

## Série d'exercices

### R105 Introduction aux bases de données et SQL

#### Exercice 4 :

Un agriculteur fait de la vente directe de ses produits ou animaux qu'il élève. Il vend des lapins, des poules, des dindes, des veaux, des cochons. Selon la saison il vend aussi des légumes (choux, pomme de terre, carottes, ...) et des fruits (fraises, poires, pommes, ...) il ne fait que de la vente directe. Suite à votre discussion, il ressort les informations suivantes.

A l'heure actuelle, les ventes sont inscrites sur trois cahiers distincts :

- Un pour les animaux
- Un pour les fruits
- Un pour les légumes

Tout est vendu au kilos, les animaux sont pesés vivants avant d'être vendus. Il souhaiterait un logiciel simple pour saisir les ventes journalières et pouvoir éditer un récapitulatif mensuel par type de vente (animaux, légumes, fruits) et par produit (poulet, lapins, poireaux, poires, ...) pour sa comptabilité.

- Créer le modèle conceptuel des données
- Concevoir le modèle logique de données
- Finir par le modèle physique des données

#### Solution :

#### **Entités et Attributs :**

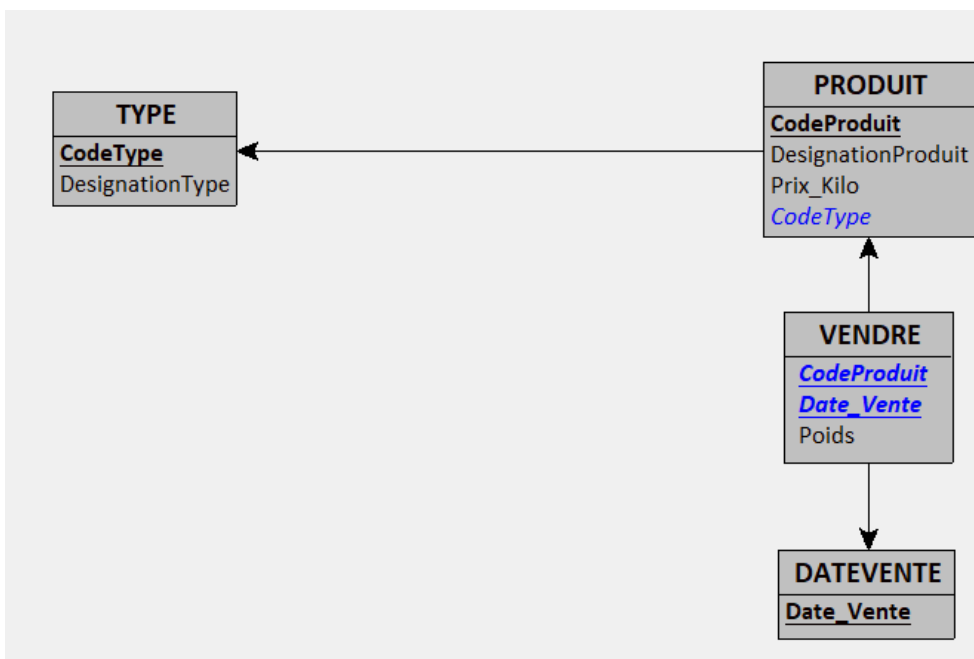
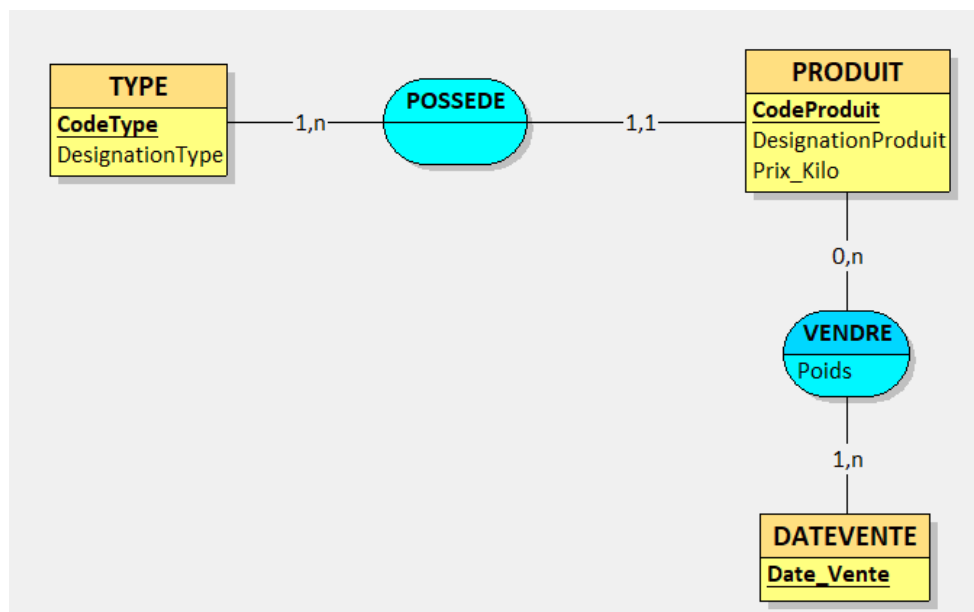
1. **Produit :**
  - Code\_Produit (PK)
  - Nom\_Produit
  - Prix\_Kilo
2. **Type\_Produit :**
  - Code\_Produit (PK)
  - Designation
3. **Date :**
  - Date de vente (PK)

#### **Associations et Cardinalités :**

- **Produit - date :**
  - Un produit peut ne pas être vendu à une date donnée et peut faire l'objet de plusieurs ventes (0,N).
  - Une vente concerne un seul ou plusieurs produits (1, N).
- **Produit - TypeProduit :**
  - Un produit possède un seul et un seul type (animal, légume, fruit) (1,1)
  - Un type peut être possédé par plusieurs produits (1, N).

### Schéma relationnel :

- **Types**(CodeType, Désignation)
- **Produit** (Code Produit, Nom\_Produit, Prix\_Kilo, #codeType)
- **Date**(DateDeVente)
- **Vendre** (#Code Produit, #DateDeVente, Poids)



**Attention :** si on sépare les fruits, les légumes et les animaux (une Entité pour chacun d'entre eux), on risque d'avoir des imperfections. Imaginons la structure physique de la table Vendre avec le schéma suivant :

Vendre (#codeAnimal, #codeLegumes, #CodeFruit, #DateVente, Poids)

#codeAnimal	#codeLegumes	#CodeFruit	#DateVente	Poids
1	NULL	NULL	06/09/24	2
2	15	5	09/09/24	5

La 1 ère ligne apparait avec deux codes NULL, le fichier risque de contenir des cellules vides, ça va l'alourdir inutilement

La ligne 2 présente un inconvénient majeur, à quel produit correspond le poids inscrit (5 kilos)

Avec notre modèle :

CodeType	DésignationType
1	Animaux
2	Fruits
3	Légumes

CodeProduit	DesignationProduit	Prix_Kilo	#CodeType
1	Lapin	7	1
2	Veau	11	1
3	Salade	1.2	3
4	Endives	11	3
5	Pommes	5	2
6	Noisettes	15	2

CodeProduit	Date_Vente	Poids
1	07/09/24	2
1	08/09/24	1
6	08/09/24	0.5
2	08/09/24	4

Le modèle devient cohérent, la première ligne de la table Vendre, il a été vendu un produit de code numéro 1 (Animal) un lapin à 7€ le kilos le 7 septembre pour 2 kilos. Le total du sera calculé, il n'est pas nécessaire de le stocker.

