



Développeur

“ Ce test a pour but d’évaluer les connaissances et les compétences requises pour le poste de Développeur
L’objectif est de nous assurer de votre capacité, à partir d’une problématique donnée, à savoir modéliser un flux de données, rédiger des requêtes SQL et construire un algorithme simple. ”





Apports de céréales

Problématique

Nous sommes dans le contexte métier d'une coopérative agricole.

En période de collecte, les agriculteurs viennent à la coopérative apporter les céréales qu'ils ont récoltées.

Informatiquement, pour chaque camion se présentant chez elle, la coopérative enregistrera un « Apport » pour **1** Agriculteur et **1** céréale (Mais, Ble, Orge, ...)

L'édition résultant de cette apport pourrait se présenter comme suit :

Coopérative Grainbow

Centre ville

APPORT

SILO DE RECEPTION

SILO CEREALES BIOS

Rue des silos

46500 BIO

Agriculteur n° CB6

Route de Bayonne

Bat 1

64000 PAU

NUMERO DE PIECE : **APC001238**

DATE D'OPERATION : 10/04/2024

TRANSPORT : AGRICULTEUR

IMMATRICULATION :

RECOLTE : 2023

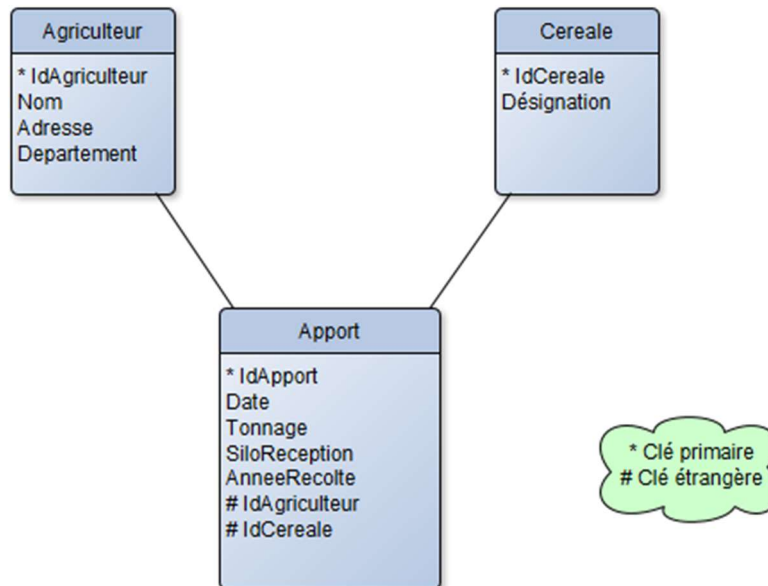
CEREALE : 11-18

DESIGNATION : **BLE PAYSAN**

POIDS LIVRE : **19,420** TO

Modélisation

Sur la base des informations ci-dessus, construisez un Diagramme Entité-Relation simplifié qui décrit l'organisation des données stockées dans le système.



SQL

A partir du diagramme construit précédemment écrire les requêtes SQL permettant de :

- 1) Connaitre la liste des agriculteurs (Id et Nom) ayant apporté la céréale avec la désignation « Ble Paysan » sur l'année 2024

Version courte:

```
SELECT A.idAgriculteur, A.Nom
FROM Agriculteur A
INNER JOIN Apport AP ON A.idAgriculteur = AP.idAgriculteur
INNER JOIN Céréale C ON AP.idCereale = C.idCereale
WHERE C.Designation = 'Ble Paysan' AND YEAR(AP.Date) = 2024;
```

Version longue:

```
SELECT Agriculteur.idAgriculteur, Agriculteur.Nom
FROM Agriculteur
INNER JOIN Apport ON Agriculteur.idAgriculteur = Apport.idAgriculteur
INNER JOIN Céréale ON Apport.idCereale = Céréale.idCereale
WHERE Céréale.Designation = 'Ble Paysan' AND YEAR(Apport.Date) = 2024;
```

- 2) Connaitre le tonnage global (somme des apports toutes céréales confondues) apporté par chaque agriculteur entre le 01/07/2023 et le 01/08/2023.

Si l'agriculteur n'a rien apporté sur cette période le tonnage devra être égal à 0.

Colonnes à afficher : Nom agriculteur, département, tonnage

Version courte:

```
SELECT A.Nom AS 'Nom agriculteur', A.Département,
COALESCE(SUM(AP.Tonnage), 0) AS 'tonnage'
FROM Agriculteur A
LEFT JOIN Apport AP ON A.idAgriculteur = AP.idAgriculteur AND AP.Date >=
'2023-07-01' AND AP.Date < '2023-08-01'
GROUP BY A.idAgriculteur, A.Nom, A.Département;
```

Version longue:

```
SELECT Agriculteur.Nom AS 'Nom agriculteur', Agriculteur.Département,
COALESCE(SUM(Apport.Tonnage), 0) AS 'tonnage'
FROM Agriculteur
LEFT JOIN Apport ON Agriculteur.idAgriculteur = Apport.idAgriculteur AND
Apport.Date >= '2023-07-01' AND Apport.Date < '2023-08-01'
GROUP BY Agriculteur.idAgriculteur, Agriculteur.Nom,
Agriculteur.Département;
```

Les Versions courtes sont plus succinctes, mais la lisibilité peut être moins évidente avec les alias.

Algorithmique

Le but du projet est de créer un mini site web en React sur lequel on trouvera à minima :

- un listing Agriculteurs : ajout et modification de données via un formulaire (bien penser à faire une validation des données)
- un listing Céréales : ajout et modification de données via un formulaire (bien penser à faire une validation des données)
- Un formulaire simple de saisie d'Apport (Agriculteur, Céréale, Poids)
- Une suppression de données

L'utilisation de bibliothèques tierces pour la réalisation des listings et des formulaires est autorisée, mais ces éléments peuvent aussi très bien être faits "à la main" (en pur React / HTML / css).

Pas besoin de Web API, les données peuvent être stockées en mémoire.

Pas besoin de faire quelque chose de joli, le design ne sera pas jugé.

Le but du projet est de voir comment vous maîtrisez React et si les bonnes pratiques de programmation sont respectées.

Les sources du projet ainsi que les réponses aux exercices précédents pourront être transmises à Grainbow par le moyen le plus simple pour le candidat (dépôt GIT, archive ZIP...)

Un bref descriptif du projet et des bibliothèques utilisées et / ou des principes de programmation mis en place sera apprécié. |