

Functions	
bin(4)	100. Converts a decimal number to a binary number
CASE WHEN Quantity > 30 THEN "The quantity is greater than 30" WHEN Quantity = 30 THEN "The quantity is 30" ELSE "The quantity is something else" END	can be used in SELECT,DELETE,INSERT,UPDATE
SELECT CASE city WHEN 'calgary' THEN 'Y' ELSE 'N' END calgary FROM...	
CAST("14:06:10" AS TIME)	CAST(value AS type)
SELECT COALESCE(NULL, 1, 2, 'W3Schools.com');	1. Return the first non-null expression in a list
IF(500<1000, "YES", "NO")	IF(condition, value_if_true, value_if_false)
IFNULL(NULL, 500)	IFNULL(expression, alt_value)

NUMERIC	
ABS()	
AVG()	
CEIL()	
FLOOR()	
LOG() LOG10() LOG2()	
PI	

NUMERIC (cont)	
POWER(3,2) or POW(3,2)	9
RAND()	a value between 0 (inclusive) and 1 (exclusive)
RAND(4)	RAND(seed), a repeatable sequence of random numbers if a seed value is used
ROUND(-135.375, 2)	135.38, ROUND(number, decimal_places)
ROUND(-135.375)	135
SIGN(number)	if >0, 1. If = 0, 0. If <0, -1
SQRT(64)	

Join	
SELECT x.a, b, c FROM x INNER JOIN y ON x.id=y.id	
SELECT o.a, c.b, e.d FROM ((orders o INNER JOIN customers c ON o.id=c.id) INNER JOIN employee e ON e.id=o.eid);	
LEFT JOIN	
RIGHT JOIN	
FULL OUTER JOIN	
(SELECT a FROM aa) UNION (SELECT a FROM bb)	# of columns, and data types the same and in the same order
SELECT a,b FROM aa,bb	cartesian join, return table of length(aa*bb)
UNION ALL	Union all will not eliminate duplicate rows

String	
ASCII(character)	number code of the first character
CONCAT("SQL ", "is ", "fun!")	Concatenate several expressions together
CONCAT_WS("-", "SQL ", "is ", "fun!")	SQL-is-fun!
FIELD("c", "a", "b", "c", "d", "e")	3. FIELD(value, val1, val2, val3, ...). position of the value in the list of values
FIND_IN_SET("c", "a,b,c,d,e")	3. the position of a string in a comma-separated string list. not found=0
INSERT("W3Schools.com", 1, 9, "Example")	Example.com. INSERT(string, position, number, substring)
INSTR("W3Schools.com", "COM")	11. return position
LEFT(string, number_of_chars)	extracts a substring from a string (starting from left).
LENGTH(string)	
LOWER(string)	
LPAD("SQL", 10, "ABC")	ABCABCASQL. LPAD(string, length, pad_string)
LTRIM(" SQL Tutorial")	Remove leading spaces from a string



String (cont)	
MID("SQL Tutorial", 5, 3) or SUBSTR(string, start, length)	Tut. MID(string, start, length)
POSITION("3" IN "W3Schools.com") or LOCATE("3", "W3Schools.com")	2
REPEAT(string, number)	
REPLACE("SQL Tutorial", "SQL", "HTML")	REPLACE(string, from_substring, to_substring)
REVERSE(string)	
RIGHT("SQL Tutorial is cool", 4)	cool
RPAD("SQL Tutorial", 20, "ABC")	
RTRIM("ABC ")	
SPACE(10)	Return a string with 10 spaces
UPPER()	
TRIM()	Removes leading and trailing spaces from a string
STRCMP(string1, string2)	If string1=string2, 0 If string1<string2, -1 If string1>string2, 1
SUBSTRING_INDEX("www.w3schools.com", ".", 2)	www.w3schools, substring of string before number of occurrences of delimiter

Date	
ADDDATE("2017-06-15", INTERVAL -10 DAY) or DATE_ADD(...)	ADDDATE(date, INTERVAL value unit) [SECOND MINUTE HOUR DAY WEEK MONTH QUARTER YEAR ...]
ADDDATE("2017-06-15", 1)	ADDDATE(date, days)
CURDATE() or CURRENT_DATE()	current date as a "YYYY-MM-DD" or YYYYMMDD
CURDATE() + 1	Tomorrow
CURTIME() or CURRENT_TIME()	current time as a "HH-MM-SS" or HHMMSS
CURRENT_TIMESTAMP()	"YYYY-MM-DD HH-MM-SS"
DATE("2017-06-15 09:34:21")	extracts the date value from a date or datetime expression
DATEDIFF("2017-06-25 09:34:21", "2017-06-15 15:25:35")	the difference in days between two date values
DATE_FORMAT(BirthDate, "%W %M %e %Y")	%W Weekday name in full, %e Day of the month as a numeric value (0 to 31)
DAY("2017-06-15"), DAYOFMONTH	15.DAY(date)
DAYNAME(date)	
DAYOFWEEK("2017-06-15")	1=Sunday, ..., 7=Saturday.
DAYOFYEAR("2017-06-15")	a number from 1 to 366

Date (cont)	
EXTRACT(WEEK FROM "2017-06-15")	EXTRACT(unit FROM date)
YEAR(date)	
WEEK(date)	
SECOND(date)	
MONTH(date)	
MINUTE(date)	
STRFTIME('%Y', BD) AS year	extract part of date-time format data
STRFTIME('%m', BD) AS month	
DATE('now')	
STRFTIME('%Y %m %d', 'now')	
DATE('now') - BD AS age	
view	
CREATE VIEW x AS SELECT...	
DROP VIEW x	delete view