

Брянский государственный технический университет

Кафедра «Информатика и программное
обеспечение»

Доцент, к.т.н.
Копелиович Дмитрий Игоревич

БАЗЫ ДАННЫХ

Вид учебной нагрузки	Объём в часах 6-й семестр
Лекции	34
Лабораторные работы	34
Курсовой проект	+
Экзамен	+
Всего аудиторных занятий	68
Самостоятельная работа	72
Итого	140

ТРЕБОВАНИЯ ГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО СТАНДАРТА

Назначение и основные компоненты системы баз данных; обзор современных систем управления базами данных (СУБД); уровни представления баз данных; понятия схемы и подсхемы; модели данных; иерархическая, сетевая и реляционная модели данных; схема отношения; язык манипулирования данными для реляционной модели; реляционная алгебра и язык SQL; проектирование реляционной базы данных, функциональные зависимости, декомпозиция отношений, транзитивные зависимости, проектирование с использованием метода сущность - связь; изучение одной из современных СУБД по выбору; создание и модификация базы данных; поиск, сортировка, индексирование базы данных, создание форм и отчетов; физическая организация базы данных; хешированные, индексированные файлы; защита баз данных; целостность и сохранность баз данных.

ОСНОВНЫЕ ПОНЯТИЯ

- **Информация** – сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний.
- **Данные** – информация, представленная в виде, позволяющем автоматизировать ее сбор, хранение и дальнейшую обработку человеком или информационным средством. Данные – это запись в соответствующем коде наблюдения, факта, объекта, пригодная для коммуникации, интерпретации, обработки и получения новой информации.

- **Информационная система** – система, предназначенная для обработки информации.
- **Обработка данных** – это совокупность задач, осуществляющих преобразование массивов данных. Обработка данных включает в себя ввод данных в ЭВМ, отбор данных по каким-либо критериям, преобразование структуры данных, перемещение данных на внешней памяти ЭВМ, вывод данных, являющихся результатом решения задач, в табличном или в каком-либо ином удобном для пользователя виде.
- **Система обработки данных (СОД)** – это набор аппаратных и программных средств, осуществляющих выполнение задач по управлению данными.

- **База данных** – совокупность взаимосвязанных систематизированных, структурированных данных, при такой минимальной избыточности, которая допускает их использование оптимальным для одного или нескольких приложений в определенной предметной области. Доступ к информации, хранящейся в БД, осуществляется с помощью специальных методов.
- **Предметная область** – это отраженная в БД, совокупность объектов реального мира с их связями, относящихся к некоторой области знаний и имеющая практическую ценность для пользователя.
- **Объект** – элемент предметной области, который можно четко идентифицировать.
- **Атрибут** – это логически неделимый элемент, относящийся к свойству некоторого объекта или процесса.

- Множество всех возможных значений атрибута называют **доменом**.
- Совокупность атрибутов характеризующих один объект называется **записью**. Тип записи определяется свойствами объекта.
- **Ключ** – атрибут или совокупность атрибутов, однозначно определяющие запись.

Система управления базами данных

- **Система управления базами данных (СУБД) (DBMS)** – система программного обеспечения, имеющая средства обработки на языке БД, позволяющая обрабатывать обращения к БД, которые поступают от прикладных программ или конечных пользователей, и поддерживать целостность базы данных.

СУБД выполняет следующие действия:

