SQL ASSIGNMENT-5

Firstly, we have to create tables for given database

```
import sqlite3

movie_db=sqlite3.connect("my_database.movie_db")

cur = movie_db.cursor()

#create table 'country'

s = "CREATE TABLE country (country_id VARCHAR(5) NOT
NULL,country_iso_code VARCHAR(10) NOT NULL,country_name
TEXT(15) NOT NULL)"

cur.execute(s)

# create table 'production country'

s = "CREATE TABLE production_country(movie_id INT(5) NOT
NULL,country_id VARCHAR(5) NOT NULL)"

cur.execute(s)

# create table 'language'

s = "CREATE TABLE language(language_id INT(5) NOT
NULL,language_code VARCHAR(10),language_name TEXT(10) NOT
NULL)"

cur.execute(s)

# create table 'movie_languages'

s = "CREATE TABLE movie_languages(movie_id INT(5) NOT
NULL,language_id INT(5) NOT NULL,language_role_id VARCHAR(10))"

cur.execute(s)

# create table 'language_role'

s = "CREATE TABLE language_role(role_id INT(5) NOT
NULL,language_role VARCHAR(10) NOT NULL)"
```

```
cur.execute(s)

# create table 'genre'

s = "CREATE TABLE genre(genre_id INT(5) NOT NULL,genre_name
VARCHAR(10) NOT NULL)"

cur.execute(s)

# create table 'movie_genre'

s = "CREATE TABLE movie_genre(genre_id INT(5) NOT NULL,movie_id
INT(5) NOT NULL)"

cur.execute(s)

# create table 'keyward'

s = "CREATE TABLE keyward(keyward_id INT(5) NOT
NULL,keyward_name VARCHAR(10) NOT NULL)"

cur.execute(s)

# create table 'movie_keywards'

s = "CREATE TABLE movie_keywards(movie_id INT(5) NOT
NULL,keyward_id INT(5) NOT NULL)"

cur.execute(s)

# create table 'movie'

s = "CREATE TABLE movie(movie_id INT(5) NOT NULL,title
VARCHAR(15) NOT NULL,budget INT(8),homepage VARCHAR(25)
NOT NULL,overview VARCHAR(35) NOT NULL,popularity
VARCHAR(30) NOT NULL,release_date DATE(8) NOT NULL,revenue
INT(8) NOT NULL,runtime TIME(3) NOT NULL,movie_status TEXT(10)
NOT NULL,tagline VARCHAR(50) NOT NULL, votes_avg INT(5) NOT
NULL,votes_count INT(5) NOT NULL)"

cur.execute(s)

# create table 'movie_company'
```

```
s = "CREATE TABLE movie_company(movie_id INT(5) NOT NULL,company_id INT(5) NOT NULL)"

cur.execute(s)

# create table 'production_company'

s = "CREATE TABLE production_company(company_id INT(5) NOT NULL,company_name VARCHAR(15) NOT NULL)"

cur.execute(s)

# create table 'movie_cast'

s = "CREATE TABLE movie_cast(movie_id INT(5) NOT NULL,gender_id INT(5) NOT NULL,person_id INT(5) NOT NULL,character_name TEXT(15) NOT NULL,cast_order INT(3) NOT NULL)"

cur.execute(s)

# create table 'movie_crew'

s = "CREATE TABLE movie_crew(person_id INT(5) NOT NULL,movie_id INT(5) NOT NULL,department_id INT(5) NOT NULL,job TEXT(25) NOT NULL)"

cur.execute(s)

# create table 'gender'

s = "CREATE TABLE gender(gender_id INT(5) NOT NULL,gender TEXT(10) NOT NULL)"

cur.execute(s)

# create table 'person'

s = "CREATE TABLE person(person_id INT(5) NOT NULL,person_name TEXT(20) NOT NULL)"

cur.execute(s)

# create table 'department'
```

```
s = "CREATE TABLE department(department  id INT(5) NOT NULL,department  name TEXT(20) NOT NULL)"
cur.execute(s)

# 1. Write SQL query to show all the data in the Movie table.
sql = "SELECT * FROM movie"
cur.execute(sql)
result=cur.fetchall()
for r in result:
    print(r)

# 2. Write SQL query to show the title of the longest runtime movie.
sql = "SELECT title FROM 'movie' WHERE runtime==(SELECT max(runtime) FROM 'movie')"
cur.execute(sql)
result=cur.fetchall()
for r in result:
    print(r)

#3. Write SQL query to show the highest revenue generating movie title.
sql = "SELECT title FROM 'movie' WHERE revenue==(SELECT max(revenue) FROM 'movie')"
cur.execute(sql)
result=cur.fetchall()
```

```
for r in result:
    print(r)
```

```
# 4. Write SQL query to show the movie title with maximum value of revenue/budget.
sql = "SELECT title FROM 'movie' WHERE budget==(SELECT max(budget) FROM 'movie')"
cur.execute(sql)
result=cur.fetchall()
for r in result:
    print(r)
```

```
# 5. Write a SQL query to show the movie title and its cast details like name of the person, gender, character name, cast order.
sql = "SELECT movie.title,person.person_name,gender.gender,movie_cast.character_name,movie_cast.cast_order FROM movie INNER JOIN movie ON movie.movie_id=movie_cast.movie_id,INNER JOIN person ON person.person_id=movie_cast.person_id,INNER JOIN gender ON gender.gender_id = movie_cast.gender_id "
cur.execute(sql)
result=cur.fetchall()
for r in result:
    print(r)
```

#7. Write a SQL query to show all the genre_id in one column and genre_name in second column.

```
sql = "SELECT * FROM genre"

cur.execute(sql)

result=cur.fetchall()

for r in result:

  print(r)
```

9. Write a SQL query to show movie name in first column, no. of crew members in second column and umber of cast members in third column

```
sql = "SELECT movie.movie_id, movie.title

    FROM movie_cast INNER JOIN movie ON movie_cast.movie_id = movie.movie_id

    FROM movie_crew INNER JOIN movie ON movie_crew.movie_id = movie.movie_id"

cur.execute(sql)

result=cur.fetchall()

for r in result:

  print(r)
```

#10. Write a SQL query to list top 10 movies title according to popularity column in decreasing order.

```
sql = "SELECT TOP 10 title FROM movie ORDER BY popularity DESC"

cur.execute(sql)
```

```
result=cur.fetchall()
for r in result:
    print(r)
```

#11. Write a SQL query to show the name of the 3rd most revenue generating movie and its revenue.

```
sql = "SELECT title FROM `movie` ORDER BY `revenue` DESC LIMIT 1 OFFSET 2;"
cur.execute(sql)
result=cur.fetchall()
for r in result:
    print(r)
```

#12. Write a SQL query to show the names of all the movies which have “rumoured” movie status.

```
sql = "SELECT title FROM 'movie' WHERE movie_status ='rumoured'"
cur.execute(sql)
result=cur.fetchall()
for r in result:
    print(r)
```

#15. Write a SQL query to show the title of top 20 movies arranged in decreasing order of their budget.

```
sql = "SELECT TOP 20 title FROM movie ORDER BY budget DESC"
cur.execute(sql)
result=cur.fetchall()
```

```
for r in result:
    print(r)
```