Oracle PL/SQL Capstone Exam 28/08/2024

Oracle Code: -

set serveroutput on;

-- create table movies

create table movies (

movie\_id number generated by default as identity primary key,

title varchar2(100) not null,

director varchar2(100),

release\_year number(4),

genre varchar2(50)

);

-- create table customers

create table customers (

customer\_id number generated by default as identity primary key,

name varchar2(100) not null,

email varchar2(100) unique,

phone varchar2(15)

);

-- Insert movies data

insert into movies (title, director, release\_year, genre) values ('Gharat Ganpati', 'Navjyot Bandiwadekar', 2024, 'Marathi Film');

insert into movies (title, director, release\_year, genre) values ('Dharmaveer', 'Pravin Tarde', 2022, 'Marathi Film');

insert into movies (title, director, release\_year, genre) values ('Swatantrya Veer Savarkar', 'Randeep Hooda', 2024, 'Bollywood');

insert into movies (title, director, release\_year, genre) values ('Khel Khel Mein', 'Mudassar Aziz', 2024, 'Bollywood');

insert into movies (title, director, release\_year, genre) values ('3 Idiots', 'Rajkumar Hirani', 2009, 'Bollywood');

insert into movies (title, director, release\_year, genre) values ('Housefull 2', 'Sajid Khan', 2012, 'Bollywood');

select \* from movies;

-- update one row from movies table

update movies set release\_year=2024 where title = 'Gharat Ganpati';

select \* from movies;

-- delete one row from movies table

delete from movies where movie\_id=6;

select \* from movies;

-- Insert customers data

insert into customers (name, email, phone) values ('Geeta Joshi', 'geeta.joshi@example.com', '9765432108');

insert into customers (name, email, phone) values ('Anita Desai', 'anita.desai@example.com', '8765432109');

insert into customers (name, email, phone) values ('Nisha Kapoor', 'nisha.kapoor@example.com', '6543210987');

insert into customers (name, email, phone) values ('Rahul Khanna', 'rahul.khanna@example.com', '9876543210');

insert into customers (name, email, phone) values ('Siddharth Patel', 'siddharth.patel@example.com', '7654321098');

insert into customers (name, email, phone) values ('Rajesh Kumar', 'rajesh.kumar@example.com', '5432109876');

select \* from customers;

-- create movie view

create view movie\_view as

select movie\_id, title, director, release\_year, genre

from movies;

-- create procedure for add movie

create or replace procedure add\_movie (

p\_title in varchar2,

p\_director in varchar2,

p\_release\_year in number,

p\_genre in varchar2

) as

begin

insert into movies (title, director, release\_year, genre)

values (p\_title, p\_director, p\_release\_year, p\_genre);

commit;

end;

/

-- create declare block for add movie

declare

p\_title varchar2(100) := '&input\_title';

p\_director varchar2(100) := '&input\_director';

p\_release\_year number := 2024;

p\_genre varchar2(50) := '&input\_genre';

begin

insert into movies (title, director, release\_year, genre)

values (p\_title, p\_director, p\_release\_year, p\_genre);

commit;

dbms\_output.put\_line('movie inserted successfully: ' || p\_title);

end;

/

select \* from movies;

-- create procedure for read movie

create or replace function read\_movie (

p\_movie\_id in number

) return sys\_refcursor as

v\_cursor sys\_refcursor;

begin

open v\_cursor for

select title, director, release\_year, genre

from movies

where movie\_id = p\_movie\_id;

return v\_cursor;

end;

/

-- create declare block for read movie

declare

p\_movie\_id number := 2;

v\_title varchar2(100);

v\_director varchar2(100);

v\_release\_year number;

v\_genre varchar2(50);

begin

select title, director, release\_year, genre

into v\_title, v\_director, v\_release\_year, v\_genre

from movies

where movie\_id = p\_movie\_id;

dbms\_output.put\_line('title: ' || v\_title || ', director: ' || v\_director || ', release year: ' || v\_release\_year || ', genre: ' || v\_genre);

exception

when no\_data\_found then

dbms\_output.put\_line('no movie found with id ' || p\_movie\_id);

end;

/

select \* from movies;

