You wish to query a database which is a subset of the IMDb Internet Movie Database. [Note: The first course practical used such a database.] Recall that the database schema has table movies with key movie_id, the table people with key person_id, and the table genres with key genre_id. The table has_genre implements a relationship between movies and genres and has the key (movie_id, genre_id). The table plays_role implements a relationship between movies and people and has the key (movie_id, person_id, role).

(a) Write an SQL query to return the number of movies that are romantic comedies. [6 marks]

(b) Complete the following SQL so that it returns records of the form

\[
\text{pid1, pid2, movie_id}
\]

where pid1 and pid2 are identifiers of co-actors with roles in the romantic comedy with identifier movie_id. This should be a symmetric table so that if pid1, pid2, m is in the result, then so should be pid2, pid1, m. However, it should not include records where pid1 and pid2 are equal.

\[
\text{select R1.person_id as pid1,}
\text{R2.person_id as pid2,}
\text{M.movie_id as movie_id}
\text{from .... your code goes here ....}
\]

[7 marks]

(c) Complete the following SQL so that it returns records of the form

\[
\text{name1, title1, name2, title2, name3}
\]

that can be interpreted as follows:

- Actors name1 and name2 are co-actors in a romantic comedy title1.
- Actors name2 and name3 are co-actors in a romantic comedy title2.
- However, neither actor name1 has a role in the movie associated with title2 and name3, nor does actor name3 have a role in the movie associated with title1 and name1.

\[
\text{select P1.name as name1, M1.title as title1,}
\text{P2.name as name2, M2.title as title2,}
\text{P3.name as name3}
\text{from .... your code goes here ....}
\]