Cheatography

SQL commands Cheat Sheet by sjm via cheatography.com/62460/cs/16026/

Terminology - Basic Manipulation

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

C

By **sjm** cheatography.com/sjm/ Published 24th July, 2019. Last updated 24th July, 2019. Page 1 of 3.

Wildcards

*	Matches a	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 condit- ion2;
OR	Operator used to combine multiple conditions in a WHERE clause; ANY must be true.	WHERE column condition1 OR column condit- ion2;
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.

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

Cheatography

SQL commands Cheat Sheet by sjm via cheatography.com/62460/cs/16026/

Terminology - Aggregate Functions	
aggregates	Calculations performed on multiple rows of a table.
aggregate functions	Combine multiple rows together to form a single value of more meaningful information.
clause	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 que property.	uery based on an aggregate
4.	ORDER BY	Sort results by column.	ORDER BY column ASC/DESC

Clauses (cont)

5.	LIMIT	Maximum number of rows to return.
ie.		
SEL	ECT column	, AGG(co lumn)
FROI	M table	
CLA	USE 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

Combining tables - JOIN

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

0	
JOIN <i>(inner</i> <i>join)</i>	combine rows from different tables if the join condition is true; drops unmatched rows
LEFT JOIN	return every row in the <i>left/right</i> table; if join condition
/ RIGHT	not met, $\ensuremath{\operatorname{NULL}}$ values used to fill in columns from the
JOIN	<i>right/left</i> table
OUTER	return unmatched rows from both tables; unmatched
JOIN	fields filled with NULL
CROSS	combine all rows of 1 table with all rows of another
JOIN	table; does NOT require joining on a specific column

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

Published 24th July, 2019. Last updated 24th July, 2019. Page 2 of 3. Sponsored by Readable.com Measure your website readability! https://readable.com

