BDD : Agence Bancaire

**Membres du projet :**

Batisse Dylann

Brault Yann

Cousson Antoine

Sabatier Juliette

Table des matières

[1 - Description du sujet 2](#_Toc92613408)

[2- Description des données 2](#_Toc92613409)

[2-1 Dictionnaire des données 2](#_Toc92613410)

[2-2 Description textuelle des associations 4](#_Toc92613411)

[2-3 Modèle Entité-Association 5](#_Toc92613412)

[2-4 Méthodes des types 5](#_Toc92613413)

[3 – Requêtes 7](#_Toc92613414)

[3-1 Requêtes de mises à jour 7](#_Toc92613415)

[3-1-1 Requêtes impliquant une table 7](#_Toc92613416)

[3-1-2 Requêtes impliquant deux tables 8](#_Toc92613417)

[3-1-3 Requêtes impliquant plus de deux tables 10](#_Toc92613418)

[3-2 Requêtes de suppression 11](#_Toc92613419)

[3-1-1 Requêtes impliquant une table 11](#_Toc92613420)

[3-1-2 Requêtes impliquant deux tables 11](#_Toc92613421)

[3-1-3 Requêtes impliquant plus de deux tables 12](#_Toc92613422)

[3-3 Requêtes de consultation 13](#_Toc92613423)

[3-1-1 Requêtes impliquant une table 13](#_Toc92613424)

[3-1-2 Requêtes impliquant deux tables 13](#_Toc92613425)

[3-1-3 Requêtes impliquant plus de deux tables 14](#_Toc92613426)

# 1 - Description du sujet

La base de données de notre banque permet la création et gestion de comptes bancaires dans plusieurs agences bancaires localisées un peu partout dans le monde, mais aussi la gestion des employés de ces mêmes agences. Elle permet aussi de conserver l’historique des transactions faites d’un compte bancaire d’un client à un autre.

# 2- Description des données

## 2-1 Dictionnaire des données

**Types :**

A -> Alphanumérique

N -> Numérique

AN -> Alphanumérique (ou numérique sans calcul)

D -> Date

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Code Mnémonique** | **Description** | **Types** | **Format des données** | **Taille** | **Contraintes** |
| country | Pays d’une adresse | A | Varchar2 | 50 |  |
| city | Ville d’une adresse | A | Varchar2 | 50 |  |
| streetName | Rue d’une adresse | A | Varchar2 | 100 |  |
| streetNo | Numéro de rue d’une adresse | AN | number | 5 | Between 1 and 99999 |
| empNo | Numéro d’un employé | AN | number | 8 | Primary key |
| eName | Nom de l’employé | A | Varchar2 | 15 |  |
| prenoms | Prénoms de l’employé | A | Varray(varchar2) |  |  |
| job | Travail de l’employé | A | Varchar2 | 20 | 'Ingenieur', 'Banquier', 'Vigile', 'Avocat’, 'Secretaire', 'Directeur', 'Analyste', 'PDG' |
| sal | Salaire de l’employé | N | number | 7 ;2 |  |
| cv | CV de l’employé |  | CLOB |  |  |
| birthDate | Date de naissance de l’employé | D | date |  |  |
| employementDate | Date d’embauche de l’employé | D | date |  | >birthDate |
| agencyNo | Numéro de l’agence | AN | number | 4 | Primary key |
| aName | Nom de l’agence | A | Varchar2 | 30 |  |
| tNum | Numéro identifiant la transaction | AN | number | 8 | Primary key |
| amount | Montant de la transaction | N | number | 11 ;4 | Primary key |
| accountNo | Numéro du compte bancaire | AN | number | 4 | Primary key |
| accountType | Type de compte | A | Varchar2 | 20 | 'Livret A', 'Compte Epargne', 'Compte Courant', 'PEL', 'PEL Pro' |
| balance | Solde du compte | A | number | 10 ;4 |  |
| bankCeilling | Plafond du compte | N | number | 10 ;4 |  |
| numCli | Numéro identifiant de client | AN | number | 8 | Primary key |
| cName | Nom du client | A | Varchar2 | 15 |  |
| job | Travail du client | A | Varchar2 | 15 |  |
| Sal | Salaire du client | N | number | 8 |  |
| birthDate | Date de naissance du client | D | date |  | JJ.MM.AAAA |
| project | Project du client | CLOB |  |  |  |
| tNum | Identifant de la transaction | AN | number | 8 | Primary key |
| Amount | Montant de la transaction | N | number | 11 ;4 |  |

