

Классификация моделей данных

Одними из основополагающих в концепции баз данных являются обобщенные категории "*данные*" и "*модель данных*".

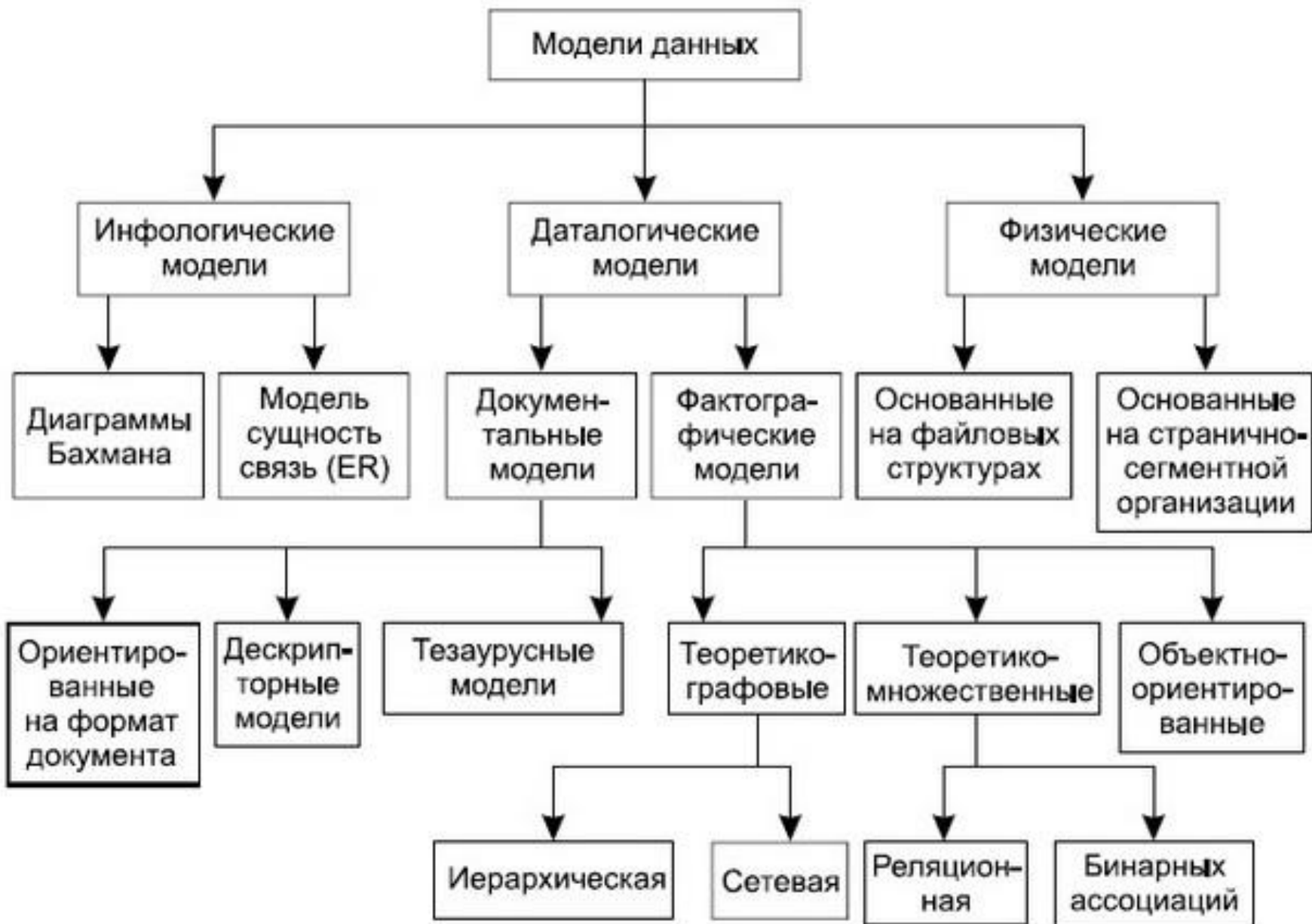
Данные — это набор конкретных значений, параметров, характеризующих объект, условие, ситуацию или любые другие факторы.

- *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 некоторая абстракция, которая, будучи приложима к конкретным данным, позволяет пользователям и разработчикам трактовать их уже как информацию, то есть сведения, содержащие не только данные, но и взаимосвязь между ними.

