархическая:

Пример – структура фирмы, структура организации просмотра программ и файлов в компьютере.





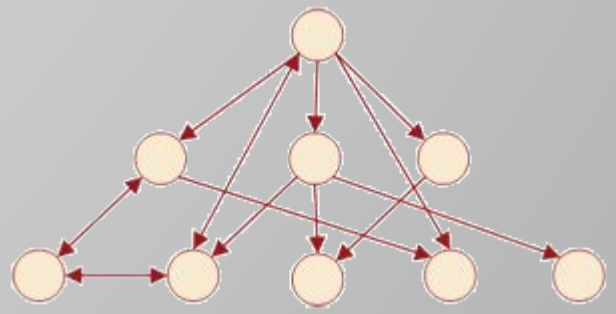
Простой ключ

Номер	Автор	Название	Год	Полка
001	Беляев А.Р.	Звезда КЭЦ	1990	3
002	Олеша Ю.К.	Избранное	1987	5
003	Беляев А.Р.	Избранное	1994	1

В БД «Домашняя библиотека» у разных книг могут совпадать значения полей, но инвентарный номер у каждой книги свой

Сетевая:

Пример – одноранговая компьютерная сеть, Интернет.



Достоверность и непротиворечивость данных

Проблема: как обеспечить работу крупных ИС с возможностью одновременной работы с нескольких рабочих мест в которых одновременно изменяется содержимое БД (вводятся, обновляются, удаляются данные и производится выборка из БД).

Результаты, получаемые от ИС, будут достоверны и непротиворечивы, если коллективная работа автоматически производится согласованно, т.е.:

- оператор, желающий обновить или удалить данные, не может выполнить операцию до тех пор, пока не закончится аналогичная операция над теми же данными, которую ранее начал, но еще не закончил другой оператор,
- оператор, формирующий отчеты, не сможет воспользоваться данными, которые начал, но еще не закончил формировать другой оператор,
- оператор формирующий данные, не сможет выполнить операцию над данными, которыми пользуется другой оператор, начавший, но не закончивший формировать отчет.

В этом случае все результаты, получаемые от ИС, будут соответствовать согласованному состоянию БД.

Классическая транзакция

Классическая транзакция – последовательность изменения БД и/или выборки из базы данных, воспринимаемая системой управления базой данных (СУБД) как единое (атомарное) действие. БД находится в целостном состоянии при условии успешного завершения транзакции. Если транзакция не может быть завершена, то СУБД производит полный ее откат, ликвидируя в БД результаты всех операций изменения, произведенных при выполнении транзакции.

В персональных БД возможен откат транзакции в случаях:

- нарушение целостности базы данных при окончании транзакции,
- аварийное выключение питания,
- аварии внешнего носителя БД.

Проще всего обрабатывать запросы последовательно, но существуют методы (метод полной сериализации), позволяющие максимально перемешивать запросы и операторы изменения БД, поступающие от разных транзакций, с тем лишь условием, что конечный результат выполнения всего набора транзакций будет эквивалентен результату их последовательного выполнения.

Иногда создается дополнительный файл внешней памяти - журнал БД, в который помещаются записи, соответствующие каждой операции изменения БД, а также записи о начале и конце каждой транзакции.

Системы управления базами данных

Для взаимодействия пользователя с базами данных используют системы управления данными (СУБД). СУБД - это ПО, предназначенное для создания на ЭВМ общей базы данных для множества приложений, поддержания ее в актуальном состоянии и обеспечения эффективного доступа пользователей к содержащимся в ней данным в рамках предоставленных им полномочий. СУБД используются для того, чтобы не повторять одни и те же способы хранения, выбора и модификации сложных данных в каждом приложении.

СУБД являются посредниками между логической структурой данных, необходимых разным приложениям, и физическими хранилищами данных (файловой системой персонального компьютера, сервера или группы серверов).

Функции СУБД:

- **управление данными во внешней памяти** - обеспечение необходимых структур внешней памяти как для хранения непосредственных данных, так и для служебных целей, например, для ускорения доступа к данным;
- **управление транзакциями** – условие особенно важное в многопользовательских СУБД; при соответствующем механизме управления транзакциями пользователь может почувствовать себя единственным пользователем СУБД;
- **журнализация и восстановление БД после сбоев** - способность СУБД восстановить последнее согласованное состояние после аппаратного или программного сбоя; во всех случаях придерживаются "упреждающей" записи в журнал суть которой заключается в том, что запись об изменении любого объекта БД должна попасть во внешнюю память журнала раньше, чем она попадет во внешнюю память основной части БД;
- **поддержка специальных языков БД,** содержащих все необходимые средства для работы с БД и обеспечивающих базовый пользовательский интерфейс с БД.

