**SQL Commands**

Glossary of commonly used SQL commands.

**Background**

SQL, **S**tructured **Q**uery **L**anguage, is a programming language designed to manage data stored in relational databases. SQL operates through simple, declarative statements. This keeps data accurate and secure, and it helps maintain the integrity of databases, regardless of size.

Here’s an appendix of commonly used commands.

**Commands**

**ALTER TABLE**

ALTER TABLE table\_name

ADD column\_name datatype;

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

**AND**

SELECT column\_name(s)

FROM table\_name

WHERE column\_1 = value\_1

AND column\_2 = value\_2;

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

**AS**

SELECT column\_name AS 'Alias'

FROM table\_name;

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

**AVG()**

SELECT AVG(column\_name)

FROM table\_name;

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

**BETWEEN**

SELECT column\_name(s)

FROM table\_name

WHERE column\_name BETWEEN value\_1 AND value\_2;

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

**CASE**

SELECT column\_name,

CASE

WHEN condition THEN 'Result\_1'

WHEN condition THEN 'Result\_2'

ELSE 'Result\_3'

END

FROM table\_name;

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

**COUNT()**

SELECT COUNT(column\_name)

FROM table\_name;

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.

**CREATE TABLE**

CREATE TABLE table\_name (

column\_1 datatype,

column\_2 datatype,

column\_3 datatype

);

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.

**DELETE**

DELETE FROM table\_name

WHERE some\_column = some\_value;

DELETE statements are used to remove rows from a table.

**GROUP BY**

SELECT column\_name, COUNT(\*)

FROM table\_name

GROUP BY column\_name;

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.

**HAVING**

SELECT column\_name, COUNT(\*)

FROM table\_name

GROUP BY column\_name

HAVING COUNT(\*) > value;

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

**INNER JOIN**

SELECT column\_name(s)

FROM table\_1

JOIN table\_2

ON table\_1.column\_name = table\_2.column\_name;

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

**INSERT**

INSERT INTO table\_name (column\_1, column\_2, column\_3)

VALUES (value\_1, 'value\_2', value\_3);

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

**IS NULL / IS NOT NULL**

SELECT column\_name(s)

FROM table\_name

WHERE column\_name IS NULL;

IS NULL and IS NOT NULL are operators used with the WHERE clause to test for empty values.

**LIKE**

SELECT column\_name(s)

FROM table\_name

WHERE column\_name LIKE pattern;

LIKE 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\_name

LIMIT number;

LIMIT is a clause that lets you specify the maximum number of rows the result set will have.

**MAX()**

SELECT MAX(column\_name)

FROM table\_name;

MAX() is a function that takes the name of a column as an argument and returns the largest value in that column.

**MIN()**

SELECT MIN(column\_name)

FROM table\_name;

MIN() is a function that takes the name of a column as an argument and returns the smallest value in that column.

**OR**

SELECT column\_name

FROM table\_name

WHERE column\_name = value\_1

OR column\_name = value\_2;

OR is an operator that filters the result set to only include rows where either condition is true.

**ORDER BY**

SELECT column\_name

FROM table\_name

ORDER BY column\_name ASC | DESC;

ORDER BY is a clause that indicates you want to sort the result set by a particular column either alphabetically or numerically.

**OUTER JOIN**

SELECT column\_name(s)

FROM table\_1

LEFT JOIN table\_2

ON table\_1.column\_name = table\_2.column\_name;

An outer join will combine rows from different tables even if the join condition is not met. Every row in the *left* table is returned in the result set, and if the join condition is not met, then NULL values are used to fill in the columns from the *right* table.

**ROUND()**

SELECT ROUND(column\_name, integer)

FROM table\_name;

ROUND() is a function that takes a column name and an integer as arguments. It rounds the values in the column to the number of decimal places specified by the integer.

**SELECT**

SELECT column\_name

FROM table\_name;

SELECT statements are used to fetch data from a database. Every query will begin with SELECT.

**SELECT DISTINCT**

SELECT DISTINCT column\_name

FROM table\_name;

SELECT DISTINCT specifies that the statement is going to be a query that returns unique values in the specified column(s).

**SUM**

SELECT SUM(column\_name)

FROM table\_name;

SUM() is a function that takes the name of a column as an argument and returns the sum of all the values in that column.

**UPDATE**

UPDATE table\_name

SET some\_column = some\_value

WHERE some\_column = some\_value;

UPDATE statements allow you to edit rows in a table.

**WHERE**

SELECT column\_name(s)

FROM table\_name

WHERE column\_name operator value;

WHERE is a clause that indicates you want to filter the result set to include only rows where the following *condition* is true.

**WITH**

WITH temporary\_name AS (

SELECT \*

FROM table\_name)

SELECT \*

FROM temporary\_name

WHERE column\_name operator value;

WITH clause lets you store the result of a query in a temporary table using an alias. You can also define multiple temporary tables using a comma and with one instance of the WITH keyword.

The WITH clause is also known as common table expression (CTE) and subquery factoring.