WORKSHEET 5 SQL

Refer the following ERD and answer all the questions in this worksheet. You have to write the queries using

MySQL for the required Operation.

⬚ The movie table contains information about each movie. There are text descriptions such as title and

overview. Some fields are more obvious than others: revenue (the amount of money the movie made), budget

(the amount spent on creating the movie). Other fields are calculated based on data used to create the data

source: popularity, votes_avg, and votes_count. The status indicates if the movie is Released, Rumoured, or in

Post-Production.

⬚ The country list contains a list of different countries, and the movie_country table contains a record of which

countries a movie was filmed in (because some movies are filmed in multiple countries). This is a standard

many-to-many table, and you'll find these in a lot of databases.

⬚ The same concept applies to the production_company table. There is a list of production companies and a

many-to-many relationship with movies which is captured in the movie_company table.

⬚   The languages table has a list of languages, and the movie_languages captures a list of languages in a movie.

The difference with this structure is the addition of a language_role table.

⬚   This language_role table contains two records: Original and Spoken. A movie can have an original language

(e.g. English), but many Spoken languages. This is captured in the movie_languages table along with a role.

▢	Genres define which category a movie fits into, such as Comedy or Horror. A movie can have multiple

genres, which is why the movie_genres table exists.WORKSHEET

▢ The same concept applies to keywords, but there are a lot more keywords than genres. I'm not sure what

qualifies as a keyword, but you can explore the data and take a look. Some examples as "paris", "gunslinger",

or "saving the world".

▢ The cast and crew section of the database is a little more complicated. Actors, actresses, and crew members

are all people, playing different roles in a movie. Rather than have separate lists of names for crew and cast,

this database contains a table called person, which has each person's name.

▢	The movie_cast table contains records of each person in a movie as a cast member. It has their character

name, along with the cast_order, which I believe indicates that lower numbers appear higher on the cast list.

▢ The movie_cast table also links to the gender table, to indicate the gender of each character. The gender is

linked to the movie_cast table rather than the person table to cater for characters which may be a different

gender than the person, or characters of unknown gender. This means that there is no gender table linked to

the person table, but that's because of the sample data.

▢	The movie_crew table follows a similar concept and stores all crew members for all movies. Each crew

member has a job, which is part of a department (e.g. Camera).

QUESTIONS:

1. Write SQL query to show all the data in the Movie table.
   Answer-results=cur.execute('select * from Movie')

   results.fetchall()
2. Write SQL query to show the title of the longest runtime movie.
   Answer-result=cur.execute("select max(runtime) from Movie")
   for i in result:
   print(i)
3. Write SQL query to show the highest revenue generating movie title
   Answer- result=cur.execute("select max(revenue) from Movie")
   for i in result:
   print(i)
4. Write SQL query to show the movie title with maximum value of revenue/budget.
   Answer-- result=cur.execute("select titile from Movie where max(revenue) ")
   for i in result:
   print(i)
5. Write a SQL query to show the movie title and its cast details like name of the person, gender, character name, cast order