

# Documentation sur MCD, MLD et SQL

**MERISE** est une méthode d'analyse et de conception des systèmes d'information basée sur le principe de la séparation des données et des traitements. Elle possède plusieurs modèles qui sont répartis sur 3 niveaux (Le niveau conceptuel, le niveau logique ou organisationnel, le niveau physique).

Merise propose un ensemble de formalismes et de règles destinées à modéliser de manière indépendante les données et les traitements du système d'information.

Modéliser consiste à donner une représentation structurée d'un système réel, afin de reproduire son fonctionnement sur ordinateur et de répondre de manière efficace et rapide à des questions que l'on pourrait se poser sur le système réel.

## C'est quoi un système d'information ?

«Un système d'information (noté SI) représente l'ensemble des éléments (membres, ordinateurs, postes de travail, règles et méthodes, etc.) participant à la collecte, à la saisie, à la gestion, au stockage, au traitement, au transport et à la diffusion de l'information au sein d'une organisation »

## ○ Modèle conceptuel de données(MCD)

La **modélisation conceptuelle des données** a pour but de décrire de façon formelle les données qui seront utilisées par le système d'information, sans se préoccuper ni des contraintes d'organisation, ni du gestionnaire de bases de données utilisé, ni des traitements. Le formalisme (langage) adopté par la méthode Merise pour réaliser cette description est basé sur le modèle « entité-association »

## ■ Les entités :

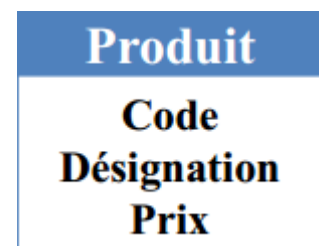
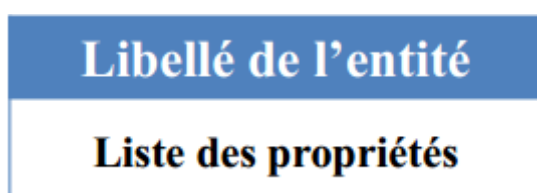
Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire.

Chaque entité est composée de données élémentaires, appelées propriétés, permettant de la décrire.

**Notation** : Les entités sont représentées par un rectangle séparé en deux champs :

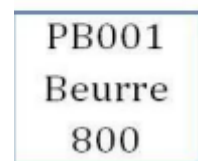
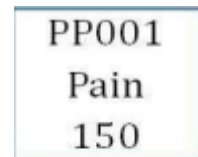
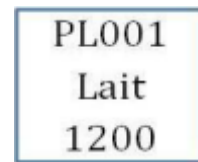
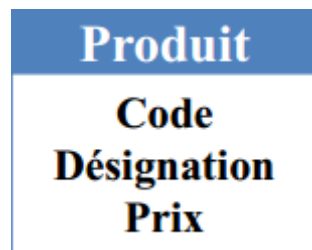
- Le champ du haut contient le libellé, qui est unique pour chaque entité.
- Le champ du bas contient la liste des propriétés de l'entité.

**Exemple :**



## Entités : Occurrence

Créer une occurrence d'une entité consiste à donner une valeur réelle à chaque propriété de l'entité.



## Entités : Identifiant

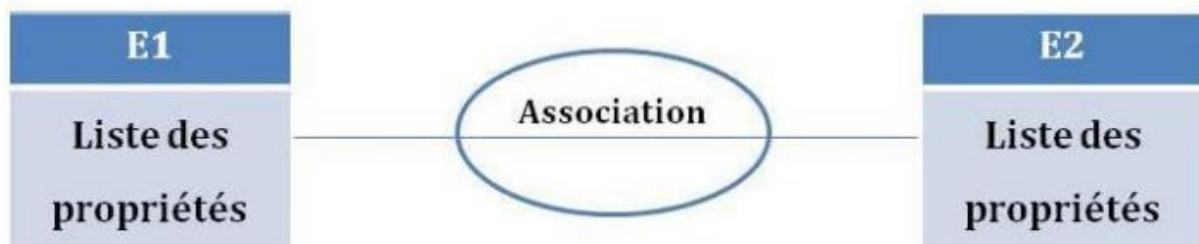
Un identifiant est un ensemble de propriétés (une ou plusieurs) permettant de désigner de façon unique un objet de l'entité.

**L'identifiant** est une propriété particulière d'un objet telle qu'il n'existe pas deux occurrences de cette entité pour lesquelles cette propriété pourrait prendre une même valeur.

