Farees Siddiqui Project 2 (Lab 5-10)

100-780-513

Lab 05

Inserts.sql

```
CREATE EXTENSION IF NOT EXISTS "uuid-TABLEossp";

--

INSERT INTO "Ticket" VALUES --

    (uuid_generate_v4(), (SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Black Panther'), (SELECT "CinemaID" FROM "Cinema" WHERE "cinemaName" = 'Cineplex Cinemas Yonge-Dundas and VIP'), '2020-04-01', '02:55:00', '02:15:00', 'IMAX', 15.99),

    (uuid_generate_v4(), (SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Endgame'), (SELECT "CinemaID" FROM "Cinema" WHERE "cinemaName" = 'AMC Saratoga 14'), '2020-04-01', '02:55:00', '03:02:00', '3D', 13.99),

    (uuid_generate_v4(), (SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Infinity War'), (SELECT "CinemaID" FROM "Cinema" WHERE "cinemaName" = 'AMC Sunset Place 24'), '2020-04-01', '02:55:00', '02:30:00', 'Regular', 7.99),

    (uuid_generate_v4(), (SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Age of Ultron'), (SELECT "CinemaID" FROM "Cinema" WHERE "cinemaName" = 'Cineplex Cinemas Oshawa'), '2020-04-01', '02:55:00', '02:20:00', 'IMAX', 15.99),

    (uuid_generate_v4(), (SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'The Avengers'), (SELECT "CinemaID" FROM "Cinema" WHERE "cinemaName" = 'AMC Boston Common 19'), '2020-04-01', '02:55:00', '02:23:00', 'Regular', 7.99);



INSERT INTO "Awards" VALUES --

    (uuid_generate_v4(), 'Best Picture', 'best picture'),

    (uuid_generate_v4(), 'Best Director', 'best director'),
    (uuid_generate_v4(), 'Best Actor', 'best actor'),

    (uuid_generate_v4(), 'Best Actress', 'best actress'),

    (uuid_generate_v4(), 'Best Screenplay', 'best screenplay');



INSERT INTO "Festivals" VALUES --

    (uuid_generate_v4(), 'Toronto International Film Festival'),

    (uuid_generate_v4(), 'Sundance Film Festival'),

    (uuid_generate_v4(), 'Cannes Film Festival'),

    (uuid_generate_v4(), 'Venice Film Festival'),

    (uuid_generate_v4(), 'Berlin International Film Festival');



INSERT INTO "Movie" VALUES --

    (uuid_generate_v4(), (SELECT "festivalID" FROM "Festivals" WHERE "festivalName"
= 'Toronto International Film Festival'), 'Black Panther', '2018-02-16', 7,
200000000, 1347597973),

    (uuid_generate_v4(), (SELECT "festivalID" FROM "Festivals" WHERE "festivalName"
= 'Sundance Film Festival'), 'The Avengers', '2012-05-04', 8, 220000000,
1518815515),

    (uuid_generate_v4(), (SELECT "festivalID" FROM "Festivals" WHERE "festivalName"
= 'Cannes Film Festival'), 'Avengers: Age of Ultron', '2015-05-01', 7, 250000000,
1402809540),

    (uuid_generate_v4(), (SELECT "festivalID" FROM "Festivals" WHERE "festivalName"
= 'Toronto International Film Festival'), 'Avengers: Infinity War', '2018-04-27',
8, 321000000, 2048359754),

    (uuid_generate_v4(), (SELECT "festivalID" FROM "Festivals" WHERE "festivalName"
= 'Venice Film Festival'), 'Avengers: Endgame', '2019-04-26', 8, 356000000,
2797501328);
```

```
INSERT INTO "Genre" VALUES --

    ('Action', 'Movies in the action genre are fast-paced and include a lot of
action like fight scenes, chase scenes, and slow-motion shots. They can feature
superheroes, martial arts, or exciting stunts. These high-octane films are more
about the execution of the plot rather than the plot itself'),

    ('Sci-Fi', 'Science fiction is a genre of fiction in which the stories often
tell about science and technology of the future. It is important to note that
science fiction has a relationship with the principles of science—these stories
involve partially true- partially fictitious laws or theories of science'),

    ('Horror', 'Horror is a genre of literature, film, and television that is meant
to scare, startle, shock, and even repulse audiences. The key focus of a horror
novel, horror film, or horror TV show is to elicit a sense of dread in the reader
through frightening images, themes, and situations'),

    ('Thriller', 'Thriller is a genre of fiction, having numerous, often
overlapping subgenres. Thrillers are characterized and defined by the moods they
elicit, giving viewers heightened feelings of suspense, excitement, surprise,
anticipation and anxiety'),

    ('Romance', 'The romance genre is defined by two aspects that can be found in
every romance book or novel: a central love story between characters. an
emotionally satisfying, happy ending');



INSERT INTO "location" VALUES --

    (uuid_generate_v4(), 'Toronto', 'Ontario', 'Canada'),

    (uuid_generate_v4(), 'San Jose', 'California', 'USA'),

    (uuid_generate_v4(), 'Miami', 'Florida', 'USA'),

    (uuid_generate_v4(), 'Oshawa', 'Ontario', 'Canada'),

    (uuid_generate_v4(), 'Boston', 'Massachusetts', 'USA');



INSERT INTO "Director" VALUES --

    (uuid_generate_v4(), (SELECT "locationID" FROM "location" WHERE city =
'Toronto'), 'CooglerRyan', 1986),
    (uuid_generate_v4(), (SELECT "locationID" FROM "location" WHERE city = 'San
Jose'),'WhedonJoss', 1964),

    (uuid_generate_v4(), (SELECT "locationID" FROM "location" WHERE city =
'Boston'),'RussoAnthony', 1970),

    (uuid_generate_v4(), (SELECT "locationID" FROM "location" WHERE city =
'Oshawa'),'RussoJoe', 1971),



INSERT INTO "Actor" VALUES --

    (ARRAY[uuid_generate_v4()], (SELECT "locationID" FROM "location" WHERE 'city' =
'Toronto'), NULL, '1965-04-04', 'Downey-Robert', 'Dark Brown');

    (ARRAY[uuid_generate_v4()], (SELECT "locationID" FROM "location" WHERE 'city' =
'San Jose'), NULL, '1981-06-13', 'EvansChris', 'Green'),

    (ARRAY[uuid_generate_v4()], (SELECT "locationID" FROM "location" WHERE 'city' =
'Boston'), NULL, '1984-11-22', 'JohanssonScarlett', 'Green'),

    (ARRAY[uuid_generate_v4()], (SELECT "locationID" FROM "location" WHERE 'city' =
'Oshawa'), NULL, '1983-08-11', 'HemsworthChris', 'Blue'),

    (ARRAY[uuid_generate_v4()], (SELECT "locationID" FROM "location" WHERE 'city' =
'Miami'), NULL, '1967-11-22', 'RuffaloMark', 'Dark Brown');



INSERT INTO "University" VALUES --

    ('New York University', (SELECT "directorID" FROM "Director" WHERE
"Director.fName-lName" = "CooglerRyan"), 'Science', 28772, true, 'white-purple',
'New York University is a private research university in New York City'),

    ('Massachusetts Institute of Technology', (SELECT "directorID" FROM "Director"
WHERE "Director.fName-lName" = "WhedonJoss"), 'Mathematics', 11376, true, 'red-
black-silver-cardinal', 'The Massachusetts Institute of Technology is a private
land-grant research university in Cambridge, Massachusetts. Established in 1861,
MIT has since played a key role in the development of modern technology and
science, ranking among the top academic institutions in the world'),

    ('University of Toronto', (SELECT "directorID" FROM "Director" WHERE
"Director.fName-lName" = "RussoAnthony"), 'Arts and Science', 72389, false, 'white-
```

```
blue', 'The University of Toronto is a public research university in Toronto,
Ontario, Canada, located on the grounds that surround Queens Park. It was founded
by royal charter in 1827 as Kings College, the first institution of higher learning
in Upper Canada'),

    ('Queens University', (SELECT "directorID" FROM "Director" WHERE
"Director.fName-lName" = "RussoJoe"), 'Computing', 24143, false, 'blue-gold-red',
'Queens University at Kingston, commonly known as Queens University or simply
Queens, is a public research university in Kingston, Ontario, Canada. Queens holds
more than 1,400 hectares of land throughout Ontario and owns Herstmonceux Castle in
East Sussex, England'),

    ('Princeton University', (SELECT "directorID" FROM "Director" WHERE
"Director.fName-lName" = "BodenAnna"), 'Science', 8623, true, 'black-orange',
'Princeton University is a private Ivy League research university in Princeton, New
Jersey. Founded in 1746 in Elizabeth as the College of New Jersey, Princeton is the
fourth-oldest institution of higher education in the United States and one of the
nine colonial colleges chartered before the American Revolution.');



INSERT INTO "movie genre" VALUES --

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Black Panther'),
'Action'),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Endgame'),
'Sci-Fi'),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Infinity War'),
'Thriller'),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Age of
Ultron'), 'Sci-Fi'),



INSERT INTO "movieDirector" VALUES --

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Black Panther'), (SELECT
"directorID" FROM "Director" WHERE "fName-lName" = 'CooglerRyan')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Endgame'),
(SELECT "directorID" FROM "Director" WHERE "fName-lName" = 'RussoJoe')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Infinity War'),
(SELECT "directorID" FROM "Director" WHERE "fName-lName" = 'RussoAnthony')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Age of Ultron'), (SELECT "directorID" FROM "Director" WHERE "fName-lName" = 'WhedonJoss')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'The Avengers'), (SELECT "directorID" FROM "Director" WHERE "fName-lName" = 'WhedonJoss'));

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'The Avengers'), 'Action');



INSERT INTO "Cinema" VALUES --

    (uuid_generate_v4(), (SELECT "locationID" FROM "location" WHERE city = 'Toronto'), 'Cineplex Cinemas Yonge-Dundas and VIP'),

    (uuid_generate_v4(), (SELECT "locationID" FROM "location" WHERE city = 'San Jose'), 'AMC Saratoga 14'),

    (uuid_generate_v4(), (SELECT "locationID" FROM "location" WHERE city = 'Miami'), 'AMC Sunset Place 24'),

    (uuid_generate_v4(), (SELECT "locationID" FROM "location" WHERE city = 'Oshawa'), 'Cineplex Cinemas Oshawa'),

    (uuid_generate_v4(), (SELECT "locationID" FROM "location" WHERE city = 'Boston'), 'AMC Boston Common 19');



INSERT INTO "playedIn" VALUES --

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Black Panther'), (SELECT "CinemaID" FROM "Cinema" WHERE "cinemaName" = 'Cineplex Cinemas Yonge-Dundas and VIP')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Endgame'), (SELECT "CinemaID" FROM "Cinema" WHERE "cinemaName" = 'AMC Saratoga 14')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Infinity War'), (SELECT "CinemaID" FROM "Cinema" WHERE "cinemaName" = 'AMC Sunset Place 24')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Age of
Ultron'), (SELECT "CinemaID" FROM "Cinema" WHERE "cinemaName" = 'Cineplex Cinemas
Oshawa')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'The Avengers'), (SELECT
"CinemaID" FROM "Cinema" WHERE "cinemaName" = 'AMC Boston Common 19'));



INSERT INTO "movie awards" VALUES --

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Black Panther'), (SELECT
"AwardID" FROM "Awards" WHERE "awardName" = 'Best Picture')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Endgame'),
(SELECT "AwardID" FROM "Awards" WHERE "awardName" = 'Best Picture')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Infinity War'),
(SELECT "AwardID" FROM "Awards" WHERE "awardName" = 'Best Picture')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Age of
Ultron'), (SELECT "AwardID" FROM "Awards" WHERE "awardName" = 'Best Picture')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'The Avengers'), (SELECT
"AwardID" FROM "Awards" WHERE "awardName" = 'Best Picture'));



-- insert movie cast into movie cast table

INSERT INTO "movie cast" VALUES --

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Black Panther'), (SELECT
"actorID" FROM "Actor" WHERE "fName-lName" = 'Evans-Chris')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Endgame'),
(SELECT "actorID" FROM "Actor" WHERE "fName-lName" = 'Downey-Robert')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Infinity War'),
(SELECT "actorID" FROM "Actor" WHERE "fName-lName" = 'Hemsworth-Chris')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'Avengers: Age of
Ultron'), (SELECT "actorID" FROM "Actor" WHERE "fName-lName" = 'Johansson-
Scarlett')),

    ((SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'The Avengers'), (SELECT
"actorID" FROM "Actor" WHERE "fName-lName" = 'Ruffalo-Mark'));
```

