Correction TD Normalisation

Exercice I

1) **Couverture minimale** : Application de l'algorithme de calcul d'une couverture minimale**.**

Etape 1. F' := F.

<u>Etape 2.a</u> : On fait en sorte d'avoir des parties droites de DF ne contenant qu'un seul élément : F' = {
AB → C, B → D, B → E, BDL → K, BHJL → C, C → A, C → B
CEL → K, CIL → G, CIL → K, D → B, D → E, EIK → C, EIK → G
EIK → L }

<u>Etape 2.b :</u> On supprime des parties gauches les attributs non nécessaires.

✓ **AB → C** : est ce qu'on peut remplacer **AB → C par A → C ou B → C ???**

- **C ∈ A^+?**
  On calcule A^+ :
  A^+ = A, ⇒ C ∉ A^+...... (1) on ne peut pas remplacer **AB → C par A → C**
- **C ∈ B^+?**
  On calcule B^+ :
  B^+ = BDE ⇒ C ∉ B^+. ...... (2) on ne peut pas remplacer **AB → C par B → C**

✓ Pour **B → D et B → E** il n'y a rien à faire car les parties gauches ne contiennent qu'un attribut.

✓ **BDL → K** : est ce qu'on peut remplacer BDL → K par DL → K ou BD → K ou bien BL → K ?

- K ∈ **(DL)$^+$** ?
  On calcule (DL)$^+$:
  (DL)$^+$ **= DLBEK.** ⇒ K ∈ (DL)$^+$
  Donc en connaissent DL, on connait K. On peut donc remplacer BDL → K par DL → K

On fait la même chose avec **DL → K** : peut-on remplacer DL → K par D → K ? Ou, *autrement dit,* $K \in D^+$? La réponse est non.

$K \in L^+$? La réponse est non.

Alors on remplace BDL → K par DL → K.

Remarque ; On n'aura pas besoin de vérifier les autres possibilités (BD → K ou bien BL → K).

✓ **BHJL → C :** est ce qu'on peut remplacer BHJL → C par **HJL → C ou BJL → C ou BHL → C ou bien BHJ → C ?**

- **C** ∈ **(HJL)$^+$** ?
  On calcule (HJL)$^+$:
- **(HJL)$^+$** = HGL ⇒ **C** ∉ **(HJL)$^+$** ; donc on ne peut remplacer BHJL → C **par HJL → C.**

De même, on vérifie pour les autres :

- **(BJL)$^+$** = BGLDEK ⇒ **C** ∉ **(BJL)$^+$** ; donc on ne peut remplacer BHJL → C **par BJL → C.**
- **(BHL)$^+$** = BHLDEK ⇒ **C** ∉ **(BHL)$^+$** ; donc on ne peut remplacer BHJL → C **par BHL → C.**
- **(BHJ)$^+$** = BHGDE ⇒ **C** ∉ **(BHJ)$^+$** donc on ne peut remplacer BHJL → C **par BHJ → C.**

Donc la DF **BHJL → C** est minimale.

Pour **C → A et C → B** il n'y a rien à faire car les parties gauches ne contiennent qu'un attribut.

On fait les mêmes calcules pour les DFs restantes :

- ✓ **CEL → K :** Après calcul, on remplace donc CEL → K par CL → K car $K \in CL^+$ (mais $K \notin C^+$ et $K \notin L^+$).
- ✓ **CIL → G** est minimale
- ✓ **CIL → K** : Après calcul, on remplace donc CIL → K par CL → K qui est déjà dans F' et qu'on a déja testé.

Pour **D → B et D → E** il n'y a rien à faire car les parties gauches ne contiennent qu'un attribut.

- ✓ **EIK → C :** C'est une DF élémentaire.
- ✓ **EIK → G :** C'est une DF élémentaire.
- ✓ **EIK → L :** C'est une DF élémentaire

<u>3ème étape :</u> On supprime une à une les DF et on regarde si on peut la retrouver à l'aide des autres.

Ici, c'est le cas de B → E que l'on peut retrouver grace à B → D puis D → E.

C'est aussi le cas de CIL → G car avec C, I, et L on obtient K (CL → K), B (C → B), puis D (B→ D) et E (D → E), et enfin G par (EIK → G)

Enfin, CL → K peut aussi être enlevée : si on connaît C et L, on a A (C → A), B (C → B), puis D (B → D), et enfin K (DL → K) La couverture minimale est donc

F^0 = {AB → C ; B → D ; DL → K ; BHJL → C ; C → A ; C → B ; D → B ; D → E ; EIK → C ; EIK→ G ; EIK → L}

Clés candidates ?

