

Базы данных

Лекция №1

Виноградова М.В.

МГТУ им. Н.Э. Баумана (ИУ5)

КМ и баллы

- Семестр = 100/60
- ЛР + РК + КР (макет)
- Оценка
 - 60 = 3 (удв)
 - 71 = 4 (хор)
 - 85 = 5 (отл)

Зеленый коридор

- Все КМ в срок (ЛР, ДЗ, РК)
- Посещение всех лекций

База данных

- Структурированные данные
- Длительное (постоянное) хранение
- Под управлением СУБД

СУБД

- Язык DDL (Data Definition Language) – определяет новые базы данных(БД) и их схемы.
- Язык DML (Data Manipulation Language) – задает запросы.
- Длительное хранение больших объемов структурированных данных.
- Многопользовательская среда.

Поколения СУБД

- Первые БД появились в конце 60-х годов:
 - Бронирование авиабилетов; Банковская система; Корпоративные системы.
 - Файловая система + модель описания структуры
 - «иерархическая» (дерево); «сетевая» (граф)
- Реляционные БД появляются с 70-х годов:
 - Язык SQL (Structured Query Language)
 - Отношения (таблицы)
 - Реляционная алгебра; оптимизация запросов
 - Основные до 90-х

Постреляционные БД

- Поколение 90-х – 2000-х
 - Объектные,
 - Объектно-реляционные,
 - Полуструктурированные (XML).
- NoSql (с 2009)
 - Графовые,
 - Колоночные,
 - Документные,
 - Ключ-значение.

Вопросы создания БД

- Проектирование БД
 - Модель;
 - Структура;
 - Типы данных;
 - Связь элементов.
- Программирование приложений БД
 - На стороне клиента;
 - На стороне сервера БД.
- Распределенная обработка
 - Кластеры;
 - Фрагментация и репликация.

Модели данных


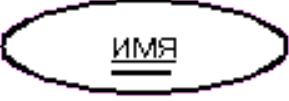

- Концептуальные:
 - ER-модель (сущность-связь);
- Логические:
 - Реляционная (SQL; нормализация);
 - Объектно-реляционная (расширение SQL);
 - Объектная (ODL);
 - Полуструктурированная (XML),
 - Агрегатные;
 - Графовая.

ER – модель

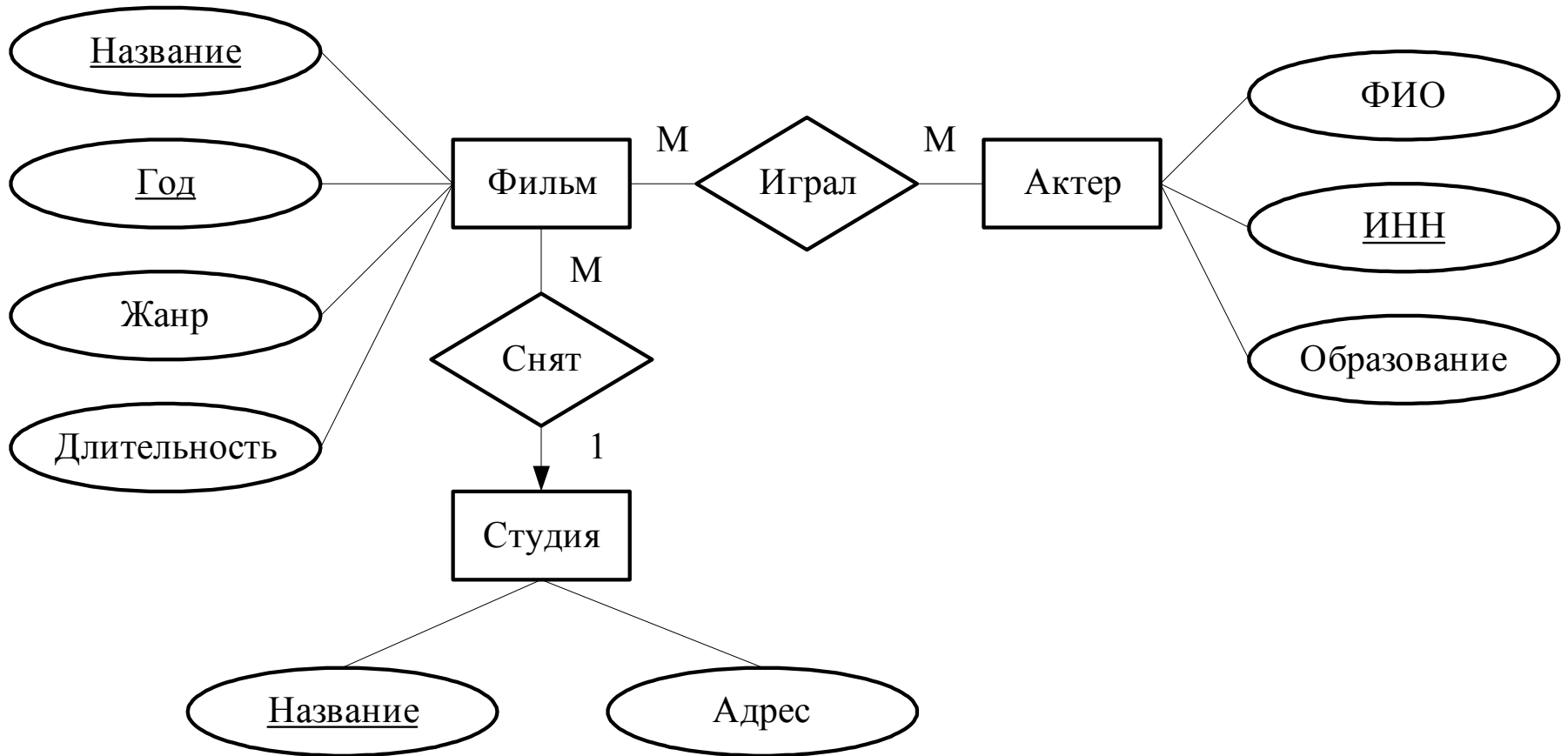
- Entity-relationship model.
- Задаёт структуру БД на концептуальном уровне.
- Объекты предметной области, их атрибуты и связи между ними.
- Использует графическую нотацию.
- Нотации Чена, Мартина, IDEF1X, Баркера и др.

