CSCI 3030 Project Part II

LO, Cheuk Him Calvin #100514352

**Lab 6 1.a. List all directors who were born in *Canada*.**

**Query:**

SELECT "Name", "Surename" FROM "Directors", "Locations"

WHERE "location\_id" = "Place\_of\_Birth\_id" AND

"Country" = 'Canada'

**Result:**

Name ; Surename

"Denis";"Villeneuve"

**Lab 6 1.b. Which movies are directed by *Steven Spielberg?***

**Query:**

SELECT "Title" FROM "Direct"

INNER JOIN "Movies" ON "Movies"."movie\_id" = "Direct"."movie\_id"

INNER JOIN "Directors" ON "Directors"."director\_id" = "Direct"."director\_id"

WHERE "Directors"."Name" = 'Steven' AND "Directors"."Surename" = 'Spielberg'

**Result:**

/\* Changed From David Lynch to Steven Spielberg \*/

Title

"Bridge of Spies"

**Lab 6 1.c.i. List all the actors who played in the movies for which cost of production was more than *$1M*. Compute cost in CAD.**

**Query:**

SELECT "Name","Surename", ("Budget" \* 1.311) AS "CAD" FROM "Act"

INNER JOIN "Movies" ON "Movies"."movie\_id" = "Act"."movie\_id"

INNER JOIN "Actors" ON "Actors"."actor\_id" = "Act"."actor\_id"

WHERE "Movies"."Budget" >= '1000000'

**Result:**

Name ; Surename ; CAD

"Michael";"Fassbender";$39,330,000.00

"Kate";"Winslet";$39,330,000.00

"Seth";"Rogen";$39,330,000.00

"Tom";"Hanks";$52,440,000.00

"Mark";"Rylance";$52,440,000.00

"Alan";"Alda";$52,440,000.00

"Matt";"Damon";$141,588,000.00

"Jessica";"Chastain";$141,588,000.00

"Kristen";"Wiig";$141,588,000.00

"Robert";"De Niro";$52,440,000.00

"Anne";"Hathaway";$52,440,000.00

"Rene";"Russo";$52,440,000.00

"Emily";"Blunt";$39,330,000.00

"Josh";"Brolin";$39,330,000.00

"Benicio";"Del Toro";$39,330,000.00

"Caleb";"Castille";$32,775,000.00

"Sean";"Astin";$32,775,000.00

"Jon";"Voight";$32,775,000.00

"Amy";"Poehler";$229,425,000.00

"Bill";"Hader";$229,425,000.00

"Lewis";"Black";$229,425,000.00

"Matthew";"McConaughey";$216,315,000.00

"Anne";"Hathaway";$216,315,000.00

"Jessica";"Chastain";$216,315,000.00

"Benedict";"Cumberbatch";$18,354,000.00

"Keira";"Knightley";$18,354,000.00

"Matthew";"Goode";$18,354,000.00

"Chris";"Pratt";$256,824,900.00

"Vin";"Diesel";$256,824,900.00

"Bradley";"Cooper";$256,824,900.00

**Lab 6 1.c.ii. List all the actors who played in the movies for which cost of production was more than *$1M*. Compute cost in JPY**

**Query:**

SELECT "Name","Surename", ("Budget" \* 121.165) AS "JPY" FROM "Act"

INNER JOIN "Movies" ON "Movies"."movie\_id" = "Act"."movie\_id"

INNER JOIN "Actors" ON "Actors"."actor\_id" = "Act"."actor\_id"

WHERE"Movies"."Budget" >= '1000000'