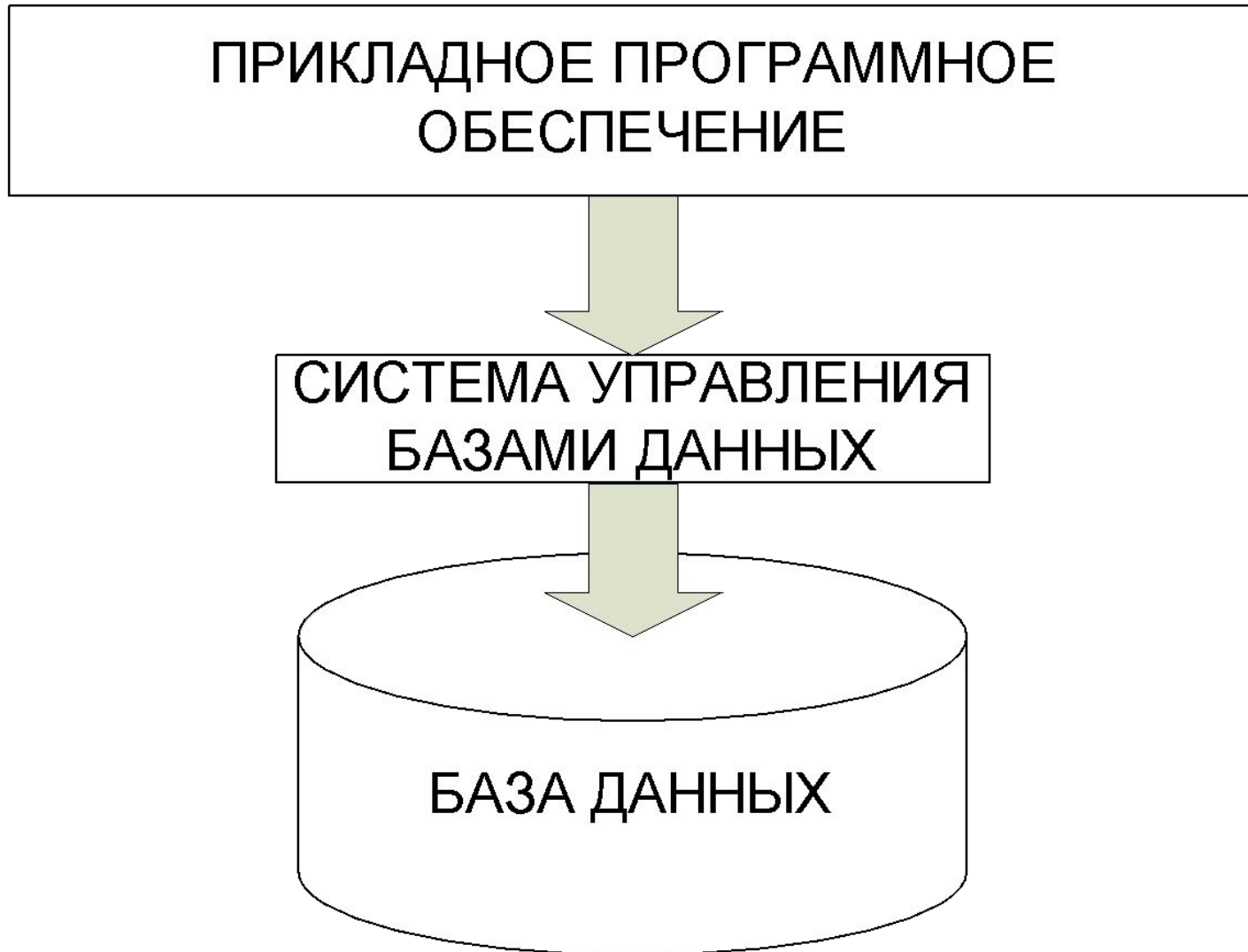
- Позволяет определять базу данных с указанием типа данных, их структуры, ограничений для данных, хранимых в базе.
- Позволяет вставлять, обновлять, удалять и извлекать данные из базы.
- Осуществляет защиту данных

Защита данных

- Системы защиты от несанкционированного доступа
- Системы поддержки целостности данных, обеспечивающей непротиворечивое состояние хранимых данных.
- Системы управления параллельной работой приложений, контролирующей процессы совместного доступа к файлам.
- Системы восстановления, позволяющей восстановить базу данных до предыдущего непротиворечивого состояния.

- **Система базы данных** – совокупность базы данных, СУБД, соответствующего оборудования и персонала.

