# Technical Specification

***Solution: Clinic Reporting System - BI Solution***

***Owner:*** *Karla Rivas*

***Date:*** *March 2017*

## Change Log

|  |  |  |
| --- | --- | --- |
| **Who** | **When** | **What** |
|  |  |  |
|  |  |  |
|  |  |  |

# 

[**Technical Specification**](#_5x0anaz1yirx) **1**

[Change Log](#_q1lmpfprz8ic) 1

[1. Introduction](#_sbr21s1l7ca) 2

[1.1 Summary](#_ett6qrek0e6k) 2

[2. Solution Naming Conventions](#_9kx0usga991e) 2

[2.1 Data Warehouse Database (SQL Server)](#_jdlvndaj50u8) 2

[2.2 ETL Staging Database (SQL Server)](#_sxgu5db7q0k) 2

[2.3 ETL Processing Objects (Integration Services)](#_3vedyjmdztyc) 3

[3. Data Model Overview](#_48c4kch0r8p4) 3

[3.1 Bus Matrix](#_upm04v23c6ze) 3

[3.2 Data Warehouse Objects](#_m7i51u8mcpls) 3

[4. Data Integration](#_9cf9l35vwtmq) 4

[4.1 Solution Overview](#_hsqvz2m2qzyd) 4

[4.4 Patients Data Integration](#_ww702m47tu5h) 5

[4.5 Visits Data Integration](#_alf8e1bbfgz) 6

[4.6 Other Supporting Objects](#_fugpwhvj840c) 7

[5. ETL Solution Overview](#_jvoqh39thd2x) 7

[5.1 Data Sources](#_velptnuk4y3l) 8

[5.2 Solution Overview](#_nohpe5909t5n) 8

[5.3 Supporting scripts](#_xf9j9n76wu1m) 9

[5.4 Other Supporting Objects](#_r3lo10h696z9) 9

[6. The Tabular Solution](#_md0tz18xnm1v) 9

[7. SSIS Package Overview](#_hasp4ltj8of9) 10

# 

## 1. Introduction

### 1.1 Summary

The client provides health services. It has three clinics located in three different locations Bellevue, Kirkland, and Redmon. All the information collected by each of the clinics (Patient profiles and medical consultations) is submitted at the end of the day to a corporate office where all the information is centralized.

Actually, all this process is manual. Each clinic has staff members that collect the information and upload the data in a shared folder at the end of the day. There are three shared folders, one for each clinic. At the corporate office, there is another group of people in charge of validate the data submitted (in the files), and load the information in the databases. If some information is wrong, then is returned to the clinic for review.

In order to make this process more efficient, the client requires to automate these process. Also it requires an ETL system to load the data warehouse, and provide some reporting capability. In the following sections, details about the solution are provided.

## 2. Solution Naming Conventions

This section provides a summary of the naming conventions used in this project

### 2.1 Data Warehouse Database (SQL Server)

|  |  |  |  |
| --- | --- | --- | --- |
| Object Type | Convention | Example | Description |
| Fact Table | FactTableName | FactSales | A table that holds measures and dimensional keys |
| Dimension Table | DimTableName | DimAuthors | A table that holds dimensional keys and descriptive dimensional attributes |
| View | sta\_group\_table\_view | sta\_clinic\_doctors\_view | A saved select statement used for reporting.  For this project group can be visits (all the data related to the medical consultation) and clinics |
| Stored Proc. | sta\_clinics\_clinics\_view | sta\_clinics\_clinics\_view | A saved set of statements used for reporting |

### 2.2 ETL Staging Database (SQL Server)

|  |  |  |  |
| --- | --- | --- | --- |
| Object Type | Convention | Example | Description |
| Table | sta\_group\_table | sta\_clinics\_clinics\_view | A table used to temporarily hold data for the ETL process  For this project group can be visits (all the data related to the medical consultation) and clinics |
| View | sta\_group\_table\_view | sta\_clinics\_clinics\_view | A saved select statement that provides ETL data  For this project group can be visits (all the data related to the medical consultation) and clinics |
| Stored Proc. | sta\_clinics\_clinics\_view | sta\_clinics\_clinics\_view | A set of saved statements that provides ETL data |

