FINAL PROJECT REPORT

IST 659

Group Zimin

Aditya Sanjay Pawar, Zimin Zhu, Yixing Zhu, Tian Tian

December 11, 2023

Introduction

Collecting and sifting large amounts of data for comparison is essential in the movie-making industry. In the pre-production phase of a movie, producers often need to select potentially suitable directors, crew members, actors, and locations based on the budget or script (Clevé, 2006). Therefore, the goal of this project is to apply the techniques and skills of this course to meet the needs of producers in the pre-screening process and to help clients better analyze and more appropriately and efficiently allocate the funds needed for their film projects. The business problems we hope to solve in this project are:

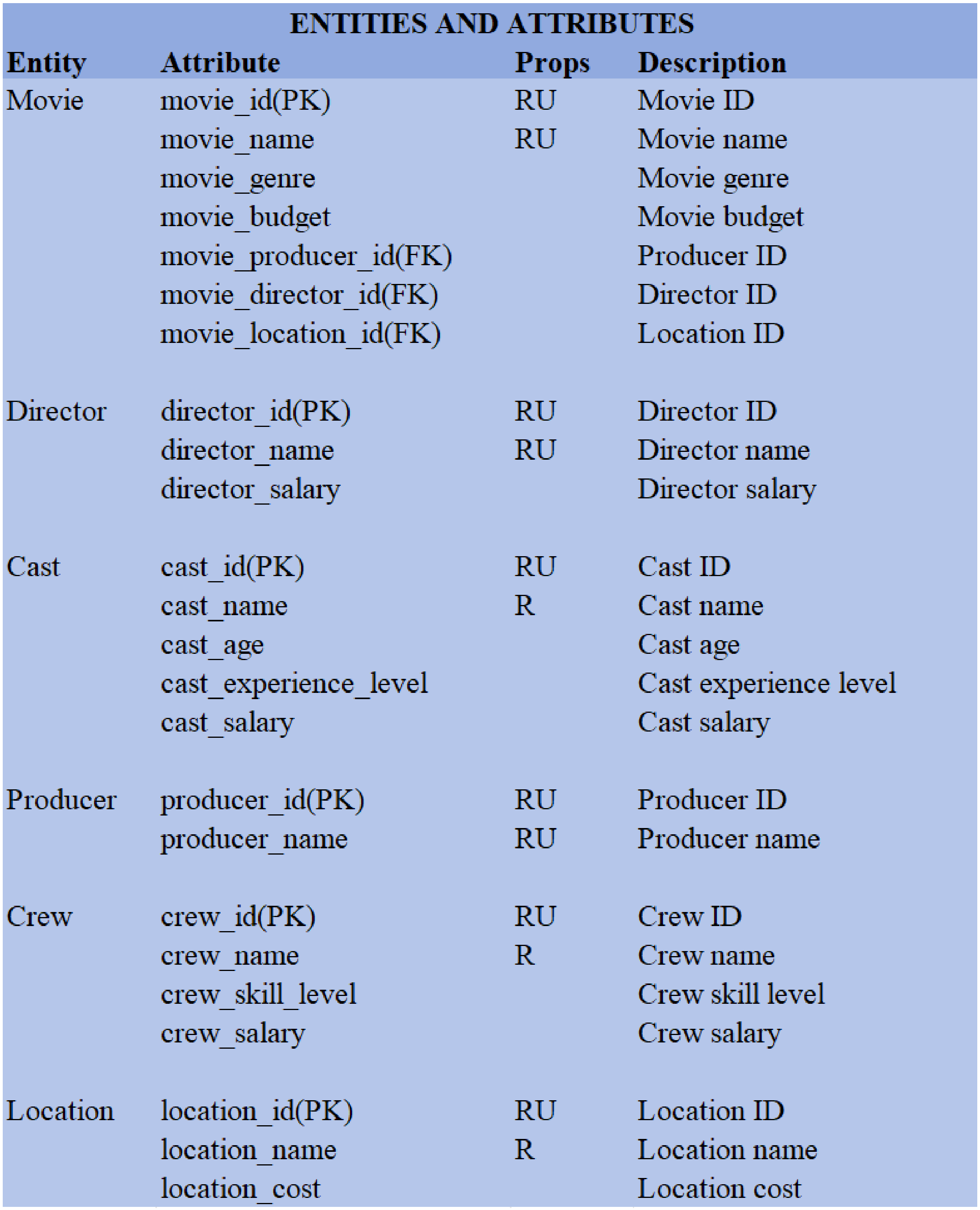
1. What is the range of expenses required for specific types of movies (action, comedy, etc.)?
2. Which actors and actresses have starred in certain specific types of movies and were hired on lower budgets?
3. Which shoots took place at locations that cost less than budget?
4. Which crew members are experienced in making certain types of movies and are paid lower wages?
5. Which directors have experience making certain types of movies and how much do they get paid?

Data Analysis

Our dataset consists of movie information, producer information, director information, crew member information, cast information, and location information. We refer to the information required for movie production on the Internet Movie Database (IMDb) and American Film Institute (AFI) and select some of them as our entities. Most of the information is randomly generated data. We created 8 tables based on the dataset: crews, producers, locations, movies, directors, casts, movie_cast, and movie_crew. the last two tables are related tables, which are used in many-to-many relationships to connect the crews table to the movies table and the casts table to the movies table.

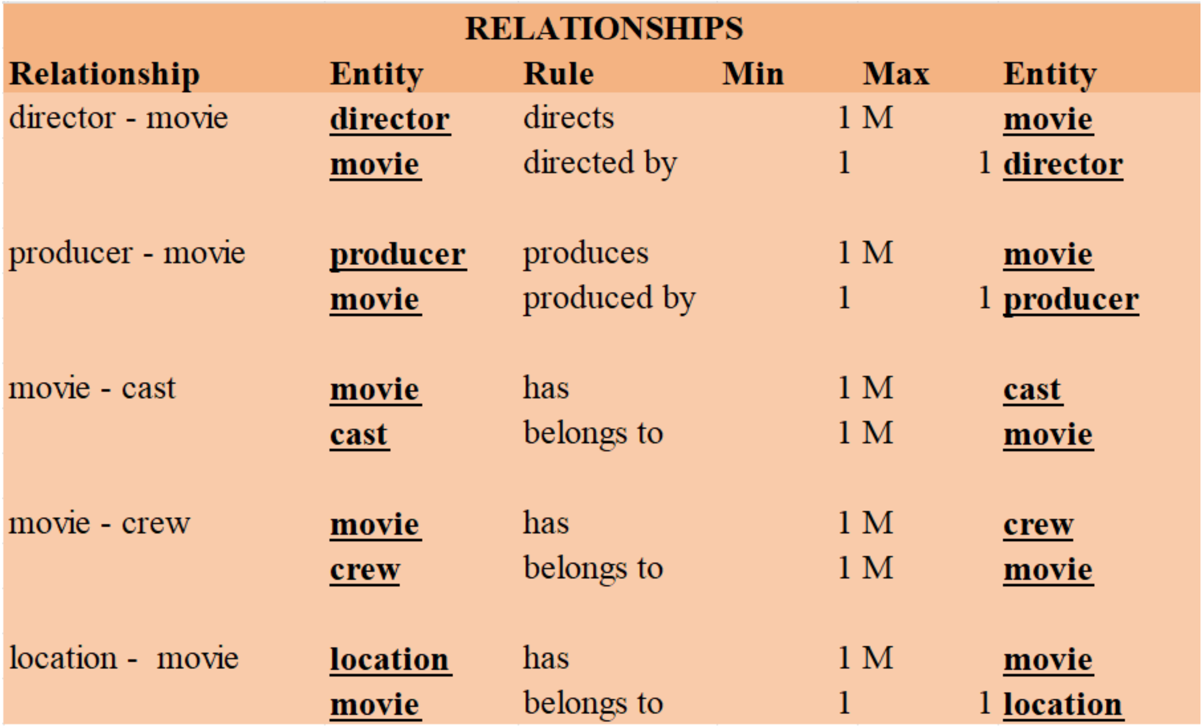
The logical and conceptual data model diagrams (Figure 1 & Figure 2) were created using diagrams.net.

Table 1 *Entities and Attributes*



ENTITIES AND ATTRIBUTES			
Entity	**Attribute**	**Props**	**Description**
Movie	movie_id(PK)	RU	Movie ID
	movie_name	RU	Movie name
	movie_genre		Movie genre
	movie_budget		Movie budget
	movie_producer_id(FK)		Producer ID
	movie_director_id(FK)		Director ID
	movie_location_id(FK)		Location ID
Director	director_id(PK)	RU	Director ID
	director_name	RU	Director name
	director_salary		Director salary
Cast	cast_id(PK)	RU	Cast ID
	cast_name	R	Cast name
	cast_age		Cast age
	cast_experience_level		Cast experience level
	cast_salary		Cast salary
Producer	producer_id(PK)	RU	Producer ID
	producer_name	RU	Producer name
Crew	crew_id(PK)	RU	Crew ID
	crew_name	R	Crew name
	crew_skill_level		Crew skill level
	crew_salary		Crew salary
Location	location_id(PK)	RU	Location ID
	location_name	R	Location name
	location_cost		Location cost

Table 2 *Relationships*



RELATIONSHIPS					
Relationship	**Entity**	**Rule**	**Min**	**Max**	**Entity**
director - movie	**director**	directs	1	M	**movie**
	movie	directed by	1	1	**director**
producer - movie	**producer**	produces	1	M	**movie**
	movie	produced by	1	1	**producer**
movie - cast	**movie**	has	1	M	**cast**
	cast	belongs to	1	M	**movie**
movie - crew	**movie**	has	1	M	**crew**
	crew	belongs to	1	M	**movie**
location - movie	**location**	has	1	M	**movie**
	movie	belongs to	1	1	**location**