Классификация моделей данных



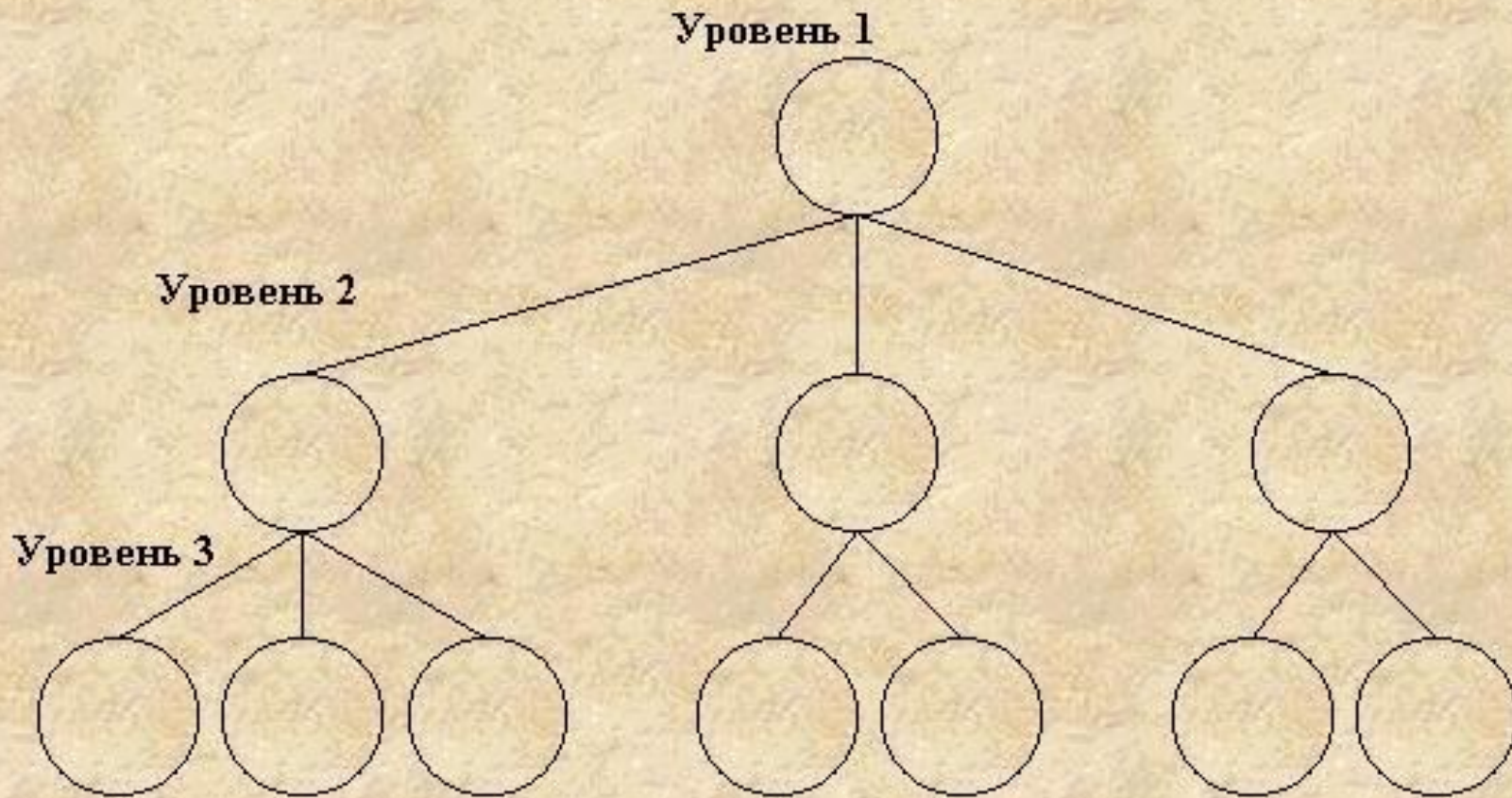
Инфологические, или *семантические модели* отражают в естественной и удобной для разработчиков и других пользователей форме информационно-логический уровень абстрагирования, связанный с фиксацией и описанием объектов предметной области, их свойств и их взаимосвязей.

Инфологические модели данных используются на ранних стадиях проектирования для описания структур данных в процессе разработки приложения, а *даталогические* модели уже поддерживаются конкретной СУБД.

- *Документальные модели данных* соответствуют представлению о слабоструктурированной информации, ориентированной в основном на свободные форматы документов, текстов на естественном языке.
- *Модели, основанные на языках разметки документов*, связаны прежде всего со стандартным общим языком разметки — SGML (Standart Generalised Markup Language), который был утвержден ISO в качестве стандарта еще в 80-х годах.