### 2.3 ETL Processing Objects (Integration Services)

|  |  |  |  |
| --- | --- | --- | --- |
| Object Type | Convention | Example | Description |
| Object Type | **Convention** | **Example** | **Description** |
| File Connection | NameofFile Connection | NewPatients Connection | A SSIS File connection object |
| Database Connection | DatabaseName Connection | DWClinicReportData Connection | A SSIS DB connection object |
| Task | Purpose ObjectTypeObjectName | Load DimDate SQL Task | A SSIS task object |

## 3. Data Model Overview

This section provides an overview of the Clinic Report Data warehouse

### 3.1 Bus Matrix

Clinic Reporting Solution, provides to users the ability to analyze:

* Current staff workload (doctors)
* Total of Patients attended over time and incomes

## 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Process** | **Dimensions** | | | | | |
| **Doctors** | **Clinics** | **Patients** | **Procedures** | **Dates** | **Shifts** |
| Visits | X | X | X | X | X |  |
| DoctorsShifs | X | X |  |  | X | X |

### 3.2 Data Warehouse Objects

The following table provides a summary of the available objects in the Clinic Data warehouse

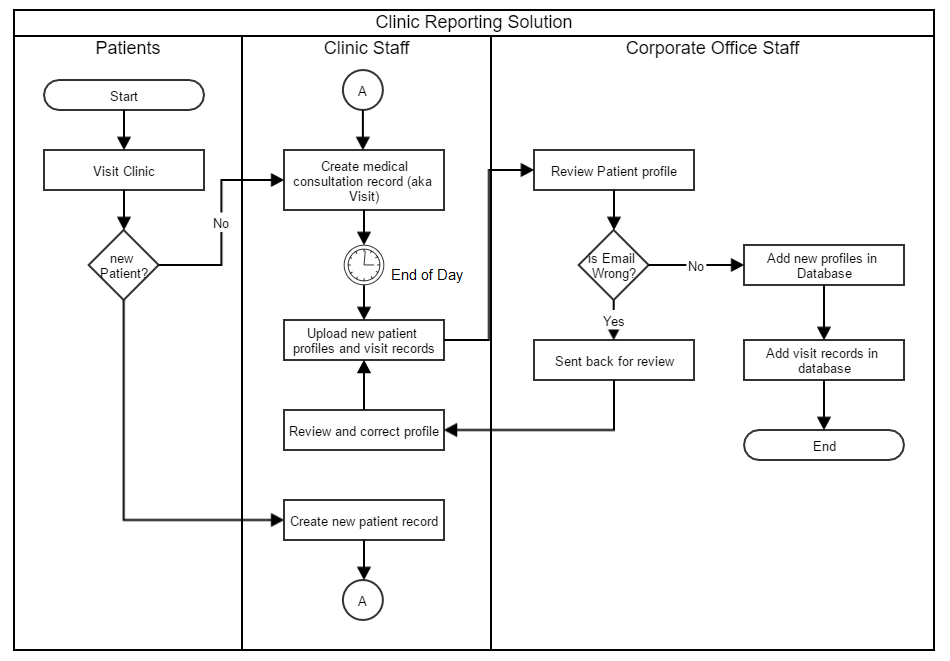
|  |  |
| --- | --- |
| Table | Description |
| DimPatients | Patient’s profiles |
| DimClinics | Clinic information |
| DimDoctors | Medical staff |
| DimShifts | Available shifts assigned to medical staff |
| DimDates | Calendar information |
| Procedures | Medical procedures follow by the clinic staff |
| FactVisits | Tracks all the medical consultations provided by the medical staff and the charges applied for the service |
| FactDoctorShifts | Tracks the workload of medical staff |

## 4. Data Integration

In addition to the ETL process requested to support the Clinic Data warehouse, the client also needed an additional process to automate the entry of patients and medical consultations (visits) in the system.

### 4.1 Solution Overview

The following figure depicts the current process to input new patients and medical consultation records:



When patients visit the any of the clinics for the first time a new record is generated to create their profile. For both new and old patients, every time they arrive to any of the clinics a medical consultation (from now on visit) record is generated. At the end of the day all the data is collected in two separate files in each of the clinics, one for patient's data and another one with the details of the visit, and these files are uploaded in a shared location. Corporate office staff collect the files from each clinic and review the information. If they find inconsistencies in the patient data, they sent back the information to clinic for review. If no issue is found then the profile is created in the database. In addition, all visit records are added in the database.

