MERISE
REGLES DE PASSAGE DU MODELE CONCEPTUEL DES DONNEES AU MODELE LOGIQUE DES DONNEES

I- RAPPEL SUR LES NIVEAUX D'ABSTRACTION

Dans la méthode Merise, l'étude du système d'information s'effectue selon plusieurs niveaux de préoccupation dits d'abstraction. A chaque niveau d'abstraction, l'étude des données donne lieu à la production d'un modèle de données :

⇐ Au ***niveau conceptuel***, le ***Modèle Conceptuel des Données*** utilise une représentation sous forme d'un schéma Entité-Association (MEA). Ce schéma permet de représenter les liens entre les données indépendamment de toute organisation.

⇐ Au ***niveau logique***, le ***Modèle Logique des Données*** décrit l'organisation des données sur support informatique. Les données peuvent être mémorisées dans des fichiers ou dans une base de données.
Dans le cas où les données sont mémorisées dans un système de gestion de bases de données relationnelles, la représentation du Modèle Logique des Données est un **schéma relationnel.**

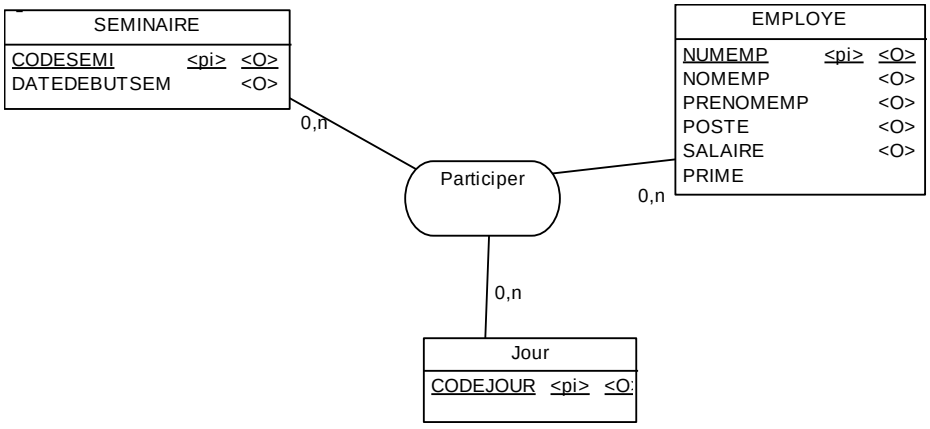
⇐ Au ***niveau physique***, le ***Modèle Physique des Données*** est celui du système de stockage retenu. Par exemple dans le cas d'un SGBD-R Oracle, c'est le schéma de la base Oracle qui constitue le modèle physique des données.

II- REGLES DE BASE

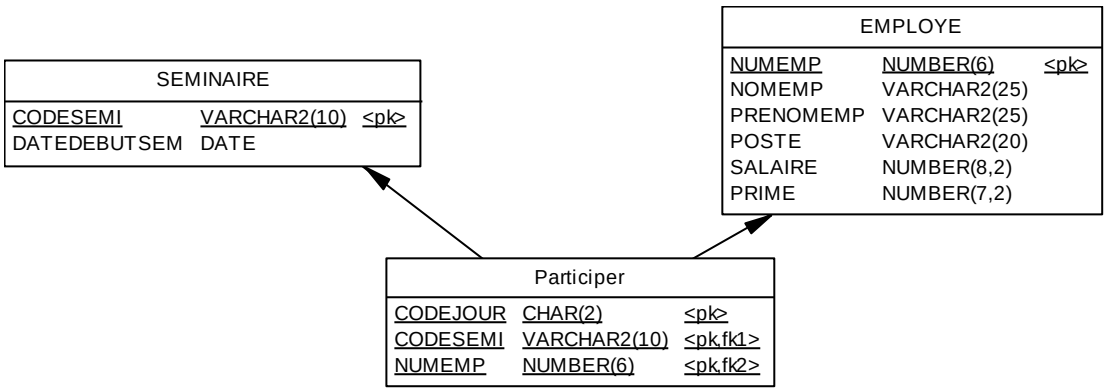
Pour obtenir un modèle relationnel à partir du Modèle conceptuel des données, il faut appliquer les règles suivantes :

Règle 1 : Toute entité qui n'est porteuse que de son identifiant (attribut spatio-temporel en général), n'apparaît plus dans le modèle relationnel (entité JOUR par exemple),

Exemple
Soit le MCD :



MLD correspondant :

