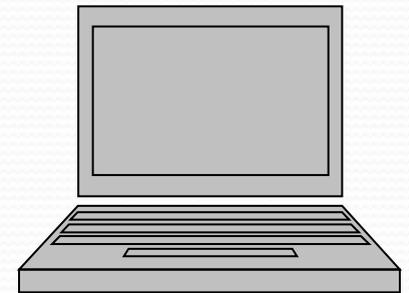


РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ СОЦИАЛЬНЫЙ УНИВЕРСИТЕТ



Тема: Базы данных и системы управления
БД

5.1. Основные понятия

Базы данных представляют собой совокупность сведений о конкретных объектах реального мира в какой-либо предметной области (*под которой принято понимать часть реального мира, подлежащего изучению для организации управления и автоматизации*).

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро производить выборку с произвольным сочетанием признаков, т.е. извлекать информацию. Сделать это возможно в случае, если данные **структурированы** (*систематизированы*).

Неструктурированными называют данные, записанные в простой текстовой форме (*организовать поиск необходимых данных, хранящихся в неструктурированном виде довольно сложно, а упорядочить подобную информацию практически не реально*).

Чтобы систематизировать данные и автоматизировать поиск, необходимо выработать определённые соглашения о способах представления данных. После проведения несложной структуризации информация о студентах, например, будет выглядеть в виде таблицы:

№ З/к	Фамилия	Имя	Отчество	Факультет	Курс	Группа
Д11222	Иванов	Иван	Иванович	Социальный	1	СОЦ-Д-1-1
Д22111	Петров	Петр	Петрович	Экономический	2	ЭКО-Д-1-1
Д33222	Сидоров	Сидор	Сидорович	Юридический	3	Юр-Д-1-1

База данных (БД) представляет собой поименованную совокупность структурированных данных, относящихся к определенной предметной области. Управление базами данных, осуществляются централизованно с помощью специального программного инструментария - системы управления базами данных. **Система управления базами данных (СУБД)** - это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации (*СУБД можно рассматривать как программную оболочку, которая находится между базой данных и пользователем*). Она обеспечивает централизованный контроль защиты и целостности данных, доступ к данным, их обработку, формирование отчетов и другие операции и процедуры.

Входящие в состав современных СУБД средства совместно обеспечивают выполнение следующих действий:

- описание данных, их структуры, первичный ввод, пополнение информации в базе данных;
- удаление устаревшей информации из базы данных;
- корректировку данных для поддержания их актуальности;
- упорядочение (*сортировку*) данных по некоторым признакам;
- поиск информации по некоторым признакам;
- подготовку и генерацию отчетов;
- защиту информации и разграничение доступа пользователей к ней;
- резервное сохранение и восстановление базы данных;
- поддержку интерфейса с пользователями;
- оказание помощи;
- защиту от необдуманных действий;
- предоставляют несколько вариантов выполнения одних и тех же действий.

Архитектурно СУБД состоит из двух основных компонентов: **языка описания данных** (*предназначен для формализованного описания типов данных, их структур и взаимосвязей*) и **языка манипулирования данными** (*выполняющего операции с базой данных - наполнение, обновление, удаление, выборку информации*). Помимо них к СУБД следует отнести средства (*или языки*) **подготовки отчетов** (*позволяющие подготовить отчеты на основе информации, найденной в базе данных, по заданным формам*).

По технологии обработки данных БД подразделяются на следующие типы: централизованные и распределенные. Централизованная база данных хранится в памяти одной вычислительной системы. Распределенная база данных состоит из нескольких, возможно пересекающихся или даже дублирующих друг друга частей, хранимых в различных ЭВМ вычислительной сети.

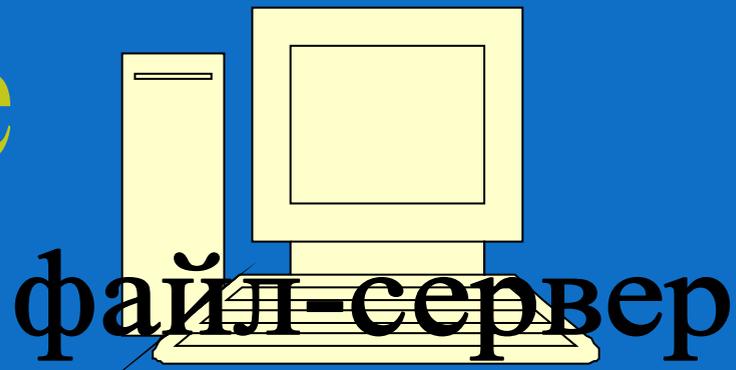
По способу доступа к данным базы данных делятся на базы данных с локальным доступом (БД хранится на том же компьютере, где и обрабатывается) и базы данных с удаленным (сетевым) доступом (БД хранится на одном из компьютеров, входящих в компьютерную сеть).

Системы централизованных баз данных с сетевым доступом полагают такие архитектуры:

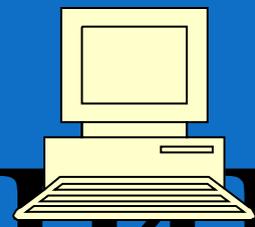
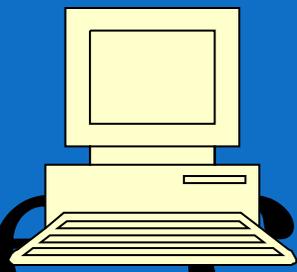
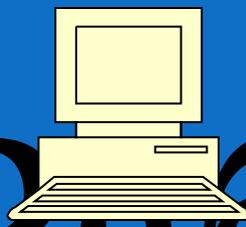
- ▶ файл-сервер;
- ▶ клиент-сервер.

Файл-сервер. Архитектура систем БД с сетевым доступом имеет условием выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине хранится совместно используемая централизованная БД. Все другие машины сети выполняют функции рабочих станций. Файлы БД в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится обработка (*при большой интенсивности доступа к одним и тем же данным производительность информационной системы падает*).

