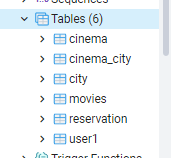
Report

Name: Aruzhan Amanova

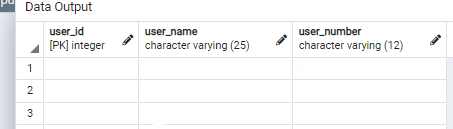
Group: IT-2004

Project name: Movie Booking System

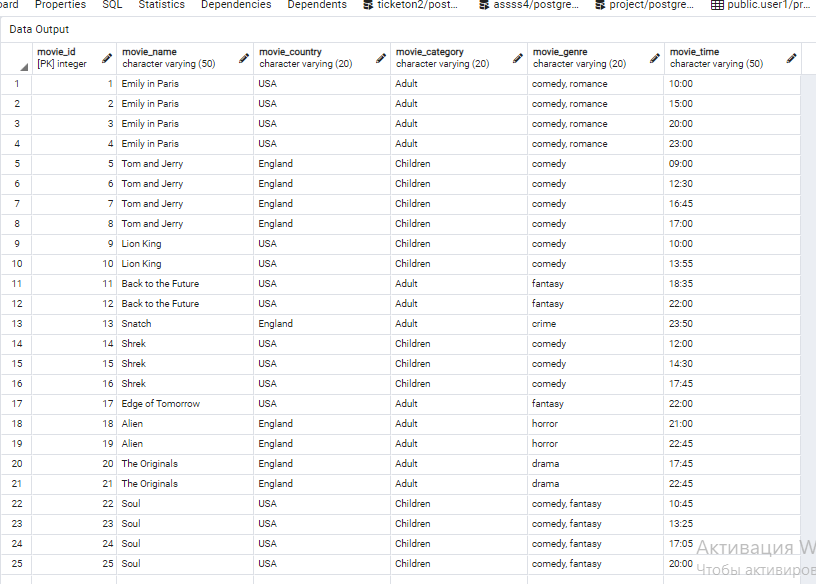
The topic of my project is "Movie Booking System". It is a console application that helps the user to select a movie and book it. Initially, I created a database with 6 tables(user1, movies, reservation, cinema, city, cinema\_city), which is connected to IntelliJia IDEA.

  
The purpose of the tables (for what purpose they were created):

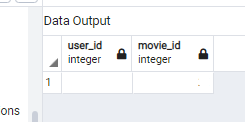
1. The **"user1" table** stores information about users. The data that the user enters in the Console is redirected to this table and stored there.



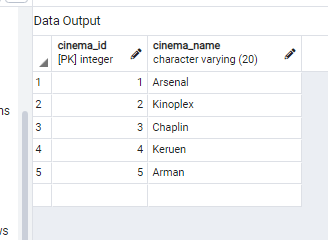
1. The **“movies” table** stores all the information about the available movies. This table is needed to access the list of movies and select the desired movie from there. This table was previously filled with 25 rows with different movies, which are then displayed to the user.



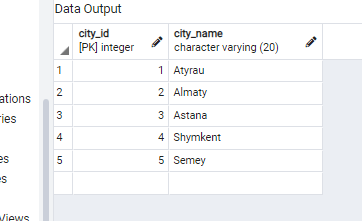
1. The **“reservation” table** is needed to store information about the reservation, more precisely, information about which user has reserved which movie.



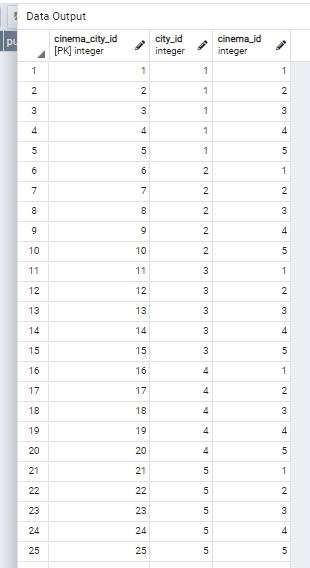
1. The **“cinema” table** is needed to store information about cinemas.