Lab 06

SQL1.sql

```
-- Farees Siddiqui

-- 100 780 513
-- List all directors that were born in canada

SELECT "fName-lName" FROM "Director" WHERE "locationID" IN (SELECT "locationID"
FROM "location" WHERE "country" = 'Canada');

-- List all movies which are directed by 'David Lynch'

SELECT "movieTitle" FROM "Movie" WHERE "movieID" IN (SELECT "movieID" FROM
"movieDirector" WHERE "directorID" IN (SELECT "directorID" FROM "Director" WHERE
"fName-lName" = 'WhedonJoss'));

--List all the actors who played in the movies for which cost of production was
more than $1M.

SELECT "fName-lName" FROM "Actor" WHERE "actorID" IN (SELECT "actorID" FROM "movie
cast" WHERE "movieID" IN (SELECT "movieID" FROM "Movie" WHERE "budget" > 1000000));

-- List all the actors who played in the movies for which cost of production was
more than $1M in other currencies

SELECT "fName-lName", "budget",

"budget" * 1.36 AS "'CAD'",

"budget" * 147.46 AS "'JPY'",

"budget" * 61.53 AS "'RUB'",

"budget" * 1.0 AS "'EUR'",

"budget" * 1.0 AS "'CHF'"

FROM (("Actor"

INNER JOIN "movie cast" ON "Actor"."actorID" = "movie cast"."actorID")

INNER JOIN "Movie" on "Movie"."movieID" = "movie cast"."movieID")
```

```
WHERE "Movie"."budget" > 1000000;

-- Part D

SELECT "fName-lName" FROM "Director" WHERE "fName-lName" LIKE 'Ru%';

-- Part E

SELECT *

FROM "Movie"

WHERE "Movie"."movieID" IN (

    SELECT "movieID"

    FROM "movie genre"

    WHERE "genreName" =

        (SELECT "genreName" FROM "Genre" WHERE "Genre"."genreName" = 'Sci-Fi')

    AND "movieID" IN

        (SELECT "movieID"

         FROM "movie cast"

         WHERE "actorID" IN

            (SELECT "actorID"

             FROM "Actor"

             WHERE "Actor"."dateOfBirth" < DATE('1982-10-30')

            )

        )

);

-- Part F

SELECT "Actor1"."fName-lName", "Actor2"."fName-lName"
```

```
    FROM "Actor" as "Actor1", "Actor" as "Actor2"

        WHERE ("Actor1"."eyecolour" = 'Dark Brown'

        AND "Actor2"."eyecolour" = 'Dark Brown'

        AND "Actor1"."fName-lName" != "Actor2"."fName-lName"

        AND "Actor1"."fName-lName" < "Actor2"."fName-lName");
```

Lab 06 Results

Results.txt

```
Farees Siddiqui
100780513

Query:
SELECT "fName-lName" FROM "Director" WHERE "locationID" IN (SELECT "locationID"
FROM "location" WHERE "country" = 'Canada');

Result:
"CooglerRyan"
"RussoJoe"

Query:
SELECT "movieTitle" FROM "Movie" WHERE "movieID" IN (SELECT "movieID" FROM
"movieDirector" WHERE "directorID" IN (SELECT "directorID" FROM "Director" WHERE
"fName-lName" = 'WhedonJoss'));

Result:
"Avengers: Age of Ultron"
"The Avengers"

Query:
SELECT "fName-lName" FROM "Actor" WHERE "actorID" IN (SELECT "actorID" FROM "movie
cast" WHERE "movieID" IN (SELECT "movieID" FROM "Movie" WHERE "budget" > 1000000));

Result:
"Downey-Robert"
"Evans-Chris"
```

```
"Ruffalo-Mark"
"Hemsworth-Chris"
"Johansson-Scarlett"

Query:
SELECT "fName-lName", "budget",
"budget" * 1.36 AS "'CAD'",
"budget" * 147.46 AS "'JPY'",
"budget" * 61.53 AS "'RUB'",
"budget" * 1.0 AS "'EUR'",
"budget" * 1.0 AS "'CHF'"
FROM (("Actor"
INNER JOIN "movie cast" ON "Actor"."actorID" = "movie cast"."actorID")
INNER JOIN "Movie" on "Movie"."movieID" = "movie cast"."movieID")
WHERE "Movie"."budget" > 1000000;

Result:
"Evans-Chris"           200000000       272000000.00    29492000000.00
12306000000.00  200000000.0     200000000.0
"Downey-Robert"         356000000       484160000.00    52495760000.00
21904680000.00  356000000.0     356000000.0
"Hemsworth-Chris"           321000000   436560000.00    47334660000.00
19751130000.00  321000000.0     321000000.0
"Johansson-Scarlett"    250000000       340000000.00    36865000000.00
15382500000.00  250000000.0     250000000.0
"Ruffalo-Mark"          220000000       299200000.00    32441200000.00
13536600000.00  220000000.0     220000000.0

Query:
SELECT "fName-lName" FROM "Director" WHERE "fName-lName" LIKE 'Ru%';

Result:
"RussoAnthony"
"RussoJoe"

Query:

SELECT *
FROM "Movie"
WHERE "Movie"."movieID" IN (
        SELECT "movieID"
        FROM "movie genre"
        WHERE "genreName" =
                (SELECT "genreName" FROM "Genre" WHERE "Genre"."genreName" = 'Sci-
Fi')
        AND "movieID" IN
                (SELECT "movieID"
                 FROM "movie cast"
                 WHERE "actorID" IN
                        (SELECT "actorID"
                         FROM "Actor"
                         WHERE "Actor"."dateOfBirth" < DATE('1982-10-30')
                        )
                )
);

Result:
"dfc75159-d2a6-455b-8c77-44997db7bc6f"  "2b397197-3d72-4535-942b-46ecc3530739"
"Avengers: Endgame"     "2019-04-26"    8       356000000       2797501328

Query:
SELECT "Actor1"."fName-lName", "Actor2"."fName-lName"
    FROM "Actor" as "Actor1", "Actor" as "Actor2"
        WHERE ("Actor1"."eyecolour" = 'Dark Brown'
        AND "Actor2"."eyecolour" = 'Dark Brown'
        AND "Actor1"."fName-lName" != "Actor2"."fName-lName"
        AND "Actor1"."fName-lName" < "Actor2"."fName-lName");

Result:
"Downey-Robert" "Ruffalo-Mark"
```

