REPUBLIQUE DU SENEGAL

Un Peuple – Un But – Une Foi

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE

UNIVERSITE GASTON BERGER DE SAINT-LOUIS

U.F.R DE SCIENCES APPLIQUEES ET DE TECHNOLOGIE

MASTER 1 GDIL

Rapport de projet Base de données Relationnelle objet

SUJET : gestion de compte bancaire

Présenté par

DAOUDA FICKOU P31 2277 : fickou.daouda@ugb.edu.sn
MOUHAMADOU GUEYE P31 2127 : gueye.mouhamadou7@ugb.edu.sn
MAME ANTA BATHILY P27 2290 : bathily.mame-anta@ugb.edu.sn

Année académique : 2023/2024

Ce projet consiste à concevoir, implémenter et manipuler une base de données relationnelle-objet pour gérer les clients, leurs comptes (épargne et chèque), ainsi que les mouvements financiers associés. La base de données doit respecter les contraintes d'intégrité référentielle et les spécifications techniques fournies.

Conception de la Base de Données

La base de données est composée des classes et tables suivantes :

- **Client** :
  - Attributs : codeClient (clé primaire), nom, prenom, adresse, numerotelephone (VARRAY de taille 5).
  - Contrainte : Un client peut avoir jusqu'à 5 numéros de téléphone.

- **CompteGenerique** (classe abstraite) :
  - Attributs : CodeCompte (cléprimaire), solde, Dateouverture, referenceClient (clé étrangère vers Client).
  - Contrainte : Un client peut posséder plusieurs comptes (épargne ou chèque).

- **CompteEpargne** (hérite de CompteGenerique) :
  - Attributs : tauxInt (taux d'intérêt).

- **CompteCheque** (hérite de CompteGenerique) :
  - Attributs : nombreOperation, Signatures (nested table contenant codeClient et droitSurCompte)
  - Contrainte : droitSurCompte doit être dans l'ensemble {'R', 'D'}.

- **Mvt** :
  - Attributs : codeClient, CodeCompte, dateOperation, montant.
  - Contrainte : Enregistre les mouvements sur les comptes chèques.

Contraintes d'Intégrité

- Clés primaires : codeClient, CodeCompte.
- Clés étrangères : referenceClient dans CompteGenerique référence Client.
- Contraintes de domaine : droitSurCompte dans {'R', 'D'}.

Création et Peuplement de la base

1. Création de la base

bdro_Bancaire.sql > {} create table comptechequetable > comptecheque

```sql
create or replace type telephone_varray as
   varray(5) of varchar(9);

create or replace type t_signature as object (
      codeclient      number,
      droitsurcompte char(1)
)

create or replace type t_signature_table as
   table of t_signature;

create table client (
   codeclient number primary key,
   nom        varchar2(25) not null,
   prenom     varchar2(25) not null,
   adresse    varchar2(35),
   telephone  telephone_varray
);

create or replace type comptegenerique as object (
      codecompte       varchar2(20),
      solde            number,
      dateouverture    date,
      referenceclient number
) not instantiable not final;

create table comptegeneriquetable of comptegenerique (
   constraint pk_compte primary key ( codecompte ),
   constraint fk_client foreign key ( referenceclient )
      references client ( codeclient )
);


create or replace type compteepargne under comptegenerique (
   tauxint number
);


create table compteepargnetable of compteepargne (
   constraint pk_compte_epargne primary key ( codecompte )
);


create or replace type comptecheque under comptegenerique (
      nombreoperation number,
      signatures       t_signature_table
);


create table comptechequetable of comptecheque (
   constraint pk_compte_cheque primary key ( codecompte )
)
nested table signatures store as tab_signatures;



create table mvt (
   codeclient    number not null,
   codecompte    varchar2(20) not null,
   dateoperation date default sysdate,
   montant       number not null,
   constraint fk_mvt_client foreign key ( codeclient )
      references client ( codeclient ),
   constraint pk_mvt primary key ( codeclient,
                                   codecompte,
                                   dateoperation )
);
```

