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

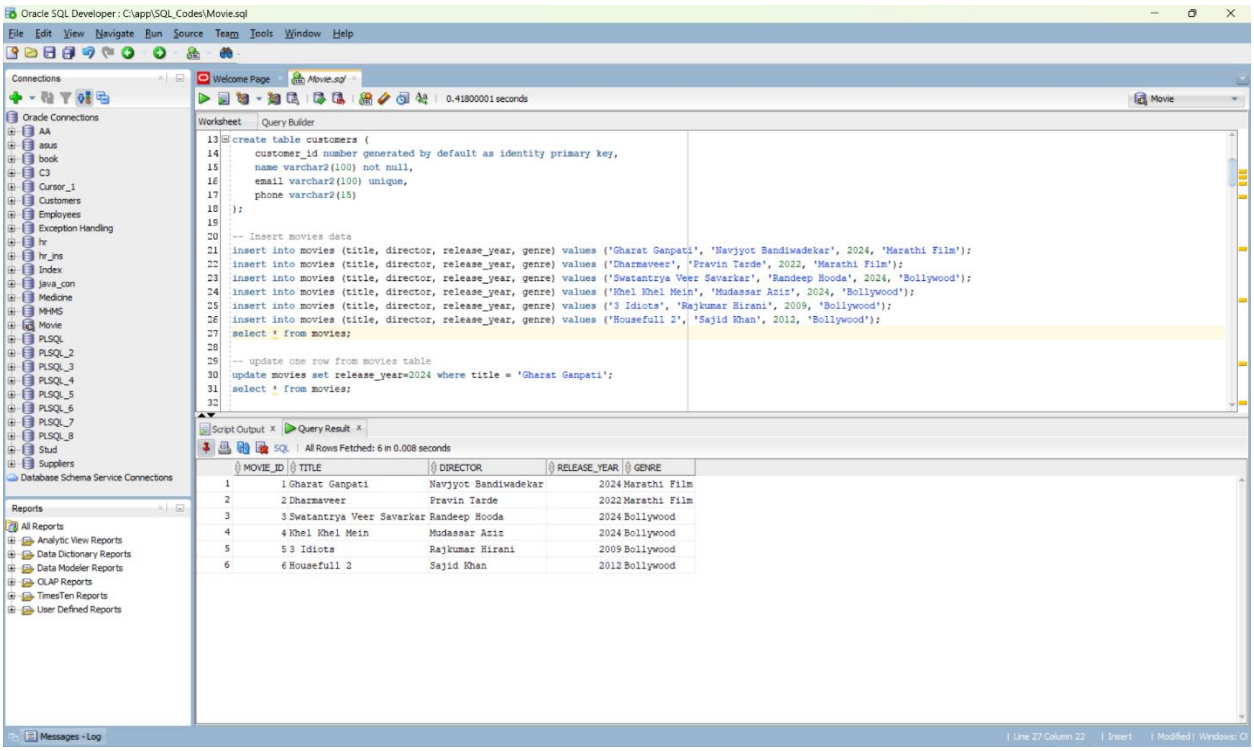
```
package com.movie.service;
import java.sql.SQLException;
import java.util.Scanner;
import com.movie.dao.MovieDaoImpl;
import com.movie.model.Movie;
public class MovieService {
    private static MovieDaoImpl dao;
    public static void main(String[] args) {
        dao = new MovieDaoImpl();
        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");
```

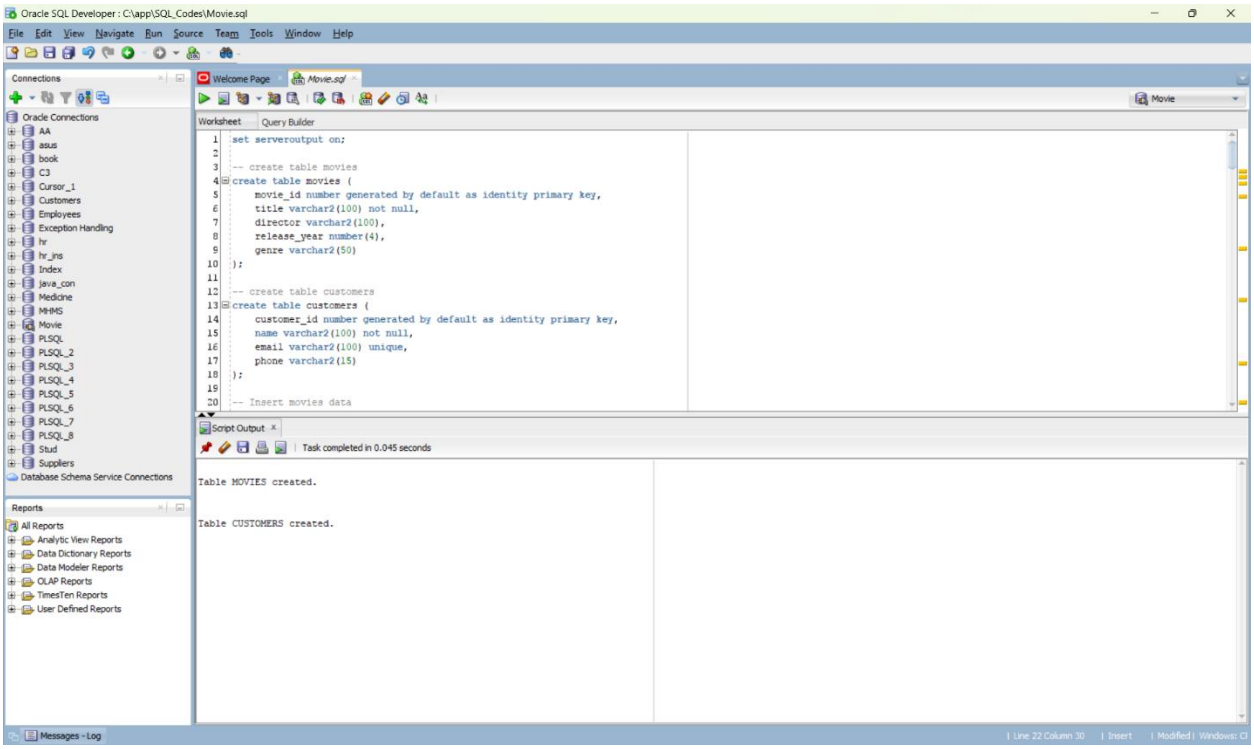
```
            System.out.println("2. Read movie details");
            System.out.println("3. Update movie details");
            System.out.println("4. Delete a movie");
            System.out.println("5. Exit");
            choice = sc.nextInt();
            sc.nextLine();
            try {
                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);
```

