CentraleSupélec

SACLAY CAMPUS, COMPUTER SCIENCE DEPARTMENT
2019-2020

Models and Systems for Big Data Management
Relational Databases & Structured Query Language

The purpose of this practical work is to work with a relational database. We will use a dataset that consists of a collection of movies we will store in a relational database managed by PostgreSQL [1] server. SQL Developer [2] will be used as a client to connect to PostgreSQL server and to submit `sql` queries.

1 POSTGRESQL AND SQL DEVELOPER

1. The Postgresql server is already running and waiting for client connection requests on the port number gven by default xxxx (5432 for macOs).

2. Launch SQL Developer client from a terminal using the command `sqldeveloper`, open preferences in ORACLE SQL Developer tools bar, then in database section choose third party JDBC driver, click "Add Entry" button, pick the PostgreSQL JDBC Driver (to download first `postgresql-42.2.2.jar`), [3] then click OK to make it enabled.

3. Now you can create a connection to Postgresql server to access to `postgres` database by specifying:
   a) a name for the connection,
   b) `localhost` as the location of the server,
   c) server port number,
   d) `postgres` as user and password.

   The work sheet enables you to execute SQL queries by running all the script or running only the selected statement. The script output displays results.

[1] `https://www.postgresql.org`
[2] `http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html`
[3] PostgreSQL JDBC 4.2 Driver, 42.2.2 `https://jdbc.postgresql.org/download.html`

2 DOWNLOAD DATA, CREATE SCHEMA, INSERT DATA

1. Execute the following command to import data from `artists_movies.sql` file (available on Edunao) in the database in the work sheet window: `@/PATH/artists_movies.sql` where `PATH` is the path to access to the downloaded file.

2. Which tables are created, how many tuples in each table ? What is the meaning of each tuple ? What about the 1^{st}, 2^{nd} and 3^{th} normal forms.
   Deux tables sont créées :
   `movies` et `moviesReferences`. `artists` est en 2NF et 3NF (1NF par définition) si on considère $Id \rightarrow last\_name, first\_name, birth\_date$.
   `moviesReferences` n'est pas en 2NF si on considère $id \rightarrow title, year, genre, summary$ et $actors\_id, Id \rightarrow actors\_role$.

3. Execute the following SQL queries in the following order to create new tables and to insert data:

   a) Create `movies` table:

   ```
   CREATE TABLE movies(
     id VARCHAR(8) PRIMARY KEY,
     title VARCHAR(35),
     year INT,
     genre VARCHAR(15),
     summary VARCHAR(4406),
     country VARCHAR(3)
   ) ;
   ```

   b) Insert movies referenced in `movies` table from `moviesReferences` table:

   ```
   INSERT INTO movies (
     SELECT  id, title , year , genre , summary , country
     FROM moviesreferences
     GROUP BY id, title , year , genre , summary , country
   ) ;
   ```

   c) Create `movies_directors` table:

   ```
   CREATE TABLE movies_directors (
     director_id  VARCHAR(10), movie_id VARCHAR(8) ,
     PRIMARY KEY(director_id, movie_id),
     FOREIGN KEY(director_id )  references artists(id),
     FOREIGN KEY(movie_id )   references movies (id)
   );
   ```

   d) Insert data in `movies_directors` table using directors referenced in `moviesreferences` table:

   ```
   INSERT INTO movies_directors (
     SELECT distinct director_id, id FROM moviesreferences
   ```

```
);
```

e) Create `movies_actors` table:

```
CREATE TABLE movies_actors (
  actor_id VARCHAR(10), actor_role VARCHAR(30), movie_id VARCHAR(8) ,
  unique (actor_id, movie_id),
  foreign key(actor_id )  references artists(id),
  foreign key(movie_id )   references movies (id)
);
```

f) Insert data in `movies_actors` table using actors referenced in `moviesReferences` table:

```
INSERT INTO movies_actors (
  SELECT DISTINCT actors_id, actors_role, id FROM moviesreferences
);
```

g) Remove `moviesReferences` table:

```
DROP TABLE moviesReferences;
```

4. Analyse the final schema/tables and give the underlying Entity Association model. What about the 1st, 2nd and 3th normal forms.

