

# Programming Assignment 1

CSE 132A Fall 2019

Due: by midnight on Friday, Nov 1 (electronically via gradescope)

Recall our schema of a bank:

```
customer (name: string, credit: integer)
loan (no: string, type: string, minCredit: integer)
borrower (cname: string, lno: string, due: date)
```

where borrower.cname and borrower.lno are foreign keys referencing customer, respectively loan, whose keys are name, respectively no (number). Attribute loan.minCredit indicates the minimum credit required of a customer to qualify for that loan.

1. Write the following queries in SQL. Make sure that the output contains no duplicates.
  - (a) Customers' credit rating may decrease after they got a loan. Find (customer,loan) pairs where the customer no longer satisfies the minimum credit required by the loan.  
The output schema should be (name, no).
  - (b) Find for each customer the number of loans she took. If a customer took no loans, list her name with a count of zero. The output schema should be (name, loanCount).
  - (c) List the loan type(s) with the smallest number of borrowers. The output schema should be (loanType).
  - (d) List the customers who took every *type* of loan (at least one loan from every type). The output schema should be (name).
  - (e) List the loans who have a strictly greater number of borrowers than the average number of borrowers over all loans of that type. The output schema should be (no).
  - (f) Find the pairs of customers who took the exact same loans (i.e. all loans taken by one are taken by the other and viceversa). The output schema should be (name1, name2).  
The answer should contain exactly one tuple for each pair of customers who take exactly the same loans. For each pair of such customers, always choose the tuple that is lexicographically smaller (e.g. "(Jane,John)" but not "(John,Jane)").
2. Formulate the following updates in SQL. You may use a sequence of update commands but you are not allowed to change the schema of the database.
  - (a) Change the type of all "jumbo" loans to "student" and the type of all *original* "student" loans to "jumbo".
  - (b) Delete all "jumbo" loans as well as the customers who took them. Do this carefully to avoid violating the referential integrity constraints.

*What to turn in:* Turn in two SQL files. They should be named 'query.sql' and 'update.sql'. Each file should include all SQL statements for the corresponding exercise and should be an executable Postgres script (to this end, make sure to finish each statement with a semicolon (;)).

# Programming Assignment 1

CSE 132A Fall 2019

Due: by midnight on Friday, Nov 1 (electronically via gradescope)

Recall our schema of a bank:

```
customer (name: string, credit: integer)
loan (no: string, type: string, minCredit: integer)
borrower (cname: string, lno: string, due: date)
```

where borrower.cname and borrower.lno are foreign keys referencing customer, respectively loan, whose keys are name, respectively no (number). Attribute loan.minCredit indicates the minimum credit required of a customer to qualify for that loan.

1. Write the following queries in SQL. Make sure that the output contains no duplicates.
  - (a) Customers' credit rating may decrease after they got a loan. Find (customer,loan) pairs where the customer no longer satisfies the minimum credit required by the loan.  
The output schema should be (name, no).
  - (b) Find for each customer the number of loans she took. If a customer took no loans, list her name with a count of zero. The output schema should be (name, loanCount).
  - (c) List the loan type(s) with the smallest number of borrowers. The output schema should be (loanType).
  - (d) List the customers who took every *type* of loan (at least one loan from every type). The output schema should be (name).
  - (e) List the loans who have a strictly greater number of borrowers than the average number of borrowers over all loans of that type. The output schema should be (no).
  - (f) Find the pairs of customers who took the exact same loans (i.e. all loans taken by one are taken by the other and viceversa). The output schema should be (name1, name2).  
The answer should contain exactly one tuple for each pair of customers who take exactly the same loans. For each pair of such customers, always choose the tuple that is lexicographically smaller (e.g. "(Jane,John)" but not "(John,Jane)").
2. Formulate the following updates in SQL. You may use a sequence of update commands but you are not allowed to change the schema of the database.
  - (a) Change the type of all "jumbo" loans to "student" and the type of all *original* "student" loans to "jumbo".
  - (b) Delete all "jumbo" loans as well as the customers who took them. Do this carefully to avoid violating the referential integrity constraints.

*What to turn in:* Turn in two SQL files. They should be named 'query.sql' and 'update.sql'. Each file should include all SQL statements for the corresponding exercise and should be an executable Postgres script (to this end, make sure to finish each statement with a semicolon (;)).

$$a \in \mathbb{R} \setminus \{27\}$$