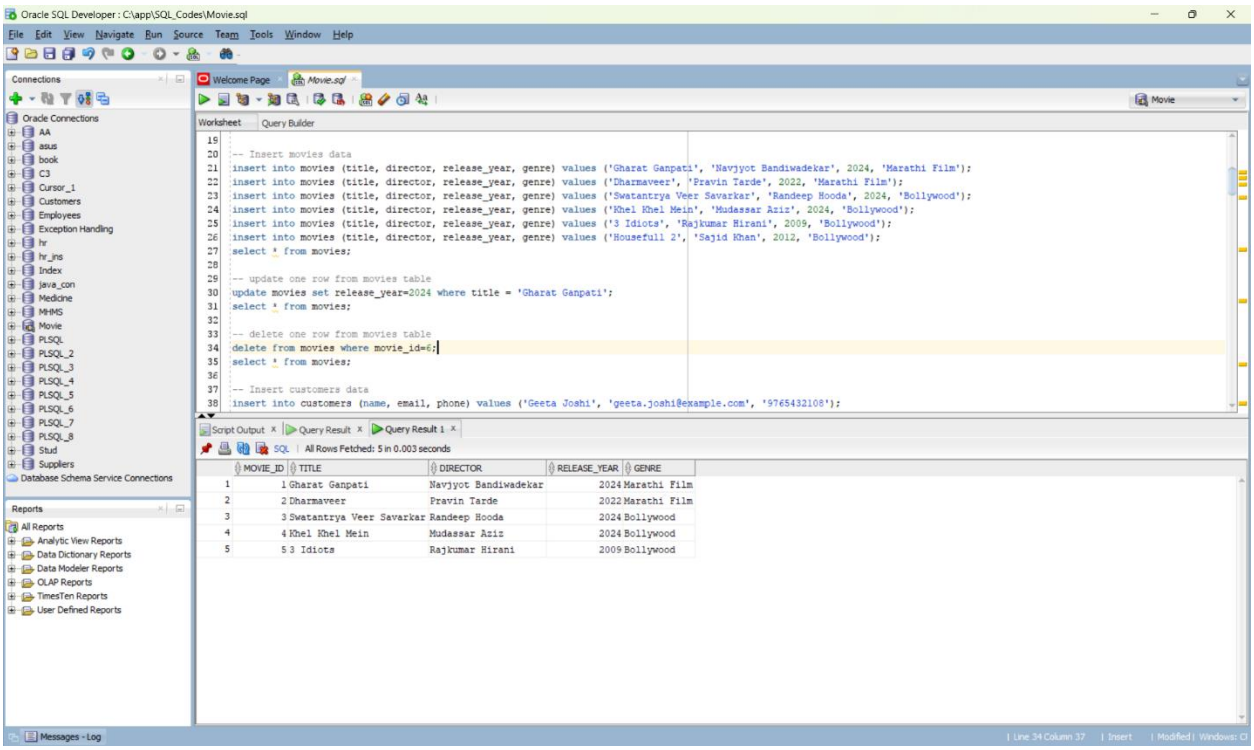
```
                    if (movie != null) {
                        System.out.println(movie);
                    } else {
                        System.out.println("No movie found with ID: " +
id);
                    }
                    break;
                case 3:
                    System.out.println("Enter 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.");
```

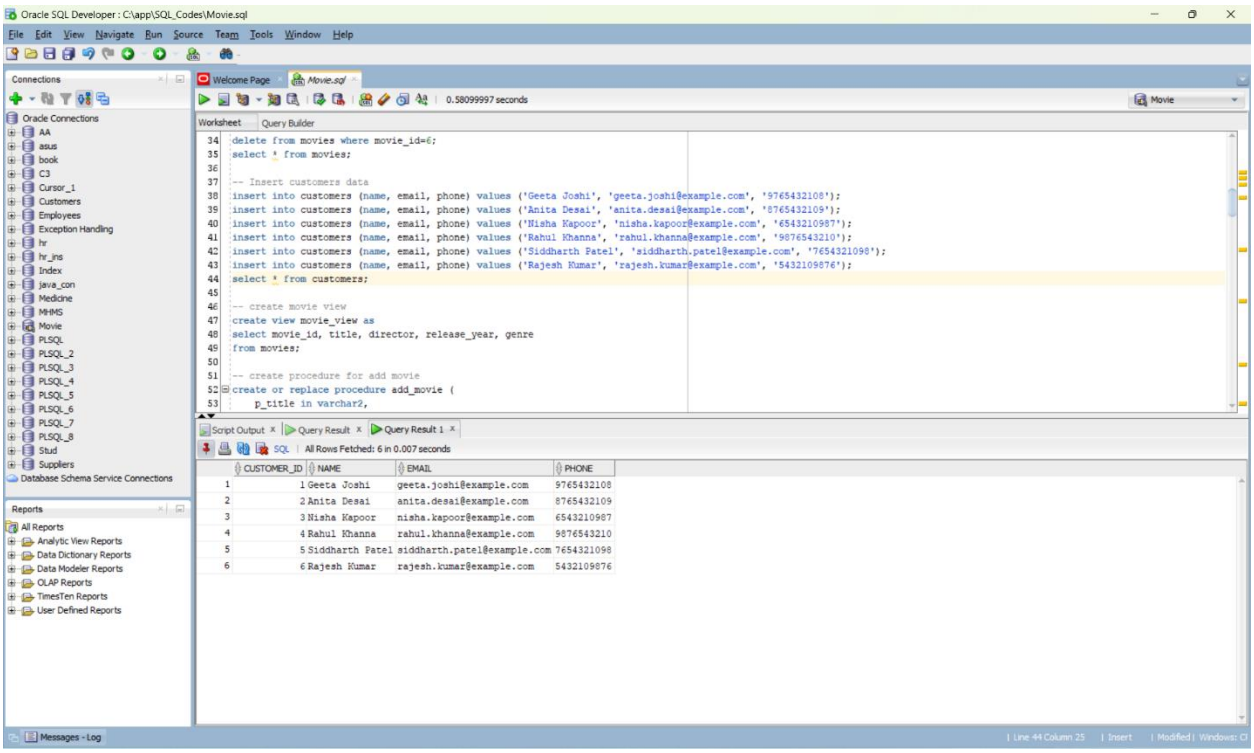
```
                    break;
                case 5:
                    System.out.println("Exited the program");
                    break;
                default:
                    System.out.println("Invalid choice. Please enter a
number between 1 and 5.");
                    break;
                }
            } catch (SQLException e) {
                System.out.println("An error occurred: " + e.getMessage());
            }
        } while (choice != 5);
        sc.close();
    }
}
```