**Result:**

Name ; Surename ; JPY

"Michael";"Fassbender";$3,634,950,000.00

"Kate";"Winslet";$3,634,950,000.00

"Seth";"Rogen";$3,634,950,000.00

"Tom";"Hanks";$4,846,600,000.00

"Mark";"Rylance";$4,846,600,000.00

"Alan";"Alda";$4,846,600,000.00

"Matt";"Damon";$13,085,820,000.00

"Jessica";"Chastain";$13,085,820,000.00

"Kristen";"Wiig";$13,085,820,000.00

"Robert";"De Niro";$4,846,600,000.00

"Anne";"Hathaway";$4,846,600,000.00

"Rene";"Russo";$4,846,600,000.00

"Emily";"Blunt";$3,634,950,000.00

"Josh";"Brolin";$3,634,950,000.00

"Benicio";"Del Toro";$3,634,950,000.00

"Caleb";"Castille";$3,029,125,000.00

"Sean";"Astin";$3,029,125,000.00

"Jon";"Voight";$3,029,125,000.00

"Amy";"Poehler";$21,203,875,000.00

"Bill";"Hader";$21,203,875,000.00

"Lewis";"Black";$21,203,875,000.00

"Matthew";"McConaughey";$19,992,225,000.00

"Anne";"Hathaway";$19,992,225,000.00

"Jessica";"Chastain";$19,992,225,000.00

"Benedict";"Cumberbatch";$1,696,310,000.00

"Keira";"Knightley";$1,696,310,000.00

"Matthew";"Goode";$1,696,310,000.00

"Chris";"Pratt";$23,736,223,500.00

"Vin";"Diesel";$23,736,223,500.00

"Bradley";"Cooper";$23,736,223,500.00

**Lab 6 1.c.iii. List all the actors who played in the movies for which cost of production was more than *$1M*. Compute cost in RUB.**

**Query:**

SELECT "Name","Surename", ("Budget" \* 62.89) AS "RUB" FROM "Act"

INNER JOIN "Movies" ON "Movies"."movie\_id" = "Act"."movie\_id"

INNER JOIN "Actors" ON "Actors"."actor\_id" = "Act"."actor\_id"

WHERE "Movies"."Budget" >= '1000000'

**Result:**

Name ; Surename ; RUB

"Michael";"Fassbender";$1,886,700,000.00

"Kate";"Winslet";$1,886,700,000.00

"Seth";"Rogen";$1,886,700,000.00

"Tom";"Hanks";$2,515,600,000.00

"Mark";"Rylance";$2,515,600,000.00

"Alan";"Alda";$2,515,600,000.00

"Matt";"Damon";$6,792,120,000.00

"Jessica";"Chastain";$6,792,120,000.00

"Kristen";"Wiig";$6,792,120,000.00

"Robert";"De Niro";$2,515,600,000.00

"Anne";"Hathaway";$2,515,600,000.00

"Rene";"Russo";$2,515,600,000.00

"Emily";"Blunt";$1,886,700,000.00

"Josh";"Brolin";$1,886,700,000.00

"Benicio";"Del Toro";$1,886,700,000.00

"Caleb";"Castille";$1,572,250,000.00

"Sean";"Astin";$1,572,250,000.00

"Jon";"Voight";$1,572,250,000.00

"Amy";"Poehler";$11,005,750,000.00

"Bill";"Hader";$11,005,750,000.00

"Lewis";"Black";$11,005,750,000.00

"Matthew";"McConaughey";$10,376,850,000.00

"Anne";"Hathaway";$10,376,850,000.00

"Jessica";"Chastain";$10,376,850,000.00

"Benedict";"Cumberbatch";$880,460,000.00

"Keira";"Knightley";$880,460,000.00

"Matthew";"Goode";$880,460,000.00

"Chris";"Pratt";$12,320,151,000.00

"Vin";"Diesel";$12,320,151,000.00

"Bradley";"Cooper";$12,320,151,000.00

**Lab 6 1.c.iv. List all the actors who played in the movies for which cost of production was more than *$1M*. Compute cost in EUR.**

**Query:**

SELECT "Name","Surename", ("Budget" \* 0.91) AS "EUR" FROM "Act"

INNER JOIN "Movies" ON "Movies"."movie\_id" = "Act"."movie\_id"

INNER JOIN "Actors" ON "Actors"."actor\_id" = "Act"."actor\_id"

WHERE "Movies"."Budget" >= '1000000'

