

Papa Ibrahima Seck DIENG

Documentation sur MCD, MLD et SQL

I. Notion d'analyse et de conception :

1) Mérise :

MERISE est une méthode française née dans les années 70, développée initialement par Hubert Tardieu. Elle fut ensuite mise en avant dans les années 80, à la demande du Ministère de l'Industrie qui souhaitait une méthode de conception des SI.

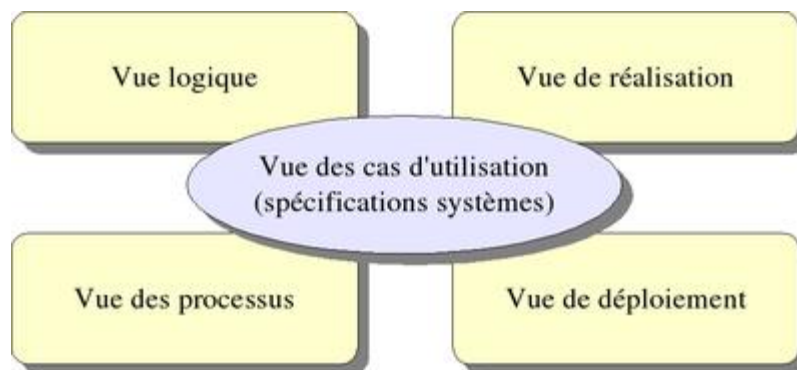
MERISE est donc une méthode d'analyse et de conception des SI basée sur le principe de la séparation des données et des traitements. Elle possède un certain nombre de modèles (ou schémas) qui sont répartis sur 3 niveaux :

- Le niveau conceptuel,
- Le niveau logique ou organisationnel,
- Le niveau physique.

2) UML :

UML est un langage unifié pour la modélisation objet et la conception d'applications de bases de données. Il est né en octobre 1994 chez Rational Rose Corporation. Il est conçu pour modéliser divers types de systèmes, de taille quelconque et pour tous les domaines d'application (gestion, système embarqué, etc.).

UML est fortement inspiré de l'approche 4+1 vues (figure 2) définies par P. Kruchten pour exprimer les diverses perspectives de l'architecture d'un système informatique.



3) Etude comparative :

MERISE (Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise) est une méthode d'analyse et de réalisation des systèmes d'information qui est élaborée en plusieurs étapes: schéma directeur, étude préalable, étude détaillée et la réalisation. Alors que UML (Unified Modeling Language), est un langage de modélisation des systèmes standard, qui utilise des diagrammes pour représenter chaque aspect d'un système : statique, dynamique,...en s'appuyant sur la notion d'orienté objet qui est un véritable atout pour ce langage.

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

II. Concepts Analyse et Conception :

1. MCD :

Le MCD est une représentation graphique de haut niveau qui permet facilement et simplement de comprendre comment les différents éléments sont liés entre eux à l'aide de diagrammes codifiés dont les éléments suivants font partie :

- Les entités (1 rectangle = 1 objet) ;
- Les propriétés (la liste des données de l'entité) ;
- Les relations qui expliquent et précisent comment les entités sont reliées entre elles (les ovales avec leurs « pattes » qui se rattachent aux entités) ;
- Les cardinalités (les petits chiffres au dessus des « pattes »).

Utilisé assez tôt en conception de base de données, le MCD évoluera ensuite vers les autres outils de Merise, à savoir le MPD et le MLD.

2. MLD :

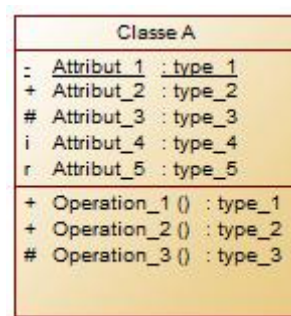
Il est aussi appelé modèle relationnel (lorsqu'on travaille avec une base de données relationnelle. On emploie souvent l'abréviation suivante : MLD : Modèle logique des données

Et quelquefois, les abréviations suivantes sont employées : - MLDR : Modèle logique de données relationnelles - MRD : Modèle relationnel de données - MLRD : Modèle relationnel logique de données Le MCD (Modèle Conceptuel de Données) 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.

3. Diagramme de classes :

Le diagramme de classe est une représentation abstraite et statique des classes qui composent le système et leurs relations.

Une classe est représentée par un rectangle ‘‘compartmenté’’ en 3 parties.



- Le premier compartiment contient le nom de la classe
- Le deuxième compartiment contient les attributs de la classe.
- Le troisième compartiment contient les méthodes ou opérations de la classe.

Les attributs et les méthodes ont sur leur droite leur type (int, string, date, etc.) et sur leur gauche leur portée. Les différents types de portée sont :

- *Public* : il est représenté par un plus (+)
- *Private* : il est représenté par un moins (-)
- *Protected* : il est représenté par un dièse (#)
- *Internal* : il est représenté par un (i)
- *Protected internal* : il est représenté par un (r)

À chaque fois l'attribut ou les attributs soulignés dans la classe représentent la clé primaire.

III. SQL

1. LDD :

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

- 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);

Table est le nom que l'on donne à la table. Colonnei est le nom d'une colonne. Typei est le type des données contenues dans Colonnei

Exemple :

CREATE TABLE STAGE (Num-Stage NUMBER(3) NOT NULL PRIMARY KEY,
Libellé-Stage VARCHAR2(15) NOT NULL, Nb-jours NUMBER(2),
Type-Stage VARCHAR2(15),
Num-Cat NUMBER);

- La commande ALTER TABLE permet de modifier la définition d'une table.

ALTER TABLE table

ADD

MODIFY (colonne1 type1, colonne2 type2, ..., colonnen typen)

DROP

Exemple :

ALTER TABLE STAGIAIRE ADD NomEntr VARCHAR2(20);

ALTER TABLE STAGIAIRE MODIFY NomEntr VARCHAR2(30);

ALTER TABLE STAGIAIRE DROP NomEntr; Ajout d'un attribut Modification dans le type d'un attribut Suppression d'un attribut.

Attention, certaines modifications peuvent poser des problèmes d'intégrité dans la base !

- La commande DROP 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.

DROP TABLE table

Exemple :

DROP TABLE STAGIAIRE;

2. LMD :

L'instruction SELECT est la base du LMD, elle permet de renvoyer une table contenant les données correspondantes aux critères qu'elle contient.

Seules les clauses "SELECT" et "FROM" sont obligatoires. La forme générale d'une instruction SELECT (on parle également de *phrase SELECT* ou *requête*) est :

```
SELECT [ALL] | [DISTINCT] * | <liste de champs ou d'instructions d'agrégation>
FROM <liste de tables>
WHERE <condition>
GROUP BY <champs de regroupement>
HAVING <condition>
ORDER BY <champs de tri> [DESC] | [ASC]
```

Une autre forme est :

```
SELECT [ALL] | [DISTINCT] * | <liste de champs ou d'instructions  
d'agrégation>  
FROM <table de base>  
<liste de jointures>  
GROUP BY <champs de regroupement>  
HAVING <condition>  
ORDER BY <champs de tri> [DESC] | [ASC]
```