Christian M Rapp
Promineo Tech Beck End Developer Course | Week 8 Research

Writing Prompt #2

Research different constructs that are available in MySQL when retrieving data. What types of constructs can be put into the WHERE clause? Please include the syntax for this type of request, and examples.

When pulling data from a MySQL database, a typical call to pull data would look something like this:

```
SELECT  data_to_select
FROM    table_to_pull_from
WHERE   conditions_to_satisfy;
```

If you wanted to see all of the data in a table, you would simply use the wildcard * in your SELECT statement.

ex: **`SELECT * FROM employees;`**

`Using the above example of a typical call, we selected all (*) from the TABLE employees`

The **WHERE** clause is used to specify a certain condition. For example, using the **employees** table referenced above, you could do a select of salaried employees. Assuming the table employees had a column of **Type** (indicating salaried, hourly, manager, etc.) you could use the following statement to pull the information of all salaried employees:

```
SELECT * FROM employees WHERE Type = "salaried";
```

The **WHERE** clause uses operators to perform its actions. Just like Java, there are many operators available. Here's a list of some of the operators available with a description for each and some example syntax:

= (equals operator): `SELECT * FROM dogs WHERE Breed = 'German Shepherd';`

Typical comparison operators (work the same is in Java):
> (greater than) >= (greater than or equal to) < (less than) <= (less than or equal to)

```
Example: SELECT * FROM employees WHERE Hours <= 40;
selects all employees that have work 40 hours or less
```

The NOT (or not equal to) operator is different in MySQL. In Java programming you indicate not equals with **!= .** In MySQL you use the word **NOT**

```
EXAMPLE: SELECT * FROM dogs WHERE NOT Breed = 'German Shepherd';
```

C**onditional operators** (much like && and || in Java) can be used to make queries more specific:

```
AND EXAMPLE: SELECT * FROM dogs WHERE Breed = 'German Shepherd' AND Color = 'White';
OR EXAMPLE: SELECT * FROM dogs WHERE Breed = 'German Shepherd' OR Breed = 'Yorkie;
```

The above are just a few examples of constructs that can be used in the **WHERE** clause. It's worth noting that the **WHERE** clause is not only used in **SELECT** statements. It can also be used in **UPDATE** and **DELETE** statements as well.

Christian M Rapp
Promineo Tech Beck End Developer Course | Week 8 Research

Writing Prompt #3
List 3-5 interesting facts that you have learned about MySQL from your research?

1) The **WHERE** clause, which is used to filter records to meet a certain condition, behaves similar to **comparison** and **conditional** statements in Java programming.

2) There are many other built-in functions in MySQL that will help reduce programming on the Client Side to filter / format results. One example would be the **ORDER BY** keyword. This keyword will let you sort your results by a particular results set. When using ORDER BY, the results are returned in ascending order by default. For example:

   `SELECT * FROM dogs ORDER BY Breed; would` return results similar to "Beagle, Collie, German Shepherd, Yorkie" ***`(SELECT * FROM dogs ORDER BY Breed ASC; would be the explicit call)`***

   Adding the DESC keyword would return the results in descending order:
   `SELECT * FROM dogs ORDER BY Breed DESC; would` return results similar to "Yorkie. German Shepherd, Collie, Beagle"

3) Another handy statement built into MySQL is the **GROUP BY** statement. The GROUP BY statement allows you to collect data and group the result by one (or multiple) columns. An example I can think of where this would be handy would be a customer database. You could sort the database by city or zip code (or city **AND** zip code) and create a list of customers in that selected area. If you combined this with the COUNT statement, you could quickly find out how many customers you had in a certain city and zip code:

```
SELECT COUNT(CustomerID), City, ZipCode (COUNT counts the records matching the criteria provided)
FROM Customers
GROUP BY City, ZipCode;                  (By default, results are returned in ascending order)
```

One step further: you could gather the email or mailing addresses of these customers and send them special offers.

There's so much more built-in to MySQL that will help aid in programming the DB Client. I can't wait to discover more!

RESEARCH CREDITS
https://dev.mysql.com/doc/refman/8.0/en/retrieving-data.html
https://www.w3schools.com/mysql/mysql_where.asp
https://www.javatpoint.com/mysql-group-by