COMP3311 19T3 Database Systems

Assignment 2

SQL, Views, PLpgSQL

Last updated: **Thursday 17th October 9:56pm**
Most recent changes are shown in red ... older changes are shown in brown.

Aims

The aims of this assignment are to:

- populate an RDBMS with a real dataset, and analyse the data
- write SQL queries, views and PLpgSQL functions to solve information requests

Admin

Submission:	use the command `give cs3311 ass2 ass2.sql` or use Webcms3
Deadline:	**Sunday 27th October 23:59**
Late Penalty:	Late submissions will have marks deducted from the maximum achievable mark at the rate of 0.7% of the total mark *per hour* that they are late (i.e., around 17% per day).

Description

In this assignment, you will work with a copy of the Internet Movie Database (aka IMDB). This database has information about movies, TV series, actors, directors, etc. The database for the assignment is actually a *very small* subset of the complete IMDB database. It only deals with works from 2018 and 2019, removes many TV series, and deals with a subset of people involved in movies, so don't expect to find your favourite old actor or movie. The actual database is over 50GB.

Some of the terminology IMDB uses may require some explanation:

Names

Since the database deals with a wide variety of humans, animals and animated characters that appear in video artifacts, the term "people" isn't broad enough. IMDB uses the term "Name" to cover all of the entities that appear in video artifacts. An example of where this term is in common use "They got some big **names** for the new movie".

Titles

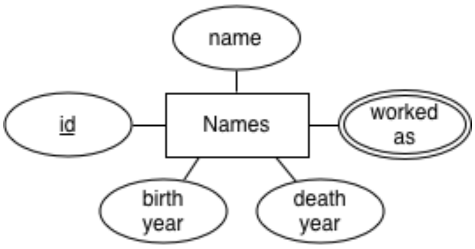
Since the database deals with different kinds of video artifacts (movies, TV series, documentaries, etc.) it uses the term "Title" to cover all of them. An example of where this term is in common use "The new **title** by James Cameron is hot".

Aliases

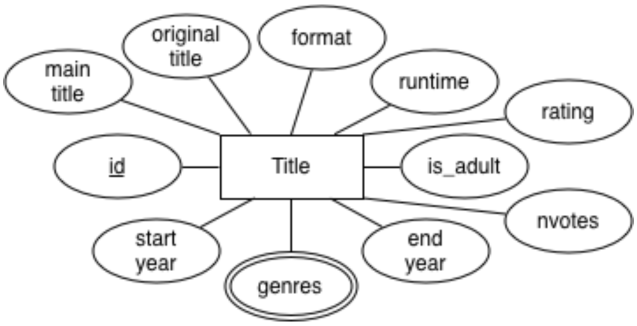
Movies (and other media) and released in different forms in different regions of the world. Some versions are cut, to fit with local laws. Others are dubbed or subtitled, to fit the local language. The title is also often changed, and to a phrase with quite a different meaning to the original. The various versions of a video work are called "Aliases".