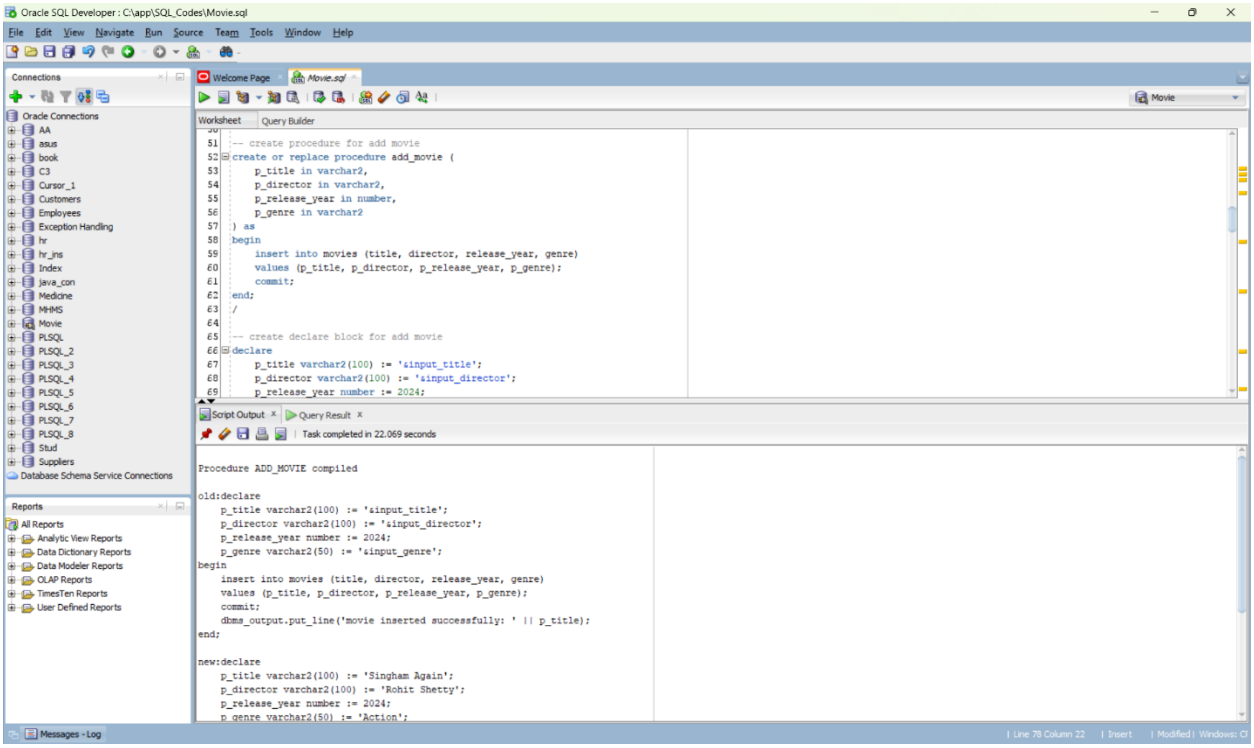
OUTPUT: -

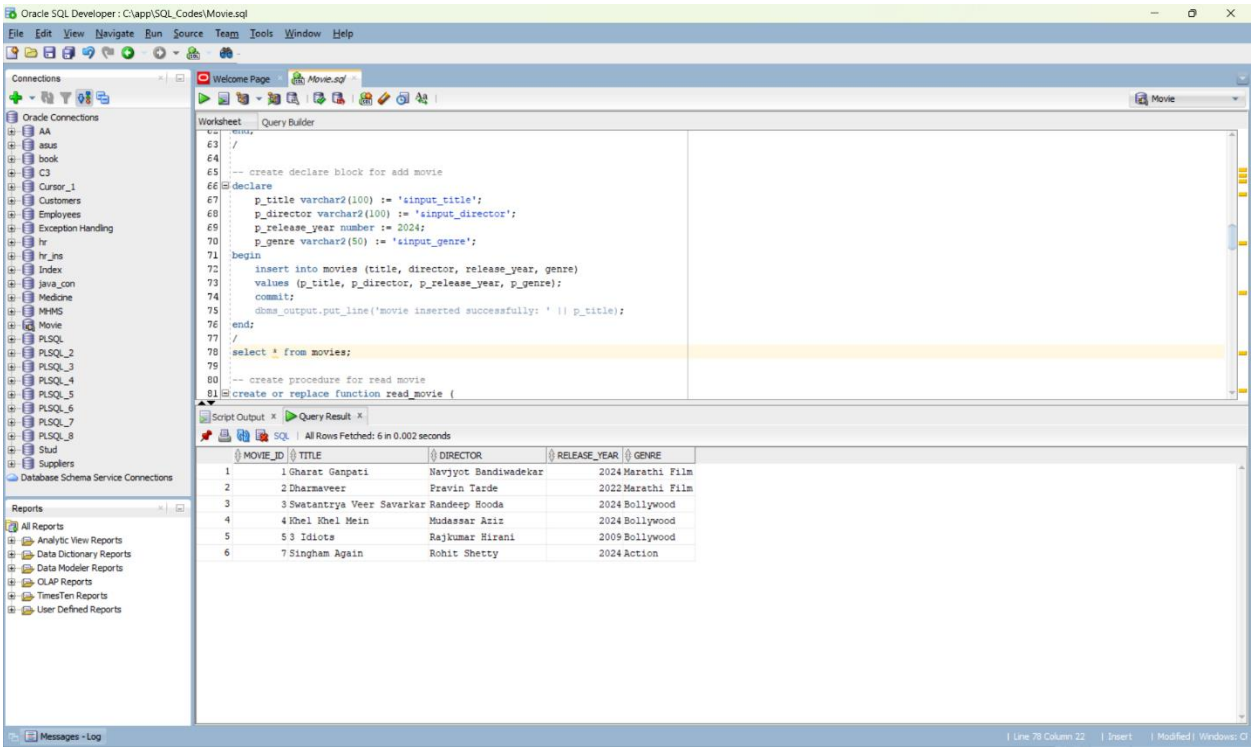


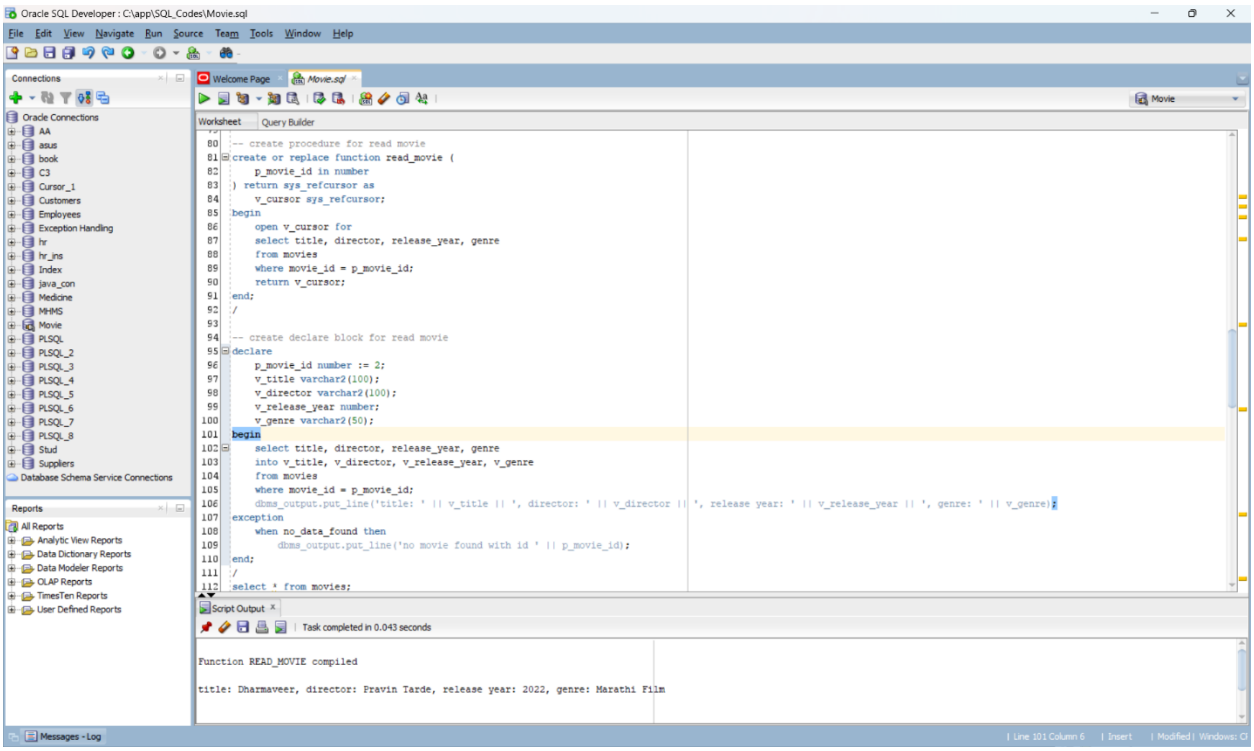

Oracle SQL Developer : C:\app\SQL_Codes\Movie.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
asus
book
C3
Cursor_1
Customers
Employees
Exception Handling
hr
hr_ins
Index
java_con
Medicine
MHMS
Movie
PLSQL
PLSQL_2
PLSQL_3
PLSQL_4
PLSQL_5
PLSQL_6
PLSQL_7
PLSQL_8
Stud
Suppliers
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
Movie.sql
Movie
Worksheet
Query Builder
1 set serveroutput on;
2
3 -- create table movies
4 create table movies (
5 movie_id number generated by default as identity primary key,
6 title varchar2(100) not null,
7 director varchar2(100),
8 release_year number(4),
9 genre varchar2(50)
10);
11
12 -- create table customers
13 create table customers (
14 customer_id number generated by default as identity primary key,
15 name varchar2(100) not null,
16 email varchar2(100) unique,
17 phone varchar2(15)
18);
19
20 -- Insert movies data
Script Output
Task completed in 0.045 seconds
Table MOVIES created.
Table CUSTOMERS created.
Messages - Log
Line 22 Column 30
Insert
Modified
Windows: C

