

# **Systèmes d'Information & Base de données**

**-- Systèmes d'information, MCD&MLD --**

**-- Base de données & SGBD --**

**Abdeslam EN-NOUAARY, PhD , Ingénieur fonctionnel**

**[abdeslam@inpt.ac.ma](mailto:abdeslam@inpt.ac.ma)**

**Anass RACHDI, Docteur , Ingénieur**

**[anass.rach@gmail.com](mailto:anass.rach@gmail.com)**

**Avril 2018**

# Structure de cours

## A. Première partie

- Architecture d'un système d'information
- Modélisation des données & des traitements

## B. Deuxième partie

- Les bases de données & les SGBD

## C. Troisième partie

- Le langage de requête structurée sur les BDR (SQL)

# Objectifs de la première partie

- **Définition de l'architecture et les fonctionnalités d'un système d'information représentant la colonne vertébrale de tout organisme**
- **Modélisation des données & des traitements**
- **Transposition du schéma : Entité/Association en schéma relationnel**
- **Normalisation du schéma relationnel (formes normales)**

# Système d'information

- **Un système ?**

**C'est un ensemble d'éléments en interaction qui vise à atteindre un certain objectif.**

- **Une information ?**

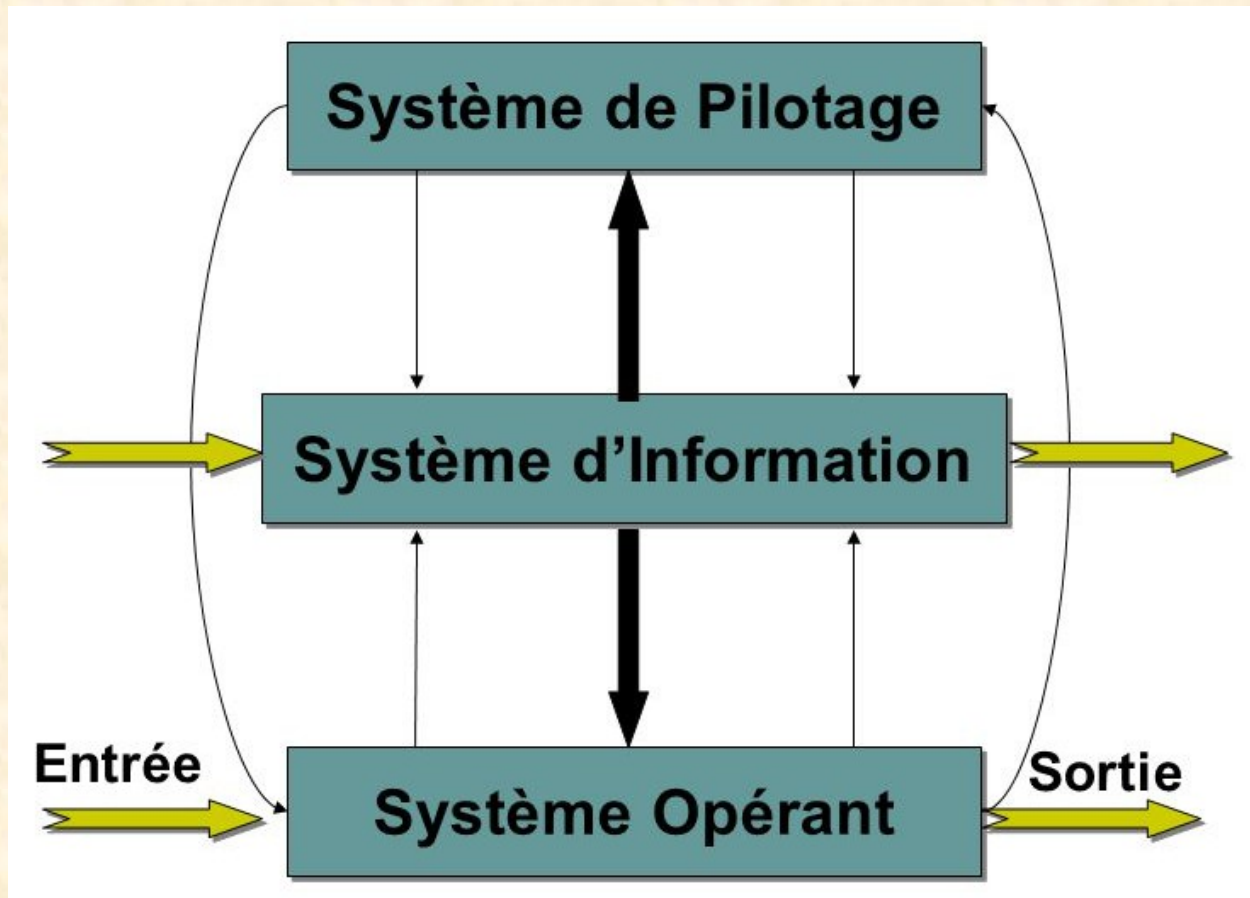
**Donnée : Signe + code**

**Information : Donnée + Modèle d'interprétation**

- **Le système d'information (SI) est un ensemble de ressources qui permet de saisir, mémoriser, traiter et communiquer l'information.**
- **Il s'agit d'un système socio-technique composé de deux sous-systèmes, l'un social et l'autre technique. Le sous-système social est composé de la structure organisationnelle . Le sous-système technique est composé des technologies (serveurs, logiciels, base de données...) et des processus métiers concernés par le SI.**

# Systeme d'information

- Le système d'information est un système de couplage entre le système de pilotage et le système opérant



# Le système de pilotage & le système opérant

- **Le système de pilotage : c'est l'endroit où les décisions sont prises en matière de :**
  - Fixation d'objectifs
  - Contrôle des résultats obtenus
  - Correction des erreurs si nécessaire
  - ...
- **Le système opérant : c'est le système qui effectue des transformations physiques et/ou intellectuelles du flux reçu par l'organisme. Le flux peut être de :**
  - L'argent
  - Matière brute
  - Équipements (produit fini)
  - L'information
  - ....



