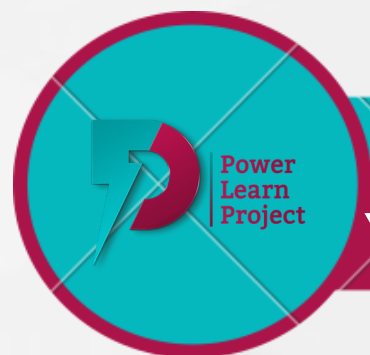


MySQL Filtering Data: WHERE Clause and Beyond

week 11



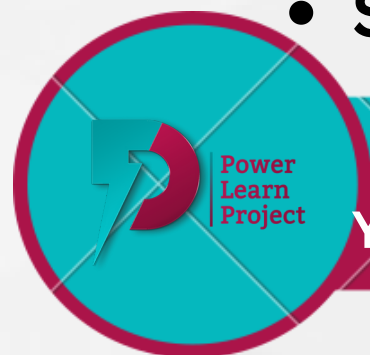
Database Design

You are live with Stanley Munga

Learning Objectives

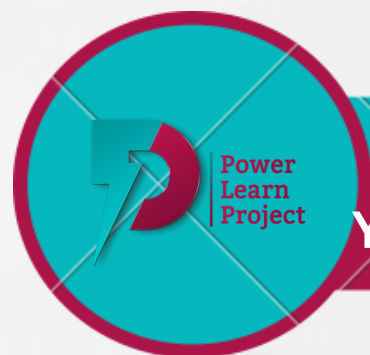
Objectives

- Understand the basic syntax of the SELECT statement.
- Learn the role of the FROM clause in specifying the table.
- Introduce and use the WHERE clause for data filtering.
- Explore conditions such as equality, inequality, greater than, and less than.
- Combine conditions using logical operators (AND, OR, NOT).
- Use wildcard characters for pattern matching and the LIKE operator.
- Handle NULL values with IS NULL and IS NOT NULL.
- Perform range queries with BETWEEN and IN operators.
- Sort query results using the ORDER BY clause with ASC and DESC.



I. Introduction to Filtering Data

In MySQL, the WHERE clause is used to filter rows returned by a query based on specified conditions. It enables you to extract specific data from a table, making your queries more focused and relevant. Additionally, other clauses and operators contribute to effective data filtering.



II. The WHERE Clause

1. Basic Syntax:

SELECT column1, column2, ...

FROM table_name

WHERE condition;

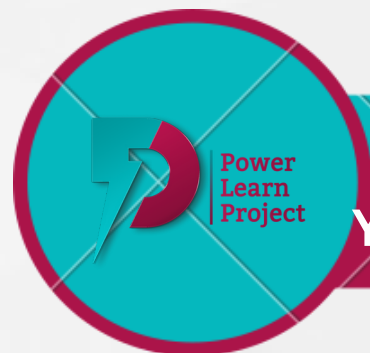
2. Operators in WHERE Clause:

Comparison Operators:

=, != or <>, <, <=, >, >=

Logical Operators:

AND, OR, NOT



3. Examples:

-- Simple equality condition

```
SELECT * FROM employees WHERE department_id = 10;
```

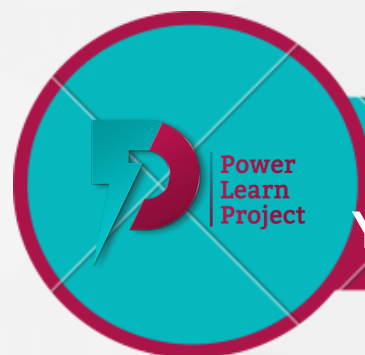
-- Using AND operator

```
SELECT * FROM orders WHERE order_date >= '2023-01-01' AND order_status = 'Shipped';
```

-- Using OR operator

```
SELECT * FROM products WHERE category = 'Electronics' OR category = 'Appliances';
```

III. Additional Filtering Techniques



1. IN Operator:

Allows you to specify multiple values in a WHERE clause.

```
SELECT * FROM employees WHERE department_id IN (10, 20, 30);
```

2. LIKE Operator:

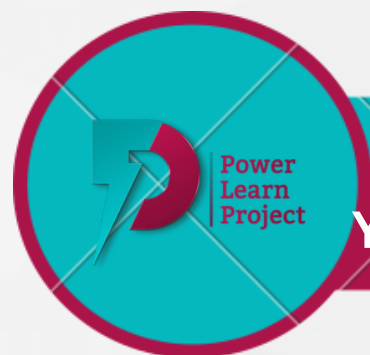
Used for pattern matching in string comparisons.

```
SELECT * FROM customers WHERE customer_name LIKE 'A%';
```

3. BETWEEN Operator:

Filters results within a specified range.

```
SELECT * FROM orders WHERE order_date BETWEEN '2023-01-01' AND '2023-12-31';
```



4. IS NULL / IS NOT NULL:

Filters results where a column is (not) set to NULL.

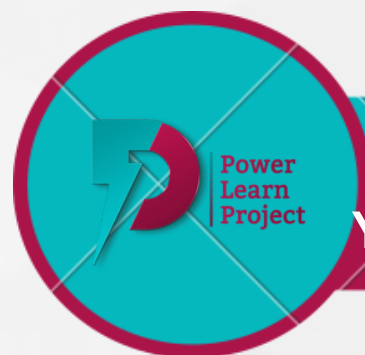
```
SELECT * FROM products WHERE expiration_date IS NULL;
```

IV. Combining Conditions

1. Using Parentheses:

Helps to control the order of evaluation when combining multiple conditions.

```
SELECT * FROM employees WHERE (salary > 50000 AND department_id = 20) OR  
(salary >  
60000 AND department_id = 30);;
```



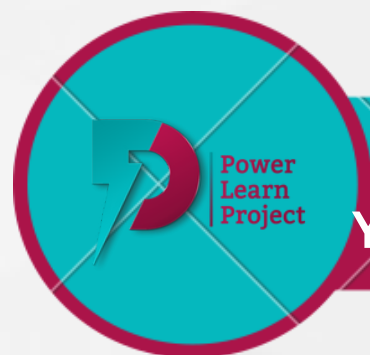
V. Conclusion

Effectively filtering data is crucial for obtaining meaningful insights from your MySQL databases.

The WHERE clause, along with various operators and additional filtering techniques, allows you

to tailor your queries to specific criteria, ensuring that the retrieved data meets your requirements. By mastering these filtering techniques, you enhance your ability to extract

relevant information from large datasets efficiently.



Database Design

You are live with Stanley Munga