

Rapport de Projet : Gestion des Prêts Bancaires

PROJET GROUPE 1

NDIASSE FALL
SERIGNE TALLA
FATOU KINE MBAYE
MAMADOU MAHFOUZ BALDE
FALLOU THIAW
FATOUMATA KANOUTE

Boubacar faty | Modélisation Base de donnée | 31/01/2025

Introduction

Contexte du projet : La gestion des prêts bancaires est un processus complexe qui nécessite une grande précision et une organisation rigoureuse. Une base de données efficace est essentielle pour assurer le suivi des prêts, des remboursements, et pour gérer les risques associés.

Objectifs du projet : Ce projet vise à développer une base de données robuste et conviviale pour gérer tous les aspects des prêts bancaires, de la demande initiale au remboursement final. Les objectifs spécifiques incluent :

- Digitalisation des processus de gestion des prêts.
- Amélioration de l'efficacité et de la précision du suivi des prêts et des remboursements.
- Facilitation de la gestion des prêts.

Portée du projet : Le projet couvrira les fonctionnalités suivantes :

- Gestion des clients (enregistrement).
- Gestion des demandes de prêts (saisie, validation, refus).
- Gestion des prêts (création, suivi).
- Gestion des remboursements (enregistrement, calcul des soldes).
- Gestion des taxes (calcul, application).

ANALYSE DES BESOINS

Identification des acteurs :

- Clients
- Chargés de prêts

Recueil des besoins : Les besoins ont été recueillis par le biais du cahier de charge, de la recherche sur le Net et d'ateliers de travail. Les principaux besoins identifiés sont :

- Suivi précis des informations des clients et des prêts.
- Automatisation des calculs (montant des mensualités, soldes, taxes).
- Gestion des risques (identification des prêts à risque, suivi des retards de paiement).
- Facilité d'utilisation et interface conviviale.
- Sécurité des données.

Modélisation des processus : (Diagramme Merise inclure en annexe)

CONCEPTION DE LA BASE DE DONNEES

Modèle conceptuel (MCD) : (Schéma MCD inclure en annexe)

- Entités : Client, DemandePrêt, Prêt, Remboursement, Taxe
- Relations :
 - Un client peut faire plusieurs demandes de prêt.
 - Un prêt est lié à une demande de prêt.
 - Un prêt peut avoir plusieurs remboursements.
 - Un prêt peut être soumis à plusieurs taxes.

Modèle logique (Schéma SQL) : (Schéma SQL détaillé inclure en annexe)

- Tables :
 - Clients (NumClient _, nom, adresse, etc.)
 - DemandePrêt (Num_ordre, montant, durée, NumClient, etc.)
 - Prêt (Num_Prets, montant, taux, durée, mensualité, Num_ordre, etc.)
 - Remboursement (Num_Rem, montant, date, etc.)
 - Taxe (Num_taxe, montant, date, etc.)
- Clés primaires et étrangères : (Définies dans le schéma SQL)
- Contraintes d'intégrité : (Définies dans le schéma SQL)

DEVELOPPEMENT DE LA BASE DE DONNEES

Environnement de développement :

- SGBD : MySQL
- Langage de programmation : SQL, PHP, HTML, JAVASCRIPT
- Framework : Bootstrap
- Outil de modélisation : Jmerise, AnalyseSI

Création des tables : (Scripts SQL pour la création des tables en annexe)

Insertion des données : (Exemples de données insérées pour les tests en annexe)

FONCTIONNALITES DEVELOPPEES

Gestion des clients :

- Enregistrement : Formulaire pour saisir les informations du client.

Gestion des demandes de prêts :

- Saisie : Formulaire pour saisir les détails de la demande (montant, durée, client).
- Validation : Vérification des critères d'éligibilité (pas d'autres prêts en cours, pas de suspension).
- Refus : Enregistrement du motif du refus.
- Création : Création du prêt après approbation de la demande.

Gestion des prêts :

- Suivi : Suivi de l'état du prêt (actif, révisé).

Gestion des remboursements :

- Enregistrement : Enregistrement des remboursements avec date et montant.
- Calcul des soldes : Mise à jour automatique du solde du prêt.

Gestion des taxes :

- Calcul : Calcul automatique des taxes en cas de retard de paiement.
- Application : Enregistrement de l'application de la taxe.