Нотация Чена.

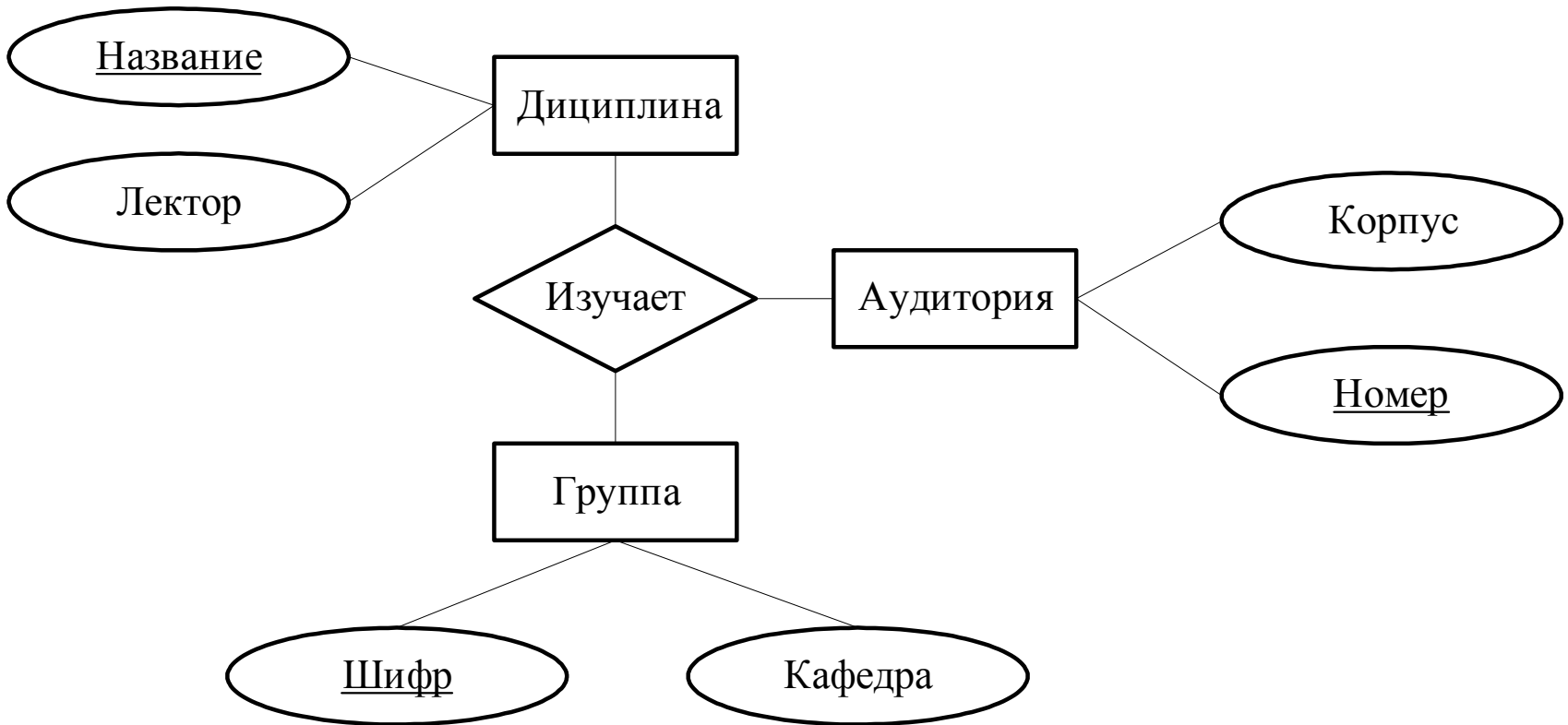
Основные элементы

Элемент модели	Графическое обозначение	Описание элемента
Сущность (набор сущностей)		Определяет объект предметной области или набор подобных объектов
Атрибут		Характеризует сущность. Может относиться к связи
Ключ (первичный ключ)		Соответствует атрибуту(ам), идентифицирующему(им) сущность
Связь		Соединяет две или более сущности. Связи могут быть 1–1, 1–М, М–1 и М–М.