**Result:**

Name ; Surename ; EUR

"Michael";"Fassbender";$27,300,000.00

"Kate";"Winslet";$27,300,000.00

"Seth";"Rogen";$27,300,000.00

"Tom";"Hanks";$36,400,000.00

"Mark";"Rylance";$36,400,000.00

"Alan";"Alda";$36,400,000.00

"Matt";"Damon";$98,280,000.00

"Jessica";"Chastain";$98,280,000.00

"Kristen";"Wiig";$98,280,000.00

"Robert";"De Niro";$36,400,000.00

"Anne";"Hathaway";$36,400,000.00

"Rene";"Russo";$36,400,000.00

"Emily";"Blunt";$27,300,000.00

"Josh";"Brolin";$27,300,000.00

"Benicio";"Del Toro";$27,300,000.00

"Caleb";"Castille";$22,750,000.00

"Sean";"Astin";$22,750,000.00

"Jon";"Voight";$22,750,000.00

"Amy";"Poehler";$159,250,000.00

"Bill";"Hader";$159,250,000.00

"Lewis";"Black";$159,250,000.00

"Matthew";"McConaughey";$150,150,000.00

"Anne";"Hathaway";$150,150,000.00

"Jessica";"Chastain";$150,150,000.00

"Benedict";"Cumberbatch";$12,740,000.00

"Keira";"Knightley";$12,740,000.00

"Matthew";"Goode";$12,740,000.00

"Chris";"Pratt";$178,269,000.00

"Vin";"Diesel";$178,269,000.00

"Bradley";"Cooper";$178,269,000.00

**Lab 6 1.c.v. List all the actors who played in the movies for which cost of production was more than *$1M*. Compute cost in CHF.**

**Query:**

SELECT "Name","Surename", ("Budget" \* 0.99) AS "CHF" FROM "Act"

INNER JOIN "Movies" ON "Movies"."movie\_id" = "Act"."movie\_id"

INNER JOIN "Actors" ON "Actors"."actor\_id" = "Act"."actor\_id"

WHERE "Movies"."Budget" >= '1000000'

**Result:**

Name ; Surename ; CHF

"Michael";"Fassbender";$29,700,000.00

"Kate";"Winslet";$29,700,000.00

"Seth";"Rogen";$29,700,000.00

"Tom";"Hanks";$39,600,000.00

"Mark";"Rylance";$39,600,000.00

"Alan";"Alda";$39,600,000.00

"Matt";"Damon";$106,920,000.00

"Jessica";"Chastain";$106,920,000.00

"Kristen";"Wiig";$106,920,000.00

"Robert";"De Niro";$39,600,000.00

"Anne";"Hathaway";$39,600,000.00

"Rene";"Russo";$39,600,000.00

"Emily";"Blunt";$29,700,000.00

"Josh";"Brolin";$29,700,000.00

"Benicio";"Del Toro";$29,700,000.00

"Caleb";"Castille";$24,750,000.00

"Sean";"Astin";$24,750,000.00

"Jon";"Voight";$24,750,000.00

"Amy";"Poehler";$173,250,000.00

"Bill";"Hader";$173,250,000.00

"Lewis";"Black";$173,250,000.00

"Matthew";"McConaughey";$163,350,000.00

"Anne";"Hathaway";$163,350,000.00

"Jessica";"Chastain";$163,350,000.00

"Benedict";"Cumberbatch";$13,860,000.00

"Keira";"Knightley";$13,860,000.00

"Matthew";"Goode";$13,860,000.00

"Chris";"Pratt";$193,941,000.00

"Vin";"Diesel";$193,941,000.00

"Bradley";"Cooper";$193,941,000.00

**Lab6 1.d.** Using operator LIKE find all the directors whose surname start with letter “*A*” and “*D*”.

**Query:**

SELECT "Name", "Surename" FROM "Directors"

WHERE "Surename" LIKE 'A%' OR "Surename" LIKE 'D%'

**Result:**

Name;Surename

"Pete";"Docter"

"Ronnie";"Del Carmen"

