SQL Programming Project

CS-6360 Database Design, Fall 18

VXR180002 - Vishal Chandar Ramachandran

Design Document

Database Contains 4 tables (models in Django):

1. Contact_information
2. Address_information
3. Phone_number_information
4. Dates_information

Assumptions are made as given in the problem statement and data is loaded into these 4 tables using the master data given in Contacts.csv

Each of these 4 tables are created as model classes in Django. I have adopted the schema design given in the assignment instructions.

How data was loaded/updated into the 4 tables:

1. Created a master table with all data in Contacts.csv provided
2. Create contact_information table and update the Django model by picking only attributes as given in the contacts table in the assignment instructions and save these rows into the table using Django.
5. Create address_information table and update the Django model by picking only attributes as given in the address table in the assignment instructions and save these rows into the table using Django.
3. Create phone_number_information relationand update the Django model by picking only attributes as given in the phonenumber table in the assignment instructions and save these rows into the table using Django.
4. Create date_information relationand update the Django model by picking only attributes as given in the date table in the assignment instructions and save these rows into the table using Django.
5. Foreign keys are linked where required as mentioned in assignment instructions.
6. New entries will go to individual tables
7. Updating entries will happen in the existing entries
8. All row details are cascaded to maintain integrity
9. Edit and delete options for each are provided in the GUI
10. Auto increment of specific keys is provided as required in assignment instructions.

How Django works:

1. Django works on models, views, templates and routes
2. Each model is a table in the database
3. Each route is a url endpoint eg. /app/endpoint
4. Each route is linked to a view.
5. On hitting the route in the browser, the call/data is transferred to the view which is a function in Python.
6. This view decides what to return to the browser window.
7. View returns a html template that couples design and data together
8. This data bound template is displayed to the user.
9. Both GET and POST calls are supported a view is called
10. GET merely fetches data from the view
11. POST sends some data to the view
12. GET does not change the current state of the model/database.
13. POST provides data that could be used to change the current state of the model/database.

How view and model was coupled in this assignment:

1. I have used a basic form template available in Django to display the various rows/table content in the database
2. /all_contacts is an endpoint that displays all contact information in table form.
3. The view fetches all data in the contacts table from the database
4. This data is then modeled into a table in the html template.
5. This html is returned to the browser and a formatted table is displayed.
6. Similarly, where contents are displayed in the form of a table, this above-mentioned logic was used.

How updates and insert is handled in this assignment:

1. Similar to how a database table contents is displayed in the form of table, a row in the table is presented to edit/ create newly by the user.
2. In case of modify, an existing row from the model/table is provided as a form with its attributes and containing data for the user to edit.
3. Upon changing contents of the form and confirming submission, the form is then validated against the model schema in Python(which is the table schema in the database) and then a save (or update in database terms) is performed on this object onto the database.
4. In case of insert new data into a model/table, a new form containing attributes of the desired model.