# SQL

Максим Крижановски Комфо Софтуерен инженер



#### Модел на база данни

Описва как са структурирани данните и какви са основните операции за манипулиране с тях.

- Нерелационни
- Релационни

#### Релационни бази данни

- •Най-разпространеният модел
- Разработен от Едгар Код (E. Codd)
- •Данните се съхраняват като записи в таблици, които могат да имат релации (връзки) с други таблици
- Има стандартизиран език за обработка на данните SQL (Structured Query Language)

# RDBMS - Система за управление на база данни

Софтуерът, който контролира операциите с данните, съхранението им и достъпа до тях..

#### Примери:

MS SQL Server, Oracle, IBM DB2, MySQL, PostgreSQL

#### Релационен модел - компоненти

- Данните се съхраняват в таблици съвкупност от стойности, подредени в редове и колони
- Редовете имат едни и същи полета
- Колоните имат уникално име и определящ съдържанието им тип

### Hayчете SQL DDL

CREATE, ALTER, DROP...

#### **DML**

- SELECT
- INSERT
- •UPDATE
- •DELETE



#### Movies

id	Title	Length	Year
1	Nindja Movie	120	1990
2	Die Hard 5	140	2011
3	Trainspoting	210	1982
4	Geen Mile	310	1991
5	Titanic	314	2000
6	Нор	122	2010

#### Таблицата в една база данни

- Редовете имат еднаква структура
- Колоните имат име и тип (число, символен низ, дата или др.)
- Типовете са различни според различните DBMS

#### Извличане на данни (Select)

#### **SELECT \* FROM Movies**

id	Title	Length	Year
1	Nindja Movie	120	1990
2	Die Hard 5	140	2011
3	Trainspoting	210	1982
4	Geen Mile	310	1991
5	Titanic	314	2000
6	Нор	122	2010

#### Извличане на данни (Select)

#### SELECT id, title, length, year FROM Movies

id	Title	Length	Year
1	Nindja Movie	120	1990
2	Die Hard 5	140	2011
3	Trainspoting	210	1982
4	Geen Mile	310	1991
5	Titanic	314	2000
6	Нор	122	2010

### Where клауза

#### SELECT id, title, length, year FROM Movies WHERE title = 'Die Hard 5'

id	Title	Length	Year
2	Die Hard 5	140	2011

# SELECT id, title, length, year FROM Movies WHERE length<130

ìd	Title	Length	Year
1	Nindja Movie	120	1990
6	Нор	122	2010

## Order by клауза

# SELECT id, title, length, year FROM Movies WHERE length<130 ORDER BY year DESC LIMIT 2

id	Title	Length	Year
6	Нор	122	2010
1	Nindja Movie	120	1990

# SELECT MAX(Length) AS 'MaxLength' FROM Movies

MaxLength

314

MAX, MIN, AVG, COUNT

SELECT COUNT(\*) AS 'MoviesCount' FROM Movies GROUP BY Year

### INSERT, UPDATE

INSERT INTO Movies(id, title, length, year)
SELECT 7, 'Rambo', 144, 1982

#### Hayчете SQL

**UPDATE Movies** 

*SET year = 1987* 

WHERE id = 7

# Разцъкай



#### Релации

- Съпоставка на записи от една със записи от друга таблица
- Обикновено се базират на primary key-foreign key отношения
- Спестяват повтарянето на информация
- Множественост:
  - -1 x 1 (one-to-one)
  - $-1 \times N (N \times 1)$  (one-to-many)
  - –N x M (many-to-many)