Lab 07

SQL2.sql

```
-- Task 2

-- Siddiqui Farees (100-780-513)

-- 1 a) From table actors find the average age
```

```
SELECT AVG(AGE(CURRENT_DATE, "dateOfBirth"))  as "average age" FROM "Actor";



-- 1 b) Find the number of different countries in which actors from the movie “The
Avengers” were born

SELECT COUNT(DISTINCT "location"."country") FROM "location", "Actor", "Movie",
"movie cast"

    WHERE "Movie"."movieID" = (SELECT "movieID" FROM "Movie" WHERE "movieTitle" =
'The Avengers')



-- 1 c) Find the number of actors with Dark Brown eyes

SELECT COUNT(*) FROM "Actor" WHERE "eyecolour" = 'Dark Brown';



-- 1 d) what is the number of movies in which Robert Downey Jr. was playing

SELECT COUNT(*) FROM "Movie" WHERE "movieID" IN (SELECT "movieID" FROM "movie cast"
WHERE "movie cast"."actorID" IN (SELECT "actorID" FROM "Actor" WHERE "fName-lName"
= 'Downey-Robert'));

-- 1 e) Find the minimum, average, and maximum budget for each type of movie

SELECT "Genre"."genreName",

    MIN("Movie"."budget") as "MINIMUM",

    AVG("Movie"."budget") as "AVERAGE",

    MAX("Movie"."budget") as "MAXIMUM"

FROM "Genre", "Movie"

WHERE "Movie"."movieID" IN

    (SELECT "movieID" FROM "movie genre" WHERE "movie genre"."genreName" =
"Genre"."genreName")

GROUP BY "Genre"."genreName";
```

```
-- 1 f) Find the average rating for each movie which has a director born in Toronto or blue eyecolour

SELECT "movieTitle", "rating"

FROM "Movie"

WHERE "movieID" IN

    (SELECT "movieID"

     FROM "movieDirector"

     WHERE "directorID" in

        (SELECT "directorID"

         FROM "Director"

         WHERE "directorID" IN

            (SELECT "directorID"

             FROM "location"

             WHERE city = 'Toronto'

            )

        )

    );

-- 1 g) Find all the movies that have actors born in atleast 2 different countries (this answer seems very wrong)

SELECT "movieID" FROM "movie cast"

WHERE "actorID" IN (SELECT "actorID" FROM "Actor"

WHERE "locationID" IN (SELECT "locationID" FROM "location"

WHERE "country" IN (SELECT "country" FROM "location" GROUP BY "country" HAVING COUNT("country") > 2)));
```

```
-- 1 h) Find how many awards each movie in total has received and rank the movies (display the title) by the amount of awards

SELECT "Movie"."movieTitle", COUNT("movie awards"."movieID") AS "awards"

    FROM "Movie"

    INNER JOIN "movie awards" ON "Movie"."movieID" = "movie awards"."movieID"

    GROUP BY "Movie"."movieTitle"

    ORDER BY "awards" DESC;
```

Lab 07 Results

Results.txt

```
Query:
    -- 1 a) From table actors find the average age
    SELECT AVG(AGE(CURRENT_DATE, "dateOfBirth"))  as "average age" FROM "Actor";

Result:
    "44 years 1 mon 27 days 06:51:25.701943"

Query:
    -- 1 b) Find the number of different countries in which actors from the movie “The Avengers” were born
    SELECT COUNT(DISTINCT "location"."country") FROM "location", "Actor", "Movie", "movie cast"
        WHERE "Movie"."movieID" = (SELECT "movieID" FROM "Movie" WHERE "movieTitle" = 'The Avengers')

Result:
    "4"

Query:
    -- 1 c) Find the number of actors with Dark Brown eyes
    SELECT COUNT(*) FROM "Actor" WHERE "eyecolour" = 'Dark Brown';

Result:
    "2"
```

```
Query:
    -- 1 d) what is the number of movies in which Robert Downey Jr. was playing
    SELECT COUNT(*) FROM "Movie" WHERE "movieID" IN (SELECT "movieID" FROM "movie
cast" WHERE "movie cast"."actorID" IN (SELECT "actorID" FROM "Actor" WHERE "fName-
lName" = 'Downey-Robert'));

Result:
    "1"


Query:
    -- 1 e) Find the minimum, average, and maximum budget for each type of movie
    SELECT "Genre"."genreName",
        MIN("Movie"."budget") as "MINIMUM",
        AVG("Movie"."budget") as "AVERAGE",
        MAX("Movie"."budget") as "MAXIMUM"
    FROM "Genre", "Movie"
    WHERE "Movie"."movieID" IN
        (SELECT "movieID" FROM "movie genre" WHERE "movie genre"."genreName" =
"Genre"."genreName")
    GROUP BY "Genre"."genreName";

Result:
    "Action"        200000000   210000000.00000000      220000000
    "Sci-Fi"        250000000   303000000.00000000      356000000
    "Test Genre"        256000000       256000000.00000000      256000000
    "Thriller"      321000000   321000000.00000000      321000000

Query:
    -- 1 f) Find the average rating for each movie which has a director born in
Toronto or blue eyecolour
    SELECT "movieTitle", "rating"
    FROM "Movie"
    WHERE "movieID" IN
        (SELECT "movieID"
        FROM "movieDirector"
        WHERE "directorID" in
            (SELECT "directorID"
            FROM "Director"
            WHERE "directorID" IN
                (SELECT "directorID"
                FROM "location"
                WHERE city = 'Toronto'
                )
            )
        );

Result:

    "Black Panther"                 7
    "The Avengers"                  8
    "Avengers: Age of Ultron"   7
    "Avengers: Infinity War"    8
    "Avengers: Endgame"         8
    "Test Movie Name"           9

Query:
    -- 1 g) Find all the movies that have actors born in atleast 2 different
countries
    SELECT "movieID" FROM "movie cast"
    WHERE "actorID" IN (SELECT "actorID" FROM "Actor"
    WHERE "locationID" IN (SELECT "locationID" FROM "location"
    WHERE "country" IN (SELECT "country" FROM "location" GROUP BY "country" HAVING
COUNT("country") > 2)));

-- (The results are the UUID of each movie)
Result:
    "1a6711e7-4572-4529-aba3-2c0fa829fb41"
    "0d505440-b4a5-4ccf-bd78-a15ac095121e"
    "e56fdb62-e63d-4f6f-9aa5-b8f4c94faf0e"


-- Query:
    -- 1 h) Find how many awards each movie in total has received and rank the
movies (display the title) by the amount of awards
    SELECT "Movie"."movieTitle", COUNT("movie awards"."movieID") AS "awards"
        FROM "Movie"
        INNER JOIN "movie awards" ON "Movie"."movieID" = "movie awards"."movieID"
        GROUP BY "Movie"."movieTitle"
        ORDER BY "awards" DESC;

Result:
    "Test Movie Name"           2
    "The Avengers"                  1
    "Avengers: Infinity War"    1
    "Avengers: Age of Ultron"   1
    "Avengers: Endgame"         1
    "Black Panther"                 1
```

Lab 08

lab08.java

```java
import java.sql.*;

public class lab08 {

    private String url = "jdbc:postgresql://localhost:5432/postgres";
    private String user = "postgres";
    private String password = "Sadeem7apple";

    public Connection connect() throws SQLException {
        return DriverManager.getConnection(url, user, password);
    }

    public void getBooks() {

        String sql = "SELECT * FROM \"Actor\"";

        try (Connection conn = connect();
                Statement stmt = conn.createStatement();
                ResultSet rs = stmt.executeQuery(sql)) {
            System.out.println(rs);
            displayResult(rs);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }

    }

    public void displayResult(ResultSet rs) throws SQLException {
        while (rs.next()) {
            System.out.println(rs.getString("actorID") + "\t"
                    + rs.getString("fName-lName") + "\t"
                    + rs.getString("locationID"));
        }
    }

    public static void main(String[] args) {
        lab08 app = new lab08();
        try {
            app.connect();
            System.out.println("Successful Connection");
            app.getBooks();

        } catch (SQLException ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```

Lab08 Output

output.txt

