**Day 1-2: Deep Dive into Filtering & Sorting**

**Subtopic 1: Basic Selection and Filtering (SELECT, WHERE, ORDER BY)**

* **Definition:** These are the foundational clauses of a query. SELECT specifies the columns you want to see, WHERE filters the rows based on a condition, and ORDER BY sorts the final result.
* **Use/Importance:** This combination is the most fundamental way to retrieve specific, organized data from a database. It allows you to ask targeted questions like "Show me all employees in the Sales department, sorted by their last name."
* **Syntax/Structure:**

SELECT column1, column2

FROM table\_name

WHERE condition

ORDER BY column\_name [ASC|DESC];

* **Example:** To get the names and salaries of employees in the 'Engineering' department, sorted from highest to lowest salary.

SELECT first\_name, last\_name, salary

FROM employees

WHERE department = 'Engineering'

ORDER BY salary DESC;

**Subtopic 2: The IN and NOT IN Operators**

* **Definition:** The IN operator is a shorthand for multiple OR conditions. It checks if a column's value matches any value in a provided list. NOT IN does the opposite.
* **Use/Importance:** It drastically simplifies queries when you need to filter by a specific set of items, making the code cleaner and more readable than a long chain of OR statements.
* **Syntax/Structure:**

WHERE column\_name IN (value1, value2, ...);

* **Example:** To find all employees who are in the 'Sales', 'Marketing', or 'Finance' departments.

SELECT employee\_id, first\_name, department

FROM employees

WHERE department IN ('Sales', 'Marketing', 'Finance');

**Subtopic 3: The BETWEEN Operator**

* **Definition:** The BETWEEN operator selects values within a given range. The range is inclusive, meaning it includes the start and end values.
* **Use/Importance:** It's perfect for filtering data based on a continuous range, such as dates, prices, or ages. It's more concise and often more efficient than using AND with >= and <= operators.
* **Syntax/Structure:**

WHERE column\_name BETWEEN start\_value AND end\_value;

* **Example:** To find all orders placed between January 1, 2025, and January 31, 2025.

SELECT order\_id, order\_date, total\_amount

FROM orders

WHERE order\_date BETWEEN '2025-01-01' AND '2025-01-31';

**Subtopic 4: The LIKE Operator and Wildcards (%, \_)**

* **Definition:** The LIKE operator is used in a WHERE clause to search for a specified pattern in a text column. It uses two main wildcards:
  + %: Represents zero, one, or multiple characters.
  + \_: Represents a single character.
* **Use/Importance:** This is essential for performing "fuzzy" searches on text data when you don't know the exact value. It's the backbone of many search functionalities.
* **Syntax/Structure:**

SQL

WHERE column\_name LIKE 'pattern%';

* **Example:** To find all customers whose last name starts with 'S'.

SELECT first\_name, last\_name

FROM customers

WHERE last\_name LIKE 'S%';

* **Example 2:** To find a 4-letter product name that starts with 'C' and ends with 'e'.

SELECT product\_name

FROM products

WHERE product\_name LIKE 'C\_\_e';

**Subtopic 5: Handling NULL Values**

* **Definition:** A NULL value in a database represents a missing or unknown value. You cannot use standard comparison operators like = or != to find them. Instead, you must use IS NULL or IS NOT NULL.
* **Use/Importance:** Real-world data is often incomplete. Correctly identifying rows with or without missing data is crucial for accurate analysis and data cleaning.
* **Syntax/Structure:**

WHERE column\_name IS NULL;

* **Example:** To find all employees who have not been assigned a manager.

SELECT employee\_id, first\_name

FROM employees

WHERE manager\_id IS NULL;

**Quick Recap:**

* **WHERE** is used to filter rows based on one or more conditions.
* **ORDER BY** sorts the final output in ascending (ASC) or descending (DESC) order.
* Use **IN** to check if a value exists within a list of options.
* Use **BETWEEN** to select records that fall within an inclusive range (dates or numbers).
* Use **LIKE** with wildcards (%, \_) to search for patterns in text.
* Always use **IS NULL** or **IS NOT NULL** to specifically check for missing values.

**Practice Tasks:**

1. **Task 1:** From a products table, write a query to find all products where the product\_name contains the word 'classic'.
2. **Task 2:** From an orders table, select all orders placed in the last quarter (e.g., between '2025-07-01' and '2025-09-30').
3. **Task 3:** From a customers table, write a query to find all customers who do not have a phone number listed (phone\_number is NULL).
4. **Task 4:** Find all employees with the job title of 'Sales Rep' or 'Marketing Coord' and order them by their hire date, starting with the most recent.