

CPS510 - A3

By: Isaac Martin, Bowie Chau, Hachi Ndu

Instructor: Dr. S. B. Tajali

TA: George Lopez

Section #: 06

Lab Number: 3

Sept 18, 2024

Table Creation:

```
CREATE TABLE movie(  
    movieID INT PRIMARY KEY NOT NULL,  
    movieName VARCHAR(50) NOT NULL,  
    moviePoster VARCHAR(50) NOT NULL,  
    movieDirector VARCHAR(50) NOT NULL,  
    movieDescription VARCHAR(50) NOT NULL,  
    moviePrice FLOAT NOT NULL,  
    movieRating INT NOT NULL,  
    movieGenre VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE admin(  
    adminUsername VARCHAR(50) PRIMARY KEY NOT NULL,  
    adminEmail VARCHAR(50) NOT NULL UNIQUE,  
    adminPassword VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE customer(  
    customerID INT PRIMARY KEY,  
    firstName VARCHAR(50) NOT NULL,  
    lastName VARCHAR(50) NOT NULL,  
    customerUsername VARCHAR(50) NOT NULL UNIQUE,  
    customerPassword VARCHAR(50) NOT NULL,  
    email VARCHAR(50) NOT NULL,  
    customerBirthday VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE payment(  
    paymentID INT PRIMARY KEY,  
    billingAddress VARCHAR(50) NOT NULL,  
    cardHolderName VARCHAR(50) NOT NULL,  
    customerID INT NOT NULL,  
    CONSTRAINT customerIDFK FOREIGN KEY (customerID) REFERENCES  
customer(customerID)  
);  
  
CREATE TABLE membership(  
    membershipID INT PRIMARY KEY,  
    startDate DATE NOT NULL,  
    endDate DATE NOT NULL,  
    subscriptionCharge INT NOT NULL
```

);

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Connections' pane shows a tree view for the 'hachicps510' database, including tables (ADMIN, CUSTOMER, MEMBERSHIP, MOVIE, PAYMENT), views, indexes, and other database objects. The main workspace is divided into two panes: 'Worksheet' and 'Query Builder'. The 'Worksheet' pane contains a SQL script with the following content:

```
--create movie store tables
CREATE TABLE movie(
    movieID INTEGER PRIMARY KEY NOT NULL,
    movieName VARCHAR(50) NOT NULL,
    moviePoster VARCHAR(100) NOT NULL,
    movieDirector VARCHAR(50) NOT NULL,
    movieDescription VARCHAR(50) NOT NULL,
    moviePrice INT NOT NULL,
    movieRating INTEGER NOT NULL,
    movieGenre INTEGER NOT NULL
);

CREATE TABLE admin(
    adminUsername VARCHAR(50) PRIMARY KEY NOT NULL,
    adminEmail VARCHAR(50) NOT NULL UNIQUE,
    adminPassword VARCHAR(50) NOT NULL
);

CREATE TABLE customer(
    customerID INT PRIMARY KEY,
    firstName VARCHAR(50) NOT NULL,
    lastName VARCHAR(50) NOT NULL,
    customerUsername VARCHAR(50) NOT NULL UNIQUE,
    customerPassword VARCHAR(50) NOT NULL
);
```

Below the script, the 'Script Output' pane shows the execution results:

```
Table MOVIE created.

Table ADMIN created.

Table CUSTOMER created.

Table PAYMENT created.

Table MEMBERSHIP created.
```

The 'Script Output' pane also indicates that the task was completed in 0.062 seconds.

```

CREATE TABLE customer(
    customerID INT PRIMARY KEY,
    firstName VARCHAR(50) NOT NULL,
    lastName VARCHAR(50) NOT NULL,
    customerUsername VARCHAR(50) NOT NULL UNIQUE,
    customerPassword VARCHAR(50) NOT NULL,
    email VARCHAR(50) NOT NULL,
    customerBirthDay VARCHAR(50) NOT NULL
);

CREATE TABLE payment(
    paymentID INT PRIMARY KEY,
    billingAddress VARCHAR(50) NOT NULL,
    cardHolderName VARCHAR(50) NOT NULL,
    customerID INT NOT NULL,
    CONSTRAINT customerIDFK FOREIGN KEY (customerID) REFERENCES customer(customerID)
);