```
Successful Connection
org.postgresql.jdbc.PgResultSet@1b083826
{3d4dcd29-acb9-46c5-b7d5-f7e3353e6c11}  Downey-Robert           4b0ce179-ae27-4762-
a4e0-b92ca008c381
{d1579b56-1f37-4fde-ab86-5225795e2cb0}  Evans-Chris             5138a119-cb80-411a-
baee-b0765922b62d
{045b04f4-7b6f-4fa9-a13d-058da06502a7}  Ruffalo-Mark            74ab37b4-7371-4a2d-
a0c1-eb19b736f49b
{7d6fa880-ef9b-41ee-9e56-5ecf82af8a92}  Hemsworth-Chris         09d25ea9-b121-4f9a-
9fca-bdd8f21e84a5
{6baf4aff-adf9-462d-99c8-aecdcbc15a79}  Johansson-Scarlett      a69b49fe-03c0-4f93-
8fb8-17923d199c5c
{e9838673-fca8-486b-a067-caf69f34437a}  Test Actor Name         b3b4586b-8da8-4223-
8b38-444eea8dcac1
```

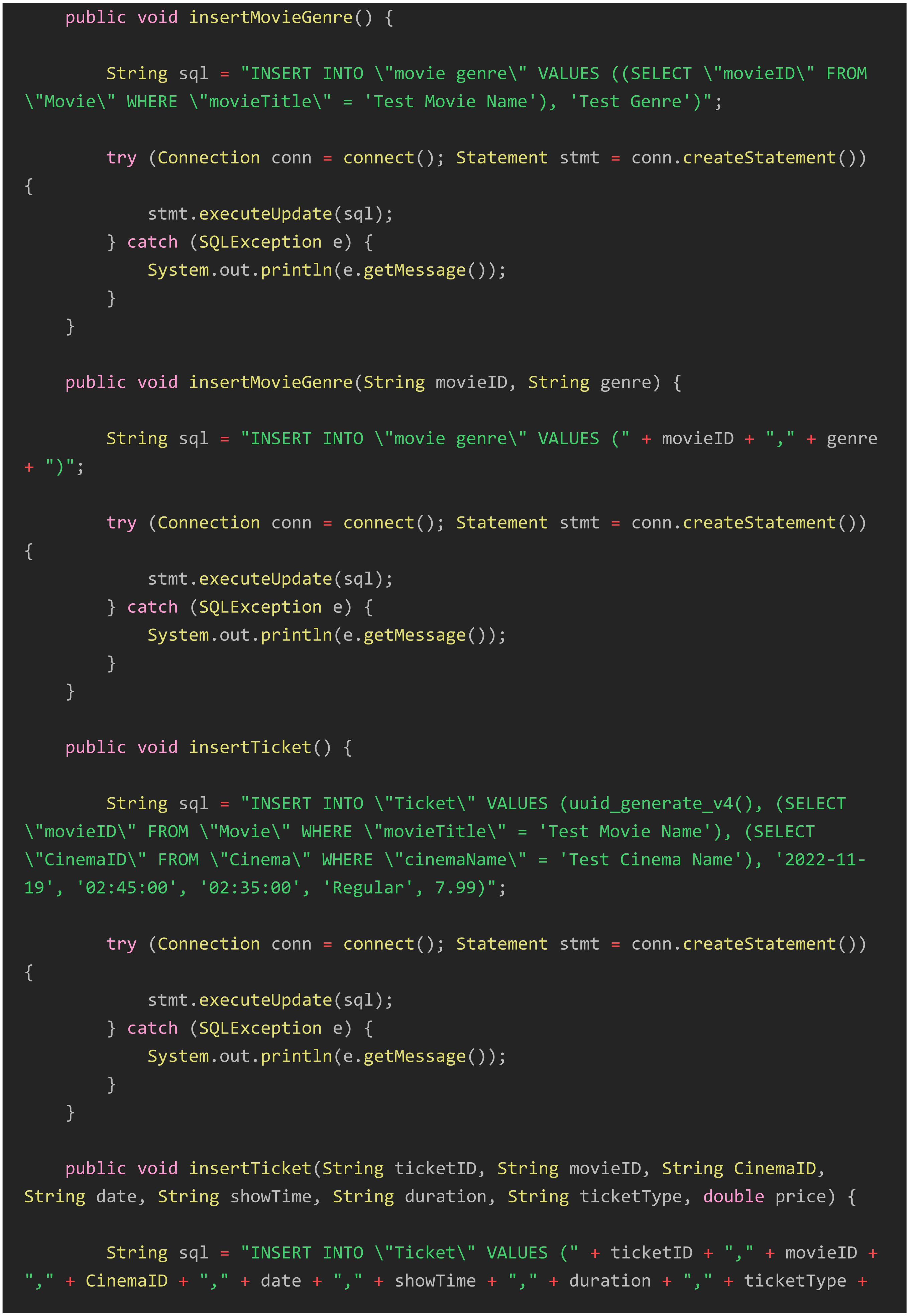
Lab 09

lab09.java

```
import java.sql.*;

public class lab09 {
    private String url = "jdbc:postgresql://localhost:5432/postgres";
    private String user = "postgres";
    private String password = "Sadeem7apple";

    public Connection connect() throws SQLException {
        return DriverManager.getConnection(url, user, password);
    }

    public void getTable(String table) {
        String sql = "SELECT * FROM \"" + table + "\"";
        try (Connection conn = connect();
                Statement stmt = conn.createStatement();
                ResultSet rs = stmt.executeQuery(sql)) {
            displayResult(rs);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertFestivals() {
        String sql = "INSERT INTO \"Festivals\" VALUES (uuid_generate_v4(), 'Test
Festival')";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }


    public void insertFestivals(String festivalID, String festivalName) {
        String sql = "INSERT INTO \"Festivals\" VALUES (" + festivalID + "," +
festivalName + ")";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertAwards() {
        String sql = "INSERT INTO \"Awards\" VALUES (uuid_generate_v4(), 'Test
Award', 'Test Category')";
        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertAwards(String AwardID, String AwardName, String
AwardCategory) {
        String sql = "INSERT INTO \"Awards\" VALUES (" + AwardID + "," + AwardName
+ "," + AwardCategory + ")";
        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertGenre() {

        String sql = "INSERT INTO \"Genre\" VALUES ('Test Genre', 'Test Genre
Description')";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertGenre(String GenreName, String GenreDescription) {
        String sql = "INSERT INTO \"Genre\" VALUES (" + GenreName + "," +
GenreDescription + ")";
        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertLocation() {
        String sql = "INSERT INTO \"location\" VALUES (uuid_generate_v4(), 'Test
Location City', 'Test Location Province', 'Test Location Country')";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertLocation(String LocationID, String LocationCity, String
LocationProvince, String LocationCountry) {

        String sql = "INSERT INTO \"location\" VALUES (" + LocationID + "," +
LocationCity + "," + LocationProvince + "," + LocationCountry + ")";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertDirector() {
        String sql = "INSERT INTO \"Director\" VALUES (uuid_generate_v4(), (SELECT
\"locationID\" FROM \"location\" WHERE city = 'Test Location City'), 'Test Director
Name', 1984)";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }


    public void insertDirector(String DirectorID, String LocationID, String
DirectorName, int DirectorBirthYear) {

        String sql = "INSERT INTO \"Director\" VALUES (" + DirectorID + "," +
LocationID + "," + DirectorName + "," + DirectorBirthYear + ")";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
```

```
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertMovie() {

        String sql = "INSERT INTO \"Movie\" VALUES (uuid_generate_v4(), (SELECT
\"festivalID\" FROM \"Festivals\" WHERE \"festivalName\" = 'Test Festival'), 'Test
Movie Name', '2022-11-19', 9, 256000000, 4797508348)";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertMovie(String movieID, String festivalID, String movieName,
String movieReleaseDate, int rating, int movieBudget, int gross) {

        String sql = "INSERT INTO \"Movie\" VALUES (uuid_generate_v4(), (SELECT
\"festivalID\" FROM \"Festivals\" WHERE \"festivalName\" = 'Test Festival'), 'Test
Movie Name', '2022-11-19', 9, 256000000, 4797508348)";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertActor() {

        String sql = "INSERT INTO \"Actor\" VALUES (ARRAY[uuid_generate_v4()],
(SELECT \"locationID\" FROM \"location\" WHERE city = 'Test Location City'), NULL,
'1984-01-23', 'Test Actor Name', 'Grey')";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertActor(String actorID, String locationID, String
universityName, String actorDOB, String actorName, String actorHairColor) {

        String sql = "INSERT INTO \"Actor\" VALUES (" + actorID + "," + locationID
+ "," + universityName + "," + actorDOB + "," + actorName + "," + actorHairColor +
")";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertUniversity() {

        String sql = "INSERT INTO \"University\" VALUES ('Test University Name',
(SELECT \"directorID\" FROM \"Director\" WHERE \"fName-lName\" = 'Test Director
Name'), 'Science', 26123, false, 'green-white', 'test university description')";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertUniversity(String universityName, String directorID, String
faculty, int universitySize, String isPublic, String universityColors, String
universityDescription) {

        String sql = "INSERT INTO \"University\" VALUES (" + universityName + "," +
directorID + "," + faculty + "," + universitySize + "," + isPublic + "," +
universityColors + "," + universityDescription + ")";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertCinema() {
        String sql = "INSERT INTO \"Cinema\" VALUES(uuid_generate_v4(), (SELECT
\"locationID\" FROM \"location\" WHERE city = 'Test Location City'), 'Test Cinema
Name')";
        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }



    public void insertCinema(String cinemaID, String locationID, String cinemaName)
{

