Part 1 - Data

The schema used is in 3NF (Third Normal Form). Each table has a primary key, and non-key columns are dependent on the primary key. There aren't any dependencies between non-key columns in the same table, and all the foreign key relationships are made between primary and foreign keys of their respective tables.

Relations	**Attributes**
movie	**Primary:** id
genre	**Primary:** id
movie_genre	**Primary:** id **Foreign:** movie_id, genre_id
keyword	**Primary:** id
movie_keyword	**Primary:** id **Foreign:** movie_id, keyword_id
production_company	**Primary:** id
movie_production_company	**Primary:** id **Foreign:** movie_id, production_company_id
production_country	**Primary:** iso_3166_1
movie_production_country	**Primary:** id **Foreign:** movie_id, production_country_id
spoken_language	**Primary:** iso_639_1
movie_spoke_language	**Primary:** id **Foreign:** movie_id, spoken_language_id

movie - This table contains information about each movie, such as its budget, title, and release date.

genre - This table contains information about each movie, such as its budget, title, and release date.

movie_genre - This table contains a mapping between movies and their respective genres. movie_id and genre_id are foreign keys that reference the id of the movie and genre tables respectively.

keyword - This table contains keywords associated with movie. The id column is the primary key of this table.

movie_keyword - This table contains a mapping between movies and their respective keywords. The movie_id and keyword_id are foreign keys that reference the id of the movie and keyword tables, respectively.

production_company - This table contains information about production companies.

movie_production_company - This table contains a mapping between movies and their associated production companies. The movie_id and production_company_id columns are foreign keys that reference the id of the movie and production_company tables, respectively.

production_country - This table contains information about countries that produce movies. iso_3166_1 is a two-letter country code. (Such as US, CA, CN)

movie_production_country - This table contains a mapping between movies and their associated production countries. The movie_id and production_country_id columns are foreign keys that reference the id of the movie and production_country tables, respectively.

spoken_language - This table contains information about languages spoken in movies. iso_639_1 is a two-letter language code based (Such as en, af, pt),

movie_spoken_language - This table contains a mapping between movies and their associated spoken languages. The movie_id and spoken_language_id columns are foreign keys that reference the id of the movie and spoken_language tables, respectively.

Part 2 - Queries

The following are the top 5 tuples returned when running the queries.

1. What is the average budget of all movies?

SELECT AVG(budget) FROM movie

budget
29045039.8753

2. Show the movies that were produced in the United States

```
SELECT m.title, pc.name
    FROM movie m
    JOIN movie_production_company mpc ON m.id = mpc.movie_id
    JOIN production_company pc ON mpc.production_company_id = pc.id
    JOIN movie_production_country mpcn ON m.id = mpcn.movie_id
    JOIN production_country pcn ON mpcn.production_country_id = pcn.iso_3166_1
    WHERE pcn.name = 'United States of America'
```

title	name
Avatar	Ingenious Film Partners
Avatar	Twentieth Century Fox Film Corporation
Avatar	Dune Entertainment
Avatar	Lightstorm Entertainment
Pirates of the Caribbean: At World's End	Walt Disney Pictures

3. Show the top 5 movies that made the most revenue.

```
SELECT title, revenue
    FROM movie
    ORDER BY revenue DESC
    LIMIT 5
```

title	revenue
Avatar	2787965087
Titanic	1845034188
The Avengers	1519557910
Jurassic World	1513528810
Furious 7	1506249360

4. What movies have both the genre Science Fiction and Mystery.

```
SELECT m.title, GROUP_CONCAT(DISTINCT g.name) AS genres
    FROM movie AS m
    INNER JOIN movie_genre AS mg ON m.id = mg.movie_id
    INNER JOIN genre AS g ON mg.genre_id = g.id
    WHERE g.name IN ('Science Fiction', 'Mystery')
    GROUP BY m.title
    HAVING COUNT(DISTINCT g.name) = 2
```

title	genre
2001: A Space Odyssey Mystery	Mystery,Science Fiction
Atlas Shrugged Part II	Mystery,Science Fiction
Atlas Shrugged Part III: Who is John Galt?	Mystery,Science Fiction
Beneath the Planet of the Apes	Mystery,Science Fiction
Blindness	Mystery,Science Fiction

5. Find the movies that have a popularity greater than the average popularity.

```
SELECT title, popularity
    FROM movie
    WHERE popularity > (SELECT AVG(popularity) FROM movie)
    ORDER BY popularity DESC
```

title	popularity
Minions	875.581
Interstellar	724.248
Deadpool	514.57
Guardians of the Galaxy	481.099
Mad Max: Fury Road	434.279