

LA MÉTHODE MERISE

... en quelques mots.

Contenu

Bases de données relationnelles	1
Bases de données SQL	1
Le langage SQL	2
Système de Gestion de Bases de Données	2
La méthode Merise, en quelques mots	3
Les étapes de la démarche Merise	5

Bases de données relationnelles

Une base de données relationnelle est un espace de stockage où l'information est organisée de manière cohérente dans des tableaux à deux dimensions appelés des **relations** ou **tables**.

Graphiquement, une table peut se représenter comme un tableau de données dans une feuille de calcul Excel.

L'image ci-dessous représente 4 tables : Personne, Pays, Région et Ville :

Personne			
#Numero	Nom	Prenom	LieuNaissance
1	Durand	Pierre	60200
2	Dupont	Marie	60200

Pays			
#Nom	Population	Superficie	Dirigeant
France	60	500001,01	2
Allemagne	80	600000,23564	2
Espagne	40	350000,1	1

Région			
#Pays	#Nom	Superficie	Dirigeant
France	Picardie	50	1
Espagne	Picardie	40	1
France	Normandie	30	2

Ville				
#CodePostal	Nom	Pays	Région	Dirigeant
F60200	Compiègne	France	Picardie	1
F60300	Senlis	France	Picardie	2
F60301	Senlis	France	Picardie	2
E8000	Senlis	Espagne	Picardie	2

Pour interagir avec une base de données relationnelle, on utilise un langage relationnel tel que le SQL (Structured Query Language).

Bases de données SQL

Les bases de données SQL existent depuis plusieurs décennies et sont largement utilisées depuis les années 1990.

Les données stockées dans une base SQL sont organisées en **tables** reliées entre elles.

La structure et le type des données sont rigides (fixés à l'avance).

Dans les projets liés à une base SQL, la base de données doit être modélisée avant d'implémenter une logique métier. Il est donc impossible d'ajouter des données tant que les tables ne sont pas définies (avec leurs colonnes et leur type de données).

La définition des tables est ce qu'on appelle un « Schéma ». Ce schéma contient donc les informations sur la structure des tables, les clés primaires, les index, les contraintes, les déclencheurs et les procédures stockées.

Le schéma de la base de données doit être conçu et mis en œuvre avant que toute logique métier puisse être développée pour manipuler les données.

Il est évidemment possible de modifier un schéma SQL existant mais de gros changements rendent la mise à jour des objets métiers compliquée.

Le langage SQL

le langage SQL est un langage normalisé permettant de définir, organiser et manipuler les données.

Le langage SQL est divisé en plusieurs sous-ensembles :

- **DDL : Data Definition Langage** (Langage de Définition des Données)
 - o Permet de manipuler les structures de données.
- **DML : Data Manipulation Langage** (Langage de Manipulation des Données)
 - o Permet de manipuler les données.
- **DCL : Data Control Langage** (Langage de Contrôle des Données)
 - o Permet de contrôler les accès aux données.
- **TCL : Transaction Control Langage** (Langage de Contrôle des Transactions)
 - o Permet de contrôler, valider et annuler les transactions (lots d'instructions).

Système de Gestion de Bases de Données

Acronymes :

- SGBDR (Système de Gestion de Bases de Données Relationnelles)
- SGBD (Système de Gestion de Bases de Données)
- DBMS en anglais (Data Base Management System)

Un Système de Gestion de Bases de Données Relationnelles permet de stocker, administrer et gérer des bases de données relationnelles.

Tous les SGBDR SQL proposent les fonctionnalités liées au langage SQL et à ses sous-ensembles.

Le langage SQL, bien que standardisé depuis de nombreuses années, peut légèrement différer d'un SGBD à un autre.

Il est primordial de bien sélectionner le SGBD qui sera adapté au système cible.

Ce choix se fera en fonction de plusieurs critères tels que le volume de données à traiter par transaction, le niveau de sécurité attendu, les performances ou le budget (le prix d'un SGBD varie de gratuit à plusieurs centaines de milliers d'euros par licence).