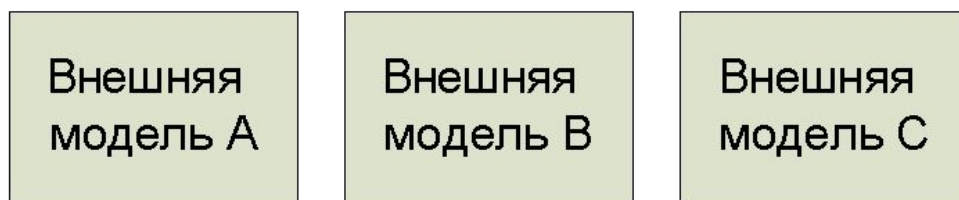
Компоненты системы обработки данных.



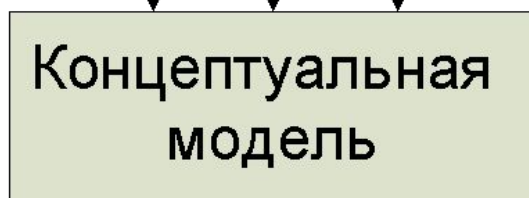
Трехуровневая архитектура

Уровни представления базы данных

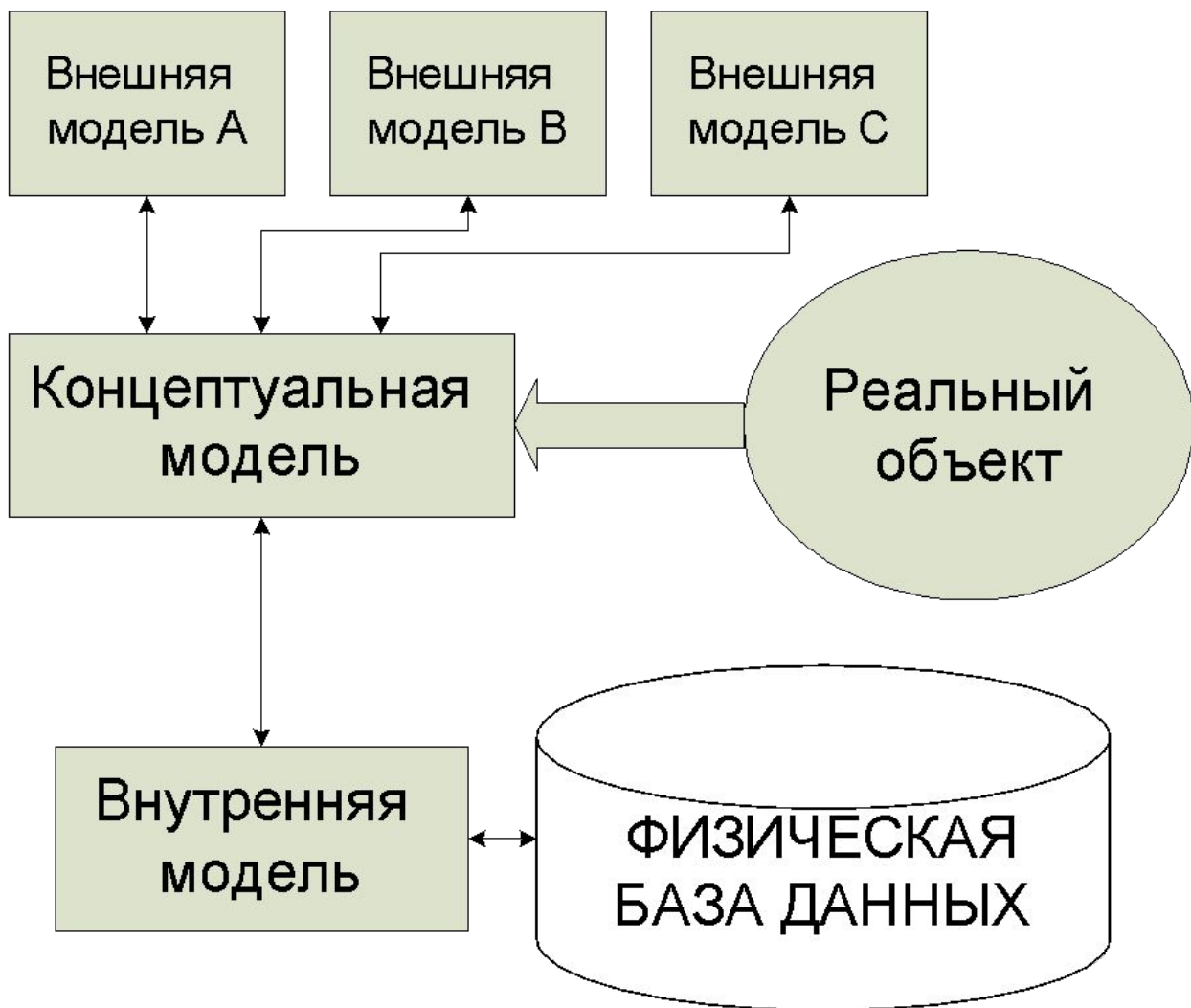
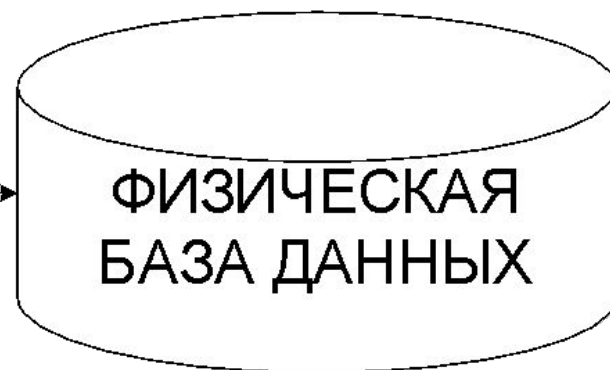
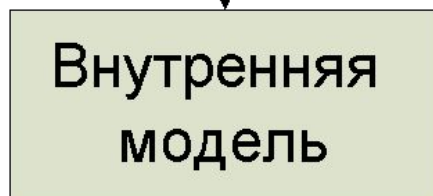
Внешний
уровень