#### Релации - примери

One-to-one

• Държава <> столица

One-to-many

- Държава <> градове
- Many-to-many
- Студент <> изборен курс

#### Movies

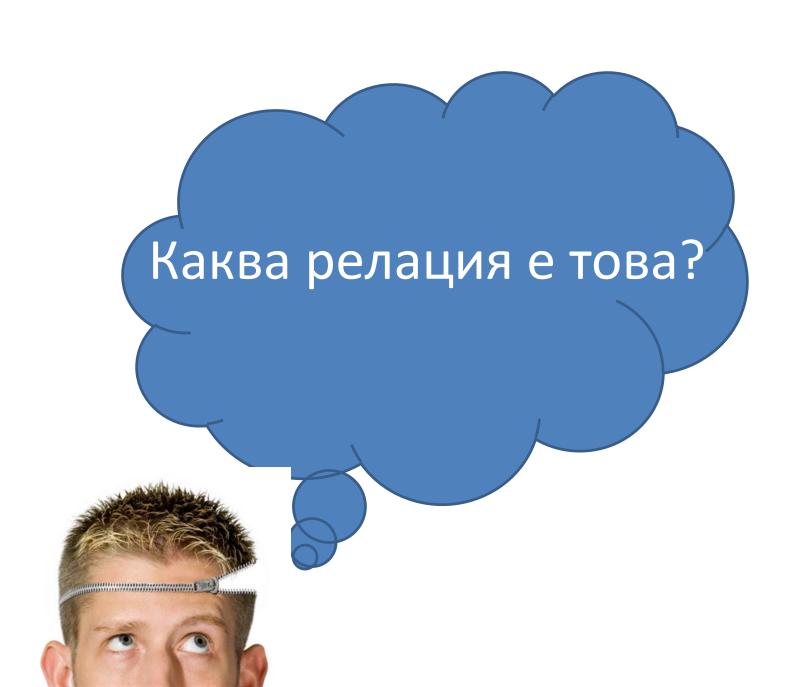
id	Title	Length	Year
1	Nindja Movie	120	1990
2	Die Hard 5	140	2011
3	Trainspoting	210	1982
4	Geen Mile	310	1991
5	Titanic	314	2000
6	Нор	122	2010

# Bloggers

id	Name	FavoriteMovie	BlogAddress
1	Ivaylo Hristov	Die Hard 5	hristov.bg
2	Milen Ivanov	Nindja Movie	alabala.com
3	Leila Kudsen	Die Hard 5	leila.dk

# Bloggers

id	Name	FavMovieID	BlogAddress
1	Ivaylo Hristov	2	hristov.bg
2	Milen Ivanov	1	alabala.com
3	Leila Kudsen	2	leila.dk



#### SELECT name, blogAddress FROM Bloggers

Name	BlogAdress
Ivaylo Hristov	hristov.bg
Milen Ivanov	alabala.com
Leila Kudsen	leila.dk

```
SELECT name, blogAddress,

(

SELECT title FROM Movies

WHERE id = Bloggers.FavMoviewID

) as FavMovie

FROM Bloggers
```

Name	FavMovie	BlogAddress
Ivaylo Hristov	Die Hard 5	hristov.bg
Milen Ivanov	Nindja Movie	alabala.com
Leila Kudsen	Die Hard 5	leila.dk

### Join – Свързване на таблици

**SELECT** 

Bloggers.name,

Movie.title,

Bloggers.BlogAddress

FROM Bloggers JOIN Movies

ON Bloggers. FavMovieID = Movies.id

Name	FavMovie	BlogAddress
Ivaylo Hristov	Die Hard 5	hristov.bg
Milen Ivanov	Nindja Movie	alabala.com
Leila Kudsen	Die Hard 5	leila.dk

#### Join – Свързване на таблици

- •Inner
- Left (Outer)
- Right (Outer)

