

Основные модели данных

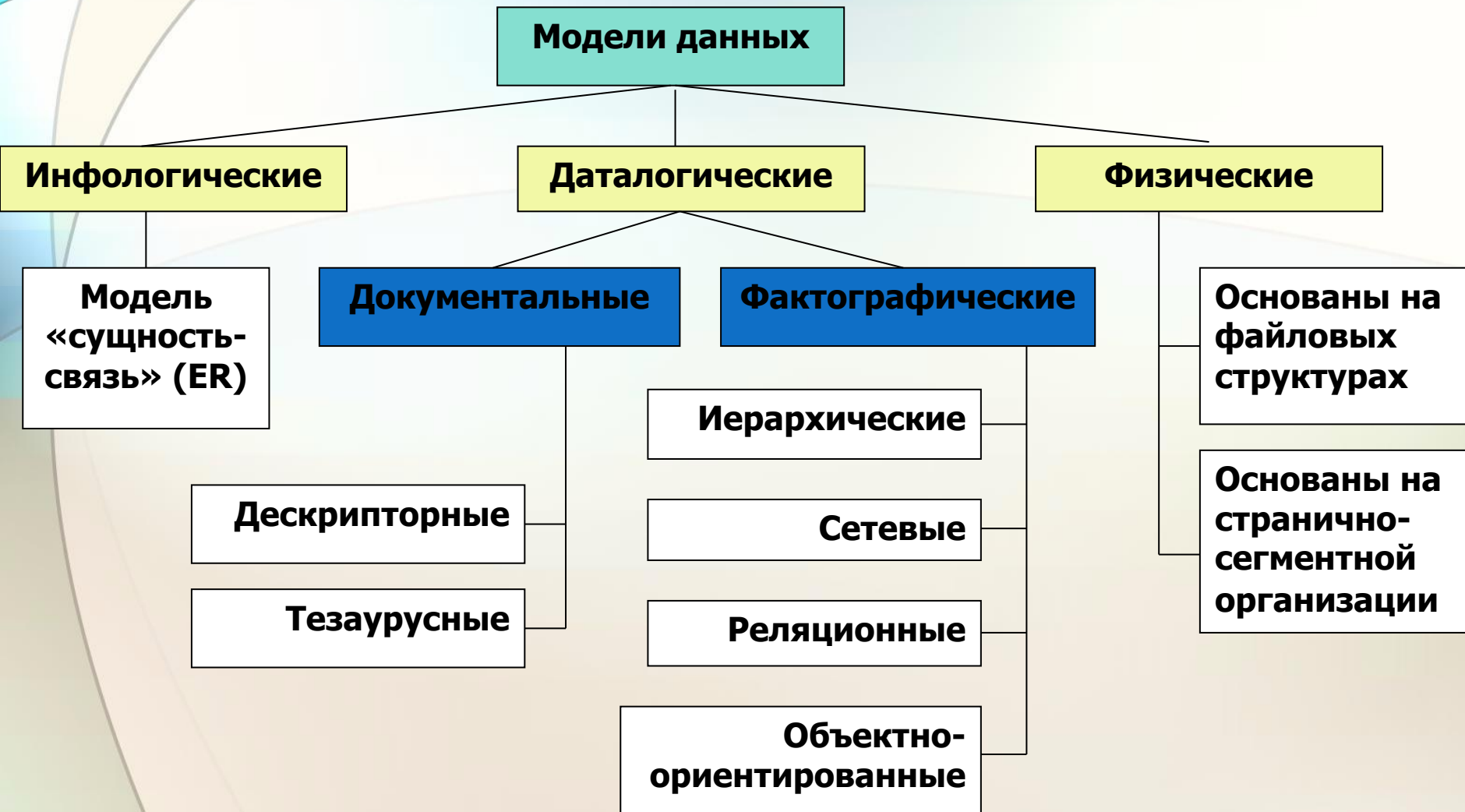
*Модель данных – это совокупность
структуры данных и операций
их обработки*

