

**ЗАПИСАТЬ В  
ТЕТРАДЬ!**

**ТЕМА УРОКА.**

**Системы управления базами  
данных. Реляционная модель  
данных (§5)**

**Добрый день, дорогие дети! Я приветствую вас на  
дистанционном уроке.**

**Желаю вам успешного освоения учебного  
материала. Я постаралась сделать материал урока  
понятным и доступным для вас.**

Человек в процессе информационной деятельности собирает и накапливает сведения об окружающем мире. До появления ВТ вся информация хранилась обычно в письменном или печатном виде. Однако чем больше были объемы информации, с которыми приходилось оперировать человеку, тем острее вставал вопрос сохранения информации и ее обработки.

**Вспомним!** Как называется информация, предназначенная для обработки .....

**ЗАПИСАТЬ В**

Мы будем рассматривать не просто данные, а базы данных. Не всякий блок информации можно считать БД.

**ТЕТРАДЬ!**

*База данных – это совокупность данных, которые обладают свойствами структурированности и взаимосвязанности, а также независимости от прикладных программ.*

Поясню, что означают перечисленные свойства БД.

Чтобы пользователь легко мог **находить нужную** ему информацию, последняя **должна быть организована** определенным образом. Это касается не только информации, которая хранится в компьютере, но любой информации об объектах реального мира. Например, удобно находить нужную книгу **в библиотеке, пользуясь каталогом**. Не составляет труда отыскать интересующие вас объявления в газете.

Такая легкость поиска возможна благодаря тому, что данные в каталоге или в газете **имеют структуру**, или, другими словами, **структурированы**. Все книги описаны одинаковым образом: автор, название, издательство, год издания и т.д. Все объявления о продажах размещены по рубрикам и также имеют определенную структуру: краткое описание товара, цена, телефон.

**Устройство БД обычно сложнее**, чем устройство простого каталога или набора газетных объявлений.

Это обусловлено прежде всего свойством **взаимосвязанности** данных в базе, которое поясню на таком примере. Допустим, вы хотели, помимо каталожных карточек, описывающих каждую книгу, иметь карточки с информацией о каждом авторе (год рождения, литературный жанр, хобби и т.д.). Если бы такие карточки были созданы, вы получили бы пример взаимосвязанных данных: сведения по отдельной книге связаны с информацией об авторе. Эта связь осуществляется через определенный параметр – фамилию автора.

Наконец, последнее из перечисленных свойств БД – это их **независимость от прикладных программ**. БД конструируются таким образом, чтобы с ними можно было работать в различных программных средах и на различных компьютерных платформах.

*Управляющая программа, предназначенная для хранения, поиска и обработки данных в базе, называется системой управления базами данных (сокращенно СУБД).*

Современные СУБД – это программные приложения, которые позволяют решать многообразные задачи. Все существующие системы удовлетворяют, как правило, следующим **требованиям**.

- **Возможности манипулирования данными** (ввод, выбор, вставка, обновление, удаление и др.). Основные операции с данными выполняются под управлением СУБД. Важными показателями при этом являются производительность СУБД, стоимость хранения и использования данных, простота обращения к базе данных и проч.
- **Возможность поиска и формирование запросов**. С помощью запросов пользователь может оперативно получать различного рода информацию, которая хранится в базе данных.
- **Обеспечение целостности** (согласованности) данных. При использовании данных многими пользователями важно обеспечить корректность операций, при которых не может быть нарушена согласованность данных. Нарушение согласованности данных чревато их необратимой потерей.
- **Обеспечение защиты и секретности**. Кроме защиты от некорректных действий пользователей, важно обеспечить защиту данных от несанкционированного доступа и от аппаратных сбоев. Проникновение в базу лиц, не имеющих на это право, может иметь следствием разрушение данных. Секретность БД позволяет определять круг лиц, имеющих доступ к информации, и порядок доступа.

В настоящее время существует множество СУБД, которые различаются архитектурой, внутренним языком программирования, ОС под управлением которой они работают, а также другими характеристиками. Среди СУБД, которые устанавливаются в небольших организациях и ориентированы на работу с конечными пользователями, наиболее популярны **Access**

## Модели БД

### Реляционная модель данных

Практически все современные СУБД используют реляционную модель данных. В основе этой модели, предложенной Е.Ф. Коддом 1970 г., лежит понятие отношения (по- английски relation). Речь идет об отношениях, заключенных в двумерных (то есть обычных) таблицах. Кодд, показал, что таблицы могут быть использованы для хранения данных об объектах реального мира.

Рассмотрим пример реляционной модели данных. Допустим, нужно составить БД о подготовке к походу, в который отправятся ученики школы. Назовем эту базу «Поход». В ней прежде всего должна быть таблица, отражающая сведения об участниках похода. Эта таблица будет состоять из полей «Код участника», «Фамилия», «Имя», «Класс», «Адрес», «Телефон» (ТАБЛИЦА 1).