2. Peuplement de la base

compte.sql > SYSDATE

```sql
-- Comptes Épargne
INSERT INTO CompteEpargnetable (CodeCompte, solde, DateOuverture, referenceClient, tauxInt)
VALUES ('CE101', 500000, SYSDATE, 2, 2.5);

INSERT INTO CompteEpargnetable (CodeCompte, solde, DateOuverture, referenceClient, tauxInt)
VALUES ('CE102', 300000, SYSDATE, 4, 3.0);

INSERT INTO CompteEpargnetable (CodeCompte, solde, DateOuverture, referenceClient, tauxInt)
VALUES ('CE103', 450000, SYSDATE, 3, 2.8);


INSERT INTO CompteEpargnetable (CodeCompte, solde, DateOuverture, referenceClient, tauxInt)
VALUES ('CE108', 500000, SYSDATE, 2, 2.5);

INSERT INTO CompteEpargnetable (CodeCompte, solde, DateOuverture, referenceClient, tauxInt)
VALUES ('CE128', 300000, SYSDATE, 4, 3.0);

INSERT INTO CompteEpargnetable (CodeCompte, solde, DateOuverture, referenceClient, tauxInt)
VALUES ('CE158', 450000, SYSDATE, 3, 2.8);

-- Comptes Chèques (avec `NESTED TABLE` vide pour les signatures)
INSERT INTO CompteChequetable (CodeCompte, solde, DateOuverture, referenceClient, nombreOperation, signatures)
VALUES ('CC201', 200000, SYSDATE, 1, 5, t_signature_table());

INSERT INTO CompteChequetable (CodeCompte, solde, DateOuverture, referenceClient, nombreOperation, signatures)
VALUES ('CC202', 150000, SYSDATE, 2, 3, t_signature_table());

INSERT INTO CompteChequetable (CodeCompte, solde, DateOuverture, referenceClient, nombreOperation, signatures)
VALUES ('CC203', 100000, SYSDATE, 3, 4, t_signature_table());

INSERT INTO CompteChequetable (CodeCompte, solde, DateOuverture, referenceClient, nombreOperation, signatures)
VALUES ('CC204', 50000, SYSDATE, 4, 2, t_signature_table());
```

3. Insertion des signataires pour les comptes courants CompteCourant3 et CompteCourant9

ajoutSignataires.sql > ...

```sql
DECLARE
    signatairesCC203 t_signature_table := t_signature_table(t_signature(1, 'R'), t_signature(2, 'D'));
    signatairesCC204 t_signature_table := t_signature_table(t_signature(3, 'R'), t_signature(4, 'D'));
BEGIN
    UPDATE CompteChequetable SET Signatures = signatairesCC203 WHERE CodeCompte = 'CC203';
    UPDATE CompteChequetable SET Signatures = signatairesCC204 WHERE CodeCompte = 'CC204';
    COMMIT;
END;
/
```

4. Affectation des six comptes épargnes

```sql


UPDATE CompteEpargnetable SET referenceClient = 4 WHERE CodeCompte = 'CE128';
UPDATE CompteEpargnetable SET referenceClient = 4 WHERE CodeCompte = 'CE158';
UPDATE CompteEpargnetable SET referenceClient = 4 WHERE CodeCompte = 'CE101';
UPDATE CompteEpargnetable SET referenceClient = 4 WHERE CodeCompte = 'CE102';
UPDATE CompteEpargnetable SET referenceClient = 4 WHERE CodeCompte = 'CE103';
UPDATE CompteEpargnetable SET referenceClient = 4 WHERE CodeCompte = 'CE108';

COMMIT;

UPDATE CompteEpargnetable SET referenceClient = 4 WHERE CodeCompte = 'CE108';
```