Классификация моделей данных

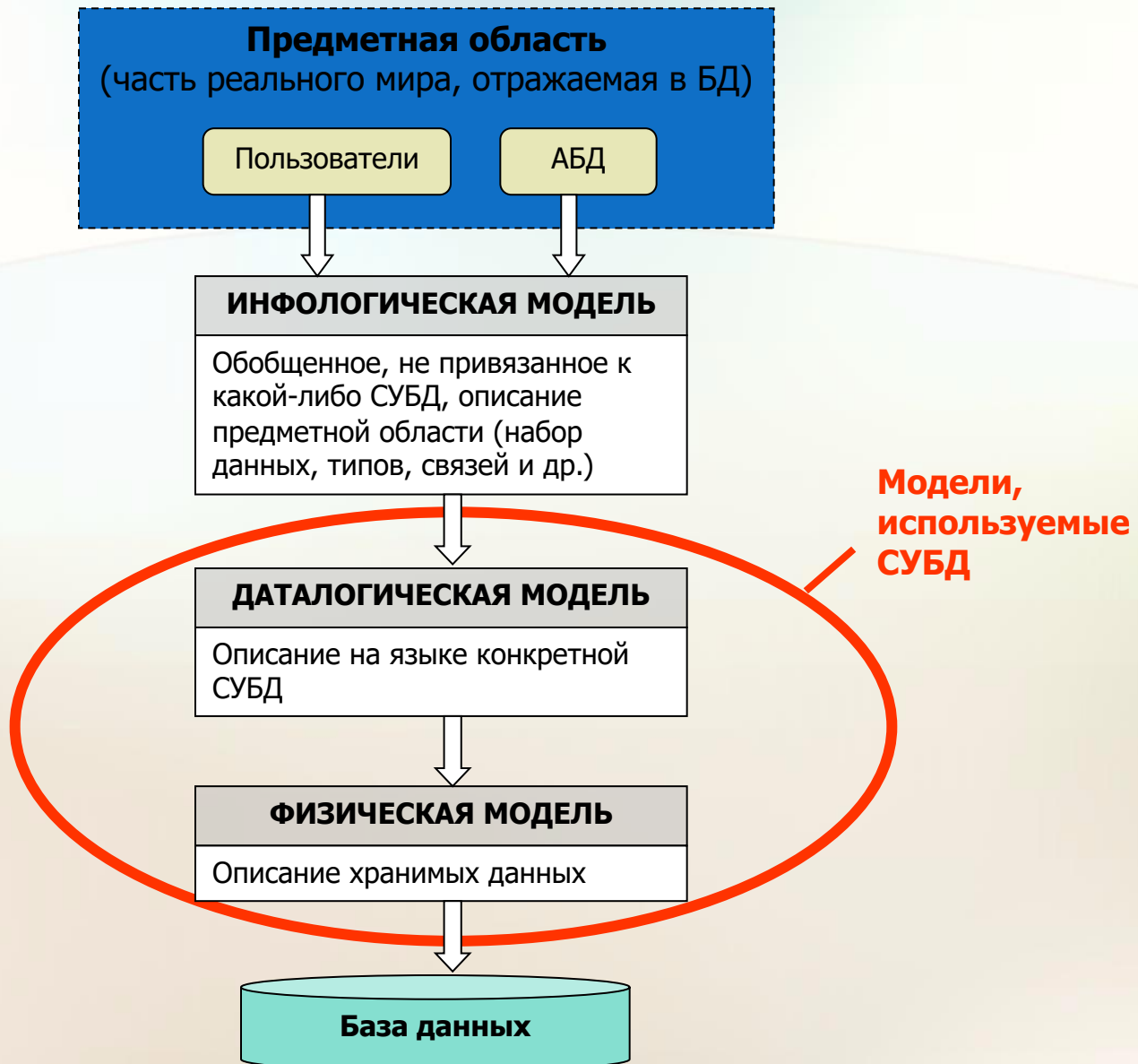
Модель данных

- это некоторая абстракция, которая, будучи применима 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



Уровни моделей данных (последовательность разработки БД)