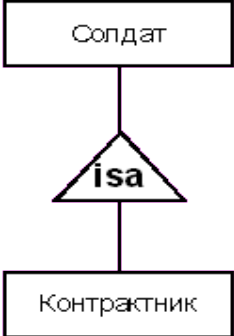
Пример простой ER-модели



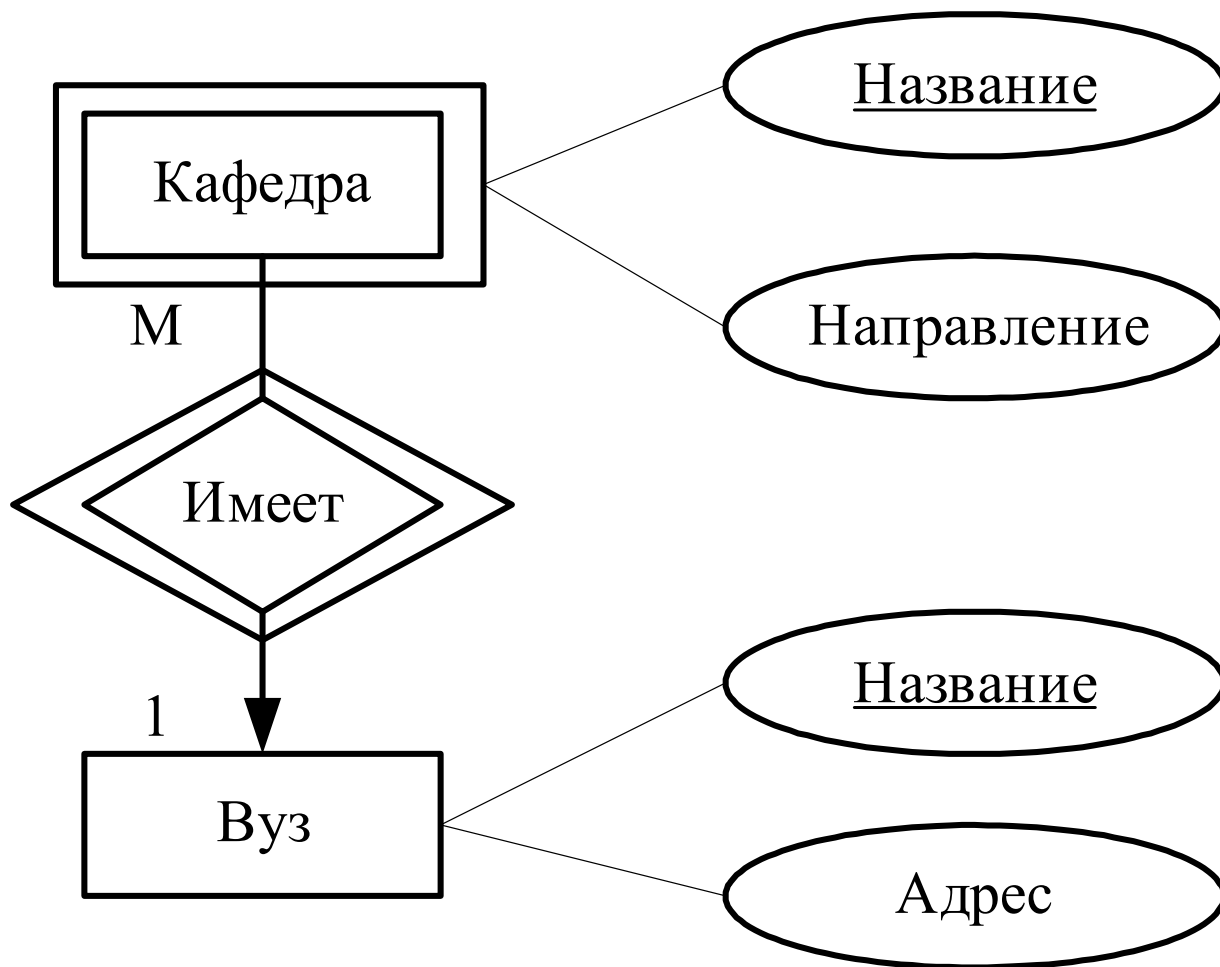
Пример тернарной связи



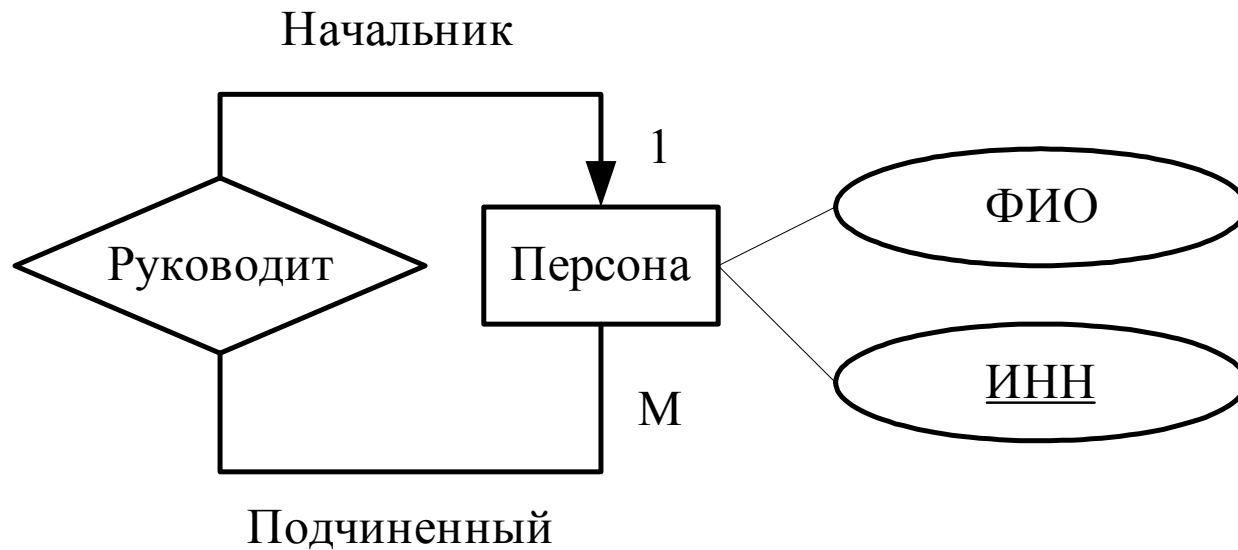
Дополнительные элементы ER-модели

Элемент модели	Графическое обозначение	Описание элемента
