

Modèle Conceptuel de Données

MCD

Christian Bonhomme

CNAM 2015-2016

Définition

Une **Base de Données** (ou *database*) est un système d'informations permettant de stocker et de retrouver des données brutes en rapport avec un thème ou une activité.

Ces données sont stockées, gérées et utilisées par un Système de Gestion de Base de Données (SGBD).

En pratique un SGBD est un programme serveur situé sur une machine qui exploite des bases de données locales et/ou distantes. Ce programme serveur est sollicité par des applications clientes qui en extraient de l'information au moyen de *requêtes*.

Historique

1959 : Naissance du groupement américain CODASYL (*Conference On DATA System Languages*) ;

En 1965, Charles Bachman conçoit l'architecture Ansi/Sparc encore utilisée de nos jours (trois niveaux : schéma interne (SI), schéma conceptuel (SC) et schémas externes (SE) ;

1967 : CODASYL donne naissance à une entité nommée DBTG (*DataBase Task Group*) ;

1968 : Dick Pick crée Pick, un système d'exploitation contenant un système de gestion de base de données « multivaluée » ;

1970 : Edgar Franck Codd (IBM), dans sa thèse sur l'algèbre relationnelle, jette les bases du modèle relationnel des bases de données ;

1971 : DBTG rédige les spécifications d'un langage de manipulation et de définition des données. (DML et DDL) pour la normalisation d'un modèle de base de données en réseau ;

1974 : Naissance du langage SQL (*Structured Query Langage*) ;

1975 : Peter Chen invente le modèle entité-association ;

1979 : Commercialisation de la première base de données relationnelle, Oracle ;

1986 : SQL est normalisé ;

1992 : Norme SQL-2, appelée aussi SQL-92.

1995 : Sortie de la première version de MySQL ;

1999 : Norme SQL-3, appelée aussi SQL-99.

Mise en oeuvre

- Analyse de la situation existante et des besoins ;
- Création d'une série de modèles conceptuels, qui permettent de représenter tous les aspects du métier et des besoins ;
- A partir des modèles, implémentation d'une base de données.

Analyse

L'analyse de l'existant et des besoins se fait en interviewant les différents acteurs et futurs utilisateurs de la base données.

Cette étape est la plus délicate, car c'est d'elle que dépendra la pertinence de la base données et de son utilisation.

Elle repose sur la qualité des interviews et l'expertise d'analyse du concepteur.

Cette analyse se traduit en général par la rédaction d'un Cahier Des Charges (CDC).

Modèle Conceptuel de Données (MCD)

A partir de l'analyse de l'existant et des besoins de l'entreprise ou du métier, le MCD représente de façon structurée les données qui seront utilisées par le système d'information.

Le MCD décrit la **sémantique**, c'est à dire le sens attaché à ces données et à leurs rapports et non à l'utilisation qui peut en être faite.

Le MCD représente graphiquement les entités et leurs relations sous forme de diagrammes.

Modèle Logique de Données (MLD)

Le MLD découle du MCD, mais en précisant la volumétrie, la structure et l'organisation des données.

Le MLD est exprimé dans un formalisme logique adapté au SGBD envisagé.

