22- NOTE: WHERE clause is used to filter data

23- NOTE: WHERE clause always comes right after FROM clause.

24- Display all the rows where amount(column) is 0(value) from the table(payment).

```
SELECT
*
FROM
payment
WHERE amount = 0
```

24- Display count of all the rows where amount(column) is 0(value) from the table(payment).

```
SELECT
COUNT(*)
FROM
payment
WHERE amount = 0
```

24- Display details about a customer whose first_name(column) is ADAM(value) from table(customer).

```
SELECT
*
FROM
customer
WHERE first_name = ’ADAM’
```

25- NOTE: ’ADAM’ & ’Adam’ will give different results

26- Display the count of payments(rows) made by customer_id(column)=100 from table(payment).

```
SELECT
count(*)
FROM
payment
WHERE customer_id=100
```

27- Display the last_name(column) of the customer(row) where first_name(column)=’ERICA’ from table(customer).

```
SELECT
last_name
FROM
customer
WHERE first_name=’ERICA’
```

28- NOTE: WHERE operators
# = : Equal to
# != : Not Equal to
# <> : Not Equal to
# >= : Greater than or Equal to
# <= : Less than or Equal to
# is null : Value is null
# is not null : Value is not null

29- Display all rows where amount(column) is greater or equal to 10(value) in descending order from table(payment).

```
SELECT
*
FROM
payment
WHERE amount >= 10
ORDER BY amount DESC;
```

30- Display the count of rentals that are not returned yet i.e. count the rows where return_date(column) is null(value) from table rental.

```
SELECT
COUNT(*)
FROM
rental
WHERE return_date IS NULL
```

31- Display payment_id,amount of all payments(rows) where amount(column) is less than or equal to $2(value) from table(payment)

```
SELECT
payment_id, amount
FROM
payment
WHERE amount <= 2
```

32- NOTE: We can chain AND and OR clause with WHERE clause to apply mutiple conditions. AND will be processed before OR.

33- Display all rows where amount is 10.99 or 9.99 and customer_id = 426 from table(payment).

```
SELECT
*
FROM
payment
WHERE ( amount = 10.99
OR amount = 9.99 )
AND customer_id = 426
```

34- Display details(rows) of the customer with customer_id(column) 322, 346 and 354 where amount(column) is either less than 2 or greater than 10 from table(payment). Order rows by ascending order of customer_id (column) and descending order of amount(column).

```
SELECT
*
FROM
payment
WHERE
(customer_id = 322 OR customer_id = 346 OR customer_id = 354 )
AND
```

```
(amount < 2 OR amount > 10)
ORDER BY customer_id ASC,amount DESC
```

35- NOTE: Use BETWEEN .... AND .... clause to filter values in given range (inclusive of border values).

36- NOTE: Use NOT BETWEEN .... AND .... clause to filter values not in given range (border value not included).

37- Display rows where amount(column) is in between 1.99 and 6.99 from table(payment).

```
SELECT
*
FROM
payment
WHERE amount BETWEEN 1.99 AND 6.99 ### 1.99 & 6.99 will be included ###
```

38- Display all rows except where amount(column) is in between 1.99 and 6.99 from table(payment).

```
SELECT
*
FROM
payment
WHERE amount NOT BETWEEN 1.99 AND 6.99 ### 1.99 & 6.99 will NOT be included ###
```

39- NOTE:
Following query:
- WHERE date BETWEEN ’2020-01-24’ AND ’2020-01-26’

is internally read as:
- WHERE date BETWEEN ’2020-01-24 00:00’ AND ’2020-01-26 00:00’

which is 24/01/2020 12:00 A.M. to 26/01/2020 12:00 A.M.

- which means for date 26/01/2020 time after 12:00 A.M. is not included.

To add date 26/01/2020 use:
- WHERE date BETWEEN ’2020-01-24 00:00’ AND ’2020-01-26 23:59’

40- Display all rows that has rental_date(column) between 24/05/2005 and 26/05/2005 in descending order
from table(rental)

```
SELECT
*
FROM
rental
WHERE rental_date BETWEEN ’2005-05-24’ AND ’2005-05-26 23:59’
ORDER BY rental_date DESC
```

41- Display count of all payments(rows) made between payment_date(column) 26/01/2020 and 27/01/2020
with amount(column) between 1.99 and 3.99 from table(payment).

```
SELECT
COUNT(*)
FROM
payment
WHERE amount BETWEEN 1.99 AND 3.99
```

```
AND payment_date BETWEEN ’2020-01-26’ AND ’2020-01-27 23:59’
```

42- NOTE: IN operator used to list only values that are in given set (...,...,...,...,...).

NOTE: NOT IN operator used to list only values that are NOT in given set (...,...,...,...).

43- Display the data of customers with customer_id = 256,323,456,100,99,67 from table(customer).

```
SELECT
*
FROM
customer
WHERE customer_id IN (256,323,456,100,99,67)
```

44- Display payments(rows) made by customers with customer_id(column) = (12,25,67,93,124,234)(values)
of amount(column) = (4.99,7.99,9.99) in month of January from table(payment).

```
SELECT
*
FROM
payment
WHERE customer_id IN (12,25,67,93,124,234)
AND amount IN (4.99,7.99,9.99)
AND payment_date BETWEEN ’2020-01-01’ AND ’2020-01-31 23:59’
```

45- NOTE: Use LIKE operator with WHERE clause to filter by matching against a pattern.

46- Wildcards of LIKE clause:

- % : Represents 0 or more characters.
- _ : Represents single character.

47- LIKE operator is case-sensitive.

48- Display rows of actor(column) where first_name starts with ’A’.

```
SELECT *
FROM actor
WHERE first_name LIKE ’A%’
```

49- NOTE: ILIKE operator is case-insensitive LIKE operator.

50- Display rows of film that have word ’Drama’ anywhere in its description(column) from the table(film).

```
SELECT *
FROM film
WHERE description LIKE ’%Drama%’
```

51- Display list of actors that have ’T’ in 3rd place in their first_name(column) from table(actor).

```
SELECT *
FROM actor
WHERE first_name LIKE ’_ _ _D%’ ### spaces here are for understanding only ###
```

52- Display all rows where customer’s first_name(column) is of 3 letters and last_name(column) ends with ’Y’ or ’X’ from table(customer).

```
SELECT first_name,last_name
FROM customer
WHERE first_name LIKE ’___’
AND (last_name LIKE ’%X’ OR last_name LIKE ’%Y’)
```

53- NOTE: -- single line comment

54- NOTE: /* multiple line comment */

55- NOTE: AS clause is used to give alias (another name).

56- Display count of films as no_of_movies(alias) from table(film) where description has the word ’Saga’ anywhere in it and the film title either starts with ’A’ or ends with ’R’.

```
SELECT COUNT(film_id) as no_of_movies
from film
WHERE description LIKE ’%Saga%’
AND (title LIKE ’A%’ OR title LIKE ’%R’)
```

57- Display rows of customers in descending order of their last name where their first name has letter ’ER’ in it and second letter of their name is ’A’ from the table (customer).

```
SELECT *
FROM customer
WHERE first_name LIKE ’%ER%’
AND first_name LIKE ’_A%’
ORDER BY last_name DESC
```

58- Display rows of payment done on day 01/05/2020 where either amount was 0 or in between 3.99-7.99
from table (payment).

```
SELECT COUNT(payment_id)
FROM payment
WHERE (amount=0 OR amount BETWEEN 3.99 AND 7.99)
AND (payment_date BETWEEN ’2020-05-01 00:00’ AND ’2020-05-01 23:59’)
```