
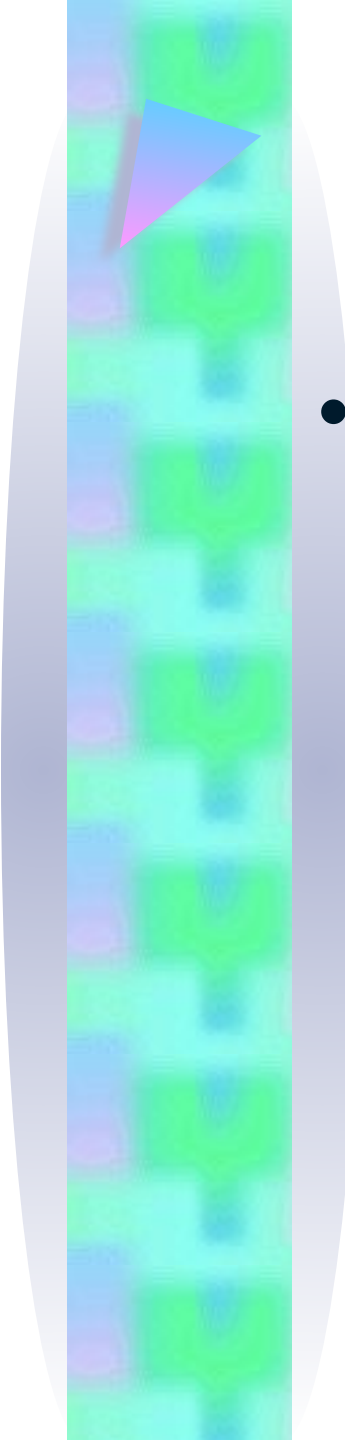
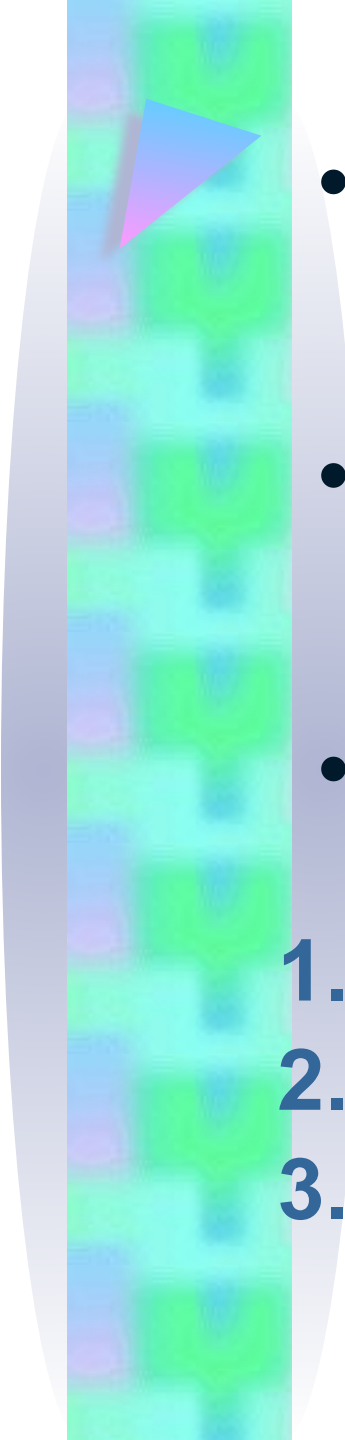


ОСНОВЫ РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ

- Обобщенные структуры называют моделями данных, т.к. они отражают представление пользователя о данных реального мира.

- 
- Любая модель данных содержит три компоненты:
 - структура данных - описывает точку зрения пользователя на представление данных
 - 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.
 - ограничения целостности - механизм поддержания соответствия данных предметной области на основе формально описанных правил.

