

LIF10 – FONDEMENTS DES BASES DE DONNÉES

TD – Introduction aux dépendances

Licence informatique – printemps 2012–2013

Résumé

Exemples d'exercices $\text{\LaTeX} 2_{\epsilon}$

Exercice 1 : vérification des dépendances en SQL

Prouver que $r, s \models R[X] \subseteq S[Y]$ si et seulement si $|\pi_X(r) \setminus \pi_Y(s)| = 0$, en déduire une requête SQL qui permet de tester la satisfaction d'une dépendance d'inclusion.

Answer of exercise 1

On procède par équivalence successives. Par définition, $r, s \models R[X] \subseteq S[Y]$ est équivalent à $\pi_X(r) \subseteq \pi_Y(s)$ ce qui à son tour équivaut à $\pi_X(r) \setminus \pi_Y(s) = \emptyset$ qui équivaut à $|\pi_X(r) \setminus \pi_Y(s)| = 0$. Pour SQL, on va traduire assez directement $|\pi_X(r) \setminus \pi_Y(s)|$ et tester si le résultat est 0 :

```
SELECT COUNT(*)
FROM (SELECT DISTINCT X FROM r
      MINUS
      SELECT DISTINCT Y FROM s)
```

Exercice 2 : inférence des dépendances

On rappelle les règles d'inférences suivantes pour les dépendances fonctionnelles.

$Y \subseteq X$ σ_R (réflexivité)	$X \rightarrow Y$	$X \rightarrow Y$	$X \rightarrow Y$ $X \rightarrow Z$ σ_C (composition)
σ_A (augmentation)	$WX \rightarrow WY$	$X \rightarrow Y$	$X \rightarrow YZ$ $X \rightarrow YZ$ σ_D (décomposition)
$Y \rightarrow Z$ σ_T (transitivité)	$X \rightarrow Z$	$X \rightarrow Y$ $X \rightarrow Y$ $WY \rightarrow Z$ σ_P	(pseudo-transitivité) $WX \rightarrow Z$

Soit Σ l'ensemble des dépendances suivantes

$BC \rightarrow A$	$D \rightarrow BE$
$AC \rightarrow B$	$B \rightarrow DE$
$AE \rightarrow C$	$C \rightarrow E$

- En utilisant les règles le système d'Armstrong augmenté des règles de composition et décomposition, prouver que les DF suivantes appartiennent à $\Sigma^+ = \{f \mid \Sigma \models f\}$:
 - $AD \rightarrow C$
 - $CD \rightarrow A$
- Même question en calculant la fermeture des parties gauches à partir de l'algorithme de fermeture.

Answer of exercise 2

- $D \rightarrow BE$ aug. $AD \rightarrow ABE$ $AE \subseteq ABE$ refl. $ABE \rightarrow AE$ trans. $AD \rightarrow AE$ $AE \rightarrow C$ trans. $AD \rightarrow C$
 - $D \rightarrow BE$ decomp. $D \rightarrow B$ aug. $CD \rightarrow BC$ $BC \rightarrow A$ trans. $CD \rightarrow A$
- Pour $AD \rightarrow C$, les étapes successives de l'algorithme sont les suivantes :

- (a) $closure = AD$
 - (b) $closure = ABDE$ en utilisant $D \rightarrow BE$
 - (c) $closure = ABCDE$ en utilisant $AE \rightarrow C$
 - (d) comme $C \subseteq AD^+$, on en déduit $AD \rightarrow C$ par correction de l'algorithme
2. Pour $CD \rightarrow A$, les étapes successives de l'algorithme sont les suivantes :
- (a) $closure = CD$
 - (b) $closure = BCDE$ en utilisant $D \rightarrow BE$
 - (c) $closure = ABCDE$ en utilisant $BC \rightarrow A$
 - (d) comme $A \subseteq CD^+$, on en déduit $CD \rightarrow A$ par correction de l'algorithme

Algorithme 1 : fermeture d'un ensemble d'attributs par des dépendances fonctionnelles

Data : Σ un ensemble de DF, X un ensemble d'attributs.

Result : X^+ , la fermeture de X par Σ .

```

1  $unused := \Sigma$ 
2  $closure := X$ 
3 repeat
4    $closure' := closure;$ 
5   if  $W \rightarrow Z \in unused$  and  $W \subseteq closure$  then
6      $unused := unused - \{W \rightarrow Z\};$ 
7      $closure := closure \cup Z;$ 
8 until  $closure' = closure;$ 
9 return  $closure;$ 
```