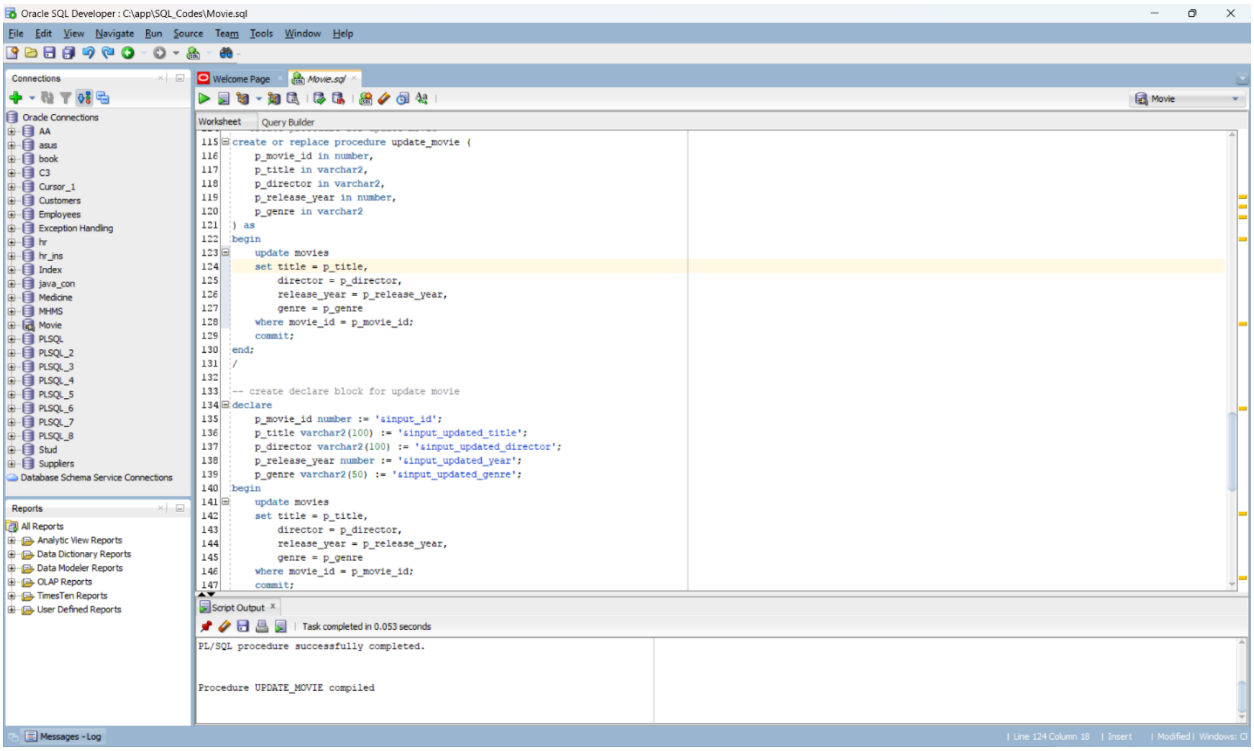

Oracle SQL Developer : C:\app\SQL_Codes\Movie.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
asus
book
C3
Cursor_1
Customers
Employees
Exception Handling
hr
hr_ins
Index
java_con
Medicine
MHMS
Movie
PLSQL
PLSQL_2
PLSQL_3
PLSQL_4
PLSQL_5
PLSQL_6
PLSQL_7
PLSQL_8
Stud
Suppliers
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
Movie.sql
Movie
Worksheet
Query Builder
19
20 -- Insert movies data
21 insert into movies (title, director, release_year, genre) values ('Gharat Ganpati', 'Navjyot Bandiwadekar', 2024, 'Marathi Film');
22 insert into movies (title, director, release_year, genre) values ('Dharmaveer', 'Pravin Tarde', 2022, 'Marathi Film');
23 insert into movies (title, director, release_year, genre) values ('Swatantrya Veer Savarkar', 'Randeep Hooda', 2024, 'Bollywood');
24 insert into movies (title, director, release_year, genre) values ('Khel Khel Mein', 'Mudassar Aziz', 2024, 'Bollywood');
25 insert into movies (title, director, release_year, genre) values ('3 Idiots', 'Rajkumar Hirani', 2009, 'Bollywood');
26 insert into movies (title, director, release_year, genre) values ('Housefull 2', 'Sajid Khan', 2012, 'Bollywood');
27 select * from movies;
28
29 -- update one row from movies table
30 update movies set release_year=2024 where title = 'Gharat Ganpati';
31 select * from movies;
32
33 -- delete one row from movies table
34 delete from movies where movie_id=6;
35 select * from movies;
36
37 -- Insert customers data
38 insert into customers (name, email, phone) values ('Geeta Joshi', 'geeta.joshi@example.com', '9765432108');
Script Output
Query Result
Query Result 1
SQL
All Rows Fetched: 5 in 0.003 seconds
MOVIE_ID TITLE DIRECTOR RELEASE_YEAR GENRE
1 1 Gharat Ganpati Navjyot Bandiwadekar 2024 Marathi Film
2 2 Dharmaveer Pravin Tarde 2022 Marathi Film
3 3 Swatantrya Veer Savarkar Randeep Hooda 2024 Bollywood
4 4 Khel Khel Mein Mudassar Aziz 2024 Bollywood
5 5 3 Idiots Rajkumar Hirani 2009 Bollywood
Messages - Log
Line 34 Column 37
Insert
Modified
Windows: C

