

CHEIKH BABACAR DIOUF

DOCUMENTATION SUR LE MCD, MLD ET SQL

INTRODUCTION

Un système d'information ou SI, est l'ensemble des moyens humains, matériels et immatériels mis en œuvre afin de gérer l'information au sein d'une unité, d'une organisation comme une entreprise.

Pour arriver à bien réaliser ces objectifs, des méthodes comme merise ou l'UML ont été mise en place. Ces méthodes fonctionnent suivant un certain nombre de protocoles que nous verrons dans la suite de notre exposée.

I) NOTION D'ANALYSE ET DE CONCEPTION

a) LA METHODE MERISE

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.

b) L'UML

C'est un langage de modélisation graphique de données à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture logicielle. Les différents éléments représentables sont :

- Activité d'un objet/logiciel
- Acteurs
- Processus
- Schéma de base de données
- Composants logiciels
- Réutilisation de composants

Grâce aux outils de modélisation UML, il est également possible de générer automatiquement tout ou partie du code d'une application logicielle, par exemple en langage Java, à partir des divers documents réalisés.

c)ETUDE COMPARATIVE

Comme le montre la représentation ci-dessous, nous pouvons distinguer les grandes différences entre la méthode Merise et l'UML.

Comparaison entre Merise & UML	
Merise	UML
Méthode d'analyse et de conception de système d'information	Langage de représentation d'un SI
Méthodes de modélisation de données et traitements orienté bases de données relationnelles	Système de notation orienté objet
relationnel	Objet
Franco-Français	Internationale
Schéma directeur, étude préalable, étude détaillée et la réalisation	Langage de modélisation des systèmes standards, qui utilise des diagrammes pour représenter chaque aspect d'un système : statique, dynamique... en s'appuyant sur la notion d'orienté objet

29

II)CONCEPTS ANALYSE ET CONCEPTION

Pour la conception de SI, on passe par différents stades, suivants ces stades nous allons nous intéresser au Modèle Conceptuel des Données (MCD) et au Modèle Logique de Données (MLD).

Les modèles conceptuels permettent de représenter la réalité telle qu'elle est perçue par les utilisateurs Ils constituent une base de dialogue et de validation entre concepteur et utilisateur Leurs diagrammes utilisent des notations simples et précises, compréhensible de manière visuelle

a) MCD

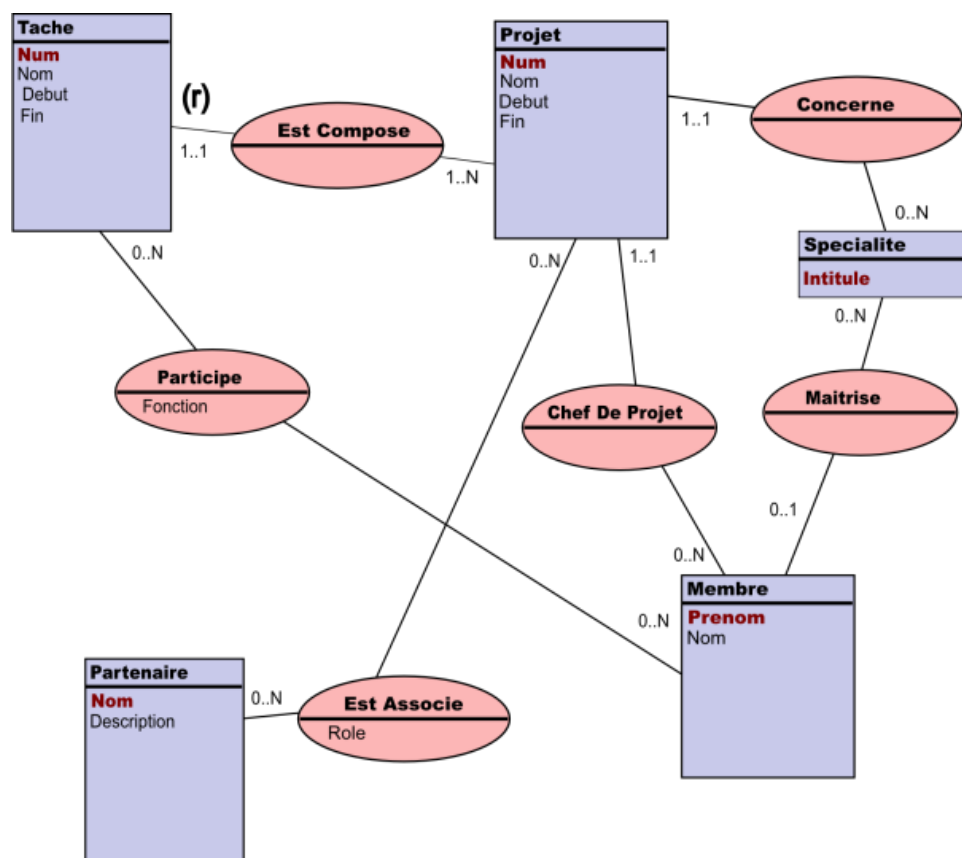
Le MCD est l'élément le plus connu de MERISE et certainement le plus utile. Il permet d'établir une représentation claire des données du SI et définit les dépendances fonctionnelles de ces données entre elles.