Типы моделей БД

Иерархические

Сетевые

Реляционные

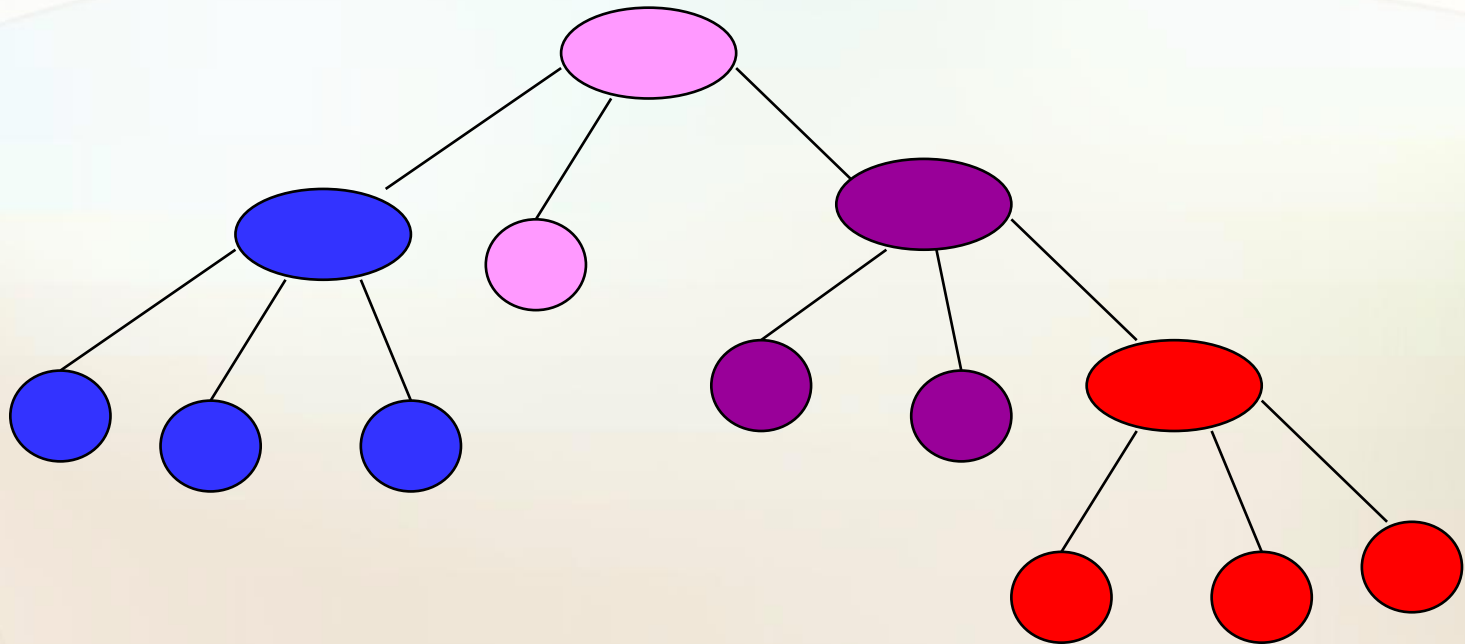
Иерархическая модель

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

- Объекты, связанные иерархическими отношениями, образуют ориентированный граф

Иерархическая модель

Существует строгая подчиненность элементов: один главный, остальные подчиненные. Например, система каталогов на диске.



Иерархическая модель

- Свойства иерархической модели:
 - каждый узел связан только с одним вышестоящим узлом, кроме вершины
 - иерархическая модель данных имеет только одну вершину, узел не подчинен более никаким узлам
 - от каждого узла существует единственный путь к вершине
 - связь не может быть установлена между объектами, находящимися через уровень
 - связь между узлами первого уровня не определяется

Иерархическая модель данных



Иерархическая модель

Преимущества:

- Простота понимания и использования, быстрота доступа к данным.
- Простота оценки операционных характеристик благодаря заранее заданным взаимосвязям и тому, что операции манипулирования данными являются навигационными, т.е. разработчик приложения явным образом описывает всю процедуру обработки.

