

РАЗДЕЛ 3. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Тема 3.1. Базы данных

План:

1. Виды хранения данных
2. Виды СУБД
3. Базы данных
4. Реляционная база данных

1. Виды хранения данных

Локальные или персональные **СУБД (системы управления базами данных)**, когда все компоненты информационной системы (программное обеспечение и база данных) размещаются на одном компьютере. При этом работа с базой данных осуществляется в однопользовательском режиме. К таким **СУБД** относятся **FoxPro, dBase** и ряд других. В настоящее время разработка локальных **СУБД** не производится по причине повсеместного внедрения сетевых информационных технологий.

Многопользовательские **СУБД**, когда база данных информационной системы, как правило, размещенная на выделенном для этого компьютере – сервере – доступна двум и более пользователям по локальной вычислительной сети или по сети Интернет. К **СУБД** этого класса относятся **Oracle, Microsoft SQL Server, MySQL, PostgreSQL, Informix, Cache** и др

Персональные **СУБД** широко используются в информационно-справочных системах, не работающих с большими массивами данных. Использование персональных **СУБД** позволяет не только эффективно организовывать работу с данными при отсутствии локальной вычислительной сети, но и поддерживать независимую работу клиентского приложения за счет наличия собственных форматов хранения данных. Краткая характеристика некоторых персональных **СУБД** приведена в таблице.

Наименование	Краткая характеристика
Lotus Approach	Позволяет выполнять все виды обработки данных. Имеет очень простой интерфейс. СУБД тесно интегрирована с базами данных Notes и электронными таблицами Lotus 1-2-3 . Поддерживает технологию электронного обмена сообщениями MAPI .
MS Access	Полнофункциональная СУБД , обладающая богатым набором визуальных средств, многочисленными мастерами и мощным языком программирования Visual Basic for Applications . Имеет гибкую систему подготовки отчетов. Поддерживаются технологии ODBC и OLE 2.0 . СУБД тесно интегрирована со всеми приложениями MS Office .
MS Visual FoxPro	Одна из наиболее быстрых персональных СУБД , сочетающая технологию xBase и объектно-ориентированный язык программирования. Имеет богатый набор визуальных средств разработки и мастеров для быстрого построения приложений и отчетов. Поддерживаются технологии ActiveX, ODBC и OLE 2.0 . Позволяет создавать OLE-сервера и имеет очень развитые средства разработки и поддержки приложений “клиент-сервер”.
Paradox	Поддерживает все виды работы с данными. Для визуального выполнения стандартных задач имеется специальное средство Experts . Наделен собственным достаточно сложным языком ObjectPAL . Поддерживает технологии OLE 2.0, ActiveX, MAPI и ODBC .

Табл.1

2. Виды СУБД

По способу доступа к БД различают файл-серверные и клиент серверные СУБД.

При использовании **файл-серверной технологии** файлы данных располагаются централизованно на выделенном компьютере – файл-сервере.

На каждом клиентском компьютере (рабочей станции) устанавливается клиентская программа и экземпляр ядра **СУБД**.

Доступ к данным осуществляется через локальную вычислительную сеть. Обработка данных выполняется непосредственно на рабочих станциях.

Недостатком этой архитектуры является большая загрузка локальной вычислительной сети и высокие требования, предъявляемые к аппаратно-программным средствам рабочих станций, а **сомнительным преимуществом** – низкая нагрузка на центральный процессор сервера.

На данный момент **файл-серверная** технология доступа к данным считается устаревшей.

К файл-серверным СУБД относятся, например, **Microsoft Access, Borland Paradox** и др.

Клиент-серверные СУБД разделяют функции приложения пользователя (называемого клиентом) и сервера. Приложение-клиент формирует запрос к серверу, на котором расположена **БД**, на структурном языке запросов **SQL**, являющимся промышленным стандартом в мире реляционных БД.

Удаленный сервер принимает запрос и переадресует его **SQL-серверу БД** – специальной программе, управляющая удаленной базой данных.

SQL-сервер обеспечивают интерпретацию запроса, его выполнение, формирование результата и выдачу его приложению-клиенту.

