

MySQL Cheat Sheet

www.databasestar.com

SELECT Query

SELECT col1, col2 FROM table JOIN table2 ON table1.col = table2.col WHERE condition GROUP BY column_name HAVING condition ORDER BY col1 ASCIDESC;

SELECT Keywords

DISTINCT: Removes SELECT DISTINCT product_name duplicate results FROM product;

BETWEEN: Matches a SELECT product_name

FROM product value between two

WHERE price BETWEEN 50 AND 100; other values (inclusive)

SELECT product_name IN: Matches to any of FROM product the values in a list

WHERE category IN

('Electronics', 'Furniture');

LIKE: Performs

SELECT product_name wildcard matches using FROM product WHERE product_name _ or % LIKE '%Desk%";

Joins

SELECT t1.*, t2.* FROM t1 join_type t2 ON t1.col = t2.col;

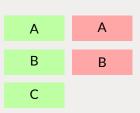
Table 1 Table 2 Α В

INNER JOIN: show all matching records in both tables.

LEFT JOIN: show all records from left table, and any matching records from right table.

RIGHT JOIN: show all records from right table, and any matching records from left table.

FULL JOIN: show all records from both tables, whether there is a match or not.



D

CASE Statement

CASE name Simple Case

> WHEN 'John' THEN 'Name John' WHEN 'Steve' THEN 'Name Steve'

ELSE 'Unknown'

END

CASE Searched Case

WHEN name='John' THEN 'Name John' WHEN name='Steve' THEN 'Name Steve' ELSE 'Unknown' **END**

Common Table Expression

WITH queryname AS (SELECT col1, col2 FROM firsttable) SELECT col1, col2.. FROM queryname...;

Modifying Data

INSERT INTO tablename Insert (col1, col2...) VALUES (val1, val2);

Insert from a INSERT INTO tablename Table (col1, col2...) SELECT col1, col2...

Insert Multiple INSERT INTO tablename (col1, Rows co12...) **VALUES**

> (valA1, valB1), (valA2, valB2), (valA3, valB3);

Update UPDATE tablename SET col1 = val1

WHERE condition;

Update with UPDATE t

a Join SET col1 = val1FROM tablename t INNER JOIN table x

ON t.id = x.tidWHERE condition;

DELETE FROM tablename Delete WHERE condition;

Indexes

CREATE INDEX indexname Create Index ON tablename (cols);

Drop Index DROP INDEX indexname;

Set Operators

UNION: Shows unique rows from two result sets.



UNION ALL: Shows all rows from two result sets.

exist in both result sets.



MINUS: Shows rows that exist in the first result set but not the second.

INTERSECT: Shows rows that

Aggregate Functions

- SUM: Finds a total of the numbers provided
- COUNT: Finds the number of records
- AVG: Finds the average of the numbers provided • MIN: Finds the lowest of the numbers provided
- MAX: Finds the highest of the numbers provided

Common Functions

- LENGTH(string): Returns the length of the provided string
- INSTR(string, substring): Returns the position of the substring within the specified string.
- CAST(expression AS datatype): Converts an expression into the specified data type.
- ADDDATE(input_date, days): Adds a number of days to a specified date.
- NOW: Returns the current date, including time.
- CEILING(input_val): Returns the smallest integer greater than the provided number.
- FLOOR(input_val): Returns the largest integer less than the provided number.
- ROUND(input_val, [round_to]): Rounds a number to a specified number of decimal places.
- TRUNCATE(input_value, num_decimals): Truncates a number to a number of decimals.
- REPLACE(whole_string, string_to_replace, replacement_string): Replaces one string inside the whole string with another string.
- SUBSTRING(string, start_position): Returns part of a value, based on a position and length.

Create Table

```
Create Table
                CREATE TABLE tablename (
                  column_name data_type
```

Create Table with Constraints

```
CREATE TABLE tablename (
  column_name data_type NOT NULL,
  CONSTRAINT pkname PRIMARY KEY (col),
  CONSTRAINT fkname FOREIGN KEY (col)
REFERENCES other_table(col_in_other_table),
  CONSTRAINT ucname UNIQUE (col),
  CONSTRAINT ckname CHECK (conditions)
);
```

Create Temporary CREATE TEMPORARY TABLE

Table tablename (colname datatype);

DROP TABLE tablename; Drop Table

Alter Table

DROP COLUMN columnname;

ALTER TABLE tablename Add Column ADD columnname datatype;

ALTER TABLE tablename Drop Column

Modify Column

columnname newcolumnname newdatatype;

ALTER TABLE tablename CHANGE

ALTER TABLE tablename CHANGE Rename Column COLUMN currentname TO newname;

Add Constraint ALTER TABLE tablename ADD CONSTRAINT constraintname

constrainttype (columns);

ALTER TABLE tablename DROP **Drop Constraint** constraint_type constraintname;

ALTER TABLE tablename Rename Table RENAME TO newtablename;

Window/Analytic Functions

```
function_name ( arguments ) OVER (
[query_partition_clause]
[ORDER BY order_by_clause
[windowing_clause] ] )
```

Example using RANK, showing the student details and their rank according to the fees_paid, grouped by gender:

```
SELECT
student_id, first_name, last_name, gender, fees_paid,
RANK() OVER (
  PARTITION BY gender ORDER BY fees_paid
) AS rank_val
FROM student;
```

Subqueries

```
SELECT id, last_name, salary
Single Row
                FROM employee
                WHERE salary = (
                  SELECT MAX(salary)
                  FROM employee
                );
                SELECT id, last_name, salary
Multi Row
                FROM employee
                WHERE salary IN (
                  SELECT salary
                  FROM employee
                  WHERE last_name LIKE 'C%'
                );
```