



DigiSchool : Gestion des Données Scolaires avec MongoDB

Auteurs : Filip DABROWSKI, Marwa BENYAHIA





Agenda de la Présentation

1. Objectif du Projet
2. Organisation du Travail
3. Structure du Projet
4. Modèle de Données NoSQL
5. Avantages et Limites du Modèle
6. Création de l'Application
7. Fonctionnalités Clés
8. Lancer et Utiliser l'Application
9. Modèle de la Base NoSQL
10. Conclusion
11. Auteurs





Objectif du Projet

Le projet DigiSchool vise à développer une application qui permet une gestion efficace des données scolaires. Le but est de créer une API accessible aux parents d'élèves, qui peuvent consulter des informations concernant leurs enfants, y compris les notes, les professeurs, et les classes.

Notre application utilise des technologies comme Python, FastAPI, et MongoDB pour atteindre une grande flexibilité et simplicité d'utilisation.



Organisation du Travail

Le travail a été réalisé en binôme, par Filip DABROWSKI et Marwa BENYAHIA. Le projet a été livré en utilisant Git, conformément aux exigences du cahier des charges. L'accent a été mis sur la collaboration et la gestion du code source pour assurer une qualité constante.





Structure du Projet

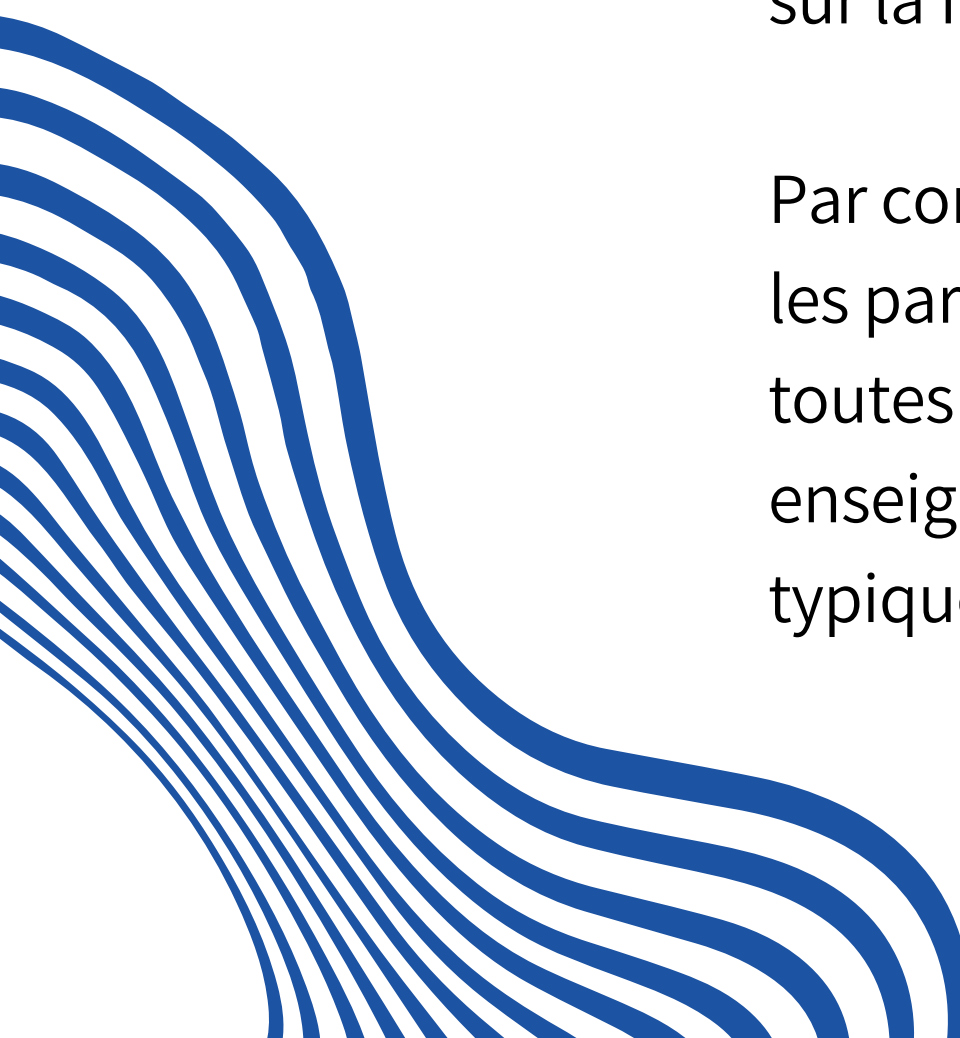
Le projet est structuré de manière à faciliter la maintenance et la compréhension :

