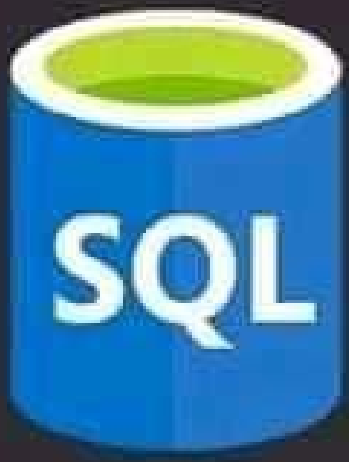
SQL Commands

With Example



Technicabd

CREATE TABLE

CREATE TABLE creates a new table in the database. It allows you to specify the name of the table and the name of each column in the table.

```
CREATE TABLE table_name (
    column_1 datatype,
    column_2 datatype,
    column_3 datatype
);
```

ALTER TABLE

ALTER TABLE lets you add columns to a table in a database.

```
ALTER TABLE table_name
ADD column_name datatype;
```

SQL

Technicabd

INSERT

INSERT statements are used to add a new row to a table.

```
INSERT INTO table_name (column_1,
            column_2, column_3)
VALUES (value_1, 'value_2', value_3);
```

Technicabd

DELETE

DELETE statements are used to remove rows from a table.

```
DELETE FROM table_name
WHERE some_column = some_value;
```

AS

AS is a keyword in SQL that allows you to rename a column or table using an alias.

```
SELECT column_name AS 'Alias'
FROM table_name;
```

SQL

Technicabd

COUNT()

COUNT() is a function that takes the name of a column as an argument and counts the number of rows where the column is not NULL.

```
SELECT COUNT(column_name)
FROM table_name;
```

GROUP BY

GROUP BY is a clause in SQL that is only used with aggregate functions. It is used in collaboration with the SELECT statement to arrange identical data into groups.

```
SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name;
```

HAVING

HAVING was added to SQL because the WHERE keyword could not be used with aggregate functions.

```
SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name
HAVING COUNT(*) > value;
```

AVG()

AVG() is an aggregate function that returns the average value for a numeric column.

```
SELECT AVG(column_name)
FROM table_name;
```

BETWEEN

The BETWEEN operator is used to filter the result set within a certain range. The values can be numbers, text or dates.

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN
      value_1 AND value_2;
```

AND

AND is an operator that combines two conditions. Both conditions must be true for the row to be included in the result set.

```
SELECT column_name(s)
FROM table_name
WHERE column_1 = value_1
  AND column_2 = value_2;
```

SQL

Technicabd

CASE

CASE statements are used to create different outputs (usually in the SELECT statement). It is SQL's way of handling if-then logic.

```
SELECT column_name,
  CASE
    WHEN condition THEN 'Result_1'
    WHEN condition THEN 'Result_2'
    ELSE 'Result_3'
  END
FROM table_name;
```

INNER JOIN

An inner join will combine rows from different tables if the join condition is true.

```
SELECT column_name(s)
FROM table_1
JOIN table_2
  ON table_1.column_name =
     table_2.column_name;
```