CSCE 5350.001

Fundamentals of Database Systems

Project Part 2

Naga Vara Pradeep Yendluri

11646461

nagavarapradeepyendluri@my.unt.edu

Project Description:

The Movie Producer Management System is an application that is being developed for a movie production company like Universal Studios. The system is designed to store and manage information about the company's movies, artists, songs, employees and various other aspects of the movie production process. The system will store information about the producing site locations, movie-script-inventory, sponsoring companies, employee data, and payroll. It will also store information about the artists and the movies they have worked on, as well as the various aspects of the movie production process, such as soundtracks, awards, and more.

We have Identified the following entities and relations for the movie producer management system.

1. **Movies:** The entity 'Movies' provides information about the different movies produced by the company. It has 5 attributes, including the movie id, movie title, release date, duration and script inventory id. This entity is important for keeping track of the different movies produced by the company and the information related to each movie.
2. **Artists:** The entity 'Artists' provides information about the actors involved in the movies. It has 4 attributes, including the artist id, artist name, artist date of birth and gender. This entity is important for maintaining the information about the actors, their age and date of birth, which is required for casting actors for various roles.
3. **Genre:** The entity 'Genre' provides information about the genre to which the movie belongs. It has 2 attributes, genre id, and genre name. This entity is important for categorizing the movies into different genres, which helps in better management and analysis of the movies.
4. **Sponsoring Companies:** The entity 'Sponsoring Companies' provides information about the companies that sponsor the movies. It has 2 attributes, including company id and company name. This entity is important for tracking the sponsorship deals and the companies that sponsor the movies.
5. **Site Locations:** The entity 'Site Locations' provides information about the different producing sites, including their addresses and buildings. It has 3 attributes, including location id, name, and address. This entity is important for tracking the different producing sites and their details, which is essential for managing the movie production process.
6. **Buildings:** The entity 'Buildings' provides information about the buildings in each producing site. It has 4 attributes, including Building id, name, location id and purpose.

This entity is important for tracking the different types of buildings present in the producing sites, which is essential for managing the resources and maintenance of the buildings.

7. **Movie Script Inventory:** The entity 'Movie Script Inventory' provides information about the movie scripts. It has 2 attributes, including script inventory id and script inventory name. This entity is important for tracking the different movie scripts and the information related to each script.
8. **Employees:** The entity 'Employees' provides information about the employees of the company. It has 4 attributes, including employe id, name, designation, and phone. This entity is important for maintaining the information about the employees, their job title and contact information, which is required for managing the human resources of the company.
9. **Payroll:** The entity 'Payroll' provides information about employee payroll data. It has 4 attributes, including payroll id, salary, employee id and hours worked. This entity is important for tracking the payroll information of the employees, which is essential for managing the finances of the company.
10. **Songs:** The entity 'Songs' provides information about different soundtracks used in the movies. It has 4 attributes, including song id, song name, movie id, and singer name. This entity is important for tracking the different soundtracks used in the movies and the information related to each soundtrack.

Binary Relations:

• One-to-Many relation:

1. Every Employee will have one payroll associated to them each month. So, the relation between employee and payroll is one to many.
2. Each script inventory will have many movies associated with. Hence, the relation between movie script inventory and movies will also be one to many.
3. Each movie can have multiple songs/soundtracks in it and inversely there can be multiple songs in a movie. So, the relation between songs and movies is one to many.
4. Each site location can have multiple buildings in them, and inversely multiple buildings can be located at a single location. So, the relation between these two entities will be one-to-many relation.

• Many-to-Many relation:

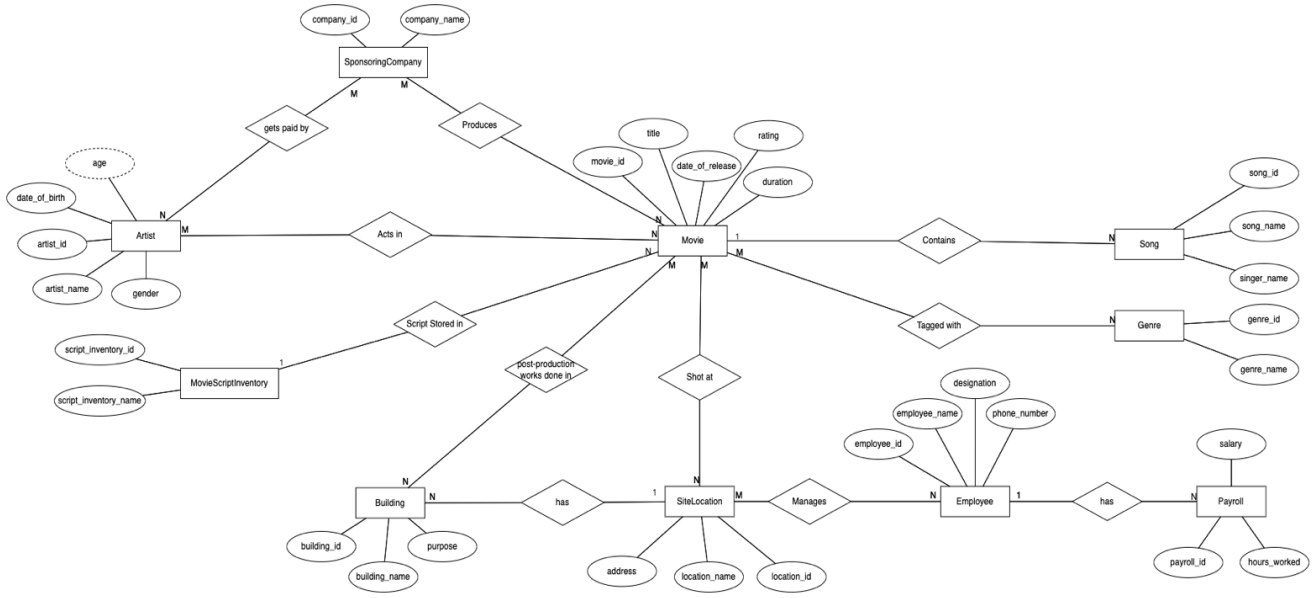
1. Each movie can have many artists performing in it and inversely many actors can act in many different movies. Hence, the relation between movies and artists will be many to many.
2. Many Sponsoring companies can sponsor for many movies in this system. Inversely, many movies can get sponsorship from different sponsoring companies. Hence, the relation between these two entities is many to many relation.
3. Many movies can be shot at different locations and inversely many locations can concurrently host shootings for many movies in different buildings so the relation between these two entities will also be many to many relation.
4. Many movies can have many genres so the relation between these two entities will also be many to many relation.

5. Many employees can manage many site locations so the relation between these two entities will also be many to many relation.
6. Many artists gets paid by many sponsoring companies so the relation between these two entities will also be many to many relation.

Assumptions:

1. Movies and Artists: It is assumed that each movie has one or more actors, and each actor is associated with one or more movies.
2. Genre: It is assumed that each movie belongs to multiple genres like adventure, action, drama, etc.
3. Sponsoring Companies: It is assumed that each movie is sponsored by one or more companies and each company sponsors one or more movies.
4. Site Locations and Buildings: It is assumed that each producing site has one or more buildings and each building is located at a single site.
5. Movie Script Inventory: It is assumed that each movie is associated with one unique movie scripts and each script is associated with a single movie.
6. Employee: It is assumed that each employee of the company has a unique employee ID, and each employee belongs to a single job title.
7. Payroll: It is assumed that each employee has payroll information, and each employee ID is associated with a single payroll record.
8. Songs: It is assumed that each movie has one or more soundtracks, and each soundtrack is associated with a single movie.
9. Movie title must be unique.
10. Ratings must be between 1 to 10 and no decimals values.
11. Employees get paid by weekly basis hence we used 40 hours constraint in the table.
12. Company name must be unique.
13. Phone number has 10 characters.
14. Employee will have unique phone number.
15. One employee can manage multiple locations. m-m
16. Duration of movie must be > 0 minutes.
17. Artist gender must be of F(female), M(male), T(transgender).

ER Diagram



ER Relations:

Relations transformed to schema.

Movie (**movie_id**, title, rating, date_of_release, duration, script_inventory_id)

Songs (**song_id**, song_name, singer_name, movie_id)

Genre (**genre_id**, genre_name)

TaggedWith (**movie_id**, **genre_id**)

SiteLocation (**location_id**, location_name, address)

ShotAt (**location_id, movie_id**)

Building (**building_id**, building_name, purpose, location_id)

PostProductionDoneIn (**movie_id, building_id**)

Employees (**employee_id**, employee_name, designation, phone_number)

Manages (**employee_id, location_id**)

Payroll (**payroll_id**, salary, employee_id, hours_worked)

SponsoringCompany (**company_id**, company_name)

getsPaidBy (**artist_id, company_id**)

Produces (**company_id, movie_id**)

Artist (**artist_id**, artist_name, date_of_birth, gender)

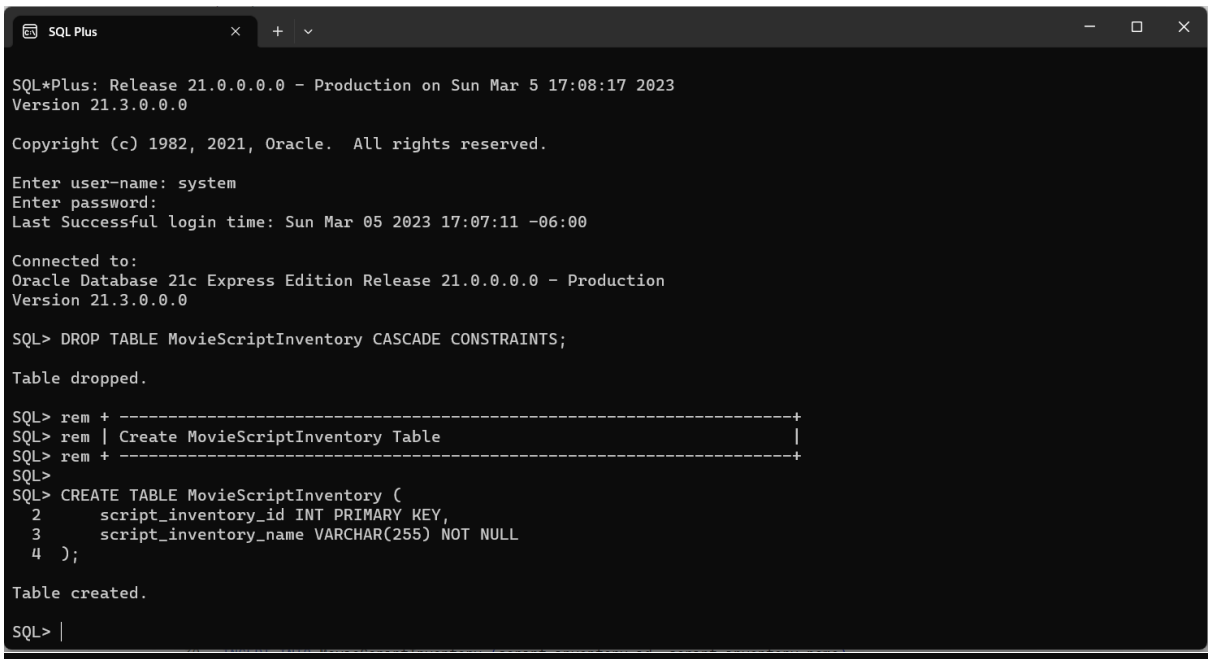
MovieScriptInventory (**script_inventory_id**, script_inventory_name)

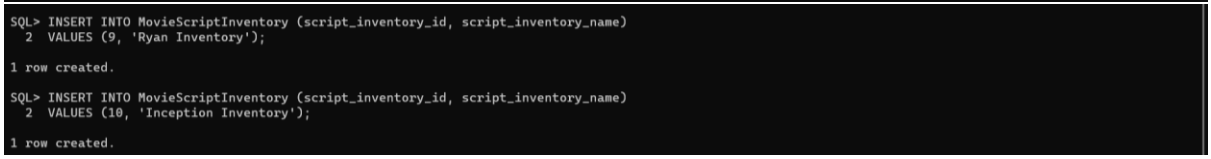
ActsIn(**movie_id, artist_id**)

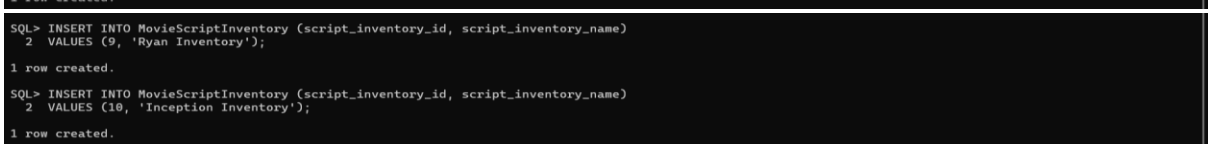
Tables:

MovieScriptInventory:

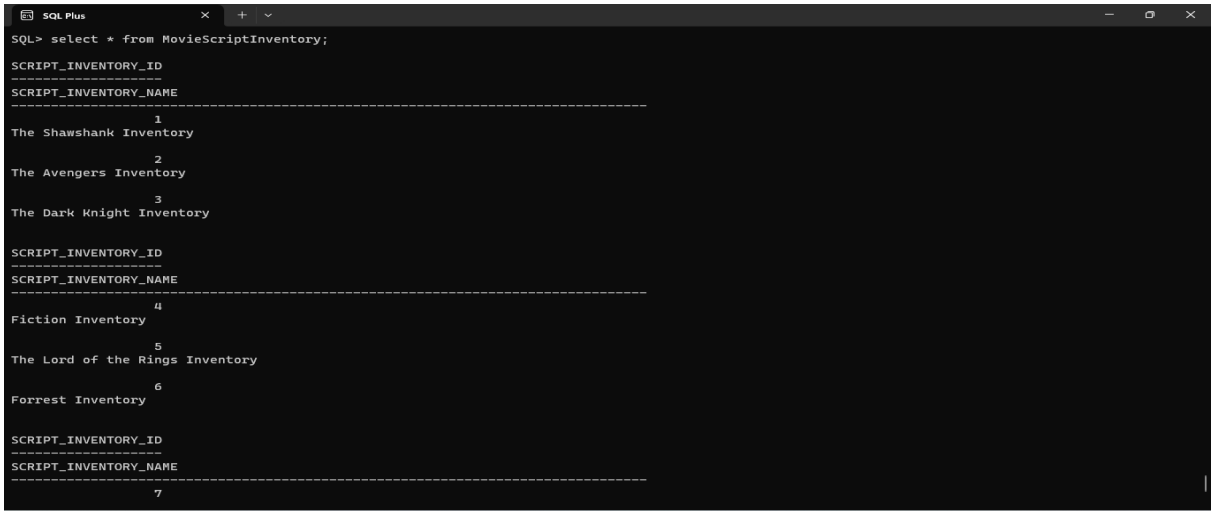
```
CREATE TABLE MovieScriptInventory (
  script_id INT PRIMARY KEY,
  script_name VARCHAR(255) NOT NULL);
```

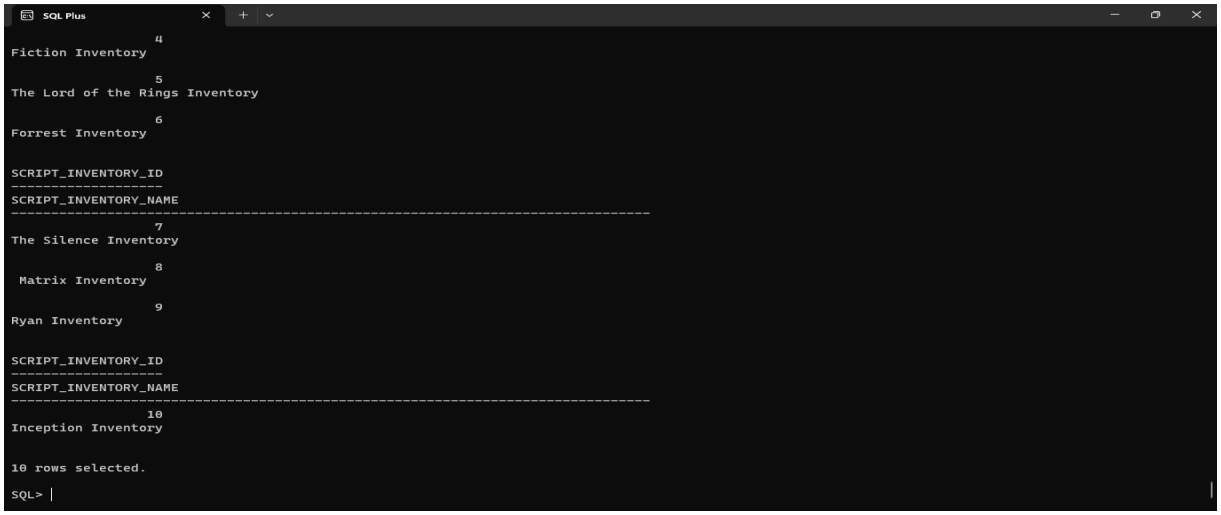






OUTPUT:





Movie:

```
CREATE TABLE Movie (
    movie_id INT NOT NULL,
    title VARCHAR(255) NOT NULL UNIQUE,
    rating INT NOT NULL,
    date_of_release DATE NOT NULL,
    duration INT NOT NULL,
    script_inventory_id INT NOT NULL,
    PRIMARY KEY (movie_id),
    FOREIGN KEY (script_inventory_id) REFERENCES
MovieScriptInventory(script_inventory_id),
    CONSTRAINT rating_constraint_violated CHECK (rating > 0 and rating < 11),
    CONSTRAINT duration_constraint_violated CHECK (duration > 0));
```

```
SQL> DROP TABLE Movie CASCADE CONSTRAINTS;

Table dropped.

SQL> rem  +----------------------------------------------------------------------+
SQL> rem  |  Create Movie Table                                                  |
SQL> rem  +----------------------------------------------------------------------+
SQL>
SQL> CREATE TABLE Movie (
  2      movie_id INT NOT NULL,
  3      title VARCHAR(255) NOT NULL UNIQUE,
  4      rating INT NOT NULL,
  5      date_of_release DATE NOT NULL,
  6      duration INT NOT NULL,
  7      script_inventory_id INT NOT NULL,
  8      PRIMARY KEY (movie_id),
  9      FOREIGN KEY (script_inventory_id) REFERENCES MovieScriptInventory(script_inventory_id),
 10      CONSTRAINT rating_constraint_violated CHECK (
 11          rating > 0
 12          and rating < 11
 13      ),
 14      CONSTRAINT duration_constraint_violated CHECK (duration > 0)
 15  );

Table created.

SQL>
```