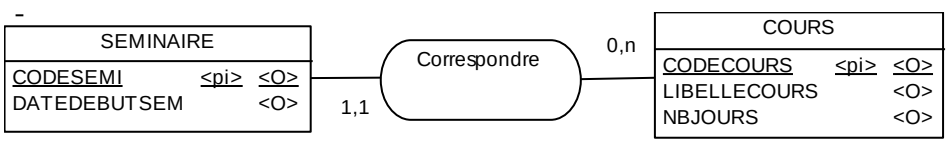


Règle 2 : Toutes les autres entités deviennent des relations dont la clef primaire est l'identifiant de l'entité

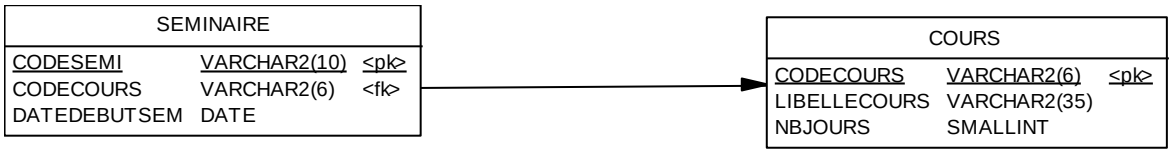
Exemples : voir les entités SEMINAIRE et EMPLOYE ci-dessus

Règle 3 : Toute association binaire fonctionnelle (qui contient une cardinalité 1,1 ou 0,1) sur une branche, donne naissance à une clef étrangère dans la relation issue de l'entité supportant la cardinalité 1,1 ou 0,1. Cette clé étrangère est la clé primaire de l'entité supportant la cardinalité 1,n ou 0,n. Dans le cas de la cardinalité 0,1, la clé étrangère peut avoir une valeur nulle, qu'il faut bien sûr le préciser de manière très explicite dans la définition de la table.

Exemple
Soit le MCD :

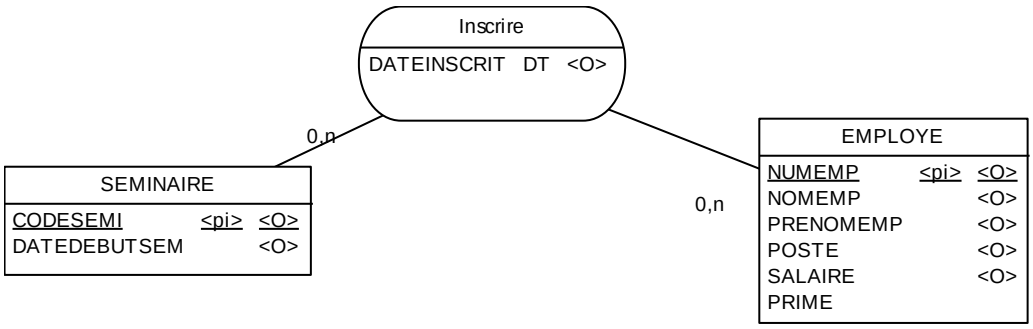


MLD correspondant :

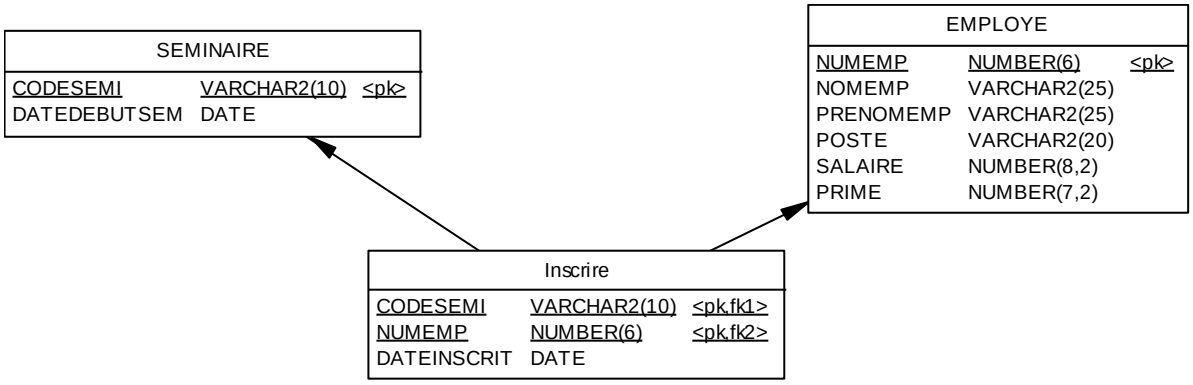


Règle 4 : les associations N:M ((1,n) ou (0,n) vers (0,n) ou (1,n), ou les associations ternaires et plus deviennent des relations dont la clef primaire est d'ordre m, et correspondant à la concaténation des m identifiants des entités concourant à la définition de l'association,

