BASE DE DONNÉES – TD6

OPTIMISATION

**LÊ NGUYỄN MINH HƯƠNG**

# 1. ÉLIMINATION DE REDONDANCES :

Personnel ( NuméroEmployé, Nom, Prénom, DateEmbauche, IntituléFonction )

+ NuméroEmployé NUMBER(4) ( 4 octets )

+ Nom CHAR(20) ( 20 octets )

+ Prénom CHAR(20) ( 20 octets )

+ DateEmbauche DATE ( 7 octets )

+ IntituléFonction CHAR(20) ( 20 octets )

Problème : Trop de redondances pour l’attribut IntituléFonction

🡺 OCCUPATION : 1000\*(4 + 20 + 20 + 7 + 20 ) = 71 000 (octets )

Nouveau schéma relationnel :

( xxx : clé primaire xxx : clé externe )

Personnel ( NuméroEmployé, Nom, Prénom, DateEmbauche, NuméroFonction )

+ NuméroEmployé NUMBER(4) ( 4 octets )

+ Nom CHAR(20) ( 20 octets )

+ Prénom CHAR(20) ( 20 octets )

+ DateEmbauche DATE ( 7 octets )

+ NuméroFonction NUMBER(4) ( 4 octets )

Fonction ( NuméroFonction, IntituléFonction )

+ NuméroFonction NUMBER(4) ( 4 octets )

+ IntituléFonction CHAR(20) ( 20 octets )

🡺 OCCUPATION : 1000\*( 4 + 20 +20 + 7 + 4 ) + 25\*( 4 + 20 ) = 55 600

|  |  |  |
| --- | --- | --- |
|  | Ancien schéma | Nouveau schéma |
| Avantage | 1 seule table à manipuler, facile de voir des données désirées | 2 tables, il faut réaliser des jointures pour connaître des données désirées |
| Inconvénient | Grande occupation de mémoire car des données de l’attribut de type CHAR IntituléFonction sont répétées 40 fois | Optimisation de l’occupation de mémoire, CHAR (20 octets) est remplacé par NUMBER (4 octets) |

# 2. CHOIX D’ UNE CLÉ :

Personnes ( Nom, Prénom, DateNaissance, Adresse, Téléphone )

Voiture ( NuméroImmatriculation, Marque, Couleur, Nom, Prénom, DateNaissance )

Nouvel schéma relationnel :

Personnes ( NuméroPersonne, Nom, Prénom, DateNaissance, Adresse, Téléphone )

Voiture ( NuméroImmatriculation, Marque, Couleur, NuméroPersonne )

|  |  |  |
| --- | --- | --- |
|  | Ancien schéma | Nouveau schéma |
| Avantage | Données précises, facile à connaître les informations désirées | Optimisation du schéma, moins d’espace de mémoire est occupé, des redondances sont réduites |
| Inconvénient | Il occupe un grand espace de mémoire | Difficile à connaître toutes les informations désiréees sans réalisaer une jointure entre des tables |

# 3. CONTRAINTE D’INTÉGRITÉ RÉFÉRENTIELLE BASÉE SUR PLUSIEURS ATRIBUTS :

R ( a, b, c )

S ( d, e, a, b )

( S.a, S.b ) référence ( R.a, R.b )

- On a :

( a, b ) 🡪 c

d 🡪 e

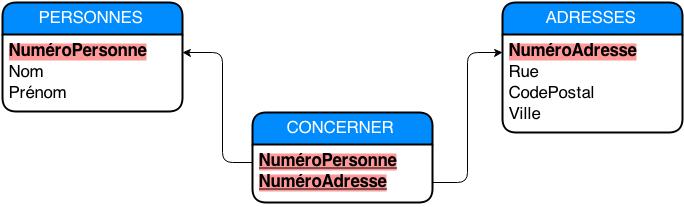
d 🡪 ( a, b )

🡺 d 🡪 e, d 🡪 c, d 🡪 ab

Nouveau schéma relationnel :

R ( d, a, b, c, e )

# 4. EXPLOITATION DES CONTRAINTES D’ INTÉGRITÉ ( 1/ 3 ) :

- Ancien schéma relationnel :

Le nombre de lignes dans la table CONCERNER = 2 \* le nombre de lignes dans la table PERSONNES.