Хранение



Передача файлов БД для обработки



Рабочие станции

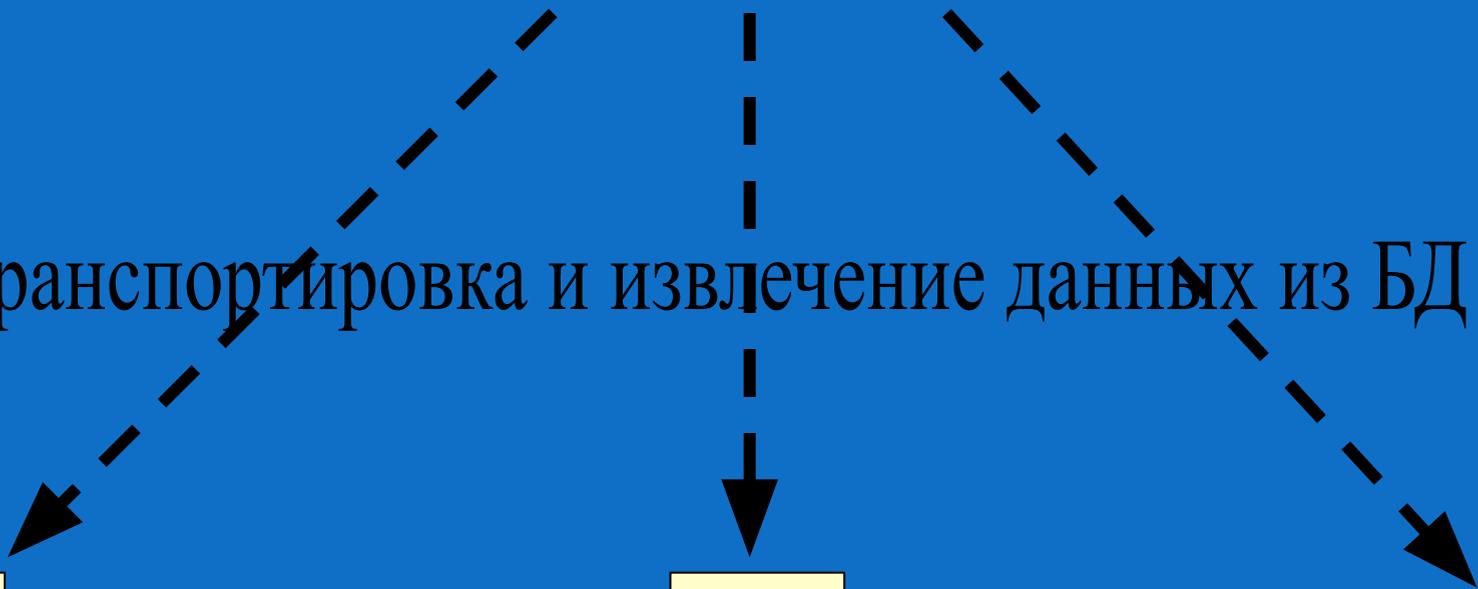
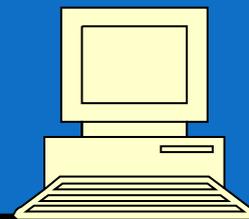
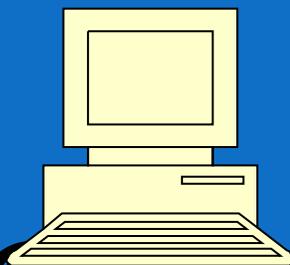
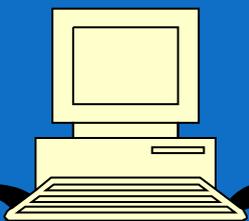
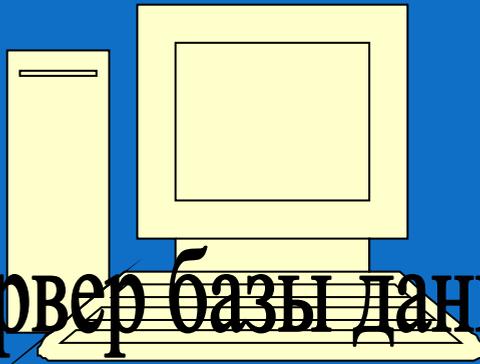
Клиент-сервер. В этой концепции подразумевается, что помимо хранения централизованной БД центральная машина (сервер базы данных) должна обеспечивать выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиентом (рабочей станцией), порождает поиск и извлечение данных на сервере. Извлеченные данные транспортируются по сети от сервера к клиенту. Спецификой архитектуры клиент-сервер является использование языка запросов SQL (Structured Query Language – язык структурированных запросов).

Хранение и обработка

Сервер базы данных

Транспортировка и извлечение данных из БД

Рабочие станции



В структуре информационных систем существует более широкое понятие (*чем база данных*) - это «Банк данных» (*data bank*) - хранилище реальной совокупности данных, система специальным образом организованных данных (*баз данных*), программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

Банк данных включает в свой состав вычислительную систему, базу данных (БД) и систему управления базой данных (СУБД). Кроме того в БНД предусматривается словарь данных, использование специальных методов доступа к одним и тем же данным разных пользователей и др.

Банк данных

В структуре информационных систем существует более широкое понятие (чем база данных) – это **Банк данных** - система специальным образом организованных данных (баз данных), программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных. Банк данных включает в свой состав вычислительную систему, базу данных (БД) и систему управления базой данных (СУБД). Кроме того в БнД входят словарь данных, использование специальных методов доступа к одним и тем же данным разных пользователей и др.



Ядром БД является база данных. Программные средства БД представляют собой комплекс, обеспечивающий взаимодействие всех частей информационной системы при ее функционировании.

Основу программных средств БД представляет программная компонента (СУБД). В ней можно выделить ядро, трансляторы и утилиты. Большинство СУБД работает в среде универсальных операционных систем и взаимодействует с ОС при обработке обращений к БД, поэтому можно считать, что ОС также входит в состав БД. Для обработки запросов к базе данных пишутся соответствующие программы, которые представляют прикладное программное обеспечение БД.

С банком данных в процессе его функционирования взаимодействуют пользователи разных категорий. Кроме конечных пользователей, присутствуют и специфические пользователи БД - специалисты, обеспечивающие создание, функционирование и развитие БД. В такую группу входят администратор и обслуживающий персонал.

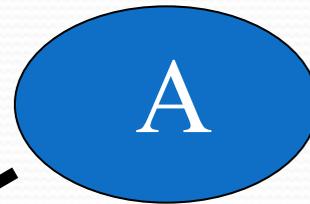
5.2. Виды моделей данных

Основой любой базы данных является модель данных (представляет множество структур данных и взаимосвязи между ними). Различают иерархическую, сетевую и реляционную модели данных (базы данных базируются на их использовании, на комбинации этих моделей или на некотором их подмножестве).

Иерархическая модель данных

Иерархическая структура представляет совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево).

Уровень 1



Уровень 2



...



...



Уровень 3



..



..



..