- Code source dans le dossier **src** pour les modèles, contrôleurs, et routes.
- Configurations dans **.env** pour gérer les variables d'environnement.
- Documentations dans le dossier **docs** pour guider les développeurs.



Modèle de Données NoSQL

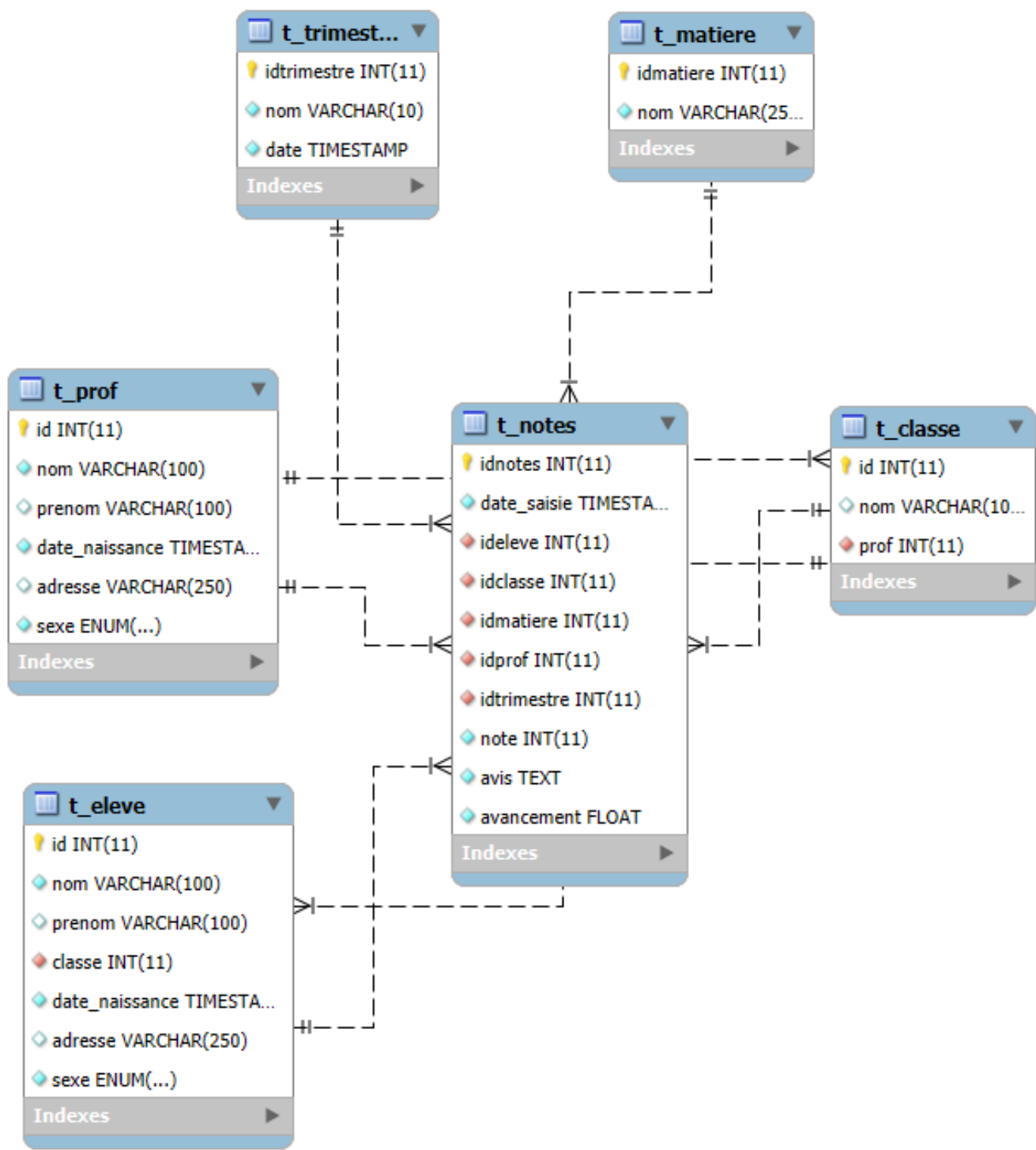
Nous avons pris la décision de convertir toutes les tables SQL en documents imbriqués dans MongoDB en partant de **l'hypothèse que l'application sera principalement utilisée par les parents d'élèves**. L'application a été conçue pour répondre spécifiquement aux besoins des parents, en mettant l'accent sur la facilité d'accès aux informations concernant leurs enfants.



