RESTAURANT DATABASE NON-RELATIONAL DATA

Introduction:

Non-relational data, unlike relational data, is unstructured data that cannot be stored in the tables of a relational database. They can be broadly classified into BLOB (Binary Large Objects) and CLOB (Character Large Objects). There are more complexities involved in storing and retrieving data of this kind within a database. They may be files such as '.jpg', '.png', '.txt', '.pdf', '.docx' etc. In our Restaurant database, we wanted to model the **Recipe** of the items on the restaurant menu as the non-relational data within a ".**txt**" file. For this, we first created the recipe as a BLOB data field in a standard "create table statement". After this we tried several techniques to handle (Store and Retrieve) the ".txt" files within the MySQL server. Although we ran into many hurdles, we learnt a great deal about non-traditional data/non-relational data The following section Discusses the commands we came across in order to store and retrieve non-relational data within the MYSQL Server.

Experiments: To insert and handle non-relational data, here are a few commands we attempted:

- set recipe=LOAD_FILE('/usa/abhay/data.txt')
  where item_name='chowmein';

  LOAD_FILE: Although the query gets executed, the data.txt did not get inserted into the class server due to permission issues.

- mysql -e "update menu_items set recipe= x'$(xxd -p /usa/abhayh/data.txt | tr -d \\n)'
  where item_name='chowmein'" -u tejaswi -p restaurantdb;

  Running the above command on the cisc637 server, we could successfully add the text data present within 'data.txt' directly into the recipe column into the database. For instance. The same is achieved by running Blob.java.

Item_name	Type	Ingredient1	Ingredient2	Ingredient3	Recipe
chowmein	chinese	Chowmein noodles	chicken	Hotsauce	To make the **chow mein** sauce, put the water, oyster sauce and soy ……..

- Also, We tried the utl_raw.cast_to_raw" function, but it was not supported on the MySQL server.