SQL Plus

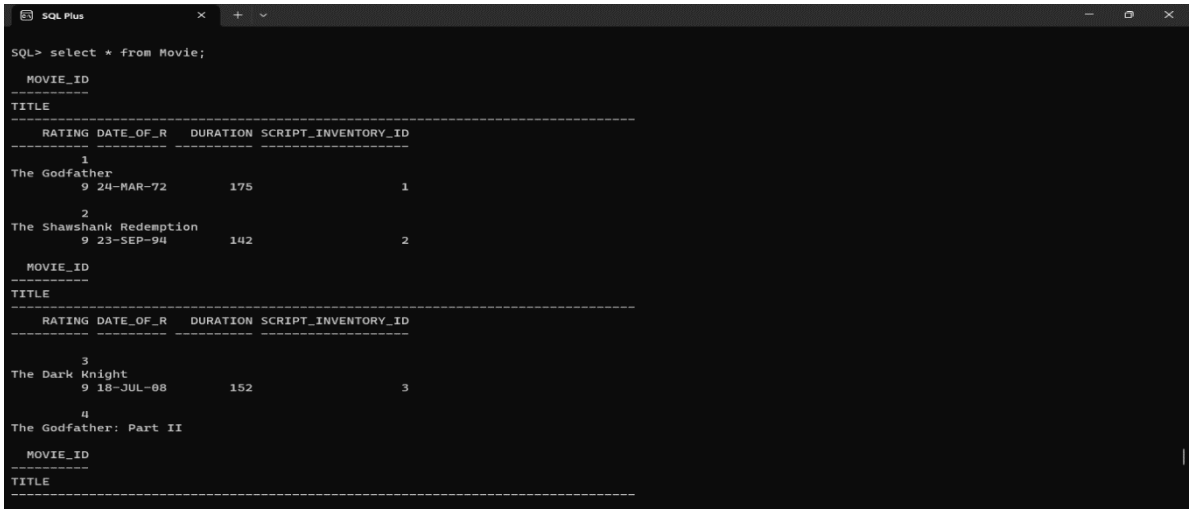
```
Table created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (1, 'The Godfather', 9, TO_DATE('1972-03-24', 'yyyy-mm-dd'), 175, 1);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (2, 'The Shawshank Redemption', 9, TO_DATE('1994-09-23', 'yyyy-mm-dd'), 142, 2);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (3, 'The Dark Knight', 9, TO_DATE('2008-07-18', 'yyyy-mm-dd'), 152, 3);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (4, 'The Godfather: Part II', 9, TO_DATE('1974-12-20', 'yyyy-mm-dd'), 202, 1);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (5, 'The Lord of the Rings: The Return of the King', 9, TO_DATE('2003-12-17', 'yyyy-mm-dd'), 201, 4);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (6, 'Pulp Fiction', 9, TO_DATE('1994-10-14', 'yyyy-mm-dd'), 154, 5);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (7, 'Schindler''s List', 8, TO_DATE('1993-12-15', 'yyyy-mm-dd'), 195, 6);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (8, 'The Lord of the Rings: The Fellowship of the Ring', 8, TO_DATE('2001-12-19', 'yyyy-mm-dd'), 178, 4);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (9, 'The Lord of the Rings: The Two Towers', 8, TO_DATE('2002-12-18', 'yyyy-mm-dd'), 179, 4);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
```

SQL Plus

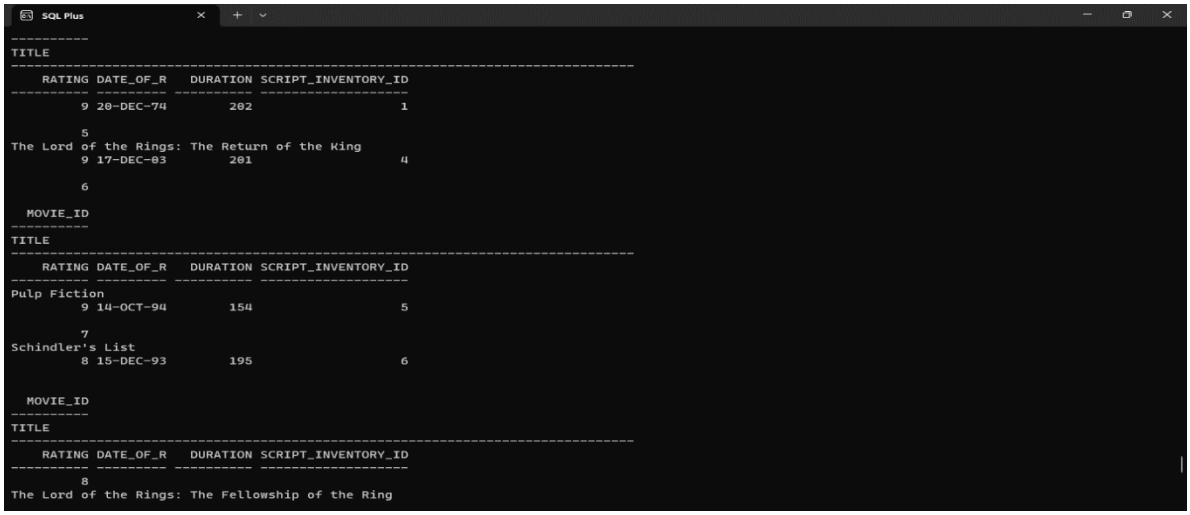
```
SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (10, 'Forrest Gump', 8, TO_DATE('1994-07-06', 'yyyy-mm-dd'), 142, 7);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (11, 'Inception', 8, TO_DATE('2010-07-16', 'yyyy-mm-dd'), 148, 8);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (12, 'Fight Club', 8, TO_DATE('1999-10-15', 'yyyy-mm-dd'), 139, 9);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (13, 'The Matrix', 8, TO_DATE('1999-03-31', 'yyyy-mm-dd'), 136, 10);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (14, 'Goodfellas', 8, TO_DATE('1990-09-19', 'yyyy-mm-dd'), 146, 9);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (15, 'The Silence of the Lambs', 8, TO_DATE('1991-02-14', 'yyyy-mm-dd'), 118, 9);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (16, 'The Departed', 8, TO_DATE('2006-10-06', 'yyyy-mm-dd'), 151, 8);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (17, 'Saving Private Ryan', 8, TO_DATE('1998-07-24', 'yyyy-mm-dd'), 169, 9);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (18, 'Terminator 2: Judgment Day', 8, TO_DATE('1991-07-03', 'yyyy-mm-dd'), 137, 6);

1 row created.

SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (19, 'The Prestige', 8, TO_DATE('2006-10-20', 'yyyy-mm-dd'), 130, 8);
```

```
SQL> INSERT INTO Movie (movie_id, title, rating, date_of_release, duration, script_inventory_id)
  2  VALUES (20, 'Gladiator', 8, TO_DATE('2000-05-01', 'yyyy-mm-dd'), 155, 6);

1 row created.
```

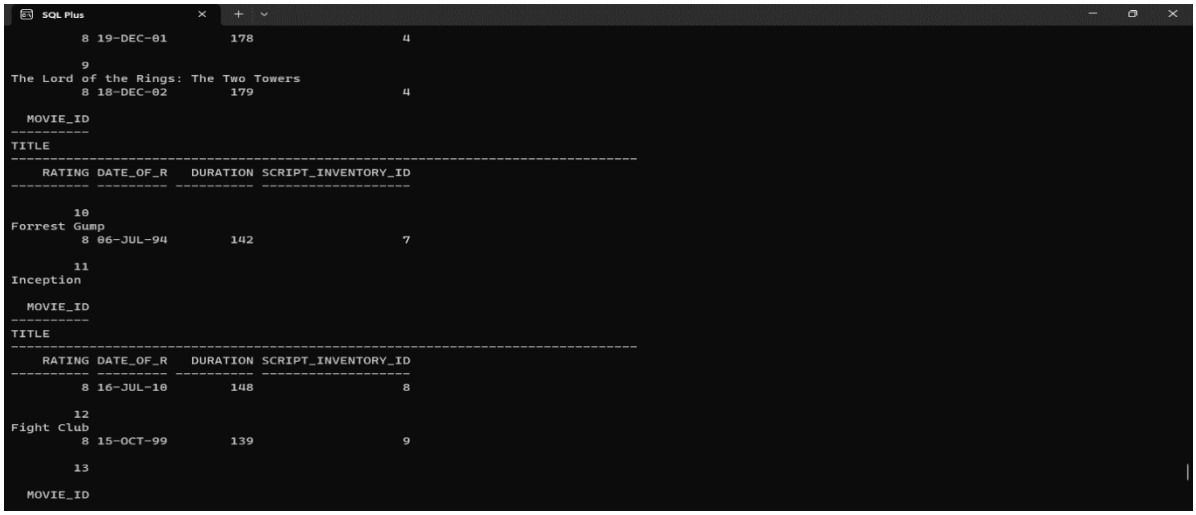
OUTPUT:



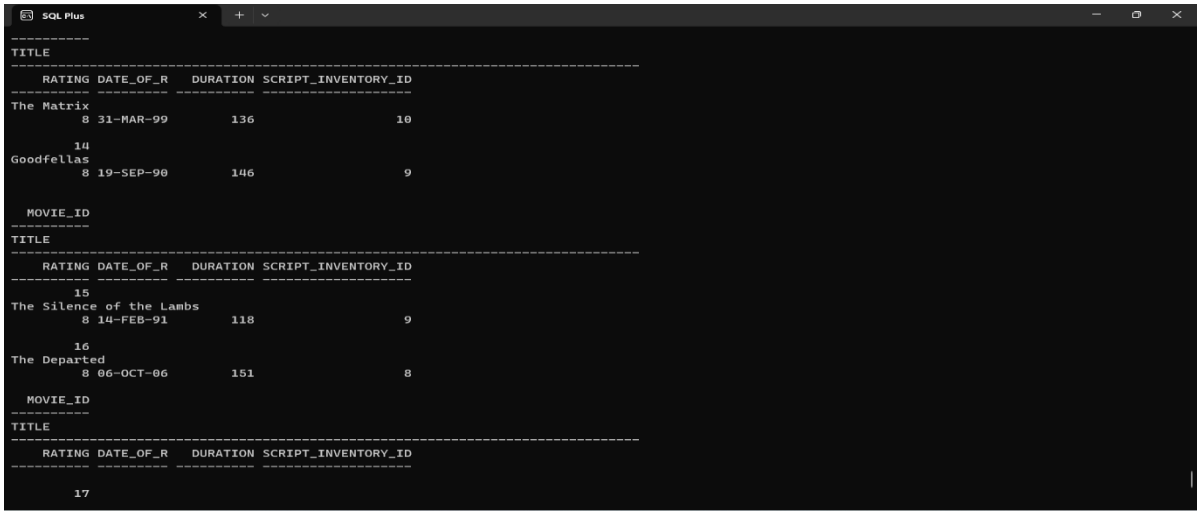
```
SQL Plus

SQL> select * from Movie;

  MOVIE_ID
----------
TITLE
--------------------------------------------------------------------------------
    RATING DATE_OF_R   DURATION SCRIPT_INVENTORY_ID
---------- --------- ---------- --------------------
         1
The Godfather
         9 24-MAR-72        175                    1

         2
The Shawshank Redemption
         9 23-SEP-94        142                    2

  MOVIE_ID
----------
TITLE
--------------------------------------------------------------------------------
    RATING DATE_OF_R   DURATION SCRIPT_INVENTORY_ID
---------- --------- ---------- --------------------

         3
The Dark Knight
         9 18-JUL-08        152                    3

         4
The Godfather: Part II

  MOVIE_ID
----------
TITLE
--------------------------------------------------------------------------------
```

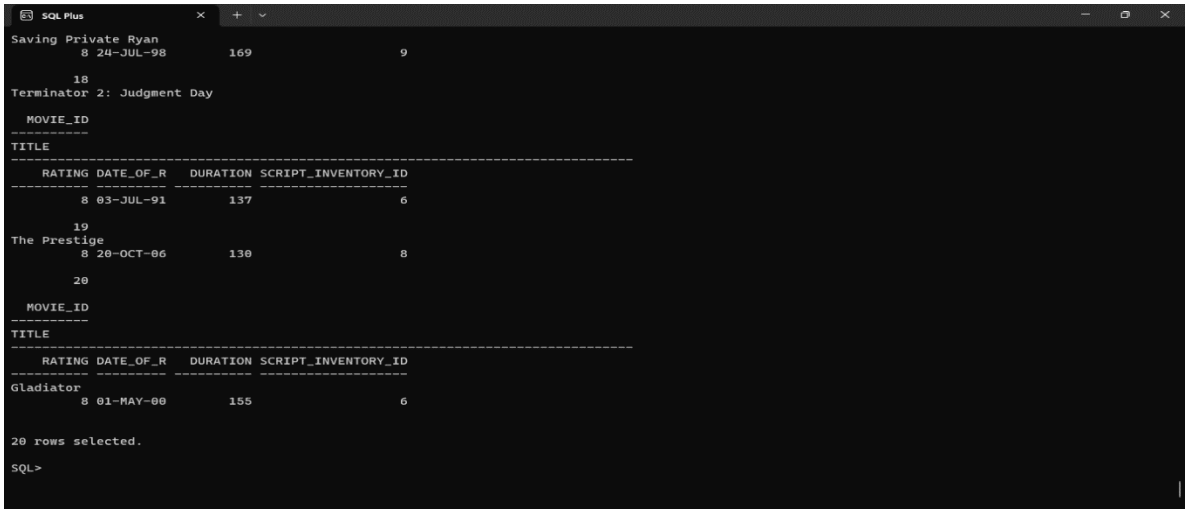


```
SQL Plus

----------
TITLE
--------------------------------------------------------------------------------
    RATING DATE_OF_R   DURATION SCRIPT_INVENTORY_ID
---------- --------- ---------- --------------------
         9 20-DEC-74        202                    1

         5
The Lord of the Rings: The Return of the King
         9 17-DEC-03        201                    4

         6

  MOVIE_ID
----------
TITLE
--------------------------------------------------------------------------------
    RATING DATE_OF_R   DURATION SCRIPT_INVENTORY_ID
---------- --------- ---------- --------------------
Pulp Fiction
         9 14-OCT-94        154                    5

         7
Schindler's List
         8 15-DEC-93        195                    6


  MOVIE_ID
----------
TITLE
--------------------------------------------------------------------------------
    RATING DATE_OF_R   DURATION SCRIPT_INVENTORY_ID
---------- --------- ---------- --------------------
         8
The Lord of the Rings: The Fellowship of the Ring
```



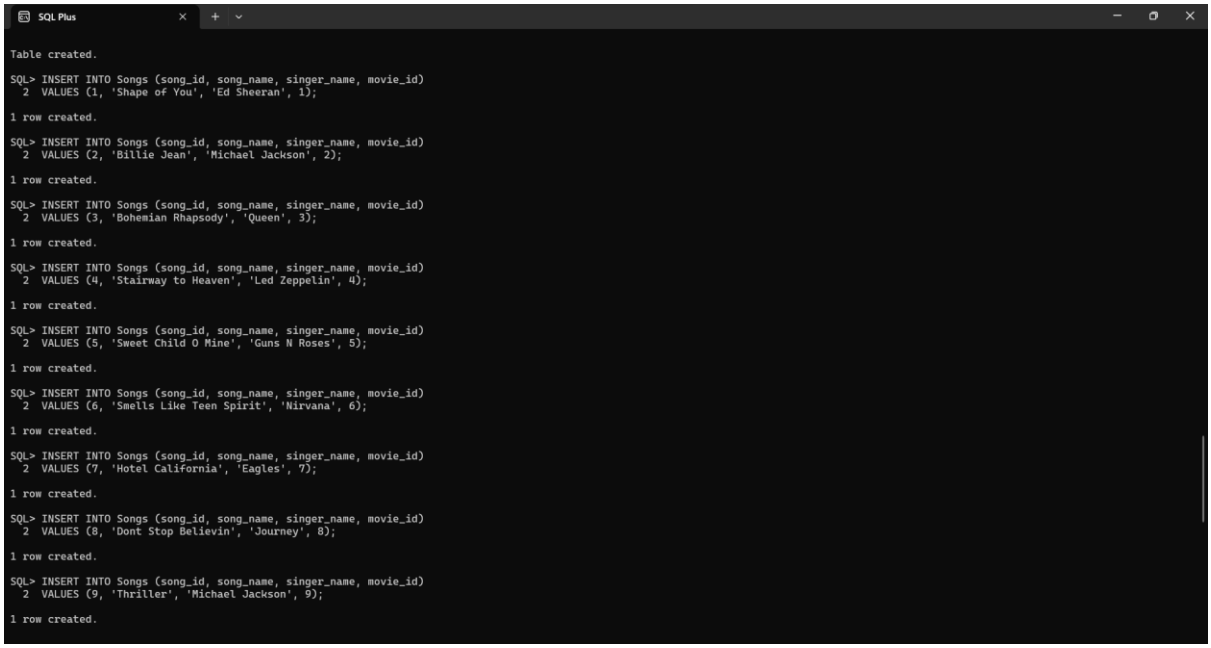
```
SQL Plus

         8 19-DEC-01        178                    4

         9
The Lord of the Rings: The Two Towers
         8 18-DEC-02        179                    4

  MOVIE_ID
----------
TITLE
--------------------------------------------------------------------------------
    RATING DATE_OF_R   DURATION SCRIPT_INVENTORY_ID
---------- --------- ---------- --------------------

        10
Forrest Gump
         8 06-JUL-94        142                    7

        11
Inception

  MOVIE_ID
----------
TITLE
--------------------------------------------------------------------------------
    RATING DATE_OF_R   DURATION SCRIPT_INVENTORY_ID
---------- --------- ---------- --------------------
         8 16-JUL-10        148                    8

        12
Fight Club
         8 15-OCT-99        139                    9

        13

  MOVIE_ID
```