- 
- **В процессе исторического развития в СУБД использовались следующие модели данных: реляционная, иерархическая, сетевая.**

- 
- **Основы реляционной модели данных были впервые изложены в Е.Коддом в 1970 г.**
 - **Наиболее распространенная трактовка реляционной модели данных принадлежит К.Дейту.**
 - **Реляционная модель состоит из трех частей:**
 1. **структурной части;**
 2. **целостной части;**
 3. **манипуляционной части.**

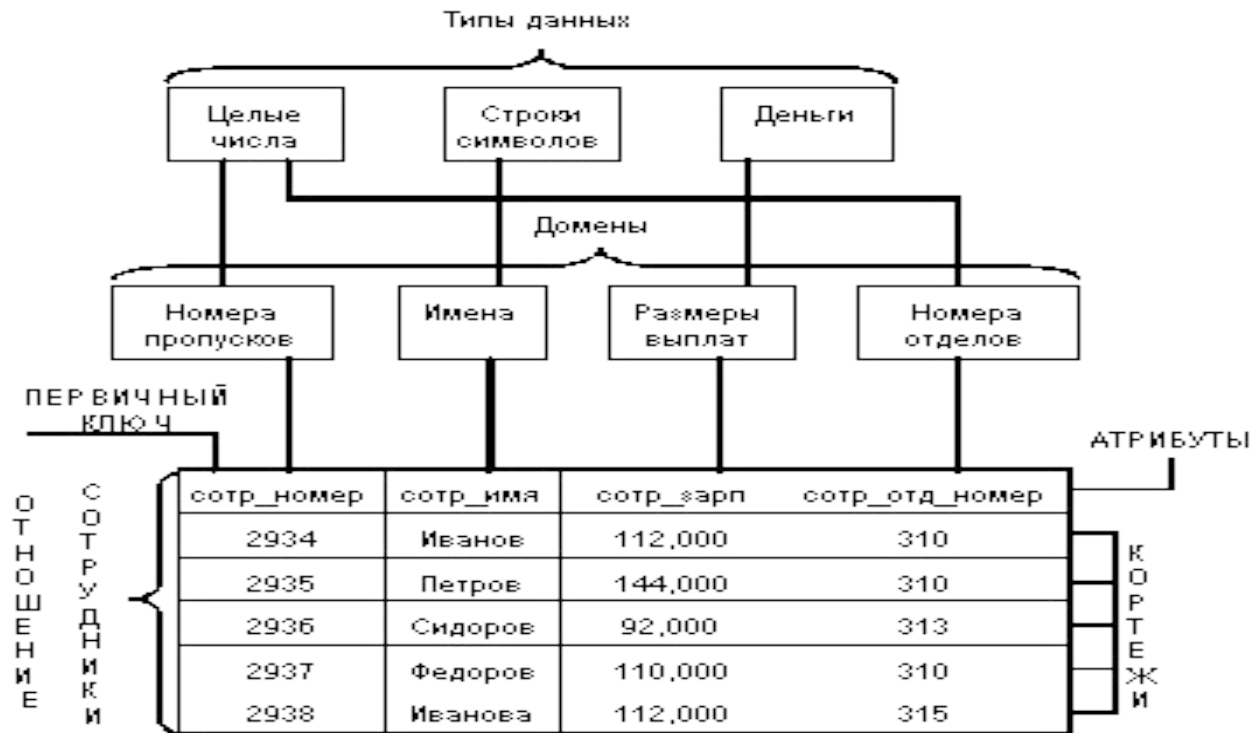
- **Структурная часть** описывает, какие объекты рассматриваются реляционной моделью. Постулируется, что единственной структурой данных, используемой в реляционной модели, являются нормализованные n-арные отношения.
- **Целостная часть** описывает ограничения специального вида, которые должны выполняться для любых отношений в любых реляционных базах данных. Это *целостность сущностей* и *целостность внешних ключей (по ссылкам)*.
- **Манипуляционная часть** описывает два эквивалентных способа манипулирования реляционными данными - *реляционную алгебру* и *реляционное исчисление*.

