**EDI DATABASE**

Database with fields to produce EDI X12 file. In our case we use EDI X12 837 Healthcare Claim (X12 Release 4010).

Sometimes you might not know all the fields you need to produce valid EDI X12. If you have sample EDI X12 file you

can find out all the fields using our ediReports product. It simply prints all data with field names in CSV-Excel format.

Documentation explaining EDI X12 message layout and structure. You should have some document that would list all

the EDI X12 segments and elements that you need to create in the EDI X12 file. Contact your trading partner or

supplier for simple documentation on EDI X12 message you need to translate from the database. This documentation

usually lists some specific required segments and provides EDI X12 message number and release version number.

Once mapping is done you do not have to recreate it again simply save it into the file with extension \*.xmp. You can run map

files using other utilities that come in the package

We recommend approach that usually gives good performance and is easy to understand: create as many SQL queries as

many main loops you will have in EDI X12 file. Some of our customers go to extreme by creating separate query for each EDI

X12 segment they will output. This will work but will result in many separate SQL queries and usually more queries you have

slower database connection gets. In other words, more separate queries – more work for the database to fetch all the data.

Granted you will not feel performance difference on few small EDI X12 files. But if your goal is to produce lots of big files performance turning is important. In this map we only use 3 queries: header, detail and detail2. NM1 segments do not form separate queries but are mapped to

different fields in the each query depending on the main loop NM1 is in.

Extracting the Data

It involves extracting the data from different heterogeneous data sources. Data extraction from a transactional system varies as per the requirement and the ETL tool in use. It is normally done by running scheduled jobs in off-business hours like running jobs at night or over the weekend.

Transforming the Data

It involves transforming the data into a suitable format that can be easily loaded into a DW system. Data transformation involves applying calculations, joins, and defining primary and foreign keys on the data. For example, if you want % of total revenue which is not in database, you will apply % formula in transformation and load the data. Similarly, if you have the first name and the last name of users in different columns, then you can apply a concatenate operation before loading the data. Some data doesn’t require any transformation; such data is known as direct move or pass through data.

Data transformation also involves data correction and cleansing of data, removing incorrect data, incomplete data formation, and fixing data errors. It also includes data integrity and formatting incompatible data before loading it into a DW system.

Loading the Data into a DW System

It involves loading the data into a DW system for analytical reporting and information. The target system can be a simple delimited flat file or a data warehouse.

ETL Tool Function

A typical ETL tool-based data warehouse uses staging area, data integration, and access layers to perform its functions. It’s normally a 3-layer architecture.

 Staging Layer – The staging layer or staging database is used to store the data extracted from different source data systems.

 Data Integration Layer – The integration layer transforms the data from the staging layer and moves the data to a database, where the data is arranged into hierarchical groups, often called dimensions, and into facts and aggregate facts. The combination of facts and dimensions tables in a DW system is called a schema.

 Access Layer – The access layer is used by end-users to retrieve the data for analytical reporting and information.

Here is a list of the common tasks involved in ETL Testing –

1. Understand the data to be used for reporting

2. Review the Data Model

3. Source to target mapping

4. Data checks on source data

5. Packages and schema validation

6. Data verification in the target system

7. Verification of data transformation calculations and aggregation rules

8. Sample data comparison between the source and the target system

9. Data integrity and quality checks in the target system

10. Performance testing on data

ETL Testing

ETL testing involves the following operations:

1. Validation of data movement from the source to the target system.

2. Verification of data count in the source and the target system.

3. Verifying data extraction, transformation as per requirement and expectation.

4. Verifying if table relations – joins and keys – are preserved during the transformation.

Common ETL testing tools include QuerySurge, Informatica, etc.

Database Testing

Database testing stresses more on data accuracy, correctness of data and valid values. It involves the following operations:

1. Verifying if primary and foreign keys are maintained.

2. Verifying if the columns in a table have valid data values.

3. Verifying data accuracy in columns. Example: Number of months column shouldn’t have a value greater than 12.

4. Verifying missing data in columns. Check if there are null columns which actually should have a valid value.

Common database testing tools include Selenium, QTP, etc.