SQL Basics (Part 2)

Kian

Before you start...

- If you are **new here**, do check out the part 1 series here.
- Otherwise, welcome back to this series!

Topics we will cover...

- Where
- And, or, not
- Order by
- In
- Group by
- Like
- Having

Creating a table

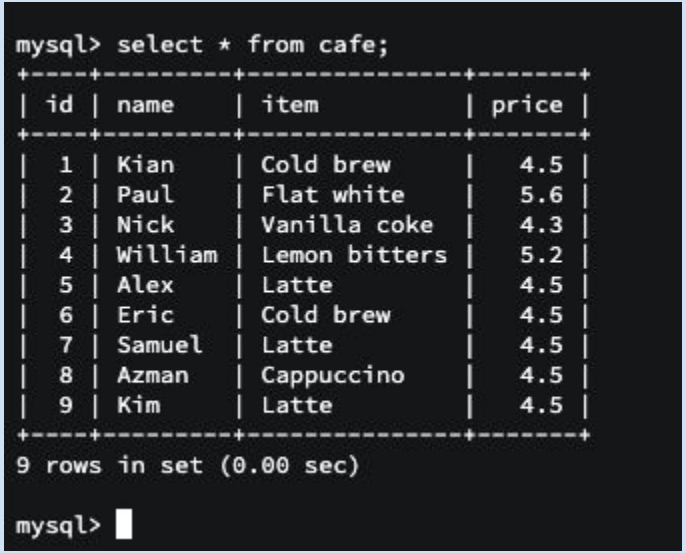
- Before you continue with the following exercises, I will need you to create a table call ‘cafe’ in the database call ‘test_database’.
- The cafe table should have:
    - **Id** (INT, NOT NULL, AUTO_INCREMENT and PRIMARY KEY)
    - **Name** (varchar(255) not null)
    - **Item** (varchar(255))
    - **Price** (float)

Creating a table

- Now, I need you to insert the entries

Name	Item	Price
Kian	Cold brew	$ 4.50
Paul	Flat White	$ 5.60
Nick	Vanilla coke	$ 4.30
William	Lemon bitters	$ 5.20
Alex	Latte	$ 4.50
Eric	Cold brew	$ 4.50
Samuel	Latte	$ 4.50
Azman	Cappuccino	$ 4.50
Kim	Latte	$ 4.50

You should be able to get this...



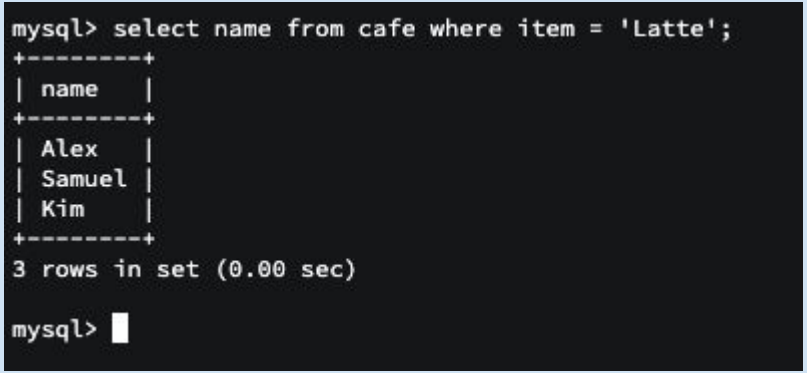
```
mysql> select * from cafe;
+----+---------+---------------+-------+
| id | name    | item          | price |
+----+---------+---------------+-------+
|  1 | Kian    | Cold brew     |   4.5 |
|  2 | Paul    | Flat white    |   5.6 |
|  3 | Nick    | Vanilla coke  |   4.3 |
|  4 | William | Lemon bitters |   5.2 |
|  5 | Alex    | Latte         |   4.5 |
|  6 | Eric    | Cold brew     |   4.5 |
|  7 | Samuel  | Latte         |   4.5 |
|  8 | Azman   | Cappuccino    |   4.5 |
|  9 | Kim     | Latte         |   4.5 |
+----+---------+---------------+-------+
9 rows in set (0.00 sec)

mysql> 
```