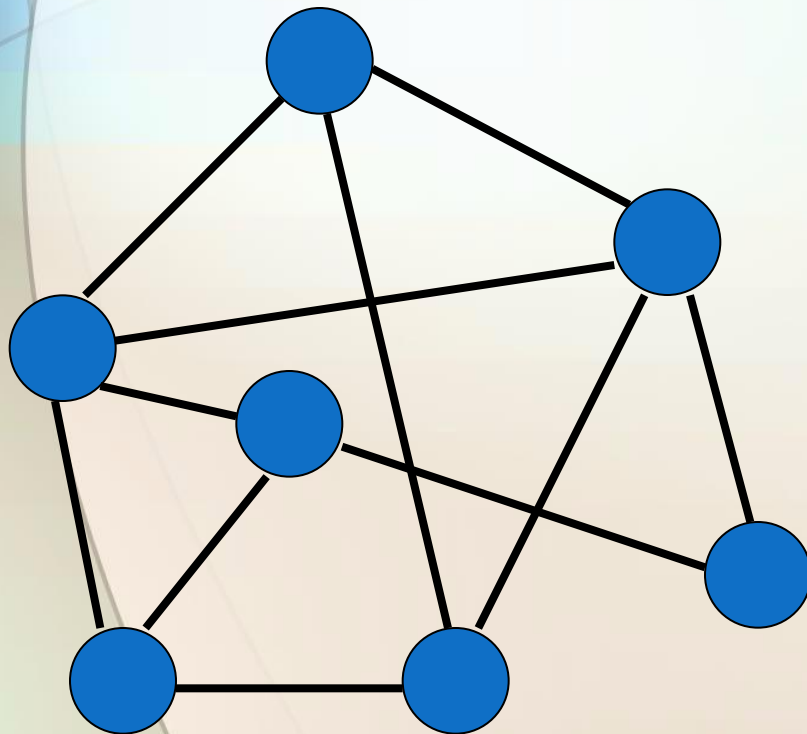
Иерархическая модель

Недостатки:

- Отсутствие универсальности
 - Не всякую информацию можно выразить в иерархической модели данных
- Исключительно навигационный принцип доступа к данным
- Доступ к данным только через корневой элемент
- Трудность реализации взаимосвязей «многие-ко-многим». При этом искусственно созданная структура иерархической модели становится громоздкой, может потребоваться хранение избыточных данных на физическом уровне.

Сетевая модель данных

Сетевые БД более гибкие: нет явно выраженного главного элемента и существует возможность установления горизонтальных связей. Например, организация информации в Интернете (W W W).



Сетевая модель
базы данных
представляет
совокупность объектов
различного уровня,
однако схема связей
между объектами
может быть любой.