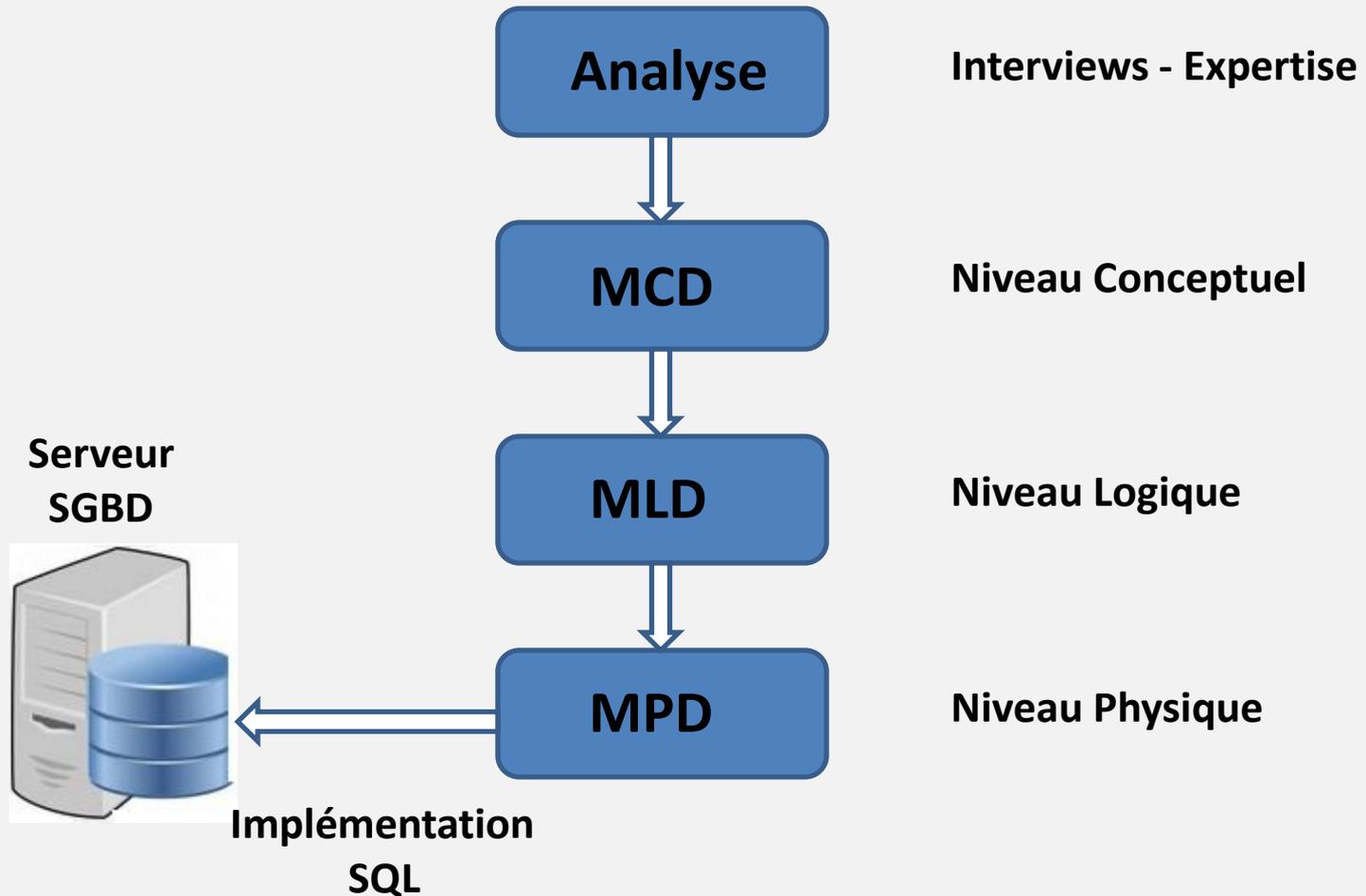
Le MLD sera ensuite transformé et adapté en fonction des spécificités du langage de définition des données spécifique à l'outil retenu pour devenir un Modèle Physique de Données (MPD).

Modèle Physique de Données (MPD)

Le MPD consiste à implémenter le MLD dans un SGBD.

Le langage généralement utilisé pour ce type d'implémentation est SQL (Structured Query Language).

Conception d'une Base de Données



MCD : Définitions

Entité : Objet pourvu d'une existence propre.

Catégories, Produits, Clients sont des entités.

Association : Relation entre entités.

Achète est une relation entre l'entité Client et l'entité Produits.

Attributs ou propriétés : Plus petit élément d'information d'une entité ou d'une association.

Nom, Référence, Prix, etc. sont des attributs de l'entité Produits.

Instance ou occurrence : Valeur d'un attribut.

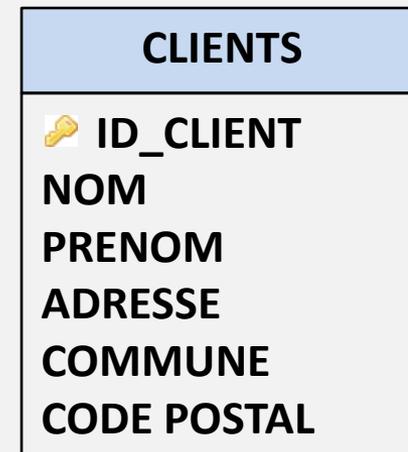
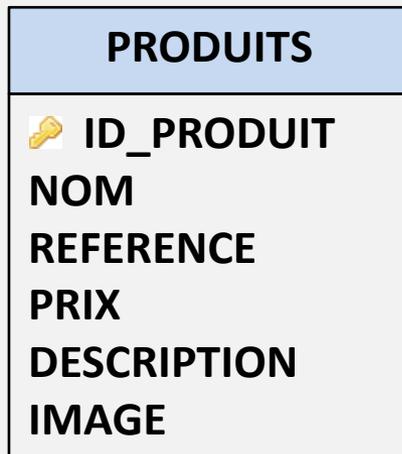
Fleur est une instance de l'attribut Nom.

