

Rapport de Projet NoSQL

4^{ème} année

Ingénierie Informatique et Réseaux

Sous le thème

GESTION DES CIRCUITS ET COURSES AUTOMOBILES

Réalisé par :

Chemaou El Fihri Bassma

Bennani Rania

Encadré par : Pr. Amimi Rajae

Dédicaces

Parce que chaque rêve réalisé porte en lui les empreintes de ceux qui nous ont soutenus, et que derrière chaque succès se cachent des mains tendues et des cœurs généreux... je dédie ce travail à ceux qui ont éclairé mon chemin.

À nos chers parents,

Pour leur soutien inconditionnel, leurs sacrifices inestimables et leurs encouragements constants tout au long de notre parcours académique. Pour leur patience, leur amour et leur confiance en nos capacités. Pour avoir été notre source de motivation dans les moments difficiles et notre première source de fierté dans nos réussites.

À nos familles,

Qui ont cru en nous depuis le premier jour, qui nous ont donné la force de persévérer face aux obstacles, et qui ont toujours été présentes pour nous soutenir moralement et matériellement. Pour leur compréhension durant les longues heures de travail et leur célébration de chaque petite victoire.

À tous les passionnés de technologie et d'innovation,

Qui voient dans chaque défi une opportunité d'apprentissage et de dépassement, qui croient au pouvoir de la technologie pour transformer le monde et améliorer notre quotidien. À ceux qui n'ont jamais cessé d'apprendre et d'explorer de nouveaux horizons.

À nos enseignants,

Qui nous ont transmis le savoir et la passion pour l'informatique et les technologies émergentes.

À tous ceux qui croient en l'importance de choisir les bonnes solutions technologiques pour répondre aux défis de demain,

Que ce travail soit le reflet de votre amour et de votre confiance en moi.

Remerciement

Au terme de ce travail, nous tenons à exprimer notre profonde gratitude et nos sincères remerciements à tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce projet.

Nous adressons nos remerciements les plus respectueux à **Madame AMIMI Rajae**, notre professeur, pour sa disponibilité, ses conseils judicieux, son suivi rigoureux et ses orientations précieuses qui nous ont permis de mener à bien ce projet. Son expertise dans le domaine des bases de données NoSQL nous a été d'une aide inestimable.

Je tiens à remercier cordialement tout le corps professionnel et administratif de **L'École Marocaine Des Sciences De L'Ingénieur De Rabat** pour leur soutien tout au long de ma formation. Leur dévouement et la qualité de leur enseignement m'ont permis d'acquérir les compétences nécessaires pour mener à bien ce projet.

Nous remercions également l'ensemble du corps professoral et administratif de la **4ème année Ingénierie Informatique et Réseaux** pour la qualité de la formation dispensée et pour nous avoir fourni les connaissances et compétences nécessaires à l'accomplissement de ce travail.

Nos remerciements s'étendent à nos collègues et camarades de promotion pour les échanges enrichissants, l'entraide et l'esprit d'équipe qui ont marqué notre parcours universitaire.

Enfin, merci à tous ceux qui ont participé de près ou de loin à l'élaboration de ce projet, à tous ceux qui ont cru en moi et m'ont encouragé dans cette aventure.

Résumé

Le présent projet s'inscrit dans le cadre de l'étude et de l'application des bases de données NoSQL. Il porte sur la conception et la mise en œuvre d'un système de gestion des circuits et courses automobiles, domaine caractérisé par une grande diversité et un volume important de données hétérogènes.

Le monde des courses automobiles génère quotidiennement des informations variées concernant les circuits, les dates des événements, les localisations géographiques, les capacités d'accueil des tribunes et les différentes classes de compétition. La gestion de ces données par le biais de bases de données relationnelles traditionnelles s'avère limitée en raison de la rigidité de leur structure et de leur difficulté à s'adapter aux évolutions constantes des besoins.

Face à cette problématique, nous avons opté pour une approche NoSQL basée sur MongoDB, qui offre la flexibilité nécessaire pour stocker des données semi-structurées et permet d'effectuer des requêtes performantes. Notre système organise les informations en deux collections principales : **tracks** pour les circuits et **racers** pour les courses associées, permettant ainsi d'établir des relations logiques entre ces entités.

