

ЛЕКЦИЯ

Тема: «Модели данных»

Дисциплина «Базы данных в ИС»

ОП «Информационные системы»

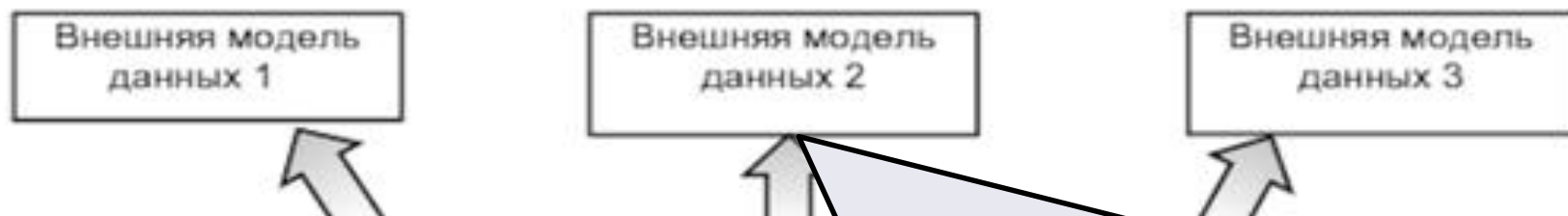
Авторы: ст.преп.каф.ИВС Саданова Б.М.
ст.преп.каф.ИВС Олейникова А.В.

План лекции:

1. Трехуровневая модель системы управления базой данных
2. Понятие модели данных, компоненты модели данных
3. Классификация моделей данных
4. Иерархические системы
5. Сетевые системы
6. Реляционные системы

Цель лекции: дать общее представление о модели данных СУБД как средства для представления концептуальной модели при создании базы данных, рассмотреть типовые модели данных (сетевая модель, иерархическая модель, реляционная модель)

Трехуровневая модель системы управления базой данных



Физический уровень — собственно данные, расположенные в файлах или в страничных структурах, расположенных на внешних носителях информации.

Концептуальный уровень — основное управляющее звено, БД представлена в наиболее общем виде, который объединяет данные, используемые всеми приложениями, работающими с БД.

Отражает обобщенную модель предметной области.

Как любая модель, концептуальная модель отражает только существенные, с точки зрения обработки, особенности объектов реального мира.

Трехуровневая модель системы управления базой данных

- *Логическая независимость* предполагает возможность изменения одного приложения без корректировки других приложений, работающих с этой же базой данных.
- *Физическая независимость* предполагает возможность переноса хранимой информации с одних носителей на другие при сохранении работоспособности всех приложений, работающих с данной базой данных.

Модель данных

Данные:

- это набор конкретных значений, параметров, характеризующих объект, условие, ситуацию или любые другие факторы. Примеры данных: Петров Николай Степанович, \$30 и т. д.
- данные не обладают определенной структурой, данные становятся информацией тогда, когда пользователь задает им определенную структуру, то есть осознает их смысловое содержание.