## 2-2 Description textuelle des associations

**Location :**

**Client :**

* Un client est forcément client dans une unique agence
* Un client possède 0, un ou plusieurs comptes bancaires

**Agence :**

* Une agence est localisée à une unique location
* Une agence possède 0, un ou plusieurs clients
* Une agence possède 0, un ou plusieurs employés

**Transaction :**

* Une transaction est faite par 2 clients
* Une transaction est faite par 2 comptes bancaires

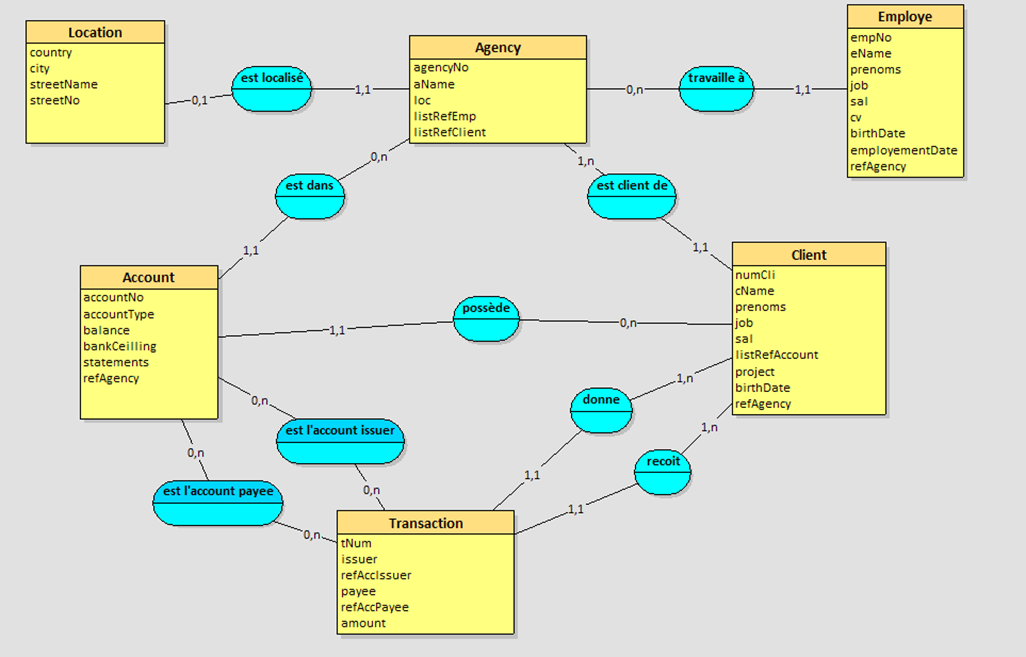
**Employé :**

* Un employé travaille dans une unique agence

**Compte bancaire :**

* Un compte est lié à une unique agence
* Un compte possède 0, une ou plusieurs transactions

## 2-3 Modèle Entité-Association



## 2-4 Méthodes des types

**Location :**

* Méthode d’ordre
  + compLoc
* Méthode de gestion des liens N
  + deleteLoc
* Méthodes applicatives de consultation
  + getLocStatic
  + getCountryStatic
* Méthodes CRUD
  + getCountry, getCity, getStreetName, getStreetNo
  + updateCountry, updateCity, updateStreetName, updateStreetNo

**Client :**

* Méthode d’ordre
  + compCli
* Méthode de gestions des liens N
  + deleteClient
  + getAccounts
  + getAgency
  + addLinkListAccount, deleteLinkListAccount, updateLinkListAccount
* Méthodes applicatives de consultations
  + getClientStatic
  + getAgencyStatic
  + getAccountsStatic
* Méthodes CRUD
  + getCName, getNumCli, getJob, getSal, getProject, getBirthDate
  + updateCName, updatePrenoms, updateJob, updateSal, updateProject

**Agency :**

* Méthodes d’ordre
  + compAgency
* Méthodes de gestions des liens N
  + updateLoc
  + addLinkListEmploye, deleteLinkListEmploye, updateLinkListEmploye
  + addLinkListClient, deleteLinkListClient, updateLinkListClient
* Méthodes applicatives de consultations
  + getAgencyStatic
  + getLocationStatic
  + getInfoEmpStatic
* Méthodes CRUD
  + getAgencyNo, getAName, getLoc, getEmployes
  + updateAName