**Lab6 1.e.** Find all the *comedy* movies in which there is a playing actor who is less than *40* (age).

**Query:**

SELECT "Title", "Name", "Surename" FROM "Act"

INNER JOIN "Movies" ON "Movies"."movie\_id" = "Act"."movie\_id"

INNER JOIN "Actors" ON "Actors"."actor\_id" = "Act"."actor\_id"

INNER JOIN "Genre" ON "Genre"."genre\_id" = "Movies"."genre\_id"

WHERE "Genre"."Type" = 'Comedy' AND

DATE\_PART('year', age("Date\_of\_Birth")) < 40

**Result:**

Title ; Name ; Surename

"The Intern";"Anne";"Hathaway"

**Lab6 1.f.** Find all the pairs of actors who have *blue* eye color. Produce pairs in alphabetic order, e.g., (Howard before Lynch) and do not produce pairs like (Lynch, Lynch)

**Query:**

SELECT A1."Surename", A2."Surename" FROM "Actors" A1, "Actors" A2

WHERE A1."Eye\_Color" = 'Blue' AND A2."Eye\_Color" = 'Blue' AND

A1."Surename" < A2."Surename"

**Result:**

Surename ; Surename

"Fassbender";"Winslet"

"Fassbender";"Hanks"

"Fassbender";"Rylance"

"Fassbender";"Wiig"

"Fassbender";"Russo"

"Fassbender";"Voight"

"Fassbender";"Poehler"

"Fassbender";"McConaughey"

"Fassbender";"Goode"

"Hanks";"Winslet"

"Hanks";"Rylance"

"Hanks";"Wiig"

"Hanks";"Russo"

"Hanks";"Voight"

"Hanks";"Poehler"

"Hanks";"McConaughey"

"Rylance";"Winslet"

"Rylance";"Wiig"

"Rylance";"Voight"

"Alda";"Fassbender"

"Alda";"Winslet"

"Alda";"Hanks"

"Alda";"Rylance"

"Alda";"Damon"

"Alda";"Wiig"

"Alda";"Russo"

"Alda";"Blunt"

"Alda";"Voight"

"Alda";"Poehler"

"Alda";"McConaughey"

"Alda";"Cumberbatch"

"Alda";"Goode"

"Alda";"Cooper"

"Damon";"Fassbender"

"Damon";"Winslet"

"Damon";"Hanks"

"Damon";"Rylance"

"Damon";"Wiig"

"Damon";"Russo"

"Damon";"Voight"

"Damon";"Poehler"

"Damon";"McConaughey"

"Damon";"Goode"

"Wiig";"Winslet"

"Russo";"Winslet"

"Russo";"Rylance"

"Russo";"Wiig"

"Russo";"Voight"

"Blunt";"Fassbender"

"Blunt";"Winslet"

"Blunt";"Hanks"

"Blunt";"Rylance"

"Blunt";"Damon"

"Blunt";"Wiig"

"Blunt";"Russo"

"Blunt";"Voight"

"Blunt";"Poehler"

"Blunt";"McConaughey"

"Blunt";"Cumberbatch"

"Blunt";"Goode"

"Blunt";"Cooper"

"Voight";"Winslet"

"Voight";"Wiig"

"Poehler";"Winslet"

"Poehler";"Rylance"

"Poehler";"Wiig"

"Poehler";"Russo"

"Poehler";"Voight"

"McConaughey";"Winslet"

"McConaughey";"Rylance"

"McConaughey";"Wiig"

"McConaughey";"Russo"

"McConaughey";"Voight"

"McConaughey";"Poehler"

"Cumberbatch";"Fassbender"

"Cumberbatch";"Winslet"

"Cumberbatch";"Hanks"

"Cumberbatch";"Rylance"

"Cumberbatch";"Damon"

"Cumberbatch";"Wiig"

"Cumberbatch";"Russo"

"Cumberbatch";"Voight"

"Cumberbatch";"Poehler"

"Cumberbatch";"McConaughey"

"Cumberbatch";"Goode"

"Goode";"Winslet"