```

hachicps510.sql × moviequery.sql × PAYMENT ×					
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes					
📌 📝 🔗 Actions...					
	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_T
1	CUSTOMERIDFK	Foreign_Key	(null)	HNDU	CUST
2	SYS_C001903307	Check	"BILLINGADDRESS" IS NOT NULL	(null)	(nul
3	SYS_C001903308	Check	"CARDHOLDERNAME" IS NOT NULL	(null)	(nul
4	SYS_C001903309	Check	"CUSTOMERID" IS NOT NULL	(null)	(nul
5	SYS_C001903310	Primary_Key	(null)	(null)	(nul

(CustomerID referenced as foreign key in payment)

Testing:

```
INSERT INTO movie (movieID, movieName, moviePoster, movieDirector, movieDescription,
moviePrice, movieRating, movieGenre)
```

```
VALUES
```

```
(1,'Aladdin','https://originalvintagemovieposters.com/wp-content/uploads/2020/03/Aladdin-9002-scaled.jpg', 'Guy Ritchie', 'Aladdin, a kind thief, woos Jasmin, the princess',10, 7, 'Family');
```

(Creates movie1)

```
INSERT INTO movie (movieID, movieName, moviePoster, movieDirector, movieDescription,
moviePrice, movieRating, movieGenre)
```

```
VALUES (10, 'CREED',
```

```
'https://image.tmdb.org/t/p/original/1BfTsk5VWuw8FCocAhCyqnRbEzq.jpg', 'Ryan Coogler',
'Boxing', 10.99, 10, 'Action');
```

(Creates movie10)

```
INSERT INTO movie (movieID, movieName, moviePoster, movieDirector, movieDescription, moviePrice, movieRating, movieGenre)
VALUES (1, 'Aladdin', 'https://originalvintagemovieposters.com/wp-content/uploads/2020/03/Aladdin-9002-scaled.jpg', 'Guy Ritchie', 'Alad

INSERT INTO movie (movieID, movieName, moviePoster, movieDirector, movieDescription, moviePrice, movieRating, movieGenre)
VALUES (2, 'CREED', 'CREEDimg', 'Ryan Coogler', 'Boxing', 10.99, 10, 'Action');
```

MOVIEID	MOVIEName	MOVIEPOSTER
1	1 Aladdin	https://originalvintagemovieposters.com/wp-content/uploads/2020/03/Aladdin-9002-scaled.jpg
2	2 CREED	CREEDimg

```
INSERT INTO movie (movieID, movieName, moviePoster, movieDirector, movieDescription,
moviePrice, movieRating, movieGenre)
```

```
VALUES (2,'Avatar The Way Of The Water','https://dylancolestudio.com/movie-posters', 'James
Cameron', 'Ex marine falls in love with the Navi culture',15, 9, 'Family');
```

(Creates movie2)

```
INSERT INTO movie (movieID, movieName, moviePoster, movieDirector, movieDescription,
moviePrice, movieRating, movieGenre)
```

```
VALUES (3,'The Jungle Book','https://dylancolestudio.com/movie-posters', 'Jon Favreau', 'An
adventure story about a man-cub named Mowgli',12, 7, 'Family');
```

(Creates movie3)

```

INSERT INTO movie (movieID, movieName, moviePoster, movieDirector, movieDescription, moviePr
VALUES (1, 'Aladdin', 'https://originalvintagemovieposters.com/wp-content/uploads/2020/03/Ala

INSERT INTO movie (movieID, movieName, moviePoster, movieDirector, movieDescription, moviePr
VALUES (10, 'CREED', 'https://image.tmbd.org/t/p/original/1BfTsk5VWuw8FCocAhCyqnRbEzq.jpg',

INSERT INTO movie (movieID, movieName, moviePoster, movieDirector, movieDescription, moviePr
VALUES (2, 'Avatar The Way Of The Water', 'https://dylancolestudio.com/movie-posters', 'James

INSERT INTO movie (movieID, movieName, moviePoster, movieDirector, movieDescription, moviePr
VALUES (3, 'The Jungle Book', 'https://dylancolestudio.com/movie-posters', 'Jon Favreau', 'An