Les SGBDR SQL les plus connus sont :

- Oracle Database : <https://www.oracle.com/>
- Microsoft SQL Server : <https://www.microsoft.com/fr-fr/sql-server/>
- MySQL : <https://www.mysql.com/fr/>
- MariaDB : <https://mariadb.org/> (Fork de MySQL 100% compatible)
- PostgreSQL : <https://www.postgresql.org/>

D'autres systèmes de gestion de bases de données :

https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es#Quelques_SGBD

La méthode Merise, en quelques mots

La méthode Merise est une méthode d'analyse, de conception et de modélisation de systèmes d'information. Cette méthode de gestion de projet est particulièrement adaptée à la conception de bases de données relationnelles et aux systèmes d'information de grande taille.

Les projets Merise sont en général des projets de grande ampleur et concernent principalement :

- l'informatisation de processus métiers.
- la refonte d'un S.I existant complexe dans un environnement « grand système ».
- la fusion de plusieurs systèmes en un ensemble plus grand et plus complexe.

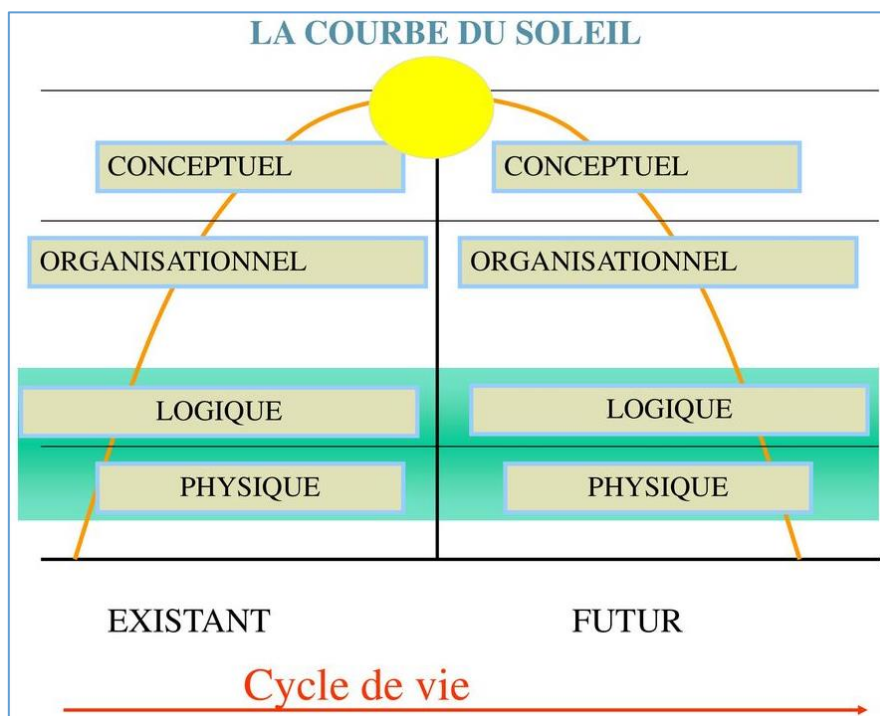
La méthode Merise propose une démarche articulée selon 3 axes :

1. Cycle de décision

- étude préalable
- définition du projet (étude détaillée)
- détails de la réalisation et de la mise en œuvre du système d'information
- chaque étape est documentée et marquée par une prise de décision