Figure 1 *Conceptual Model of Film Production*

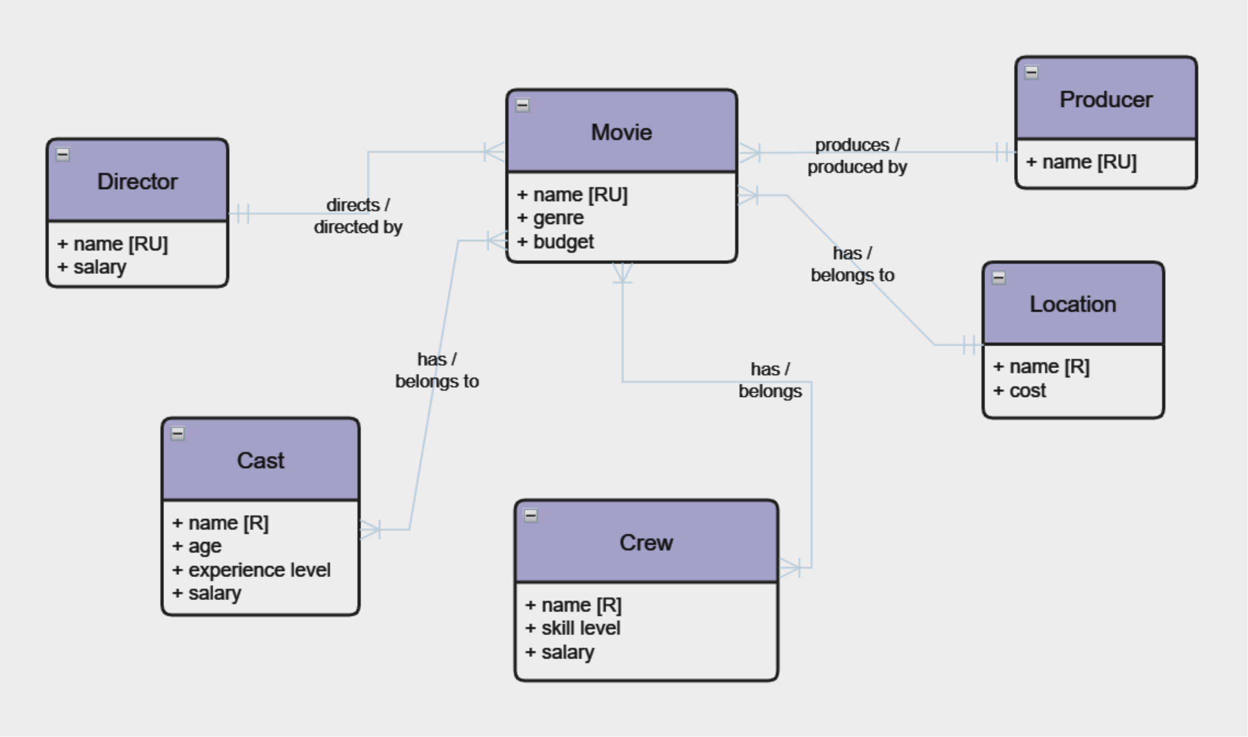
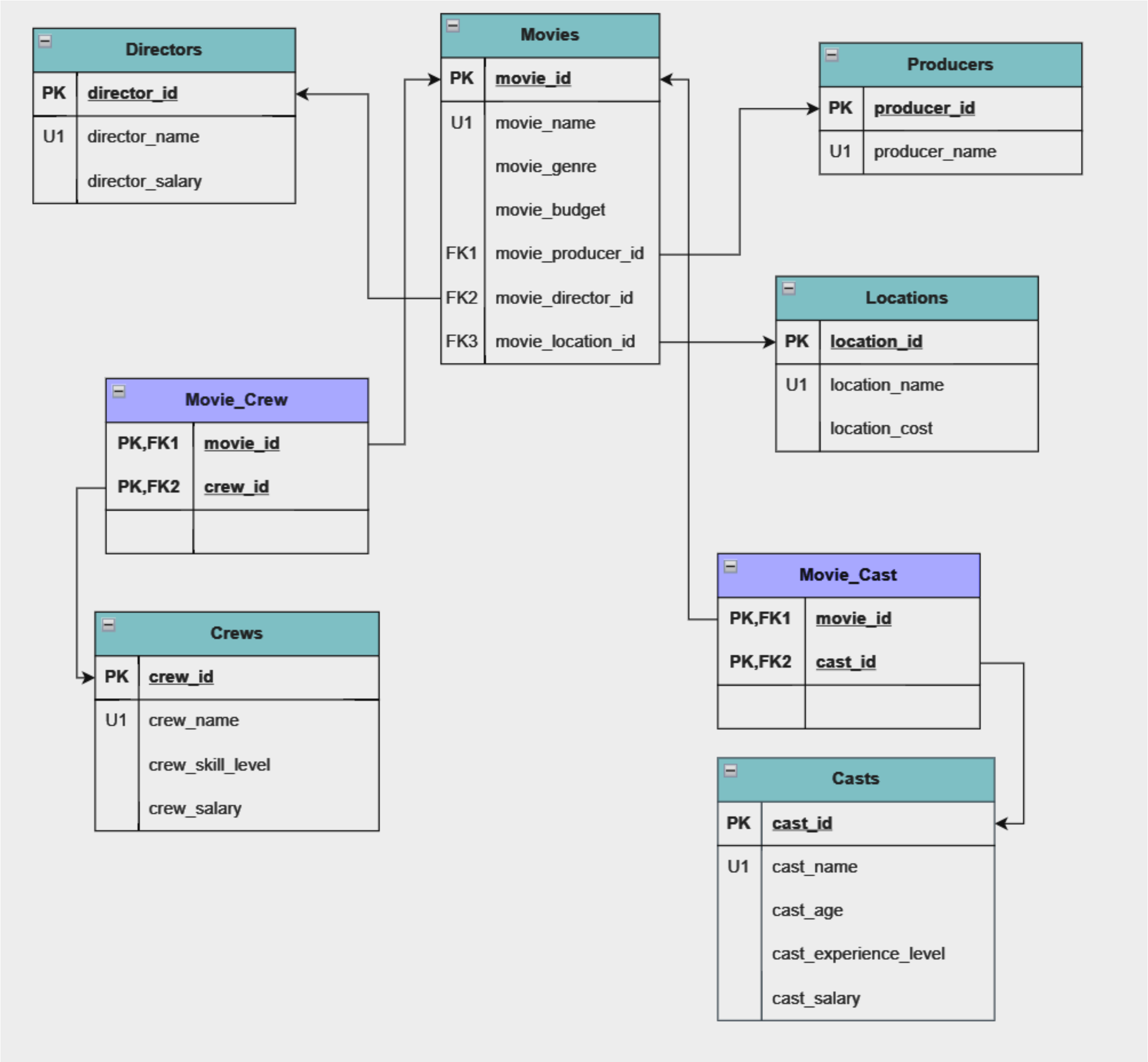


Figure 2 *Logical Model of Film Production*

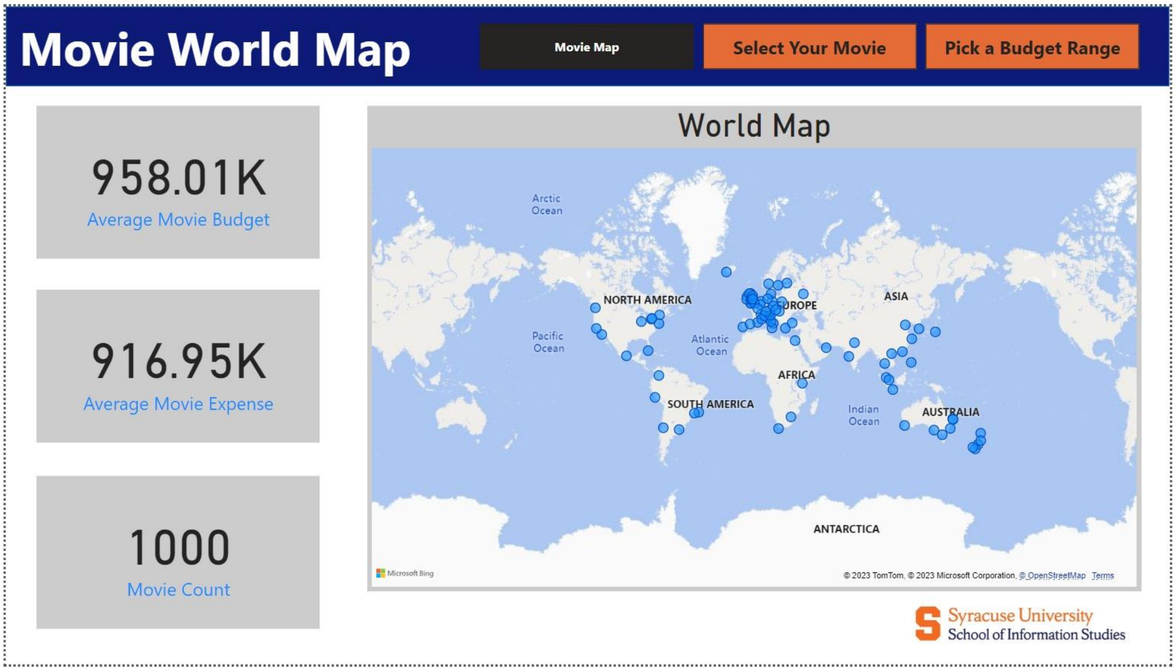


Visualization and Application

We chose PowerBI for data visualization because it is very easy to use and has very diverse features. With our application, users can select the "movie map" option to see a map of the locations of a thousand movies on a global map (Figure 3), as well as the average production budget and average spending of those movies.

Next, selecting "select your movie" allows the user to make more detailed filters (Figure 4), such as selecting a movie genre or title, and learning about the movie's production information. Finally, selecting "pick a budget range" allows the user to select a budget range according to their needs (Figure 5), and to see how the cost of movie genres within that budget range compares to the approximate budget.

Figure 3 *Movie World Map*



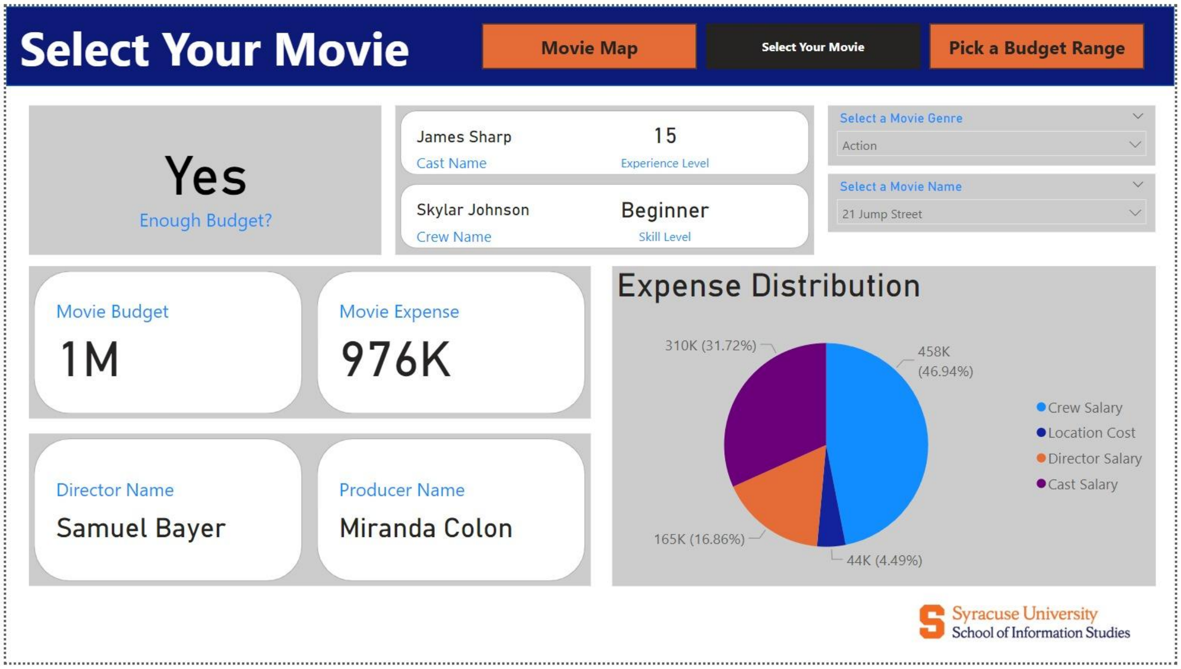
Movie World Map

958.01K
Average Movie Budget

916.95K
Average Movie Expense

1000
Movie Count

Figure 4 *Detailed Movie Information*



Select Your Movie

Yes
Enough Budget?