-- create procedure for update movie

create or replace procedure update\_movie (

p\_movie\_id in number,

p\_title in varchar2,

p\_director in varchar2,

p\_release\_year in number,

p\_genre in varchar2

) as

begin

update movies

set title = p\_title,

director = p\_director,

release\_year = p\_release\_year,

genre = p\_genre

where movie\_id = p\_movie\_id;

commit;

end;

/

-- create declare block for update movie

declare

p\_movie\_id number := '&input\_id';

p\_title varchar2(100) := '&input\_updated\_title';

p\_director varchar2(100) := '&input\_updated\_director';

p\_release\_year number := '&input\_updated\_year';

p\_genre varchar2(50) := '&input\_updated\_genre';

begin

update movies

set title = p\_title,

director = p\_director,

release\_year = p\_release\_year,

genre = p\_genre

where movie\_id = p\_movie\_id;

commit;

dbms\_output.put\_line('movie updated successfully.');

exception

when no\_data\_found then

dbms\_output.put\_line('no movie found with id ' || p\_movie\_id);

end;

/

select \* from movies;

-- create procedure for delete movie

create or replace procedure delete\_movie (

p\_movie\_id in number

) as

begin

delete from movies

where movie\_id = p\_movie\_id;

commit;

end;

/

-- create declare block for delete movie

declare

p\_movie\_id number := 6;

begin

delete from movies

where movie\_id = p\_movie\_id;

commit;

dbms\_output.put\_line('movie deleted successfully.');

exception

when no\_data\_found then

dbms\_output.put\_line('no movie found with id ' || p\_movie\_id);

end;

/

select \* from movies;

-- Drop tables

drop table movies;

drop table customers;

Java Code: -

Movie.java

package com.movie.model;

public class Movie {

private int id;

private String title;

private String director;

private int releaseYear;

private String genre;

public Movie(int id, String title, String director, int releaseYear, String genre) {

this.id = id;

this.title = title;

this.director = director;

this.releaseYear = releaseYear;

this.genre = genre;

}

public Movie() {

this.id = 0;

this.title = "";

this.director = "";

this.releaseYear = 0;

this.genre = "";

}

public String toString() {

return "Movie Details: \nID = " + id +

"\nTitle = " + title +

"\nDirector = " + director +

"\nRelease Year = " + releaseYear +

"\nGenre = " + genre + "\n";

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getDirector() {

return director;

}

public void setDirector(String director) {

this.director = director;

}

public int getReleaseYear() {

return releaseYear;

}

public void setReleaseYear(int releaseYear) {

this.releaseYear = releaseYear;

}

public String getGenre() {

return genre;

}

public void setGenre(String genre) {

this.genre = genre;

}

}

MovieDao.java

package com.movie.dao;

import java.sql.Connection;

import java.sql.SQLException;

import com.movie.model.Movie;

public interface MovieDao {

void addMovie(Movie movie) throws SQLException;

Movie readMovie(int id) throws SQLException;

void updateMovie(Movie movie) throws SQLException;

void deleteMovie(int id) throws SQLException;

Connection getConnection() throws SQLException;

}

MovieDaoImpl.java

**package** com.movie.dao;

**import** java.sql.CallableStatement;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Types;

**import** com.movie.model.Movie;

**public** **class** MovieDaoImpl **implements** MovieDao {

**private** Connection con;

**public** **void** addMovie(Movie movie) **throws** SQLException {

con = getConnection();

String sql = "{call add\_movie(?, ?, ?, ?)}";

CallableStatement cst = con.prepareCall(sql);

cst.setString(1, movie.getTitle());

cst.setString(2, movie.getDirector());

cst.setInt(3, movie.getReleaseYear());

cst.setString(4, movie.getGenre());

cst.executeUpdate();

cst.close();

con.close();

}

**public** Movie readMovie(**int** id) **throws** SQLException {

con = getConnection();

String sql = "{? = call read\_movie(?)}";

CallableStatement cst = con.prepareCall(sql);

cst.registerOutParameter(1, Types.***REF\_CURSOR***);

cst.setInt(2, id);

cst.execute();

ResultSet rs = (ResultSet) cst.getObject(1);

Movie movie = **null**;

