

COMP2400/6240 - Relational Databases

SQL Sample Questions

Due date: TBD

Question 1

Unknown Marks

The relational database `moviedb` has the following database schema:

`MOVIE`(title, production_year, country, run_time, major_genre)
primary key : {title, production_year}

`PERSON`(id, first_name, last_name, year_born)
primary key : {id}

`AWARD`(award_name, institution, country)
primary key : {award_name}

`RESTRICTION_CATEGORY`(description, country)
primary key : {description, country}

`DIRECTOR`(id, title, production_year)
primary key : {id, production_year}
foreign keys : [title, production_year] \subseteq `MOVIE`[title, production_year]

`WRITER`(id, title, production_year)
primary key : {id, production_year}
foreign keys : [title, production_year] \subseteq `MOVIE`[title, production_year]
[id] \subseteq `PERSON`[id]

`CREW`(id, title, production_year, contribution)
primary key : {id, title, production_year}
foreign keys : [title, production_year] \subseteq `MOVIE`[title, production_year]
[id] \subseteq `PERSON`[id]

`SCENE`(title, production_year, scene_no, description)
primary key : {title, production_year, scene_no}
foreign keys : [title, production_year] \subseteq `MOVIE`[title, production_year]

`ROLE`(id, title, production_year, description, credits)
primary key : {title, production_year, description}
foreign keys : [title, production_year] \subseteq `MOVIE`[title, production_year]
[id] \subseteq `PERSON`[id]

`RESTRICTION`(title, production_year, description, country)
primary key : {title, production_year, description, country}
foreign keys : [title, production_year] \subseteq `MOVIE`[title, production_year]
[description, country] \subseteq `RESTRICTION_CATEGORY`[description, country]

`APPEARANCE`(title, production_year, description, scene_no)
primary key : {title, production_year, description, scene_no}
foreign keys : [title, production_year, scene_no] \subseteq `SCENE`[title, production_year, scene_no]
[title, production_year, description] \subseteq `ROLE`[title, production_year, description]

MOVIE_AWARD(title, production_year, award_name, year_of_award, category, result)
 primary key : {title, production_year, award_name, year_of_award, category}
 foreign keys : [title, production_year] \subseteq MOVIE[title, production_year]
 [award_name] \subseteq AWARD[award_name]

CREW_AWARD(id, title, production_year, award_name, year_of_award, category, result)
 primary key : {id, title, production_year, award_name, year_of_award, category}
 foreign keys : [id, title, production_year] \subseteq CREW[id, title, production_year]
 [award_name] \subseteq AWARD[award_name]

DIRECTOR_AWARD(title, production_year, award_name, year_of_award, category, result)
 primary key : {title, production_year, award_name, year_of_award, category}
 foreign keys : [title, production_year] \subseteq DIRECTOR[title, production_year]
 [award_name] \subseteq AWARD[award_name]

WRITER_AWARD(id, title, production_year, award_name, year_of_award, category, result)
 primary key : {id, title, production_year, award_name, year_of_award, category}
 foreign keys : [id, title, production_year] \subseteq WRITER[id, title, production_year]
 [award_name] \subseteq AWARD[award_name]

ACTOR_AWARD(title, production_year, description, award_name, year_of_award, category, result)
 primary key : {title, production_year, description, award_name, year_of_award, category}
 foreign keys : [award_name] \subseteq AWARD[award_name]
 [title, production_year, description] \subseteq ROLE[title, production_year, description]

There are five different categories of actor awards. A movie can only win one or more actor awards. Your task is to answer the following SQL query that may contain one or more *single myqueries.sql*.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

- 1.1 Find all the movies produced in Australia. List the titles and production years.

```
SELECT title, production_year
FROM movie
WHERE lower(country) = 'australia';
```

- 1.2 How many persons are there in this database? List that number.

```
SELECT count(id)
FROM person;
```

