

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ РЕСПУБЛИКИ КРЫМ  
"КЕРЧЕНСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ"

Презентация на тему: Иерархические, сетевые и реляционные БД

Выполнила  
студентка группы ПК-17 1/9  
Пиякова Е.К.

2020

# БАЗЫ ДАННЫХ

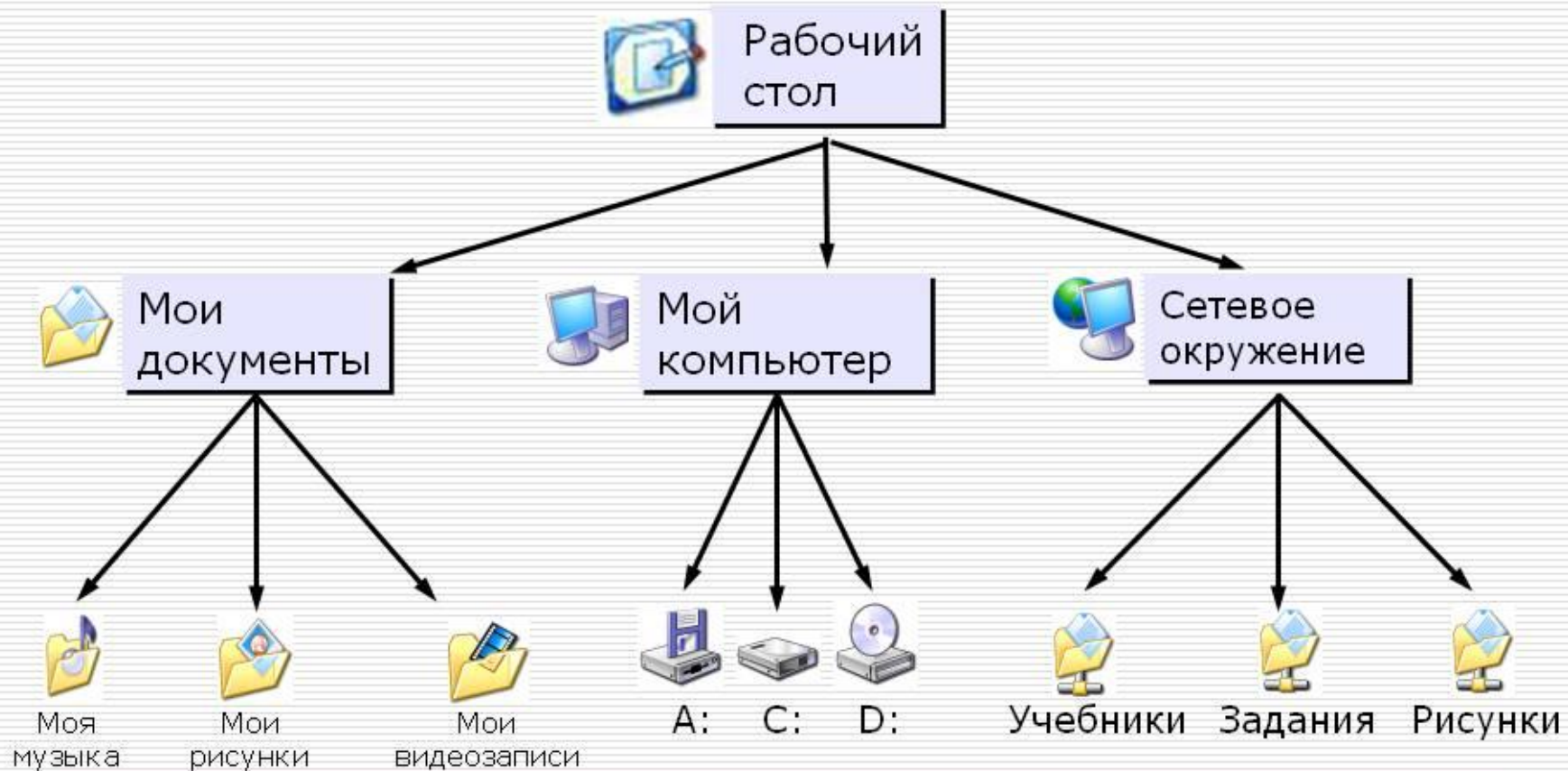
Базы данных представляют собой информационные модели, содержащие данные об объектах и их свойствах. Базы данных хранят информацию о группах объектов с одинаковым набором свойств. Информация в базах данных хранится в упорядоченном виде. Например, все записи в записной книжке упорядочены по алфавиту. Существует несколько различных структур информационных моделей и соответственно различных типов моделей данных: табличные, иерархические и сетевые.

