**Basic Commands in SQL:**

|  |  |  |
| --- | --- | --- |
| Command | Syntax | Description |
| **ALTER table** | ALTER TABLE table\_name ADD column\_name datatype; | It is used to add columns to a table in a database |
| **AND** | SELECT column\_name(s)FROM table\_nameWHERE column\_1 = value\_1  AND column\_2 = value\_2; | It is an operator that is used to combine two conditions |
| **AS** | SELECT column\_name AS ‘Alias’FROM table\_name; | It is a keyword in SQL that is used to rename a column or table using an alias name |
| **AVG** | SELECT AVG(column\_name)FROM table\_name; | It is used to aggregate a numeric column and return its average |
| [**SQL BETWEEN**](https://intellipaat.com/blog/tutorial/sql-tutorial/sql-between/) | SELECT column\_name(s)FROM table\_nameWHERE column\_name BETWEEN value\_1 AND value\_2; | It is an operation used to filter the result within a certain range |
| **CASE** | SELECT column\_name,CASEWHEN condition THEN ‘Result\_1’WHEN condition THEN ‘Result\_2’ELSE ‘Result\_3’ENDFROM table\_name; | It is a statement used to create different outputs inside a SELECT statement |
| **COUNT** | SELECT COUNT(column\_name)FROM table\_name; | It is a function that takes the name of a column as an argument and counts the number of rows when the column is not NULL |
| **Create TABLE** | CREATE TABLE table\_name (  column\_1 datatype,   column\_2 datatype,   column\_3 datatype); | It is used to create a new table in a database and specify the name of the table and columns inside it |
| **DELETE** | DELETE FROM table\_nameWHERE some\_column = some\_value; | It is used to remove the rows from a table |
| **GROUP BY** | SELECT column\_name, COUNT(\*)FROM table\_nameGROUP BY column\_name; | It is a clause in SQL used for [aggregate functions](https://intellipaat.com/blog/tutorial/sql-tutorial/sql-functions/" \t "_blank) in collaboration with the SELECT statement |
| **HAVING** | SELECT column\_name, COUNT(\*)FROM table\_nameGROUP BY column\_nameHAVING COUNT(\*) > value; | It is used in SQL because the WHERE keyword cannot be used in aggregating functions |
| [**INNER JOIN**](https://intellipaat.com/blog/tutorial/sql-tutorial/inner-join-in-sql/) | SELECT column\_name(s)FROM table\_1JOIN table\_2  ON table\_1.column\_name = table\_2.column\_name; | It is used to combine rows from different tables if the Join condition goes TRUE |
| **INSERT** | INSERT INTO table\_name (column\_1, column\_2, column\_3) VALUES (value\_1, ‘value\_2’, value\_3); | It is used to add new rows to a table |
| **IS NULL/ IS NOT NULL** | SELECT column\_name(s)FROM table\_nameWHERE column\_name IS NULL; | It is an operator used with the WHERE clause to check for the empty values |
| [**LIKE**](https://intellipaat.com/blog/tutorial/sql-tutorial/like-and-between-operators-in-sql/) | SELECT column\_name(s)FROM table\_nameWHERE column\_name LIKE pattern; | It is a special operator used with the WHERE clause to search for a specific pattern in a column |
| **LIMIT** | SELECT column\_name(s)FROM table\_nameLIMIT number; | It is a clause to specify the maximum number of rows the result set must have |
| **MAX** | SELECT MAX(column\_name)FROM table\_name; | It is a function that takes a number of columns as an argument and returns the largest value among them |
| **MIN** | SELECT MIN(column\_name)FROM table\_name; | It is a function that takes a number of columns as an argument and returns the smallest value among them |
| **OR** | SELECT column\_nameFROM table\_nameWHERE column\_name = value\_1   OR column\_name = value\_2; | It is an operator that is used to filter the result set to contain only the rows where either condition is TRUE |
| **ORDER BY** | SELECT column\_nameFROM table\_nameORDER BY column\_name ASC | DESC; | It is a clause used to sort the result set by a particular column either numerically or alphabetically |
| [**OUTER JOIN**](https://intellipaat.com/blog/tutorial/sql-tutorial/full-join-sql/) | SELECT column\_name(s)FROM table\_1LEFT JOIN table\_2  ON table\_1.column\_name = table\_2.column\_name; | It issued to combine rows from different tables even if the condition is NOT TRUE |
| **ROUND** | SELECT ROUND(column\_name, integer)FROM table\_name; | It is a function that takes the column name and an integer as an argument and rounds the values in a column to the number of decimal places specified by an integer |
| **SELECT** | SELECT column\_name FROM table\_name; | It is a statement that is used to fetch data from a database |
| [**SELECT DISTINCT**](https://intellipaat.com/blog/tutorial/sql-tutorial/distinct/) | SELECT DISTINCT column\_nameFROM table\_name; | It is used to specify that the statement is a query that returns unique values in specified columns |
| **SUM** | SELECT SUM(column\_name)FROM table\_name; | It is a function used to return the sum of values from a particular column |
| [**UPDATE**](https://intellipaat.com/blog/tutorial/sql-tutorial/update-query/) | UPDATE table\_nameSET some\_column = some\_valueWHERE some\_column = some\_value; | It is used to edit rows in a table |
| [**WHERE**](https://intellipaat.com/blog/tutorial/sql-tutorial/where-clause/) | SELECT column\_name(s)FROM table\_nameWHERE column\_name operator value; | It is a clause used to filter the result set to include the rows where the condition is TRUE |
| **WITH** | WITH temporary\_name AS (SELECT \*FROM table\_name)SELECT \*FROM temporary\_nameWHERE column\_name operator value; | It is used to store the result of a particular query in a temporary table using an alias |

**Commands and syntax for querying data from a single table and multiple tables:**

|  |  |
| --- | --- |
| Single Table | Multiple Table |
| SELECT c1 FROM t To select the data in Column c1 from table t | SELECT c1, c2 FROM t1 INNER JOIN t2 on conditionSelect column c1 and c2 from table t1 and perform an inner join between t1 and t2 |
| SELECT \* FROM t To select all rows and columns from table t | SELECT c1, c2 FROM t1 LEFT JOIN t2 on condition Select column c1 and c2 from table t1 and perform a left join between t1 and t2 |
| SELECT c1 FROM t WHERE c1 = ‘test’ To select data in column c1 from table t, where c1=test | SELECT c1, c2 FROM t1 RIGHT JOIN t2 on condition Select column c1 and c2 from table t1 and perform a right join between t1 and t2 |
| SELECT c1 FROM t ORDER BY c1 ASC (DESC) To select data in column c1 from table t either in ascending or descending order | SELECT c1, c2 FROM t1 FULL OUTER JOIN t2 on condition Select column c1 and c2 from table t1 and perform a full outer join between t1 and t2 |
| SELECT c1 FROM t ORDER BY c1LIMIT n OFFSET offset To skip the offset of rows and return the next n rows | SELECT c1, c2 FROM t1 CROSS JOIN t2 Select column c1 and c2 from table t1 and produce a Cartesian product of rows in a table |
| SELECT c1, aggregate(c2) FROM t GROUP BY c1 To group rows using an aggregate function | SELECT c1, c2 FROM t1, t2Select column c1 and c2 from table t1 and produce a Cartesian product of rows in a table |
| SELECT c1, aggregate(c2) FROM t GROUP BY c1HAVING condition Group rows using an aggregate function and filter these groups using ‘HAVING’ clause | SELECT c1, c2 FROM t1 A INNER JOIN t2 B on condition Select column c1 and c2 from table t1 and join it to itself using INNER JOIN clause |