- **Тезаурусные модели** основаны на принципе организации словарей, содержат определенные языковые конструкции и принципы их взаимодействия в заданной грамматике. Эти модели эффективно используются в системах-переводчиках, особенно многоязыковых переводчиках.
- **Дескрипторные модели** — самые простые из документальных моделей, они широко использовались на ранних стадиях использования документальных баз данных. В этих моделях 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.

Иерархическая модель данных



Иерархическая модель данных

строится по принципу иерархии типов объектов, то есть один тип объекта является главным, а остальные, находящиеся на низших уровнях иерархии, – подчиненными.

Между главным и подчиненными объектами устанавливается взаимосвязь *«один ко многим»*.

Узлы и ветви образуют иерархическую древовидную структуру.

Узел является совокупностью атрибутов, описывающих объект. Наивысший в иерархии узел называется корневым (это главный тип объекта).

Корневой узел находится на первом уровне.

Зависимые узлы (подчиненные типы объектов) находятся на втором, третьем и др. уровнях.

Объекты, связанные иерархическими отношениями, образуют ориентированный граф.

Иерархическая модель данных

Основные понятия иерархической структуры:

Узел (элемент) – совокупность атрибутов данных, описывающих некоторый объект (на схеме это вершины графа).

Иерархическое дерево имеет только одну вершину (**корень**), неподчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне.

Таким образом, **иерархическая модель** представляет собой древовидный граф с записями в виде узлов (сегментами) и множествами в виде **ребер** (связь).

Иерархическая модель данных

Основные *операции* манипуляции с БД:

- поиск элемента в БД;
- переход от одного дерева к др.;
- перемещение от записи к записи внутри дерева;
- вставка некоторой записи;
- удаление элемента.

Достоинства:

- эффективное использование памяти и неплохие показатели временных затрат на выполнение операций;
- пригодны для формирования БД с теми данными, которые сами по себе имеют иерархическую структуру.

Недостатки:

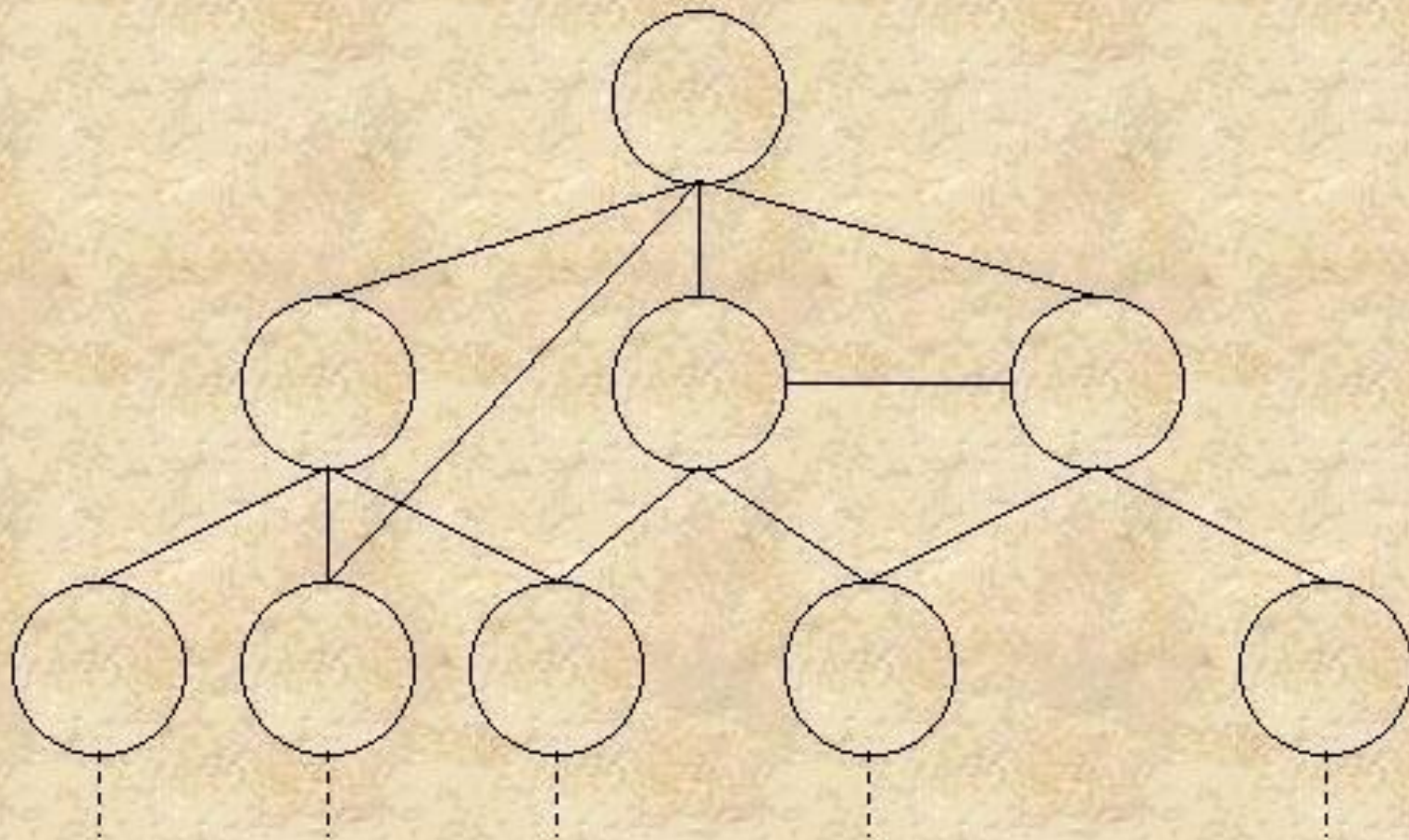
- громоздкость;
- сложность физической реализации для больших древовидных структур.

