Basic SQL Commands

| **Statement** | **How to Use It** | **Other Details** |
| --- | --- | --- |
| SELECT | SELECT **Col1**, **Col2**, … | Provide the columns you want |
| FROM | FROM **Table** | Provide the table where the columns exist |
| LIMIT | LIMIT \*\*10 \*\* | Limits based number of rows returned |
| ORDER BY | ORDER BY **Col** | Orders table based on the column. Used with **DESC**. |
| WHERE | WHERE **Col > 5** | A conditional statement to filter your results |
| LIKE | WHERE **Col LIKE '%me%'** | Only pulls rows where column has 'me' within the text |
| IN | WHERE **Col IN ('Y', 'N')** | A filter for only rows with column of 'Y' or 'N' |
| NOT | WHERE **Col NOT IN ('Y', 'N')** | **NOT** is frequently used with **LIKE** and **IN** |
| AND | WHERE \*\*Col1 > 5 AND Col2 < 3 \*\* | Filter rows where two or more conditions must be true |
| OR | WHERE **Col1 > 5 OR Col2 < 3** | Filter rows where at least one condition must be true |
| BETWEEN | WHERE **Col BETWEEN 3 AND 5** | Often easier syntax than using an **AND** |

**DISTINCT** is always used in **SELECT** statements, and it provides the unique rows for all columns written in the **SELECT** statement. Therefore, you only use **DISTINCT** once in any particular **SELECT** statement.

**HAVING** is the “clean” way to filter a query that has been aggregated, but this is also commonly done using a [subquery](https://community.modeanalytics.com/sql/tutorial/sql-subqueries/). Essentially, any time you want to perform a **WHERE** on an element of your query that was created by an aggregate, you need to use **HAVING** instead.

**DATE\_TRUNC** allows you to truncate your date to a particular part of your date-time column. Common truncations are **day, month,** and**year.**

You can reference the columns in your select statement in **GROUP BY** and **ORDER BY** clauses with numbers that follow the order they appear in the select statement. For example

**SELECT** standard\_qty, COUNT(\*)

**FROM** orders

**GROUP BY** 1 (this 1 refers to standard\_qty since it is the first of the columns included in the select statement)

**ORDER BY** 1 (this 1 refers to standard\_qty since it is the first of the columns included in the select statement)

The CASE statement always goes in the **SELECT** clause.

1. **CASE** must include the following components: **WHEN**, **THEN**, and **END**. **ELSE** is an optional component to catch cases that didn’t meet any of the other previous **CASE** conditions.
2. You can make any conditional statement using any conditional operator (like [**WHERE**](https://community.modeanalytics.com/sql/tutorial/sql-where/)) between **WHEN** and **THEN**. This includes stringing together multiple conditional statements using **AND** and **OR**.
3. You can include multiple **WHEN** statements, as well as an **ELSE** statement again, to deal with any unaddressed conditions.