Le développement technique repose sur un backend Python utilisant la bibliothèque PyMongo pour interagir avec la base de données MongoDB, tandis que le framework Django assure la création d'une interface front-end intuitive et interactive pour la visualisation et la manipulation des données.

L'architecture adoptée suit une approche polyglotte, conçue pour être évolutive et capable d'intégrer d'autres technologies NoSQL selon les besoins futurs. Cette conception permet d'envisager l'ajout de Redis pour la gestion de cache et de données en temps réel, Neo4j pour la modélisation de relations complexes entre pilotes, équipes et sponsors, ou encore Cassandra pour le stockage d'historiques massifs de courses.

Ce projet démontre la pertinence et l'efficacité des solutions NoSQL pour la gestion de données complexes et évolutives, et constitue une base solide pour des extensions futures répondant à des exigences métier plus avancées.

Mots-clés : NoSQL, MongoDB, Gestion de courses automobiles, Architecture polyglotte, Python, Django, PyMongo, Base de données orientée documents

Abstract

This project is conducted as part of the study and application of NoSQL databases. It focuses on the design and implementation of a management system for racing circuits and automobile races, a domain characterized by high diversity and significant volume of heterogeneous data.

The world of automobile racing generates daily various information concerning circuits, event dates, geographical locations, grandstand capacities, and different competition classes. Managing this data through traditional relational databases proves limited due to the rigidity of their structure and their difficulty in adapting to constantly evolving needs.

Faced with this challenge, we opted for a NoSQL approach based on MongoDB, which offers the necessary flexibility to store semi-structured data and enables high-performance queries. Our system organizes information into two main collections: **tracks** for circuits and **aces** for associated races, thereby establishing logical relationships between these entities.

The technical development relies on a Python backend using the PyMongo library to interact with the MongoDB database, while the Django framework ensures the creation of an intuitive and interactive front-end interface for data visualization and manipulation.

The adopted architecture follows a polyglot approach, designed to be scalable and capable of integrating other NoSQL technologies according to future needs. This design allows for the potential addition of Redis for cache management and real-time data, Neo4j for modeling complex relationships between drivers, teams and sponsors, or Cassandra for storing massive race histories.

This project demonstrates the relevance and efficiency of NoSQL solutions for managing complex and evolving data, and provides a solid foundation for future extensions meeting more advanced business requirements.

Keywords: NoSQL, MongoDB, Racing management, Polyglot architecture, Python, Django, PyMongo, Document-oriented databas

Table des matières

Dédicaces	i
Remerciement.....	ii
Résumé	iii
Abstract	iv
Liste des figures	vi
Liste des abréviations	vii
Chapitre 1 : Contexte, Problématique et Justification du choix NoSQL :	3
1.1 Contexte :	3
1.2 Problématique :	3
1.3 Justification du choix NoSQL :	3
Chapitre 2 : Objectifs du projet et Technologies utilisées	4
2.1 Objectifs du projet	4
2.2 Technologies utilisées	4
Chapitre 3 : Architecture polyglotte et Modèle de données	5
3.1 Qu'est-ce qu'une architecture polyglotte ?	5
3.2 Architecture polyglotte du projet	5
3.3 Modèle de données choisi	6
Bénéfices du projet	7
Conclusion	7

Liste des figures

Figure 1:Architecture polyglotte du projet	5
Figure 2:Partie de fichier JSON de la base de données tracks	6
Figure 3:Partie de fichier JSON de la base de données races.....	6

Liste des abréviations

Abbréviation	Signification
NoSQL	Not Only SQL (Non seulement SQL)
JSON	JavaScript Object Notation
CRUD	Create, Read, Update, Delete
ID	Identifiant

Dans le contexte actuel de la transformation numérique et de l'évolution rapide des technologies de gestion de données, le monde des courses automobiles, reconnu pour sa complexité technique et son innovation constante, génère quotidiennement un volume massif de données hétérogènes concernant les circuits, les événements sportifs, les dates de compétition, les localisations géographiques, les capacités d'accueil des infrastructures et les différentes classes de course.

