

Utilisation des bases de données

CM1-2 : Merise

Mickaël Martin-Nevot

V1.3.1



Cette œuvre est mise à disposition selon les termes de la [licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage à l'Identique 3.0 non transposé](https://creativecommons.org/licenses/by-nc-sa/3.0/).

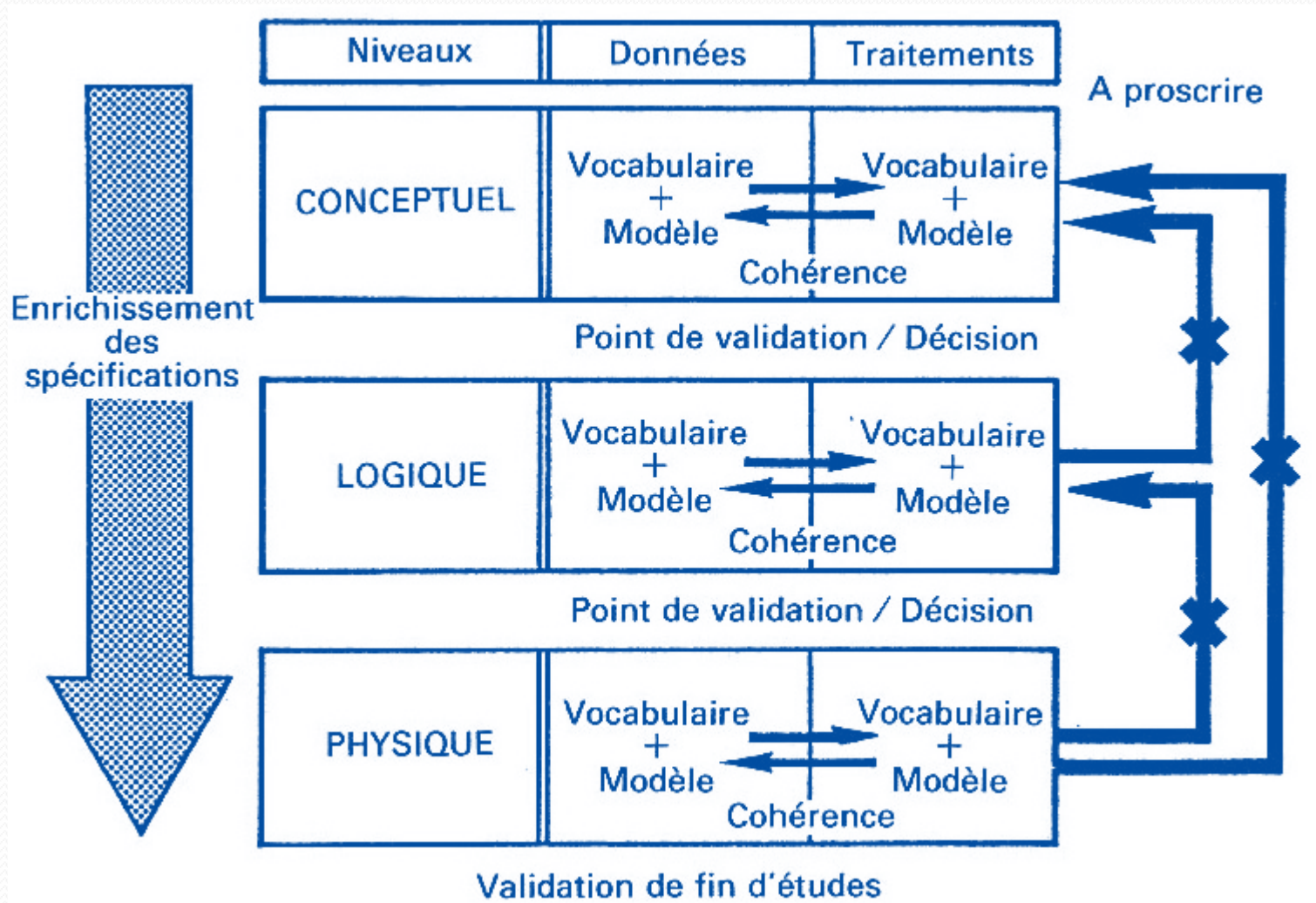
Merise

- Méthode d'analyse, de conception et de développement d'un système d'information informatisé
- Méthode spécifiquement française
- Essentiellement pour les SI avec BD relationnelle
- Permet **analyse systémique** et **modélisation**