James Sharp — Cast Name
15 — Experience Level

Skylar Johnson — Crew Name
Beginner — Skill Level

Select a Movie Genre: Action

Select a Movie Name: 21 Jump Street

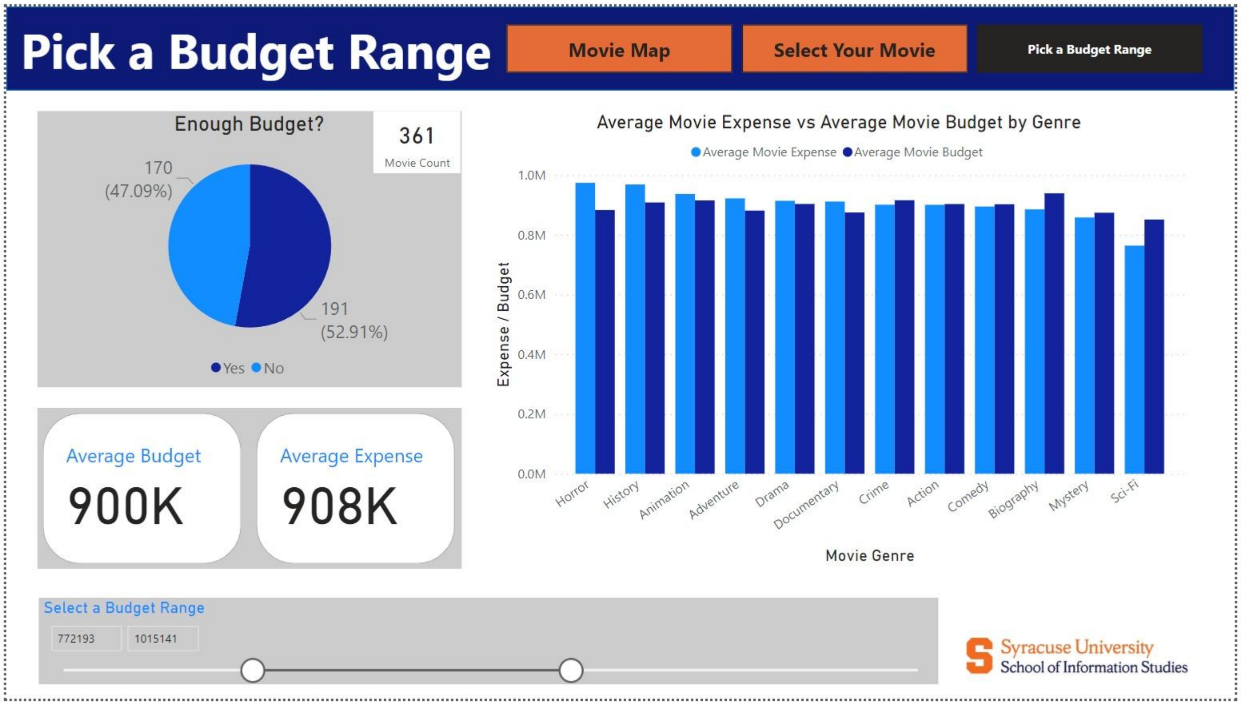
Movie Budget
1M

Movie Expense
976K

Director Name
Samuel Bayer

Producer Name
Miranda Colon

Figure 5 *Budget Range of Movies*



Conclusion and Reflection

Our project showcases a practical application of data analysis in the film industry, aiming to assist producers in making informed decisions during the pre-screening process. By leveraging SQL databases and Power BI for visualization, we established a system to explore and understand various movie-related aspects.

During the project, we used a variety of skills regarding databases, learning how to create tables, insert data into them, and filter and analyze data. We created a proper SQL database with multiple tables that established links between entities. Additionally, we learned how to visualize data using Power BI to present our data results more intuitively.

In trying to achieve the goals of the project, we encountered some difficulties. Initially, we hoped to be able to use real film production data found on the internet for our analysis, but none of the data was complete or exactly what we needed given the confidentiality issues, so in the end we considered using some of the information used to create the tables, but populated them with randomly generated data.

Reference

American Film Institute.
https://www.afi.com/

Chistyakova, T. B. (2019). *Big Data Analysis in Film Production.*
https://link.springer.com/chapter/10.1007/978-3-030-32579-4_18

Clevé, B. (2006). *Film Production Management.*
https://books.google.com/books?hl=en&lr=&id=0-Yp6udpdbgC&oi=fnd&pg=PP2&dq=film+production+cost+management&ots=6341yBopk2&sig=vGkvJfeMO73iSicQq5iVCxfZVYQ#v=onepage&q&f=false

Internet Movie Database.
https://www.imdb.com/

Yang, S. (2023). *Analysis of top box office film poster marketing scheme based on data mining and deep learning in the context of film marketing.* PLOS ONE 18(1): e0280848.
https://doi.org/10.1371/journal.pone.0280848

Appendix A

Team Log Recording

Team Members	Tasks
Zimin Zhu (team manager)	● Create Project guidelines (week 1) ● Set E-R Requirement (week 2) ● Grab and downloaded the raw data sets (week 3-4) ● Developed queries for answering business problems (week 7-8) ● Created and implemented trigger (week 9)
Aditya Sanjay Pawar	● Created project guidelines and objectives (week 1) ● Wroted create/drop table queries (week 5) ● Conducted logical data modeling (week 6) ● Conducted conceptual data modeling (week 6) ● Developed queries for answering business problems (week7)
Yixing Zhu	● Create project guidelines and objectives (week 1) ● Adjusted the conceptual model entity relationships (week7) ● Create random crew information (week 7) ● Conducted insert values (week 8) ● Create user interface (week 9)
Tian Tian	● Create project guidelines and objectives (week1) ● Conducted raw data set cleaning (week 3-4) ● Develop queries for answering business problems (week7) ● Summarized group highlights and wrote the project report (week9) ● Created and implemented trigger (week 9)

Timeline

Week 1 (10/9/2023-10/15/2023)

- Assigned roles and responsibilities for each team member
- Set up a group chat to communicate and share updates

Week 2 (10/16/2023-10/22/2023)

- Discussed about the idea
- Wrote the project proposal

Week 3 (10/23/2023-11/29/2023)

- Identifying the data needs for the DBMS designs
- Brainstormed ideas for the database structure and user interface

Week 4 (10/30/2023-11/5/2023)

- Set up a basic layout of the model
- Came up with a potential table and attributes for the management system

Week 5 (11/6/2023-11/12/2023)

- Build relationship between the possible entities
- Finalized the database structure

Week 6 (11/13/2023-11/19/2023)

- Started building the database using SQL

Week 7 (11/20/2023-11/26/2023)

- Developed a conceptual data model
- Developed a logical data model

Week 8 (11/27/2023-12/3/2023)

- Building the database and implemented procedures and triggers
- Designed the user interface for the management system

- Fix the relationship issues between tables

Week 9 (12/4/2023-12/10/2023)

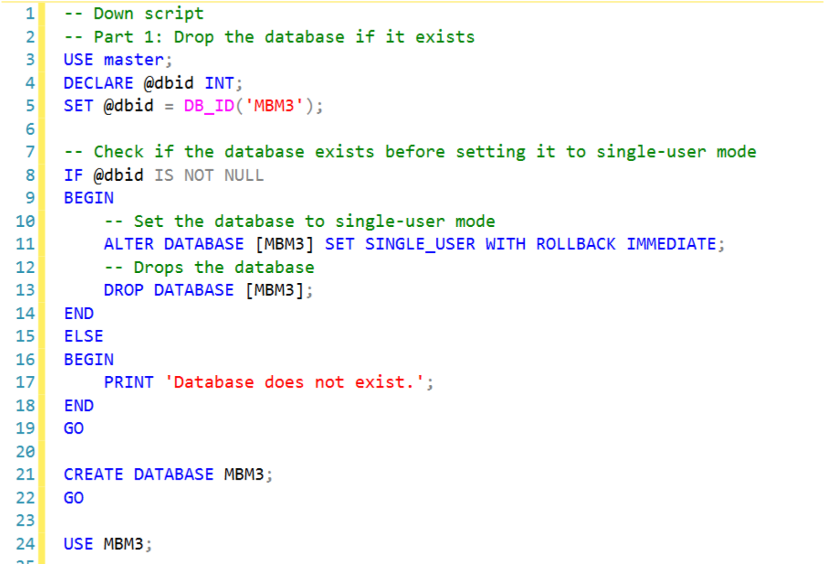
- Tested the database with data to ensure accuracy and functionality
- Made final revisions to the database and user interface
- Cross training

Week 10 (12/11/2023-12/12/2023)

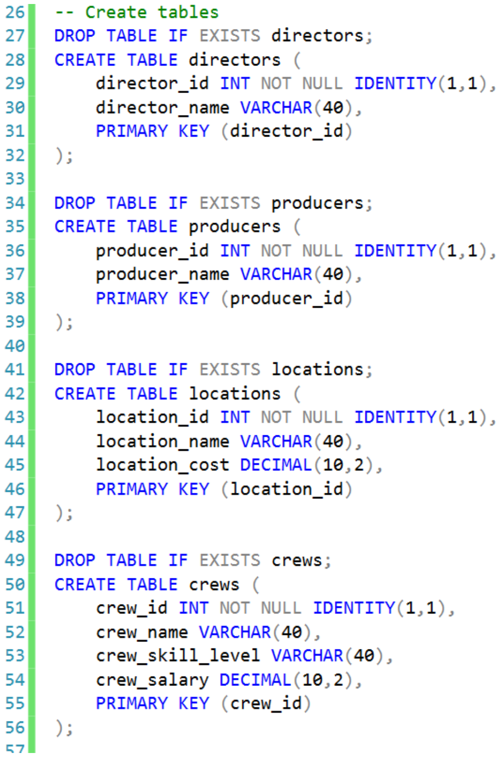
- Presented the film management system to the class
- Record the reflection and presentation video

Appendix B

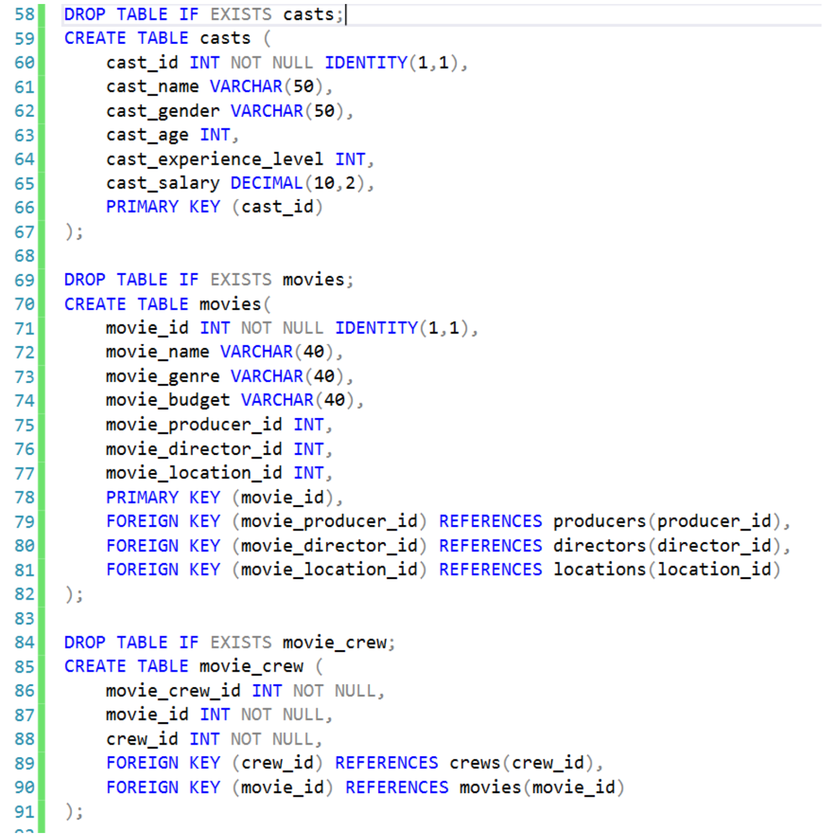
SQL Script



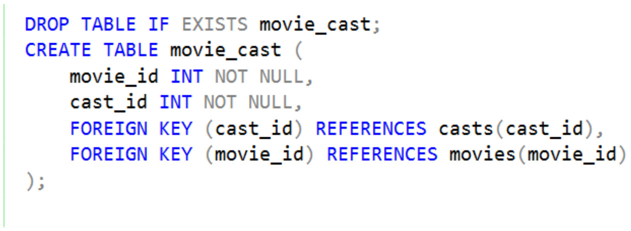
```sql
-- Down script
-- Part 1: Drop the database if it exists
USE master;
DECLARE @dbid INT;
SET @dbid = DB_ID('MBM3');

-- Check if the database exists before setting it to single-user mode
IF @dbid IS NOT NULL
BEGIN
    -- Set the database to single-user mode
    ALTER DATABASE [MBM3] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
    -- Drops the database
    DROP DATABASE [MBM3];
END
ELSE
BEGIN
    PRINT 'Database does not exist.';
END
GO

CREATE DATABASE MBM3;
GO

USE MBM3;
```