Exemple :
Soit le MCD suivant :

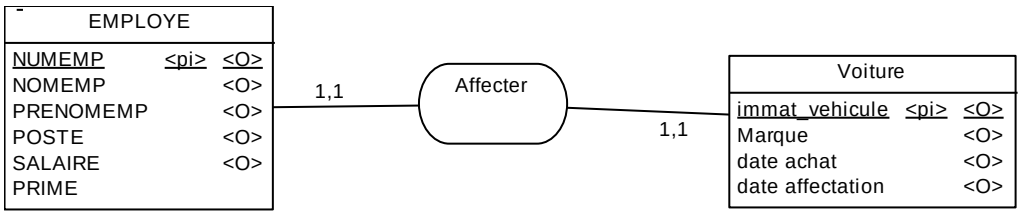


MLD correspondant :

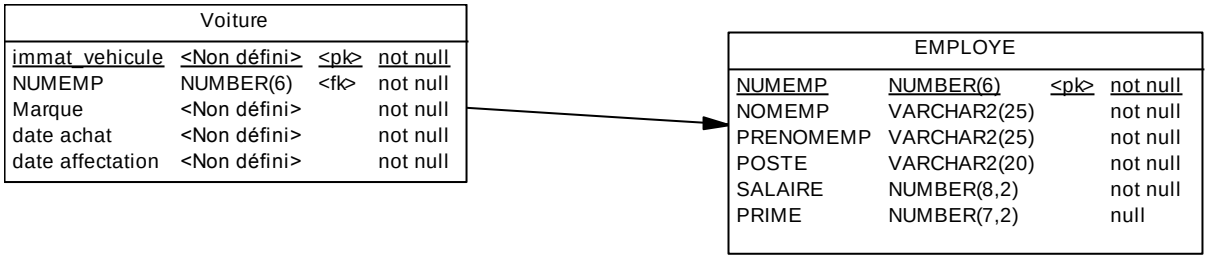


Règle 5 : Cas des associations binaires 1,1 vers 1,1 : On s'assurera que l'on ne peut pas fusionner les deux entités. Si tel est le cas, on ne créera qu'une seule clé étrangère du côté de l'entité la plus stable dans le temps. ☛ **Remarque** : certains AGL ne permettent pas ce cas de figure. Dans ce cas, il faut dégrader une cardinalité 0,1

Exemple
Soit le MCD :



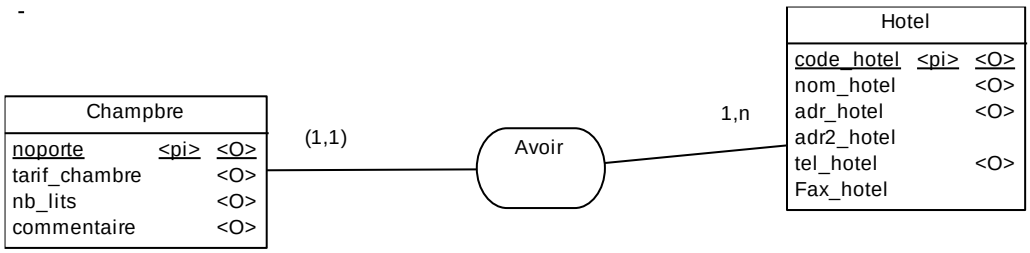
MLD correspondant :



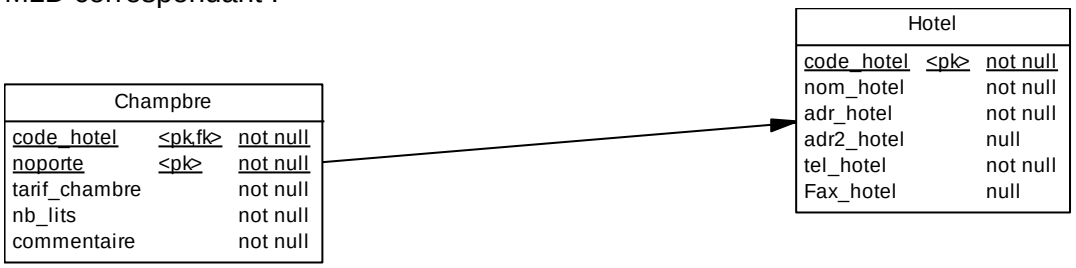
Règle 6 : Cas des associations binaires supportant un lien identifiant : l'entité faible supportant la cardinalité (1,1) fournit une relation ayant pour identifiant son identifiant et l'identifiant de l'entité associée.