Структурная часть модели реляционной БД

Основные понятия реляционных баз данных

- тип данных
- домен
- атрибут
- кортеж
- ключ
- отношение
- схема отношения

Отношение «Сотрудники»



Реляциялық мәліметтер қоймасының негізгі түсініктері



Атрибуттар

Кортеждер

Сур.3 — «Қызметкерлер» қатынасы

Тип данных

- Понятие *тип данных* в реляционной модели данных полностью адекватно понятию типа данных в языках программирования.
- В современных реляционных БД Структурная часть модели реляционной БД допускается хранение символьных, числовых данных, битовых строк, специализированных числовых данных (таких как "деньги"), а также специальных "темпоральных" данных (дата, время, временной интервал).

Домен

- **Домен определяется заданием некоторого базового типа данных, к которому относятся элементы домена, и произвольного логического выражения, применяемого к элементу типа данных. Если вычисление этого логического выражения дает результат "истина", то элемент данных является элементом домена.**

Домен

- **Домен** - потенциальное множество значений данного типа. Например, домен "Имена" определен на базовом типе строк символов, но в число его значений могут входить только те строки, которые могут изображать имя (в частности, такие строки не могут начинаться с мягкого знака).
- **Семантическая нагрузка понятия домена:** данные считаются сравнимыми только в том случае, когда они относятся к одному домену. В нашем примере значения доменов "Номера пропусков" и "Номера групп" относятся к типу целых чисел, но не являются сравнимыми.

Свойства домена

Домен имеет уникальное имя (в пределах базы данных).

Домен определен на некотором простом типе данных или на другом домене.

Домен может иметь некоторое логическое условие, позволяющее описать подмножество данных, допустимых для данного домена.

Домен несет определенную смысловую нагрузку.

Пример: домен D , имеющий смысл "возраст сотрудника" можно описать как следующее подмножество множества натуральных чисел:

$$D = \{x \in \mathbb{N} : x \geq 18 \text{ and } x \leq 60\}$$

Домен

- Если **тип данных** - множеством всех возможных значений данного типа, то **домен** - подмножество в этом множестве.
- Отличие домена от понятия подмножества: домен отражает семантику, определенную предметной областью.
- Может быть несколько доменов, совпадающих как подмножества, но несущие различный СМЫСЛ.

Домен

- **Пример: домены "Вес детали" и "Имеющееся количество" можно одинаково описать как множество неотрицательных целых чисел, но смысл этих доменов будет различным, и это будут различные домены.**
- **Основное значение доменов в том, что домены ограничивают сравнения. Некорректно сравнивать значения из различных доменов, даже если они имеют одинаковый тип.**

Кортеж

- **Кортеж, соответствующий данной схеме отношения, - это множество пар {имя атрибута, значение}, которое содержит одно вхождение каждого имени атрибута, принадлежащего схеме отношения.**
- **"Значение" является допустимым значением домена данного атрибута.**
- **Степень или "арность" кортежа, т.е. число элементов в нем, совпадает с "арностью" соответствующей схемы отношения. То есть, кортеж - это набор именованных значений заданного типа.**

Кортеж, отношение

- **Отношение** - это множество кортежей, соответствующих одной схеме отношения. Схему отношения называют заголовком отношения, а отношение как набор кортежей - телом отношения.
- Житейским представлением отношения является таблица, заголовком которой является схема отношения, а строками - кортежи отношения-экземпляра; в этом случае имена атрибутов именуют столбцы этой таблицы.
- **Реляционная база данных** - это набор отношений, имена которых совпадают с именами схем отношений в схеме БД.

Схема отношения, схема базы данных

- **Схема отношения** - это именованное множество пар {имя атрибута, имя домена}. Степень или "арность" схемы отношения определяет мощность этого множества.
- Степень отношения СОТРУДНИКИ равна четырем, то есть оно является 4-арным.
- **Схема БД** (в структурном смысле) - это набор именованных схем отношений.