1. The **“city” table** is needed to store information about cities.



1. The **“cinema\_city” table** was created to link cities with movie theaters. This table shows which movie theater in which city the user wants to book.



**Main class**. As the name itself says, this class is the most important and special because here is the main part of the code; this is where the entire project is focused on.

package com.company;  
import java.sql.\*;  
import java.util.Scanner;  
public class Main {  
  
 public static void main(String[] args) {  
 Connection connection = null;  
 Statement stmt = null;  
 ResultSet rs = null;  
 Scanner in = new Scanner(System.*in*);  
 try { //try statement is used to handle exceptions

// Here we load the driver’s class file into memory at the runtime  
 Class.*forName*("org.postgresql.Driver");

//Establish the connection  
 connection = DriverManager.*getConnection*("jdbc:postgresql://localhost:5432/project", "postgres", "0000");  
 if (connection != null){ //this if statement is used to check whether the connection is established or not  
 stmt = connection.createStatement();  
 //System.out.println("connected"); //  
 }  
 else {  
 System.*out*.println("no connection");  
 }  
 }catch (Exception e) {  
 System.*out*.println(e);  
 }  
 `while(true){  
 try {  
 *Menu*(); //calling the “menu” function that I’ve created below  
 int san = in.nextInt(); //accepts the number of the menu’s option  
 in.nextLine(); //going to the next line  
 if (san == 1){ //the body of the first option – “To book”  
 System.*out*.println("Enter your name: ");  
 String name = in.nextLine();  
 User user = new User();  
 user.setName(name);  
 System.*out*.println("Enter your phone number: ");  
 String phone\_number = in.nextLine();  
 user.setPhone\_number(phone\_number);  
 PreparedStatement prst = connection.prepareStatement("insert into user1(user\_name, user\_number) values(?,?) ");  
 prst.setString(1,name);  
 prst.setString(2,phone\_number);  
 prst.executeUpdate();  
 System.*out*.println("Choose a category:");  
 System.*out*.println("1: Children");  
 System.*out*.println("2: Adult");  
 san = in.nextInt(); //accepts the number(the option category)  
 if (san == 1){ //if the entered number equals to the 1, which means Children category, then the following statements will run  
 rs = stmt.executeQuery("select \* from movies where movie\_category='Children'"); //shows all the movies with the Children category  
 while(rs.next()){  
 System.*out*.println("Movie\_id:" + rs.getInt("movie\_id") + '\t' +" Name: " + rs.getString("movie\_name") + '\t' + " Country: " + rs.getString("movie\_country") + '\t' + " Category: " + rs.getString("movie\_category") + '\t' + " Genre: " + rs.getString("movie\_genre") + '\t' + " Time: " + rs.getString("movie\_time")); //outputs the booked information  
 }  
 int currentID = 0;  
 rs = stmt.executeQuery("select user\_id from user1 where user1.user\_id = user\_id");  
 while (rs.next()) {  
 currentID = rs.getInt("user\_id");  
 }  
 int currentIDMovie = 0;  
 rs = stmt.executeQuery("select movie\_id from movies where movies.movie\_id = movie\_id");  
 while (rs.next()) {  
 currentIDMovie = rs.getInt("movie\_id");  
 }  
 while (true) {  
 int selected = in.nextInt();  
 System.*out*.println("Input '-1' as soon as you finish booking."); //message for the user, if the user finished booking, then he/she should enter “-1” to finish the booking procedure  
 if (selected == -1) { //after entering the number “-1”, the user will choose the city and cinema  
 System.*out*.println("Choose a city: ");  
 rs = stmt.executeQuery("select \* from city"); //query for outputting all cities  
 while(rs.next()){  
 System.*out*.println("City\_ID: " + rs.getString("city\_id") + '\t' + " Name of the city: " + rs.getString("city\_name")); //outputting the message  
 }  
 int chooseCity = in.nextInt();  
 String chooseCity1 = "";  
 in.nextLine();  
 if(chooseCity == 1){ //the meaning of the digits  
 chooseCity1 = "Atyrau";  
 }  
 else if(chooseCity == 2){  
 chooseCity1 = "Almaty";  
 }  
 else if(chooseCity == 3){  
 chooseCity1 = "Astana";  
 }  
 else if(chooseCity == 4){  
 chooseCity1 = "Shymkent";  
 }  
 else if(chooseCity == 5){  
 chooseCity1 = "Semey";  
 }  
 System.*out*.println("Choose a cinema");  
 rs = stmt.executeQuery("select \* from cinema"); //query for outputting all cinemas of the chosen city  
 while(rs.next()){  
 System.*out*.println("Cinema\_ID: " + rs.getString("cinema\_id") + '\t' + " Name of the cinema: " + rs.getString("cinema\_name")); //outputting the chosen cinema  
 }  
 int chooseCinema = in.nextInt();  
 in.nextLine();  
 String chooseCinema1 = "";  
 if(chooseCinema == 1){ //the meaning of numbers  
 chooseCinema1 = "Arsenal";  
 }  
 else if(chooseCinema == 2){  
 chooseCinema1 = "Kinoplex";  
 }  
 else if(chooseCinema == 3){  
 chooseCinema1 = "Chaplin";  
 }  
 else if(chooseCinema == 4){  
