

Concevoir la base de données

TRAVAIL

Nom : AIT SI ELARABI

Prénom : IDRISS

Group : F

Nom : Elmarhraoui

Prénom : Ayoub

Group : F

Encadré par : M. Amine Chaoulid

Méthode Merise :

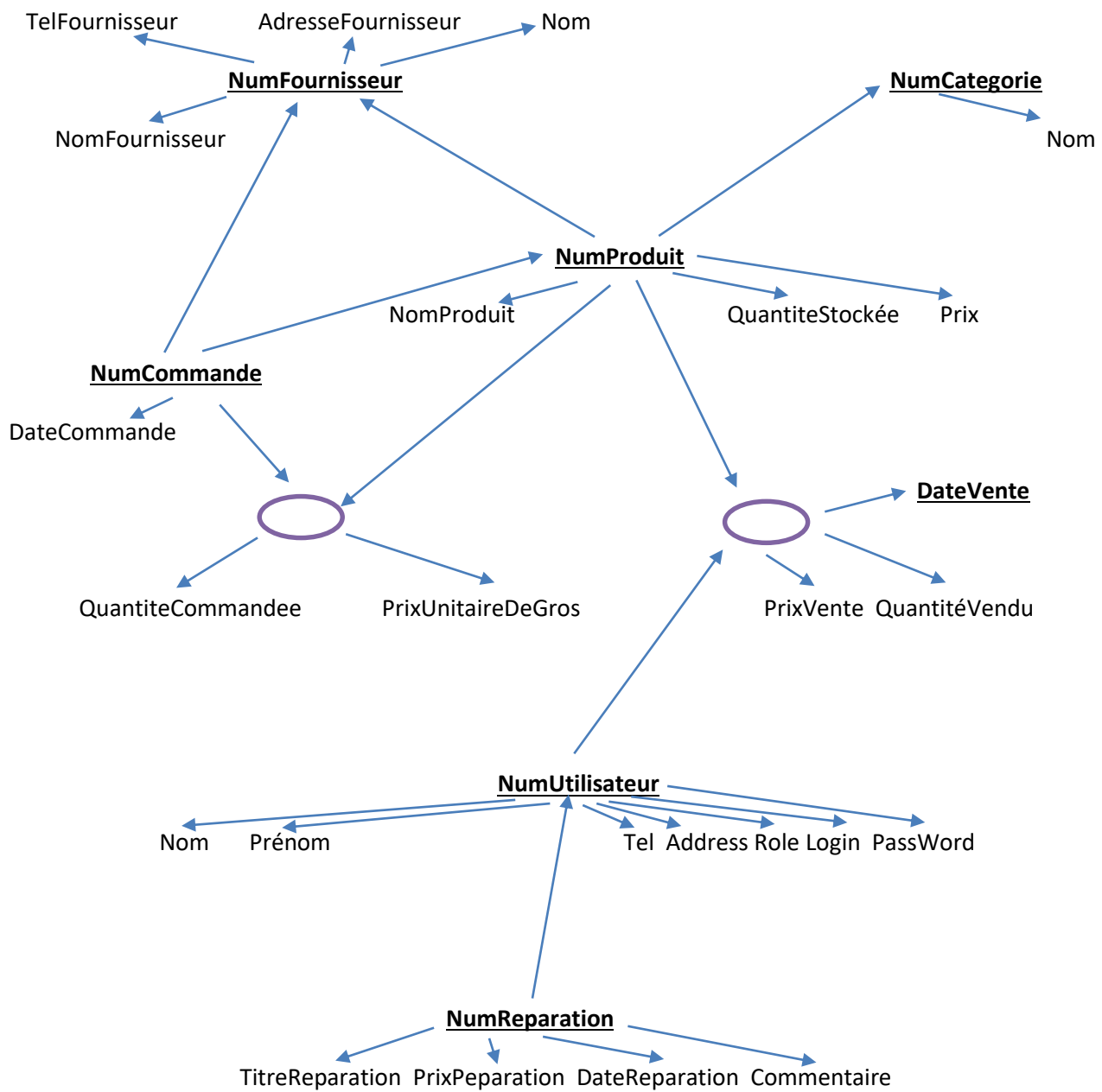
I. Dictionnaire de données :

