

**Exercice n° 1 : (3pts)**

1. La clé primaire de la relation Restaurant est **Num\_Menu** (avec justification)
2. La plus grande forme normale de Restaurant :  
**1FN**: vérifiée car tous les attributs de la relation Restaurant sont atomiques  
**2FN**: vérifiée car la relation Restaurant est en 1FN et tous ses attributs non clé sont en Df irréductibles (élémentaires) avec sa clé primaire Num\_Menu  
**3FN**: la relation Restaurant doit être en 2FN (vérifiée) et tous ses attributs non clé doivent être en Df directe avec sa clé primaire Num\_Menu : **Non vérifiée** car Nom\_plat n'est pas en Df directe avec Num\_Menu puisque Num\_Plat  $\rightarrow$  Nom\_Plat.
3. Décomposition de la relation Restaurant en 3FN.  
Principe de normalisation : effectuer des projections pour éliminer les dépendances transitives.  
Si  $R(\underline{A}, B, C)$  et  $A$  est la clé primaire de  $R$  et  $B \rightarrow C$  est une DF satisfaite par  $R$ , alors  $R$  peut être décomposée en :  $R_2(\underline{A}, B)$  et  $R_1(\underline{B}, C)$

Restaurant (Num\_menu, Nom\_Menu, Num\_Plat, Nom\_Plat, Type\_Plat )  
Nom\_plat n'est pas en Df directe avec Num\_Menu (Num\_Plat  $\rightarrow$  Nom\_Plat), De même pour Type\_plat (Num\_Plat  $\rightarrow$  Type\_Plat)

Restaurant sera décomposée en  
Menu (Num\_menu, Nom\_Menu, Num\_Plat) en 3FN  
Plats(Num\_Plat, Nom\_Plat, Type\_Plat) en 3FN

Autre Solution : On peut appliquer l'algorithme de synthèse pour déterminer la décomposition 3FN

**Exercice n°2 : (9pts)**

**Algébrique :**

- 1-  $\pi_{\text{NumAut}}(\text{Ecrit}) - \pi_{\text{NumAut}}(\text{Ecrit} - \pi_{\text{NumEdit, NumLivre}}(\text{livres}))$
- 2-  $\pi_{\text{NumEdit, NumEdit}}(\text{Editeur} \bowtie (\pi_{\text{NumEdit}}(\text{Editeur}) - \pi_{\text{NumEdit}}(\text{Livres} \bowtie \sigma_{\text{NumAut}=120}(\text{Ecrit}))))$

**Prédicatif :**

- 3-  $\{x.\text{NomAut} / \text{Auteur}(x) \wedge \text{Ecrit}(y) \wedge \text{Livres}(z) \wedge x.\text{NumAut}=y.\text{NumAut} \wedge y.\text{NumLivre}=z.\text{Numlivre} \wedge y.\text{NumAut}=z.\text{NumEdit}\}$
- 4-  $\{x.\text{NomEdit} / \text{Editeur}(x) \wedge \text{Livres}(y) \wedge \text{Ecrit}(z) \wedge \forall y (x.\text{NumEdit}=y.\text{NumEdit} \rightarrow \exists z (y.\text{NumLivre}=z.\text{Numlivre} \wedge z.\text{NumAut}=y.\text{NumEdit}))\}$

**SQL:**

- 5- 

```
SELECT NumAut FROM Ecrit x, Livres y
WHERE x.Numlivre=y.Numlivre AND AnnéeEdit between (2000 AND 2012)
GROUP BY NumAut
HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM Ecrit x', Livres y'
WHERE x'.Numlivre=y'.Numlivre AND AnnéeEdit between (2000 AND 2012)
GROUP BY NumAut;
```

## 6- Solution 1 (avec MINUS)

```
(SELECT NumAut FROM Ecrit x, Livre y, Editeur z
WHERE x.Numlivre=y.NumLivre AND y.NumEdit=z.NumEdit
AND z.NomEdit ='SCIENCE ET SAVOIR')
MINUS
```

```
(SELECT NumAut FROM ecrit x, Livre y, Editeur z
WHERE x.Numlivre=y.NumLivre AND y.NumEdit=z.NumEdit
AND z.NomEdit <>'SCIENCE ET SAVOIR');
```

## Solution 2 (avec NOT EXISTS)

```
SELECT NumAut FROM Auteur a
WHERE NOT EXISTS (SELECT * FROM ecrit x
WHERE x.NumAut= a.NumAut
AND NOT EXISTS (SELECT * FROM Livre y, Editeur z
WHERE x.NumLivre=y.numLivre AND y.NumEdit=z.NumEdit
AND z.NomEdit ='SCIENCE ET SAVOIR'));
```

## Exercice n° 3 : (8pts)

### 1. Ensemble des dépendances fonctionnelles :

Codp → nomp, desp, famp

numz → desz, locz, surz

numz, date-debut, date-fin, codp → quantitéMax (quantité max à pêcher par poisson par zone par période)

numv → nomv, typv, vitv, lonv, prov, gerv

codep → nomp, villep, surp, dirp

nums → codep, numv, **gerv**, datesortie, durée\_estim

nume → **codep**, **numv**, **gerv**, dateentré, durée\_eff, **nums** (un bon d'entrée correspond à 1 bon de sortie)

nume, numz, codp → quantitésP (quantité pêchée par poisson par zone)

### 2. Application de l'algorithme de synthèse :

#### Etape 1 : Couverture minimale

Propriété 1: Membres droits des DFs singletons : Décomposer les Dfs

Propriété 2 : Membres gauches des DFs irréductibles : Vérifiée

Propriété 3 : Aucune Df n'est redondante : Enlever les Dfs redondantes qui sont :

nums → gerv      nume → codep      nume → numv      nume → gerv (justifier)

#### Etape 2 : Grouper les Dfs par membres gauches

#### Etape 3 : Construire les relations : Les relations obtenues sont :

Poisson (Codp , nomp, desp, famp)

Zone (numz, desz, locz, surz)

Pêche (numz, date-debut, date-fin, codep , quantitéMax)

Navire (numv , nomv, typv, vitv, lonv, prov, gerv)

Port (codep , nomp, villep, surp, dirp)

Bon\_ sortie (nums , codep, numv, datesortie, durée\_estim)

Bon\_entrée (nume, dateentré, durée\_eff, **nums**)

Zones\_visitées (nume, numz, codp , quantitésP)

Zones\_ciblées (nums, numz)