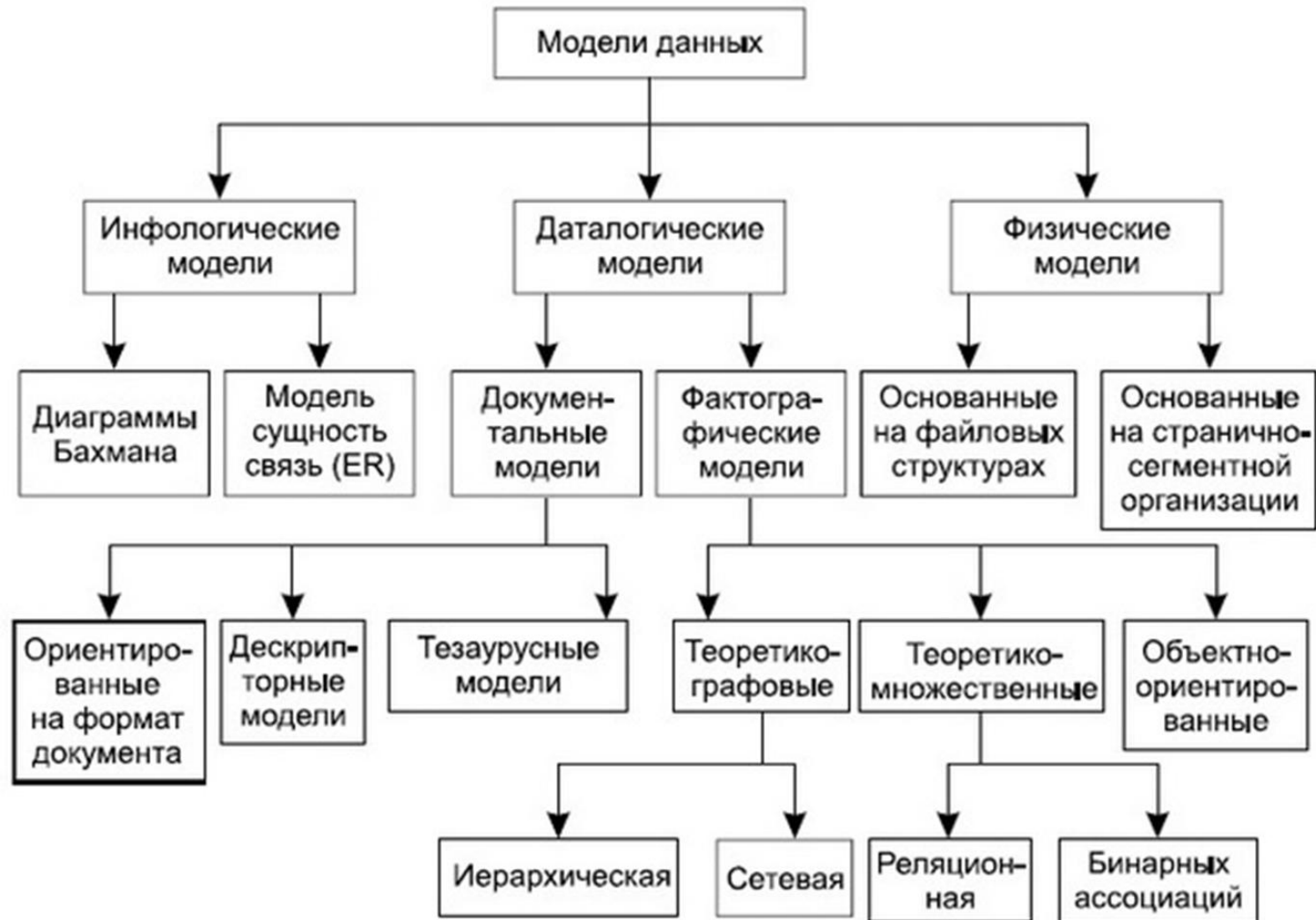
Модель данных :

это формализованное описание, отражающее состав и типы данных, а также взаимосвязь между ними

Модель данных

- Любая модель данных содержит три компоненты:
 1. *структуру данных* - описывает точку зрения пользователя на представление данных
 2. *набор допустимых операций*, выполняемых на структуре данных. Модель данных предполагает, как минимум, наличие языка определения данных (ЯОД), описывающего структуру их хранения, и языка манипулирования данными (ЯМД), включающего операции извлечения и модификации данных.
 3. *ограничения целостности* - механизм поддержания соответствия данных предметной области на основе формально описанных правил.