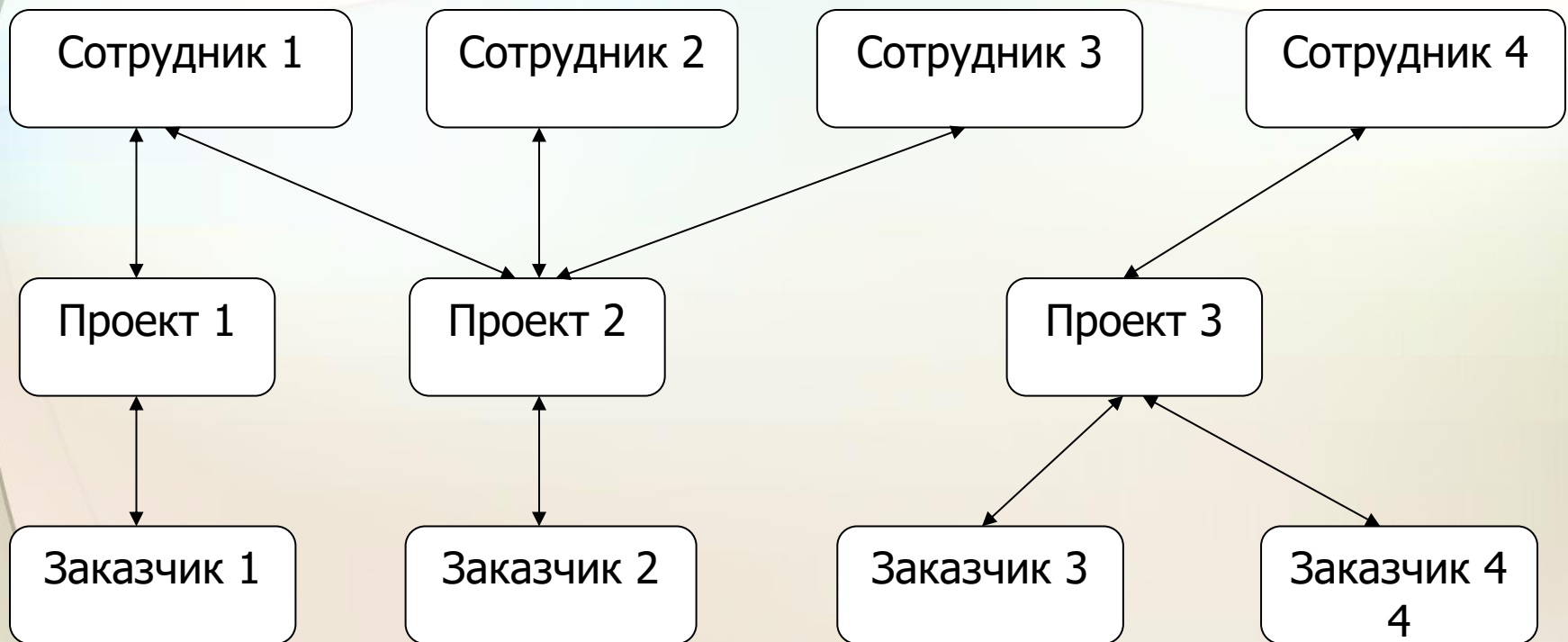
Сетевая модель

- Элементами этой модели являются: уровень, узел, связь
- Отличия в том, что элемент одного уровня может быть связан с любым числом элементов соседнего уровня
- Не существует подчиненности уровней друг другу

Свойства сетевой модели:

- Связь не может быть установлена между объектами, находящимися через уровень
- Связь между узлами первого уровня не определяется

Сетевая модель данных



Сетевая модель

Преимущества:

- универсальность
- возможность доступа к данным через значения нескольких отношений

Сетевая модель

Недостатки:

- сложность
 - обилие понятий, вариантов их взаимосвязей и способов реализации
- допустимость только навигационного принципа доступа к данным
- В частности, прикладной программист должен детально знать логическую структуру базы данных, поскольку ему необходимо осуществляя навигацию среди различных экземпляров наборов и экземпляров записей. Другими словами, программист должен представлять «свое» текущее положение в экземплярах наборов при «продвижении» по базе данных.

Реляционная модель данных

Теоретической основой этой модели стала теория отношений, основу которой заложили два логика — американец Чарльз Содерс Пирс (1839-1914) и немец Эрнст Шредер (1841-1902).

В руководствах по теории отношений было показано, что множество отношений замкнуто относительно некоторых специальных операций, то есть образует вместе с этими операциями абстрактную алгебру.

Это важнейшее свойство отношений было использовано в реляционной модели для разработки языка манипулирования данными, связанного с исходной алгеброй.

Реляционная модель данных

Реляционная модель (от английского "relation"-отношение) основана на отношениях между таблицами, которые содержат информацию.

Реляционные базы данных позволяют хранить информацию в **отношениях** (нескольких «плоских» (двухмерных) таблицах), связанных между собой посредством совместно используемых полей данных, называемых ключами. Реляционные базы данных предоставляют более простой доступ к оперативно составляемым отчетам (обычно через SQL) и обеспечивают повышенную надежность и целостность данных благодаря отсутствию избыточной информации.

Реляционная модель данных

Реляционная система – это система, основанная на следующих принципах:

- Данные пользователя представлены только в виде таблиц
- Элементы: *поле* (столбец), *запись* (строка) и *таблица* (отношение)
- Используется в реляционных системах
- пользователю предоставляются операторы, генерирующие новые таблицы из старых (для выборки данных)

Реляционная модель данных

Преимущества:

- простота
 - в такой модели всего одна информационная конструкция, формализующая табличное представление
- теоретическое обоснование
 - существуют четкие методы формализации данных в таблицах
- независимость данных
 - при изменении БД необходимы бывают лишь минимальные изменения прикладных программ

Достоинства и недостатки РМД

- Достоинства:
 - ✓ наличие теоретического базиса;
 - ✓ максимальная степень независимости данных от программ;
 - ✓ наличие декларативного языка запросов.
- Недостатки:
 - ✓ низкая эффективность выполнения запросов;
 - ✓ отсутствие однозначного соответствия между сущностями предметной области и таблицами реляционной базы данных.