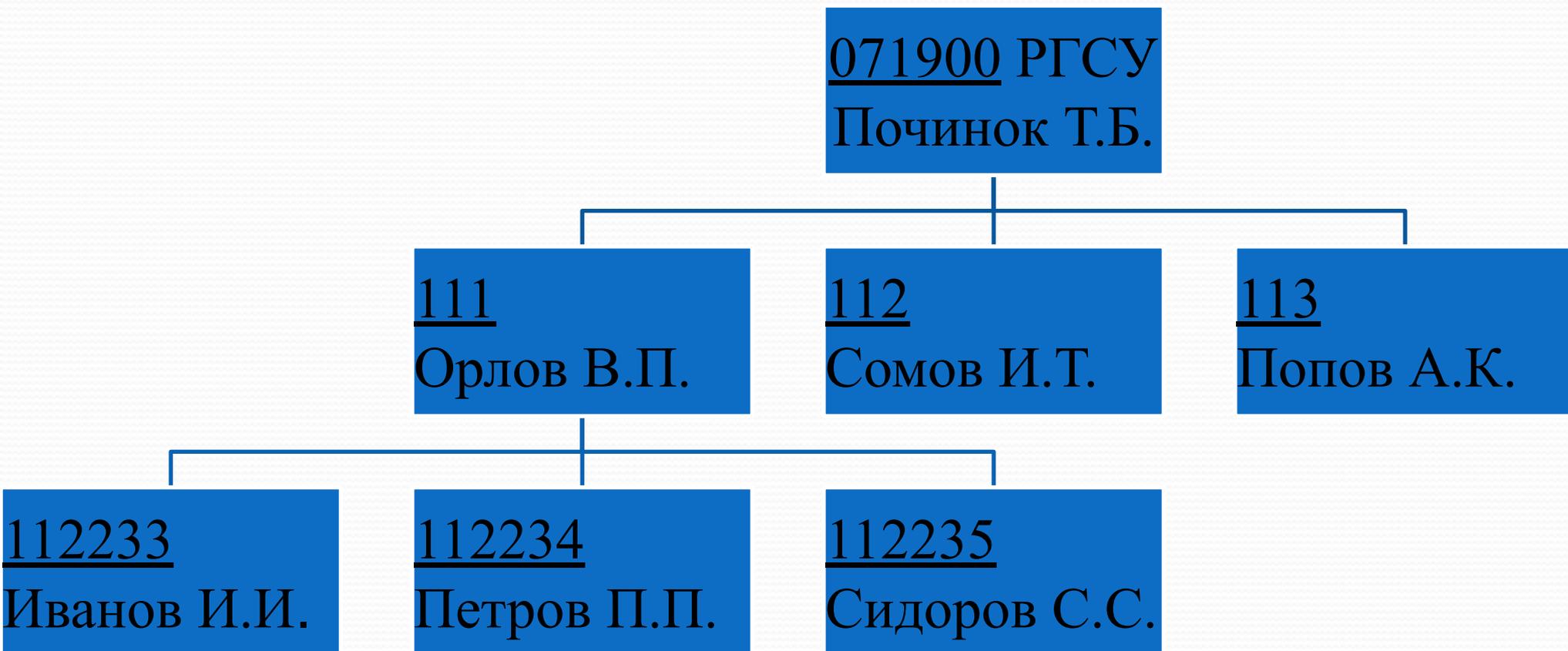
К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь.

Узел - это совокупность атрибутов данных, описывающих некоторый объект (*на схеме узлы представляются вершинами графа*). В иерархической модели должно соблюдаться правило: каждый порожденный узел не может иметь больше одного порождающего узла; в структуре может быть только один непорожденный узел - **корень**. Узлы, не имеющие выходных стрелок, носят название **листьев**. Узел интегрируется как **запись**. Для поиска необходимой записи нужно двигаться от корня к листьям, т.е. сверху вниз, что значительно упрощает доступ.

Связь – линия, связывающая узлы нижнего уровня с одним узлом вышележащего уровня.

Уровень – номер слоя узлов, отсчитанный от корня.

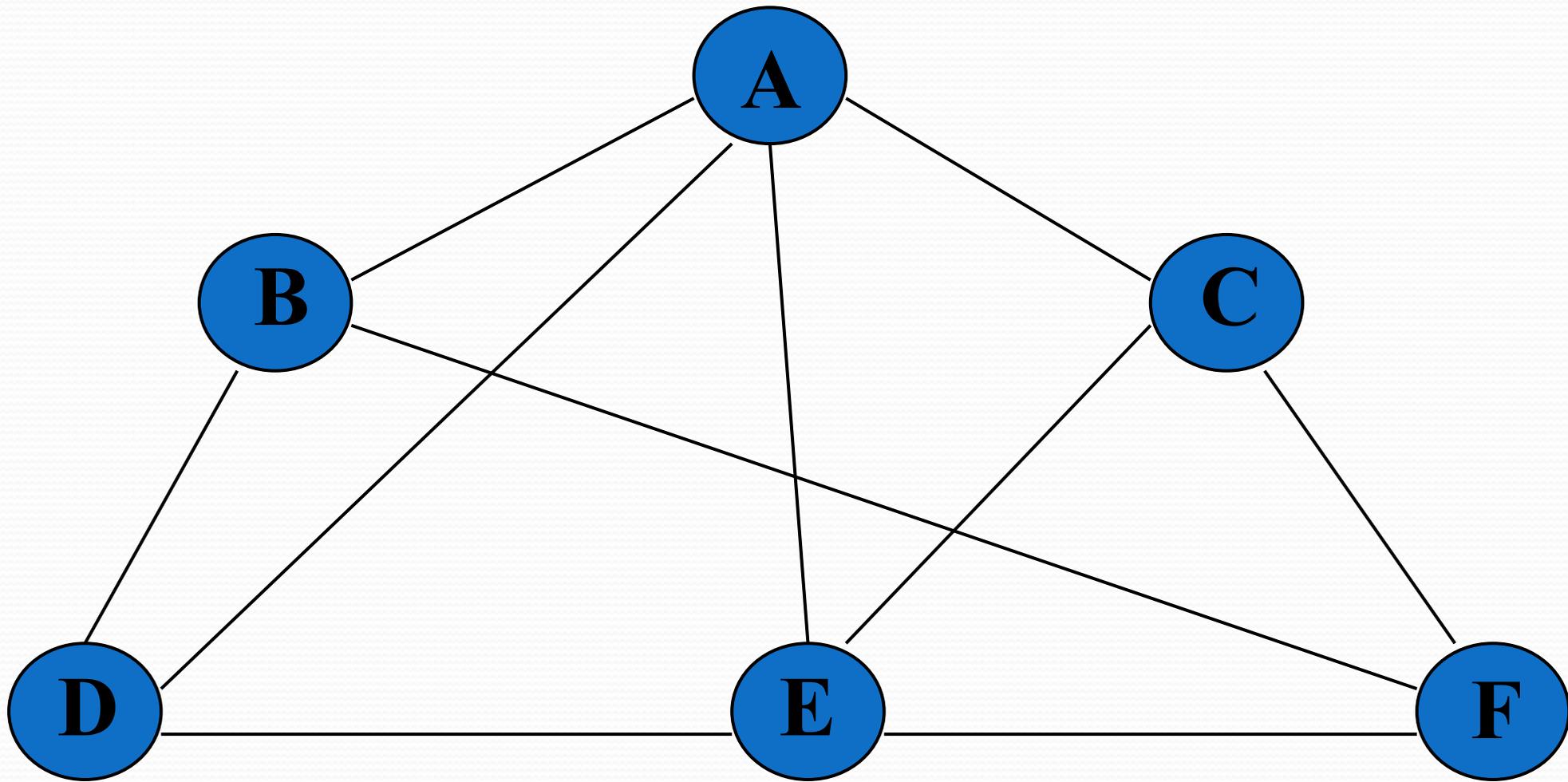
Пример иерархической структуры БД



Студент (номер зачетной книжки,
фамилия, имя, отчество)