# ИЕРАРХИЧЕСКИЕ БАЗЫ ДАННЫХ

Иерархические базы данных графически могут быть представлена как дерево, состоящее из объектов различных уровней. Между объектами существуют связи. Каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка (объект, более близкий к корню) к потомку (объект более низкого уровня). При этом объект - предок может не иметь потомков, или иметь их несколько. Объект - потомок обязательно имеет только одного предка. Объекты, имеющие общего предка, называются близнецами. Примером иерархической базы данных является каталог папок Windows, с которым можно работать, запустив ПРОВОДНИК. Верхний уровень занимает папка Рабочий стол. На втором уровне находятся папки Мой компьютер, Мои документы и так далее, которые являются потомками папки Рабочий стол, а между собой являются близнецами. В свою очередь, папка Мой компьютер является предком по отношению к папкам третьего уровня.

**Иерархическая БД** – это набор данных в виде многоуровневой структуры (дерева).

---



# СЕТЕВЫЕ БАЗЫ ДАННЫХ

- Сетевая модель расширяет иерархическую модель, позволяя группировать связи между записями в множества. С логической точки зрения связь — это не сама запись. Связи лишь выражают отношения между записями. Как и в иерархической модели, связи ведут от родительской записи к дочерней, но на этот раз поддерживается множественное наследование.
- Следуя спецификации CODASYL, сетевая модель поддерживает DDL (Data Definition Language — язык определения данных) и DML (Data Manipulation Language — язык обработки данных). Это специальные языки, предназначенные для определения структуры базы данных и составления запросов. Несмотря на их наличие, программист по-прежнему должен знать структуру базы данных.

- В сетевой модели допускаются отношения "многие ко многим", а записи не зависят друг от друга. При удалении записи удаляются и все ее связи, но не сами связанные записи.
- В сетевой модели требуется, чтобы связи устанавливались между существующими записями во избежание дублирования и искажения целостности. Данные можно изолировать в соответствующих таблицах и связать с записями в других таблицах.
- Программисту не нужно заботиться о том, как организуется физическое хранение данных на диске. Это ослабляет зависимость приложений и данных. Но в сетевой модели требуется, чтобы программист помнил структуру данных при формировании запросов.
- Оптимальную структуру базы данных сложно сформировать, а готовую структуру трудно менять. Если вид таблицы претерпевает изменения, все отношения с другими таблицами должны быть установлены заново, чтобы не нарушилась целостность данных. Сложность подобной задачи приводит к тому, что программисты зачастую отменяют некоторые ограничения целостности ради упрощения приложений.



# РЕЛЯЦИОННАЯ БАЗА ДАННЫХ

Определение «реляционные базы данных» означает, что таблицы в этой БД могут вступать в отношения и находиться в связи между собой

Например, одна таблица может ссылаться на другую таблицу. Это часто требуется, чтобы сократить объём и избежать дублирования информации.

В сценарии с дневником погоды пользователь вводит название своего города. Это название сохраняется вместе с погодными данными.

Но можно поступить иначе:

- Создать новую таблицу с именем [cities](#).
- Все города в России известны, поэтому их все можно добавить в одну таблицу.
- Переделать форму, изменив поле ввода города с текстового на поле типа [select](#), чтобы пользователь не вписывал город, а выбирал его из списка.
- При сохранении погодной записи, в поле для города поставить ссылку на соответствующую запись из таблицы городов.



Так мы решим сразу две задачи:

- Сократим объём хранимой информации, так как погодные записи больше не будут содержать название города;
- Избежим дублирования: все пользователи будут выбирать один из заранее определённых городов, что исключит опечатки.

Связи между таблицами в БД бывают разных видов.

В примере выше использовалась связь типа «один-ко-многим», так как одному городу может соответствовать множество погодных записей, но не наоборот!

Бывают связи и других типов: «один-к-одному» и «многие-ко-многим», но они используются значительно реже.

# Реляционные базы данных