Слабая сущность		Не может существовать без поддерживающей сущности
Поддерживающая связь		Соединяет зависимую сущность с поддерживающими ее
Связь ISA		Расширяет базовую сущность дополнительными атрибутами. Аналог наследования

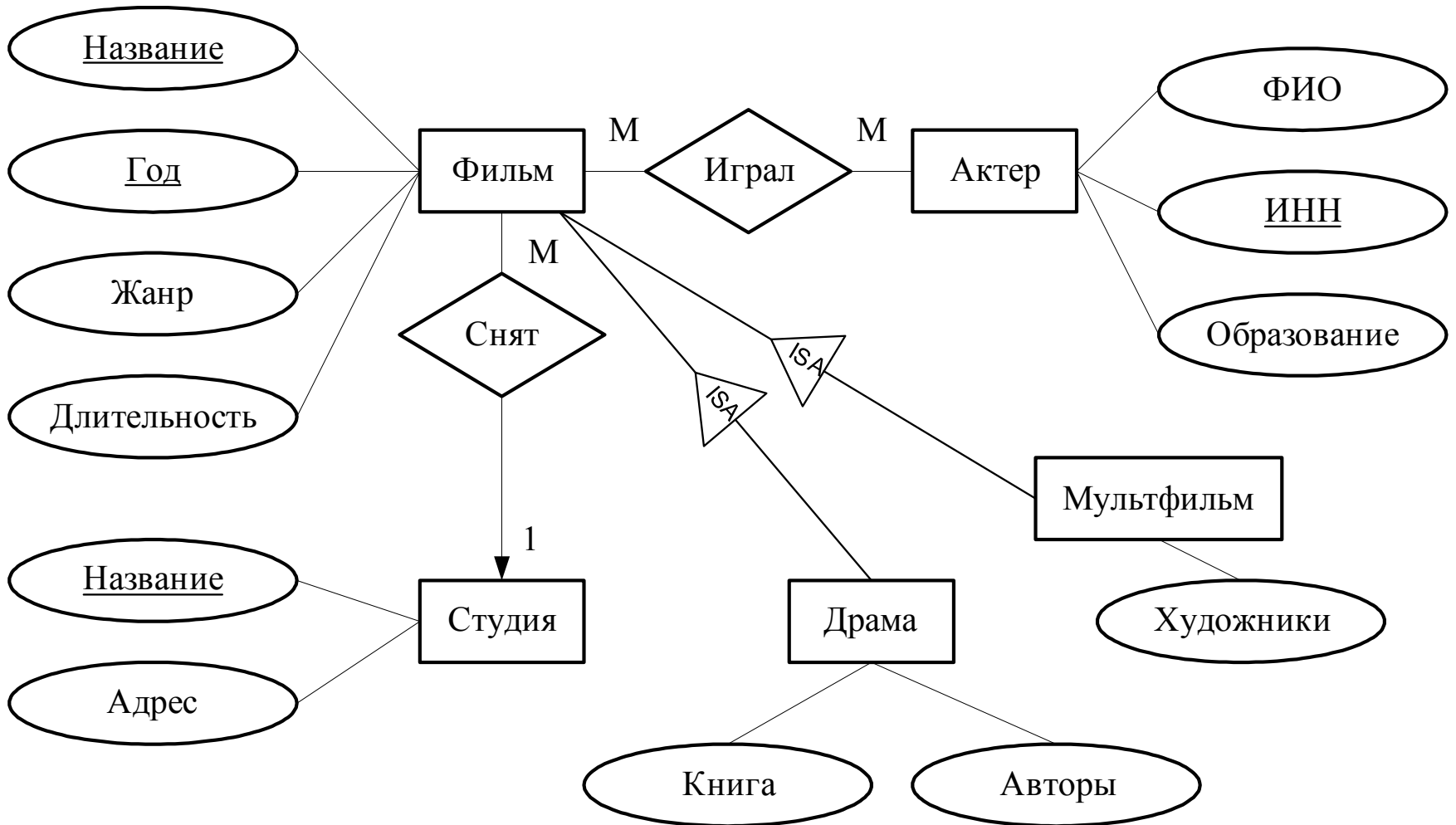
Пример поддерживающей (идентифицирующей) связи



