Base de données Tennis – TP 4 – SQL

Dans ce TP nous nous intéressons à une BD sur le tennis. Nous utiliserons ici les tables suivantes (les clés primaires sont soulignées) :

Joueurs(nom, prenom, age, nationalite)
Rencontre(nomgagnant, nomperdant, lieutournoi, annee, score)
Gain(nomjoueur, nomsponsor, lieutournoi, annee, rang, prime)
Sponsor(nom, lieutournoi, annee, adresse, montant)

… et les contraintes de clé étrangère :

```
Rencontre(nomgagant) REFERENCES Joueurs(nom)
Rencontre(nomperdant) REFERENCES Joueurs(nom)
Gain(nomjoueur) REFERENCES Joueur(nom)
Gain(nomsponsor, lieutournoi, annee) REFERENCES Sponsor(nom,
lieutournoi, annee)
```

1. Création de tables

Dans l'interface phpMyAdmin de votre SGBD MySQL, il existe deux manières de créer des tables : l'une est graphique, l'autre utilise des instructions SQL. La manière graphique est plus simple, mais elle ne permet pas de gérer beaucoup de contraintes. En particulier, elle ne permet pas de gérer les contraintes de clé étrangères, ce qui est trop limitatif.

Pour créer une table de manière graphique, il faut entrer le nom de la table, puis donner le nombre de champs. Il faut commencer en créant les tables qui n'ont pas de clés étrangères.

Question 1 : Donnez un ordre correct pour la création des tables.

Commençons à créer la table **Joueurs**. ATTENTION ! Les majuscules et minuscules sont importantes. Le logiciel fait la différence entre **Joueurs** et **joueurs**.

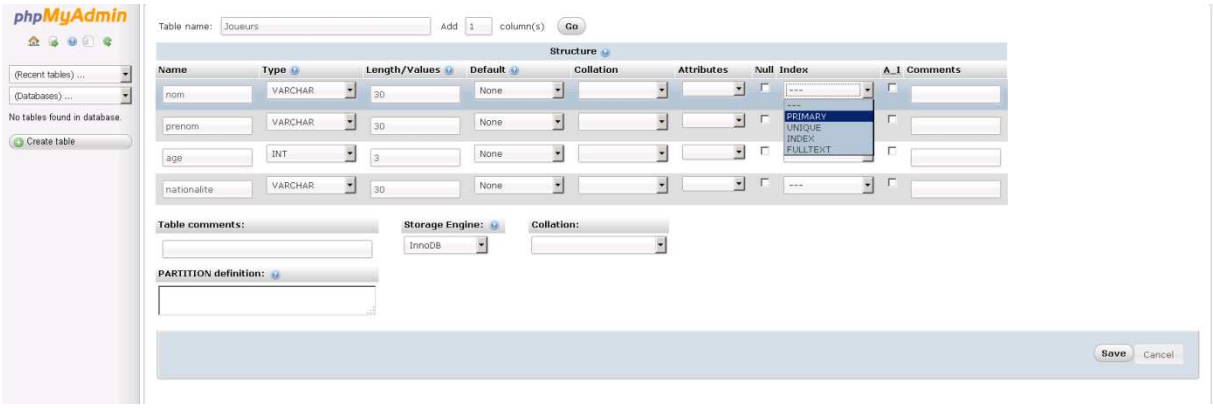

Figure 1 : Formulaire de création de tables dans phpMyAdmin

Il ne faut remplir que les champs : *Champ (Name), Type, Taille (Length)* et *Clé (Index)*. La clé se trouve à la droite de l'écran (comme montré sur la Figure 1). Il faut cocher/sélectionner les lignes des

attributs qui participent à la clé primaire (ici seulement l'attribut **nom**). Il est également très important de choisir comme type de la table la valeur *INNODB*. Notez que vous pouvez également indiquer la clé primaire après la création de la table dans la page affichant les métadonnées de la table (voir la Figure 2).