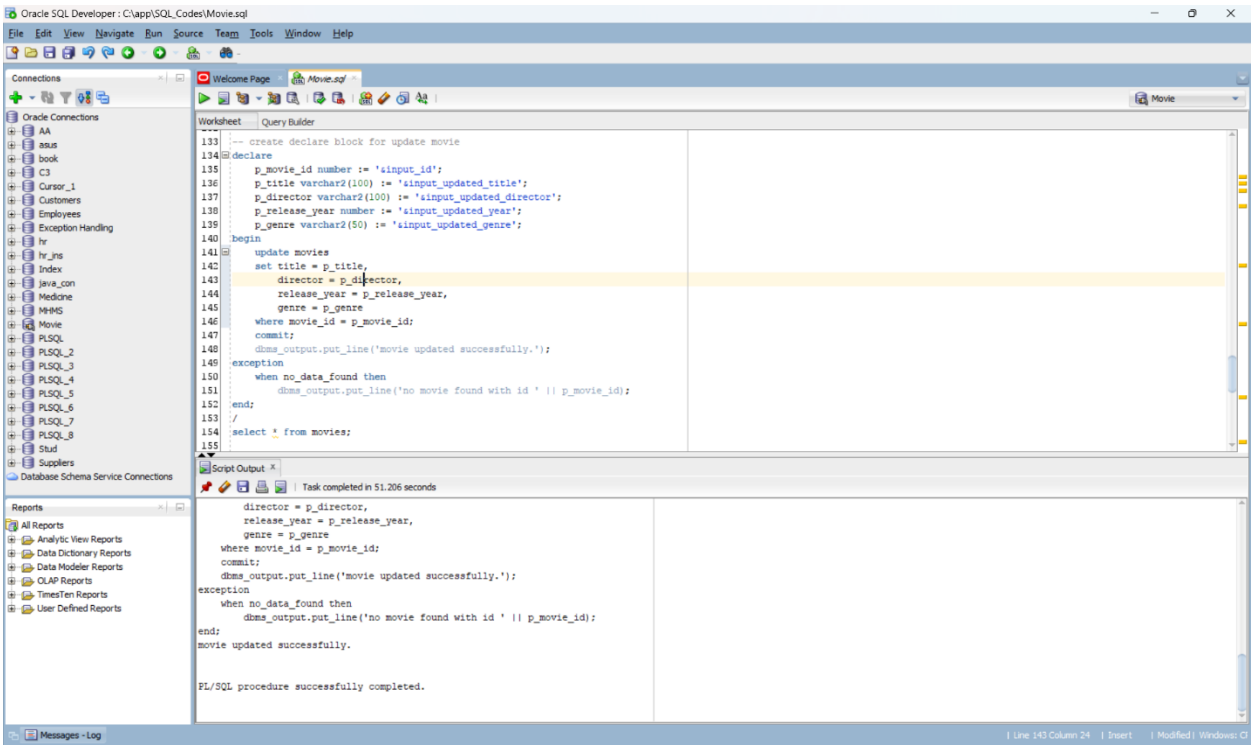

Oracle SQL Developer : C:\app\SQL_Codes\Movie.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
asus
book
C3
Cursor_1
Customers
Employees
Exception Handling
hr
hr_ins
Index
java_con
Medicine
MHMS
Movie
PLSQL
PLSQL_2
PLSQL_3
PLSQL_4
PLSQL_5
PLSQL_6
PLSQL_7
PLSQL_8
Stud
Suppliers
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
Movie.sql
0.58099997 seconds
Movie
Worksheet
Query Builder
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,
Script Output
Query Result
Query Result 1
SQL | All Rows Fetched: 6 in 0.007 seconds
CUSTOMER_ID NAME EMAIL PHONE
1 1 Geeta Joshi geeta.joshi@example.com 9765432108
2 2 Anita Desai anita.desai@example.com 8765432109
3 3 Nisha Kapoor nisha.kapoor@example.com 6543210987
4 4 Rahul Khanna rahul.khanna@example.com 9876543210
5 5 Siddharth Patel siddharth.patel@example.com 7654321098
6 6 Rajesh Kumar rajesh.kumar@example.com 5432109876
Messages - Log
Line 44 Column 25
Insert
Modified
Windows: CI