"Goode";"Hanks"

"Goode";"Rylance"

"Goode";"Wiig"

"Goode";"Russo"

"Goode";"Voight"

"Goode";"Poehler"

"Goode";"McConaughey"

"Cooper";"Fassbender"

"Cooper";"Winslet"

"Cooper";"Hanks"

"Cooper";"Rylance"

"Cooper";"Damon"

"Cooper";"Wiig"

"Cooper";"Russo"

"Cooper";"Voight"

"Cooper";"Poehler"

"Cooper";"McConaughey"

"Cooper";"Cumberbatch"

"Cooper";"Goode"

**Lab7 1.a. From table Actors find the average age.**

**Query:**

SELECT AVG(DATE\_PART('year', age("Date\_of\_Birth"))) AS "Average\_Age" FROM "Actors"

**Result:**

Average\_Age

46.8888888888889

**Lab7 1.b. Find the number of different countries in which actors from the movie “*Bridge of Spies*” were born.**

**Query:**

SELECT count(DISTINCT "Country") FROM "Act"

INNER JOIN "Movies" ON "Movies"."movie\_id" = "Act"."movie\_id"

INNER JOIN "Actors" ON "Actors"."actor\_id" = "Act"."actor\_id"

INNER JOIN "Locations" ON "Locations"."location\_id" = "Actors"."Place\_of\_Birth\_id"

WHERE "Movies"."Title" = 'Bridge of Spies'

**Result:**

/\* Changed from Beautiful Mind to Bridge of Spies\*/

count

2

**Lab7 1.c. Find the number of actors with *green* eye color.**

**Query:**

SELECT count("actor\_id") FROM "Actors"

WHERE "Eye\_Color" = 'Green'

**Result:**

count

2

**Lab7 1.d. What is the number of movies in which *Anne Hathaway* was playing?**

**Query:**

SELECT COUNT("Movies"."movie\_id") FROM "Act"

INNER JOIN "Movies" ON "Movies"."movie\_id" = "Act"."movie\_id"

INNER JOIN "Actors" ON "Actors"."actor\_id" = "Act"."actor\_id"

WHERE "Actors"."Name" = 'Anne' AND "Actors"."Surename" = 'Hathaway'

**Result:**

/\* Changed from Brad Pitt to Anne Hathaway\*/

Count

2

**Lab7 1.e. Find the minimum, average and maximum budget for each type of movie.**

**Query:**

SELECT "Genre"."Type",

CAST(MIN(CAST("Movies"."Budget" AS DECIMAL)) AS MONEY) AS "Min",

CAST(AVG(CAST("Movies"."Budget" AS DECIMAL)) AS MONEY) AS "Avg",

CAST(MAX(CAST("Movies"."Budget" AS DECIMAL)) AS MONEY) AS "Max"

FROM "Movies"

INNER JOIN "Genre" ON "Movies"."genre\_id" = "Genre"."genre\_id"

GROUP BY "Genre"."Type"

**Result:**

Type;Min;Avg;Max

"Action";$30,000,000.00;$112,950,000.00;$195,900,000.00

"Comedy";$40,000,000.00;$40,000,000.00;$40,000,000.00

"Adventure";$108,000,000.00;$136,500,000.00;$165,000,000.00

"Drama";$25,000,000.00;$25,000,000.00;$25,000,000.00

"Animation";$175,000,000.00;$175,000,000.00;$175,000,000.00

"Biography";$14,000,000.00;$28,000,000.00;$40,000,000.00

**Lab7 1.f. Find the average rating for each movie which has director born in *London* or *blue* eye color.**

**Query:**

SELECT "Directors"."Name" , "Title" ,"Rating" FROM "Direct"

INNER JOIN "Movies" ON "Movies"."movie\_id" = "Direct"."movie\_id"

INNER JOIN "Directors" ON "Directors"."director\_id" = "Direct"."director\_id"

WHERE

("Directors"."Place\_of\_Birth\_id" IN (SELECT "location\_id"