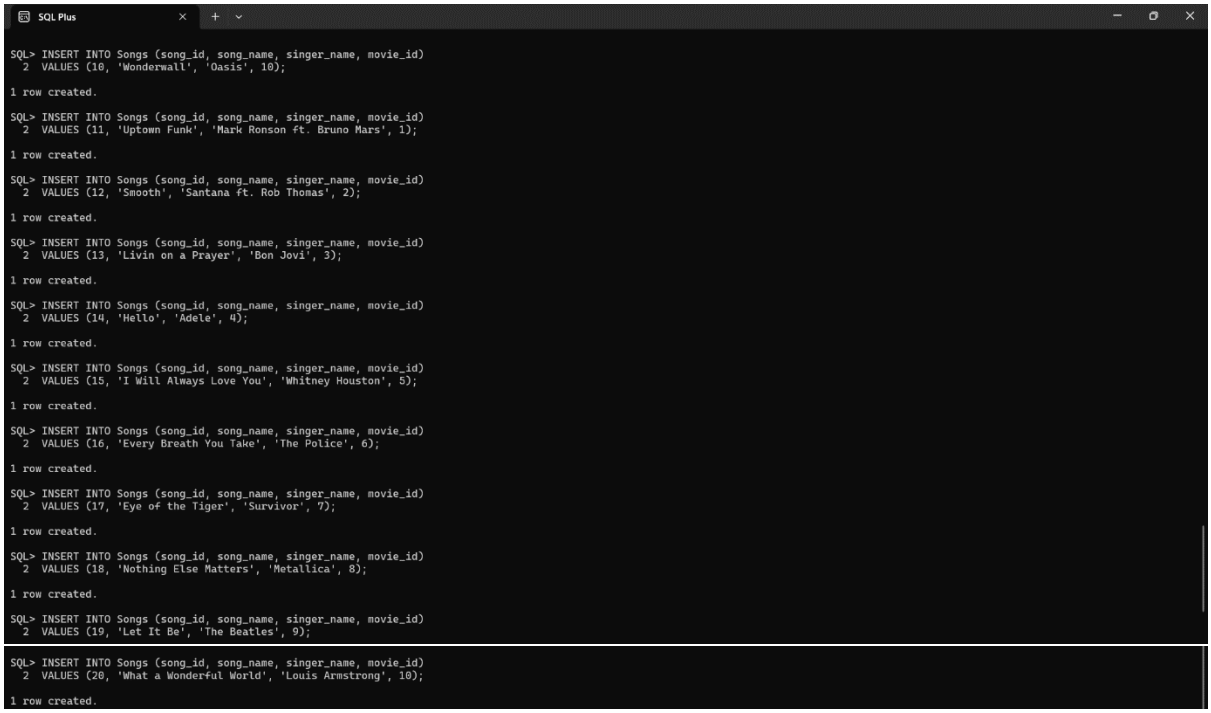




Songs:

```
CREATE TABLE Songs (
  song_id INT NOT NULL,
  song_name VARCHAR(255) NOT NULL,
  singer_name VARCHAR(255) NOT NULL,
  movie_id INT NOT NULL,
  PRIMARY KEY (song_id),
  FOREIGN KEY (movie_id) REFERENCES Movie(movie_id)
);
```

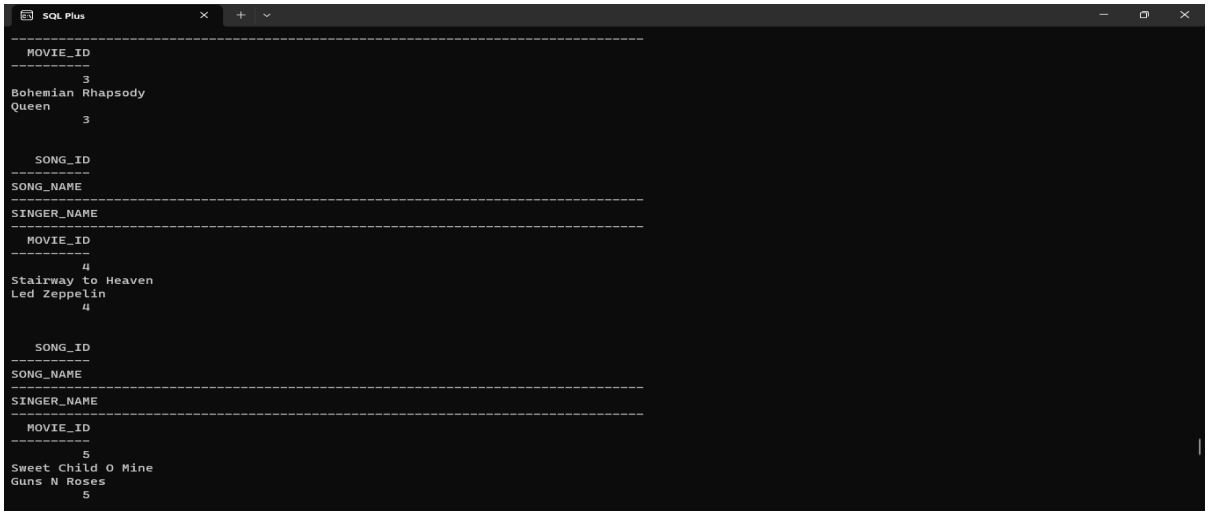

SQL Plus
Table created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (1, 'Shape of You', 'Ed Sheeran', 1);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (2, 'Billie Jean', 'Michael Jackson', 2);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (3, 'Bohemian Rhapsody', 'Queen', 3);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (4, 'Stairway to Heaven', 'Led Zeppelin', 4);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (5, 'Sweet Child O Mine', 'Guns N Roses', 5);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (6, 'Smells Like Teen Spirit', 'Nirvana', 6);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (7, 'Hotel California', 'Eagles', 7);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (8, 'Dont Stop Believin', 'Journey', 8);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (9, 'Thriller', 'Michael Jackson', 9);
1 row created.


SQL Plus
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (10, 'Wonderwall', 'Oasis', 10);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (11, 'Uptown Funk', 'Mark Ronson ft. Bruno Mars', 1);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (12, 'Smooth', 'Santana ft. Rob Thomas', 2);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (13, 'Livin on a Prayer', 'Bon Jovi', 3);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (14, 'Hello', 'Adele', 4);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (15, 'I Will Always Love You', 'Whitney Houston', 5);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (16, 'Every Breath You Take', 'The Police', 6);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (17, 'Eye of the Tiger', 'Survivor', 7);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (18, 'Nothing Else Matters', 'Metallica', 8);
1 row created.
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (19, 'Let It Be', 'The Beatles', 9);

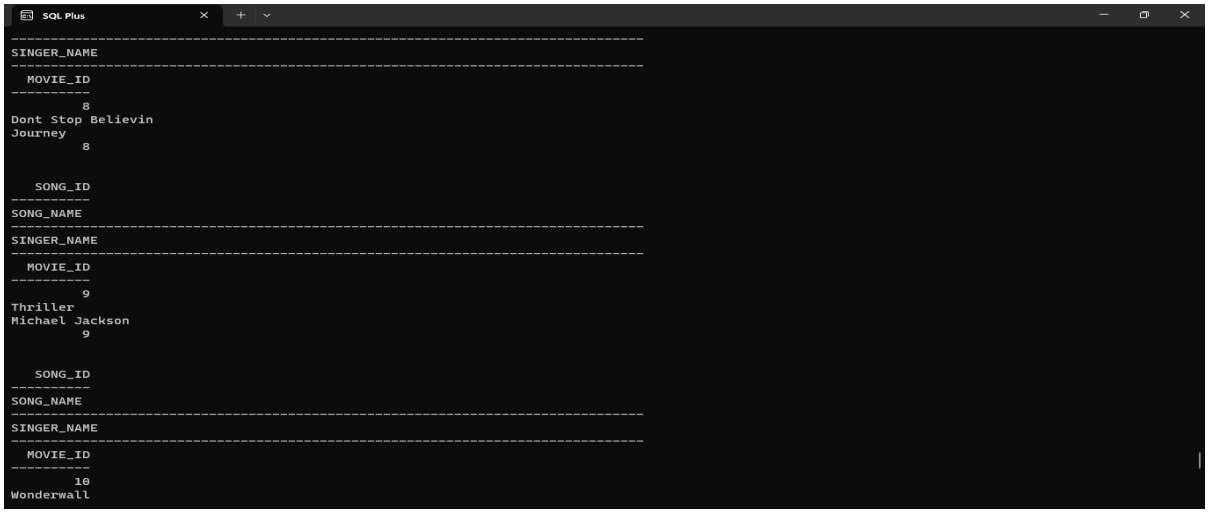
SQL> INSERT INTO Songs (song_id, song_name, singer_name, movie_id)
2 VALUES (20, 'What a Wonderful World', 'Louis Armstrong', 10);
1 row created.

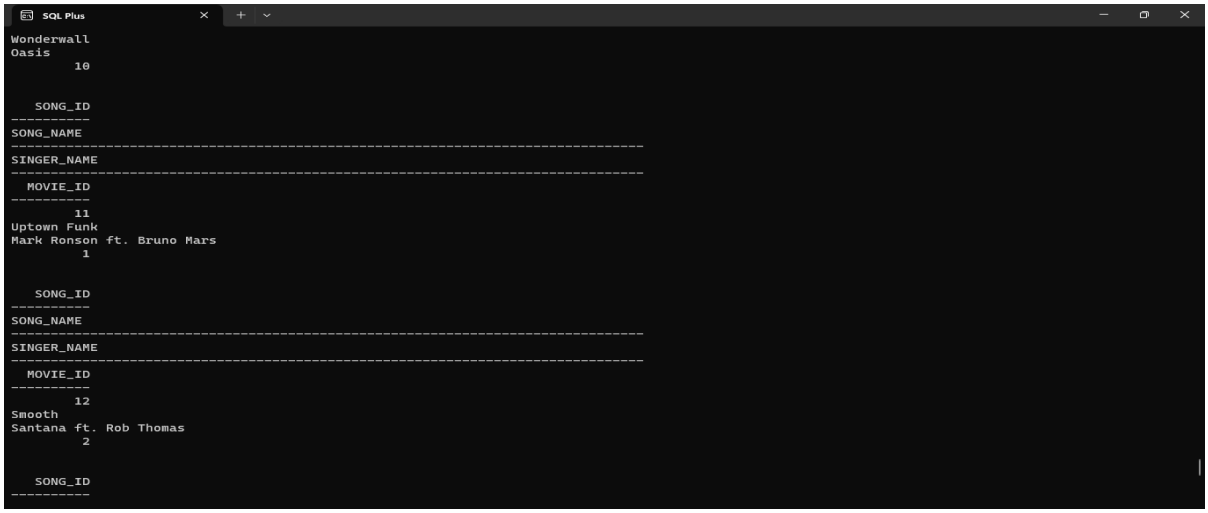
OUTPUT:

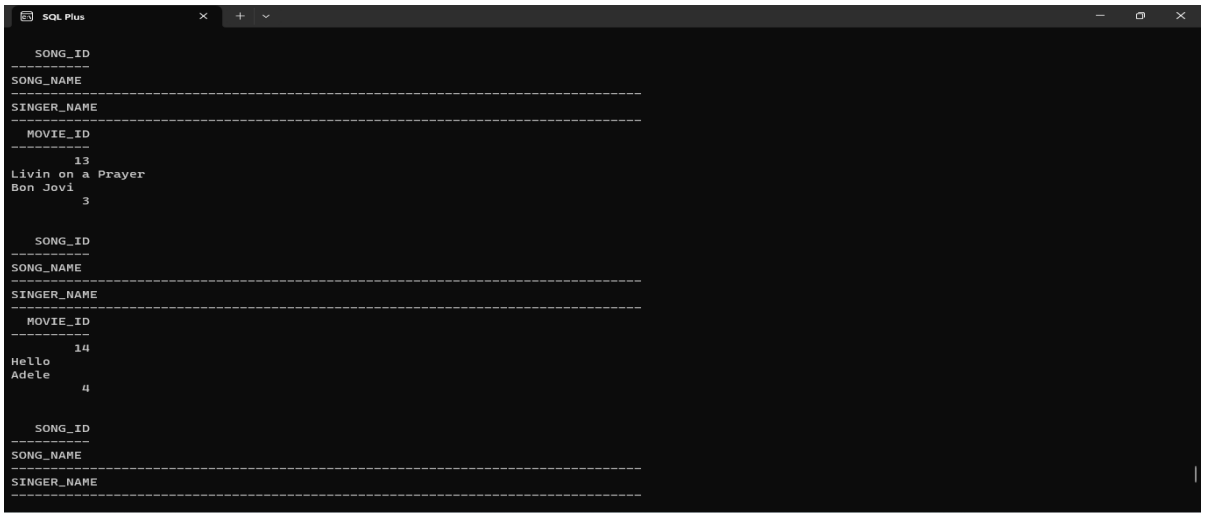

SQL Plus
SQL> select * from Songs;
SONG_ID
SONG_NAME
SINGER_NAME
MOVIE_ID
1
Shape of You
Ed Sheeran
1
SONG_ID
SONG_NAME
SINGER_NAME
MOVIE_ID
2
Billie Jean
Michael Jackson
2
SONG_ID
SONG_NAME
SINGER_NAME

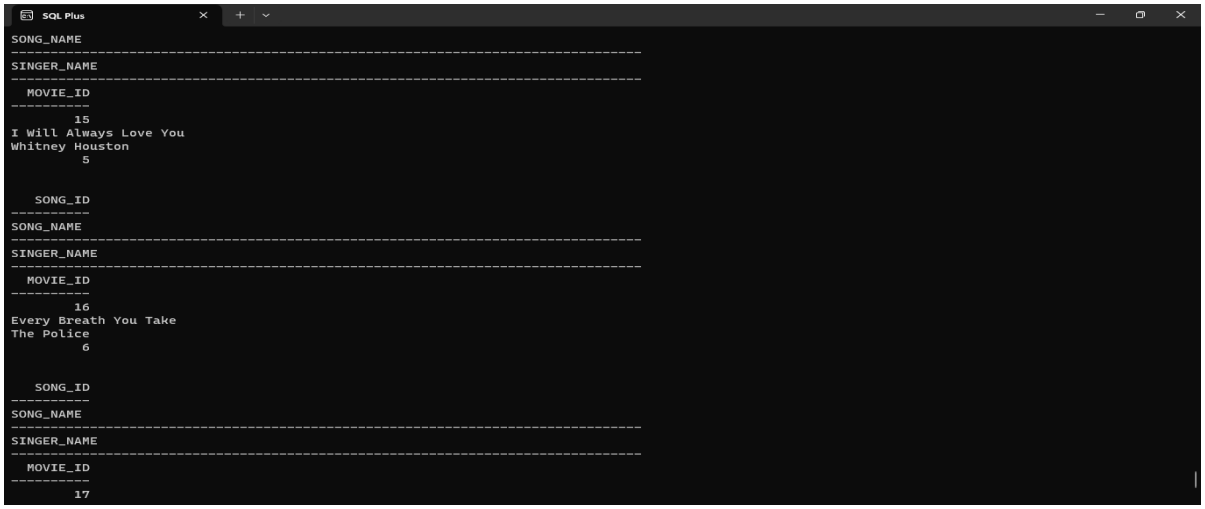

SQL Plus
MOVIE_ID
3
Bohemian Rhapsody
Queen
3
SONG_ID
SONG_NAME
SINGER_NAME
MOVIE_ID
4
Stairway to Heaven
Led Zeppelin
4
SONG_ID
SONG_NAME
SINGER_NAME
MOVIE_ID
5
Sweet Child O Mine
Guns N Roses
5

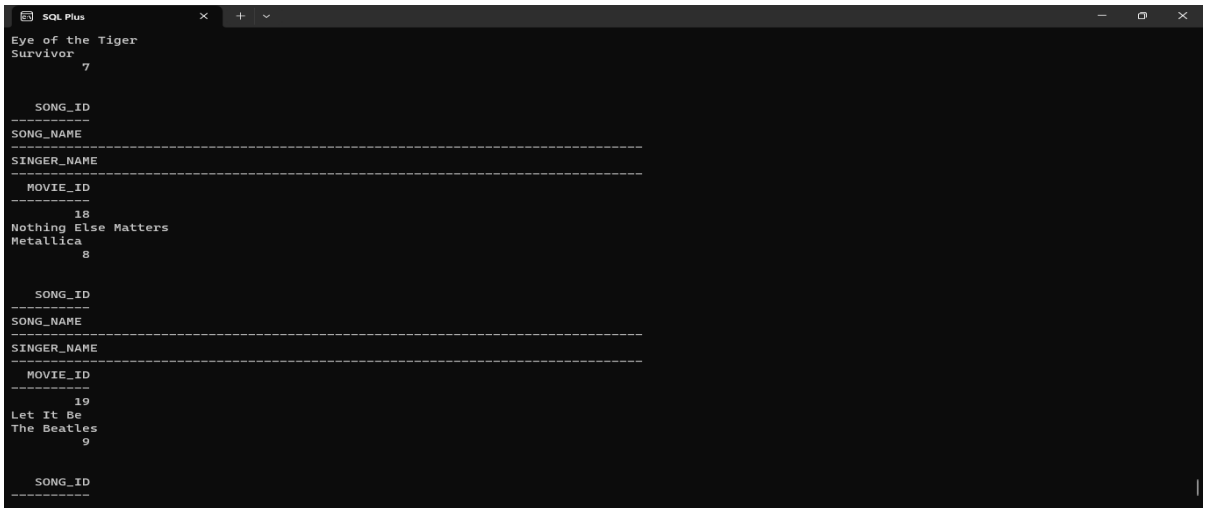

SQL Plus
5
SONG_ID
SONG_NAME
SINGER_NAME
MOVIE_ID
6
Smells Like Teen Spirit
Nirvana
6
SONG_ID
SONG_NAME
SINGER_NAME
MOVIE_ID
7
Hotel California
Eagles
7
SONG_ID
SONG_NAME

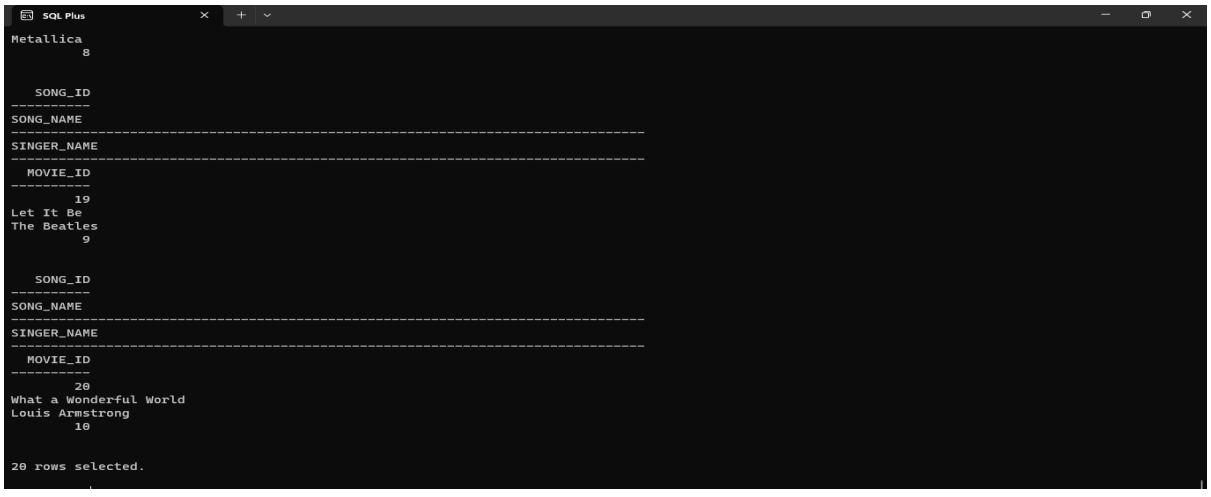

SQL Plus
SINGER_NAME
MOVIE_ID
8
Dont Stop Believin
Journey
8
SONG_ID
SONG_NAME
SINGER_NAME
MOVIE_ID
9
Thriller
Michael Jackson
9
SONG_ID
SONG_NAME
SINGER_NAME
MOVIE_ID
10
Wonderwall