La problématique identifiée révèle plusieurs lacunes dans les systèmes traditionnels : la rigidité des bases de données relationnelles face à la diversité des informations, l'absence de flexibilité pour s'adapter aux évolutions constantes du domaine, des difficultés à gérer efficacement les relations complexes entre circuits et courses, et un manque d'outils permettant des recherches rapides et des analyses statistiques pertinentes. Ces défis impactent directement l'efficacité opérationnelle et la qualité de l'exploitation des données.

Face à cette situation, notre mission consiste à développer un système de gestion intégré basé sur une approche NoSQL permettant de structurer, stocker et exploiter efficacement les données relatives aux circuits et courses automobiles. L'objectif principal est d'améliorer la gestion, la visibilité et l'accessibilité des informations tout en offrant une architecture flexible et évolutive.

Cette solution propose des fonctionnalités adaptées basées sur MongoDB comme base de données principale, Python et PyMongo pour le backend, et Django pour offrir une interface utilisateur claire et interactive. Le système permettra de gérer les deux entités principales : les circuits (tracks) et les courses (races), tout en facilitant les recherches, les statistiques et l'évolution vers une architecture polyglotte.

Ce rapport s'articule autour de six chapitres permettant de comprendre la démarche suivie pour aboutir à une solution pertinente et innovante.

Le premier chapitre présente le contexte général du projet en exposant les spécificités du domaine des courses automobiles, identifie la problématique liée à la gestion des données avec les systèmes traditionnels, et justifie le choix d'une approche NoSQL comme solution adaptée.

Le deuxième chapitre définit les objectifs précis du projet et détaille l'ensemble des technologies utilisées pour développer notre solution, en expliquant le rôle de chaque composant (MongoDB, Python, PyMongo et Django) dans l'architecture globale.

Le troisième chapitre introduit le concept d'architecture polyglotte, décrit son application dans notre projet, et présente le modèle de données retenu avec les deux collections principales (tracks et races) ainsi que leurs structures et relations.

La conclusion expose les bénéfices apportés par cette solution en termes de flexibilité, performance et évolutivité, avant de conclure sur la pertinence de l'approche NoSQL et les perspectives d'évolution du système vers une architecture polyglotte complète.

Chapitre 1 : Contexte, Problématique et Justification du choix NoSQL :

1.1 Contexte :

Le monde des courses automobiles, secteur en constante évolution et caractérisé par une activité intense, génère quotidiennement une multitude de données extrêmement variées et complexes. Ces informations couvrent un large spectre d'éléments essentiels à la gestion efficace des événements sportifs. Elles incluent notamment les caractéristiques techniques et géographiques des circuits telles que leur localisation précise, leur longueur en kilomètres, la configuration de leurs virages, la capacité d'accueil de leurs tribunes et leurs spécificités techniques. S'ajoutent à cela les détails opérationnels des courses elles-mêmes, comprenant les dates précises des événements, les horaires des différentes sessions (essais, qualifications, course), les classes de compétition concernées (Formule 1, Formule E, rallye, endurance), les listes de participants inscrits, ainsi que de nombreuses autres données évolutives nécessitant une gestion à la fois flexible et performante. Cette diversité et ce volume important d'informations hétérogènes imposent des exigences particulières en termes de stockage, d'organisation et d'exploitation des données.

1.2 Problématique :

Les systèmes de gestion traditionnels basés sur des bases de données relationnelles présentent plusieurs limitations dans ce contexte. La structure rigide des tables relationnelles rend difficile l'adaptation aux changements fréquents des données. Les relations entre circuits et courses nécessitent des jointures complexes qui impactent les performances. De plus, l'ajout de nouvelles informations impose souvent une restructuration coûteuse du schéma de la base de données.

1.3 Justification du choix NoSQL :

Face à ces contraintes, nous avons opté pour une approche NoSQL, spécifiquement MongoDB, pour plusieurs raisons. MongoDB offre une flexibilité de schéma permettant de stocker des documents JSON avec des structures variables. Cette base orientée documents facilite la modélisation naturelle des entités circuits et courses sans nécessiter de jointures complexes. Les requêtes sont rapides grâce à l'indexation efficace, et l'architecture permet une évolution progressive vers un système polyglotte intégrant d'autres technologies NoSQL selon les besoins futurs.