```

Script Output x

Task completed in 0.063 seconds

```

)
startDate DATE NOT NULL,
endDate DATE NOT NULL,
subscriptionCharge INT NOT NULL
)
Error report -
SQL Error: ORA-00955: name is already used by an existing object
00955. 00000 - "name is already used by an existing object"
*Cause:
*Action:

1 row inserted.

1 row inserted.

1 row inserted.

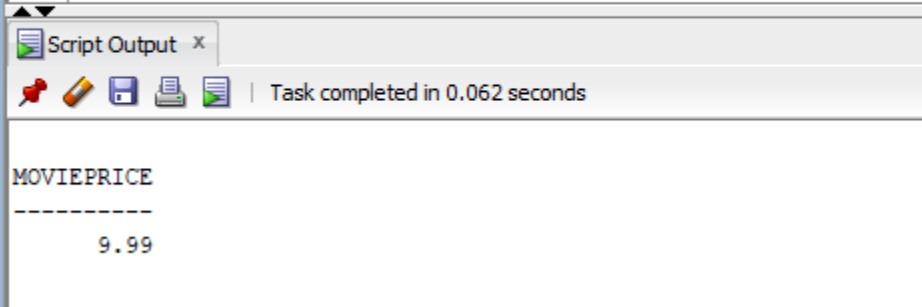
1 row inserted.

```

MOVIEID	MOVIE NAME	MOVIEPOSTER	MOVIEDIRECTOR	MOVIE
1	Aladdin	https://originalvintagemovieposters.com/wp-content/uploads/2020/03/Aladdin-9002-scaled.jpg	Guy Ritchie	Aladdi
2	10 CREED	https://image.tmbd.org/t/p/original/1BfTsk5VWuw8FCocAhCyqnRbEzq.jpg	Ryan Coogler	Boxing
3	2 Avatar The Way Of The Water	https://dylancolestudio.com/movie-posters	James Cameron	Ex mar
4	3 The Jungle Book	https://dylancolestudio.com/movie-posters	Jon Favreau	An adv

SELECT moviePrice FROM movie where movieid = 10;
(Returns the price of movie10 = 9.99)

```
--return price
SELECT moviePrice FROM movie where movieid = 10;
```

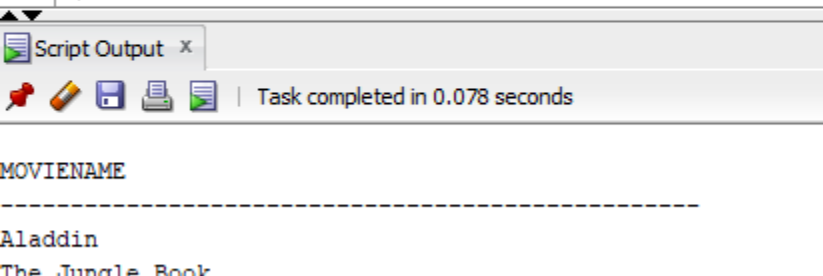


The screenshot shows a script output window titled "Script Output x". It contains a red pushpin icon, an eraser icon, a save icon, a print icon, and a refresh icon. Below the icons, it says "Task completed in 0.062 seconds". The output of the query is displayed in a table format with the column header "MOVIEPRICE" and a single row with the value "9.99".

MOVIEPRICE
9.99

SELECT movieName
FROM MOVIE
WHERE moviePrice = 10.99;
(Returns movies with price 10.99)

```
--return movies with price 10.99
SELECT movieName
FROM MOVIE
WHERE moviePrice = 10.99;
```



The screenshot shows a script output window titled "Script Output x". It contains a red pushpin icon, an eraser icon, a save icon, a print icon, and a refresh icon. Below the icons, it says "Task completed in 0.078 seconds". The output of the query is displayed in a table format with the column header "MOVIEName" and two rows with the values "Aladdin" and "The Jungle Book".

MOVIEName
Aladdin
The Jungle Book