**Transaction :**

* Méthodes d’ordre
  + compTransaction
* Méthodes de gestions des liens N
* Méthodes applicatives de consultations
  + getTransactionStatic
  + getIssuerStatic
  + getPayeeStatic
* Méthodes CRUD
  + getTNum, getIssuer, getPayee, getAmount, getRefAccIssuer, getRefAccPayee

**Employe :**

* Méthodes d’ordre
* Méthodes de gestions des liens N
* Méthodes applicatives de consultations
* Méthodes CRUD

**Account :**

* Méthodes d’ordre
  + compAccount
* Méthodes de gestions des liens N
  + updateAgency
  + addLinkListTransaction, deleteLinkListTransaction, updateLinkListTransaction
* Méthodes applicatives de consultations
  + getAccountStatic
  + getAgencyStatic
  + getStatementsStatic
* Méthodes CRUD
  + getAccountNo, getAccountType, getBalance, getBankCeilling, getAgency, getStatements
  + updateBankCeilling , updateBalance

# 3 – Requêtes

## 3-1 Requêtes de mises à jour

### 3-1-1 Requêtes impliquant une table

Requête 1 :

        declare

            loc location\_t;

        begin

            select value(lo) into loc from

            o\_location lo where lo.country='France' and lo.city='Paris' and lo.streetName='Rue de la paix' and lo.streetNo=8;

            loc.updateCountry('a');

        end;

        /

Requête 2 :

        declare

            loc location\_t;

        begin

            select value(lo) into loc from

            o\_location lo where lo.country='France' and lo.city='Paris' and lo.streetName='Rue de la paix';

            loc.updateStreetName('Place Vendome');

        end;

        /

Requête 3 :

        declare

        cli client\_t;

        begin

            select value(oc) into cli from

            o\_client oc where oc.numCli='7';

            cli.updateJob('Directeur');

            cli.updateSal(5000);

        end;

        /

### 3-1-2 Requêtes impliquant deux tables

Requête 1 :

        DECLARE

        refOe1 ref EMPLOYE\_T;

        agency agency\_t;

        refAgency ref agency\_t;

        agency2 agency\_t;

        refAgency2 ref agency\_t;

        refOe2 ref employe\_t;

        emp1 employe\_t;

        emp2 employe\_t;

        begin

            select value(oa), ref(oa) into agency, refAgency

            from o\_agency oa where oa.agencyNo=1;

            select value(oa), ref(oa) into agency2, refAgency2

            from o\_agency oa where oa.agencyNo=2;

            select value(oe), ref(oe) into emp1, refOe1

            from o\_employe oe where oe.empNo=1;

            select value(oe), ref(oe) into emp2, refOe2

            from o\_employe oe where oe.empNo=3;

            emp1.updateAgency(null);

            emp2.updateAgency(refAgency);

            agency.updateLinkListEmploye(refOe1, refOe2);

            agency2.deleteLinkListEmploye(refOe2);

        end;

        /

Requête 2 :

        DECLARE

        refOe ref EMPLOYE\_T;

        agency agency\_t;

        refAgency ref agency\_t;

        emp employe\_t := employe\_t( 25,  'DEBREUILLE',  TABPRENOMS\_T('Louis'), 'Vigile', 1500, Empty\_clob(),

            to\_date('11-11-1970','DD-MM-YYYY'),  to\_date('11-12-2012','DD-MM-YYYY'),  refAgency );

        begin

            select value(oa), ref(oa) into agency, refAgency

            from o\_agency oa where oa.agencyNo=4;

            emp.refAgency:=refAgency;

            insert into o\_employe oe

            values(emp) returning ref(oe) into refOe;

            agency.addLinkListEmploye(refOe);

        end;

        /

### 3-1-3 Requêtes impliquant plus de deux tables

Requête 1 :

        update (select \* from o\_transaction ot

        where ot.amount > 3000 and ot.issuer.job='Ecrivain'

        and ot.issuer.refAgency.loc.country='France') set amount=5000;

        /

Requête 2 :

        update (select \* from o\_employe oe

        where oe.empNo>5 and oe.refAgency.agencyNo>5

        and oe.refAgency.loc.country='USA') set sal=10000;

        /

## 3-2 Requêtes de suppression

### 3-1-1 Requêtes impliquant une table

Requête 1 :

        delete from o\_client oc where oc.numCli=11;

        /

Requête 2 :

        delete from o\_agency oa where oa.agencyNo=4;

