

Analyse et conception de base de données

PLAN

Introduction

- I. Notion analyse et conception
- II. Concepts Analyse et Conception
- III. SQL

INTRODUCTION

La gestion d'informations se faisait jadis à partir de fiches manuscrites puis dactylographiées, classées bien souvent selon un index . Avec l'informatique, un certain nombre d'opérations peuvent être automatisées et l'enregistrement des informations est donc passé de la fiche manuelle à un support informatique (Bande magnétique, disque dur, disquette etc...). Ces informations peuvent être organisées dans des fichiers dans un premier temps. Ces fichiers se sont organisés en un véritable ensemble dévolu à la gestion des informations de l'entreprise.

C'est ainsi qu'un ensemble structuré de données, dénommé **Base de données**, enregistrées sur des supports, accessibles par l'ordinateur, représentant les informations du monde réel et pouvant être interrogées et mises à jour par une communauté d'utilisateurs, marque la révolution. La base de données est destinée à la gestion, au stockage, à l'actualisation et à la consultation d'entités de différentes natures (et de leurs données), sachant que ces entités ont un lien les unes avec les autres.

I.

NOTION ANALYSE ET CONCEPTION

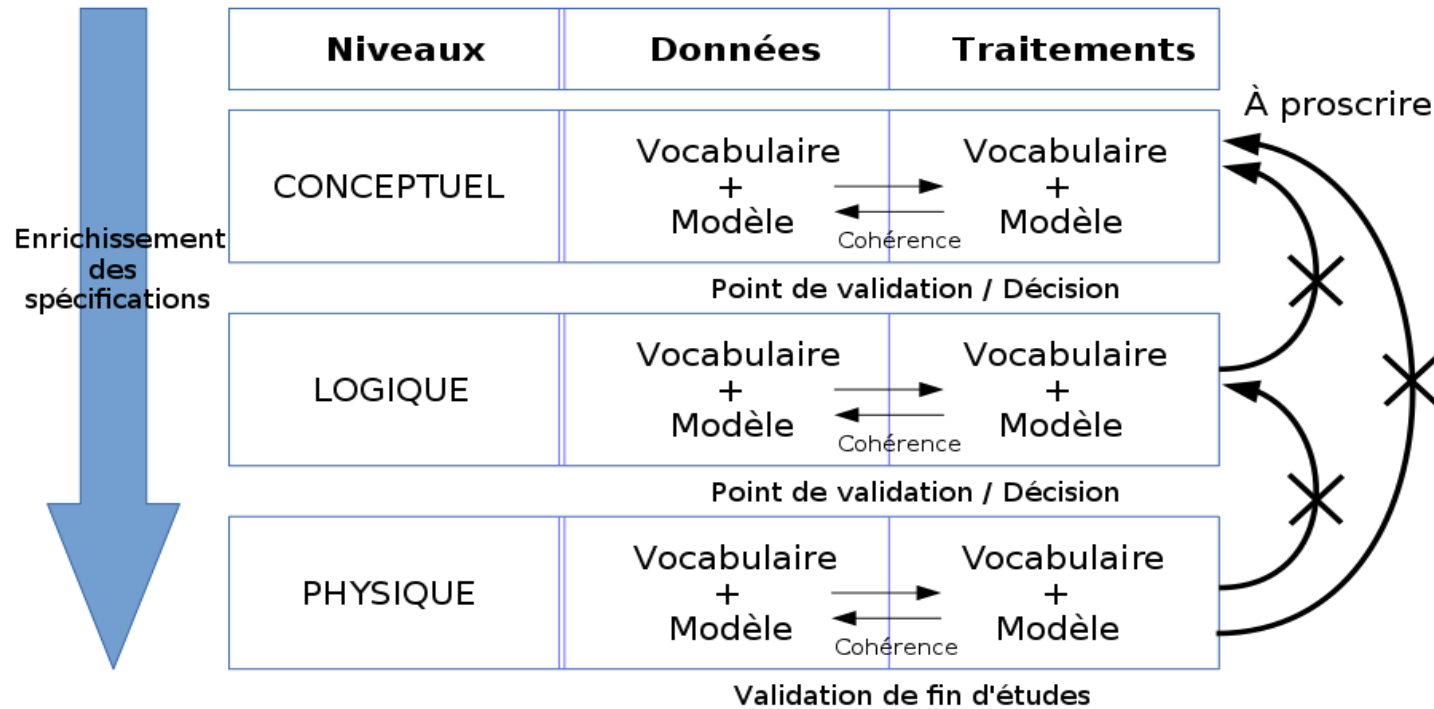
Concevoir un système d'informations c'est avant tout développer des plans de réalisation qui répondent aux besoins des utilisateurs tels qu'exprimés dans la phase d'analyse. La modélisation est une étape fondamentale dans la conception de logiciels. La conception d'une base de données est un processus qui consiste en l'observation d'une situation réelle pour aboutir à la définition de la base de données correspondante.

