

# РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Реляционная модель данных была предложена сотрудником фирмы IBM Эдгаром Коддом. Математик по образованию, Кодд предложил использовать для обработки данных аппарат теории множеств (объединение, пересечение, разность, декартово произведение).

Он показал, что любое представление данных сводится к совокупности двумерных таблиц особого вида, известного в математике как отношение (relation), отсюда и название- реляционные базы данных.

**Реляционная модель позволила решить одну из важнейших задач в управлении базами данных - обеспечить независимость представления и описания данных от прикладных программ. Как следствие, упрощение проектирования и программирования баз данных.**

**Физическое размещение данных в реляционных базах на внешних носителях легко осуществляется с помощью обычных файлов.**

# Основные принципы

---

- 1. Связь между данными должна устанавливаться в соответствии с их внутренними логическими взаимоотношениями**
- 2. В реляционных СУБД одной командой могут обрабатываться целые файлы данных**

# **Достоинства реляционных СУБД**

---

- 1. Наличие небольшого набора абстракций, которые позволяют сравнительно просто моделировать большую часть распространенных предметных областей**
- 2. Наличие простого и мощного математического аппарата, опирающегося на теорию множеств и математическую логику**
- 3. Возможность манипулирования данными без необходимости знания конкретной физической организации баз данных во внешней памяти**

# Базовые понятия реляционной модели данных

## Основные понятия:

1. Отношение

2. Тип данных

3. Сущность

4. Домен

5. Атрибут

6. Кортеж

7. Ключ

# Базовые понятия реляционной модели данных

---

Отношение представляет собой двумерную таблицу, содержащую некоторые данные.

Понятие *тип данных* в реляционной модели данных полностью эквивалентно соответствующему понятию в алгоритмических языках.

**!** Набор поддерживаемых типов данных определяется СУБД и может сильно различаться в разных системах.

## Основные типы данных:

1. Целочисленные
2. Вещественные
3. Строковые
4. Специализированные типы данных для денежных величин
5. Специализированные типы данных для временных величин (Дата/Время)

# **Базовые понятия реляционной модели данных**

---

**Сущность – это обособленный объект или событие, информацию о котором необходимо сохранять в базе данных и который имеет определенный набор свойств-атрибутов.**

**Для сущностей различают тип и экземпляр.**

**Тип характеризуется именем и списком свойств, а экземпляр-конкретными значениями свойств.**



# Базовые понятия реляционной модели данных

---

**Атрибуты** представляют собой свойства, характеризующие сущность.

В структуре таблицы каждый атрибут именуется и ему соответствует заголовок некоторого столбца таблицы.

# Базовые понятия реляционной модели данных

---

Атрибуты бывают:

## 1. Идентифицирующие и описательные.

Идентифицирующие атрибуты имеют уникальное значение для сущностей данного типа и являются потенциальными ключами. Они позволяют однозначно распознавать экземпляры сущности.

Из потенциальных ключей выбирается один первичный ключ. В качестве первичного ключа обычно выбирается потенциальный ключ, по которому чаще происходит обращение к экземплярам записи. Первичный ключ должен включать в свой состав минимально необходимое для идентификации количество атрибутов.

Остальные атрибуты называются описательными.

# **Базовые понятия реляционной модели данных**

---

## **2. Простые и составные.**

Простой атрибут состоит из одного компонента, его значение неделимо.

Составной атрибут является комбинацией нескольких компонентов, возможно принадлежащих разным типам данных (например, адрес)

Для обеспечения высокой скорости работы с большими базами данных необходимо использовать простые атрибуты;

## **3. Однозначные и многозначные.**

Атрибуты могут иметь соответственно одно или много значений для каждого экземпляра сущности;

## **4. Основные и производные.**

Значение основного атрибута не зависит от других атрибутов.

Значение производного атрибута вычисляется на основе значений других атрибутов (например, возраст человека вычисляется на основе даты его рождения и текущей даты).

# Базовые понятия реляционной модели данных

---

**Домен** – множество всех возможных значений определенного атрибута отношения.

**Кортеж**, соответствующий данной схеме отношения, представляет собой множество пар {имя атрибута, значение}, которое содержит одно вхождение каждого имени атрибута.

Аргумент «значение» является допустимым значением домена данного атрибута.

**Пример:** {Специальность, Программист},  
{Курс, 3}

# Базовые понятия реляционной модели данных

атрибуты

№ студ. билета	ФИО	Дата рождения	Курс	Специальность
23982829	Алексеев Д.А.	12.03.1984	1	Металлург
23569875	Яковлев Н.В.	25.12.1983	2	Прокатчик
22565458	Михайлов В.В.	14.02.1983	2	Электрик
23696568	Афанасьев А.В.	03.10.1982	3	Металлург
23154786	Кузнецов В.И.	27.03.1984	1	Программист
22963236	Смирнов А.Д.	02.02.1984	1	Программист

К  
О  
Р  
Т  
Е  
Ж  
И

домены



Таблица - Отношение

# Базовые понятия реляционной модели данных

---

## Домены отношения «Студент»

- ❑ домены `Номера_студенческих_билетов` и `Номера_курсов` базируются на целочисленном типе. В число его значений могут входить только те целые числа, которые могут обозначать номер студенческого билета (положительное число) и номер курса (от 1 до 5)
- ❑ Домены `Фамилии_инициалы` и `Специальности` для столбцов `ФИО` и `Специальность` соответственно будут базироваться на строковом типе данных (не должны начинаться с `ъ` и `ь` знака)
- ❑ Домены `Даты_рождения` для столбца `Дата_рождения` определяется на базовом временном типе данных – содержит только допустимый диапазон дат рождения студентов