        /

### 3-1-2 Requêtes impliquant deux tables

Requête 1 :

        declare

        agency agency\_t;

        refOe ref employe\_t;

        begin

        select value(ag) into agency from

        o\_agency ag where ag.agencyNo=1;

        update o\_employe oe set oe.refAgency=null

        where oe.refAgency.agencyNo=1 and oe.empNo=1 returning ref(oe) into refOe;

        agency.deleteLinkListEmploye(refOe);

        end;

        /

**Requête 2 :**

        declare

        accountIssuer account\_t;

        accountPayee account\_t;

        refTrans ref transaction\_t;

        begin

        select value (ac) into accountIssuer from

        o\_account ac where ac.accountNo=1;

        select value (ac) into accountPayee from

        o\_account ac where ac.accountNo=2;

        update O\_TRANSACTION ot set ot.refAccIssuer=null, ot.refAccPayee=null

        where ot.refAccIssuer.accountNo=1 and ot.refAccPayee.accountNo=2 and ot.tNum=1 returning ref(ot) into refTrans;

        accountIssuer.deleteLinkListTransaction(refTrans);

        accountPayee.deleteLinkListTransaction(refTrans);

        end;

        /

### 3-1-3 Requêtes impliquant plus de deux tables

Requête 1 :

            delete from o\_transaction ot where ot.amount > 4000 and ot.issuer.numCli=10 and ot.refAccPayee.accountNo=11;

            /

Requête 2 :

            delete from o\_transaction ot

            where ot.amount > 3000 and ot.issuer.job='Ecrivain'

            and ot.issuer.refAgency.loc.country='France';

            /

## 3-3 Requêtes de consultation

### 3-1-1 Requêtes impliquant une table

Requête 1 avec order by:

        select ag.agencyNo, ag.aName

        from o\_agency ag

        order by ag.aName;

        /

Requête 2 avec group by:

        select ot.payee, SUM(ot.amount) from o\_transaction ot

        group by ot.payee;

        /

Requête 3 :

            select \* from o\_account oa where oa.accountType='PEL';

            /

Requête 4 :

            select \* from o\_client oc where oc.job='Médecin';

            /

Requête 5 :

            select \* from o\_employe oe where oe.sal > 2000;

            /

### 3-1-2 Requêtes impliquant deux tables

Requête 1 avec interne :

            select \* from o\_transaction ot left join o\_client oc on ot.issuer.numCli=oc.numCli

            where ot.issuer.numCli=1;

            /

Requête 2 avec group by :

            select ot.issuer.cName, SUM(ot.amount)

            from o\_transaction ot where ot.issuer.refAgency.agencyNo=1

            group by ot.issuer;

            /

Requête 3 avec tri :

            select \* from o\_transaction ot where ot.issuer.refAgency.aName='pine bank' and ot.amount > 3000

            ORDER BY ot.amount;

            /

Requête 4 avec externe :

            select \* from o\_transaction ot left outer join o\_client oc

            on (ot.issuer.numCli=oc.numCli and oc.job='Médecin');

            /

Requête 5 :

            select \* from o\_client oc

            where oc.job='Ingenieur' and oc.refAgency.aName='agence le cèdre';

            /

### 3-1-3 Requêtes impliquant plus de deux tables

Requête 1 avec externe :

            SELECT \*

            FROM o\_transaction ot

                left outer join o\_agency oa

                    on(ot.payee.refAgency.agencyNo = oa.agencyNo and oa.loc.country='Italie');

                    /

Requête 2 avec group by :

            select ot.issuer.numCli, ot.issuer.cName, ot.issuer.prenoms, ot.issuer.job, SUM(ot.amount) from o\_transaction ot

            where ot.amount > 3000 and ot.issuer.birthDate > to\_date('11-12-1960','DD-MM-YYYY')

            and ot.refAccIssuer.accountType='PEL' group by ot.issuer;

            /

Requête 3 avec tri :

            select \* from o\_transaction ot

            where ot.amount > 3000 and ot.issuer.job='Ecrivain'

            and ot.issuer.refAgency.loc.country='France' ORDER BY issuer.cName DESC;

            /

Requête 4 interne :

            SELECT \* FROM o\_transaction  ot LEFT JOIN o\_client oc ON ot.payee.numCli = oc.numCli

            where ot.issuer.refAgency.loc.country='France';

            /

Requête 5 :

            select \* from o\_transaction ot

            where ot.amount > 3000 and ot.issuer.birthDate > to\_date('11-12-1970','DD-MM-YYYY')

            and ot.refAccIssuer.accountType='Compte Epargne';

            /