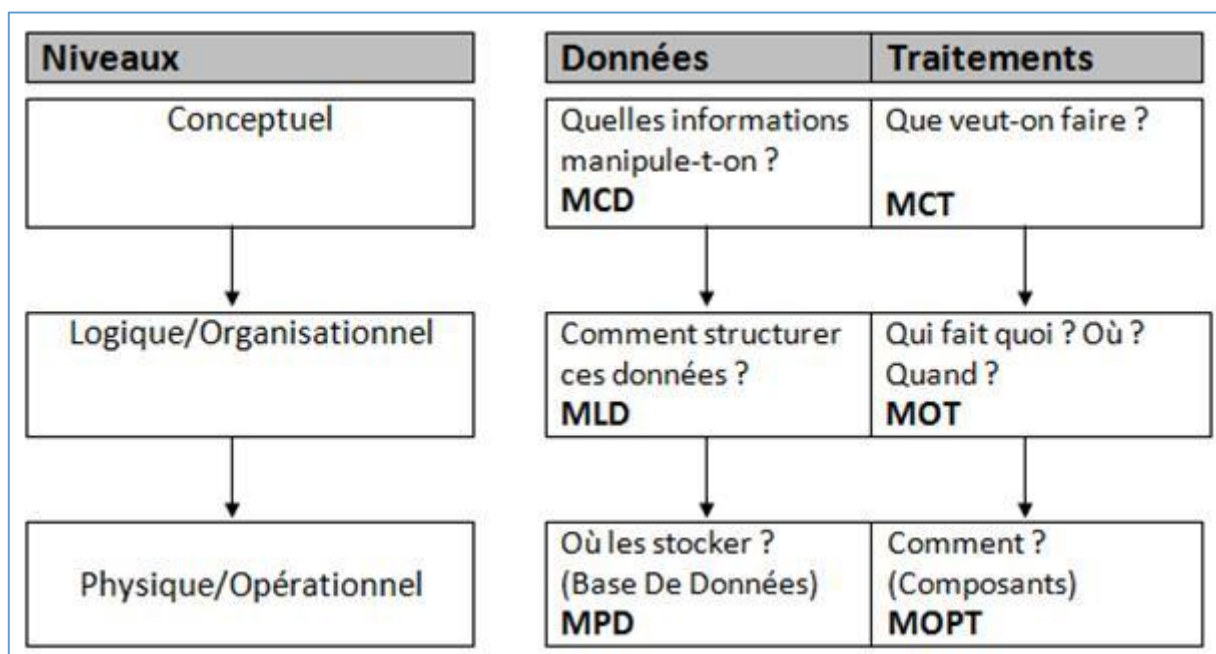
2. Cycle de vie

- analyse de l'existant
- conception
- réalisation
- maintenance



3. Cycle d'abstraction

- du conceptuel à l'opérationnel (du plus abstrait au plus concret)



La méthode Merise distingue les **données** et les **traitements**, même si les interactions entre les deux sont profondes et s'enrichissent mutuellement (validation des données par les traitements et réciproquement).

Nous nous intéressons ici à la modélisation des **données**.

Les étapes de la démarche Merise

"Pour concevoir un système d'information robuste, il faut, en premier lieu, identifier les données à traiter et les relations entre ces données."

<p>A N A L Y S E</p>	<p>1. Recueillir les informations</p> <ul style="list-style-type: none"> a. L'interview b. Les documents c. Les contraintes sur les données <p>2. Constituer le dictionnaire des données</p> <ul style="list-style-type: none"> a. Repérer les données à représenter b. Supprimer toute donnée "calculée" c. Préciser les contraintes liées à chaque donnée
<p>C O N C E P T I O N</p>	<p>3. Établir le modèle conceptuel</p> <ul style="list-style-type: none"> a. Repérer et créer les entités b. Attribuer à chaque entité un identifiant et compléter le dictionnaire des données c. Placer les propriétés dans les entités d. Repérer et placer les associations <p>4. Valider le modèle</p> <ul style="list-style-type: none"> a. Identifier les dépendances fonctionnelles b. Respect de la 1^{ère} Forme Normale (1FN) c. Respect de la 2^{ème} Forme Normale (2FN) d. Respect de la 3^{ème} Forme Normale (3FN) <p>5. Transformer le modèle conceptuel en modèle logique</p> <ul style="list-style-type: none"> a. Respect des règles de transformation b. Les entités et associations deviennent des relations c. Les identifiants deviennent des clés primaires d. Les associations impliquent la création de clés étrangères dans les relations
<p>R É A L I S A T I O N</p>	<p>6. Implémenter le modèle physique</p> <ul style="list-style-type: none"> a. Le modèle logique devient un schéma relationnel b. Les relations deviennent des tables c. Les attributs deviennent des champs (ou colonnes) d. Les clés primaires impliquent la création de contraintes d'unicité e. Les clés étrangères impliquent la création de contraintes d'intégrité relationnelle f. Choix du Système de Gestion de Base de Données g. Le schéma est implémenté dans un langage relationnel (création des tables et des contraintes) h. Le schéma est testé et validé par un ou plusieurs jeux de données (tests d'intégrité) i. Les processus métiers et de sécurité sont implémentés (procédures stockées, déclencheurs...) j. L'ensemble est testé en conditions réelles (tests fonctionnels) k. Livraison

--- FIN DU DOCUMENT ---

<http://www.arfp.asso.fr>