SI : système d'information

Analyse systémique : méthode de réflexion qui permet d'aborder des aspects complexes et de les simplifier en les modélisant

Abstraction



Merise : Niveau conceptuel

Données

- Modèle conceptuel des données (**MCD**)
 - Décrit les **données** et leurs attributs
 - La **définition sémantique des données** permet l'appréhension complète des informations
 - Qu'est ce qu'on va gérer comme données, comment sont elles organisées

Traitement

- Modèle conceptuel des traitements (**MCT**)
 - Décrit les **opérations** à réaliser avec les données
 - La reconnaissance des **traitements** fondamentaux met en évidence les objectifs du système
 - Qu'est ce qu'on va réaliser comme traitement

Quoi ?

Merise : Niveau logique

Données

- Modèle logique des données (**MLD**)
 - Description du système, **indépendamment du SGBD**
 - Passage « automatique » au modèle relationnel

Traitement

- Modèle logique des traitements (**MLT**)
 - Description indépendante de la machine
 - **Structuration** en procédure

Validation et optimisation du MLD par rapport aux traitements

Qui ? Quand ? Où ?

Merise : Niveau physique

Données

- Modèle physique des données (**MPD**)
 - **Description interne des données** en fonction du logiciel SGBD
 - Définition des contraintes, structures d'accès, etc.

Traitement

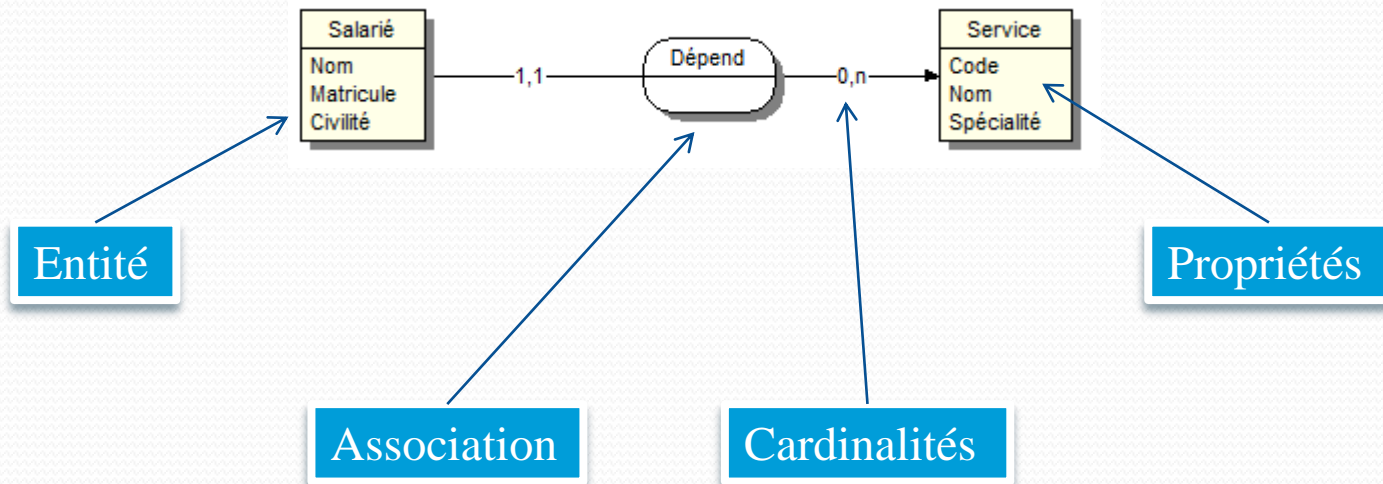
- Modèle physique des traitements (**MPT**)
 - Description de l'**architecture** des traitements
 - Spécifications détaillées de la **programmation** (algorithmes)

Comment ? Avec quoi ?

