PROJECT 2 REPORT

ONYEDIKACHI KALU

100557846

LAB 6

1. a.
```
SELECT "Director"."Name”
FROM "Director","Location"
    WHERE "Location"."Country" = 'Canada' AND "Location"."LocationID" =
"Director"."LocationID";
```

b.
```
SELECT "Movie"."Name"
FROM "Movie","Director","Director Movie Relationship"
WHERE "Movie"."MovieID" = "Director Movie Relationship"."MovieID"
    AND "Director"."DirectorID" = "Director Movie Relationship"."DirectorID"
    AND "Director"."Name" = 'Louis Leterrier';
```

c.
```
SELECT "Actor"."Name", "Movie"."Budget" as "USD",
    ("Movie"."Budget")*1.27 as "CAD", ("Movie"."Budget")*113.75 as "JPY",
    ("Movie"."Budget")*59.25 as "RUB", ("Movie"."Budget")*0.86 as "EUR",
("Movie"."Budget")*1.00 as "CHF"
FROM "Movie","Actor","Actor movie Relationship"
WHERE  "Movie"."MovieID" = "Actor movie Relationship"."MovieID"
    AND "Actor"."ActorID" = "Actor movie Relationship"."ActorID"
    AND "Movie"."Budget" >= 1000000;
```

d.
```
SELECT "Name"
FROM "Director"
WHERE "Name" LIKE 'J%' OR "Name" LIKE 'S%';
```

e.
```
SELECT "Movie"."Name"
FROM "Movie", "Genre", "Movie Category"
WHERE "Movie"."MovieID" = "Movie Category"."MovieID"
    AND "Movie Category"."GenreID" = "Genre"."GenreID"
AND "Genre"."Name" = 'Comedy';
```

f.
```
SELECT "b1"."Name", "b2"."Name"
```

FROM "Actor" "b1", "Actor" "b2"

WHERE "b1"."Eye Color" = 'Blue' AND "b2"."Eye Color" = 'Blue' AND "b1"."Name" < "b2"."Name";

2 a. SELECT AVG("Show time"."Price") as "Average price of ticket"

FROM "Show time"

b. SELECT "Location"."Country", "Actor"."Name"

FROM "Location", "Actor", "Actor movie Relationship", "Movie"

WHERE "Movie"."Name" = 'Iron Man' AND "Movie"."MovieID" = "Actor movie Relationship"."MovieID"

AND "Actor movie Relationship"."ActorID" = "Actor"."ActorID"

AND "Location"."LocationID" = "Actor"."LocationID"

c. SELECT COUNT("Eye Color")as "Number of people with brown eyes"

FROM "Actor"

WHERE "Actor"."Eye Color" = 'Brown'

d. SELECT COUNT("Movie"."Name")

FROM "Actor","Movie", "Actor movie Relationship"

WHERE "Actor"."ActorID" = "Actor movie Relationship"."ActorID"

AND "Movie"."MovieID" = "Actor movie Relationship"."MovieID"

AND "Actor"."Name" = 'Chris Evans'

e. SELECT "Genre"."Name", MIN("Movie"."Budget"),MAX("Movie"."Budget"), AVG("Movie"."Budget")

FROM "Movie", "Genre", "Movie Category"

WHERE "Movie"."MovieID" = "Movie Category"."MovieID"

AND "Genre"."GenreID" = "Movie Category"."GenreID"

GROUP BY "Genre"."Name"

f. SELECT AVG("Review"."Critic Rating") as "Average Critic rating"