Классификация моделей данных

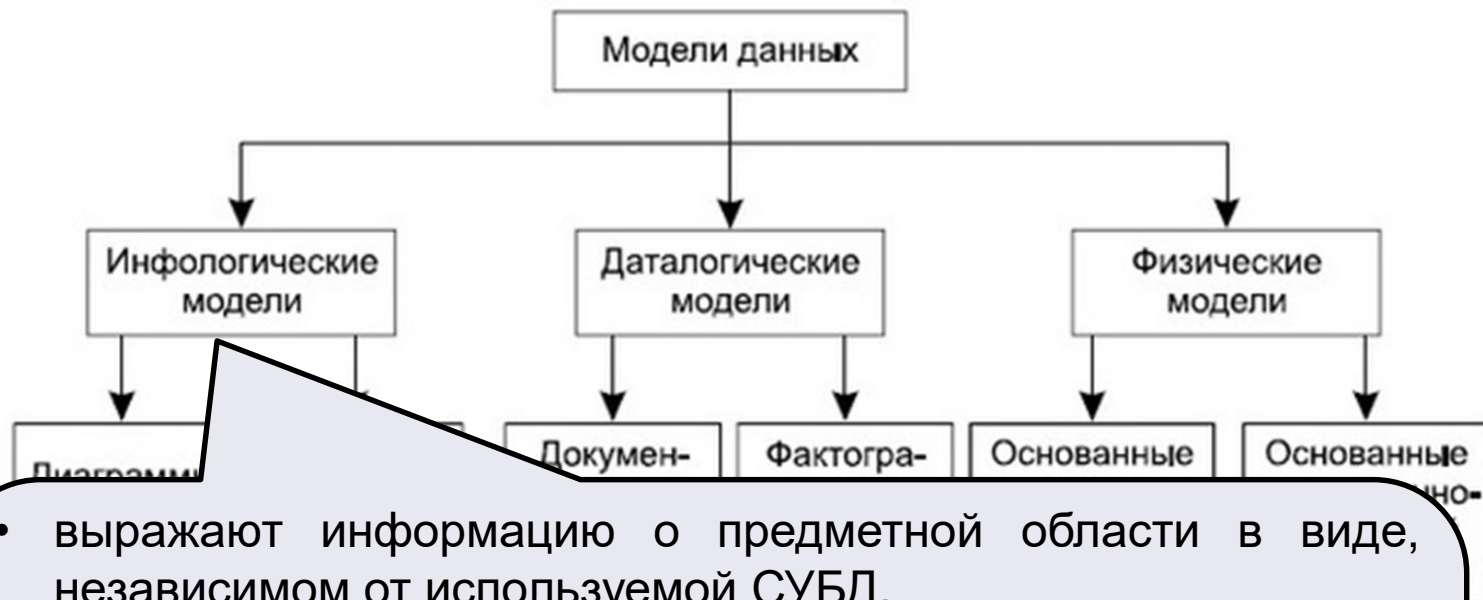


Классификация моделей данных



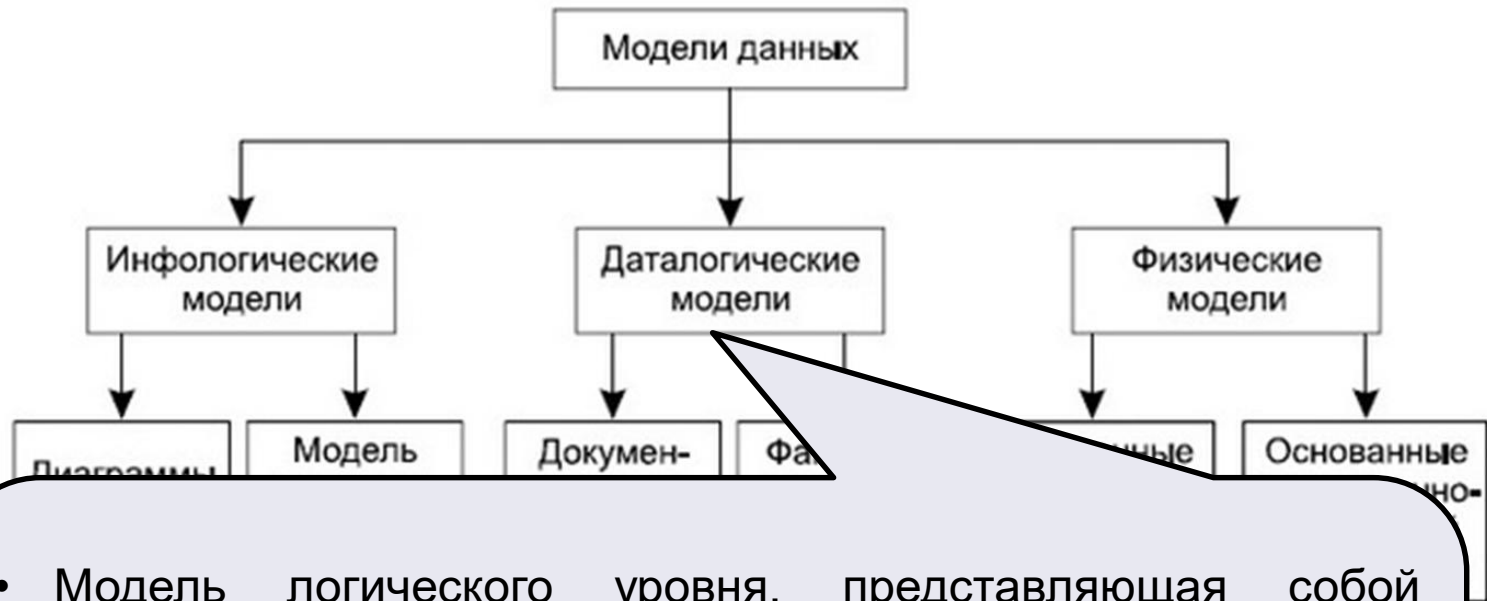
- оперируют категориями, касающимися организации внешней памяти и структур хранения, используемых в данной операционной среде.
- используются различные методы размещения данных, основанные на файловых структурах: это организация файлов прямого и последовательного доступа, индексных файлов и инвертированных файлов, файлов, использующих различные методы хэширования, взаимосвязанных файлов.
- используется страничная организация данных.