SQL Plus
Wonderwall
Oasis
10
SONG_ID
SONG_NAME
SINGER_NAME
MOVIE_ID
11
Uptown Funk
Mark Ronson ft. Bruno Mars
1
SONG_ID
SONG_NAME
SINGER_NAME
MOVIE_ID
12
Smooth
Santana ft. Rob Thomas
2
SONG_ID


SQL Plus
SONG_ID
SONG_NAME
SINGER_NAME
MOVIE_ID
13
Livin on a Prayer
Bon Jovi
3
SONG_ID
SONG_NAME
SINGER_NAME
MOVIE_ID
14
Hello
Adele
4
SONG_ID
SONG_NAME
SINGER_NAME







Genre:

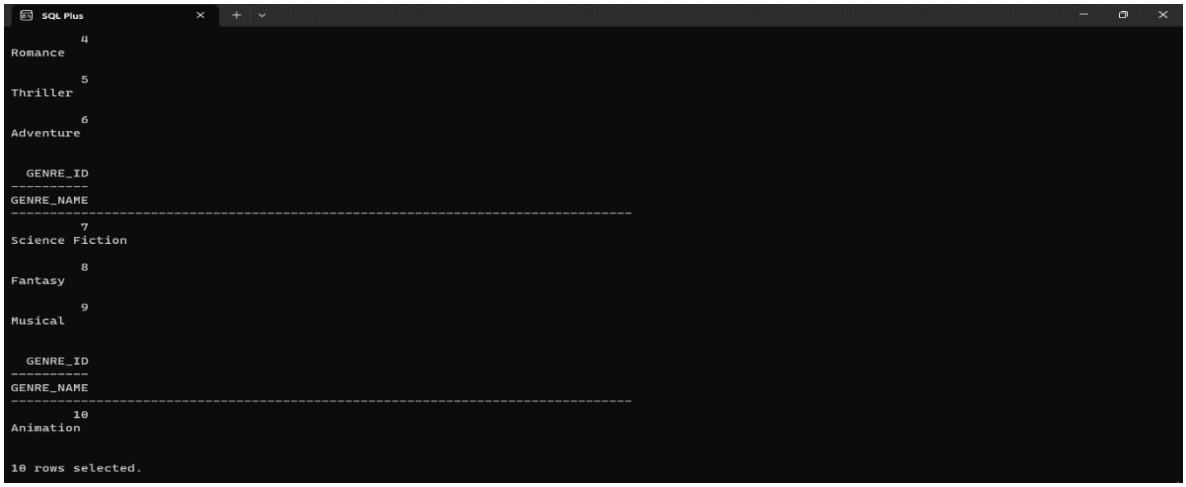
```
CREATE TABLE Genre (
  genre_id INT NOT NULL,
  genre_name VARCHAR(255) NOT NULL,
  PRIMARY KEY (genre_id)
);
```

```
SQL> DROP TABLE Genre CASCADE CONSTRAINTS;

Table dropped.

SQL> rem  +-----------------------------------------------------------------------+
SQL> rem  |  Create Genre Table                                                   |
SQL> rem  +-----------------------------------------------------------------------+
SQL>
SQL> CREATE TABLE Genre (
  2      genre_id INT NOT NULL,
  3      genre_name VARCHAR(255) NOT NULL,
  4      PRIMARY KEY (genre_id)
  5  );

Table created.

SQL>
```

```
SQL Plus
  2      genre_id INT NOT NULL,
  3      genre_name VARCHAR(255) NOT NULL,
  4      PRIMARY KEY (genre_id)
  5  );

Table created.

SQL> INSERT INTO Genre (genre_id, genre_name) VALUES (1, 'Action');

1 row created.

SQL> INSERT INTO Genre (genre_id, genre_name) VALUES (2, 'Comedy');

1 row created.

SQL> INSERT INTO Genre (genre_id, genre_name) VALUES (3, 'Drama');

1 row created.

SQL> INSERT INTO Genre (genre_id, genre_name) VALUES (4, 'Romance');

1 row created.

SQL> INSERT INTO Genre (genre_id, genre_name) VALUES (5, 'Thriller');

1 row created.

SQL> INSERT INTO Genre (genre_id, genre_name) VALUES (6, 'Adventure');

1 row created.

SQL> INSERT INTO Genre (genre_id, genre_name) VALUES (7, 'Science Fiction');

1 row created.

SQL> INSERT INTO Genre (genre_id, genre_name) VALUES (8, 'Fantasy');

1 row created.

SQL> INSERT INTO Genre (genre_id, genre_name) VALUES (9, 'Musical');

1 row created.

SQL> INSERT INTO Genre (genre_id, genre_name) VALUES (10, 'Animation');

1 row created.

SQL>
```

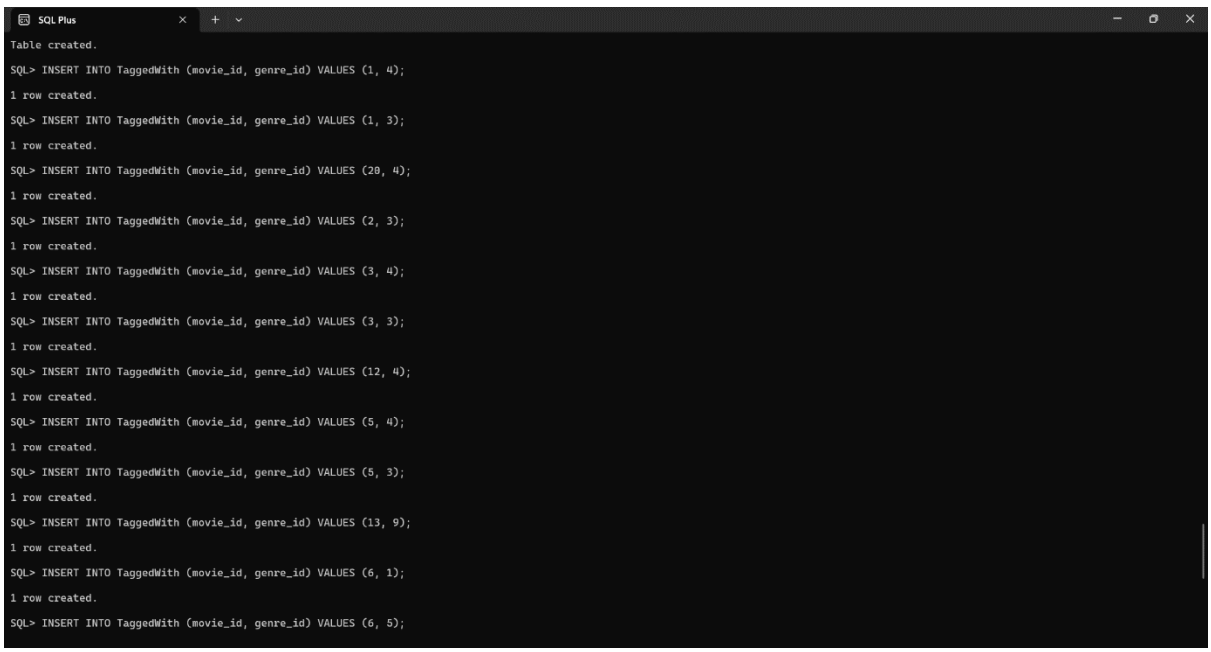
OUTPUT:

```
SQL Plus
SQL> select * from  Genre;

  GENRE_ID
----------
GENRE_NAME
--------------------------------------------------------------------------------
         1
Action

         2
Comedy

         3
Drama


  GENRE_ID
----------
GENRE_NAME
--------------------------------------------------------------------------------
         4
Romance

         5
Thriller

         6
Adventure


  GENRE_ID
----------
GENRE_NAME
--------------------------------------------------------------------------------
         7
```



Tagged With:

```
CREATE TABLE TaggedWith (
  movie_id INT NOT NULL,
  genre_id INT NOT NULL,
  PRIMARY KEY (movie_id, genre_id),
  FOREIGN KEY (movie_id) REFERENCES Movie(movie_id),
  FOREIGN KEY (genre_id) REFERENCES Genre(genre_id)
);
```



```
SQL Plus

1 row created.

SQL> INSERT INTO TaggedWith (movie_id, genre_id) VALUES (6, 5);

1 row created.

SQL> INSERT INTO TaggedWith (movie_id, genre_id) VALUES (11, 1);

1 row created.

SQL> INSERT INTO TaggedWith (movie_id, genre_id) VALUES (7, 5);

1 row created.

SQL> INSERT INTO TaggedWith (movie_id, genre_id) VALUES (8, 6);

1 row created.

SQL> INSERT INTO TaggedWith (movie_id, genre_id) VALUES (8, 7);

1 row created.

SQL> INSERT INTO TaggedWith (movie_id, genre_id) VALUES (13, 8);

1 row created.

SQL> INSERT INTO TaggedWith (movie_id, genre_id) VALUES (9, 6);

1 row created.

SQL> INSERT INTO TaggedWith (movie_id, genre_id) VALUES (9, 7);

1 row created.

SQL> INSERT INTO TaggedWith (movie_id, genre_id) VALUES (14, 8);

1 row created.

SQL> INSERT INTO TaggedWith (movie_id, genre_id) VALUES (10, 1);

1 row created.

SQL> INSERT INTO TaggedWith (movie_id, genre_id) VALUES (10, 2);

1 row created.

SQL>
```

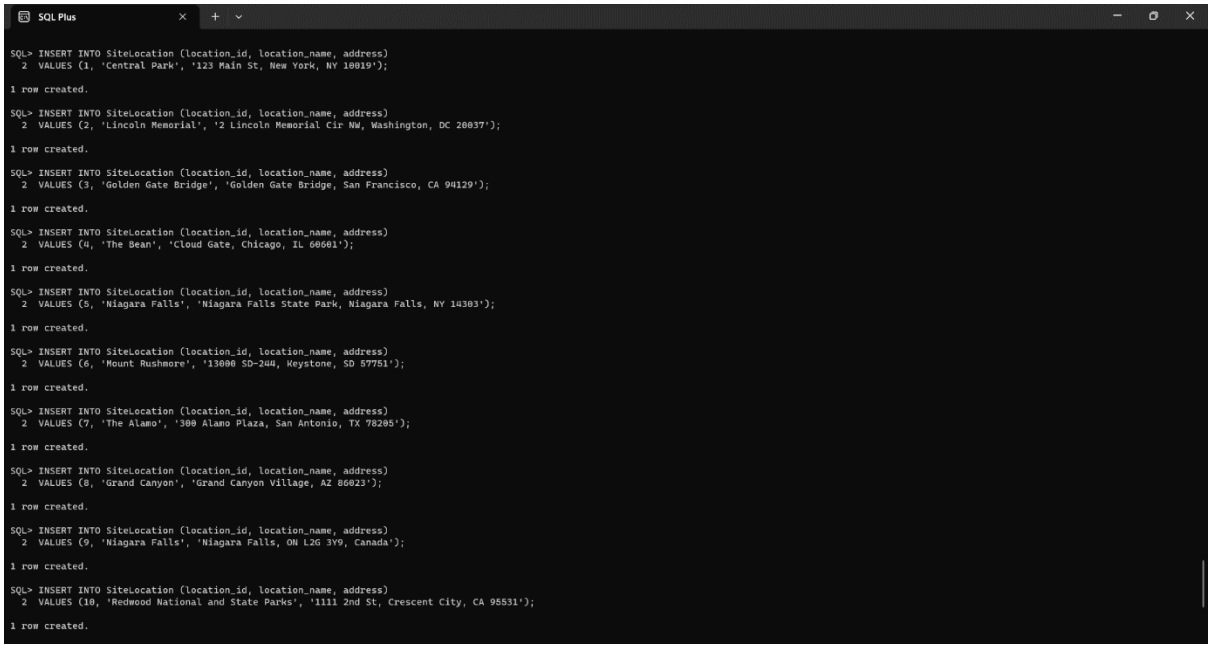
OUTPUT:

```
SQL> select * from TaggedWith;

  MOVIE_ID   GENRE_ID
---------- ----------
         1          4
         1          3
        20          4
         2          3
         3          4
         3          3
        12          4
         5          4
         5          3
        13          9
         6          1

  MOVIE_ID   GENRE_ID
---------- ----------
         6          5
        11          1
         7          5
         8          6
         8          7
        13          8
         9          6
         9          7
        14          8
        10          1
        10          2

22 rows selected.
```

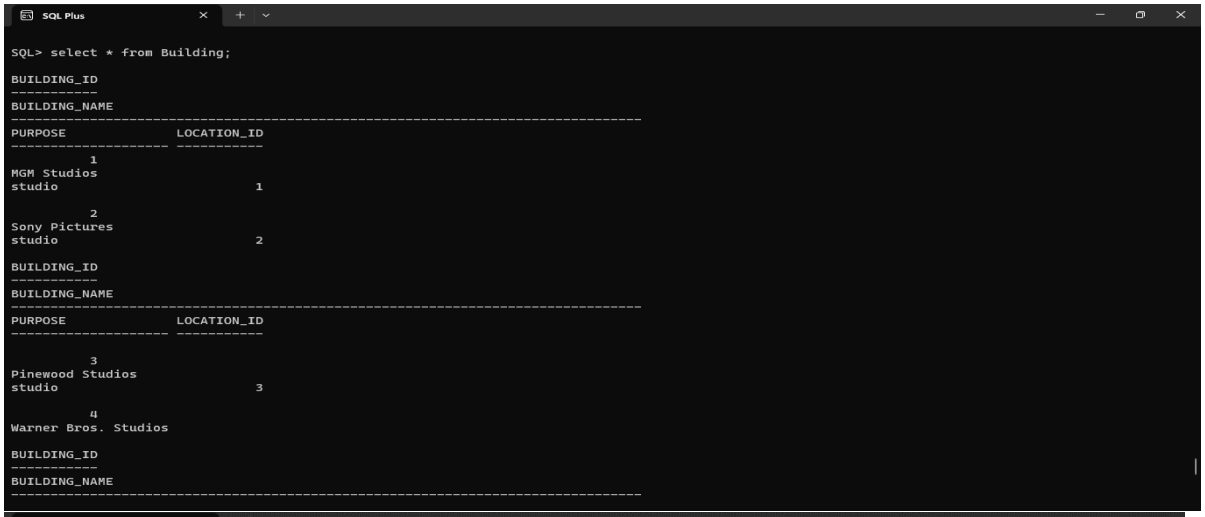
SiteLocation:

```
CREATE TABLE SiteLocation (
  location_id INT NOT NULL,
  location_name VARCHAR(255) NOT NULL,
  address VARCHAR(255) NOT NULL,
  PRIMARY KEY (location_id)
);
```

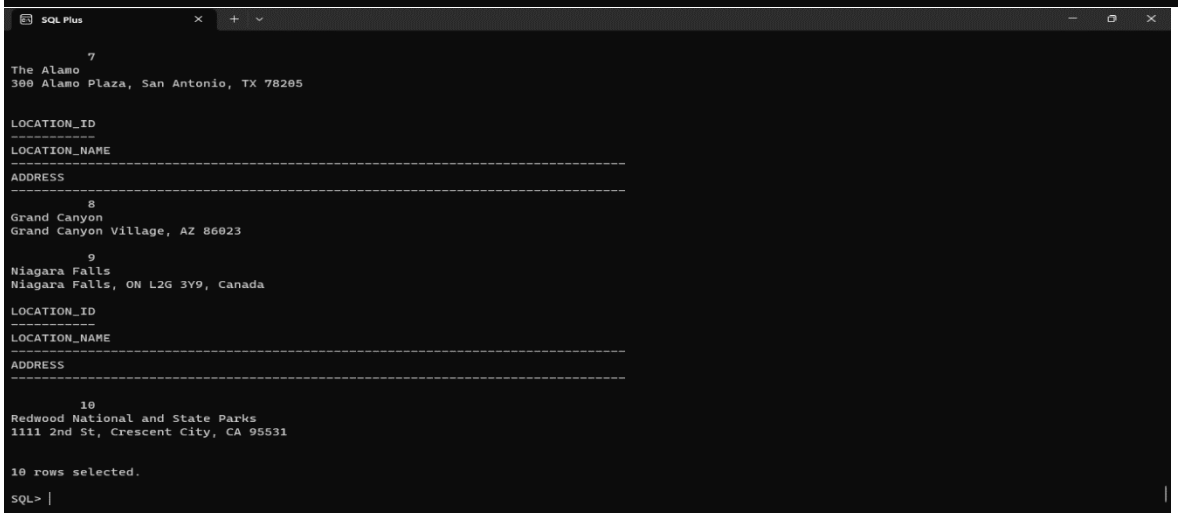
```
SQL> DROP TABLE SiteLocation CASCADE CONSTRAINTS;

Table dropped.

SQL> rem  +-----------------------------------------------------------------------+
SQL> rem  |  Create SiteLocation Table                                            |
SQL> rem  +-----------------------------------------------------------------------+
SQL>
SQL> CREATE TABLE SiteLocation (
  2      location_id INT NOT NULL,
  3      location_name VARCHAR(255) NOT NULL,
  4      address VARCHAR(255) NOT NULL,
  5      PRIMARY KEY (location_id)
  6  );

Table created.

SQL>
```


SQL Plus
SQL> INSERT INTO SiteLocation (location_id, location_name, address)
2 VALUES (1, 'Central Park', '123 Main St, New York, NY 10019');
1 row created.
SQL> INSERT INTO SiteLocation (location_id, location_name, address)
2 VALUES (2, 'Lincoln Memorial', '2 Lincoln Memorial Cir NW, Washington, DC 20037');
1 row created.
SQL> INSERT INTO SiteLocation (location_id, location_name, address)
2 VALUES (3, 'Golden Gate Bridge', 'Golden Gate Bridge, San Francisco, CA 94129');
1 row created.
SQL> INSERT INTO SiteLocation (location_id, location_name, address)
2 VALUES (4, 'The Bean', 'Cloud Gate, Chicago, IL 60601');
1 row created.
SQL> INSERT INTO SiteLocation (location_id, location_name, address)
2 VALUES (5, 'Niagara Falls', 'Niagara Falls State Park, Niagara Falls, NY 14303');
1 row created.
SQL> INSERT INTO SiteLocation (location_id, location_name, address)
2 VALUES (6, 'Mount Rushmore', '13000 SD-244, Keystone, SD 57751');
1 row created.
SQL> INSERT INTO SiteLocation (location_id, location_name, address)
2 VALUES (7, 'The Alamo', '300 Alamo Plaza, San Antonio, TX 78205');
1 row created.
SQL> INSERT INTO SiteLocation (location_id, location_name, address)
2 VALUES (8, 'Grand Canyon', 'Grand Canyon Village, AZ 86023');
1 row created.
SQL> INSERT INTO SiteLocation (location_id, location_name, address)
2 VALUES (9, 'Niagara Falls', 'Niagara Falls, ON L2G 3Y9, Canada');
1 row created.
SQL> INSERT INTO SiteLocation (location_id, location_name, address)
2 VALUES (10, 'Redwood National and State Parks', '1111 2nd St, Crescent City, CA 95531');
1 row created.