## ■ Les associations :

Une association (appelée aussi parfois relation) est un lien sémantique entre des entités. Le nom décrit le type d'association qui relie les entités (généralement un verbe).

Une association peut relier une ou plusieurs entités.



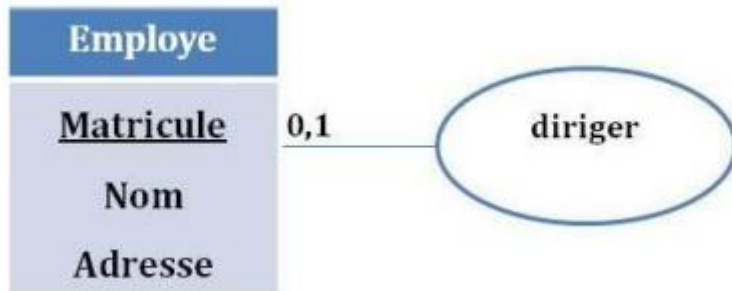
## ■ Cardinalités :

- Les cardinalités permettent de caractériser (ajouter une contrainte ou règle de gestion) le lien qui existe entre une entité et l'association à laquelle elle est reliée.
- La cardinalité est composée d'un couple comportant une borne minimale et une borne maximale :

- La borne minimale (généralement **0** ou **1**) décrit le nombre minimum de fois qu'une occurrence d'une entité peut participer à une association.
- La borne maximale (généralement **1** ou **n**) décrit le nombre maximum de fois qu'une entité peut participer à une association.

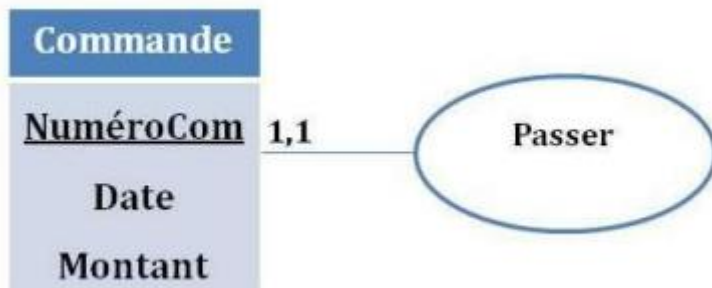
**0,1** : Une occurrence de l'entité E est reliée au plus à une occurrence de l'association.

Exemple : **Un employé ne peut diriger qu'un seul département à la fois.**



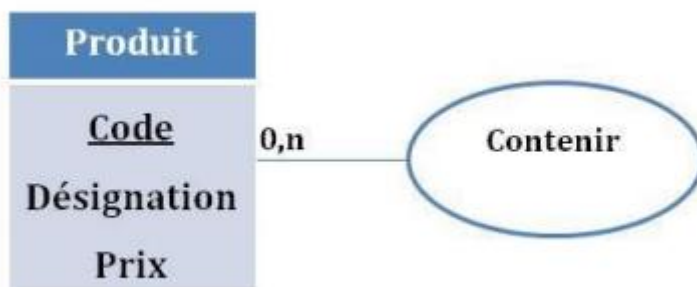
**1,1** : Une occurrence de l'entité E est reliée forcément à une et une seule occurrence de l'association.

Exemple : **Une commande est passée par un et un seul client.**



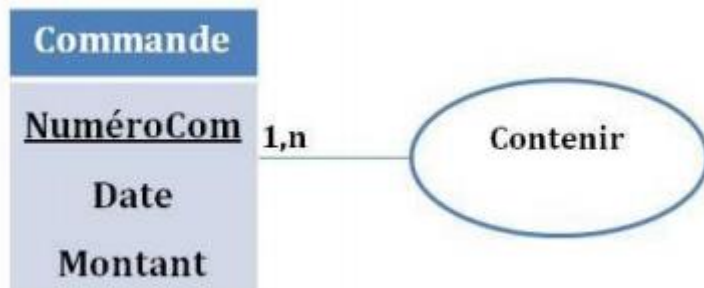
**0, n** : Une occurrence de l'entité E peut être reliée à plusieurs occurrences de l'association ou pas du tout.