TESTS ET VALIDATION

Plan de test : (Cas de test détaillés en annexe)

- Tests unitaires : Vérification du bon fonctionnement de chaque fonctionnalité.
- Tests d'intégration : Vérification de l'interaction entre les différentes fonctionnalités.
- Exécution des tests : (Non complète)

CONCLUSION

Synthèse des résultats : Le projet a permis de développer une base de données fonctionnelle et efficace pour la gestion des prêts bancaires.

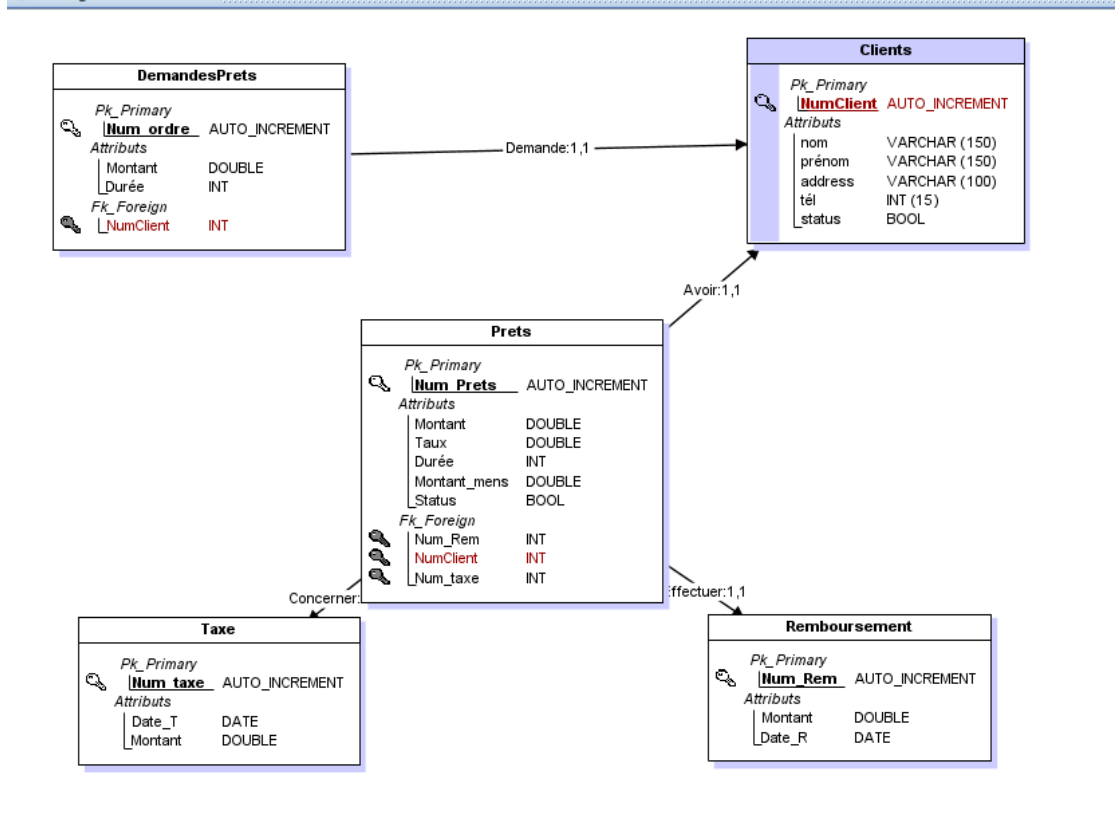
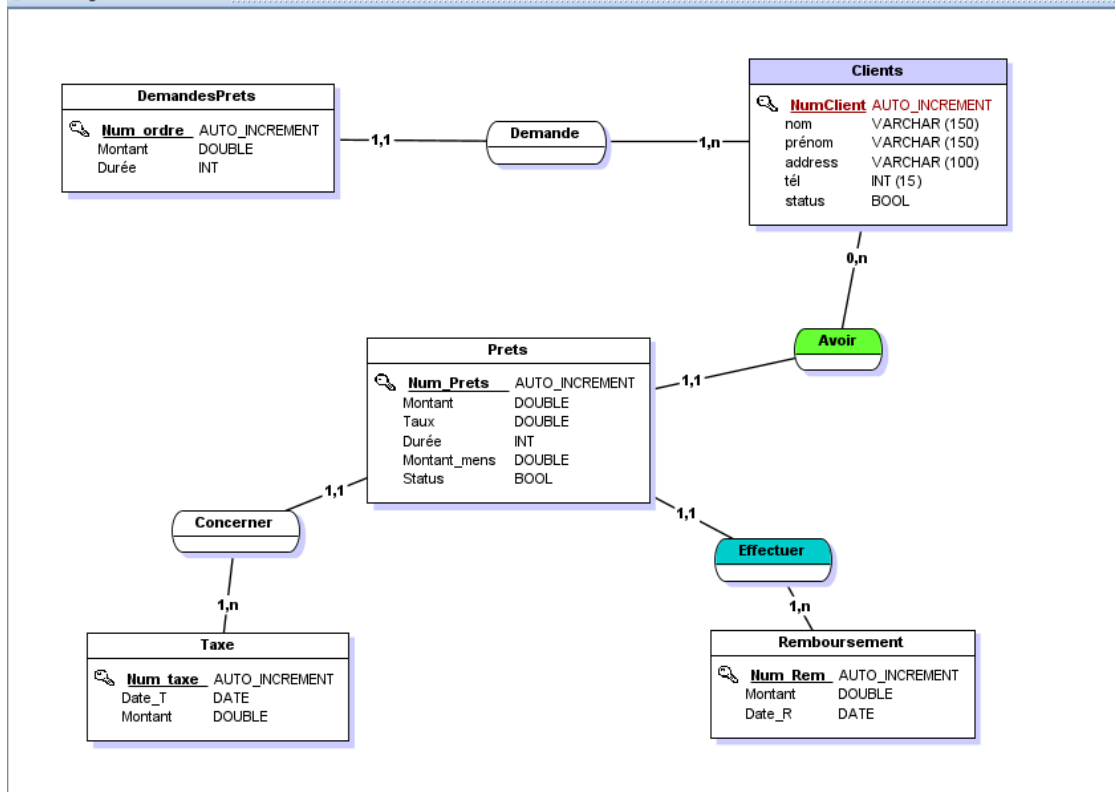
Perspectives d'évolution :

