COMP2400/6240 - Relational Databases

Sample SQL Questions

Spoiler: Example solutions follow on page 3. Try to solve the questions yourself before you look at the solutions!

The Movie Database

These sample SQL questions refer to the `moviedb` database, which is available in the PostgreSQL DBMS in the CSIT lab environment (including via `partch`). To connect to the `moviedb` database, add its name to the command when you start the `psql` client in a terminal, like this:

```
u1234567@n11XltYZ:~$ psql moviedb
```

The schema of the `moviedb` databse is shown in Figure 1 below. Remember that you can also view the database schema using the `\dt` command in `psql` when you are connected to the database.

Questions

Your task is to write SQL queries that answer the following questions. For each question, your answer must be a *single SQL query* (that may contain subqueries).

1. How many unique persons are in the database?
2. How many different first names of persons are there in database?
3. List all distinct first names of persons in the database, along with for each the number of persons that have that first name, ordered by that number.
4. How many persons share the most common first name?
5. What is the most common first name?
6. Find all the movies made in Australia. List the titles and production years of these Australian movies.
7. Which directors have directed crime movies (i.e., movies whose major genre is 'crime')? List the directors' first and last names, and the titles and production years of these crime movies.
8. Which directors have directed crime movies (i.e., movies whose major genre is 'crime')? List *only* the directors' first and last names, *without repetition*.
9. List all movies (title and production year) for which any award, of any kind, has been won, along with the total number of awards won for each movie, ordered by the number of awards (most award-winning movie first).
10. List all movies (title and production year) that were written and directed by one (and only one!) person.
11. List all movies (title and production year) that had an 'R' restriction in the USA but did *not* have an 'R' restriction in Australia.
12. What is the maximum total number of persons who worked (in any capacity, i.e., either actor, crew, director or writer) on any movie?
13. Find all the writers who have only written movies with at least one other writer (i.e., have never written a movie on their own). List their ids, first and last names.

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` : {title, production_year}
`foreign keys` : [title, production_year] ⊆ MOVIE[title, production_year]
[id] ⊆ PERSON[id]

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

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

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

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

RESTRICTION(title, production_year, description, country)
`primary key` : {title, production_year, description, country}
`foreign keys` : [title, production_year] ⊆ MOVIE[title, production_year]
[description, country] ⊆ 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] ⊆ SCENE[title, production_year, scene_no]
[title, production_year, description] ⊆ 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] ⊆ MOVIE[title, production_year]
[award_name] ⊆ 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] ⊆ CREW[id, title, production_year]
[award_name] ⊆ 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] ⊆ DIRECTOR[title, production_year]
[award_name] ⊆ 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] ⊆ WRITER[id, title, production_year]
[award_name] ⊆ 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` : [title, production_year, description] ⊆ ROLE[title, production_year, description]
[award_name] ⊆ AWARD[award_name]

Figure 1: The `moviedb` schema.

Example solutions

1. How many unique persons are in the database?
   Solution:

   ```
   select
       count(*)
   from
       person;
   ```

   OR

   ```
   select
       count(id)
   from
       person;
   ```

   Why do these two queries give the same answer?

2. How many different first names of persons are there in database?

   Solution:
   First, to list the all person's first names (in alphabetical order):

   ```
   select
       first_name
   from
       person
   order by
       first_name;
   ```

   To list the all *distinct* first names:

   ```
   select distinct
       first_name
   from
       person
   order by
       first_name;
   ```

   To count *how many* distinct first names:

   ```
   select
       count(distinct first_name)
   from
       person;
   ```

   What will

   ```
   select
       count(first_name)
   from
       person;
   ```

   produce? Do you understand why?

3. List all distinct first names of persons in the database, along with for each the number of persons that have that first name, ordered by that number.
   Solution:

   ```
   select
       first_name,
       count(*) as number_with_name
   from
       person
   group by
       first_name
   order by
       number_with_name asc;
   ```

4. How many persons share the most common first name?
   Solution:

```
select
    max(number_with_name)
from (
    -- use the query from above as a table subquery
    select
        first_name,
        count(*) as number_with_name
    from
        person
    group by
        first_name
);
```

5. What is the most common first name?
   Solution:

```
select
    first_name
from (
    -- like above, use the "name, count" query as a
    -- table subquery
    select
        first_name,
        count(*) as number_with_name
    from
        person
    group by
        first_name
)
where
    -- but here we need a restriction to the row(s) that
    -- have a count equal to the maximum
    number_with_name = (
        -- so we need the query to get the maximum as a
        -- correlated subquery:
        select
            max(number_with_name)
        from (
            select
                first_name,
                count(*) as number_with_name
            from
                person
            group by
                first_name
        )
    );
```

   which can also be written

```
-- the "with" keyword allows us to define a named subquery
-- once, and then use in several places within our query:
with name_count as (
    select
        first_name,
        count(*) as number_with_name
    from
        person
    group by
        first_name
)
select
    first_name
```

```
from
    name_count  -- use it here
where
    number_with_name = (
        select
            max(number_with_name)
        from
            name_count  -- and here
    );
```

