

## Interrogation écrite 2 – Mai 2023 – BD-L1 SF – Algèbre relationnelle

Seule une feuille A4 de notes manuscrites personnelles est autorisée. Tout moyen de communication est interdit.

### Introduction

On considère une base de données gérant un ensemble d'épreuves de course à pied. Un modèle relationnel de celle-ci est représenté à la figure 1. Vous avez déjà vu cette base en TD. Dans cette base de données, on suppose que toutes les clés identifiées par `XXX_ID` sont des entiers qui sont auto incrémentés.

- La table `TYPE_EPREUVE` (`TY_ID`) contient les données sur les différents types d'épreuves ("Semi marathon", par exemple).
- La table `CHAMPIONNAT` (`CH_ID`) contient les données sur les championnats.
- La table `MANIFESTATION` (`MA_ID`) contient les données sur les manifestations servant pour un certain championnat.
- La table `EPREUVE` (`EP_ID`) contient les données sur les épreuves proposées. Chaque épreuve a un type donnée, et fait partie d'une manifestation.
- La table `CLUB` (`CL_ID`) contient les données sur les clubs de course à pied.
- La table `COUREUR` (`CO_ID`) contient les données sur les coureur-euse-s.
- La table `ADHESION` (`AD_COUREUR_FK`, `AD_ANNEE`) contient les données d'adhésion des coureur-euse-s aux clubs.
- La table `CATEGORIE_AGE` (`CA_ID`) contient les données sur les différentes catégories d'âges des coureur-euse-s.
- La table `INSCRIPTION` (`IN_ID`) contient les données d'inscription des des coureur-euse-s à des épreuves, dans une certaine catégorie d'âge.

### Exercice 1 – Requêtes d'extraction d'information

Écrire les requêtes SQL permettant d'obtenir les éléments suivants.

**Question 1.1.** Liste de tous les "Semi marathon" (date, nom de la manifestation) qui ont eu lieu en 2016, classé par ordre chronologique. Vous pouvez utiliser la fonction `YEAR(date)` pour extraire l'année de la donnée `date` passée en paramètre.

---

---

---

---

---

---

---

**Question 1.2.** Pour chaque catégorie d'âge (afficher son identifiant et son libellé) et chaque année, meilleur temps sur un semi-marathon. On utilisera les fonctions `TIME_TO_SEC` et `SEC_TO_TIME` qui respectivement convertissent un temps en secondes et des secondes en temps.

---

---

---

---

Nom:

Prénom:

---

---

---

---

---

**Question 1.3.** Nom des manifestations et année où il y avait plus de 1000 inscrit-e-s, toutes épreuves confondues à la manifestation.

---

---

---

---

---

---

---

**Question 1.4.** Identifiant, des coureur-euse-s qui se sont inscrit-e-s à des épreuves mais qui n'en ont jamais terminée une.

---

---

---

---

---

**Question 1.5.** Identifiant, nom et prénom des coureur-euse-s qui se sont inscrit-e-s à des épreuves où Dorine Tabary figurait parmi la liste des personnes inscrites.

---

---

---

---

---

---

---

---

**Question 1.6.** Identifiant des coureur-euse-s qui comptabilisent le plus d'inscriptions à des épreuves.

---

---

---

---

---

---

---

---

---

**Question 1.7.** Inventer une requête intéressante d'extraction d'information et donner le code SQL associé, qui doit contenir une sous-requête synchronisée ou une division. Cette requête ne sera pas un copier-coller d'une requête vue en TD.

