

# EXAMEN DE BASE DE DONNÉES NON-RELATIONNELLES (NOSQL) - SESSION DE RATTRAPAGE

---

**Université :** UPB – Université Polytechnique de Bingerville

**Professeur :** Robert Yavo

**Date :** 2 Décembre 2024

**Durée :** 3 heures

**Langage :** MongoDB, JavaScript, Python, HTML

**Supports :** Cours de Base de Données Relationnelles (RY)

**Étudiant :** Abouho Sana Franklin Prince YAVO

## RÉPONSES AUX QUESTIONS

---

**NB :** Les réponses aux questions posées dans le sujet données ci-dessus peuvent être accompagnées de fichiers SQL ou PDF.

### 1) Diagramme de Modèle de Données NoSQL

#### Description des collections

##### 1. Événements Olympiques (evenementsOlympiques)

- **Champs :**
  - `_id` (ObjectId)
  - `NoÉvénement` (String)
  - `Année` (Number)
  - `DateDébut` (Date)
  - `DateFin` (Date)
  - `Pays` (String)

##### 2. Compétitions (competitions)

- **Champs :**
  - `_id` (ObjectId)
  - `NomCompétition` (String)
  - `DateHeureDébut` (Date)
  - `DateHeureFin` (Date)
  - `ÉvénementOlympique` (Référence à `evenementsOlympiques` )

- NoStade (Référence à stades )
- Athlètes (Array de Références à athletes )
- Officiers (Array de Références à officiers )
- Médaille (Array de {athlète, médaille})
- Ville (String)
- Images (Array de URLs)
- Vidéos (Array de URLs)

### 3. Stades (stades)

- **Champs :**

- \_id (ObjectId)
- NomStade (String)
- Pays (String)
- Province (String)
- Ville (String)
- Adresse (String)
- Téléphone (String)

### 4. Athlètes (athletes)

- **Champs :**

- \_id (ObjectId)
- Nom (String)
- Prénom (String)
- Pays (String)
- Âge (Number)
- Jeux (Array de Références à jeux )
- Photo (URL)
- Médaille (Array de {type de médaille, compétition})

### 5. Officiers (officiers)

- **Champs :**

- \_id (ObjectId)
- Nom (String)
- Prénom (String)
- Pays (String)
- Âge (Number)
- Jeux (Array de Références à jeux )
- Téléphone (String)

## 6. Jeux (jeux)

- Champs :
  - \_id (ObjectId)
  - NoJeux (String)
  - NomJeux (String)
  - Compétitions (Array de Références à competitions )
  - Athlètes (Array de Références à athletes )