Ref234 est une instance de l'attribut Référence.

24 est une instance de l'attribut Prix.

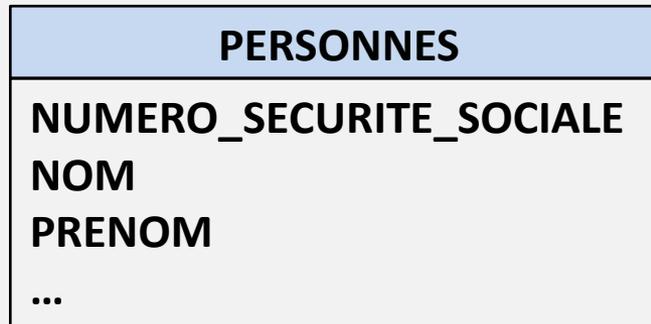
MCD : Entité

Une entité a une existence propre et est décrit au moins par un **identifiant** et possède généralement une liste d'attributs qui décrit ses caractéristiques.



MCD : identifiant d'une entité

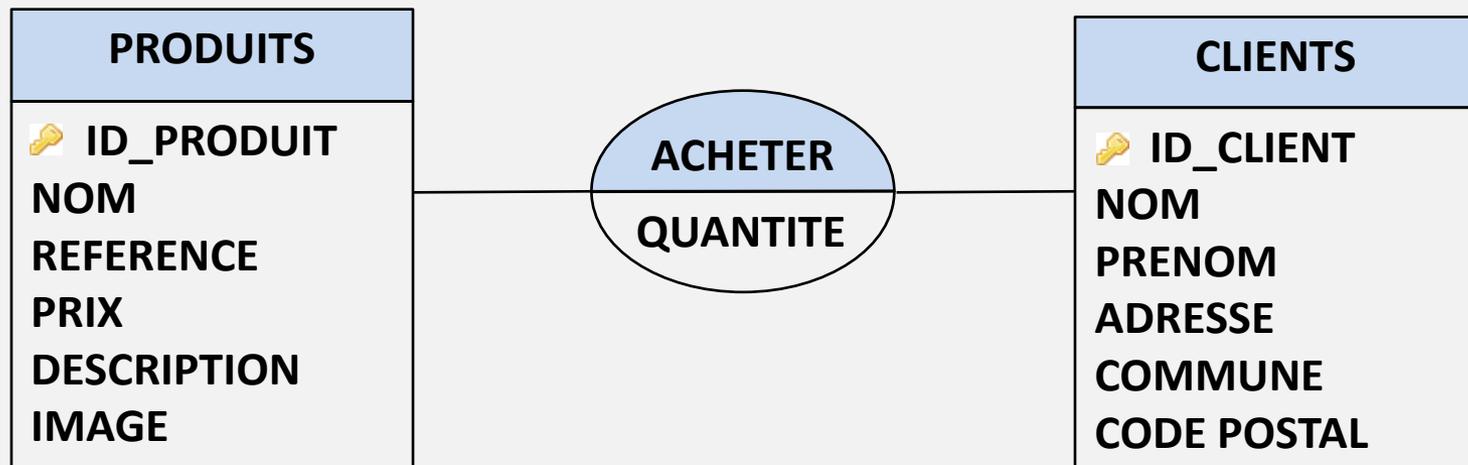
L'identifiant ou clef d'une entité est un attribut qui identifie les caractéristiques de l'entité, c'est-à-dire les valeurs de ses attributs.



MCD : associations

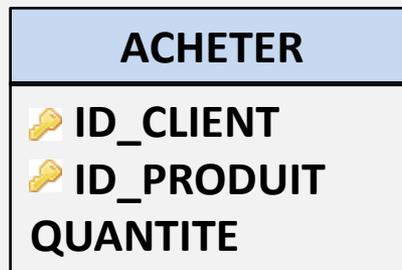
Une association ou une relation décrit le lien existant entre deux entités ou plus.

Elle est dépourvue d'existence propre et peut être porteuse d'attributs ou non.



MCD : identifiant d'une association

L'identifiant d'une association est l'identifiant obtenu en ajoutant ou en concaténant les identifiants des entités reliées par l'association.



MCD : Instances

L'**instance d'un attribut** est la valeur prise par cet attribut.

L'**instance d'une entité** est caractérisée par l'ensemble des valeurs de ces attributs.

