Compte-rendu - Base de données : projet
Une postface

Xian YANG, M1 Bio-Statistique

04 décembre 2016

Puisque souvent les difficultés rencontrées font aussi partie du bénéfice du travail, je me permets d'écrire cette « postface » sous forme de notes points par points avec toutes les 3 questions mélangées. Comme beaucoup de temps a déjà été investi dans le rapport et la programmation, ici je serai le plus concis possible.

- Tout d'abord, ce projet m'a amené à mettre mon premier pas dans la pratique de traitement de données et m'a fait découvrir d'autres fonctionnalités de MySQL que celles qu'on avait apprises en cours. A vrai dire, un cours de quelques mois comme le nôtre n'est pas là pour apprendre aux étudiants de statistiques quelques notions de base en base de données, puisqu'avec un niveau de confiance de 99% nous ne deviendrons jamais ingénieurs de base de données. Quelle entreprise ose d'ailleurs nous recruter, un groupe d'étudiants qui ne savaient en fin du cours que saisir des n-uplets à clavier, et donner sa base de données dans nos mains ? En revanche, le cours sert à mon avis à nous fournir un point de départ et une compétence, une compétence qui nous permettra un jours, lorsqu'on aura de nouveau besoin de cette matière, de savoir comment poser une question, de savoir où chercher la réponse, de pouvoir évoluer tout seul sans cours ni prof. De ce point de vu, le but du cours a été pour moi parfaitement atteint.

- Concernant ce qui est de méthode de travail, je me suis beaucoup appuyé au site stackoverflow.com pendant la programmation. C'est vrai qu'il vaut mieux consulter le manuel de MySQL parce que c'est la Bible, c'est vrai que le manuel d'en ligne est tellement complet qu'on trouvera certainement une réponse une fois qu'on sait le lire. Mais de chercher ce dont on a besoin parmi les 100 paramètres optionnels listés dans une formule dont 98 inconnus et impertinents n'est pas forcément la manière la plus efficace non plus pour un débutant. Sachant qu'au niveau que nous avons en ce moment, les questions qu'on puisse se poser se sont vues posées probablement déjà cent fois ailleurs, je tiens à dire que stackoverflow est un bon compagnon des programmateurs débutants qui leur fournit dans la plupart des cas un exemple intuitif et illustratif.

- La question de « big data » mise à part, je ne vois pas beaucoup de difficultés dans le projet, que ce soit le schéma entité/association, le schéma conceptuel ou la construction et la suppression des tables dans MySQL. Souvent, du moment que le nombre des objets à transmettre en relations ne dépasse pas 20, leur liens logiques en terme de base de données ont une forte chance de coïncider avec leur liens physiques. Donc si le but est juste les codes, la structure de la base de données se fait en 20 secondes dans la tête, on n'a pas forcément besoin de tel où tel schéma. Après bien sûr lorsque la complexité du problème augmente en progression géométrique, la partie théorique jouera un rôle considérablement important.

- Le premier vrai problème qui m'a bloqué c'est les contraintes d'intégrités de champ (un attribut ne peut prendre que telle ou telle valeur) et les CI plus sophistiquées disons globales (l'insertion n'est valide que lorsque certaines conditions globales soient atteintes). N'ayant pas beaucoup parlé en cours des CI autres que les CI d'entité et les CI référentielles, une première recherche dans de différentes sources du langage SQL m'a guidé naturellement à la commande `check`. Or ceci n'est qu'un piège particulier de MySQL ! On peut bien en mettre une centaine sans qu'aucun ne fonctionne ! Les `check` sont en effet tous ignorés par MySQL. Finalement je suis tombé sur la bonne réponse : le *trigger*. C'est là que j'ai commencé à sentir le goût de programmation avec

MySQL, bien qu'il m'ait fallu apprendre la boucle, la condition, les codes et messages d'erreur de SQL.

- Tout au long du projet je n'ai cessé de tenter d'établir ma base de données avec des jeux de données de taille réelle. Si la saisie de données à la main est encore faisable avec une centaine de médecins, elle n'est plus drôle avec les 600 malades. Donc comment générer, importer et exporter des données (de test) volumineuses dans et depuis une base de données ? Avec cette question j'ai découvert une série d'outils très utiles à cet effet. En premier lieu il y a des générateurs de données sur l'internet, dont le fameux genaratedata.com. C'est grâce à lui que j'ai eu mes 214 médecins et mes 216 chambres. L'avantage de ce site c'est qu'il peut fournir des noms et prénoms francophones... Mais les fonctionnalités de ce genre de sites sont très limitées en terme de génération de nombres aléatoires (comme MySQL mais MySQL te laisse au moins programmer). De ce fait, pour les six cents malades avec toutes les tranches d'âges et de sexe différents, dont une partie doit porter une contre-indication, je suis tourné au logiciel R, parce que R est très fort en génération des données de divers caractères à une proportion exactement comme on veut. Cela étant dit, le paquet qui s'occupe de fabriquer des noms et des prénoms ne produit malheureusement que des noms anglophones... Donc finalement j'ai un hôpital dont tous les médecins sont français et dont tous les patients parlent anglais. Mais ce n'est pas très grave ! Puisque j'ai stocké ces 600 malades dans un fichier csv séparé j'ai du aussi apprendre les méthodes d'importation des données en MySQL, employées une deuxième fois plus tard avec la liste des médicaments. Et finalement j'ai généré plus de quatre milles d'ordonnances à partir d'une fusion des tables `malade` et `medecin` que j'avais en main et cette fois-ci uniquement avec des fonctions de MySQL. Il faut que je critique la fonction `rand()` proposée par MySQL qui est très faible ! Elle ne sait que générer un nombre entre 0 et 1. Du coup pour obtenir une date aléatoire entre deux autres il a fallu d'abord transformer la différence en nombre de jours, multiplié par ce nombre entre 0 et 1, l'arrondir à un nombre entier, puis l'additionner à nouveau à la date la plus petite. Et avec les définitions de variables `set` imaginez combien de lignes ça fait. Quelle torture ! Alors qu'en R c'est juste 3 mots. Les architectes de base de données comment testent-ils une nouvelle base avec des Go de données compatibles ? Ça m'intéresserait.

- Un dernier point pour finir, ce travail m'a entraîné entre autres l'utilisation de LaTeXet d'Excel (pour faire le schéma E/A), un bénéfice secondaire qui mérite le temps investi !