

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(10) 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(c.Count) AS max, d.Count AS avg
2  FROM IMDB.MOVIES m,
3       IMDB.ACTORS a,
4       (SELECT COUNT(*) AS Count FROM IMDB.MOVIES) c
5       (SELECT AVG(COUNT(*)) AS Count FROM IMDB.MOVIES) d
6       WHERE m.TITLE_ID = a.TITLE_ID
7       AND TITLE_YEAR >= 2004
8  UNION
9  SELECT MAX(c.Count) AS max, d.Count AS avg
10 FROM IMDB.MOVIES m,
11      IMDB.ACTRESSES a,
12      (SELECT COUNT(*) AS Count FROM IMDB.MOVIES) c
13      (SELECT AVG(COUNT(*)) AS Count FROM IMDB.MOVIES) d
14      WHERE m.TITLE_ID = a.TITLE_ID
15      AND TITLE_YEAR >= 2004
16
17 -----
18 SELECT MAX(Count) max, AVG(Count) avg FROM
19      (SELECT COUNT(*) AS Count
20       FROM IMDB.MOVIES AS m, IMDB.ACTORS AS a
21       WHERE m.TITLE_ID = a.TITLE_ID
22       AND TITLE_YEAR >= 2004)
23 UNION
24 SELECT MAX(Count) max, (AVG(Count) avg FROM
25      (SELECT COUNT(*) AS Count
26       FROM IMDB.MOVIES AS m, IMDB.ACTRESSES AS a
27       WHERE m.TITLE_ID = a.TITLE_ID
28       AND 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
2
3
4  INSERT INTO Type (id, name)
5  VALUES ('film', 'Ein_Film');
6
7  INSERT INTO Entry (id, name, production_year, type)
8  (SELECT TITLE_ID,
9         TITLE_TITLE,
10        TITLE_YEAR,
11        TITLE_TYPE
12  FROM IMDB.MOVIES

```

```

13     WHERE TITLE_YEAR = 2001
14     AND     TITLE_TYPE = 'film');
15
16
17
18
19
20 INSERT INTO Person (name, task)
21 VALUES
22     SELECT a AS name, b AS task
23     FROM Person, Task
24     WHERE
25
26
27 INSERT INTO Task (id, name)
28 VALUES (1, 'Actor'),
29         (2, 'Actress'),
30         (3, 'Writer'),
31         (4, 'Director'),
32         (5, 'Cinematographer'),
33         (6, 'Composer'),
34         (7, 'Costume_Designer'),
35         (8, 'Producer'),
36         (9, 'Editor');

```