

**Бази от данни**

**Упражнение 2:**  
**Заявки върху**  
**две и повече таблици**

**Димитър Димитров**

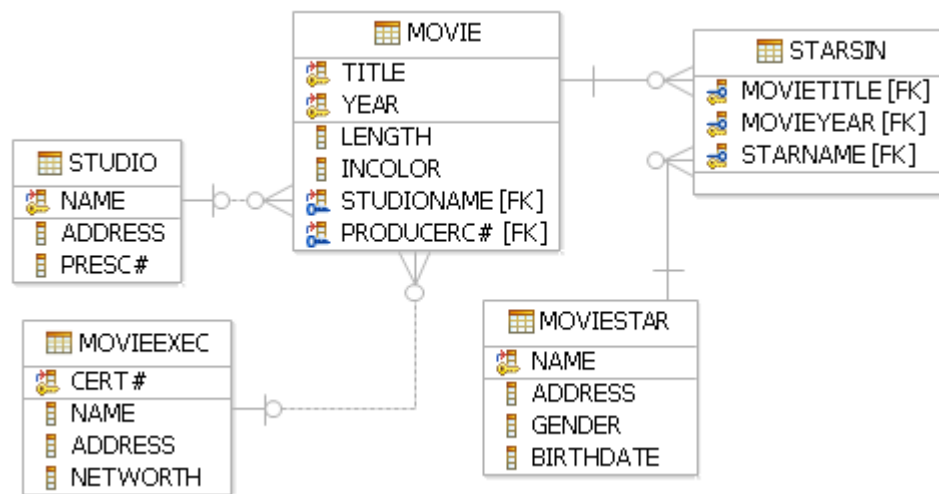
# Въведение

- В практиката рядко ще ни се налага да извличаме информация, съдържаща се само в една таблица на дадена база от данни
- Ще разгледаме две възможности:
  - Съединение на две таблици
  - Обединение, сечение и разлика

# Съединение на таблици

- Искаме за всеки цветен филм да изведем следната информация:

- Заглавие
- Година
- Име на студиото
- Адрес на студиото



- Кои таблици са ни нужни?

- Ще изградим решението стъпка по стъпка
- Започваме отдалеч:  
**SELECT \* FROM Movie; -- 10 реда**  
**SELECT \* FROM Studio; -- 5 реда**
- Да експериментираме:  
**SELECT \***  
**FROM Movie, Studio; -- 50 реда**  
– Получихме декартово произведение

- Премахваме безсмислените (за конкретната задача) комбинации:

```
SELECT *  
FROM Movie, Studio  
WHERE studioname = name;
```

- Колоните в условието не са случайни
- Близко сме до целта

- Работещо решение на задачата:

```
SELECT title, year, studioname, address  
FROM movie, studio  
WHERE studioname = name AND incolor = 'y';
```

# Синтаксис с JOIN

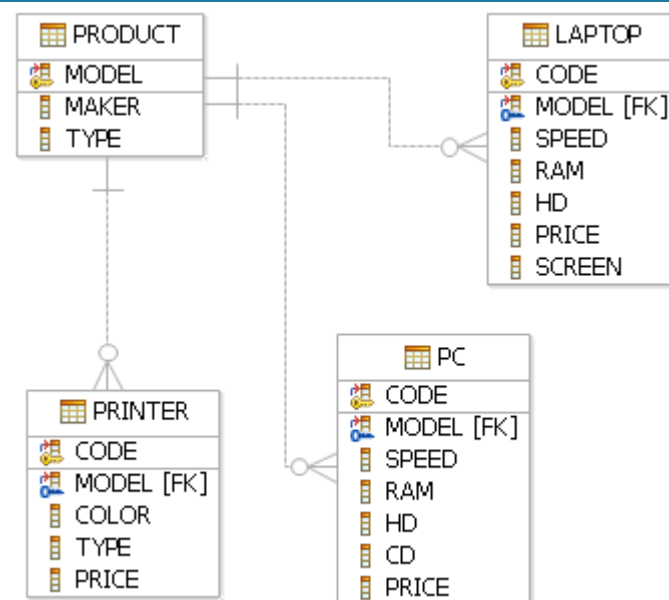
- Въпреки че е коректен, няма да използваме горния синтаксис, а следния:

```
SELECT title, year, studioname, address  
FROM movie  
JOIN studio ON studioname = name  
WHERE incolor = 'y';
```