Операции над отношениями

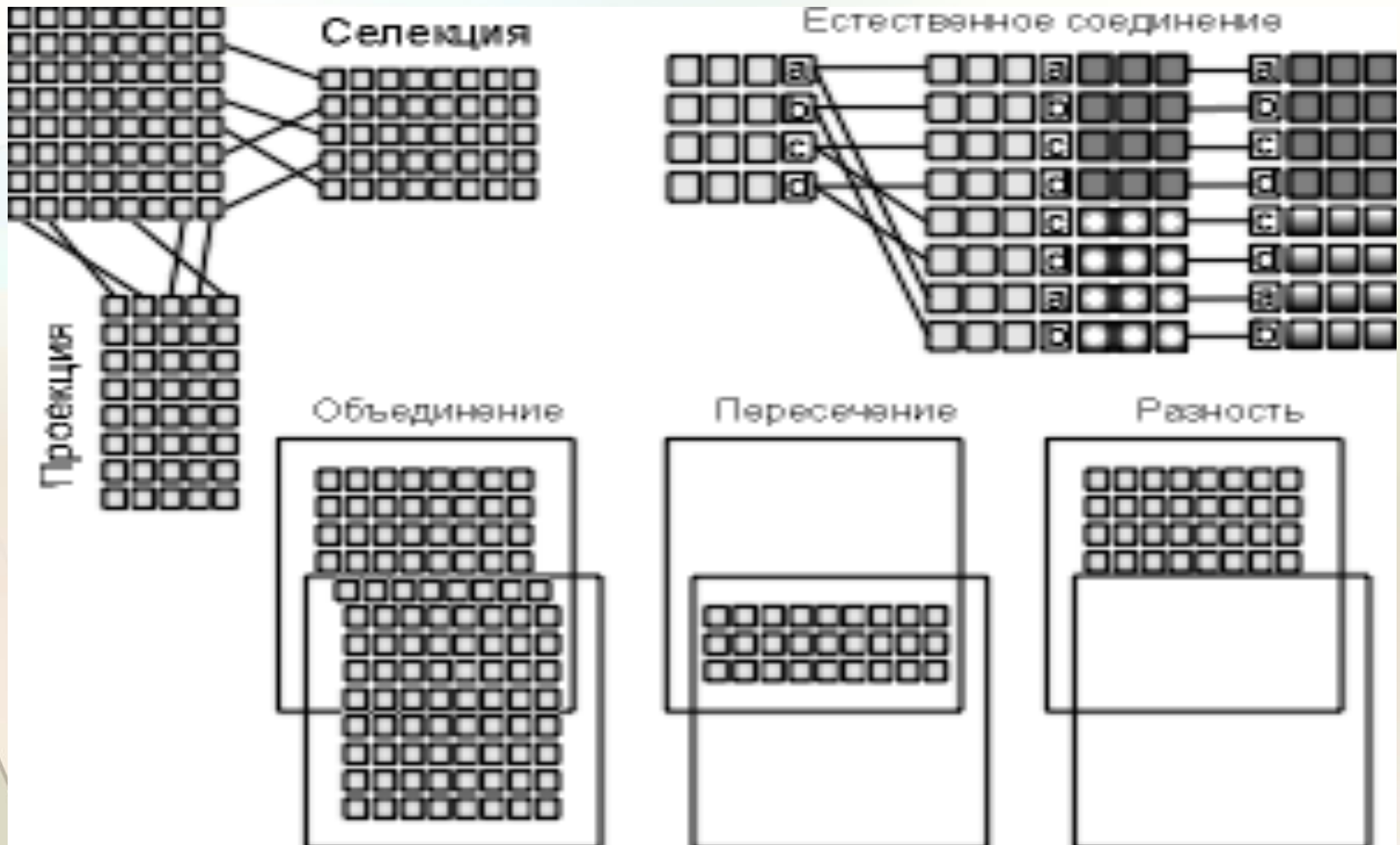
- **Реляционная алгебра** – Процедурный язык обработки реляционных таблиц.