FROM "Movie", "Review", "Actor", "Location" , "Actor movie Relationship"

```
WHERE "Actor"."LocationID" = "Location"."LocationID" AND
        "Location"."Town" = 'Toronto' AND
"Movie"."MovieID" = "Actor movie Relationship"."MovieID"
AND "Actor"."ActorID" = "Actor movie Relationship"."ActorID"
AND "Movie"."MovieID" = "Review"."MovieID"
OR ("Movie"."MovieID" = "Actor movie Relationship"."MovieID"
AND "Actor"."ActorID" = "Actor movie Relationship"."ActorID"
AND "Actor"."Eye Color" = 'Blue')
```

g.
```
SELECT "Movie"."Name"
FROM "Movie", "Genre", "Movie Category"
WHERE "Movie"."MovieID" = "Movie Category"."MovieID"
        AND "Genre"."GenreID" = "Movie Category"."GenreID"
        GROUP BY "Movie"."Name"
        HAVING COUNT("Movie"."Name") >= 2
```

h.
```
SELECT "Movie"."Name"
FROM "Movie","Show time"
WHERE "Movie"."MovieID" = "Show time"."MovieID"
        AND "Show time"."Price" <= ALL (SELECT "Price"
                                        FROM "Show time");
```

RESULT

1. a. 'Brad Bird'

   'Shane Black'

   b. 'Iron Man'

   c. 'Robert Downey Jr','140000000','177800000.00','15925000000.00','8295000000.00','120400000.00','140000000.00'

'Robert Downey Jr','220000000','279400000.00','25025000000.00','13035000000.00','189200000.00','220000000.00'

'Chris Hemsworth','220000000','279400000.00','25025000000.00','13035000000.00','189200000.00','220000000.00'

' Chris Evans','220000000','279400000.00','25025000000.00','13035000000.00','189200000.00','220000000.00'

'Samuel L. Jackson','220000000','279400000.00','25025000000.00','13035000000.00','189200000.00','220000000.00'

'Chris Hemsworth','140000000','177800000.00','15925000000.00','8295000000.00','120400000.00','140000000.00'

'Chris Evans','140000000','177800000.00','15925000000.00','8295000000.00','120400000.00','140000000.00'

'Samuel L. Jackson','150000000','190500000.00','17062500000.00','8887500000.00','129000000.00','150000000.00'

'Meryl Streep','150000000','190500000.00','17062500000.00','8887500000.00','129000000.00','150000000.00'

'Leonardo DiCaprio','200000000','254000000.00','22750000000.00','11850000000.00','172000000.00','200000000.00'

'Channing Tatum','150000000','190500000.00','17062500000.00','8887500000.00','129000000.00','150000000.00'

'Jonah Hill','140000000','177800000.00','15925000000.00','8295000000.00','120400000.00','140000000.00'

'Ryan Gosling','220000000','279400000.00','25025000000.00','13035000000.00','189200000.00','220000000.00'

'Ryan Reynolds','220000000','279400000.00','25025000000.00','13035000000.00','189200000.00','220000000.00'

d. 'Jon Favreau'

'Joss Whedon'

'Steve McQueen'

'Shane Black'

e. 'Shutter Island'

'Get Out'

f. 'Chris Hemsworth','Meryl Streep'

'Channing Tatum','Chris Hemsworth'

'Channing Tatum','Meryl Streep'

2. a. '12.7000000000000000'

b. 'United States','Robert Downey Jr'

'United States','Chris Hemsworth'

'United States','Chris Evans'

c. '5'

d. '2'

e. 'Horror','17100000','160000000','88550000.000000000000'

'Action','140000000','200000000','165000000.00000000'

'Thriller','29000000','150000000','89500000.000000000000'

'Family','30000000','92000000','61000000.000000000000'

'Comedy','4500000','80000000','42250000.000000000000'

'Adventure','140000000','200000000','159500000.00000000'

'Drama','120000000','120000000','120000000.000000000000'

'Sci-Fi','140000000','232300000','190383333.33333333'

'Animation','52000000','55000000','53500000.000000000000'

f. '78.9571938168846611'

g. 'The Incredible Hulk'

'Iron Man 2'

'Iron Man'

h. 'Iron Man'