# Cycle de vie d'un SI

- Un SI est en ensemble des données et des traitements (qui agissent sur ces données).
- Lors de développement d'un SI, l'étude (données et traitement) se fait séparément et/ou en parallèle. L'étude d'un SI se fait sur 4 niveaux :

	Données	Traitements	
<b>Niveau conceptuel</b>	<b>MCD</b> <i>Modèle Conceptuel de Données</i>	<b>MCT</b> <i>Modèle Conceptuel de Traitements</i>	<b>SIO</b> <i>Système d'Information Organisationnel</i>
<b>Niveau organisationnel</b>	<b>MOD</b> <i>Modèle Organisationnel de Données</i>	<b>MOT</b> <i>Modèle Organisationnel de Traitements</i>	
<b>Niveau logique</b>	<b>MLD</b> <i>Modèle Logique de Données</i>	<b>MLT</b> <i>Modèle Logique de Traitements</i>	<b>SII</b> <i>Système d'Information Informatisé</i>
<b>Niveau physique</b>	<b>MPD</b> <i>Modèle Physique de Données</i>	<b>MPT</b> <i>Modèle Physique de Traitements</i>	

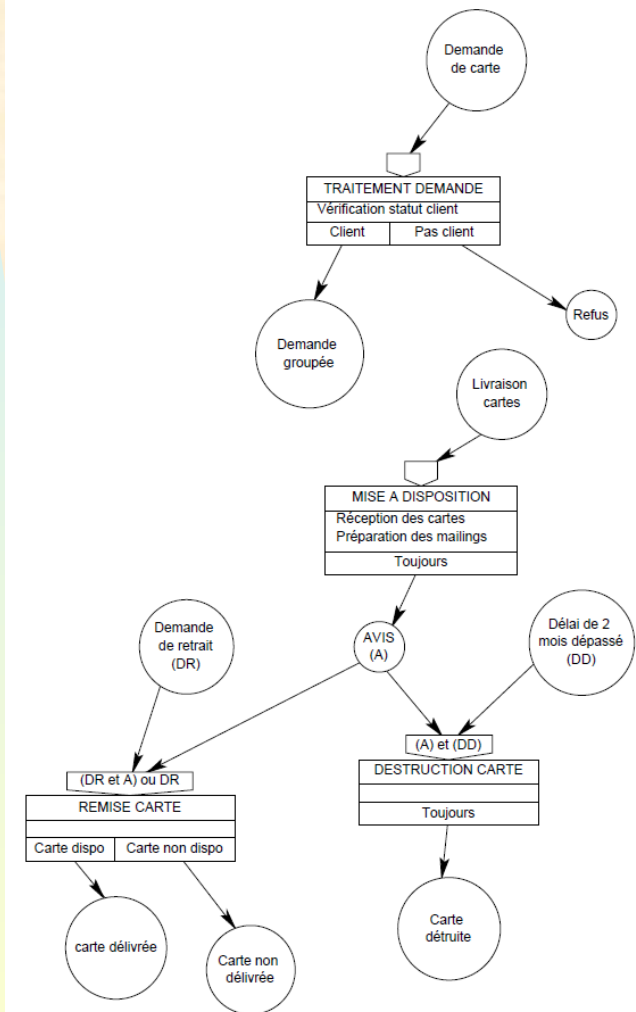
# Cycle de vie des traitements

- **MCT** : Le modèle conceptuel des traitements permet de traiter la dynamique du système d'information, c'est-à-dire les opérations qui sont réalisées en fonction d'événements. il permet de définir simplement ce qui doit être fait, mais il ne dit pas quand, comment ni où...
- **MOT** : Le modèle organisationnel des traitements s'attache à décrire les propriétés des traitements non traitées par le modèle conceptuel des traitements, c'est-à-dire : le temps, les ressources, le lieu...
- **MLT** : Le modèle logique des traitements concerne la description des unités logiques de traitement. Unité logique de traitement est une tâche organisée et exécutée d'une manière automatique
- **MPT** : Le modèle physique de traitement n'est rien que la présentation de la manière dont les différents modules seront agencés sous forme de menu.

