

NET-FIRST! – Aufgabenblatt 3

3.1 Aufgabe

Entry(id, name, production_year, type → Type)
Person(id, name, task → Task)
Type(id, name, description)
Task(id, name)
participates(person → Person, task → Task, Name)

3.2 Aufgabe

1. Entry

```
1 CREATE TABLE Entry (  
2     id INTEGER,  
3     name VARCHAR(255),  
4     production_year DATE,  
5     type INTEGER REFERENCES Type  
6 );
```

2. Person

```
1 CREATE TABLE Person (  
2     id INTEGER,  
3     name VARCHAR(255),  
4     task INTEGER REFERENCES Task  
5 );
```

3. Type

```
1 CREATE TABLE Type (  
2     id INTEGER,  
3     name VARCHAR(255),  
4     description VARCHAR(255)  
5 );
```

4. Task

```

1 CREATE TABLE Task (
2     id INTEGER,
3     name VARCHAR(255)
4 );

```

5. participates

```

1 CREATE TABLE participates (
2     id INTEGER NOT NULL REFERENCES Person,
3     id INTEGER NOT NULL REFERENCES Task,
4     name VARCHAR(255) REFERENCES Task
5 );

```

3.3 Aufgabe

Anfrage 1: Finde den *title* und *type* aller *Movies* seit 2004.

```

1 SELECT TITLE_TITLE, TITLE_TYPE
2 FROM IMDB.MOVIES
3 WHERE TITLE_YEAR >= '2004';

```

Anfrage 2: Finde die Name aller *Actors* (und *Actresses*), die in mindestens einem der Filme aus Anfrage 1 mitgespielt haben. Gib jeden Namen nur einmal aus!

```

1 SELECT DISTINCT NAME_NAME
2 FROM IMDB.ACTORS a, IMDB.MOVIES m
3 WHERE a.TITLE_ID = m.TITLE_ID
4 AND m.TITLE_YEAR >= '2004';

```

Anfrage 3: Finde heraus, wie viele *Movies* es seit 2004 gegeben hat (die Anzahl)!

```

1 SELECT COUNT(*)
2 FROM IMDB.MOVIES
3 WHERE TITLE_YEAR >= '2004'
4 AND TITLE_TYPE='film';

```

Anfrage 4: Finde die maximale und durchschnittliche Anzahl von *Movies*, in denen *Actors* (und *Actresses*) seit 2004 mitgespielt haben. *Actors* (und *Actresses*), die nur in einem *Movie* mitgespielt haben, sollen nicht mit betrachtet werden!

Anfrage 5: Finde die *names* aller *Actors* (und *Actresses*), die in mindestens 30 *Movies* (nicht *Serien*) seit 2004 mitgespielt haben und die Anzahl der *Movies* in denen sie mitgespielt haben, absteigend sortiert nach der Anzahl der *title* und ihre *names* in alphabetischer Reihenfolge.

3.4 Aufgabe

```
1 INSERT INTO Entry (Name, production_year, type)
2 VALUES
3     SELECT x AS Name, y AS production_year, z AS type
4     FROM blah;
```