Классификация моделей данных



- выражают информацию о предметной области в виде, независимом от используемой СУБД.
- отражают в естественной и удобной форме информационно-логический уровень абстрагирования, связанный с фиксацией и описанием объектов предметной области, их свойств и их взаимосвязей.
- используются на ранних стадиях проектирования для описания структур данных в процессе разработки приложения

Классификация моделей данных



- Модель логического уровня, представляющая собой отображение логических связей между элементами данных независимо от их содержания и среды хранения.
- строится в терминах информационных единиц, допустимых в той СУБД, в среде которой будет создаваться БД.
- описывает собственно данные (информацию, которая будет записана в памяти компьютера) и связи между данными.

Классификация моделей данных

Документальные модели данных

- соответствуют представлению о слабоструктурированной информации, ориентированной в основном на свободные форматы документов и текстов на естественном языке.
- используют стандартный общий язык разметки — SGML (Standart Generalised Markup Language), языки HTML (HyperText Markup Language) и XML (Extensible Markup Language) которые позволяют определять оформление элементов документа

Классификация моделей данных

Тезаурусные модели

- основаны на принципе организации словарей, содержат определенные языковые конструкции и принципы их взаимодействия в заданной грамматике
- эффективно используются в системах-переводчиках, особенно многоязыковых переводчиках.
- принцип хранения информации в этих системах подчиняется тезаурусным моделям.

Классификация моделей данных

Дескрипторные модели

- самые простые из документальных моделей, они широко использовались на ранних стадиях использования документальных баз данных.
- каждому документу соответствовал дескриптор — описатель. Этот дескриптор имел жесткую структуру и описывал документ в соответствии с теми характеристиками, которые требуются для работы с документами в разрабатываемой документальной БД.
- обработка информации в таких базах данных велась исключительно по дескрипторам

Классификация моделей данных

Теоретико-графовые модели:

- отражают совокупность объектов реального мира в виде графа взаимосвязанных информационных объектов.
- в зависимости от типа графа выделяют иерархическую или сетевую модели.
- исторически эти модели появились раньше и в настоящий момент они используются реже, чем более современная реляционная модель данных.

Классификация моделей данных

Теоретико-множественные модели:

- отражают совокупность объектов реального мира в виде отношений - таблиц
- включают реляционную модель и модель бинарных отношений

Объектно-ориентированные модели:

- структура модели графически представлена в виде дерева, узлами которого являются объекты
- логическая структура похожа на структуру иерархической базы, но отличается методами манипулирования данными.
- *инкапсуляция* ограничивает область видимости имени свойства пределами того объекта, в котором оно определено.
- *Наследование*, наоборот, распространяет область видимости свойства на всех потомков объекта.
- *Полиморфизм* означает способность одного и того же программного кода работать с разнотипными данными.

Иерархические системы

Структура данных

- состоит из упорядоченного набора деревьев, т.е из упорядоченного набора нескольких экземпляров одного типа дерева.
- тип дерева состоит из одного "корневого" типа записи и упорядоченного набора из нуля или более типов поддеревьев (каждое из которых является некоторым типом дерева). Тип дерева в целом представляет собой иерархически организованный набор типов записи.
- организация данных в СУБД иерархического типа определяется в терминах: *элемент, агрегат, запись (группа), групповое отношение, база данных.*

Иерархические системы

- **Атрибут (элемент данных)** - наименьшая единица структуры данных. Каждому элементу при описании базы данных присваивается уникальное имя. По этому имени к нему обращаются при обработке. Элемент данных называют **полем**.
- **Запись** - именованная совокупность атрибутов. Использование записей позволяет за одно обращение к базе получить некоторую логически связанную совокупность данных. Именно записи изменяются, добавляются и удаляются. Тип записи определяется составом ее атрибутов. Экземпляр записи - конкретная запись с конкретным значением элементов.

Иерархические системы

- **Групповое отношение** - иерархическое отношение между записями двух типов.
- Родительская запись (владелец группового отношения) называется *исходной записью*, а дочерние записи (члены группового отношения) - *подчиненными*.
- Иерархическая база данных может хранить только такие древовидные структуры.

Иерархические системы

- Корневая запись каждого дерева обязательно должна содержать *ключ с уникальным значением*.
- Ключи некорневых записей должны иметь уникальное значение только в рамках группового отношения.
- Каждая запись идентифицируется *полным сцепленным ключом* - совокупность ключей всех записей от корневой по иерархическому пути.
- При графическом изображении групповые отношения изображают дугами ориентированного графа, а типы записей - вершинами (диаграмма Бахмана).