LAB 7 – 8

```
package main;


import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Time;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;


public class Main {


static DateFormat sdf = new SimpleDateFormat("hh:mm:ss");


	public static Time getTime(String time) throws ParseException {
		long date = sdf.parse(time).getTime();
		Time t = new Time(date);
		return t;
	}


	//For queries that return 2 columns
	public static void printResult(ResultSet rs) throws SQLException {
		System.out.println("Query Returned returned: ");
		while (rs.next())
```

```
            {
               System.out.println(rs.getString(1));
            }
            rs.close();
            System.out.println("\n");
}



public static void main(String[] argv)
 {

    Connection conn;
    ResultSet rs = null;
    try
    {
     Class.forName("org.postgresql.Driver");
     String dbURL = "jdbc:postgresql://localhost:5432/LAB 5/ PROJECT";
     String user = "postgres";
     String pass = "Arsenal17";

     conn = DriverManager.getConnection(dbURL, user, pass);
     Statement stmt = conn.createStatement();
     PreparedStatement p = null;
     String sql;

     //1
                    sql = "INSERT INTO \"Movie\" " +
            "VALUES (22, 'Django Unchained', 'December 25, 2012', 100000000)";
                   stmt.executeUpdate(sql);
```

```
p = conn.prepareStatement("INSERT INTO \"Movie\" VALUES "
                    + "(?, ?, ?, ?)");
p.setInt(1, 23);
p.setString(2, "Deadpool");
p.setString(3, "February 12, 2016");
p.setInt(4, 58000000);
p.executeUpdate();

//2
sql = "INSERT INTO \"Writer\""
                    + "VALUES (11, 'Quentin Tarantino')";
stmt.executeUpdate(sql);

p = conn.prepareStatement("INSERT INTO \"Writer\" VALUES "
                    + "(?, ?)");
p.setInt(1, 12);
p.setString(2, "Rhett Reese");
p.executeUpdate();



//3
sql = "INSERT INTO \"Genre\""
                    + "VALUES (11, 'Fiction', 'All made up')";
stmt.executeUpdate(sql);

p = conn.prepareStatement("INSERT INTO \"Genre\" VALUES "
                    + "(?, ?, ?)");
p.setInt(1, 12);
```

```
p.setString(2, "Anime");
p.setString(3, " Japanese term for hand-drawn or computer animation. ");
p.executeUpdate();



//4
sql = "INSERT INTO \"Producer\""
                    + "VALUES (11, 'Stacey Sher')";
stmt.executeUpdate(sql);

p = conn.prepareStatement("INSERT INTO \"Producer\" VALUES "
                    + "(?, ?)");
p.setInt(1, 12);
p.setString(2, "Simon Kinberg");
p.executeUpdate();



//5
 sql = "INSERT INTO \"Location\""
                    + "VALUES (11, 1 ,'Canada', 'Ontario', 'Oshawa')";
stmt.executeUpdate(sql);

p = conn.prepareStatement("INSERT INTO \"Location\" VALUES "
                    + "(?, ?, ?, ?,?)");
p.setInt(1, 12);
p.setInt(2, 2);
p.setString(3, "Canada");
p.setString(4, "Ontario");
p.setString(5, "Guelph");
```

```
        p.executeUpdate();



        //6
        sql = "INSERT INTO \"Actor\""
                            + "VALUES (11, 5, 'Jamie Foxx', 'Black')";
        stmt.executeUpdate(sql);

        p = conn.prepareStatement("INSERT INTO \"Actor\" VALUES "
                            + "(?, ?, ?, ?)");
        p.setInt(1, 12);
        p.setInt(2, 12);
        p.setString(3, "T. J. Miller");
        p.setString(4, "Brown");
        p.executeUpdate();



        //7
    sql = "INSERT INTO \"Director\""
                            + "VALUES (11, 6, 'Quentin Tarantino')";
        stmt.executeUpdate(sql);

        p = conn.prepareStatement("INSERT INTO \"Director\" VALUES "
                            + "(?, ?, ?)");
        p.setInt(1, 12);
        p.setInt(2, 12);
        p.setString(3, "Tim Miller");
        p.executeUpdate();
```

```
//8
sql = "INSERT INTO \"Review\""
                    + "VALUES (22, 86, 90)";
stmt.executeUpdate(sql);

p = conn.prepareStatement("INSERT INTO \"Review\" VALUES "
                    + "(?, ?, ?)");
p.setInt(1, 23);
