

Terminology - Basic Manipulation

<i>SQL</i>	A programming language designed to manipulate & manage data stored in relational databases
<i>relational database</i>	A database that organizes information into one or more tables.
<i>table</i>	A collection of data organized into rows & columns.
<i>statement</i>	A string of characters that the database recognizes as a valid command.
<i>primary key</i>	Column in table that is unique to each row w/ no NULL values.
<i>foreign key</i>	Primary key of table1 that appears in table2.

Commands - Basic Manipulation

SHOW DATABASES	list all available databases
USE database	use specified database
SHOW TABLES [FROM database]	list tables in database
DESCRIBE table	list column headers in table
SHOW FIELDS FROM table	list all fields
SHOW COLUMNS FROM table	list all columns (fields) + column type etc
SHOW COLUMNS FROM table	list all columns (fields) + column type etc
SHOW INDEX FROM table	list all indexes from table

Terminology - queries

operators Operators create a condition that can be evaluated as either *true* or *false*.

Commands - operators

=	equal to
!=	not equal to
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
IS NULL	is null
IS NOT NULL	is not null

Wildcards

*	Matches any number or type of character(s).
_	Matches any individual character.
%	Matches zero or more missing letters in the pattern.

Commands - queries

SELECT	Identify columns to return in query.	SELECT column FROM table;
AS	Renames a column or table using an alias.	SELECT column AS 'alias' FROM table;
DISTINCT	Used to return unique values in the output. Filters out all duplicate values in the specified column(s).	SELECT DISTINCT column FROM table;
LIKE	Operator used with WHERE clause to search for a specific pattern in a column.	WHERE column LIKE 'text'; (or NOT LIKE)
AND	Operator used to combine multiple conditions in a WHERE clause; ALL must be true.	WHERE column condition1 AND column condition2;
OR	Operator used to combine multiple conditions in a WHERE clause; ANY must be true.	WHERE column condition1 OR column condition2;
BETWEEN	Operator used in a WHERE clause to filter the result set within a certain range (numbers, text, or dates).	WHERE column BETWEEN 'A' AND 'B';

BETWEEN two letters is not* inclusive of the 2nd letter.

BETWEEN two numbers is* inclusive of the 2nd number.



Terminology - Aggregate Functions

<i>aggregates</i>	Calculations performed on multiple rows of a table.
<i>aggregate functions</i>	Combine multiple rows together to form a single value of more meaningful information.
<i>clause</i>	A clause is used with aggregate functions; used in collaboration with the SELECT statement.

Commands - Aggregate Functions

COUNT ()	Count the number of rows	SELECT COUNT (column) FROM table ;
SUM ()	The sum of the values in a column	SELECT SUM (column) FROM table ;
MAX () / MIN ()	The largest/smallest value in a column	SELECT MAX (column) FROM table ;
AVG ()	The average (mean) of the values in a column	SELECT AVG (column) FROM table ;
ROUND ()	Round the values in a column	SELECT ROUND (column, integer) FROM table ;

Clauses

1. WHERE	Restrict the results of a query based on values of individual rows within a column.
2. GROUP BY	A clause used with aggregate functions to combine data from one or more columns. Arrange identical data into groups.
3. HAVING	Limit the results of a query based on an aggregate property.
4. ORDER BY	Sort results by column. ORDER BY column ASC/DESC

Clauses (cont)

5. LIMIT Maximum number of rows to return.

ie.

```
SELECT column, AGG(column)
FROM table
CLAUSE column;
```

Clauses can refer to a column name, or to a column reference number (assigned by order column referred to in statement).

If-then - CASE

```
SELECT columns,
CASE
WHEN column condition1 THEN action1
WHEN column condition2 THEN action2
ELSE action3
END AS 'renamed_column'
FROM table;
```

Combining tables - JOIN

JOIN (<i>inner join</i>)	combine rows from different tables if the join condition is true; drops unmatched rows
LEFT JOIN / RIGHT JOIN	return every row in the <i>left/right</i> table; if join condition not met, NULL values used to fill in columns from the <i>right/left</i> table
OUTER JOIN	return unmatched rows from <i>both</i> tables; unmatched fields filled with NULL
CROSS JOIN	combine all rows of 1 table with all rows of another table; does NOT require joining on a specific column



Combining tables - JOIN (cont)

UNION stacks 1 dataset on top of another; tables must have same # columns & same data types/order columns

```
SELECT * FROM
table1 UNION
SELECT * FROM
table2;
```

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

ie.

```
SELECT table1.column1,
COUNT(*) AS renamed_output
FROM table1
CROSS JOIN table2
WHERE table2.column1 <= table1.column1
AND table2.column2 >= table1.column1
GROUP BY table1.column1;
```

Combining tables - WITH statements

FY!! MySQL prior to version 8.0 doesn't support the WITH clause.

```
WITH previousQueryAlias AS (
SELECT column1,
COUNT(column2) AS renamedOutputColumn
FROM table1
GROUP BY column1
)
SELECT table2.column1,
previousQueryAlias.renamedOutputColumn
FROM previousQueryAlias
JOIN table2
ON table2.column1 = previousQueryAlias.column1;
```

Commands - String Functions

STRCMP("string1","string2")	compare strings
LOWER("string")	convert to lower case
UPPER("string")	convert to upper case
LTRIM/RTRIM("string")	left or right trim
SUBSTRING("string","inx1","inx2")	substring of a string
CONCAT("string1","string2")	concatenate



By **sjm**
cheatography.com/sjm/

Published 24th July, 2019.
 Last updated 24th July, 2019.
 Page 3 of 3.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>