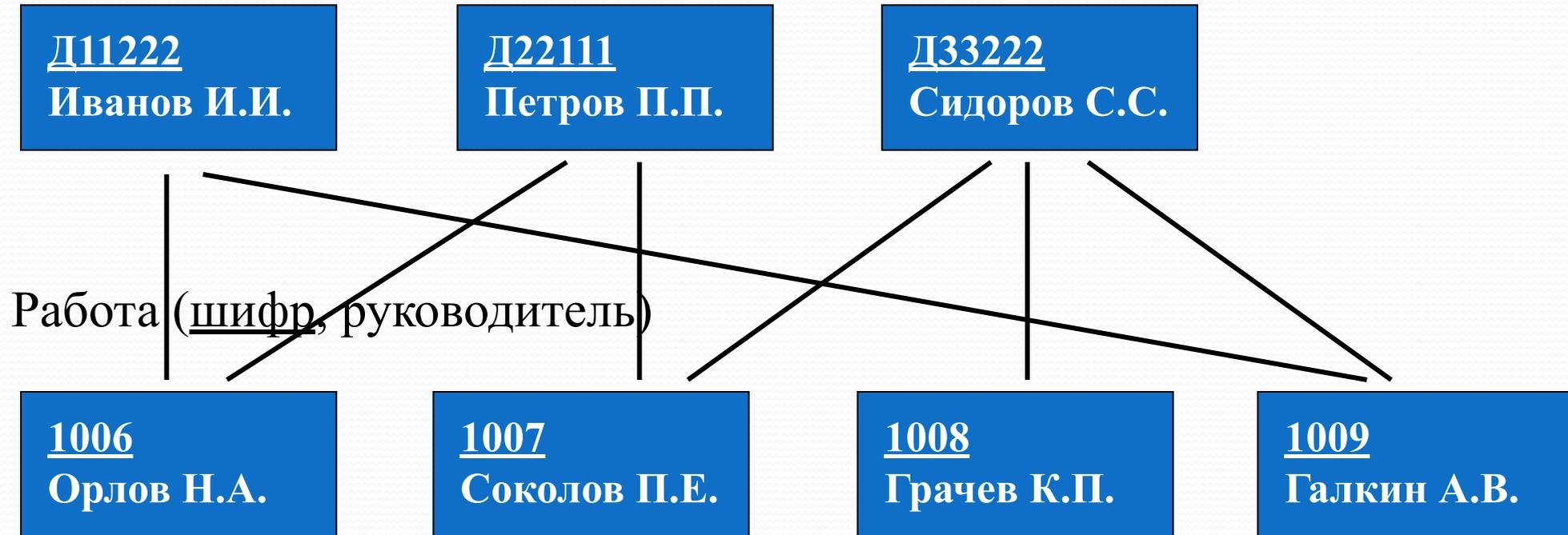
Достоинствами рассматриваемой модели являются наличие промышленных СУБД, поддерживающих данную модель, простота понимания используемого принципа иерархии, обеспечение определенного уровня независимости данных. Недостатками же являются жесткая фиксированность взаимосвязей между элементами данных, вследствие чего любые изменения связей требуют изменения структуры. Быстрота доступа в иерархической модели достигнута за счет потери информационной гибкости. Указанные недостатки ограничивают применение иерархической структуры.

Сетевая структура (модель данных) имеет те же составляющие, что и иерархическая (*уровень, узел, связь*), но каждый узел может быть связан с любым другим узлом (*представлена в виде диаграммы связей*).



Пример сетевой структуры БД

Студент (номер зачетной книжки, фамилия, имя, отчество)



Примером сложной сетевой структуры может служить структура базы данных, содержащей сведения о студентах, участвующих в научно-исследовательских работах (НИР). Возможно участие одного студента в нескольких НИР, а также нескольких студентов в разработке одной НИР.



Достоинство сетевой модели БД - большая информационная гибкость по сравнению с иерархической моделью. Основным же недостатком сетевой модели состоит в ее сложности. Сохраняется и общий для обеих моделей изъян - достаточно жесткая структура, что препятствует развитию информационной базы и системы управления.

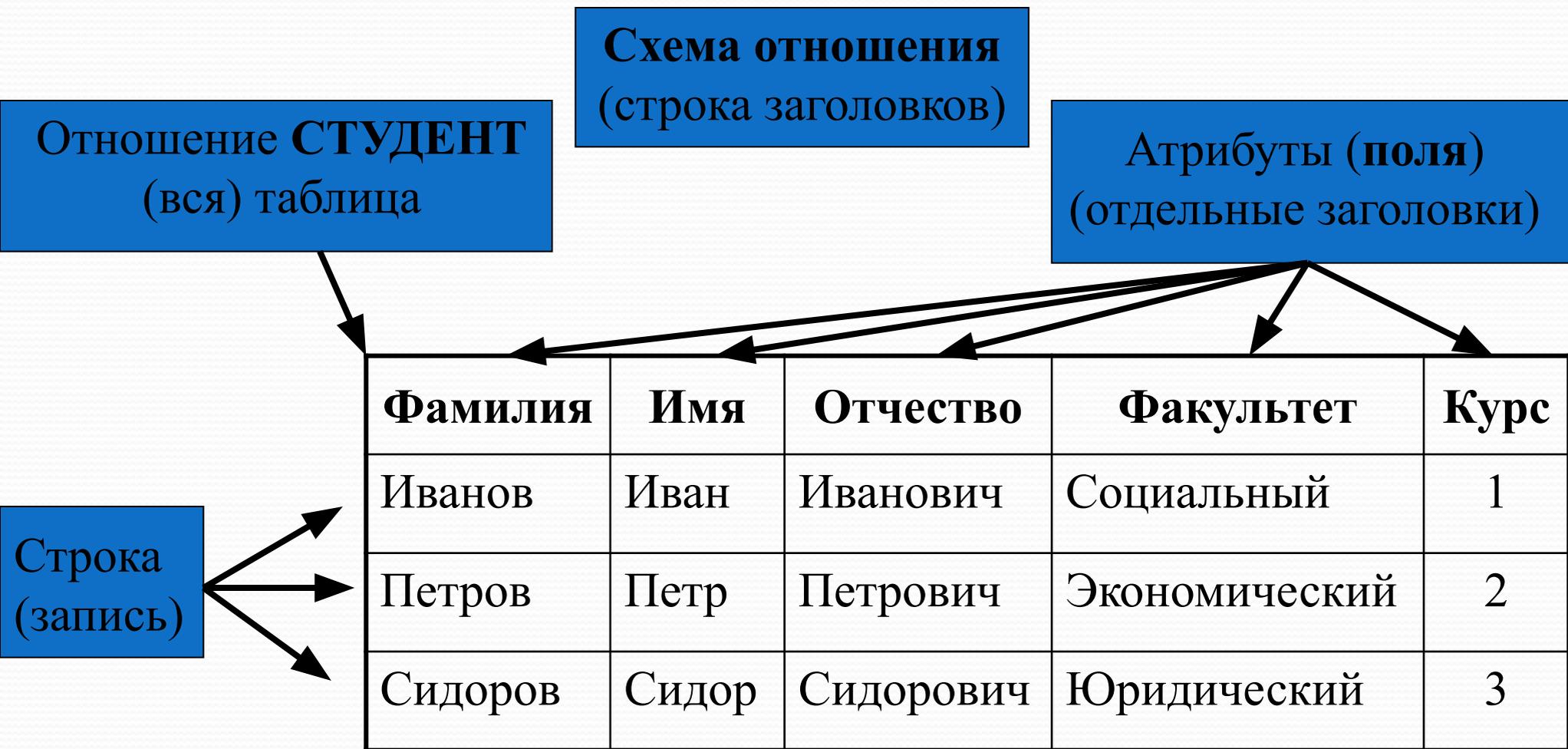
Реляционная модель данных характеризуется простотой структуры данных, удобным для пользователя табличным представлением, возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных. Она ориентирована на организацию данных в виде двумерных таблиц, где каждая таблица представляет собой двумерный массив и обладает следующими свойствами:

- каждый элемент таблицы - один элемент данных;
- все столбцы в таблице однородные (*имеют одинаковый тип и длину*);
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным;
- количество записей в таблице не ограничено.



