

1. How to compile and run this program

- 1) To run the code first explicitly compile DataAccessLayer.java using:

```
javac -cp mysql-connector.jar:. DataAccessLayer.java
```

- 2) Next compile the Main file using:

```
javac Main.java
```

- 3) To run the code use:

```
java -cp mysql-connector.jar:. Main -i or -q
```

2. What does this program do

This program is designed to add some random data into a schema for Bank accounts and transactions and satisfy 3 different types of queries.

The program uses an sql file to create the database schema. The -i option is used to add randomly synthesised sensible data to the database i.e. 50 users, their user credentials that include address information, names, contact numbers, PAN card numbers etc.. Next it generates some random transactions in the following format.

After adding the data once it wont allow user to corrupt the database by adding more users. For that checkData() function does it work and notifies the user.

And with -q option it will take user arguments for queries and give generated output.

All the outputs are sent by DataAccessLayer to Main using Data Transfer Objects i.e. CategoryExDTO and TransactionDTO.

3. A description of how this program works (i.e. its logic)

The program has been divided into several different functions:

-
- 1) `checkData()` : check if data already exists and does not allow further addition as it would corrupt the usage
 - 2) `getNewDummyNames()` : reads two text files containing names and surnames and generates 50 random combinations to be used in the database.
 - 3) `getNewDummyAddress()` : This function populates 3 tables , City, State and Country in a normalized(updated from given format) manner. First it adds countries, then while adding states assigns random CountryIDs from the previous table to all the states. Similarly for Cities random StateIDs are assigned.
 - 4) `addNewContacts()` : generates and populates random contacts into the `contact_information` table following 10 users per city condition.
 - 5) `addNewTransactions()`: generates transactions in the following manner:
The accounts are created at random but on serialized dates, for account 1 no transaction is possible as no other account has yet been generated, for account 2 one transaction is generated with account 1, for account 3, 2 transactions are generated one with account 2 and another with account 1. Similarly account 4 generates 3 transactions with account 3,2 and 1 and so on.
The transaction begins from Opening Date and goes till 5 days.
This will take a good amount of time to run as it is a huge number of transactions.
 - 6) `RandomDate`, `RandomPAN`, `RandomContact` and `RandomPostal` are used to generate the random data.
 - 7) `Cities.txt` , `states.txt`, `countries.txt` , `transaction_categories.txt`, `dummy_names.txt`, `dummy_sur_names.txt` are used for legible data creation.

Function Flow:

- 1) `getNewDummyAddress()`
- 2) `app.dal.addNewContacts(cities)`
- 3) `addNewUsers()` calls `getNewDummyNames()` for names
- 4) `addNewBankAccounts()`
- 5) `addNewTransactions();`
- 6) `queryExpenditure2()`

Sample Outputs:

```

prerna@prerna-pc:~/Desktop/src/csl310$ javac -cp mysql-connector.jar:. DataAccessLayer.java
prerna@prerna-pc:~/Desktop/src/csl310$ javac Main.java
prerna@prerna-pc:~/Desktop/src/csl310$ java -cp mysql-connector.jar:. Main -i

Enter DB login: root
Enter DB password:
New Addresses Generated
New Contacts Added
New Names Generated
New Users Added
New Bank Accounts Generated
Generating Transactions...
6125 new Transactions Added
12250 entries entered in Database
prerna@prerna-pc:~/Desktop/src/csl310$ java -cp mysql-connector.jar:. Main -i

Enter DB login: root
Enter DB password:
50 users and corresponding data already present.
prerna@prerna-pc:~/Desktop/src/csl310$ java -cp mysql-connector.jar:. Main -q

Enter DB login: root
Enter DB password:
Enter 1 to print the transactions recorded for a given account between given two dates.
Enter 2 to print Category wise spending of a given person in a given month.
Enter 3 to print Category wise spending in a given month by all persons (taken together) who live in a given city.
Your Selection:
1
Enter AccountID: 4
Enter start date Format(yyyy-mm-dd): 1941-09-08
Enter end date: Format(yyyy-mm-dd): 1942-09-23
Results fetched for AccountID 4 are:
ID: 8276 Type: Credit Date: 1941-10-27 Amount: 3994 Account: 4 Category: medical Remarks: Successful :)
ID: 8358 Type: Credit Date: 1941-10-28 Amount: 1487 Account: 4 Category: salary Remarks: Successful :)
ID: 8440 Type: Credit Date: 1941-10-29 Amount: 554 Account: 4 Category: dining Remarks: Successful :)
ID: 8522 Type: Credit Date: 1941-10-30 Amount: 206 Account: 4 Category: salary Remarks: Successful :)
ID: 8604 Type: Credit Date: 1941-10-31 Amount: 77 Account: 4 Category: entertainment Remarks: Successful :)
ID: 8688 Type: Credit Date: 1942-04-02 Amount: 3501 Account: 4 Category: Tax Remarks: Successful :)
ID: 8772 Type: Credit Date: 1942-04-03 Amount: 1303 Account: 4 Category: Tax Remarks: Successful :)
ID: 8856 Type: Credit Date: 1942-04-04 Amount: 485 Account: 4 Category: Tax Remarks: Successful :)
ID: 8940 Type: Credit Date: 1942-04-05 Amount: 181 Account: 4 Category: dining Remarks: Successful :)
ID: 9024 Type: Credit Date: 1942-04-06 Amount: 67 Account: 4 Category: money transfer Remarks: Successful :)

prerna@prerna-pc:~/Desktop/src/csl310$ java -cp mysql-connector.jar:. Main -q

Enter DB login: root
Enter DB password:
Enter 1 to print the transactions recorded for a given account between given two dates.
Enter 2 to print Category wise spending of a given person in a given month.
Enter 3 to print Category wise spending in a given month by all persons (taken together) who live in a given city.
Your Selection:
2
Enter Person ID: 4
Enter month: 3
Results fetched for PersonID 4 for month: 3 are:
Category : dining Expenditure : 2052
Category : entertainment Expenditure : 7108
Category : grocery Expenditure : 24447
Category : medical Expenditure : 18714
Category : money transfer Expenditure : 964
Category : phone bill Expenditure : 3169
Category : salary Expenditure : 2817
Category : Tax Expenditure : 739
prerna@prerna-pc:~/Desktop/src/csl310$ java -cp mysql-connector.jar:. Main -q

Enter DB login: root
Enter DB password:
Enter 1 to print the transactions recorded for a given account between given two dates.
Enter 2 to print Category wise spending of a given person in a given month.
Enter 3 to print Category wise spending in a given month by all persons (taken together) who live in a given city.
Your Selection:
3
Enter month: 3
Enter cityName Shinjuku
Results fetched for City Shinjuku for month 3 are:
Category : dining Expenditure : 28832
Category : entertainment Expenditure : 14831
Category : grocery Expenditure : 28668
Category : medical Expenditure : 13017
Category : money transfer Expenditure : 13480
Category : phone bill Expenditure : 20678
Category : salary Expenditure : 15314
Category : Tax Expenditure : 29111
prerna@prerna-pc:~/Desktop/src/csl310$

```