

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   PRIMARY KEY(id)  
5 );
```

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, entry, task)
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  WITH
2      Avrg AS (
3          SELECT AVG(COUNT(*)) AS average FROM IMDB.MOVIES)
4  (SELECT MAX(c.Count) AS max
5   FROM IMDB.MOVIES m,
6        IMDB.ACTORS a,
7        (SELECT COUNT(*) AS Count FROM IMDB.MOVIES) c
8        WHERE m.TITLE_ID = a.TITLE_ID
9              AND TITLE_YEAR >= 2004
10 UNION
11 SELECT MAX(c.Count) AS max
12 FROM IMDB.MOVIES m,
13      IMDB.ACTRESSES a,
14      (SELECT COUNT(*) AS Count FROM IMDB.MOVIES) c
15      WHERE m.TITLE_ID = a.TITLE_ID
16            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  INSERT INTO Task (id, name)
17  VALUES (1, 'Actor'),
18          (2, 'Actress'),
19          (3, 'Writer'),
20          (4, 'Director'),
21          (5, 'Cinematographer'),
22          (6, 'Composer'),
23          (7, 'Costume_Designer'),
24          (8, 'Producer'),

```

```

25         (9, 'Editor');
26
27 INSERT INTO DBLAB01.Person (id, name)
28     (SELECT a.NAME_ID, a.NAME_NAME
29     FROM IMDB.ACTORs a, DBLAB01.Entry
30     WHERE a.TITLE_ID = Entry.id);
31 INSERT INTO participates (person, task, entry)
32     (SELECT NAME_NAME, 1, TITLE_ID
33     FROM IMDB.ACTORs, Entry
34     WHERE TITLE_ID = Entry.id);
35
36 INSERT INTO DBLAB01.Person (id, name)
37     (SELECT a.NAME_ID, a.NAME_NAME
38     FROM IMDB.ACTRESSES a, DBLAB01.Entry
39     WHERE a.TITLE_ID = Entry.id);
40 INSERT INTO participates (person, task, entry)
41     (SELECT NAME_NAME, 2, TITLE_ID
42     FROM IMDB.ACTRESSES, Entry
43     WHERE TITLE_ID = Entry.id);
44
45 INSERT INTO DBLAB01.Person (id, name)
46     (SELECT a.NAME_ID, a.NAME_NAME
47     FROM IMDB.WRITERS a, DBLAB01.Entry
48     WHERE a.TITLE_ID = Entry.id);
49 INSERT INTO participates (person, task, entry)
50     (SELECT NAME_NAME, 3, TITLE_ID
51     FROM IMDB.WRITERS, Entry
52     WHERE TITLE_ID = Entry.id);
53
54 INSERT INTO DBLAB01.Person (id, name)
55     (SELECT a.NAME_ID, a.NAME_NAME
56     FROM IMDB.DIRECTORS a, DBLAB01.Entry
57     WHERE a.TITLE_ID = Entry.id);
58 INSERT INTO participates (person, task, entry)
59     (SELECT NAME_NAME, 4, TITLE_ID
60     FROM IMDB.DIRECTORS, Entry
61     WHERE TITLE_ID = Entry.id);
62
63 INSERT INTO DBLAB01.Person (id, name)
64     (SELECT a.NAME_ID, a.NAME_NAME
65     FROM IMDB.CINEMATOGRAPHERS a, DBLAB01.Entry
66     WHERE a.TITLE_ID = Entry.id);
67 INSERT INTO participates (person, task, entry)
68     (SELECT NAME_NAME, 5, TITLE_ID
69     FROM IMDB.CINEMATOGRAPHERS, Entry
70     WHERE TITLE_ID = Entry.id);
71
72 INSERT INTO DBLAB01.Person (id, name)
73     (SELECT a.NAME_ID, a.NAME_NAME
74     FROM IMDB.COMPOSERS a, DBLAB01.Entry
75     WHERE a.TITLE_ID = Entry.id);

```

```

76 INSERT INTO participates (person, task, entry)
77     (SELECT NAME_NAME, 6, TITLE_ID
78     FROM IMDB.COMPOSERS, Entry
79     WHERE TITLE_ID = Entry.id);
80
81 INSERT INTO DBLAB01.Person (id, name)
82     (SELECT a.NAME_ID, a.NAME_NAME
83     FROM IMDB.COSTUME_DESIGNERS a, DBLAB01.Entry
84     WHERE a.TITLE_ID = Entry.id);
85 INSERT INTO participates (person, task, entry)
86     (SELECT NAME_NAME, 7, TITLE_ID
87     FROM IMDB.COSTUME_DESIGNERS, Entry
88     WHERE TITLE_ID = Entry.id);
89
90 INSERT INTO DBLAB01.Person (id, name)
91     (SELECT a.NAME_ID, a.NAME_NAME
92     FROM IMDB.PRODUCERS a, DBLAB01.Entry
93     WHERE a.TITLE_ID = Entry.id);
94 INSERT INTO participates (person, task, entry)
95     (SELECT NAME_NAME, 8, TITLE_ID
96     FROM IMDB.PRODUCERS, Entry
97     WHERE TITLE_ID = Entry.id);
98
99 INSERT INTO DBLAB01.Person (id, name)
100     (SELECT a.NAME_ID, a.NAME_NAME
101     FROM IMDB.EDITORS a, DBLAB01.Entry
102     WHERE a.TITLE_ID = Entry.id);
103 INSERT INTO participates (person, task, entry)
104     (SELECT NAME_NAME, 9, TITLE_ID
105     FROM IMDB.EDITORS, Entry
106     WHERE TITLE_ID = Entry.id);

```