Design Document

Individual Project – CS6360.5UI Database Design

Name – Biswadip Mandal; NetId – BXM200000

Schema:

The schema is the same as given in the assignment description.

Technologies:

The Django frame and its powerful ORM features has been used to build the schema as well handling the end user interaction.

Inserting data in the Tables:

There are four tables – *Contact, Address, Phone, Date*. While parsing the given csv, each row is processed to insert in the four tables. The dataset is loaded using the pandas read_csv feature and each row is parsed while validating the rows.

Contact Table

One row is inserted for each row in the csv. Only the contact id and first name are mandatory field for this table.

Address Table

Address information is inserted based on the data availability. Ideally, two rows for home and work address are inserted in the table for each contact. If one or both addresses are not available, 1 or zero rows are inserted respectively to the *Address* table. Note that for an address to be available, at-least one of the address attributes (i.e., state, city) has to be available. `address_type` is a mandatory field for this table and allow any input from (work, home).

Phone Table

Phone information is inserted in a similar way as address. Maximum three and minimum zero rows can be inserted for one contact. Single record can have a home phone, cell phone and work phone. `area_code` is inferred from the phone number automatically while saving any row to the Phone Table. `phone_type` is a mandatory field for this table and allow any input from (cell, work, home).

Date Table

Date is also inserted in the same way as address. Maximum one and minimum zero rows can be inserted for one contact. `date_type` is a mandatory field for this table. While inserting the initial records, the `date_type` can be `birth`.

Saving Space

It is worth noting that not all information for each user is available. For example, only 645 contacts out of 1000 have work phone numbers. Having a separate *Phone* Table saves memory as only 645 rows for work phone number is created in the *Phone* table

Functionalities

After the initial processing and insertion is done, searching, adding, updating, deleting and viewing record follows the following logics:

Searching:
When searching for any record, a filter query is performed in Contact Table, Address Table and Phone Table. All fields are searched except the primary keys (contact id, address id) and columns that not record-specific (i.e., address type). While filtering few column values are matched exactly with the search query while few others are matched even if the search query is part of the value (i.e., the search query `Biswa` would consider records where first name is Biswadip). The decision for whether to do an exact search or part-matching search is done based on various facts such as typical length of the value (area code is only of length 3, so searching only with two characters provides little selectivity) or specific attribute properties (Names, city names are often abbreviated and hence a part-matching search makes more sense).

Add New contact:
Adding new contact is very simple. If the required columns are not present, the query is rejected. Data values for each table is validated before inserting. The contact id is mandatory and required to be provided by the user while adding a new contact.

Updating Contact:
Updating works the same way as add. However, for address type only one entry for each address type is allowed for every user. For an example, a contact cannot have multiple work address. Hence, when a new address for a contact id comes with address type work, the existing work address is updated. Also note, if a work address doesn't already exist, a new entry is created in the *Address* Table while updating work address.

Deleting Contact:
Deleting is performed by finding the row with input contact id and deleting the row and cascading the delete across the database.

View Contact:
For viewing a record, all the tables are filtered by the contact id and the filtered information is mapped to the view display form fields to display available information.

Key Assumption

For the current table, a contact can have a single work address and a single home address. I have assumed a person doesn’t have multiple workplaces or multiple home addresses for simplicity. This was done to accommodate the static display of all information of a user. Also, there is no mention of other data types and hence I have decided to view only the birth date on view page. Users are however allowed to enter other forms of date (i.e., anniversary date)