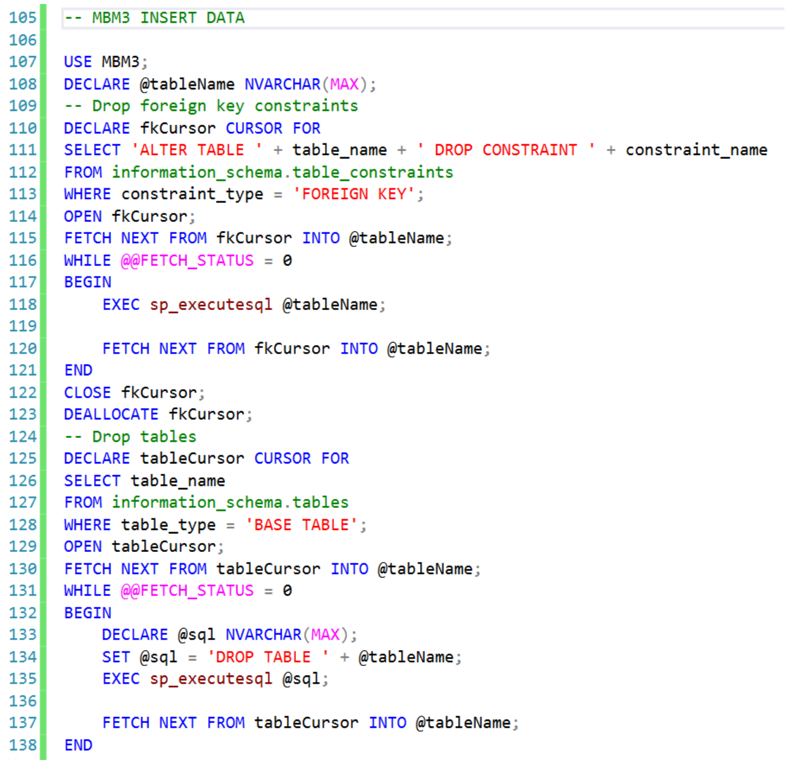
```sql
-- Create tables
DROP TABLE IF EXISTS directors;
CREATE TABLE directors (
    director_id INT NOT NULL IDENTITY(1,1),
    director_name VARCHAR(40),
    PRIMARY KEY (director_id)
);

DROP TABLE IF EXISTS producers;
CREATE TABLE producers (
    producer_id INT NOT NULL IDENTITY(1,1),
    producer_name VARCHAR(40),
    PRIMARY KEY (producer_id)
);

DROP TABLE IF EXISTS locations;
CREATE TABLE locations (
    location_id INT NOT NULL IDENTITY(1,1),
    location_name VARCHAR(40),
    location_cost DECIMAL(10,2),
    PRIMARY KEY (location_id)
);

DROP TABLE IF EXISTS crews;
CREATE TABLE crews (
    crew_id INT NOT NULL IDENTITY(1,1),
    crew_name VARCHAR(40),
    crew_skill_level VARCHAR(40),
    crew_salary DECIMAL(10,2),
    PRIMARY KEY (crew_id)
);
```



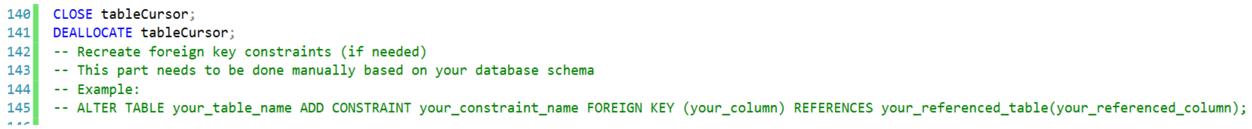
```sql
DROP TABLE IF EXISTS casts;
CREATE TABLE casts (
    cast_id INT NOT NULL IDENTITY(1,1),
    cast_name VARCHAR(50),
    cast_gender VARCHAR(50),
    cast_age INT,
    cast_experience_level INT,
    cast_salary DECIMAL(10,2),
    PRIMARY KEY (cast_id)
);

DROP TABLE IF EXISTS movies;
CREATE TABLE movies(
    movie_id INT NOT NULL IDENTITY(1,1),
    movie_name VARCHAR(40),
    movie_genre VARCHAR(40),
    movie_budget VARCHAR(40),
    movie_producer_id INT,
    movie_director_id INT,
    movie_location_id INT,
    PRIMARY KEY (movie_id),
    FOREIGN KEY (movie_producer_id) REFERENCES producers(producer_id),
    FOREIGN KEY (movie_director_id) REFERENCES directors(director_id),
    FOREIGN KEY (movie_location_id) REFERENCES locations(location_id)
);

DROP TABLE IF EXISTS movie_crew;
CREATE TABLE movie_crew (
    movie_crew_id INT NOT NULL,
    movie_id INT NOT NULL,
    crew_id INT NOT NULL,
    FOREIGN KEY (crew_id) REFERENCES crews(crew_id),
    FOREIGN KEY (movie_id) REFERENCES movies(movie_id)
);
```



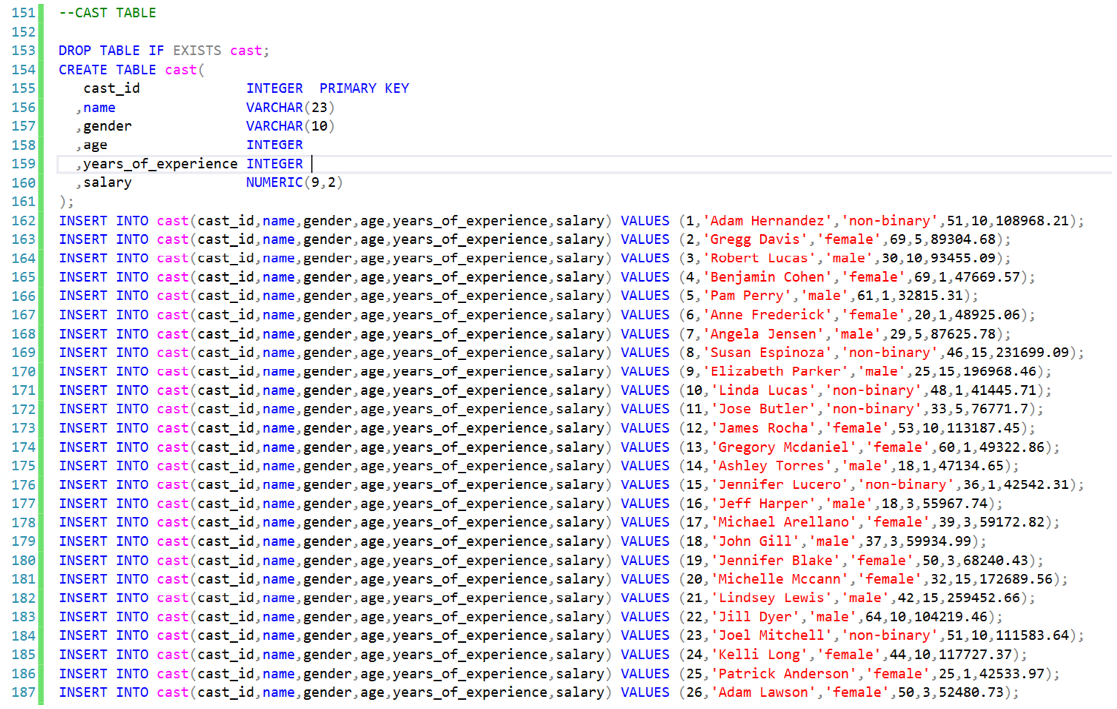
```sql
DROP TABLE IF EXISTS movie_cast;
CREATE TABLE movie_cast (
    movie_id INT NOT NULL,
    cast_id INT NOT NULL,
    FOREIGN KEY (cast_id) REFERENCES casts(cast_id),
    FOREIGN KEY (movie_id) REFERENCES movies(movie_id)
);
```