Операции над множествами, к которым относятся операции: **объединения, пересечения, разности, деления и декартова произведения.**

- **Реляционное исчисление** – Непроцедурный язык создания запросов.

Специальные операции над отношениями, к которым относятся операции: **проекции, соединения, выбора.**

Каждая операция реляционной алгебры использует одну или несколько таблиц (отношений) в качестве ее операндов и продуцирует в результате новую таблицу, т.е. позволяет "разрезать" или "склеивать" таблицы



Пример декартова произведения

<i>Должность</i>
директор
инженер
экономист

<i>ФИО</i>
Белов С.Ю.
Рогов А.И.
Панина А.А.
Волкова Н.М.

<i>Оклад</i>
40000
75000

<i>ФИО</i>	<i>Должность</i>	<i>Оклад</i>
Белов С.Ю.	директор	40000
Белов С.Ю.	директор	75000
Белов С.Ю.	инженер	40000
Белов С.Ю.	инженер	75000
Белов С.Ю.	экономист	40000
Белов С.Ю.	экономист	75000
Рогов А.И.	директор	40000
Рогов А.И.	директор	75000
Рогов А.И.	инженер	40000
Рогов А.И.	инженер	75000
Рогов А.И.	экономист	40000
...
Волкова Н.М.	экономист	40000

Полужирным шрифтом выделены записи, имеющие соответствие в предметной области.

Элементы реляционной модели

Атрибут — характеристика сущности или объекта. У атрибута есть имя и тип данных.

Домен (domain) - компонент реляционной базы данных, предназначенный для формирования и описания множества допустимых значений атрибута.

Кортеж (tuple) - набор связанных (логически соединенных) полей (строка таблицы базы данных).

Кардинальное число, мощность (cardinality) — выражает значение связности компонентов. Задаёт число допустимых вхождений сущностей, связанных с единым местонахождением объединённой сущности (количество строк).

Мощность отношения - число кортежей в отношении (проще говоря, число записей).

Количество атрибутов в отношении называется **степенью, или рангом**, отношения. Можно говорить об ***n-арности отношения***.

Организация связей между таблицами

Связь один-ко-многим: Отделы – Сотрудники

Таблица «Сотрудники»

<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i><u>Отдел</u></i>
023	Волкова Елена Павловна	2
113	Белов Сергей Юрьевич	1
101	Рогов Сергей Михайлович	2
056	Панина Анна Алексеевна	1
...
098	Фролов Юрий Вадимович	9

Таблица «Отделы»

<i>Номер отдела</i>	<i>Название отдела</i>
1	Информационный отдел
2	Администрация
3	Отдел кадров
...	...
9	Проектный отдел

«Номер отдела» - первичный ключ в таблице «Отделы»

«Отдел» – внешний ключ в таблице «Сотрудники»

Организация связей между таблицами

Связь многие-ко-многим: Проекты – Сотрудники

Таблица «Сотрудники»

ФИО	Номер
Волкова Е.П.	023
Белов С.Ю.	113
Рогов С.М.	101
Панина А.А.	056
Фролов Ю.В.	098
...	...

Таблица «Участие»


<u>Участник</u>	Роль	<u>Проект</u>
113	исполнитель	23/Н
101	руководитель	18-К
056	исполнитель	18-К
101	консультант	09/Р
098	руководитель	23/Н
...

Таблица «Проекты»

Шифр	Название проекта
23/Н	АИС "Налог"-2
18-К	ИПС "Жители"
09/Р	ГИС "Город"
...	...