5. Insertion des opérations pour chaque compte

```sql
-- Opérations des comptes épargnes (faites par les titulaires)
INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (2, 'CE101', SYSDATE - 10, 50000); -- Dépôt

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (4, 'CE102', SYSDATE - 8, -20000); -- Retrait

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (3, 'CE103', SYSDATE - 7, 70000); -- Dépôt

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (2, 'CE104', SYSDATE - 5, -15000); -- Retrait

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (4, 'CE108', SYSDATE - 3, 80000); -- Dépôt

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (3, 'CE128', SYSDATE - 2, -30000); -- Retrait

-- Opérations des comptes chèques (faites par le titulaire ou un signataire)
INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (1, 'CC201', SYSDATE - 9, 100000); -- Dépôt du titulaire

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (2, 'CC202', SYSDATE - 6, -50000); -- Retrait du titulaire

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (3, 'CC203', SYSDATE - 4, 25000); -- Dépôt du titulaire

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (4, 'CC204', SYSDATE - 2, -10000); -- Retrait du titulaire

-- Opérations sur les comptes avec signataires
INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (1, 'CC203', SYSDATE - 1, -20000); -- Retrait par le signataire (codeClient 1)

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (2, 'CC203', SYSDATE, 50000); -- Dépôt par le signataire (codeClient 2)

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (3, 'CC204', SYSDATE - 1, -15000); -- Retrait par le signataire (codeClient 3)

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (4, 'CC204', SYSDATE, 60000); -- Dépôt par le signataire (codeClient 4)

COMMIT;
```


```sql
INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (2, 'CC202', SYSDATE - 6, -50000); -- Retrait du titulaire

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (3, 'CC203', SYSDATE - 4, 25000); -- Dépôt du titulaire

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (4, 'CC204', SYSDATE - 2, -10000); -- Retrait du titulaire

-- Opérations sur les comptes avec signataires
INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (1, 'CC203', SYSDATE - 1, -20000); -- Retrait par le signataire (codeClient 1)

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (2, 'CC203', SYSDATE, 50000); -- Dépôt par le signataire (codeClient 2)

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (3, 'CC204', SYSDATE - 1, -15000); -- Retrait par le signataire (codeClient 3)

INSERT INTO mvt (codeClient, CodeCompte, dateOperation, montant)
VALUES (4, 'CC204', SYSDATE, 60000); -- Dépôt par le signataire (codeClient 4)

COMMIT;
```


```sql
INSERT INTO mvt (codeClient, CodeCompte, montant)
VALUES (88, 'CC148', 30000);

COMMIT;
```

Manipulation de la base

1. Ajout du client 88 du compte cheque CC148

```sql
new-client.sql > ...
INSERT INTO Client (codeClient, nom, prenom, adresse, telephone)
VALUES (88, 'Diouf', 'Modou', 'Sor – Saint-Louis', telephone_varray('775514416'));

INSERT INTO CompteChequetable (CodeCompte, solde, DateOuverture, referenceClient, nombreOperation, Signatures)
VALUES ('CC148', 0, SYSDATE, 88, 0, t_Signature_Table());
```

Réponses aux Requêtes

1. Liste des clients sans leur numéro de téléphone

```sql
Run | Tab | JSON | Ask Copilot
SELECT nom, prenom, adresse
FROM Client;
```

Query Result

SQL | All Rows Fetched: 7 in 0.015 seconds

	NOM	PRENOM	ADRESSE
1	Diouf	Modou	Sor – Saint-Louis
2	Ba	Aissatou	Dakar
3	Sow	Moussa	Saint-Louis
4	Diagne	Fatou	Thiès
5	Ndiaye	Cheikh	Kaolack
6	Faye	Awa	Louga
7	Gueye	Ibrahima	Ziguinchor

2. La liste des clients avec la liste de leur numéro de téléphone

```sql
Run | Tab | JSON | Select | Ask Copilot
SELECT codeClient, nom, prenom, adresse,
       LISTAGG(t.COLUMN_VALUE, ', ') WITHIN GROUP (ORDER BY t.COLUMN_VALUE) AS numeros
FROM Client Cl, TABLE(Cl.telephone) t
GROUP BY codeClient, nom, prenom, adresse;
```

Script Output | Query Result

SQL | All Rows Fetched: 6 in 0.035 seconds

	CODECLIENT	NOM	PRENOM	ADRESSE	NUMEROS
1	1	Ba	Aissatou	Dakar	771234567, 778889900
2	2	Sow	Moussa	Saint-Louis	765432100, 767891234
3	3	Diagne	Fatou	Thiès	770098765, 775678912
4	4	Ndiaye	Cheikh	Kaolack	771112233, 776543210
5	5	Faye	Awa	Louga	773344556
6	88	Diouf	Modou	Sor – Saint-Louis	775514416

3. La liste des comptes cheque

```
Run | Tab | JSON | Ask Copilot
SELECT CodeCompte, solde, DateOuverture, nombreOperation
FROM CompteChequetable
ORDER BY CodeCompte;
```

Script Output × Query Result ×

SQL | All Rows Fetched: 5 in 0.021 seconds

	CODECOMPTE	SOLDE	DATEOUVERTURE	NOMBREOPERATION
1	CC148	0	04-FEB-25	0
2	CC201	200000	04-FEB-25	5

4. La liste des comptes épargne

```
Run | Tab | JSON | Ask Copilot
SELECT codecompte,solde,DateOuverture,tauxInt
FROM CompteEpargnetable ;
```

