* SELECT is the clause we use every time we want to query information from a database.
* AS renames a column or table.
* DISTINCT return unique values.
* WHERE is a popular command that lets you filter the results of the query based on conditions that you specify.
* LIKE and BETWEEN are special operators.
* AND and OR combines multiple conditions.
* ORDER BY sorts the result.
* LIMIT specifies the maximum number of rows that the query will return.
* CASE creates different outputs.

AS is a keyword in SQL that allows you to rename a column or table using an alias. The new name can be anything you want as long as you put it inside of single quotes. Here we renamed the name column as Titles.

SELECT imdb\_rating AS 'IMDb'

FROM movies;

DISTINCT is used to return unique values in the output. It filters out all duplicate values in the specified column(s).

SELECT DISTINCT genre

FROM movies;

WHERE clause filters the result set to only include rows where the following condition is true.

SELECT \*

FROM movies

WHERE year > 2014;

LIKE is a special operator used with the WHERE clause to search for a specific pattern in a column when you want to compare similar values.

SELECT \*

FROM movies

WHERE name LIKE 'Se\_en';

% is a wildcard character that matches zero or more missing letters in the pattern.

SELECT \*

FROM movies

WHERE name LIKE '%man%';

It is not possible to test for NULL values with comparison operators, such as = and !=.

Instead, we will have to use these operators:

* IS NULL
* IS NOT NULL

SELECT name, imdb\_rating

FROM movies

WHERE imdb\_rating IS NULL;

The BETWEEN operator can be used in a WHEREclause to filter the result set within a certain range. The values can be numbers, text or dates.

* BETWEEN two letters *is not* inclusive of the 2nd letter.
* BETWEEN two numbers *is* inclusive of the 2nd number.

SELECT \*

FROM movies

WHERE name BETWEEN 'A' AND 'J';

or

SELECT \* FROM movies

WHERE year BETWEEN 1970 AND 1979

AND imdb\_rating > 8;

or

SELECT \* FROM movies

WHERE year < 1985 AND genre = 'horror';

or

SELECT \*

FROM movies

WHERE year > 2014 OR genre = 'action';

We can sort the results using ORDER BY, either alphabetically or numerically.

 ORDER BY always goes after WHERE (if WHERE is present).

LIMIT is a clause that lets you specify the maximum number of rows the result set will have.

SELECT \*

FROM movies

ORDER BY imdb\_rating DESC

LIMIT 3;

A CASE statement allows us to create different outputs (usually in the SELECTstatement). we can rename the column to ‘Review’ using AS:

SELECT name,

CASE

WHEN genre = 'romance' OR genre = 'comedy'

THEN 'Chill'

ELSE 'Intense'

END AS 'Mood'

FROM movies;

Or

SELECT name,

CASE

WHEN imdb\_rating > 8 THEN 'Fantastic'

WHEN imdb\_rating > 6 THEN 'Poorly Received'

ELSE 'Avoid at All Costs'

END AS 'Review'

FROM movies;

SELECT DISTINCT cuisine

FROM nomnom;

SELECT \*

FROM nomnom

WHERE cuisine = 'Chinese';

SELECT \*

FROM nomnom

WHERE review >= 4;

SELECT \*

FROM nomnom

WHERE cuisine = 'Italian'

AND price LIKE '%$$$%';

SELECT \*

FROM nomnom

WHERE name LIKE '%meatball%';

SELECT \*

FROM nomnom

WHERE neighborhood = 'Midtown'

OR neighborhood = 'Downtown'

OR neighborhood = 'Chinatown';

SELECT \*

FROM nomnom

WHERE health IS NULL;

SELECT \* FROM nomnom

ORDER BY review DESC

LIMIT 10;

SELECT name,

CASE

WHEN review > 4.5 THEN 'Extraordinary'

WHEN review > 4 THEN 'Excellent'

WHEN review > 3 THEN 'Good'

WHEN review > 2 THEN 'Fair'

ELSE 'Poor'

END AS 'Review'

FROM nomnom;

SELECT full\_name, email

FROM transaction\_data

WHERE full\_name = 'Art Vandelay' OR full\_name LIKE '% der %';