### **Exercice 5 :**

Comme toutes les personnes de votre village font appel à vos services lorsqu'ils ont un problème informatique, vous vous dites que c'est peut re le moment de vous mettre à votre compte.

Pour démarrer il vous faut un petit logiciel vous permettant de saisir vos interventions pour faciliter la tenue de votre comptabilité.

Ce logiciel permettra la saisie des coordonnées des clients et le matériel sur lequel vous êtes intervenu.

Vous décider d'appliquer un prix horaire différent selon le type d'intervention (certaines réparation ou manipulation complexes doivent être facturée plus cher.

Pour certaines pannes vous devez vendre le composant neuf. Le logiciel devra donc intégrer la vente de matériel inhérent à la réparation

- Concevoir le dictionnaire des données simplifié
- Créer le modèle conceptuel des données
- Concevoir le model logique de données
- Finir par le modèle physique des données

**Solution :**

## **1. Dictionnaire des Données Simplifié**

Voici un dictionnaire des données simplifié pour le logiciel de gestion d'interventions. Chaque entité, attribut et description est détaillée.

Entité	Attribut	Description
<b>Client</b>	ID_Client	Identifiant unique du client
	Nom	Nom du client
	Prénom	Prénom du client
	Adresse	Adresse du client
	Téléphone	Numéro de téléphone
	Email	Adresse email du client
<b>Matériel</b>	ID_Matériel	Identifiant unique du matériel
	Désignation	Nom ou description du matériel (ordinateur, imprimante, etc.)
	Fabricant	
	Date d'achat	
<b>Intervention</b>	Num_Intervention	Identifiant unique de l'intervention
	Date_Intervention	Date de l'intervention
	Type_Intervention	Type d'intervention (réparation, maintenance, installation, etc.)
	Temps_passé	
	Prix_Horaire	Prix horaire appliqué pour le type d'intervention

Entité	Attribut	Description
	Code_intervention	
	Désignation	
<b>Composant</b>	ID_Composant	Identifiant unique du composant
	Désignation	Nom ou description du composant
	Prix_Achat	Prix d'achat du composant

## 2. Modèle Conceptuel de Données (MCD)

Le MCD décrit les entités et les relations entre elles pour organiser les données. Voici les entités et associations définies pour ce projet :

### Entités et Attributs :

#### 1. Client

- Code\_Client (PK)
- Nom
- Prénom
- Adresse
- CP
- Ville
- Téléphone
- Email

#### 2. Matériel

- Code\_Matériel (PK)
- Désignation
- Fabricant
- Date d'achat

#### 3. Intervention

- Num\_Intervention (PK)
- Description panne
- Date\_Intervention
- Temps passé

#### 4. Composant

- référence\_Composant (PK)
- Désignation
- Prix\_Achat

#### 5. TypeIntervention

- Code\_Intervention (PK)
- Désignation
- Prix\_Horaire

### Associations :

- Client - Matériel :

- Un client possède un ou plusieurs matériels (1,N).
- Un matériel est possédé par un et un seul client (1,1).
- **Matériel - Intervention :**
  - Un matériel peut avoir aucune ou plusieurs interventions (0,N).
  - Une intervention concerne un et un seul matériel (1,1).
- **Intervention – Composant :**
  - Une intervention peut inclure (utiliser) aucun ou plusieurs composants (0,N).
  - Un composant peut être utilisé dans aucune ou plusieurs interventions (0,N).
- **Intervention – Type d'intervention :**
  - Une intervention est classifiée par un et un seul type (1,1).
  - Un type d'intervention peut correspondre zéro ou plusieurs interventions (0,N).

## Résumé des Dépendances Fonctionnelles :

- Les clés primaires (ID\_Client, ID\_Matériel, ID\_Intervention, ID\_Composant, ID\_Vente) déterminent l'ensemble des autres attributs de leur table respective.
- Les clés étrangères établissent des dépendances fonctionnelles entre les tables, permettant de retrouver des informations détaillées à partir des identifiants des clients, des matériels, des interventions et des composants.

Ces dépendances fonctionnelles sont nécessaires pour garantir la cohérence et l'intégrité des données dans un système relationnel.

Le **graphe des dépendances fonctionnelles** permet de représenter visuellement les relations de dépendance entre les attributs d'une base de données. Voici comment on peut représenter les dépendances fonctionnelles pour le modèle de gestion des interventions informatiques. Les nœuds représentent les attributs, et les flèches montrent les dépendances entre eux.

### 1. Table Client

```
code_Client --> Nom
              --> Prénom
              --> Adresse
              --> CP
              --> Ville
              --> Téléphone
              --> Email
```

### 2. Table Matériel

```
Code_Matériel --> Désignation
               --> Fabricant
               --> date d'achat
```

### 3. Table Intervention

```
Num_Intervention --> DescriptionPanne
                  --> Date_Intervention
                  --> TempsPassé
```

### Table TypeIntervention

Code\_Intervention --> Designation  
 --> Prix\_Horaire

#### 4. Table Composant

ID\_Composant --> Désignation  
 --> Prix\_Achat

### 6. Dépendances Liées aux Clés Étrangères

**De Code\_Materiel vers les informations du client dans Client :**

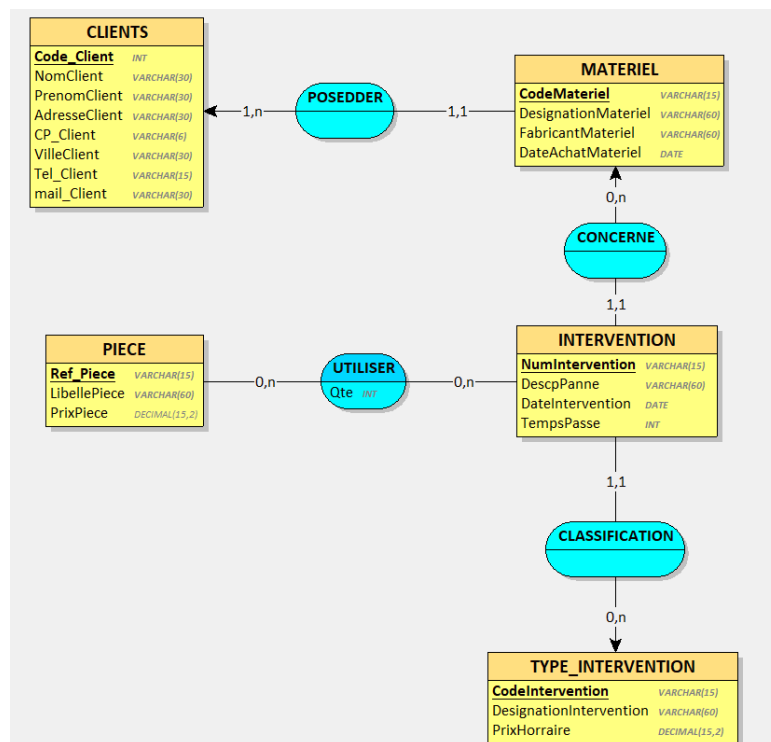
Code\_Matériel --> Désignation  
 --> Fabricant  
 --> date d'achat  
 --> CodeClient