Пример ролей связи



Пример ER-модели



Ограничения на ER-модели

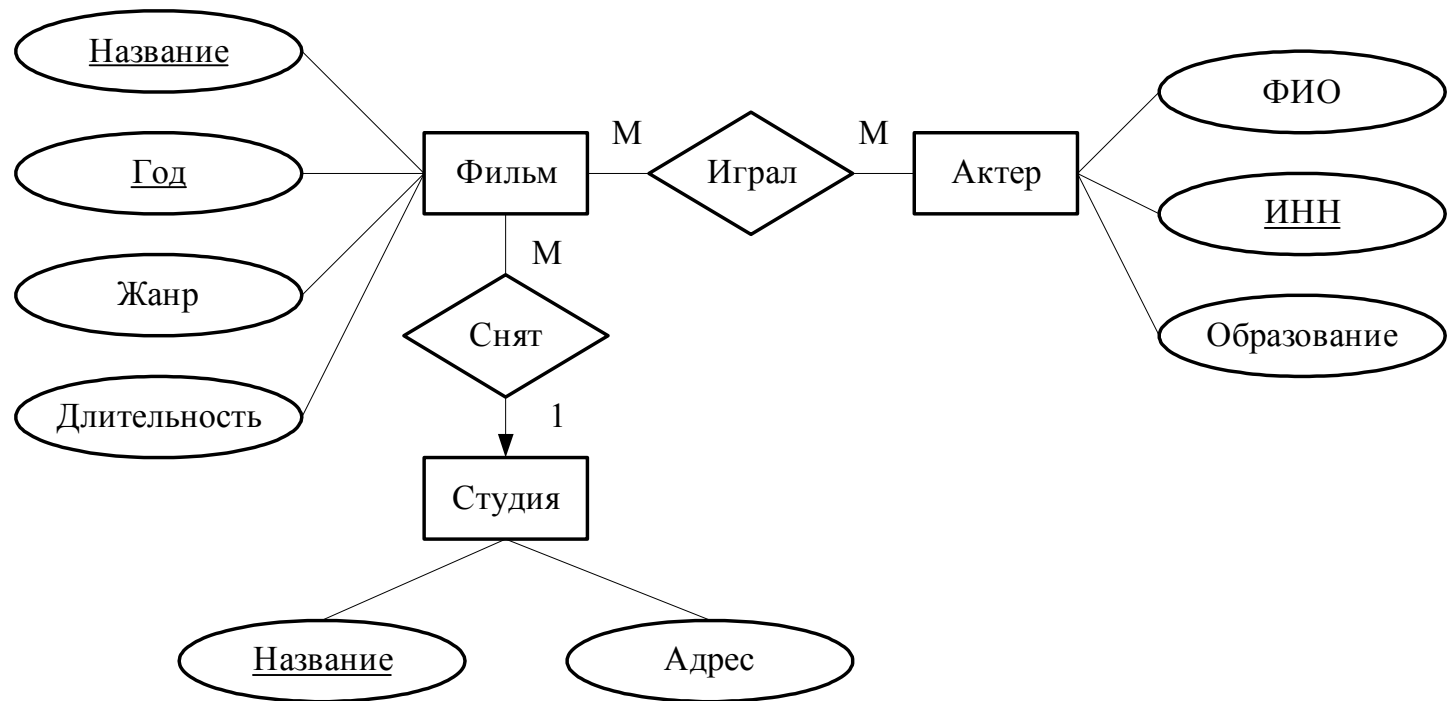
- ограничение связи типом множественности (1–M, 1–1, M–M);
- ограничения, накладываемые слабой сущностью, ключами или связью ISA;
- ограничения домена, уникальности или общего вида (например, в связи участвует не более трех сущностей)
- указывают как комментарий около элемента/связи.

