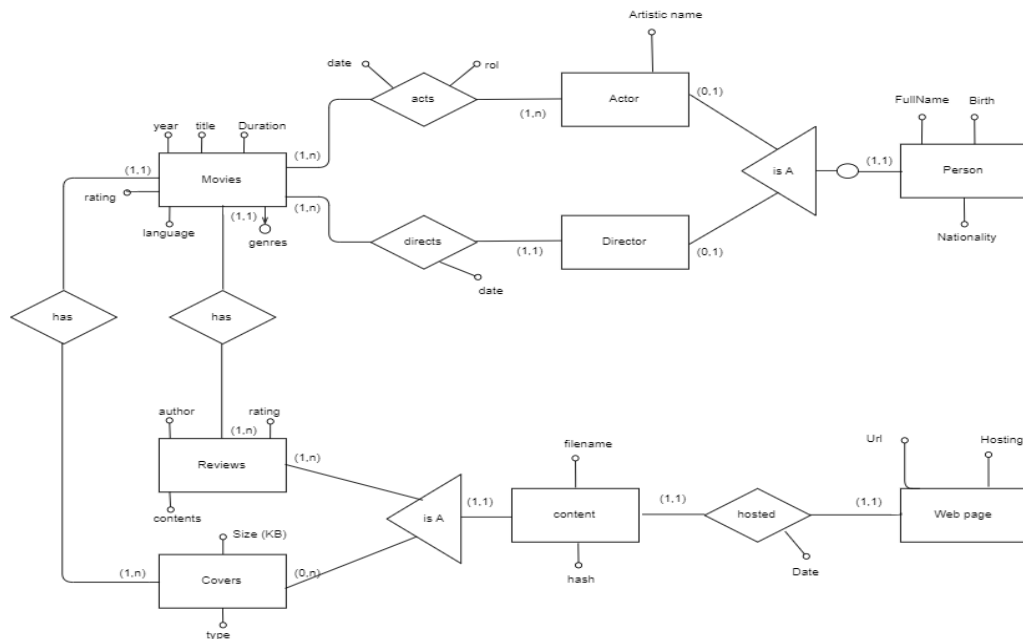


NeatFlint The Internet Movies database



We want to build a database for our new Start-up Neatflint (our motto is "In the beginning all we got was a flint"). The database will contain info about Actors, Actresses and Directors and they movies they have been involved with. For that purpose, you're given an EER model to begin with:



Given this EER model you are asked to provide:

- A textual description of the model, so the board of directors & stakeholders will be able to understand what you are building is what they asked for.
- Provide a set of primary keys for every entity looking to use as many natural keys as possible. Also provide, for every entity, any other candidate keys that may exist.
- Construct a Relational Model considering the following premises:
 - o An actor/actress can play a main role or a secondary one.
 - o Covers are downloadable files (from a web page) representing different artistic concepts (covers, posters etc.) and can exist in different formats and layouts.
 - o Web pages are different kind of projects like forums, journals, blogs and movie libraries.
- Provide a brief data dictionary with the domain of every attribute and constraints of interest.

Now proceed to Part 2...

NeatFlint The Internet Movies database (Part2)

We want to build a database for our new Start-up Neatflint (our motto is "In the beginning all we got was a flint"). The database will contain info about Actors, Actresses and Directors and the movies they have been involved with. For that purpose, you were given an EER model to begin with and asked to design a Relational Model by using the rules of conversion seen at class.

Now we are providing you're a set of raw data files in CSV format that resemble the model you are constructing from the E/R diagram as close as possible, your mission is accomplish the following tasks:

1. We assume your relational model is already created with the all tables and constraints (table constraints, primary and foreign key constraints) in place
2. Inspect the files to devise an interim schema name suitable for loading all files into interim tables: **use the prefix 'import_' in all schema names you create to separate interim tables from the final ones.** i.e. 'import_actors' for the interim table holding actor data.
3. Load all the data into the interim tables using COPY and compare those table schemas with the final ones (the ones your created for the E/R model): **Are the schemes compatible to each other? Does the data fit all the existing constraints in your relational model?** Describe what changes, either to your relational model or just to imported data will be needed to definitely move the data into the Relational model.
4. Construct all query transforming the data from the "import" tables and moving it into your relational model tables. **These query must do their job without switching off any constraints in your model.**
5. **Final carry out the final import phase by running all your queries and populating your relational model.**

NeatFlint The Internet Movies database (Part3)

Now you are supposed to have a working database with a model populated with data from movies, reviews, actors, directors and the relationships among them properly constrained as depicted in the E/R model you were provided in Part1.

We are providing you with a view of our database relational model (made with PGModeler) so you can check against your own. Your answers(results) to the queries may display subtle differences from ours if you haven't considered participations or outer joins where applicable. We provide you in advance the results for a few sample queries so you can check against your own data.

```
angel=> \i import_schema.sql
BEGIN
creating schema
CREATE SCHEMA
creating interim tables import_*
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
Bulk loading data into import_movies tables
COPY 1129
Bulk loading data into import_actors tables
COPY 997
Bulk loading data into import_directors tables
COPY 560
Bulk loading data into import_movie_reviews tables
COPY 1129
Bulk loading data into import_movie_medias tables
COPY 1109
Bulk loading data into import_movies_directors tables
COPY 147
Bulk loading data into import_movies_actores tables
COPY 210
creating and populating people from actors and directors
CREATE TABLE
INSERT 0 1534
creating and populating directors
CREATE TABLE
INSERT 0 560
creating and populating actors
CREATE TABLE
INSERT 0 997
creating and populating movies
CREATE TABLE
INSERT 0 1126
```

```

creating and populating genres
CREATE TABLE
INSERT 0 3431
creating and populating websites from reviews and covers
CREATE TABLE
INSERT 0 7
creating and populating reviews
CREATE TABLE
INSERT 0 1129
ROLLBACK
angel=>

```

Now consider the following questions and **provide the Relational algebra, SQL, and results in every case:**

1. Given the query "Show the name of people which is actor and also director" Construct the relational algebra expression for that query and also provide the equivalent query in SQL code and compute the results from the database.
2. What is the overall minute count for all movies in which Tom Cruise is acting? Group the results by genre and provide the SQL code as well.
3. Show people acting or directing any Horror movies. Provide the relational expression for your SQL query as well.
4. How many people, considering actors and directors can you count from the movie "The Lord of the Rings: The Return of the King"? Show the SQL code.
5. Show directors and the movies they directed for those people being also actors in addition to being directors.
6. Provide the relational expression for all actors born before Dec 31 1980 and also the equivalent SQL expression. Provide also the result from your database.
7. Show the overall number of movies by genre with most popular genres first.
8. What movies share the same title but have different year? Provide a SQL code for this query, show them in lexicographical order and finally provide the relational algebra expression.
9. Provide the relational algebra for the query showing the best ranked movies (by rating). Provide the SQL query as well and finally show the corresponding results.
10. Provide the query to show movies having the same average rating and the results.