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.

Cet outil permet d'échanger entre informaticiens et non-informaticiens sur l'outil à informatiser. On peut ainsi à partir d'un MCD valider et préciser des règles qui s'appliqueront à la future base de données (d'après le MCD en image ci-dessous)

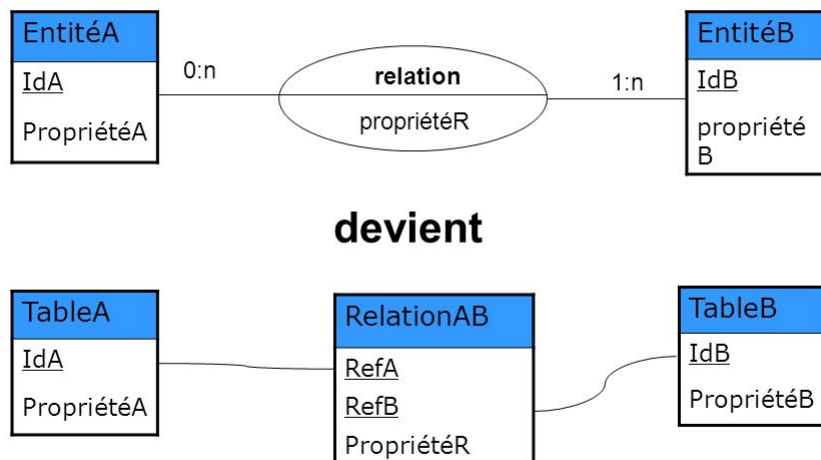


b) LE MLD

Une fois le MCD établi, nous sommes en mesure de le traduire en système logique : MLD.

Pour le passage entre les deux on applique les règles suivants :

Passage du MCD au MLD



IV)SQL

Chaque SGBD possède ses propres spécificités et caractéristiques. Pour présenter ces différences, les logiciels de gestion de bases de données sont cités, tels que : MySQL, PostgreSQL, SQLite, Microsoft SQL Server ou encore Oracle.

comme dans la vie, pour que des personnes puissent se comprendre, elles doivent parler le même langage et bien en informatique, c'est pareil.

Pour que les différents logiciels et le moteur de base de données puissent se comprendre, ils utilisent un langage appelé SQL.

Ce langage est complet. Il va être utilisé pour :

- Lire les données,
- Ecrire les données,
- Modifier les données,
- Supprimer les données
- Il permettra aussi de modifier la structure de la base de données :
 - Ajouter des tables,
 - Modifier les tables,
 - les supprimer
 - Ajouter, ou supprimer des utilisateurs,
 - Gérer les droits des utilisateurs,
 - Gérer les bases de données : en créer de nouvelles, les modifier, etc ...

Comme vous pouvez le voir, les possibilités sont nombreuses.

Ce langage est structuré (comme son nom l'indique), c'est à dire que la syntaxe est toujours la même et respecte des normes très précises.

a) LDD

Le LDD comprend les commandes pour la définition des données, qui sont CREATE (créer), DROP (supprimer), et ALTER (modifier).

Instruction CREATE TABLE

Cette instruction permet de créer une table : il faut définir son nom, les différents champs de la table avec leur type ainsi que des caractéristiques comme par exemple des clés.