При этом ресурсы клиентского компьютера не участвуют в физическом выполнении запроса. Клиентский компьютер лишь отправляет запрос к серверной БД и получает результат, после чего интерпретирует его необходимым образом и представляет пользователю.

Таким образом, к аппаратно-программным средствам клиента не предъявляются особые требования. Так как клиентскому приложению посылается результат выполнения запроса, по сети «путешествуют» только те данные, которые необходимы клиенту.

В итоге снижается нагрузка на сеть. Поскольку выполнение запроса происходит там же, где хранятся данные (на сервере), нет необходимости в пересылке больших пакетов данных.

Кроме того, **SQL-сервер**, если это возможно, оптимизирует полученный запрос таким образом, чтобы он был выполнен в минимальное время с наименьшими накладными расходами. Всё это повышает быстродействие системы и снижает время ожидания результата запроса.

При использовании клиент-серверной технологии существенно повышается степень сохранности данных, поскольку правила целостности данных определяются на сервере и являются едиными для всех приложений, использующих эту **БД**. Все перечисленное можно отнести к преимуществам клиент-серверных технологий. В качестве весьма условного недостатка необходимо указать на высокие требования, предъявляемые к аппаратно-программным средствам сервера, который должен одновременно работать с десятками и даже сотнями клиентов.

Клиент-серверными СУБД являются: **Oracle, Microsoft SQL Server, Firebird, Interbase** и др.

В последнее время в связи с распространением Интернет-технологий, при разработке информационных систем широко используется **трехуровневая архитектура клиент-серверной технологии**.

В ней на клиентском уровне (т.н. «тонкий клиент») используется универсальный **браузер**, на уровне сервера находится сервер **БД**, а на промежуточном уровне располагаются **Web-сервер**.

Такое архитектурное решение позволяет уменьшить сетевой трафик, делает компоненты взаимозаменяемыми и повышает уровень безопасности.

Браузер посылает **Web-серверу** запросы на доставку **Web-страниц** или данных. Web-сервер обслуживает заявки на Web-страницы, преобразует их в форму, понятную серверу БД, и передает их серверу БД.

Затем сервер БД выполняет работу по обслуживанию запроса и возвращает результат Web-серверу, который преобразует их в формат, приемлемый для браузера, и передает их по сети клиенту.

Физически **WEB-сервер** и **СУБД** могут располагаться как на одном сервере, так и на разных серверах. В последнем случае производительность информационной системы еще более увеличивается из-за распределения вычислительной нагрузки между серверами.

Выбор системы управления баз данных для построения конкретной информационной системы представляет собой сложную многопараметрическую задачу и является одним из важных этапов ее создания.

Выбранный программный продукт должен удовлетворять как текущим, так и будущим потребностям конечных пользователей.

При этом следует учитывать финансовые затраты на приобретение необходимого оборудования (серверов и рабочих станций), самой СУБД, разработку необходимого программного обеспечения на ее основе, а также обучение персонала.

Практика показывает, что решение об использовании той или иной СУБД зачастую принимает один человек, причем даже не руководитель предприятия, а руководитель информационного подразделения или группы разработчиков.

При этом он может опираться отнюдь не на технические критерии, а на собственные предпочтения, например на уровень знаний или опыт работы с тем или иным программным обеспечением.

3. Базы данных

База данных (БД) — структурированный организованный набор данных, описывающих характеристики какой-либо системы объектов.

Система управления базами данных (СУБД) — специализированная программа (чаще — комплекс программ), предназначенная для манипулирования базой данных.

В зависимости от математической модели представления данных, БД и, соответственно, СУБД разделяются на сетевые, иерархические, реляционные, объектно-реляционные и объектно-ориентированные.

В настоящий момент самыми распространенными являются **реляционные СУБД**.

Их реализации существуют на всех платформах (от персональных компьютеров до мэйнфреймов), для всех операционных систем и для всех применений — от простейших продуктов, предназначенных для ведения картотек индивидуального пользования, до сложнейших распределенных многопользовательских систем.

Все эти СУБД основаны на реляционной модели данных, разработанной **Э.Ф.Коддом** в 70-х годах XX столетия.

В ее основе лежит мощный и вместе с тем изящный математический аппарат реляционной алгебры, которая в свою очередь базируется на целом ряде математических дисциплин, среди которых — логика, исчисление предикатов, теория множеств и пр.