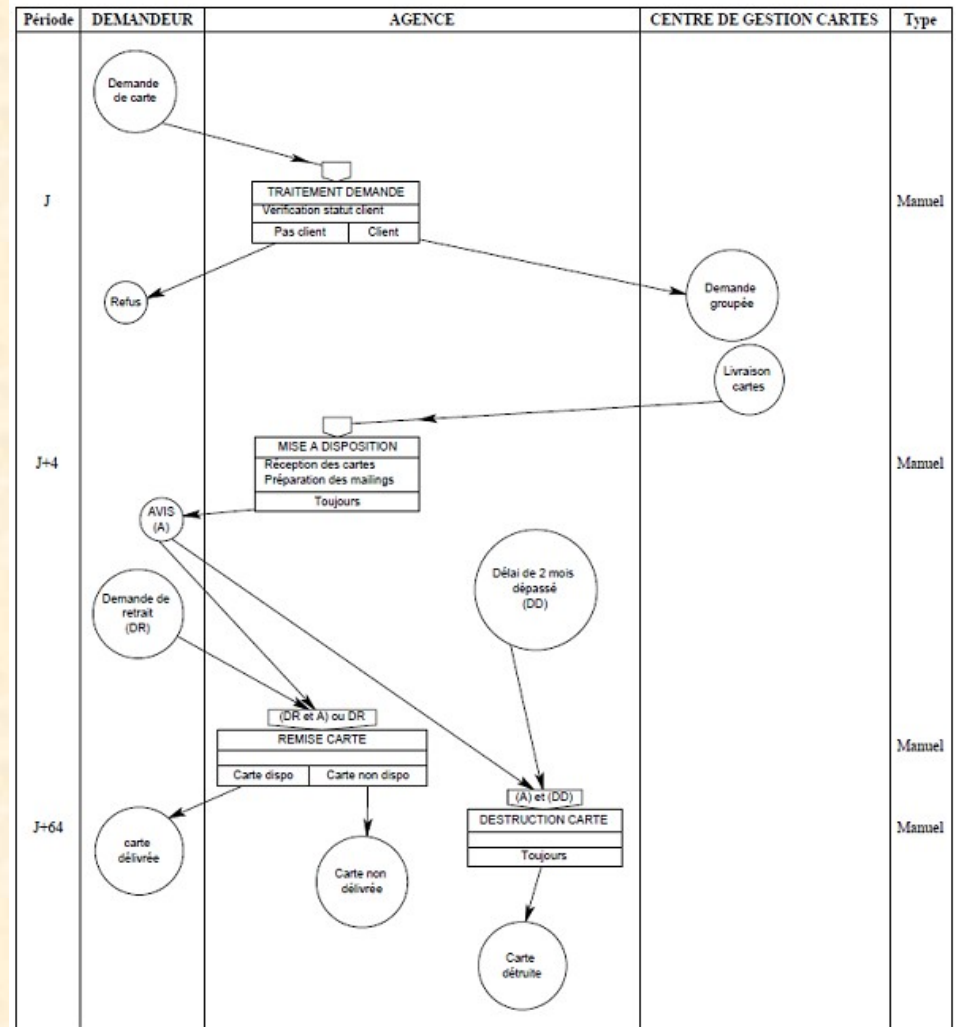


# Cycle de vie des traitements

## • MCT



## • MCO



# Cycle de vie des données

- **MCD** : Il a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités et d'association.
- **MOD** : Lorsqu'une organisation est éclatée entre plusieurs sites, son système d'information doit être accessible depuis ces différents endroits. Il est donc nécessaire de construire des Modèle Organisationnel de Données (MOD) afin de répondre à la question « qui gère/stocke quoi ? »
- **MLD** : Le modèle logique des données consiste à décrire la structure de données (tables) utilisée sans faire référence à un langage de programmation.
- **MPD** : le modèle physique des données (MPD) consiste à implanter une base de données dans un SGBDR. Le langage utilisé pour ce type d'opération est le SQL

# Modélisation des Données

## Formalise Entité/Association

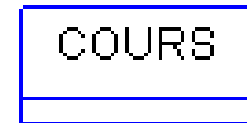
### Entité

- Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire.
- Un type-entité désigne un ensemble d'entités qui possèdent une sémantique et propriétés communes permettant de le décrire
- Exemple : Les personnes, les livres, les cours sont des type-entités.

# Modélisation des Données

## Entité

- Les type-entités sont représentés par un rectangle. Ce rectangle est séparé en deux champs : le champ du haut contient le libellé (en MAJISCULE). le champ du bas contient la liste des attributs d u type-entité (A voir plus tard)
- Une entité est une occurrence ou instance de son type-entité



# Modélisation des Données

## Formalise Entité/Association

### Association

- Une association (appelée aussi parfois relation) représente les liens sémantiques qui peuvent exister entre plusieurs entités.
- Le nom doit être un verbe à l'infinitif, à la forme passive ou bien accompagné d'un adverbe. Exemple : Enseigner, AvoirLieuDans. Notez que l'initiale du nom de l'association est en majuscule.

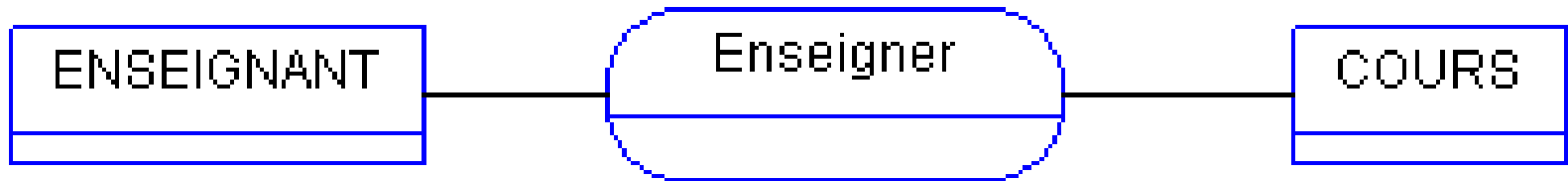
Exemple : L'enseignant enseigne un cours . Le type- relation ici est « Enseigner » qui existe entre les deux type-entités « ENSEIGNANT » et « COURS »



# Modélisation des Données

## Association

- Les type-associations sont représentés par des ellipses (parfois des hexagones) dont l'intitulé décrit le type de relation qui relie les type-entité.
- Les attributs de l'association sont placés dans la partie inférieure au cartouche.



# Modélisation des Données

## Association

- Les type-entités intervenant dans un type-association sont appelés participants.
- La dimension ou l'arité d'un type-association est le nombre de type-entités participant à cette relation
- Un type-association est dit :
  - N-aire : dans le cas général,
  - Binaire : dans le cas où  $n=2$  (la collection contient deux type-entités),
  - Ternaire : dans le cas où  $n=3$ .
  - Les type-associations avec  $n>2$ , bien qu'existants, sont rares et problématiques

# Modélisation des Données

## Attribut/valeur

- Un attribut (ou propriété) est une caractéristique associée à un type-entité ou à un type association.
- Une valeur au niveau du type-entité ou de type-association, chaque attribut possède un domaine qui définit l'ensemble des valeurs possibles qui peuvent être choisies pour lui (entier, chaîne de caractères, booléen...)

ENSEIGNANT
<u>matricule</u>
nom
prénom
date de naissance
spécialité

# Modélisation des Données

## Attribut/valeur

- Règles :
  - Dans le modèle entité-association, chaque attribut est destiné à recevoir une valeur.
  - Un attribut est une donnée élémentaire, ce qui exclut les données calculées ou dérivées.
  - Un attribut peut être placé dans un type-association uniquement lorsqu'il dépend de toutes les entités liées par le type-association.
  - Un type-association peut ne pas posséder d'attribut explicite et cela est relativement fréquent, mais nous verrons qu'il possède au moins des attributs implicites

# Modélisation des Données

## Attribut/valeur

- Un identifiant (ou clé) d'un type-entité ou d'un type-association, est un ensemble minimal d'attributs qui permet d'identifier chaque entité ou association de manière unique. Par exemple dans le cas du type-entité ENSEIGNANT, le matricule est l'identifiant de l'instance ou l'occurrence car nous sommes sûrs que nous ne pouvons pas avoir deux professeurs ayant le même matricule.
- Nous pouvons donc voir qu'un identifiant est représenté dans ce modèle comme un attribut à la seule différence qu'il est souligné