*Двумерные таблицы состоят из строк, называемых в терминологии баз данных записями, и столбцов, которые именуются полями*

Код участника	Фамилия	Имя	Класс	Адрес	Телефон
1	Арбузов	Константин	8 А	ул. Песчаная, 3а, кв.17	25-44-93
2	Бирюкова	Ирина	8 Б	ул.Луговая, 15	33-62-76
3	Вендик	Константин	8 А	ул. Песчаная, 3, кв.43	25-37-12
4	Крамаренко	Вячеслав	8 В	пр. Победы, 1, кв.66	34-25-98
5	Крамаренко	Наталья	7 Б	пр. Победы, 1, кв.66	34-25-98

**ТАБЛИЦА 1. «Участники»**

Назовем эту таблицу «Участники» (такое же имя будет иметь и отношение, представленное данной таблицей). Обратите внимание, что поле «Код участника» данной таблицы имеет уникальные значения, которые не повторяются ни в одной из записей. Такое поле при включении таблицы в БД будет иметь статус *ключевого поля*. На роль ключевого не годятся ни поле «Фамилия» (в таблице имеются однофамильцы), ни любое другое поле.

БД строится на основе не одной, а множества таблиц. Введем в БД «Поход» еще одну таблицу «Снаряжение» (ТАБЛИЦА 2). В нее будут заноситься сведения о снаряжении, которое должны взять с собой участники.

Код снаряжения	Код участника	Снаряжение
10	1	Палатка
20	2	Котелок
30	2	Веревка
40	4	Топор
50	5	Котелок
60	5	Фонарь
70	4	Нож

**ТАБЛИЦА 2. «Снаряжение»**

Записи в этой таблице составлены из полей «Код снаряжения», «Код участника» и «Снаряжение». Первое поле («Код снаряжения») является ключевым полем: для каждой записи оно принимает уникальные значения, в отличие от остальных полей, в которых имеются дублирующие значения (например, повторяются коды участников 2, 4, 5 и снаряжение «Котелок»).

**? Каким образом можно использовать информацию приведенных двух таблиц?**

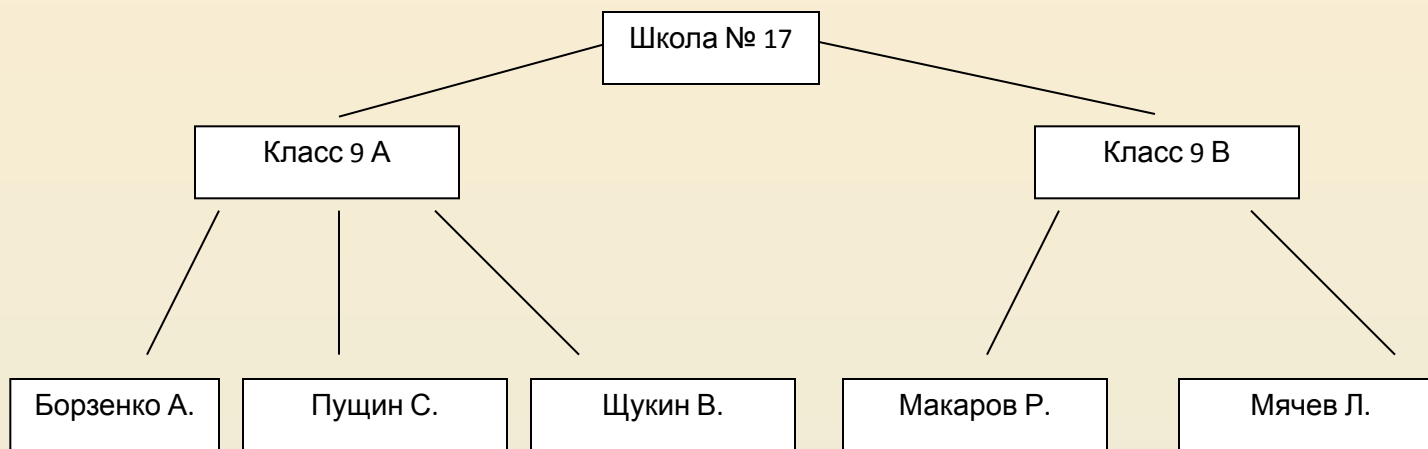
Допустим, нас интересует, кто возьмет в поход палатку. Из таблицы «Снаряжение» находим, что палатку, имеющую код 10, должен взять участник с кодом 1. Из таблицы «Участники» видим, что код 1 имеет Арбузов Константин. Таким образом, палатку должен взять участник Арбузов К. эту информацию мы получили благодаря связи таблиц «Участники» и «Снаряжение» через общее поле «Код участника».

