



Projet Accès aux données

# Table des matières

A. Contexte	p. 1
B. Le système central (SGBDR)	p. 1
C. Le système distribué (MongoDB)	p. 2
i. Système SPORT SALUT	p. 2
ii. Système GameEZ	p. 3
iii. Système MEDIDONC	p. 4
iv. Que faire de ces syntaxes	p. 4
D. Le programme en lui-même	p. 4
E. Modalités	p. 5
i. Bonus potentiels	p. 5
ii. Malus potentiels	p. 5

## Contexte

Vous devrez réaliser une plateforme centrale, une base de données de produits avec leurs références.

Le système central dispose d'une base de données relationnelles de type MySQL ou MariaDB. Il stockera des produits et des fournisseurs, ainsi que la disponibilité des produits (stock). Le détail des colonnes de ces tables est présenté plus loin dans ce document.

Cette base de données de produits est utilisée par de nombreux systèmes divers et variés de sites e-commerce en ligne. Chacun de ces systèmes utilise sa propre syntaxe. Pour autant, les équipes gérant la base centrale ne s'occupent pas des sites e-commerce.

Votre projet possèdera donc deux parties :

- Une base de données relationnelle pour stocker les produits et leurs fournisseurs
- Une base de données MongoDB pour stocker les produits et leurs fournisseurs avant vente sur les sites e-commerce, que l'on appellera plus tard revendeur ou magasin.

## Le système central (SGBDR)

Le système central possède trois tables :

- produits
- fournisseurs
- categories

Colonnes de la table produits :

- id
- nom
- description
- prix\_achat
- statut (disponible, en rupture)
- date\_creation
- date\_modification

Colonnes de la table fournisseurs :

- id
- nom
- date\_creation

Colonnes de la table categories :

- id
- nom
- date\_creation

Il faudra également penser à faire les relations entre les différentes tables.

Vous devrez créer les procédures/fonctions suivantes :

- Créer un produit. Si le fournisseur et la catégorie sont spécifiés, crée également le fournisseur et/ou la catégorie
- Modifier un produit
- Modifier un fournisseur
- Supprimer un fournisseur. Cela supprimera tous les produits associés

Ces procédures/fonctions seront utilisées par votre programme : vous devrez donc créer un utilisateur sur le SGBDR qui n'aura accès qu'aux fonctions SELECT et à l'appel de ces procédures.

**Dit autrement : votre programme ne pourra pas utiliser des requêtes commençant par `INSERT`, `UPDATE` ou `DELETE`.**

Il faudra en plus créer une procédure par système distribué (expliqué plus bas dans ce document). Par exemple, un magasin revendeur nommé "Sport Salut" ayant le droit de récupérer les articles de sport devrait posséder une procédure `produits_sportsalut` ramenant uniquement les articles de sport de la base de données. La structure des données est laissée au choix de chacun des systèmes distribués.

Vous pouvez créer, si vous le souhaitez, une nouvelle table contenant les systèmes distribués et les catégories auxquels ceux-ci ont accès.

## Le système distribué (MongoDB)

Dans ce système distribué, on retrouvera des données au format JSON qui ont des structures beaucoup plus flexibles. Voici le structures :

### Système SPORT SALUT

Voici un produit défini par SPORT SALUT :

```
[
  {
    "nom_produit": "Nom du produit",
    "description_produit" : "Description du produit",
    "nom_fournisseur": "Nom fournisseur",
    "en_stock": "Oui", // Peut aussi être "Non"
    // Formule de calcul automatique : prix_achat + 20%
    "prix": 20,
  },
  // ...
]
```

On imagine que cette société s'intéresse exclusivement aux articles de sport.

## Système GameEZ

Voici un type de produit défini par GameEZ :

```
[
  {
    "results": [
      {
        "product": {
          "product_name": "Nom du produit",
          "product_description": "Description du produit",
          // Calcul automatique : +10% si jeu vidéo,
          // +15% si jeu de société
          "product_price": 20,
          "product_status": "available", // ou "unavailable"
        },
        "seller": {
          "seller_name" : "Nom du fournisseur",
          "seller_creation_date" : "Date de création du fournisseur"
        }
      }
    ]
  },
  // ...
]
```

On suppose que cette société s'intéresse aux arts ludiques de manière générale, donc jeux vidéo et jeux de société.

## Système MEDIDONC

Voici un type de produit défini par MEDIDONC :

```
{
  "p_name": "Mon produit",
  "p_description" : "Description du produit",
  "p_last_update": "2024-02-10 18:00:00", // Dernière modification
  "p_status": "En stock", // Peut être "Rupture de stock"
  "p_seller": "{ \"id\":1, \"name\": \"Nom fournisseur\", \"creation_date\": \"2024-01-12\" }"
}
```

On suppose que cette société se spécialise dans les articles de santé, et cela peut inclure certains articles de sport.

## Que faire de ces syntaxes

Votre objectif est de stocker dans votre base MongoDB les différents produits et fournisseurs tel quel ou presque. Le principe est d'avoir un minimum de traçabilité.

## Le programme en lui-même

Une fois ces deux systèmes de gestion de bases de données réalisés, vous aurez besoin d'un programme ou d'un site web qui permet d'ajouter des produits à la base centrale. Ce programme permettra de saisir les différents champs associés aux produits et aux fournisseurs : les créer, les modifier et les supprimer, par le biais de `CALL`.

Vous aurez également besoin, pour vérifier le bon fonctionnement, d'avoir un programme qui récupère périodiquement les données des différents sites e-commerce. Ce programme se contentera d'appeler les trois procédures liées à chacun des magasins, par le biais de `CALL`.

Bonus : votre programme pourrait être suffisamment flexible pour activer et désactiver des "systèmes" (par exemple : désactiver le site e-commerce SPORT SALUT).

Si vous préférez, vous pouvez créer un seul programme : l'important est que les éléments demandés existent et soient fonctionnels.

## Modalités

Les modalités de ce projet sont les suivantes :

- Ce projet devra être réalisé en groupe de 4 à 5 personnes
- Votre rendu devra être effectué dans une archive ZIP envoyée sur Yparéo, nous vous conseillons donc une taille inférieure à 25Mo (voir livrables).
- Les seuls langages autorisés pour ce projet sont HTML, CSS et JavaScript côté front, ainsi JavaScript, SQL côté back. Une tolérance est permise pour TypeScript.
- Vous êtes autorisés à utiliser yarn, npm, pnpm, tant que les seuls packages installés ne permettent pas de résoudre une fonctionnalité.
- Les frameworks client comme serveur sont autorisés, ainsi que leurs extensions, tant qu'ils ne permettent pas de résoudre une fonctionnalité.

## Bonus potentiels

L'ajout de fonctionnalités supplémentaires ou un travail particulièrement qualitatif sur un élément du projet peut donner droit à des points supplémentaires.

## Malus potentiels

Des malus pourront être appliqués :

- En cas de copier/coller trop important d'une source, que ce soit internet ou un autre groupe
- Retard d'envoi du projet
- Hors sujet
- Faille de sécurité manifeste et vue en cours (requêtes SQL non préparées, mots de passe non hashés)
- En cas d'une utilisation abusive de LLM (ChatGPT etc)