

## СУБД

Специализированные программные продукты  
(СУБД)

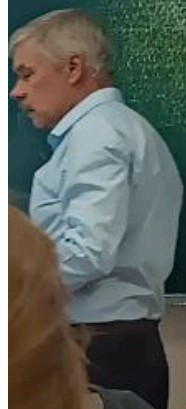
(Database Management systems –DBMS)

- Средства постоянного хранения данных
- Возможность совместного доступа к данным
- Сокращение избыточности данных
- Обеспечение целостности данных
- Интерфейс программирования
- Управление транзакциями
- Защита данных



## Архитектура СУБД

В соответствии с рекомендациями ANSI-SPARC (*American National Standards Institute, Standards Planning And Requirements Committee*), СУБД имеют в своём составе 3 уровня:





- логический уровень содержит абстракции, в которых будут описываться данные, хранимые в СУБД. На этом же уровне находятся соглашения о способе работы с абстракциями логического уровня: приведения их к необходимой структуре, а также чтению, добавлению, изменению и удалению данных;



- прикладной уровень содержит абстракции логического уровня, приведённые к структуре, достаточной для решения прикладной задачи;



- физический уровень содержит алгоритмы и структуры данных, обслуживающие логический уровень. Теоретически, устройство физического уровня скрыто от нас.



## Классификации СУБД

По модели данных

- иерархические
- реляционные
- сетевые и т.д.

По степени распределённости

- локальные СУБД
- распределённые СУБД

По способу доступа к БД

- Файл-серверные
- Клиент-серверные
- Встраиваемые



## ИСТОРИЯ

**1960** – первые коммерческие СУБД оперировали файловыми системами

**1970** - Codd E.F. Data Models in Databases Management.

Реляционные базы данных

**1975** — основана компания Microsoft.



1977 — основана компания Software Development Laboratories (ныне — Oracle Corporation).

1984 - Дэйв и Билл Фултон Фергюсон (Fox Software) был разработан FoxBase, который бежал со скоростью лисы.



База данных является *самодокументированной (self-describing)*: она содержит, в дополнение к исходным данным пользователя, описание собственной структуры. Это описание называется *словарем данных (data dictionary)*, *каталогом данных (data directory)* или *метаданными (metadata)*.





Самодокументированность обуславливает независимость программ от данных. Иначе говоря, она дает возможность определить структуру и содержимое базы данных путем обращения к самой базе данных.



## Проектирование БД

1. В базе данных должны иметься данные, достаточные для получения ответов на все необходимые вопросы.
2. В базе данных не должны возникать противоречия.
3. Структура базы данных не должна часто меняться.



Любой проект базы данных вписывается в некоторую модель.

**Модель сущность-связь (Entity-Relationship, ER)** – сущности, атрибуты сущностей, связи между сущностями.

**Реляционная модель** – таблицы, поля, записи.

Для каждой модели известны best practices – приёмы, зарекомендовавшие себя и желательные к применению.

Сначала рассмотрим ER-модель, а потом - реляционную.



# Модель «сущность-связь»

Entity-relationship model  
(Peter Chen, 1976)



## Сущности

Сущность – это некоторый объект, за которым пользователь хотел бы наблюдать.

Сущности одного и того же типа группируются в классы сущностей.



## Пример

### СТУДЕНТ

№ студенческого билета
Фамилия
Имя
Город
Улица
Дом
Телефон



## Атрибуты

У сущностей есть *атрибуты* (attributes), или, как их иногда называют, *свойства* (properties), которые описывают характеристики сущности. В модели «сущность—связь» предполагается, что все экземпляры данного класса сущностей имеют одинаковые атрибуты.



## Атрибуты

Исходное определение модели «сущность—связь» включает в себя композитные атрибуты (composite attributes) и многозначные атрибуты (multi-valued attributes). В качестве примера композитного атрибута можно привести атрибут Адрес, состоящий из группы атрибутов {Улица, Город, Индекс}.



## Идентификаторы

Экземпляры сущностей имеют *идентификаторы* (identifiers) — атрибуты, с помощью которых эти экземпляры именуется, или идентифицируются.



## Идентификаторы

Идентификатор экземпляра сущности состоит из одного или более атрибутов сущности. Идентификатор может быть *уникальным* (unique) либо *неуникальным* (nonunique).



## Связи

Взаимоотношения сущностей выражаются *связями* (relationships). Модель «сущность—связь» включает в себя классы связей и экземпляры связей.



## Связи

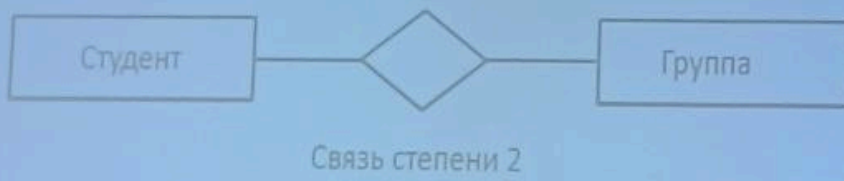
*Классы связей* (relationship classes) — это взаимоотношения между классами сущностей, а *экземпляры связи* (relationship instances) — взаимоотношения между экземплярами сущностей. У связей могут быть атрибуты.



