Issue Highlights and Recommendations

Xingyuan Mu

August 3rd, 2022

# Executive Summary

Raw data row counts: 129,358

Aggregated list row counts: 33,491

Sql scripts lines: 522

Max execution time per query: 9 seconds

Deliverable file list:

Aggregated\_list.csv;

Assignment\_2.sql;

Assignment Documentation.docx

# Difficulties and Solutions

## Data Quality

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Category of column | Columns | Quality | Major Challenge | Solution |
| Id purpose columns | Report\_id, report\_period | Good | Report\_period in String type, hamper proper sorting | Cast report\_period to integer |
| Grouping purpose columns | Test\_no, serial\_no | Bad | more than 10% NULL values, some happening at beginning of a new group which critically hampers the grouping and aggregation | Replace NULL with ‘no data’, to start a new group.  In real work, will exam the pattern of ‘no data’ happenings and provide customized solution.  For Assignment, just leave it as ‘no data’. |
| Measure columns | Start\_len, end\_len, total\_len\_worked | bad | More than 10% NULL values | Use coalesce to allow end\_len, total\_len\_worked fill each other’s null values |
| Information only columns | Size, type | Good with some exceptions |  | Validate that size, type is almost one to one relationship with serial number, therefore leave these 2 columns untouched. |

## Handle NonAlphaNumeric requirement

Write a function to stripe off NonAlphaNumeric characters;

Use function in test\_no an serial\_no treatment.

## Interpret what is the rule of grouping

From the requirement narrative, it seems complex, however by carefully examining the illustration, I get the conclusion, either test\_no goes to a new value, or serial\_no goes to a new value, it is the starting row of a new group.

|  |  |  |  |
| --- | --- | --- | --- |
| REPORT\_ID | REPORT\_PERIOD | TEST\_NO | SERIAL\_NO |
| 123 | 1 | 1A | AB |
| 123 | 2 | 1A | AB111 |
| 123 | 3 | 1A | AB111 |
| 123 | 6 | 2A | C22 |
| 123 | 7 | 2A | C22 |
| 123 | 8 | 2AD | E3 |
| 123 | 9 | 2A | F4 |

## Handle whether current value is a new value in an ordered list

Sql server built-in windowing function lag() provide all I need here.

According to requirement of different columns, use lag() either by ascending order or by descending order, either with partition by or without.

I know other friends write store-procedure and use loops to do comparison, the performance is very poor, after properly indexing the procedure still takes 30 minutes to produce result list.

With proper sorting and lag(), my solution runs under 5 minutes including all analysis, discussion, manipulation and final result list generation.

# Recommendation to data collection

Since the major difficulty is NULL value for columns like test\_no, serial\_no, I recommend data owner to put check or rules to improve data quality at source.

At the start of a new report\_id, and/or report\_period, test\_no and serial\_no can not be NULL;

At the on-going events within a report\_id, and/or a report\_period, proper deault value of test\_no, serial\_no, start\_len, end\_len, total\_len\_worked can be designed to guarantee data quality and work efficiency at the same time.