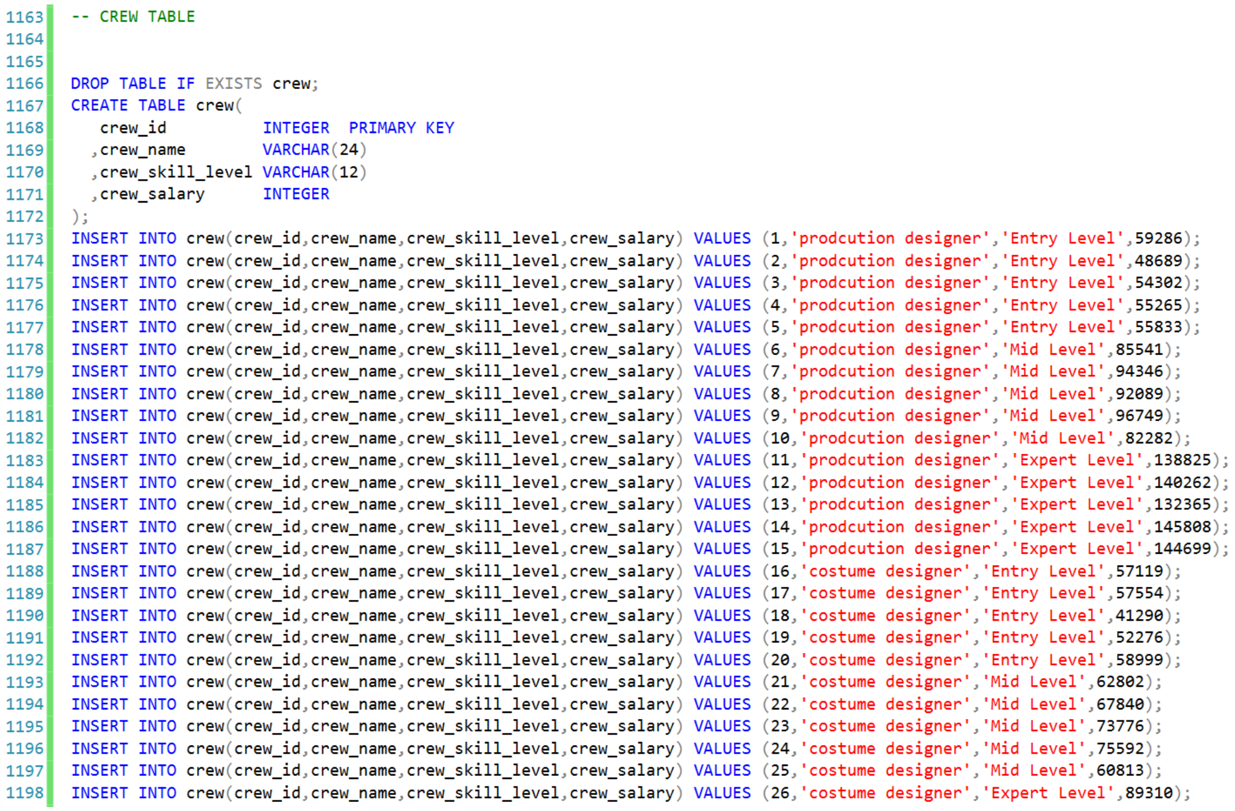
```sql
-- MBM3 INSERT DATA

USE MBM3;
DECLARE @tableName NVARCHAR(MAX);
-- Drop foreign key constraints
DECLARE fkCursor CURSOR FOR
SELECT 'ALTER TABLE ' + table_name + ' DROP CONSTRAINT ' + constraint_name
FROM information_schema.table_constraints
WHERE constraint_type = 'FOREIGN KEY';
OPEN fkCursor;
FETCH NEXT FROM fkCursor INTO @tableName;
WHILE @@FETCH_STATUS = 0
BEGIN
    EXEC sp_executesql @tableName;

    FETCH NEXT FROM fkCursor INTO @tableName;
END
CLOSE fkCursor;
DEALLOCATE fkCursor;
-- Drop tables
DECLARE tableCursor CURSOR FOR
SELECT table_name
FROM information_schema.tables
WHERE table_type = 'BASE TABLE';
OPEN tableCursor;
FETCH NEXT FROM tableCursor INTO @tableName;
WHILE @@FETCH_STATUS = 0
BEGIN
    DECLARE @sql NVARCHAR(MAX);
    SET @sql = 'DROP TABLE ' + @tableName;
    EXEC sp_executesql @sql;

    FETCH NEXT FROM tableCursor INTO @tableName;
END
```



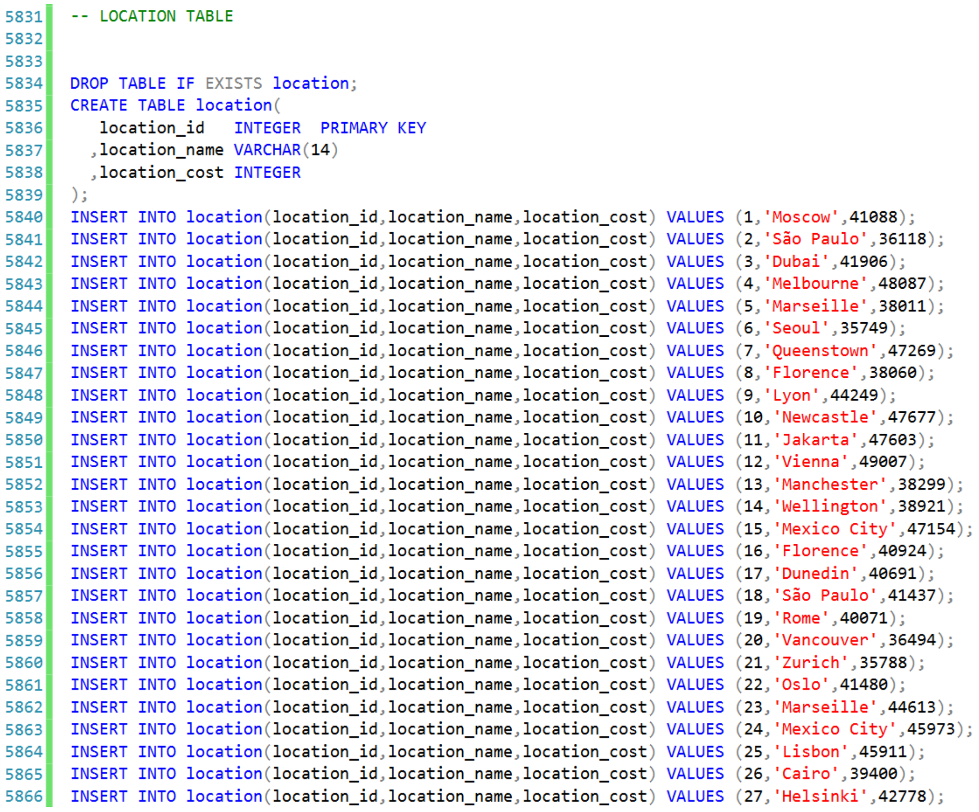
```sql
CLOSE tableCursor;
DEALLOCATE tableCursor;
-- Recreate foreign key constraints (if needed)
-- This part needs to be done manually based on your database schema
-- Example:
-- ALTER TABLE your_table_name ADD CONSTRAINT your_constraint_name FOREIGN KEY (your_column) REFERENCES your_referenced_table(your_referenced_column);
```



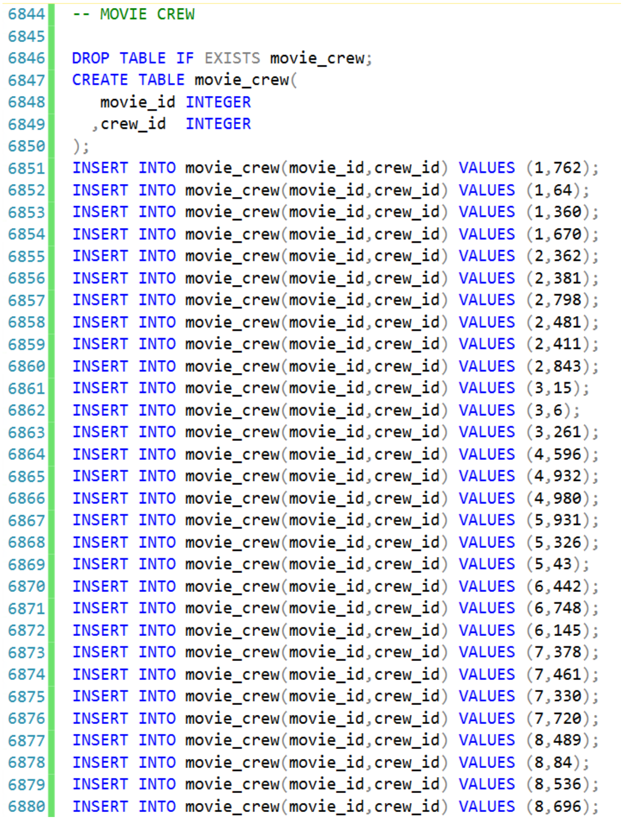
```sql
--CAST TABLE

DROP TABLE IF EXISTS cast;
CREATE TABLE cast(
   cast_id              INTEGER  PRIMARY KEY
  ,name                 VARCHAR(23)
  ,gender               VARCHAR(10)
  ,age                  INTEGER
  ,years_of_experience INTEGER
  ,salary               NUMERIC(9,2)
);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (1,'Adam Hernandez','non-binary',51,10,108968.21);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (2,'Gregg Davis','female',69,5,89304.68);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (3,'Robert Lucas','male',30,10,93455.09);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (4,'Benjamin Cohen','female',69,1,47669.57);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (5,'Pam Perry','male',61,1,32815.31);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (6,'Anne Frederick','female',20,1,48925.06);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (7,'Angela Jensen','male',29,5,87625.78);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (8,'Susan Espinoza','non-binary',46,15,231699.09);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (9,'Elizabeth Parker','male',25,15,196968.46);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (10,'Linda Lucas','non-binary',48,1,41445.71);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (11,'Jose Butler','non-binary',33,5,76771.7);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (12,'James Rocha','female',53,10,113187.45);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (13,'Gregory Mcdaniel','female',60,1,49322.86);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (14,'Ashley Torres','male',18,1,47134.65);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (15,'Jennifer Lucero','non-binary',36,1,42542.31);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (16,'Jeff Harper','male',18,3,55967.74);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (17,'Michael Arellano','female',39,3,59172.82);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (18,'John Gill','male',37,3,59934.99);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (19,'Jennifer Blake','female',50,3,68240.43);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (20,'Michelle Mccann','female',32,15,172689.56);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (21,'Lindsey Lewis','male',42,15,259452.66);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (22,'Jill Dyer','male',64,10,104219.46);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (23,'Joel Mitchell','non-binary',51,10,111583.64);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (24,'Kelli Long','female',44,10,117727.37);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (25,'Patrick Anderson','female',25,1,42533.97);
INSERT INTO cast(cast_id,name,gender,age,years_of_experience,salary) VALUES (26,'Adam Lawson','female',50,3,52480.73);
```