ENSEIGNANT
<u>matricule</u>
nom
prénom
date de naissance
spécialité



# Modélisation des Données

## Attribut/valeur

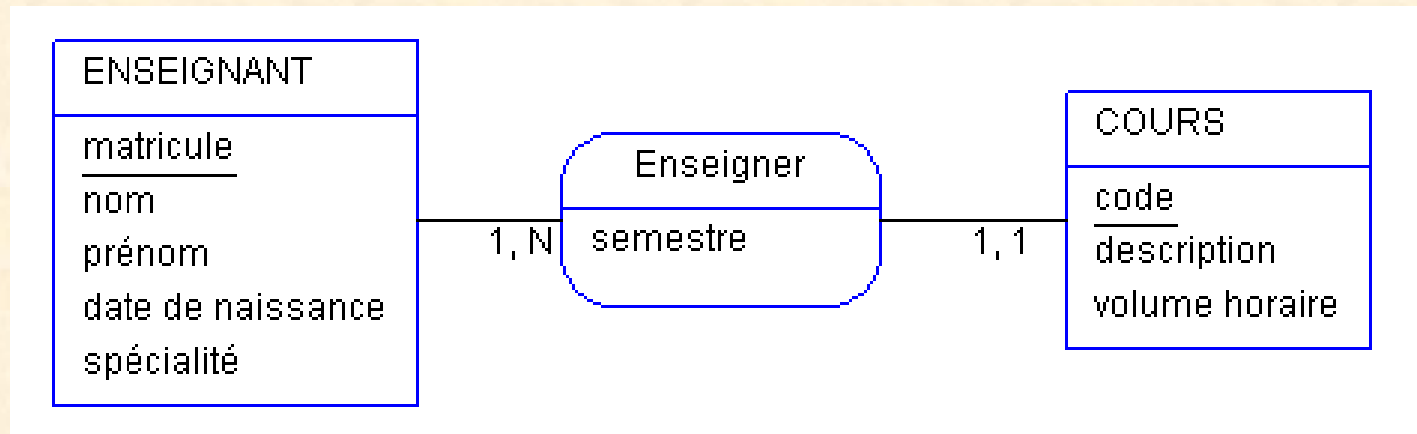
- Un identifiant (ou clé) d'un type-entité ou d'un type-association, est un ensemble minimal d'attributs qui permet d'identifier chaque entité ou association de manière unique. Par exemple dans le cas du type-entité ENSEIGNANT, le matricule est l'identifiant de l'instance ou l'occurrence car nous sommes sûrs que nous ne pouvons pas avoir deux professeurs ayant le même matricule.
- Nous pouvons donc voir qu'un identifiant est représenté dans ce modèle comme un attribut à la seule différence qu'il est souligné

ENSEIGNANT
<u>matricule</u>
nom
prénom
date de naissance
spécialité

# Modélisation des Données

## Cardinalités d'un type-association

- Les cardinalités d'une patte reliant un type-association et un type-entité précisent le nombre de fois minimal et maximal d'interventions d'une entité du type-entité dans une association de type-association. La cardinalité minimale doit être inférieure ou égale à la maximale



- Un enseignant peut Enseigner un ou plusieurs cours.
- Un cours ne peut être Enseigné que par un et un seul enseignant

# Modélisation des Données

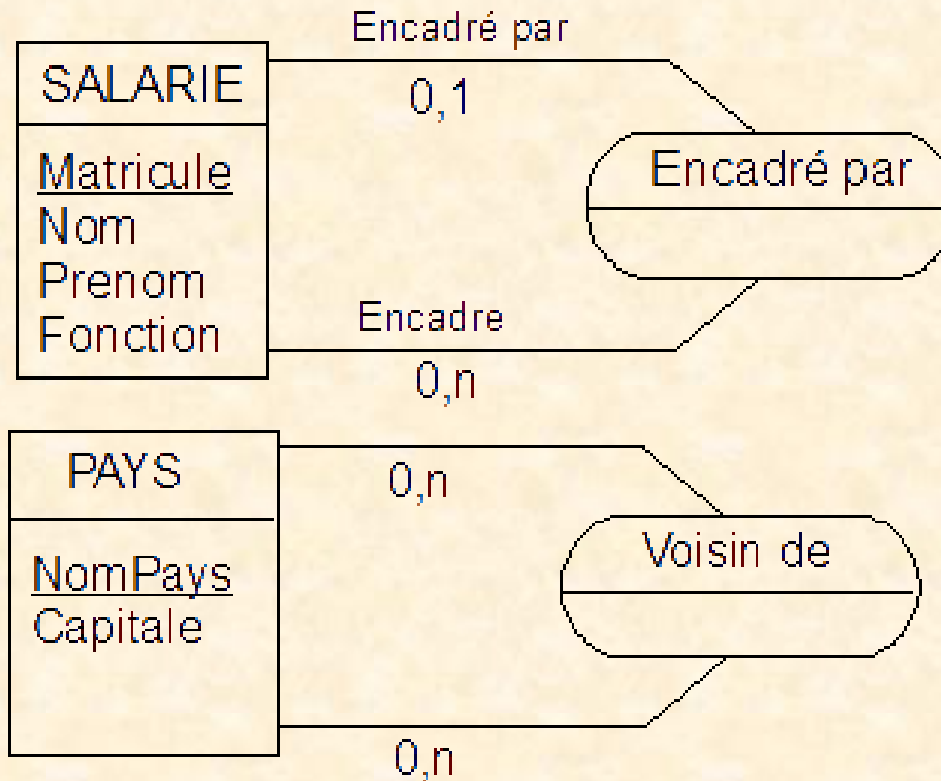
## Cardinalités d'un type-association

- Règles :
  - L'expression de la cardinalité est obligatoire pour chaque patte d'un type-association.
  - Il ne peut y avoir de cardinalité maximale égale à 0, car elle rendrait le type-association inutile.
  - Une cardinalité minimale est toujours 0 ou 1 et une cardinalité maximale est toujours 1 ou n.
  - Si une cardinalité est connue et vaut 2 ou plus, alors nous considérons qu'elle est indéterminée et vaut n. En effet, si cette valeur est définie lors de la conception, il se peut qu'elle évolue dans le futur. Elle est donc considérée n comme inconnue dès la conception.

