IS 631101

ENTERPRISE DATABASE MANAGEMENT

MSHA

Project Members:

Pranjali Telavane(pct9)

Agrim Sachdeva(as2936)

Prithvi Vadapally(pav23)

IS`-631 Project

Database:

**MSHA- Mine Safety and Health Administration**

About the Dataset:

The dataset contains 5 tables and more the 2.1 million data rows. The dataset we are using is of the Mine Safety and Health Administration which has the mine accident, inspection and violation data. This data is from the United States Department of Safety website. The data has the following details:

1. Mine ID.
2. Mine Name, curr\_mine\_nm, district and address of the Mine.
3. Latitude and longitude.
4. Other details like actv\_ind, inactv\_dt and naics\_cd related to the commodity.

**Proposed Database Model**

We have normalized the dataset into 5 tables which is linked to the main data table. The name of our normalized table is as follows:

1. msha\_commodity\_lookup
2. msha\_contractor
3. msha\_controller\_history
4. msha\_mine
5. msha\_inspection

Major areas that were normalized and consolidated were:

1. Contractor Details
2. Longitude, Latitude
3. Geolocation
4. Controller Details
5. State

**Management problem, including the business rules you choose to model and implement in your database design :**

* Initially we started with DDL creation from the data dictionary and ended up creating unwanted tables.
* We then imported all the data and then after running the stored procedure we create duplicate column names and inserted data in to different tables.
* We also created a cursor for master table where in the data from imported dataset is transferred to master table using normalization form.
* Using the address (city, state, zip code) we called google map geocode API to extract longitude and latitude of that address.
* We then update the data into geospatial data table.
* Using data inserted we calculated the geolocation using longitude and latitude from   
  Geospatial data table.
* We ask the user to insert the longitude and latitude calculate the distance.
* The VB code had to be macro-enabled to run without error.

**How you handle missing and unexpected data elements in the dataset?**

* The stored procedure name given for reference had different name than that of the one used in the database. So that had to be corrected for correcting the bad data.
* For missing data we incorporated google map geocode API. This google API gives free access for only first 2500 API calls. Due to which we only took data of one small state that is West Virginia having approx. 3880 rows.
* we implemented Geospatial data using the geographic data functions.
* Be that as it may, for unforeseen information components we utilized diverse functions accessible inside MYSQL server.

**Data Querying Performance:**

* The data querying performance was greatly impacted after the normalization of the data.
* Initially it used to take about 120 sec appx to retrieve data from 2.1 millions of records.
* After normalization it managed to reduce within 60 secs.