OUTPUT:


SQL Plus
SQL> select * from Building;
BUILDING_ID
BUILDING_NAME
PURPOSE LOCATION_ID
1
MGM Studios
studio 1
2
Sony Pictures
studio 2
BUILDING_ID
BUILDING_NAME
PURPOSE LOCATION_ID
3
Pinewood Studios
studio 3
4
Warner Bros. Studios
BUILDING_ID
BUILDING_NAME


SQL Plus
SQL> select * from SiteLocation;
LOCATION_ID
LOCATION_NAME
ADDRESS
1
Central Park
123 Main St, New York, NY 10019
2
Lincoln Memorial
2 Lincoln Memorial Cir NW, Washington, DC 20037
LOCATION_ID
LOCATION_NAME
ADDRESS
3
Golden Gate Bridge
Golden Gate Bridge, San Francisco, CA 94129
4
The Bean
LOCATION_ID
LOCATION_NAME



Building:

```
CREATE TABLE Building (
    building_id INT NOT NULL,
    building_name VARCHAR(255) NOT NULL,
    purpose VARCHAR(20) NOT NULL,
    location_id INT NOT NULL,
    PRIMARY KEY (building_id),
    FOREIGN KEY (location_id) REFERENCES SiteLocation(location_id),
   CONSTRAINT purpose_constraint_violated CHECK (purpose IN ('production', 'post-production', 'office', 'studio', 'storage', 'visual effects', 'color grading', 'sound design', 'mixing')));
```

```
SQL> DROP TABLE Building CASCADE CONSTRAINTS;

Table dropped.

SQL> rem  +----------------------------------------------------------------------+
SQL> rem  |  Create Building Table                                               |
SQL> rem  +----------------------------------------------------------------------+
SQL>
SQL> CREATE TABLE Building (
  2      building_id INT NOT NULL,
  3      building_name VARCHAR(255) NOT NULL,
  4      purpose VARCHAR(20) NOT NULL,
  5      location_id INT NOT NULL,
  6      PRIMARY KEY (building_id),
  7      FOREIGN KEY (location_id) REFERENCES SiteLocation(location_id),
  8      CONSTRAINT purpose_constraint_violated CHECK (
  9          purpose IN (
 10              'production',
 11              'post-production',
 12              'office',
 13              'studio',
 14              'storage',
 15              'visual effects',
 16              'color grading',
 17              'sound design',
 18              'mixing'
 19          )
 20      )
 21  );

Table created.

SQL>
```

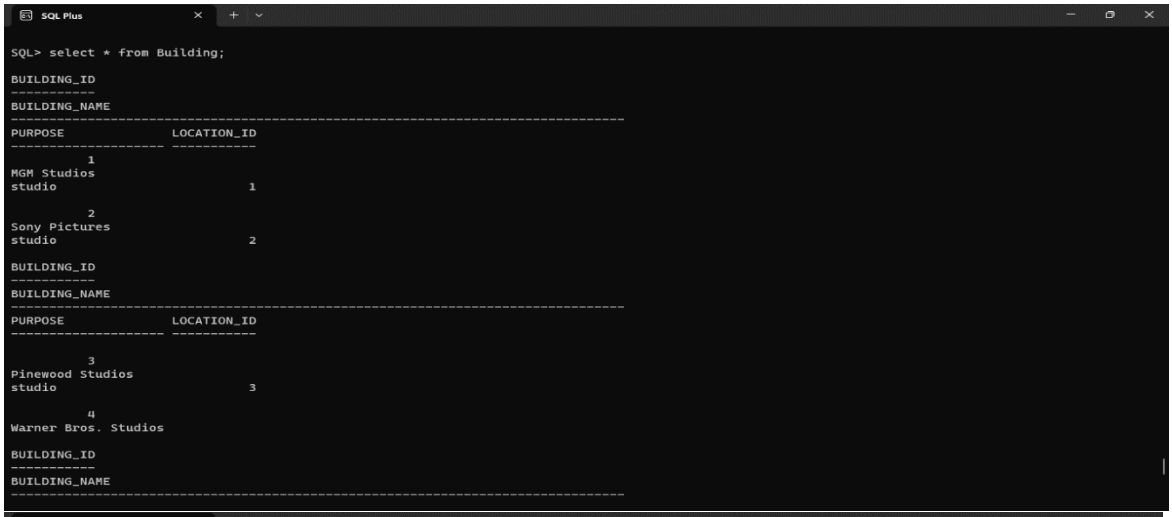
SQL Plus

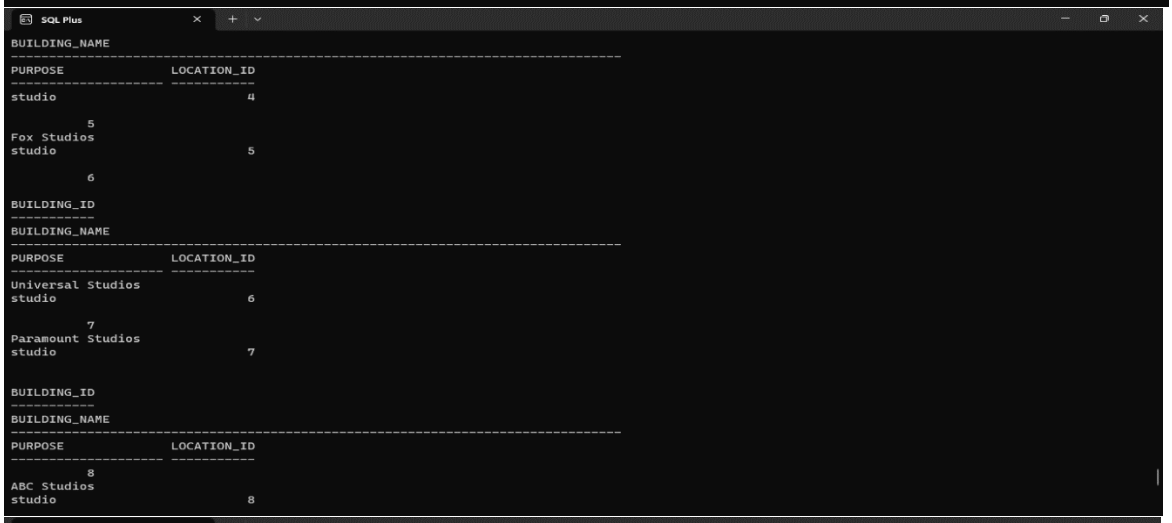
```
SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (1, 'MGM Studios', 'studio', 1);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (2, 'Sony Pictures', 'studio', 2);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (3, 'Pinewood Studios', 'studio', 3);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (4, 'Warner Bros. Studios', 'studio', 4);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (5, 'Fox Studios', 'studio', 5);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (6, 'Universal Studios', 'studio', 6);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (7, 'Paramount Studios', 'studio', 7);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (8, 'ABC Studios', 'studio', 8);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (9, 'Columbia Pictures', 'studio', 9);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (10, 'Marvel Studios', 'studio', 10);

1 row created.
```

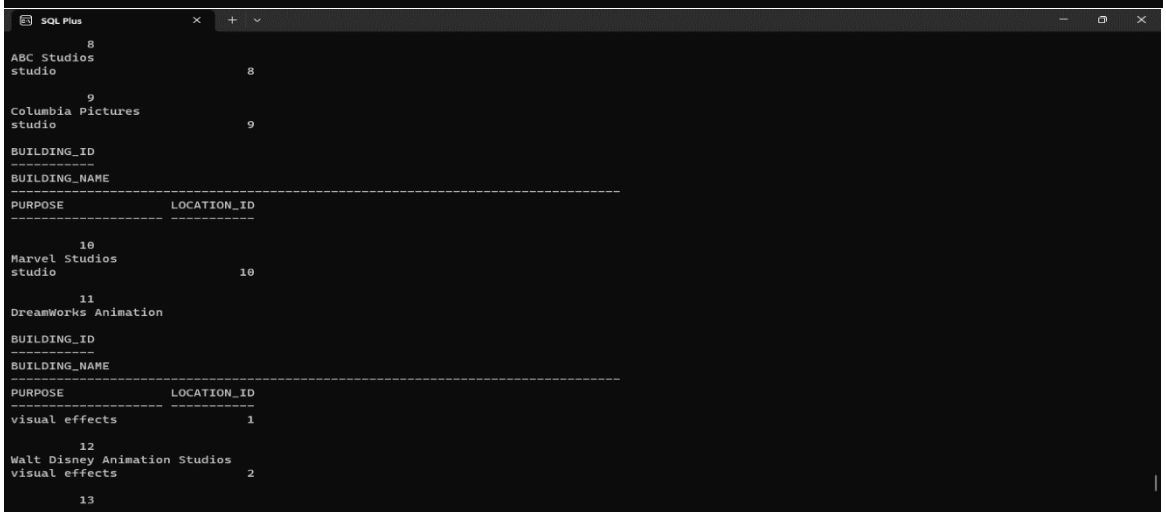
SQL Plus

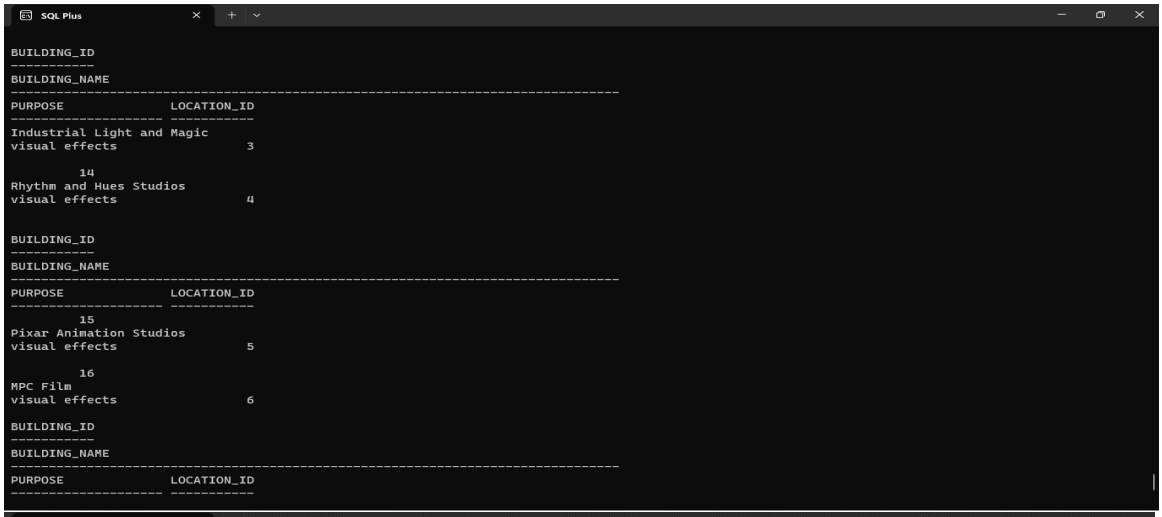
```
SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (11, 'DreamWorks Animation', 'visual effects', 1);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (12, 'Walt Disney Animation Studios', 'visual effects', 2);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (13, 'Industrial Light and Magic', 'visual effects', 3);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (14, 'Rhythm and Hues Studios', 'visual effects', 4);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (15, 'Pixar Animation Studios', 'visual effects', 5);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (16, 'MPC Film', 'visual effects', 6);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (17, 'Double Negative', 'visual effects', 7);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (18, 'Sony Pictures Imageworks', 'visual effects', 8);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (19, 'Framestore', 'visual effects', 9);

1 row created.

SQL> INSERT INTO Building (building_id, building_name, purpose, location_id)
  2  VALUES (20, 'Technicolor SA', 'color grading', 10);

1 row created.
```

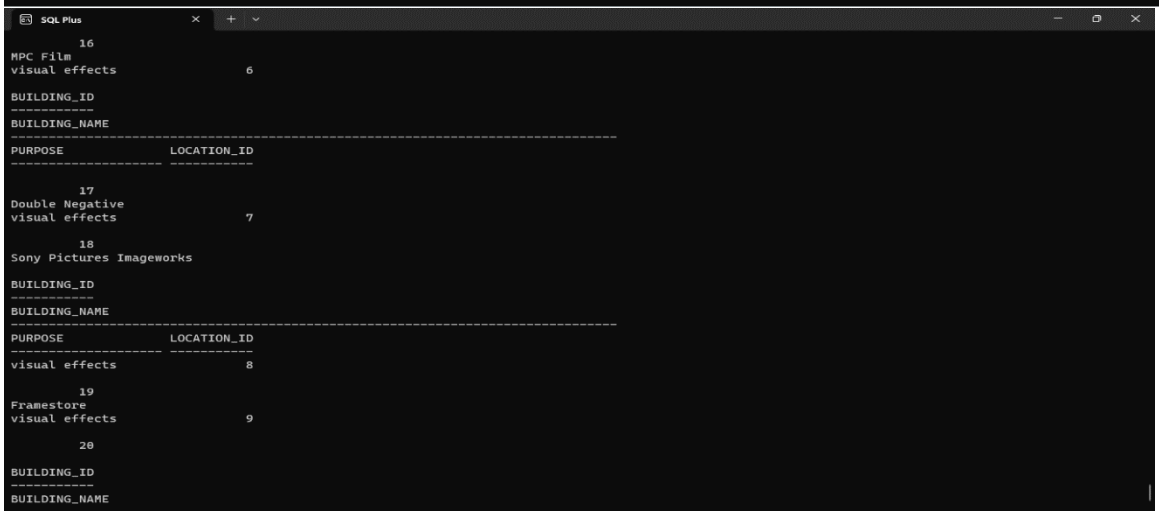
OUTPUT:

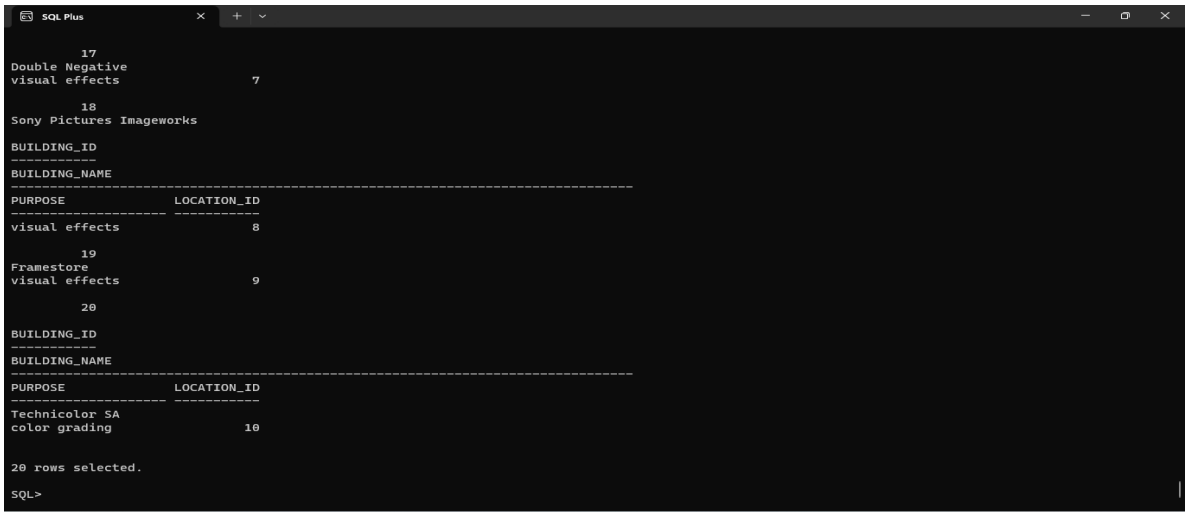

SQL Plus
SQL> select * from Building;
BUILDING_ID
BUILDING_NAME
PURPOSE LOCATION_ID
1
MGM Studios
studio 1
2
Sony Pictures
studio 2
BUILDING_ID
BUILDING_NAME
PURPOSE LOCATION_ID
3
Pinewood Studios
studio 3
4
Warner Bros. Studios
BUILDING_ID
BUILDING_NAME


SQL Plus
BUILDING_NAME
PURPOSE LOCATION_ID
studio 4
5
Fox Studios
studio 5
6
BUILDING_ID
BUILDING_NAME
PURPOSE LOCATION_ID
Universal Studios
studio 6
7
Paramount Studios
studio 7
BUILDING_ID
BUILDING_NAME
PURPOSE LOCATION_ID
8
ABC Studios
studio 8


SQL Plus
8
ABC Studios
studio 8
9
Columbia Pictures
studio 9
BUILDING_ID
BUILDING_NAME
PURPOSE LOCATION_ID
10
Marvel Studios
studio 10
11
DreamWorks Animation
BUILDING_ID
BUILDING_NAME
PURPOSE LOCATION_ID
visual effects 1
12
Walt Disney Animation Studios
visual effects 2
13







ShotAt:

```
CREATE TABLE ShotAt (
  location_id INT NOT NULL,
  movie_id INT NOT NULL,
  PRIMARY KEY (location_id, movie_id),
  FOREIGN KEY (location_id) REFERENCES SiteLocation(location_id),
  FOREIGN KEY (movie_id) REFERENCES Movie(movie_id)
```