Реляционная модель

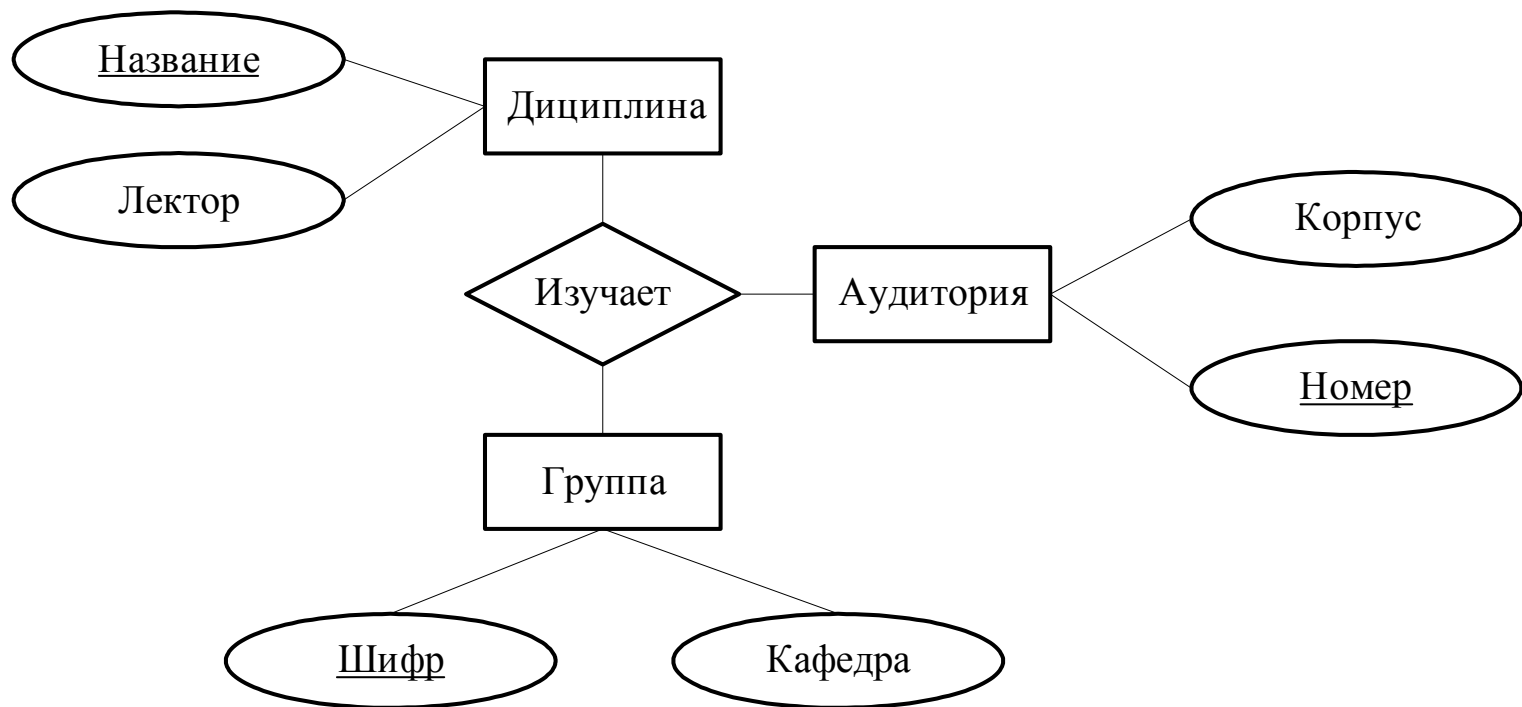
- Особенности:
 - Атомарные атрибуты;
 - Отношения;
 - Схема отношения;
 - Кортеж;
 - Домен;
 - Схема БД.
- Преимущества:
 - Нормализация / SQL / Оптимизация.

