Assignment 1

Deadline: *22 January, 2016, midnight*

1 BUILD AND POPULATE

A banking information system is abstracted into the following 6 tables (given along with their schema). The primary keys and foreign keys of each of the tables is underlined and stated.

- **Branch(Bname, Bcity , Assets)**
  Bname is the primary key.
- **Customer(Cname , City , Cid)**
  Cid is the primary key.
- **Depositor(Cid , AccNo)**
  Multiple accounts for a single Cid is possible. **Cid** is a foreign key referencing **Customer**. **AccNo** is a foreign key referencing **Account.AccNo**.
- **Borrower(Cid , LoanNo)**
  Multiple loans for a single Cid possible. **Cid** is a foreign key referencing **Customer**. **LoanNo** is a foreign key referencing **Loan.LoanNo**.
- **Account(AccNo , Bname , Balance)**
  AccNo is the primary key. **Bname** is a foreign key referencing **Branch**.
- **Loan(LoanNo, Bname, Amount)**
  LoanNo is the primary key. **Bname** is a foreign key referencing **Branch**.

You are required to build the above tables as your bank database. Populate the tables whilst adhering to the necessary constraints. Automating the process is advised.

2 QUERIES

Design SQL statements for the following queries and updates to the database. You should run your queries on your sample database using the MySQL (or equivalent) system. Submit your queries in a .txt file on the course page on moodle.cse.iitk.ac.in. Each query should end with a semi-colon. Also submit the details of the database which you created (number of records in each table.)

1. Find customers (Cname) having at most one account in branch with *Bname* 'IIT Kanpur.
2. Find customers(Cname) having at most one account in every branch they have an account in, that is, they shouldn't have more than one accounts at any single branch. Do not include customers who do not
3. Find customers having accounts in every branch of the bank.
4. Find the avg account balance of 'IIT Kanpur' branch.
5. Find customers who have at least one account or at least one loan but not both.
6. Find all pairs of customers who have exactly the same number of accounts and the same number of loans in each branch.
7. Update balance of all accounts (AccNo) through all branches as follows:
   5% to all accounts with Balance < X.
   10% increment to the remaining.
   Choose X to ensure a fairly even split based on how you populate your database.