p.setInt(2, 88);
p.setInt(3, 90);
p.executeUpdate();


//9
sql = "INSERT INTO        \"Famous Quote\""
                    + "VALUES ('I like the way you die boy.', 22)";
stmt.executeUpdate(sql);

p = conn.prepareStatement("INSERT INTO \"Famous Quote\" VALUES "
                    + "(?, ?)");
p.setString(2, "Daddy Needs To Express Some Rage.");
p.setInt(1, 23);
p.executeUpdate();

//10

sql = "INSERT INTO        \"Awards\""
```

```
                    + "VALUES (11, 21, 'Golden Globe Award for Best Screenplay -
Motion Picture 2013')";
                stmt.executeUpdate(sql);

                p = conn.prepareStatement("INSERT INTO \"Awards\" VALUES "
                        + "(?, ?, ?)");
                p.setInt(1, 12);
                p.setInt(2, 23);
                p.setString(3, "MTV Movie Award for Best Fight 2016");
                p.executeUpdate();


                //11
                sql = "INSERT INTO      \"Cinema\""
                        + "VALUES (11, 'Cinema Town')";
                stmt.executeUpdate(sql);
                p = conn.prepareStatement("INSERT INTO \"Cinema\" VALUES "
                        + "(?, ?)");
                p.setInt(1, 12);
                p.setString(2, "Theatre of dreams");
                p.executeUpdate();


                //12
                sql = "INSERT INTO      \"Show time\""
                        + "VALUES (22, 11, 12)";
                stmt.executeUpdate(sql);
                p = conn.prepareStatement("INSERT INTO \"Show time\" VALUES "
                        + "(?, ?, ?)");
```

```
p.setInt(1, 12);
p.setInt(2, 12);
p.setInt(3, 15);
p.executeUpdate();



//13
sql = "INSERT INTO        \"Movie Category\""
                    + "VALUES (22, 6)";
stmt.executeUpdate(sql);

p = conn.prepareStatement("INSERT INTO \"Movie Category\" VALUES "
                    + "(?, ?)");
p.setInt(1, 23);
p.setInt(2, 6);
p.executeUpdate();

//14
 sql = "INSERT INTO        \"Writer Movie Relationship\""
                    + "VALUES (22, 11)";
stmt.executeUpdate(sql);
p = conn.prepareStatement("INSERT INTO \"Writer Movie Relationship\"
VALUES "
                    + "(?, ?)");
p.setInt(1, 23);
p.setInt(2, 12);
p.executeUpdate();
```

```
                    //15
                    sql = "INSERT INTO        \"Movie Producer Relationship\""
                                + "VALUES (22, 11)";
                    stmt.executeUpdate(sql);
                    p = conn.prepareStatement("INSERT INTO \"Movie Producer Relationship\"
VALUES "
                                + "(?, ?)");
                    p.setInt(1, 23);
                    p.setInt(2, 12);
                    p.executeUpdate();



                    //16
                    sql = "INSERT INTO        \"Director Movie Relationship\""
                                + "VALUES (11, 22)";
                    stmt.executeUpdate(sql);
                    p = conn.prepareStatement("INSERT INTO \"Director Movie Relationship\"
VALUES "
                                + "(?, ?)");
                    p.setInt(1, 12);
                    p.setInt(2, 23);
                    p.executeUpdate();



                    //17
                    sql = "INSERT INTO        \"Actor movie Relationship\""
                                + "VALUES (11, 22)";
                    stmt.executeUpdate(sql);
                    p = conn.prepareStatement("INSERT INTO \"Actor movie Relationship\" VALUES
"
```

```
                                        + "(?, ?)");
                        p.setInt(1, 12);
                        p.setInt(2, 23);
                        p.executeUpdate();



                //SELECT statements
                //********* 1 ************
                //sql = "SELECT \"Director\".\"Name\"" +
                //              "                       FROM \"Director\", \"Location\"" +
                //              "                       WHERE \"Location\".\"Country\" = 'Canada' AND
