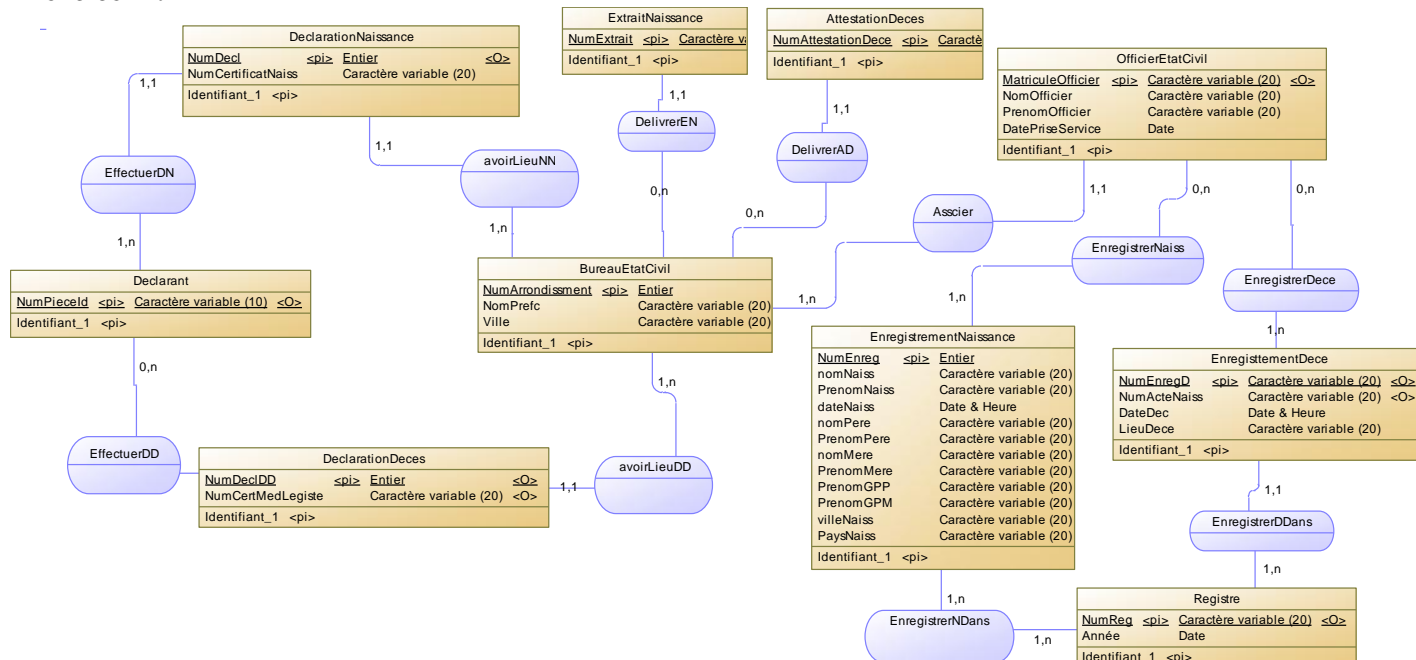




Correction

Exercice 1 :



Exercice 2 :

Rappel énoncé: La relation suivante décrit des commandes faites par des clients, avec les produits et quantités commandées par client.

Commandes (NumCom, DateCom, NumCli, AdrCli, NumProd, Prix, Qte)

- Quelle est la clé de cette relation ?
- En quelle forme normale elle est ?
- La mettre en 3FN le cas échéant.

Correction :

1.

Avant de chercher la clé, il faut d'abord déterminer les DFs. on a les règles de gestion suivantes :

- NumCom** → **DateCom**, **NumCli**, **AdrCli** (Une commande est faite par un seul client avec une adresse donnée et à une date donnée)
- NumCom**, **NumProd** → **Prix**, **Qte** (Dans une commande, un produit a un prix donnée est commandé avec une quantité donnée)
- NumCli** → **AdrCli** (Un client a une seule adresse)
- NumProd** → **Prix** (Il y a un seul prix pour un produit)

On peut alors dire que (**NumCom**, **NumProd**) déterminent tous les autres attributs. C'est donc une clé.