Le monde informatique dispose de plusieurs méthodes de modélisation de base de données. Dans cette partie on va présenter la modélisation systématique **MERISE** et celle graphique **UML** (unified modeling language).

- **MERISE**

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).

La méthode Merise d'analyse et de conception propose une démarche articulée simultanément selon 3 axes(cycle de vie, cycle de décision et cycle d'abstraction) pour hiérarchiser les préoccupations et les questions auxquelles répondre lors de la conduite d'un projet.



La méthode MERISE préconise d'analyser séparément données et traitements, à chaque niveau.

L'étude conceptuelle Merise s'attache aux invariants de l'entreprise ou de l'organisme du point de vue du métier : quelles sont les activités, les métiers gérés par l'entreprise, quels sont les grands processus traités, de quoi parle-t-on en matière de données, quelles notions manipule-t-on ?... et ce indépendamment des choix techniques (comment fait-on ?) ou d'organisation (qui fait quoi ?) qui ne seront abordés que dans les niveaux suivants.

Au niveau conceptuel on veut décrire, après abstraction, le modèle (le système) de l'entreprise ou de l'organisme :

- le Modèle conceptuel des données (ou MCD), schéma représentant la structure du système d'information, du point de vue des données, c'est-à-dire les dépendances ou relations entre les différentes données du système d'information (par exemple : le client, la commande, les produits, etc.),
- et le Modèle conceptuel des traitements (ou MCT), schéma représentant les traitements, en réponse aux événements à traiter (par exemple : la prise en compte de la commande d'un client).

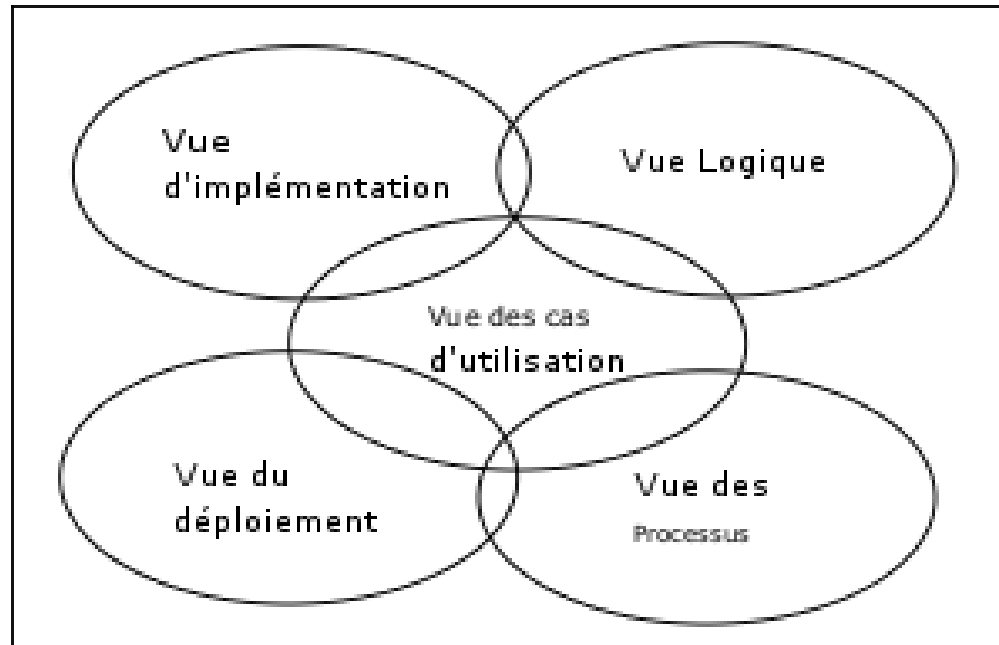
Dans l'idéal, le MCD et le MCT d'une entreprise sont stables, à périmètre fonctionnel constant, et tant que le métier de l'entreprise ne varie pas. La modélisation ne dépend pas du choix d'un progiciel ou d'un autre, d'une automatisation ou non des tâches à effectuer, d'une organisation ou d'une autre, etc.