SQL WHERE

- The WHERE clause is used to filter out records.
- It specifies the condition you want to look into.
- Syntax:
    - SELECT <column1> from <table> where <condition>;
    - Eg of <condition>:
        - <column1> > 10
        - <column1> = “True”

SQL WHERE

- Now, we will like to pick customers who ordered latte
  - SELECT name from cafe where item = ‘latte’;



```
mysql> select name from cafe where item = 'Latte';
+--------+
| name   |
+--------+
| Alex   |
| Samuel |
| Kim    |
+--------+
3 rows in set (0.00 sec)

mysql> 
```

Exercise 1

Now, I will like you to choose customers and item with price above $ 4.50

SQL AND, OR, NOT

- The WHERE clause can be combined with AND, OR, and NOT operators.
- The AND and OR operators are used to filter records based on more than one condition:
  - The AND operator displays a record if all the conditions separated by AND are TRUE.
  - The OR operator displays a record if any of the conditions separated by OR is TRUE.

The NOT operator displays a record if the condition(s) is NOT TRUE.

Syntax:

1. SELECT <Column1> from <table> where <condition1> **and** <condition2>...
2. SELECT <Column1> from <table> where <condition1> **or** <condition2>...
3. SELECT <Column1> from <table> where not <condition1>**...**

Exercise 2

Now, I will like you to show all columns with price above $ 4.50 but not flat white.

SQL ORDER BY

- The ORDER BY keyword is used to sort the result-set in ascending or descending order.
- The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

- SELECT <Column1> from <table> ORDER BY <column1>, <column2> ...

Exercise 3

Now, I will like you to select item and order them by price in a descending order

SQL IN

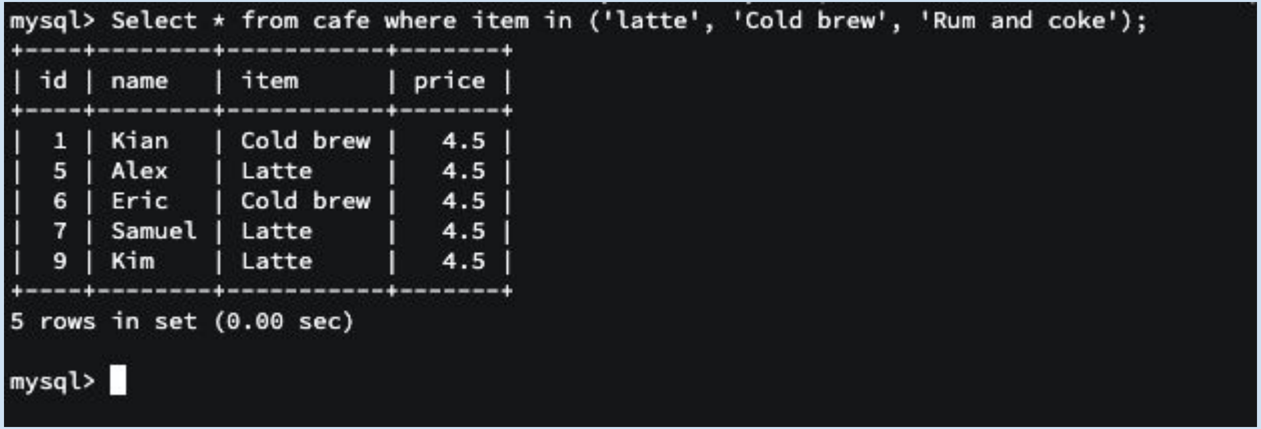
- The IN operator allows you to specify multiple values in a WHERE clause.
- The IN operator is a shorthand for multiple OR conditions.
- Syntax:
    - SELECT <column1> FROM <table> WHERE <column1> IN (value1, value2, ...);