## Связи

Класс связей может затрагивать несколько классов сущностей. Число классов сущностей, участвующих в связи, называется *степенью связи (relationship degree)*.







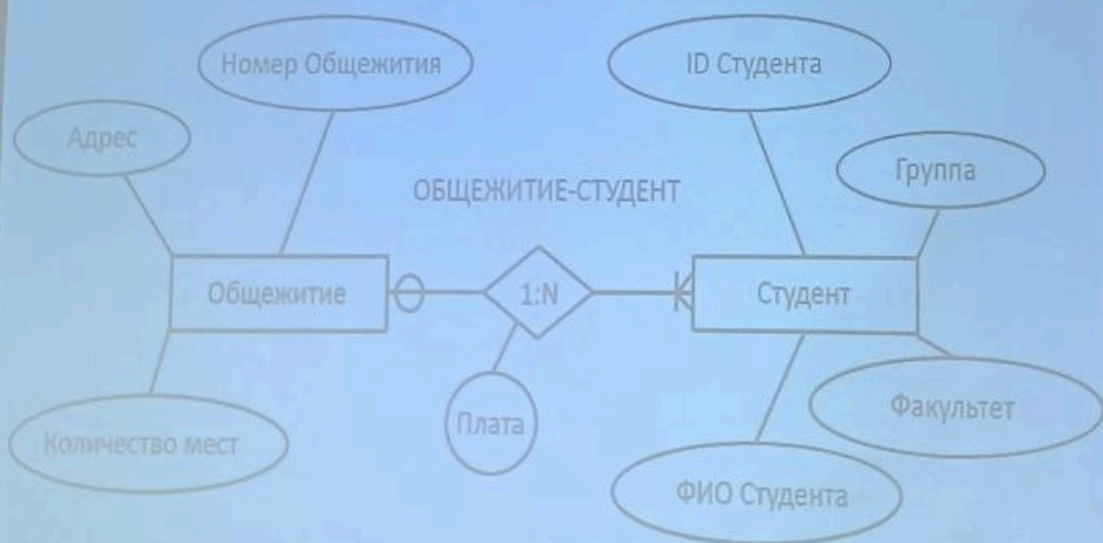




Числа внутри ромба, символизирующего связь, обозначают максимальное количество сущностей на каждой стороне связи. Эти ограничения называются *максимальными кардинальными числами*, а совокупность из двух таких ограничений для обеих сторон связи называется *максимальной кардинальностью (maximum cardinality)* связи.



## Диаграммы «сущность—связь»





## Слабые сущности

В модели «сущность—связь» определен особый тип сущности, называемый *слабой сущностью* (weak entity). К слабым сущностям относятся такие сущности, которые могут существовать в базе данных только в том случае, если в ней присутствует сущность некоторого другого типа. Сущность, не являющаяся слабой, называется *сильной сущностью* (strong entity).





Идентификационно-зависимая сущность



Идентификатор:  
№ дома

Идентификатор:  
№ дома+№ квартиры



## Представление многозначных атрибутов при помощи слабых сущностей

Многозначные атрибуты представляются в модели «сущность—связь» путем создания новой слабой сущности и построения связи вида «один ко многим».





## Подтипы сущностей

Некоторые сущности имеют необязательные наборы атрибутов; эти сущности часто представляются с помощью *подтипов* (subtypes) сущностей.

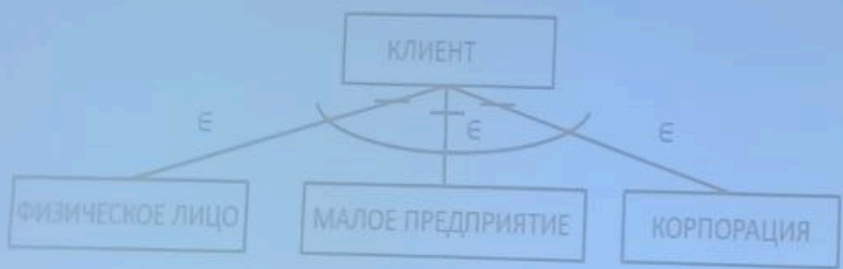


## Подтипы сущностей

### КЛИЕНТ

ID клиента  
Имя клиента  
Сумма к оплате  
Адрес  
№ пенсионной страховки  
Имя директора  
ИНН  
Телефон  
Доверенное лицо