Преобразование ER модели к реляционной

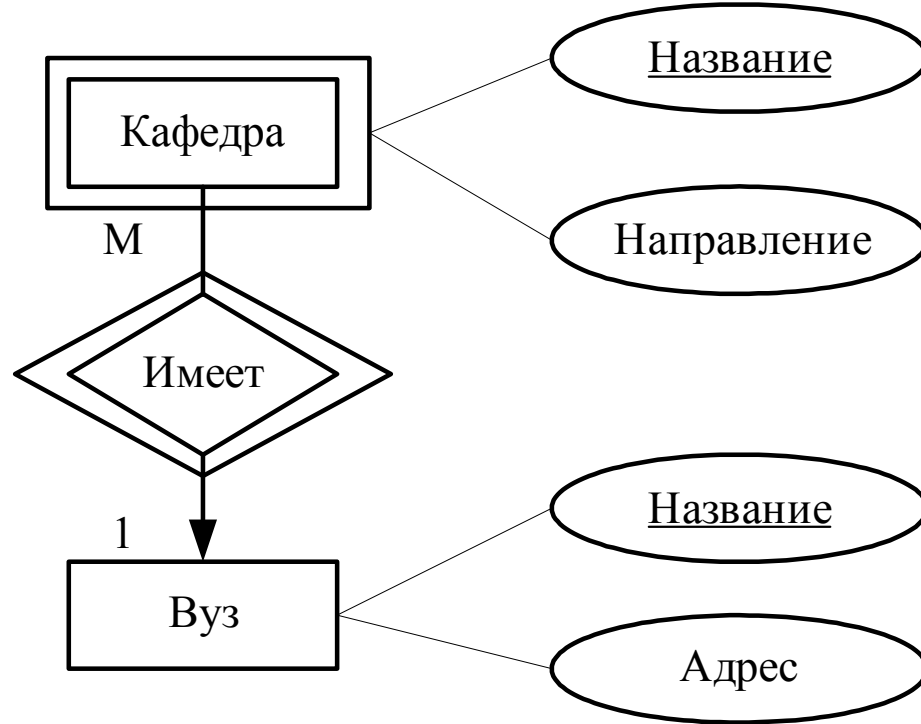
- Сущность → схема отношения (те же атрибуты).
- Атрибут → атрибут.
- Ключ → ключевой атрибут.
- Связь 1-М → внешний ключ на стороне многих.
- Связь М-М → схема отношения
 - содержит ключевые атрибуты связанных сущностей + свои атрибуты,
 - атрибуты могут быть переименованы.
- Слабая сущность → схема отношения
 - атрибуты слабой сущности + ключевые атрибуты всех поддерживающих сущностей.
- Связь слабой сущности → аналогично обычной связи



- Актёр (ИНН, Ф.И.О, Образование)
- Фильм (Название, Год, Длительность, Жанр, Студия)
- Студия (Название, Адрес)
- Актёр–Фильм (Название, Год, ИНН)



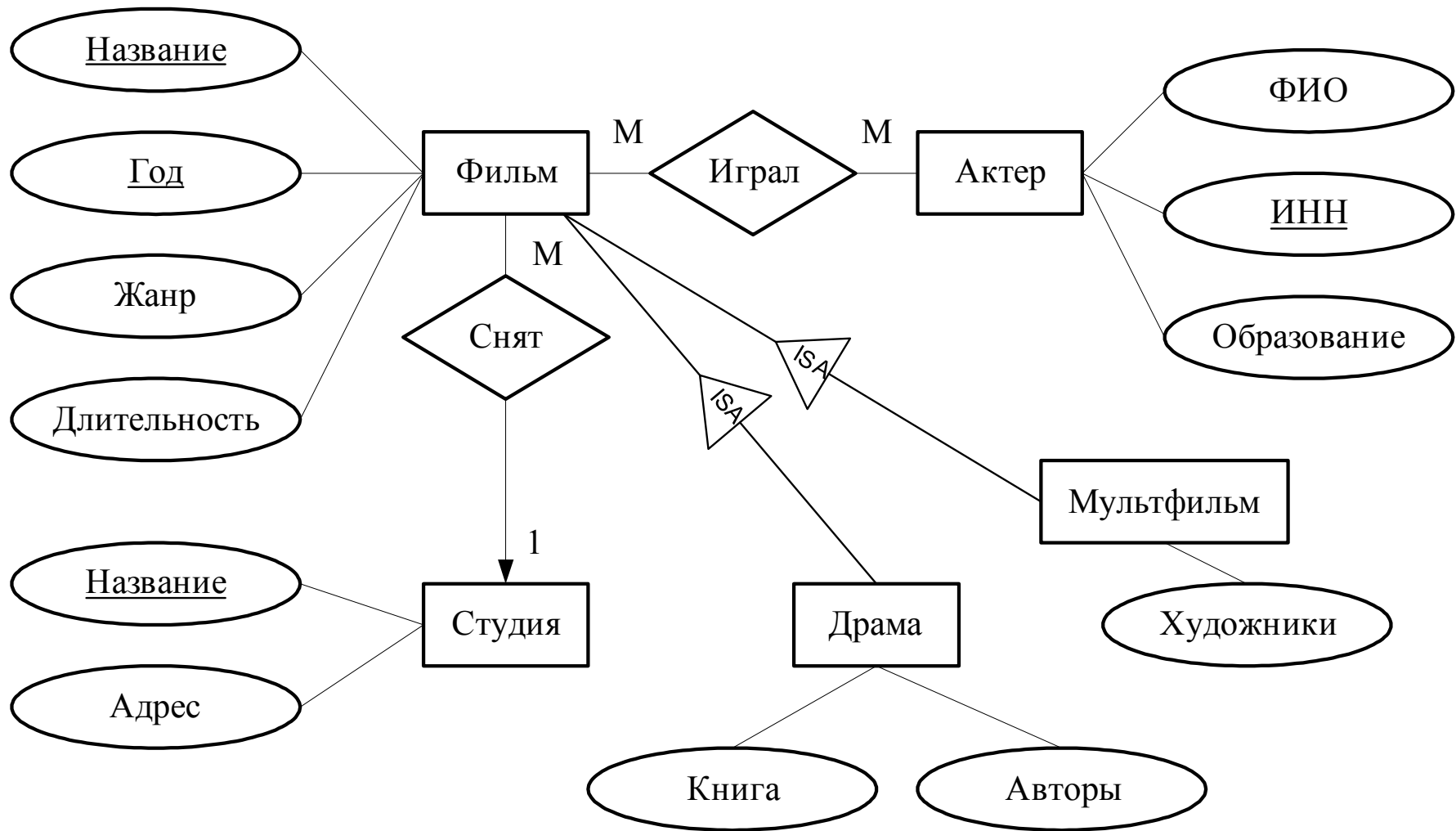
- Дисциплина (Название, Лектор)
- Группа (Шифр, Кафедра)
- Аудитория (Номер, Корпус)
- Занятие (Дисциплина, Группа, Аудитория)



