

finger↑tips

Data Intelligence Solutions

SQL Cheat sheet

QUERYING DATA FROM A TABLE

Select

Query all rows and columns from a table

```
SELECT * FROM table1;
```

Query data in columns col1, col2 from a table

```
SELECT col1, col2 FROM table1;
```

Alias

For rename a columns name

```
SELECT col1 as Newname from table1;
```

Where

filter the rows by given condition

```
SELECT col1, col2 from table1 WHERE col1=1 and col2=4
```

Distinct

Query will return all unique values from column

```
SELECT DISTINCT col1 from table1;
```

Like

used in a WHERE clause to search for a specified pattern in a column.

```
SELECT col1, col2, FROM table1 WHERE col1 LIKE 's%';
```

Between

Filter result between given range

```
SELECT * from table1 where col1 between value1 and value2;
```

Limit

Limiting amount of result

```
SELECT * from table1 LIMIT 5;
```

```
select * from worker where limit 0,2;
```

Even or ODD ROWS

For even Rows

```
SELECT * from table1 where MOD(col1,2) = 0;
```

For odd Rows

```
SELECT * from table1 where MOD(col1,2) = 1;
```

IN & Not in

To get Record having any of the given values

```
SELECT * from table1 where col1 IN (val1,val2,val3)
```

To get Record without any of the given values

```
SELECT * from table1 where col1 NOT IN (val1,val2,val3)
```

STRING FUNCTION

Substring

Used for a show substring from original string

```
SELECT SUBSTRING('HELLO',1,2)
```

INSTR

returns the position of the first occurrence of a string in another string.

```
SELECT INSTR('FINGERTIPS','FIN')
```

TRIM

TRIM is used to remove space at beginning or end of selection

```
SELECT LTRIM(' hello '); (left trim)
```

```
SELECT RTRIM(' hello '); (right trim)
```

```
SELECT LTRIM(RTRIM(' hello '));
```

Length

LEN is used to find character length in column

```
SELECT LEN(col1) from table1;
```

Replace

For replace string with original string

```
SELECT REPLACE('peter steve Tom', 'steve','Billy')
```

CONCAT

For concatenate two columns or 2 string

```
SELECT CONCAT(col1,col2) from table1
```

GROUP BY

Result of select query can be grouped by columns using GROUP BY

```
SELECT col1, col2 from table1 GROUP BY col1;
```

Having

Filter groups using HAVING clause

```
SELECT col1, col2 FROM table1 GROUP BY col1 HAVING condition;
```

ORDER BY

Sort a table in ascending & descending order

```
SELECT * from table1 ORDER BY col1;
```

```
SELECT * from table1 ORDER BY col1 desc;
```

Group by with having & order by

```
SELECT col1, col2 FROM table1 GROUP BY col1 HAVING condition order by col1;
```

Subquery

Use a subquery to filter the result set

```
Select * from table1 where col1 = (select max(col1) from table1);
```

Current Date and Time

Show The Current Date and Time

```
SELECT CURDATE();  
SELECT NOW();
```

Statistics

```
select max(col1) from table1;
```

```
select min(col1) from table1;
```

```
select avg(col1) from table1;
```

```
select stddev_pop(col1) from table1;
```

```
select var_pop(col1) from table1;
```

Aggregate

```
SELECT COUNT(col1) FROM table1;
```

```
SELECT AVG (col1) FROM table1;
```

```
SELECT SUM (col1) FROM table1;
```

JOINS

Inner join table1 and table 2

```
SELECT table1.col1, table2.col2 FROM table1 INNER JOIN  
table2 ON table1.col1 = table2.col2;
```

Left join table1 and table 2

```
SELECT table1.col1, table2.col2 FROM table1 LEFT JOIN  
table2 ON table1.col1 = table2.col2;
```

Right join table1 and table 2

```
SELECT table1.col1, table2.col2 FROM table1 RIGHT JOIN  
table2 ON table1.col1 = table2.col2;
```

Full Outer join table1 and table 2

```
SELECT table1.col1, table2.col2 FROM table1 FULL  
OUTER JOIN table2 ON table1.col1 = table2.col2;
```

Multiple Joins table1, table2 and table3

```
SELECT table1.col1, table2.col2, table3.col3 FROM  
table1 INNER JOIN table2 ON table1.col1 = table2.col2  
LEFT JOIN table3 ON table1.col1 = table3.col3;
```

SQL OPERATORS

UNION

Combine rows from two queries

Union

```
select * from Worker union select * from Bonus;
```

Union All

```
SELECT col1, col2 FROM table1 UNION [ALL] SELECT col1,  
col2 FROM table2;
```

INTERSECTION

Return the intersection of two queries

```
SELECT col1, col2 FROM table1 INTERSECT SELECT col1,  
col2 FROM table2;
```

MINUS

Subtract a result set from another result set

```
SELECT col1, col2 FROM table1 MINUS SELECT col1, col2  
FROM table2;
```

Copy

Clone or copy table

Create table table2 Select * from table1;

IF CLAUSE

Return "A" if the condition is TRUE, or "B" if the condition is FALSE:

```
select *, if(col1 > 100000, "A","B") as Newname from  
table1;
```

Delete the table from the database

```
DROP TABLE table1;
```

Add a new column to the table

```
ALTER TABLE table1 ADD column;
```

Drop column c from the table

```
ALTER TABLE table1 DROP COLUMN col1 ;
```

Remove all data in a table

```
TRUNCATE TABLE table1;
```

Update new value in the column c1 for all rows

```
UPDATE table1 SET col1 = new_value;
```

Delete all data in a table

```
DELETE FROM table1;
```

SQL Constraints

NOT NULL - Ensures that a column cannot have a NULL value

UNIQUE - Ensures that all values in a column are different

PRIMARY KEY - A combination of a NOT NULL and UNIQUE.

Uniquely identifies each row in a table

FOREIGN KEY - Uniquely identifies a row/record in another table

CHECK - Ensures that all values in a column satisfies a specific condition

DEFAULT - Sets a default value for a column when no value is specified

INDEX - Used to create and retrieve data from the database very quickly