# Modélisation des Données

## Cardinalités d'un type-association

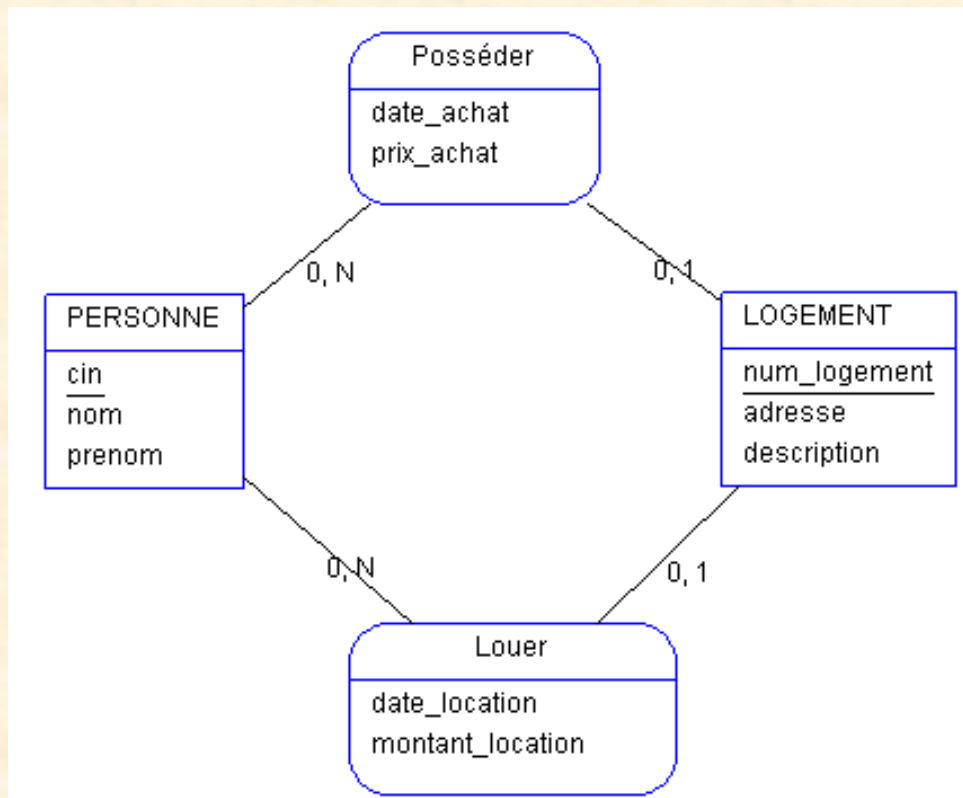
- Un cas spécial d'une association réflexive (unaire) :



# Modélisation des Données

## Cardinalités d'un type-association

- Un cas spécial d'une association plurielle :





# Modélisation des Données

## Identifiant d'un type-association

Catégories de Type-association	Description
Type-association 1 vers 1	La cardinalité maximale sur chacune de ses pattes est 1
Type-association 1 vers n	Une cardinalité maximale est à 1 et une cardinalité maximale est à n.
Type-association n vers n	La cardinalité maximale de chacune de ses pattes est à n.

# Modélisation des Données

## Identifiant d'un type-association

- Règles :
  - L'identifiant d'un type-association **n vers n** est composé des identifiants de ses participants. Cet identifiant n'est pas mentionné sur le modèle, il est implicite.
  - L'identifiant d'un type-association **1 vers n** est composé des identifiants du participant situé du côté de la cardinalité maximale de 1. Cet identifiant n'est pas mentionné sur le modèle, il est implicite.
  - Un type-association **1 vers 1**, possède deux identifiants candidats issus des deux participants. Il suffit d'en choisir un. Cet identifiant n'est pas mentionné sur le modèle, il est implicite (**Il y a des exceptions !!!!**).

# Modélisation des Données

## Exercice de cours

Un établissement de crédit (une banque) souhaite informatiser son SI pour suivre ses clients d'une manière automatique. Il désire ainsi stocker les coordonnées de chaque client (nom, prénom, adresse), et les comptes dont elle dispose ainsi que leur solde (sachant par ailleurs que certains comptes ont plusieurs bénéficiaires). On stockera également les opérations relatives

à ces comptes (retrait et dépôt, avec leur date et le montant).

Questions :

1. Identifier les différentes entités et leurs propriétés pour cette gestion.
2. Préciser les différentes associations entre les entités et ajouter les propriétés pour les associations porteuses de propriétés.
3. Préciser les cardinalités pour les différentes associations

# Transformation du MCD au MLD

- Le Modèle Logique de Données est aussi appelé modèle relationnel.
- Le modèle de données relationnel est fondé sur la notion de relation : un tableau à deux dimensions qui contient un ensemble de n-uplets (les lignes). Quand on se focalise plus sur le stockage, les relations sont souvent appelées des tables et les n-uplets des enregistrements. Les entrées dans les tables sont appelées des valeurs.
- Le MCD ne peut pas être implanté dans une base de données sans modification.
- Il est obligatoire de transformer ce modèle. On dit qu'on effectue un passage du modèle conceptuel de données vers le modèle logique de données.
- Le MLD pourra être implanté dans une base de données relationnelle

# Transformation du MCD au MLD

## Transformation des entités

- Toute entité est transformée en table.
- Une ligne correspond à un enregistrement, une colonne correspond à un champ
- Son identifiant devient la clé primaire de la table.
- Les autres propriétés deviennent les attributs de la table.