Cheatography

SQL commands Cheat Sheet by sjm via cheatography.com/62460/cs/16026/

Combining tables - JOIN (cont) UNION stacks 1 dataset on top of another; table1 UNION & stacks 1 dataset on top of another; table1 UNION & same data types/order columns & same data types/order columns & select * FROM table1. UNION SELECT * FROM table1 JOIN table1 GENERATION AS renamed_output FROM table1 CROSS JOIN table2 WHERE table2.column1 <= table1.column1 GROUP BY table1.column1; COUNT (*) AS renamed_output FROM table1 CROSS JOIN table2 FYI! MySQL prior to version 8.0 doesn't v					
tables must have same # columns & same data types/order columns 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;FYI! MySQL prior to version 8.0 doesn't super the WITH clause.FYI! MySQL prior to version 8.0 doesn't super the WITH clause.FYI! MySQL prior to version 8.0 doesn't super the WITH clause.FYI! MySQL prior to version 8.0 doesn't super the WITH clause.GROUP BY column1, previousQueryAlias AS (SELECT column1, COUNT (column2) AS renamedOutputColumn FROM table1 GROUP BY column1)SELECT table2.column1, previousQueryAlias.renamedOutputColumn1; previousQueryAlias.renamedOutputColumn1;SELECT table2.column1, previousQueryAlias.renamedOutputColumn1;SELECT table2.column1 (STRCMP ("string1", "string2")Commands - String FunctionsSTRCMP("string1", "string2")GOUP EX column1 (STRCMP ("string1", "inx1")SUBSTRING("string", inx1", inx2")SUBSTRING("string", inx1", inx2")	Combini	Combining tables - JOIN (cont)			
& same data types/order colums 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 GROUP BY table1.column1; TOMDING table2.column2 >= table1.column1 GROUP BY table1.column1; FYI! MySQL prior to version 8.0 doesn't suport the WITH clause. FYI! MySQL prior to version 8.0 doesn't suport the WITH clause. FYI! MySQL prior to version 8.0 doesn't suport the WITH clause. FYI! MySQL prior to version 8.0 doesn't suport the WITH clause. FYI! MySQL prior to version 8.0 doesn't suport the WITH clause. FYI! MySQL prior to version 8.0 doesn't suport the WITH clause. FYI! MySQL prior to version 8.0 doesn't suport the WITH clause. FYI! MySQL prior to version 8.0 doesn't suport the WITH clause. FYI! MySQL prior to version 8.0 doesn't suport the WITH clause. FYI! MySQL prior to version 8.0 doesn't suport the WITH clause. FYI! MySQL prior to version 8.0 doesn't suport the WITH clause. FROM table1 GROUP BY column1, previousQueryAlias.renamedOutputColumn FROM table2 oN table2.column1, previousQueryAlias.renamedOutputColumn FROM previousQueryAlias.renamedOutputColumn FROM previousQueryAlias.renamedOutput column from previousQueryAlias.renamedOu	UNION	stacks 1 dataset on top of anoth	ner; SELECT * FROM		
table2; selleCT * 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 GROUP BY table1.column1; TOTAL table2.column2 >= table1.column1 GROUP BY table1.column1; FYI! MySQL prior to version 8.0 doesn't support the WITH clause. FYI! MySQL prior to version 8.0 doesn't support the WITH clause. FYI! MySQL prior to version 8.0 doesn't support the WITH clause. FYI! MySQL prior to version 8.0 doesn't support the WITH clause. FYI! MySQL prior to version 8.0 doesn't support the WITH clause. FYI! MySQL prior to version 8.0 doesn't support the WITH clause. FYI! MySQL prior to version 8.0 doesn't support the WITH clause. FYI! MySQL prior to version 8.0 doesn't support the WITH clause. FYI! MySQL prior to version 8.0 doesn't support the WITH clause. FYI! MySQL prior to version 8.0 doesn't support the WITH clause. FROM table1 GROUP BY column1, previousQueryAlias AS (SELECT table2.column1, previousQueryAlias.renamedOutputColumn FROM previousQueryAlias.renamedOutputColumn FROM previousQueryAlias.re		tables must have same # colum	ins table1 UNION		
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; FTM 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, previousQueryAlias.renamedOutputColumn FROM table1 GROUP BY column1, count (column1, count (column2) AS renamedOutputColumn FROM table1 GROUP BY column1, count (column1) SELECT table2.column1, count (column2) AS renamedOutputColumn FROM table1 GROUP BY column1, count (column1), count (column1), count (column2) AS renamedOutputColumn FROM table1 GROUP BY column1, count (column1), count (column2) AS renamedOutputColumn FROM table1 GROUP BY column1, count (column1), count (column2) AS renamedOutputColumn FROM table1 GROUP BY column1, from previousQueryAlias.column1;		& same data types/order colum	NS SELECT * FROM		
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 GROUP BY table1.column1; Combining tables - WITH statements FY!! MySQL prior to version 8.0 doesn't suport the WITH clause. WITH previousQueryAlias AS (SELECT column1, COUNT (column2) AS renamedOutputColumn FROM table1 GROUP BY column1, count (column2) AS renamedOutputColumn FROM table1 GROUP BY column1, previousQueryAlias.renamedOutputColumn FROM previousQueryAlias.renamedOutputColumn FROM previousQueryAlias JOIN table2 oN table2.column1 = previousQueryAlias.column1; PreviousQueryAlias.renamedOutputColumn FROM previousQueryAlias JOIN table2 oN table2.column1 = previousQueryAlias.column1 previousQueryAlias JOIN table2 oN table2.column1 = previousQueryAlias.column1; COMPART String1", "string2") convert to lower case <th></th> <th></th> <th>table2;</th>			table2;		
