ENHANCED PERFORMANCE AND ADDED DATA COLUMNS

# Project Objectives

1. Retro-engineer the older below views to use “FDS\_ALL\_LAST\_DOWNLOADS\_FTBL” instead of calculating it on the fly using a SELECT statement in the FROM clause to increase the performance.

FDS\_CONFIGS\_LAST\_TRX\_FVW

FDS\_DIAGS\_LAST\_TRX\_FVW

FDS\_ERRORS\_LAST\_TRX\_FVW

FDS\_SHOTSUMMARY\_LAST\_TRX\_FVW

FDS\_STATUSES\_LAST\_TRX\_FVW

1. Speed up the scheduled job “FDS\_UpdateLastDownloadsTable” by changing the T-SQL statement to use temp tables.

1. Change “FDS\_UNIFIED\_LAST\_DOWNLOADS\_FVW” to have all of the columns from the main data tables (it currently has a subset of popular columns).

# Testing

All changed T-SQL must be compared to their older counterparts. Testing will first confirm a performance improvement (Run Time). After that has been confirmed, then the results will be compared.

A well-documented testing procedure will be followed with results documented to provide data analysis and proof that desired target goals have been met.

General Testing Guidelines

1. T-SQL scripts will be developed and documented designed to produce measured results by simulating varying data sets.
2. Result sets between current and proposed replacement scripts will be loaded into 2 tables and then a comparison script will be run to check for any differences in the data.
3. A standardized set of measurements will be recorded while the above scripts are run.

# Specific Testing Guidelines

1. Make sure that no database load is running so that the source tables aren’t changing.
2. Truncate the current target table, FDS.DW.FDS\_ALL\_LAST\_DOWNLOADS\_FTBL.
3. Create a copy of the target table FDS.DW.FDS\_ALL\_LAST\_DOWNLOADS\_FTBL.
4. Run the current job upload code. Monitor CPU, IO, tempdb.
5. Run the new job upload code into an empty copy of the target table. Monitor CPU, IO, tempdb.
6. Rerun 4 and 5 twice each time to test the source process. Monitor CPU, IO, tempdb.
7. Compare the table contents (“except”) and the performance.
8. Run the new views against the new source. Monitor CPU, IO, tempdb.
9. Run the old views against the original source. Monitor CPU, IO, tempdb.
10. Repeat steps 8 and 9 multiple times with the @DAYSBACK set to maybe 200.
11. Repeat steps 8 and 9 multiple times with DATA\_SET\_DATE changed to new values to force inserts and updates. Monitor CPU, IO, tempdb.

# Specific Testing Steps

TESTING THE UPLOAD FOR FDS.DW.FDS\_ALL\_LAST\_DOWNLOADS\_FTBL

1. After each completed batch run a script that that shows in Temp DB user\_objects\_alloc\_page\_count and **internal\_objects alloc\_page\_count.**
2. Run T-SQL that inserts all historical data into a copy of FDS.DW.FDS\_ALL\_LAST\_DOWNLOADS\_FTBL table using the FDS\_UpdateLastDownloadsTableALLTIMEUsingIndTempTablesMergescriptFinal script.
3. During the above processes run SQL Profiler to measure CPU, Reads, Writes, Duration and Run DT (StartTime).
4. Execute the existing job so a clean historical download into the FDS.DW.FDS\_ALL\_LAST\_DOWNLOADS\_FTBL table after truncating the table.
5. During the above processes run SQL Profiler to measure CPU, Reads, Writes, Duration and Run DT (StartTime).
6. **Run a compare between** FDS.DW.FDS\_ALL\_LAST\_DOWNLOADS\_FTBL table and the copy and make sure the data matches.
7. Truncate FDS.DW.FDS\_ALL\_LAST\_DOWNLOADS\_FTBL and the copy table.
8. Execute steps 1 and 3 immediately after running them again so there will be no database updates (test data collection timing only)
9. During the above processes run SQL Profiler to measure CPU, Reads, Writes, Duration and Run DT (StartTime).
10. **Run 1 and 3 after running the changes to the** FDS.DW.FDS\_ALL\_LAST\_DOWNLOADS\_FTBL and the copy table to force inserts and updates. Run for several values and after each run a comparison on the returned rows. After each run compare timings and TempDB.

Once the above is completed – Testing New Views

1. Run T-SQL that inserts all historical data into a copy of FDS.DW.FDS\_ALL\_LAST\_DOWNLOADS\_FTBL table using the FDS\_UpdateLastDownloadsTableALLTIMEUsingIndTempTablesMergescriptFinal script after truncating table.
2. Run a comparison between the old views and the new views using the copy of FDS.DW.FDS\_ALL\_LAST\_DOWNLOADS\_FTBL. Limit the returned rows by limiting and varying the number of lasers in the WHERE clause.
3. During the above processes run SQL Profiler to measure CPU, Reads, Writes, Duration and Run DT (Start Time).
4. After measuring performance, create a combined statement of old and new views and run an EXCEPT against the results.
5. For the new unified view run a timing test recording CPU, Reads, Writes, Duration and Run DT (Start Time).