

Case Study – SAP Data Validation for Queries

Abstract

In Client's systems, 194 queries are executed every day. There is a budget session when 74 queries are run and budgets are planned. These queries are hence very important to the port. By nature BW stores large volume of data. For analytical purposes, the final dataset is frequently aggregated and the transactional detail is lost. When users look at the analytical reports generated out of the BW system, the results they see cannot be easily verified. As a result, creating user confidence in data quality becomes difficult and user trust is eroded, which can ultimately undermine the success of the EDW initiative. The users wanted to verify that the data been pulled out and the results of the query are identical thereby having correct budget calculations.

The client wanted to develop a process and software for the verification of data loaded into BW overnight.

This would verify that the data loaded into BW does in fact match the corresponding ECC data. This report would immediately detect discrepancies in the data, rather than allowing them to go undetected as was the case in past.

In this document we showcase what principles were used by V3it to perform data validation and how we have automated the whole process.

Principles followed:

Understand the source of your data - it was very important to know and understand the data of each application area in the source system, and how it is was entered and stored in the source system database. Data validation was not a simple process. It was not purely technical, but also required a quality management mindset, and a commitment to data validation process. Data validation required understanding of the associated business processes, data entry modes, and data warehousing theory. Without this, the validation functions could not achieve their full potential.

Note:

Validation did not mean that we were required to check everything about the data. Rather, it was detrimental of us to ensure that critical elements like the number of records and/or sums of certain value fields (key figures) that arrived and were updated properly into the data warehouse. With this knowledge, setting up our validation rules and processes became more 'mechanical'.

The key things to know were:

- 1) Which fields are populated by the application?
- 2) Which of these fields are utilized in data warehouse reporting?
- 3) What is the frequency with which this information is loaded into the data warehouse?

Case Study – SAP Data Validation for Queries

The BW queries which ran every night and were saved as excel files. We were required to customize our validation approach - not all BW objects created were same, nor were they validated in the same way. Validation frequency, selection parameters, and comparison object definition differed depending upon BW object type – an operational data store and the different record types stored.

Options used for Data Validation by the Client

Manual validation

Manual data validation was performed in some cases while in some other cases they would manually compare datasets in the source and BW systems for comparison and validation of 'statistically significant' representative data sample according to a pre-written validation test script. This led to undetected data errors and omissions, due to exclusion and human error. In some other cases they would manually determine which table brought up the data from the R/3 system (via SE16n), and then use the Manage function in BW to view the data records in the ODS or Cube (note: to sum records, or to capture record counts in BW, they had to export the data into a spreadsheet for external manual validation). This was thus very complex and a tedious work, which required dedication of work hours from knowledgeable resources to perform this validation accurately.

Data Validation techniques developed by V3iT:

Report program was written to extract data from various sources and tables. This was then downloaded in excel file. VB macros were developed to compare the results and report discrepancies.

This was all automated to run every day at a fixed time to trigger the R/3 system and the VB macro.

Technologies used

Following technologies were used:

SAP ECC (R/3)

Microsoft Excel VB Macros

IMPACT/ BENEFITS

Data discrepancies were detected.

The application provided user friendly interface which enabled the user to plan during budget sessions.

Key challenges faced:

The customer environment used Excel 2003 where in sorting for only 3 fields was allowed. Hence macros written had to be made compatible to the 2003 version.