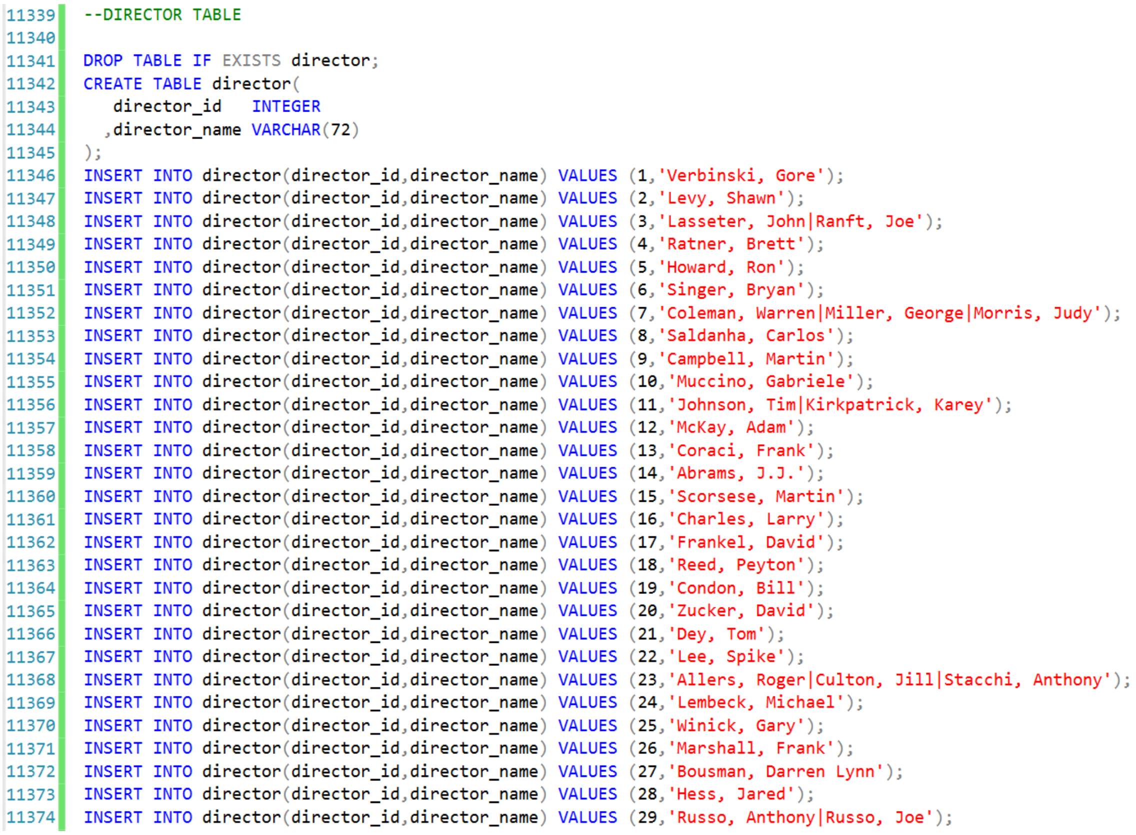
```sql
-- CREW TABLE


DROP TABLE IF EXISTS crew;
CREATE TABLE crew(
   crew_id          INTEGER  PRIMARY KEY
  ,crew_name        VARCHAR(24)
  ,crew_skill_level VARCHAR(12)
  ,crew_salary      INTEGER
);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (1,'prodcution designer','Entry Level',59286);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (2,'prodcution designer','Entry Level',48689);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (3,'prodcution designer','Entry Level',54302);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (4,'prodcution designer','Entry Level',55265);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (5,'prodcution designer','Entry Level',55833);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (6,'prodcution designer','Mid Level',85541);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (7,'prodcution designer','Mid Level',94346);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (8,'prodcution designer','Mid Level',92089);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (9,'prodcution designer','Mid Level',96749);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (10,'prodcution designer','Mid Level',82282);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (11,'prodcution designer','Expert Level',138825);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (12,'prodcution designer','Expert Level',140262);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (13,'prodcution designer','Expert Level',132365);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (14,'prodcution designer','Expert Level',145808);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (15,'prodcution designer','Expert Level',144699);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (16,'costume designer','Entry Level',57119);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (17,'costume designer','Entry Level',57554);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (18,'costume designer','Entry Level',41290);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (19,'costume designer','Entry Level',52276);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (20,'costume designer','Entry Level',58999);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (21,'costume designer','Mid Level',62802);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (22,'costume designer','Mid Level',67840);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (23,'costume designer','Mid Level',73776);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (24,'costume designer','Mid Level',75592);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (25,'costume designer','Mid Level',60813);
INSERT INTO crew(crew_id,crew_name,crew_skill_level,crew_salary) VALUES (26,'costume designer','Expert Level',89310);
```



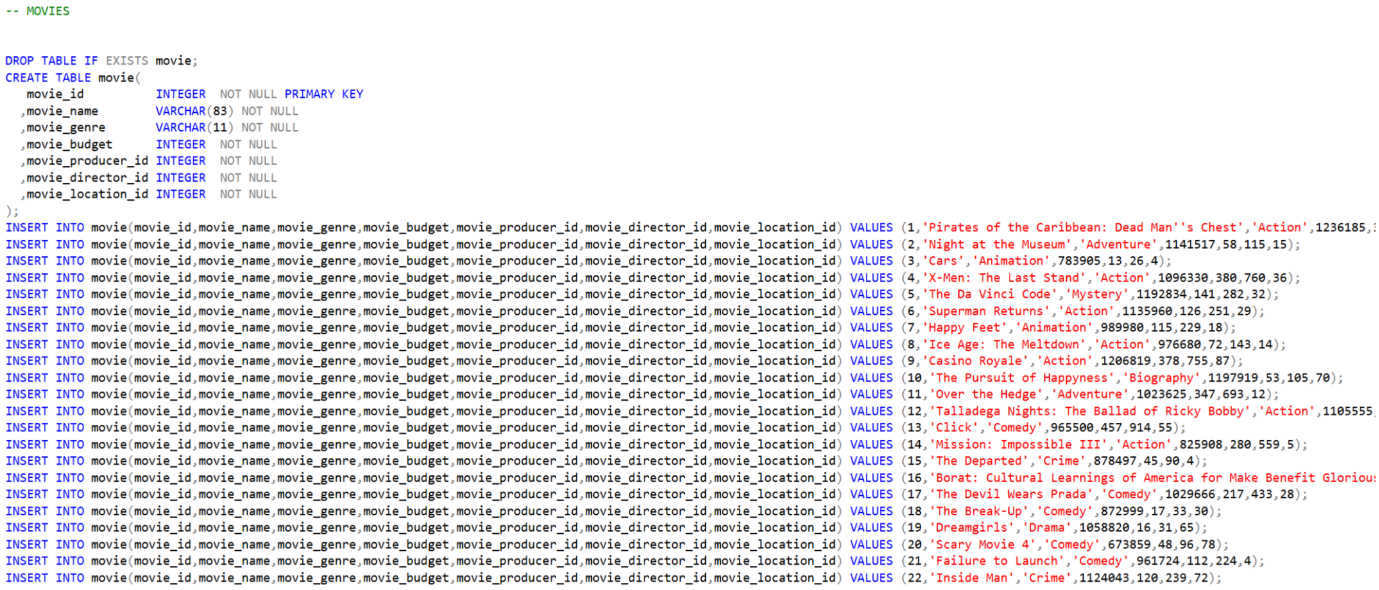
```sql
-- LOCATION TABLE


DROP TABLE IF EXISTS location;
CREATE TABLE location(
   location_id   INTEGER  PRIMARY KEY
  ,location_name VARCHAR(14)
  ,location_cost INTEGER
);
INSERT INTO location(location_id,location_name,location_cost) VALUES (1,'Moscow',41088);
INSERT INTO location(location_id,location_name,location_cost) VALUES (2,'São Paulo',36118);
INSERT INTO location(location_id,location_name,location_cost) VALUES (3,'Dubai',41906);
INSERT INTO location(location_id,location_name,location_cost) VALUES (4,'Melbourne',48087);
INSERT INTO location(location_id,location_name,location_cost) VALUES (5,'Marseille',38011);
INSERT INTO location(location_id,location_name,location_cost) VALUES (6,'Seoul',35749);
INSERT INTO location(location_id,location_name,location_cost) VALUES (7,'Queenstown',47269);
INSERT INTO location(location_id,location_name,location_cost) VALUES (8,'Florence',38060);
INSERT INTO location(location_id,location_name,location_cost) VALUES (9,'Lyon',44249);
INSERT INTO location(location_id,location_name,location_cost) VALUES (10,'Newcastle',47677);
INSERT INTO location(location_id,location_name,location_cost) VALUES (11,'Jakarta',47603);
INSERT INTO location(location_id,location_name,location_cost) VALUES (12,'Vienna',49007);
INSERT INTO location(location_id,location_name,location_cost) VALUES (13,'Manchester',38299);
INSERT INTO location(location_id,location_name,location_cost) VALUES (14,'Wellington',38921);
INSERT INTO location(location_id,location_name,location_cost) VALUES (15,'Mexico City',47154);
INSERT INTO location(location_id,location_name,location_cost) VALUES (16,'Florence',40924);
INSERT INTO location(location_id,location_name,location_cost) VALUES (17,'Dunedin',40691);
INSERT INTO location(location_id,location_name,location_cost) VALUES (18,'São Paulo',41437);
INSERT INTO location(location_id,location_name,location_cost) VALUES (19,'Rome',40071);
INSERT INTO location(location_id,location_name,location_cost) VALUES (20,'Vancouver',36494);
INSERT INTO location(location_id,location_name,location_cost) VALUES (21,'Zurich',35788);
INSERT INTO location(location_id,location_name,location_cost) VALUES (22,'Oslo',41480);
INSERT INTO location(location_id,location_name,location_cost) VALUES (23,'Marseille',44613);
INSERT INTO location(location_id,location_name,location_cost) VALUES (24,'Mexico City',45973);
INSERT INTO location(location_id,location_name,location_cost) VALUES (25,'Lisbon',45911);
INSERT INTO location(location_id,location_name,location_cost) VALUES (26,'Cairo',39400);
INSERT INTO location(location_id,location_name,location_cost) VALUES (27,'Helsinki',42778);
```



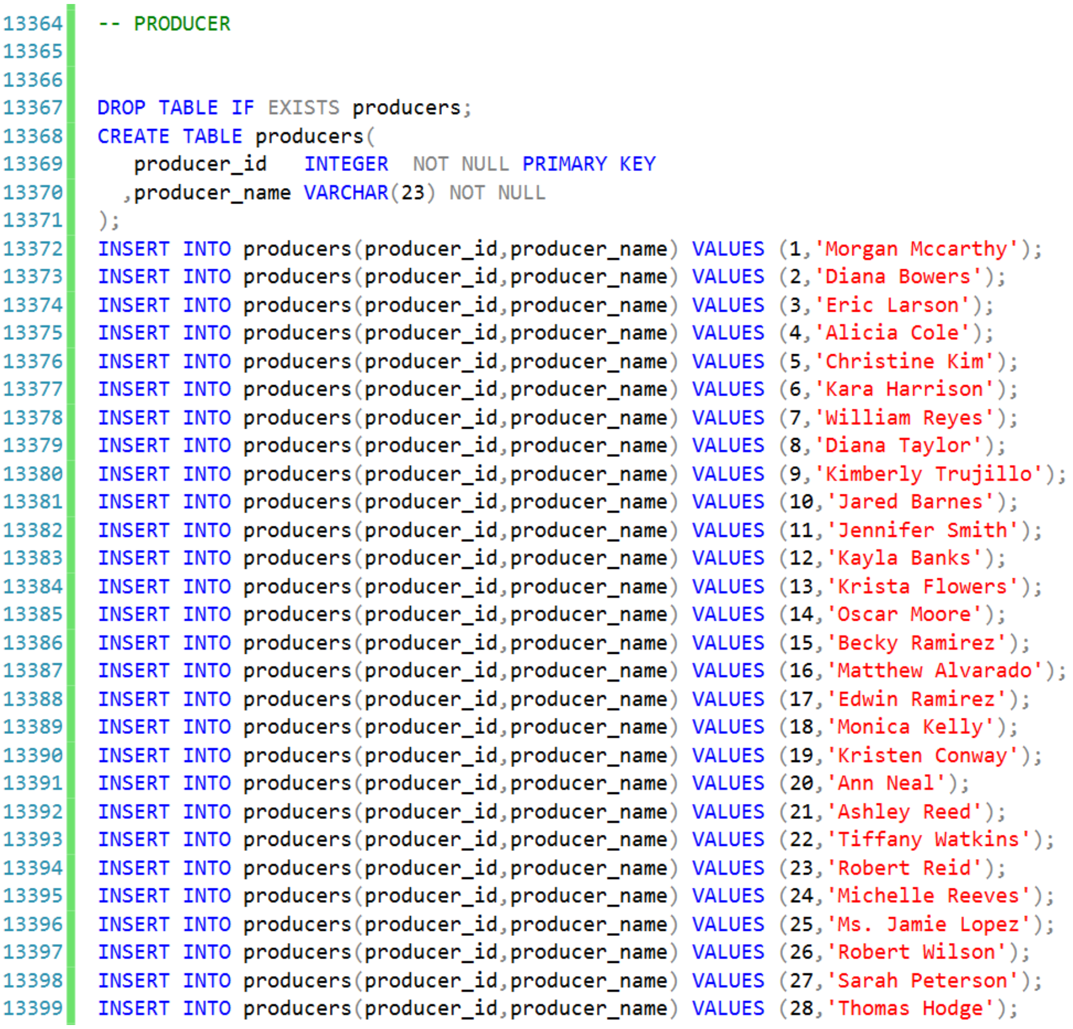
```sql
-- MOVIE CREW

DROP TABLE IF EXISTS movie_crew;
CREATE TABLE movie_crew(
   movie_id INTEGER
  ,crew_id  INTEGER
);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (1,762);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (1,64);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (1,360);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (1,670);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (2,362);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (2,381);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (2,798);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (2,481);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (2,411);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (2,843);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (3,15);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (3,6);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (3,261);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (4,596);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (4,932);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (4,980);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (5,931);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (5,326);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (5,43);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (6,442);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (6,748);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (6,145);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (7,378);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (7,461);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (7,330);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (7,720);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (8,489);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (8,84);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (8,536);
INSERT INTO movie_crew(movie_id,crew_id) VALUES (8,696);
```