FROM "Locations" WHERE

"Locations"."City" = 'London') OR

"Directors"."Eye\_Color" = 'Blue')

**Result:**

/\* Changed from Toronto to London \*/

Name;Title;Rating

"Danny";"Steve Jobs";7

"Steven";"Bridge of Spies";8

"Nancy";"The Intern";8

"Denis";"Sicario";8

"Andrew";"Woodlawn";7

"Jon";"Woodlawn";7

"Ronnie";"Inside Out";9

"Christopher";"Interstellar";9

"James";"Guardians of the Galaxy";8

**Lab7 1.g. Find all the movies that have actors born in at least two different countries.**

**Query:**

SELECT "Title" FROM "Act"

INNER JOIN "Movies" ON "Movies"."movie\_id" = "Act"."movie\_id"

INNER JOIN "Actors" ON "Actors"."actor\_id" = "Act"."actor\_id"

INNER JOIN "Locations" ON "Locations"."location\_id" = "Actors"."Place\_of\_Birth\_id"

GROUP BY "Title"

HAVING COUNT(DISTINCT("Country")) >= 2

**Result:**

Title

"Bridge of Spies"

"Sicario"

"Steve Jobs"

**Lab7 1.h. Write a query of your choice with a subquery and aggregation.**

**Query:**

/\*Find the number of movies for every actor from UK acted\*/

Select "Actors"."Name", "Actors"."Surename",

(SELECT COUNT("Act"."actor\_id") FROM "Act"

WHERE "Actors"."actor\_id" = "Act"."actor\_id")

FROM "Actors"

INNER JOIN "Locations" ON "Actors"."Place\_of\_Birth\_id" = "Locations"."location\_id"

WHERE "Country" = 'UK'

**Result:**

Name;Surename;count

"Kate";"Winslet";1

"Mark";"Rylance";1

"Emily";"Blunt";1

"Benedict";"Cumberbatch";1

"Keira";"Knightley";1

"Matthew";"Goode";1

**Lab 8&9 Java Code:**

/\* CSCI3030 Project JDBC

Cheuk Him Calvin, Lo

100514352

Compile: javac movie\_jdbc.java

Run: java cp postgresql-9.4-1205.jdbc4.jar; movie\_jdbc

All insert record default id is 40

\*/

import java.sql.\*;

import java.text.SimpleDateFormat;

public class movie\_jdbc {

public static void main(String[] argv) {

PreparedStatement pst = null;

ResultSet rs = null;

Connection conn = null;

String query = null;

System.out.println("----------------------------------MOVIE DB-------------------------------------");

try {

Class.forName("org.postgresql.Driver");

} catch (ClassNotFoundException e) {

System.out.println(e.getMessage());

return;

}

try {

String host = "/localhost:5432";

String db = "/movie\_db";

String username = "postgres";

String pw = "admin";

conn= DriverManager.getConnection("jdbc:postgresql:/" + host + db, username, pw );

} catch (SQLException e) {

System.out.println(e.getMessage());

return;

}

if (conn != null) {

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

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

} else {

System.out.println("Failed to connect...");

}

// Delete all id 40

try {

query = "DELETE FROM \"Act\" WHERE movie\_id = ?;";

int id = 40;

pst = conn.prepareStatement(query);

pst.setInt(1, id);

pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

try {

query = "DELETE FROM \"Direct\" WHERE movie\_id = ?;";

int id = 40;

pst = conn.prepareStatement(query);

pst.setInt(1, id);

pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

try {

query = "DELETE FROM \"Actors\" WHERE actor\_id = ?;";

int id = 40;

pst = conn.prepareStatement(query);

pst.setInt(1, id);

pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

try {

query = "DELETE FROM \"Genre\" WHERE genre\_id = ?;";

int id = 40;

pst = conn.prepareStatement(query);

pst.setInt(1, id);

pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

try {

query = "DELETE FROM \"Movies\" WHERE movie\_id = ?;";

int id = 40;

pst = conn.prepareStatement(query);

pst.setInt(1, id);

pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

try {

query = "DELETE FROM \"Locations\" WHERE location\_id = ?;";

int id = 40;

pst = conn.prepareStatement(query);

pst.setInt(1, id);

pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

try {

query = "DELETE FROM \"Directors\" WHERE director\_id = ?;";

int id = 40;

pst = conn.prepareStatement(query);

pst.setInt(1, id);

pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

// Insert Genre

try {

query = "INSERT INTO \"Genre\" (\"genre\_id\",\"Type\") VALUES (?,?)";

int genre\_id = 40;

