SQL Syntaxes

- Transact SQL (T-SQL) - Used in Microsoft Applications (SSMS, Azure ADF, etc.)
- Procedural Language SQL (PL/SQL) - Used in Snowflake

Types of Functions

- Aggregate - A function that returns one value across all rows row
- Scalar - A function that returns one value per row
  - Conditional - Return values based on logical operations using each expression passed to the function
  - Conversion - Convert an expression of any Snowflake data type to another data type
  - Date & Time - Construct, convert, extract, or modify DATE/TIME/TIMESTAMP data
  - Numeric - Perform operations such as rounding and exponentiation
  - String - Perform operations on a string input value, and return a string or numeric value
- Window - Aggregate functions that can operate on a subset of rows within the set of input rows

Functions

Function Type	Function Name	Function Description	Syntax	Formula
Aggregate, Window	AVG	Returns the average of non-NULL records	Both	AVG(<field_name>)
Conditional	BETWEEN, NOT BETWEEN	When the input expression (numeric or string) is within the	Both	BETWEEN/NOT BETWEEN <lower_bound> AND <upper_bound>

		specified lower and upper boundary		
Conditional	CASE	Cascading "if-then-else" statement	Both	CASE WHEN <condition_1> THEN <result_1> WHEN <condition_2> THEN <result_2> ELSE <result_3> END <alias_name>
Conversion	CAST	Converts a value of one data type into another data type	Both	CAST(<field_name> AS <data_
Conditional	COALESCE	Returns the first non-NULL expression among its arguments, or NULL if all its arguments are NULL	PL	COALESCE(<field_name_1>, <field_name_2>, ..., <field_nam
String	CONCAT	Concatenates one or more strings. If any of the values is NULL, the result is also NULL	Both	CONCAT(<field_name_1>, <field_name_2>, ..., <field_nam <field_name_1> || <field_name_ ... || <field_name_N>
String	CONTAINS	Returns true if field_name contains value. Both expressions must be text/string data type	Both	CONTAINS(<field_name>, <val
Conversion	CONVERT	Converts a value of one	T	CONVERT(<data_type>, <field_name>)

		data type into another data type		or CONVERT(<data_type>, <field_name>, <format>)
Aggregate, Window	COUNT	Returns either the number of non-NULL records for the specified columns, or the total number of records	Both	COUNT(*) or COUNT(<field_name>) or COUNT(DISTINCT <field_nar
Context	CURRENT_DATE	Returns the current date of the system	PL	CURRENT_DATE() or CURRENT_DATE
Context	CURRENT_TIME	Returns the current time for the system	PL	CURRENT_TIME
Date & Time	DATE_PART	Extracts the specified date or time part from a date, time, or timestamp	Both	DATE_PART(<date_or_time_part <field_name_or_date_or_time>
Date & Time	DATEADD	Adds the specified value for the specified date or time part to a date, time, or timestamp	Both	DATEADD(<date_or_time_part> <value>, <field_name_or_date_or_time>
Date & Time	DATEDIFF	Calculates the difference between two date, time, or timestamp values based on the date or time part requested. The function	Both	DATEDIFF(<date_or_time_part> <field_name_or_date_or_time_1 <field_name_or_date_or_time_2 or <field_name_or_date_or_tim <field_name_or_date_or_time_2

		returns the result of subtracting the first value from the second value The minus sign (-) can also be used to subtract dates		
Date & Time	DAY	Extracts the corresponding day from a date or timestamp	Both	DAY(<field_name>)
Date & Time	GETDATE()	Returns the current database system timestamp as a datetime value	T	GETDATE()
String	LIKE	Allows case-insensitive matching of strings based on comparison with a pattern. Unlike the LIKE function, string matching is case-insensitive		<field_name> ILIKE <expressio
Conditional	IN, NOT IN	Tests whether its argument is or is not one of the members of an explicit list	Both	<field_name> IN/NOT IN <value_or_sub_query>

		or the result of a subquery		
Conditional	IS NULL, IS NOT NULL	Determines whether an expression is NULL or is not NULL	Both	<field_name> IS NULL <field_name> IS NOT NULL
	IDENTITY	Sets up a column as an identity column (Unique identifier) and automatically assigns a number value for each record as it is inserted into the table		IDENTITY(1,1)
String	LEFT	Returns a leftmost substring of its input. LEFT(STR,N) is equivalent to SUBSTR(STR, 1,N)	Both	LEFT(<field_name>, <length>)
String	LENGTH, LEN	Returns the length of an input string or binary value	Both	LEN(<field_name>) or LENGTH(<field_name>)
String	LIKE	Allows case-sensitive matching of strings based on comparison with a pattern. Unlike the ILIKE function, string	Both	<field_name> LIKE <expressior

		matching is case-sensitive		
	LIMIT	Limits the amount of records that are returned in the results	PL	LIMIT <number_of_records_to_return
String	LTRIM	Removes leading characters, including whitespace, from a string	Both	LTRIM(<field_name>, <charact
Aggregate, Window	MEDIAN	Determines the median of a set of values	Both	MEDIAN(<field_name>)
Aggregate, Window	MIN, MAX	Returns the minimum or maximum value for the records within a field. NULL values are ignored unless all the records are NULL, in which case a NULL value is returned	Both	MIN(<field_name>) MAX(<field_name>)

