T-202-GAG1: Exercise 05

Readings

Ramakrishnan & Gehrke: Chapters 3 (esp. 3.6), 5 (esp. 5.8-5.9), 16.6.1.

Preparation

Create a new database called Exercise05, and use the associated scripts to create and populate the database. The command script CREATE.sql first drops and then creates four tables: People (P), Bills (B), Accounts (A), and AccountRecords (R). The script FILL.sql then inserts 200 individuals into the People table, as well as many records into Accounts, AccountRecords and Bills.

The Accounts and AccountRecords tables are an example of master-detail design, where the Accounts table contains (very simple) metadata about accounts, while the AccountRecords table contains detailed records of the accounts.

The intention is that aBalance is the current amount of the account and that aDate is the date of the last change to the account. Then, for each record, rAmount should be the amount of the transaction (deposit, withdrawal) of the account, rDate should be the date of that transaction, and rBalance should be the state of the account following the transaction. Thus, the following two queries should return the same data for any account *X*:

```
SELECT A.aDate, A.aBalance
FROM Accounts A
WHERE A.AID = X;

SELECT R.rDate, R.rBalance
FROM AccountRecords R
WHERE R.AID = X
  AND R.RID = (SELECT MAX(R1.RID)
               FROM AccountRecords R1
               WHERE R1.AID = R.AID);
```

Accounts can be overdrawn to the amount A.aOver (i.e., A.aBalance ≥ –A.aOver).

AccountRecords contains all transactions made on each account. Each record includes an R.aType attribute, which denotes the type of the record:

- B for Bills
- T for Transfers (Deposit or Withdrawal).
- L for loans.
- O for other.

All records due to bills will therefore have R.aType value of B. When a loan is granted and deposited into the account, the account record will be of R.aType L.

The Bills table contains both paid and unpaid bills for a person. The B.bIsPaid attribute indicates if the person has paid the bill or not. A bill always has a positive amount, but when paid, the same negative amount should be deducted from an account. A bill also has a B.bDueDate, which is the date it should be paid.

To signal an error in a trigger, or a function in general, you can use the command RAISE EXCEPTION, for example:

```
RAISE EXCEPTION 'Height cannot be negative!'
USING ERRCODE = '45000';
```

See the following web-page for more details:
https://www.postgresql.org/docs/current/static/plpgsql-errors-and-messages.html

The Assignment

For this assignment, you can work in groups of two or three. **Do as many of the following tasks as you can in the span of 90 minutes.** If you find that you could not complete all the tasks, then finish the assignment **later**, when you have time, as these are very useful exercises.

1. Create a **view** to combine the information from the People and AccountRecords tables. Use this view to select all records for a particular individual, ordered by AID and RID.
2. Create a **procedure** to insert a new account. (This could be used to grant access to certain people for creating accounts.)
3. Create a **trigger** on the People table to a) ensure that gender is correct (either 'M' or 'F') and b) that height is > 0. If either is not correct, then abort the transaction.
4. Create a **procedure** to find all accounts that do not satisfy the constraints on aDate, aBalance, rDate and rBalance above.

   The data should have one such account. You can also test this by modifying data in Accounts or AccountRecords directly.

   Note: To do this well, you need to worry about accounts with aBalance = 0 and no entries in AccountRecords.
5. Create an **integrity constraint** (foreign key, CHECK constraint, or trigger) to ensure that only account records of type 'B', 'T', 'L' and 'O' can be created.

Deliverables

Submit one text file (*.sql) containing the SQL code that you wrote for the assignment. Use comments to briefly explain what each code construct (procedure, trigger, …) does.

Additional Exercises

Exercises: 5.10; finish the rest of this assignment; then use your imagination and experience and make up your own exercises.