Термины в реляционной модели данных имеют соответствующие "табличные" синонимы

Реляционный термин	Соответствующий "табличный" термин
База данных	Набор таблиц
Схема базы данных	Набор заголовков таблиц
Отношение	Таблица
Заголовок отношения	Заголовок таблицы
Тело отношения	Тело таблицы
Атрибут отношения	Наименование столбца таблицы
Кортеж отношения	Строка таблицы
Степень (-арность) отношения	Количество столбцов таблицы
Мощность отношения	Количество строк таблицы
Домены и типы данных	Типы данные в ячейках таблицы

Фундаментальные свойства отношений

- **Отсутствие кортежей-дубликатов. В отношении нет одинаковых кортежей.**
- **Отсутствие упорядоченности кортежей. Кортежи не упорядочены (сверху вниз).**
- **Отсутствие упорядоченности атрибутов. Атрибуты отношений не упорядочены (слева направо).**
- **Атомарность значений атрибутов. Все значения атрибутов атомарны.**

Целостная часть модели реляционной БД

- 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. Сложные сущности реального мира представляются в реляционной БД в виде нескольких кортежей нескольких отношений.

Пример: требуется представить в реляционной БД:

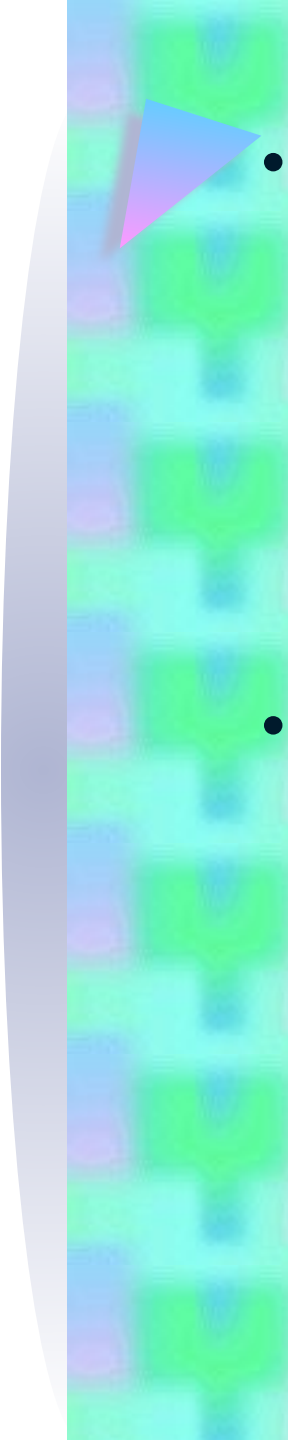
1. сущность **ОТДЕЛ** с атрибутами ОТД_НОМЕР (номер отдела), ОТД_КОЛ (количество сотрудников), ОТД_СОТР (набор сотрудников отдела).
2. сущность **ОТД_СОТР** с атрибутами СОТР_НОМЕР (номер сотрудника), СОТР_ИМЯ (имя сотрудника), СОТР_ЗАРП (заработная плата сотрудника).


При проектировании БД имеем два отношения:


1. ОТДЕЛЫ (ОТД_НОМЕР, ОТД_КОЛ)
2. СОТРУДНИКИ (СОТР_НОМЕР, СОТР_ИМЯ, СОТР_ЗАРП, СОТР_ОТД_НОМ)

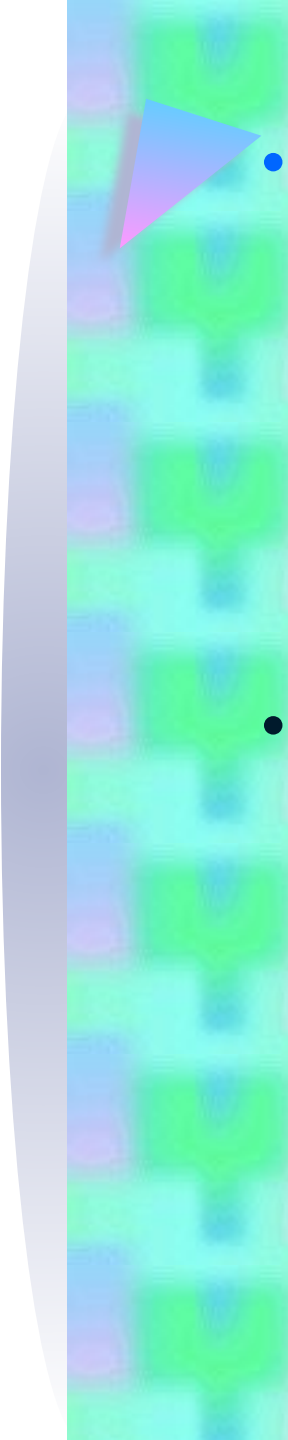
Пример ненормализованного отношения «Отделы»