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; COUND table2.column1; FY!! MySQL prior to version 8.0 doesn't suport the WITH clause. WITH previousQueryAlias AS (SELECT column1, COUNT (column2) AS renamedOutputColumn FROM table1 GROUP BY column1, COUNT (column2) AS renamedOutputColumn FROM table1 GROUP BY column1, previousQueryAlias.renamedOutputColumn FROM previousQueryAlias JOIN table2 ON table2.column1 = previousQueryAlias.column1; COUNT table2 MITH previousQueryAlias JOIN table2 ON table2.column1 = freviousQueryAlias.column1; STRCMP("string1", "string2") Compare strings LOWER("string") CONVET to upper case UPPER("string") SUBSTRING("string", "inx1", "inx2") SUBSTRING("string", "inx1", "inx2")	SELECT	*			
ie. SELECT table1.column1, COUNT (*) AS renamed_output FROM table1 CROSS JOIN table2 WHERE table2.column1 <= table1.column1 AND table2.column1 >= table1.column1 GROUP BY table1.column1; FY!! MySQL prior to version 8.0 doesn't suport the WITH clause. FY!! MySQL prior to version 8.0 doesn't suport the WITH clause. WITH previousQueryAlias AS (SELECT column1, COUNT (column2) AS renamedOutputColumn FROM table1 GROUP BY column1, previousQueryAlias.renamedOutputColumn FROM previousQueryAlias.renamedOutputColumn FROM previousQueryAlias.renamedOutputColumn FROM previousQueryAlias.renamedOutputColumn FROM previousQueryAlias.renamedOutputColumn FROM previousQueryAlias.column1; SELECT table2.column1 = previousQueryAlias.column1; FROM rable2 ON table2 ON table2.column1 = previousQueryAlias.column1; EXECMP("string1", "string2") convert to lower case UPPER("string") convert to upper case UPPER("string") kotting ("string") kotting ("stri	FROM ta	ble1			
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 FYI! MySQL prior to version 8.0 doesn't suport the WITH clause. WITH previousQueryAlias AS (SELECT column1, COUNT (column2) AS renamedOutputColumn FROM table1 GROUP BY column1, count (column1, previousQueryAlias.renamedOutputColumn FROM previousQueryAlias JOIN table2 oN table2.column1 = previousQueryAlias.column1; Commands - String Functions STRCMP("string1","string2") compare strings LOWER("string1","string2") convert to lower case UPPER("string") convert to upper case LOWER("string") kusting of a string					
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 FYI! 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 JOIN table2 ON table2.column1 = previousQueryAlias STRCMP("string1", "string2") Compare strings LOWER("string") SUBSTRING("string", "inx1", "inx2") SUBSTRING("string", "inx1", "inx2") SUBSTRING("string", "inx1", "inx2")	on tabl	e1.id = table2.id;			
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 FYI! 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 JOIN table2 ON table2.column1 = previousQueryAlias STRCMP("string1", "string2") Compare strings LOWER("string") SUBSTRING("string", "inx1", "inx2") SUBSTRING("string", "inx1", "inx2") SUBSTRING("string", "inx1", "inx2")	ie.				
FROM table1CROSS JOIN table2WHERE table2.column1 <= table1.column1AND table2.column2 >= table1.column1GROUP BY table1.column1;Combining tables - WITH statementsFYI! 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.renamedOutputColumn1;FROM previousQueryAlias JOIN table2 ON table2.column1 = previousQueryAlias.column1;Commands - String FunctionsSTRCMP("string1", "string2")convert to lower caseUPPER("string")convert to lower caseLOWER("string")left or right trimSUBSTRING("string", "inx1", "inx2")substring of a string		table1.column1,			
CROSS JOIN table2 WHERE table2.column1 <= table1.column1 AND table2.column2 >= table1.column1 GROUP BY table1.column1; Combining tables - WITH statements FYI! 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; 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") left or right trim	COUNT	(*) AS renamed_output			
WHERE table2.column1 <= table1.column1	FROM t	able1			
AND table2.column2 >= table1.column1; GROUP BY table1.column1; Combining tables - WITH statements FYI! 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- STRCMP("string1", "string2") compare strings LOWER("string") convert to upper case UPPER("string") left or right trim SUBSTRING("string", "inx1", "inx2") substring of a string	CROSS	JOIN table2			
GROUP BY table1.column1; Combining tables - WITH statements FYI! 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	WHERE t	able2.column1 <= table1.column	1		
Combining tables - WITH statementsFYI! MySQL prior to version 8.0 doesn't suport the WITH clause.WITH previousQueryAlias AS (SELECT column1, COUNT (column2) AS renamedOutputColumnFROM table1 GROUP BY column1)SELECT table2.column1, previousQueryAlias.renamedOutputColumnFROM previousQueryAlias.renamedOutputColumnFROM previousQueryAlias JOIN table2 ON table2.column1 = previousQueryAlias.column1;Commands - String FunctionsSTRCMP("string1", "string2")convert to lower caseUPPER("string")convert to upper caseLTRIM/RTRIM("string")left or right trimSUBSTRING("string", "inx1", "inx2")substring of a string	AND ta	ble2.column2 >= table1.column1			