Par conséquent, la majorité des requêtes est orientée autour des étudiants, car les parents sont les principaux utilisateurs. Ce modèle permet de regrouper toutes les informations pertinentes sur un étudiant (notes, classes, enseignants) dans un seul document, simplifiant ainsi les accès fréquents et typiques pour les parents.

Migration des Données de SQL à NoSQL

SQL

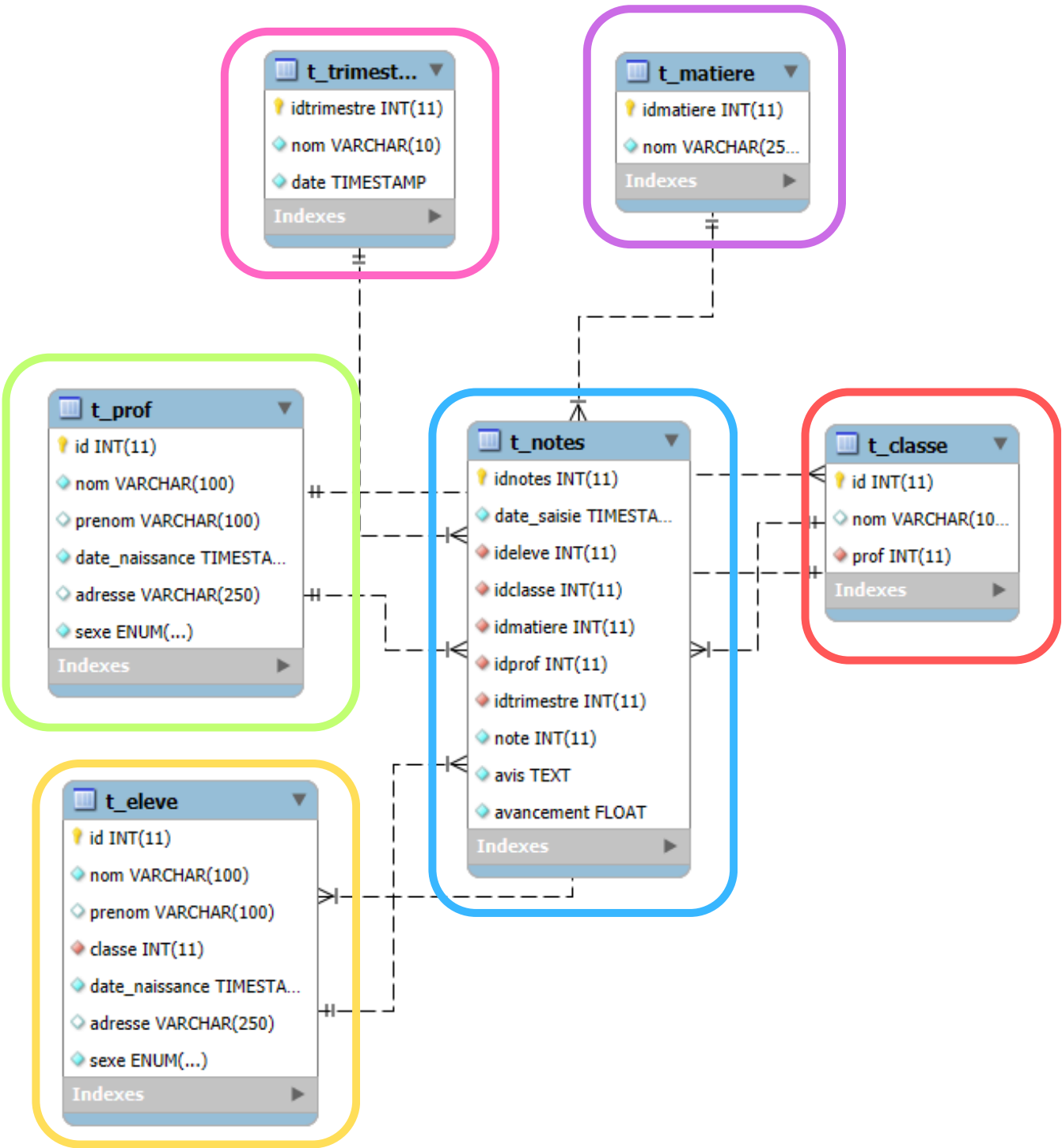


NoSQL

```
{
  "_id": "<student_id>",
  "nom": "<student_name>",
  "prenom": "<student_firstname>",
  "date_naissance": "<date_of_birth>",
  "adresse": "<address>",
  "sexe": "<gender>",
  "classe": {
    "class_id": "<class_id>",
    "nom": "<class_name>",
    "prof": {
      "prof_id": "<teacher_id>",
      "nom": "<teacher_lastname>",
      "prenom": "<teacher_firstname>"
    }
  },
  "notes": [
    {
      "note_id": "<note_id>",
      "matiere": {
        "matiere_id": "<subject_id>",
        "nom": "<subject_name>"
      },
      "note": "<grade_value>",
      "date_saisie": "<date_entered>",
      "trimestre": {
        "trimestre_id": "<trimester_id>",
        "nom": "<trimester_name>"
      },
      "avis": "<comment>",
      "avancement": "<progress>"
    }
  ]
}
```


Migration des Données de SQL à NoSQL

SQL



NoSQL

```
{
  "_id": "<student_id>",
  "nom": "<student_name>",
  "prenom": "<student_firstname>",
  "date_naissance": "<date_of_birth>",
  "adresse": "<address>",
  "sexe": "<gender>",
  "classe": {
    "class_id": "<class_id>",
    "nom": "<class_name>",