L'**instance d'une association** est l'ensemble formé par l'instance de chacune des entités et une instance de chacune des éventuelles entités de l'association.

MCD : types d'associations

Le type d'une association est déterminé par le lien qui relie deux entités.

Ces liens n'ont pas d'existence propre, ils n'existent que si des entités existent.

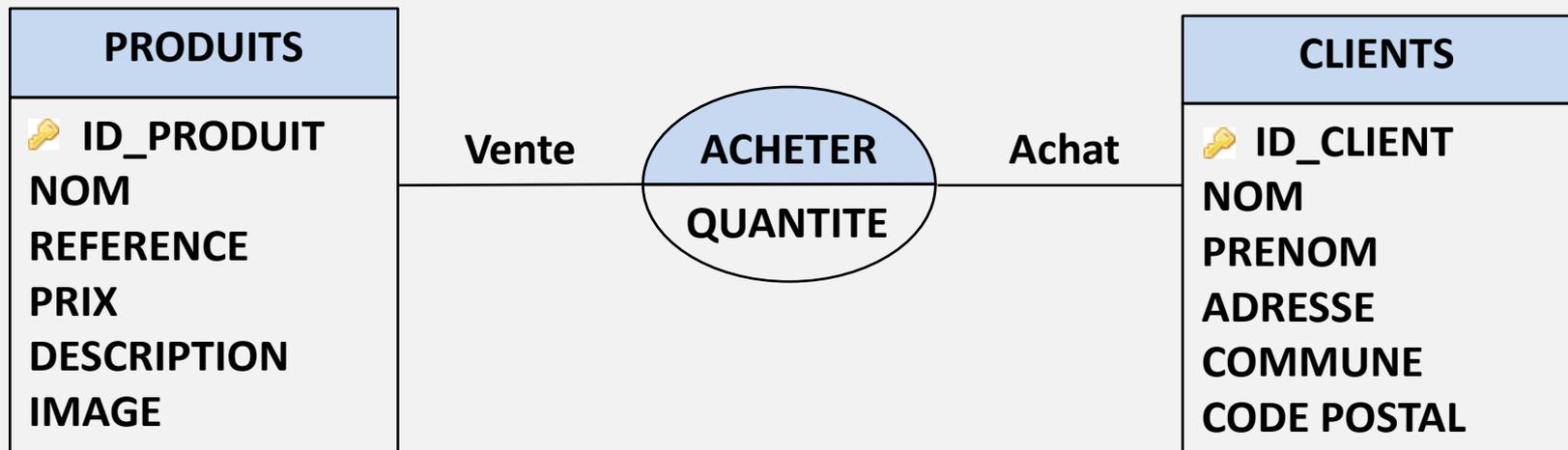
Quel est le lien qui unit un client à un produit ?
c'est l'action d'**acheter**.

Des types d'associations ne peuvent pas être liés entre eux, et deux types d'entités peuvent être liés par des types d'associations différents.

MCD : rôle d'un type d'associations

Un type d'entités qui intervient dans un type d'associations y joue un **rôle**.

On peut donner un nom à ce rôle mais si toutefois aucun nom n'est donné alors par défaut ce sera celui du type d'entités situé à cette extrémité.



MCD : Cardinalité

La **cardinalité** d'un lien entre une entité et une association indique le minimum et le maximum de fois qu'une entité peut être en relation avec l'association.

La **cardinalité** minimale est de 0 ou 1.

La **cardinalité** maximale est N.

Un client a acheté au moins un produit.

Un client peut avoir acheter plusieurs produits.