Démarche

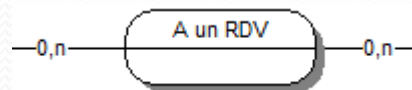
1. Définir l'application (MCT) :
 - Définir l'objectif précis
 - Définir les sorties
2. Définir les données (MCD) :
 - Recensement des données nécessaires
 - Définir l'organisation des données
3. Définir les requêtes nécessaires pour l'application (MLT)
4. Validation :
 - Modélisation compatibles avec les requêtes ou recommencement
5. Définir le MLD, vérifier la validation puis définir le MPD

MCD



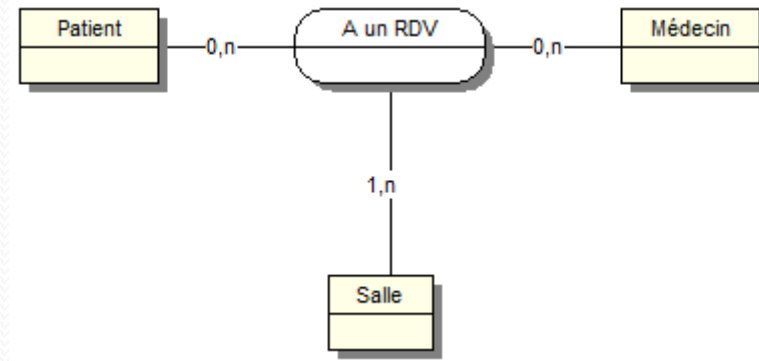
Vocabulaire

- **Entité (TE)** : « objet » pourvue d'une existence propre
- **Association (TA)** : relation entre entités, dépourvue d'existence propre
- **Propriété** : plus petit élément d'information caractérisant partiellement une entité ou une association
- **Occurrence** : valeur d'une propriété, d'une entité ou d'une association
- **Cardinalité** : nombre minimum et maximum d'occurrences d'une association pour une occurrence d'entité (indiqué sur chaque arc)



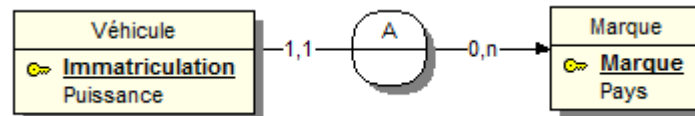
Classes d'association

- **Réursive** (ou *réflexive*) : relie la même entité
- **Binaire** : relie deux entités
- **Ternaire** : relie trois entités
- **N-aire** : relie n entités



Identifiant et agrégation

- **Identifiant** (absolu) :
 - **Entité** : une (ou plusieurs) propriété(s) qui définissent chaque occurrence de l'entité
 - **Association** : concaténation des identifiants des entités associées
- **Agrégation** (identifiant relatif) :
 - Spécifie qu'une entité est nécessaire pour en spécifier une autre (cardinalité maximale de 1) : association hiérarchique