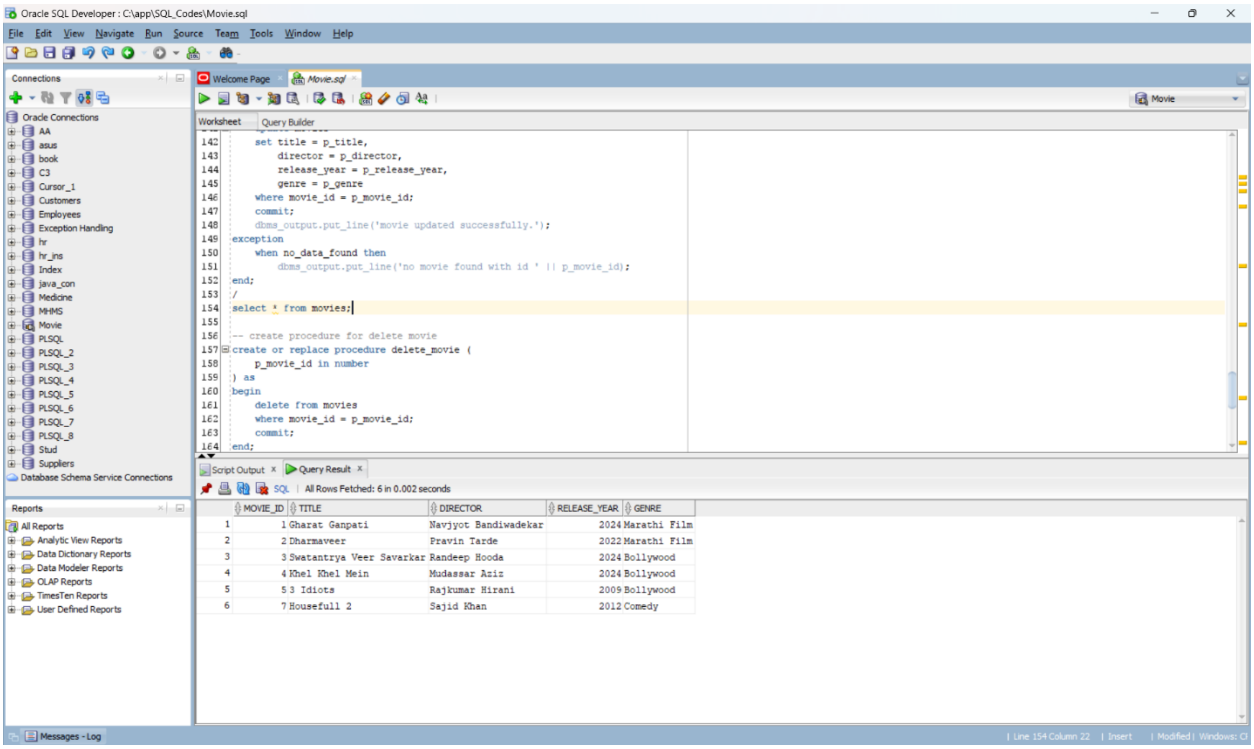

Oracle SQL Developer : C:\app\SQL_Codes\Movie.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
asus
book
C3
Cursor_1
Customers
Employees
Exception Handling
hr
hr_ins
Index
java_con
Medicine
MHMS
Movie
PLSQL
PLSQL_2
PLSQL_3
PLSQL_4
PLSQL_5
PLSQL_6
PLSQL_7
PLSQL_8
Stud
Suppliers
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
Movie.sql
Movie
Worksheet
Query Builder
-- 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;
Script Output
Query Result
Task completed in 22.069 seconds
Procedure ADD_MOVIE compiled
old: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;
new:declare
p_title varchar2(100) := 'Singham Again';
p_director varchar2(100) := 'Rohit Shetty';
p_release_year number := 2024;
p_genre varchar2(50) := 'Action';
Messages - Log
Line 78 Column 22
Insert
Modified
Windows: CI

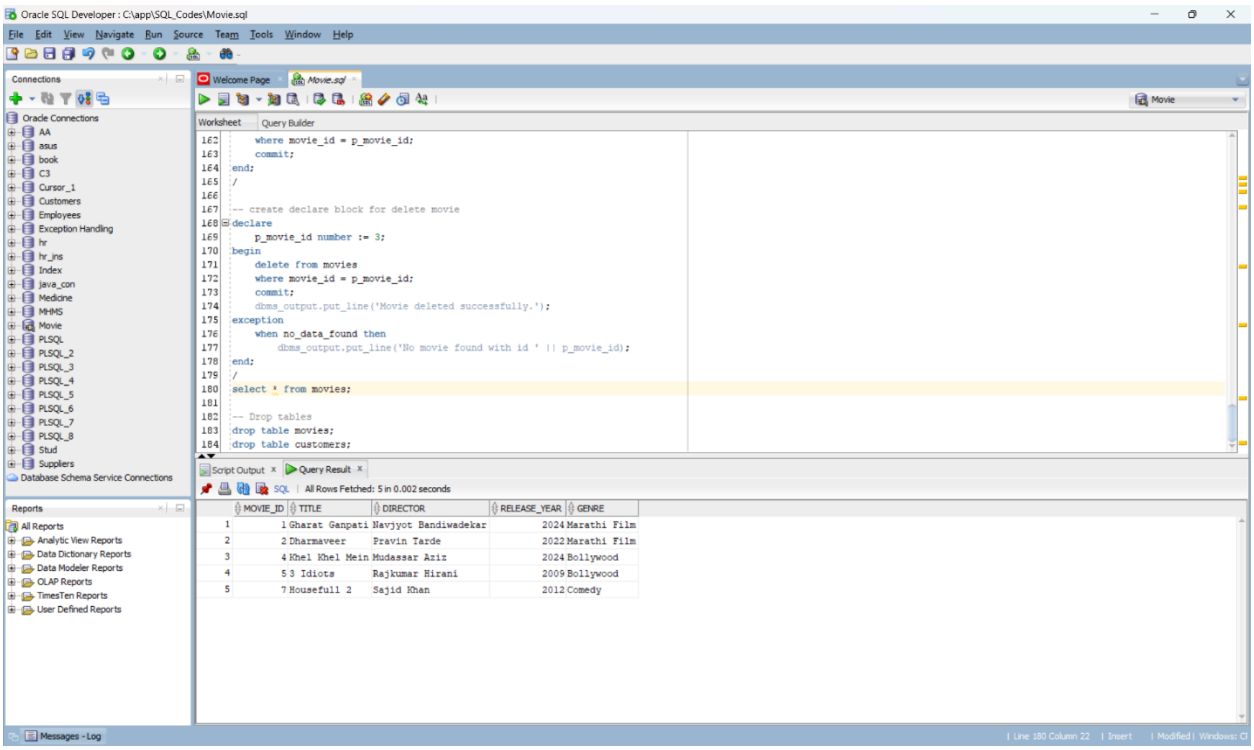

Oracle SQL Developer : C:\app\SQL_Codes\Movie.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
asus
book
C3
Cursor_1
Customers
Employees
Exception Handling
hr
hr_ins
Index
java_con
Medicine
MHMS
Movie
PLSQL
PLSQL_2
PLSQL_3
PLSQL_4
PLSQL_5
PLSQL_6
PLSQL_7
PLSQL_8
Stud
Suppliers
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
Movie.sql
Worksheet
Query Builder
Movie
/
-- 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 (
Script Output
Query Result
All Rows Fetched: 6 in 0.002 seconds
MOVIE_ID TITLE DIRECTOR RELEASE_YEAR GENRE
1 1 Gharat Ganpati Navjyot Bandiwadekar 2024 Marathi Film
2 2 Dharmaveer Pravin Tarde 2022 Marathi Film
3 3 Swatantrya Veer Savarkar Randeep Hooda 2024 Bollywood
4 4 Khel Khel Mein Mudassar Aziz 2024 Bollywood
5 5 3 Idiots Rajkumar Hirani 2009 Bollywood
6 7 Singham Again Rohit Shetty 2024 Action
Messages - Log
Line 78 Column 22
Insert
Modified
Windows: CI