Иерархические системы

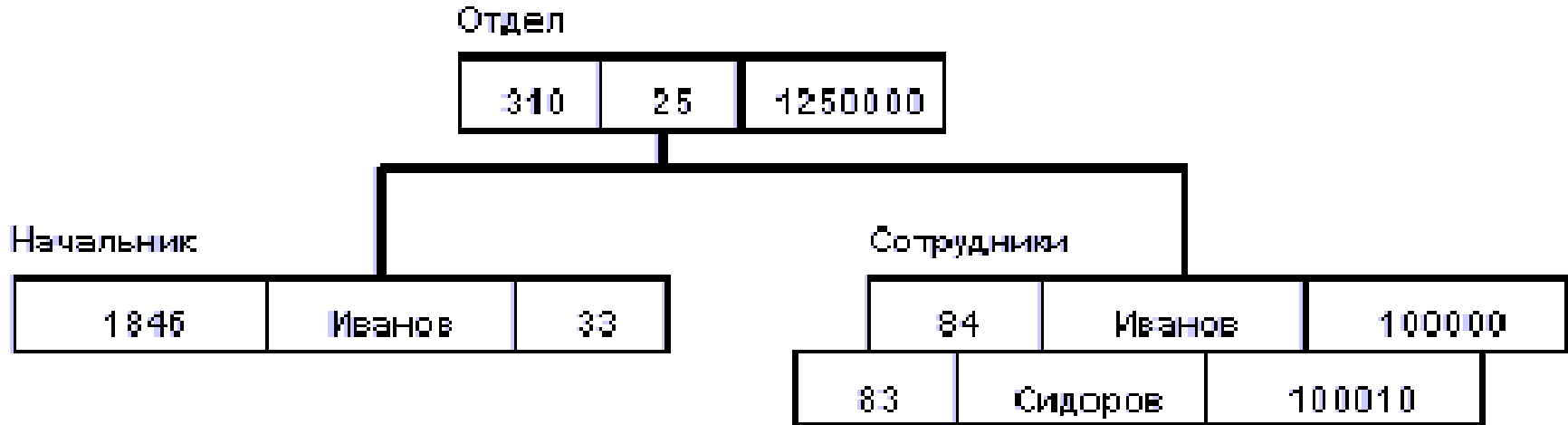
- Для групповых отношений в иерархической модели обеспечивается *автоматический режим включения и фиксированное членство*. Это означает, что для запоминания любой некорневой записи в БД должна существовать ее родительская запись. При удалении родительской записи автоматически удаляются все подчиненные.

Пример типа дерева



Иерархические системы

База данных



- Все экземпляры данного типа потомка с общим экземпляром типа предка называются **близнецами**.
- Для БД определен полный порядок обхода - сверху-вниз, слева-направо.

Иерархические системы

Манипулирование данными

- Примеры типичных операторов манипулирования иерархически организованными данными :
- Найти указанное дерево БД (например, отдел 310);
- • Перейти от одного дерева к другому;
- • Перейти от одной записи к другой внутри дерева (например, от отдела - к первому сотруднику);
- • Перейти от одной записи к другой в порядке обхода иерархии;
- • Вставить новую запись в указанную позицию;
- • Удалить текущую запись.

Иерархические системы

Манипулирование данными

- автоматически поддерживается целостность ссылок между предками и потомками.
- *основное правило*: никакой потомок не может существовать без своего родителя.
- поддержание целостности по ссылкам между записями, не входящими в одну иерархию, *не поддерживается*

Иерархические системы

Пример СУБД

- Типичным представителем (наиболее известным и распространенным) является Information Management System (IMS) фирмы IBM. Первая версия появилась в 1968 г.
- До сих пор поддерживается много баз данных, что создает существенные проблемы с переходом как на новую технологию БД, так и на новую технику.