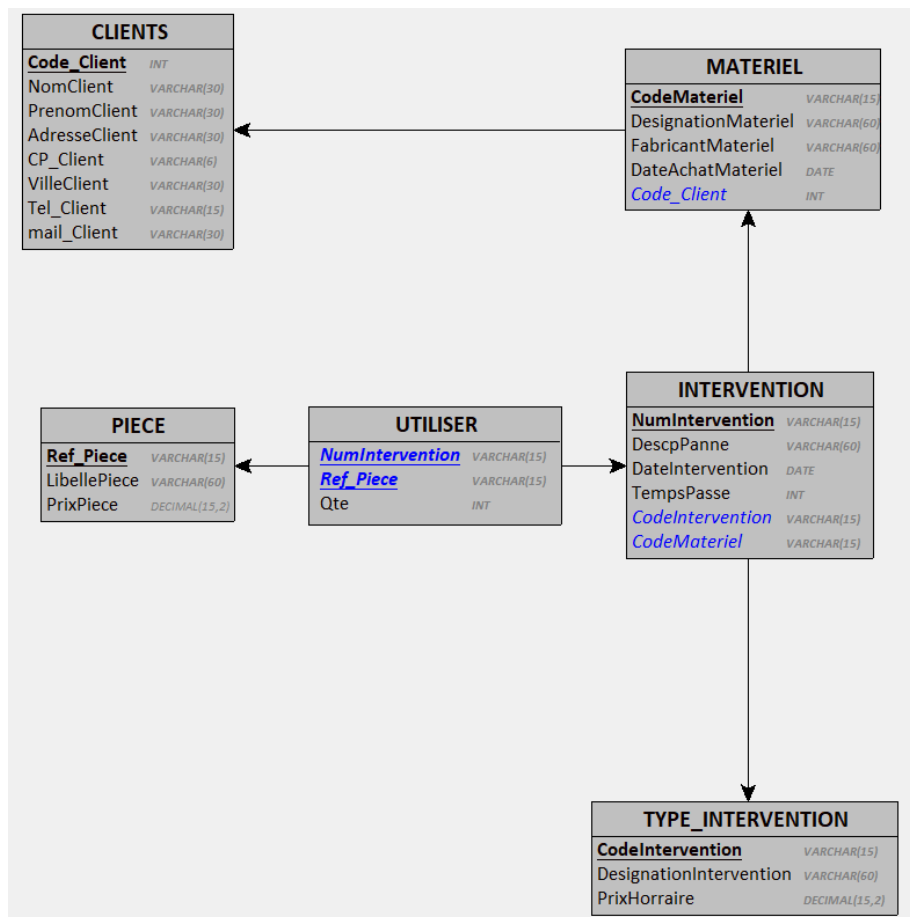
**De Intervention vers les informations du client dans Client :**

Num\_Intervention --> DescriptionPanne  
 --> Date\_Intervention  
 --> TempsPassé  
 --> Code\_Intervention  
 --> Code\_Matériel

**entre Intervention et Composant :**

Num\_Intervention, ID\_Composant --> Quantité utilisée





## Schéma relationnel des données :

**Client** (Code\_Client, Nom, Prénom, Adresse, CP, Ville, Téléphone, Email)

**Matériel** (Code\_Matériel, Désignation, Fabricant, Date d'achat, #Code\_Client)

**Intervention** (Num Intervention, Description\_panne, Date\_Intervention, Temps\_passé, #Code\_Matériel, #Code\_Intervention)

**Composant**(référence Composant, Désignation, Prix\_Achat)

**TypeIntervention** (Code Intervention, Désignation, Prix\_Horaire)

**Utiliser** (#référence Composant, #Num Intervention, Qté)

### Exercice 6 :

Vous êtes missionné par un de vos amis qui exerce la profession d'agent immobilier pour lui réaliser un petit programme. Il désire un logiciel dans lequel il peut inscrire son fichier des maisons, des propriétaires et des locataires avec les règles de gestion suivantes :

- Une maison appartient à une ou plusieurs personnes
- Une personne peut être propriétaire d'une maison et en louer une autre



- Concevoir le dictionnaire des données simplifié
- Donner les dépendances fonctionnelles associées
- Créer le modèle conceptuel des données
- Concevoir le modèle logique de données
- Finir par le modèle physique des données

### Solution :

## 1. Dictionnaire des Données Simplifié

Le dictionnaire des données identifie les entités (tables) et les attributs (colonnes) pour ce système de gestion immobilière.

Entité	Attribut	Description
<b>Personne</b>	ID_Personne	Identifiant unique d'une personne (propriétaire ou locataire)
	Nom	Nom de la personne
	Prénom	Prénom de la personne
<b>Maison</b>	ID_Maison	Identifiant unique d'une maison
	Adresse	Adresse de la maison
	Ville	Ville où se trouve la maison
<b>Posséder</b>	ID_Personne (FK)	Référence vers l'identifiant de la personne propriétaire
	ID_Maison (FK)	Référence vers l'identifiant de la maison possédée
<b>habiter</b>	ID_Personne (FK)	Référence vers l'identifiant du locataire
	ID_Maison (FK)	Référence vers l'identifiant de la maison louée

## 3. Modèle Conceptuel de Données (MCD)

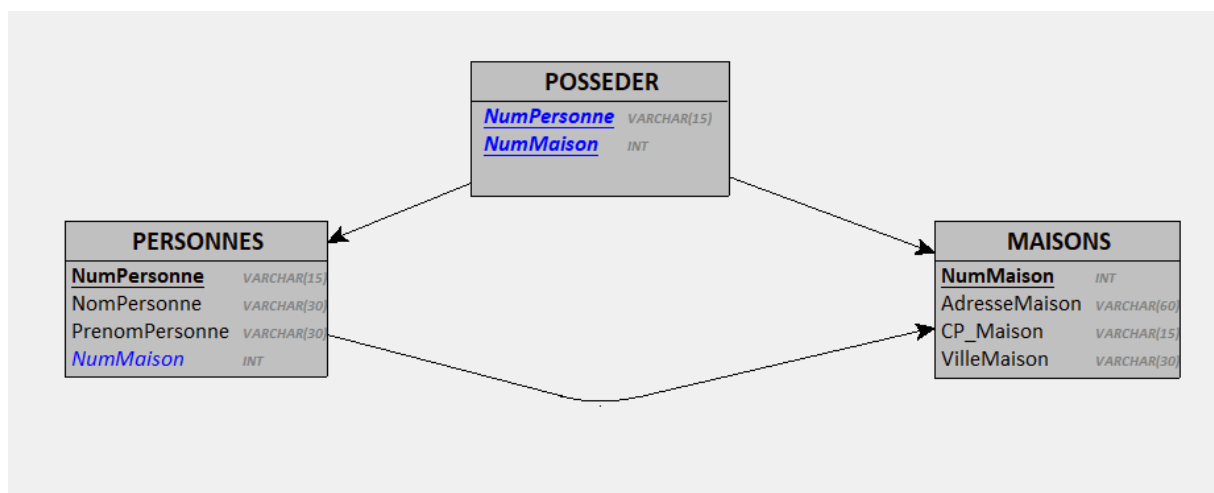
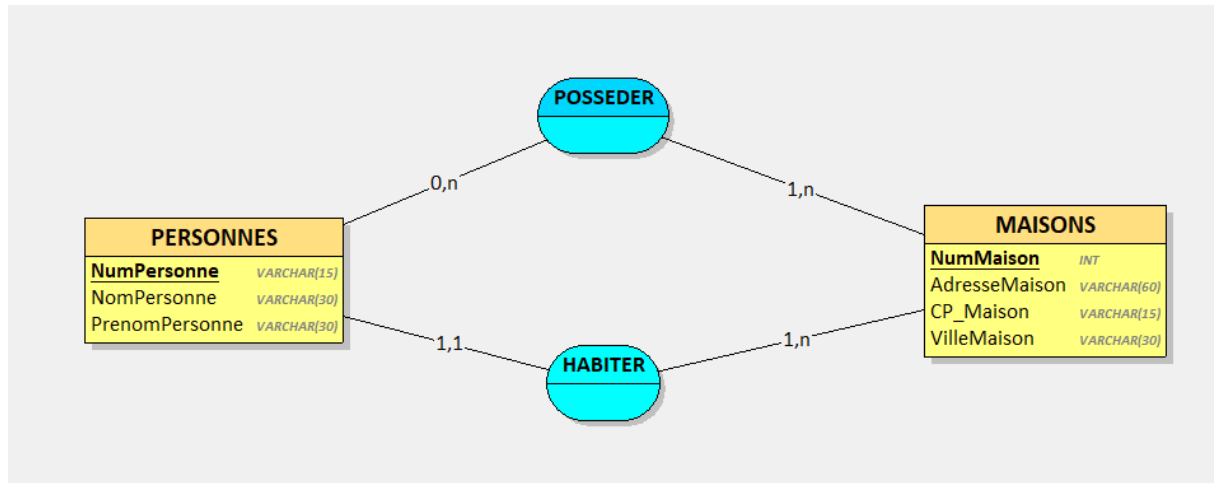
Le modèle conceptuel présente les entités, attributs et relations.

