



Tajamul Khan

SQL Cheat Sheet



Select Data



SELECT: Retrieves specific columns from a table.
`SELECT column1, column2 FROM table_name;`

DISTINCT: Removes duplicate rows from the result.
`SELECT DISTINCT column1 FROM table_name;`

WHERE: Filters rows based on a condition.

`SELECT * FROM table_name WHERE column1 = 'v1';`

ORDER BY: Sorts result set by one or more columns.

`SELECT * FROM table_name ORDER BY column1 ASC;`

LIMIT / FETCH: Limits the number of rows returned.

`SELECT * FROM table_name LIMIT 10;`

LIKE: Searches for patterns in text columns.

`SELECT * FROM table_name WHERE col1 LIKE 'A%';`

IN: Filters rows with specific values.

`SELECT * FROM table_name WHERE col1 IN ('v1', 'v2');`

BETWEEN: Filters rows within a range of values.

`SELECT * FROM table_name WHERE column1
BETWEEN 10 AND 20;`



Tajamul Khan



Aggregate Functions



COUNT(): Returns the number of rows.
`SELECT COUNT(*) FROM table_name;`

SUM(): Calculates the sum of a numeric column.
`SELECT SUM(column1) FROM table_name;`

AVG(): Calculates the average of a numeric column.
`SELECT AVG(column1) FROM table_name;`

MIN(): Returns the smallest value in a column.
`SELECT MIN(column1) FROM table_name;`

MAX(): Returns the largest value in a column.
`SELECT MAX(column1) FROM table_name;`

GROUP BY: Groups rows for aggregation.
`SELECT col1, COUNT(*) FROM t1 GROUP BY col1;`

HAVING: Filters grouped rows based on a condition.
`SELECT column1, COUNT(*) FROM t1 GROUP BY column1 HAVING COUNT(*) > 5;`

DISTINCT COUNT(): Counts unique values in column.
`SELECT COUNT(DISTINCT col1) FROM table_name;`



Tajamul Khan



Table Joins



INNER JOIN: Returns rows with matching values in both tables.

```
SELECT * FROM table1 INNER JOIN table2 ON  
table1.id = table2.id;
```

LEFT JOIN: Returns all rows from the left table and matching rows from the right table.

```
SELECT * FROM table1 LEFT JOIN table2 ON table1.id  
= table2.id;
```

RIGHT JOIN: Returns all rows from the right table and matching rows from the left table.

```
SELECT * FROM table1 RIGHT JOIN table2 ON  
table1.id = table2.id;
```

FULL OUTER JOIN: Returns rows when there is a match in either table.

```
SELECT * FROM table1 FULL OUTER JOIN table2 ON  
table1.id = table2.id;
```

CROSS JOIN: Returns the Cartesian product of both tables.

```
SELECT * FROM table1 CROSS JOIN table2;
```

SELF JOIN: Joins a table with itself.

```
SELECT a.column1, b.column1 FROM table_name a,  
table_name b WHERE a.id = b.parent_id;
```



Tajamul Khan



Data Manipulation



INSERT INTO: Adds new rows to a table.

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

UPDATE: Updates existing rows in a table.

`UPDATE table_name SET col1 = 'value' WHERE id = 1;`

DELETE: Removes rows from a table.

`DELETE FROM table_name WHERE column1 = 'value';`

MERGE: Combines INSERT, UPDATE, and DELETE based on a condition.

`MERGE INTO table_name USING source_table ON
condition WHEN MATCHED THEN UPDATE SET
column1 = value WHEN NOT MATCHED THEN INSERT
(columns) VALUES (values);`

TRUNCATE: Removes all rows from a table without logging.

`TRUNCATE TABLE table_name;`

REPLACE: Deletes existing rows and inserts new rows (MySQL-specific).

`REPLACE INTO table_name VALUES (value1, value2);`



Tajamul Khan



Data Definition



CREATE TABLE: Creates a new table.

`CREATE TABLE table_name (id INT PRIMARY KEY,
name VARCHAR(50));`

ALTER TABLE: Modifies an existing table.

`ALTER TABLE table_name ADD column2 INT;`

DROP TABLE: Deletes a table.

`DROP TABLE table_name;`

CREATE INDEX: Creates an index on a table.

`CREATE INDEX idx_name ON table_name (column1);`

DROP INDEX: Removes an index.

`DROP INDEX idx_name ON table_name;`

CREATE VIEW: Creates virtual table based on query.

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

DROP VIEW: Deletes a view.

`DROP VIEW view_name;`

RENAME TABLE: Renames an existing table.

`RENAME TABLE old_table_name TO new_table_name;`



Tajamul Khan



Important Functions



CONCAT(): Concatenates strings.

`SELECT CONCAT(first_name, '', last_name) FROM table_name;`

SUBSTRING(): Extracts a substring from a string.

`SELECT SUBSTRING(column1, 1, 5) FROM table_name;`

LENGTH(): Returns the length of a string.

`SELECT LENGTH(column1) FROM table_name;`

ROUND(): Rounds a number to a specified number of decimal places.

`SELECT ROUND(column1, 2) FROM table_name;`

NOW(): Returns the current timestamp.

`SELECT NOW();`

DATE_ADD(): Adds a time interval to a date.

`SELECT DATE_ADD(NOW(), INTERVAL 7 DAY);`

COALESCE(): Returns the first non-null value.

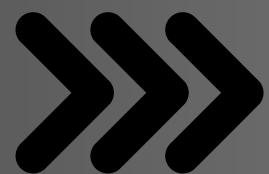
`SELECT COALESCE(column1, column2) FROM table_name;`

IFNULL(): Replaces NULL values with specified value.

`SELECT IFNULL(col1, 'default') FROM table_name;`



Tajamul Khan



**Found
Helpful ?**

Re~~p~~ost



Follow for more!