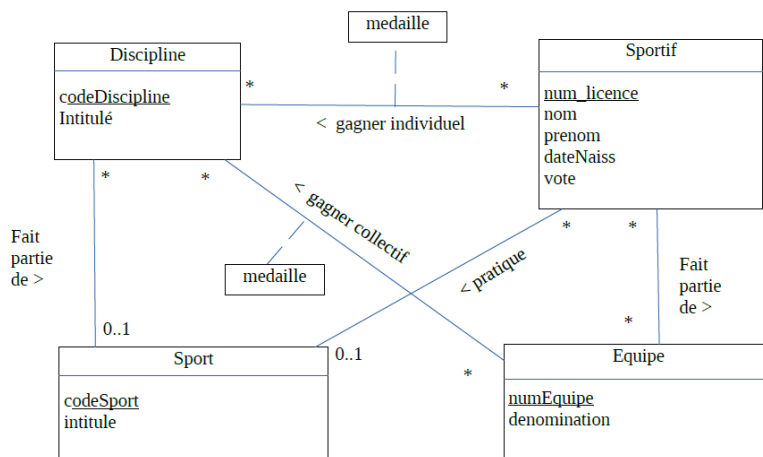


Nous allons utiliser dans ce TP une base de données déjà construite sur les Jeux Olympiques de Londres 2012. Voici le schéma relationnel utilisé. Cette base de données sera hébergée sur votre machine locale en utilisant **docker**

Voici son diagramme de classes :



et son schéma relationnel :

Sport (codeSport, intitule)

Discipline (codeDiscipline, intitulé, # code_Sport)

Sportif (num_licence, nom, prenom, date_naiss, vote, # code_sport)

Equipe (numEquipe, denomination)

gagner_individuel (#code_discipline, # num_licence, medaille)

appartenir_equipe (# num_licence, # num_equipe)

gagner_collectif (#code_discipline, # num_equipe, medaille)

Etape 1 : accès à mysql

Pour utiliser **mysql**, nous allons utiliser docker pour créer un serveur local **mysql**.

Utilisation de docker en local

- Nous allons utiliser docker sur la machine locale. Pour cela créer un répertoire **docker** et à l'intérieur deux répertoires **www** et **mysql**.
- Copier dans le répertoire **docker** le fichier de configuration **docker-compose.yml** disponible sur moodle. Ce fichier configure 3 conteneurs docker : 1 pour un serveur web, un pour **mysql** et un pour l'outil d'administration **phpmyadmin**.
- Pour lancer les conteneurs docker, se placer dans le répertoire **docker** et taper : **docker compose up -d**
- Sur votre navigateur local, vous pouvez ouvrir **phpmyadmin** grâce au lien **localhost:10001** (login=root, password=licinfo2020 : ces informations sont dans le fichier de configuration **docker-compose.yml**)
- Créez alors une base de données nommée **jo**, puis vous allez importer les tables de la base de données **jo** dans votre base de données. (Pour cela dans **PhpMyAdmin**, se positionner sur la base données **jo**, puis sélectionner l'onglet **importer**. Choisissez alors le fichier **jo.sql** disponible sur moodle) Vous travaillerez désormais sur cette base de données.
- Pour accéder à **mysql** en java il faudra utiliser le port **10002**

Etape 2 : premier programme java

Nous allons dans cette partie accéder en **JAVA** à la base de données précédemment créée sur **localhost**

- Récupérer sur moodle le premier programme **tp3-etape2.java** qui est un programme qui propose un menu permettant d'accéder aux données de la base de données. Créez un projet java eclipse pour y mettre le fichier java.
- Pour fonctionner, le programme doit disposer d'un connecteur **JDBC**. Celui-ci est disponible sur moodle, télécharger-le. Il faut ensuite l'installer dans eclipse :
 - clic droit sur le projet, choisir **Properties**
 - Dans **Java Build Path** choisir **Libraries**
 - Puis **Add External Jars**
- Tester alors le programme :
 - vérifier qu'on accède bien à la liste de tous les sports
 - vérifier que l'on peut ajouter un sport.
- Vérifier alors qu'on ne peut pas ajouter un sport dont le **CODE_SPORT** existe déjà
- Pour gérer ce problème, le mieux est que le **CODE_SPORT** soit incrémenté automatiquement. Modifier alors la base de données pour que l'attribut **CODE_SPORT** soit incrémenté automatiquement. Pour faire cela, il faut ajouter à l'attribut la propriété **AutoIncrement (AI)**, accessible dans l'onglet **structure**.

Lorsque l'on souhaite ajouter l'AutoIncrement, phpMyadmin refuse car il y a des clés étrangères associées à cet attribut : pour régler le problème il faut supprimer les clés étrangères incriminées (pour cela dans l'onglet Structure, choisir la Vue Relationnelle et supprimer la contrainte de clé étrangère), ajouter l'AutoIncrement, puis recréer les contraintes de clés étrangères.

Modifier alors le programme java pour ne plus avoir à demander le `CODE_SPORT` du sport à ajouter.

Etape 3 : ajout de fonctionnalités

Ajouter maintenant les fonctionnalités suivantes au programme java :

- Permettre de récupérer la liste des sportifs d'un sport donnée (noms et prénoms)
- Afficher la liste des sportifs qui ont gagné une médaille d'or en individuel et afficher aussi dans quelle discipline
- Afficher la liste des équipes présentes et pour chacune le nom des joueurs qui la composent
- Permettre de supprimer un sportif donné par son nom. **Attention** : si on supprime un sportif, il faut d'abord le supprimer des équipes auxquelles il appartient puis de la table `GAGNER_INDIVIDUEL`. Pour cela, soit on le fait depuis le programme java, soit, et c'est le mieux, on le fait dans la base données en choisissant l'option `DELETE ON CASCADE` pour la clé étrangère de `GAGNER_INDIVIDUEL`. Pour cela sélectionner la table `GAGNER_INDIVIDUEL`, puis dans l'onglet Structure choisir la vue relationnelle. Le même problème se pose pour `GAGNER_COLLECTIF` et `APPARTENIR_EQUIPE`

Etape 4 : interface graphique java

Nous allons maintenant utiliser une interface graphique pour manipuler les données.

- Récupérer sur moodle le programme `tp3-etape4.java` qui est un programme qui propose une interface graphique permettant de manipuler des données de la base de données.
- Tester le programme :
 - vérifier que l'on peut modifier l'intitulé d'un sport
 - vérifier que l'on peut ajouter un sport (Il faudra ajouter la fonctionnalité d'auto incrément pour le `CODE_SPORT`)

Etape 5 : ajout de fonctionnalités

Ajouter maintenant les fonctionnalités suivantes :

- Faire les modifications nécessaires pour qu'il ne soit pas possible d'ajouter un sport dont l'intitulé existe déjà
- Faire les modifications nécessaires pour qu'un intitulé de sport commence par une majuscule
- Ajouter un bouton Supprimer qui permet de supprimer le sport correspondant à la ligne sélectionnée
- Lorsque l'on double clique sur un sport une fenêtre s'ouvre contenant la liste des sportifs de ce sport