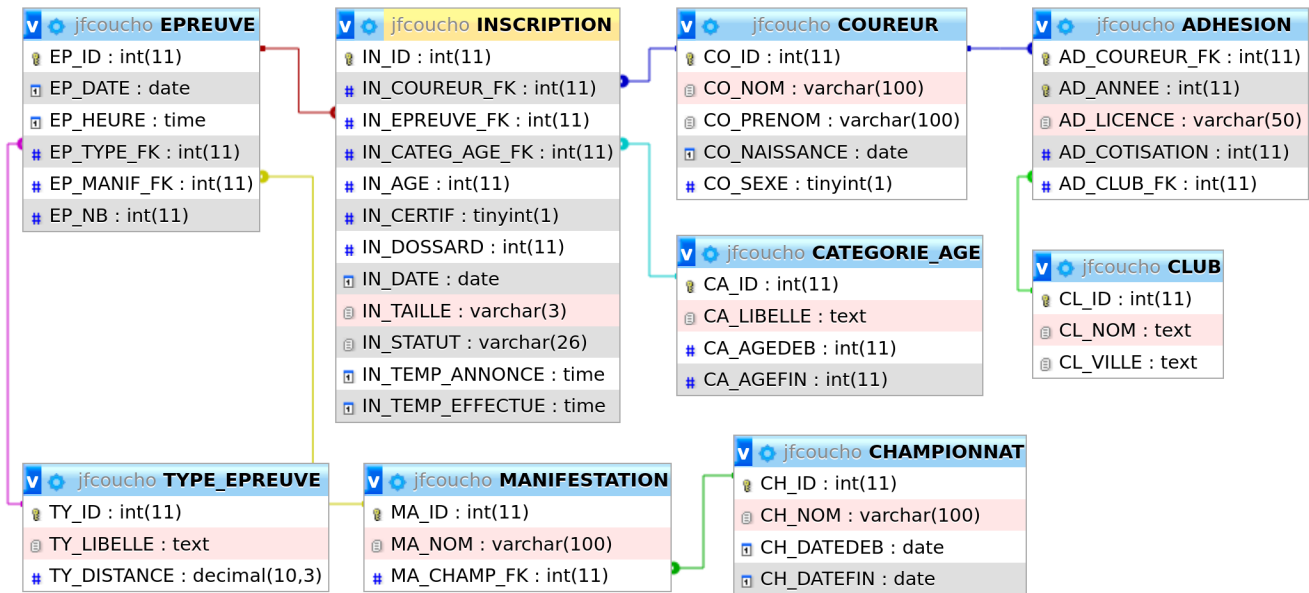


FIGURE 1 – Modèle relationnel

## Exercice 2 – Requêtes d’ajout de données

Écrire les requêtes SQL permettant d’ajouter dans la bases les éléments suivants.

**Question 2.1.** Catégorie “Master 9” des athlètes entre 80 ans et 84 ans.

**Question 2.2.** L’épreuve de type “60km” (qui n’est pas dans la base) de la manifestation “Trail des Forts” (qui n’est pas dans la base) dont le départ a eu lieu le samedi 13 mai 2023 à 8h (’08 :00 :00’) et qui était limitée à 1000 inscrite-s. Cette manifestation faisait partie du championnat des “courses nature 2023” (qui est déjà dans la base) dont l’identifiant CH\_ID est 3.

---



---



---



---



---



---

## Exercice 3 – Normalisation

On considère une version simplifiée de la base de donnée présentée plus haut. Celle ci est initialement codée sous la forme de la relation universelle suivante :

U (EP\_ID, EP\_DATE, EP\_TYPE, EP\_LIBELLE, CO\_ID, CO\_NOM, CO\_PRENOM, CO\_NAISS, CO\_SEXE, DOSSARD, TEMPS\_EFFECTUE, CA\_AGE, CA\_LIBELLE, CA\_AGEDEB, CA\_AGEFIN)

On a les dépendances fonctionnelle suivantes :

- $DF_1$  : EP\_ID  $\rightarrow$  EP\_DATE, EP\_TYPE, EP\_LIBELLE ;
- $DF_2$  : CO\_ID  $\rightarrow$  CO\_NOM, CO\_PRENOM, CO\_NAISS, CO\_SEXE, CA\_AGE ;
- $DF_3$  : CA\_AGE  $\rightarrow$  CA\_LIBELLE, CA\_AGEDEB, CA\_AGEFIN ;
- $DF_4$  : EP\_ID, DOSSARD  $\rightarrow$  CO\_ID, TEMPS\_EFFECTUE.

**Question 3.1.** Expliquer chaque dépendance fonctionnelle.

- $DF_1$  : EP\_ID  $\rightarrow$  EP\_DATE, EP\_TYPE, EP\_LIBELLE :