Exemple : **Un produit peut être contenu dans plusieurs commandes**

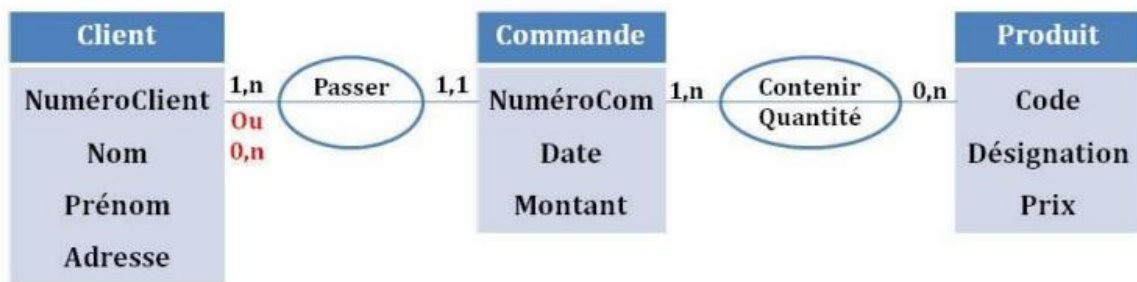


**1, n** : Une occurrence de l'entité E est reliée au moins à une occurrence de l'association.

Exemple : Une commande contient au moins un produit.



Exemple d'un MCD :



**NB** : Pour une situation donnée, il n'existe pas une « **solution unique** ». Un modèle exprime un **point de vue** et reflète des **besoins** en information. Le bon modèle est celui qui est **accepté** par les **personnes concernées** par le projet.

## ○ Modèle Logique de données (MLD)

- ❖ La description conceptuelle a permis de représenter le plus fidèlement possible les réalités de l'univers à informatiser.
- ❖ Mais cette représentation ne peut pas être directement acceptée et manipulée par un système informatique.
- ❖ Il est donc nécessaire de passer du niveau conceptuel à un second niveau plus proche des capacités des systèmes informatiques.
- ❖ Ce niveau, appelé le niveau logique est une traduction du modèle conceptuel en intégrant les choix d'organisation des données.
- ❖ Nous obtenons à ce niveau le modèle logique des données qui consiste à décrire la structure des données utilisée sans faire référence à un langage de programmation.

Dans le vocabulaire de représentation du MLD, nous distinguons :

- les tables et les champs
- les clés (primaires et étrangères) et
- des flèches qui matérialisent les relations entre les tables

Les tables correspondent aux entités ou associations du MCD. Elles sont représentées sous forme de rectangle divisé en deux parties dans le sens de la hauteur.

La partie du haut contient le nom de la table et qui est le même que le nom de l'entité ou de l'association correspondante.

La partie du bas contient ce qu'on appelle des champs et qui correspondent aux propriétés des entités ou associations dans le MCD.

Client
<u>NumeroClient</u>
Nom
Prenom
Adresse

Exemple de contenu de la table client.

NumeroClient	Nom	Prenom	Adresse
CL001	Diouf	Matar	PA U18 Villa ZZZ DK
CL002	Diallo	Yacine	7 Rue BD St-Louis
...	...	...	...

## Les clés primaires :

Les lignes d'une table sont uniques, i.e. qu'il existe au moins une ou un ensemble de colonnes qui sert à identifier les lignes : il s'agit de la clé primaire de la table.

Les champs de la table dans le MLD qui constituent la clé primaire sont soulignés et correspondent aux identifiants des entités ou associations dans le MCD.

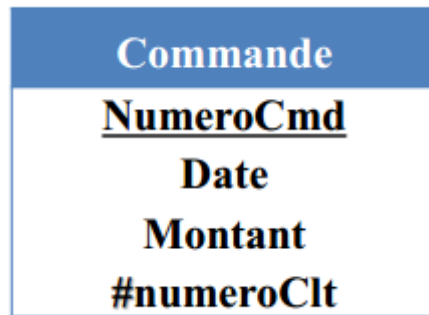
Ces champs doivent toujours prendre des valeurs non nulles et celles-ci ne doivent pas changer au cours du temps.

## Les clés étrangères

Les clés étrangères sont des champs d'une table correspondant à des clés primaires dans d'autres tables et permettant de traduire la notion de contrainte d'intégrité référentielle.

Ainsi, la connaissance d'une ligne dans cette table implique la connaissance de sa ligne correspondante dans une autre table.

On fait précéder d'un # les clés étrangères.



Dans cet exemple, NumeroCmd représente une clé primaire et #numeroClt représente une clé étrangère.

## Les flèches :

Les liens entre les clés étrangères et leurs clés primaires associées sont symbolisés par un connecteur (une flèche) orienté vers la table contenant la clé étrangère.