На поверхностном уровне реляционную базу данных можно представить как набор взаимосвязанных таблиц, каждая из которых описывает какой либо набор объектов — «отношений» в терминах реляционной БД

4. Реляционная база данных

На рисунках Рис.1, Рис.2 и Рис.3 показана схема организации таблицы реляционной базы данных. В строках (кортежах или **записях**) таблицы хранятся параметры (атрибуты) объектов. На этапе проектирования БД каждому атрибуту присваивается свой **тип данных**. Например, атрибут «**Специальность**», содержащий название специальностей, имеет тип данных **Текстовый**, а атрибут «**УМК**» имеет тип **Гиперссылка**.

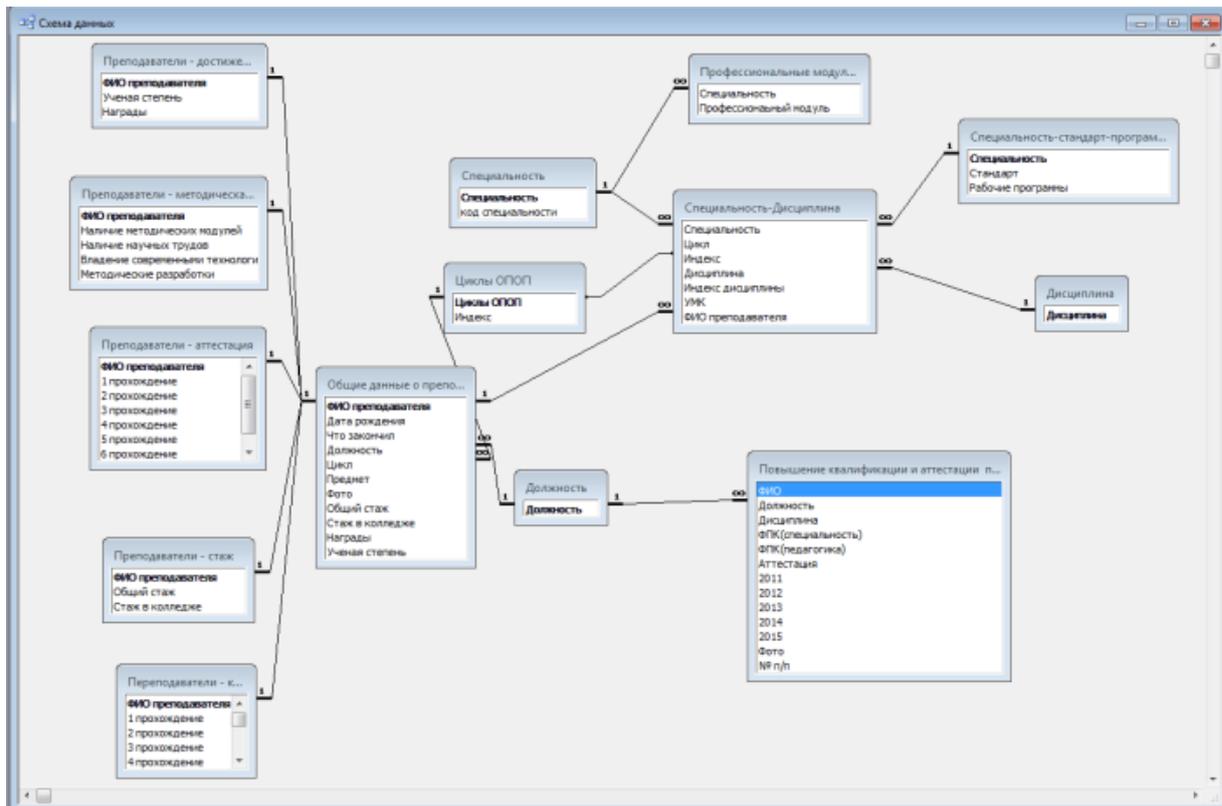


Рис. 1

Запрос Специальность-стандарт-рабочая программа : запрос на выборку

Специальность	Стандарт	Рабочие программы
Акушерское дело	ФГОС-3\Акуш дело.pdf	Рабочие программы\Акуш дело
Лабораторная диагностика	ФГОС-3\Лаб диагн.pdf	Рабочие программы\Лаб диагн
Лечебное дело	ФГОС-3\Леч дело.pdf	Рабочие программы\Леч дело
Сестринское дело	ФГОС-3\Сестр дело.pdf	Рабочие программы\Сестр дело
Стоматология ортопедическая	ФГОС-3\Стом.ортопед.pdf	Рабочие программы\Стом.ортоп
Стоматология профилактическая	ФГОС-3\Стом.проф.pdf	Рабочие программы\Стом.проф
Фармация	ФГОС-3\Фармация.pdf	Рабочие программы\Фармация
*		

Рис. 2

db2012(1) : база данных (формат Access 2000)

Специальность-Дисциплина : таблица

Имя поля	Тип данных
Специальность	Текстовый
Цикл	Текстовый
Индекс	Текстовый
Дисциплина	Текстовый
Индекс дисциплины	Текстовый
УМК	Гиперссылка
ФИО преподавателя	Текстовый

Рис. 3

Домен

Допустимое множество значений каждого атрибута называется доменом. Например, домен «**ФИО преподавателя**» в нашем примере определен как «**короткая строка**», но в число его значений допустимо вхождение только тех строк, которые могут соответствовать фамилии (в частности, такие строки не могут начинаться с мягкого или твердого знака и не могут содержать цифр).

Домен «**Индекс**», определенный как «**короткое целое без знака**» не может иметь отрицательное значение.

Первичный ключ

Отношения (таблицы реляционной базы данных) не могут содержать кортежей-дубликатов. Для этого каждый кортеж имеет свой уникальный в пределах таблицы-отношения идентификатор – **первичный ключ**. Зачастую первичный ключ генерируется самой СУБД. Такой ключ называется **суррогатным** и представляет собой дополнительное служебное поле, в которое заносятся заведомо уникальные значения из возрастающей числовой последовательности.

