

CPS510 - A4

By: Isaac Martin, Bowie Chau, Hachi Ndu

Instructor: Dr. S. B. Tajali

TA: George Lopez

Section #: 06

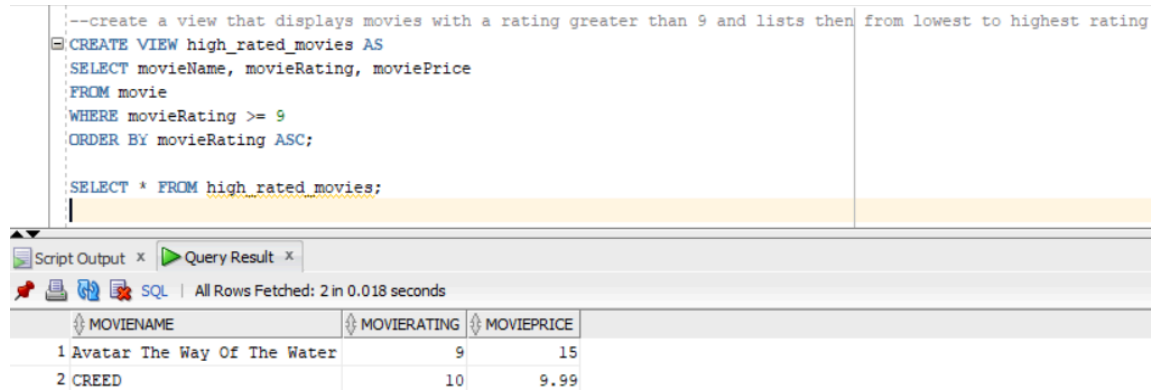
Lab Number: 4

Oct 1st, 2024

--create a view that displays movies with a rating greater than 9 and lists then from lowest to highest rating

```
CREATE VIEW high_rated_movies AS
SELECT movieName, movieRating, moviePrice
FROM movie
WHERE movieRating >= 9
ORDER BY movieRating ASC;
```

SELECT * FROM high_rated_movies;



The screenshot shows a SQL script editor with the following code:

```
--create a view that displays movies with a rating greater than 9 and lists then from lowest to highest rating
CREATE VIEW high_rated_movies AS
SELECT movieName, movieRating, moviePrice
FROM movie
WHERE movieRating >= 9
ORDER BY movieRating ASC;

SELECT * FROM high_rated_movies;
```

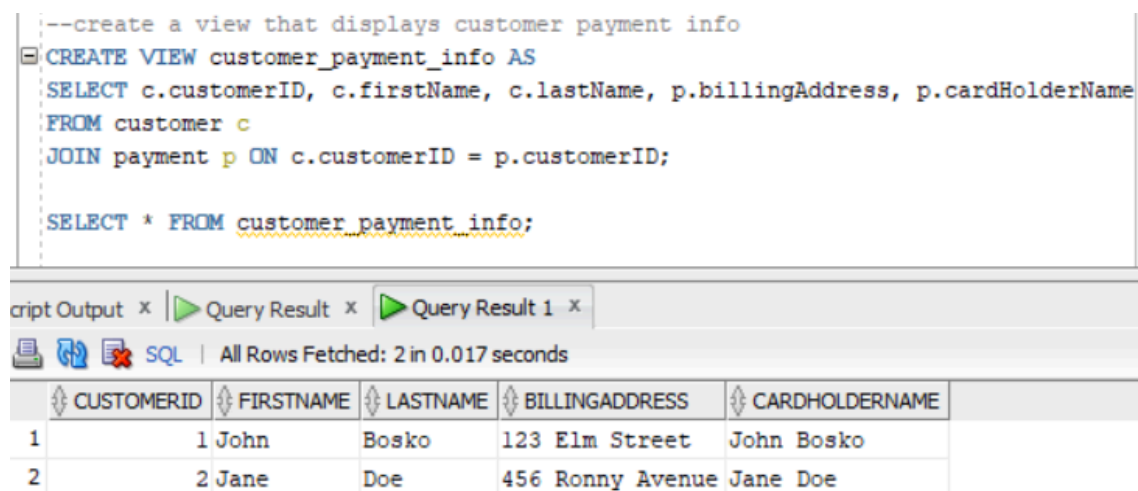
Below the script, the 'Query Result' tab is active, displaying the results of the second query. The status bar indicates 'All Rows Fetched: 2 in 0.018 seconds'.

| | MOVIE NAME | MOVIE RATING | MOVIE PRICE |
|---|-----------------------------|--------------|-------------|
| 1 | Avatar The Way Of The Water | 9 | 15 |
| 2 | CREED | 10 | 9.99 |

--create a view that displays customer info from customer table and payment info from payment table

```
CREATE VIEW customer_payment_info AS
SELECT c.customerID, c.firstName, c.lastName, p.billingAddress, p.cardHolderName
FROM customer c
JOIN payment p ON c.customerID = p.customerID;
```

SELECT * FROM customer_payment_info;



The screenshot shows a SQL script editor with the following code:

```
--create a view that displays customer payment info
CREATE VIEW customer_payment_info AS
SELECT c.customerID, c.firstName, c.lastName, p.billingAddress, p.cardHolderName
FROM customer c
JOIN payment p ON c.customerID = p.customerID;

SELECT * FROM customer_payment_info;
```

