

Datenbanksysteme 2, 1. Übung

Lösungshinweise

Aufgabe 1.1 Ihr Lieblingsfilm in der Movie-Datenbank

Wenn im Folgenden von Filmen gesprochen wird, sind alle Einträge in der Tabelle MOVIE gemeint, also auch TV-Filme, Serien, etc.

- a) Finden Sie alle Informationen über Ihren Lieblingsfilm in der Datenbank heraus. Nutzen Sie auch Informationen über den Film, die nicht aus der Tabelle MOVIE stammen, sondern aus anderen Tabellen sich auf den entsprechenden Film beziehen.

Das Ergebnis hängt natürlich von Ihrem Lieblingsfilm ab. Beispiele sind:

```
SELECT * FROM moviedb.movie WHERE title like 'NAME%';  
  
SELECT * FROM moviedb.movie JOIN moviedb.plays ON (id = movie) JOIN  
    moviedb.person ON (player = id)  
WHERE title like 'NAME%';  
  
SELECT * FROM moviedb.movie JOIN moviedb.rating ON (id = movie)  
WHERE title like 'NAME%';
```

Aufgabe 1.2 Beispiel-Anfragen an die Movie-Datenbank

Formulieren Sie für die folgenden Aufgaben eine SQL-Anfrage und werten Sie sie auf der Datenbank aus. Falls mehrere Tabellen benötigt werden, empfiehlt sich die Formulierung der Anfrage mit explizitem JOIN in der FROM-Klausel.

Wenn im Folgenden von Filmen gesprochen wird, sind alle Einträge in der Tabelle MOVIE gemeint, also auch TV-Filme, Serien, etc.

- a) Suchen Sie alle Informationen aus der Tabelle MOVIE für Filme aus dem Jahr 1989.

```
select * from moviedb.movie where year = 1989;
```

7.443 Ergebnisse

- b) Finden Sie die Titel aller Filme sowie die darin auftretenden Personen (Charakter und Name des Schauspielers/der Schauspielerin), die entweder vor dem Jahr 1975 oder nach dem Jahr 2006 gedreht wurden. Verwenden Sie nicht UNION/UNION ALL!

```
select m.title, p.character, ps.name  
from moviedb.movie m join moviedb.plays p on (m.id = p.movie)  
join moviedb.person ps on (p.player = ps.id)  
where m.year < 1975 or m.year > 2006;
```

5.261.381 Ergebnisse

- c) Formulieren Sie die Anfrage aus Teil b) erneut; verwenden Sie dieses Mal UNION bzw. UNION ALL.

```
select m.title, p.character, ps.name
```

```

from moviedb.movie m
join moviedb.plays p on (m.id = p.movie)
join moviedb.person ps on (p.player = ps.id)
where m.year < 1975
union all
select m.title, p.character, ps.name
from moviedb.movie m
join moviedb.plays p on (m.id = p.movie)
join moviedb.person ps on (p.player = ps.id)
where m.year > 2006;

```

5.261.381 Ergebnisse

- d) Finden Sie die Titel aller Filme, für die es keinen Regisseur in der Datenbank gibt. Formulieren Sie die Anfrage dabei ohne die Verwendung von EXISTS.

```

select distinct title
from moviedb.movie m
left join moviedb.directs d on (m.id = d.movie)
where d.movie is null;

```

155.343 Ergebnisse, Duplikatfrei 155.343

- e) Formulieren Sie die Anfrage aus Teil d) erneut und verwenden Sie dabei EXISTS in sinnvoller Weise. Falls sie EXISTS nicht kennen, recherchieren Sie ggf. die Verwendung dieser Klausel.

```

select title
from moviedb.movie m
where not exists
    (select * from moviedb.directs d where d.movie = m.id);

```

- f) Geben sie alle Filme (ID und Titel) mit ihren Regisseuren (ID und Name) und Ratings aus. Sortieren Sie das Ergebnis nach Regisseur und pro Regisseur nach Rating absteigend.

```

SELECT m.id as mid, m.title, p.id as pid, p.name, r.rating
FROM moviedb.person p
JOIN moviedb.directs d ON (p.id = d.director)
JOIN moviedb.movie m ON (d.movie = m.id)
JOIN moviedb.rating r ON (m.id = r.movie)
ORDER BY p.name, r.rating DESC, m.id ASC;

```

352.383 Ergebnisse

Aufgabe 1.3 Beispiel-Anfragen an die Movie-Datenbank mit Gruppierungen

Formulieren Sie für die folgenden Aufgaben eine SQL-Anfrage und werten Sie sie auf der Datenbank aus. Falls mehrere Tabellen benötigt werden, empfiehlt sich die Formulierung der Anfrage mit explizitem JOIN in der FROM-Klausel.

Wenn im Folgenden von Filmen gesprochen wird, sind alle Einträge in der Tabelle MOVIE gemeint, also auch TV-Filme, Serien, etc.

- a) Berechnen Sie die Summe aller Budgets sowie das minimale und maximale Budget für alle Filme des Jahres 1989, bei denen das Budget in USD angegeben ist. (1 Anfrage!)

```
SELECT SUM(budget) AS SUMME, MIN(budget) AS MIN, MAX(budget) AS MAX
FROM moviedb.budget b
JOIN moviedb.movie m ON (b.movie = m.id)
WHERE year = 1989 AND currency_symbol LIKE '%USD%';
```

Ergebnis:

Summe = 1.633.121.442

Max = 70.000.000

Min = 0

- b) Finden Sie die Namen aller Personen, die sowohl Darsteller und als auch Regisseur sind.

```
SELECT DISTINCT p.id, p.name
FROM moviedb.plays pl
JOIN moviedb.person p ON (p.id = pl.player)
JOIN moviedb.directs di ON (di.director = pl.player);
```

Ergebnis: 122.777

- c) Finden Sie die Namen aller Personen, die entweder Darsteller oder Regisseur, aber nicht beides sind.

```
SELECT p.id, p.name
FROM moviedb.person p
Where p.id in (
    SELECT NVL(d.director, p.player) id
    FROM moviedb.directs d
    FULL OUTER JOIN moviedb.plays p ON (d.director = p.player)
    WHERE d.director IS NULL OR p.player IS NULL
);
```

Ergebnis: 2.794.238

- d) Ermitteln Sie alle Genres, die in Kinofilmen aus dem Jahr 1989 eingesetzt wurden zusammen mit der Anzahl von Kinofilmen, in denen sie in dem Jahr verwendet wurden. (Jedes Genre sollte nur einmal auftauchen!).

```
select g.genre AS genrename, count(*) AS anzahl
from moviedb.movie m join moviedb.genre g on (m.id = g.movie)
where m.year = 1989 and m.type = 'C'
group by g.genre;
```

Ergebnisse: 25

- e) Schränken Sie die Abfrage aus i) so ein, dass nur Genres aufgelistet werden, zu denen es in 1989 mindestens 100 Kinofilme gab.

```
select g.genre, count(*)
from moviedb.movie m join moviedb.genre g on (m.id = g.movie)
where m.year = 1989 and m.type = 'C'
group by g.genre
having count(*) > 100;
```

Ergebnisse: 15

- f) Die gleiche Anfrage wie in i) aber nach Häufigkeit absteigend sortiert.

```
select g.genre, count(*)
from moviedb.movie m join moviedb.genre g on (m.id = g.movie)
where m.year = 1989 and m.type = 'C'
group by g.genre
order by count(*) desc;
```

Ergebnisse: 25