        String sql = "INSERT INTO \"Cinema\" VALUES(" + cinemaID + "," + locationID
+ "," + cinemaName + ")";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertMovieAward() {

        String sql = "INSERT INTO \"movie awards\" VALUES ((SELECT \"movieID\" FROM
\"Movie\" WHERE \"movieTitle\" = 'Test Movie Name'), (SELECT \"AwardID\" FROM
\"Awards\" WHERE \"awardName\" = 'Test Award'))";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertMovieAward(String movieID, String awardID) {
        String sql = "INSERT INTO \"movie awards\" VALUES (" + movieID + "," +
awardID + ")";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertMovieCast() {

        String sql = "INSERT INTO \"movie cast\" VALUES ((SELECT \"movieID\" FROM
\"Movie\" WHERE \"movieTitle\" = 'Test Movie Name'), (SELECT \"actorID\" FROM
\"Actor\" WHERE \"fName-lName\" = 'Test Actor Name'))";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertMovieCast(String movieID, String actorID) {

        String sql = "INSERT INTO \"movie cast\" VALUES (" + movieID + "," +
actorID + ")";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertPlayedIn() {

        String sql = "INSERT INTO \"playedIn\" VALUES ((SELECT \"movieID\" FROM
\"Movie\" WHERE \"movieTitle\" = 'Test Movie Name'), (SELECT \"CinemaID\" FROM
\"Cinema\" WHERE \"cinemaName\" = 'Test Cinema Name'))";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
```

```
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertPlayedIn(String movieID, String cinemaID) {

        String sql = "INSERT INTO \"playedIn\" VALUES (" + movieID + "," + cinemaID
+ ")";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertMovieDirector() {

        String sql = "INSERT INTO \"movieDirector\" VALUES ((SELECT \"movieID\"
FROM \"Movie\" WHERE \"movieTitle\" = 'Test Movie Name'), (SELECT \"directorID\"
FROM \"Director\" WHERE \"fName-lName\" = 'Test Director Name'))";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertMovieDirector(String movieID, String directorID) {

        String sql = "INSERT INTO \"movieDirector\" VALUES (" + movieID + "," +
directorID + ")";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
```



```
    public void insertMovieGenre() {

        String sql = "INSERT INTO \"movie genre\" VALUES ((SELECT \"movieID\" FROM
\"Movie\" WHERE \"movieTitle\" = 'Test Movie Name'), 'Test Genre')";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertMovieGenre(String movieID, String genre) {

        String sql = "INSERT INTO \"movie genre\" VALUES (" + movieID + "," + genre
+ ")";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertTicket() {

        String sql = "INSERT INTO \"Ticket\" VALUES (uuid_generate_v4(), (SELECT
\"movieID\" FROM \"Movie\" WHERE \"movieTitle\" = 'Test Movie Name'), (SELECT
\"CinemaID\" FROM \"Cinema\" WHERE \"cinemaName\" = 'Test Cinema Name'), '2022-11-
19', '02:45:00', '02:35:00', 'Regular', 7.99)";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void insertTicket(String ticketID, String movieID, String CinemaID,
String date, String showTime, String duration, String ticketType, double price) {

        String sql = "INSERT INTO \"Ticket\" VALUES (" + ticketID + "," + movieID +
"," + CinemaID + "," + date + "," + showTime + "," + duration + "," + ticketType +
"," + price + ")";

        try (Connection conn = connect(); Statement stmt = conn.createStatement())
{
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void displayResult(ResultSet rs) throws SQLException {
        ResultSetMetaData rsmd = rs.getMetaData();
        int columnsNumber = rsmd.getColumnCount();
        while (rs.next()) {
            for (int i = 1; i <= columnsNumber; i++) {
                if (i > 1)
                    System.out.print(",  ");
                String columnValue = rs.getString(i);
                System.out.print(columnValue + " (" + rsmd.getColumnName(i) + ")");
            }
            System.out.println("");
        }
    }

    // Queries

    public void getCanadianDirectors() {
        String sql = "SELECT \"fName-lName\" FROM \"Director\" WHERE \"locationID\"
IN (SELECT \"locationID\" FROM \"location\" WHERE \"country\" = 'Canada')";
        try (Connection conn = connect();
                Statement stmt = conn.createStatement();
                ResultSet rs = stmt.executeQuery(sql)) {
            System.out.println("Canadian Directors:");
            displayResult(rs);
            System.out.println("----------------");
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void getMoviesByDirector(String directorName) {
        String sql = "SELECT \"movieTitle\" FROM \"Movie\" WHERE \"movieID\" IN
(SELECT \"movieID\" FROM \"movieDirector\" WHERE \"directorID\" IN (SELECT
\"directorID\" FROM \"Director\" WHERE \"fName-lName\" = ?))";
        try (Connection conn = connect();
                PreparedStatement pstmt = conn.prepareStatement(sql)) {
```

```java
            pstmt.setString(1, directorName);
            ResultSet rs = pstmt.executeQuery();
            System.out.printf("Movies by [%s]:\n", directorName);
            displayResult(rs);
            System.out.println("----------------");
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void getActorsWhereProductionIsGreaterThanX(int production) {
        String sql = "SELECT \"fName-lName\" FROM \"Actor\" WHERE \"actorID\" IN
(SELECT \"actorID\" FROM \"movie cast\" WHERE \"movieID\" IN (SELECT \"movieID\"
FROM \"Movie\" WHERE \"budget\" > ?))";
        try (Connection conn = connect();
                PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setInt(1, production);
            ResultSet rs = pstmt.executeQuery();
            System.out.printf("Actors where production is greater than [%d]:\n",
production);
            displayResult(rs);
            System.out.println("----------------");
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void getSimilarDirectorNames(String pattern) {
        String sql = "SELECT \"fName-lName\" FROM \"Director\" WHERE \"fName-
lName\" LIKE ?";
        try (Connection conn = connect();
                PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setString(1, pattern);
            ResultSet rs = pstmt.executeQuery();
            System.out.printf("Similar Director Names to pattern [%s]:\n",
pattern);
            displayResult(rs);
            System.out.println("----------------");
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public static void main(String[] args) {
        lab09 app = new lab09();
        try {
            app.connect();
            System.out.println("Successful Connection");
            app.insertFestivals(); // Inserted a new festival, if ran again will
insert a
            app.insertAwards();
            app.insertGenre();
            app.insertLocation();
            app.insertDirector();
            app.insertActor();
            app.insertMovie();
            app.insertUniversity();
            app.insertCinema();
            app.insertMovieAward();
            app.insertMovieCast();
            app.insertPlayedIn();
            app.insertMovieDirector();
            app.insertFestivals("uuid_generate_v4()", "'Test Festival 2'");
            app.insertAwards("uuid_generate_v4()", "'Test Award 2'", "'Test Award
Category 2'");
            app.insertGenre("'Test Genre 2'", "'Test Genre 2 Description'");
            app.insertLocation("uuid_generate_v4()", "'Test Location 2 City'",
"'Test Location 2 State'", "'Test Location 2 Country'");

            app.insertDirector("uuid_generate_v4()", "(SELECT \"locationID\" FROM
\"location\" WHERE city = 'Test Location 2 City')", "'Test Director 2 Name'",
1990);

            app.insertActor("ARRAY[uuid_generate_v4()]",
                    "(SELECT \"locationID\" FROM \"location\" WHERE city = 'Test
Location 2 City')", "NULL", "'1978-05-17'", "'Test Actor 2 Name'", "Black");

            app.insertMovie("uuid_generate_v4()", "(SELECT \"festivalID\" FROM
\"Festivals\" WHERE \"festivalName\" = 'Test Festival 2')", "'Test Movie 2 Name'",
"'2022-12-30'", 4, 256000000, 479750834);



            app.insertUniversity("'Test University 2 Name'", "(SELECT
\"directorID\" FROM \"Director\" WHERE \"fName-lName\" = 'Test Director Name')",
"'Arts'", 31000, "false", "'yellow-green'", "'Test University 2 Description'");


            app.insertCinema("uuid_generate_v4()", "(SELECT \"locationID\" FROM
\"location\" WHERE city = 'Test Location 2 City')", "'Test Cinema 2 Name'");
```

```
            app.insertMovieAward("'7827b6e1-fbdd-43b4-b220-a607e99028f9'", "(SELECT \"AwardID\" FROM \"Awards\" WHERE \"awardName\" = 'Test Award 2')");
            app.insertMovieCast("'7827b6e1-fbdd-43b4-b220-a607e99028f9'", "'{e9838673-fca8-486b-a067-caf69f34437a}'");
            app.insertPlayedIn("'7827b6e1-fbdd-43b4-b220-a607e99028f9'", "(SELECT \"CinemaID\" FROM \"Cinema\" WHERE \"cinemaName\" = 'Test Cinema 2 Name')");

            System.out.println("-----Festivals-----");

            app.getTable("Festivals");

            System.out.println("------Awards-------");

            app.getTable("Awards");