### Entités et Attributs :

- Personne**
  - ID\_Personne (PK)
  - Nom
  - Prénom
- Maison**
  - ID\_Maison (PK)
  - Adresse
  - Ville
- Posséder** (association entre *Personne* et *Maison*)
  - ID\_Personne (FK)
  - ID\_Maison (FK)
- habiter** (association entre *Personne* et *Maison*)
  - ID\_Personne (FK)
  - ID\_Maison (FK)

## Associations :

- **Personne - posséder** : Une personne peut être propriétaire de zéro ou plusieurs maisons (0,n) , et une maison peut appartenir à une ou plusieurs personnes (1,n)
- **Personne - habiter** : Une personne peut habiter une et une seule maison (1,1), et une maison peut être habitée par plusieurs personnes (1,n).



## 5. schéma relationnel :

**Personne**(ID\_Personne , Nom, Prénom, # ID\_Maison)

**Maison** (ID\_Maison, Adresse, Ville)

**Posseder** (#ID\_Personne, #ID\_Maison)

**habiter** (#ID\_Personne, #ID\_Maison)

## Exercice 7 :

ASSURAUTO comme son nom l'indique est une petite société d'assurance spécialisée dans les contrats d'assurance automobiles. Malgré son envergure restreinte (elle dispose tout de même de plusieurs agences et plusieurs employés sur le territoire, elle assure aussi bien les véhicules de tourisme que les véhicules utilitaires.

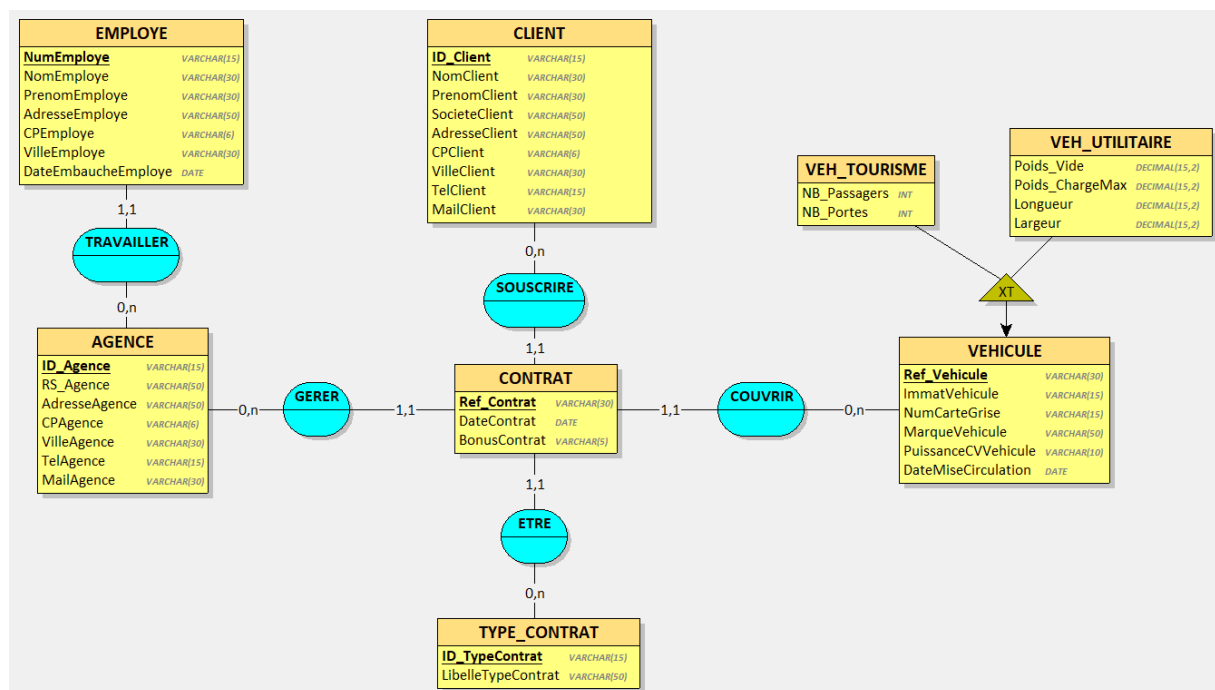
Pour assurer un véhicule, son propriétaire, dont on enregistre le nom, le prénom, l'adresse et les coordonnées (téléphone, mail, ...) doit fournir au conseiller de l'agence la carte grise du véhicule afin que l'on enregistre son type, sa marque, son numéro d'immatriculation, sa date de mise en circulation et sa puissance fiscale. Il s'agit d'un véhicule de tourisme, on enregistre aussi le nombre de porte et de passagers autorisés tandis que s'il s'agit d'un véhicule utilitaire on enregistre le poids à vide, le poids autorisé en charge, la longueur, la largeur.

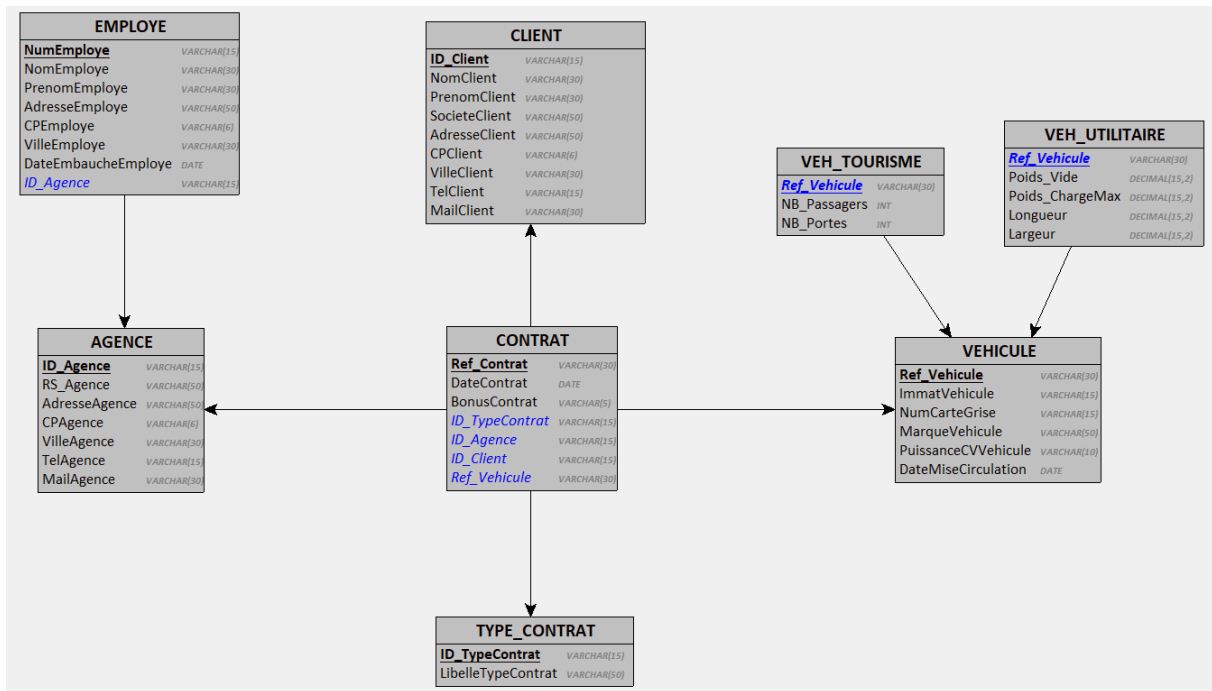
Chaque contrat, établie à une certaine date est référencé par un numéro de contrat et est d'une certaine catégorie : tout risque, au tiers, ...