 chooseCinema1 = "Keruen";  
 }  
 else if(chooseCinema == 5){  
 chooseCinema1 = "Arman";  
 }  
 PreparedStatement prst2 = connection.prepareStatement("");  
 System.*out*.println("The reservation confirmed!" );  
 System.*out*.println("The chosen city is " + chooseCity1);  
 System.*out*.println("The chosen cinema is " + chooseCinema1);  
 System.*out*.println("Your unique ID is " + currentID + ". Please use it to check your reservation."); //the final output of the booking procedure  
 break;  
 }  
 PreparedStatement prst1 = connection.prepareStatement("insert into reservation(user\_id, movie\_id) values (?, ?)");  
 prst1.setInt(1, currentID);  
 prst1.setInt(2, selected);  
 prst1.executeUpdate();  
 }  
 }  
 else if(san == 2){ //the body of the second option – Adult category  
 rs = stmt.executeQuery("select \* from movies where movie\_category='Adult'"); //gets information about movies with the Adult category  
 while(rs.next()){  
 System.*out*.println("Movie\_id:" + rs.getInt("movie\_id") + '\t' +" Name: " + rs.getString("movie\_name") + '\t' + " Country: " + rs.getString("movie\_country") + '\t' + " Category: " + rs.getString("movie\_category") + '\t' + " Genre: " + rs.getString("movie\_genre") + '\t' + " Time: " + rs.getString("movie\_time"));//outputs this information  
 }  
 int currentID = 0;  
 rs = stmt.executeQuery("select user\_id from user1 where user1.user\_id = user\_id");  
 while (rs.next()) {  
 currentID = rs.getInt("user\_id");  
 }  
 int currentIDMovie = 0;  
 rs = stmt.executeQuery("select movie\_id from movies where movies.movie\_id = movie\_id");  
 while (rs.next()) {  
 currentIDMovie = rs.getInt("movie\_id");  
 }  
 while (true) {   
 int selected = in.nextInt();//message for the user, if the user finished booking, then he/she should enter “-1” to finish the booking procedure  
 if (selected == -1) { // after entering the number “-1”, the user will choose the city and cinema  
 System.*out*.println("Choose a city: ");  
 rs = stmt.executeQuery("select \* from city");  
 while(rs.next()){  
 System.*out*.println("City\_ID: " + rs.getString("city\_id") + '\t' + " Name of the city: " + rs.getString("city\_name"));  
 }  
 int chooseCity = in.nextInt();  
 String chooseCity1 = "";  
 in.nextLine();  
 if(chooseCity == 1){  
 chooseCity1 = "Atyrau";  
 }  
 else if(chooseCity == 2){  
 chooseCity1 = "Almaty";  
 }  
 else if(chooseCity == 3){  
 chooseCity1 = "Astana";  
 }  
 else if(chooseCity == 4){  
 chooseCity1 = "Shymkent";  
 }  
 else if(chooseCity == 5){  
 chooseCity1 = "Semey";  
 }  
 System.*out*.println("Choose a cinema");  
 rs = stmt.executeQuery("select \* from cinema");  
 while(rs.next()){  
 System.*out*.println("Cinema\_ID: " + rs.getString("cinema\_id") + '\t' + " Name of the cinema: " + rs.getString("cinema\_name"));  
 }  
 int chooseCinema = in.nextInt();  
 in.nextLine();  
 String chooseCinema1 = "";  
 if(chooseCinema == 1){  
 chooseCinema1 = "Arsenal";  
 }  
 else if(chooseCinema == 2){  
 chooseCinema1 = "Kinoplex";  
 }  
 else if(chooseCinema == 3){  
 chooseCinema1 = "Chaplin";  
 }  
 else if(chooseCinema == 4){  
 chooseCinema1 = "Keruen";  
 }  
 else if(chooseCinema == 5){  
 chooseCinema1 = "Arman";  
 }  
 PreparedStatement prst2 = connection.prepareStatement("");  
 System.*out*.println("The reservation confirmed!" );  
 System.*out*.println("The chosen city is " + chooseCity1);  
 System.*out*.println("The chosen cinema is " + chooseCinema1);  
 System.*out*.println("Your unique ID is " + currentID + ". Please use it to check your reservation.");  
 break;  
 }  
 PreparedStatement prst1 = connection.prepareStatement("insert into reservation(user\_id, movie\_id) values (?, ?)");  
 prst1.setInt(1, currentID);  
 prst1.setInt(2, selected);  
 prst1.executeUpdate();  
 }  
 }  
 }else if(san == 2){//the body of the second menu’s option – “To check reservation”  
 System.*out*.println("Enter your ID: ");  
 int enteredID = in.nextInt(); //the user enters the unique ID that was given at the end of the booking procedure  
 ResultSet rss = stmt.executeQuery("select movie\_name, movie\_time from movies where movies.movie\_id = (select movie\_id from reservation where reservation.user\_id = '"+enteredID+"')"); //gets information from movies table with the similar id number that was entered by the user above  
 while(rss.next()){  
 System.*out*.println("Movie name: " + rss.getString("movie\_name") + '\t' + " Date and Time: " + rss.getString("movie\_time")); //outputs the message about the movie  
 }  
 }  
 else if(san == 3){ //the body for the third menu’s option, which is exit  
 System.*exit*(0); //the program stops  
 }  
 }catch (Exception e){  
 System.*out*.println(e);  
 }  
 }  
 }  
 public static void Menu(){//creating a Menu  
 System.*out*.println("1: To book");  
 System.*out*.println("2: To check reservation");  
 System.*out*.println("3: Exit");  
 }  
  