**if** (rs.next()) {

movie = **new** Movie(id, rs.getString("title"), rs.getString("director"), rs.getInt("release\_year"), rs.getString("genre"));

}

rs.close();

cst.close();

con.close();

**return** movie;

}

**public** **void** updateMovie(Movie movie) **throws** SQLException {

con = getConnection();

String sql = "{call update\_movie(?, ?, ?, ?, ?)}";

CallableStatement cst = con.prepareCall(sql);

cst.setInt(1, movie.getId());

cst.setString(2, movie.getTitle());

cst.setString(3, movie.getDirector());

cst.setInt(4, movie.getReleaseYear());

cst.setString(5, movie.getGenre());

cst.executeUpdate();

cst.close();

con.close();

}

**public** **void** deleteMovie(**int** id) **throws** SQLException {

con = getConnection();

String sql = "{call delete\_movie(?)}";

CallableStatement cst = con.prepareCall(sql);

cst.setInt(1, id);

cst.executeUpdate();

cst.close();

con.close();

}

**public** Connection getConnection() **throws** SQLException {

String URL = "jdbc:oracle:thin:@localhost:1521:xe";

String USER = "SYS AS SYSDBA";

String PWD = "Atharva2003";

**return** DriverManager.*getConnection*(URL, USER, PWD);

}

}

MovieService.java

package com.movie.service;

import java.sql.SQLException;

import java.util.Scanner;

import com.movie.dao.MovieDaoImpl;

import com.movie.model.Movie;

public class MovieService {

private static MovieDaoImpl *dao*;

public static void main(String[] args) {

*dao* = new MovieDaoImpl();

Movie movie = null;

int id, releaseYear;

String title, director, genre;

Scanner sc = new Scanner(System.*in*);

System.*out*.println("Welcome to Movie Store");

int choice;

do {

System.*out*.println("Enter your choice: ");

System.*out*.println("1. Add new movie");

System.*out*.println("2. Read movie details");

System.*out*.println("3. Update movie details");

System.*out*.println("4. Delete a movie");

System.*out*.println("5. Exit");

choice = sc.nextInt();

sc.nextLine();

try {

switch (choice) {

case 1:

System.*out*.println("Enter the movie ID: ");

id = sc.nextInt();

sc.nextLine();

System.*out*.println("Enter the movie title: ");

title = sc.nextLine();

System.*out*.println("Enter the movie director: ");

director = sc.nextLine();

System.*out*.println("Enter the movie release year: ");

releaseYear = sc.nextInt();

sc.nextLine();

System.*out*.println("Enter the movie genre: ");

genre = sc.nextLine();

movie = new Movie(id, title, director, releaseYear, genre);

*dao*.addMovie(movie);

System.*out*.println("Movie added successfully.");

break;

case 2:

System.*out*.println("Enter the movie ID: ");

id = sc.nextInt();

sc.nextLine();

movie = *dao*.readMovie(id);

if (movie != null) {

System.*out*.println(movie);

} else {

System.*out*.println("No movie found with ID: " + id);

}

break;

case 3:

System.*out*.println("Enter the movie ID to update: ");

id = sc.nextInt();

sc.nextLine();

System.*out*.println("Enter the new movie title: ");

title = sc.nextLine();

System.*out*.println("Enter the new movie director: ");

director = sc.nextLine();

System.*out*.println("Enter the new movie release year: ");

releaseYear = sc.nextInt();

sc.nextLine();

System.*out*.println("Enter the new movie genre: ");

genre = sc.nextLine();

movie = new Movie(id, title, director, releaseYear, genre);

*dao*.updateMovie(movie);

System.*out*.println("Movie updated successfully.");

break;

case 4:

System.*out*.println("Enter the movie ID to delete: ");

id = sc.nextInt();

sc.nextLine();

*dao*.deleteMovie(id);

System.*out*.println("Movie deleted successfully.");

break;

case 5:

System.*out*.println("Exited the program");

break;

default:

System.*out*.println("Invalid choice. Please enter a number between 1 and 5.");

break;

}

} catch (SQLException e) {

System.*out*.println("An error occurred: " + e.getMessage());

}

} while (choice != 5);

sc.close();

}

}

OUTPUT: -





















