

## CASE STUDY–Data Integration

### CLIENT: Leading Consumer Healthcare Vendor

#### Background:

The project addressed the need of efficient integration of multiple data items with the cloud based CRM system that was in use in the client's organization.

The client used a Cloud based CRM system consisting

- Account Management (Hospitals, clinics, pharmacies etc.)
- Contact Management (HCPs, prescribers, non-prescribers, nurse practitioners etc.) for FFE (Field Force Enablement)
- Call Management (Appointments with HCPs)
- Call Center System, providing Service Request management, Task Management and other functions.

The project involved data integration of the client's "Master Data Management" MDM with the CRM application.

The source data items (like Product, Territory, User, Account, Contact, Alignment etc.) from different source systems (including the Data Access Layer and MDM, CRM) were retrieved and staged in the Data Service Hub (DSH).

This data from DSH was then processed and managed according to predefined business rules using the Informatica ETL Tool. Finally, the prepared data was pushed to the CRM application through web service calls.

#### Challenges

When the initial POC was done to connect the CRM system from Informatica, the complex approach chosen caused several issues:

- Delay in Batch Processing: Individual processes were taking a lot of time to complete due to higher data volumes.
- Lower Productivity: As downstream applications were not getting the output-data on time, the productivity of reporting solutions was significantly reduces.
- Limited Information Sharing across Applications: Applications were unable to share information that should be shared (such as Doctor Information, User Representative Information, Product information) which adversely affected the field representatives' activities.
- Unnecessary Administrative/ Support Overhead: As each of the processes were taking longer to run, support teams had to monitor the jobs/status for an unnecessarily long period of time.

#### CASE SUMMARY

*Leading Consumer Healthcare Vendor*

#### Requirement:

Optimization of Cloud CRM and Informatica Data Integration.

#### Approach:

Enabling multiple data operations through a single login session and identifying datasets for parallel processing.

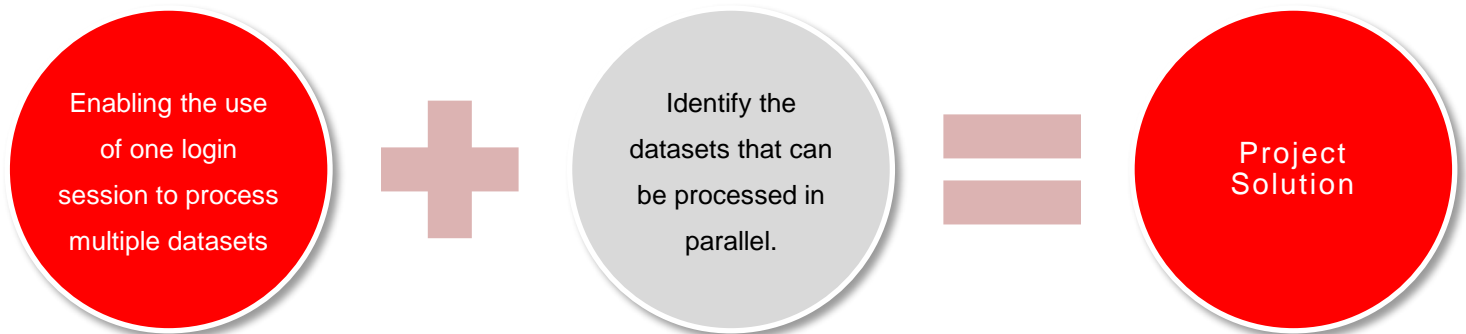
#### Result:

- Faster batch processing
- Increased productivity
- Reduced administrative overheads.

Due to these issues, client business was significantly affected as company representatives were unable to show relevant product information during doctor or hospital visits. Paramount's technical lead was tasked with putting together a more efficient solution to the integration requirement that enabled processes to work faster and for essential data to be available when required.

### **Approach:**

The Login-Data Operation-Logout method used for the initial integration worked on a record-by-record level, which made processes slow. The solution approach taken was focused on achieving the following goals –



### **Technology**

Informatica Power Center 9.6.1, Oracle 9i Database PL-SQL Developer, UNIX Solaris, HTML, XML, Tivoli Workload Scheduler (Maestro)

### **Solution:**

The key components identified as part of the long term solution were the use of parallel data processing, along with the utilization of a single login cookie for accomplishing multiple dataset operations. These were achieved through the following changes -

- Creation of a process to capture the cookie information after the first login to the CRM system into the local database.
- Usage of the stored Cookie information to perform multiple data operations.
- The logout process was modified to be performed only at the end of each data load once all the load processes were complete.
- Created multiple instances of time consuming ETL processes at the Informatica server to enable parallel processing.
- Create object specific parameter files so that all processes can run in parallel

**Results:**

Implementing an effective end-to-end single login and parallel processing system resulted in