            System.out.println("------Genre--------");

            app.getTable("Genre");

            System.out.println("-----Locations-----");

            app.getTable("location");

            System.out.println("------Director-----");

            app.getTable("Director");

            System.out.println("------Actor--------");

            app.getTable("Actor");

            System.out.println("------Movie--------");

            app.getTable("Movie");

            System.out.println("----University-----");

            app.getTable("University");

            System.out.println("------Cinema-------");

            app.getTable("Cinema");

            System.out.println("-----Movie Awards-----");

            app.getTable("movie awards");
            System.out.println("-----Movie Cast-----");
            app.getTable("movie cast");
            System.out.println("-----Played in-----");
            app.getTable("playedIn");
            System.out.println("-----Movie Director-----");
            app.getTable("movieDirector");
            System.out.println("-----Movie Genre-----");
            app.getTable("movie genre");
            System.out.println("-----Ticket-----");
            app.getTable("Ticket");

            // Queries
            app.getCanadianDirectors();
            app.getMoviesByDirector("WhedonJoss");
            app.getActorsWhereProductionIsGreaterThanX(1000000);
            app.getSimilarDirectorNames("Ru%");

        } catch (SQLException ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```

Lab 09 Output

output.txt

```
Successful Connection
-----Festivals-----
827c99be-c2d3-40ed-ac3b-03b782f42769 (festivalID),  Toronto International Film
Festival (festivalName)
8331ff8e-e86f-4545-81de-97acf485c355 (festivalID),  Sundance Film Festival
(festivalName)
aa8a8076-edab-47f7-b6bf-ae650e21f6b0 (festivalID),  Cannes Film Festival
(festivalName)
2b397197-3d72-4535-942b-46ecc3530739 (festivalID),  Venice Film Festival
(festivalName)
390b9f31-47a0-4145-af96-adad30e74419 (festivalID),  Berlin International Film
Festival (festivalName)
169d3bf0-1771-4afd-8bcf-43131cfd9b41 (festivalID),  Test Festival (festivalName)
2f0a0f64-c910-4444-af58-0c517abe5888 (festivalID),  Test Festival 2 (festivalName)
------Awards-------
46387b74-903b-4d6e-9bf5-f4dac9c67e2f (AwardID),  Best Picture (awardName),  best
picture (awardCategory)
d397607c-e1ae-4f0a-838a-e0e6eff2b829 (AwardID),  Best Director (awardName),  best
director (awardCategory)
1dc2e4bc-6871-424e-a93d-57d5f48375e7 (AwardID),  Best Actor (awardName),  best
actor (awardCategory)
f38d0db4-3155-41fb-abd1-0101c4e11818 (AwardID),  Best Actress (awardName),  best
actress (awardCategory)
da2a1fe7-f16e-4f66-a64f-3e23a5ae9cfe (AwardID),  Best Screenplay (awardName),  best
screenplay (awardCategory)
edbbc1bd-98f5-4c2a-9f9f-a7252ce8cb90 (AwardID),  Test Award (awardName),  Test
Category (awardCategory)
2196196e-6b37-4409-a671-325c9191485b (AwardID),  Test Award 2 (awardName),  Test
Award Category 2 (awardCategory)
------Genre--------
Action (genreName),  Movies in the action genre are fast-paced and include a lot of
action like fight scenes, chase scenes, and slow-motion shots. They can feature
superheroes, martial arts, or exciting stunts. These high-octane films are more
about the execution of the plot rather than the plot itself (Description)
Sci-Fi (genreName),  Science fiction is a genre of fiction in which the stories
often tell about science and technology of the future. It is important to note that
science fiction has a relationship with the principles of science?these stories
involve partially true- partially fictitious laws or theories of science
```

```
(Description)
Horror (genreName),  Horror is a genre of literature, film, and television that is
meant to scare, startle, shock, and even repulse audiences. The key focus of a
horror novel, horror film, or horror TV show is to elicit a sense of dread in the
reader through frightening images, themes, and situations (Description)
Thriller (genreName),  Thriller is a genre of fiction, having numerous, often
overlapping subgenres. Thrillers are characterized and defined by the moods they
elicit, giving viewers heightened feelings of suspense, excitement, surprise,
anticipation and anxiety (Description)
Romance (genreName),  The romance genre is defined by two aspects that can be found
in every romance book or novel: a central love story between characters. an
emotionally satisfying, happy ending (Description)
Test Genre (genreName),  Test Genre Description (Description)
Test Genre 2 (genreName),  Test Genre 2 Description (Description)
-----Locations-----
4b0ce179-ae27-4762-a4e0-b92ca008c381 (locationID),  Toronto (city),  Ontario
(state),  Canada (country)
5138a119-cb80-411a-baee-b0765922b62d (locationID),  San Jose (city),  California
(state),  USA (country)
74ab37b4-7371-4a2d-a0c1-eb19b736f49b (locationID),  Miami (city),  Florida (state),
USA (country)
09d25ea9-b121-4f9a-9fca-bdd8f21e84a5 (locationID),  Oshawa (city),  Ontario
(state),  Canada (country)
a69b49fe-03c0-4f93-8fb8-17923d199c5c (locationID),  Boston (city),  Massachusetts
(state),  USA (country)
b3b4586b-8da8-4223-8b38-444eea8dcac1 (locationID),  Test Location City (city),
Test Location Province (state),  Test Location Country (country)
6f989189-81ba-4212-94b8-a1ba7f943f48 (locationID),  Test Location 2 City (city),
Test Location 2 State (state),  Test Location 2 Country (country)
------Director-----
2d2c9794-f77d-4d60-a0cc-68716434618f (directorID),  4b0ce179-ae27-4762-a4e0-
b92ca008c381 (locationID),  CooglerRyan (fName-lName),  1986 (yearOfBirth)
b2ccd76a-7360-4901-8cdd-e8050b1b53d9 (directorID),  5138a119-cb80-411a-baee-
b0765922b62d (locationID),  WhedonJoss (fName-lName),  1964 (yearOfBirth)
6aed248a-888f-43a8-801c-5ad8bc965e55 (directorID),  a69b49fe-03c0-4f93-8fb8-
17923d199c5c (locationID),  RussoAnthony (fName-lName),  1970 (yearOfBirth)
62d379e5-a71c-4190-a284-14786639ca8f (directorID),  09d25ea9-b121-4f9a-9fca-
bdd8f21e84a5 (locationID),  RussoJoe (fName-lName),  1971 (yearOfBirth)
a2836203-8639-4864-9ab4-47a3d33d381d (directorID),  74ab37b4-7371-4a2d-a0c1-
eb19b736f49b (locationID),  BodenAnna (fName-lName),  1979 (yearOfBirth)
```

```
a2e2f980-ebca-4262-bea9-80f2d5219933 (directorID),  b3b4586b-8da8-4223-8b38-
444eea8dcac1 (locationID),  Test Director Name (fName-lName),  1984 (yearOfBirth)
e4c7d9c8-4a21-4391-a669-dd87e85a7b55 (directorID),  6f989189-81ba-4212-94b8-
a1ba7f943f48 (locationID),  Test Director 2 Name (fName-lName),  1990 (yearOfBirth)
------Actor--------
{3d4dcd29-acb9-46c5-b7d5-f7e3353e6c11} (actorID),  4b0ce179-ae27-4762-a4e0-
b92ca008c381 (locationID),  null (universityName),  1965-04-04 (dateOfBirth),
Downey-Robert (fName-lName),  Dark Brown (eyecolour)
{d1579b56-1f37-4fde-ab86-5225795e2cb0} (actorID),  5138a119-cb80-411a-baee-
b0765922b62d (locationID),  null (universityName),  1981-06-13 (dateOfBirth),
Evans-Chris (fName-lName),  Green (eyecolour)
{045b04f4-7b6f-4fa9-a13d-058da06502a7} (actorID),  74ab37b4-7371-4a2d-a0c1-
eb19b736f49b (locationID),  null (universityName),  1967-11-22 (dateOfBirth),
Ruffalo-Mark (fName-lName),  Dark Brown (eyecolour)
{7d6fa880-ef9b-41ee-9e56-5ecf82af8a92} (actorID),  09d25ea9-b121-4f9a-9fca-
bdd8f21e84a5 (locationID),  null (universityName),  1983-08-11 (dateOfBirth),
Hemsworth-Chris (fName-lName),  Blue (eyecolour)
{6baf4aff-adf9-462d-99c8-aecdcbc15a79} (actorID),  a69b49fe-03c0-4f93-8fb8-
17923d199c5c (locationID),  null (universityName),  1984-11-22 (dateOfBirth),
Johansson-Scarlett (fName-lName),  Green (eyecolour)
{78ece862-ce2a-4bf7-9ba3-10f3010918ed} (actorID),  b3b4586b-8da8-4223-8b38-
444eea8dcac1 (locationID),  null (universityName),  1984-01-23 (dateOfBirth),  Grey
(fName-lName),  null (eyecolour)
{e9838673-fca8-486b-a067-caf69f34437a} (actorID),  b3b4586b-8da8-4223-8b38-
444eea8dcac1 (locationID),  null (universityName),  1984-01-23 (dateOfBirth),  Test
Actor Name (fName-lName),  Grey (eyecolour)
------Movie--------
1a6711e7-4572-4529-aba3-2c0fa829fb41 (movieID),  827c99be-c2d3-40ed-ac3b-
03b782f42769 (festivalID),  Black Panther (movieTitle),  2018-02-16 (releaseDate),
7 (rating),  200000000 (budget),  1347597973 (gross)
e56fdb62-e63d-4f6f-9aa5-b8f4c94faf0e (movieID),  8331ff8e-e86f-4545-81de-
97acf485c355 (festivalID),  The Avengers (movieTitle),  2012-05-04 (releaseDate),
8 (rating),  220000000 (budget),  1518815515 (gross)
0d505440-b4a5-4ccf-bd78-a15ac095121e (movieID),  aa8a8076-edab-47f7-b6bf-
ae650e21f6b0 (festivalID),  Avengers: Age of Ultron (movieTitle),  2015-05-01
(releaseDate),  7 (rating),  250000000 (budget), 1402809540 (gross)
33dadbed-4df2-4d2f-8660-dc12e274d822 (movieID),  827c99be-c2d3-40ed-ac3b-
03b782f42769 (festivalID),  Avengers: Infinity War (movieTitle),  2018-04-27
(releaseDate),  8 (rating),  321000000 (budget), 2048359754 (gross)
dfc75159-d2a6-455b-8c77-44997db7bc6f (movieID),  2b397197-3d72-4535-942b-
```

```
46ecc3530739 (festivalID),  Avengers: Endgame (movieTitle),  2019-04-26
(releaseDate),  8 (rating),  356000000 (budget), 2797501328 (gross)
75f5c5ae-ff75-4e03-bf56-85e50337c19b (movieID),  169d3bf0-1771-4afd-8bcf-
43131cfd9b41 (festivalID),  Test Movie Name (movieTitle),  2022-11-19