The part of the process automated in the one performed by the corporate office staff, with the current automation, patients profiled will be automatically validated by the process (to be more specific email data) and those records with incorrect values are gathered in a separate file to be reviewed later. All the data that pass the quality test will be added in the database.

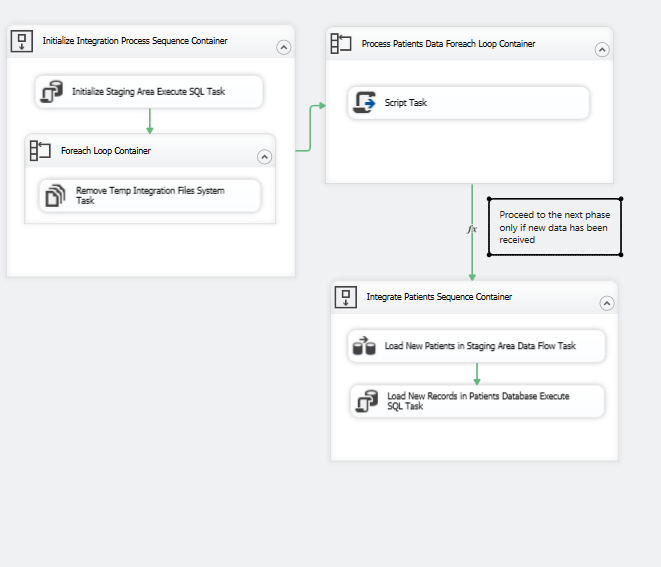
4.3 Data Sources

There are two data sources for this integration

|  |  |  |
| --- | --- | --- |
| Source | Type | Description |
| YYYYMMDDNewPatients.csv | CSV File | Patients Profiles |
| YYYYMMDDVisits.csv | CSV File | Visits Data |

### 4.4 Patients Data Integration

The following image shows the current process configured in SSIS to integrate the new patient's profiles.

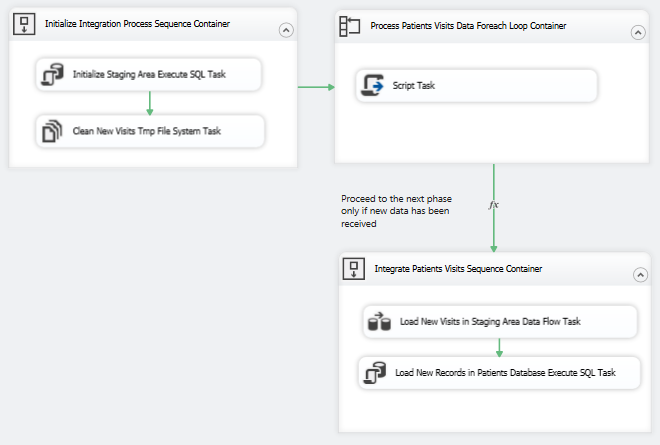


### 

|  |  |  |
| --- | --- | --- |
| Step | Object | Description |
| 1 | Initialize Staging Area Execute SQL Task | Re-create the staging table used to temporarily store patient data |
| 2 | Foreach Loop Container | Get the list of temporary files using during the integration process |
| 2.1 | Remove Temp Integration File System Task | Removes each of the files identified in the step 1 |
| 3 | Process Patients Data Foreach Loop Container | Get the list of files with Patient data |
| 3.1 | Script Task | From each of the files identified in the step 3, evaluate the email field and if the value is incorrect store the record in a separate file for further review |
| 4 | Load New Patients in staging Area Data Flow Task | Gets all the new patient profiles from file and load them in an staging table |
| 5 | Load New Records in Patients Database Execute SQL Task | Load the new records in the OLTP system |

### 4.5 Visits Data Integration

The following image shows the current process configured in SSIS to integrate the new visit information



|  |  |  |
| --- | --- | --- |
| Step | Object | Description |
| 1 | Initialize Staging Area Execute SQL Task | Re-create the staging table used to temporarily store visit data |
| 2 | Clean New Visits Tmp File System Task | Removes temporary files |
| 3 | Process Patients Visits Data Foreach Loop Container | Get the list of files with Visit data |
| 3.1 | Script Task | From each of the files identified in the step 3 conform the data to be integrated later |
| 4 | Load New Visits in Staging Area Data Flow Task | Load Visit data in staging table |
| 5 | Load New Records in Patients Database Execute SQL Task | Load data into OLTP system |