Главное достоинство суррогатного ключа состоит в том, что он никогда не изменяется, поскольку не является информативным полем таблицы (не несёт никакой информации об описываемом записью объекте).

Первичный ключ в таблице является базовым уникальным идентификатором для записей. Значение первичного ключа используется везде, где нужно указать на конкретную запись. На использовании первичных ключей основана организация связей между таблицами реляционной БД.

Внешний ключ

Чтобы организовать между двумя таблицами связь типа «**один к одному**» или «**один ко многим**» в одну из связываемых таблиц добавляют поле, содержащее значение первичного ключа записи в связанной таблице (такое поле называют внешним ключом).

Состав базы данных

Базы данных включают в себя десятки и сотни таблиц-отношений, которые подразделяются на два класса: **информационные таблицы и таблицы-справочники**.

В информационные таблицы заносятся параметры-атрибуты объектов.

Справочники же содержат заведомо известные или ранее определенные значения вариантов этих параметров.

Наличие в базе данных справочников обеспечивает:

- **Компактное хранение данных** в информационных таблицах. В них заносятся и хранятся не сами значения параметров, а их коды, взятые из справочников.
- **Однозначность вариантов значений** параметров объектов. При наличии справочника пользователь выбирает необходимое значение параметра из предлагаемого ему списка вариантов, а не заносит их вручную.
- **Совместимость различных баз данных**. Использование одних и тех же справочников в различных БД.

Проектирование базы данных

Проектирование баз данных для информационных систем – сложная, трудоемкая и длительная работа, требующая высокой квалификации участвующих в ней специалистов.

В процессе создания и функционирования информационных систем потребности пользователей постоянно изменяются или уточняются, что еще более усложняет разработку и сопровождение таких систем.

Эти проблемы могут быть существенно облегчены за счет применения современных структурных методов, среди которых центральное место занимают методологии структурного анализа, в частности – **CASE-технологии**.

Они представляет собой совокупность методологий анализа, проектирования, разработки и сопровождения сложных систем и поддерживается комплексом взаимосвязанных средств автоматизации

CASE-технология – это инструментарий для системных аналитиков, разработчиков и программистов, заменяющий бумагу и карандаш компьютером, автоматизируя процесс проектирования и разработки программного обеспечения.