Un produit peut ne pas être acheté

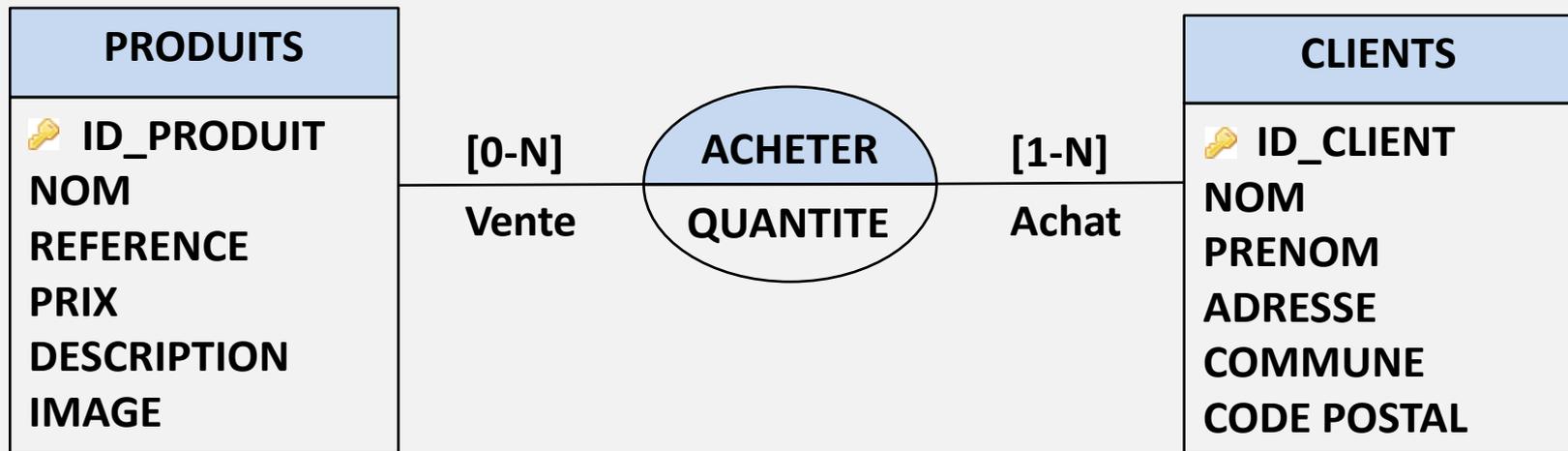
Un produit peut être acheté par plusieurs clients.

MCD : Cardinalité

La **cardinalité** d'un lien entre une entité et une association indique le minimum et le maximum de fois qu'une entité peut être en relation avec l'association. Elle est notée sous la forme **[min-max]**.

La **cardinalité** minimale est de 0 ou 1.

La **cardinalité** maximale est N.



MCD : contraintes d'intégrité

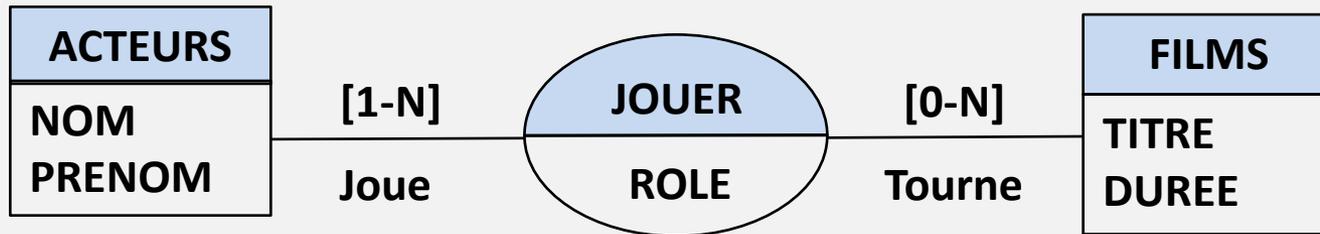
Ces contraintes assurent à tout moment la cohérence logique des données au sein de notre base de données.

Les contraintes de base :

- les identifiants ;
- les attributs obligatoires ;
- les contraintes de cardinalité des rôles.

Lire un MCD

Prenons le MCD simple suivant :

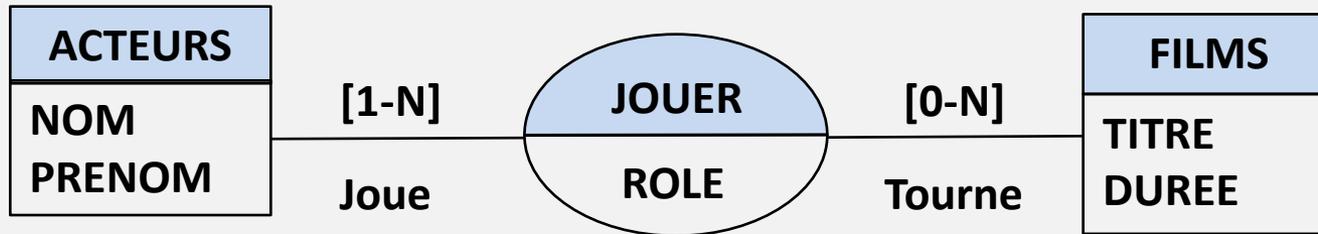


Comment lire ce schéma ?

Nous avons :

- 2 entités : **ACTEURS** et **FILMS** ;
- 1 association : **JOUER**.

Lire un MCD



L'entité **ACTEURS** contient 2 attributs : **NOM** et **PRENOM**.