Le contrat est attaché à la personne, pas au véhicule : lorsqu'il y a changement de véhicule le propriétaire concerne le bonus ou le malus attaché à ce contrat qui est alors reporté sur le nouveau véhicule.

- Concevoir le dictionnaire des données simplifié
- Donner les dépendances fonctionnelles associées
- Créer le modèle conceptuel des données
- Concevoir le model logique de données

### Solution :





Nouvelle notion d'héritage que vous allez approfondir un peu plus tard, ici Véhicule de tourisme et Utilitaires héritent des propriétés de l'entité Véhicule.

XT indique qu'un véhicule est soit utilitaire soit de tourisme mais pas les deux.

L'héritage est l'association entre deux entités permettant d'exprimer que l'une est plus générale que l'autre. L'héritage implique une transmission automatique des propriétés (attributs) d'une entité A à une entité A'. Dire que A' hérite de A équivaut à dire que A' est une sous-entité de A. On peut également dire que A est une généralisation de A' et que A' est une spécialisation de A. Outre qu'il permet de représenter une relation courante dans le monde réel, l'héritage a un avantage pratique, celui de factoriser la définition de propriétés identiques pour des entités proches.

**AGENCE** = (ID\_Agence VARCHAR(15), RS\_Agence VARCHAR(50), AdresseAgence VARCHAR(50), CPAgence VARCHAR(6), VilleAgence VARCHAR(30), TelAgence VARCHAR(15), MailAgence VARCHAR(30));

**CLIENT** = (ID\_Client VARCHAR(15), NomClient VARCHAR(30), PrenomClient VARCHAR(30), SocieteClient VARCHAR(50), AdresseClient VARCHAR(50), CPClient VARCHAR(6), VilleClient VARCHAR(30), TelClient VARCHAR(15), MailClient VARCHAR(30));

**VEHICULE** = (Ref\_Vehicule VARCHAR(30), ImmatVehicule VARCHAR(15), NumCarteGrise VARCHAR(15), MarqueVehicule VARCHAR(50), PuissanceCVVehicule VARCHAR(10), DateMiseCirculation DATE);

**TYPE\_CONTRAT** = (ID\_TypeContrat VARCHAR(15), LibelleTypeContrat VARCHAR(50));

**VEH\_TOURISME** = (#Ref\_Vehicule, NB\_Passagers INT, NB\_Portes INT);

**VEH\_UTILITAIRE** = (#Ref\_Vehicule, Poids\_Vide DECIMAL(15,2), Poids\_ChargeMax DECIMAL(15,2), Longueur DECIMAL(15,2), Largeur DECIMAL(15,2));

**EMPLOYE** = (NumEmploye VARCHAR(15), NomEmploye VARCHAR(30), PrenomEmploye VARCHAR(30), AdresseEmploye VARCHAR(50), CPEmploye VARCHAR(6), VilleEmploye VARCHAR(30), DateEmbaucheEmploye DATE, #ID\_Agence);

**CONTRAT** = (Ref\_Contrat VARCHAR(30), DateContrat DATE, BonusContrat VARCHAR(5), #ID\_TypeContrat, #ID\_Agence, #ID\_Client, #Ref\_Vehicule);

### Exercice 8 :

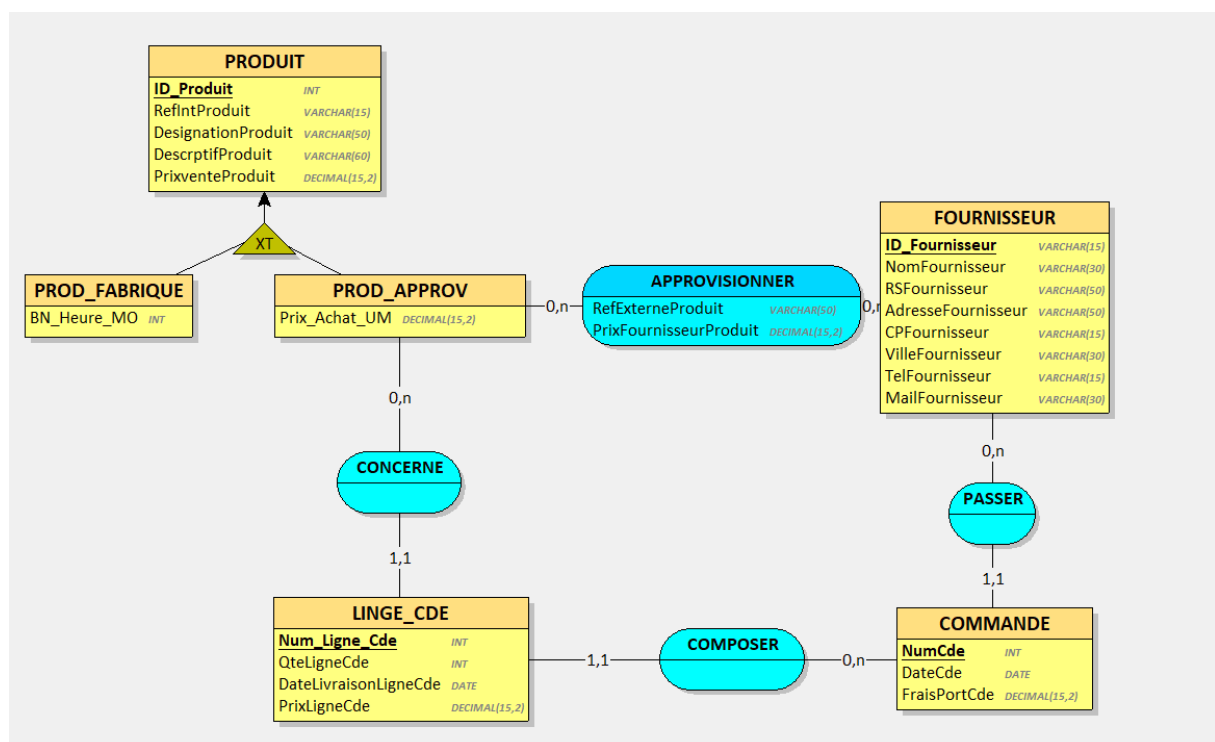
L'entreprise XProd fabrique et commercialise divers produits. Ils sont identifiés par une référence propre à XProd et on enregistre une désignation (libellé court), un descriptif (libellé long) et un prix de vente catalogue unitaire hors taxes. Dans la base de données, elle gère deux types de produits :