# Transformation du MCD au MLD

## Transformation des entités

ENSEIGNANT
<u>matricule</u>
nom
prenom
cin
spécialité

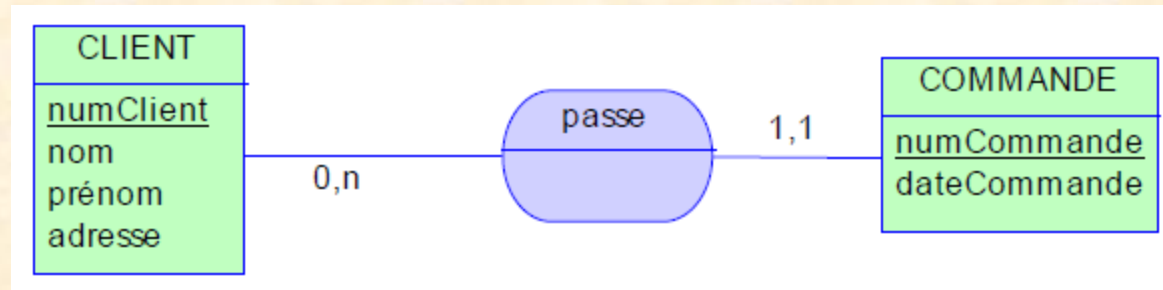


ENSEIGNANT (matricule, nom, prenom, cin, spécialité)

# Transformation du MCD au MLD

## Transformation des associations du type $(x,n) - (x,1)$

- Nous dupliquons la clé primaire de la table basée sur l'entité à cardinalité  $(x,n)$  dans la table basée sur l'entité à cardinalité  $(x,1)$ . Cet attribut est appelé clé étrangère.



CLIENT(numClient , nom , prenom , adresse)

COMMANDE(numCommande , dateCommande , #numClient)

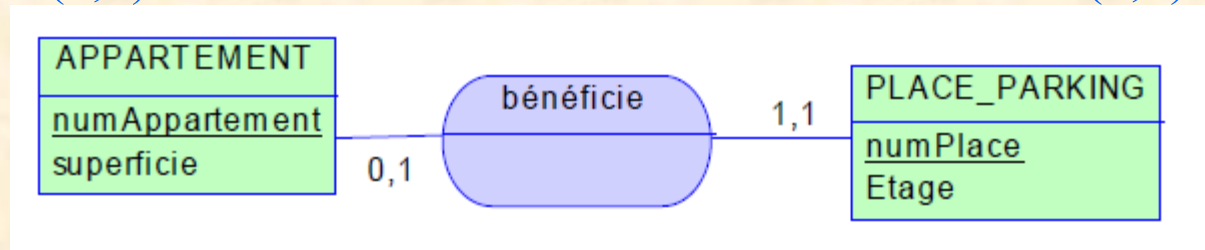
# Transformation du MCD au MLD

## Transformation des relations binaires du type $(x,1) - (x,1)$

Nous devons distinguer plusieurs cas :

- Relation binaire (0,1)-(1,1)

Nous dupliquons la clé de la table basée sur l'entité à cardinalité (0,1) dans la table basée sur l'entité à cardinalité (1,1).



APPARTEMENT (numAppartement , superficie)

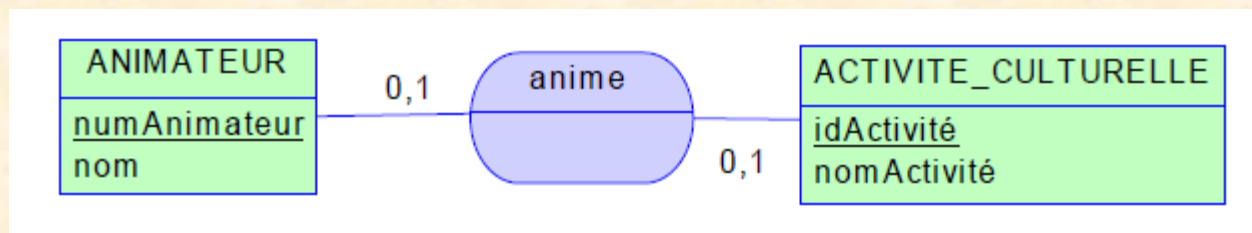
PLACE\_PARKING (numPlace , Etage , #numAppartement)

# Transformation du MCD au MLD

## Transformation des relations binaires du type (x,1) – (x,1)

- Relation binaire (0,1)-(0,1)

Nous créons une table qui reflète cette association ayant comme clé primaire une clé composée des clés primaires des 2 tables.



ANIMATEUR (numAnimateur, nom)

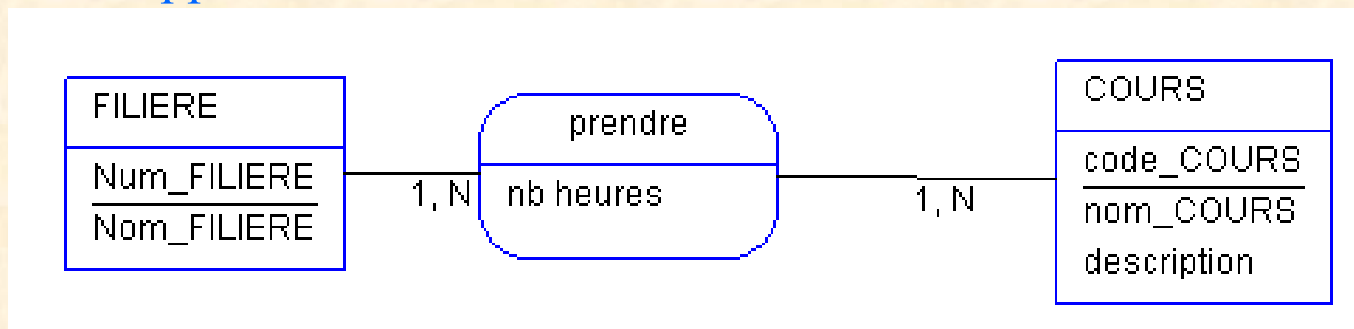
ACTIVITE\_CULTURELLE (idActivite , nomActivite)

ANIMER (numAnimateur, idActivite)

# Transformation du MCD au MLD

## Transformation des relations binaires du type (x,n) – (x,n)

Nous créons une table supplémentaire ayant comme clé primaire une clé composée des clés primaires des 2 tables. Lorsque la relation contient elle-même des propriétés, celles-ci deviennent attributs de la table supplémentaire. Une propriété de la relation qui est soulignée devra appartenir à la clé primaire composée de la table supplémentaire.



FILIERE (Num\_FILIERE, Nom\_FILIERE)

COURS (code\_COURS, nom\_COURS, description\_COURS)

prendre (Num\_FILIERE, code\_COURS, nb\_heures)

# Les règles de Normalisation

## Première Forme Normale-

A un instant donné dans une table, un attribut ne peut prendre qu'une valeur et non pas, un ensemble ou une liste de valeurs. Si un attribut prend plusieurs valeurs, alors ces valeurs doivent faire l'objet d'une table supplémentaire, en association avec la première.

