

Corrigé type

Exercice I (6 pts)

a) Citer trois problèmes liés à la présence de la redondance de données dans une BD? Dans quels cas cette redondance est nécessaire?

1.
 - *Gaspillage d'espace de stockage.*
 - *Incohérence de base de données, par exemple, problème de la propagation des mises à jour.*
 - *Gaspillage de ressources humaines, par exemple, une même donnée peut être saisie à plusieurs reprises, avec un risque toujours présent de commettre des erreurs au moment d'entrer l'information au sein du système.*

2. Cette redondance est nécessaire dans *le cas de la clé étrangère pour assurer les liens entre les différentes informations.*

b) Est-ce qu'une clé étrangère peut prendre la valeur **NULL**? Justifier votre réponse.

Oui, si elle ne fait pas partie de la clé primaire.

c) démontrer la règle suivante en utilisant les axiomes d'Armstrong : $(X \rightarrow Y) \wedge (WY \rightarrow Z) \Rightarrow (WX \rightarrow Z)$
 $X \rightarrow Y \Rightarrow WX \rightarrow WY$ par augmentation de W ... (1) et on a $WY \rightarrow Z$... (2)
 Par transitivité de (1) et (2) on obtient $WX \rightarrow Z$

d) Pourquoi normalise-t-on le schéma d'une base de données relationnelle?

Réduire les redondances et donc éliminer les anomalies possibles lors de la mise à jour.

e) Soient deux relations **R1(A:D1, B:D2, C:D3)** et **R2(B:D2, C:D3)**.

1. Exprimer en **SQL** la division de R1 par R2 ($R1 \div R2$).

```
SELECT A FROM R1 R3
WHERE not exists (SELECT B, C FROM R2
                  WHERE not exists (SELECT A, B, C FROM R1
                                   WHERE A = R3.A and B = R2.B and C = R2.C));
```

2. Travaillant sur la relation R1, donner une requête en algèbre relationnelle qui ne retourne aucun tuple dans le cas où la dépendance fonctionnelle $B \rightarrow C$ est respectée.

```
R2 ← R1
Resultat ← R1 ⋈(R1.B = R2.B) ∧ (R1.C ≠ R2.C) R2
```

Exercice II (4 pts)

On considère le schéma relationnel de la base de données MICRO.

```
CLIENT (noC, nomC, prenom, Adresse)
PRODUIT (noP, libelle, marque, prix, QteStock)
VENTE (noC, noP, dateVte, QteVendue)
```

a) Formuler en algèbre relationnelle les requêtes suivantes :

1. Les numéros des clients qui ont acheté au moins un produit de la marque 'DELL'.

```
Resultat ←  $\Pi_{noC} (\delta_{marque = "DELL"} (PRODUIT) \bowtie VENTE)$ 
```

2. La liste des produits les moins chers (N'utiliser pas la fonction d'agrégat **MIN**)

```
P1 ← PRODUIT
P2 ← PRODUIT
Resultat ← PRODUIT ⋈(\Pi_{prix} (PRODUIT) - (\Pi_{P2.prix} (P1 ⋈P1.prix < P2.prix P2)))
```

b) Formuler en **SQL** les requêtes données ci-dessous.

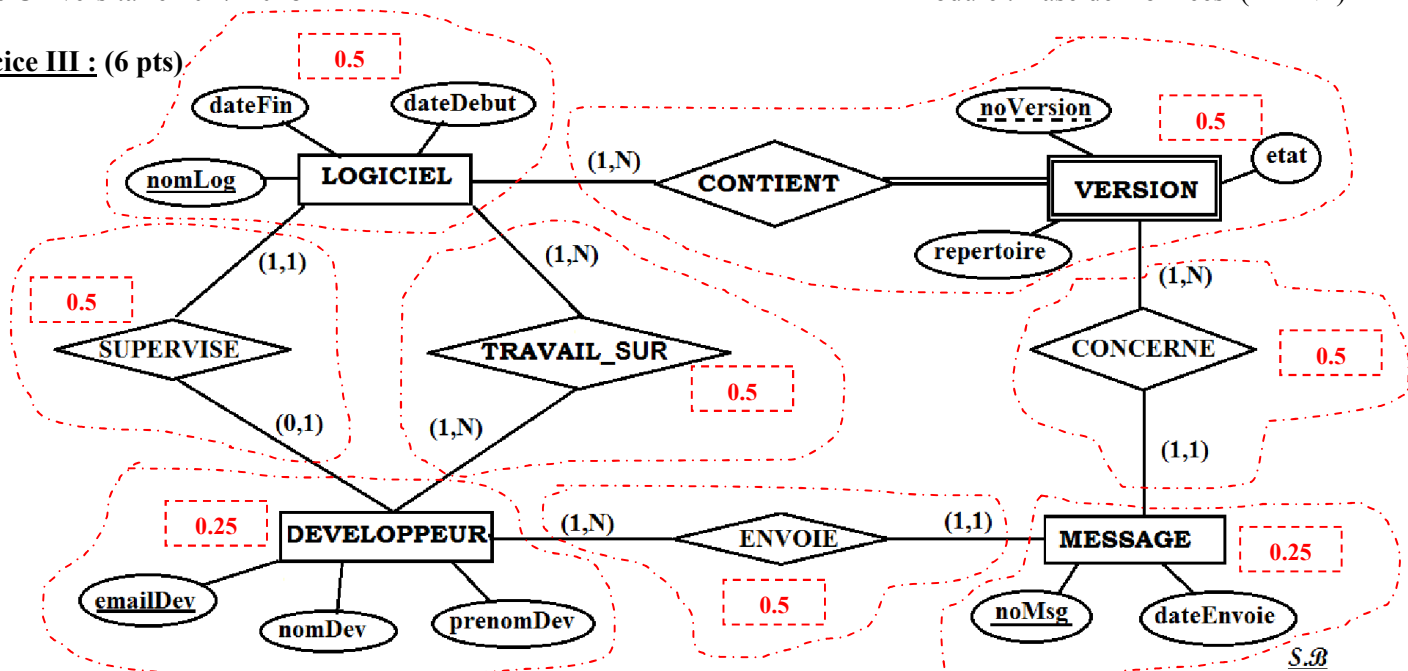
1. Nom et prénom des clients qui ont acheté plus d'un produit de la marque 'DELL'.

```
SELECT nomC, prenom
FROM PRODUIT P INNER JOIN (CLIENT C INNER JOIN VENTE V ON C.noC = V.noC) ON
P.noP = V.noP
WHERE marque = "DELL"
GROUP BY C.noC, nomC, prenom
HAVING COUNT(distinct V.noP) > 1;
```