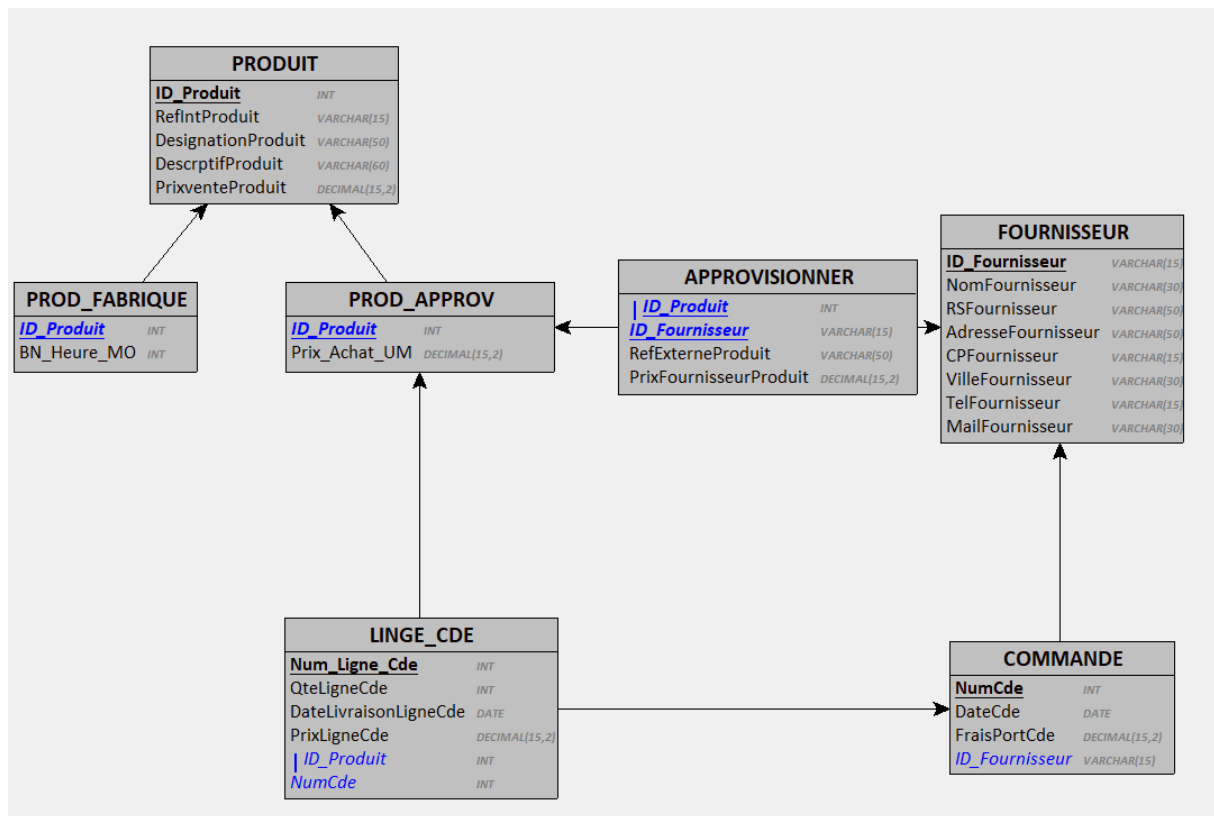
Les produits qu'elle fabrique pour lesquels on enregistre le nombre moyen d'heures de main d'œuvre nécessaire à leur fabrication. Les produits dits approvisionnés parce qu'elle ne les fabrique pas : ils sont achetés à un ou plusieurs fournisseurs à un prix d'achat unitaire moyen.

Pour ne pas dépendre d'un fournisseur, enregistré par sa raison sociale adresse, ... pour chaque produit approvisionné l'entreprise a établi une liste de fournisseurs capables de livrer ce produit. Bien entendu pour un même produit chaque fournisseur peut avoir sa propre référence et un prix différent. Lorsque XProd passe une commande à une certaine date fournisseur, elle essaie de grouper plusieurs lignes de commande : une par produit dans une certaine quantité avec sa date de livraison prévue, pour réduire les frais de livraison de la commande et essayer de négocier un prix d'achat unitaire inférieur au prix catalogue du fournisseur.

- Concevoir le dictionnaire des données simplifié
- Donner les dépendances fonctionnelles associées
- Créer le modèle conceptuel des données
- Concevoir le model logique de données

### Solution :





**PRODUIT** = (ID\_Produit INT, RefIntProduit VARCHAR(15), DesignationProduit VARCHAR(50), DescrptifProduit VARCHAR(60), PrixventeProduit DECIMAL(15,2));

**PROD\_FABRIQUE** = (#ID\_Produit, BN\_Heure\_MO INT);

**PROD\_APPROV** = (#ID\_Produit, Prix\_Achat\_UM DECIMAL(15,2));

**FOURNISSEUR** = (ID\_Fournisseur VARCHAR(15), NomFournisseur VARCHAR(30), RSFournisseur VARCHAR(50), AdresseFournisseur VARCHAR(50), CPFournisseur VARCHAR(15), VilleFournisseur VARCHAR(30), TelFournisseur VARCHAR(15), MailFournisseur VARCHAR(30));

**COMMANDE** = (NumCde INT, DateCde DATE, FraisPortCde DECIMAL(15,2), #ID\_Fournisseur);

**LINGE\_CDE** = (Num\_Ligne\_Cde INT, QteLigneCde INT, DateLivraisonLigneCde DATE, PrixLigneCde DECIMAL(15,2), #ID\_Produit, NumCde);

**APPROVISIONNER** = (#ID\_Produit, ID\_Fournisseur, RefExterneProduit VARCHAR(50), PrixFournisseurProduit DECIMAL(15,2));

### Exercice 9 :

Les fédérations de sport proposent des compétitions composées de plusieurs sports ou épreuves comme le biathlon, triathlon et autre décathlon vous ont demandé de développer un logiciel générique pouvant gérer l'organisation de leurs compétitions. Voici quelques éléments vous permettant de commencer.

- Les sportifs s'inscrivent à une compétition.
- Lors de cette inscription on enregistre son nom, son prénom, son adresse et ses coordonnées.
- Il se voit attribuer un numéro de dossard dans cette compétition, qui servira à retrouver son dossier d'insertion
- Un sportif peut être licencié à la fédération via un club ou pas
- Les amateurs sont parfois autorisés à courir

C'est pourquoi pour un sportif licencié on enregistre bien sur son numéro de licence et son club, tandis que pour le sportif amateur on exige un certificat médical daté d'au de mois de 3 mois délivré par un médecin pour des raison d'assurance