Концептуальный
уровень



Внутренний
уровень



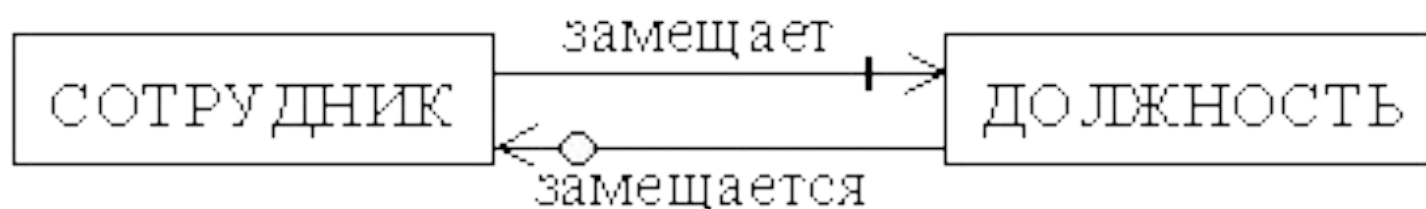
Предметная область
информационной системы.
Основные понятия, модель
сущность-связь.

Независимость от данных

Основным назначением трехуровневой архитектуры является обеспечение независимости от данных, которая означает, что изменения на нижних уровнях никак не влияют на верхние уровни. Различают два типа независимости от данных: логическую и физическую.

- Логическая независимость от данных означает полную защищенность внешних схем от изменений, вносимых в концептуальную схему.
- Физическая независимость от данных означает защищенность концептуальной схемы от изменений, вносимых во внутреннюю схему.