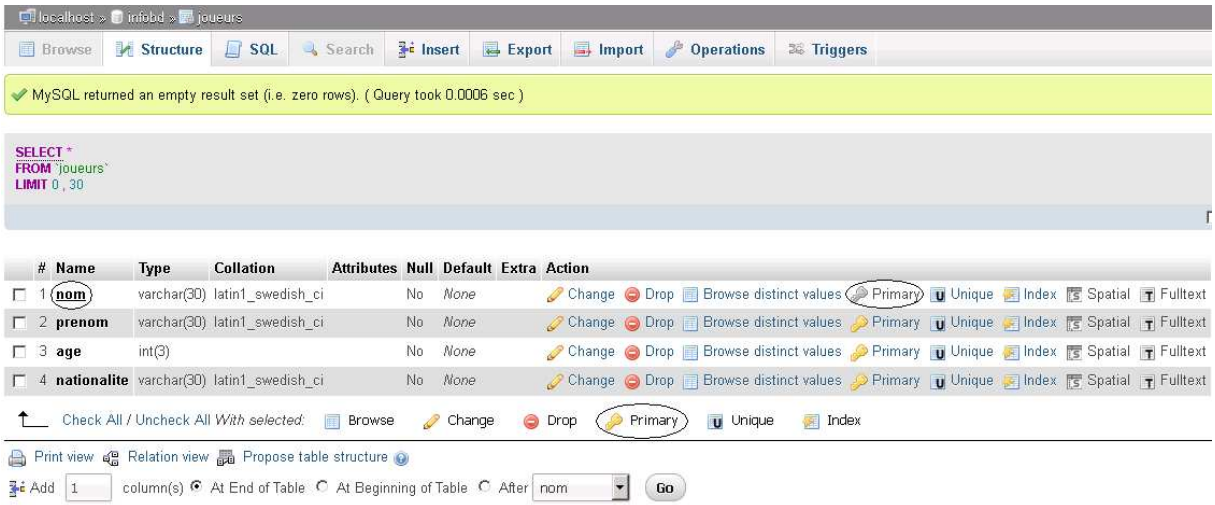

Figure 2 : Affichage des métadonnées de la table Joueurs

Question 2 : Créez les tables **Joueur** et **Sponsor** en utilisant l'interface graphique.

Selon la version de phpMyAdmin que vous avez installé, vous remarquerez qu'une requête SQL est générée lorsque vous créez une table (en appuyant sur le bouton **sauvegarder**, de la forme *CREATE TABLE* … Conservez cette requête dans un fichier texte pour une utilisation ultérieure :

```
CREATE TABLE Joueurs (
nom VARCHAR(30) NOT NULL,
prenom VARCHAR(30) NOT NULL,
age INT(3) NOT NULL,
nationalite VARCHAR(30) NOT NULL,
PRIMARY KEY (nom)
);
```

Les guillemets encadrant les noms des attributs sont générés automatiquement. Nous déconseillons d'une manière générale leur utilisation lorsque vous écrivez vous-même le code SQL. De même, il est fortement déconseillé d'utiliser des caractères accentués (é, è, ô, û, …) pour les noms des tables ou des attributs. En utilisant la fenêtre SQL, nous allons écrire le code SQL de création de tables nécessitant une ou plusieurs clés étrangères. Pour ce faire, nous écrivons une requête similaire à celle qu'on vient de générer, en ajoutant les lignes de clé étrangères :

```
CREATE TABLE Rencontre(
nomgagnant varchar(30),
nomperdant varchar(30),
lieutournoi varchar(30),
annee int(3),
score varchar(15),
PRIMARY KEY (nomgagnant, nomperdant, lieutournoi, annee),
FOREIGN KEY (nomgagnant) REFERENCES Joueurs (nom),
```

```
FOREIGN KEY (nomperdant) REFERENCES Joueurs (nom)
);
```

Question 3 : Créez les tables restantes.

2. Insertion de données

En cliquant sur le nom d'une table, puis sur *insérer*, on arrive à un formulaire permettant de remplir une table avec des données.

Il suffit de ne remplir que le premier formulaire comme montré dans la Figure 3. Choisir « exécuter » pour rajouter cette entrée à la relation. Il faut bien se rappeler les contraintes qui existent sur une table lorsque vous faites des insertions : comme *nom* est la clé primaire de la table **Joueurs**, il sera impossible d'entrer deux fois un joueur dont le nom serait « Federer ». De même, il est impossible de créer une rencontre entre deux joueurs, si on ne les a pas préalablement insérés dans la table **Joueurs**. De même pour la table **Gain**, il faut d'abord préciser le sponsor…

