

A SQL Cheat Sheet from [OnlineWebApplication.com](https://www.onlinewebapplication.com)

What is SQL?

SQL is used to communicate with a database. It is the standard language for relational database management systems. SQL statements are used to perform tasks such as updating data on a database or retrieving data from a database.

SQL Syntax

SQL is designed to have an English-like syntax to make it easier to write and understand. Here are some basic syntax rules:

- SQL is not case sensitive: SELECT is the same as select.
- Statements can be on one or more lines.
- Keywords cannot be abbreviated or split across lines.
- Clauses must appear in the correct order within a statement.

Basic Commands

- SELECT: Extracts data from a database.
- UPDATE: Updates data in a database.
- DELETE: Deletes data from a database.
- INSERT INTO: Inserts new data into a database.
- CREATE DATABASE: Creates a new database.
- ALTER DATABASE: Modifies a database.
- CREATE TABLE: Creates a new table in the database.
- ALTER TABLE: Modifies a table.
- DROP TABLE: Deletes a table.
- CREATE INDEX: Creates an index (search key).
- DROP INDEX: Deletes an index.

Retrieving Data

Selecting Data

To retrieve data from a database, the SELECT statement is used. Here's how you can use it:

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

To select all columns from a table, use the asterisk (*) symbol:

```
SELECT * FROM table_name;
```

Distinct Values

To select distinct values from a table:

```
SELECT DISTINCT column_name FROM table_name;
```

Where Clause

The WHERE clause is used to filter records:

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

And, Or, Not

- AND and OR can be used to combine multiple conditions:

```
SELECT column1, column2 FROM table_name WHERE condition1 AND condition2;
```

- NOT negates a condition:

```
SELECT column1 FROM table_name WHERE NOT condition;
```

Order By

The ORDER BY keyword is used to sort the result set:

```
SELECT column1, column2 FROM table_name ORDER BY column1 [ASC|DESC], column2 [ASC|DESC];
```

Limit

To limit the number of rows returned:

```
SELECT column1, column2 FROM table_name LIMIT number;
```

Group By

The GROUP BY statement groups rows that have the same values in specified columns into summary rows:

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

Having

The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions:

```
SELECT column_name, COUNT(*) FROM table_name GROUP BY column_name HAVING COUNT(*) > 10;
```

Modifying Data

Inserting Data

To add new rows to a table:

```
INSERT INTO table_name (column1, column2) VALUES (value1, value2);
```

Updating Data

To modify existing records:

```
UPDATE table_name SET column1 = value1 WHERE condition;
```

Deleting Data

To delete records:

```
DELETE FROM table_name WHERE condition;
```

Advanced SQL

Joins

SQL JOIN clause is used to combine rows from two or more tables:

- **INNER JOIN:** Returns records that have matching values in both tables.

```
SELECT columns FROM table1 INNER JOIN table2 ON table1.column_name = table2.column_name;
```

- **LEFT (OUTER) JOIN:** Returns all records from the left table, and the matched records from the right table.

```
SELECT columns FROM table1 LEFT JOIN table2 ON table1.column_name = table2.column_name;
```

- **RIGHT (OUTER) JOIN:** Returns all records from the right table, and the matched records from the left table.

```
SELECT columns FROM table1 RIGHT JOIN table2 ON table1.column_name = table2.column_name;
```

- **FULL (OUTER) JOIN:** Returns all records when there is a match in either left or right table.

```
SELECT columns FROM table1 FULL OUTER JOIN table2 ON table1.column_name = table2.column_name;
```

Subqueries

A subquery is a query nested inside another query. It can be used in SELECT, INSERT, UPDATE, or DELETE statements wherever expressions are allowed:

```
SELECT column_name FROM table_name WHERE column_name IN (SELECT column_name FROM table_name WHERE condition);
```

Indexes

Indexes are special lookup tables that the database search engine can use to speed up data retrieval:

```
CREATE INDEX index_name ON table_name (column1, column2);
```

Transactions

Transactions ensure data integrity and handle errors:

```
BEGIN TRANSACTION;  
INSERT INTO table_name VALUES(value1, value2);  
UPDATE table_name SET column1 = value1 WHERE condition;  
COMMIT;
```

Views

Views are virtual tables based on the result-set of an SQL statement:

```
CREATE VIEW view_name AS  
SELECT column1, column2 FROM table_name WHERE condition;
```

Stored Procedures

Stored procedures are SQL statements saved and used repeatedly:

```
CREATE PROCEDURE procedure_name AS  
sql_statement GO;
```

Triggers

Triggers are automatic actions triggered by database operations:

```
CREATE TRIGGER trigger_name  
BEFORE INSERT ON table_name FOR EACH ROW  
EXECUTE procedure_name;
```