);

```
SQL> DROP TABLE ShotAt CASCADE CONSTRAINTS;

Table dropped.

SQL> rem  +----------------------------------------------------------------------+
SQL> rem  |  Create ShotAt Table                                                 |
SQL> rem  +----------------------------------------------------------------------+
SQL>
SQL> CREATE TABLE ShotAt (
  2      location_id INT NOT NULL,
  3      movie_id INT NOT NULL,
  4      PRIMARY KEY (location_id, movie_id),
  5      FOREIGN KEY (location_id) REFERENCES SiteLocation(location_id),
  6      FOREIGN KEY (movie_id) REFERENCES Movie(movie_id)
  7  );

Table created.

SQL>
```

```
SQL Plus

SQL> INSERT INTO ShotAt (location_id, movie_id) VALUES (1, 1);

1 row created.

SQL> INSERT INTO ShotAt (location_id, movie_id) VALUES (2, 1);

1 row created.

SQL> INSERT INTO ShotAt (location_id, movie_id) VALUES (2, 2);

1 row created.

SQL> INSERT INTO ShotAt (location_id, movie_id) VALUES (3, 2);

1 row created.

SQL> INSERT INTO ShotAt (location_id, movie_id) VALUES (4, 3);

1 row created.

SQL> INSERT INTO ShotAt (location_id, movie_id) VALUES (4, 4);

1 row created.

SQL> INSERT INTO ShotAt (location_id, movie_id) VALUES (5, 5);

1 row created.

SQL> INSERT INTO ShotAt (location_id, movie_id) VALUES (6, 6);

1 row created.

SQL> INSERT INTO ShotAt (location_id, movie_id) VALUES (7, 7);

1 row created.

SQL> INSERT INTO ShotAt (location_id, movie_id) VALUES (7, 8);

1 row created.
```

OUTPUT:

```
SQL> select * from ShotAt;

LOCATION_ID   MOVIE_ID
----------- ----------
          1          1
          2          1
          2          2
          3          2
          4          3
          4          4
          5          5
          6          6
          7          7
          7          8
          7          9

LOCATION_ID   MOVIE_ID
----------- ----------
          8         10
          9         11
         10         12
         10         13
         10         14
         10         15
         10         16
         10         17
         10         18
         10         19

21 rows selected.

SQL>
```

PostProductionDoneIn:

```
CREATE TABLE PostProductionDoneIn (
  movie_id INT NOT NULL,
  building_id INT NOT NULL,
  PRIMARY KEY (movie_id, building_id),
  FOREIGN KEY (movie_id) REFERENCES Movie(movie_id),
  FOREIGN KEY (building_id) REFERENCES Building(building_id)
```

);

```
SQL> DROP TABLE PostProductionDoneIn CASCADE CONSTRAINTS;

Table dropped.

SQL> rem + ------------------------------------------------------------------------+
SQL> rem | Create PostProductionDoneIn Table                                      |
SQL> rem + ------------------------------------------------------------------------+
SQL>
SQL> CREATE TABLE PostProductionDoneIn (
  2      movie_id INT NOT NULL,
  3      building_id INT NOT NULL,
  4      PRIMARY KEY (movie_id, building_id),
  5      FOREIGN KEY (movie_id) REFERENCES Movie(movie_id),
  6      FOREIGN KEY (building_id) REFERENCES Building(building_id)
  7  );

Table created.

SQL>
```

```
SQL Plus

SQL> INSERT INTO PostProductionDoneIn (movie_id, building_id) VALUES (1, 1);

1 row created.

SQL> INSERT INTO PostProductionDoneIn (movie_id, building_id) VALUES (2, 2);

1 row created.

SQL> INSERT INTO PostProductionDoneIn (movie_id, building_id) VALUES (3, 3);

1 row created.

SQL> INSERT INTO PostProductionDoneIn (movie_id, building_id) VALUES (4, 4);

1 row created.

SQL> INSERT INTO PostProductionDoneIn (movie_id, building_id) VALUES (5, 5);

1 row created.

SQL> INSERT INTO PostProductionDoneIn (movie_id, building_id) VALUES (6, 6);

1 row created.

SQL> INSERT INTO PostProductionDoneIn (movie_id, building_id) VALUES (7, 7);

1 row created.

SQL> INSERT INTO PostProductionDoneIn (movie_id, building_id) VALUES (8, 8);

1 row created.

SQL> INSERT INTO PostProductionDoneIn (movie_id, building_id) VALUES (9, 9);

1 row created.

SQL> INSERT INTO PostProductionDoneIn (movie_id, building_id) VALUES (10, 10);

1 row created.
```

OUTPUT:

```
SQL> select * from PostProductionDoneIn;

  MOVIE_ID BUILDING_ID
---------- -----------
         1           1
         2           2
         3           3
         4           4
         5           5
         6           6
         7           7
         8           8
         9           9
        10          10
        11          11

  MOVIE_ID BUILDING_ID
---------- -----------
        12          12
        13          13
        14          14
        15          15
        16          16
        17          17
        18          18
        19          19
        20          20

20 rows selected.

SQL>
```

Employees:

```
CREATE TABLE Employees (
    employee_id INT NOT NULL,
    employee_name VARCHAR(255) NOT NULL,
    designation VARCHAR(255) NOT NULL,
```

phone_number VARCHAR(10) NOT NULL UNIQUE,
PRIMARY KEY (employee_id),
CONSTRAINT designatoin_constraint_violated CHECK (
designation IN ('choreographers', 'security','sound engineer', 'makeup artist', 'electrician', 'janitor','manager')));

```
SQL> DROP TABLE Employees CASCADE CONSTRAINTS;

Table dropped.

SQL> rem + ------------------------------------------------------------------------+
SQL> rem | Create Employees Table                                                 |
SQL> rem + ------------------------------------------------------------------------+
SQL>
SQL> CREATE TABLE Employees (
  2      employee_id INT NOT NULL,
  3      employee_name VARCHAR(255) NOT NULL,
  4      designation VARCHAR(255) NOT NULL,
  5      phone_number VARCHAR(10) NOT NULL UNIQUE,
  6      PRIMARY KEY (employee_id),
  7      CONSTRAINT designatoin_constraint_violated CHECK (
  8          designation IN (
  9              'choreographer',
 10              'security',
 11              'sound engineer',
 12              'makeup artist',
 13              'electrician',
 14              'janitor',
 15              'manager'
 16          )
 17      )
 18  );

Table created.

SQL>
```

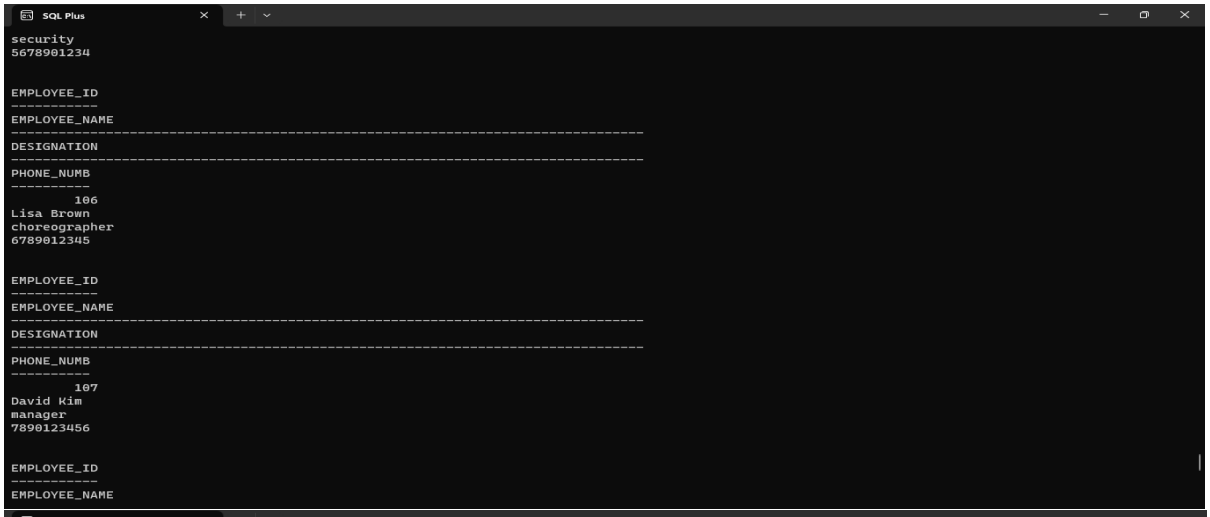
```
SQL Plus
Table created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (101, 'John Doe', 'electrician', '1234567890');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (102, 'Jane Smith', 'makeup artist', '2345678901');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (103, 'Bob Johnson', 'janitor', '3456789012');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (104, 'Sara Lee', 'sound engineer', '4567890123');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (105, 'Mike Tyson', 'security', '5678901234');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (106, 'Lisa Brown', 'choreographer', '6789012345');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (107, 'David Kim', 'manager', '7890123456');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (108, 'Jessica Lee', 'electrician', '8901234567');

1 row created.
```

```
SQL Plus

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (109, 'Jason Lee', 'makeup artist', '9012345678');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (110, 'Emily Johnson', 'janitor', '0123456789');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (111, 'Alex Lee', 'sound engineer', '1234509876');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (112, 'Alice Kim', 'security', '2345609876');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (113, 'Brian Kim', 'choreographer', '3456709876');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (114, 'Cathy Lee', 'manager', '4567809876');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (115, 'Daniel Lee', 'electrician', '5678909876');

1 row created.

SQL> INSERT INTO Employees (employee_id, employee_name, designation, phone_number)
  2  VALUES (116, 'Ellen Kim', 'makeup artist', '6789019876');

1 row created.
```

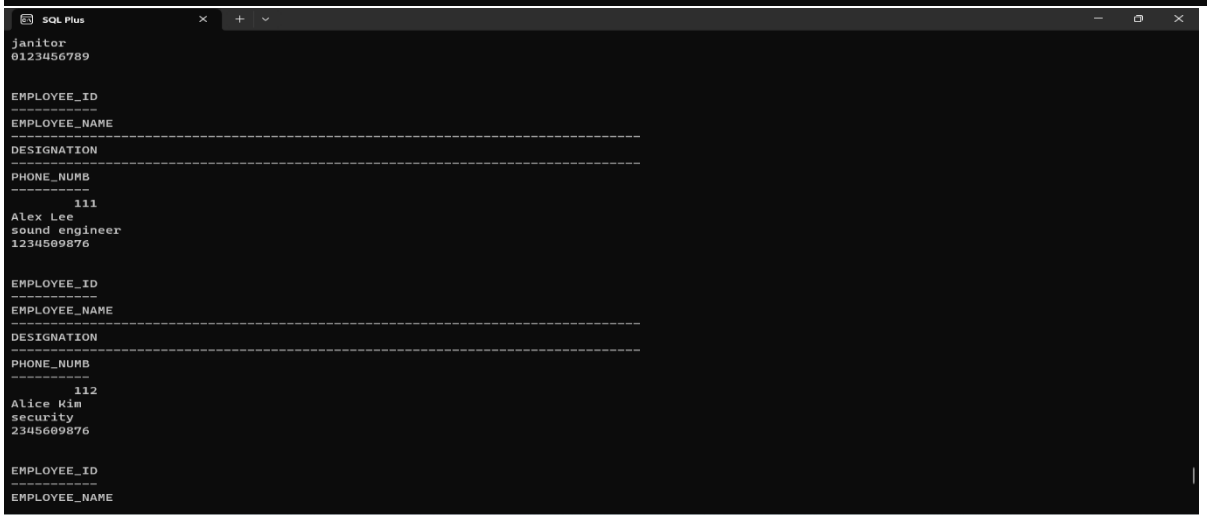
OUTPUT:

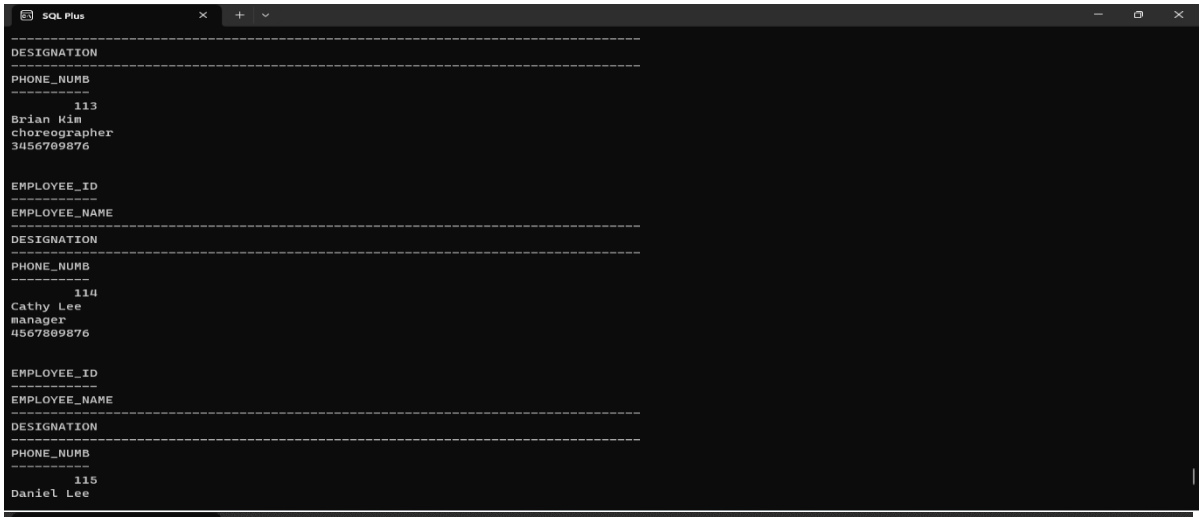
```
SQL Plus

SQL> select * from Employees;

EMPLOYEE_ID
-----------
EMPLOYEE_NAME
--------------------------------------------------------------------------------
DESIGNATION
--------------------------------------------------------------------------------
PHONE_NUMB
----------
       101
John Doe
electrician
1234567890


EMPLOYEE_ID
-----------
EMPLOYEE_NAME
--------------------------------------------------------------------------------
DESIGNATION
--------------------------------------------------------------------------------
PHONE_NUMB
----------
       102
Jane Smith
makeup artist
2345678901


EMPLOYEE_ID
-----------
EMPLOYEE_NAME
--------------------------------------------------------------------------------
DESIGNATION
```

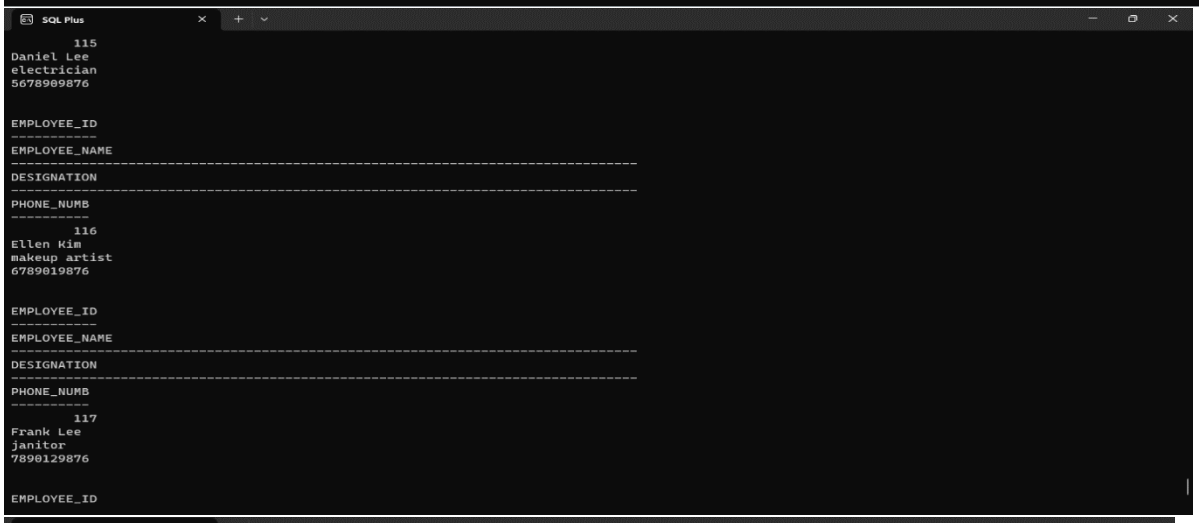
```
SQL Plus

--------------------------------------------------------------------------------
PHONE_NUMB
----------
       103
Bob Johnson
janitor
3456789012


EMPLOYEE_ID
-----------
EMPLOYEE_NAME
--------------------------------------------------------------------------------
DESIGNATION
--------------------------------------------------------------------------------
PHONE_NUMB
----------
       104
Sara Lee
sound engineer
4567890123


EMPLOYEE_ID
-----------
EMPLOYEE_NAME
--------------------------------------------------------------------------------
DESIGNATION
--------------------------------------------------------------------------------
PHONE_NUMB
----------
       105
Mike Tyson
security
```

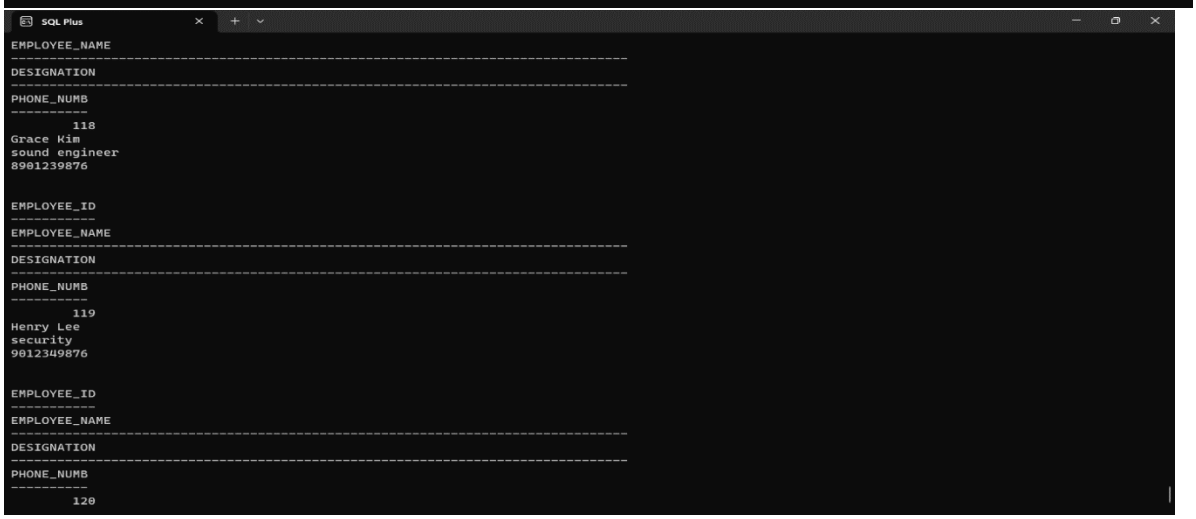

SQL Plus
security
5678901234
EMPLOYEE_ID
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
106
Lisa Brown
choreographer
6789012345
EMPLOYEE_ID
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
107
David Kim
manager
7890123456
EMPLOYEE_ID
EMPLOYEE_NAME