Aggregate, Window	MODE	Returns the most frequent value for the values within a field. NULL values are ignored. If all the values are NULL, or there are 0 rows, then the function returns NULL	Both	MODE(<field_name>)
Date & Time	MONTH	Extracts the corresponding month from a date or timestamp	Both	MONTH(<field_name>)
Window	RANK	Returns the rank of a value within an ordered group of values The rank value starts at 1 and continues up sequentially If two values are the same, they have the same rank	Both	RANK() OVER(ORDER BY <field_name> ASC/DESC) or RANK() OVER(PARTITION BY <field_name> ORDER BY <field_name> ASC/DESC)
String	REPEAT	Builds a string by repeating the input for the specified number of times	PL	REPEAT(<field_name_or_value> <number_of_times_to_repeat>)

String	REPLACE	Removes all occurrences of a specified substring, and optionally replaces them with another string	Both	REPLACE(<field_name_or_value <value_to_be_replaced>, <value_to_replace_with>)
String	REPLICATE	Builds a string by repeating the input for the specified number of times	T	REPLICATE(<field_name_or_valu <number_of_times_to_repeat>)
String	REVERSE	Reverses the order of characters in a string, or of bytes in a binary value	Both	REVERSE(<field_name>)
String	RIGHT	Returns a rightmost substring of its input	Both	RIGHT(<field_name>, <length>
Numeric	ROUND	Returns rounded values for field/value	Both	ROUND(<field_name>, <number_of_zeroes>)
Window	ROW_NUMBER	Returns a unique row number for each row within a window partition The row number starts at 1 and continues up sequentially	Both	ROW_NUMBER() OVER(ORDER <field_name> ASC/DESC) or ROW_NUMBER() OVER (PARTITION BY <field_name> C BY <field_name> ASC/DESC)

String	RTRIM	Removes trailing characters, including whitespace, from a string	Both	RTRIM(<field_name>, <charact
String	SPLIT	Splits a given string with a given separator and returns the result in an array of strings	PL	SPLIT(<field_name>, <separatc
String	SPLIT_PART	Splits a given string at a specified character and returns the requested part If any parameter is NULL, NULL is returned	PL	SPLIT_PART(<field_name>, <character_delimiter>, <part_number>)
String	STRING_SPLIT	Splits a given string at a specified character and returns the requested part. If any parameter is NULL, NULL is returned	PL	STRING_SPLIT(<field_name>, <character_delimiter>) or STRING_SPLIT(<field_name> <character_delimiter>, <part_number>)
String	SUBSTR, SUBSTRING	Returns the portion of the string or binary value from base_expr, starting from	Both	SUBSTR(<field_name>, <start_chararcter_number>, <length>) SUBSTRING(<field_name>, <start_chararcter_number>, <length>)

		the character/byte specified by start_expr, with optionally limited length		
Aggregate, Window	SUM	Returns the sum of non-NULL records for expr. You can use the DISTINCT keyword to compute the sum of unique non-NULL values. If all records inside a group are NULL, the function returns NULL	Both	SUM(<field_name>)
Conversion	TO_VARCHAR	Converts the input field to a string. For NULL input, the output is NULL	PL	TO_VARCHAR(<field_name>) or TO_VARCHAR(<field_name> <format>)
Date & Time	TO_DATE, DATE	Converts an input field to a date. For NULL input, the output is NULL	PL	TO_DATE(<field_name>) or TO_DATE(<field_name>, <format>) or DATE(<field_name>) or DATE(<field_name>, <forma
Conversion	TO_NUMBER	Converts an input field to a fixed-point number. For NULL input, the output is NULL	PL	TO_NUMBER(<field_name>) or TO_NUMBER(<field_name>, <number_of_characters, <number_of_decimals>>)

Conditional	TOP	Limits the amount of records that are returned in the results	T	TOP(<number_of_records_to_re
String	TRIM	Removes leading and trailing characters from a string	Both	TRIM(<field_name>) or TRIM(<field_name>, <chara _to_trim>)
String	UPPER	Returns the input string field with all characters converted to uppercase	Both	UPPER(<field_name>)
Date & Time	YEAR	Extracts the corresponding year from a date or timestamp	Both	YEAR(<field_name>)

Edit **Contact Intranet Support**