- UML

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie.

L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientées objet.

Vue UML:



► Concept de modélisation spécifié par l'UML:

Le développement d'un système est axé sur trois modèles de systèmes globaux :

- **Fonctionnel** : ce sont des diagrammes de cas d'utilisation, qui décrivent la fonctionnalité du système du point de vue de l'utilisateur.
- **Objet** : ce sont des diagrammes de classes qui décrivent la structure d'un système en termes d'objets, attributs, associations et opérations.
- **Dynamique** : ce sont des diagrammes d'interaction, diagrammes états-transitions et diagrammes d'activités utilisés pour décrire le comportement interne du système.

On visualise ces modèles de système grâce à deux types distincts de diagrammes : structurel et comportemental.

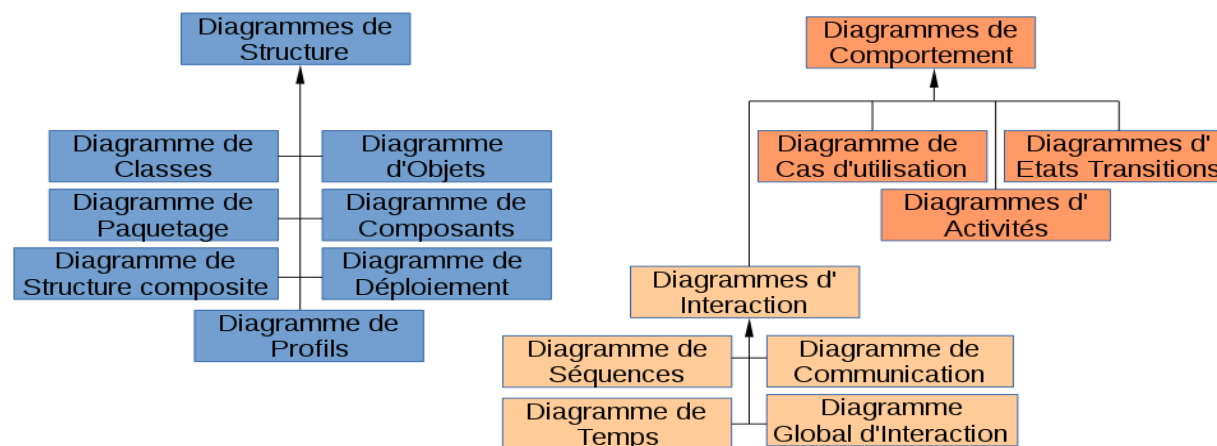
► Concept orienté objet dans l'UML:

Les objets dans UML sont des entités du monde réel qui existent autour de nous. Dans le développement de logiciels, les objets peuvent être utilisés pour décrire - ou modéliser - le système en cours de création sous un angle pertinent. Les objets permettent également la décomposition de systèmes complexes en éléments compréhensibles qui permettent de construire les pièces une par une.

Voici quelques concepts fondamentaux d'un monde orienté objet :

- **Objets** Représentent une entité et le module de base
- **Classe** Plan d'un objet
- **Abstraction** Comportement d'une entité du monde réel
- **Encapsulation** Mécanisme qui consiste à relier les données et à les cacher du monde extérieur
- **Héritage** Mécanisme par lequel de nouvelles classes sont créées à partir d'une classe existante
- **Polymorphisme** Définit le mécanisme sous différentes formes.

- Les *diagrammes* sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie. Il en existe quatorze depuis UML 2.3.