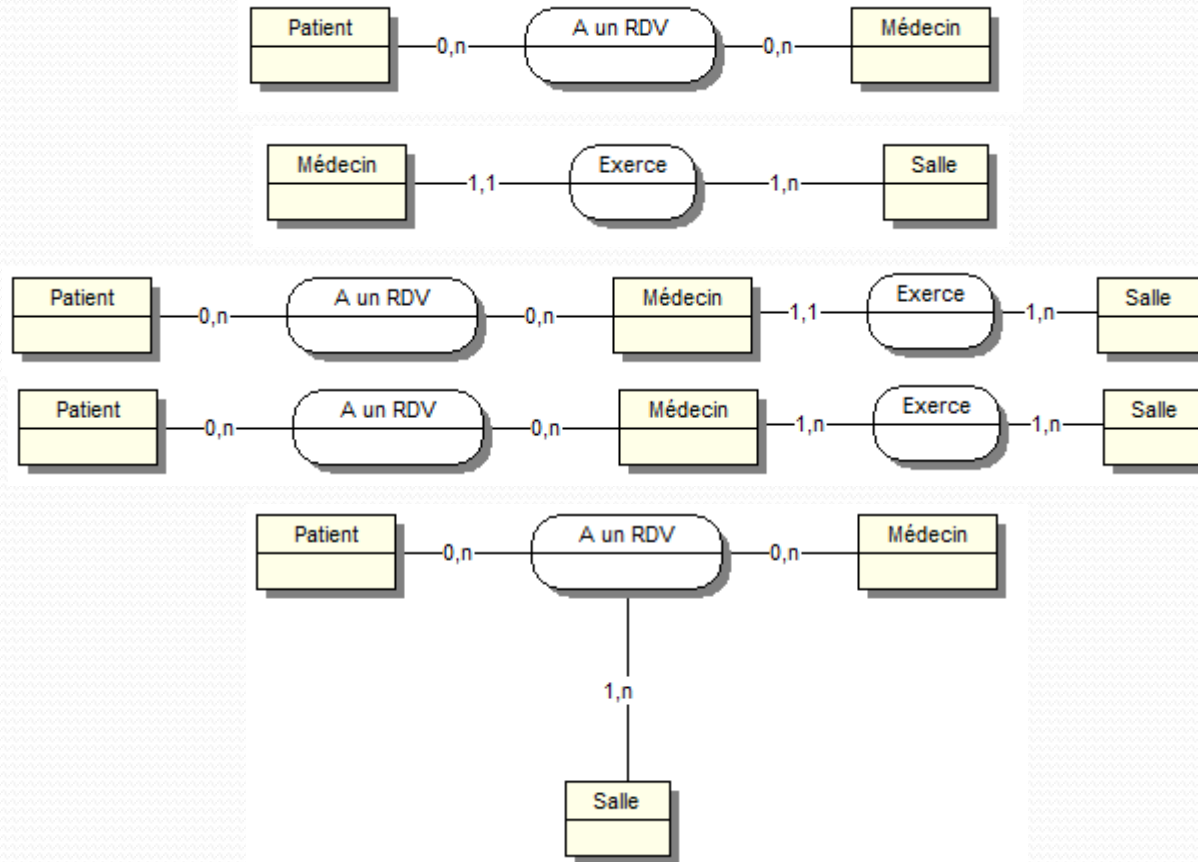


Tout entité doit avoir un identifiant

Entité ou association ?

- Souvent le **contexte** permet de décider
- Lorsqu'on ne parvient pas à trouver d'identifiant pour une entité, il faut se demander s'il ne s'agit pas plutôt d'une association
- Lorsque toutes les arcs d'un TA portent la cardinalité $(1, 1)$, il faut se demander si ce TA et les TE liés ne décrivent pas en fait un seul TE

Exemples de MCD



Règle d'or : énoncer le réel (avec des phrases)

Cas pratique



Formes normales

- **Les formes normales sont des règles permettant d'éviter les redondances en base de données**
- Défauts des redondances :
 - Perte de place
 - Risques d'incohérences
 - Difficultés de mise à jour
- **Cinq formes normales :**
 - De plus en plus complexes
 - De moins en moins essentielles

Seules les **trois premières formes normales** sont traitées dans le cadre de ce cours

Première forme normale

- 1NF
- Propriété **atomique** :
 - **Scalaire** (élémentaire)
 - **Non répétitive** (mono-valué : pas de liste de valeurs)
- Valeur **constante** (date de naissance plutôt que âge)

La notion d'élémentaire est relative : une adresse postale complète (a priori composée) peut être considérée comme élémentaire si elle est toujours manipulée comme telle (pas de tri par ville par exemple)

Il n'est pas dérangeant de séparer les propriétés groupées, mais pas l'inverse !