Schema

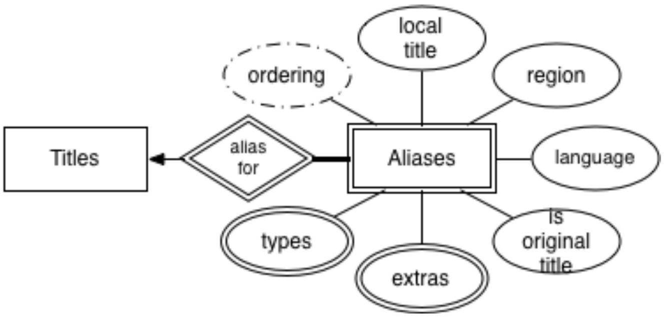
Names have the following ER model



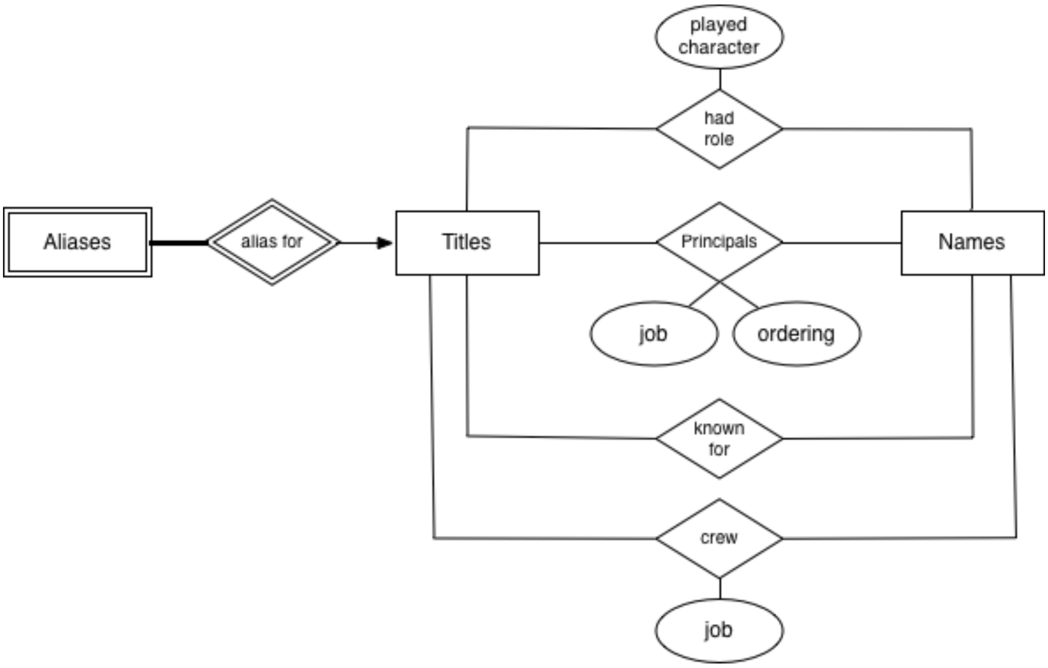
Titles have the following ER model



Aliases for titles have the following ER model



The above entities are linked together as follows



Getting Started

Make sure you read this entire specification thoroughly, then create and load the database using the commands:

```
grieg% dropdb a2
grieg% createdb a2
grieg% psql a2 -f /home/cs3311/web/19T3/assignments/ass2/files/imdb.sql
```

If you're working on your own machine with PostgreSQL installed, you'll need to download a copy of the dump file from:

```
/home/cs3311/web/19T3/assignments/ass2/files/imdb.sql
```

Note: use right mouse click and "Save Link As" for the above, or you might get 37MB of text dumped into your browser.

Once you've loaded the data, start a `psql` session explore the database, using the `psql` meta-commands (e.g. `\d`) and running some SQL queries.

It is useful to create a separate directory just for this assignment, and keep all of the assignment-related files there (e.g. `ass2.sql`).

We have provided a template for the `ass2.sql` file that you submit for this assignment:

```
/home/cs3311/web/19T3/assignments/ass2/files/ass2.sql
```

Note that when solving the problems above, you can define as many auxiliary views and functions as you want. These must also be included in the `ass2.sql` file you submit, and must be ordered so that the file can load into a fresh database in one pass (i.e. no forward references).

A readable copy of the schema is available in the file:

```
/home/cs3311/web/19T3/assignments/ass2/files/schema.sql
```

You do *not* need to load this schema into your database. It is already a part of `imdb.sql`.

For you to test your views and function, we have also provided a test harness, implemented as a collection of views and PLpgSQL functions. This is available in the file:

```
/home/cs3311/web/19T3/assignments/ass2/files/check.sql
```

You should also load this into your database each time you create the database.

```
$ psql a2 -f check.sql
```

The views and functions in `check.sql` are all called ass2_*XXXX*. You should avoid naming any of your views or functions like this.

Once `check.sql` is loaded, you can test individual queries, or test them all as follows:

```
grieg% psql a2
...
a2=# select check_q1();
...
a2=# select check_all();
```

Exercises

Below are ten information requests on the Internet Movie Database for you to answer ...

