Feuille 4 de Introduction aux SI et aux BD, L2 2014-2015
VALABLE POUR 2 SEANCES

Exercice 1(Création de tables en SQL)

Un concepteur d'une base de données pour la gestion de l'ensemble des distributeurs automatiques de sodas d'un campus univérsitaire a décidé d'organiser les informations selon les tables :

- VEND(IdDistrib,IdBoisson, Prix), munie de la dépéndance fonctionnelle:
  IdDistrib IdBoisson $\rightarrow$ Prix.
- Boisson(IdBoisson,NomBoisson,Sucre), oú Dom(Sucre)=$\{`O', `N'\}$ (OUI, NON) et la dépéndance fonctionnelle associée est :
  IdBoisson $\rightarrow$ Sucre.
- AIME(NomPersonne,IdBoisson), avec { NomPersonne,IdBoisson } comme clé primaire.

Vous retrouvez ici la base étudiée dans le DS. Ce qu'on vous demande dans cet exercice est de:

1. Utiliser la commande `create table` de SQL pour créer les trois tables. N'oubliez pas que toute valeur d'un attribut composant la clé (primaire) d'une table doit forcement respecter la contrainte `NOT NULL`., et, pour chaque table, votre déclaration doit indiquer la clésprimaire et, éventuéllement, la clé étrangére.
2. Utiliser la commande `INSERT... INTO...VALUES` de SQL pour insérer au moins une ligne de valeurs dans chaque table.

Exercice 2(Calcul de ${X^+}_F$)

On dit qu'un ensemble de dépéndances fonctionnelles F implique une dépéndance fonctionnelle $X \rightarrow Y$, X et Y étant deux ensembles d'attributs, si toute table R qui rend vraie toute dépéndance de F rend forcement vraie $X \rightarrow Y$.

En cours on a vu un algoritme permettant de décider si F implique ou pas une dépéndence fonctionnelle $X \rightarrow Y$. Cet algoritme utilise comme sous-routine l'algorithme de calcul de la fermeture d'un ensemble d'attributs modulo un ensemble de dépendances fonctionnelles F, notéé ${X^+}_F$. Utiliser l'algoritme de décision du cours pour démontrer ou réfuter les assertions suivantes :

1. $F_1 : \{A \rightarrow BC, A \rightarrow E\}$ implique $A \rightarrow BCE$

2. $F_2 : \{A \to BC, BCE \to F\}$ implique $AE \to F$

3. $F_3 : \{C \to B\}$ implique $C \to BD$

Si vous réfutez une assertion i, construisez aussi un contre-exemple : une table R_i telle que toute dépéndance de F_i est vraie mais la dépéndance qui suit le mot $\langle\langle$ implique $\rangle\rangle$ est fausse.

Exercice 3(Intuitions sur SPI)

La table *Planning_ Examens* a le schéma S suivant :

$$Etudiant\ Examen\ Heure\ Date$$

qui est équipé de l'ensemble de dépendances fonctionnelles suivant :

$$\{Etudiant\ Examen\ \to\ Heure\ Date,\quad Etudiant\ Heure\ Date\ \to Examen\}$$

On décompose S en $S_1 = Etudiant\ Examen$ et $S_2 = Etudiant\ Heure\ Date$. Donnez un exemple de valeurs pour la table initiale *Planning_ Examens* qui montre que cette décomposition est problématique, au sens qu'elle comporte une *perte d'information* négative (elle n'est pas SPI).

Exercice 4(Algoritmes pour tester si SPI)

1. Soit $S = ABCD,\ F = \{A \to B,\ B \to C\}$ et la décomposition AB, BC, CD. En utilisant l'algorithme *chase* du cours, décider si cette décomposition est SPI ou pas selon F. Si ce ne l'est pas, fournir un contre-exemple.

2. Exactement les mêmes questions qu'avant, mais en prenant, au lieu de F, l'ensemble de dépendances $F' = \{A \to B,\ B \to C,\ C \to D\}$.

3. Soit $S = ABCD$, et soit F" l'ensemble de dépendances :
   $\{AB \to C,\ BC \to D\}$.
   Soit la décomposition ABC, BCD. Mêmes questions qu'avant, mais, attention, ici, selon le cours, vous avez le choix entre **deux** algorithmes possible pour tester SPI. Pourquoi ?

Exercice 5(Intuitions sur SPD)

Soit R une table ayant comme schéma : $Num\_Vol\ Date\ Porte\ Heure\ Destination$ et comme ensemble de dépendances :
$F = \{Num\_Vol\ Date\ \to\ Porte,\ \ Num\_Vol\ \to\ Destination\ Heure,$
$Date\ Porte\ Heure\ \to\ Num\_Vol\}$
on décompose R en $R_1 = Num\_Vol\ Destination\ Heure$ et $R_2 = Num\_Vol\ Porte\ Date$. Indiquer les ensembles F_{R_1} et F_{R_2} des dépendances applicables aux deux nouveaux schémas qui sont les projections de F sur, respectivement R_1 et R_2 (voir le cours pour la définition précise).
Donner une suite d'insertions dans la base ainsi décomposée qui respecte F_{R_1} et F_{R_2} mais telle que, si on fait la jointure des relations obtenues, on obtient un n-uplet qui viole au moins une des contraintes établies en origine par F (laquelle ?).

On aura ainsi montré que cette décomposition *fait perdre des dépéndances*, ce qui veut dire qu'elle n'est pas SPD selon F.

Exercice 6(Tester si SPD)

- **Algorithme SPD : complément de cours**

  Entrée : Un ensemble S d'attributs (ceux d'une table), un ensemble F de dépendances fonctionnelles et une décomposition d de S: $R_1, R_2, \ldots, R_n$.
  Sortie : Vrai si d est SPD, faux sinon.

  1. Initialisation : SPD = vrai;
  2. Itération principale : Tant que SPD et il existe $X \rightarrow Y \in F$ non traitée faire : Répeter :
  3. jusqu'a ce que $Y \subseteq Z$ ou Z ne change pas.

Algorithme SPD : questions d'application

En appliquant cet algorithme, tester si les deux décompositions d_1 et d_2 suivantes suivantes sont SPD (*sans perte de dependances*) par rapport Ãă F ou pas :

1. Le schéma Ãă décomposer est $ABCD$, $F = \{A \rightarrow C,\ D \rightarrow C, BD \rightarrow A\}$ et d_1 est la décomposition en 3 schémas : AB, ACD et BCD.
2. Le schéma Ãă décomposer est $ABCD$, $F = \{A \rightarrow B,\ B \rightarrow CD,\ A \rightarrow D\}$ et d_2 est la décomposition en 2 schémas : AB et et BCD.