

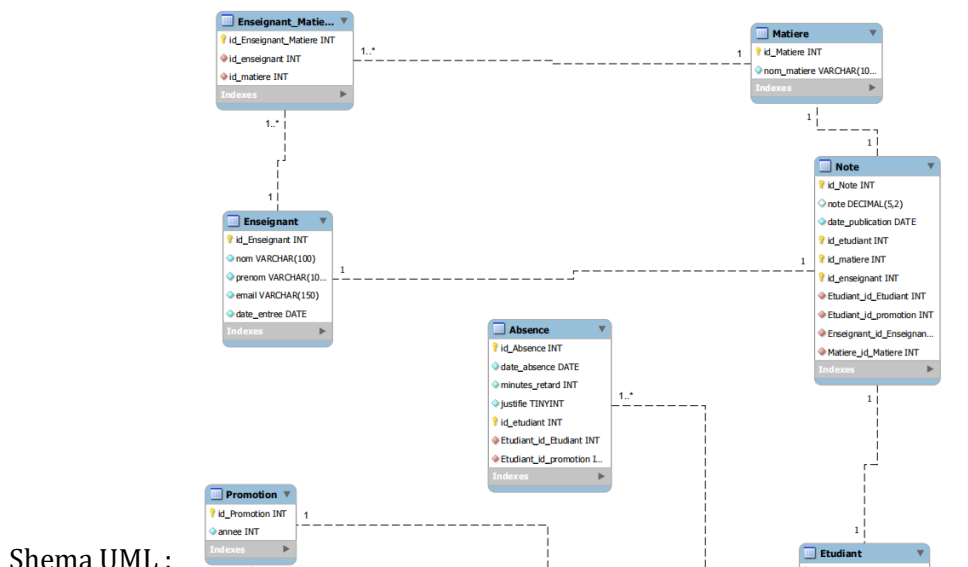
Rapport du Projet : Système de Gestion Informatique pour une École

Introduction

Ce rapport présente le processus de réalisation du projet de système de gestion informatique pour une école. L'objectif principal était de concevoir une base de données relationnelle pour gérer efficacement les données des étudiants, des enseignants, des matières, des promotions, des notes et des absences. Le projet inclut également des analyses à travers des requêtes SQL.

Étape 1 : Analyse des besoins et conception UML

Dans cette étape, nous avons analysé les besoins fournis par le client pour structurer les données à travers un schéma UML (ERD). Les entités principales identifiées sont : Enseignant, Enseignant matière, Étudiant, Matière, Promotion, Note, Absence et Historique Promotion.



Étape 2 : Normalisation de la base de données en 3FN

Après avoir défini les tables, nous avons appliqué la normalisation jusqu'à la troisième forme normale (3FN) pour éliminer les redondances et assurer l'intégrité des données. Voici un résumé des étapes de normalisation :

1. Vérification de la 1NF : Les colonnes contiennent des valeurs atomiques et chaque table possède une clé primaire.
2. Vérification de la 2NF : Toutes les colonnes non-clés dépendent de la totalité de la clé primaire.
3. Vérification de la 3NF : Aucune dépendance transitive entre colonnes non-clés.

Étape 3 : Création des tables SQL

Un script SQL a été écrit pour créer les tables normalisées, définir les relations entre elles, et ajouter les contraintes nécessaires (clés primaires, étrangères, unique, etc.). Le script est fourni dans un fichier séparé.

Exquise création des tables SQL :

```
1 • USE gestion_ecole;
2
3 • CREATE TABLE Enseignant (
4     id_enseignant INT AUTO_INCREMENT PRIMARY KEY,
5     nom VARCHAR(100) NOT NULL,
6     prenom VARCHAR(100) NOT NULL,
7     email VARCHAR(150) UNIQUE NOT NULL,
8     date_entree DATE NOT NULL
9 );
10 • CREATE TABLE Matiere (
11     id_matiere INT AUTO_INCREMENT PRIMARY KEY,
12     nom_matiere VARCHAR(100) NOT NULL
13 );
14
15 • CREATE TABLE Enseignant_Matiere (
16     id_enseignant INT,
17     id_matiere INT,
18     PRIMARY KEY (id_enseignant, id_matiere),
19     FOREIGN KEY (id_enseignant) REFERENCES Enseignant(id_enseignant),
```

Étape 4 : Requêtes SQL pour l'analyse

Des requêtes SQL ont été écrites pour répondre aux besoins analytiques du client :

1. Liste des étudiants avec nom et prénom renommés en last_name et first_name.

```
2 • SELECT nom AS last_name, prenom AS first_name
3     FROM Etudiant;
4
```

2. Liste des retards non justifiés de plus de 10 minutes, triée par durée décroissante.

```
1
2 • SELECT date_absence, minutes_retard, justifie, id_etudiant
3     FROM Absence
4     WHERE justifie = FALSE AND minutes_retard > 10
5     ORDER BY minutes_retard DESC;
6
```

3. Notes entre 15 et 20 publiées cette année avec informations sur l'étudiant et l'enseignant.

```

2 • SELECT
3     Note.note,
4     Note.date_publication,
5     Etudiant.nom AS etudiant_nom,
6     Etudiant.prenom AS etudiant_prenom,
7     Enseignant.email AS enseignant_email
8 FROM
9     Note
10 JOIN
11     Etudiant ON Note.id_etudiant = Etudiant.id_etudiant
12 JOIN
13     Enseignant ON Note.id_enseignant = Enseignant.id_enseignant
14 WHERE
15     Note.note BETWEEN 15 AND 20
16     AND YEAR(Note.date_publication) = YEAR(CURDATE());
17

```

Étape 5 : Insertion de données (Bonus)

Un script SQL d'insertion a été écrit pour ajouter des données de test dans les tables de la base. Cela permet de vérifier le bon fonctionnement des requêtes analytiques et de tester l'intégrité des données.

```

2
3 • INSERT INTO Enseignant (nom, prenom, email, date_entree)
4 VALUES
5     ('Nom_Enseignant_1', 'Prenom_Enseignant_1', 'enseignant1@exemple.com', '2020-09-01'),
6     ('Nom_Enseignant_2', 'Prenom_Enseignant_2', 'enseignant2@exemple.com', '2021-01-15');
7
8
9 • INSERT INTO Matiere (nom_matiere)
10 VALUES
11     ('Matière 1'),
12     ('Matière 2');
13
14
15 • INSERT INTO Enseignant_Matiere (id_enseignant, id_matiere)
16 VALUES
17     (1, 1),
18     (2, 2);
19

```

Conclusion

Ce projet a permis de concevoir et de tester une base de données relationnelle répondant aux besoins du client. La normalisation en 3FN garantit la cohérence et l'intégrité des données, tandis que les requêtes SQL offrent des outils d'analyse puissants.