## 2) Création de la base de données NoSQL

### Création des collections

Voir le fichier `ytg_no_sql.js` pour le script de création des collections.

```
use("ytg");

const WORKING_COLLECTIONS = [
  "evenementsOlympiques",
  "competitions",
  "stades",
  "athletes",
  "officiers",
  "jeux",
  "restaurants",
];

// Création des collections

for (const collection of WORKING_COLLECTIONS) {
  db.getCollection(collection)
    ? db.getCollection(collection).drop()
    : db.createCollection(collection);
}

// Insertion des documents dans les collections
db.getCollection("evenementsOlympiques").insertMany([
  {
    noEvenement: "JO2024",
    annee: 2024,
    dateDebut: new Date("2024-07-26"),
    dateFin: new Date("2024-08-11"),
    pays: "France",
  },
]);

db.getCollection("competitions").insertMany([
  {
    noStade: "STADE1",
```

```
    nomCompétition: "Volley-ball",
    dateHeureDebut: new Date("2024-07-26T08:00:00"),
    dateHeureFin: new Date("2024-07-26T10:00:00"),
    evenementOlympique: "JO2024",
    ville: "Paris",
    images: ["https://example.com/image1.jpg"],
    videos: ["https://example.com/video1.mp4"],
    athletes: ["ATH1", "ATH2"],
    officiers: ["OFF1", "OFF2"],
  },
  {
    noStade: "STADE2",
    nomCompétition: "Cyclisme",
    dateHeureDebut: new Date("2024-07-26T08:00:00"),
    dateHeureFin: new Date("2024-07-26T10:00:00"),
    evenementOlympique: "JO2024",
    ville: "Marseille",
    images: ["https://example.com/image2.jpg"],
    videos: ["https://example.com/video2.mp4"],
    athletes: ["ATH3", "ATH4"],
    officiers: ["OFF5", "OFF1"],
  },
  {
    noStade: "STADE3",
    nomCompétition: "Football",
    dateHeureDebut: new Date("2024-07-26T08:00:00"),
    dateHeureFin: new Date("2024-07-26T10:00:00"),
    evenementOlympique: "JO2024",
    ville: "Lyon",
    images: ["https://example.com/image3.jpg"],
    videos: ["https://example.com/video3.mp4"],
    athletes: ["ATH5", "ATH2"],
    officiers: ["OFF1", "OFF4"],
  },
  {
    noStade: "STADE4",
    nomCompétition: "Handball",
    dateHeureDebut: new Date("2024-07-26T08:00:00"),
    dateHeureFin: new Date("2024-07-26T10:00:00"),
    evenementOlympique: "JO2024",
    ville: "Nice",
    images: ["https://example.com/image4.jpg"],
    videos: ["https://example.com/video4.mp4"],
    athletes: ["ATH5", "ATH3"],
    officiers: ["OFF3", "OFF2"],
  },
  {
    noStade: "STADE5",
    nomCompétition: "Tennis",
    dateHeureDebut: new Date("2024-07-26T08:00:00"),
    dateHeureFin: new Date("2024-07-26T10:00:00"),
    evenementOlympique: "JO2024",
    ville: "Bordeaux",
```

```
images: ["https://example.com/image5.jpg"],
videos: ["https://example.com/video5.mp4"],
athletes: ["ATH1", "ATH4"],
officiers: ["OFF1", "OFF2"],
},
]);

db.getCollection("stades").insertMany([
{
  noStade: "STADE1",
  nomStade: "Stade de France",
  pays: "France",
  province: "Île-de-France",
  ville: "Saint-Denis",
  adresse: "93216 Saint-Denis",
  telephone: "+33 1 55 93 00 00",
},
{
  noStade: "STADE2",
  nomStade: "Vélodrome",
  pays: "France",
  province: "Provence-Alpes-Côte d'Azur",
  ville: "Marseille",
  adresse: "3 Boulevard Michelet, 13008 Marseille",