# Базовые понятия реляционной модели данных

---

**Степень отношения – это число атрибутов (количество полей таблицы)**

**Схема отношения - список имен атрибутов отношения с указанием имен доменов(или типов, если домены не поддерживаются).**

## Схема отношения «Студент»

**Студент {№\_студ.\_билета Номера\_студенческих\_билетов  
ФИО Фамилии\_инициалы  
Дата\_рождения Даты\_рождения  
Курс Номера\_курсов  
Специальность Специальности}**

# **Базовые понятия реляционной модели данных**

---

**В некоторых случаях какой-либо атрибут отношения может быть неприменим.**

**Пример. Студенту не присвоен еще номер студенческого билета.**

**Пустое значение – это не ноль и не пустая строка, это неизвестное значение атрибута, которое не определено в данный момент времени и может быть определено позднее.**



**NULL- обозначение пустых значений атрибутов**



# Базовые понятия реляционной модели данных

---

<b>Элемент реляционной модели</b>	<b>Форма представления</b>
<b>Отношение</b>	<b>Таблица</b>
<b>Схема отношения</b>	<b>Строка заголовков столбцов таблицы (заголовок таблицы)</b>
<b>Кортеж</b>	<b>Строка таблицы</b>
<b>Сущность</b>	<b>Описание свойств объекта</b>
<b>Атрибут</b>	<b>Заголовок столбца таблицы</b>
<b>Домен</b>	<b>Множество допустимых значений атрибута</b>
<b>Значение атрибута</b>	<b>Значение поля в записи</b>
<b>Первичный ключ</b>	<b>Один или несколько атрибутов</b>
<b>Тип данных</b>	<b>Тип значений элементов таблицы</b>

# Ключи отношения

---

Первичный ключ (Primary key) - некоторый атрибут или набор атрибутов, однозначно определяющий каждый кортеж отношения и обеспечивающий уникальность строк таблицы

Первичный ключ по определению уникален, в отношении не может быть двух разных кортежей с одинаковыми значениями первичного ключа




По правилу целостности первичный ключ не может быть полностью или частично пустым, т.е. иметь значение NULL



Для каждого отношения первичный ключ может быть только один

# Ключи отношения

Клиенты	
Имя поля	Тип данных
 Код клиента	Числовой
Название клиента	Текстовый
Контактное лицо	Текстовый
Признак группы	Логический
Телефон	Текстовый
Адрес	Текстовый

Клиенты						
Код клиент	Название клиента	Контактное лицо	Признак группы	Телефон	Адрес	
1000	Дарвиш Ф.К.	Дарвиш Ф.К.	<input checked="" type="checkbox"/>	(123)543-31-35	Ленина 58 кв. 12	
1001	Мефрилов А.С.	Мефрилов А.С.	<input checked="" type="checkbox"/>	(111)111-11-11	Карла Маркса 12 кв. 17	
1010	Харьков К.Е.	Харьков К.Е.	<input type="checkbox"/>	(516)434-35-43	Янгельский район Пролетарская 11	
1011	ПАО ММК	Джаноб А.С.	<input checked="" type="checkbox"/>	(134)864-38-43	Советский перекрёсток 19 кв. 12	
1100	ООО Хлебпром	Ходжи Д.В.	<input checked="" type="checkbox"/>	(483)438-73-84	Вокзальная 22 кв. 16	
1101	ОАО Ситно	Хон Ж.Д.	<input checked="" type="checkbox"/>	(843)483-48-73	Московская 12 кв. 17	
1110	Мирзо В.А.	Мирзо В.А.	<input checked="" type="checkbox"/>	(448)742-34-87	Московская 26 кв. 56	
1111	Мулло А.Д.	Мулло А.Д.	<input checked="" type="checkbox"/>	(466)431-67-46	9 мая 16 кв. 29	
6666	Устоз В.А.	Устоз В.А.	<input type="checkbox"/>	(464)643-13-83	Владиславская 56 кв 10	
9895	Машхади А.К.	Машхади А.К.	<input checked="" type="checkbox"/>	(468)468-46-46	Пионерская 23 кв. 21	
*			<input type="checkbox"/>			

# Первичный ключ отношения

---

## По количеству атрибутов:

1. *Простой ключ* – ключ, содержащий только один атрибут
2. *Составной ключ* – ключ, состоящий из нескольких атрибутов

## По содержанию информации:

1. *Искусственный* – ключ, созданный самой СУБД или пользователем с помощью некоторой процедуры, который сам по себе не содержит информации. Система поддерживает искусственный ключ, но он никогда не показывается пользователю
2. *Естественный ключ* – ключ, в котором используются значимые атрибуты, т.е. содержит информацию.



В качестве ключа используется целочисленный тип.

# Ключи отношения

---

**Внешний ключ (Foreign key)** – это атрибут (или множество атрибутов) одного отношения, являющийся ключом другого отношения.

Внешний ключ отношения является ссылкой на первичный ключ другого отношения.

По правилу ссылочной целостности внешний ключ может быть либо пустым, либо соответствовать значению первичного ключа, на который он ссылается.

За счет внешних ключей реализуются связи между отношениями базы данных.

# Ключи отношения