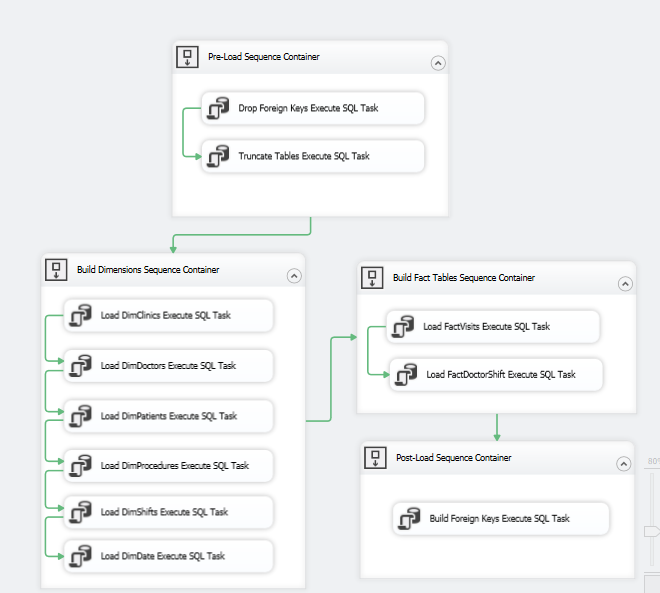
### 4.6 Other Supporting Objects

There are other database objects in the database used to support this process

|  |  |  |
| --- | --- | --- |
| # | Object | Description |
| 1 | sta\_patients\_new\_tmp | Stores new patient profiles temporarily |
| 2 | sta\_visits\_new\_tmp | Stores new visit records temporarily |
| 3 | usp\_create\_patients\_staging\_table | Creates the sta\_patients\_new\_tmp table |
| 4 | usp\_create\_visits\_staging\_table | Creates the sta\_visits\_new\_tmp table |
| 5 | usp\_integrate\_patients | Grabs new patient records and insert them in the Patients.dbo.Patients table |
| 6 | usp\_integrate\_visits | Grabs new visit records and insert them in the Patients.dbo.visits table |

## 5. ETL Solution Overview

Next figure displays the current etl flow follow to load the data from the OLTP system into the data warehouse. Due to current data warehouse doesn’t track changes over time in any dimension, flush and fill technique is used.



### 5.1 Data Sources

The following table shows the list of source systems that feed the DWClinicReportData data warehouse.

|  |  |  |
| --- | --- | --- |
| Schema Name | RDBMS | Description |
| Patients | SQLServer | Information about patients and visits |
| DoctorSchedules | SQLServer | Information about medical staff and schedules |

### 5.2 Solution Overview

The following table lists the objects in the ETL process.

|  |  |  |
| --- | --- | --- |
| Step | Object | Description |
| 1 | Drop Foreign Keys | Removes all foreign keys from both FactVisits and FactDoctorShifts tables |
| 2 | Truncate Tables Execute SQL Task | Removes all the data in each Dimension and Fact Table |
| 3 | Load DimClinics | Fill DimClinics with new data |
| 4 | Load DimDoctors | Fill DimDoctors with new data |
| 5 | Load DimPatients | Fill DimPatients with new data |
| 6 | Load DimProcedures | Fill DimProcedures with new data |
| 7 | Load DimShifts | Fill DimShifts with new data |
| 8 | Load DimDate | Fill DimDate with new data |
| 9 | Load FactVisits | Fill FactVisits with new data |
| 10 | Load FactDoctorsShfit | Fill FactDoctorShift with new data |
| 11 | Build Foreign Keys | Re-creates all the foreign keys in Fact Visits and FactDoctorShifts tables |

