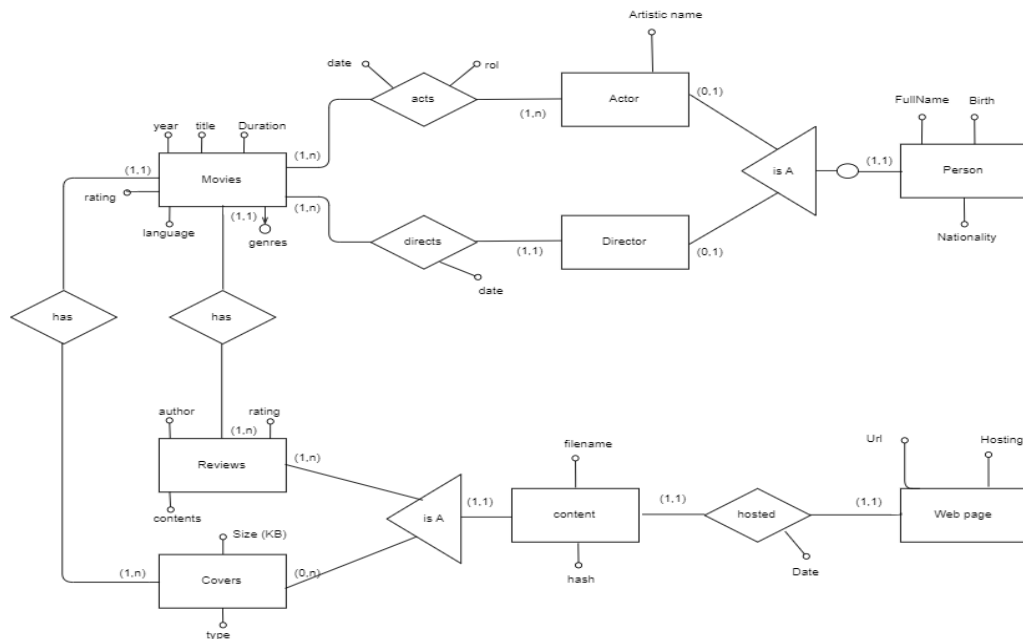


NeatFlint The Internet Movies database



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 they 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:

We assume your relational model is already created with the all tables and constraints (table constrains, primary and foreign key constraints) in place.

1. 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.
2. 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.
3. **Provide all queries 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 constrains in your model.**
4. **Finally carry out the final import phase by running all your queries and populating your relational model.**

Deliverables:

A document containing the following items:

1. **SQL code used and the rationale followed to create your relational model (tables and constraints) from the E/R and the rationales and rules followed in the process.**
2. **SQL code used and the rationale followed for loading the data from the files into the interim tables 'import_***
3. **SQL code used and the rationale followed for changes and alterations in tables and constraints in your relational model in order to accommodate the data**
4. **SQL code used and the rationale followed to transform and populate your model with data from the interim tables.**