#### **SELECT**

Bloggers.name,

Movie.title,

Bloggers.BlogAddress

FROM Bloggers LEFT JOIN Movies

ON Bloggers. FavMovieID = Movies.id

#### **Primary Key**

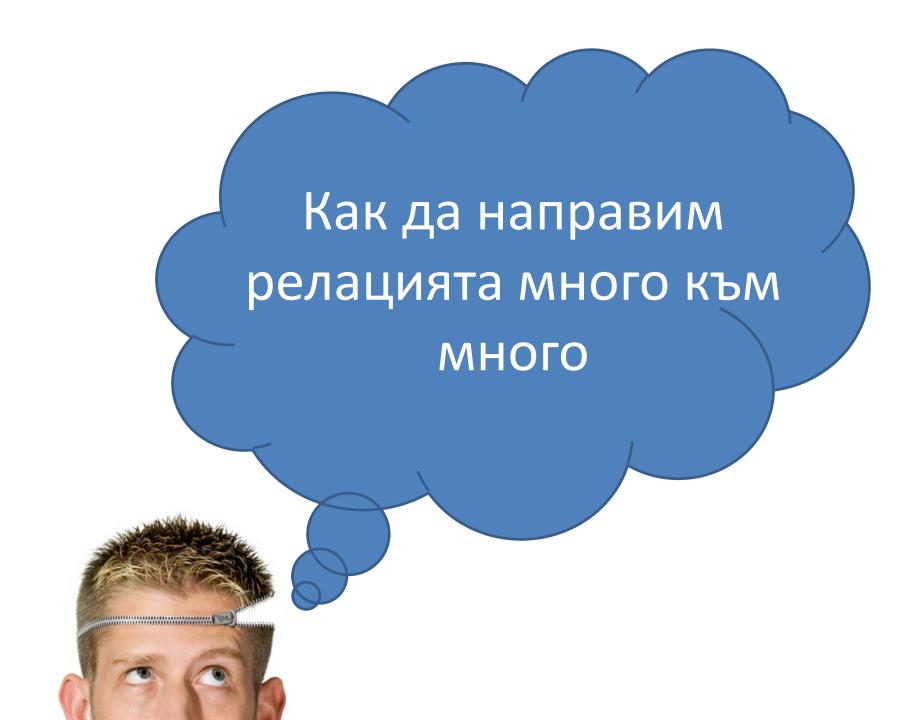
- Състои се от една или повече колони
- Уникално идентифицира всеки запис в таблицата
- Два записа са различни точно когато имат различни стойности в тези колони
- Auto-increment

#### Foreign Key

- Указва една или повече колони от една таблица като свързани с една или повече колони от друга
  - Сочи уникални стойности
  - Служи като връзка (релация) между двете таблици

#### **Actors**

id	Name	Country	Age
1	Ivaylo Hristov	Bulgaria	28
2	Bruce Willis	USA	60
3	Katy Perry	USA	26
4	Gerard Butler	Ireland	40



#### StartsIn

id	ActorID	MovieID
1	1	1
2	1	2
3	2	1
4	2	3

#### Нормализация

- Процес на реорганизация на данните за поефективно използване и съхранение
- •Премахва дублиращите се данни
- •Предотвратява аномалии при:
- –Добавяне
- –Промяна
- -Изтриване
- •Денормализация

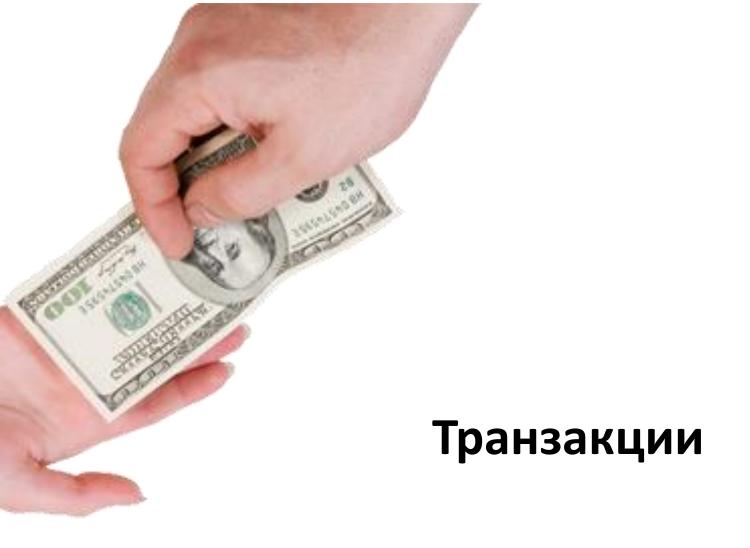
### Нормализация

id	Title_en	Title_bg	Length	Year
1	Die Hard 5	Умирай трудно 5	120	1990

id	Title	Lang	Length	Year
1	Die Hard 5	En	120	1990
2	Умирай трудно 5	Bg	120	1990

#### Индекси

- DB сървъра достъпва данните:
  - Чрез сканиране на таблицата (table scan)
  - –Използвайки индекси(B-tree или hash table)
- Ускорява намирането и извличането на данни
- Primary Key-а също е индекс!



#### **Constraints**



### **PHPMyAdmin**



# THE END...