- Кафедра (Название, Направление, Вуз)
- Вуз (Название, Адрес)

Преобразование иерархии сущностей (в реляции)

1. Для каждой сущности → отдельное отношение (объектный подход)
 - его атрибуты + все атрибуты предков.
2. Для каждой сущности → связанное отношение (сущностный подход)
 - его атрибуты + ключевые атрибуты корня (FK),
 - в запросах используют соединение.
3. Одно отношение для всех сущностей иерархии (подход пустых значений)
 - со всеми возможными атрибутами,
 - для отсутствующих значений – NULL.



- При *сущностном подходе*
 - Фильм (Название, Год, Длительность, Жанр, Студия)
 - Мультфильм (Название, Год, Художники)
 - Драма (Название, Год, Книга, Авторы)
- При *объектном подходе*
 - Фильм (Название, Год, Длительность, Жанр, Студия)
 - Мультфильм (Название, Год, Длительность, Жанр, Студия, Художники)
 - Драма (Название, Год, Длительность, Жанр, Студия, Книга, Авторы)
- При *подходе пустых значений*
 - Фильм (Название, Год, Длительность, Жанр, Студия, Художники, Книга, Авторы).

Описание реляционной модели на языке SQL

```
CREATE TABLE имя_таблицы
(
    Имя_поля тип_поля [размер] [ограничение_поля]
                                [ DEFAULT значение ] ,
    ....
    Ограничения целостности ,
    ....
);
```

Атомарные типы

- **integer** — целое поле;
- **char** (количество_символов) — строка из фиксированного количества символов;
- **varchar** (число_символов) — строка символов, размером не более указанного;
- **smallint** — целое число;
- **float** — действительное число;
- **date** — дата;
- **time** — время.

Ограничения

- Для столбца:
 - **NOT NULL** – значение всегда не пустое ;
 - **UNIQUE** — все значения столбца должны быть уникальны;
 - **PRIMARY KEY** — первичный ключ и уникальность;
 - **FOREIGN KEY** — внешний ключ,
 - **CHECK** (условие) —условие для проверки значения столбца. В скобках м.б. условие, выборка, запрос.
- Ограничения для таблицы:
 - **CHECK** — ограничение на значения полей кортежа;
 - **PRIMARY KEY** —первичный ключ,
 - **FOREIGN KEY** — ограничение ссылочной целостности.

Ограничения внешнего ключа

FOREIGN KEY (имя_столбца_внешнего_ключа [,...n])

REFERENCES имя_родительской_таблицы

(имя_столбца_родительской_таблицы [,...n]),

[ON UPDATE

{ CASCADE | SET NULL |
SET DEFAULT | NO ACTION }]

[ON DELETE

{ CASCADE | SET NULL |
SET DEFAULT | NO ACTION }

Пример описания таблиц на SQL

```
CREATE TABLE Stud
(
    sid INTEGER PRIMARY KEY UNIQUE,
    sname VARCHAR(50) NOT NULL,
    addr VARCHAR(300)
);
CREATE TABLE Actor
(
    inn CHAR(10) PRIMARY KEY,
    fio VARCHAR(200),
    edu VARCHAR(50)
);
```

Пример описания таблиц на SQL

```
CREATE TABLE Film
(
  title varchar(100),
  year integer,
  length integer NOT NULL,
  type char(20) CHECK( type IN ('комедия', 'драма',
                                'фантастика') ),
  stud integer REFERENCES Stud(sid)
      ON UPDATE CASCADE
      ON DELETE SET NULL,
  PRIMARY KEY (title, year),
  CHECK ( length>0 AND year>1894 )
);
```


Пример описания таблиц на SQL

```
CREATE TABLE FA
(
  act CHAR(10) NOT NULL REFERENCES Actor(inn)
  ON DELETE CASCADE
  ON UPDATE SET NULL,
  fname VARCHAR(50) NOT NULL,
  fyear INTEGER NOT NULL,
  FOREIGN KEY(fname, fyear) REFERENCES Film(name,
  YEAR)
  ON DELETE CASCADE
  ON UPDATE SET NULL,
  PRIMARY KEY(act, fname, fyear)
);
```