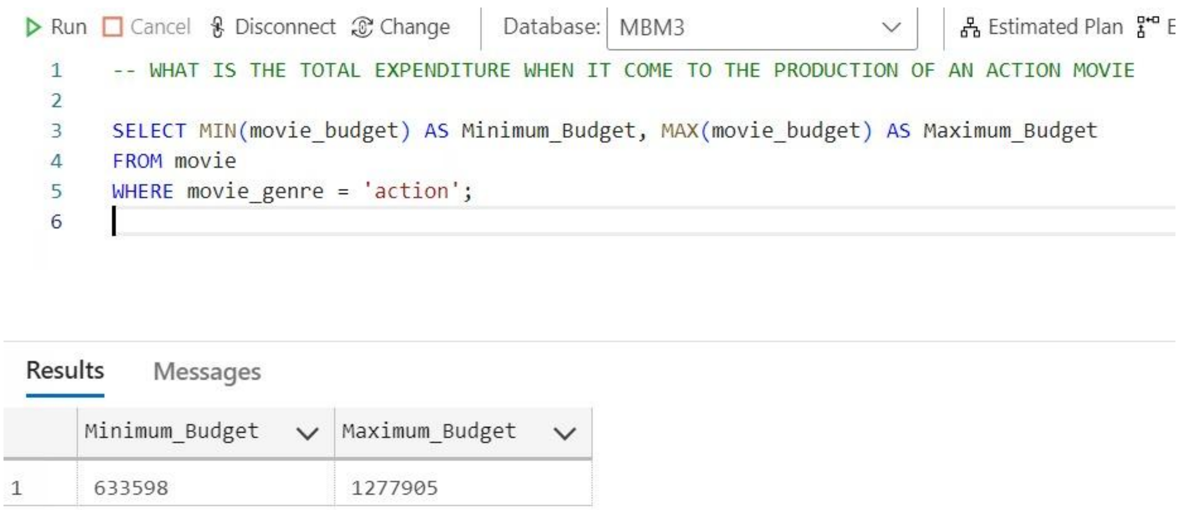
```sql
--DIRECTOR TABLE

DROP TABLE IF EXISTS director;
CREATE TABLE director(
   director_id   INTEGER
  ,director_name VARCHAR(72)
);
INSERT INTO director(director_id,director_name) VALUES (1,'Verbinski, Gore');
INSERT INTO director(director_id,director_name) VALUES (2,'Levy, Shawn');
INSERT INTO director(director_id,director_name) VALUES (3,'Lasseter, John|Ranft, Joe');
INSERT INTO director(director_id,director_name) VALUES (4,'Ratner, Brett');
INSERT INTO director(director_id,director_name) VALUES (5,'Howard, Ron');
INSERT INTO director(director_id,director_name) VALUES (6,'Singer, Bryan');
INSERT INTO director(director_id,director_name) VALUES (7,'Coleman, Warren|Miller, George|Morris, Judy');
INSERT INTO director(director_id,director_name) VALUES (8,'Saldanha, Carlos');
INSERT INTO director(director_id,director_name) VALUES (9,'Campbell, Martin');
INSERT INTO director(director_id,director_name) VALUES (10,'Muccino, Gabriele');
INSERT INTO director(director_id,director_name) VALUES (11,'Johnson, Tim|Kirkpatrick, Karey');
INSERT INTO director(director_id,director_name) VALUES (12,'McKay, Adam');
INSERT INTO director(director_id,director_name) VALUES (13,'Coraci, Frank');
INSERT INTO director(director_id,director_name) VALUES (14,'Abrams, J.J.');
INSERT INTO director(director_id,director_name) VALUES (15,'Scorsese, Martin');
INSERT INTO director(director_id,director_name) VALUES (16,'Charles, Larry');
INSERT INTO director(director_id,director_name) VALUES (17,'Frankel, David');
INSERT INTO director(director_id,director_name) VALUES (18,'Reed, Peyton');
INSERT INTO director(director_id,director_name) VALUES (19,'Condon, Bill');
INSERT INTO director(director_id,director_name) VALUES (20,'Zucker, David');
INSERT INTO director(director_id,director_name) VALUES (21,'Dey, Tom');
INSERT INTO director(director_id,director_name) VALUES (22,'Lee, Spike');
INSERT INTO director(director_id,director_name) VALUES (23,'Allers, Roger|Culton, Jill|Stacchi, Anthony');
INSERT INTO director(director_id,director_name) VALUES (24,'Lembeck, Michael');
INSERT INTO director(director_id,director_name) VALUES (25,'Winick, Gary');
INSERT INTO director(director_id,director_name) VALUES (26,'Marshall, Frank');
INSERT INTO director(director_id,director_name) VALUES (27,'Bousman, Darren Lynn');
INSERT INTO director(director_id,director_name) VALUES (28,'Hess, Jared');
INSERT INTO director(director_id,director_name) VALUES (29,'Russo, Anthony|Russo, Joe');
```



```sql
-- MOVIES


DROP TABLE IF EXISTS movie;
CREATE TABLE movie(
   movie_id          INTEGER  NOT NULL PRIMARY KEY
  ,movie_name        VARCHAR(83) NOT NULL
  ,movie_genre       VARCHAR(11) NOT NULL
  ,movie_budget      INTEGER  NOT NULL
  ,movie_producer_id INTEGER  NOT NULL
  ,movie_director_id INTEGER  NOT NULL
  ,movie_location_id INTEGER  NOT NULL
);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (1,'Pirates of the Caribbean: Dead Man''s Chest','Action',1236185,
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (2,'Night at the Museum','Adventure',1141517,58,115,15);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (3,'Cars','Animation',783905,13,26,4);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (4,'X-Men: The Last Stand','Action',1096330,380,760,36);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (5,'The Da Vinci Code','Mystery',1192834,141,282,32);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (6,'Superman Returns','Action',1135960,126,251,29);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (7,'Happy Feet','Animation',989980,115,229,18);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (8,'Ice Age: The Meltdown','Action',976680,72,143,14);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (9,'Casino Royale','Action',1206819,378,755,87);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (10,'The Pursuit of Happyness','Biography',1197919,53,105,70);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (11,'Over the Hedge','Adventure',1023625,347,693,12);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (12,'Talladega Nights: The Ballad of Ricky Bobby','Action',1105555
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (13,'Click','Comedy',965500,457,914,55);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (14,'Mission: Impossible III','Action',825908,280,559,5);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (15,'The Departed','Crime',878497,45,90,4);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (16,'Borat: Cultural Learnings of America for Make Benefit Gloriou
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (17,'The Devil Wears Prada','Comedy',1029666,217,433,28);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (18,'The Break-Up','Comedy',872999,17,33,30);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (19,'Dreamgirls','Drama',1058820,16,31,65);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (20,'Scary Movie 4','Comedy',673859,48,96,78);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (21,'Failure to Launch','Comedy',961724,112,224,4);
INSERT INTO movie(movie_id,movie_name,movie_genre,movie_budget,movie_producer_id,movie_director_id,movie_location_id) VALUES (22,'Inside Man','Crime',1124043,120,239,72);
```

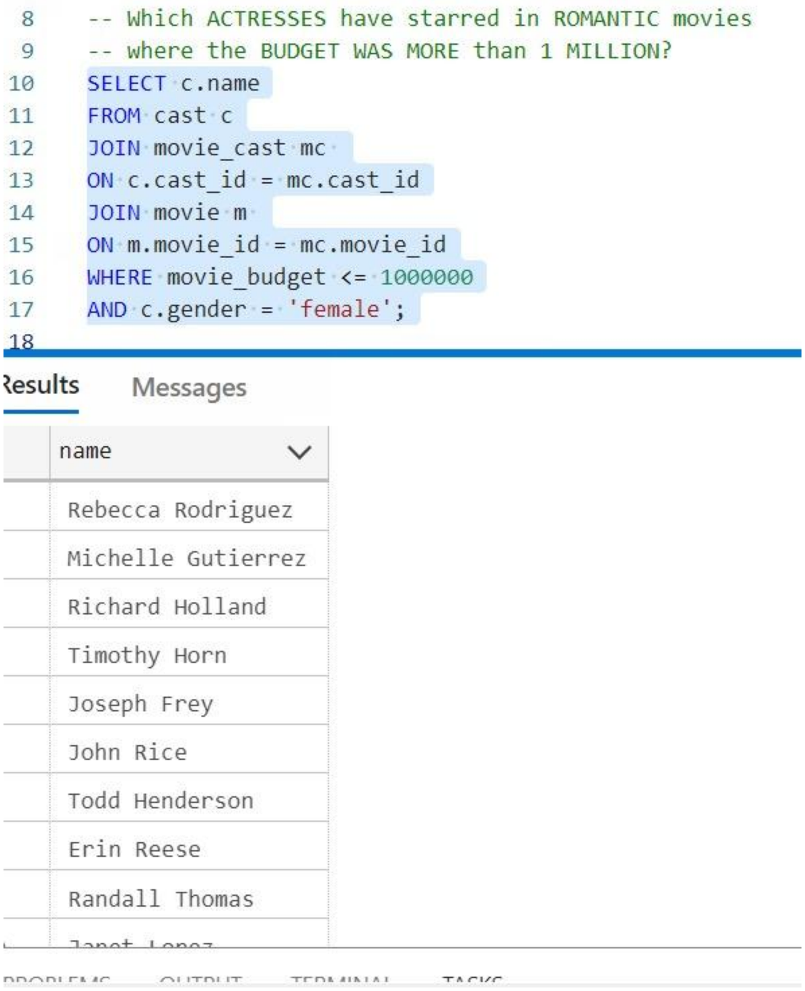


```sql
-- PRODUCER


DROP TABLE IF EXISTS producers;
CREATE TABLE producers(
   producer_id   INTEGER  NOT NULL PRIMARY KEY
  ,producer_name VARCHAR(23) NOT NULL
);
INSERT INTO producers(producer_id,producer_name) VALUES (1,'Morgan Mccarthy');
INSERT INTO producers(producer_id,producer_name) VALUES (2,'Diana Bowers');
INSERT INTO producers(producer_id,producer_name) VALUES (3,'Eric Larson');
INSERT INTO producers(producer_id,producer_name) VALUES (4,'Alicia Cole');
INSERT INTO producers(producer_id,producer_name) VALUES (5,'Christine Kim');
INSERT INTO producers(producer_id,producer_name) VALUES (6,'Kara Harrison');
INSERT INTO producers(producer_id,producer_name) VALUES (7,'William Reyes');
INSERT INTO producers(producer_id,producer_name) VALUES (8,'Diana Taylor');
INSERT INTO producers(producer_id,producer_name) VALUES (9,'Kimberly Trujillo');
INSERT INTO producers(producer_id,producer_name) VALUES (10,'Jared Barnes');
INSERT INTO producers(producer_id,producer_name) VALUES (11,'Jennifer Smith');
INSERT INTO producers(producer_id,producer_name) VALUES (12,'Kayla Banks');
INSERT INTO producers(producer_id,producer_name) VALUES (13,'Krista Flowers');
INSERT INTO producers(producer_id,producer_name) VALUES (14,'Oscar Moore');
INSERT INTO producers(producer_id,producer_name) VALUES (15,'Becky Ramirez');
INSERT INTO producers(producer_id,producer_name) VALUES (16,'Matthew Alvarado');
INSERT INTO producers(producer_id,producer_name) VALUES (17,'Edwin Ramirez');
INSERT INTO producers(producer_id,producer_name) VALUES (18,'Monica Kelly');
INSERT INTO producers(producer_id,producer_name) VALUES (19,'Kristen Conway');
INSERT INTO producers(producer_id,producer_name) VALUES (20,'Ann Neal');
INSERT INTO producers(producer_id,producer_name) VALUES (21,'Ashley Reed');
INSERT INTO producers(producer_id,producer_name) VALUES (22,'Tiffany Watkins');
INSERT INTO producers(producer_id,producer_name) VALUES (23,'Robert Reid');
INSERT INTO producers(producer_id,producer_name) VALUES (24,'Michelle Reeves');
INSERT INTO producers(producer_id,producer_name) VALUES (25,'Ms. Jamie Lopez');
INSERT INTO producers(producer_id,producer_name) VALUES (26,'Robert Wilson');
INSERT INTO producers(producer_id,producer_name) VALUES (27,'Sarah Peterson');
INSERT INTO producers(producer_id,producer_name) VALUES (28,'Thomas Hodge');
```



