

Структура данных и система запросов на примерах баз данных различного назначения: юридические, библиотечные, налоговые, социальные, кадровые и др.

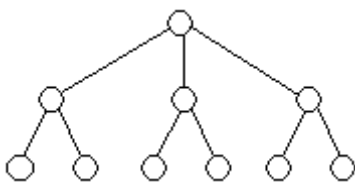
База данных (БД) предназначена для хранения больших объемов данных во внешней памяти вычислительной системы. Эта совокупность специальным образом организованных данных, отражающая состояние объектов в некоторой предметной области. Предметной областью может быть книжный фонд библиотеки, музейный фонд, кадровый состав предприятия, учебный процесс в школе. База данных составляет часть соответствующей информационной системы.

Понятия “данные” и “модель данных” являются центральными для баз данных. Категория “данные” означает параметры, конкретные значения, которые определяют объект. Например, 1 байт, Иванов А.П. Данные не имеют определенной структуры. Логическую структуру хранимой в БД информации называют моделью представления информации. Это некоторая абстракция, которая отражает взаимосвязь данных.

Рассмотрим основные модели данных.

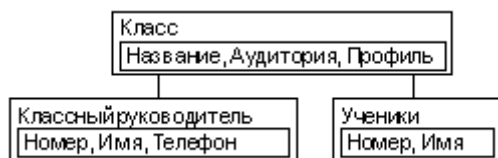
Иерархическая модель

В реальном мире многие связи соответствуют некоторой иерархии, когда один объект выступает как родительский, а с ним связано множество подчиненных объектов. Связи между объектами описывают с помощью упорядоченного графа, дерева, которое упрощенно можно представить в следующем виде:

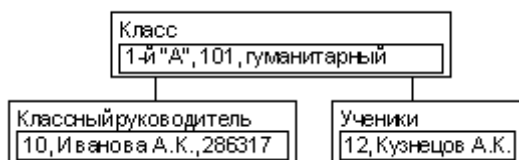


Основными информационными единицами иерархической модели являются сегмент и поле. Поле — это минимальная неделимая единица данных, сегмент называется записью. В модели сегменты объединяются в ориентированный древовидный граф. Ребра графа отражают иерархические связи между сегментами. “Дерево” состоит из одного корневого сегмента и упорядоченного набора подчиненных сегментов.

Возможен следующий пример организации данных типа “дерево”:

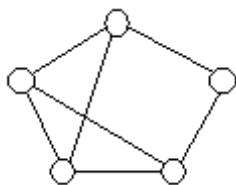


Конкретный экземпляр записей в базе данных с приведенной схемой “дерево” может, например, выглядеть следующим образом:



Сетевая модель

С помощью сетевой модели можно отобразить взаимосвязи объектов в виде произвольного графа:



Для описания сетевой базы данных используют понятия “запись” и “связь”. Связь определяется для двух записей: предка и потомка. В иерархической модели запись-потомок имела только одну запись-предок, в сетевой модели данных такого ограничения нет, запись-потомок может иметь произвольное число записей-предков.

Пример схемы сетевой базы данных приведен на следующем рисунке, связи здесь обозначены надписями на соединяющих линиях.



Сетевая БД состоит из набора записей и соответствующих связей.

Реляционная модель

Эта модель данных основана на понятии отношения. Наглядной формой отношения является двумерная таблица. С помощью таблицы можно описать простейший вид связей между данными: деление одного объекта на множество подобъектов, которым соответствуют строки таблицы.

С. 100
Модель 03/19
Эксп. 1

Строки таблицы имеют одинаковую структуру и называются записями. Записи содержат информацию об отдельном объекте — о классе в нашем примере. Записи состоят из полей. Поле — это определенное свойство, атрибут объекта: название класса, его профиль. Поле имеет определенный тип, который определяет множество значений и множество операций, выполняемых с данными. Основными типами данных, применяемыми в БД, являются числовой, символьный, логический и дата.

Для каждой таблицы реляционной БД должен быть указан главный ключ. Им может быть имя одного и нескольких полей, однозначно определяющих запись. Значение этого поля не может повторяться в различных записях, оно уникально. Именно по этому полю можно отделить одну запись от другой. Название класса может послужить главным ключом таблицы **Класс**.

В рамках одной таблицы нельзя описать сложные логические структуры данных из предметной области, поэтому применяют связывание таблиц.

