

## 1 Description

Read the whole description before you start:

A Bank asks you to design a database to store all informations about that Bank. The bank has different branches. Important information about the branches is the number of employees, the number of accounts, the city and the manger ID. Every branch is rolled by a manger. A manager might be responsible of several branches. In one city, there might be several branches. However, no manger roles two branches in the same city.

Every client is of age at least 18. For every client, we want to store information of name, address and age. A client can have several accounts. In addition, there are two types of special accounts: shared (two DIFFERENT clients have the same account) and VIP accounts. Every client can have at most 3 accounts (no matter they are normal, shared or VIP). For every account, we want to store the owner(s) and the balance (in Euro). For shared accounts, we also want to store the relation between the two owners and both of the owner must be clients. The VIP must save money in different currencies (only dollar, yuan, British pounds and at least one of them). We want to store information about each currency and the balance. In addition, the balance of a VIP account must be at least 10.000 of any currency (except Euro). There might be a client that has no account.

Every client has exactly one contact person. This person is an employee in the bank (also, a manager is an employee). We want to store information about this employee, the name, salary, the manger of this employee and the clients that this person is responsible of. When a manger of a branch is changed, all of the employees become under the role of the new manger.

Every transfer that a client sends must be in Euro. The same account can not send twice in the same date. The amount must be at most 2000 Euro. Every time a transfer from an account is done, the balance of the account decreases the same amount (automatically).

## 2 The Task

We want to design a database that involves all information above. There are of course several schema that serve this goal. You need to design a schema that avoids redundancy, update anomaly and deletion anomaly. After that, you design a UML diagram that describes the schema. After that you write the SQL scripts with inserting at least two or three examples for every table.

### 3 Submission

1. You need to submit two files. The first one is PDF which draws the UML diagram you designed (choose any software to draw. A suggestion is ipe).
2. The second file is .sql that has the scripts to create this database. In .sql file, put comments to describe every part of your scripts.
3. You can submit the project in groups. Every group can submit one project (then, all group member have the same grade). Every group is at most 4 members. Every group must register in this file. Open it using your browser (Open in ONLYOFFICE), not Excel.
4. You can send me your project via my email ([george.krait@inria.fr](mailto:george.krait@inria.fr)). The deadline is Jan 25. I will receive the projects until 12 pm (at noon). After that, no project is accepted.