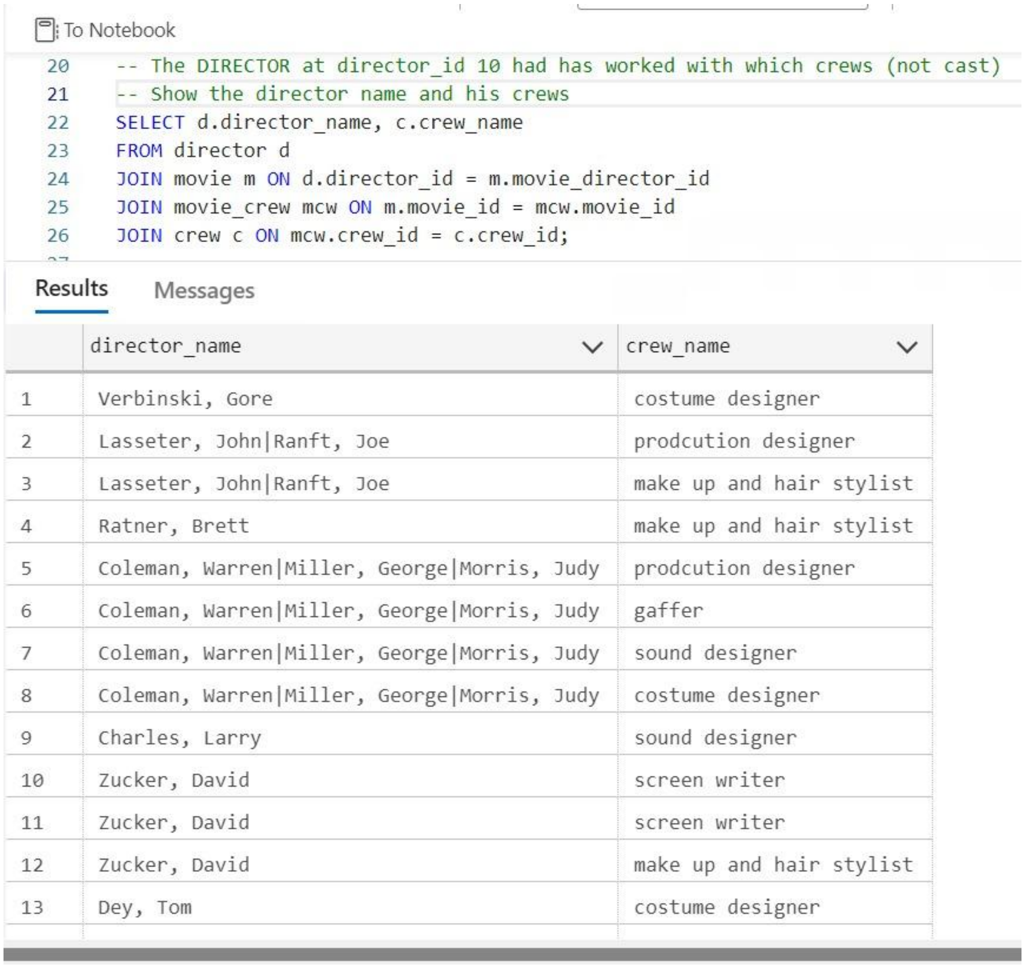
```sql
-- WHAT IS THE TOTAL EXPENDITURE WHEN IT COME TO THE PRODUCTION OF AN ACTION MOVIE

SELECT MIN(movie_budget) AS Minimum_Budget, MAX(movie_budget) AS Maximum_Budget
FROM movie
WHERE movie_genre = 'action';
```

	Minimum_Budget	Maximum_Budget
1	633598	1277905



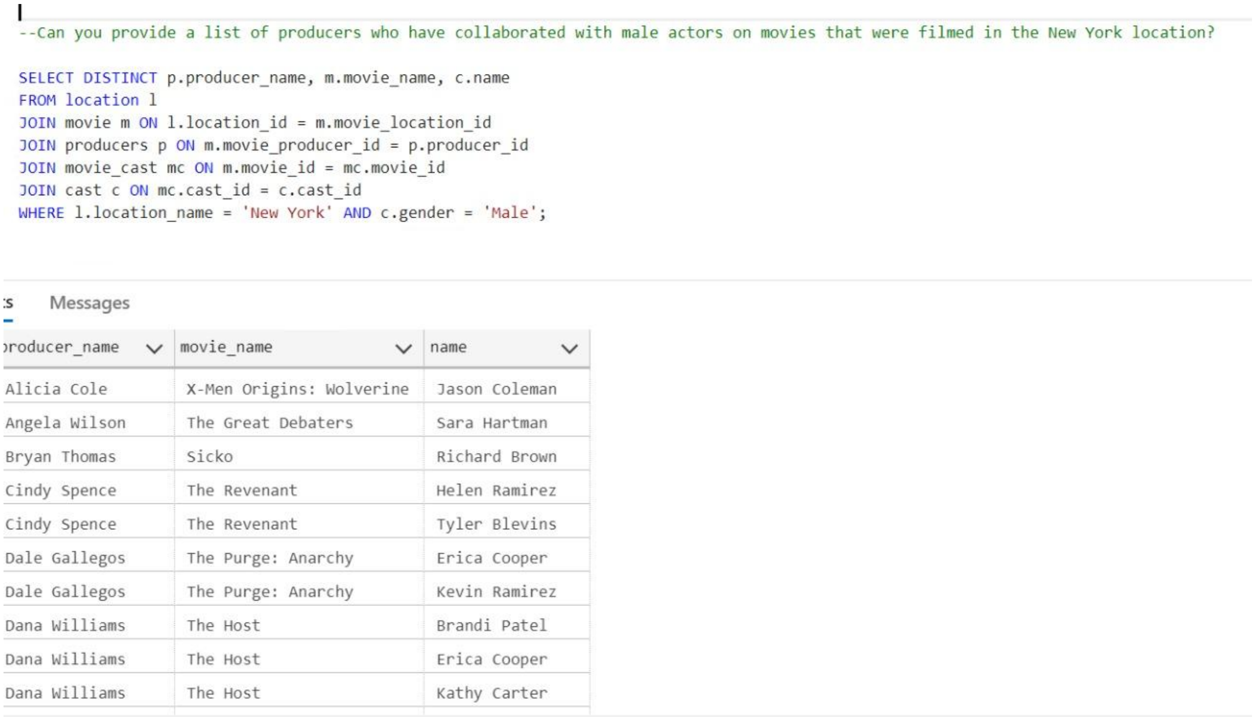
```sql
-- Which ACTRESSES have starred in ROMANTIC movies
-- where the BUDGET WAS MORE than 1 MILLION?
SELECT c.name
FROM cast c
JOIN movie_cast mc
ON c.cast_id = mc.cast_id
JOIN movie m
ON m.movie_id = mc.movie_id
WHERE movie_budget <= 1000000
AND c.gender = 'female';
```

name
Rebecca Rodriguez
Michelle Gutierrez
Richard Holland
Timothy Horn
Joseph Frey
John Rice
Todd Henderson
Erin Reese
Randall Thomas
Janet Lopez



```sql
-- The DIRECTOR at director_id 10 had has worked with which crews (not cast)
-- Show the director name and his crews
SELECT d.director_name, c.crew_name
FROM director d
JOIN movie m ON d.director_id = m.movie_director_id
JOIN movie_crew mcw ON m.movie_id = mcw.movie_id
JOIN crew c ON mcw.crew_id = c.crew_id;
```

	director_name	crew_name
1	Verbinski, Gore	costume designer
2	Lasseter, John|Ranft, Joe	prodcution designer
3	Lasseter, John|Ranft, Joe	make up and hair stylist
4	Ratner, Brett	make up and hair stylist
5	Coleman, Warren|Miller, George|Morris, Judy	prodcution designer
6	Coleman, Warren|Miller, George|Morris, Judy	gaffer
7	Coleman, Warren|Miller, George|Morris, Judy	sound designer
8	Coleman, Warren|Miller, George|Morris, Judy	costume designer
9	Charles, Larry	sound designer
10	Zucker, David	screen writer
11	Zucker, David	screen writer
12	Zucker, David	make up and hair stylist
13	Dey, Tom	costume designer



```sql
--Can you provide a list of producers who have collaborated with male actors on movies that were filmed in the New York location?

SELECT DISTINCT p.producer_name, m.movie_name, c.name
FROM location l
JOIN movie m ON l.location_id = m.movie_location_id
JOIN producers p ON m.movie_producer_id = p.producer_id
JOIN movie_cast mc ON m.movie_id = mc.movie_id
JOIN cast c ON mc.cast_id = c.cast_id
WHERE l.location_name = 'New York' AND c.gender = 'Male';
```

roducer_name	movie_name	name
Alicia Cole	X-Men Origins: Wolverine	Jason Coleman
Angela Wilson	The Great Debaters	Sara Hartman
Bryan Thomas	Sicko	Richard Brown
Cindy Spence	The Revenant	Helen Ramirez
Cindy Spence	The Revenant	Tyler Blevins
Dale Gallegos	The Purge: Anarchy	Erica Cooper
Dale Gallegos	The Purge: Anarchy	Kevin Ramirez
Dana Williams	The Host	Brandi Patel
Dana Williams	The Host	Erica Cooper
Dana Williams	The Host	Kathy Carter