Propriétés	Description	Type
NumFournisseur	Numéro du Fournisseur	Entier
TelFournisseur	Téléphone du Fournisseur	Chaine de caractères
AdresseFournisseur	Adresse du Fournisseur	Chaine de caractères
NomFournisseur	Nom du Fournisseur	Chaine de caractères
NumUtilisateur	Numéro d'utilisateur	Entier
Nom	Nom d'utilisateur	Chaine de caractères
Prénom	Prénom d'utilisateur	Chaine de caractères
Tel	Téléphone d'utilisateur	Chaine de caractères
Adresse	Adresse d'utilisateur	Chaine de caractères
Rôle	Rôle d'utilisateur	Chaine de caractères
Login	Login d'utilisateur	Chaine de caractères
PassWord	Mot de passe d'utilisateur	Chaine de caractères
NumCategorie	Numéro de la catégorie	Entier
Nom	Nom de la catégorie	Chaine de caractères
NumReparation	Numéro de la réparation	Entier
TitreReparation	Titre de la réparation	Chaine de caractères
PrixReparation	Prix de la réparation	Réel
DateReparation	Date de la réparation	Date
Commentaire	Commentaire sur la réparation	Chaine de caractères
NumProduit	Numéro de Produit	Entier
NomProduit	Nom de Produit	Chaine de caractères
QuantiteStockee	Quantité stockée de Produit	Entier
Prix	Prix de Produit	Réel
NumCommande	Numéro de Commande	Entier
DateCommande	Date De la Commande	Date
QuantiteCommandee	Quantité de produit commandé	Entier
DateVente	Date du Vente	Date
PrixVente	Prix du Vente	réel
QuantiteVendu	Quantité Vendu	Entier
PrixUnitaireDeGros	Prix Unitaire de Produit De Gros	réel

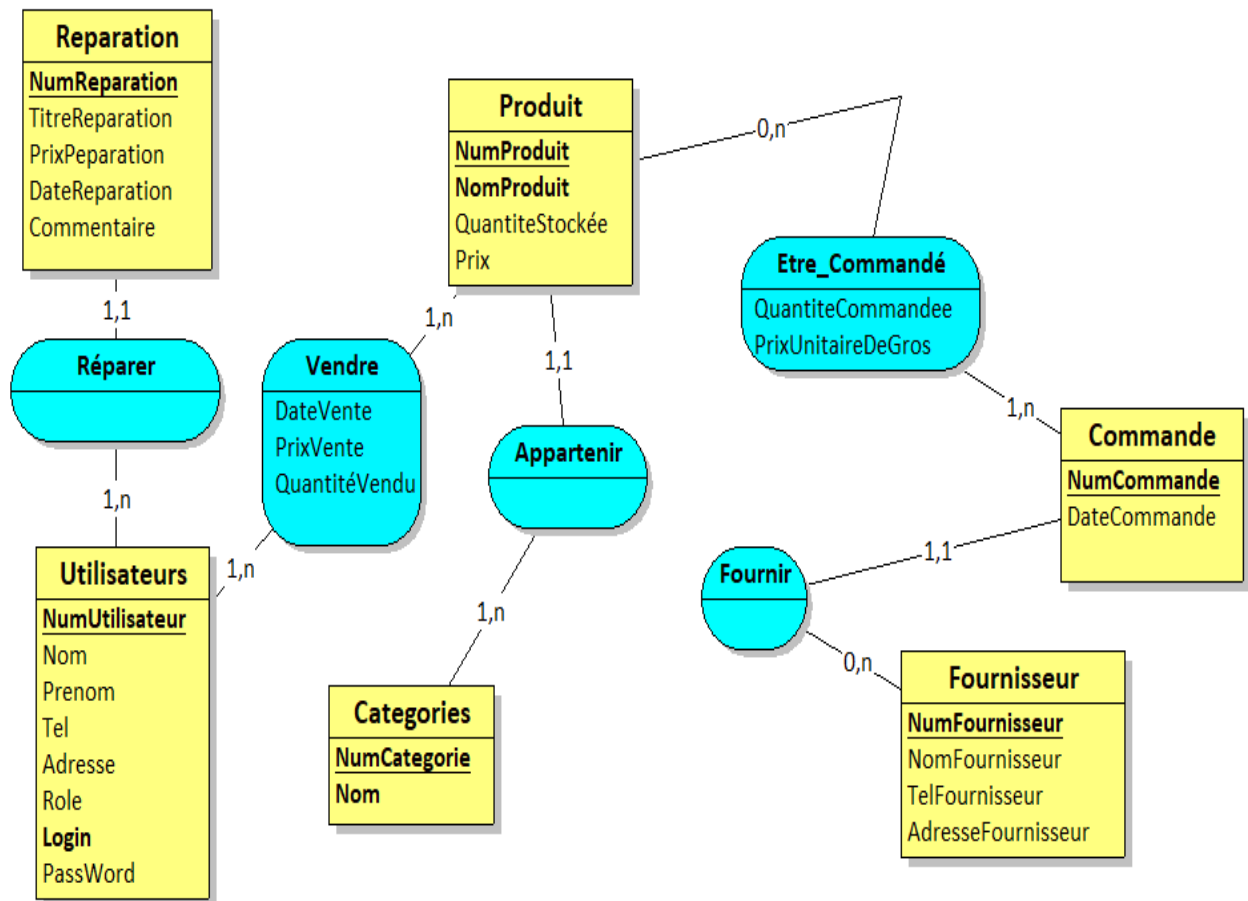
II. Ensemble de règles de gestion :

- Un utilisateur fait plusieurs réparations et ventes.
- Un produit appartient à une seule catégorie.
- Un produit peut être vendu et commandé plusieurs fois.
- Un fournisseur peut faire plusieurs livraison ou aucun.