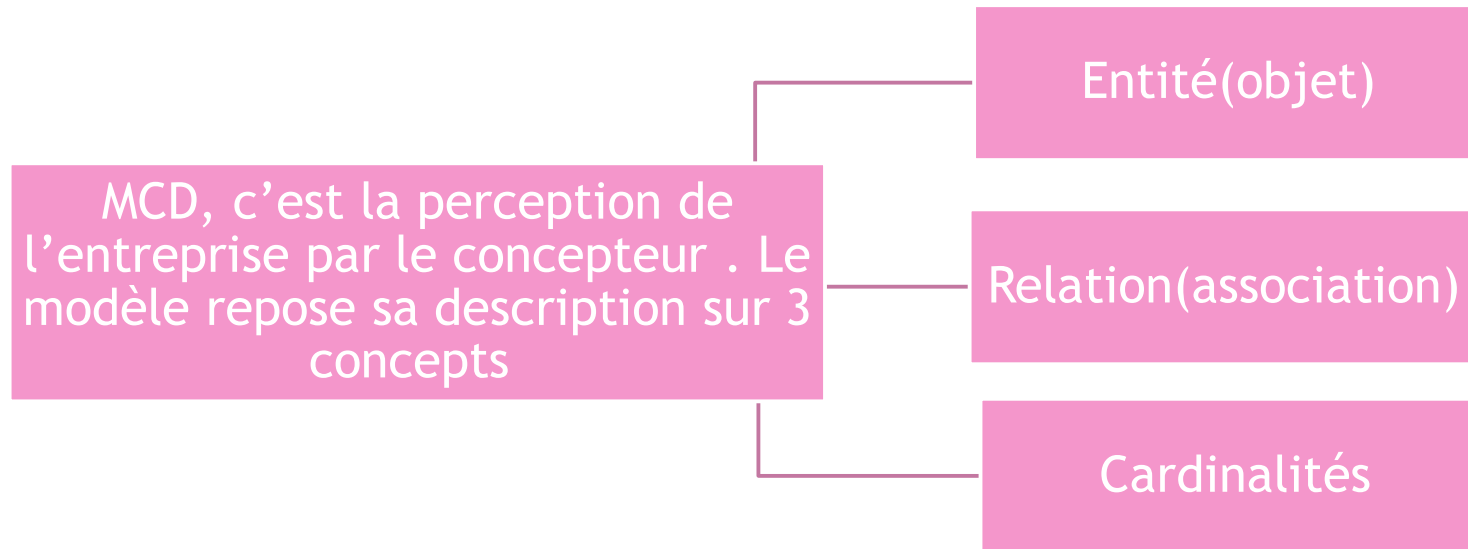
- Etude comparative

Simplement, Merise est orienté "relationnel" et UML est orienté "objet".
De plus Merise a pour but la modélisation des traitements et la conception d'un modèle de base de données et UML a pour but lui de générer un modèle de programme orienté objet.

Merise est plus "abstrait" que UML qui lui représente des objets "plus réaliste" (plus proche de la BDD).

II. CONCEPT ANALYSE ET CONCEPTION

1. MCD(modèle conceptuel des données)



Entité

Une entité est une représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire. Chaque entité est composée des propriétés, données élémentaires permettant de la décrire.

Propriétés

Une propriété est un attribut ou une donnée élémentaire perçue sur le Système d'Information, elle peut concerner une entité ou une relation. En d'autres termes, une propriété est une information rattachée à une entité.

Identifiant de l'entité

C'est une propriété particulière telle qu'à chaque valeur de la propriété corresponde une et une seule occurrence de l'Entité. Il permet d'identifier d'une manière unique et sans ambiguïté une occurrence.

Occurrence

L'occurrence d'une entité correspond au nombre de fois que cette entité est valorisée.

Relation (Association)

Une association (relation) est un lien sémantique entre plusieurs entités. Selon le nombre d'entités qu'elle relie, une association peut être :

Réursive (réflexive) :
relie la même entité ;

Binaire : relie deux entités
Ternaire : relie trois entités
\$ N-aire : relie n-entités.

Cardinalité

Une cardinalité est le nombre de fois minimum et maximum qu'une entité participe à une relation. En bref la démarche de la réalisation du MCD peut être résumée comme suit :

Déterminer la liste des entités
Pour chaque entité, déterminer la liste des propriétés et parmi celle-ci, déterminer l'identifiant

Déterminer les relations entre les entités ;
Pour chaque relation, déterminer les propriétés si possibles ;
Placer les cardinalités ;
Déterminer le CIF (Contrainte d'Intégrité Fonctionnelle) ;
Vérifier les règles de normalisation.

► Formalisme de représentation

Après avoir conçu le MCD, on doit le vérifier sur base des quelques règles :

Concernant les Entités

- Il ne doit exister un identifiant pour chaque entité
- Toutes les propriétés autre que l'identifiant doivent être en dépendance fonctionnelle complète et direct de l'identifiant.

Concernant les relations

- Toute les propriétés d'une relation doivent dépendre complètement de l'identifiant de la relation

Concernant l'ensemble du MCD

- Une propriété ne peut apparaitre qu'une et une seule fois dans le même MCD
- Les propriétés qui sont les résultats d'un calcul ne doivent pas en principe se figurer dans un MCD sauf si elles sont indispensables à la compréhension de celui-ci.

Règle de bonne manière dans le schéma du MCD

- Pour les entités, utilisez le nom commun au pluriel ;
- Pour les associations, utilisez un verbe à l'infinitif ;
- Pour les attributs (propriétés), utilisez un nom au singulier éventuellement accompagné du nom de l'entité ;
- Préférez un identifiant court pour rendre la recherche plus rapide.