      OR

    - SELECT <column1> FROM <table> WHERE <column1> IN (SELECT STAEMENT);

Example

Select * from cafe where item in (‘Latte’, ‘Cold brew’, ‘Rum and coke’);



```
mysql> Select * from cafe where item in ('latte', 'Cold brew', 'Rum and coke');
+----+--------+-----------+-------+
| id | name   | item      | price |
+----+--------+-----------+-------+
|  1 | Kian   | Cold brew |   4.5 |
|  5 | Alex   | Latte     |   4.5 |
|  6 | Eric   | Cold brew |   4.5 |
|  7 | Samuel | Latte     |   4.5 |
|  9 | Kim    | Latte     |   4.5 |
+----+--------+-----------+-------+
5 rows in set (0.00 sec)

mysql> 
```

Exercise 4

Now, I want you to use the IN operator to select all the records where drink is cold brew or vanilla coke.

SQL GROUP BY

- The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".
- The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.
- Syntax:
  - SELECT <Column1> from <table> where <condition1> group by <column1> order by <column1>;

Exercise 5

- Create the following:

Column 1: count	Column 2: item
Group by according to item name	

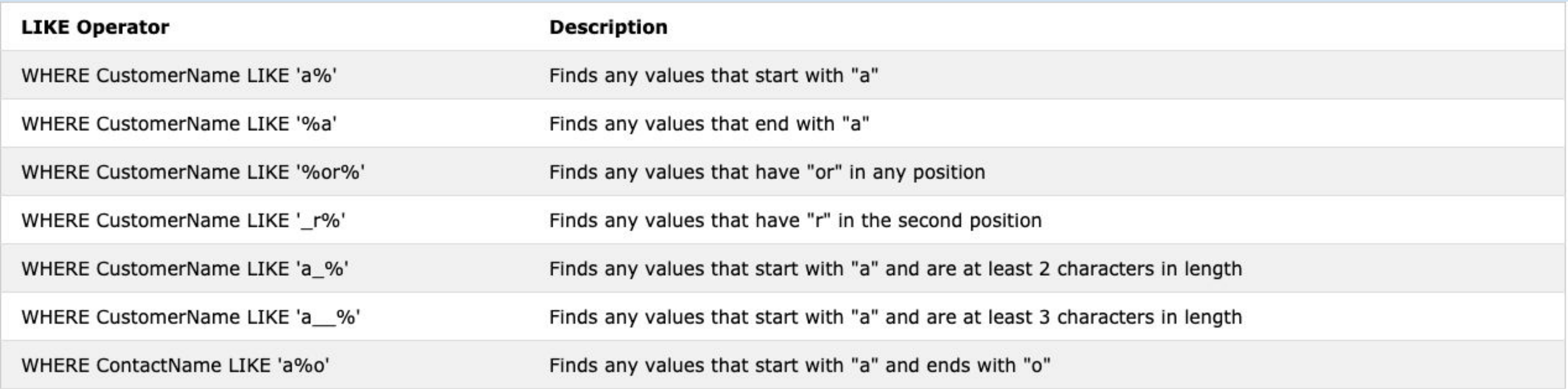
SQL Like

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the LIKE operator:

% - The percent sign represents zero, one, or multiple characters

_ - The underscore represents a single character

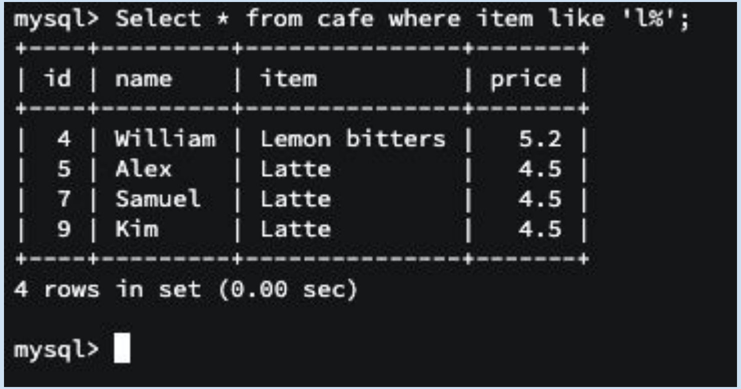


LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%'	Finds any values that start with "a" and are at least 2 characters in length
WHERE CustomerName LIKE 'a__%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