   Au final nous avons une entité `movies`, une entité `artists`, et les deux associations `movies_directors` *-*) et `movies_actors` (*-*). Chaque entité et association a des attributs.

   En analysant le code de création des tables, en plus des contraintes d'unicité (`primary key` et `unique`), les associations sont exprimés par les clés étrangères: un acteur ou un directeur est désigné avec une `foreign key` qui référence l'identifiant `artists`. De même pour le film.

   On ne peut pas insérer un tuple dans `movies_actors` si le film ou l'artiste n'existent pas. De même on ne peut pas supprimer de `movies` un film référencé par une clé étrangère (dans `movies_directors`ouy `movies_actors`). Essayer par ex.
   `INSERT INTO movies_actors ('1', 'betty', '1')`
   `INSERT INTO movies_actors ('random', 'betty', '1').`

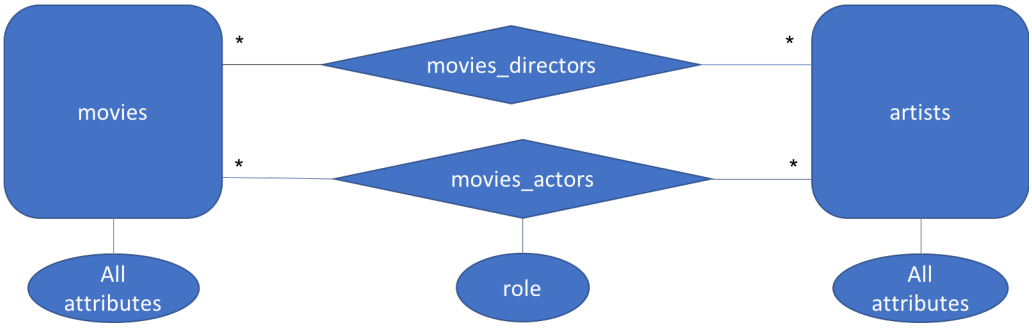
   Oui le schéma est en 2ème normale (pas de dépendance partielle à la clé) et 3ème forme normale (pas de dépendance transitive à la clé)

3 Querying data

Write the following SQL queries:

1. What are the genres of the movies? how many distinct genres of movies?

```
select distinct genre from movies;
select count(distinct genre) from movies;
```



2. What are the titles and the years of French movies released from1960 to 2010?

```
select title, year from movies where country='FR'
and year >=1960 and year <=2010;
```

3. What are the names and the roles of the actors who played in 'The Dark Knight Rises' movie? Give at least two different equivalent queries (one of them should use a sub-query).

```
select a.first_name, a.last_name, ma.actor_role
from movies m , artists a, movies_actors ma
where m.title='The Dark Knight Rises'
and m.id=ma.movie_id and a.id=ma.actor_id;
OR (more optimized query)
select a.first_name, a.last_name, ma.actor_role
from  artists a, movies_actors ma
where a.id= ma.actor_id and
ma.movie_id in (select id from movies where title='The Dark Knight Rises');
```

4. What are the names and the roles of the actors who directed and played in a movie (the same movie)?

```
select  a.first_name, a.last_name, ma.actor_role
from movies_directors md, movies_actors ma, artists a
where md.movie_id=ma.movie_id and
md.director_id = ma.actor_id and a.id=ma.actor_id;
```

5. What is the number of movies by country? by country and year. Retrieve the answers in different kind of orders.

```
select country, count(*)  from movies group by country;
select country, year, count(*) from movies group by country, year
order by year asc;
select country, year, count(*) from movies group by country, year
order by year, country desc;
```