2. MLD(modèle logique des données)

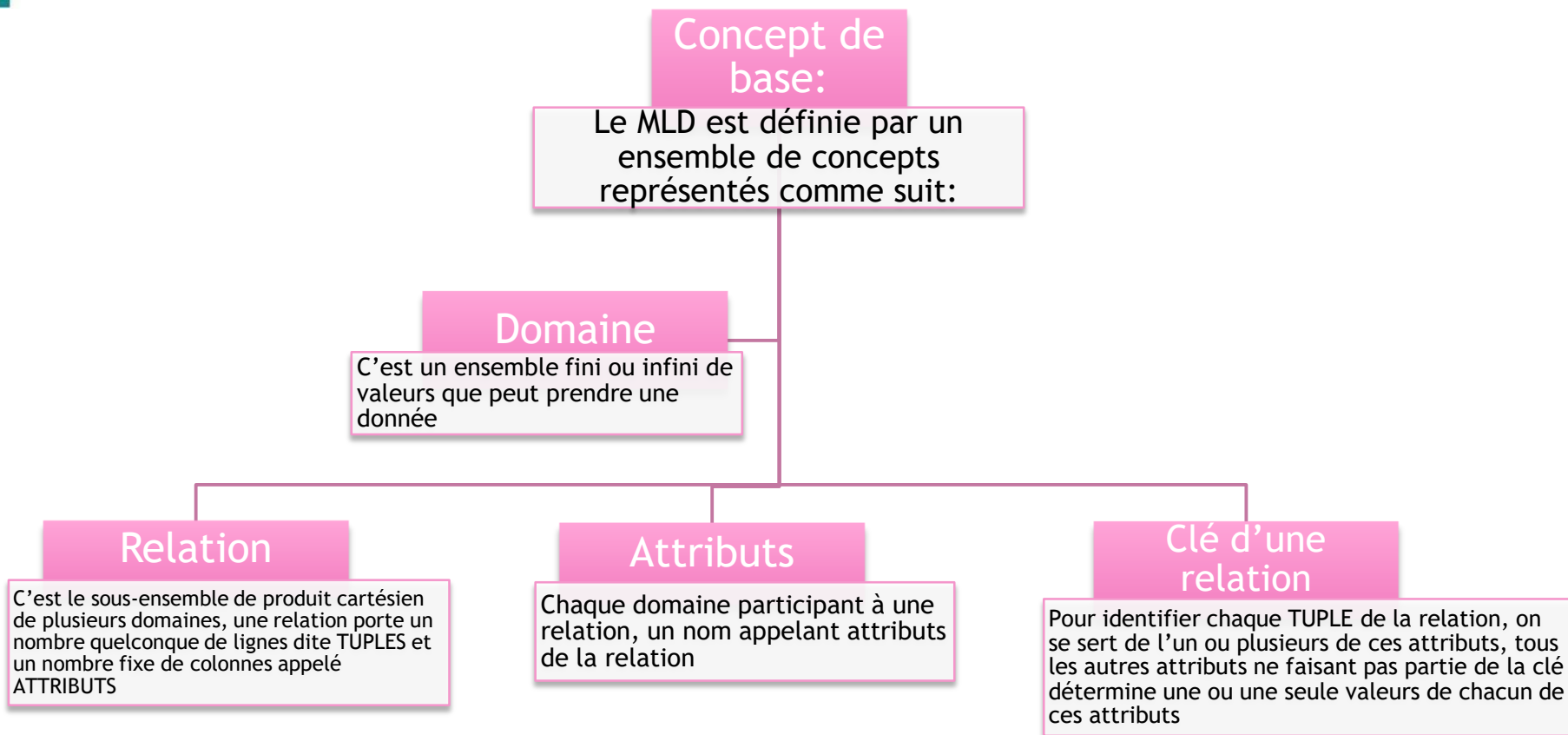
Le modèle logique de données est une représentation du MCD valide en fonction du matériel et logique utilisés dans l'automatisation. Le modèle logique est le modèle conceptuel plus la réponse aux contraintes d'organisation des données. En outre, il nous permet de préfigurer le temps d'accès et l'espace nécessaire.