6. Find all the movies made in Australia. List the titles and production years of these Australian movies.

   Solution:

```
select
    title,
    production_year
from
    movie
where
    country = 'Australia';
```

   For case-insensitive matching:

```
select
    title,
    production_year
from
    movie
where
    lower(country) = 'australia';
```

7. Which directors have directed crime movies (i.e., movies whose major genre is 'crime')? List the directors' first and last names, and the titles and production years of these crime movies.

```
select
    first_name,
    last_name,
    title,
    production_year
from
    -- join person and director to get first_name and
    -- last_name of directors
    person
    join director using (id)
    -- join with movie which has attribute major_genre
    join movie using (title, production_year)
where
    -- restrict to 'crime' genre
    major_genre = 'crime';
```

8. Which directors have directed crime movies (i.e., movies whose major genre is 'crime')? List *only* the directors' first and last names, *without repetition*.

```
select distinct
    first_name,
    last_name
from
    person
    join director using (id)
    join movie using (title, production_year)
where
    major_genre = 'crime';
```

(Try the previous and this query for some of the other genres, e.g., 'drama' or 'action', to see the difference.)

9. List all movies (title and production year) for which any award, of any kind, has been won, along with the total number of awards won for each movie, ordered by the number of awards (most award-winning movie first).

```
select title,
       production_year,
       (select count(*)
        from   movie_award
        where  movie.title = movie_award.title
          and  movie.production_year = movie_award.production_year
          and  lower(movie_award.result) = 'won'
       ) +
       (select count(*)
        from   actor_award
        where  movie.title = actor_award.title
          and  movie.production_year = actor_award.production_year
          and  lower(actor_award.result) = 'won'
       ) +
       (select count(*)
        from   crew_award
        where  movie.title = crew_award.title
          and  movie.production_year = crew_award.production_year
          and  lower(crew_award.result) = 'won'
       ) +
       (select count(*)
        from   director_award
        where  movie.title = director_award.title
          and  movie.production_year = director_award.production_year
          and  lower(director_award.result) = 'won'
       ) +
       (select count(*)
        from   writer_award
        where  movie.title = writer_award.title
          and  movie.production_year = writer_award.production_year
          and  lower(writer_award.result) = 'won'
       ) as total_awards
from   movie
where  total_awards > 0
order by total_awards desc;
```

10. List movies (title and production year) that were written and directed by one (and only one!) person.

```
select title,
       production_year
  from writer
       join director using (id, title, production_year)
 where (select count(*)
          from writer as w2
         where w2.title = writer.title
          and  w2.production_year = writer.production_year
       ) = 1
   and (select count(*)
          from director as d2
         where d2.title = director.title
           and d2.production_year = director.production_year
       ) = 1;
```

The counting condition (or some equivalent) is necessary, as there can be movies with more than one writer, for which one of the writers is also the director.

11. List movies (title and production year) that had an 'R' restriction in the USA but did not have an 'R' restriction in Australia.

```
-- movies that have a different restriction (not R) in Australia
select title,
       production_year
  from      restriction as r1
       join restriction as r2 using (title, production_year)
 where r1.country = 'USA'
   and r1.description = 'R'
   and r2.country = 'Australia'
   and r2.description <> 'R'
union
-- movies that have NO restriction in Australia
select title,
       production_year
  from restriction as r1
 where r1.country = 'USA'
   and r1.description = 'R'
   and not exists (
               select *
                 from restriction as r2
                where r1.title = r2.title
                  and r1.production_year = r2.production_year
                  and r2.country = 'Australia'
           );
```

Note: The answer should include both movies that had a different restriction in Australia and movies that had no restriction in Australia.

Can you write the query in a different way that gives the same result?

12. What is the maximum total number of persons who worked (in any capacity, i.e., either actor, crew, director or writer) on any movie?

```
select max(total)
from   ( select title,
                production_year,
                count(*) as total
         from (  select distinct title, production_year, id from role
                union
                 select distinct title, production_year, id from crew
                union
                 select distinct title, production_year, id from director
                union
                 select distinct title, production_year, id from writer
              ) as worked_on
         group by title, production_year
       ) worked_on_count;
```

Note that some movies have the same person doing multiple jobs (e.g., writer and director, director and producer, actor in two roles, etc); we want the maximum number of distinct persons here.

13. Find all the writers who have only written movies with at least one other writer (i.e., have never written a movie on their own). List their ids, first and last names.

```
with single_writer_movies as (select title, production_year
                                from writer
                               group by title, production_year
                              having count(id) = 1),
     single_writers as (select id
                          from writer natural join single_writer_movies),
     co_writers as (select id
                      from writer
                    except
                    select id
                      from single_writers)
select id, first_name, last_name
  from co_writers natural join person;
```