Важнейшим понятием реляционных моделей данных является сущность. Сущность - это объект любой природы, данные о котором хранятся в БД. Данные о сущности (*ее взаимосвязях и отношениях*), как было отмечено, хранятся в двумерных таблицах, которые называют реляционными (*от английского relation - отношение*).

В качестве примера реляционной таблицы можно представить информацию о студентах, обучающихся в вузе.



Сама реляционная таблица – это **отношение**. Строка заголовков - **схема отношения**. В отношении каждый конкретный экземпляр сущности представлен **строкой**, состоящей из совокупности логически связанных полей.

5.3. Структурные элементы БД

Понятие базы данных связано со следующими представлениями о структурных элементах: **поле, запись, таблица, ключ.**

Поле – наименьший элемент информации (*элементарная логически неделимая единица информации*). Для описания поля используются следующие характеристики: **ИМЯ** (*в нашем примере: Фамилия, Имя, Отчество, Факультет, Курс, Группа*); **ТИП** (*зависит от типа хранящейся в нем информации*); **ДЛИНА** (*количество символов*); **ТОЧНОСТЬ** (*количество десятичных знаков для отображения дробной части числа*) для числовых данных.

Запись - набор полей, описывающих структуру хранящейся в таблице информации, которые в одной записи должны быть логически связаны друг с другом (*в таблице каждая строка – это запись*).

Таблица - набор записей одного и того же типа, имеющая имя, уникальное для данной базы.

При работе с реляционными моделями используется как математическая терминология, так и терминология, исторически принятая в сфере обработки данных. Например, отношение (*являясь математическим отражением или описанием некоторой сущности реального мира*) с точки зрения обработки данных представляет собой таблицу. Поскольку в локальных базах данных каждая таблица размещается в отдельном файле, то отношение можно отождествлять с файлом. Чтобы не возникало разночтений, приведем основные формальные реляционные термины и соответствующие им неформальные эквиваленты:

отношение, таблица, файл (*для локальных баз данных*);

кортеж, строка, запись;

атрибут, столбец, поле.

Ключ – одно или несколько полей, однозначно определяющих конкретную запись в таблице (*ее уникальность*). Бывают **первичные ключи** (*однозначно идентифицирующие экземпляр записи*) и **вторичные** (*выполняющие роль поисковых признаков*). В отношении СТУДЕНТ первичным ключом может быть поле ФАМИЛИЯ, если во всем списке нет однофамильцев (*это будет простой ключ*). Если есть однофамильцы, то для однозначной идентификации, необходимо выбрать совокупность полей – ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО (*которые создадут составной первичный ключ*). На практике обычно в качестве ключевого выбирают поле, в котором совпадения заведомо исключены. Таким полем в отношении СТУДЕНТ может служить номер зачетной книжки.



Первичный ключ должен обладать следующими свойствами:

- уникальностью - в таблице может быть только один первичный ключ;

- не избыточностью - не должно быть полей, которые, будучи удаленными из первичного ключа, не нарушат его уникальность;

- в состав первичного ключа не должны входить поля типа комментариев и графическое.

Для связи реляционных таблиц необходимо ввести в таблицы одинаковые по типу поля, по которым определится связь между записями таблиц. Связи бывают следующих типов «один к одному», «один ко многим», «многие ко многим».

Связь **один к одному** (1 : 1) предполагает, что в каждый момент времени одному экземпляру информационного объекта **A** соответствует не более одного экземпляра информационного объекта **B** и наоборот.

При связи **один ко многим** (1 : M) одному экземпляру информационного объекта **A** соответствует 0, 1 или более экземпляров объекта **B**, но каждый экземпляр объекта **B** связан не более чем с 1 экземпляром объекта **A**.

Связь **многие ко многим** (M : M) предполагает, что в каждый момент времени одному экземпляру информационного объекта **A** соответствует 0, 1 или более экземпляров объекта **B** и наоборот.

5.4. Этапы проектирования базы данных

Проектирование БД начинается с анализа предметной области (*анализа ее содержания со стороны смысла*). Определяются структура и состав данных предметной области, информационно-логическая (*инфологическая*) модель (*инструмент превращения предметной области в комплекс объектов информации и их связей*), которая, вместе алгоритмами обработки данных позволяет построить каноническую модель (*схему*) информационной базы (*каноническая модель отображает в структурированном виде информационную модель предметной области, позволяет сформировать логические записи, их элементы и взаимосвязи между ними*), а затем перейти к логической схеме и далее - к физическому уровню реализации.

Итак, начало проектирования БД – это проектирование информационно-логической модели предметной области, куда входит:

- определение предметной области;
- выделение информационных объектов и и определение ключевых реквизитов ;
- выявление связей между информационными объектами.

При проектировании базы данных необходимо решить вопрос о наиболее эффективной структуре данных. Процесс уменьшения избыточности информации в базе данных называется **нормализацией**. Нормализация - это разбиение таблицы на две или более, обладающих лучшими свойствами при вводе, изменении и удалении данных. Окончательная цель нормализации сводится к получению такого проекта базы данных, в котором исключена избыточность информации.

5.5. Проектирование с помощью СУБД MS Access

Популярность СУБД MS Access (*Access - в переводе с английского означает «доступ»*) обусловлена следующими причинами:

- Access является одной из самых понятных систем как для профессионалов, так и для начинающих пользователей;
- система имеет полностью русифицированную версию;
- интегрирована с пакетами Microsoft Office;
- позволяет использования OLE технологии;
- широко и наглядно представлена справочная система;
- существует набор «мастеров» по разработке объектов, облегчающий создание таблиц, форм и отчетов.

