IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Building audit logging for data ingestion frameworks – Identifying gaps and mitigating them – A use case scenario

Bincy George
Associate
Data Engineering
Cognizant Technology Solutions

Abstract: This use case scenario explains the importance of proper logging mechanism in tracking data loads using batch data ingestion frameworks. Issues that can exist in common logging mechanisms and how those can be avoided is explained with the help of a sample scenario.

INTRODUCTION

It is necessary to have some kind of logging in any data ingestion framework to track all the data loaded through the framework. In daily operation handling of batch data ingestion jobs, it is inevitable to have the audit logging done correctly to monitor, track and resolve failures. Consider a batch ingestion framework, with an audit table called as summary audit table, used for logging in all data loads done using the framework code.

This involves logging into fields like job name, feed name/ file name, start time, end time, count of records, status, and error message to write all refined load details to it.

PROBLEMS WITH EXISTING AUDIT LOGGING

The existing audit logging have issues with logging such as:

No logging enabled for failure scenarios.
No error message capture
No date factor in start time and end time columns
No direct logging of the target table name

MITIGATING THE GAPS

The issues were affecting the usability of the summary audit table. Below section explains what changes are made in the code to solve these problems.

3.1Adding date factor to start time and end time fields.

The format used for starttime was using variable Starttimeformat which had the format of only HH:mm:ss. Changing this to yyyy-MM-dd HH:mm:ss adds date factor to the entries made to the audit table and helps in better tracking of the jobs based on run date and time.

Code snippet in scala below:
Previous:
var Starttimeformat = new SimpleDateFormat("HH:mm:ss");
Fixed code:
var Starttimeformat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

.

3.2 Adding table name to the audit logging.

For The target table name can be used to understand which exact table got loaded as part of the data load. To add this factor, without altering the existing summary audit table schema, the table name was incorporated to an existing field 'field name'.

Code snippet in scala below:
val summaryAuditInfo = batchId + "|" + dataSet + "|" + Filevendor + "|" + market_Code + "|" + feedname + "," + refDB + "." + refTableName + "|" + job_name + "|" + starttime toString() + "|" + endtime.toString() + "|" + file_count + "|" + Successflag + "|" + errorMessage

3.3 Adding logging and error capture for failure scenarios.

Error capture and failure logging can be made possible with a try catch code block addition to the existing code. The library *ExceptionUtils* is used to get the root cause of failure and to log the same in error message column of the audit table.

Code snippet in scala below:

```
import org.apache.commons.lang3.exception.ExceptionUtils
try
{
//your code goes here
}
} catch {
     case ex: Throwable =>
      {
        var errorMessage = ExceptionUtils.getRootCauseMessage(ex)
        Successflag = "Failure"
        val job_name = "DATAQUALITY"
        val summaryAuditInfo = batchId + "|" + dataSetFromEnv + "|" + vendorFromEnv + "|" + Market_CD + "|" + feedname + "," + refDB + "." + refTableName + "|" + job_name + "|" + Statstartime.toString() + "|" + StatsEndTime.toString() + "|" + rawfile_count + "|" + Successflag + "|" + errorMessage
            //code for audit table entry
       sys.exit(1)
       }
   }
```

CONCLUSION

Logging of batch data loads in a more accurate way is extremely helpful in tracking the data loads over time and can be used for reporting purposes and get useful insights. Accurate entries of both successful and failed loads should be considered for logging. While logging into the audit table, it is necessary to have basic details like table name, date, and time of ingestion and error messages to be entered into the audit tables.