  telephone: "+33 4 91 13 20 00",
},
{
  noStade: "STADE3",
  nomStade: "Groupama Stadium",
  pays: "France",
  province: "Auvergne-Rhône-Alpes",
  ville: "Décines-Charpieu",
  adresse: "10 Avenue Simone Veil, 69150 Décines-Charpieu",
  telephone: "+33 4 72 10 00 00",
},
{
  noStade: "STADE4",
  nomStade: "Allianz Riviera",
  pays: "France",
  province: "Provence-Alpes-Côte d'Azur",
  ville: "Nice",
  adresse: "Boulevard des Jardiniers, 06200 Nice",
  telephone: "+33 4 97 07 00 00",
},
{
  noStade: "STADE5",
  nomStade: "Matmut Atlantique",
  pays: "France",
  province: "Nouvelle-Aquitaine",
  ville: "Bordeaux",
  adresse: "Cours Jules Ladoumègue, 33300 Bordeaux",
  telephone: "+33 5 56 16 20 20",
},
],
```

```
]);
```

```
db.getCollection("athletes").insertMany([
  {
    noAthlete: "ATH1",
    nom: "Doe",
    prenom: "John",
    pays: "USA",
    age: 25,
    jeux: ["JO2024"],
    photo: "https://example.com/johndoe.jpg",
    medaille: [{ type: "Or", competition: "Volley-ball" }],
  },
  {
    noAthlete: "ATH2",
    nom: "Smith",
    prenom: "Jane",
    pays: "USA",
    age: 22,
    jeux: ["JO2024"],
    photo: "https://example.com/janesmith.jpg",
    medaille: [{ type: "Argent", competition: "Volley-ball" }],
  },
  {
    noAthlete: "ATH3",
    nom: "Johnson",
    prenom: "Alice",
    pays: "USA",
    age: 28,
    jeux: ["JO2024"],
    photo: "https://example.com/alicejohnson.jpg",
    medaille: [{ type: "Bronze", competition: "Cyclisme" }],
  },
  {
    noAthlete: "ATH4",
    nom: "Brown",
    prenom: "Bob",
    pays: "USA",
    age: 30,
    jeux: ["JO2024"],
    photo: "https://example.com/bobbrown.jpg",
    medaille: [{ type: "Or", competition: "Tennis" }],
  },
  {
    noAthlete: "ATH5",
    nom: "Williams",
    prenom: "Will",
    pays: "USA",
    age: 26,
    jeux: ["JO2024"],
    photo: "https://example.com/willwilliams.jpg",
    medaille: [{ type: "Argent", competition: "Handball" }],
  },
],
```

```
]);
```

```
db.getCollection("officiers").insertMany([
  {
    noOfficier: "OFF1",
    nom: "Doe",
    prenom: "John",
    pays: "USA",
    age: 35,
    jeux: ["J02024"],
    telephone: "+1 555 555 5555",
  },
  {
    noOfficier: "OFF2",
    nom: "Smith",
    prenom: "Jane",
    pays: "USA",
    age: 32,
    jeux: ["J02024"],
    telephone: "+1 555 555 5556",
  },
  {
    noOfficier: "OFF3",
    nom: "Johnson",
    prenom: "Alice",
    pays: "USA",
    age: 38,
    jeux: ["J02024"],
    telephone: "+1 555 555 5557",
  },
  {
    noOfficier: "OFF4",
    nom: "Brown",
    prenom: "Bob",
    pays: "USA",
    age: 40,
    jeux: ["J02024"],
    telephone: "+1 555 555 5558",
  },
  {
    noOfficier: "OFF5",
    nom: "Williams",
    prenom: "Will",
    pays: "USA",
    age: 36,
    jeux: ["J02024"],
    telephone: "+1 555 555 5559",
  },
]);
```

```
db.getCollection("jeux").insertMany([
  {
    noJeux: "J02024",
```

```
    nomJeux: "Jeux Olympiques 2024",  
    competitions: ["Volley-ball", "Cyclisme", "Football", "Handball", "Tennis"],  
    athletes: ["ATH1", "ATH2", "ATH3", "ATH4", "ATH5"],  
    officiers: ["OFF1", "OFF2", "OFF3", "OFF4", "OFF5"],  
  },  
]);
```