String type ="SCI-FI";

pst = conn.prepareStatement(query);

pst.setInt(1, genre\_id);

pst.setString(2,type);

rs = pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

// Insert movie

try {

query = "INSERT INTO \"Movies\" (\"movie\_id\", \"Title\", \"ReleaseDate\", \"Rating\", \"Language\") VALUES (?,?,?,?,?)";

int movie\_id = 40;

String title = "Captain America: Civil War";

java.sql.Date releaseDate= null;

try {

java.util.Date releaseDate\_1= new SimpleDateFormat("yyyy-MM-dd").parse("2016-05-06");

releaseDate = new java.sql.Date(releaseDate\_1.getTime());

} catch(Exception e) {

System.out.println("Error in converting date");

}

int rating = 0;

String language = "English";

pst = conn.prepareStatement(query);

pst.setInt(1, movie\_id);

pst.setString(2, title);

pst.setDate(3, releaseDate);

pst.setInt(4, rating);

pst.setString(5,language);

pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

// Add Location

try {

query = "INSERT INTO \"Locations\"(\"location\_id\",\"City\", \"Country\") VALUES(?,?,?);";

int location\_id = 40;

String city = "Kingston upon Thames";

String country = "UK";

pst = conn.prepareStatement(query);

pst.setInt(1, location\_id);

pst.setString(2,city);

pst.setString(3,country);

pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

//Add Director

try {

query = "INSERT INTO \"Directors\" (\"director\_id\", \"Name\", \"Surename\") VALUES(?,?,?)";

int director\_id = 40;

String name = "Anthony";

String surename = "Russo";

pst = conn.prepareStatement(query);

pst.setInt(1,director\_id);

pst.setString(2, name);

pst.setString(3, surename);

pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

// Add actor

try{

query = "INSERT INTO \"Actors\" (\"actor\_id\", \"Name\", \"Surename\", \"Date\_of\_Birth\", \"Eye\_Color\") VALUES(?,?,?,?,?)";

int actor\_id = 40;

String name = "Chris";

String surename = "Evans";

java.sql.Date date\_of\_birth= null;

try {

java.util.Date date\_of\_birth\_1= new SimpleDateFormat("yyyy-MM-dd").parse("1981-06-13");

date\_of\_birth = new java.sql.Date(date\_of\_birth\_1.getTime());

} catch(Exception e) {

System.out.println("Error in converting date");

}

String eye\_color = "Blue";

pst = conn.prepareStatement(query);

pst.setInt(1, actor\_id);

pst.setString(2, name);

pst.setString(3, surename);

pst.setDate(4,date\_of\_birth);

pst.setString(5,eye\_color);

rs = pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

// Add Act

try {

query = "INSERT INTO \"Act\" (\"movie\_id\",\"actor\_id\") VALUES (?,?);";

int movie\_id = 40;

int actor\_id = 40;

pst = conn.prepareStatement(query);

pst.setInt(1,movie\_id);

pst.setInt(2,actor\_id);

pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

// Add Direct

try {

query = "INSERT INTO \"Direct\" (\"movie\_id\",\"director\_id\") VALUES (?,?);";

int movie\_id = 40;

int director\_id = 40;

pst = conn.prepareStatement(query);

pst.setInt(1,movie\_id);

pst.setInt(2,director\_id);

pst.executeQuery();

} catch (SQLException e) {

System.out.println(e.getMessage());