\"Location\".\"LocationID\" = \"Director\".\"LocationID\" ";
                        //rs = stmt.executeQuery(sql);
                        //printResult(rs);


                p = conn.prepareStatement("SELECT \"Movie\".\"Name\"" +
                        "FROM \"Movie\",\"Director\",\"Director Movie Relationship\"" +
                        "WHERE \"Movie\".\"MovieID\" = \"Director Movie Relationship\".\"MovieID\"
" +
                                "AND \"Director\".\"DirectorID\" = \"Director Movie
Relationship\".\"DirectorID\" "+
                                "AND \"Director\".\"Name\" = ?");
                        p.setString(1, "Louis Leterrier");
                        rs = p.executeQuery();
                        printResult(rs);



                //********* 2 **********
                sql = "SELECT \"Actor\".\"Name\", \"Movie\".\"Budget\" as \"USD\", " +
```

```
                    "(\"Movie\".\"Budget\")* 1.27 as \"CAD\", (\"Movie\".\"Budget\")*
113.75 as \"JPY\"," +
                    "(\"Movie\".\"Budget\")* 59.25 as \"RUB\", (\"Movie\".\"Budget\")*
0.86 as \"EUR\"," +
                "(\"Movie\".\"Budget\")* 1.00 as \"CHF\"" +
                "FROM \"Movie\",\"Actor\",\"Actor movie Relationship\" "+
                "WHERE  \"Movie\".\"MovieID\" = \"Actor movie Relationship\".\"MovieID\" " +
                    "AND \"Actor\".\"ActorID\" = \"Actor movie Relationship\".\"ActorID\""
+
                    "AND \"Movie\".\"Budget\" >= 1000000";
                    rs = stmt.executeQuery(sql);
            printResult(rs);


        p = conn.prepareStatement(" SELECT \"Movie\".\"Name\"\r\n" +
                "                   FROM \"Movie\", \"Genre\", \"Movie Category\"\r\n" +
                "                   WHERE \"Movie\".\"MovieID\" = \"Movie
Category\".\"MovieID\"\r\n" +
                "                           AND \"Movie Category\".\"GenreID\" =
\"Genre\".\"GenreID\" \r\n" +
                "                   AND \"Genre\".\"Name\" = ?;");
                p.setString(1, "Comedy");
                rs = p.executeQuery();
                printResult(rs);



        //******** 3 **********
        sql = "SELECT \"Name\"\r\n" +
                "                   FROM \"Director\"\r\n" +
                "                   WHERE \"Name\" LIKE 'J%' OR \"Name\" LIKE 'S%'";
```

```
                    rs = stmt.executeQuery(sql);
            printResult(rs);

        p = conn.prepareStatement("SELECT COUNT(\"Location\".\"Country\")\n" +
                    "                FROM \"Location\", \"Actor\", \"Actor movie Relationship\", \"Movie\"\r\n" +
                    "                WHERE \"Movie\".\"Name\" = ? AND \"Movie\".\"MovieID\" = \"Actor movie Relationship\".\"MovieID\"\r\n" +
                    "                AND \"Actor movie Relationship\".\"ActorID\" = \"Actor\".\"ActorID\"\r\n" +
                    "        AND \"Location\".\"LocationID\" = \"Actor\".\"LocationID\"");
                    p.setString(1, "Iron Man");
                    rs = p.executeQuery();
                    printResult(rs);



        //********  4 *******
        sql = "SELECT AVG(\"Show time\".\"Price\") as \"Average price of ticket\"\r\n" +
                    "                FROM \"Show time\"";
                    rs = stmt.executeQuery(sql);
            printResult(rs);

        p = conn.prepareStatement("SELECT COUNT(\"Eye Color\")as \"Number of people with brown eyes\"\r\n" +
                    "                FROM \"Actor\"\r\n" +
                    "                WHERE \"Actor\".\"Eye Color\" = ?");
                    p.setString(1, "Brown");
                    rs = p.executeQuery();
                    printResult(rs);
```

```
            } catch (Exception ex)
            {
                ex.printStackTrace();
                    }
            }
}
```

