**ETL Challenge: Converting SQL data to No-SQL data**

Team members: Shyam Sundar Ramamoorthy, Prasanth Prahladan

1. **Data cleaning:**

Some sql data has errors when importing to sql server. So, data was cleaned and successfully imported

1. **Analyzing relationship between tables:**

From the given sql database, we analyzed what kind of mapping are present between tables. We figured most of them are many-to-many mapping. Since MongoDB is a document store, we tried to convert all the tables to single collection.

MongoDB features:

1. Embedding Data:

When there is one-many mapping, data can be embedded in a document as a list

1. Referencing Data:

For many-many mapping, referencing is used to map data from one collection to other. Since sql data already has relationship, we came to solution that code for mapping is not needed.

1. **Tools used:**

Both tools are open-source.

1. Mongoimport:

To convert csv data to mongo document

1. Mongify:

To import sql data to mongo database. It translates ddl file(sql) to rb file(mongodb). It connects to sql server and mongo server and converts all tables present in the database. It also retains the relationship between data.

Mongify dependencies: ruby, activerecord

1. PyMongo:

To insert, update, find records (CRUD) operations using python programming.

1. **Scripts attached:**

script.sh – installs mongify, load sql data to sql server, creates ddl mapping, loads sql, csv data to mongo database, export mongo data to json

insert-update.py – script for testing incremental update operations

**References**:

1) RDBMS to MongoDB Migration Guide (A MongoDB White Paper)

2) <https://code.tutsplus.com/articles/mapping-relational-databases-and-sql-to-mongodb--net-35650>

3) <https://www.mongodb.com/webinars>

4) <https://www.mongodb.com/blog/post/transitioning-from-relational-databases-to-mongodb>