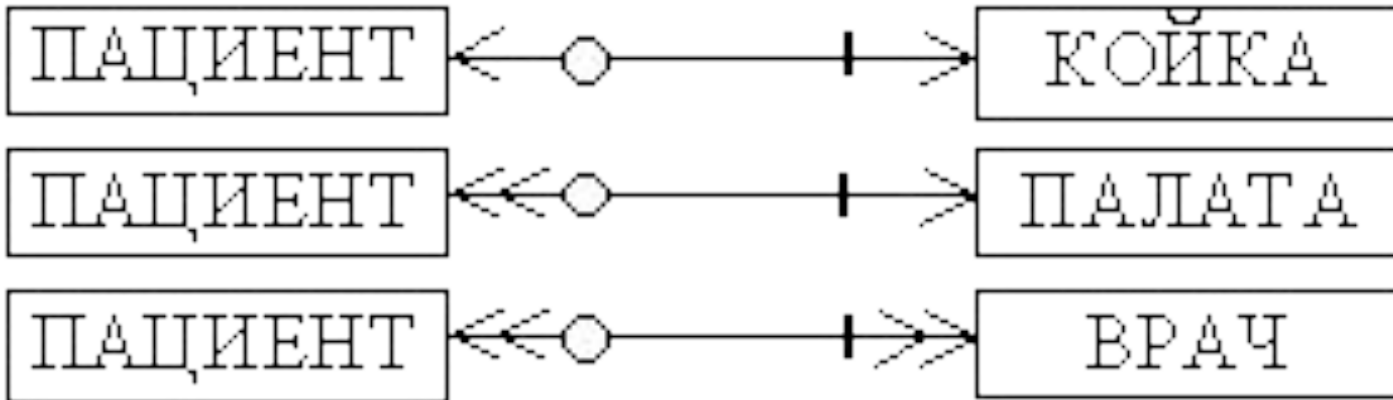
Обязательная и факультативная связи



—|> — обязательная связь

—○> — факультативная связь

Типы множественных связей



Этапы концептуального проектирования базы данных:

- Определение объектов.
- Определение связей.
- Определение атрибутов объектов.
- Определение доменов атрибутов.
- Определение атрибутов, являющихся потенциальными первичными ключами.
- Создание диаграммы «сущность-связь»

Модели данных

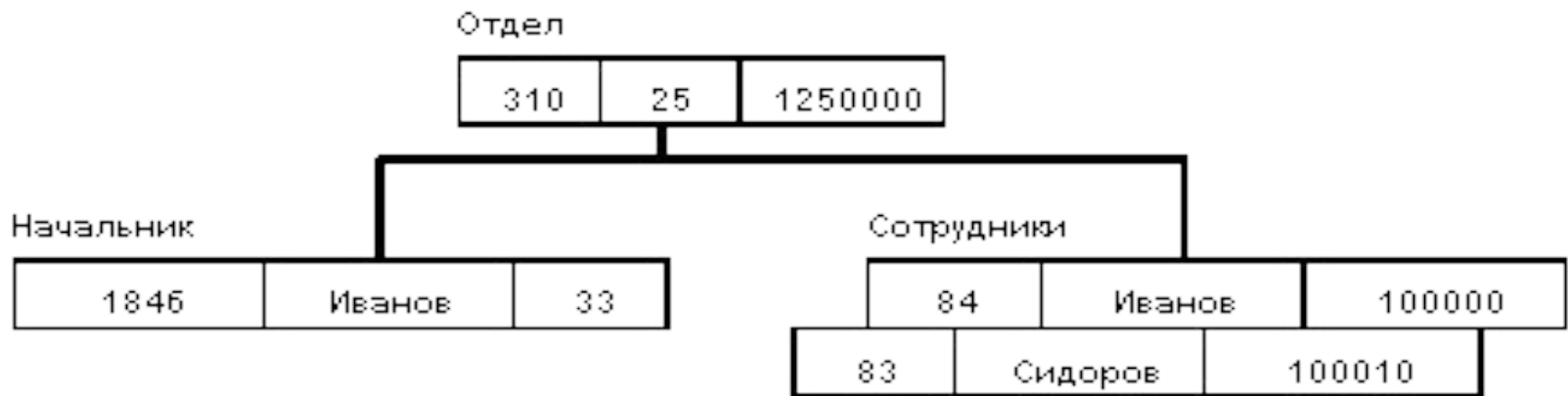
- Иерархическая
- Сетевая
- Реляционная
- Объектно-ориентированная
- Объектно-реляционная