Diminue la complexité de traitement, permet une recherche plus rapide et facilite les mises à jours régulières

Deuxième et troisième FN

- Deuxième forme normale :
 - 2NF
 - Modèle en 1NF
 - Un attribut non identifiant ne dépend pas que d'une partie de l'identifiant
- Troisième forme normale :
 - 3NF
 - Modèle en 2NF
 - Pas de dépendance transitive : une propriété non identifiant ne dépend pas d'une ou plusieurs propriétés ne participant pas à l'identifiant

Du MCD normalisé au schéma relationnel

- **Règle 1:**

- Chaque entité (qui a des propriétés) devient une relation, son identifiant la clef primaire et ses attributs les propriétés de l'entité

- **Règle 2 :**

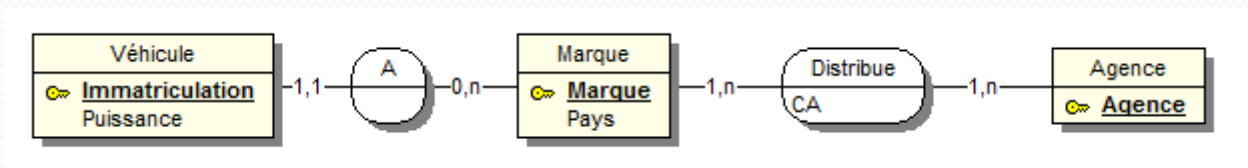
- Chaque entité d'une agrégation portant un arc d'une cardinalité maximale de 1 ajoute à la relation résultant l'identifiant des autres entités (en temps qu'attribut non clef)

- **Règle 3:**

- Les associations de type $n:m$ (ou ayant des propriétés) donnent lieu à la création de nouvelles relation :
 - Les identifiants des entités associées forment la clef primaire
 - Les propriétés de l'association deviennent des attributs simples

Exemple de schéma relationnel

- MCD :



- Schéma relationnel :

- Véhicule (Immatriculation, Puissance, *Marque*#)
- Marque (Marque, Pays)
- Agence (Agence)
- Distribue (*Marque*#, *Agence*#, CA)

Entités **statiques** (autonomes) : Véhicule, Marque, Agence

Entités **dépendantes** (associatives) : Distribue

Cas pratique



Maintenance

- Pour limiter le temps et les coûts de maintenance, il faut porter ses efforts sur les étapes antérieures

	Répartition effort dév.	Origine des erreurs	Coût de la maintenance
Définition des besoins	6%	56%	82%
Conception	5%	27%	13%
Codage	7%	7%	1%
Intégration Tests	15%	10%	4%
Maintenance	67%		

Outils

- AnalyseSI (génère script SQL, logiciel libre)
- MeriseSuite
- Power Designer / Power AMC
- Win'design
- Jmerise



PowerAMC

WIN DESIGN

Aller plus loin

- Modèle conceptuel de communication (MCC)
- Modèle conceptuel de traitement (MCT)
- Modèle logique de traitements (MLT)
- Modèle physique de données (MPD)
- Modèle physique de traitements (MPT)
- Contrainte d'intégrité fonctionnelle (CIF)
- Évènements internes
- Dépendances fonctionnelles
- Forme normale de Boyce-Codd (BCNF), 4NF, 5NF
- Processus de développement
- Modèles successifs de produits

Liens

- Document classique :
 - Luc Bouganim. *Cours de bases de données.*
 - Laurent Carmignac. *Introduction aux Bases De Données.*
 - Pierre Gérard. *MERISE, Modélisation de système d'information.*
 - Michel Divine. *Parlez-vous Merise ?*
 - Marie Paule Dumond. *La méthode Merise, aide à la conception d'un système d'information.*

Crédits

Auteur

Mickaël Martin-Nevot

mmartin.nevot@gmail.com

- Laurent Carmignac



Carte de visite électronique

Relecteurs

Cours en ligne sur : www.mickaël-martin-nevot.com

