**1. Системы управления базами данных**

***Система управления базой данных (СУБД*)** — это комплекс языковых и программных средств, предназначенных для создания, ведения и совместного использования базы данных одним или многими пользователями.

**Под *архитектурой*** *СУБД*понимают совокупность основных характеристик компьютера и программных средств, обеспечивающих функционирование СУБД.

**Структурно все объекты СУБД разбиты на семь основных типов:**

* Таблицы;
* Формы;
* Запросы;
* Отчеты;
* Страницы;
* Макросы;
* Модули.

**Таблица –** это отформатированное место памяти, которое создается с целью хранения в нем данных. Таблицы определяют структуру БД и хранят всю информацию, имеющуюся в БД. Чтобы не повредить данные, находящиеся в таблицах, у пользователя нет прямого доступа к таблицам.

**Форма –** это шаблон, с помощью которого осуществляется ввод информации в базу данных и, при необходимости, просмотр этой информации на экране монитора. Эти шаблоны, как правило, соответствуют привычному для пользователя виду документа. Формы можно распечатывать на твердую копию.

**Запрос –** это требование, которое формируется пользователем для осуществления выборки нужных данных из одной или нескольких таблиц, связанных между собой. С помощью запроса можно также изменить (обновить, удалить, добавить) данные в существующие таблицы или на базе этих таблиц создать новые таблицы. Запрос может формироваться с помощью готового образца (бланка) с помощью средств *QBE (Query By Example – запрос по образцу)* или же в соответствии с инструкцией специального языка структурированных запросов – SQL.

**Отчет –** это отформатированный выходной документ, содержащий необходимые сведения и предназначенный для вывода на печать. Отчет формируется на основе запроса и тех условий отбора записей, которые в этом запросе прописаны.

**Страница –** это специальный объект базы данных, фактически являющийся страницей доступа к данным и представляет собой специализированный тип Web-страниц, предназначенных для просмотра и работы через Интернет с данными, хранящимися в БД офисного пакета MS Access. Сама по себе страница не является базой данных, но содержит компоненты, через которые осуществляется связь, переданной Web-страницы с базой данных. Страница доступа к данным может также включать данные других источников, например, MS Excel.

Объекты типа **Макросы** и **Модули** относятся к средствам автоматизации обработки данныхв БД. Они создаются на языке VBA (Visual Basic for Application), который встроен в инструментальные средства офиса.

**Макрос –** это набор команд, предназначенных для выполнения каких-либо действий, реализуемых одной командой. Удобство использования макросов состоит в том, что он позволяет объединить в одном блоке разрозненные операции обработки данных, при этом достаточно одного нажатия клавиши вызова макроса, чтобы выполнился значительный объем работы по управлению сложным процессом, например, распечатка отчета.

**Модуль –** это своего рода «контейнер» для кода VBA, в котором содержатся одна или несколько подпрограмм (процедур или функций), обеспечивающих выполнение тех или иных операций.

**СУБД классифицируются** по многим признакам, основными из которых являются:

* По поддерживаемым моделям данных (иерархическая, сетевая, реляционная);
* По наличию встроенных языков манипулирования данными (NDL – язык сетевых БД; SQL – язык структурированных запросов для реляционных БД и т.д.);
* По наличию языков для разработки прикладных программ (включаемые языки высокого уровня; базовые языки СУБД – *VBA, Lotus Script, Object PAL* и т.д.);
* По наличию объектов СУБД: таблиц, запросов, форм, отчетов, макросов, модулей, страниц доступа к удаленным данным и др.;
* По уровню использования с другими информационными системами (локальные, технологии файл/сервер, технологии клиент/сервер, распределенные и т.д.);
* По возможности интеграции данных с другими БД и СУБД;
* По возможности интеграции и обработки данных разных видов (числовой, текстовой, графической, звуковой, видеоинформации) в соответствии с технологией OLE, которая используется в Windows и Internet.

**2. Логические модели данных**

**Логическая модель данных** - описание объектов предметной области, их атрибутов и взаимосвязей между ними в том объеме, в котором они подлежат непосредственному хранению в базе данных системы.

Логическая модель строится в несколько этапов с постепенным приближением к оптимальному для данных условий варианту. Эффективность такой модели зависит от того, насколько близко она отображает изучаемую предметную область. К предметной области относятся объекты (документы, счета, операции над ними и пр.), а также характеристики данных объектов, их свойства, взаимодействие и взаимное влияние.

Таким образом, при построении логической модели данных сначала выявляются те объекты, которые интересуют пользователей проектируемой базы данных. Затем для каждого объекта формулируются характеристики и свойства, достаточно полно описывающие данный объект. Эти характеристики в дальнейшем будут отражены в базе данных как соответствующие поля.

Выделяют следующие **виды логических моделей** базы данных:

* иерархическая
* сетевая
* реляционная

**Иерархическая модель** представляет собой древовидную структуру, которая выражает связи подчинения нижнего уровня высшему. Это облегчает поиск информации в том случае, если запросы имеют такую же структуру.

**Сетевая модель** отличается от предыдущей наличием также и горизонтальных связей. Это усложняет как модель, так и саму базу данных и средства ее управления.