(releaseDate),  9 (rating),  256000000 (budget),  4797508348 (gross)
7827b6e1-fbdd-43b4-b220-a607e99028f9 (movieID),  169d3bf0-1771-4afd-8bcf-
43131cfd9b41 (festivalID),  Test Movie Name (movieTitle),  2022-11-19
(releaseDate),  9 (rating),  256000000 (budget),  4797508348 (gross)
----University-----
New York University (universityName),  2d2c9794-f77d-4d60-a0cc-68716434618f
(directorID),  Science (department),  28772 (studentCount),  t (private),  white-
purple (color),  New York University is a private research university in New York
City (description)

Massachusetts Institute of Technology (universityName),  b2ccd76a-7360-4901-8cdd-
e8050b1b53d9 (directorID),  Mathematics (department),  11376 (studentCount),  t
(private),  red-black-silver-cardinal (color),  The Massachusetts Institute of
Technology is a private land-grant research university in Cambridge, Massachusetts.
Established in 1861, MIT has since played a key role in the development of modern
technology and science, ranking among the top academic institutions in the world
(description)

University of Toronto (universityName),  6aed248a-888f-43a8-801c-5ad8bc965e55
(directorID),  Arts and Science (department),  72389 (studentCount),  f (private),
white-blue (color),  The University of Toronto is a public research university in
Toronto, Ontario, Canada, located on the grounds that surround Queens Park. It was
founded by royal charter in 1827 as Kings College, the first institution of higher
learning in Upper Canada (description)

Queens University (universityName),  62d379e5-a71c-4190-a284-14786639ca8f
(directorID),  Computing (department),  24143 (studentCount),  f (private),  blue-
gold-red (color),  Queens University at Kingston, commonly known as Queens
University or simply Queens, is a public research university in Kingston, Ontario,
Canada. Queens holds more than 1,400 hectares of land throughout Ontario and owns
Herstmonceux Castle in East Sussex, England (description)

Princeton University (universityName),  a2836203-8639-4864-9ab4-47a3d33d381d
(directorID),  Science (department),  8623 (studentCount),  t (private),  black-
orange (color),  Princeton University is a private Ivy League research university
in Princeton, New Jersey. Founded in 1746 in Elizabeth as the College of New
```

```
Jersey, Princeton is the fourth-oldest institution of higher education in the
United States and one of the nine colonial colleges chartered before the American
Revolution. (description)

Test University Name (universityName),  a2e2f980-ebca-4262-bea9-80f2d5219933
(directorID),  Science (department),  26123 (studentCount),  f (private),  green-
white (color),  test university description (description)

Test University 2 Name (universityName),  a2e2f980-ebca-4262-bea9-80f2d5219933
(directorID),  Arts (department),  31000 (studentCount),  f (private),  yellow-
green (color),  Test University 2 Description (description)
------Cinema-------
bc4db6ed-80aa-49b4-b9f8-6f461af5fc1c (CinemaID),  4b0ce179-ae27-4762-a4e0-
b92ca008c381 (locationID),  Cineplex Cinemas Yonge-Dundas and VIP (cinemaName)
58615812-eb08-497a-a9c4-96a7cccb20c9 (CinemaID),  5138a119-cb80-411a-baee-
b0765922b62d (locationID),  AMC Saratoga 14 (cinemaName)
9de61cd1-61a8-4009-8bc8-b2b8ca1d0728 (CinemaID),  74ab37b4-7371-4a2d-a0c1-
eb19b736f49b (locationID),  AMC Sunset Place 24 (cinemaName)
a6f0ff9a-705e-4165-9c41-71d231c7b987 (CinemaID),  09d25ea9-b121-4f9a-9fca-
bdd8f21e84a5 (locationID),  Cineplex Cinemas Oshawa (cinemaName)
031333a0-e455-421f-82fd-5694d72c9235 (CinemaID),  a69b49fe-03c0-4f93-8fb8-
17923d199c5c (locationID),  AMC Boston Common 19 (cinemaName)
192c3cb1-fb86-48f2-a33e-222b83c65084 (CinemaID),  b3b4586b-8da8-4223-8b38-
444eea8dcac1 (locationID),  Test Cinema Name (cinemaName)
d628ca15-ad11-4aa1-95f3-6e74bd84a259 (CinemaID),  6f989189-81ba-4212-94b8-
a1ba7f943f48 (locationID),  Test Cinema 2 Name (cinemaName)
-----Movie Awards-----
1a6711e7-4572-4529-aba3-2c0fa829fb41 (movieID),  46387b74-903b-4d6e-9bf5-
f4dac9c67e2f (AwardID)
dfc75159-d2a6-455b-8c77-44997db7bc6f (movieID),  46387b74-903b-4d6e-9bf5-
f4dac9c67e2f (AwardID)
33dadbed-4df2-4d2f-8660-dc12e274d822 (movieID),  46387b74-903b-4d6e-9bf5-
f4dac9c67e2f (AwardID)
0d505440-b4a5-4ccf-bd78-a15ac095121e (movieID),  46387b74-903b-4d6e-9bf5-
f4dac9c67e2f (AwardID)
e56fdb62-e63d-4f6f-9aa5-b8f4c94faf0e (movieID),  46387b74-903b-4d6e-9bf5-
f4dac9c67e2f (AwardID)
75f5c5ae-ff75-4e03-bf56-85e50337c19b (movieID),  edbbc1bd-98f5-4c2a-9f9f-
a7252ce8cb90 (AwardID)
7827b6e1-fbdd-43b4-b220-a607e99028f9 (movieID),  2196196e-6b37-4409-a671-
```

```
325c9191485b (AwardID)
-----Movie Cast-----
1a6711e7-4572-4529-aba3-2c0fa829fb41 (movieID),  {d1579b56-1f37-4fde-ab86-5225795e2cb0} (actorID)
dfc75159-d2a6-455b-8c77-44997db7bc6f (movieID),  {3d4dcd29-acb9-46c5-b7d5-f7e3353e6c11} (actorID)
33dadbed-4df2-4d2f-8660-dc12e274d822 (movieID),  {7d6fa880-ef9b-41ee-9e56-5ecf82af8a92} (actorID)
0d505440-b4a5-4ccf-bd78-a15ac095121e (movieID),  {6baf4aff-adf9-462d-99c8-aecdcbc15a79} (actorID)
e56fdb62-e63d-4f6f-9aa5-b8f4c94faf0e (movieID),  {045b04f4-7b6f-4fa9-a13d-058da06502a7} (actorID)
75f5c5ae-ff75-4e03-bf56-85e50337c19b (movieID),  {e9838673-fca8-486b-a067-caf69f34437a} (actorID)
7827b6e1-fbdd-43b4-b220-a607e99028f9 (movieID),  {e9838673-fca8-486b-a067-caf69f34437a} (actorID)
-----Played in-----
1a6711e7-4572-4529-aba3-2c0fa829fb41 (movieID),  bc4db6ed-80aa-49b4-b9f8-6f461af5fc1c (CinemaID)
dfc75159-d2a6-455b-8c77-44997db7bc6f (movieID),  58615812-eb08-497a-a9c4-96a7cccb20c9 (CinemaID)
33dadbed-4df2-4d2f-8660-dc12e274d822 (movieID),  9de61cd1-61a8-4009-8bc8-b2b8ca1d0728 (CinemaID)
0d505440-b4a5-4ccf-bd78-a15ac095121e (movieID),  a6f0ff9a-705e-4165-9c41-71d231c7b987 (CinemaID)
e56fdb62-e63d-4f6f-9aa5-b8f4c94faf0e (movieID),  031333a0-e455-421f-82fd-5694d72c9235 (CinemaID)
75f5c5ae-ff75-4e03-bf56-85e50337c19b (movieID),  192c3cb1-fb86-48f2-a33e-222b83c65084 (CinemaID)
7827b6e1-fbdd-43b4-b220-a607e99028f9 (movieID),  192c3cb1-fb86-48f2-a33e-222b83c65084 (CinemaID)
7827b6e1-fbdd-43b4-b220-a607e99028f9 (movieID),  d628ca15-ad11-4aa1-95f3-6e74bd84a259 (CinemaID)
-----Movie Director-----
1a6711e7-4572-4529-aba3-2c0fa829fb41 (movieID),  2d2c9794-f77d-4d60-a0cc-68716434618f (directorID)
dfc75159-d2a6-455b-8c77-44997db7bc6f (movieID),  62d379e5-a71c-4190-a284-14786639ca8f (directorID)
33dadbed-4df2-4d2f-8660-dc12e274d822 (movieID),  6aed248a-888f-43a8-801c-5ad8bc965e55 (directorID)
```

```
0d505440-b4a5-4ccf-bd78-a15ac095121e (movieID),  b2ccd76a-7360-4901-8cdd-
e8050b1b53d9 (directorID)
e56fdb62-e63d-4f6f-9aa5-b8f4c94faf0e (movieID),  b2ccd76a-7360-4901-8cdd-
e8050b1b53d9 (directorID)
75f5c5ae-ff75-4e03-bf56-85e50337c19b (movieID),  a2e2f980-ebca-4262-bea9-
80f2d5219933 (directorID)
-----Movie Genre-----
1a6711e7-4572-4529-aba3-2c0fa829fb41 (movieID),  Action (genreName)
dfc75159-d2a6-455b-8c77-44997db7bc6f (movieID),  Sci-Fi (genreName)
33dadbed-4df2-4d2f-8660-dc12e274d822 (movieID),  Thriller (genreName)
0d505440-b4a5-4ccf-bd78-a15ac095121e (movieID),  Sci-Fi (genreName)
e56fdb62-e63d-4f6f-9aa5-b8f4c94faf0e (movieID),  Action (genreName)
75f5c5ae-ff75-4e03-bf56-85e50337c19b (movieID),  Test Genre (genreName)
-----Ticket-----
a5e2de2b-a9bd-4c15-81e6-e867e735cd2a (ticketID),  1a6711e7-4572-4529-aba3-
2c0fa829fb41 (movieID),  bc4db6ed-80aa-49b4-b9f8-6f461af5fc1c (CinemaID),  2020-04-
01 (date),  02:55:00 (showTime),  02:15:00 (duration),  IMAX (cinemaType),  16
(price)
c87767d8-2801-44c4-b705-e853eaf0c148 (ticketID),  dfc75159-d2a6-455b-8c77-
44997db7bc6f (movieID),  58615812-eb08-497a-a9c4-96a7cccb20c9 (CinemaID),  2020-04-
01 (date),  02:55:00 (showTime),  03:02:00 (duration),  3D (cinemaType),  14
(price)
2cc35122-8bfa-4657-a8e3-41dbcc4023ff (ticketID),  33dadbed-4df2-4d2f-8660-
dc12e274d822 (movieID),  9de61cd1-61a8-4009-8bc8-b2b8ca1d0728 (CinemaID),  2020-04-
01 (date),  02:55:00 (showTime),  02:30:00 (duration),  Regular (cinemaType),  8
(price)
a623aa51-bb88-4e16-8d4d-5443cc28523b (ticketID),  0d505440-b4a5-4ccf-bd78-
a15ac095121e (movieID),  a6f0ff9a-705e-4165-9c41-71d231c7b987 (CinemaID),  2020-04-
01 (date),  02:55:00 (showTime),  02:20:00 (duration),  IMAX (cinemaType),  16
(price)
5b8ed3f4-4384-4753-89d6-72b48a22aff3 (ticketID),  e56fdb62-e63d-4f6f-9aa5-
b8f4c94faf0e (movieID),  031333a0-e455-421f-82fd-5694d72c9235 (CinemaID),  2020-04-
01 (date),  02:55:00 (showTime),  02:23:00 (duration),  Regular (cinemaType),  8
(price)
16641e3f-6a9f-4c62-a345-e9cf91fe719e (ticketID),  75f5c5ae-ff75-4e03-bf56-
85e50337c19b (movieID),  192c3cb1-fb86-48f2-a33e-222b83c65084 (CinemaID),  2022-11-
19 (date),  02:45:00 (showTime),  02:35:00 (duration),  Regular (cinemaType),  8
(price)

-----Database Queries-----
```

```
Canadian Directors:
CooglerRyan (fName-lName)
RussoJoe (fName-lName)
----------------
Movies by [WhedonJoss]:
Avengers: Age of Ultron (movieTitle)
The Avengers (movieTitle)
----------------
Actors where production is greater than [1000000]:
Downey-Robert (fName-lName)
Evans-Chris (fName-lName)
Ruffalo-Mark (fName-lName)
Hemsworth-Chris (fName-lName)
Johansson-Scarlett (fName-lName)
Test Actor Name (fName-lName)
----------------
Similar Director Names to pattern [Ru%]:
RussoAnthony (fName-lName)
RussoJoe (fName-lName)
----------------
```