}`

**Movie class**. This class is created for initializing fields and adding information about Movie. In this class, I’ve created getter and setter functions to get and set the values;   
1) name is used to initialize the name of the movie; through this field the user and the program can enter and get names of the movies;  
2) time is used to show the time of the movie;   
3) country and genre fields are created to store information about the movie’s country and genre

package com.company;  
  
public class Movie {  
 private String name; //name of the movie  
 private String time; //time of the movie  
 private String country; //the country where it was shooted  
 private String genre; //the genre of the movie  
 public Movie(String name, String time, String country, String genre){ //the first constructor with all arguments  
 setName(name); //setting the name of the movie  
 setTime(time); //setting the time of the movie  
 setCountry(country); //setting the country of the movie  
 setGenre(genre); //setting the genre of the movie  
 }  
 public Movie(){ //empty constructor  
 }  
 public void override(){   
 System.*out*.println("Parentclass");  
 }  
  
 public String getName() { //getter  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getTime() {  
 return time;  
 }  
  
 public void setTime(String time) {  
 this.time = time;  
 }  
  
 public String getCountry() {  
 return country;  
 }  
  
 public void setCountry(String country) {  
 this.country = country;  
 }  
  
 public String getGenre() {  
 return genre;  
 }  
  
 public void setGenre(String genre) {  
 this.genre = genre;  
 }  
}

**User class.** This class is created for initializing fields and adding information about the user. In this class, I’ve created getter and setter functions to get and set the values;   
1) name is used to initialize the user’s name;  
2) phone\_number initialize the user’s phone number

package com.company;  
  
public class User {  
 private String name; //the user’s name  
 private String phone\_number; //the user’s phone number  
 public User(){ //empty constructor  
  
