✏️

Basic SQL

SQL SELECT

We use SQL SELECT queries to exactly tell the database what data we need to fetch and display.

The most basic SELECT Query contains two essential SQL keywords : SELECT followed by FROM .

```
SELECT column1, column2, ...
FROM table_name;
```

```
SELECT review_id, submit_date, stars
FROM reviews;
```

SQL WHERE

The SQL WHERE clause can be combined with the SELECT statement to output only certain rows, based on if they meet certain conditions:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

```
SELECT *
FROM reviews
WHERE stars <4;
```

Here “*” this means select all from table.

For multiple condition

```
SELECT *
FROM reviews
WHERE stars < 4
  AND user_id = 362;
```

All Logic Operators are valid here . We can also write ≠ as <> both are same which means Not equal.

SQL AND Operator

```
SELECT *
FROM reviews
WHERE stars > 3
AND stars < 5
AND product_id != 50001;
```

SQL OR Operator

```
SELECT *
FROM reviews
WHERE (stars = 3 OR stars = 4)
AND review_id > 5000;
```

SQL NOT Operator

```
SELECT * FROM reviews
WHERE NOT rating = 5;
```

SQL BETWEEN

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

SQL IN

The IN operator allows us to specify multiple values in a single line’s WHERE clause , instead of the more tedious approach of having to use multiple OR condition to filter for multiple values.

```
SELECT ...
FROM ...
WHERE column IN (....);
```

SQL LIKE

It allows you to filter rows based on whether a string matches a certain pattern . For ex. find me all medicines whose name starts with “A and end in Y” into the snippet drug LIKE ‘A%Y’

```
SELECT ...
FROM ...
WHERE column LIKE ...
  AND/OR column NOT LIKE ...;
```

```
SELECT product_id,
manufacturer,
drug
FROM pharmacy_sales
WHERE drug LIKE '%Relief%'
```

All codes are just to see syntax.

5 Ways To Filter Data with WHERE in SQL

- how `WHERE` allows us to filter rows based on specified conditions
- how `AND and OR` allow you to combine multiple filtering conditions
- how `BETWEEN` allows you to filter on a range of values
- how `IN` allows you to specify a list of values that you'd like to filter on
- how `LIKE` allows you to match a value against a pattern

SQL ORDER BY

It allows you to reorder your result based on the data in one or more columns.

```
SELECT column1, column2
FROM table_name
WHERE condition(s)
ORDER BY column;
```

For ex. Imagine you had a table of medicines, and how well they sold at CVS Pharmacy. Running `SELECT * FROM pharmacy_sales` would result in this un-ordered mess:

We could sort this alphabetically based on the drug's name, using `ORDER BY`:

```
SELECT product_id, drug, units_sold
FROM pharmacy_sales
ORDER BY drug;
```

If we want our data in descending Order

```
SELECT column1, column2
FROM table_name
WHERE condition(s)
ORDER BY column DESC;
```