Lab 10

lab10.XML

```xml
<MOVIES>

    <MOVIE>

        <TITLE> The Avengers </TITLE>

        <PRODUCTIONYEAR> 2012 </PRODUCTIONYEAR>

        <BUDGET> 220000000 </BUDGET>

        <DIRECTOR> Joss Whedon </DIRECTOR>

        <RATING> 7 </RATING>

        <DURATION> 2:23 </DURATION>
        <ACTORS>
            <ACTOR>
                <NAME> Robert Downey Jr. </NAME>
                <AGE> 57 </AGE>
                <COUNTRY> USA </COUNTRY>
            </ACTOR>
            <ACTOR>
                <NAME> Scarlett Johansson</NAME>
                <AGE> 37 </AGE>
                <COUNTRY> USA </COUNTRY>
            </ACTOR>
        </ACTORS>
    </MOVIE>
    <MOVIE>
        <TITLE> Warrior </TITLE>
        <PRODUCTIONYEAR> 2011 </PRODUCTIONYEAR>
        <BUDGET> 750000 </BUDGET>
        <DIRECTOR> Gavin O'Connor </DIRECTOR>
        <RATING> 8.6 </RATING>
        <DURATION> 224 </DURATION>
        <ACTORS>
            <ACTOR>
                <NAME> Tom Hardy </NAME>
                <AGE> 45 </AGE>

                <COUNTRY> United Kingdom </COUNTRY>

            </ACTOR>

            <ACTOR>

                <NAME> Joel Edgerton </NAME>

                <AGE> 48 </AGE>

                <COUNTRY> Australia </COUNTRY>

            </ACTOR>

        </ACTORS>

    </MOVIE>

    <MOVIE>

        <TITLE> Real Steel </TITLE>

        <PRODUCTIONYEAR> 2011 </PRODUCTIONYEAR>

        <BUDGET> 110000000 </BUDGET>

        <DIRECTOR> Shawn Levy </DIRECTOR>

        <RATING> 7 </RATING>

        <DURATION> 127 </DURATION>

        <ACTORS>

            <ACTOR>

                <NAME> Hugh Jackman </NAME>

                <AGE> 54 </AGE>

                <COUNTRY> Australia </COUNTRY>

            </ACTOR>
            <ACTOR>

                <NAME> Dakota Goyo </NAME>

                <AGE> 23 </AGE>

                <COUNTRY> Canada </COUNTRY>

            </ACTOR>

        </ACTORS>

    </MOVIE>

</MOVIES>
```

lab10 output

lab10.txt

```
Results of X Query 1.A):

Bud
2.50

Miller
3.00

Results of X Query 1.B):
Miller
3.00

Results of 3.A):
xquery version "3.0";
for $actor in MOVIES/MOVIE/ACTORS/ACTOR
where $actor/AGE < 35
order by $actor/AGE
return $actor

Dakota Goyo
23
Canada
```

```
Result of 3.B):
xquery version "3.0";
for $movie in MOVIES/MOVIE
where $movie/PRODUCTIONYEAR <= 2018 and $movie/PRODUCTIONYEAR >= 2016
order by $movie/TITLE descending
return $movie


         Warrior
         2011
         750000
         Gavin O'Connor
         8.6
         224


                 Tom Hardy
                 45
                 United Kingdom


                 Joel Edgerton
                 48
                 Australia



         The Avengers
         2012
         220000000
         Joss Whedon
         7
         2:23

                 Robert Downey Jr.
                 57
                 USA
```

```
        Scarlett Johansson
        37
        USA




Real Steel
2011
110000000
Shawn Levy
7
127


        Hugh Jackman
        54
        Australia


        Dakota Goyo
        23
        Canada
```