    "prof": {
      "prof_id": "<teacher_id>",
      "nom": "<teacher_lastname>",
      "prenom": "<teacher_firstname>"
    }
  },
  "notes": [
    {
      "note_id": "<note_id>",
      "matiere": {
        "matiere_id": "<subject_id>",
        "nom": "<subject_name>"
      },
      "note": "<grade_value>",
      "date_saisie": "<date_entered>",
      "trimestre": {
        "trimestre_id": "<trimester_id>",
        "nom": "<trimester_name>"
      },
      "avis": "<comment>",
      "avancement": "<progress>"
    }
  ]
}
```


Avantages et Limites du Modèle

Le modèle imbriqué de MongoDB est bien adapté pour une application orientée vers les parents, car il permet d'accéder à toutes les données relatives aux étudiants en une seule requête.

Cependant, ce modèle est moins performant pour des requêtes impliquant des classes fortement imbriquées, comme celles liées aux enseignants ou des analyses à l'échelle de l'école. Si l'objectif avait été de gérer les enseignants de manière plus centrale, une approche plus normalisée aurait été nécessaire.





Création de l'Application

Pour développer l'application, nous avons utilisé **FastAPI**, qui nous permet de construire une API RESTful performante et facilement testable via Swagger.

La migration des données a été effectuée à l'aide de **SQLAlchemy** pour importer les informations de la base **SQL vers MongoDB**. Les principales fonctionnalités incluent des opérations CRUD sur les entités telles que les étudiants, les professeurs, et les classes.