- **Modèle logique de données relationnel**

C'est grâce à toutes les opérations précédentes que l'ensemble des tables de la base de données vont pouvoir être structurées de manière simple et rapide. Ce sont les entités, les cardinalités maximales et minimales qui vont jouer le rôle essentiel pendant la réalisation du MLDR. Le modèle logique de données relationnel s'obtient soit à partir du modèle conceptuel de données ou soit à partir du modèle organisationnel de données.

- **Modèle relationnel**

Le modèle de données relationnel est plus approprié à la méthode MERISE, dans le modèle relationnel on ne raisonne pas en terme d'enregistrement mais en terme de relation qui existe entre ensemble de propriétés.



- **Règles de passage du MCD au MLDR**

Toutes les entités du MCD deviennent des tables de la base de données. Les propriétés des objets du MCD deviennent des attributs dans le modèle logique de données relationnel, l'identifiant de l'entité devient la clé primaire de la table.

La clé primaire de la table à la cardinalité (X, n) devient une clé étrangère dans la table à la cardinalité $(X, 1)$, $X=1$ ou 0 . Il y a création d'une table supplémentaire ayant comme clé primaire, une clé composée des identifiants de 2 entités lorsque les cardinalités sont $(X, n)-(X, n)$, $X=1$ ou 0 .

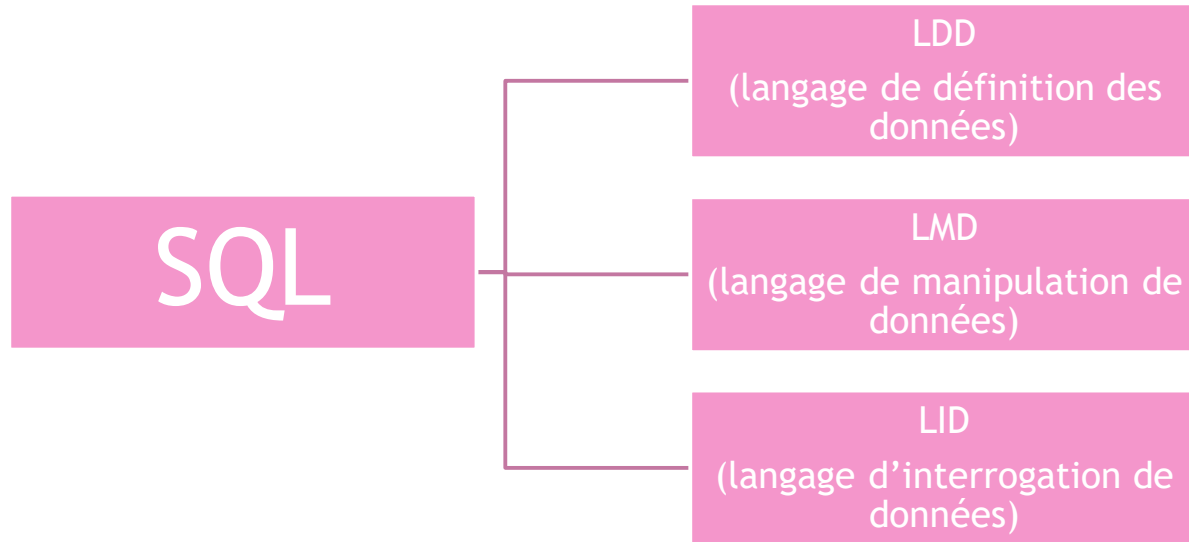
Les règles de passage du MCD au MLDR ne sont pas exhaustives.

