

## Projet de NoSQL M2 MIAGE-IF

### Enoncé

On souhaite modéliser une base de données stockant des informations issues de la *Base Carbone*<sup>1</sup>, permettant d'évaluer le bilan carbone de différents menus de restauration.

Dans cette base de données, chaque ingrédient est caractérisé par un nom et une quantité d'émission de GES (Gaz à Effet de Serre). Chaque plat est caractérisé par un nom et est associé à une et une catégorie (entrée, plat principal, dessert). Chaque plat est composé d'un ou plusieurs ingrédients). La quantité d'émissions de GES de chaque plat est calculée à partir de celles des ingrédients qui le composent. Un menu est caractérisé par un nom et est composé de plats (un seul plat par catégorie). La quantité d'émissions de GES de chaque menu est calculée à partir de celle de chacun de ses plats.

Les copies d'écran ci-dessous donnent l'exemple de 3 menus issus de la *Base Carbone*.

Types de repas	gCO <sub>2e</sub>
Entrée : légumes à la grecque	
légumes de saison	53,4
huile d'olive (1/2 c.s.)	18,2
Plat principal : poulet au riz	
poulet	774
riz	84,6
beurre	94,9
Plateau de fromages	
fromage à pâte molle	107
fromage à pâte dure	140
Pain	76
<b>Total</b>	<b>1350</b>

**Repas classique 1 (avec poulet)**

Types de repas	gCO <sub>2e</sub>
Entrée : tzatziki	
yaourt	360
concombre	129
huile d'olive (1/2 c.s.)	18,2
Plat principal : bifteck - frites	
bifteck	5370
frites	260
Dessert : tarte aux poires	
farine	46,8
poires	71
huile (1 c.s.)	32,3
<b>Total</b>	<b>6290</b>

**Repas classique 2 (avec bœuf)**

<sup>1</sup> <https://www.data.gouv.fr/fr/datasets/5f5605e63ee74d0034395703/>

Types de repas	gCO <sub>2e</sub>
Entrée : soupe de légumes	
200g légumes de saison	53.4
huile (1/2 c.s.)	15,1
Plat principal : omelette aux pommes de terre	
2 œufs	276,7
pommes de terre	15.5
huile (1/2 c.s.)	17,1
Dessert : salade de fruits	
fruits de saison	53.4
Pain	76
<b>Total</b>	<b>510</b>

**Repas végétarien 1**

La base de données doit permettre de répondre aux requêtes suivantes :

- Quelle est l'empreinte carbone d'un ingrédient donné ?
- Quelle est l'empreinte carbone d'un plat donné et quels sont les ingrédients composant un plat (avec leur empreinte carbone) ?
- Quelles sont la composition et l'empreinte carbone de chacun des plats composant un menu (avec les détails sur la composition des plats comme sur les affichages précédents par exemple) ?
- Quels sont les plats (avec leur empreinte carbone) contenant un ingrédient donné ?
- Quels sont les ingrédients, plats ou menus ayant la plus faible empreinte carbone ou une empreinte inférieure à un seuil donné ?

## Travail à réaliser

Pour ce projet, à effectuer **seul.e ou en binôme<sup>2</sup>** ; vous devez, pour chaque modèle ci-dessous, donner les instructions de création des données et les requêtes associées :

1. Un modèle clé-valeur sous *Redis*.
2. Un modèle sous *PostgreSQL* dont certains attributs sont des couples clé-valeur.
3. Un modèle document sous *MongoDB*.
4. Un modèle sous *PostgreSQL* dont certains attributs sont des données *JSON*.
5. Un modèle graphe sous *Neo4j*.

---

<sup>2</sup> Un binôme est formé de maximum 2 personnes. Aucun groupe de 3 personnes n'est autorisé.

Il n'est pas demandé de développer d'application mais uniquement d'implémenter les modèles de données et les requêtes de l'application.

Il n'est pas demandé de développer d'application mais uniquement d'implémenter les modèles de données et les requêtes de l'application. **L'objectif est de comparer les différents modèles d'un point de vue qualitatif et quantitatif.** Vous pouvez vous générer des données fictives via un programme externe, ou faire générer des données (respectant votre modèle) via un outil d'IA (type *ChatGPT*). Vous pouvez également comparer le temps d'exécution des requêtes (pour donner un ordre d'idée même si ces dernières sont exécutés sur des systèmes et des serveurs différents – cf. en annexe pour les instructions).

## Rendu du TP

Pour ce projet, vous devrez notamment :

1. Rédiger un **rapport<sup>3</sup> (10-15 pages maximum)** contenant :
  - Une description des différents modèle des données avec les commandes de création.
  - L'expression de vos requêtes pour chaque modèle et le résultat obtenu sur des exemples.
  - Une analyse de chaque modèle en indiquant les avantages de vos choix de modélisation et leurs limites.
  - Une comparaison de vos différents modèles.
  - Une conclusion indiquant les difficultés rencontrées, la répartition du travail entre les membres du binôme et le temps passé sur ce projet.
2. **Réaliser une présentation orale de 5-7 mn avec des transparents et éventuellement une démonstration**, rappelant certains points évoqués dans le rapport. Chaque membre du binôme devra intervenir de manière équitable lors de cette présentation.

## Modalités

Le projet est seul.e ou en binôme<sup>4</sup>.

Vos fichiers (dont le nom indiquera les noms de familles des membres du binôme) devront être **déposé sur moodle avant le 30/05/23 8h**. Ces fichiers devront contenir :

- La version électronique du rapport (**1 fichier pdf**)
- La version électronique de vos transparents (**1 fichier pdf**).
- Un **fichier zip contenant** : Les scripts de création de vos modèles et les requêtes (en vue de les rejouer sur les moteurs utilisés).

La **présentation orale aura lieu pendant le cours du 30/05/2023** matin en séance de cours (selon un planning, qui vous sera communiqué).

---

<sup>3</sup> Rappel : tout rapport doit comporter une introduction et une conclusion.

<sup>4</sup> Aucun groupe de 3 n'est autorisé.

## Annexe

Pour obtenir le temps d'exécution d'une requête vous avez :

- L'instruction SQL *EXPLAIN ANALYZE* sous *PostgreSQL* à placer devant la requête cf. <https://www.postgresql.org/docs/current/using-explain.html>
- L'instruction *TIME* sous *Redis* – cf. <https://redis.io/commands/time/>
- L'instruction *.explain("executionStats")* sous *MongoDB*, à placer à la fin de la requête – cf. <https://www.mongodb.com/docs/manual/tutorial/analyze-query-plan/>
- Le temps d'exécution est affiché à la fin des requêtes dans la console Neo4j

```
Query:
LOAD CSV WITH HEADERS FROM "https://raw.githubusercontent.com/inserpio/tour-de-france-2014/m
toInt(csvLine.RACE_ID), name: csvLine.RACE_NAME, from: csvLine.RACE_FROM, to: csvLine.RACE_T
csvLine.RACE_NUMBER_OF_STAGES, website: csvLine.RACE_WEBSITE }) MERGE (t:Team { id: toInt(c
sportingDirectors: csvLine.TEAM_MANAGERS }) MERGE (p:Rider { name: csvLine.RIDER_NAME, coun
number: toInt(csvLine.RIDER_NUMBER), info: csvLine.RIDER_INFO })-(p), (p)-[:RIDES_FOR { year

Query took 318 ms and returned no rows.
Updated the graph - created 221 nodes and 594 relationships set 1086 properties Result Details
```