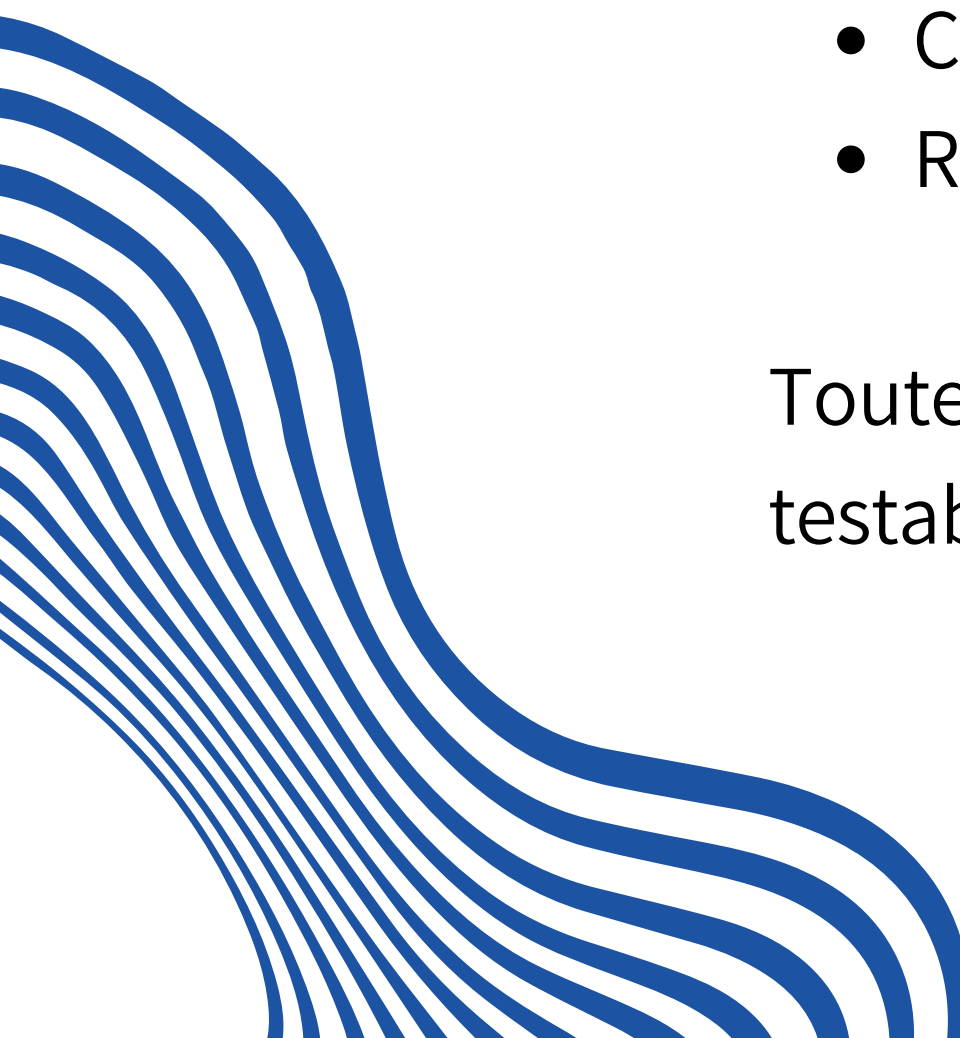


Fonctionnalités Clés

L'application permet aux utilisateurs de :

- Consulter la liste des professeurs.
- Obtenir la liste des élèves par classe.
- Obtenir les élèves en fonction d'une classe sélectionnée.
- Consulter les notes d'un élève.
- Récupérer les élèves et leurs notes par professeur.


Toutes ces fonctionnalités sont accessibles via des endpoints FastAPI et testables sur /docs grâce à Swagger.





Lancer et Utiliser l'Application

Pour lancer l'application :

1. Configurer les variables d'environnement en utilisant le fichier `.env`.
 2. Démarrer le serveur avec `run.py`.
 3. Tester l'API en accédant à `/docs` pour une documentation interactive.
- 

Conclusion et Points Importants

Le projet DigiSchool démontre comment utiliser MongoDB pour une gestion flexible des données scolaires. La structure imbriquée des documents est adaptée aux besoins des parents mais présente des limites en termes de requêtes complexes pour des utilisateurs comme les enseignants.

Le projet est extensible et peut être ajusté en fonction des besoins futurs. L'utilisation des technologies modernes comme FastAPI, PyMongo, et SQLAlchemy nous a permis de créer une application performante et robuste.



Auteurs

Filip DABROWSKI et Marwa BENYAHIA