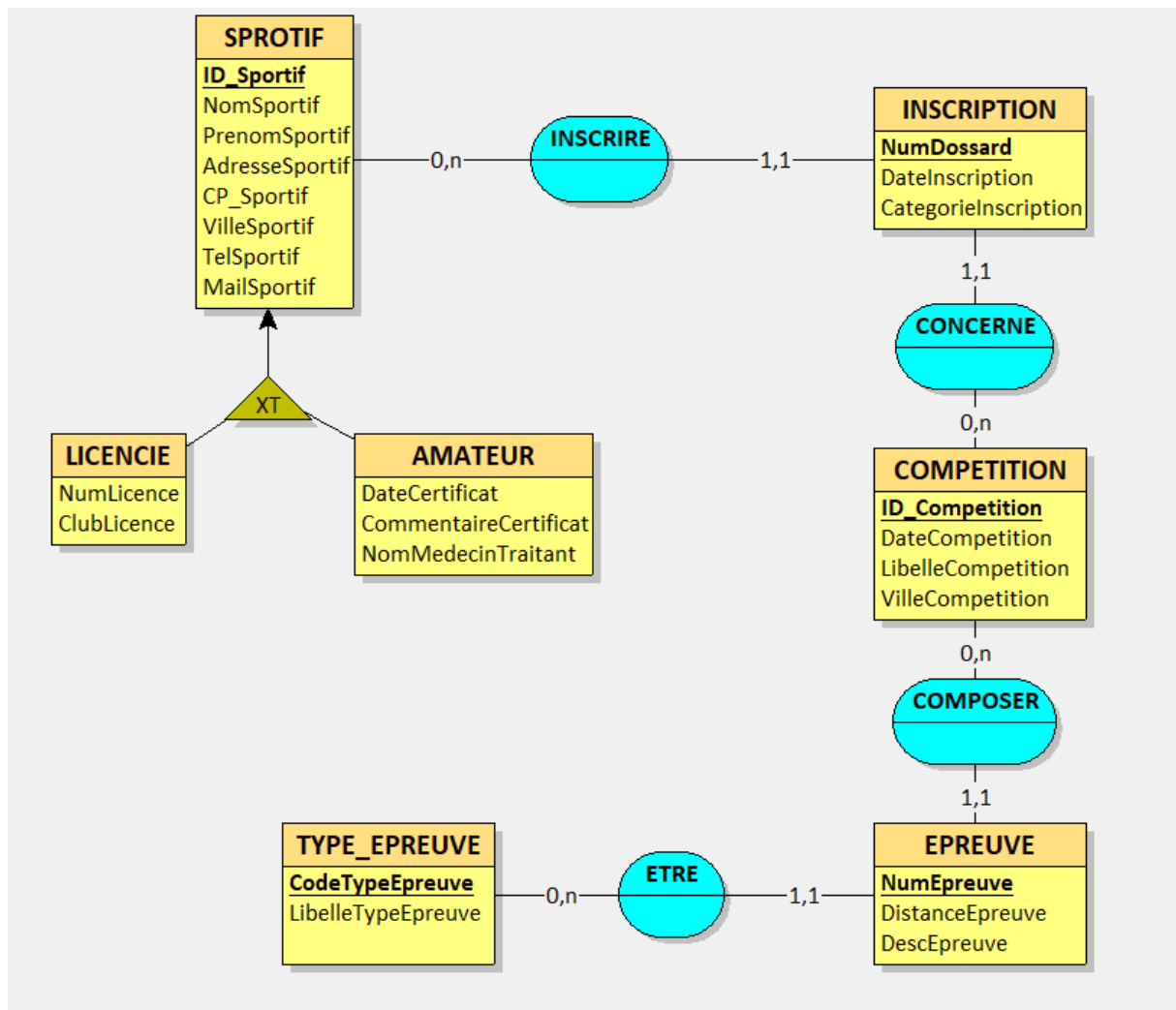
- Une compétition a lieu à une certaine date dans une certaines ville et porte éventuellement un libellé comme « grand prix de printemps »

Chaque compétition est composée d'un certain nombre de d'épreuves effectuées dans un certain ordre : pour certaines compétitions il y a d'abord une épreuve de 3 km de nation suivi de 50 km de vélo et enfin 20 km de course à pied, pour d'autres cela commence par une épreuve d'escalade d'un mur de niveau 3, suivi d'une randonnée pédestre de 10 km et se termine par un parcours en traineau tiré par des chaines.

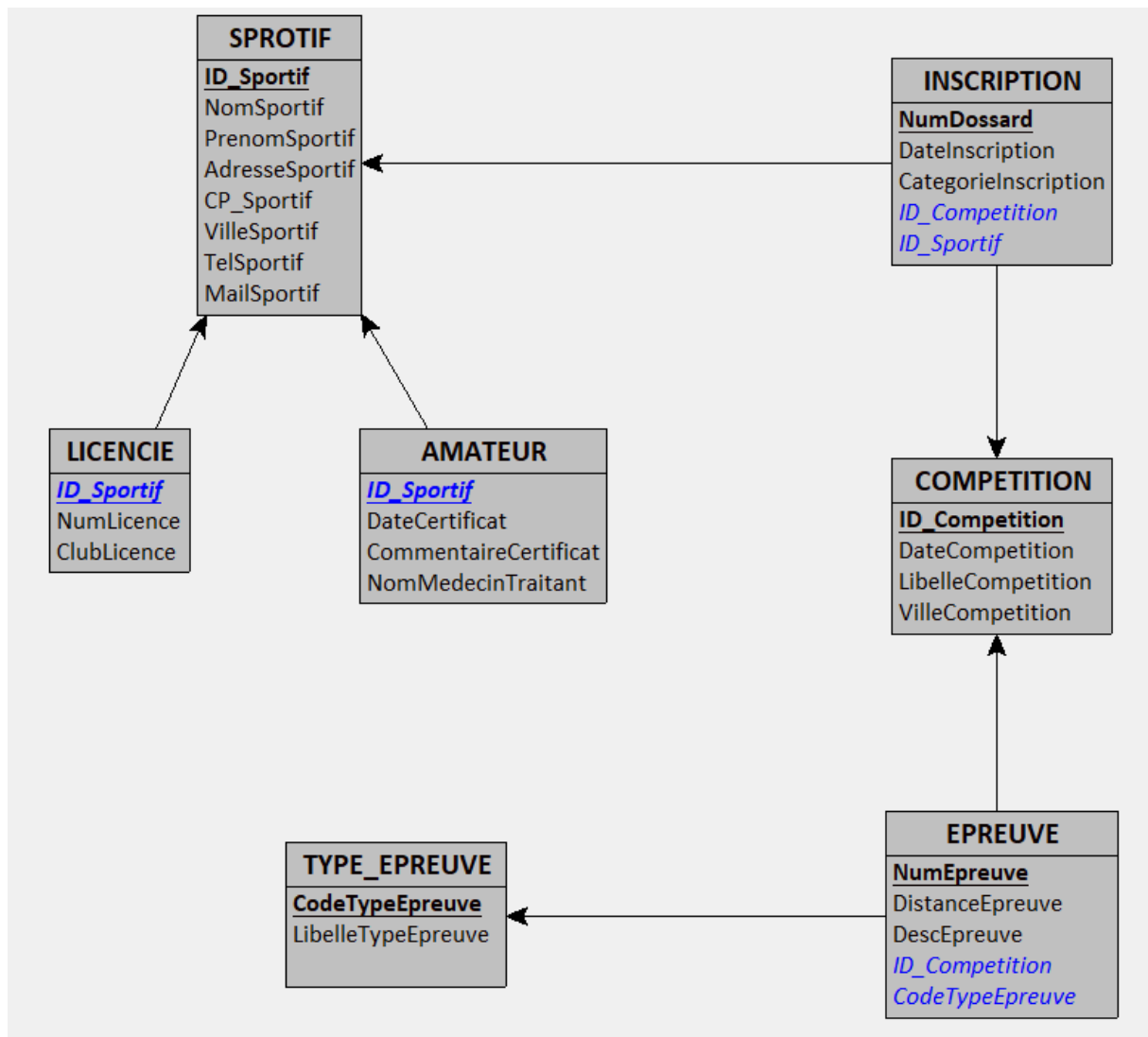
- Chaque épreuve est d'un certain type et il faut spécifier alors sa distance et les conditions de réalisation.
- Concevoir le dictionnaire des données simplifié
- Donner les dépendances fonctionnelles associées
- Créer le modèle conceptuel des données
- Concevoir le model logique de données
- Finir par le modèle physique des données