На основе модели данных строят базу данных. Далее рассмотрим реляционную БД как наиболее распространенную в настоящее время.

Системы управления базами данных

Системой управления базами данных (СУБД) называют комплекс языковых и программных средств создания, ведения и использования БД. Для персональных компьютеров используют реляционные БД, основанные на табличном представлении данных. СУБД позволяет пользователю выполнять следующие действия:

- формировать структуру БД, редактировать ее;
- вводить конкретную информацию, изменять ее;
- манипулировать данными, находить информацию, отвечающую некоторым критериям;

- сортировать данные.

Приведем примеры БД.

Правовые системы и базы данных предназначены для хранения и регулярного обновления сборников нормативных документов и подзаконных актов. Экономические реформы, проводимые в нашей стране, сопровождаются значительными изменениями законов и нормативных актов, инструкций. С этими документами работают не только юристы, но и руководители и специалисты должны постоянно иметь возможность просмотреть быстро меняющиеся требования, нормативы. В этом классе представлены отечественные информационно-правовые и справочные системы “Кодекс”, “Консультант Плюс”, “Гарант-Сервис”, “Кодекс”, “Консультант-Бухгалтер”.

Система “Консультант Плюс” позволяет вести поиск документов по официальным реквизитам или отдельным словам, по тематическому рубрикатору, по ключевым словам.

Компания “Гарант” имеет колоссальный информационный банк, охватывающий весь спектр российского законодательства и основные нормы международного права. Все документы представлены в действующей редакции, новая информация поступает ежедневно, еженедельно обновляется весь банк данных. Работа с БД производится в гипертекстовой среде с перекрестными ссылками и мощными поисковыми инструментами.

Рассмотрим СУБД Microsoft Access как наиболее популярное приложение, используемое для обучения. Приложение Microsoft Access работает на отдельном компьютере или в небольшой локальной сети. С помощью СУБД, ориентированной на персональный компьютер, можно создать небольшую личную БД или БД для небольшой организации. Система MS Access позволяет, не прибегая к программированию, выполнять основные действия с данными, расположенными в таблицах. Программа имеет стандартный для продуктов Microsoft интерфейс.

База данных приложения MS Access включает в свой состав таблицы, формы, запросы, отчеты, макросы и модули.

Таблица — основная единица хранения данных в базе. Это понятие соответствует реляционной модели данных. Таблица состоит из записей и полей. Данные в таблице можно просматривать и модифицировать, сортировать и фильтровать, распечатывать. В базе создается совокупность связанных между собой таблиц. MS Access позволяет создавать связи типа “один к одному” и “один ко многим” с помощью схемы данных. Таблица — главный объект БД, остальные объекты являются производными от нее.

Форма создается для удобного ввода, отображения и изменения данных в полях таблицы.

Запрос позволяет выбрать данные из БД, удовлетворяющие некоторому условию, обновить, добавить, удалить данные. Запрос может выбрать данные из различных таблиц. Результат будет получен так же в табличном виде.

Отчет предназначен для вывода на печать информации, сформированной из таблиц и запросов.

Страницы публикуют БД в Web для использования ее в локальных сетях и Интернете. Пользователи работают на страницах, так же как в приложении: просматривают таблицы, выполняют запросы.

Макрос задает автоматическое выполнение некоторых операций с помощью макрокоманд, например, “Печатать”, “НайтиЗапись”.

Модуль состоит из процедур на языке VBA и автоматизирует некоторое действие, которое стандартными средствами приложения MS Access выполнить затруднительно. VBA (*Visual Basic for Applications*) — язык программирования высокого уровня для создания приложений Windows. Модули используют для решения более сложных задач, которые не под силу макросам.

Построим пример базы данных учащихся в школе средствами СУБД MS Access.

База данных учащихся должна являться частью общей базы данных информационной системы образовательного учреждения. Такая база данных содержит сведения об учащихся и их родителях, о сотрудниках, учебный план, электронные классные журналы, расписание уроков и факультативных занятий, распоряжения и отчеты, методические и образовательные ресурсы. Общая база данных является одним из компонентов информационной системы, который обеспечивает управление образовательным процессом, обучающимися, кадрами, ресурсами и предоставляет возможность интерактивной связи участникам этого процесса.

Рассмотрим упрощенный вариант школьной базы данных, причем обратим внимание только на ту ее часть, которая относится к учащимся.