Иерархическая модель данных

- Пример



Сетевая модель данных



Сетевая модель данных

- Данные в такой модели представлены в виде *коллекции записей*, а *связи* – в виде наборов.
- *Сетевая модель* – это граф с записями в виде узлов графа и наборами в виде его ребер. В основу положены графы произвольной структуры, которые отражает взаимосвязи между данными в этой модели.
- В иерархической модели каждый потомок может иметь связь только с одним родителем, а в сетевой – с несколькими др. экземплярами.

Сетевая модель данных

Основные *операции* манипуляции с БД:

- поиск элемента в БД; переход от предка к некоторому потомку;
- переход от потомка к предку;
- вставка новой записи;
- удаление записи и др.

Достоинства:

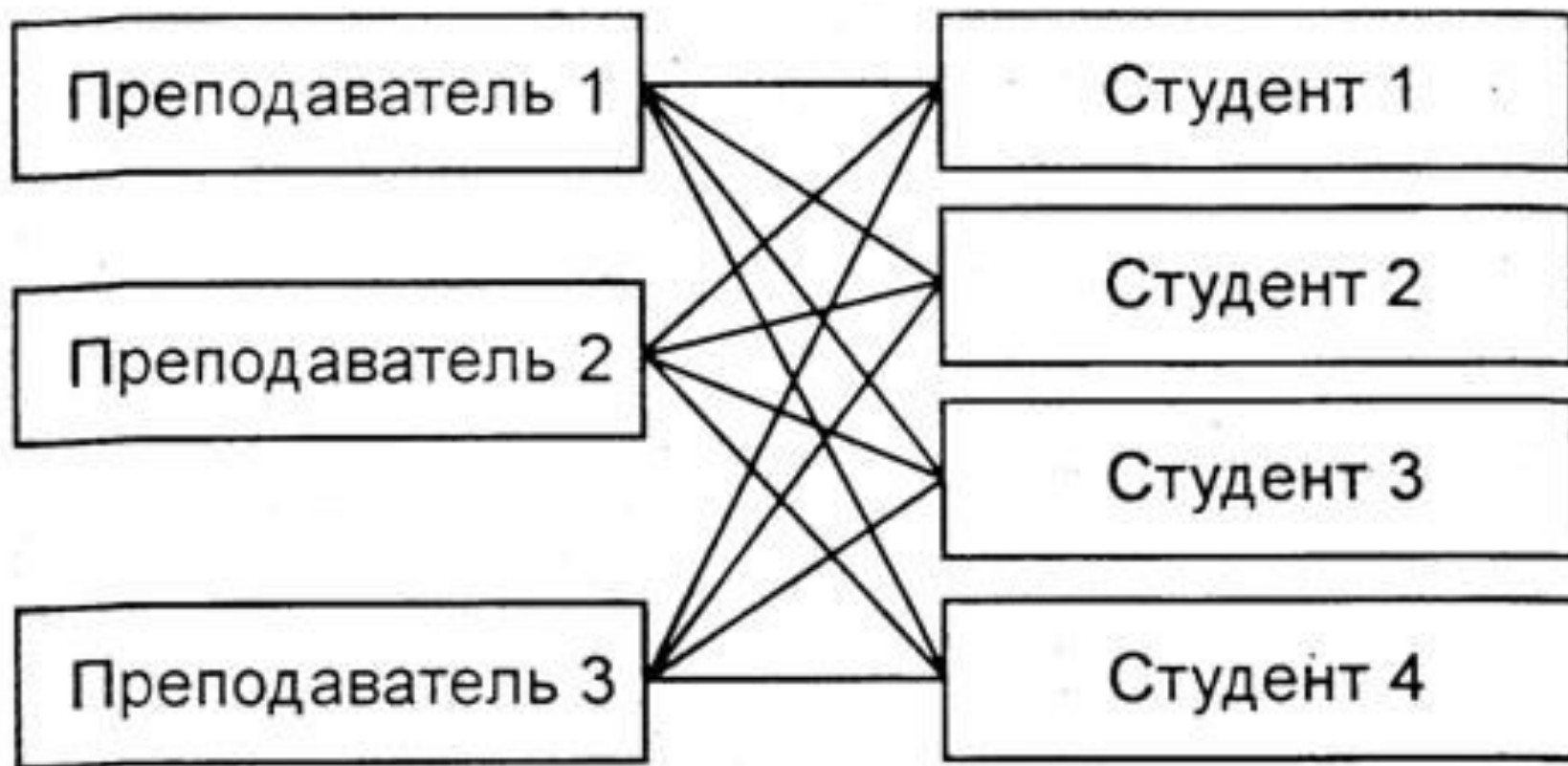
- эффективное использование затрат памяти (ресурсы) при манипулировании данными;
- использовать для решения многих задач из-за различных связей.