#### **Solution :**

Nom de la données	Format	Longueur
Nom du sportif	Alphabétique	30
Prénom du sportif	Alphabétique	20
Adresse	Alphanumérique	60
Code Postal	Alphanumérique	5
Ville	Alphanumérique	60
Téléphone	Alphanumérique	15
Mail	Alphanumérique	30
Numéro de licence	Alphanumérique	10
Club	Alphanumérique	60
Certificat médical	Alphanumérique	80
Date Certificat médical	Date	8
Médecin Traitant	Alphabétique	30
Date d'inscription	Date	8
Catégorie de l'inscription	Alphanumérique	20
Date Compétition	Date	8
Libelle Compétition	Alphanumérique	40
Ville Compétition	Alphanumérique	60
Distance Epreuve	Numérique	6
Descriptif Epreuve	Alphanumérique	60
Libellé Epreuve	Alphanumérique	60







**SPROTIF** = (ID\_Sportif INT, NomSportif VARCHAR(30), PrenomSportif VARCHAR(20), AdresseSportif VARCHAR(60), CP\_Sportif VARCHAR(5), VilleSportif VARCHAR(60), TelSportif VARCHAR(15), MailSportif VARCHAR(40));

**LICENCE** = (#ID\_Sportif, NumLicence VARCHAR(10), ClubLicence VARCHAR(60));

**AMATEUR** = (#ID\_Sportif, DateCertificat DATE, CommentaireCertificat VARCHAR(80), NomMedecinTraitant VARCHAR(30));

**COMPETITION** = (ID\_Competition VARCHAR(5), DateCompetition DATE, LibelleCompetition VARCHAR(40), VilleCompetition VARCHAR(60));

**TYPE\_EPREUVE** = (CodeTypeEpreuve VARCHAR(5), LibelleTypeEpreuve VARCHAR(80));

**INSCRIPTION** = (NumDossard VARCHAR(5), DateInscription DATE, CategorieInscription VARCHAR(20), #ID\_Competition, #ID\_Sportif);

**EPREUVE** = (NumEpreuve VARCHAR(5), DistanceEpreuve INT, DescEpreuve VARCHAR(60), #ID\_Competition, #CodeTypeEpreuve);

#### Exercice 10 :

Voici un model relationnel décrivant une nomenclature de conception d'un meuble. Le meuble est un ensemble composé de sous-ensemble et de composants divers. Un sous ensemble est élaboré grâce à un assemblage de composants.

A partir de ce model relationnel, il vous est demandé de remonter jusqu'au modèle conceptuel en passant par le model logique de données.

Ensemble (**CodeEnsemble**, désignation)

Sous-ensemble (**CodeSousEnsemble**, désignation, longueur, largeur, hauteur, PrixUnitaire)

Composants (**CodeComposant**, désignation, prix unitaire)

LienEnsSE(**#CodeEnsemble**, **#CodeSousEnsemble**, qté)

lienEnsComposant(**#CodeEnsemble**, **#CodeComposant**, Qté)

LienSEComposant(**#CodeSousEnsemble**, **#CodeComposant**, Qté)

### Solution :

Pour cette tâche, nous allons procéder de manière structurée en remontant du modèle relationnel donné jusqu'au modèle conceptuel, en passant par le modèle logique.

## 1. Modèle Relationnel (Donné)

Le modèle relationnel fourni est le suivant :

- **Ensemble** (CodeEnsemble, désignation)
- **Sous-ensemble** (CodeSousEnsemble, désignation, longueur, largeur, hauteur, PrixUnitaire)
- **Composants** (CodeComposant, désignation, prix unitaire)
- **LienEnsSE** (#CodeEnsemble, #CodeSousEnsemble, Qté)
- **lienEnsComposant** (#CodeEnsemble, #CodeComposant, Qté)
- **LienSEComposant** (#CodeSousEnsemble, #CodeComposant, Qté)

## 2. Modèle Logique de Données

Le **Modèle Logique de Données** est une représentation intermédiaire entre le conceptuel et le relationnel. Il détaille les tables et leurs relations, tout en précisant les clés primaires et étrangères.

### Tables et attributs :

1. **Table Ensemble :**
  - Clé primaire : CodeEnsemble
  - Attributs : CodeEnsemble, désignation
2. **Table Sous-ensemble :**
  - Clé primaire : CodeSousEnsemble
  - Attributs : CodeSousEnsemble, désignation, longueur, largeur, hauteur, PrixUnitaire
3. **Table Composants :**
  - Clé primaire : CodeComposant
  - Attributs : CodeComposant, désignation, PrixUnitaire
4. **Table LienEnsSE** (Relation entre un ensemble et des sous-ensembles) :

- Clé étrangère : CodeEnsemble, CodeSousEnsemble
- Attributs : Qté
- Relation entre :
  - CodeEnsemble (FK vers Ensemble)
  - CodeSousEnsemble (FK vers Sous-ensemble)
- 5. **Table lienEnsComposant** (Relation entre un ensemble et des composants) :
  - Clé étrangère : CodeEnsemble, CodeComposant
  - Attributs : Qté
  - Relation entre :
    - CodeEnsemble (FK vers Ensemble)
    - CodeComposant (FK vers Composants)
- 6. **Table LiensEComposant** (Relation entre un sous-ensemble et des composants) :
  - Clé étrangère : CodeSousEnsemble, CodeComposant
  - Attributs : Qté
  - Relation entre :
    - CodeSousEnsemble (FK vers Sous-ensemble)
    - CodeComposant (FK vers Composants)

### Clés primaires et étrangères :

- Chaque table possède sa clé primaire.
- Les tables de liens contiennent des clés étrangères qui font référence aux entités correspondantes.
- Les attributs supplémentaires (comme Qté) permettent de modéliser les quantités associées à chaque relation.

## 3. Modèle Conceptuel de Données (MCD)

Le **Modèle Conceptuel de Données (MCD)** traduit le modèle logique en entités et associations avec les cardinalités correspondantes. Il permet de visualiser le système à un niveau plus abstrait.

### Entités :

1. **Ensemble**
  - Attributs :
    - CodeEnsemble (PK)
    - désignation
2. **Sous-ensemble**
  - Attributs :
    - CodeSousEnsemble (PK)
    - désignation
    - longueur
    - largeur
    - hauteur
    - PrixUnitaire
3. **Composants**
  - Attributs :
    - CodeComposant (PK)
    - désignation
    - PrixUnitaire

## Associations :

1. **Ensemble - Sous-ensemble** (relation modélisée par `LienEnsSE`) :
  - Un ensemble peut être composé de plusieurs sous-ensembles (0,N).
  - Un sous-ensemble peut appartenir à plusieurs ensembles (0,N).
  - Attribut de l'association : `Qté` (quantité de sous-ensembles dans l'ensemble).
2. **Ensemble - Composants** (relation modélisée par `lienEnsComposant`) :
  - Un ensemble peut directement contenir plusieurs composants (0,N).
  - Un composant peut être utilisé dans plusieurs ensembles (0,N).
  - Attribut de l'association : `Qté` (quantité de composants dans l'ensemble).
3. **Sous-ensemble - Composants** (relation modélisée par `LienSEComposant`) :
  - Un sous-ensemble est constitué de plusieurs composants (0,N).
  - Un composant peut être utilisé dans plusieurs sous-ensembles (0,N).
  - Attribut de l'association : `Qté` (quantité de composants dans le sous-ensemble).