L'entité **FILMS** contient 2 attributs : **TITRE** et **DUREE**.

L'association contient un attribut : **ROLE**.

Que signifie la cardinalité **Joue [1-N]** ? Nous pouvons la lire de la manière suivante :

Un acteur joue dans un ou plusieurs films.

Que signifie la cardinalité **Tourne [0-N]** ? Nous pouvons la traduire par :

0 ou N acteurs tourne dans un film.

Ecrire un MCD

A présent partons d'une phrase pour bâtir notre schéma. Soit :
Un réalisateur réalise obligatoirement un ou plusieurs films alors qu'un film est réalisé obligatoirement par un seul réalisateur.

Nous avons :

- l'entité **REALISATEURS** qui contiendra les attributs : **NOM** et **PRENOM** ;
- L'association **REALISER** qui reliera l'entité **REALISATEURS** à l'entité **FILMS**.

La liaison entre l'association **REALISER** et l'entité **FILMS** pourrait se nommer **Réalisation** et sa cardinalité sera **[1,1]** (*un film est réalisé par un seul réalisateur*).

La liaison entre l'association **REALISER** et l'entité **REALISATEURS** pourrait se nommer **Tournage** et sa cardinalité sera **[1,N]** (*un réalisateur réalise un ou plusieurs films*).

Ecrire un MCD

De l'analyse précédente, nous en déduisons le schéma suivant :



Ecrire un MCD

Soit la nouvelle phrase :

Un film est programmé ou non à une date donnée dans des cinémas tandis qu'un cinéma programme à une date donnée au moins un film.

Nous avons :

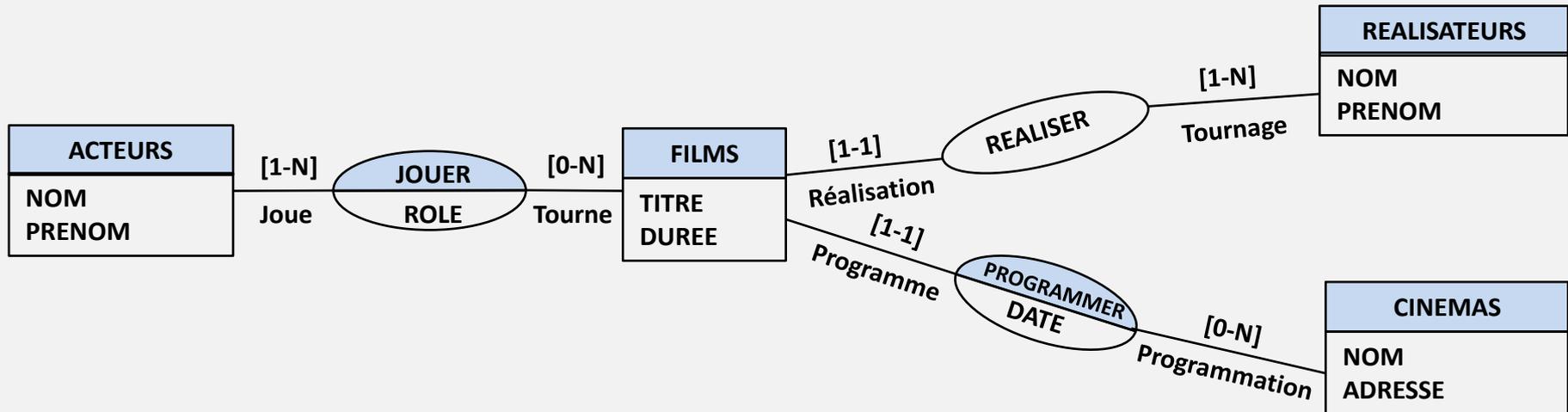
- l'entité **CINEMAS** qui contiendra les attributs : **NOM** et **ADRESSE** ;
- L'association **PROGRAMMER** qui reliera l'entité **CINEMAS** à l'entité **FILMS** contiendra un attribut **DATE**.

La liaison entre l'association **PROGRAMMER** et l'entité **FILMS** pourrait se nommer **Programme** et sa cardinalité sera **[1,1]** (*un cinéma programme à une date donnée au moins un film*).

La liaison entre l'association **PROGRAMMER** et l'entité **CINEMAS** pourrait se nommer **Programmation** et sa cardinalité sera **[0,N]** (*un film est programmé ou non à une date donnée dans des cinémas*).

Ecrire un MCD

Des différentes analyses nous en déduisons le MCD suivant :



MCD : règles de normalisation

L'objectif de la normalisation est de construire un MCD cohérent.