Разработку БД начнем с проектирования. Так как MS Access использует реляционную модель данных, информацию расположим в трех следующих таблицах — *Классы*, *Классный руководитель*, *Ученики*:

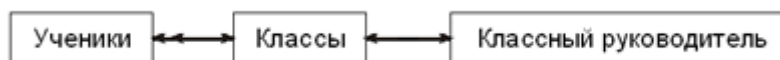
Рисунок 1

Рисунок 2 (продолжение)

Рисунок 3

Так как мы рассматриваем небольшой пример, то естественно, что в таблицах отражена только небольшая часть реальной информации. Например, в таблице **Ученики** следовало бы внести не только фамилии учеников, но и имена, и другие личные данные.

Эти три таблицы должны образовать единую систему, поэтому они взаимосвязаны:



Здесь присутствуют два типа связи: “один к одному” и “один ко многим”.

Таблицы **Классы** и **Классный руководитель** объединяет связь типа “один к одному”, она на рисунке обозначена одинарной стрелкой. При этом типе связи две таблицы можно было бы объединить в одну таблицу, так как одной записи в одной таблице соответствует одна запись в другой таблице. Информация разделена на две таблицы, так как одна таблица содержала бы слишком много полей и с ней было бы неудобно работать. Эти таблицы находятся на одном уровне иерархии.

Таблицы **Классы** и **Ученики** объединяет связь типа “один ко многим”, она обозначена двойной стрелкой. Например, в одном классе обучается несколько учеников, но каждый ученик может обучаться только в одном классе. Поэтому одной записи в таблице **Классы** соответствует несколько записей в таблице **Ученики**. Это связь между соседними уровнями иерархической структуры.

Приступим к созданию БД в приложении MS Access. После запуска программы и создания файла новой БД следует построить структуру таблиц и заполнить таблицы данными. Структуру таблиц можно задать с помощью Конструктора. На рисунке показано окно Конструктора при определении структуры таблиц **Классы**:

Классы : таблица		
	Имя поля	Тип данных
	Номер класса	Числовой
	Название класса	Текстовый
	Профиль	Текстовый
	Количество учеников	Числовой

Изображение ключа указывает на ключевое поле. Аналогично определим структуры таблиц **Классный руководитель** и **Ученик**:

Классный руководитель : таблица		
	Имя поля	Тип данных
	Номер классн_рук	Числовой
	Фамилия	Текстовый
	Телефон	Числовой
	Название класса	Текстовый
	Наличие почетного_зван	Логический

Обратим внимание на то, что в таблице **Классный руководитель** поле **Наличие почетного_зван** имеет логический тип.

Ученик : таблица		
	Имя поля	Тип данных
	Номер Ученика	Числовой
	Фамилия	Текстовый
	Возраст	Числовой
	Дом_адрес	Текстовый
	Название класса	Текстовый

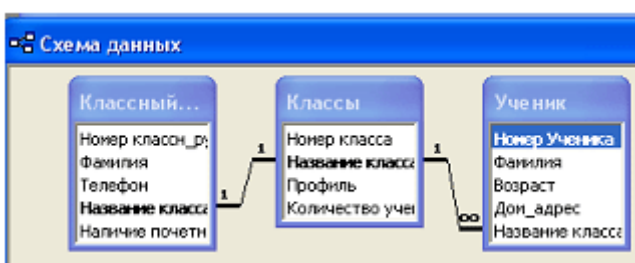
Заполним таблицы данными:

На следующем этапе определим связи между таблицами, используя **Схему данных**:

Классы : таблица				
	Номер класса	Название класса	Профиль	Количество учеников
+	1	1а	гуманитарный	25
+	2	1б	математический	24
+	3	1в	общеобразовательный	25

Классный руководитель : таблица					
	Номер классн	Фамилия	Телефон	Название клас	Наличие почет
+	1	Иванова	234566	1а	☒
+	2	Павловская	443218	1б	☐
+	3	Перова	134560	1в	☒

Ученик : таблица					
	Номер Ученика	Фамилия	Возраст	Дом_адрес	Название клас
	1	Климов	7	ул.Ленина 27,1	1а
	2	Арбузов	7	ул.Юрева 101,	1а
	3	Серов	6	пр.Декабристо	1б
	4	Николаева	7	пр.Комсомоль	1б
	5	Красина	7	ул.Лесная 2	1в
	6	Петров	6	Ул.Лесная 17	1в



Здесь отражены два типа связи. Связь “один к одному” между таблицами **Классы** и **Классный руководитель** производится через общий главный ключ “Название класса”. Связь “один ко многим” между таблицами **Классы** и **Ученик** создана с помощью главного ключа “Название класса” таблицы **Классы** и одноименного поля таблицы **Ученик**.