SQL LIKE

Try: **Select * from cafe where item like ‘l%’;**

We see items with the starting letter “l” gets chosen.

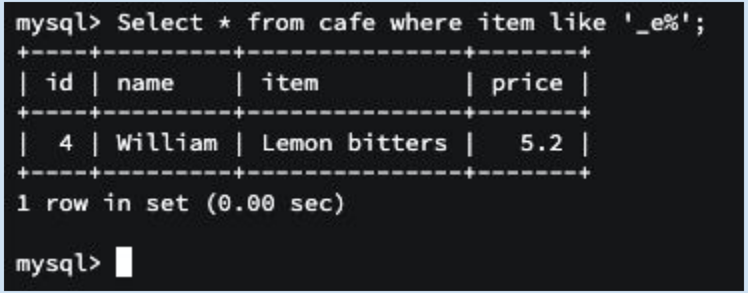


```
mysql> Select * from cafe where item like 'l%';
+----+---------+---------------+-------+
| id | name    | item          | price |
+----+---------+---------------+-------+
|  4 | William | Lemon bitters |   5.2 |
|  5 | Alex    | Latte         |   4.5 |
|  7 | Samuel  | Latte         |   4.5 |
|  9 | Kim     | Latte         |   4.5 |
+----+---------+---------------+-------+
4 rows in set (0.00 sec)

mysql> 
```

SQL LIKE

Try: **Select * from cafe where item like ‘_e%’;**

We see items with the 2nd letter “e” gets chosen.

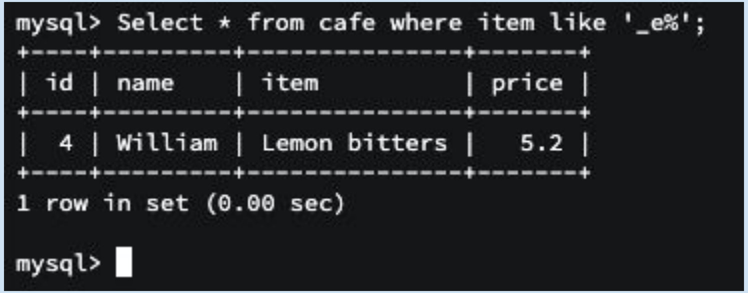


```
mysql> Select * from cafe where item like '_e%';
+----+---------+---------------+-------+
| id | name    | item          | price |
+----+---------+---------------+-------+
|  4 | William | Lemon bitters |   5.2 |
+----+---------+---------------+-------+
1 row in set (0.00 sec)

mysql> 
```

SQL LIKE

Try: **Select * from cafe where item like ‘_e%’;**

We see items with the 2nd letter “e” gets chosen.



```
mysql> Select * from cafe where item like '_e%';
+----+---------+---------------+-------+
| id | name    | item          | price |
+----+---------+---------------+-------+
|  4 | William | Lemon bitters |   5.2 |
+----+---------+---------------+-------+
1 row in set (0.00 sec)

mysql> 
```

Exercise 6

Select all customer names with item name of “tt” in any position.

SQL HAVING

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

Aggregate functions: COUNT, MIN, MAX, AVG, SUM

Syntax: SELECT column_name(s) FROM table_name WHERE condition

GROUP BY column_name(s) HAVING condition

Exercise 6

- Lists the number of customers in each item. Only include item with less than $5.50.

Part 3 will be out soon!