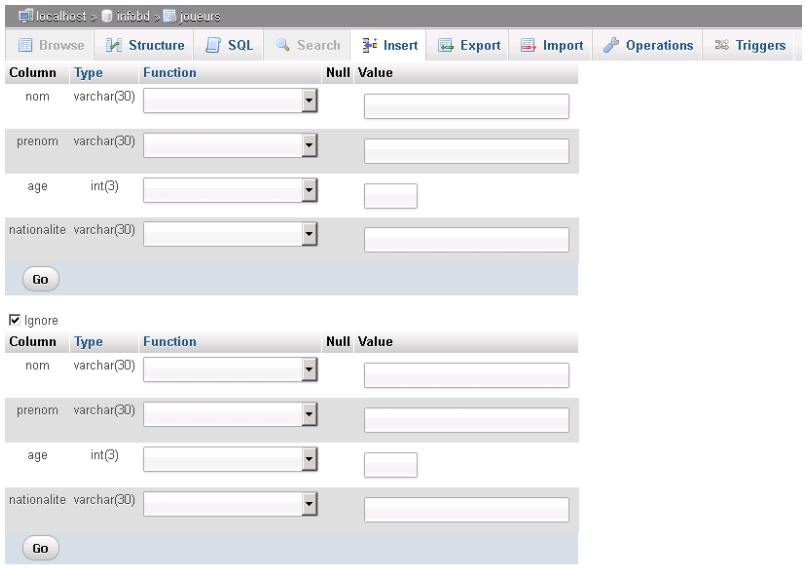

Figure 3 : Formulaire d'insertion de données

Question 4 : Ajoutes les données suivantes dans la BD en utilisant le formulaire d'insertion et des requêtes SQL (voir l'exemple ci-dessous).

INSERT INTO Joueurs VALUES ('Federer', 'Roger', 34, 'suisse');

Ajoutez les joueurs Federer, Nadal, Djokovic, Murray, Soederling, Berdych.

Ajoutez le sponsor Peugeot pour le tournoi de Roland-Garros, Wilson pour le tournoi de Flushing-Meadows et IBM pour le tournoi Open d'Australie.

Ajoutez les rencontres suivantes en 2011 :

- Australie : Djokovic bat Murray 6/4 – 6/2 – 6/3
- Roland-Garros : Nadal bat Federer 7/5 – 7/6 – 5/7– 6/1
- Wimbledon : Djokovic bat Nadal 6/4 – 6/1 – 1/6 – 6/3

- Flushing Meadows : Djokovic bat Nadal 6/2 – 6/4 – 6/7– 6/1

Ajoutez les rencontres suivantes en 2010 :

- Australie : Federer bat Murray 6/3 – 6/4 – 7/6
- Roland-Garros : Nadal bat Soederling 6/4 – 6/2 – 6/4
- Wimbledon : Nadal bat Berdych 6/3 – 7/5 – 6/4
- Flushing Meadows : Nadal bat Djokovic 6/4– 5/7 – 6/4 – 6/2

Le gagnant remporte 1.000.000 €, le perdant 500.000 €.

3. Requêtes d'interrogation

On peut poser des requêtes SQL en utilisant la fenêtre SQL. Il suffit de taper le code de la requête SELECT…FROM…WHERE… et de l'exécuter.

Question 5 : Réaliser les requêtes suivantes, et les écrire en syntaxe SQL, et les exécuter.

Requêtes simples :

- Afficher le nom et prénom des joueurs dont le prénom est Roger
- Afficher les années où s'est déroulé Roland Garros (indication : un tournoi s'est déroulé s'il y a des rencontres dans la base)
- Afficher le nom et l'âge des joueurs ayant gagné à Roland Garros, peu importe l'année.
- Afficher le nom des sponsors ayant sponsorisé un tournoi ayant lieu en France.
- Afficher le nom et prénom des joueurs ayant gagné un tournoi sponsorisé par BNP-Paribas.

Requêtes ensemblistes ou imbriquées :

- Afficher le nom de tous les joueurs ayant joué au moins un match.
- Afficher le nom des joueurs ayant joué à Wimbledon quel que soit l'année.
- Afficher le nom des joueurs ayant gagné tous leurs matchs.
- Afficher le nom des joueurs n'ayant jamais gagné moins de 1.000.000 par tournoi.
- Comptez le nombre total de matchs joués à l'US Open et affichez le résultat dans une colonne nommée nbmatchs.
- Calculer les gains totaux de Nadal dans une colonne gaintotal
- Afficher les noms et les gains (dans une colonne gaintotal) des joueurs ayant gagné au moins 2.000.000 au total.
- Affichez les noms et années des tournois où la somme versée par les sponsors est supérieure aux gains reversés aux participants.

4. Modification du schéma et du contenu de la BD

a) Ajouter/supprimer un attribut à une relation

Ajouter un nouvel attribut dans la table **Joueurs** pour stocker la taille de chaque joueur et un autre attribut dans la table Rencontre pour stocker le nom du stade de la rencontre. Utiliser une requête SQL ALTER TABLE :

```
ALTER TABLE Joueurs
ADD COLUMN taille int (10) ;
```

b) Modifier le type des attributs

Il vous est demandé de modifier les types précédemment définis, par exemple « tailles » devient int(3). Peut-on changer un type en diminuant sa taille ?

```
ALTER TABLE Joueurs
MODIFY COLUMN taille int (3) ;
```

c) Problème des valeurs nulles : not null / null

Un attribut d'une table déclaré "not null" doit nécessairement être renseigné lors de l'insertion d'un tuple. Il vous est demandé d'identifier les attributs répondant à ce critère et de modifier votre schéma en conséquence. Utiliser la commande ALTER TABLE.