Exemple :
Soit le MCD suivant :

-

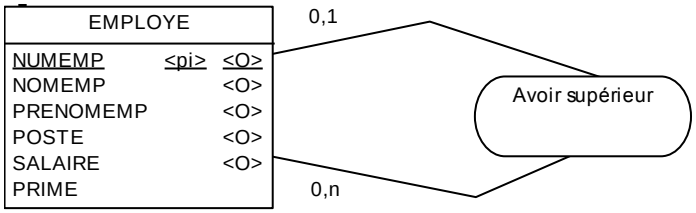


MLD correspondant :

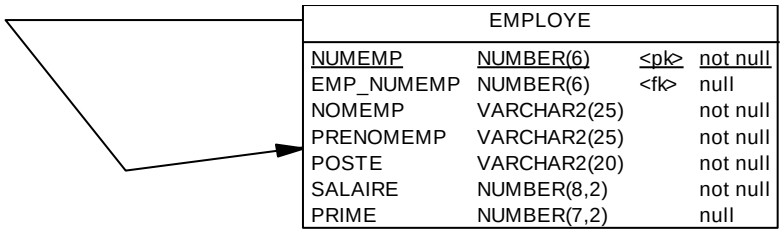


Cas particulier : les associations réflexives :
Elles se comportent exactement comme les associations binaires, fonctionnelles ou non.

Exemple
Soit le MCD :



MLD correspondant :



∗ **Remarque :**
Les clefs primaires et étrangères de chaque relation composant le modèle logique doivent être mises en évidence (par exemple souligner les clefs primaires et faire suivre les clefs étrangères d'un dièse (#)). Il est souhaitable que la légende choisie soit exposée.

III- ENRICHISSEMENT

1- LES CLES ETRANGERES

Il sera important de préciser, pour chaque clé étrangère :

- L'option de suppression,
- L'option de mise à jour;

Ces options indiquent le comportement de l'enregistrement possédant la clé étrangère, ainsi que la valeur de clé étrangère afin de préserver l'intégrité référentielle dans le cas :

- de la suppression de l'enregistrement contenant la clé primaire
- de la mise à jour de la valeur de la clé primaire

a) les options dans le cas d'une suppression

- ***Restrict*** : La suppression de l'enregistrement contenant la clé primaire n'est possible que s'il n'existe pas d'enregistrement dont la clé étrangère a la valeur de la clé primaire supprimée

- ***Cascade*** : La suppression de l'enregistrement contenant la clé primaire entraîne la suppression de tous les enregistrements dont la clé étrangère a la valeur de la clé primaire supprimée

- **Set null** : La suppression de l'enregistrement contenant la clé primaire implique la mise à null des valeurs des clés étrangères correspondantes (*à condition que la clé étrangère accepte la valeur null*).

- **Set default** : La suppression de l'enregistrement contenant la clé primaire implique la mise à la valeur par défaut des valeurs des clés étrangères correspondantes (*à condition que la clé étrangère admette une valeur par défau)t*.

b) les options dans le cas d'une mise à jour

- ***Restrict*** : La modification de la valeur de la clé primaire n'est autorisé que s'il n'existe pas d'enregistrement dont la clé étrangère a la valeur de la clé primaire modifiée.

- ***Cascade*** : La modification de la valeur de la clé primaire entraîne la mise à jour en cascade de tous les enregistrements dont la clé étrangère a la valeur de la clé primaire modifiée.

- **Set null** : La modification de la valeur de la clé primaire implique la mise à null des valeurs des clés étrangères correspondantes (*à condition que la clé étrangère accepte la valeur null*).

- **Set default** : La modification de la valeur de la clé primaire implique la mise à la valeur par défaut des valeurs des clés étrangères correspondantes (*à condition que la clé étrangère admette une valeur par défaut)*.

Ces contraintes peuvent être mises en place de trois façons :

- ***Déclarative*** : on le précise au moment de la création des tables, au moyen de la clause ***constraint***. (voir cours SQL)

- ***Trigger (déclencheur)*** : procédure compilée au sein de la base de donnée et qui se déclenche automatiquement lors d'une opération d'insertion, de mise à jour ou de suppression (voir cours PL/SQL)

- ***Applicatif*** : c'est le programme d'application qui prend en charge la vérification de ces contraintes

2- LES INDEX SECONDAIRES (CANDIDATES)

Les clés alternatives ne sont ni des clés primaires, ni des clés étrangères et pourtant il est souvent utile d'accélérer les accès à une table sur d'autres colonnes que sur la clé primaire, voire la clé étrangère.
Par exemple, la clé primaire de la table ***employe*** est numemp, sur laquelle, par défaut un index sera créé. Toutefois, on peut être amené à effectuer fréquemment des recherches sur le nom des employés (colonne nomemp). Dans ce cas là, on peut enrichir le modèle par la création d'un index sur la colonne nomemp.