- Предимства:
  - По-малка вероятност за човешка грешка
  - Консистентен с други видове съединения, които ще разгледаме

# Колони с еднакви имена

- За всеки 15-инчов лаптоп да се изведе производител, модел и цена:



```
SELECT maker, Laptop.model, price
FROM Laptop
JOIN Product
    ON Laptop.model = Product.model
WHERE screen = 15;
```



# Съединение на таблица със себе си

- Заявка, която извежда заглавията на всички филми с дължина, по-голяма от дължината на филма Star Wars:

```
SELECT m1.title  
FROM Movie m1, Movie m2  
WHERE m2.title = 'Star Wars'  
      AND m1.length > m2.length;
```

- Поне едно от срещанията на Movie трябва да има ново име
  - Може да използваме и AS: ... **Movie as m1**, ...
- Същата задача ще решим по много по-лесен начин в следващото упражнение

# Съединение на повече от две таблици

- Имената на всички актриси, които са се снимали във филм на MGM, както и заглавията на тези филми:

```
SELECT name, title  
FROM MovieStar  
JOIN StarsIn ON name = starname  
JOIN Movie ON movietitle = title AND movieyear = year  
WHERE gender = 'F' and studioName = 'MGM';
```

- Как щеше да изглежда WHERE клаузата, ако използвахме другия синтаксис?

# Обединение, сечение и разлика (1)

- Всяка SELECT заявка връща таблица
- Може да разглеждаме таблиците като мултимножества от техните редове
- Може да комбинираме резултати от две заявки

# Обединение, сечение и разлика (2)

- Пример: всички филми, които са цветни и/или в тях е играл Харисън Форд:

```
(SELECT title, year  
FROM Movie  
WHERE incolor = 'y')  
UNION  
(SELECT movietitle, movieyear  
FROM StarsIn  
WHERE starname = 'Harrison Ford');
```

- Скобите не са задължителни

# Обединение, сечение и разлика (3)

- Двете заявки трябва да връщат таблици, за които:
  - Броят на колоните е еднакъв
  - Колоните от двете таблици са от съвместими типове
  - Имената НЕ е нужно да съвпадат

# Обединение, сечение и разлика (4)

- Пример за сечение:

```
(SELECT name, address  
FROM MovieStar  
WHERE gender = 'F')  
INTERSECT  
(SELECT name, address  
FROM MovieExec  
WHERE networth > 100000000);
```

- Разлика - EXCEPT

# Обединение, сечение и разлика (4)

- Сортиране с ORDER BY:  
**(SELECT name, address  
FROM MovieStar)  
EXCEPT  
(SELECT name, address  
FROM MovieExec)  
ORDER BY address;**
- ORDER BY не се слага в подзаявка, а извън тях
  - Ако нямаше скоби, щеше да изглежда, че е част от втората подзаявка, а всъщност не е

# Отстраняване на повторения

- SELECT заявка може да върне резултат, съдържащ повтарящи се редове, вследствие на:
  - Повторения в таблица в базата (рядко)
  - Проекция (често)
- Решение – DISTINCT веднага след SELECT
  - Ако има два или повече напълно съвпадащи реда, в резултата ще остане само един от тях
- Имената на всички битки, в които има потънал кораб:

```
SELECT DISTINCT battle  
FROM Outcomes  
WHERE result = 'sunk';
```



# Повторения при обединение и т.н.

- UNION, INTERSECT и EXCEPT автоматично премахват повтарящите се редове
- Ако искаме да ги запазим – UNION ALL
  - INTERSECT ALL и EXCEPT ALL не се поддържат от SQL Server

# Допълнителен материал

- **SELECT \***  
**FROM T**  
**WHERE c1 = ... OR c2 = ...;**
- **SELECT \***  
**FROM T**  
**WHERE c1 = ...**  
**UNION**  
**SELECT \***  
**FROM T**  
**WHERE c2 = ...;**
- Втората заявка би била по-ефективна в случая, когато има индекси (ще ги учим по-късно)

