1 BigQuery vs Teradata SQL

This listing was inspired by Coursera specialization *From Data to Insights with Google Cloud Platform*. I have observed there usage of different SQL statements. I decided to extract some of them and show here, how similar queries look like in Standard/BigQuery SQL comparing to syntax commonly used in Teradata. Then, concept of arrays will be briefly mentioned.

BigQuery	Teradata
Sample selection	
SELECT * FROM *table* **LIMIT 10;**	SELECT **TOP 10** * FROM *table*;
Table creation	
CREATE **OR REPLACE** *table* AS SELECT * FROM **...;**	CREATE *table* **AS (**SELECT * FROM **...) WITH (NO) DATA;**
WITH clause order	
WITH **set1** AS (SELECT ’0’ as num), set2 AS (SELECT num FROM **set1**) SELECT * FROM set2 ;	WITH set2 AS (SELECT num FROM **set1**), **set1** AS (SELECT ’0’ as num) SELECT * FROM set2 ; -- *works also in reverse order*

Different Select functions

SELECT **CONCAT**(’part1’,’part2’);	SELECT ’part1’**||**’part2’;
SELECT **SAFE_DIVIDE**(value_to_divide, divisor);	SELECT value_to_divide / **NULLIFZERO**(divisor); -- *this solution does not work for non-numeric divisors*
SELECT * **EXCEPT(col_to_exclude)** FROM *table*;	N/A
SELECT **IF(condition, value_when_true, value_when_false)** FROM *table*;	SELECT **CASE WHEN condition THEN value_when_true ELSE value_when_false** FROM *table*;
SELECT **FORMAT("fmt",** col) FROM *table*;	SELECT **CAST**(col AS **"fmt"**) FROM *table*;
SELECT **EXTRACT(DATE FROM** date_col) FROM *table*;	SELECT **CAST**(date_col **AS DATE**) FROM *table*;
SELECT **DATE_TRUNC**(date_col, **MONTH**) FROM *table*;	SELECT **EXTRACT(MONTH FROM** date_col) FROM *table*;
SELECT **SAFE_CAST**(col AS INT64) FROM *table*;	N/A
SELECT * FROM *table1* **UNION DISTINCT** SELECT * FROM *table2*;	SELECT * FROM *table1* **UNION** SELECT * FROM *table2*;
SELECT **ANY_VALUE**(col) FROM *table*;	N/A

Concept of arrays as type in Teradata is not as flexible as in BigQuery. Before it can be used, explicit type of an array needs to be defined.

Aggregate functions

SELECT **ARRAY_AGG**(col *[LIMIT num]*) FROM *table*;	CREATE TYPE *type_name* as VARCHAR(20) ARRAY[5];
	SELECT **ARRAY_AGG**(col, NEW *type_name*) FROM *table*;

SELECT **STRING_AGG**(col *[LIMIT num]*) FROM *table*;	N/A

In Teradata, arrays are of given, predefined length. In BigQuery it is flexible and it is possible to check length of array created using following syntax:
SELECT **ARRAY_LENGTH**(*array_col*) FROM *table*;