TRAVAUX PRATIQUE (TP) :
ADMINISTRATION ET OPTIMISATION DES BASES DE DONNÉES

APPLICATION : Gestion Des Commandes

On souhaite créer une application qui consiste à gérer les commandes , les utilisateurs et les rôles d'un magazin.

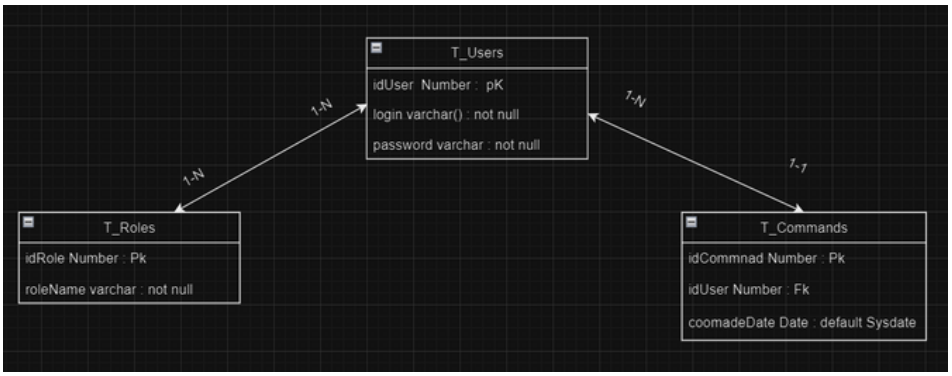
1.Configuration d'un Projet Maven avec JPA , Hibernate et Spring Data .

1. Créer un Projet Maven
2. Ajoutez les dépendances nécessaires pour utiliser Hibernate , Jpa et Spring Data avec une base de données Oracle dans le fichier **pom.xml** .
3. Créer et configurer le fichier **persistence.xml (le fichier doit etre dans cette chemin "src/main/resources/META-INF")** pour utiliser JPA avec une base de données Oracle**(NB: il faut utiliser l'ORM Hibernate)** .

2. Création des tables et l'insertion de données

1. Transformer cette schéma entité-association (LMD) au Model relationnel
2. Insérer 3 enregistrements dans chaque table

NB: pour les relations N-N il vaut mieux d'ajouter un tabled'association

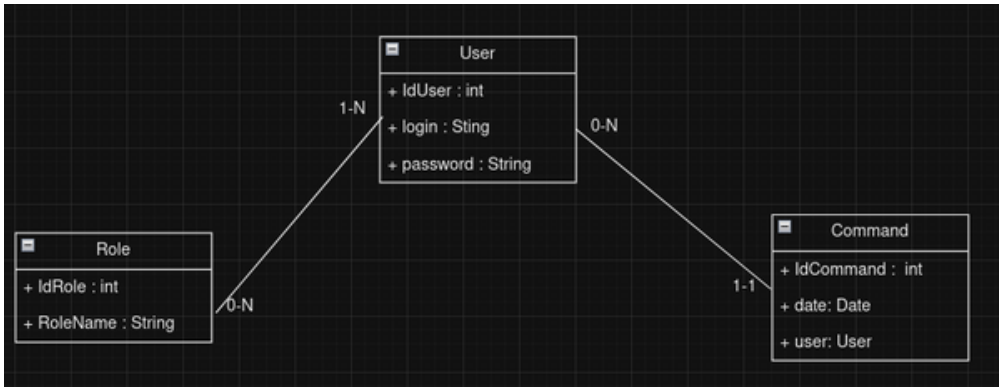


Premier Partie : JPA et Hibernate

Ici on dispose de trois classes :

- Chaque utilisateur est défini par un identifiant unique , son nom de connexion généralement appelé login et son password et peut posséder plusieurs rôles
- Un Rôle est caractérisé par un Id unique , le nom du rôle **(ici pour l'application on a trois rôles possibles : stockmanager , admin et client)**
- Chaque commande est caractérisé par un identifiant unique , la date de livraison de la commande et l'utilisateur auquel est associé la commande

Diagramme de classes :



- ## Questions :

1. Créer dans le dossier **"src/main/java"** dans le projet maven les classes nécessaires .
2. Créer les entités JPA pour chaque classe en utilisant la méthode de mapping de votre choix . **(Pour le choix de mapping avec le fichier xml nommé le fichier de mapping comme suit : <nom_classehbm.xml>)**
3. créer une interfaces qui déclare les opérations CRUDS **(CREATE - READ - UPDATE - DELETE - SEARCH) .**
   NB: Il faut utiliser une interface générique
4. créer les classes nécessaires qui implémente les méthodes de l'interface générique qu'il est déjà crée dans la question précédent .

5- pour la class qui concerne la gestion des Users il faut ajouter les méthodes suivant :

- **findByLogin(String login)**
- **findUsersByRole(Role role)**
- **assignRoleToUser(User user, Role role)**
- **findAll()**

6- pour la classe qui concerne la gestion des Roles il faut ajouter les méthodes suivant :

- **findByRoleName(String roleName)**
- **findRolesByUser(User user)**
- **findAll()**

7- pour la classe qui concerne la gestion des Commandes il faut ajouter les méthodes suivant :

- **findCommandsByUser(User user)**
- **findCommandsBetweenDates(Date startDate, Date endDate)**
- **findAll()**

8- Faire une classe de Test de tout les méthodes et vérifier l'insertion des données dans la base de données .

Deuxième Partie : Spring Data et JPA Hibernate

Recréer un projet maven pour cette deuxième partie .

1- Créer le fichier pom.xml recopier la configuration de la partie précedente puis ajouter les dépendences pour utiliser Spring Data

2- Créer le fichier **application.properties** dans le repertoire **"src/main/resources"** configurer le connection pool pour se connecter à la base de donées oracle

3- Créer les entités JPA pour chaque

4- Implémenter l'interface JpaRepository de Spring Data pour chaque entité JPA en créant une classe pour chaque entité JPA implémetant cette interface. Ces classes doivent se situer dans le même package que les entités JPA **"src/main/java"**

5- Créer une classe ApplicationTests.java sous le repertoire **src/test/java** pour effectuer les tests unitaires

6- Créer la classe **Application.java** sous **src/main/java** qui sert de point d'entré pour lancer notre application insérer quelques éléments à travers l'application et lister tous les utilisateurs et commandes pour tester la connectivité avec la base de données

7- Ajouter des méthodes suivantes dans la classe implémentant l'interface JpaRepository pour l'entité User :

- **findUserByLogin()**
- **findUserByRole()**

8-Tester ces méthodes dans l'application dans la classe **Application.java**

BONNE COURAGE