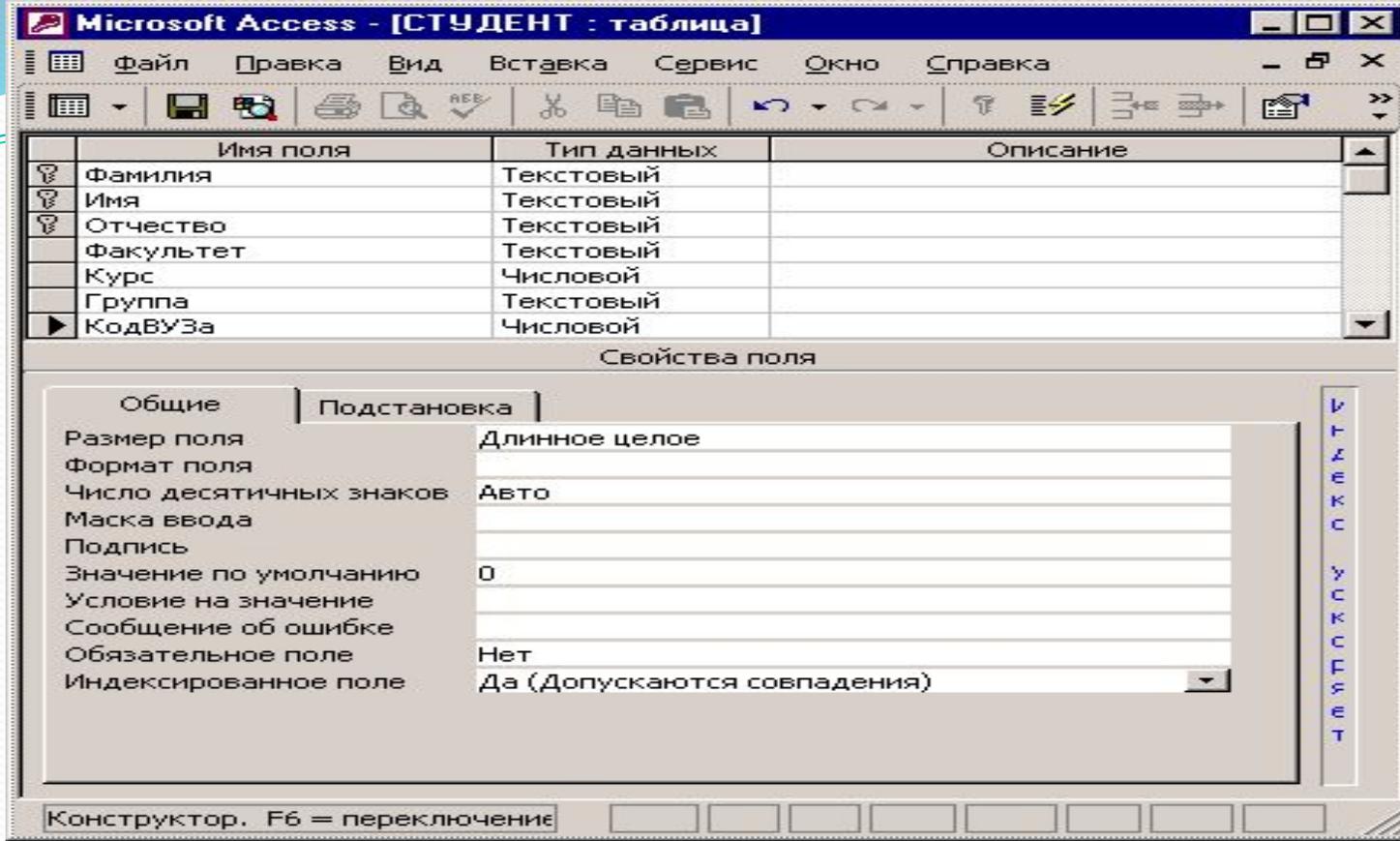
Всю базу данных по умолчанию Access хранит на диске в виде одного файла с расширением *.mdb. Данные хранятся в виде таблиц, строки которых состоят из наборов полей определенных типов. С каждой таблицей могут быть связаны индексы (*ключи*), задающие нужные пользователю порядки на множестве строк. Таблицы могут иметь однотипные поля (*столбцы*), что позволяет устанавливать между ними связи, выполнять операции реляционной алгебры.

Типичными операциями над базами данных являются:

- работа с таблицами (*создание, модификация, удаление таблиц, создание схем взаимосвязи таблиц*);
- ввод данных в таблицы (*непосредственно или с помощью формы*), проверку вводимых данных;
- поиск данных в таблицах по определенным критериям (*выполнение запросов*);
- создание отчетов о содержимом базы данных.