 }  
 public User(String name, String phone\_number){ //constructor with all created fields in arguments; this constructor is used to set the fields that were created above  
 setName(name); //setting the user name  
 setPhone\_number(phone\_number); //setting the user phone number  
 }  
 public String getName() { //getter function  
 return name;  
 }  
 public void setName(String name) { //setter function  
 this.name = name;  
 }  
 public String getPhone\_number() { //getter function  
 return phone\_number;  
 }  
 public void setPhone\_number(String phone\_number) {//setter function  
 this.phone\_number = phone\_number;  
 }  
  
}

**ChildrenMovie class.** This class is the subclass of the Movie class; the reason of creating this class is children category. The user should choose the category of the film during the booking procedure, so for that reason the ChildrenMovie class is created.

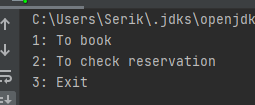
package com.company;  
  
public class ChildrenMovie extends Movie {  
 public ChildrenMovie(String name, String duration, String country, String genre){  
 super(name, duration, country, genre);  
 }  
 @Override  
 public void override(){  
 System.*out*.println("subclass1");  
 }  
}

**AdultMovie class.**This class was created with the same reason as it was with the ChildrenMovie class. The only difference is that this class is for the adult category.

package com.company;  
  
public class AdultMovie extends Movie{  
 public AdultMovie(String name, String duration, String country, String genre){  
 super(name, duration, country, genre);  
 }  
 @Override  
 public void override() {  
 System.*out*.println("subclass2");  
 }  
}

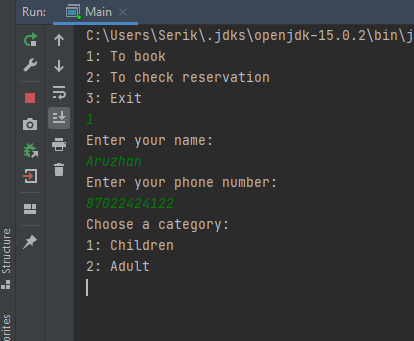
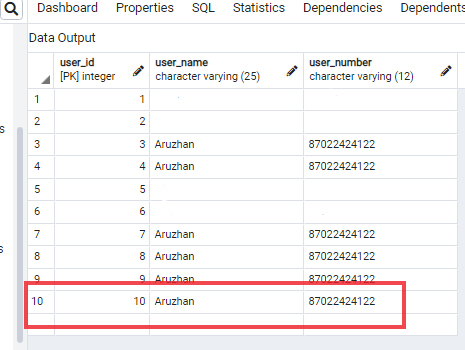
Execution:

1. The first thing that will appear is the “Menu”.

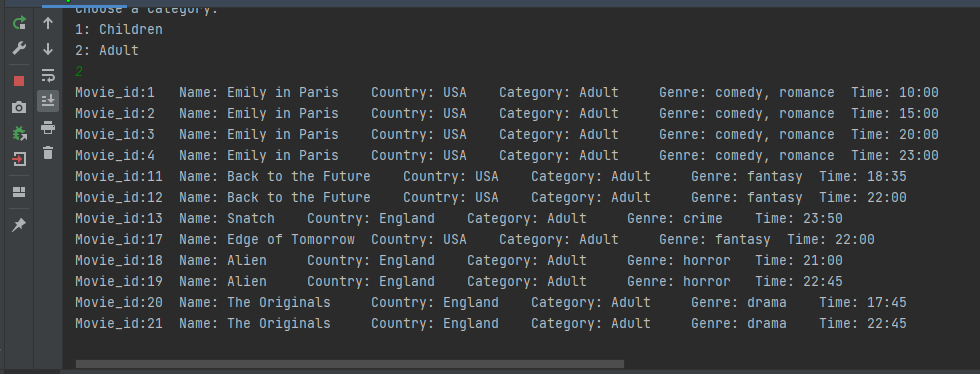


If the user wants to book the movie, then:

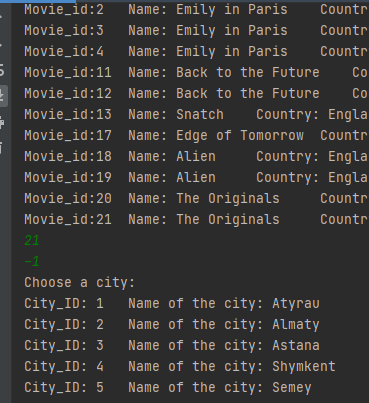
1. First, the user should enter a “1” and input his/her name and phone number. Then the program will ask to choose a category of the movie.

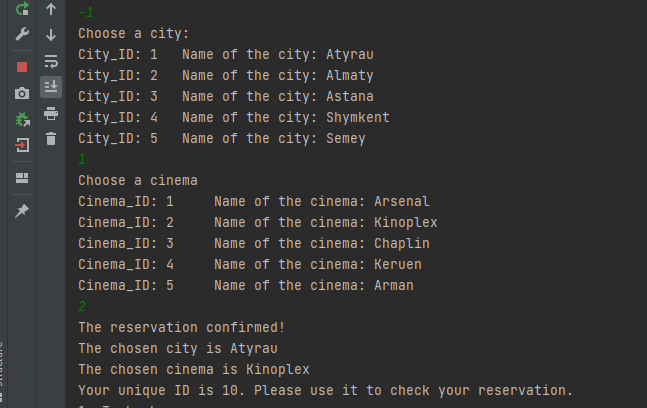
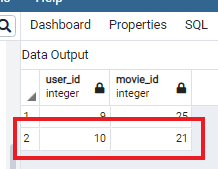
  


1. After choosing a category, there will appear a list of movies according to the chosen movie. In this try, I chose Adult category and following movies are outputted.



1. The next step is to choose a movie. After selecting the movie, the user should write the id of the movie and enter “-1” if he/she finished booking procedure. After this, the user should choose the city and the cinema where he/she would like to watch the chosen movie.

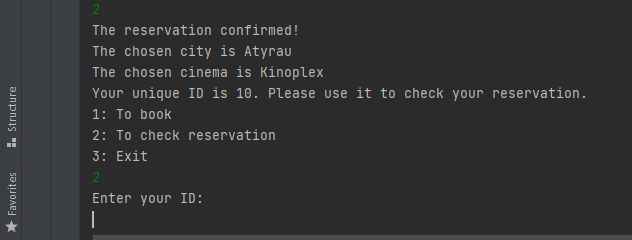


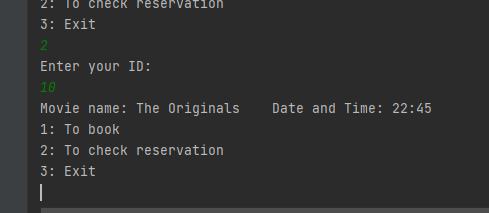
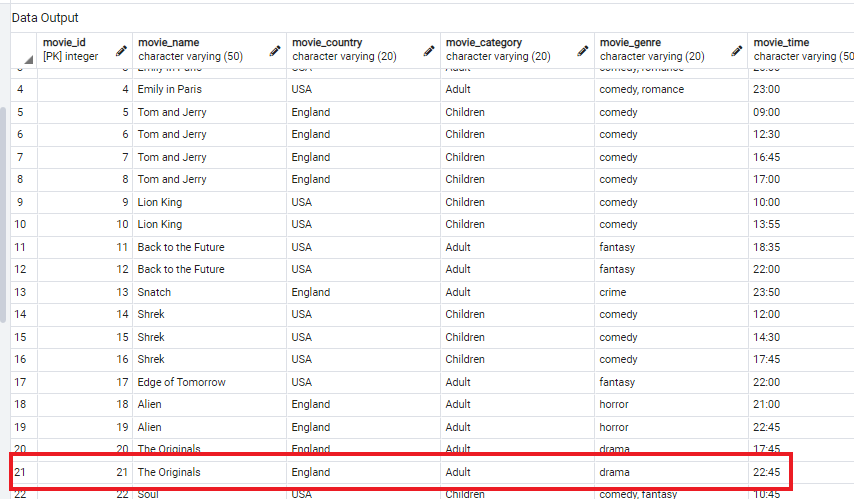
1. And at the end of the booking procedure, the program will show the message about confirmation of the reservation and information about the city, cinema and unique ID of the user.