Иерархическая модель

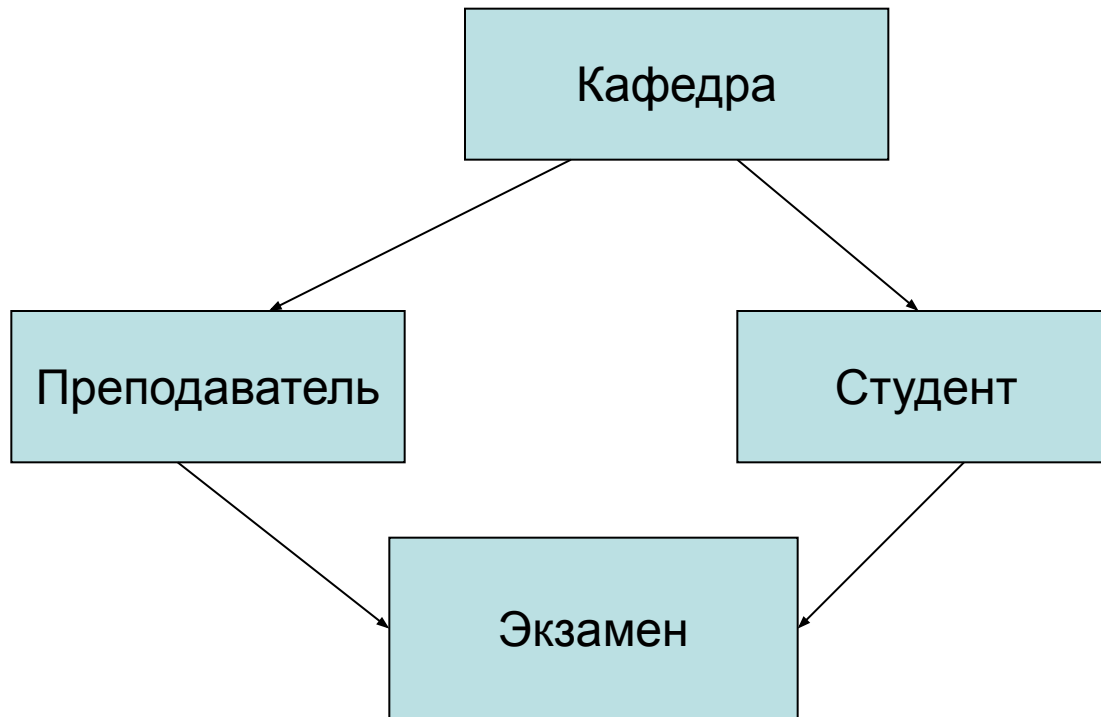
Иерархическую структуру данных можно представить в виде набора графов «древовидной структуры».

Основное правило: каждый потомок имеет только одного родителя.