Ex : l'attribut « auteurs\_livre » n'est pas atomique et prend plusieurs valeurs

LIVRE (num\_LIVRE, titre\_LIVRE, auteurs\_livre)



LIVRE (num\_LIVRE, titre\_LIVRE)

AUTEUR (num\_AUTEUR, nom\_AUTEUR)

ECRIRE (num\_LIVRE, num\_AUTEUR, nbre\_de\_page\_ecrit)



# Les règles de Normalisation

## Deuxième Forme Normale-

La deuxième forme normale (2FN) ne concerne que les tables qui ont une clé composée de plusieurs attributs.

Pour être en deuxième forme normale (2FN), il faut déjà être en 1FN et en plus respecter la règle suivante : Un attribut non clé ne dépend pas d'une partie de la clé mais de toute la clé en entier.

Ex : l'attribut « **nom\_AUTEUR** » ne dépend pas de la totalité de la clé. Il dépend seulement du « num\_AUTEUR »

ECRIRE (num\_LIVRE, num\_AUTEUR, nbre\_de page\_ecrit, **nom\_AUTEUR**)



ECRIRE (num\_LIVRE, num\_AUTEUR, nbre\_de page\_ecrit)

# Les règles de Normalisation

## Troisième Forme Normale-

Tous les attributs d'une entité doivent dépendre directement de son identifiant et d'aucun autre attribut.

Ex : l'attribut « **nom\_client** » ne dépend pas de la clé de la table COMMANDE. Il dépend seulement du « numClient » qui est un attribut non clé.

CLIENT(numClient , nom , prenom , adresse)

COMMANDE(numCommande ,dateCommande , #numClient, **nom\_client**)

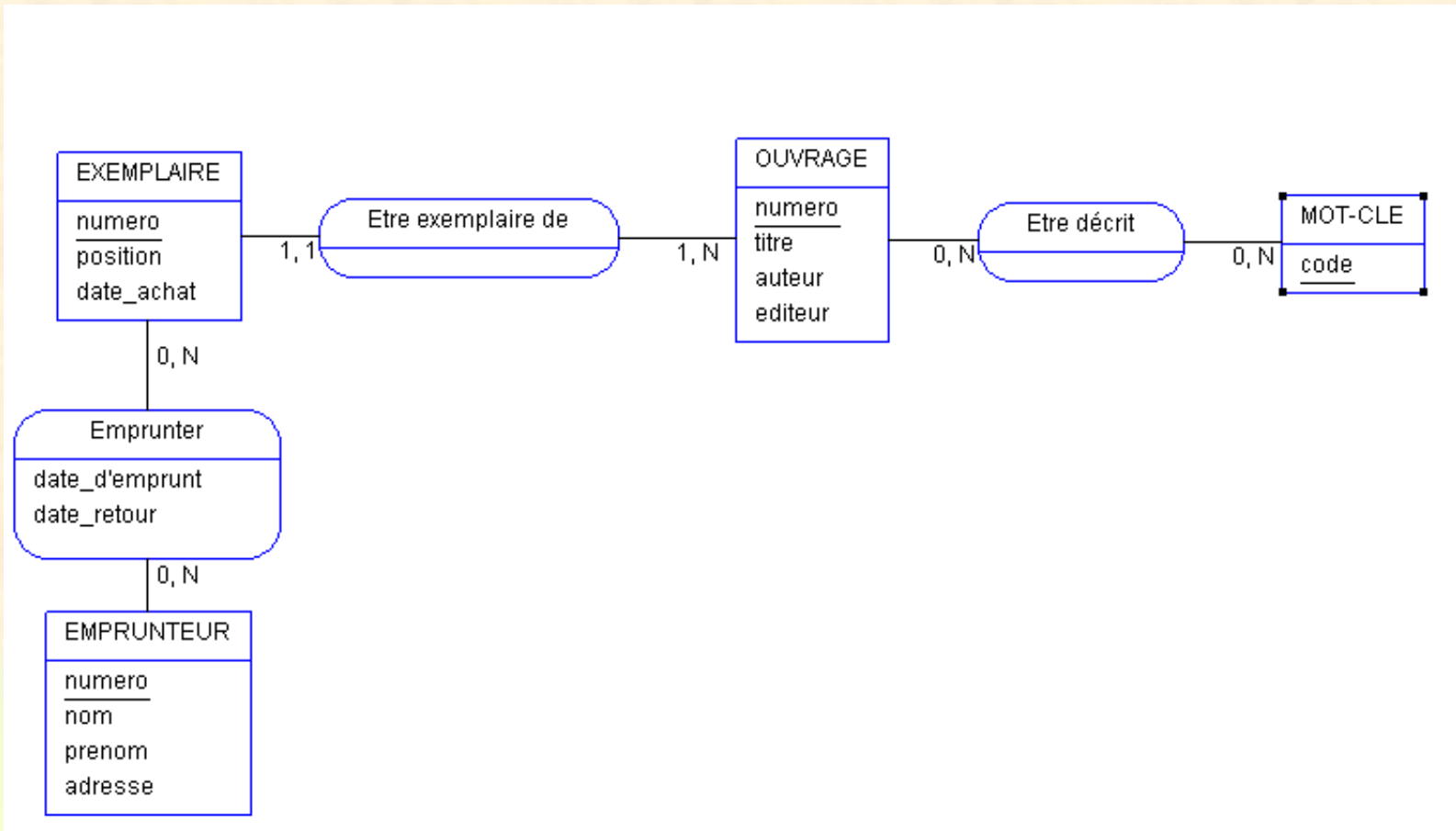


CLIENT(numClient , nom , prenom , adresse)

COMMANDE(numCommande ,dateCommande , #numClient)