Oracle SQL Developer : C:\app\SQL_Codes\Movie.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
asus
book
C3
Cursor_1
Customers
Employees
Exception Handling
hr
hr_ins
Index
java_con
Medicine
MHMS
Movie
PLSQL
PLSQL_2
PLSQL_3
PLSQL_4
PLSQL_5
PLSQL_6
PLSQL_7
PLSQL_8
Stud
Suppliers
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
Movie.sql
Worksheet
Query Builder
Movie
-- 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;
Script Output
Task completed in 0.043 seconds
Function READ_MOVIE compiled
title: Dharmaveer, director: Pravin Tarde, release year: 2022, genre: Marathi Film
Messages - Log
Line 101 Column 6
Insert
Modified
Windows: CI


Oracle SQL Developer : C:\app\SQL_Codes\Movie.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
asus
book
C3
Cursor_1
Customers
Employees
Exception Handling
hr
hr_ins
Index
java_con
Medicine
MHMS
Movie
PLSQL
PLSQL_2
PLSQL_3
PLSQL_4
PLSQL_5
PLSQL_6
PLSQL_7
PLSQL_8
Stud
Suppliers
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
Movie.sql
Movie
Worksheet
Query Builder
115 create or replace procedure update_movie (
116 p_movie_id in number,
117 p_title in varchar2,
118 p_director in varchar2,
119 p_release_year in number,
120 p_genre in varchar2
121) as
122 begin
123 update movies
124 set title = p_title,
125 director = p_director,
126 release_year = p_release_year,
127 genre = p_genre
128 where movie_id = p_movie_id;
129 commit;
130 end;
131 /
132
133 -- create declare block for update movie
134 declare
135 p_movie_id number := '&input_id';
136 p_title varchar2(100) := '&input_updated_title';
137 p_director varchar2(100) := '&input_updated_director';
138 p_release_year number := '&input_updated_year';
139 p_genre varchar2(50) := '&input_updated_genre';
140 begin
141 update movies
142 set title = p_title,
143 director = p_director,
144 release_year = p_release_year,
145 genre = p_genre
146 where movie_id = p_movie_id;
147 commit;
Script Output
Task completed in 0.053 seconds
PL/SQL procedure successfully completed.
Procedure UPDATE_MOVIE compiled
Messages - Log
Line 124 Column 18
Insert
Modified
Windows: C


Oracle SQL Developer : C:\app\SQL_Codes\Movie.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
asus
book
C3
Cursor_1
Customers
Employees
Exception Handling
hr
hr_ins
Index
java_con
Medicine
MHMS
Movie
PLSQL
PLSQL_2
PLSQL_3
PLSQL_4
PLSQL_5
PLSQL_6
PLSQL_7
PLSQL_8
Stud
Suppliers
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
Movie.sql
Movie
Worksheet
Query Builder
133 -- create declare block for update movie
134 declare
135 p_movie_id number := '&input_id';
136 p_title varchar2(100) := '&input_updated_title';
137 p_director varchar2(100) := '&input_updated_director';
138 p_release_year number := '&input_updated_year';
139 p_genre varchar2(50) := '&input_updated_genre';
140 begin
141 update movies
142 set title = p_title,
143 director = p_director,
144 release_year = p_release_year,
145 genre = p_genre
146 where movie_id = p_movie_id;
147 commit;
148 dbms_output.put_line('movie updated successfully.');
149 exception
150 when no_data_found then
151 dbms_output.put_line('no movie found with id ' || p_movie_id);
152 end;
153 /
154 select * from movies;
155
Script Output
Task completed in 51.206 seconds
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;
movie updated successfully.
PL/SQL procedure successfully completed.
Messages - Log
Line 143 Column 24
Insert
Modified
Windows: C


Oracle SQL Developer : C:\app\SQL_Codes\Movie.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
asus
book
C3
Cursor_1
Customers
Employees
Exception Handling
hr
hr_ins
Index
java_con
Medicine
MHMS
Movie
PLSQL
PLSQL_2
PLSQL_3
PLSQL_4
PLSQL_5
PLSQL_6
PLSQL_7
PLSQL_8
Stud
Suppliers
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
Movie.sql
Movie
Worksheet
Query Builder
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;
Script Output
Query Result
All Rows Fetched: 6 in 0.002 seconds
MOVIE_ID TITLE DIRECTOR RELEASE_YEAR GENRE
1 1 Gharat Ganpati Navjyot Bandiwadekar 2024 Marathi Film
2 2 Dharmaveer Pravin Tarde 2022 Marathi Film
3 3 Swatantrya Veer Savarkar Randeep Hooda 2024 Bollywood
4 4 Khel Khel Mein Mudassar Aziz 2024 Bollywood
5 5 3 Idiots Rajkumar Hirani 2009 Bollywood
6 7 Housefull 2 Sajid Khan 2012 Comedy
Messages - Log
Line 154 Column 22
Insert
Modified
Windows: C


Oracle SQL Developer : C:\app\SQL_Codes\Movie.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
AA
asus
book
C3
Cursor_1
Customers
Employees
Exception Handling
hr
hr_ins
Index
java_con
Medicine
MHMS
Movie
PLSQL
PLSQL_2
PLSQL_3
PLSQL_4
PLSQL_5
PLSQL_6
PLSQL_7
PLSQL_8
Stud
Suppliers
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
Movie.sql
Movie
Worksheet
Query Builder
where movie_id = p_movie_id;
commit;
end;
/
-- create declare block for delete movie
declare
p_movie_id number := 3;
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;
Script Output
Query Result
All Rows Fetched: 5 in 0.002 seconds
MOVIE_ID TITLE DIRECTOR RELEASE_YEAR GENRE
1 1 Gharat Ganpati Navjyot Bandiwadekar 2024 Marathi Film
2 2 Dharmaveer Pravin Tarde 2022 Marathi Film
3 4 Khel Khel Mein Mudassar Aziz 2024 Bollywood
4 5 3 Idiots Rajkumar Hirani 2009 Bollywood
5 7 Housefull 2 Sajid Khan 2012 Comedy
Messages - Log
Line 180 Column 22
Insert
Modified
Windows: C