- Amélioration de l'interface utilisateur.

Retour d'expérience : Le projet a été l'occasion d'acquérir une expérience précieuse dans la conception et le développement de bases de données pour le secteur bancaire.

Annexes

- Diagrammes (MCD, MRD, MLD)



Clients (NumClient, Nom, Prénom, Adresse, Status) ; DemandesPrets (Num_ordre, Montant, Durée, NumClient) ; Remboursement (Num_Rem, Montant, Date_R,) ; Taxe (Num_taxe, Date_T, Montant,) ; Prets (Num_Prets, Montant, Taux, Durée, Montant_mens, Status, NumClient, Num_Rem, Num_taxe) ;

- Scripts SQL pour la création des tables

```
DROP DATABASE IF EXISTS gestion_pretsbank;
CREATE DATABASE IF NOT EXISTS gestion_pretsbank;
USE gestion_pretsbank;

CREATE TABLE IF NOT EXISTS Clients(
    NumClient    Int    Auto_increment NOT NULL ,
    nom          Varchar (150) ,
    prenom       Varchar (150) ,
    address      Varchar (100) ,
    tel          Varchar (15) ,
    status       ENUM ('actif','suspendu')

    ,CONSTRAINT Clients_PK PRIMARY KEY (NumClient)

);

CREATE TABLE IF NOT EXISTS demandesPrets(
    Num_ordre    Int    Auto_increment NOT NULL ,
    Montant      Double ,
    Duree        Int,
    NumClient    Int
    ,CONSTRAINT DemandesPrets_PK PRIMARY KEY (Num_ordre)
    ,CONSTRAINT DemandesPrets_Clients_FK FOREIGN KEY (NumClient)
REFERENCES Clients(NumClient)
);

CREATE TABLE IF NOT EXISTS Remboursement(
    Num_Rem      Int    Auto_increment NOT NULL ,
    Montant      Double ,
    Date_R       Date
    ,CONSTRAINT Remboursement_PK PRIMARY KEY (Num_Rem)
```

```

);

CREATE TABLE IF NOT EXISTS Taxe(
    Num_taxe    Int    Auto_increment NOT NULL ,
    Date_T      Date ,
    Montant     Double
    ,CONSTRAINT Taxe_PK PRIMARY KEY (Num_taxe)
);

CREATE TABLE IF NOT EXISTS Prets(
    Num_Prets    Int    Auto_increment NOT NULL ,
    Montant      Double ,
    Taux         Double ,
    Duree        Int ,
    Montant_mens Double ,
    Status       ENUM ('actif','révisé') ,
    Num_Rem      Int NOT NULL ,
    NumClient    Int NOT NULL ,
    Num_taxe     Int NOT NULL
    ,CONSTRAINT Prets_PK PRIMARY KEY (Num_Prets)

    ,CONSTRAINT Prets_Remboursement_FK FOREIGN KEY (Num_Rem)
REFERENCES Remboursement(Num_Rem)
    ,CONSTRAINT Prets_Clients0_FK FOREIGN KEY (NumClient)
REFERENCES Clients(NumClient)
    ,CONSTRAINT Prets_Taxe1_FK FOREIGN KEY (Num_taxe) REFERENCES
Taxe(Num_taxe)
);