Remarque :

1- Les attributs qui se retrouves toujours à droite des DFs de F^0 sont des attributs qui ne peuvent jamais faire partie des clés candidates ; on les appelle « attributs non clé ».

Donc attributs non clé : G

2- Les attributs qui se retrouves toujours à guache des DFs de F^0 sont des attributs qui ne font toujours partie des clés candidates ; on les appelle : Attributs clé.

Donc attributs clé : I,J,H

On calcule $(IJH)^+$,

Si le résultat est égal à l'ensemble de tous les attributs de R ; alors IJH est une clé et se sera la seule clé de R.

Sinon On augmentera IJH par l'un des attributs qui se retrouve à gauche est à droite des des DF de F^0. Et on refait le calcul pour chaque augmentation.

- $(IJH)^+$ = IJH ⇒ IJH ne suffit pas pour la clé ; on procède aux calcul par augmentation.
- $(AIJH)^+$ = AIJH ⇒ AIJH n'est pas une clé ; on augmente encore avec L.
- $(AIJHL)^+$ = AIJHL ⇒ AIJHL n'est pas une clé
- $(BIJH)^+$ = BIJHDE ⇒ BIJH n'est pas une clé
- $(BIJHL)^+$ = BIJHLDEKCAG ⇒ BIJHL est pas une première clé candidate.
- Comme on a C → B alors CIJHL est une deuxième clé candidate.

- De même : on D → B alors DIJHL est une troisième clé candidate.
- On a aussi EIK→C et EIK→L alors EIJHK est une quatriàme clé candidate

Donc les clés candidates sont : BHIJL ; CHIJL ; DHIJL ; EIKHJ

Et enfin, après calcul o trouve :

Attributs clé : B,C,D,E,H,I,J,K,L

Attributs non clé : A, G.

La relation n'est pas en 2-NF car une partie non clé dépend que d'un morceau de la clé : C → A ou bien EIK→ G.

On décompose :

R1(<u>A</u>, B, C) qui est 3-NF car tous les attributs sont des attributs clé;

DFs associées : AB → C, C→A ; C→B

R2(E,I,K, G) qui est 3-NF ; DF associée : EIK→ G.

R2(B,C,D,E,H,I,J,K,L) qui est en 3 FN ; car tous ses attributs sont des attributs clé.

DFs associées : {B → D ; DL → K ; BHJL → C ; C→B ; D → B ; D → E ; EIK → C ; EIK → L}

Exercice II

Il n'y a qu'une clé candidate (donc minimale) dans cette relation : HE.

Question 2 :

Le schéma est en 2-NF mais n'est pas en 3-NF car, par exemple, la dépendance fonctionnelle C → P lie deux attributs n'appartenant pas à la clé.

Question 3 :

Appliquons l'algorithme de mise sous 3-NF donné dans le C :

1/ Couverture minimale : seule la dépendance HES → N peut être enlevée (on la retrouve via HE → S puis HS → C et enfin CE → N. Aucune des autres DF ne peut être simplifiée.

2/ On se retrouve avec les groupes suivants :

G1 = {C, P}, G2 = {H, S, C}, G3 = {H, P, S}, G4 = {C, E,N}, G5={H, E, S}.

3/ On crée les relations suivantes :

R1 = {<u>C</u>, P}, R2 = {<u>H, S</u>, C}, R3 = {<u>H, P,</u> S}, R4 = {<u>C, E</u>,N}, R5={<u>H, E</u>, S}.

4/ La clé candidate se retrouve dans la relation R5, il n'y a donc rien à faire à cette étape.

Exercice III

1) - Par transitivité, A → C. Donc A → B, C. A est donc clé candidate. Comme A n'apparait pas en partie droite de dépendance fonctionnelle, il doit être dans toute clé. Or il est lui-même clé : c'est donc la seule clé candidate.

- R est en 2FN (car la clé est composé d'un seul attribut), mais pas en 3FN (car B → C lie deux attributs qui ne sont pas dans la clé).

2) La DF B → C n'est pas respectée dans l'extension de R' car (B1, C1) et (B1, C2), donc connaître la valeur de B ne permet pas de déterminer la valeur de C.

3) Par exemple :

R"	A	B	C
	A1	B1	C1
	A2	B1	C1
	A3	B2	C1
	A4	B3	C3

4) Couverture minimale : { A → B ; B → C }

G1 = { A → B }, G2 = { B → C }

R1 = { <u>A</u>, B }, R2 = { <u>B</u>, C }