FYI! MySQL prior to version 8.0 doesn't support the WITH clause.WITH previousQueryAlias AS (SELECT column1, COUNT (column2) AS renamedOutputColumnFROM table1 GROUP BY column1)SELECT table2.column1, previousQueryAlias.renamedOutputColumnFROM previousQueryAlias. JOIN table2 ON table2.column1 = previousQueryAlias.column1;Commands - String FunctionsSTRCMP("string1", "string2")Convert to lower caseUPPER("string")convert to upper caseLTRIM/RTRIM("string")SUBSTRING("string", "inx1", "inx2")substring of a string	GROUP	BY table1.column1;			
FYI! MySQL prior to version 8.0 doesn't support the WITH clause.WITH previousQueryAlias AS (SELECT column1, COUNT (column2) AS renamedOutputColumnFROM table1 GROUP BY column1)SELECT table2.column1, previousQueryAlias.renamedOutputColumnFROM previousQueryAlias. JOIN table2 ON table2.column1 = previousQueryAlias.column1;Commands - String FunctionsSTRCMP("string1", "string2")Convert to lower caseUPPER("string")convert to upper caseLTRIM/RTRIM("string")SUBSTRING("string", "inx1", "inx2")substring of a string	Combini	ng tables - WITH statements			
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 FunctionsSTRCMP("string1", "string2")Compare strings LOWER("string")Convert to lower case LTRIM/RTRIM("string")SUBSTRING("string", "inx1", "inx2")SUBSTRING("string", "inx1", "inx2")	FYI! My	SQL prior to version 8.0 doesn't s	upport the WITH clause.		
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 FunctionsSTRCMP("string1", "string2")Compare strings LOWER("string")Convert to lower case LTRIM/RTRIM("string")SUBSTRING("string", "inx1", "inx2")SUBSTRING("string", "inx1", "inx2")	WITH pr				
FROM table1GROUP BY column1)SELECT table2.column1, previousQueryAlias.renamedOutputColumnFROM previousQueryAliasJOIN table2oN table2.column1 = previousQueryAlias.column1;Commands - String FunctionsSTRCMP("string1", "string2")compare stringsLOWER("string")convert to lower caseUPPER("string")LTRIM/RTRIM("string")SUBSTRING("string", "inx1", "inx2")substring of a string					
GROUP BY column1)SELECT table2.column1, previousQueryAlias.renamedOutputColumnFROM previousQueryAliasJOIN table2ON table2.column1 = previousQueryAlias.column1;Commands - String FunctionsSTRCMP("string1", "string2")compare stringsLOWER("string")convert to lower caseUPPER("string")LTRIM/RTRIM("string")SUBSTRING("string", "inx1", "inx2")substring of a string	COUNT	(column2) As renamedOutputC	Column		
) 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	FROM t	able1			
SELECT table2.column1, previousQueryAlias.renamedOutputColumnFROM previousQueryAliasJOIN table2 ON table2.column1 = previousQueryAlias.column1;Commands - String FunctionsSTRCMP("string1", "string2")compare stringsLOWER("string")convert to lower caseUPPER("string")LTRIM/RTRIM("string")SUBSTRING("string", "inx1", "inx2")substring of a string	GROUP	BY column1			
previousQueryAlias.renamedOutputColumnFROM previousQueryAliasJOIN table2oN table2.column1 = previousQueryAlias.column1;Commands - String FunctionsSTRCMP("string1","string2")compare stringsLOWER("string1","string2")Convert to lower caseUPPER("string")LTRIM/RTRIM("string")SUBSTRING("string", "inx1", "inx2")Substring of a string))			
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") LTRIM/RTRIM("string") SUBSTRING("string", "inx1", "inx2")	SELECT	SELECT table2.column1,			
JOIN table2ON table2.column1 = previousQueryAlias.column1;Commands - String FunctionsSTRCMP("string1", "string2")compare stringsLOWER("string")convert to lower caseUPPER("string")convert to upper caseLTRIM/RTRIM("string")left or right trimSUBSTRING("string", "inx1", "inx2")substring of a string					
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					
Commands - String FunctionsSTRCMP("string1", "string2")compare stringsLOWER("string")convert to lower caseUPPER("string")convert to upper caseLTRIM/RTRIM("string")left or right trimSUBSTRING("string", "inx1", "inx2")substring of a string					
STRCMP ("string1", "string2")compare stringsLOWER ("string")convert to lower caseUPPER ("string")convert to upper caseLTRIM/RTRIM ("string")left or right trimSUBSTRING ("string", "inx1", "inx2")substring of a string			s.columnt,		
LOWER("string")convert to lower caseUPPER("string")convert to upper caseLTRIM/RTRIM("string")left or right trimSUBSTRING("string", "inx1", "inx2")substring of a string	Comma	nds - String Functions			
UPPER("string")convert to upper caseLTRIM/RTRIM("string")left or right trimSUBSTRING("string", "inx1", "inx2")substring of a string	STRCMP	("string1","string2")	compare strings		
LTRIM/RTRIM("string") left or right trim SUBSTRING("string", "inx1", "inx2") substring of a string	LOWER('string")	convert to lower case		
SUBSTRING("string","inx1","inx2") substring of a string	UPPER('string")	convert to upper case		
	LTRIM/H	RTRIM("string")	left or right trim		
CONCAT ("string1", "string2") concatenate	SUBSTR	ING("string","inx1","inx2")	substring of a string		
	CONCAT	("string1","string2")	concatenate		

C

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