- Nouveau schéma relationnel :

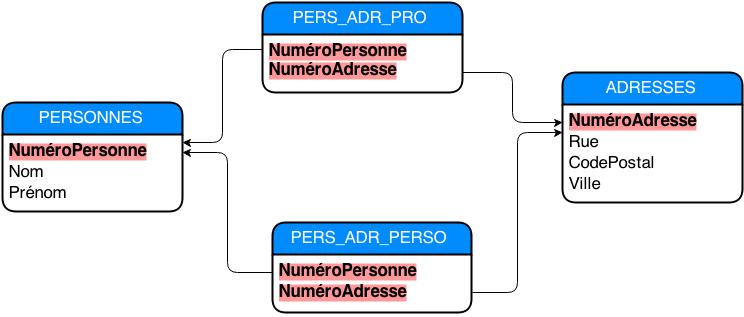
PERSONNES ( NuméroPersonne, Nom, Prénom )

ADRESSES ( NuméroAdresse, Rue, CodePostal, Ville )

PERS\_ADR\_PERSO ( NuméroPersonne, NuméroAdresse )

PERS\_ADR\_PRO ( NuméroPersonne, NuméroAdresse )

Le nombre de lignes dans 2 tables PERS\_ADR\_PERSO et PERS\_ADR\_PRO est le même que celui dans la table CONCERNER de l’ancien schéme. De plus, ce nouveau schéma permet de distinguer entre l’adresse personnelle et l’adresse professionnelle.



# 5. EXPLOITATION DES CONTRAINTES D’ INTÉGRITÉ ( 2/3 ) :

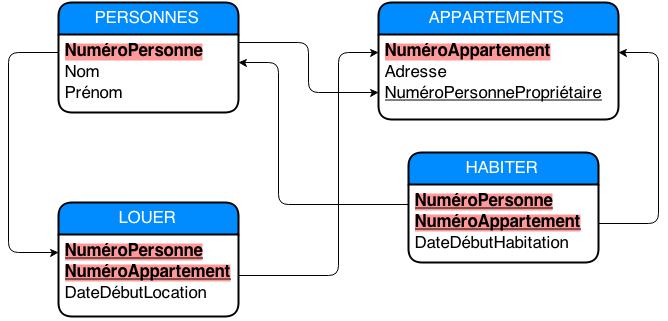
- Ancien schéma relationnel :

PERSONNES ( NuméroPersonne , Nom , Prénom )

APPARTEMENTS ( NuméroAppartement , Adresse , NuméroPersonnePropriétaire )

LOUER ( NuméroAppartement , NuméroPersonne , DateDébutLocation )

HABITER ( NuméroAppartement , NuméroPersonne , DateDébutHabitation )

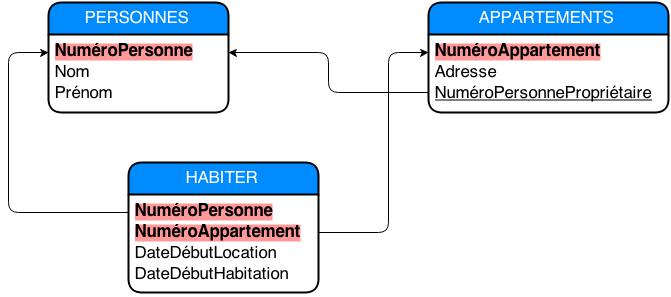


- Nouveau schéma relationnel :

PERSONNES ( NuméroPersonne, Nom, Prénom )

APPARTEMENTS ( NuméroAppartement, Adresse, NuméroPersonnePropriétaire )

HABITER ( NuméroAppartement, NuméroPersonne, DateDébutLocation, DateDébutHabitation )



- Les tables LOUER et HABITER ont un seul différent attribut : DateDébutLocation et DateDébutHabitation, alors il vaut mieux mixer ces 2 tables pour économiser de l’espace.

# 6. EXPLOITATIONS DES CONTRAINTES D’ INTÉGRITÉ :