**КЛИЕНТ**  
ID клиента  
Имя клиента  
Сумма к оплате

**МАЛОЕ ПРЕДПРИЯТИЕ**  
Адрес  
Имя директора  
ИНН

**ФИЗИЧЕСКОЕ ЛИЦО**  
Адрес  
№ пенсионной страховки

**КОРПОРАЦИЯ**  
Доверенное лицо  
ИНН  
Телефон



# Сущности

- обычные
- слабые



## Атрибуты

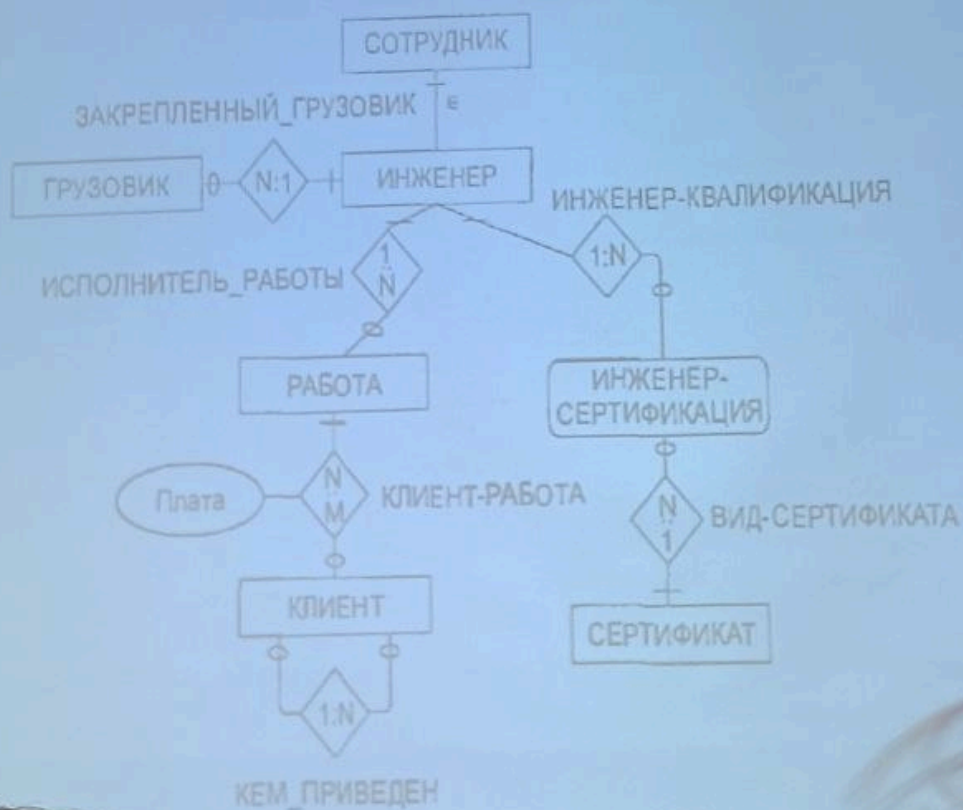
- простые
- ключевые
- составные
- однозначные
- многозначные
- производные



## Связи

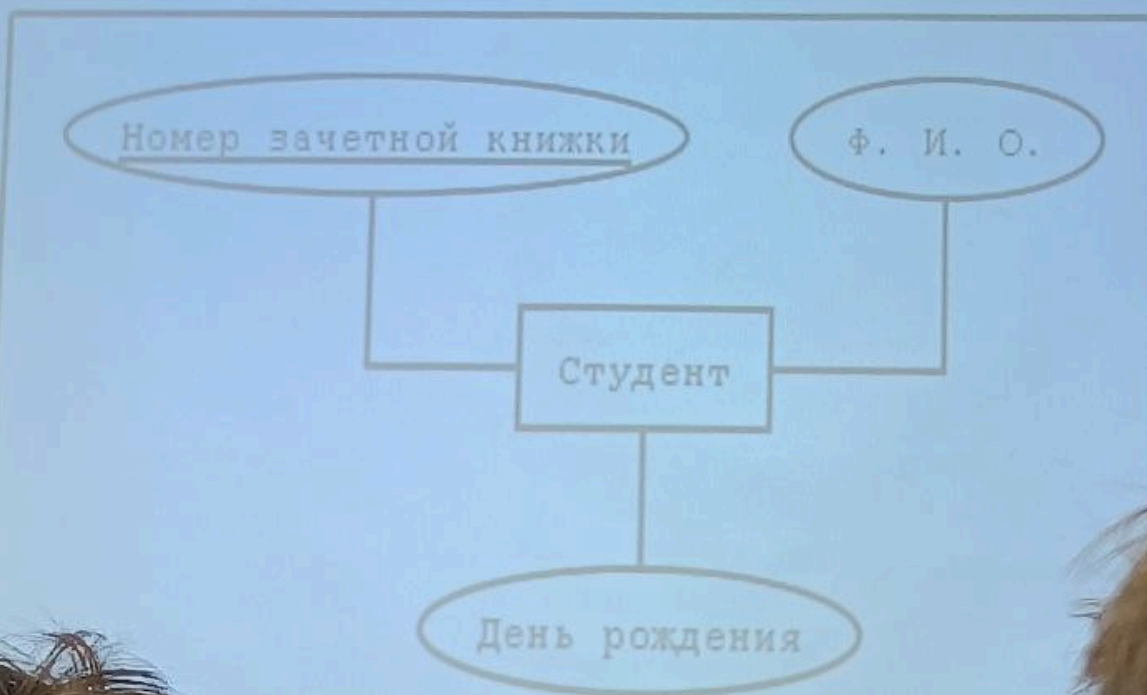
- один к одному
- один ко многим
- многие ко многим







# Сущности и атрибуты





## Связи между сущностями