LAB 9 – 10

DTD

```
<!ELEMENT movies (movie+, actor+, director+)>
<!ELEMENT movie (name, release_year, length, genre, budget, rating, gross?)>
  <!ATTLIST movie movieID ID #REQUIRED>
  <!ATTLIST movie playedIn IDREF #REQUIRED>
  <!ATTLIST movie directedBy IDREF #REQUIRED>
<!ELEMENT actor (first_name, surname, birth_location, eye_colour, age)>
  <!ATTLIST actor actorID ID #REQUIRED>
  <!ATTLIST actor movieID IDREF #REQUIRED>
<!ELEMENT director (first_name, surname, birth_year, birth_location, eye_colour)>
  <!ATTLIST director directorID ID #REQUIRED>
  <!ATTLIST director movie IDREF #IMPLIED>
<!ELEMENT movieID (#PCDATA)>
<!ELEMENT companyID (#PCDATA)>
<!ELEMENT quoteID (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT release_year (#PCDATA)>
<!ELEMENT length (#PCDATA)>
<!ELEMENT genre (#PCDATA)>
<!ELEMENT budget (#PCDATA)>
<!ELEMENT gross (#PCDATA)>
<!ELEMENT rating (#PCDATA)>
<!ELEMENT actorID (#PCDATA)>
<!ELEMENT first_name (#PCDATA)>
<!ELEMENT surname (#PCDATA)>
```

```
<!ELEMENT birth_location (#PCDATA)>
<!ELEMENT eye_colour (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT directorID (#PCDATA)>
<!ELEMENT birth_year (#PCDATA)>
```

XML

```
<?xml version="1.0" ?>
<!DOCTYPE movies SYSTEM "lab9_10.dtd">
<movies>
  <movie movieID = "1" directorID = "2" actorID = "4">
    <name>Chucky 6</name>
    <release_year>2017</release_year>
    <length>01:23:45</length>
    <genre>Horror</genre>
    <budget>1000000</budget>
    <rating>3</rating>
    <gross>2500000</gross>
  </movie>
  <movie movieID = "2" directorID = "12 7" actorID = "9">
```

```
  <name>Andrometer</name>
  <release_year>2012</release_year>
  <length>01:33:45</length>
  <genre>Horror</genre>
  <budget>2000000</budget>
  <rating>4</rating>
</movie>
<actor actorID = "6", movieID = "2">
  <first_name>Yahya</first_name>
  <surname>Imran</surname>
  <birth_location>India</birth_location>
  <eye_colour>brown</eye_colour>
  <age>22</age>
</actor>
<actor actorID = "2", movieID = "7">
  <first_name>Mustafa</first_name>
  <surname>Ali</surname>
  <birth_location>Indonesia</birth_location>
  <eye_colour>brown</eye_colour>
  <age>21</age>
</actor>
<director directorID = "4" movieID = "3">
  <first_name>Lupe</first_name>
  <surname>Fiasco</surname>
```

```
    <birth_year>1962</birth_year>
    <birth_location>USA</birth_location>
    <eye_colour>brown</eye_colour>
  </director>
  <director directorID = "1">
    <first_name>Adman</first_name>
    <surname>Imran</surname>
    <birth_year>1985</birth_year>
    <birth_location>Pakistan</birth_location>
    <eye_colour>brown</eye_colour>
  </director>
</movies>
```