Chapitre 2 : Objectifs du projet et Technologies utilisées

2.1 Objectifs du projet

L'objectif principal de ce projet est de concevoir un système de gestion efficace et performant des circuits et courses automobiles. Ce système vise à organiser et structurer l'ensemble des données en deux collections distinctes et complémentaires : la collection tracks dédiée aux circuits et la collection races pour les courses associées. Il s'agit également de relier chaque course à son circuit correspondant de manière logique et performante, tout en permettant d'effectuer des recherches rapides et pertinentes sur les circuits et les événements planifiés. Le système doit également être capable de générer des statistiques simples mais utiles, comme le nombre de courses organisées par circuit, et offrir une interface utilisateur intuitive facilitant la visualisation claire des informations et la manipulation aisée des données par les utilisateurs finaux.

2.2 Technologies utilisées

Notre solution technique s'appuie sur un ensemble de technologies modernes et complémentaires formant un écosystème cohérent et performant. MongoDB a été sélectionné comme base de données NoSQL orientée documents, constituant la solution de stockage principale grâce à sa flexibilité remarquable et ses performances élevées dans la gestion de données semi-structurées. Le langage de programmation Python a été retenu pour développer l'ensemble de la logique métier du backend, offrant une syntaxe claire et une riche bibliothèque d'outils. PyMongo, bibliothèque Python spécialisée, assure l'interaction fluide entre l'application et la base de données MongoDB, en gérant efficacement les opérations CRUD (Create, Read, Update, Delete) ainsi que l'exécution de requêtes complexes. Enfin, le framework web Django prend en charge le développement du front-end, garantissant une interface utilisateur claire, interactive et responsive. Cette technologie soigneusement choisie garantit une solution moderne, facilement maintenable et hautement évolutive, capable de s'adapter aux besoins futurs du projet.

Chapitre 3 : Architecture polyglotte et Modèle de données

3.1 Qu'est-ce qu'une architecture polyglotte ?

Une architecture polyglotte consiste à utiliser plusieurs types de bases de données au sein d'un même système, chacune étant choisie pour ses forces spécifiques. Cette approche permet d'exploiter le meilleur de chaque technologie selon les besoins : bases documentaires pour les données flexibles, bases clé-valeur pour le cache rapide, bases graphes pour les relations complexes, et bases colonnes pour les volumes massifs.

3.2 Architecture polyglotte du projet

Actuellement, notre application repose exclusivement sur MongoDB pour gérer l'ensemble des informations relatives aux circuits et aux courses automobiles. Cependant, l'architecture globale du système a été délibérément conçue et structurée pour évoluer progressivement vers une solution polyglotte complète, capable d'intégrer plusieurs technologies de bases de données selon les besoins spécifiques de chaque type de donnée. Dans une perspective d'extension future, si nous devons intégrer des informations concernant les pilotes, des résultats détaillés des compétitions et des statistiques avancées en temps réel, le système sera prêt à accueillir d'autres technologies NoSQL complémentaires. Redis pourrait être intégré pour la gestion optimale du cache et le traitement des données en temps réel, permettant notamment d'afficher des classements live pendant les courses. Neo4j, base de données orientée graphes, serait idéale pour modéliser et explorer les relations complexes et interconnectées entre pilotes, équipes, sponsors et circuits. Cassandra, quant à elle, pourrait être déployée pour stocker et gérer l'historique massif des courses accumulées sur plusieurs années, offrant une excellente scalabilité horizontale. Cette conception modulaire et flexible garantit l'adaptabilité du système aux exigences futures et aux évolutions des besoins métier.