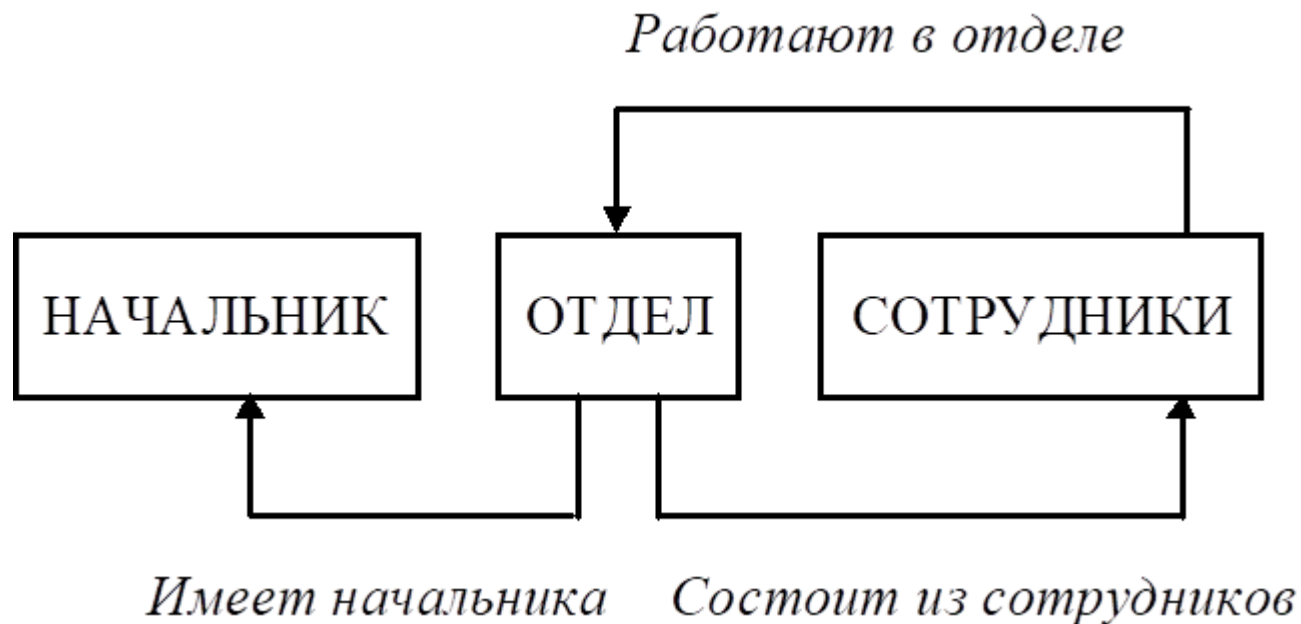
Сетевые системы

Структура данных

- Сетевой подход к организации данных является расширением иерархического. В иерархических структурах запись-потомок должна иметь в точности одного предка; в сетевой структуре данных потомок может иметь *любое число предков*.
- Сетевая БД состоит из набора записей и набора связей между этими записями, т.е. из набора экземпляров каждого типа из заданного в схеме БД набора типов записи и набора экземпляров каждого типа из заданного набора типов связи.
- *Тип связи* определяется для двух типов записи: *предка* и *потомка*.
- *Экземпляр типа связи* состоит из одного экземпляра типа записи предка и упорядоченного набора экземпляров типа записи потомка.

Сетевые системы

- Тип связи «Работают в отделе» : предок – сотрудник, потомок – отдел
- Экземпляры связи «Работают в отделе»:
 - Иванов работает в отделе 1
 - Петров работает в отделе 3 и в отделе 4



Сетевые системы

Манипулирование данными

- Примерный набор операций может быть следующим:
 - Найти конкретную запись в наборе однотипных записей (инженера Сидорова);
 - Перейти от предка к первому потомку по некоторой связи (к первому сотруднику отдела 310);
 - Перейти к следующему потомку в некоторой связи (от Сидорова к Иванову);
 - Перейти от потомка к предку по некоторой связи (найти отдел Сидорова);
 - Создать новую запись;
 - Уничтожить запись;
 - Модифицировать запись;
 - Включить в связь;
 - Исключить из связи;
 - Переставить в другую связь и т.д.

Сетевые системы

Ограничение целостности

- В принципе их поддержание не требуется, но иногда требуют целостности по ссылкам (как в иерархической модели).

Пример СУБД

- Типичным представителем является Integrated Database Management System (IDMS), предназначенная для использования на машинах основного класса фирмы IBM под управлением большинства операционных систем.

Реляционные системы

- Основы реляционной модели данных были впервые изложены в Е.Коддом в 1970 г.
- Кодд предложил использовать для обработки баз данных аппарат теории множеств и теории отношений.
- В реляционной модели данных достигается гораздо более высокий уровень абстракции данных, чем в иерархической или сетевой.
- Наиболее распространенная трактовка реляционной модели данных принадлежит К.Дейту.
- Реляционная модель состоит из трех частей:
 1. структурной части;
 2. целостной части;
 3. манипуляционной части.

Реляционные системы

- Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением
- **Структурная часть** описывает, какие объекты рассматриваются реляционной моделью.
- **Манипуляционная часть** описывает два эквивалентных способа манипулирования реляционными данными - *реляционную алгебру* и *реляционное исчисление*.
- **Целостная часть** описывает ограничения специального вида, которые должны выполняться для любых отношений в любых реляционных базах данных. Это *целостность сущностей* и *целостность внешних ключей (по ссылкам)*.