2.

a. Comme **NumCli**, entre autres attributs, ne dépend que de **NumCom**, c'est à dire une partie de la clé, la relation est en 1FN mais pas en 2FN. On décompose donc :

Commandes (NumCom, DateCom, NumCli, AdrCli)

Com-Pros (NumCom, NumProd, Prix, Qte)

b. On a la relation Commandes n'est pas en 3FN, on décompose
Commandes (NumCom, DateCom, NumCli)
Clients (NumCli, AdrCli)
 qui sont en 3FN

c. On a aussi la relation **Com-Prods** n'est pas en 2FN. On décompose en deux relations
Com-Prods (NumCom, NumProd, Qte)
Produits (NumProd, Prix)
 qui sont en 3FN.

Résultat final : les quatre dernières relations (b.) et (c.)

Commandes (NumCom, DateCom, NumCli)
Clients (NumCli, AdrCli)
Com-Prods (NumCom, NumProd, Qte)
Produits (NumProd, Prix)

Exercice 3 :

1. Schéma relationnel

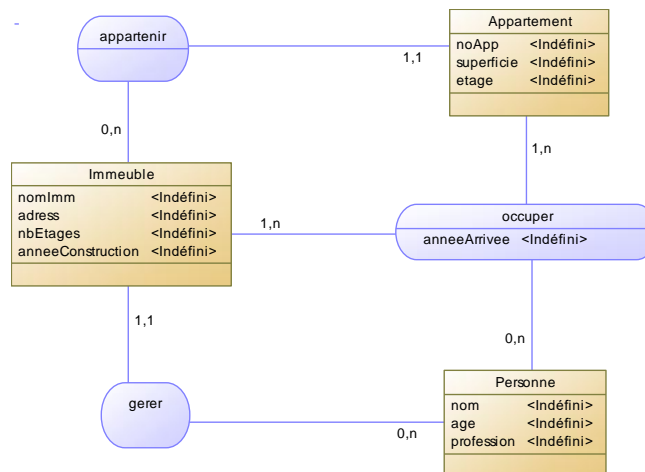
Immeuble (nomImm, Adresse , nbEtages, annéeConstruction, #nomGérant)

Appart (noApp, #nomImm, superficie, étage)

Personne (nom, age, prefession)

Occupant (nomImm, noApp, nomOcc, anneeArrivee)

2. MCD Correspondant :



3. Script SQL pour créer la table Occupant

```

CREATE TABLE Occupant (
    nomImm varchar(20),
    noApp int,
    nomOccup varchar(20),
    anneeArrivee Year,
    PRIMARY KEY (nomImm, noApp, nomOccup),
    FOREIGN KEY (nomImm) REFERENCES Immeuble(nomImm),
    FOREIGN KEY (noApp) REFERENCES Appart(noApp),
    FOREIGN KEY (nomOccup) REFERENCES Personne(nom)
); ENGINE=INNODB;
  
```

4. 4.a.

profession
Rentier
Informaticien

4.b

nomGerant	superficie
Doug	150
Doug	50
Doug	200
Doug	50
Ross	250
Ross	250

5. 5.a. $\pi_{nomOcc}(\sigma_{superficie < 70}(Appart \bowtie_{noApp=noApp} Occupant))$

5.b $\pi_{nomOcc}(Immeuble \bowtie_{nomGerant \neq nomOcc} Occupant)$

Autre solution : $\pi_{nomOcc}(\sigma_{nomGerant \neq nomOcc}(Immeuble \bowtie_{nomImm = nomImm} Occupant))$

6. 6.a. **Select** P.age, P.profession

From Personne P

Inner Join Occupant O **ON** O.nomOcc=P.nom

Inner Join Immeuble I **ON** O.nomImm=I.nomImm

Where I.nomGerant= "Ross";

6.b. **Select** A.nomImm, A.noApp

From Appart A

MINUS

Select O.nomImm, O.noApp

From Occupant O;

6.c **Select** AVG(P.age)

From Personne P

Inner Join Occupant O **On** P.nom=O.nomOccup

Inner Join Immeuble I **On** O.nomImm=I.nomImm

Where (O.anneeArrivee – I.anneeConstruction) > 20;

6.d **Select** Count(*)

From Personne P

Inner Joint Occupant O **On** P.nom = O.nomOcc

Inner Joint Appart A **On** O.noApp = A.noApp

Where P.profession = "Informaticien" **and** A.superficie > 100 **and** O.nomImm = "Koudalou";