III. Graphe de dépendances fonctionnelles entre les données :



IV. Modèle conceptuel de données (MCD) :



V. Modèle logique de données (MLD) :

- A. **Fournisseur** = (NumFournisseur, NomFournisseur, TelFournisseur, AdresseFournisseur);
- B. **Utilisateurs** = (NumUtilisateur, Nom, Prenom, Tel, Adresse, Role, Login, PassWord);
- C. **Categories** = (NumCategorie, Nom);
- D. **Reparation** = (NumReparation, TitreReparation, PrixPreparation, DateReparation, Commentaire, #NumUtilisateur);
- E. **Produit** = (NumProduit, NomProduit, QuantiteStockée, Prix, #NumCategorie);
- F. **Commande** = (NumCommande, DateCommande, #NumFournisseur);
- G. **Vendre** = (#NumProduit, #NumUtilisateur, DateVente, PrixVente, QuantitéVendu);
- H. **EtreCommandé** = (#NumCommande, #NumProduit, QuantiteCommandee, PrixUnitaireDeGros);

VI. Script SQL de la base de données :

i. Création de la base de données :

```
CREATE DATABASE Gestion_Ventes_Produits_Reparations
```

ii. Création des tables

```
CREATE TABLE Fournisseurs (
    NumFournisseur INT PRIMARY KEY,
    NomFournisseur VARCHAR(50),
```

```
TelFournisseur VARCHAR(10) CHECK (TelFournisseur LIKE '0[5-7]+'+REPLICATE('[0-9]',8)),
AdresseFournisseur VARCHAR(100)
);
```

```
CREATE TABLE Utilisateurs (
    NumUtilisateur INT PRIMARY KEY,
    Nom VARCHAR(50),
    Prenom VARCHAR(50),
    Tel VARCHAR(10) CHECK (Tel LIKE '0[5-7]+'+REPLICATE('[0-9]',8)),
    Adresse VARCHAR(100),
    Role VARCHAR(50),
    Login VARCHAR(50) NOT NULL UNIQUE,
    PassWord VARCHAR(50) NOT NULL
);
```

```
CREATE TABLE Categories (
    NumCategorie INT PRIMARY KEY,
    Nom VARCHAR(50) UNIQUE
);
```

```
CREATE TABLE Reparations(
    NumReparation INT PRIMARY KEY,
    TitreReparation VARCHAR(50),
    PrixReparation FLOAT,
    DateReparation DATE DEFAULT GETDATE(),
    Commentaire VARCHAR(50),
    NumUtilisateur INT NOT NULL FOREIGN KEY REFERENCES Utilisateurs(NumUtilisateur)
);
```

```
CREATE TABLE Produits (
    NumProduit INT PRIMARY KEY,
    NomProduit VARCHAR(50) UNIQUE,
    QuantiteStockee INT,
    Prix FLOAT,
    NumCategorie INT NOT NULL FOREIGN KEY REFERENCES Categories(NumCategorie)
);
```

```
CREATE TABLE Commandes (
    NumCommande INT PRIMARY KEY,
    DateCommande DATE,
    NumFournisseur INT NOT NULL FOREIGN KEY REFERENCES Fournisseurs(NumFournisseur)
);
```

```
CREATE TABLE Ventes (
    NumProduit INT FOREIGN KEY REFERENCES Produits(NumProduit),
    NumUtilisateur INT FOREIGN KEY REFERENCES Utilisateurs(NumUtilisateur),
    DateVente DATE DEFAULT GETDATE(),
    PrixVente FLOAT,
    QuantiteVendu INT,
```

```
PRIMARY KEY(NumProduit, NumUtilisateur,DateVente)
);
CREATE TABLE EtreCommandé (
  NumCommande INT FOREIGN KEY REFERENCES Commandes(NumCommande),
  NumProduit INT FOREIGN KEY REFERENCES Produits(NumProduit),
  QuantiteCommandee INT,
  PrixUnitaireDeGros FLOAT,
  PRIMARY KEY(NumProduit, NumCommande)
);
```