Bases de Données I -- TP1

3ASTI - INSA Centre Val de Loire

B. Nguyen

Ressources : Disponibles sur : http://www.benjamin-nguyen.fr/ENS/3ASTI-BD1/TP1/

Dans ce TP nous allons programmer les opérateurs de l'algèbre relationnelle classique. Le fichier funct.h contient les signatures des fonctions. Le fichier funct.c contient l'implémentation des fonctions de manipulation de n-uplets et de relations, ainsi que l'exemple de l'implémentation d'un opérateur : l'opérateur union (OpUnion)

Hypothèses : pour simplifier le problème, nous ne manipulons que des attributs de type entier (int). Les colonnes n'ont pas de nom, et seront simplement identifiées par leur numéro. Les relations n'ont pas de nom et sont donc identifiées par leur nom de variable.

Questions :

1) Regardez le code du programme funct.c, compilez-le et exécutez-le. Vérifiez que vous en comprenez bien le fonctionnement, et que toutes les lignes de code sont claires.

2) Regardez plus précisément le code de la fonction OpUnion. Implémentez maintenant la fonction OpInter (qui fait l'intersection de deux relations de même schéma). Ecrivez un programme main exemple pour illustrer que votre opérateur fonctionne bien.

3) Implémentez les opérateurs de restriction, de projection et de produit cartésien. Vérifiez à chaque fois par un exemple que l'opérateur fonctionne. **NB :** nous allons utiliser deux opérateurs de restriction différents : l'opérateur OpRestrictCST et OpRestrictATT. Le premier opérateur permettra d'appliquer des restrictions de type *attribut OP valeur* (où OP = {<; =; >}), tandis que le deuxième opérateur permettra d'appliquer des restrictions de type *attribut1 OP attribut2*.

4) Implémentez la jointure. Réfléchissez à un bon algorithme de jointure. On testera ensuite l'efficacité de votre algorithme sur une table assez large.

5) Définissez un langage permettant de représenter une expression de l'algèbre relationnelle. Ce langage sera une chaine de caractères que vous pourrez parser en C ou avec lex/yacc.

6) (sur papier) Donnez un algorithme permettant de passer d'une expression de la logique du premier ordre vers une expression équivalente de l'algèbre relationnelle. Définissez un langage

pour représenter une formule de la logique du premier ordre. Un utilisateur devra pouvoir écrire une requête (== expression de logique du 1er ordre) dans votre langage.

7) En utilisant lex et yacc ou bien un parseur écrit en C, écrivez un compilateur permettant de passer d'une expression de la logique du premier ordre vers l'exécution d'une requête. Félicitations, vous avez programmé un SGBD très simple !!