d) Définitions des clés

Supprimez la contrainte de clé primaire de la table **Gain** et redéfinissez la clé primaire au travers d'une requête SQL :

```
ALTER TABLE Gain
DROP PRIMARY KEY;
```

```
ALTER TABLE Gain
ADD PRIMARY KEY (nomjoueur, lieutournoi, annee);
```

Faites de même avec la clé étrangère de la table **Gain**.

e) Mise à jour des relations

Il vous est demandé de :

- mettre en majuscule les noms des sponsors (fonction UPPER) ;
- mettre en minuscule les noms des tournois qui se sont déroulés avant 2011 (fonction LOWER) ;
- ajouter 1 an de plus à l'âge de tous les joueurs.

f) Suppression de données

Il vous est demandé de :

- supprimer les rencontres de Federer au tournoi d'Australie en 2010.

5. Création de vues et requêtes sur les vues

Créez les vues suivantes :

- « Joueurs_français » contenant le nom, prénom et l'age des joueurs français. Assurez-vous qu'il y en au moins deux joueurs français dans la BD.
- « Matchs_perdus/Match_gagnés » contenant les nom et prénom du perdant/gagnant, ainsi que le lieu et l'année du tournoi, le score et les nom et prénom de l'opposant.
- « Total_gain_joueurs » contenant les nom et prénom des joueurs et la somme totale de leurs gains.

Répondez aux requêtes suivantes en utilisant les vues définies au-dessus (et les autres tables de la BD quand cela s'avère nécessaire) :

- Affichez les nom, prénom, et lieu tournoi des joueurs français ayant gagné des matchs en 2011.
- Trouvez les matchs perdus de Nadal en 2010. Affichez le lieu du tournoi et le score.
- Affichez le montant total des gains des meilleurs joueurs (i.e., ayant gagné en tout plus de 1000000).
- Affichez les matchs perdus par les joueurs français ayant des gains dans leur carrier totalisant 1000000.

6. Définition des droits d'accès

En étant connecté en tant que « root » (c.à.d., administrateur de la BD), créez des nouveaux comptes d'utilisateur pour Alice et Bob. Utilisez des requêtes CREATE USER :

```
CREATE USER 'nom_utilisateur'@'hote' identified by 'mot_de_passe';
```

où 'hote' c'est l'identifient de la machine hôte de connexion de l'utilisateur (ex., localhost, adresse IP de la machine ou '%' pour accepter toutes les adresses d'hôte).

Créez pour chaque nouvel utilisateur une nouvelle base de donnes (ex., bd_alice et bd_bob). Ensuite donnez tous les droits d'accès à Alice et à Bob à leur base de données respective :

```
GRANT ALL ON nom_bd.* TO 'nom_utilisateur'@'hote';
```

```
FLUSH PRIVILEGES;
```

Vérifiez les droits accordés à Alice et Bob :

```
SHOW GRANTS FOR nom_utilisateur;
```

Fermez la connexion root et reconnectez-vous en tant qu'Alice. Vérifiez qu'Alice peut accéder à la bd_alice, qu'elle peut créer des tables dans sa BD et qu'elle peut modifier le contenu de ces tables (insérer, supprimer, mettre à jour). Faites de même pour Bob. Vérifiez si Bob peut accéder aux tables d'Alice ou inversement, vérifiez si Alice peut accéder aux tables de Bob.

Alice veut partager des tables avec Bob. En tant qu'Alice, créez deux tables : « pour_select_bob » et « pour_update_bob » et donnez le droit de SELECT et d' INSERT/DELETE/UPDATE à Bob sur ces tables

respectivement. Connectez-vous en tant que Bob et vérifiez que ces droits d'accès sont bien pris en compte. Essayez aussi d'insérer dans la table « pour_select_bob » et de visualiser le contenu de la table « pour_update_bob ».

Droits transmissibles

Créez un nouvel utilisateur Marvin.

Alice donne à Bob le droit SELECT sur « pour_select_bob » et d' INSERT/DELETE/UPDATE sur « pour_update_bob », mais aussi la permission de transmettre ces droits à d'autres utilisateurs (WITH GRANT OPTION). Bob transmet ces droits à Marvin.

Testez si Marvin peut en effet visualiser la table « pour_select_bob » et modifier la table « pour_update_bob ».

Alice retire les privilèges qu'elle a donnés à Bob. Pour s'assurer que personne d'autre n'aura plus le droit de modifier la table « pour_update_bob », Alice retire ces droits « en cascade ». En revanche, pour le droit SELECT de la table « pour_select_bob », elle ne retire ce droit qu'à Bob. Vérifiez l'impact de ces révocations de droits sur Marvin.