

Exploitation de bases de données relationnelles et OO

IFT-287

Rapport Individuel TP3

Amandine Fouillet - 14 130 638

26 mars 2015

J'ai réalisé le TP 3 sur l'approche client-serveur avec JDBC et Java dans un groupe de seulement deux personnes. Nous avons commencé par étudier les fichiers et les modèles qui nous ont été donné puis nous nous sommes mis d'accord sur l'architecture de notre programme.

Nous avons ainsi décidé de réaliser :

- Une classe TupleXXX pour chaque table du fichier schema.sql (avec XXX le nom de la table). Cette classe servant à créer des objets Java pour chaque élément compris dans la table SQL afin de traiter chaque élément plus facilement comme un objet Java.
- Une classe XXX pour chaque table du fichier schéma.sql (avec XXX le nom de la table). Cette classe servant à effectuer des requêtes sur la base de données et récupérer les résultats qui seront traités dans une autre classe.
- Une classe GestionXXX pour chaque table du fichier schéma.sql sur lesquelles des requêtes pouvaient être demandées par l'utilisateur. C'est-à-dire pour les tables Equipe, Joueur, Match, Arbitre et Arbitrer. Ces classes permettent de traiter les résultats récupérés des classes XXX.
- Une classe Connexion pour gérer la connexion à la base de données postgresql.
- Une classe Main pour gérer la lecture des arguments des utilisateurs. Si l'utilisateur met un fichier de requête en argument, la classe main se charge de la lecture et de l'exécution des requêtes du fichier. Si l'utilisateur ne met pas de fichier en argument, la classe main se charge d'exécuter les requêtes que l'utilisateur donne l'une après l'autre.

Nous avons également rajouté une table dans la base de donnée originale, la table séquence qui nous a permis de sauvegarder les valeurs des prochaines clés primaires de toutes les autres tables de la base. Dans le code Java, une classe Séquence a été rajoutée pour gérer les requêtes vers cette nouvelle table et récupérer ou modifier les prochaines clés primaires.

Notre groupe étant constitué de seulement deux personnes, il n'a pas été difficile de définir et partager les tâches entre nous. Ainsi, nous avons réalisé les deux classes Connexion et Main ensemble puis nous nous sommes partagé les tables SQL en deux et nous avons chacun de notre côté réalisé les classes en relation avec nos tables respectives. Pour ce qui est des tests, au fur et à mesure du codage, nous avons testé chacun de notre côté les fonctions que nous réalisions puis, pour finir nous avons testé ensemble avec le fichier testTP3.dat pour régler les derniers détails. Pour ce qui est du partage de code nous avons utilisé GitHub tout au long du TP pour partager nos modifications. En procédant de cette manière, la répartition des tâches a été équitable et il n'était pas impossible pour nous de travailler en même temps sans risquer de détruire le travail de l'autre puisque nous travaillions sur des classes différentes.

Ce TP fut une découverte pour moi, bien qu'ayant déjà programmé en Java et en SQL, je n'avais jamais essayé de lier les deux langages. J'ai trouvé l'utilisation de JDBC très intéressante car elle permet de réaliser un code bien structuré et facilite la programmation, la réalisation de chaque fonction et la réutilisabilité du code.