Notes:

- in the queries, references to `title` mean `Titles.main_title`
- in most cases, the order of results doesn't matter; the testing code will use `order by` to force a specific order
- Q3 is the exception, where you are required to follow the specified order
- queries should not take more than 3 seconds to run; queries that take longer to run will be penalised

Provide SQL or PLpgSQL code for each of the following:

1. Which movies are more than 6 hours long?
   (a 6 hour movie is a frightening thought ...)

   ```
   create or replace view Q1(title) as ...
   ```

   [Expected result] (1 mark)

2. What different formats are there in Titles, and how many of each?

   ```

   create or replace view Q2(format, ntitles) as ...
   ```

   [Expected result] (1 mark)

3. What are the top 10 movies that received more than 1000 votes?

   - order the results by descending order on rating and ascending order on title
   - then take just the first 10 of the tuples, based on this ordering

   ```
   create or replace view Q3(title, rating, nvotes) as ...
   ```

   [Expected result] (1 mark)

4. What are the top-rating TV series and how many episodes did each have?

   - the rating is based on the overall rating for the series
   - the ratings of individual episodes are not relevant for this
   - "TV series" includes both regular TV series and TV mini-series

   ```
   create or replace view Q4(title, nepisodes) as ...
   ```

   [Expected result] (2 marks)

5. Which movie was released in the most languages? And how many different languages?

   - most languages = maximum number of distinct languages
   - if more than one movie has the same maximum number, return all of them

   ```
   create or replace view Q5(title,nlanguages) as ...
   ```

   [Expected result] (2 marks)

6. Which actor has the highest average rating in movies that they're known for?

   - they must been known for at least two movies that have been rated; no one-hit wonders
   - the rating is for the movie, not the person who's known for the movie
   - the person must have *worked as* an actor, but may not have acted in a movie that they're known for
   - if more than one actor has the same highest rating, return all of them

   ```
   create or replace view Q6(name) as ...
   ```

[Expected result] (2 marks)

7. For each movie with more than 3 genres, show the movie title and a comma-separated list of the genres
   - the list of genres must be in alphabetical order
   - hint: use `string_agg()`; see the PostgreSQL documentation, Chapter 9

   ```
   create or replace view Q7(title, genres) as ...
   ```

   [Expected result] (3 marks)

8. Get the names of all people who had both actor and crew roles on the same movie

   ```
   create or replace view Q8(name) as ...
   ```

   [Expected result] (2 marks)

9. Who was the youngest person to have an acting role in a movie, and how old were they when the movie started?
   - youngest is determined by their age at the time (year) the movie started shooting
   - if more than one person is equal youngest, return all of them

   ```
   create or replace view Q9(name, age) as ...
   ```

   [Expected result] (2 marks)

10. Write a PLpgSQL function that, given part of a title, shows the full title and the total size of the cast and crew
    - we consider all formats in this question, not just movies
    - total size = number of distinct people (actors, principals, crew)
    - use `ilike` to match the title to the supplied string
    - if no matching item exists, return the string "No matching titles"
    - if one or more item exists, display "TITLE has COUNT cast and crew"
    - given the data in the database, many of the counts are just 1

    ```
    create or replace function Q10(partial_title text) returns setof text ...
    ```

    [Expected result] (5 marks)

Assessment

This assignment is worth a total of **21 marks**. It will later be scaled to 12 percent for the course as described in the course outline.

Your submission (in a file called `ass2.sql`) will be auto-marked to check:

- whether it is syntactically correct;
- whether each query produces the correct results.

Queries are not worth equal marks.

If we have to fix errors in your solution before it will load, you will incur a 8 (out of total 21) mark "penalty". If your view names or attribute names are different from the names specified above, you will incur a 8 mark "administrative penalty".

Testing

Reminder: before you submit, ensure that your solution (`ass2.sql`) will work correctly in our test environment running on `grieg`:

```
grieg% dropdb a2
grieg% createdb a2
grieg% psql a2 -f /home/cs3311/web/19T3/assignments/ass2/files/files/imdb.sql
... will produce notices, and will have no errors ...
grieg% psql a2 -f ass2.sql
... will produce notices, but should have no errors ...
grieg% psql a2 -f check.sql
... will produce notices, but should have no errors ...
grieg% psql a2
... test your solution as noted above ...
```

In the real testing, we will repeat the above process using a slightly different version of the IMDB database: same schema, same views and functions, but with different data.

Have fun, *jas* and Hayden