}

// Lab 6 B

try {

query = "SELECT \"Title\" FROM \"Direct\" INNER JOIN \"Movies\" ON \"Movies\".\"movie\_id\" = \"Direct\".\"movie\_id\" INNER JOIN \"Directors\" ON \"Directors\".\"director\_id\" = \"Direct\".\"director\_id\" WHERE \"Directors\".\"Name\" = ? AND \"Directors\".\"Surename\" = ?";

String name = "Steven";

String surename = "Spielberg";

pst = conn.prepareStatement(query);

pst.setString(1, name);

pst.setString(2, surename);

rs = pst.executeQuery();

System.out.println("\n");

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

System.out.println("Lab6 1B : Which movies are directed by " + name + " " + surename + "?");

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

System.out.println("Title");

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

while (rs.next()) {

String rs\_title = rs.getString("Title");

System.out.println(rs\_title);

}

} catch (SQLException e) {

System.out.println(e.getMessage());

}

// Lab 6 D

try {

query = "SELECT \"Name\", \"Surename\" FROM \"Directors\" WHERE \"Surename\" LIKE ? OR \"Surename\" LIKE ?";

String arg1 = "A%";

String arg2 = "D%";

pst = conn.prepareStatement(query);

pst.setString(1, arg1);

pst.setString(2, arg2);

rs = pst.executeQuery();

System.out.println("\n");

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

System.out.println("Lab6 1D : Find all the directors whose surname start with letter \"" + arg1 + "\" and \"" + arg2 + "\".");

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

System.out.println("Name\tSurename");

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

while(rs.next()) {

String rs\_name = rs.getString("Name");

String rs\_surename = rs.getString("Surename");

System.out.println(rs\_name +"\t" + rs\_surename);

}

} catch (SQLException e) {

System.out.println(e.getMessage());

}

// Lab 7 C

try {

query = "SELECT count(\"actor\_id\") FROM \"Actors\" WHERE \"Eye\_Color\" = ?";

String eye\_color = "Green";

pst = conn.prepareStatement(query);

pst.setString(1,eye\_color);

rs = pst.executeQuery();

System.out.println("\n");

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

System.out.println("Lab7 1C : Find the number of actors with " + eye\_color +" eye color. ");

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

System.out.println("Count");

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

while(rs.next()) {

int rs\_count = rs.getInt("count");

System.out.println(rs\_count);

}

} catch (SQLException e) {

System.out.println(e.getMessage());

}

// Lab 7 F

try {

query = "SELECT \"Directors\".\"Name\" , \"Title\" ,\"Rating\" FROM \"Direct\" INNER JOIN \"Movies\" ON \"Movies\".\"movie\_id\" = \"Direct\".\"movie\_id\" INNER JOIN \"Directors\" ON \"Directors\".\"director\_id\" = \"Direct\".\"director\_id\" WHERE (\"Directors\".\"Place\_of\_Birth\_id\" IN (SELECT \"location\_id\" FROM \"Locations\" WHERE \"Locations\".\"City\" = ?) OR \"Directors\".\"Eye\_Color\" = ?)";

String city = "London";

String eye\_color = "Blue";

pst = conn.prepareStatement(query);

pst.setString(1, city);

pst.setString(2, eye\_color);

rs = pst.executeQuery();

System.out.println("\n");

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

System.out.println("Lab7 1F : Find the average rating for each movie which has director born in " + city + " or " + eye\_color + " eye color. ");

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

System.out.println("Name\t\t\tTitle\t\t\t\tRating");

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

while (rs.next()) {

String rs\_name = rs.getString("Name");

String rs\_title = rs.getString("Title");

int rs\_rating = rs.getInt("Rating");

System.out.println(rs\_name + "\t\t\t" + rs\_title + "\t\t\t" + rs\_rating);

}

} catch (SQLException e) {

System.out.println(e.getMessage());

}

try {

if (rs != null) { // In

rs.close();

}

if (pst != null) { // Reverse

pst.close();

}

if (conn != null) { // Order

conn.close();

}

} catch (SQLException e) { // Again, we MUST catch the errors!

System.out.println(e.getMessage());

}

}

}