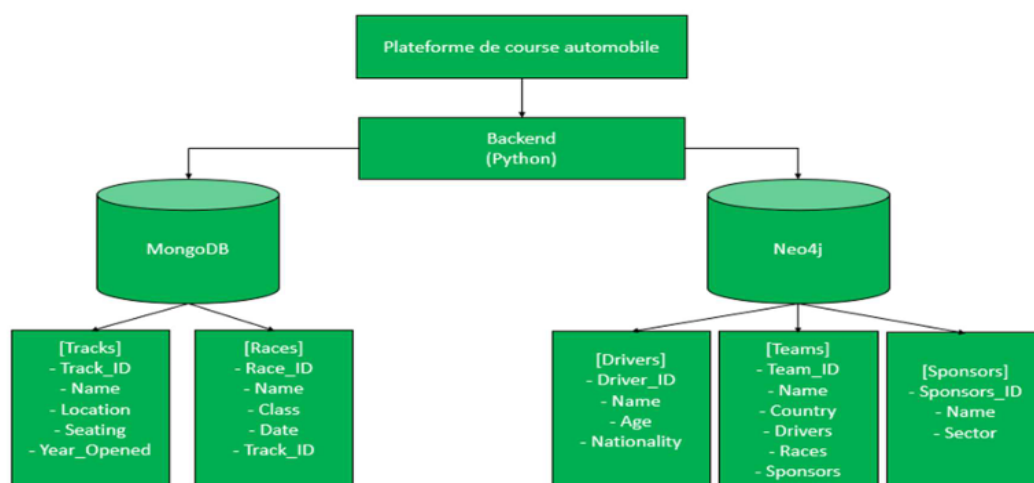


Figure 1: Architecture polyglotte du projet

3.3 Modèle de données choisi

Notre modèle de données s'organise autour de deux collections MongoDB principales :

Collection Tracks (Circuits) :

```

    _id: ObjectId('692d94bf468d1ba7b49c65f3')
  ▾ track: Array (9)
    ▾ 0: Object
      Track_ID : 1
      Name : "Auto Club Speedway"
      Location : "Fontana, CA"
      Seating : "92000"
      Year_Opened : "1997"

```

Figure 2:Partie de fichier JSON de la base de données tracks

Collection Races (Courses) :

```

  ▾ race : Array (7)
    ▾ 0: Object
      Race_ID : 1
      Name : "Rolex 24 At Daytona"
      Class : "DP/GT"
      Date : "January 26 January 27"
      Track_ID : "1"

```

Figure 3:Partie de fichier JSON de la base de données races

Le champ `track_id` dans la collection `races` établit la relation avec le circuit correspondant. Cette structure permet des requêtes performantes tout en maintenant la flexibilité nécessaire pour ajouter de nouveaux champs sans migration complexe.

Bénéfices du projet

Ce projet apporte plusieurs avantages significatifs qui en font une solution particulièrement pertinente pour la gestion des circuits et courses automobiles. La flexibilité constitue le premier atout majeur du système, offrant la capacité de faire évoluer le modèle de données sans nécessiter de restructuration majeure, permettant ainsi d'ajouter de nouveaux champs ou de modifier les structures existantes en fonction des besoins émergents. Les performances exceptionnelles représentent un autre bénéfice important, avec des requêtes rapides rendues possibles grâce aux mécanismes d'indexation efficaces de MongoDB et à l'absence de jointures complexes qui ralentissent habituellement les systèmes relationnels traditionnels. La scalabilité du système garantit une architecture prête à accueillir de nouvelles technologies et capable de gérer des volumes de données croissants au fur et à mesure de l'expansion du projet. La facilité d'utilisation est assurée par l'interface Django intuitive et conviviale, offrant aux utilisateurs finaux une expérience agréable et efficace lors de la manipulation et de la visualisation des données. Enfin, l'évolutivité du système constitue un avantage stratégique majeur, fournissant une base solide et robuste pour intégrer progressivement des fonctionnalités avancées telles que des outils d'analytics sophistiqués, le traitement de données en temps réel, ou encore la modélisation de graphes relationnels complexes, assurant ainsi la pérennité et l'adaptabilité de la solution face aux défis futurs.

Conclusion

Ce projet démontre la pertinence des bases de données NoSQL, et particulièrement MongoDB, pour gérer des données complexes et évolutives dans le domaine des courses automobiles. La solution développée répond efficacement aux problématiques identifiées en offrant flexibilité, performance et facilité d'utilisation.

L'architecture polyglotte adoptée constitue une force majeure du système, le rendant capable d'évoluer progressivement selon les besoins futurs. Que ce soit pour intégrer des données en temps réel, modéliser des relations complexes ou stocker des historiques massifs, le système est conçu pour s'adapter sans remise en question fondamentale.

Ce travail pose ainsi les fondations d'une solution moderne et pérenne, capable de répondre aux défis actuels tout en anticipant les exigences de demain. L'architecture est prête à évoluer, flexible et cohérente, offrant une base solide pour des extensions futures répondant à des besoins métier plus avancés.