### 5.3 Supporting scripts

There is a group of stored procedures used to support the ETL process, all these procedures are executed via Execute SQL Task Objects. The following table show the relation between each of the SSIS object and the stored procedures in the Database

|  |  |  |
| --- | --- | --- |
| Step | Object | Related Store Procedure |
| 1 | Drop Foreign Keys | usp\_etl\_drop\_fks |
| 2 | Truncate Tables Execute SQL Task | usp\_etl\_load\_truncate\_tables |
| 3 | Load DimClinics | usp\_etl\_load\_dimclinics |
| 4 | Load DimDoctors | usp\_etl\_load\_dimdoctors |
| 5 | Load DimPatients | usp\_etl\_load\_dimpatients |
| 6 | Load DimProcedures | usp\_etl\_load\_dimprocedures |
| 7 | Load DimShifts | usp\_etl\_load\_dimshifts |
| 8 | Load DimDate | usp\_etl\_load\_date\_dimension |
| 9 | Load FactVisits | usp\_etl\_load\_factvists |
| 10 | Load FactDoctorsShfit | usp\_etl\_load\_factdoctorshift |
| 11 | Build Foreign Keys | usp\_etl\_create\_fks |

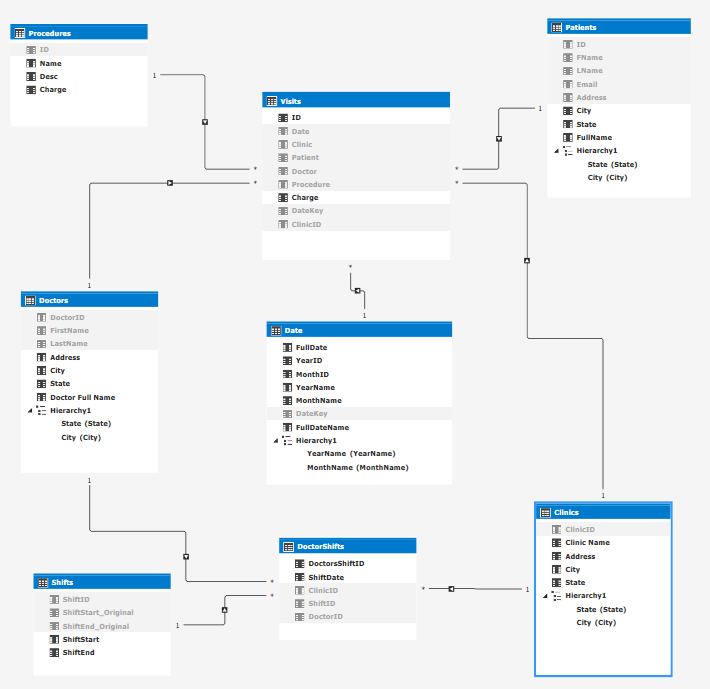
### 5.4 Other Supporting Objects

There are other database objects in the database used to support this process

|  |  |  |
| --- | --- | --- |
| # | Object | Description |
| 1 | sta\_clinics\_clinic\_view | Transformed clinic data |
| 2 | sta\_clinics\_doctors\_view | Transformed doctor’s data |
| 3 | sta\_clinics\_schedule\_view | Transformed clinic data |
| 4 | sta\_clinics\_shifts\_view | Transformed shift data |
| 5 | sta\_visits\_patients\_view | Transformed patients profiles |
| 6 | sta\_visits\_procedures\_view | Transformed procedures data |
| 7 | sta\_visits\_visits\_conformed\_view | Conforms clinic IDs |
| 8 | sta\_visits\_visits\_view | Transformed visit data |

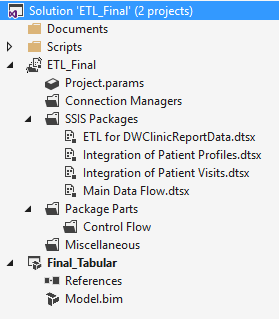
## 6. The Tabular Solution

In addition a tabular solution was developed, the following figure shows the current structure of this solution:



## 7. SSIS Package Overview

All the project has been compiled in a single SSIS Package called ETL Final



All the supporting documentation like technical references and installation manuals have been placed in the document folder.

Scripts folder contains all the SQL scripts used to build Views and stored procedures used in both the integration and ETL processes

ETL\_Final contains all the packages used for integrating and loading the data into the DWClinicReportData.

Finally, Final\_Tabular, contains the tabular model.