```

- Exemples de données

```

INSERT INTO clients (NumClient, Nom, Prenom, address, tel, status)
VALUES
(5, 'Fall', 'Fatou', 'Cite Keur Massar', 765432109, 'actif'),
(6, 'Diouf', 'Moussa', 'Sicap Dakar', 789654321, 'suspendu'),
(7, 'Sène', 'Aminata', 'Grand Yoff', 701234567, 'actif'),
(8, 'Ba', 'Ousmane', 'Guédiawaye', 777888999, 'actif'),
(9, 'Sow', 'Aïssatou', 'Hann Bel Air', 798765432, 'actif'),
(10, 'Seck', 'Cheikh', 'Parcelles Assainies', 712345678, 'actif'),
(11, 'Diagne', 'Ndèye', 'Pikine', 789012345, 'suspendu'),

```

```

(12, 'Gueye', 'Ibrahima', "Médina", 765432109, 'actif'),
(13, 'Niang', 'Coumba', "Fann", 789654321, 'actif'),
(14, 'Cissé', "Moustapha", "Mermoz", 701234567, 'actif');

INSERT INTO demandesprets (Num_ordre, Montant, Duree, NumClient) VALUES
(1005, 8000.00, 24, 5),
(1006, 15000.00, 36, 6),
(1007, 10000.00, 18, 7),
(1008, 20000.00, 48, 8),
(1009, 12000.00, 24, 9),
(1010, 18000.00, 36, 10),
(1011, 9000.00, 18, 11),
(1012, 25000.00, 48, 12),
(1013, 14000.00, 24, 13),
(1014, 22000.00, 36, 14);

INSERT INTO Remboursement (Num_Rem, Montant, Date_R) VALUES
(3005, 280.00, '2025-02-05'),
(3006, 420.00, '2025-02-10'),
(3007, 550.00, '2025-02-15'),
(3008, 300.00, '2025-03-05'),
(3009, 380.00, '2025-03-10'),
(3010, 500.00, '2025-03-15'),
(3011, 250.00, '2025-04-05'),
(3012, 400.00, '2025-04-10'),
(3013, 520.00, '2025-04-15'),
(3014, 280.00, '2025-05-05');

INSERT INTO Taxe (Num_taxe, Date_T, Montant) VALUES
(4005, '2025-02-05', 6.00),
(4006, '2025-02-10', 9.00),
(4007, '2025-02-15', 12.00),
(4008, '2025-03-05', 5.00),
(4009, '2025-03-10', 8.00),
(4010, '2025-03-15', 11.00),
(4011, '2025-04-05', 4.00),
(4012, '2025-04-10', 7.00),
(4013, '2025-04-15', 10.00),
(4014, '2025-05-05', 6.00);

```



```

INSERT INTO Prets (Num_Prets, Montant, Taux, Duree, Montant_mens,
Num_Rem, NumClient, Num_taxe, status) VALUES
(2004, 8000.00, 5.5, 24, 280.00, 3005, 5, 4005, 'actif'),
(2005, 15000.00, 4.0, 36, 420.00, 3006, 6, 4006, 'révisé'),
(2006, 10000.00, 6.5, 18, 550.00, 3007, 7, 4007, 'actif'),
(2007, 20000.00, 5.0, 48, 300.00, 3008, 8, 4008, 'actif'),
(2008, 12000.00, 4.5, 24, 380.00, 3009, 9, 4009, 'révisé'),
(2009, 18000.00, 6.0, 36, 500.00, 3010, 10, 4010, 'actif'),
(2010, 9000.00, 5.5, 18, 250.00, 3011, 11, 4011, 'actif'),
(2011, 25000.00, 4.0, 48, 400.00, 3012, 12, 4012, 'révisé'),
(2012, 14000.00, 6.5, 24, 520.00, 3013, 13, 4013, 'actif'),
(2013, 22000.00, 5.0, 36, 280.00, 3014, 14, 4014, 'actif');

```

- Dépôt Github : https://github.com/NdiasseFall/Projet_MBD_Groupe1.git