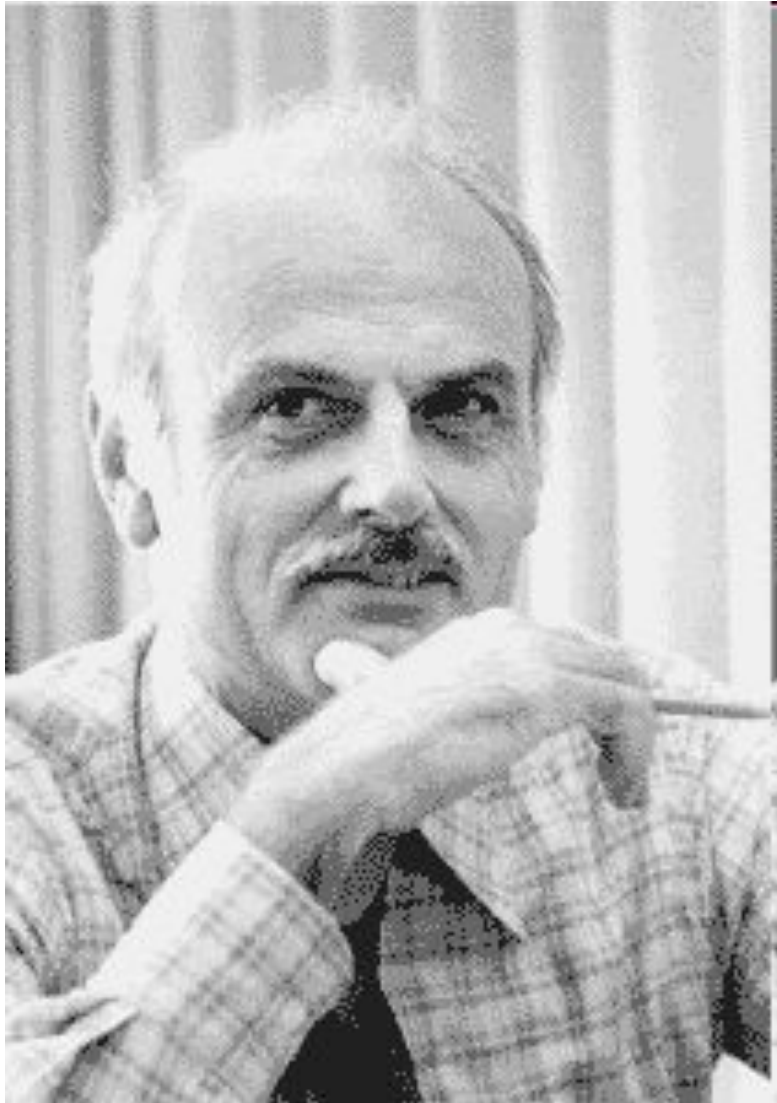
- Пример:



Сетевая модель



Реляционная модель



В 1970 году Э. Ф. Кодд (E. F. Codd), работавший в исследовательской лаборатории корпорации IBM, опубликовал очень важную и весьма своевременную статью о реляционной модели данных, позволявшей устранить недостатки прежних моделей. Вслед за этим появилось множество экспериментальных реляционных СУБД, а первые коммерческие продукты появились в конце 70-х - начале 80-х годов. Особенно следует отметить проект System R, разработанный в исследовательской лаборатории корпорации IBM, расположенной в городе Сан-Хосе, штат Калифорния, созданный в конце 70-х годов.

Реляционная модель

- 1. Данные воспринимаются пользователями как таблицы (и никак иначе).
- 2. Каждая таблица состоит из однотипных строк и имеет уникальное имя.
- 3. Строки имеют фиксированное число полей (столбцов) и значений (множественные поля и повторяющиеся группы недопустимы). Иначе говоря, в каждой позиции таблицы на пересечении строки и столбца всегда имеется в точности одно значение или ничего.
- 4. Строки таблицы обязательно отличаются друг от друга хотя бы единственным значением, что позволяет однозначно идентифицировать любую строку такой таблицы.

Реляционная модель

- 5. Столбцам таблицы однозначно присваиваются имена, и в каждом из них размещаются однородные значения данных (даты, фамилии, целые числа или денежные суммы).
- 6. Полное информационное содержание базы данных представляется в виде явных значений данных и такой метод представления является единственным. В частности, не существует каких-либо специальных "связей" или указателей, соединяющих одну таблицу с другой.
- 7. При выполнении операций с таблицей ее строки и столбцы можно обрабатывать в любом порядке безотносительно к их информационному содержанию. Этому способствует наличие имен таблиц и их столбцов, а также возможность выделения любой их строки или любого набора строк с указанными признаками.