Реляционные системы

Структура данных

- единственной структурой данных, используемой в реляционных БД, является нормализованное n -арное отношение
- структурная часть модели состоит из следующих компонент:
 - домен;
 - отношение;
 - атрибут;
 - кортеж;
 - первичный ключ.

Реляционные системы

Манипулирование данными

- два фундаментальных механизма манипулирования реляционными БД - *реляционная алгебра* и *реляционное исчисление*.
- *реляционная алгебра* базируется на классической теории множеств, а *реляционное исчисление* - на классическом логическом аппарате исчисления предикатов первого порядка.
- основной функцией манипуляционной части реляционной модели является обеспечение меры реляционности любого конкретного языка реляционных БД: язык называется *реляционным*, если он обладает не меньшей выразительностью и мощностью, чем реляционная алгебра или реляционное исчисление.

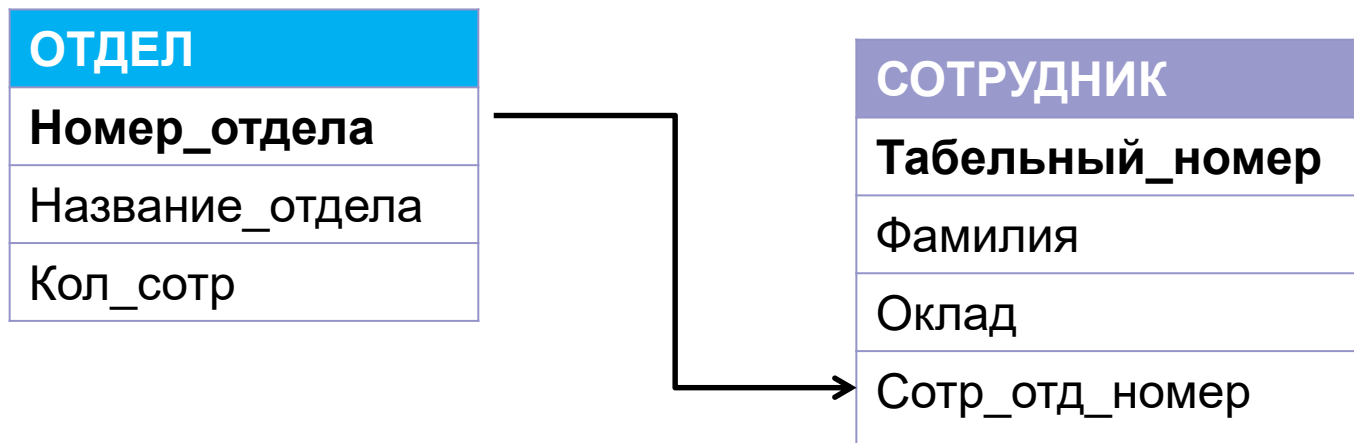
Реляционные системы

Ограничение целостности

- В целостной части реляционной модели данных фиксируются два базовых требования целостности, которые должны поддерживаться в любой реляционной СУБД:
- 1. **требование целостности сущностей**. Объекту или сущности реального мира в реляционных БД соответствуют кортежи отношений. Требование состоит в том, что любой *кортеж* любого отношения *отличим* от любого другого кортежа этого отношения, т.е. любое отношение должно обладать *первичным ключом*.
- 2. **требование целостности по ссылкам**

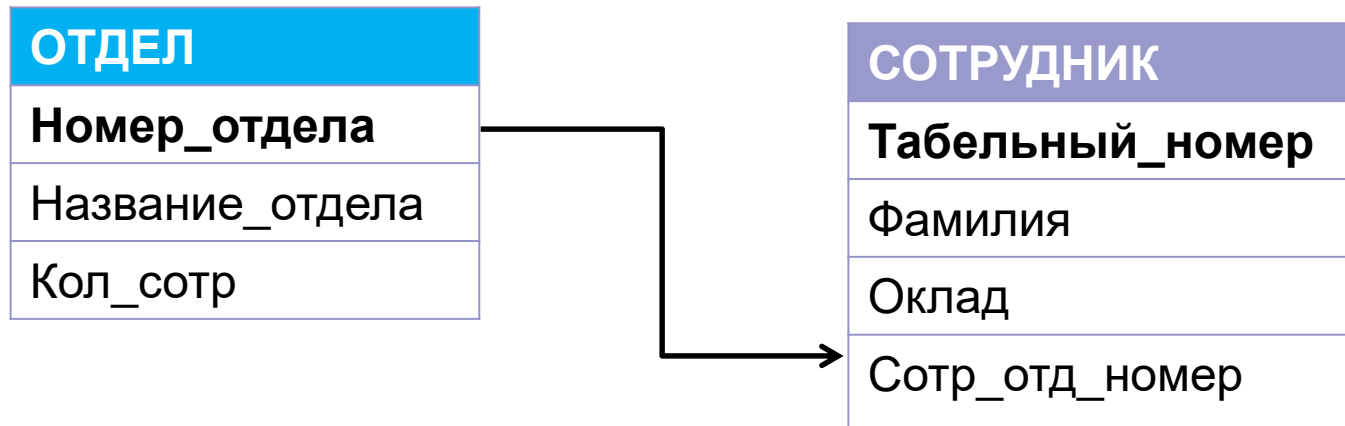
Пусть требуется представить в реляционной БД:

- 1. сущность ОТДЕЛ с атрибутами номер отдела, название отдела, количество сотрудников, набор сотрудников отдела).
- 2. сущность СОТРУДНИК с атрибутами номер сотрудника, фамилия сотрудника, заработная плата сотрудника.
- При проектировании БД имеем два отношения:
- ОТДЕЛЫ (Номер_отдела, Название_отдела, Кол_сотр)
- СОТРУДНИКИ (Табельный_номер, Фамилия, Оклад, Сотр_отд_номер)



Требование целостности по ссылкам:

- Значение атрибута *Сотр_отд_номер* в любом кортеже отношения СОТРУДНИКИ должно соответствовать значению атрибута *Номер_отдела* в некотором кортеже отношения ОТДЕЛЫ.
- Атрибут такого рода называется *внешним ключом*, т.е. его значения однозначно характеризуют сущности, представленные кортежами некоторого другого отношения (т.е. задают значения их первичного ключа).
- Отношение, в котором определен *внешний ключ*, *ссылается* на соответствующее отношение, в котором такой же атрибут является *первичным ключом*.



Целостная часть модели реляционной БД

2. Требование целостности по ссылкам

- Требование целостности по ссылкам, или требование внешнего ключа состоит в том, что для каждого значения внешнего ключа, появляющегося в *ссылающемся отношении*, в отношении, на которое ведет ссылка, *должен найтись кортеж* с таким же значением первичного ключа, либо значение внешнего ключа должно быть неопределенным (т.е. ни на что не указывать).
- Для нашего примера это означает, что если для сотрудника указан номер отдела, то этот отдел должен существовать.

Целостная часть модели реляционной БД

- Ограничения целостности сущности и по ссылкам должны поддерживаться СУБД.
- Для соблюдения *целостности сущности* достаточно гарантировать отсутствие в любом отношении кортежей с одним и тем же значением первичного ключа.
- Для соблюдения *целостности по ссылкам* необходимо:
 - ✓ При обновлении ссылающегося отношения (вставке новых кортежей или модификации значения внешнего ключа в существующих кортежах) следить за тем, чтобы *не появлялись некорректные значения внешнего ключа*.
 - ✓ при удалении кортежа из отношения, на которое ведет ссылка, существуют *3 подхода*, каждый из которых поддерживает целостность по ссылкам.

Целостная часть модели реляционной БД

2. Требование целостности по ссылкам

- 1. *запрещается* производить удаление кортежа, на который существуют ссылки (т.е. сначала нужно либо удалить ссылающиеся кортежи, либо соответствующим образом изменить значения их внешнего ключа).
- 2. при удалении кортежа, на который имеются ссылки, во всех ссылающихся кортежах значение внешнего ключа автоматически становится *неопределенным*.
- 3. каскадное удаление состоит в том, что при удалении кортежа из отношения, на которое ведет ссылка, из ссылающегося отношения *автоматически удаляются* все ссылающиеся кортежи.

Контрольные вопросы:

1. Дайте определение понятию модели данных и опишите ее компоненты
2. Приведите классификацию моделей данных
3. Иерархическая модель БД ее характеристики
4. Сетевая модель БД ее характеристики
5. Реляционная модель БД ее характеристики
6. Опишите механизмы ограничений целостности в реляционных моделях БД

Список литературы:

1. Т. Конноли. Базы данных. Проектирование, реализация и сопровождение. Теория и практика.: Пер.с англ. – М.: Изд.дом «Вильямс», 2012. – 1440 с.
2. К.Дейт. Введение в системы БД. изд.6-е. :Пер.с англ. - М.: Изд.дом "Вильямс", 2014. – 1200с.
3. Д.Ульман. Введение в системы баз данных. – М.: Издательство «Лори», 2015. – 853с.



Спасибо за внимание!