XSD

```
<?xml version="1.0" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="movies">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="movie">
          <xs:key name = "movieKey">
            <xs:selector xpath = "movie" />
            <xs:field xpath = "@movieID" />
```

```
</xs:key>
<xs:keyref name = "directorRef" refers = "directorKey">
  <xs:selector xpath = "movie" />
  <xs:field xpath = "@directedID" />
</xs:keyref>
<xs:keyref name = "actorRef" refers = "actorKey">
  <xs:selector xpath = "movie" />
  <xs:field xpath = "@actorID" />
</xs:keyref>
<xs:complexType>
  <xs:attribute name="movieID" type="xs:string"
    use = "required" />
  <xs:attribute name="actorID" type="xs:string"
    use = "required" />
  <xs:attribute name="directorID" type="xs:string"
    use = "required" />
  <xs:sequence>
    <xs:element name="name" type="xs:string"
      minOccurs = ”1” maxOccurs = ”1” />
    <xs:element name="release_year" type="xs:int"
      minOccurs = ”1” maxOccurs = ”1” />
    <xs:element name="length" type="xs:time"
      minOccurs = ”1” maxOccurs = ”1” />
    <xs:element name="genre" type="xs:string"
      minOccurs = ”1” maxOccurs = ”1” />
    <xs:element name="budget" type="xs:int"
      minOccurs = ”1” maxOccurs = ”1” />
```

```
      <xs:element name="rating" type="xs:int"
        minOccurs = ”1” maxOccurs = ”1” />
      <xs:element name="gross" type="xs:int"
        minOccurs = ”0” maxOccurs = ”1” />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="actor">
  <xs:key name = "actorKey">
    <xs:selector xpath = "actor" />
    <xs:field xpath = "@actorID" />
  </xs:key>
  <xs:keyref name = "movieRef" refers = "movieKey">
    <xs:selector xpath = "actor" />
    <xs:field xpath = "@movieID" />
  </xs:keyref>
  <xs:complexType>
    <xs:attribute name="actorID" type="xs:string"
      use = "required" />
    <xs:attribute name="movieID" type="xs:string"
      use = "required" />
    <xs:sequence>
      <xs:element name="first_name" type="xs:string"
        minOccurs = ”1” maxOccurs = ”1” />
      <xs:element name="surname" type="xs:string"
        minOccurs = ”1” maxOccurs = ”1” />
      <xs:element name="birth_location" type="xs:string"
```

```
        minOccurs = ”1” maxOccurs = ”1” />
      <xs:element name="eye_colour" type="xs:string"
        minOccurs = ”1” maxOccurs = ”1” />
      <xs:element name="age" type="xs:int"
        minOccurs = ”1” maxOccurs = ”1” />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="director">
  <xs:key name = "directorKey">
    <xs:selector xpath = "director" />
    <xs:field xpath = "@directedID" />
  </xs:key>
  <xs:keyref name = "movieRef" refers = "movieKey">
    <xs:selector xpath = "director" />
    <xs:field xpath = "@movieID" />
  </xs:keyref>
  <xs:complexType>
    <xs:attribute name="directorID" type="xs:string"
      use = "required" />
    <xs:attribute name="movieID" type="xs:string"
      use = "optional" />
    <xs:sequence>
      <xs:element name="first_name" type="xs:string"
        minOccurs = ”1” maxOccurs = ”1” />
      <xs:element name="surname" type="xs:string"
        minOccurs = ”1” maxOccurs = ”1” />
```

```
            <xs:element name="birth_year" type="xs:int"
              minOccurs = ”1” maxOccurs = ”1” />
            <xs:element name="birth_location" type="xs:string"
              minOccurs = ”1” maxOccurs = ”1” />
            <xs:element name="eye_colour" type="xs:string"
              minOccurs = ”1” maxOccurs = ”1” />
            <xs:element name="age" type="xs:int"
              minOccurs = ”1” maxOccurs = ”1” />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```