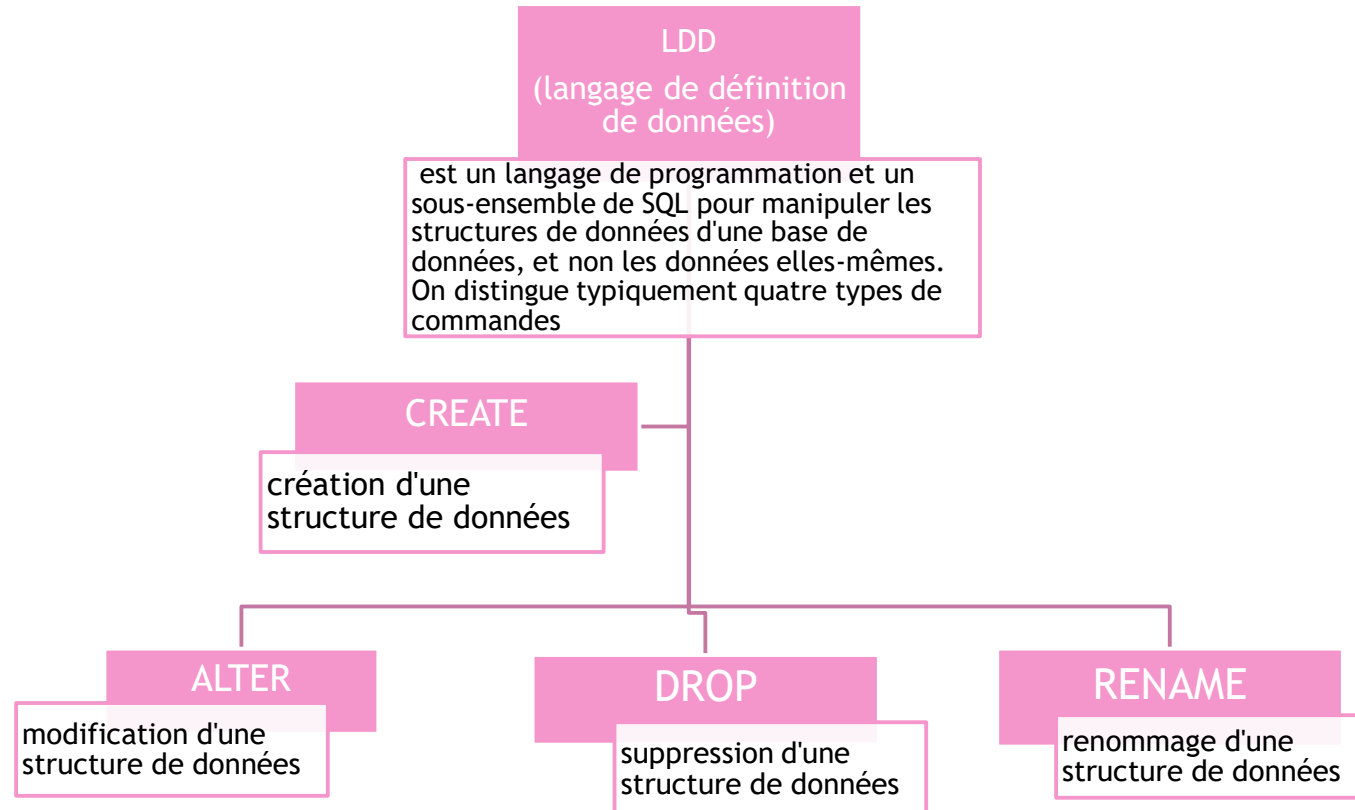
III. SQL (STRUCTURED QUERY LANGUAGE)