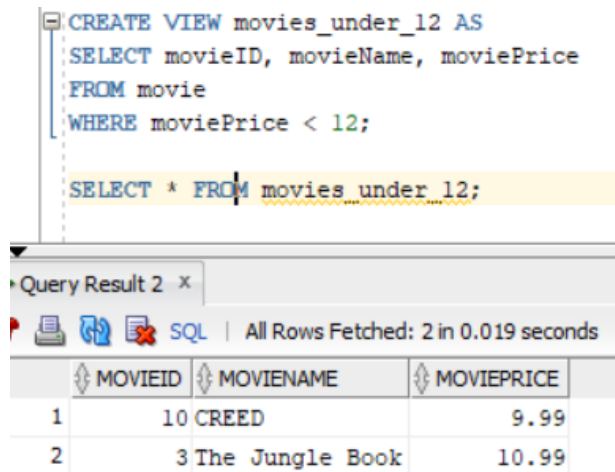
Below the script, the 'Query Result' tab is active, displaying the results of the second query. The status bar indicates 'All Rows Fetched: 2 in 0.017 seconds'.

| | CUSTOMERID | FIRSTNAME | LASTNAME | BILLINGADDRESS | CARDHOLDERNAME |
|---|------------|-----------|----------|------------------|----------------|
| 1 | 1 | John | Bosko | 123 Elm Street | John Bosko |
| 2 | 2 | Jane | Doe | 456 Ronny Avenue | Jane Doe |

--create a view that collects all movies with a price less than \$12

```
CREATE VIEW movies_under_12 AS  
SELECT movieID, movieName, moviePrice  
FROM movie  
WHERE moviePrice < 12;
```

SELECT * FROM movies_under_12;



The screenshot shows a SQL query editor with the following code:

```
CREATE VIEW movies_under_12 AS  
SELECT movieID, movieName, moviePrice  
FROM movie  
WHERE moviePrice < 12;  
  
SELECT * FROM movies_under_12;
```

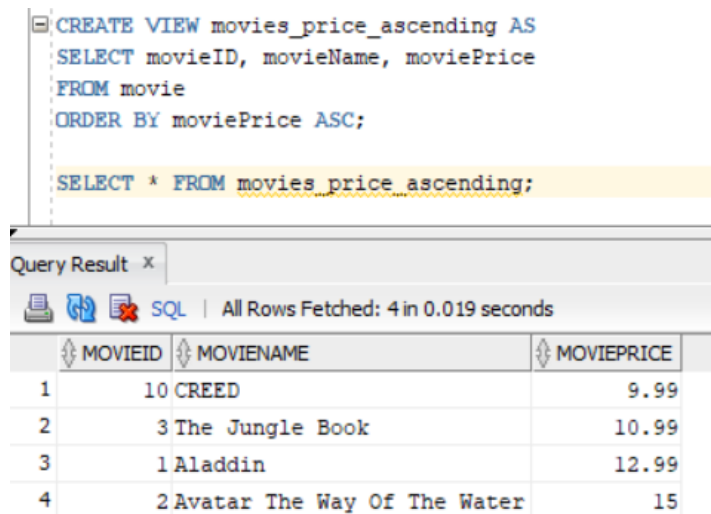
Below the editor, the 'Query Result 2' window displays the results of the second query. It shows 2 rows fetched in 0.019 seconds. The results are as follows:

| | MOVIEID | MOVIEPRICE |
|---|-------------------|------------|
| 1 | 10 CREED | 9.99 |
| 2 | 3 The Jungle Book | 10.99 |

--create a view that sorts movies based on price in ascending order

```
CREATE VIEW movies_price_ascending AS  
SELECT movieID, movieName, moviePrice  
FROM movie  
ORDER BY moviePrice ASC;
```

SELECT * FROM movies_price_ascending;



The screenshot shows a SQL query editor with the following code:

```
CREATE VIEW movies_price_ascending AS  
SELECT movieID, movieName, moviePrice  
FROM movie  
ORDER BY moviePrice ASC;  
  
SELECT * FROM movies_price_ascending;
```

Below the editor, the 'Query Result' window displays the results of the second query. It shows 4 rows fetched in 0.019 seconds. The results are as follows:

| | MOVIEID | MOVIEPRICE |
|---|-------------------------------|------------|
| 1 | 10 CREED | 9.99 |
| 2 | 3 The Jungle Book | 10.99 |
| 3 | 1 Aladdin | 12.99 |
| 4 | 2 Avatar The Way Of The Water | 15 |

```
--select distinct movie genres without repeating
SELECT DISTINCT movieGenre
FROM movie
ORDER BY movieGenre;
```

```
--select distinct movie genres without repeating
SELECT DISTINCT movieGenre
FROM movie
ORDER BY movieGenre;
```

| Query Result x | |
|--|------------|
| SQL All Rows Fetched: 2 in 0.018 seconds | |
| | MOVIEGENRE |
| 1 | Action |
| 2 | Family |

```
--create view that gets relevant information for customer billing
CREATE VIEW customer_billing_info AS
SELECT
    c.firstName,
    c.lastName,
    p.billingAddress,
    m.subscriptionCharge
FROM
    customer c
JOIN
    payment p ON c.customerID = p.customerID
JOIN
    membership m ON c.customerID = m.customerID;
```

```
--Select all from view
SELECT * FROM customer_billing_info;
```

| | FIRSTNAME | LASTNAME | BILLINGADDRESS | SUBSCRIPTIONCHARGE |
|---|-----------|----------|------------------|--------------------|
| 1 | John | Bosko | 123 Elm Street | 99 |
| 2 | Jane | Doe | 456 Ronny Avenue | 99 |