

NET-FIRST! – Aufgabenblatt 3

3.1 Aufgabe

Entry(id, name, production_year, type → Type)

Person(id, name)

Type(id, name, description)

Task(id, name)

participates(person → Person, task → Task, entry → Entry)

3.2 Aufgabe

1. Entry

```
1 CREATE TABLE Entry (  
2   id VARCHAR(400) NOT NULL,  
3   name VARCHAR(400),  
4   production_year INTEGER,  
5   type VARCHAR(10) REFERENCES Type(id),  
6   PRIMARY KEY(id)  
7 );
```

2. Person

```
1 CREATE TABLE Person (  
2   id VARCHAR(400) NOT NULL,  
3   name VARCHAR(400),  
4   task INTEGER REFERENCES Task(id),  
5   PRIMARY KEY(id)  
6 );
```

3. Type

```
1 CREATE TABLE Type (  
2   id VARCHAR(400) NOT NULL,  
3   name VARCHAR(400),  
4   description VARCHAR(400),  
5   PRIMARY KEY(id)  
6 );
```

4. Task

```
1 CREATE TABLE Task (  
2   id INTEGER NOT NULL,  
3   name VARCHAR(400),  
4   PRIMARY KEY(id)  
5 );
```

5. participates

```
1 CREATE TABLE participates (  
2   person VARCHAR(400) NOT NULL REFERENCES Person(id),  
3   task INTEGER NOT NULL REFERENCES Task(id),  
4   entry VARCHAR(400) NOT NULL REFERENCES Entry(id),  
5   PRIMARY KEY(person, task, entry)  
6 );
```

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 WHERE a.TITLE_ID = m.TITLE_ID  
3 AND m.TITLE_YEAR >= 2004  
4 UNION  
5 SELECT DISTINCT NAME_NAME  
6 FROM IMDB.ACTRESSES a, IMDB.MOVIES m WHERE a.TITLE_ID = m.TITLE_ID  
7 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!

```

1 SELECT MAX(COUNT(m.TITLE_ID)) AS MaxCount, AVG(COUNT(m.TITLE_ID)) AS AvgCount
2 FROM IMDB.MOVIES m, IMDB.ACTORS a
3 WHERE m.TITLE_ID = a.TITLE_ID
4 AND m.TITLE_YEAR >= 2004
5 UNION
6 SELECT MAX(COUNT(m.TITLE_ID)) AS MaxCount, AVG(COUNT(m.TITLE_ID)) AS AvgCount
7 FROM IMDB.MOVIES m, IMDB.ACTORS ac
8 WHERE m.TITLE_ID = ac.TITLE_ID
9 AND m.TITLE_YEAR >= 2004;

```

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 (id, name, production_year, type)
2 VALUES
3     SELECT TITLE_ID AS id,
4           TITLE_TITLE AS name,
5           TITLE_YEAR AS production_year,
6           TITLE_TYPE AS type
7 FROM IMDB.MOVIES
8 WHERE TITLE_YEAR = 2001;
9
10
11
12 INSERT INTO Person (name, task)
13 VALUES
14     SELECT a AS name, b AS task
15 FROM Person, Task
16 WHERE
17
18
19 INSERT INTO Task (id, name)
20 VALUES (1, 'Actor'),
21         (2, 'Actress'),
22         (3, 'Writer'),
23         (4, 'Director'),
24         (5, 'Cinematographer'),
25         (6, 'Composer'),
26         (7, 'Costume_Designer'),
27         (8, 'Producer'),
28         (9, 'Editor');

```