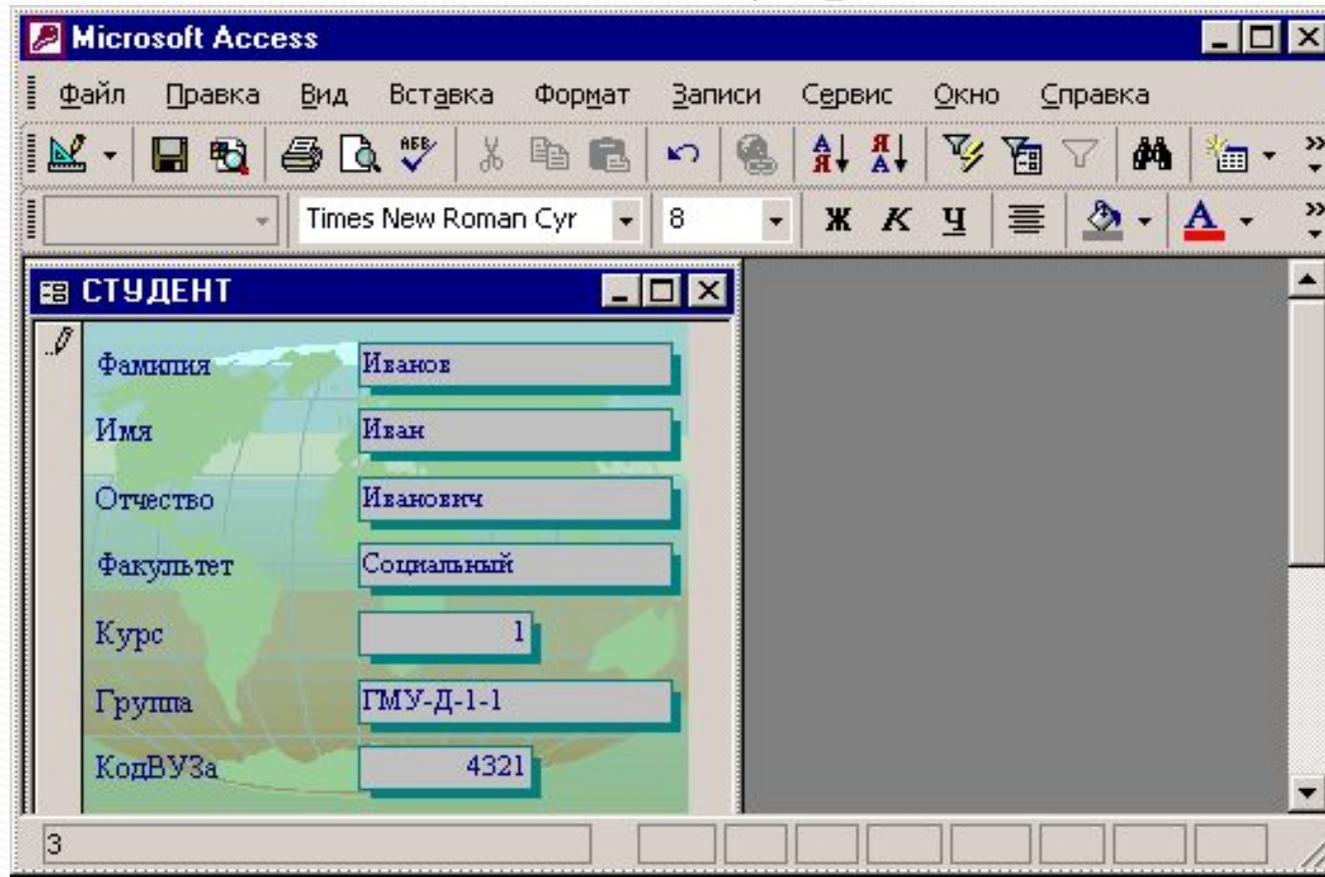


Таблица - это **объект**, который определяется и используется для хранения данных. Работа в Access начинается с определения реляционных таблиц и их полей, которые будут содержать данные, т.е. с создания макета таблицы. Заметим, что все основные объекты Access (*таблицы, запросы, отчеты, формы*) могут создаваться в режиме конструктора и в режиме мастера. Создание таблиц предпочтительней в режиме конструктора. Например:



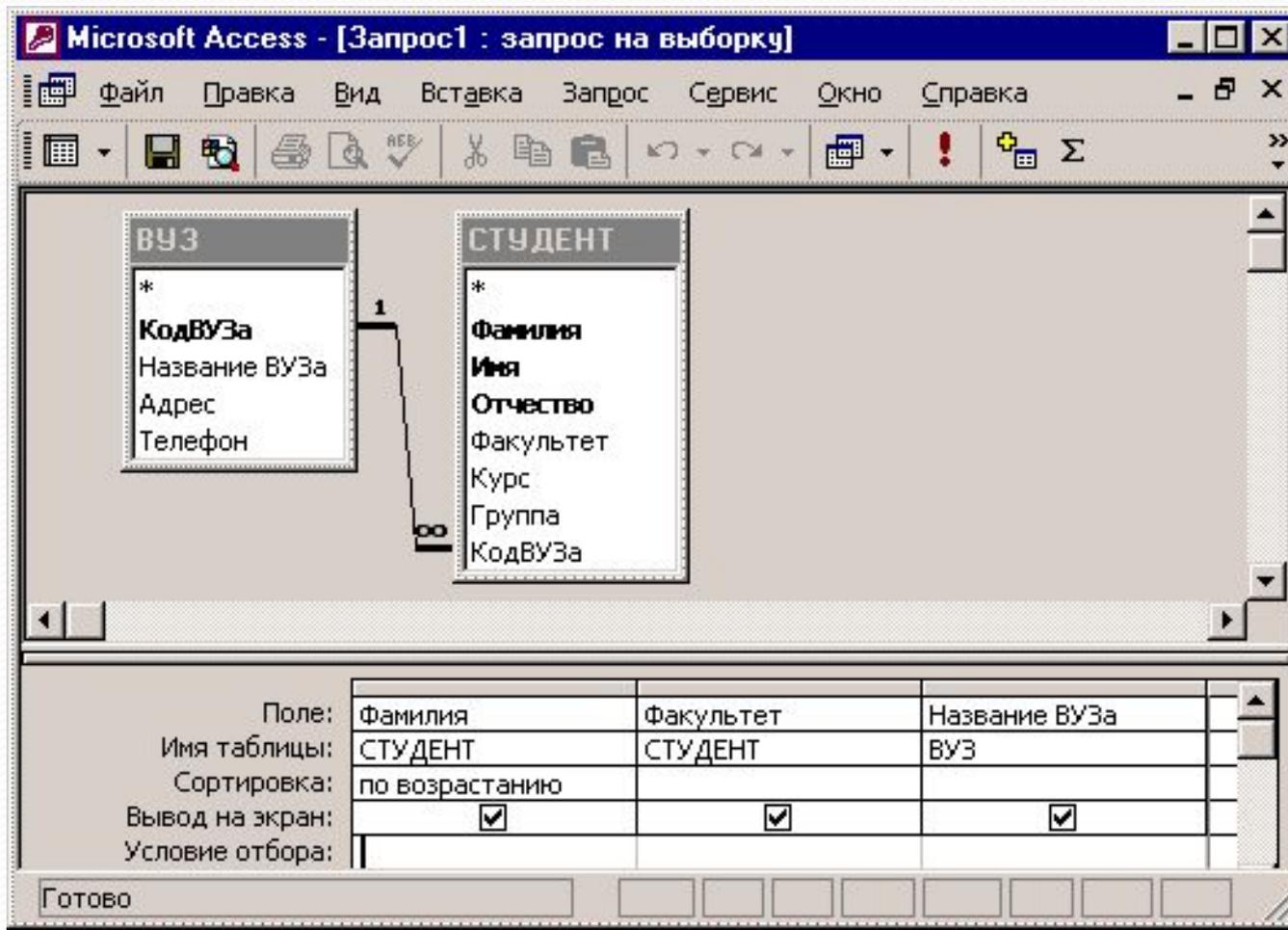
В макете задается структура таблицы, т.е. определяются типы, свойства полей, их число и названия - заголовки столбцов. При записи задается имя таблицы. После создания макетов таблиц определяются реляционные связи между ними. Режим таблицы используется для просмотра, добавления, изменения, простейшей сортировки или удаления данных.

Форма - это объект, предназначенный (в основном) для удобного ввода и отображения данных - одно из основных средств работы с базами данных в Access, используется для ввода новых записей, просмотра и редактирования уже имеющихся данных, задания параметров запросов и вывода ответов на них и др. Формы представляют собой окна с размещенными в них элементами управления.



Естественным режимом их создания является режим визуального конструирования (*при этом существуют разные возможности*). Создание формы можно автоматизировать, используя «Мастер форм». На форме можно расположить элементы управления, которые могут использоваться для ввода и отображения дат, а также для выполнения вычислений и вывода результата. Кроме того, в форму могут быть внедрены рисунки, диаграммы, аудио (звук) и видео (изображение).

Запрос - это объект, который позволяет пользователю получить нужные данные из одной или нескольких таблиц (можно создать запросы на выбор, обновление, удаление или на добавление данных). Создать запрос можно в режиме мастера или в режиме конструктора. Создадим запрос в режиме конструктора



Откроется окно, называемое бланком запроса по образцу, где в верхней части указанные таблицы предстанут вместе со схемой данных. Двойным щелчком укажем на те поля, которые будут участвовать в запросе. Пусть это будут поля из таблицы СТУДЕНТ (ФАМИЛИЯ, ФАКУЛЬТЕТ) и из таблицы ВУЗ (НАЗВАНИЕ вуза). Указанные поля появятся в нижней части бланка запроса по образцу. Таким образом создается простейший запрос, который называется запрос на выборку на языке запросов по образцу. При его активации будет построена таблица с указанием фамилии, факультета и вуза, в котором обучается студент. Закрытие запроса сопровождается присвоением ему имени.