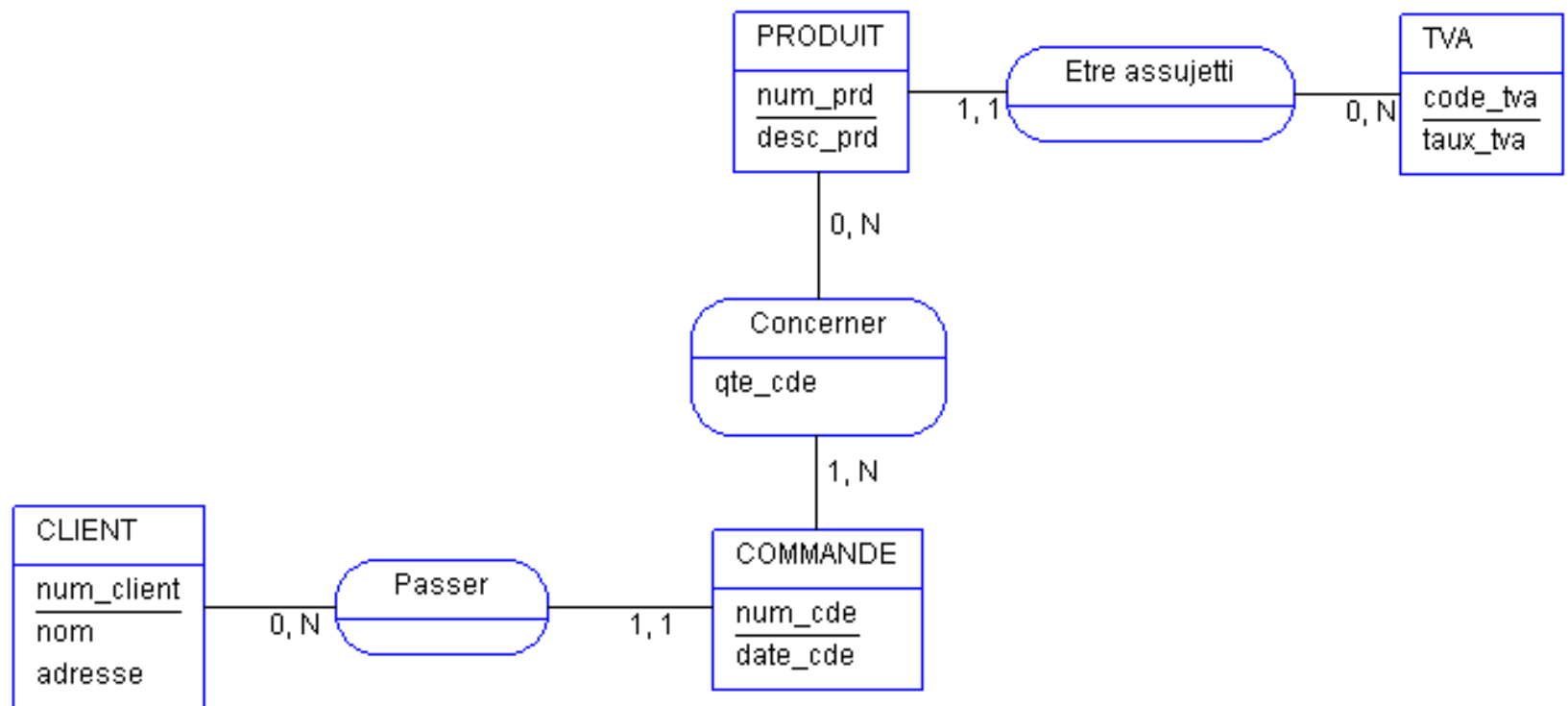
# Les règles de Normalisation

## Le passage du MCD au MLD (Exercice 1 TD1)



# Les règles de Normalisation

## Le passage du MCD au MLD (Exercice 3 TD1)



# Objectifs de la deuxième partie

- Généralités sur les Bases de Données (BD)
- Généralités sur les Systèmes de Gestion de Bases de Données (SGBD)

# Base de Données

- Une base de données (son abréviation est BD, en anglais DB, database) est une entité (informatique) dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents.
- Une base de données permet de mettre des données à la disposition d'utilisateurs pour une consultation, une insertion ou bien une mise à jour, tout en s'assurant des droits accordés à ces derniers.
- Une base de données peut être locale, c'est-à-dire utilisable sur une machine par un utilisateur, ou bien répartie, c'est-à-dire que les informations sont stockées sur des machines distantes et accessibles par réseau



# Base de Données

- Il existe plusieurs types de base de données :
  - Base de données hiérarchique : Les liens sont exclusivement de type 1 : N. Le schéma est exprimé au moyen d'un arbre ordonné, ou arborescence.
  - Base de données relationnelle : elles ont été inventées en 1970 par E.F. Codd . Elles sont constituées d'un ensemble de tableaux. Au sein de ces tableaux, les données sont classées par catégorie. Chaque tableau comporte au moins une colonne correspondant à une catégorie. Chaque colonne comporte un certain nombre de données correspondant à cette catégorie.
  - Base de données NoSQL : Elles sont utiles pour les larges ensembles de données distribuées. Si une entreprise doit analyser d'importantes quantités de données non structurées la database NoSQL est idéale. Avec l'essor du Big Data, les bases de données NoSQL sont de plus en plus utilisées.

# SGBD

- Un système de gestion de base de données (SGBD) est une solution qui permet à un ou plusieurs utilisateurs de créer et d'accéder à des données contenues dans une base de données.
- Le SGBD prend en charge les requêtes des utilisateurs ainsi que celles provenant d'autres programmes
- le SGBD garantit l'intégrité des données (s'assure qu'elles soient toujours disponibles et qu'elles soient organisées de façon cohérente) ainsi que la sécurité (s'assure que seulement ceux disposant des privilèges d'accès puissent accéder aux données).
- Le SGDB le plus courant est le système de gestion de base de données relationnelle (SGBDR). Le langage standard utilisé est le SQL (Structured Query Language).

# Exemples de SGBD

- Oracle est un SGBD relationnel (et relationnel-objet dans ses dernières versions) très reconnu pour les applications professionnelles.
- PostgreSQL est un SGBD relationnel et relationnel-objet très puissant qui offre une alternative open source aux solutions commerciales comme Oracle ou IBM.
- MySQL est un SGBD relationnel libre (licence GPL), simple d'accès et très utilisé pour la réalisation de sites Web dynamiques.
- SQLServer un système de gestion de base de données (SGBD) en langage SQL incorporant entre autres un SGBDR développé et commercialisé par la société Microsoft