Cet index secondaire consiste à créer un fichier d'index sur une clé "secondaire", formée par une ou plusieurs colonnes de la table, sans réorganisation des données : le fichier d'index va donner, pour chaque clé secondaire, l'adresse de l'enregistrement correspondant à cette clé.
Le fichier d'index est traité comme un index sur la clé primaire, à savoir comme une table qui contient, au lieu des données, des couples (clé, adresse de donnée).

Il est créé par la commande :
create index inom on employe(nomemp)

☛ **Remarque** : une clé secondaire peut être formée de colonnes faisant partie de la clé primaire.

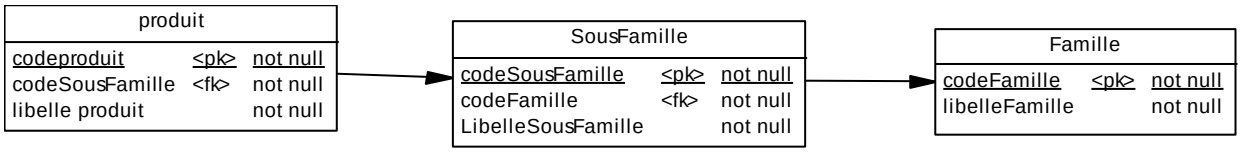
3- LA CREATION DE REDONDANCES

On peut agir sur deux points :
- ➥ La création de transitivité :
- ➥ La création de colonnes calculées

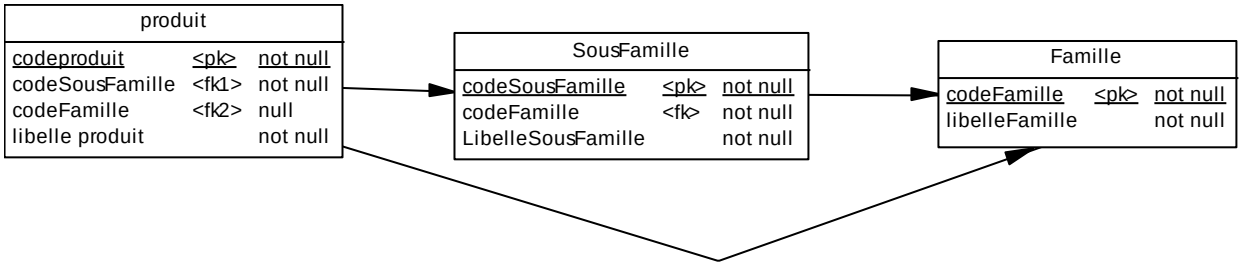
a) La création de transitivités :

La troisième forme normale, si elle garantit l'intégrité des données, ne garantit pas des performances optimales.
C'est pourquoi, au niveau du modèle logique des données, on peut être amenés à créer des transitivités afin de limiter le nombre de lectures dans les tables.
Exemple :

Soit le modèle suivant en troisième forme normale :



On peut créer une transitivité en indiquant un lien entre produit et famille, qui permettra d'accédes directement à la famille d'un produit sans passer par la sous famille.



b) La création de colonnes calculées

Dans le même ordre d'idée, on peut créer des colonnes dont le résultat peut être obtenu à partir des valeurs d'autres colonnes, dans le seul but de limiter le nombre de lectures de tables, qui est fortement pénalisant au niveau des performances.

Par exemple, dans le cas des séminaires et des cours, s'il revient fréquemment que les utilisateurs cherchent à connaître le nombre d'inscrits à un séminaire, on peut créer une propriété calculée dans séminaire fournissant directement l'information, sans avoir à aller balayer la table inscrits.

SEMINAIRE		
CODESEMI	VARCHAR(10)	<pk>
CODECOURS	VARCHAR(6)	<fk>
DATEDEBUTSEM	DATETIME	

deviendrait :

SEMINAIRE		
CODESEMI	VARCHAR(10)	<pk>
CODECOURS	VARCHAR(6)	<fk>
DATEDEBUTSEM	DATETIME	
NbInscrits	INTEGER	

Il est évident que pour toutes ces redondances d'information, il faut mettre en place des traitements qui garantissent l'intégrité des données.
Ces programmes sont souvent des triggers (déclencheurs)