Недостатки:

- сложность физической реализации;
- жесткость связи между элементами данных накладывает ряд ограничений на удобство манипуляции данными; ослаблен контроль целостности связей между записями.

Сетевая модель данных

- Пример



Реляционная модель данных

Эти модели характеризуются:

- простотой структуры данных,
- удобными для пользователя табличным представлением
- возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель данных

Реляционная модель ориентирована на организацию данных в виде *двумерных таблиц*.

Каждая *реляционная таблица* представляет собой двумерный массив и обладает следующими свойствами:

- все столбцы в таблиц – однородные (имеют одинаковый тип);
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

Реляционная модель данных

Отношения представлены в виде таблиц, строки которых соответствуют кортежам или записям, а **столбцы** – атрибутам отношений, доменам, полям.

Поле, каждое значение которого однозначно определяет соответствующую запись, называется **простым ключом** (**ключевым полем**).

Если записи однозначно определяются значениями нескольких полей, то такая таблица БД имеет **составной ключ**.

Реляционная модель данных

- **Достоинства:** простота моделирования и физическая реализация, высокая эффективность обработки данных.
- **Недостатки:** отсутствие стандартных средств идентификации каждой отдельной записи.

Реляционная модель данных

- В реляционной базе данных каждая таблица должна иметь *первичный ключ (ключевой элемент)* – поле или комбинацию полей, которые единственным образом идентифицируют каждую строку в таблице.
- Благодаря своей простоте и естественности представления реляционная модель получила наибольшее распространение в СУБД для персональных компьютеров.

Реляционная модель данных

- Строки в каждой таблице – это кортеж неструктурированных единиц данных, «атрибутов».
- Набор кортежей, составляющий таблицу, образует математическое отношение;
- Модель данных представляется множеством таблиц–отношений (**называемых также R–таблицами**); отсюда название «реляционная», т.е. *модель, представленная отношениями*.
- **Атрибуты** строк–кортежей (и таблиц–отношений) – это значения из заданных наравне с таблицами областей определения («доменов»).

Реляционная модель данных

- **Реляционная база данных** – это набор R–таблиц и только R–таблиц, т.е. считается, что никаким иным образом (переменные, массивы и т.п.) данные в базе не представлены.
- В рамках реляционной теории имеется список операций, которые можно осуществлять над R–таблицами, причем так, что результатом снова будет R–таблица.

Обычно это следующие операции:

• **базовые операции:**

- *ограничение* – исключение из таблицы некоторых строк;
- *проекция* – исключение из таблицы некоторых столбцов;
- *декартово произведение* – из двух таблиц получается третья по принципу декартова произведения двух множеств строк;
- *объединение* – объединение множеств строк двух таблиц;
- *разность* – разность множеств строк двух таблиц;
- *присвоение* – именованной таблице присваивается значение выражения над R–таблицами;

• **производные операции:**

- *группа операций соединения*;
- *пересечение* – пересечение множеств строк двух таблиц;
- *деление* – позволяет отвечать на вопросы типа: «какие студенты посещают все курсы?»;
- *разбиение* – позволяет отвечать на вопросы типа: «какие пять служащих в отделе наиболее оплачиваемы?»;
- *расширение* – добавление новых столбцов в таблицу;
- *суммирование* – в новой таблице с меньшим, чем в исходной, числом строк, строки получены как агрегирование (например, суммирование по какому-то столбцу) строк исходной.