Chaque champs d'une table possède un type :

CHAR chaîne non unicode remplacé par nchar pour des chaînes unicode

VARCHAR chaîne non unicode remplacé par nvarchar pour des chaînes unicode microsoft.

INTEGER

NUMBER

DECIMAL

FLOAT

DOUBLE

DATE

TIME

TIMESTAMP

Exemple :

```
CREATE TABLE client1 ("NOM" VARCHAR( 50 ), "PRENOM" VARCHAR( 50 ), "TELEPHONE"
VARCHAR( 50 ), "ADRESSE" VARCHAR( 200 ) );
```

Cette instruction crée une table intitulée client1 contenant 4 champs :

Un champ NOM : chaîne d'au plus 50 caractères.

Un champ PRENOM : chaîne d'au plus 50 caractères.

Un champ TELEPHONE : chaîne d'au plus 50 caractères.

Un champ ADRESSE : chaîne d'au plus 200 caractères.

Instruction DROP TABLE

```
DROP TABLE nom_table;
```

Signification : détruit la table appelée nom_table.

Exemple :

```
DROP TABLE `client1` ;
```

Cette commande permet de détruire la table client1.

INDEX

Instruction CREATE INDEX

```
CREATE [UNIQUE] INDEX nom_de_l_index ON nom_de_table
```

Signification : crée un index appelée nom_de_l_index sur la table nom_de_table.

Instruction DROP INDEX[modifier | modifier le wikicode]

```
DROP INDEX nom_de_l_index ON nom_de_table
```

Signification : détruit l'index appelé nom_de_l_index de la table nom_de_table.

b)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]
```

Détails des clauses :

SELECT

La clause SELECT permet de spécifier les informations qu'on veut récupérer. Elle contient des champs provenant de tables spécifiées dans la clause FROM, ainsi que des instructions d'agrégation portant sur ces champs.

Le nom des champs ne doit pas être équivoque, ce qui veut dire que si des champs de tables différentes ont le même nom, les champs doivent être préfixés par le nom de la table : nom_de_table.nom_de_champs

Les noms des champs sont séparés par des virgules. Si la requête comporte la clause DISTINCT, les doublons (lignes de la table de résultat ayant exactement les mêmes valeurs dans tous les champs) seront éliminés, alors qu'avec la clause ALL, tous les résultats sont renvoyés.

SELECT * permet de renvoyer tous les champs de toutes les tables spécifiées dans FROM.

Instructions d'agrégation[modifier | modifier le wikicode]

Les instructions d'agrégation permettent des opérations comme le comptage ou les sommes.

Les différentes instructions d'agrégation sont :

AVG(<champs>)

COUNT(*)

MAX (<champs>)

MIN (<champs>)

SUM (<champs>)

FROM

Clause obligatoire qui détermine sur quelles tables l'on fait la requête. Les noms des tables sont séparés par des virgules.

C)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.

Voici 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

1. 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
[GROUP      BY      ordre      des      groupes]
[HAVING
[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

```

2. Les clauses WHERE et ORDER BY

• La clause WHERE est un filtre sur les données, ce qui permet soit d'effectuer des jointures ou des restrictions. La syntaxe est la suivante :

```

WHERE

```

Le prédicat doit contenir n'importe quelle expression logique renvoyant une valeur vraie.

Exemple :

SELECT FROM WHERE 1=1	CLI_NOM T_CLIENT	CLI_NOM ----- DUPONT MARTIN BOUVIER DUBOIS DREYFUS FAURE LACOMBE DUHAMEL BOYER MARTIN ...
-----------------------------	---------------------	---

Attention : la plupart des SGBDR ne comportent pas de colonne de type booléen. Une requête comme la première risque d'échouer.

Exemple :

SELECT FROM TJ_CHB_PLN_CLI WHERE CHB_PLN_CLI_OCCUPE	* ERREUR ! bien que CHB_PLN_CLI_OCCUPE puisse être du booléen, la plupart des compilateurs SQL n'acceptent pas ce test direct.
SELECT FROM TJ_CHB_PLN_CLI WHERE CHB_PLN_CLI_OCCUPE = True	* CORRECT ... Mais sur certains compilateurs SQL il faut faire : CHB_PLN_CLI_OCCUPE = 'True' (littéral). Si le type booléen n'existe pas, alors il faut faire CHB_PLN_CLI_OCCUPE = 1 si l'on a choisi de définir les booléens comme INTEGER(1) avec 0 et 1

- La clause ORDER BY permet de définir le tri des colonnes de la réponse, soit en précisant le nom littéral de la colonne, soit en précisant son numéro d'ordre dans l'énumération qui suit le mot clef SELECT. Sa syntaxe est la suivante :

ORDER BY colonne1 | 1 [ASC ou DESC] [, colonne2 | 2 [ASC ou DESC] ...