**Реляционная модель** представляет хранимую информацию в виде таблиц, над которыми возможно выполнение логических операций (операций реляционной алгебры).

**3.Реляционные базы данных**

**Реляционная база данных** — это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа. Строка таблицы содержит данные об одном объекте (например, товаре, клиенте), а столбцы таблицы описывают различные характеристики этих объектов — атрибутов (например, наименование, код товара, сведения о клиенте). Записи, т. е. строки таблицы, имеют одинаковую структуру — они состоят из полей, хранящих атрибуты объекта. Каждое поле, т. е. столбец, описывает только одну характеристику объекта и имеет строго определенный тип данных. Все записи имеют одни и те же поля, только в них отображаются различные информационные свойства объекта.

В реляционной базе данных каждая таблица должна иметь **первичный ключ** — поле или комбинацию полей, которые единственным образом идентифицируют каждую строку таблицы. Если ключ состоит из нескольких полей, он называется составным. Ключ должен быть уникальным и однозначно определять запись. По значению ключа можно отыскать единственную запись. Ключи служат также для упорядочивания информации в БД.

Таблицы реляционной БД должны отвечать требованиям нормализации отношений. **Нормализация отношений** — это формальный аппарат ограничений на формирование таблиц, который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых в базе данных, уменьшает трудозатраты на ведение базы данных.

Над реляционными таблицами возможны следующие **операции**:

* Объединение таблиц с одинаковой структурой. Результат— общая таблица: сначала первая, затем вторая (конкатенация).
* Пересечение таблиц с одинаковой структурой. Результат — выбираются те записи, которые находятся в обеих таблицах.
* Вычитание таблиц с одинаковой структурой. Результат — выбираются те записи, которых нет в вычитаемом.
* Выборка (горизонтальное подмножество). Результат — выбираются записи, отвечающие определенным условиям.
* Проекция (вертикальное подмножество). Результат — отношение, содержащее часть полей из исходных таблиц.
* Декартово произведение двух таблиц Записи результирующей таблицы получаются путем объединения каждой записи первой таблицы с каждой записью другой таблицы.

Реляционные таблицы могут быть связаны друг с другом, следовательно, данные могут извлекаться одновременно из нескольких таблиц. Таблицы связываются между собой для того, чтобы в конечном счете уменьшить объем БД. Связь каждой пары таблиц обеспечивается при наличии в них одинаковых столбцов.

Существуют следующие **типы информационных** связей:

* один-к-одному;
* один-ко-многим;
* многие-ко-многим.

*Связь один-к-одному* предполагает, что одному атрибуту первой таблицы соответствует только один атрибут второй таблицы и наоборот.

*Связь один-ко-многим*предполагает, что одному атрибуту первой таблицы соответствует несколько атрибутов второй таблицы.

*Связь многие-ко-многим* предполагает, что одному атрибуту первой таблицы соответствует несколько атрибутов второй таблицы и наоборот.

**4. Языки баз данных**

1. *Процедурные* языки требуют от конечного пользователя знания синтаксиса, но их плюс в том, что на них можно написать стандартные программы и они гибки в использовании.
2. *Непроцедурные* языки предлагают пользователю шаблоны запросов или систему меню, из которых пользователь выбирает нужные ему вопросы.

СУБД также может иметь библиотеки программ, позволяющие делать вызовы из языков высокого уровня типа С и может вызывать внешние программы на языках высокого уровня или макроассемблер.

**В ранних СУБД** чаще всего выделялись два языка - язык определения схемы БД (**SDL** - Schema Definition Language) и язык манипулирования данными (**DML** - Data Manipulation Language).

**В современных СУБД** обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных. Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык SQL (Structured Query Language).

**5.SQL** («язык структурированных запросов») — декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

**Язык SQL сочетает средства SDL и DML**, т.е. позволяет определять схему реляционной БД и манипулировать данными. Изначально SQL был основным способом работы пользователя с базой данных и позволял выполнять следующий набор операций:

* создание в базе данных новой таблицы;
* добавление в таблицу новых записей;
* изменение записей;
* удаление записей;
* выборка записей из одной или нескольких таблиц (в соответствии с заданным условием);
* изменение структур таблиц.

Со временем SQL усложнился — обогатился новыми конструкциями, обеспечил возможность описания и управления новыми хранимыми объектами (например, индексы, представления, триггеры и хранимые процедуры) — и стал приобретать черты, свойственные языкам программирования.

**6. СУБД Access**

**Microsoft Access** — реляционная система управления базами данных корпорации Microsoft. Входит в состав пакета Microsoft Office. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA, в самом Access можно писать приложения, работающие с базами данных.

**Основные компоненты MS Access**:

* построитель таблиц;
* построитель экранных форм;
* построитель SQL-запросов (язык SQL в MS Access не соответствует стандарту ANSI);
* построитель отчётов, выводимых на печать.

Они могут вызывать скрипты на языке VBA, поэтому MS Access позволяет разрабатывать приложения и БД практически «с нуля» или написать оболочку для внешней БД.

**Взаимодействие с другими СУБД**

Инструменты MS Access, которые позволяют реализовать взаимодействие, называются «связанные таблицы» (связь с таблицей СУБД) и «запросы к серверу» (запрос на диалекте SQL, который «понимает» СУБД).