2. Numéro des produits non vendus (En utilisant **OUTER JOIN**).

```
SELECT P.noP FROM PRODUIT P LEFT OUTER JOIN VENTE V ON P.noP = V.noP
WHERE noC is NULL;
```

Exercice III : (6 pts)



LOGICIEL(nomLog, dateDebut, dateFin, emailDev*)
 VERSION(nomLog*, noVersion, repertoire, etat)
 MESSAGE(noMsg, dateEnvoie, nomLog*, noVersion, emailDev*)
 TRAVAIL_SUR(emailDev*, nomLog*)
 DEVELOPPEUR(emailDev, nomDev, prenomDev)

Les clés étrangères sont indiquées par des astérisques.

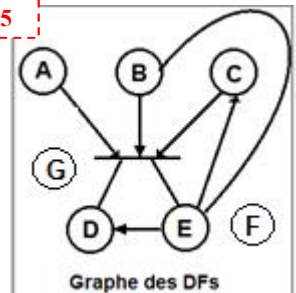
BONUS 0.5

Exercice IV (4 pts)

Soit la relation R(A, B, C, D, E, F, G) qui respecte la 1FN et satisfaire l'ensemble des dépendances fonctionnelles $\mathbb{F} = \{ABC \rightarrow DE, E \rightarrow BCD\}$.

a) Donner toutes les clés candidates de R. Justifier votre réponse. *Basant sur le graphe de DFs, on peut déduire deux clés candidates.*

<p>La première clé candidate est ABCFG</p> <p>Comme $[ABCFG]^+ = R$ ainsi que</p> <p>$[BCFG]^+ = \{BCFG\} \neq R$, $[ACFG]^+ = \{ACFG\} \neq R$, $[ABFG]^+ = \{ABFG\} \neq R$, $[ABCG]^+ = \{ABCG\} \neq R$ et $[ABCF]^+ = \{ABCF\} \neq R$.</p>	<p>La deuxième clé candidate est AEFG</p> <p>comme $[AEFG]^+ = R$ ainsi que</p> <p>$[AFG]^+ = \{AFG\} \neq R$, $[EFG]^+ = \{E, B, C, D, F, G\} \neq R$. $[AEG]^+ = \{A, E, B, C, D, G\} \neq R$ et $[AEF]^+ = \{A, E, B, C, D, F\} \neq R$.</p>
---	--



b) En quelle forme normale est R pour chaque clé candidate choisie comme clé primaire.

- Si **ABCFG** est choisie comme clé primaire alors on a **ABCFG \rightarrow DE** mais **E \rightarrow D** donc R n'est pas en 3FN, **elle est en 2FN**.
- Si **AEFG** est choisie comme clé primaire alors on a **AEFG \rightarrow BCD** mais **E \rightarrow BCD** donc R n'est pas en 2FN, **elle est en 1FN**.

c) Proposer une décomposition en 3FN pour cette relation en utilisant l'algorithme de synthèse.

En appliquant l'algorithme de synthèse, on doit tout d'abord calculer une couverture minimale de F.

min(F) = F - {ABC \rightarrow D} donc min(F) = {ABC \rightarrow E, E \rightarrow B, E \rightarrow C, E \rightarrow D} et toutes les DFs sont sous forme canonique. Toutefois, R se décompose en deux relations en 3FN :

- R1(A, B, C, E)
- R2(E, C, D)
- R3(A, B, C, F, G)