Script Output × Query Result ×

SQL | All Rows Fetched: 6 in 0.009 seconds

	CODECOMPTE	SOLDE	DATEOUVERTURE	TAUXINT
1	CE128	300000	04-FEB-25	3
2	CE158	450000	04-FEB-25	2.8
3	CE101	500000	04-FEB-25	2.5
4	CE102	300000	04-FEB-25	3
5	CE103	450000	04-FEB-25	2.8
6	CE108	500000	04-FEB-25	2.5

5. La liste des comptes cheque du client de numéro 4

```
Run | Tab | JSON | Select | Ask Copilot
SELECT Cc.CodeCompte, Cc.solde, Cc.DateOuverture, Cc.nombreOperation
FROM CompteChequetable Cc, Client Cl
WHERE Cc.referenceClient = Cl.codeClient
AND Cl.codeClient = 4;
```

Script Output × Query Result ×

SQL | All Rows Fetched: 1 in 0.026 seconds

	CODECOMPTE	SOLDE	DATEOUVERTURE	NOMBREOPERATION
1	CC204	50000	04-FEB-25	2

6. Même requête pour tous ses comptes (courant et épargne).

```
Run | Tab | JSON | Select | Ask Copilot
SELECT Ce.CodeCompte, Ce.solde, Ce.DateOuverture
FROM CompteEpargnetable Ce, Client Cl
WHERE Ce.referenceClient = Cl.codeClient
AND Cl.codeClient = 4
UNION
SELECT Cc.CodeCompte, Cc.solde, Cc.DateOuverture
FROM CompteChequetable Cc, Client Cl
WHERE Cc.referenceClient = Cl.codeClient
AND Cl.codeClient = 4;
```

Query R... ×

SQL | All Rows Fetched: 7 in 0.045 seconds

	CODECOMPTE	SOLDE	DATEOUVERTURE
1	CE128	300000	04-FEB-25
2	CE158	450000	04-FEB-25
3	CE101	500000	04-FEB-25
4	CE102	300000	04-FEB-25
5	CE103	450000	04-FEB-25
6	CE108	500000	04-FEB-25
7	CC204	50000	04-FEB-25

7. Liste des clients et numéro des comptes courants de solde inférieur à 0 FCFA donné (client à découvert)

```
Run | Tab | JSON | Select | Ask Copilot
SELECT Cl.codeClient, Cl.nom, Cl.prenom, Cc.CodeCompte, Cc.solde
FROM Client Cl, CompteChequetable Cc
WHERE Cl.codeClient = Cc.referenceClient
AND Cc.solde < 0;
```

Query Result ×

SQL | All Rows Fetched: 0 in 0.029 seconds

CODECLI...	NOM	PRENOM	CODECO...	SOLDE

8. Numéro, nom et adresse des clients titulaires d'au moins un compte épargne

```
Run | Tab | JSON | Ask Copilot
SELECT DISTINCT Cl.codeClient, Cl.nom, Cl.prenom, Cl.adresse
FROM Client Cl, CompteEpargnetable Ce
WHERE Cl.codeClient = Ce.referenceClient;
```

Query Result

SQL | All Rows Fetched: 1 in 0.034 seconds

	CODECLIENT	NOM	PRENOM	ADRESSE
1	4	Ndiaye	Cheikh	Kaolack

9. Numéro, nom et adresse des clients titulaires d'un seul compte d'épargne

```
Run | Tab | JSON | Select | Ask Copilot
SELECT Cl.codeClient, Cl.nom, Cl.prenom, Cl.adresse
FROM Client Cl, CompteEpargnetable Ce
WHERE Cl.codeClient = Ce.referenceClient
GROUP BY Cl.codeClient, Cl.nom, Cl.prenom, Cl.adresse
HAVING COUNT(Ce.CodeCompte) = 1;
```