НОМЕР_ОТДЕЛА	ОТДЕЛ		
	СОТР_НОМЕР	СОТР_ИМЯ	СОТР_ЗАРП
310	2934	Иванов	112,000
	2935	Петров	112,500
313	2937	Федоров	110,000
315	2938	Иванова	112,000

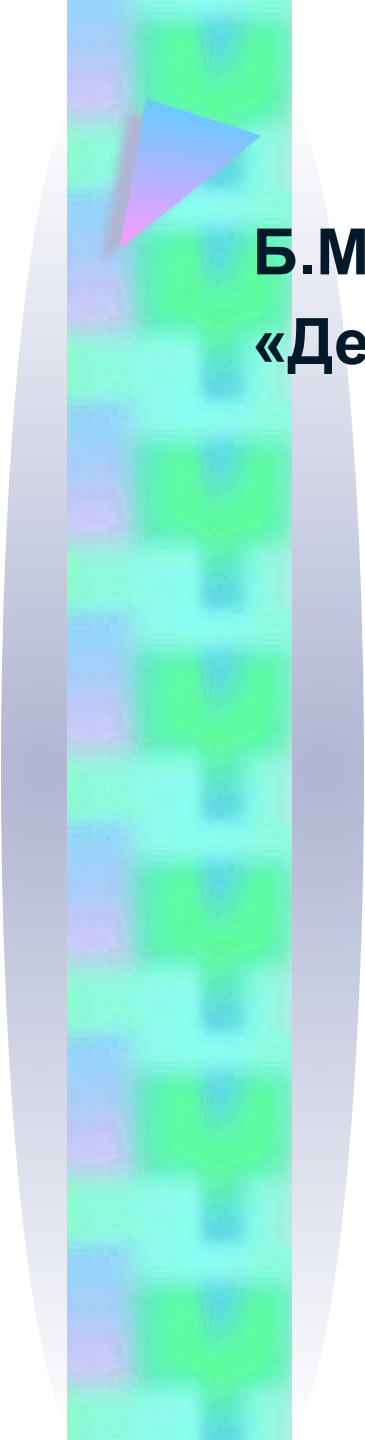
- 
- Атрибут **СОТР_ОТД_НОМ** появляется в отношении **СОТРУДНИКИ** не потому, что номер отдела является собственным свойством сотрудника, а для того, чтобы иметь возможность восстановить при необходимости полную сущность **ОТДЕЛ**.
 - Значение атрибута **СОТР_ОТД_НОМ** в любом кортеже отношения **СОТРУДНИКИ** должно соответствовать значению атрибута **ОТД_НОМ** в некотором кортеже отношения **ОТДЕЛЫ**.

- 
- Атрибут такого рода называется **внешним ключом**, поскольку его значения однозначно характеризуют сущности, представленные кортежами некоторого другого отношения.
 - Отношение, в котором определен внешний ключ, ссылается на соответствующее отношение, в котором такой же атрибут является первичным ключом.

- 
- **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.**

- 
- **При обновлении ссылающегося отношения (вставке новых кортежей или модификации значения внешнего ключа в существующих кортежах) достаточно следить за тем, чтобы не появлялись некорректные значения внешнего ключа.**
 - **Но как быть при удалении кортежа из отношения, на которое ведет ссылка???**

- 
- Существуют 3 подхода, каждый из которых поддерживает целостность 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.



**Б.М. Саданова, Г.О. Құдышева, О.А. Кан
«Деректер қоймасын ұйымдастыру»**