The next option of the menu is checking the reservation:

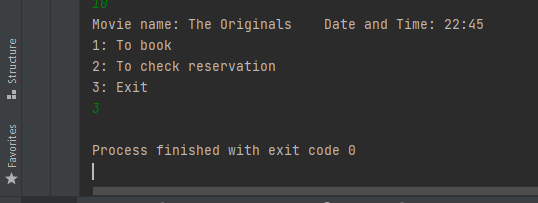
1. To check the reservation, the user first should enter 2, which means “To check the reservation” in menu.

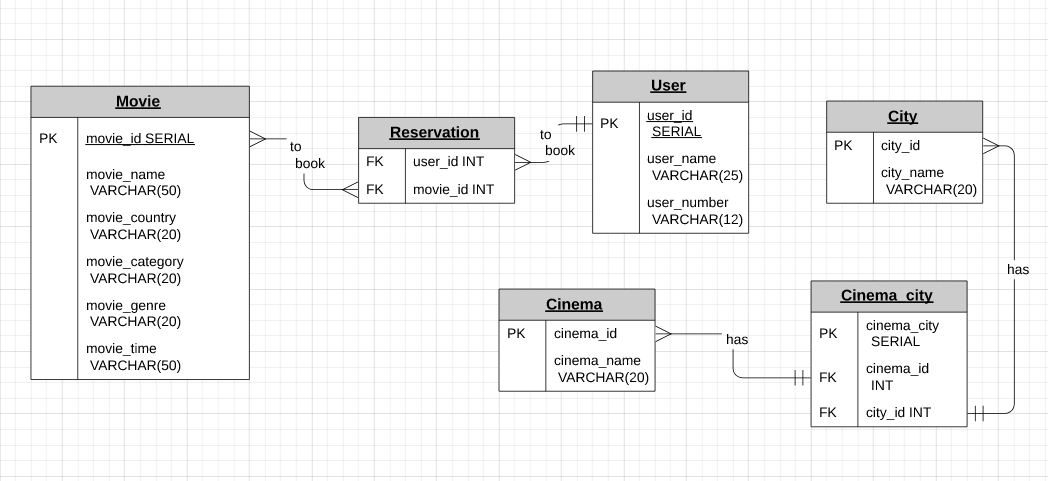


1. Then, the program will ask to user to input the unique ID that was given during the booking procedure. After inputting the ID, the program will output the information about booked film.

1. As you can see on the picture, the information that was outputted is the same that we chose during the booking procedure.
2. The last option of the menu is “Exit”, which finishes the running of the program.

  
  
**Entity-Relationship Diagram:**



Business rule:

1. Movie – to book – reservation
2. User – to book – reservation
3. Cinema – has – cinema\_city
4. City – has – cinema\_city

**Movie(optional) many to many (optional) Reservation**

One movie can be reserved several times;  
Multiple movies can be booked per reservation;

**User(required) one to many (optional) Reservation**

One user can make many reservations;  
The same reservation cannot be made by two different people;

**Cinema(optional) many to one(required) Cinema\_city  
City(optional) many to one(required) Cinema\_city**

One cinema\_city can have multiple data with the same cinema, because the same cinema is also available in other cities. By the same logic, a one cinema\_city can have multiple data with the same city, but with different cinemas.

**Github link:** https://github.com/Aruzhan02/final