- 1.3 Which writers have won a writer award in 1994? List the ids, the first and last names of the writers along with the award names.

```
SELECT w.id, first_name, last_name, award_name
FROM writer_award AS w INNER JOIN person AS p
ON w.id = p.id
WHERE year_of_award = 1994 AND lower(result) = 'won';
```

- 1.4 Which directors have directed crime movies (i.e., the major genre of the movie is crime)? List the directors' ids and the titles and production years of these crime movies.

```
SELECT d.id, m.title, m.production_year
FROM director d NATURAL JOIN movie m
WHERE m.major_genre = 'crime';
```

- 1.5 How many different crew members have worked on at least one movie produced in 1993? List that number.

```
SELECT COUNT(DISTINCT id)
FROM crew
WHERE production_year = 1993;
```

- 1.6 List all titles of movies produced in 1995 along with the corresponding number of scenes in each movie. Order your result in the ascending order of the number of scenes in each movie.

```
SELECT title, count(*)
FROM scene
WHERE production_year = 1995
GROUP BY title, production_year
ORDER BY count(*) ASC;
```

- 1.7 Which year was the youngest director (i.e., the youngest person in the database who has ever directed a movie) born in? List that year.

```
SELECT MAX(year_born)
FROM person NATURAL JOIN director;
```

- 1.8 How many directors are there in this database who have never written a movie? List that number.

```
SELECT COUNT(*)
FROM (SELECT id
      FROM director
      EXCEPT
      SELECT id
      FROM writer) AS dnm;
```

- 1.9 What is the maximum number of scenes in any movie? List that number.

```
WITH MovieSceneNum AS (SELECT COUNT(*) AS SceneNum
                       FROM scene
                       GROUP BY title, production_year)
SELECT MAX(SceneNum)
FROM MovieSceneNum;
```

- 1.10 Who has/have won a most recent writer award? List their id(s).

```
SELECT DISTINCT wd.id
FROM writer_award wd
WHERE wd.year_of_award = (SELECT MAX(year_of_award)
                        FROM writer_award
                        WHERE lower(result)='won');
```

- 1.11 Find all the movies which have won exactly two movie awards. List the titles and production years of these movies.

```
SELECT title, production_year
FROM movie_award
WHERE lower(result) = 'won'
GROUP BY title, production_year
HAVING count(*) = 2;
```

- 1.12 Who have won both writer and director awards for the same movie? Provide their ids, and the titles and production years of the corresponding movies.

```
SELECT w1.id, w1.title, w1.production_year
FROM writer_award w1
WHERE lower(w1.result) = 'won'
INTERSECT
SELECT d1.id, d1.title, d1.production_year
FROM director_award da NATURAL JOIN director AS d1
WHERE lower(da.result) = 'won';
```

- 1.13 Who have been nominated for an actor award at least twice but never won? List their ids, the first and last names, and order them in the ascending order of their last names.

```
SELECT id, first_name, last_name
FROM person
WHERE id IN (SELECT id
              FROM actor_award natural join role
              WHERE lower(result) <> 'won'
              GROUP BY id
              HAVING count(*) > 1
              EXCEPT
              SELECT id
              FROM actor_award NATURAL JOIN role
              WHERE
ORDER BY last_name ASC
```

- 1.14 Which crew members have the ids of the crew members.

```
WITH movie_wins AS (SELECT title, production_year, COUNT(*)
                     FROM movie_award
                     WHERE lower(result) = 'won'
                     GROUP BY title, production_year)
SELECT DISTINCT id
FROM crew c NATURAL JOIN movie_wins
WHERE wins = (SELECT MAX(wins)
              FROM movie_wins);
```

- 1.15 Who is/are the most productive director(s) (i.e., who has/have directed the greatest number of movies)? List their id(s).

```
WITH DirectorNum AS (SELECT id, COUNT(*) AS num
                     FROM director
                     GROUP BY id)
SELECT id
FROM DirectorNum dn
WHERE dn.num = (SELECT MAX(num)
               FROM DirectorNum);
```