### 3) Requêtes MongoDB

a) Afficher le nom, prénom des athlètes dont l'Age est compris entre 20 et 30 ans

```
// mongosh  
use("ytg");  
db.getCollection("athletes").find({ age: { $gte: 20, $lte: 30 } }, { nom: 1, prenom: 1, _id: 0 })
```

```
>_MONGOSH  
  
> use ytg  
< switched to db ytg  
> db.getCollection("athletes").find({ age: { $gte: 20, $lte: 30 } }, { nom: 1, prenom: 1, _id: 0 });  
< {  
  nom: 'Doe',  
  prenom: 'John'  
}  
{  
  nom: 'Smith',  
  prenom: 'Jane'  
}  
{  
  nom: 'Johnson',  
  prenom: 'Alice'  
}  
{  
  nom: 'Brown',  
  prenom: 'Bob'  
}  
{  
  nom: 'Williams',  
  prenom: 'Will'  
}  
ytg>
```

3-b) Afficher la liste des compétitions qui ont lieu à Paris

```
// mongosh  
use("ytg");  
db.getCollection("competitions").find({ ville: "Paris" }, { _id: 0 });
```



```
> _MONGOSH

> use ytg
< switched to db ytg
> db.getCollection("competitions").find({ ville: "Paris" }, { _id: 0 });
< {
  noStade: 'STADE1',
  nomCompetition: 'Volley-ball',
  dateHeureDebut: 2024-07-26T08:00:00.000Z,
  dateHeureFin: 2024-07-26T10:00:00.000Z,
  evenementOlympique: 'JO2024',
  ville: 'Paris',
  images: [
    'https://example.com/image1.jpg'
  ],
  videos: [
    'https://example.com/video1.mp4'
  ],
  athletes: [
    'ATH1',
    'ATH2'
  ],
  officiers: [
    'OFF1',
    'OFF2'
  ]
}
ytg> |
```

## 4) Importation des données des restaurants

a) Importer les données restaurants.json dans votre base de données de Jeux Olympique dans une collection nommée restaurants à l'aide de MongoDB Shell

```
# mongosh
mongoimport --db ytg --collection restaurants --file ytg.restaurants.json
```

```
Windows PowerShell
PS C:\laragon\www\me\M1\ytg_mcd\nosql> mongoimport --db ytg --collection restaurants --file ytg.restaurants.json
2024-12-02T22:53:39.681+0000 connected to: mongodb://localhost/
2024-12-02T22:53:39.682+0000 Failed: cannot decode array into a primitive.D
2024-12-02T22:53:39.682+0000 0 document(s) imported successfully. 0 document(s) failed to import.
PS C:\laragon\www\me\M1\ytg_mcd\nosql> mongoimport --db ytg --collection restaurants --file ytg.restaurants.json --json
Array --verbose
2024-12-02T22:53:44.313+0000 using write concern: &{majority <nil> 0s}
2024-12-02T22:53:44.321+0000 filesize: 2155 bytes
2024-12-02T22:53:44.321+0000 using fields:
2024-12-02T22:53:44.321+0000 connected to: mongodb://localhost/
2024-12-02T22:53:44.321+0000 ns: ytg.restaurants
2024-12-02T22:53:44.321+0000 connected to node type: standalone
2024-12-02T22:53:44.350+0000 10 document(s) imported successfully. 0 document(s) failed to import.
PS C:\laragon\www\me\M1\ytg_mcd\nosql> |
```

b) Écrire les requêtes MongoDB qui permettent d'afficher tous les restaurants dont la clé est "borough" et la valeur "Brooklyn". Quel est nombre total de documents affichés

```
// mongosh
use("ytg");
db.getCollection("restaurants").find({ borough: "Brooklyn" });
```

c) Écrire la requête qui affiche les restaurants qui ont un grade 'C' avec un score inférieur à 40

```
// mongosh
use("ytg");
db.getCollection("restaurants").find({ grade: "C", score: { $lt: 40 } });
```

d) Écrire une requête qui permet d'afficher la liste des valeurs distinctes de grade

```
// mongosh
use("ytg");
db.getCollection("restaurants").distinct("grade");
```

## 5) Création d'un formulaire simple pour afficher la liste des Athlètes qui ont une médaille d'or

Voir le dossier `ytg_no_sql_form` pour le formulaire simple en HTML et JavaScript.

### Installations

```
cd ytg_no_sql_form
python server.py

# ou

cd ytg_no_sql_form
python -m venv venv
python -m pip install -r requirements.txt
python -m streamlit run app.py
```

### Ressources

- `ytg_no_sql_mcd_docs.md` -> Fichier Markdown contenant les réponses aux questions de l'examen.
- `ytg_no_sql_mcd_docs.pdf` -> Fichier PDF généré à partir du fichier Markdown.
- [Visualisé sur app.eraser.io](https://app.eraser.io) -> Modèle Entité-Association et Modèle Relationnel de la base de données de la société YTG (Miro). <https://app.eraser.io/workspace/cuDkXG3fL3Q4yqkOA5SC?>

[origin=share](#)

- **ytg\_no\_sql.js** -> Script de création des collections de la base de données de la société YTG.
- **ytg\_no\_sql.zip** -> Fichiers SQL, PDF et PNG.