

## **INTRODUCTION:**

Le projet de bases de données nous permet de mettre en pratique les cours que l'on a vu en amphithéâtre, et nous pousse à faire des recherches (Manuel PostgreSQL), ce qui nous aide à améliorer nos compétences et acquérir des connaissances qui nous seront utiles dans ce domaine.

Quand on a commencé à travailler sur le projet, le premier et le plus grand problème que nous avons rencontré était de créer la meilleure modélisation possible de tables, et pour cela on a pensé à revoir les modélisations données dans nos TPs de bases de données pour s'en inspirer.

## **MODELISATION:**

À partir de là, on a fait les tables qui nous semblaient nécessaires, ces tables étant:

- Sports, qui contient tous les sports (individuels ou collectifs)
- Pays, qui contient tous les pays participants aux jeux
- Équipes, qui contient la liste de toutes les équipes
- Athlètes, qui contient la liste de tous les athlètes
- Épreuves, qui contient la liste des épreuves
- Médailles, qui contient la liste des médailles
- Matches, qui contient la liste des rencontres
- Résultats, qui contient les scores et temps obtenus par les athlètes ou équipes

On a donc fait une première modélisation qui comportait 6 tables qui nous paraissaient suffisantes, mais après avoir vu les requêtes que l'on doit réaliser, on s'est rendu compte qu'on ne pourra pas toutes les faire avec cette modélisation car on avait besoin d'attributs supplémentaire.

On a donc commencé à améliorer notre modélisation au fur et à mesure en fonction des requêtes que l'on doit implémenter.

La réalisation de nos tables se base beaucoup sur les clés étrangères, chaque table possède une clé primaire, et cette clé est utilisée pour identifier les différents éléments de la table, et est référencée dans les autres tables qui en ont besoin, par exemple, notre table «Athlètes» possède un attribut «nationalité» qui référence la clé primaire d'un élément de la table «Pays».

Cette modélisation nous permet de différencier des éléments qui ont exactement les mêmes attributs, par exemple, deux athlètes ayant le même nom, prénom, sexe, nationalité et âge peuvent coexister dans notre table et seront différenciés par leurs clés primaires.

Pour éviter les inconvénients de Ubuntu PostgreSQL, nous n'avons pas mis d'accents dans les éléments de nos tables pour éviter une erreur de serveur (Serveur SQL étant en ASCII au lieu de UTF-8).

## **REQUÊTES:**

Lors de la réalisation des requêtes, celle de difficulté une étoile nous ont posé aucun problème, car les clé étrangères que nous avons implémenter ont facilité la réalisation de ces requêtes, malgré que certaine requêtes nécessitait d'appeller plusieurs relation, ce qui rendait le code un peu long pour une requêtes assez simple(voir 2ème requetes de difficulté 1 une étoile).

Les requêtes qui nous ont vraiment posé des problèmes sont la requêtes 2 difficulté 2 étoiles, la requête 3 difficulté 2 étoiles, et la requête 4 difficulté 2 étoiles.

La requête 2 difficulté 2 étoiles, pour une raison que l'on ignore, compte les médailles 12 fois au lieu d'une fois (les pays qui ont une seul médailles en montrait 12), ce qui nous a laissé perplexe car la requête que nous avons écrite seulement sensé utiliser COUNT une seul fois. Pour remédier à ce soucis nous avons simplement ajouter «/ 12» après le COUNT du Select, ce qui donne bien les bons résultats.

La requête 3 difficulté 2 étoiles marche correctement, mais nous n'avons pas réussi à afficher cette requetes en tableau de 5 attributs comme demandé.

La requête 4 difficulté 2 étoile marche aussi correctement, mais il y a trop d'athlètes qui n'ont pas de médaille dans la base de données, ce qui provoque un problème qui fait que

l'on doit défiler les athlètes sans médailles (49 rows), ce qui bloque l'affichage des autres requêtes. Pour remédier à cela, on appuis sur la touche «Q» du clavier, afin de sauter cette requêtes, pour afficher les autres. Ceci est en soit pas un énorme problème mais plutôt un souci d'esthétique, nous pensons qu'il y a sûrement un autre raccourci pour continuer l'affichage des requêtes mais nous le connaissons pas.

Les autres requêtes plus compliqués que les requêtes cité nous ont posés quelque difficultés mais elles ont vite était remédier car nous avons trouvé des fonctions dans le manuel PostgreSQL qui nous ont permit de réaliser ces requêtes.

## **SECTION 4 DU PROJET:**

Afin de calculé l'indice prédictif de remplissage, nous avons créer une table supplémentaire «Lieux», avec les attributs:

-lid, clé primaire

-nom, nom du lieu

-indice, étant l'indice du nombre volontaire nécessaire pour chaque endroit

Et avons aussi rajouté un attribut «lieu» dans la table match, étant une clé étrangère, qui référence la clé primaire de lieu.

L'indice est calculé en fonction de taille du lieu de l'épreuve, et de combien d'épreuves auront lieu au même endroit et le même jour, par exemple un Marathon aura besoin de plus de volontaire qu'un terrain de Handball, et les épreuves d'athlétisme auront toutes lieu sur le même stade, peut-être le même jour mais pas en même temps, ce qui permet d'avoir le même groupe de volontaire pour plusieurs épreuves qui ont lieu le même jour.

Nous avons choisi cette méthode car son utilisation est simple et n'a pas nécessité une forte modification de notre base de donnée.

*-Points Fort:*

Cette méthode est très simple d'utilisation, ce qui permet à d'autre personne de pouvoir manipuler la base sans trop de difficultés.

*-Limites:*

Cette méthode ne permet pas de prévoir l'absence de volontaires, ce qui peut s'avérer être un problème.

Une solution pour améliorer cet indice serait de prendre une rangée de nombre, à la place d'un nombre définis, par exemple, avoir 10-15, au lieu d'avoir 10 en tant qu'attribut.

Athletes (aid, prenom, nom, naissance, nationalite, sexe, domaine, equipe)

Epreuves (epid, nom, type, sexe)

Medailles (mid, type, epreuve, gagnant, equipe gagnante)

Matches (maid, nom, epreuves, dates, tour)

Resultats (paid, match, participant, equipe part, temps, score)

### Contraintes :

- Equipes[nationalite]  $\subseteq$  Pays[pid]
- Equipes[domaine]  $\subseteq$  Sports[sid]
- Athletes[nationalite]  $\subseteq$  Pays[pid]
- Athletes[equipe]  $\subseteq$  Equipe[eid]
- Athletes[domaine]  $\subseteq$  Sports[sid]
- Epreuve[type]  $\subseteq$  Sports[sid]
- Medailles[epreuve]  $\subseteq$  Epreuves[epid]
- Medailles[gagnant]  $\subseteq$  Athletes[aid]
- Medailles[equipe gagnante]  $\subseteq$  Equipes[eid]
- Matches[epreuve]  $\subseteq$  Epreuves[epid]
- Resultats[participant]  $\subseteq$  Athletes[aid]
- Resultats[equipe part]  $\subseteq$  Equipes[eid]

- NOT NULL :
  - Sports (nom, individuel)
  - Equipes (nom, nationalite, sexe, domaine)
  - Athletes (prenom, nom, naissance, nationalite, sexe, domaine)
  - Epreuves (nom, type, sexe)
  - Medailles (type, epreuve)
  - Matches (epreuve, dates, tour)
  - Resultats (Match)