Pour obtenir un MCD cohérent, il faut éliminer la redondance de l'information au sein de chaque attribut et chaque entité.

Les données redondantes entraînent un gaspillage d'espace disque et créent très souvent des problèmes de maintenance.

MCD : première forme normale (1NF)

Une entité est dite de première forme normale, si tous les attributs de la relation contiennent une valeur atomique.

Par exemple :

GARAGES	
NOM	VOITURE
Piboule	Zoé, Scénic

L'entité **GARAGES** n'est pas de première forme normale, l'attribut **VOITURE** n'est pas atomique.

MCD : première forme normale (1NF)

GARAGES	
NOM	VOITURE
Piboule	Zoé, Scénic

Pour passer en première forme normale, il faut atomiser les données de l'attribut **VOITURE** comme suit:

GARAGES	
NOM	VOITURE
Piboule	Zoé
Piboule	Scénic

MCD : deuxième forme normale (2NF)

Une entité est dite de deuxième forme normale, si :

- elle est de première forme normale ;
- chaque attribut qui n'appartient pas à la clé ne dépend pas uniquement d'une partie de la clé.

Pour cet exemple, supposons que la clé de l'entité **GARAGES** soit une clé composite (**NOM-VOITURE**)

GARAGES		
NOM	VOITURE	ADRESSE
Piboule	Scénic	4, rue Marie Curie
Piboule	Zoé	4, rue Marie Curie
Morgane	Scénic	16, rue Valentin

Cet exemple n'est pas de deuxième forme normale, l'attribut **ADRESSE** ne dépend que de l'attribut **NOM**.

Dans le cas d'un changement d'adresse d'un garage, il y a un risque non négligeable d'oublier de modifier une adresse.

MCD : deuxième forme normale (2NF)

GARAGES		
NOM	VOITURE	ADRESSE
Piboule	Scénic	4, rue Marie Curie
Piboule	Zoé	4, rue Marie Curie
Morgane	Scénic	16, rue Valentin

Solution normalisée, scinder l'entité en deux :

GARAGES	
NOM	VOITURE
Piboule	Scénic
Piboule	Zoé
Morgane	Scénic

GARAGES-ADRESSES	
NOM	ADRESSE
Piboule	4, rue Marie Curie
Morgane	16, rue Valentin

Ainsi, un changement d'adresse ne donne lieu qu'à une seule modification dans l'entité **GARAGES-ADRESSES**.

MCD : troisième forme normale (3NF)

Une entité est dite de troisième forme normale, si :

- elle est de deuxième forme normale ;
- tout attribut n'appartenant pas à une clé ne dépend pas d'un autre attribut non clé.

Les attributs sont donc complètement indépendants les uns des autres.

Par exemple :

GARAGES			
NOM	ADRESSE	VILLE	CODE POSTAL
Piboule	4, rue Marie Curie	Colombes	92700
Morgane	16, rue Valentin	Versailles	78000

L'attribut **CODE POSTAL** n'est pas dépendant de la clé **NOM** de l'entité **GARAGES**, mais est fonction de l'attribut **VILLE**.

MCD : troisième forme normale (3NF)

GARAGES			
NOM	ADRESSE	VILLE	CODE POSTAL
Piboule	4, rue Marie Curie	Colombes	92700
Morgane	16, rue Valentin	Versailles	78000

Solution normalisée, scinder l'entité en deux :

GARAGES		
NOM	ADRESSE	VILLE
Piboule	4, rue Marie Curie	Colombes
Morgane	16, rue Valentin	Versailles

COMMUNES	
VILLE	CODE POSTAL
Colombes	92700
Versailles	78000

De cette manière, une modification du Code Postal pour une ville ne donnera lieu qu'à une seule modification.

La troisième forme normale :

- enlève les redondance dues aux dépendances transitives ;
- permet de ne pas perdre d'information ;
- permet de ne pas perdre des dépendances.

MCD : forme normale de Boyce-Codd (BCNF)

Tout attribut n'appartenant pas à une clé dépend de cette clé entièrement et aucun attribut de la clé ne dépend d'un autre attribut.

Considérons l'entité suivante, où la clé est une clé composite (**NOM-CARBURANT**):

GARAGES			
NOM	CARBURANT	ADRESSE	PRIX
Piboule	Diésel	4, rue Marie Curie	1.18
Piboule	SP95	4, rue Marie Curie	1.43

Elle est en troisième forme normale, mais pas en forme normale de Boyce-Codd. En effet, on peut déduire :

- du nom du garage et du carburant, le prix ;
- du nom du garage, son adresse.

