1. Classe Employe

1.1. Annoter la classe Employe en tant qu'entité pour qu'elle puisse récupérer les données de la table Employe

1.2. Ajouter un champ id de type Long et l'annoter de manière a ce qu'il puisse gérer les identifiants générés automatiquement par MySQL.

2. Initialisation interface EmployeRepository

2.1. Créer l'interface EmployeRepository dans le package repository et la faire implémenter CrudRepository

3. Initialisation classe EmployeService

3.1. Créer la classe EmployeService dans le package service et déclarer un attribut employeRepository qui est automatiquement injecté par Spring.

3.2. Ajouter dans cette classe une méthode findById qui prend en paramètre un Long et qui renvoie l'employé correspondant à l'id passé en paramètre.

3.3. Ajouter dans cette classe une méthode countAllEmploye sans paramètre qui renvoie le nombre d'employés de la table

3.4. Ajouter une méthode creerEmploye prenant en paramètre un Employe et l'ajoutant dans la base (attention Employe est abstrait...)

3.5. Ajouter une méthode deleteEmploye prenant en paramètre un Long représentant l'identifiant technique de l'employé, et permettant de supprimer cet employé

3.6.

4. Suite interface EmployeRepository

4.1. Déclarer les méthodes permettant d'effectuer les recherches suivantes :

a) Recherche d'employés par matricule
b) Recherche d'employés par nom et prénom
c) Recherche d'employés par nom sans prendre en compte la casse
d) Recherche d'employés embauchés avant une certaine date
e) Recherche d'employés embauchés après une certaine date
f) Recherche d'employés gagnant plus de X euros et ordonnés selon leur salaire (ceux qui gagnent le plus d'abord)

4.2. Modifier EmployeRepository pour lui faire implémenter PagingAndSortingRepository et ajouter une méthode permettant de rechercher les employés en fonction de leur nom, sans prendre en compte la casse, et ce de manière paginée.

4.3. Ajouter une méthode findByNomOrPrenomAllIgnoreCase prenant en parametre un String nomOuPrenom et qui recherche sans prendre en compte la casse les employés ayant ce paramètre en nom ou en prénom. Utiliser @Param.

4.4. Ajouter une méthode findEmployePlusRiches qui récupère les employés dont le salaire est supérieur au salaire moyen des employés (voir requête SQL exo 13 du TP MySQL)

5. Héritage

5.1. Mettre en place la stratégie d'héritage ad hoc par rapport à ce qui a été mis en place lors du TP MySQL

5.2. Créer un repository BaseEmployeRepository générique permettant de factoriser les méthodes d'EmployeRepository pour qu'elles puissent être utilisées dans les repository de entités qui héritent d'Employe

5.3. Créer les Repository pour les entités restantes en les faisant hériter de BaseEmployeRepository.

6. Bonus

6.1. Dans le repository de Manager, ajouter une méthode findOneWithEquipeById prenant en paramètre un Long et retournant le Manager. Essayer de récupérer l'équipe et voir l'exception levée. Corriger la méthode en ajoutant l'annotation @EntityGraph.