6. What is the number of movies by artist? First, retrieve only the identifier's artist then the first/last name's artist.

```
select  count(*) n, u.id, u.name1, u.name2 from (
(select  a.id as id, a.first_name as name1, a.last_name as name2
from  movies_actors ma, artists a
where a.id=ma.actor_id)
union all
(select a.id as id, a.first_name as name1, a.last_name as name2
from movies_directors md, artists a
where md.director_id = a.id)
) u group by u.id, u.name1, u.name2;
```

7. What is the name of the actor who played in the most of movies?

```
select id, first_name, last_name, count(*)  from movies_actors ma, artists a
where ma.actor_id=a.id
group by id, first_name, last_name having count(*) =
(Select Max(n) from (select count(*) n from movies_actors group by actor_id) as s);
```

8. What are the artists who are not associated to any movie as an actor?as a director? as neither?

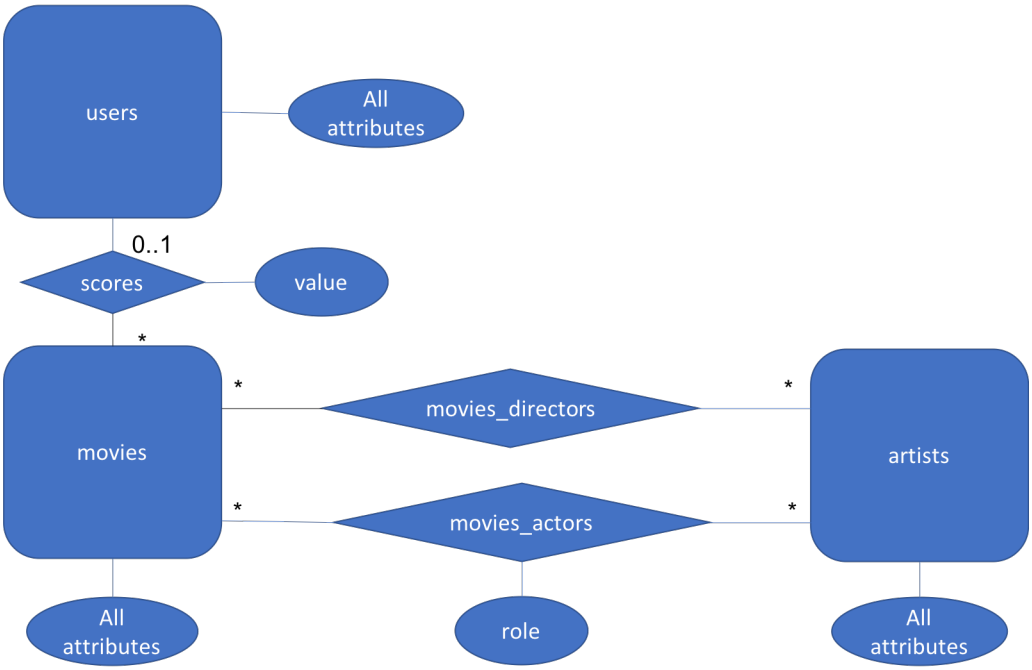
```
select id, first_name, last_name from artists where id
not in (select actor_id from movies_actors WHERE actor_id is not null)
or  id
not in (select director_id from movies_directors WHERE director_id is not null);
```

4 Enriching data

Insert new movie and artists to enrich your database using the following Wikipedia links: `https://en.wikipedia.org/wiki/The_Notebook`

On ne peut pas insérer un tuple dans `movies_actors` si le film ou l'artiste n'existent pas. De même on ne peut pas supprimer de `movies` un film référencé par une clé étrangère (dans `movies_directors` ou `movies_actors`). Une violation de contrainte est signalé par le SGBD par ex.

```
INSERT INTO movies_actors ('1', 'betty', '1')
```



5 CHANGING THE SCHEMA

Complete the EA model to take into account users who assign a score to a movie (integer between 1 and 5). Give the corresponding SQL schema.

Le schéma existant n'est pas remis en question, on rajoute les deux tables suivantes:

```
users(id PRIMARY KEY, all attributes ...)
scores((user_id FOREIGN KEY, movie_id FOREIGN KEY) PRIMARY KEY, value)
```