MCD : forme normale de Boyce-Codd (BCNF)

GARAGES			
NOM	CARBURANT	ADRESSE	PRIX
Piboule	Diésel	4, rue Marie Curie	1.18
Piboule	SP95	4, rue Marie Curie	1.43

Solution normalisée, scinder encore l'entité en deux :

GARAGES		
NOM	CARBURANT	PRIX
Piboule	Diésel	1.18
Piboule	SP95	1.43

GARAGES-ADRESSES	
NOM	ADRESSE
Piboule	4, rue Marie Curie

MCD : méthode Merise

La méthode MERISE date de 1978-1979, et fait suite à une consultation nationale lancée en 1977 par le ministère de l'Industrie dans le but de choisir des sociétés de conseil en informatique afin de définir une méthode de conception de systèmes d'information.

Les deux principales sociétés ayant mis au point cette méthode sont le CTI (Centre Technique d'Informatique) chargé de gérer le projet, et le CETE (Centre d'Etudes Techniques de l'Equipement) implanté à Aix-en-Provence.

MCD : méthode Merise

MERISE est une méthode de conception, de développement et de réalisation de projets informatiques. Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques.

MCD : méthode Merise

La méthode Merise préconise 3 niveaux d'abstraction :

- le **Modèle Conceptuel des Données (MCD)** ;
- le **Modèle Conceptuel des Traitements (MCT)** ;
- Le **Modèle Organisationnel**.

Niveau	Statique (données)	Dynamique (traitements)	
Conceptuel	MCD	MCT	Indépendant du système
Organisationnel ou logique	MLD	MOT	Quel SGBD ?
Opérationnel ou physique	MPD	MOPT	Connaissance du SGBD

MCD : méthode Merise

Logiciels de modélisation :

- PowerAMC ou AMC*Designor ;
- Open ModelSphere ;
- DBDesigner ;
- JMerise ;
- ...

MCD : JMerise

JMerise : est un logiciel dédié à la modélisation des modèles conceptuels de donnée pour Merise.

Jmerise a été développé par M. Messouci en 2011 et est en perpétuel développement.

Jmerise : <http://www.jfreesoft.com/JMerise/>

MCD : JMerise

The screenshot displays the JMerise software interface, which is used for generating database models. The main window shows the MCD (Model Conceptuel de Données) diagram, which includes entities like **Voiture**, **Personne**, **Travailler**, **Date**, **Camion**, **VoitureTouristique**, **Acheter**, **Conduire**, **Derigner**, and **CIF**. The diagram illustrates various relationships such as inheritance (XT), associations (Acheter, Conduire, Derigner), and domain definitions (D_sexe = enum("Masculin", "Feminin")).

Below the MCD diagram, the MLD (Modèle Logiciel de Données) diagram shows the logical structure of the database, including tables like **Voiture**, **VoitureTouristique**, **Travailler**, **Camion**, **Service**, and **Date**, along with their attributes and primary keys.

On the right side, the SQL code for creating the database is displayed:

```
#Ma base de données :
CREATE TABLE Voiture(
  id Varchar(25) NOT NULL ,
  id_Personne Varchar(25) NOT NULL ,
  id_Personnel Varchar(25) NOT NULL ,
  PRIMARY KEY (id)
)ENGINE=InnoDB;
CREATE TABLE Personne(
  id Varchar(25) NOT NULL ,
  nom Varchar(50),
  sexe Enum ("Feminin", "Masculin"),
  id1 Varchar(25) NOT NULL ,
  PRIMARY KEY (id)
)ENGINE=InnoDB;
CREATE TABLE VoitureTouristique(
  id_Voiture Varchar(25) NOT NULL ,
  id_Personne Varchar(25) NOT NULL ,
  id_Personnel Varchar(25) NOT NULL ,
  PRIMARY KEY (id_Voiture)
)ENGINE=InnoDB;
CREATE TABLE Camion(
  id_Voiture Varchar(25) NOT NULL ,
  id_Personne Varchar(25) NOT NULL ,
  id_Personnel Varchar(25) NOT NULL ,
  PRIMARY KEY (id_Voiture)
)ENGINE=InnoDB;
```

The interface also includes a menu bar (Fichier, Editer, Merise, Configuration, Fenêtre, A propos), a toolbar, and a sidebar with a tree view of the project structure. A play button icon is visible in the center of the MLD diagram, indicating that the model can be executed to generate the database.