Отчет - это **объект**, предназначенный для создания документа, который впоследствии может быть распечатан или включен в документ другого приложения. Для создания отчетов рационально воспользоваться Мастером отчетов, который похож на Мастера форм. Выполняется отчет за ряд шагов, в процессе которых:

- выясняются таблицы и запросы, на основе которых строится отчет;
- поля, по которым производится сортировка, фильтрация;
- стиль оформления печатного документа.

Редактировать отчет, созданный Мастером, можно с помощью конструктора.

В системное понятие «База данных» входят и другие объекты: **Макрос** - как способ структурированного описания одного или нескольких действий, которые автоматически выполняются в ответ на определенное действие. **Модуль** - программа на языке Visual Basic, с помощью которых определенный процесс разбивается на несколько небольших процедур. **Страница доступа к данным** - файл HTML и вспомогательные файлы для доступа к данным из MS Internet Explorer к данным базы данных.

Опыт использования баз данных позволяет выделить общий набор их рабочих характеристик:

• полнота - чем полнее база данных, тем вероятнее, что она содержит нужную информацию (*однако, не должно быть избыточной информации*);

• правильная организация (*чем лучше структурирована база данных, тем легче в ней найти необходимые сведения*);

• актуальность (*любая база данных может быть точной и полной, если она постоянно обновляется и в каждый момент времени полностью соответствует состоянию отображаемого ею объекта*);

• удобство для использования (*база данных должна иметь развитые методы доступа к любой части информации*).

5.6. Обобщенная технология работы в СУБД Access

Реляционная модель БД имеет дело с тремя аспектами данных: со структурой данных, с целостностью данных и с манипулированием данными. Под структурой понимается логическая организация данных в БД, под целостностью данных - безошибочность и точность информации, хранящейся в БД, под манипулированием данными - действия, совершаемые над данными в БД. Эти три аспекта отражают и основные процедуры процесса накопления данных (*хранение, актуализацию и извлечение*).

Структура интерфейса. При работе в СУБД Access на экран выводятся рабочее поле и панель управления, последняя включает меню, вспомогательную область управления и строку подсказки. **Строка меню** содержит основные режимы программы.

Вспомогательная область управления включает:

- строку состояния;
- панели инструментов;
- вертикальную и горизонтальную линейки прокрутки.

Команды для выполнения типовых операций

Совокупность команд в СУБД Access, предоставляемых в распоряжение пользователю, может быть разбита на следующие типовые группы:

- команды для работы с файлами;
- команды редактирования;
- команды форматирования;
- команды для работы с окнами;
- команды для работы в основных режимах СУБД Access (*таблица, форма, запрос, отчет*);
- получение справочной информации.

Основные этапы технологии работы в СУБД Access

В качестве основных этапов обобщенной технологии работы в СУБД Access можно выделить следующие:

- создание структуры таблиц базы данных;
- ввод и редактирование данных в таблицах;
- обработка данных, содержащихся в таблицах;
- вывод информации из базы данных.

Схема обобщенной технологии работы в СУБД

Создание структуры таблиц БД

Ввод и редактирование данных в таблицах БД

с использованием форм

без использования форм

Обработка информации Базы данных

на основе запросов

на основе программы

Вывод информации из БД

с использованием отчетов

без использования отчетов

Создание структуры таблиц базы данных

При формировании новой таблицы базы данных работа с СУБД начинается с создания структуры таблицы. Этот процесс включает определение перечня полей, из которых состоит каждая запись таблицы, а также типов и размеров полей. Практически все используемые СУБД хранят данные следующих типов: текстовый (символьный), числовой, календарный, логический, примечание.

Ввод и редактирование данных

Заполнение таблиц данными возможно как непосредственным вводом данных, так и в результате выполнения программ и запросов. Практически все СУБД позволяют вводить и корректировать данные в таблицах двумя способами:

- ▶ с помощью предоставляемой по умолчанию стандартной формы в виде таблицы;
- ▶ с помощью экранных форм, специально созданных для этого пользователем.

Обработка данных, содержащихся в таблицах

Обрабатывать информацию, содержащуюся в таблицах базы данных, можно путем использования запросов или в процессе выполнения специально разработанной программы. Большинство СУБД разрешают использовать запросы следующих типов:

- ▶ запрос-выборка, предназначенный для отбора данных, хранящихся в таблицах, и не изменяющий эти данные;
- ▶ запрос-изменение, предназначенный для изменения или перемещения данных;
- ▶ запрос с параметром, позволяющий определить одно или несколько условий отбора во время выполнения запроса.

Вывод информации из базы данных

Практически любая СУБД позволяет вывести на экран и принтер информацию, содержащуюся в базе данных из режимов таблицы или формы. Используя специальные средства создания отчетов, пользователь получает следующие дополнительные возможности вывода данных:

- ▶ включать в отчет выборочную информацию из таблиц базы данных;
- ▶ добавлять информацию, не содержащуюся в базе данных;
- ▶ выводить итоговые данные на основе информации базы данных;
- ▶ располагать выводимую в отчете информацию в любом удобном виде;
- ▶ включать в отчет информацию из разных связанных таблиц базы данных.

Тенденция развития систем управления базами данных. Современные системы управления базами данных можно разделить на универсальные и специализированные. Универсальные системы предназначены для использования в различных предметных областях. Специализированные - для описания конкретных систем. В последние годы все большее признание и развитие получают объектно-ориентированные СУБД, толчок к появлению которых дали объектно-ориентированное программирование и использование компьютера для обработки и представления практически всех форм информации, воспринимаемых человеком.

В недавнем прошлом понятие «автоматизация обработки информации» означало так называемые традиционные процедуры: сбор, регистрация, обработка, передача, хранение и выдача информации. Развитие производительных возможностей компьютеров, а также программного обеспечения позволили начать переход к обработке нетрадиционных данных (*графика, рисунки, тексты, звуковая информация, видео и т.д.*) и другой информации. Возрастает роль интеграции информации, необходимость обмена информацией между базами данных различных СУБД, работы в локальных сетях в условиях использования информации базы данных многими пользователями.



По мнению специалистов, основным направлением в развитии СУБД является разработка универсальных СУБД. При этом предусматривается:

- ▶ улучшение графического интерфейса;
- ▶ возможности работы в локальном режиме и в режиме клиента на рабочей станции;
- ▶ расширение специальных средств;
- ▶ наличие собственных языков программирования;
- ▶ применение объектной технологии типа OLE2.

Интеграция баз данных требует разработки проблем параллельной обработки информации на многопроцессорных вычислительных системах. В этом направлении предусмотрено развитие СУБД:

- без раздела ресурсов (*за каждым процессором закреплены выделенные области памяти и диски*);
- с совместным использованием дискового пространства (*все процессоры делят между собой операционную память и место на диске*).

В целом СУБД совершенствуются и развиваются с учетом расширения функциональных возможностей, а также архитектуры СУБД и качества внешнего интерфейса; расширяются возможности обработки сложных запросов с одновременным увеличением скорости обработки.

Развитие баз данных, СУБД и Internet происходит в направлении расширения их «интеллекта» за счет развития и применения интеллектуальных программных и языковых средств, что приведет к созданию баз знаний по соответствующим предметным областям и глобальной базы знаний.