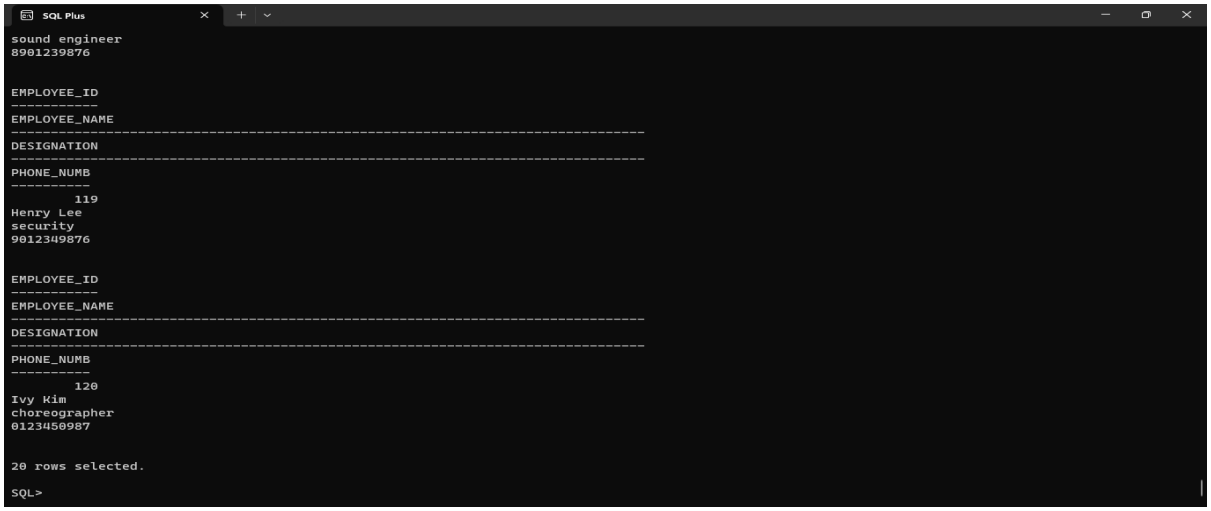

SQL Plus
DESIGNATION
PHONE_NUMB
108
Jessica Lee
electrician
8901234567
EMPLOYEE_ID
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
109
Jason Lee
makeup artist
9012345678
EMPLOYEE_ID
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
110
Emily Johnson


SQL Plus
janitor
0123456789
EMPLOYEE_ID
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
111
Alex Lee
sound engineer
1234509876
EMPLOYEE_ID
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
112
Alice Kim
security
2345609876
EMPLOYEE_ID
EMPLOYEE_NAME


SQL Plus
DESIGNATION
PHONE_NUMB
113
Brian Kim
choreographer
3456709876
EMPLOYEE_ID
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
114
Cathy Lee
manager
4567809876
EMPLOYEE_ID
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
115
Daniel Lee

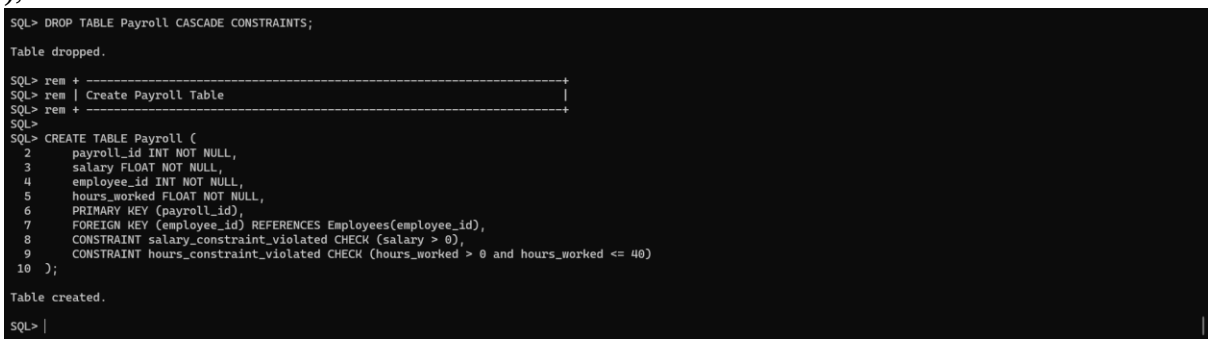

SQL Plus
115
Daniel Lee
electrician
5678909876
EMPLOYEE_ID
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
116
Ellen Kim
makeup artist
6789019876
EMPLOYEE_ID
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
117
Frank Lee
janitor
7890129876
EMPLOYEE_ID


SQL Plus
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
118
Grace Kim
sound engineer
8901239876
EMPLOYEE_ID
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
119
Henry Lee
security
9012349876
EMPLOYEE_ID
EMPLOYEE_NAME
DESIGNATION
PHONE_NUMB
120



Payroll:

```
CREATE TABLE Payroll (
    payroll_id INT NOT NULL,
    salary FLOAT NOT NULL,
    employee_id INT NOT NULL,
    hours_worked FLOAT NOT NULL,
    PRIMARY KEY (payroll_id),
    FOREIGN KEY (employee_id) REFERENCES Employees(employee_id),
    CONSTRAINT salary_constraint_violated CHECK (salary > 0),
    CONSTRAINT hours_constraint_violated CHECK (hours_worked > 0 and hours_worked
<= 40)
);
```



```
SQL Plus

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (1, 5000.00, 101, 40.00);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (2, 6000.00, 102, 38.50);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (3, 4500.00, 103, 37.00);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (4, 5500.00, 104, 39.50);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (5, 4000.00, 105, 36.00);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (6, 6500.00, 106, 40.00);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (7, 4800.00, 107, 38.50);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (8, 5200.00, 108, 37.00);

1 row created.
```

```
SQL Plus

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (9, 5800.00, 109, 39.50);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (10, 4200.00, 110, 36.00);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (11, 5100.00, 111, 40.00);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (12, 5300.00, 112, 38.50);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (13, 4700.00, 113, 37.00);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (14, 5400.00, 114, 39.50);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (15, 3900.00, 115, 36.00);

1 row created.

SQL> INSERT INTO Payroll (payroll_id, salary, employee_id, hours_worked)
  2  VALUES (16, 5900.00, 116, 40.00);

1 row created.
```

OUTPUT:

```
SQL> select * from Payroll;

PAYROLL_ID     SALARY EMPLOYEE_ID HOURS_WORKED
---------- ---------- ----------- ------------
         1       5000         101           40
         2       6000         102         38.5
         3       4500         103           37
         4       5500         104         39.5
         5       4000         105           36
         6       6500         106           40
         7       4800         107         38.5
         8       5200         108           37
         9       5800         109         39.5
        10       4200         110           36
        11       5100         111           40

PAYROLL_ID     SALARY EMPLOYEE_ID HOURS_WORKED
---------- ---------- ----------- ------------
        12       5300         112         38.5
        13       4700         113           37
        14       5400         114         39.5
        15       3900         115           36
        16       5900         116           40
        17       4600         117         38.5
        18       5000         118           37
        19       5700         119         39.5
        20       4300         120           36

20 rows selected.

SQL>
```

SponsoringCompany:

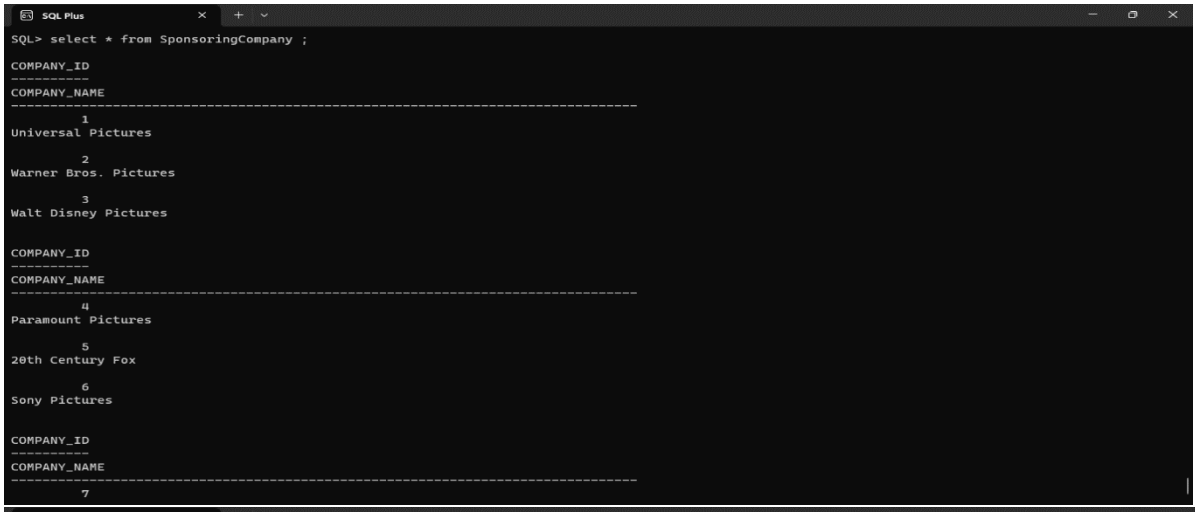
```
CREATE TABLE SponsoringCompany (
    company_id INT NOT NULL,
    company_name VARCHAR(255) NOT NULL UNIQUE,
    PRIMARY KEY (company_id)
);
```

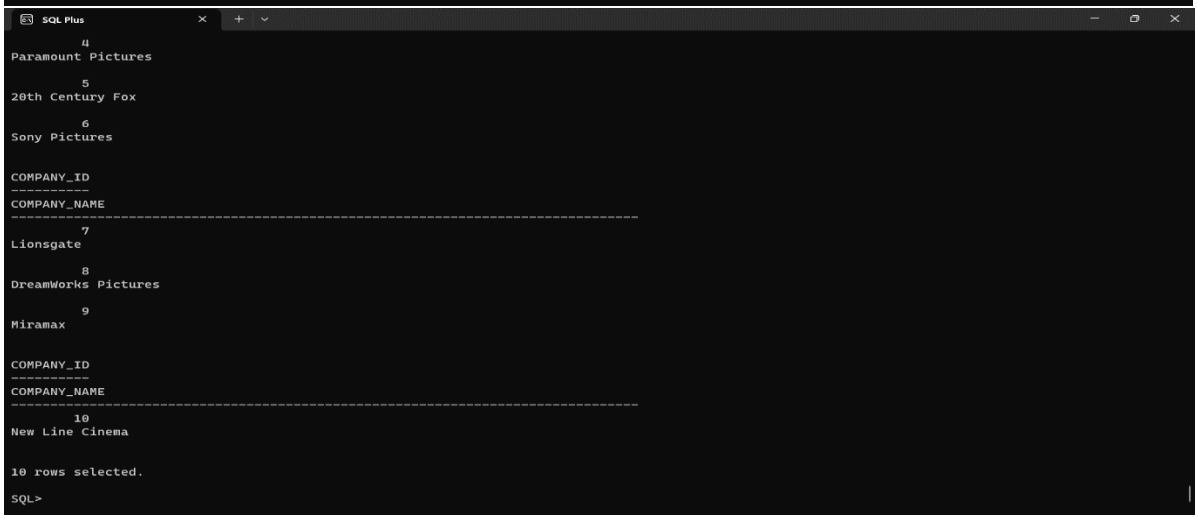
```
SQL> DROP TABLE SponsoringCompany CASCADE CONSTRAINTS;

Table dropped.

SQL> rem + ------------------------------------------------------------------------+
SQL> rem | Create SponsoringCompany Table                                         |
SQL> rem + ------------------------------------------------------------------------+
SQL>
SQL> CREATE TABLE SponsoringCompany (
  2      company_id INT NOT NULL,
  3      company_name VARCHAR(255) NOT NULL UNIQUE,
  4      PRIMARY KEY (company_id)
  5  );

Table created.

SQL>
```

```
SQL Plus
Table created.

SQL> INSERT INTO SponsoringCompany (company_id, company_name)
  2  VALUES (1, 'Universal Pictures');

1 row created.

SQL> INSERT INTO SponsoringCompany (company_id, company_name)
  2  VALUES (2, 'Warner Bros. Pictures');

1 row created.

SQL> INSERT INTO SponsoringCompany (company_id, company_name)
  2  VALUES (3, 'Walt Disney Pictures');

1 row created.

SQL> INSERT INTO SponsoringCompany (company_id, company_name)
  2  VALUES (4, 'Paramount Pictures');

1 row created.

SQL> INSERT INTO SponsoringCompany (company_id, company_name)
  2  VALUES (5, '20th Century Fox');

1 row created.

SQL> INSERT INTO SponsoringCompany (company_id, company_name)
  2  VALUES (6, 'Sony Pictures');

1 row created.

SQL> INSERT INTO SponsoringCompany (company_id, company_name)
  2  VALUES (7, 'Lionsgate');

1 row created.

SQL> INSERT INTO SponsoringCompany (company_id, company_name)
  2  VALUES (8, 'DreamWorks Pictures');

1 row created.
```

```
SQL> INSERT INTO SponsoringCompany (company_id, company_name)
  2  VALUES (9, 'Miramax');

1 row created.

SQL> INSERT INTO SponsoringCompany (company_id, company_name)
  2  VALUES (10, 'New Line Cinema');

1 row created.
```

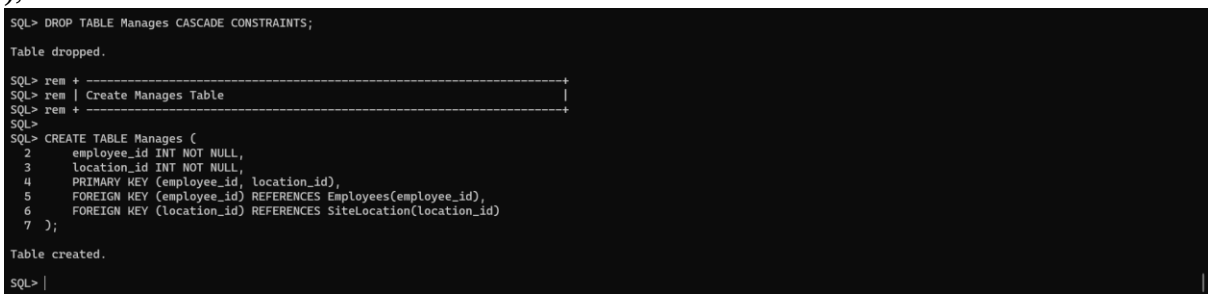
OUTPUT:





Manages:

```
CREATE TABLE Manages (
    employee_id INT NOT NULL,
    location_id INT NOT NULL,
     PRIMARY KEY (employee_id, location_id),
    FOREIGN KEY (employee_id) REFERENCES Employees(employee_id),
    FOREIGN KEY (location_id) REFERENCES SiteLocation(location_id)
);
```



```
SQL Plus

SQL> INSERT INTO Manages (employee_id, location_id) VALUES (101, 1);

1 row created.

SQL> INSERT INTO Manages (employee_id, location_id) VALUES (102, 2);

1 row created.

SQL> INSERT INTO Manages (employee_id, location_id) VALUES (103, 3);

1 row created.

SQL> INSERT INTO Manages (employee_id, location_id) VALUES (104, 4);

1 row created.

SQL> INSERT INTO Manages (employee_id, location_id) VALUES (105, 5);

1 row created.

SQL> INSERT INTO Manages (employee_id, location_id) VALUES (106, 6);

1 row created.

SQL> INSERT INTO Manages (employee_id, location_id) VALUES (107, 7);

1 row created.

SQL> INSERT INTO Manages (employee_id, location_id) VALUES (108, 8);

1 row created.

SQL> INSERT INTO Manages (employee_id, location_id) VALUES (109, 9);

1 row created.

SQL> INSERT INTO Manages (employee_id, location_id) VALUES (110, 10);

1 row created.
```

OUTPUT:

```
SQL> select * from Manages;

EMPLOYEE_ID LOCATION_ID
----------- -----------
        101           1
        102           2
        103           3
        104           4
        105           5
        106           6
        107           7
        108           8
        109           9
        110          10
        101           2

EMPLOYEE_ID LOCATION_ID
----------- -----------
        102           3
        103           4
        104           5
        105           6
        106           7
        107           8
        108           9
        109          10
        110           1

20 rows selected.

SQL>
```

Artist:

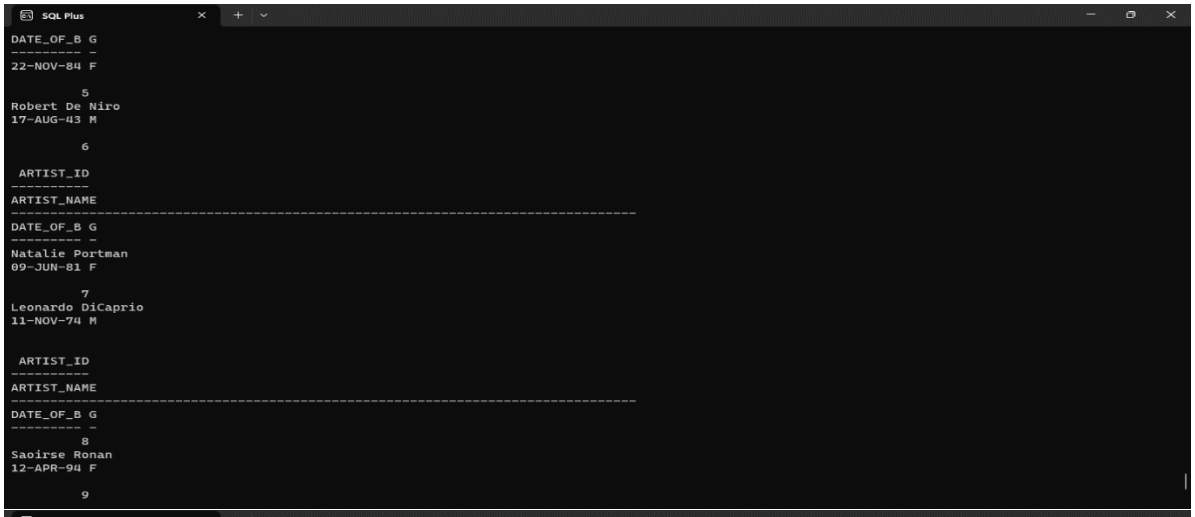
```
CREATE TABLE Artist (
  artist_id INT PRIMARY KEY,
  artist_name VARCHAR(255) NOT NULL,
  date_of_birth DATE NOT NULL,
  gender VARCHAR(1) NOT NULL,
  CONSTRAINT gender_constraint_violated CHECK (gender in ('M', 'F', 'T'))
);
```

