Kaggle.com course Notes: ADVANCED SQL.

Lets start with query optimization:

Each SQL comes with several functionalities. Like the python comes with Tensorflow, Pandas and Matplotlib . There is a Query Optimizer function in the SQL that comes with most database which can be used to optimize the several queries of the SQL.

The amount of data query uses can be so much of useful for the process of query optimization , we have show_amount_of_data_scannerd() which usually tells us the amount of data the query uses.

Show_time_to_run() is a another optimization related function and the function simply carries information that how much a typical SQL query take to executed.

Reduction in Data

If there is a way to reduce the data, it has been provided the course that 1000X reduction in data being scanned is seen to complete a query , because the raw data contained a text file that was 1000X larger than the field we might need. Hence, only selecting the column that is required helps in optimization of SQL Queries.

Another useful measure to follow by using SQL is too scan les data. If we use several function including 1:1 relationship or functions that may reduce the size of the data, we may end up optimizate the SQL queries.

Avoid N:N JOINS.

- N:N Joins is used to join N groups of rows from a column to N groups with another columns. It's a great way of joining rows from two tables. Henceforth , we end up getting more and more columns that the original query. Thus, to wind up Optimization of queries , it is good idea to reduce the usage of N:N joins while performing queries in the SQL tasks.

Joins and Unions:

- INNER JOIN, LEFT JOIN, RIGHT JOIN .
  LEFT JOIN RIGHT IS TYPICAL FORMAT WHERE LEFT APPEARS TO TAKE COMMON ELEMETS FROM LEFT COLUM AND RIGHT TAKE COLUMNS FROM THE RIGHT TABLE.

- INNER JOIN: A INNER JOIN B (COMMONL COLUMNS OF BOTH ELEMENTS ARE TAKEN)
  LEFT JOIN : ENTERIES COMMON OF A AND B ALONG WITH ENTERIES OF A ARE TAKEN.
  RIGHT JOIN : ENTRIES COMMON OF A AND B ALONG WITH ENTERIES OF B ARE TAKEN

- WHAT DOES JOIN PERFORM , DOES IT JOINS VERTICAL ELEMENTS . NO IT JOINS THROUGH HORIZONTAL ELEMENTS.WHERE AS UNION TAKES COMMON ELEMENTS VERTICALLY I.E. ROW WISE.
- DATA TYPES OF COLUMN MUST BE SAME WHILE USING UNION ELSE THERE IS AMBIGIOUS THAT WHICH IS ACTUAL DATA TYPE AND WE CAN END UP THROWING ERROR AT THE COMPUTER SCREEN.

- UNION ALL: IT IS USED TO KEEP THE DUPLICATE VALUES I.E VALUE THAT APPERS MORE THAN ONCE. AND IN ORDER TO AVOID THIS DILEMMAT OR TO AVOID DUPLICATION VALUES WE USE UNION DISTINCT.

NESTED AND REPEATED DATA:

- SCHEME , THE TERM OFTEN ASSOCIATED WITH STRUCTURE OF COLUMN. NEST COLUMS HAVE TYPE STRUCT(OR RECORD TYPE). WHEN TWO TABLES ARE COMBINED TYPICALLY, AND ONE COLUMN OCCURS AS A TYPE FOR SECOND COLUMN.
- REPEATED DATA IS POSSIBLELY OCCURRED WHEN ROW HAS MORE THAN ONE VALUE PERMITTED. AND IS TYPICALLY REFLECTED IN TABLE'S SCHEME.
- EACH ENTRY IN THE REPEATED FIELD IS AN ARRAY[SIMILAR TO THE PYTHON ARRAY TERM].
- WE USE UNNEST FUNCTION WHILE ENTERING THE REPEATED DATA'S COLUMN IN THE SQL QUERIES.

- FOR REPEATED AND NESTED DATA , WE USE THE UNNEST FUNCTION ON REPEATED DATA'S'COLUMN .AND WE CAN GIVE ALIAS NAME TO SUCH COLUMN TO REDUCE AMBIGIOUS NATURE.

ANALYTICAL FUNCTION:

- COMPLEX CALCULATION AN COMPLEX SYNTAX IS NOT EASIER , IT CREATES A , UNEASY AND DIFFICULT TO USE QUERIES. BUT WE HAVE ANALYTICAL FUNCTIONS THAT PERFORM COMPLEX CALCULATIONS WTH EASY OR LESS COMPLEX SYNTAX.

- OVER FUNCTIONS IS A WIDELY USED FUNCTION WHICH IS USED TO DEFINE A RANGE OR SET OF ROWS TO PERFORM IN CALCULATION. OVER COMES WITH THREE CLAUSES:
- PARTITION BY: THE ROW OF THE TABLE IS DIVIDED INTO DIFFERENT GROUPS. ORDER BY: EACH PARTITION IS ORDERED BY THIS CLAUSE.
- WINDOW FRAME: SET OF ROWS ARE TYPICALLY IDENTIFIED IN EACH CALCULATED BY THIS CLAUSE.

WINDOW FRAME CLAUSE: THREE WAY OF WRITING

“

- ROWS BETWEEN 1 PRECEDING AND CURRENT ROW - the previous row and the current row.
- ROWS BETWEEN 3 PRECEDING AND 1 FOLLOWING - the 3 previous rows, the current row, and the following row.
- ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING - all rows in the partition.

”.

THREE TYPES OF ANALYTICS FUNCTIONS

- ANALYTICAL FUNCTION OF NAVIGATION TYPE:
- “
- **Navigation functions** assign a value based on the value in a (usually) different row than the current row.
- **FIRST_VALUE()** (or **LAST_VALUE()**) - Returns the first (or last) value in the input
- **LEAD()** (and **LAG()**) - Returns the value on a subsequent (or preceding) row

”.

Analytics function of aggregate functions

- Min() or Max(): These function's have their goal set return Minimum or maximum of input values.
- AVG() or Sum() function(): These function's have their goat set to return the average or sum of input values.
- Count() function: Number of rows in the input is shown or returned.

- Over clause is used to perform aggregate functions.

Analytical Numbering functions:

- Numbering functions : They consider ordering , after it assigns row order starting with 1 and return the order.

- ROW_NUMBER(): The return start with 1. And the return is the order in which the row has represented itself in the input.

- RANK(): “All rows with the same value in the ordering column receive the same rank value, where the next row receives a rank value which increments by the number of rows with the previous rank value”