ASC spécifie l'ordre ascendant et DESC l'ordre descendant du tri. ASC ou DESC peuvent être

omis, dans ce cas c'est l'ordre ascendant qui est utilisé par défaut.

Bien que la clause ORDER BY ne soit pas nécessaire, il est souvent utile de trier la réponse en fonction des colonnes. En revanche le temps de réponse s'en ressent souvent. Pour spécifier l'ordre de tri, on doit placer les noms des colonnes séparées par des virgules juste après le mot clef ORDER BY, dans l'ordre voulu. On peut aussi utiliser le rang de chaque colonne dans l'ordre spécifié dans la clause SELECT.

Attention : le tri est un tri interne, il ne faut donc placer dans cette clause que les noms des colonnes présentées dans la clause SELECT.

Souvent, le fait de placer DISTINCT suffit, en général, à établir un tri puisque le moteur doit se livrer à une comparaison des lignes mais ce mécanisme n'est pas garanti car ce tri s'effectue dans un ordre non contrôlable qui peut varier d'un serveur à l'autre.

Exemple :

SELECT	CLI_NOM,	CLI_PRENOM	CLI_NOM	CLI_PRENOM
FROM		T_CLIENT	-----	-----
ORDER BY	CLI_NOM,	CLI_PRENOM	AIACH	Alexandre
ou			ALBERT	Christian
			AUZENAT	Michel
			BACQUE	Michel
SELECT	CLI_NOM,	CLI_PRENOM	BAILLY	Jean-François
FROM		T_CLIENT	BAVEREL	Frédéric
ORDER BY 1, 2			BEAUNEE	Pierre
			BENATTAR	Bernard
			BENATTAR	Pierre
			BENZAQUI	Joël
			...	

Remarque : les marqueurs NULL sont situés en premier dans l'ordre ainsi établi.

Un problème, qui n'est pas résolu, est de pouvoir choisir l'ordre des colonnes de la réponse. Sur certains serveurs cela peut être obtenu en plaçant les noms des colonnes à obtenir dans l'ordre où l'on veut les voir apparaître dans la clause SELECT mais cette possibilité n'est jamais garantie.

Attention : la clause ORDER BY est la dernière clause de tout ordre SQL et ne doit figurer qu'une seule fois dans le SELECT, même s'il existe des requêtes imbriquées ou un jeu de requêtes ensemblistes.

Voici un exemple illustrant tous les types de clauses.

Exemple : liste des loyers toutes charges comprises des locataires à Lille par ordre alphabétique.

```

SELECT      NomClt,      NumAppart,      [loyerHC]
+          charges      [AS      LoyerTCC]
3          champs      à      projeter
dont      un      champ      calculé

FROM      CLIENT,      APPARTEMENT
Tables      utiles      à      la      requête

WHERE      CLIENT.NumAppart      =
APPARTEMENT.      NumAppart
Jointures      entre      les      2      tables

AND      VilleAppart      =      «      Lille      »
Restriction      aux      clients      de      Lille

ORDER      BY      NomClt      ASC
Tri par ordre alphabétique du nom

```

CONCLUSION

Avec les différentes notions abordées dans cet exposé, il nous est maintenant possible de concevoir et réaliser des bases de données relationnelles. Il est important de maîtriser les différentes notions de ce modèle qui est aujourd'hui le plus enseigné au niveau des formations, mais aussi le plus utilisé en entreprise.

Cependant, ce modèle a souvent montré ses limites au niveau de certains systèmes d'informations. Par exemple, vous pourriez être amenés à travailler sur des systèmes d'informations où les SGBDR ne servent qu'à persister des données au format XML. Le traitement de l'information, qui doit parfois se faire en temps réel, se fait donc au niveau de l'application afin de ne pas surcharger de requêtes les serveurs de données.