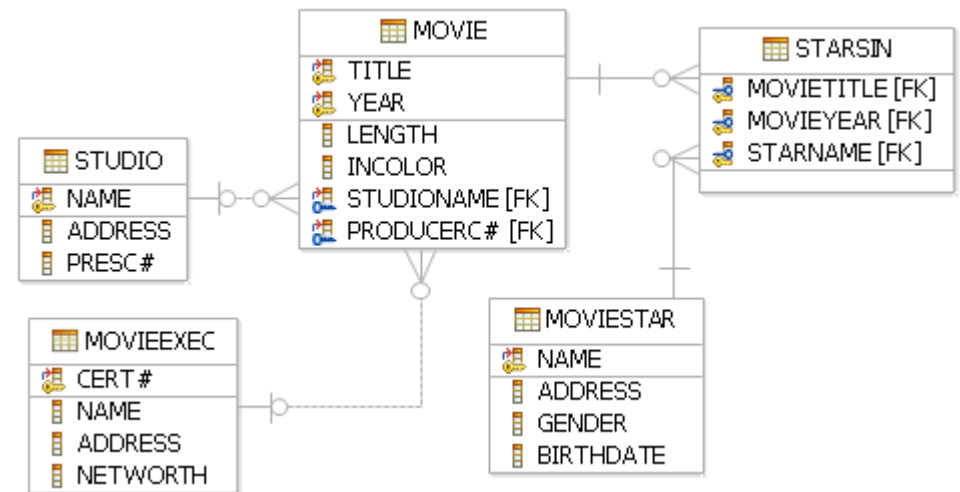
# Въпроси?

A SQL query walks into a bar and sees two tables.  
He walks up to them and says “Can I join you?”

## Следват задачи

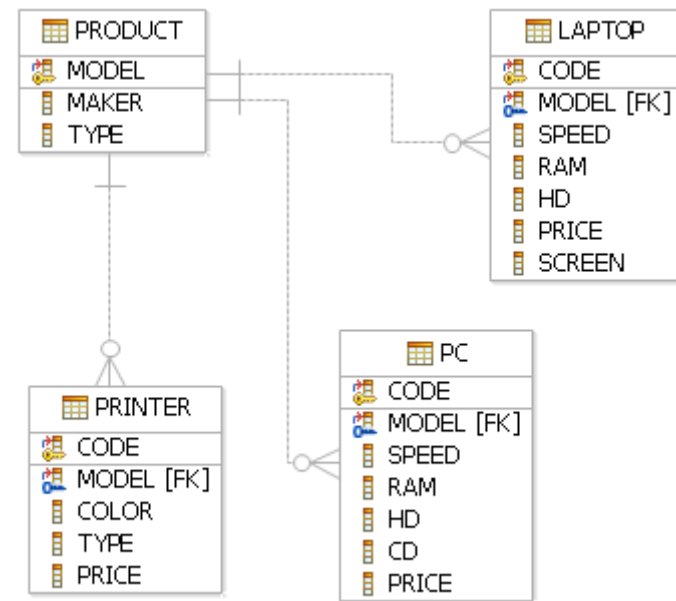
# 1. Задачи - Movies

1. Напишете заявка, която извежда имената на актрисите, участвали в Terms of Endearment.
2. Напишете заявка, която извежда имената на филмовите звезди, участвали във филми на студио MGM през 1995 г.



## 2. Задачи - РС

1. Напишете заявка, която извежда производителя и честотата на процесора на лаптопите с размер на харддиска поне 9 GB.
2. Напишете заявка, която извежда номер на модел и цена на всички продукти, произведени от производител с име 'B'. Сортирайте резултата така, че първо да се изведат най-скъпите продукти.
3. Напишете заявка, която извежда размерите на тези харддискосе, които се предлагат в поне два компютъра.
4. Напишете заявка, която извежда всички двойки модели на компютри, които имат еднаква честота на процесора и памет. Двойките трябва да се показват само по веднъж, например ако вече е изведена двойката (i, j), не трябва да се извежда (j, i)
5. Напишете заявка, която извежда производителите на поне два различни компютъра с честота на процесора поне 500 MHz.



# 3. Задачи - Ships

1. Напишете заявка, която извежда името на корабите, по-тежки (displacement) от 35000.
2. Напишете заявка, която извежда имената, водоизместимостта и броя оръдия на всички кораби, участвали в битката при Guadalcanal.
3. Напишете заявка, която извежда имената на тези държави, които имат класове кораби от тип 'bb' и 'bc' едновременно.
4. Напишете заявка, която извежда имената на тези кораби, които са били повредени в една битка, но по-късно са участвали в друга битка.