В дополнение к имеющимся двум таблицам мы можем составить и другие полезные таблицы, например, «Продукты», в которой будет приведен список продуктов и указаны участники, которые их берут. Вы можете придумать таблицы «График движения», «Памятные места», «Дежурства» и, возможно, другие, которые войдут в реляционную БД «Поход». О том, как создавать таблицы с помощью СУБД и как с таблицами работать, мы будем изучать на следующих уроках.

## Иерархическая и сетевая модели данных

При построении баз данных, кроме реляционной модели, иногда прибегают к другим видам моделей: иерархической и сетевой. Мы кратко рассмотрим эти типы моделей, хотя они имеют, скорее, исторический интерес, поскольку в основу практически всех современных СУБД положена реляционная модель.

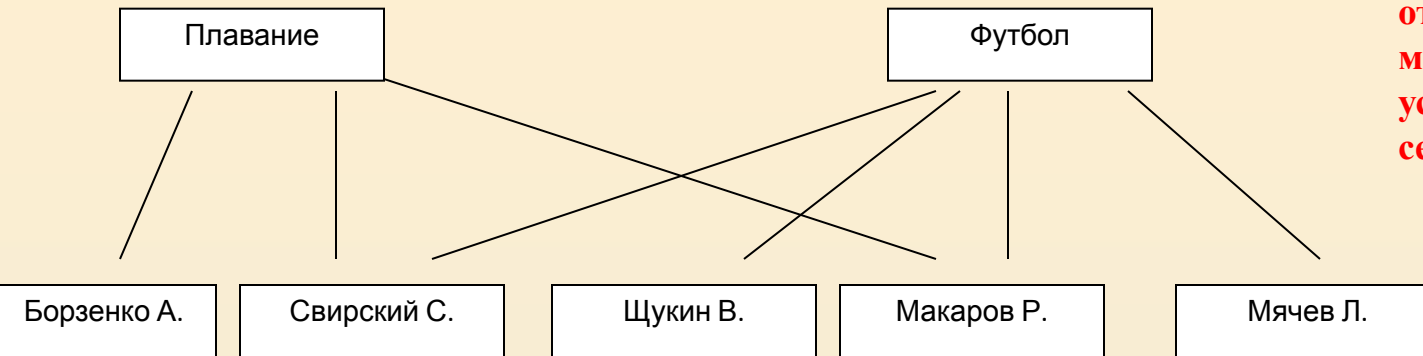
**Иерархическая модель БД представляет собой совокупность объектов различного уровня, причем объекты нижнего уровня подчинены объектам верхнего уровня**



### ***Пример иерархической модели данных «Мои школьные друзья»***

Иерархическая структура представляется перевернутым деревом, как в примере. Объекты в этой структуре соединены линиями связи. Заметим, что линии связи отвечают отношению «один ко многим». То есть одному объекту верхнего уровня отвечают много объектов нижнего уровня.

Составим модель, изображающую участие рябят в различных командах.



**Другой тип отношений, а именно: отношения «многие ко многим», устанавливается в сетевой модели данных.**

*Сетевая модель БД, как и иерархическая модель, представляет совокупность объектов различного уровня, однако схема связей между объектами может быть любой.*



# ЗАПИСАТЬ ОТВЕТЫ В ТЕТРАДЬ!

## Закрепление изученного.

1. Что называется БД?
2. Перечислите основные свойства БД.
3. Что такое СУБД?
4. Каким требованиям должны удовлетворять СУБД?
5. Что такое реляционная модель данных?
6. Приведите пример реляционной модели данных.
7. Как осуществляется связь таблиц в реляционной БД?
8. Что такое иерархическая структура данных?
9. Чем отличается сетевая модель данных от иерархической?
10. Приведите примеры иерархической и сетевой моделей данных.

### **Задание 1**

1. Создать структуру таблицы базы данных «Ученик», содержащую следующие поля: фамилия, имя, школа, класс, дата рождения, вес.
2. Определить первичный ключ таблицы.

### **Задание 2**

1. Создать структуру таблицы базы данных «Библиотека», содержащую следующие поля: инвентарный номер, автор, название, издательство, количество страниц, номер библиотеки.
2. Определить первичный ключ таблицы.

### **Задание 3**

Преобразовать приведенную ниже информацию к табличному виду, определив имя таблицы и название каждого поля:

Оля, Петя, 13, пение, 14, баскетбол, Вася, Катя, 13, хоккей, баскетбол, футбол, 15, 11, Коля, 11, танцы, Сережа.

Первичный ключ – это ключевое поле!

**-Сфотографировать выполненные задания и прислать мне на почту [ssharlota@rambler.ru](mailto:ssharlota@rambler.ru)**