В таблице «Участие»:
«Участник» – внешний ключ к таблице «Сотрудники»
«Проект» – внешний ключ к таблице «Проекты»

Пример связи внутри таблицы



<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i>Должность</i>	<i>Оклад</i>	<i><u>Начальник</u></i>
023	Волкова Елена Павловна	секретарь	26000	101
113	Белов Сергей Юрьевич	инженер	39800	205
101	Рогов Сергей Михайлович	директор	62000	NULL
205	Махова Ольга Алексеевна	начальник отдела	51300	101
...

Свойства реляционной таблицы

1. Таблица представляет собой двумерную структуру, состоящую из строк и столбцов
2. Каждая строка таблицы (*кортеж*, tuple) представляет собой отдельную сущность внутри набора сущностей
3. Каждый столбец таблицы представляет собой атрибут, и у каждого столбца есть свое имя
4. На каждом пересечении строки и столбца имеется единственное значение
5. Каждая таблица должна иметь атрибут или несколько атрибутов, уникально идентифицирующих каждую строку
6. Все значения в столбце должны отображаться в одинаковом формате. Например, если атрибуту присваивается формат целого, то все значения в столбце, представляющем данный атрибут, должны быть целыми
7. Каждый столбец имеет определенный диапазон значений, называемый *доменом атрибута* (attribute domain)
8. Порядок следования строк и столбцов для СУБД не существен

Ограничения целостности

Целостность данных (целостность отношения) - это механизм поддержания соответствия базы данных предметной области, при котором выбранные данные всегда дают один и тот же результат. В реляционной модели данных определены два базовых требования обеспечения целостности:

- целостность ссылок
- целостность сущностей (категорная сущность).

Целостность на уровне сущности

В пределах таблицы первичный ключ должен быть уникальным, чтобы однозначно идентифицировать каждую строку. Когда дело обстоит так, то говорят, что таблица *проявляет целостность на уровне сущности (entity integrity)*. Для обеспечения целостности на уровне сущности в первичном ключе недопустимы пустые значения, *null* (т.е. полное отсутствие данных).

ЦЕЛОСТНОСТЬ НА УРОВНЕ СУЩНОСТИ

- **Требование** - Все элементы первичного ключа уникальны и никакая часть первичного ключа не может быть пустой (null)
- **Назначение** - Гарантирует, что каждая сущность (логический объект) будет иметь уникальную идентификацию, а значения внешнего ключа могут должным образом ссылаться на значения первичного ключа

Целостность на уровне ссылки

- Если внешний ключ содержит значения, совпадающие с первичным ключом или пустые значения (null), говорят, что таблица (или таблицы), использующая такой ключ проявляет **целостность на уровне ссылки (referential integrity)**. Другими словами **целостность на уровне ссылки** означает, что в том случае, если внешний ключ содержит некое значение, то это значение ссылается на существующий действительный кортеж (строку) в другом отношении.

ЦЕЛОСТНОСТЬ НА УРОВНЕ ССЫЛКИ

- **Требования** - Внешний ключ может иметь или пустое значение (если только он не является частью первичного ключа данной таблицы), или значение, совпадающее со значением первичного ключа в связанной таблице. (Каждое непустое значение внешнего ключа должно ссылаться на существующее значение первичного ключа.)
- **Назначение** - Допускается, что атрибут не имеет соответствующего значения, но атрибут не может принимать недопустимые значения. Выполнение правила целостности на уровне ссылки делает невозможным удаление строки в одной таблице, где первичный ключ имеет обязательное соответствие со значением внешнего ключа в другой таблице.

Работы с `select` в MySQL

Вывод данных таблицы

- `select * from <имя таблицы>;`
- `select * from <имя таблицы> limit <количество строк>;`
- `select * from <имя таблицы> limit <с какой записи>, <количество строк>;`

Определение списка полей результата

- `select список полей> from <имя таблицы>;`

Сортировка

- `select ... order by <поле [порядок] >;`

Фильтрация

- `select ... from ... where <логическое выражение>;`

Под логическим выражением может быть:

- `<поле> in <значения>` - проверка вхождения значения поля во множество значений
- `<поле> is null` - проверка на значение null
- `<поле> like <значение>` - проверка на соответствие строки маске
- `<поле> between <начало> and <конец>` - проверка значения на вхождение в диапазон