# UML

UML (en anglais Unified Modeling Language ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la «conception orientée objet». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

[UML est avant tout un support de communication performant, qui facilite la représentation et la compréhension de solutions objet :

- Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.
- L'aspect formel de sa notation, limite les ambiguïtés et les incompréhensions.
- Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en font un langage universel.

## Etude Comparative

*Tableau 1 : tableau comparatif UML - Merise*

<b>Merise</b>	<b>UML</b>
méthode d'analyse et de conception de système d'information	langage de représentation d'un système d'information.
méthode de modélisation de données et traitements orienté bases de données relationnelles.	système de notation orienté objet.
relationnel	objet.
Franco-français	International
schéma directeur, étude préalable, étude détaillée et la réalisation.	langage de modélisation des systèmes standard, qui utilise des diagrammes pour représenter chaque aspect d'un systèmes ie: statique, dynamique,.....en s'appuyant sur la notion d'orienté objet
plus adapté à une approche théorique	plus orientée vers la conception
du "bottom up" de la base de donnée vers le code	du "top down" du modèle vers la base de donnée.



## Modélisation

- ↳ Introduction
- ↳ Définition
- ↳ Construction : Cardinalités
- ↳ **UML et Merise**



UML	MERISE
Diagramme de classes	Modèle conceptuel de données
Classe	Entité
Association (Relation)	Association (Relation)
Multiplicité	Cardinalité
Objet	Occurrence

<http://www.med.univ-rennes1.fr>

*Lim*  
Laboratoire d'Informatique Médicale



# SQL (Structured Query Language)

## I. Langage de définition des données

Le Langage de Définition des Données est la partie de SQL qui permet de décrire les tables et autres objets manipulés par les SGBD.

La commande **CREATE TABLE**

L'ordre CREATE TABLE permet de créer une table en définissant le **nom**, le **type** de chacune des colonnes de la table.

```
CREATE TABLE table (colonne1 type1, colonne2 type2, ..., colonnen typen);
```

Exemple :

```
CREATE TABLE IF NOT EXISTS `etudiant` (  
  `id_etudiant` int(11) NOT NULL AUTO_INCREMENT,  
  `Nom` varchar(20) NOT NULL,  
  `Prenom` int(11) NOT NULL,  
  PRIMARY KEY (`id_etudiant`)  
);
```

La commande **ALTER TABLE**

Cette commande permet de modifier la définition d'une table.

```
ALTER TABLE table  
{  
  ADD  
  MODIFY (colonne1 type1, colonne2 type2, ..., colonnen typen)  
  DROP
```

- Exemples :

```
ALTER TABLE STAGIAIRE
ADD NomEntr VARCHAR2(20);
```



Ajout d 'un attribut

```
ALTER TABLE STAGIAIRE
MODIFY NomEntr VARCHAR2(30);
```



Modification dans le type d 'un attribut

```
ALTER TABLE STAGIAIRE
DROP NomEntr;
```



Suppression d 'un attribut

Attention : Certaines modifications peuvent poser des problèmes d 'intégrité dans la base !

La commande **DROP TABLE**

Cette commande permet de supprimer une table de la base de données. Les lignes de la table et la définition elle-même sont détruites. L'espace occupé par la table est libéré.

Exemple : DROP TABLE ETUDIANT ;

IL y'a d'autres commande comme : **RENAME**, **TRUNCATE** etc.

## II. Langage de manipulation des données

La commande **INSERT**

Cette commande permet d'ajouter des lignes dans une table.

```
INSERT INTO table (col1, ..., coln)
```

```
VALUES (val1, ..., valn);
```

Exemple :

```
INSERT INTO `module` (`Num_Module`, `Nom_Module`, `Nb_Heures`) VALUES
```

```
(4, 'Francais', 10),
```

```
(5, 'Statistique', 20),
```

```
(6, 'infprmatique', 34);
```

La commande **UPDATE**

Cette commande permet de modifier les valeurs d'un ou de plusieurs champs, dans une ou plusieurs lignes existantes d'une table.

Exemple :

```
UPDATE table
```

```
SET col1 = exp1, col2 = exp2, ...
```

```
WHERE prédicat
```

La clause WHERE est facultative

La commande DELETE

L'ordre DELETE permet de supprimer des lignes d'une table.

Exemple : DELETE FROM etudiant

### III. Les langages d'interrogation des données

On distingue plusieurs commande comme la commande SELECT, UNION, GROUP BY, ORDER BY, HAVING etc.