Требования к СУБД:

- **Производительность и готовность** - запросы от пользователя БД удовлетворяются с такой скоростью, которая требуется для использования данных (как правило, в реальном масштабе времени).
- **Минимальные затраты** - низкая стоимость создания, хранения и использования данных.
- **Простота и легкость использования** - доступ к данным должен быть простым, исключающим возможные ошибки со стороны даже не очень квалифицированного пользователя.
- **Вертикальное и горизонтальное масштабирование** - БД может увеличиваться и изменяться без нарушения имеющихся способов использования данных.
- **Целостность** - при работе с многими пользователями элементы данных и связи между ними не должны нарушаться, а аппаратные ошибки и случайные сбои не должны приводить к необратимым потерям данных.
- **Безопасность** - защита данных от случайного или преднамеренного несанкционированного доступа.

Язык для работы с реляционными БД

Для создания, модификации и управления данными в произвольной реляционной БД язык программирования, управляемой соответствующей СУБД, используется, язык **SQL** (structured query language), т.е язык структурированных запросов.

Традиционные СУБД

Традиционные СУБД ориентируются на требования ACID к транзакционной системе:

- **Atomicity** - атомарность (транзакция не фиксируется в системе частично);
- **Consistency** – согласованность (транзакция достигающая своего нормального завершения сохраняет согласованность базы данных);
- **Isolation** – изолированность (во время выполнения транзакции параллельные транзакции не оказывают влияние на её результат);
- **Durability –** надёжность (изменения, сделанные успешно завершённой транзакцией, должны остаться сохранёнными после возвращения системы в работу).

Но свойства ACID практически невозможно обеспечить в системах с многомиллионной веб-аудиторией, вроде поисковых систем или amazon.com.

Нереляционные СУБД

Нереляционные СУБД (no relational database management system - NRDBMS) обычно называют **NoSQL** (No или Not Only Structured Query Language — не структурированный язык запросов). Он ориентируется на требования BASE, а не ACID :

- **Basic availability** - базовая доступность (каждый запрос гарантированно завершается успешно или безуспешно).
- **Soft state** - гибкое состояние (состояние системы может изменяться со временем, даже без ввода новых данных, для достижения согласования данных);
- **Eventual consistency** - согласованность в конечном счёте (данные могут быть некоторое время рассогласованы).

Согласно теореме Брюера в распределённых вычислениях можно обеспечить только два из трёх свойств. Проектировщики NoSQL-систем жертвуют согласованностью данных ради достижения двух других свойств. Нереляционные СУБД позволяют работать с БД, имеющими иерархическую или сетевую структуру.

Функционально блочный метод создания ИС

Для каждого приложения разрабатываются либо независимые функциональные продукты, либо используются готовые системы или пакеты, объединяемые с помощью специальных интерфейсных модулей. Прикладные программы, реализующие функциональные требования каждого блока, разрабатываются в отрыве друг от друга и часто в разное время. Соответственно, каждое приложение имеет и **свои собственные средства обеспечения доступа** к БД.

Недостатки: не обеспечивается концептуальное единство создаваемой ИС, что снижет надежность, производительность, целостность, технологичность и усложняет эксплуатацию.

В настоящее время применяется, как правило, для сравнительно небольших и достаточно простых ИС.

Достоинства использования СУБД:

- возможность сделать программы ввода, модификации и поиска данных независимыми от программ содержательной обработки приложений;
- минимизировать объем хранимых данных путем исключения их дублирования;
- избежать противоречий в хранимых данных;
- обеспечить сохранность и целостность информации:
- многократно использовать одни и те же данные различными прикладными программами;
- обеспечить гибкость и адаптивность структуры данных к изменяющимся информационным потребностям пользователей;
- обеспечить защиту данных от несанкционированного доступа.

Этапы проектирования ИС

- Концептуальное проектирование – разработка модели наиболее высокого уровня без ориентации на какую-либо конкретную СУБД и модель данных.
- Логическое проектирование - создание логической модели организации и взаимодействия информации в корпорации, для которой создается ИС на основе анализа потоков данных внутри корпорации, оценки объемов информации, поддерживаемых и обрабатываемых ИС.
- Физическое проектирование – выбор способов реализации ИС, методов доступа к данным, конкретной СУБД. Может влиять на логическую структуру.
- Проектирование и разработка интерфейса - выбор инструментальных средств для быстрой эффективной реализации. Производится, как правило, с помощью полуфабрикатов.

Резюме:

1. Основу любой ИС составляют БД - совокупность определенным образом организованной информации на какую-то тему; для доступа к и управления ими используется специализированное ПО – СУБД. Тематика БД называется ее предметной областью.
2. БД бывают фактографическими и документальными, хотя различие между этими типами БД условна и размыта.
3. Компьютерные БД, как правило структурированные. Существует 3 основных типа структур – реляционные, иерархические и сетевые.
4. В основе ИС лежит, с одной стороны, среда хранения и доступа к данным т.е. БД и СУБД, а с другой - простой, удобным, легко осваиваемый интерфейс.
5. При функционально блочном методе создания ИС для каждого приложения разрабатываются независимые функциональные продукты, объединяемые с помощью специальных интерфейсных модулей.
6. Более современный способ создания ИС – использование СУБД для поддержки независимости, целостности и непротиворечивости данных в условиях коллективного использования.
7. Этапы проектирования ИС включают этапы концептуального, логического, физического проектирования и разработку интерфейса.