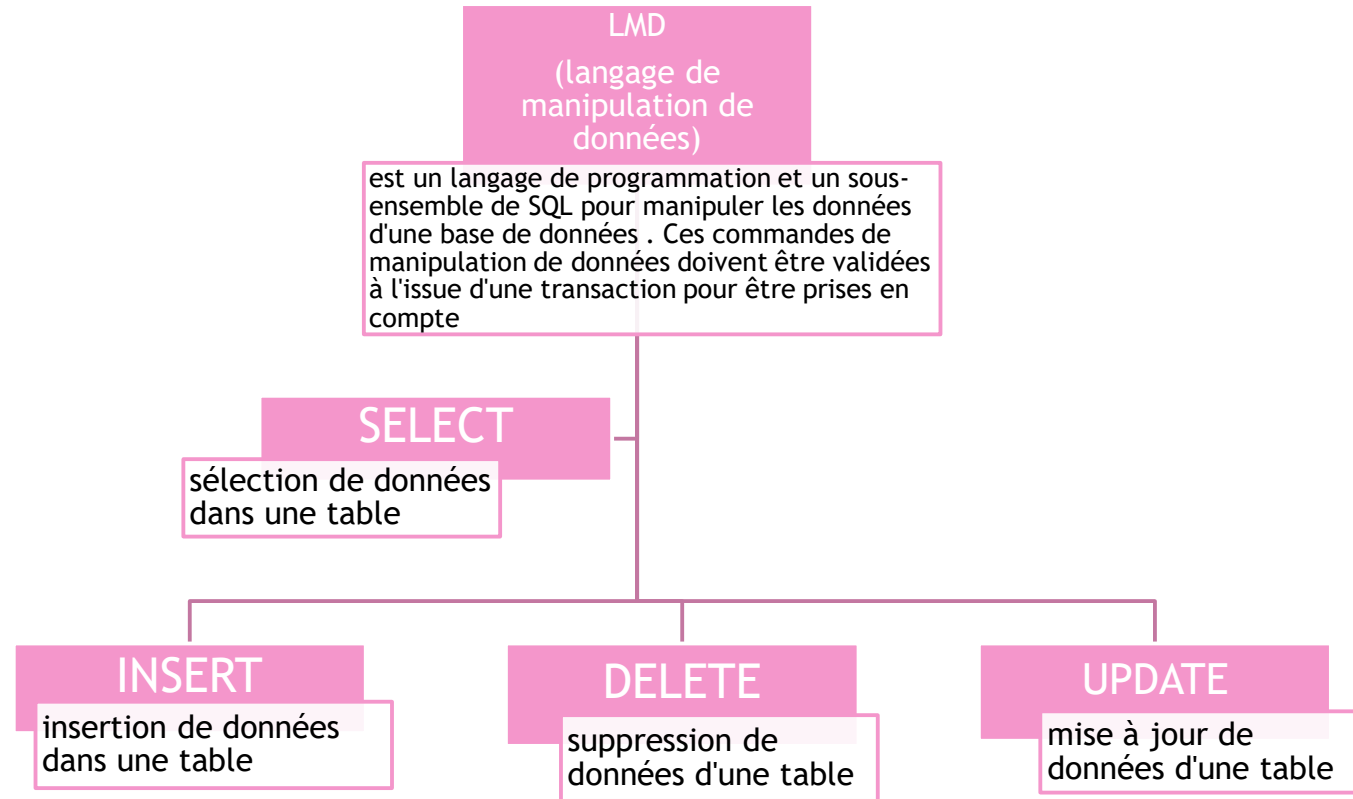
DÉFINITION

Le SQL (Structured Query Language) est un langage permettant de communiquer avec une base de données. Ce langage informatique est notamment très utilisé par les développeurs web pour communiquer avec les données d'un site web.

Les instructions SQL s'écrivent d'une manière qui ressemble à celle de phrases ordinaires en anglais. Cette ressemblance voulue vise à faciliter l'apprentissage et la lecture et couvrent 3 domaines.







Langage d'interrogation des données (LID)

Le langage d'interrogation de données (LID) permet d'établir une combinaison d'opérations portant sur des tables (relation). Le résultat de cette combinaison d'opérations est lui-même une table dont l'existence ne dure qu'un temps.

On observera qu'en fait l'ordre **SQL SELECT** est composé de six clauses dont quatre sont optionnelles. La plupart du temps, la difficulté réside dans la compréhension de la différence entre le filtre WHERE et le filtre HAVING.

Le **filtre WHERE** permet de **filtrer les données des tables** tandis que le **filtre HAVING** permet de **filtrer les données du résultat**.

Le tableau suivant représente les clauses de l'ordre **SELECT**

SELECT	NomChamp1, NomChamp2	Champs à projeter ou à calculer ou fonction d'agrégat
FROM	TABLE1, TABLE2...	Tables utiles à la requête
WHERE	Expression AND/OR	Jointures et restrictions
GROUP BY	NomChamp	Regroupement de résultat d'opérations d'agrégat
HAVING	Expression	Restriction sur l'affichage des résultats d'opérations d'agrégat
ORDER BY	NomChamp [ASC]/DESC	Critères de tri

► La commande SELECT et la clause FROM

- ❑ Le **SELECT** est la commande de base du SQL destinée à extraire des données d'une base ou calculer de nouvelles données à partir d'existantes. La syntaxe est la suivante :

```
SELECT [DISTINCT ou ALL] * ou liste de colonnes  
FROM nom de table ou de la vue  
[WHERE prédicats]  
[GROUP BY ordre des groupes]  
[HAVING condition]  
[ORDER BY ] liste de colonnes
```

A côté de cette syntaxe, il existe les agrégats suivants :

- **SUM** [...] : renvoie à la **somme** d'un champ (valeurs de données de type numérique ou date/heure).
- **AVG** [...] : renvoie la **moyenne** d'un champ (valeurs de données de type numérique ou date/heure).
- **MIN** [...] : renvoie la **valeur minimale** d'un champ (valeurs de données de type numérique, date et texte).
- **MAX** [...] : renvoie la **valeur maximale** d'un champ (valeurs de données de type numérique, date et texte).
- **COUNT** [...] : renvoie le **nombre d'enregistrements** de la table.

- Il est possible de surnommer une table dans la clause **FROM**, dans ce cas, la syntaxe de la partie FROM de la commande SELECT est la suivante :
FROM nom_de_table ou nom_de_la_vue surnom

MERCI