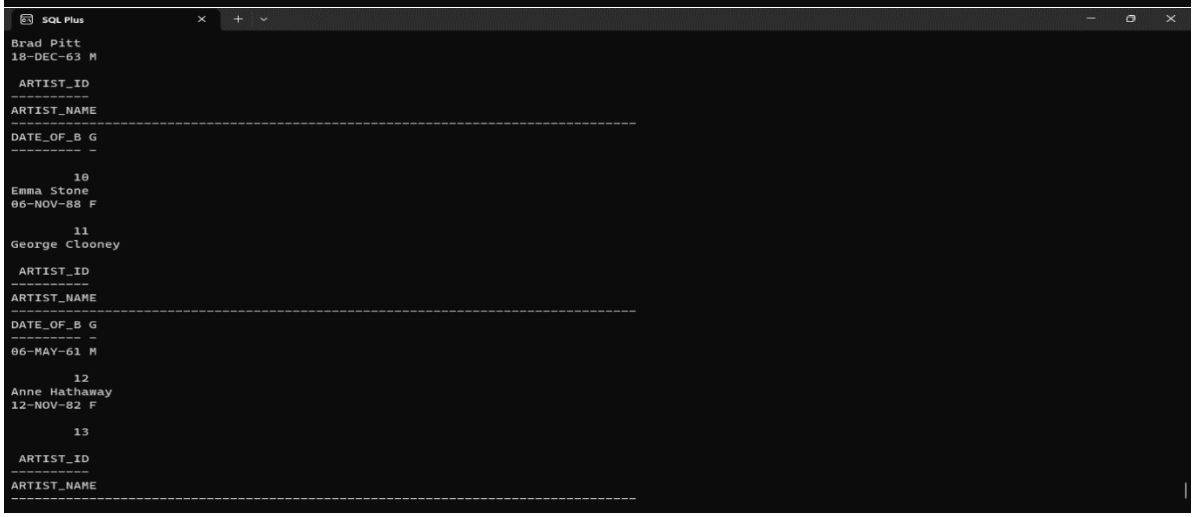
```
SQL> DROP TABLE Artist CASCADE CONSTRAINTS;

Table dropped.

SQL> rem + ------------------------------------------------------------------------+
SQL> rem | Create Artist Table                                                     |
SQL> rem + ------------------------------------------------------------------------+
SQL>
SQL> CREATE TABLE Artist (
  2    artist_id INT PRIMARY KEY,
  3    artist_name VARCHAR(255) NOT NULL,
  4    date_of_birth DATE NOT NULL,
  5    gender VARCHAR(1) NOT NULL,
  6    CONSTRAINT gender_constraint_violated CHECK (gender in ('M', 'F', 'T'))
  7  );

Table created.

SQL>
```

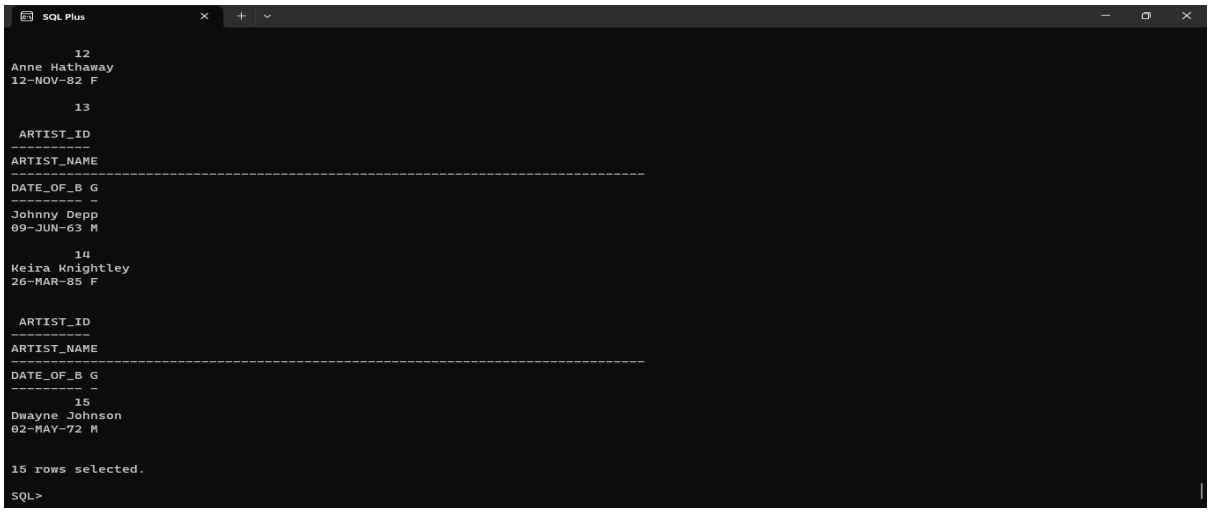
```
SQL Plus

SQL> INSERT INTO Artist (artist_id, artist_name, date_of_birth, gender) VALUES
  2  (1, 'Tom Hanks', to_date('07-09-1956', 'MM-DD-YYYY'), 'M');

1 row created.

SQL> INSERT INTO Artist (artist_id, artist_name, date_of_birth, gender) VALUES
  2  (2, 'Meryl Streep', to_date('06-22-1949', 'MM-DD-YYYY'), 'F');

1 row created.

SQL> INSERT INTO Artist (artist_id, artist_name, date_of_birth, gender) VALUES
  2  (3, 'Denzel Washington', to_date('12-28-1954', 'MM-DD-YYYY'), 'M');

1 row created.

SQL> INSERT INTO Artist (artist_id, artist_name, date_of_birth, gender) VALUES
  2  (4, 'Scarlett Johansson', to_date('11-22-1984', 'MM-DD-YYYY'), 'F');

1 row created.

SQL> INSERT INTO Artist (artist_id, artist_name, date_of_birth, gender) VALUES
  2  (5, 'Robert De Niro', to_date('08-17-1943', 'MM-DD-YYYY'), 'M');

1 row created.

SQL> INSERT INTO Artist (artist_id, artist_name, date_of_birth, gender) VALUES
  2  (6, 'Natalie Portman', to_date('06-09-1981', 'MM-DD-YYYY'), 'F');

1 row created.

SQL> INSERT INTO Artist (artist_id, artist_name, date_of_birth, gender) VALUES
  2  (7, 'Leonardo DiCaprio', to_date('11-11-1974', 'MM-DD-YYYY'), 'M');

1 row created.

SQL> INSERT INTO Artist (artist_id, artist_name, date_of_birth, gender) VALUES
  2  (8, 'Saoirse Ronan', to_date('04-12-1994', 'MM-DD-YYYY'), 'F');

1 row created.
```

OUTPUT:

```
SQL Plus

SQL> select * from Artist;

 ARTIST_ID
----------
ARTIST_NAME
--------------------------------------------------------------------------------
DATE_OF_B G
--------- -
         1
Tom Hanks
09-JUL-56 M

         2
Meryl Streep
22-JUN-49 F

 ARTIST_ID
----------
ARTIST_NAME
--------------------------------------------------------------------------------
DATE_OF_B G
--------- -

         3
Denzel Washington
28-DEC-54 M

         4
Scarlett Johansson

 ARTIST_ID
----------
ARTIST_NAME
--------------------------------------------------------------------------------
DATE_OF_B G
```







GetsPaidBy:

```
CREATE TABLE getsPaidBy (
  artist_id INT NOT NULL,
  company_id INT NOT NULL,
  PRIMARY KEY (artist_id, company_id),
  FOREIGN KEY (artist_id) REFERENCES Artist(artist_id),
  FOREIGN KEY (company_id) REFERENCES SponsoringCompany(company_id)
```

);

```
SQL> DROP TABLE getsPaidBy CASCADE CONSTRAINTS;

Table dropped.

SQL> rem + ------------------------------------------------------------------------+
SQL> rem | Create getsPaidBy Table                                                 |
SQL> rem + ------------------------------------------------------------------------+
SQL>
SQL> CREATE TABLE getsPaidBy (
  2      artist_id INT NOT NULL,
  3      company_id INT NOT NULL,
  4      PRIMARY KEY (artist_id, company_id),
  5      FOREIGN KEY (artist_id) REFERENCES Artist(artist_id),
  6      FOREIGN KEY (company_id) REFERENCES SponsoringCompany(company_id)
  7  );

Table created.

SQL>
```

```
SQL Plus

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (1, 4);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (1, 5);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (2, 3);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (2, 4);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (2, 5);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (3, 1);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (3, 2);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (3, 3);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (3, 4);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (4, 1);

1 row created.
```

```
SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (4, 2);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (4, 3);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (4, 4);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (5, 1);

1 row created.

SQL> INSERT INTO getsPaidBy (artist_id, company_id) VALUES (5, 2);

1 row created.
```

OUTPUT:

```
SQL> select * from getsPaidBy;

 ARTIST_ID COMPANY_ID
---------- ----------
         1          4
         1          5
         2          3
         2          4
         2          5
         3          1
         3          2
         3          3
         3          4
         4          1
         4          2

 ARTIST_ID COMPANY_ID
---------- ----------
         4          3
         4          4
         5          1
         5          2

15 rows selected.

SQL>
```

Produces:

```
CREATE TABLE Produces (
  company_id INT NOT NULL,
  movie_id INT NOT NULL,
  PRIMARY KEY (company_id, movie_id),
  FOREIGN KEY (company_id) REFERENCES SponsoringCompany(company_id),
  FOREIGN KEY (movie_id) REFERENCES Movie(movie_id)
);
```

```
SQL> DROP TABLE Produces CASCADE CONSTRAINTS;

Table dropped.

SQL> rem + ------------------------------------------------------------------+
SQL> rem | Create Produces Table                                            |
SQL> rem + ------------------------------------------------------------------+
SQL>
SQL> CREATE TABLE Produces (
  2      company_id INT NOT NULL,
  3      movie_id INT NOT NULL,
  4      PRIMARY KEY (company_id, movie_id),
  5      FOREIGN KEY (company_id) REFERENCES SponsoringCompany(company_id),
  6      FOREIGN KEY (movie_id) REFERENCES Movie(movie_id)
  7  );

Table created.

SQL>
```

```
SQL Plus
SQL> INSERT INTO Produces (company_id, movie_id) VALUES (1, 1);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (2, 3);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (3, 6);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (4, 9);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (5, 12);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (6, 14);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (7, 16);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (8, 18);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (9, 19);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (10, 20);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (2, 1);
```

```
SQL Plus
SQL> INSERT INTO Produces (company_id, movie_id) VALUES (2, 1);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (3, 2);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (4, 3);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (5, 4);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (6, 5);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (7, 6);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (8, 7);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (9, 8);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (10, 9);

1 row created.

SQL> INSERT INTO Produces (company_id, movie_id) VALUES (1, 10);

1 row created.

SQL>
```

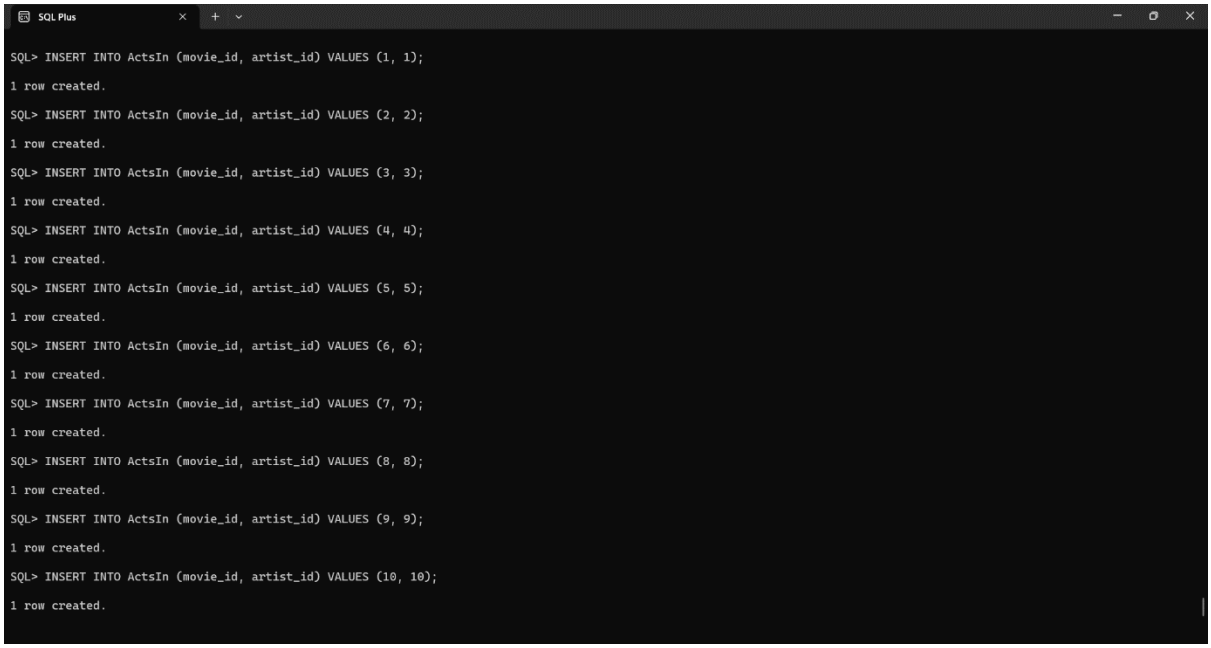
OUTPUT:

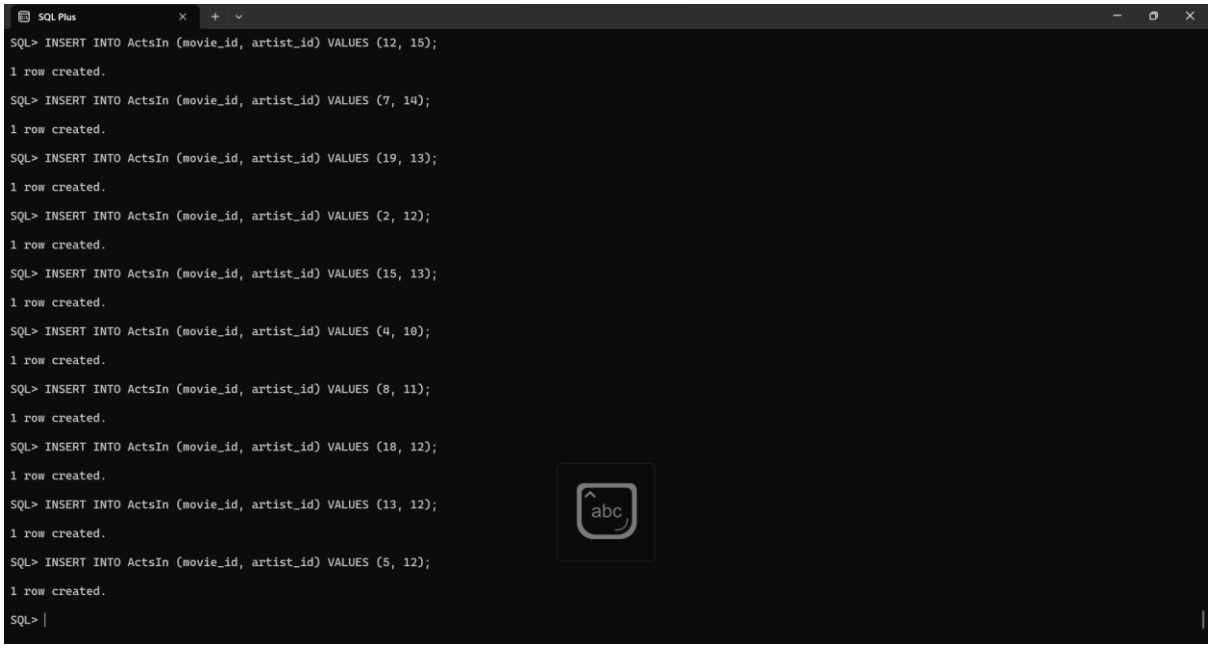
```
SQL> select * from Produces;

COMPANY_ID   MOVIE_ID
---------- ----------
         1          1
         2          3
         3          6
         4          9
         5         12
         6         14
         7         16
         8         18
         9         19
        10         20
         2          1

COMPANY_ID   MOVIE_ID
---------- ----------
         3          2
         4          3
         5          4
         6          5
         7          6
         8          7
         9          8
        10          9
         1         10

20 rows selected.

SQL>
```

ActsIn:

```
CREATE TABLE ActsIn (
   movie_id INT NOT NULL,
   artist_id INT NOT NULL,
   PRIMARY KEY (movie_id, artist_id),
   FOREIGN KEY (movie_id) REFERENCES Movie(movie_id),
   FOREIGN KEY (artist_id) REFERENCES Artist(artist_id)
);
```

```
SQL> DROP TABLE ActsIn CASCADE CONSTRAINTS;

Table dropped.

SQL> rem + ----------------------------------------------------------------------+
SQL> rem | Create ActsIn Table                                                  |
SQL> rem + ----------------------------------------------------------------------+
SQL>
SQL> CREATE TABLE ActsIn (
  2      movie_id INT NOT NULL,
  3      artist_id INT NOT NULL,
  4      PRIMARY KEY (movie_id, artist_id),
  5      FOREIGN KEY (movie_id) REFERENCES Movie(movie_id),
  6      FOREIGN KEY (artist_id) REFERENCES Artist(artist_id)
  7  );

Table created.

SQL>
```


SQL Plus
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (1, 1);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (2, 2);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (3, 3);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (4, 4);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (5, 5);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (6, 6);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (7, 7);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (8, 8);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (9, 9);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (10, 10);
1 row created.


SQL Plus
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (12, 15);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (7, 14);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (19, 13);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (2, 12);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (15, 13);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (4, 10);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (8, 11);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (18, 12);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (13, 12);
1 row created.
SQL> INSERT INTO ActsIn (movie_id, artist_id) VALUES (5, 12);
1 row created.
SQL>

OUTPUT:


SQL> select * from ActsIn;
MOVIE_ID ARTIST_ID
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
12 15
MOVIE_ID ARTIST_ID
7 14
19 13
2 12
15 13
4 10
8 11
18 12
13 12
5 12
20 rows selected.
SQL>

Individual Contribution:

Here I have created the tables Buildings and Movie-Script-Inventory, Included the foreign key(location_id), primary key(building_id), check(purpose), and NotNull constraints. I have inserted 20 tuples into the buildings tables ans also 10 tuples into movie_script_inventory table.

Obtained the tables according to inserted values in both tables.