Программа MS Access, обеспечивая целостность данных, препятствует удалению и изменению связанных данных. Изменение ключевого поля в первичной таблице приведет к автоматическому изменению ключей в связанных записях. Удаление записей из первичной таблицы вызовет соответствующее удаление записей в связанной таблице. Для этого при создании связи следует установить флажки **Каскадное обновление связанных данных** и **Каскадное удаление связанных данных**.

Для поиска информации в БД создают запросы. Запрос позволяет манипулировать данными БД: выбирать данные по некоторому условию, сортировать, обновлять, удалять и добавлять данные в БД. Простейшим является запрос на выборку. Результатом запроса на выборку будет таблица, которая содержит поля, удовлетворяющие условиям отбора.

В нашем примере Запрос 1 позволяет получить список учеников с домашними адресами, которые обучаются в классе 1а. Создадим этот запрос

в конструкторе, в качестве исходной таблицы укажем таблицу **Ученик**, в строке **Условие отбора** в поле **Название класса** зададим условие “1а”:

Запрос1 : запрос на выборку

Ученик

Номер Ученик
Фамилия
Возраст
Дом_адрес

Поле:	Фамилия	Дом_адрес	Название класса
Имя таблицы:	Ученик	Ученик	Ученик
Сортировка:			
Вывод на экран:	☒	☒	☐
Условие отбора:			"1а"
или:			

Далее получим результат выборки в таблице:

Запрос1 : запрос на выборку

	Фамилия	Дом_адрес
	Климов	ул.Ленина 27,1
	Арбузов	ул.Юрева 101,

Второй запрос выводит фамилии классных руководителей, которые не имеют почетного звания. Он основан на таблице **Классный руководитель**. Поле **Наличие почетного зван** имеет логический тип, поэтому в строку **Условие отбора** внесена константа **Нет**:

Запрос2 : запрос на выборку

Классный...

Номер класс
Фамилия
Телефон
Название кл

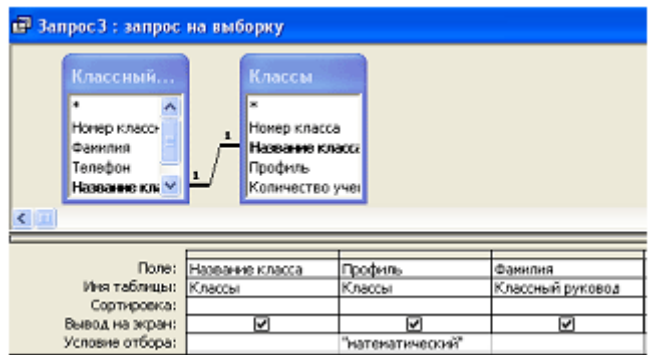
Поле:	Фамилия	Наличие почетног
Имя таблицы:	Классный руковод	Классный руковод
Сортировка:		
Вывод на экран:	☒	☒
Условие отбора:		Нет
или:		

Результат выборки:

Запрос2 : запрос на выборку

	Фамилия	Наличие почет
	Павловская	☐

Запрос может быть многотабличным. Запрос 3 выводит поля **Название класса**, **Профиль** из таблицы *Классы*, и поле **Фамилия** из таблицы *Классный руководитель*, при этом поставлено условие, что класс имеет математический профиль.



Ответ можно просмотреть в режиме таблицы:

Запрос 3 : запрос на выборку			
	Название класса	Профиль	Фамилия
	1б	математический	Павловская

В заключение нашего примера подготовим отчет. Отчеты предназначены только для вывода информации на печать. Напечатаем фамилии учеников и их домашние адреса, основываясь на таблице *Ученик*. Выберем создание отчета с помощью мастера и, следуя его указаниям, получим следующий отчет:

Ученик-отчет	
Фамилия	Дом_адрес
Арбузов	ул.Юркова 101,17
Кашин	ул.Ленина 27,14
Красина	ул.Лесная 2
Николаева	пр.Коммунальщиков 23,45
Петров	Ул.Ленина 17
Серов	пр.Дзержинское 12,86