Достоинства и недостатки «ранних» СУБД

- ***Сильные места ранних СУБД:***
- Развитые средства управления данными во внешней памяти на низком уровне;
- Возможность построения вручную эффективных прикладных систем;
- ***Недостатки:***
- Слишком сложно пользоваться;
- Фактически необходимы знания о физической организации;
- Прикладные системы зависят от этой организации;
- Их логика перегружена деталями организации доступа к БД.

Ключи

- Потенциальный ключ
- Первичный ключ
- Альтернативный ключ
- Суррогатный ключ
- Вторичный (внешний) ключ

Реляционная целостность

- целостность сущностей;
- ссылочная целостность;
- корпоративная целостность

Целостность сущностей

Целостность сущностей заключается в следующем: в базовом отношении ни один атрибут первичного ключа не может содержать отсутствующих значений, обозначаемых определителем NULL.

Определитель NULL указывает, что значение атрибута в настоящий момент неизвестно или неприемлемо для этого кортежа.

Ссылочная целостность

Ссылочная целостность означает следующее: если в отношении существует внешний ключ, то его значение должно либо соответствовать значению потенциального ключа некоторого кортежа в его базовом отношении, либо задаваться определителем NULL.

Корпоративная целостность

Администратор базы данных или пользователи могут сами определять дополнительные правила поддержки целостности данных. Такие правила называются корпоративные ограничения целостности.

ПРИМЕР:

- *ПОСТАВКИ (Номер поставки, Название товара, Цена товара, Количество, Дата поставки, Название поставщика, Адрес поставщика)*

Аномалии модификации данных.

- **1. Аномалия обновления:** в отношении *ПОСТАВКИ* она может возникнуть, если у какого-либо поставщика изменился адрес. Изменения должны быть внесены во все кортежи, соответствующие поставкам этого поставщика; в противном случае данные будут противоречивы.
- **2. Аномалия удаления:** при удалении записей обо всех поставках одного поставщика все данные о поставщике будут утеряны.
- **3. Аномалия добавления:** в нашем примере она возникнет, если с поставщиком заключен договор, но поставок от него еще не было. Информация о таком поставщике не может быть внесена в отношении *ПОСТАВКИ*, т.к. для него не определён ключ (номер поставки и название товара) и другие обязательные поля.

Правила внешних ключей

Что должно случиться при попытке удалить объект ссылки внешнего ключа? Например, удалить покупателя, для которого есть, по крайней мере, одна операция продажи. В общем существует по меньшей мере две возможности:

- **ЗАПРЕТИТЬ** – "ограничить" операцию удаления, до момента, когда не будет существовать соответствующих поставок (в противном случае операция запрещается);
- **КАСКАДИРОВАТЬ** – "каскадировать" операцию удаления, удаляя также соответствующие поставки.

- Что должно случиться при попытке обновить потенциальный ключ, на который ссылается внешний ключ? Например, при попытке обновить номер покупателя, для которого существует, по крайней мере, одна соответствующая операция. Как и для удаления здесь существует по меньшей мере две возможности:
- ЗАПРЕТИТЬ – "ограничить" операцию обновления до момента, когда не будет существовать соответствующих операций (в противном случае операция запрещается);
- КАСКАДИРОВАТЬ – "каскадировать" операцию обновления, обновляя также внешний ключ в соответствующих операциях.

Особенности проектирования реляционных БД.

- Каждое отношение соответствует одной сущности ПО и в него вносятся все атрибуты объекта, связанные с ним отношением 1:1.
- Связь типа 1:n реализуется с помощью внешнего ключа.
- Для реализации связи типа n:m между сущностями вводится дополнительное отношение, содержащее комбинации первичных ключей связанных отношений и атрибуты (свойства) этой связи.