Query Result

SQL | All Rows Fetched: 0 in 0.014 seconds

CODECLI...	NOM	PRENOM	ADRESSE

10. Pour chaque client (numéro, nom), afficher le nombre de compte épargne qu'il possède

```
Run | Tab | JSON | Select | Ask Copilot
SELECT Cl.codeClient, Cl.nom, Cl.prenom, COUNT(Ce.CodeCompte) as nbreCompte
FROM Client Cl,CompteEpargnetable Ce
WHERE  Cl.codeClient = Ce.referenceClient
GROUP BY Cl.codeClient, Cl.nom, Cl.prenom;
```

Query Result

SQL | All Rows Fetched: 1 in 0.036 seconds

	CODECLIENT	NOM	PRENOM	COUNT(CE.CODECOMPTE)
1	4	Ndiaye	Cheikh	6

11. Même requête pour afficher aussi les clients n'ayant pas de compte épargne

```
Run | Tab | JSON | Select | Ask Copilot
SELECT Cl.codeClient, Cl.nom, Cl.prenom, COALESCE(COUNT(Ce.CodeCompte), 0) AS nbComptesEpargne
FROM Client Cl
LEFT JOIN CompteEpargnetable Ce  ON Cl.codeClient = Ce.referenceClient
GROUP BY Cl.codeClient, Cl.nom, Cl.prenom;
```

Query Result

SQL | All Rows Fetched: 7 in 0.315 seconds

	CODECLIENT	NOM	PRENOM	NBCOMPTESEPARGNE
1	4	Ndiaye	Cheikh	6
2	6	Gueye	Ibrahima	0
3	88	Diouf	Modou	0
4	1	Ba	Aissatou	0
5	2	Sow	Moussa	0
6	5	Faye	Awa	0
7	3	Diagne	Fatou	0

12. Numéro, nom et nombre de compte épargne du titulaire ayant le plus de comptes épargne

```
Run | Tab | JSON | Select | Ask Copilot
SELECT Cl.codeClient, Cl.nom, COUNT(Ce.CodeCompte) AS nbComptesEpargne
FROM Client Cl
JOIN CompteEpargnetable Ce ON Cl.codeClient = Ce.referenceClient
GROUP BY Cl.codeClient, Cl.nom
ORDER BY nbComptesEpargne DESC
FETCH FIRST 1 ROWS ONLY;
```

Query Result

All Rows Fetched: 1 in 0.145 seconds

	CODECLIENT	NOM	NBCOMPTESEPARGNE
1	4	Ndiaye	6

13. Liste des clients (numéro et nom) qui ont fait des opérations (date et montant) sur le compte courant de numéro 10

```
SELECT DISTINCT Cl.codeClient, Cl.nom, Cl.prenom, m.dateOperation, m.montant
FROM Client Cl, mvt m
WHERE Cl.codeClient = m.codeClient
AND m.CodeCompte = 'CC204';
```

Query Result

All Rows Fetched: 3 in 0.023 seconds

	CODECLIENT	NOM	PRENOM	DATEOPERATION	MONTANT
1	3	Diagne	Fatou	03-FEB-25	-15000
2	4	Ndiaye	Cheikh	02-FEB-25	-10000
3	4	Ndiaye	Cheikh	04-FEB-25	60000

14. Liste des opérations sur le compte courant 8 (numéro de compte, date et montant) qui ont été faites par un client propriétaire du compte modifié

```
SELECT m.CodeCompte, m.dateOperation, m.montant
FROM mvt m
JOIN CompteChequetable cc ON m.CodeCompte = cc.CodeCompte
WHERE m.CodeCompte = 'CC203'
AND m.codeClient = cc.referenceClient;
```

Query Result

All Rows Fetched: 1 in 0.011 seconds

	CODECOMPTE	DATEOPERATION	MONTANT
1	CC203	31-JAN-25	25000

15. Même requête pour les opérations qui ont été faites par les clients non propriétaire du compte modifié

```
SELECT m.CodeCompte, m.dateOperation, m.montant
FROM mvt m
JOIN CompteChequetable cc ON m.CodeCompte = cc.CodeCompte
WHERE m.CodeCompte = 'CC203'
AND m.codeClient <> cc.referenceClient;
```

Query Result

All Rows Fetched: 2 in 0.027 seconds

	CODECOMPTE	DATEOPERATION	MONTANT
1	CC203	03-FEB-25	-20000
2	CC203	04-FEB-25	50000

Conclusion

Ce projet nous a permis de mettre en pratique les concepts de bases de données relationnelles-objets, notamment l'utilisation de VARRAY, nested tables, héritage et contraintes d'intégrité