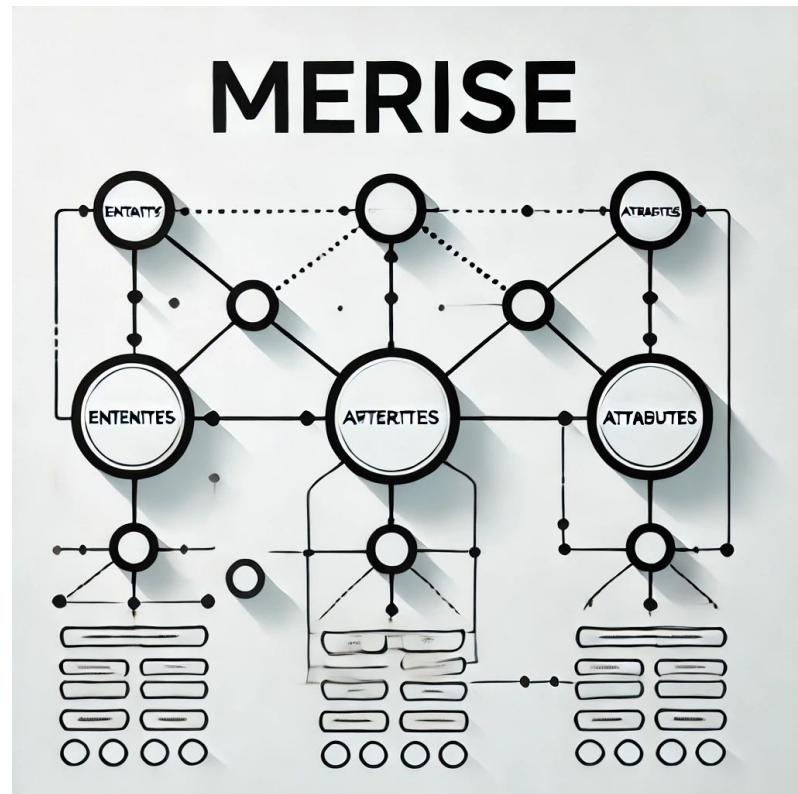




MERISE

par Romuald KOULEKO

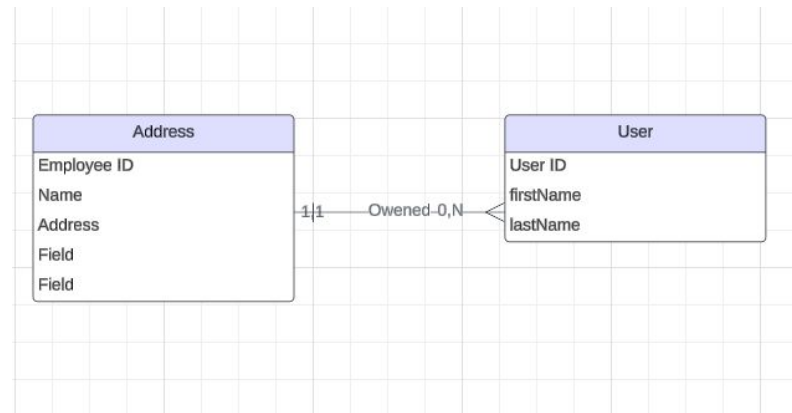
Développeur web & Formateur





Module 3

Modèle Conceptuel des Données (MCD)



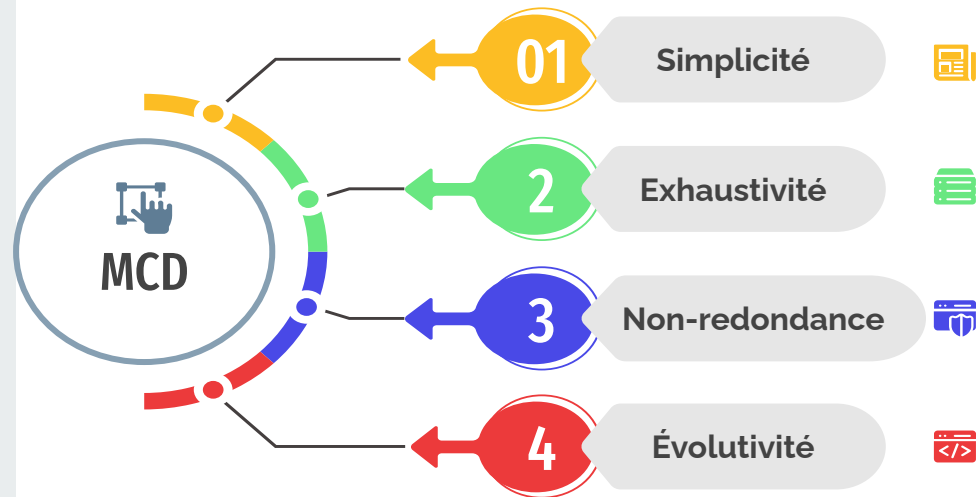
Concepts fondamentaux

Le **MCD** est une représentation graphique des données qui permet de :

1. Modéliser la structure des données du système d'information
2. Identifier les relations entre les différentes données
3. Établir les règles de gestion des données

Pourquoi utiliser le MCD ?

- Pour garantir la cohérence des données
- Pour faciliter la communication entre les différents acteurs
- Pour simplifier la maintenance future du système
- Pour optimiser la structure de la base de données

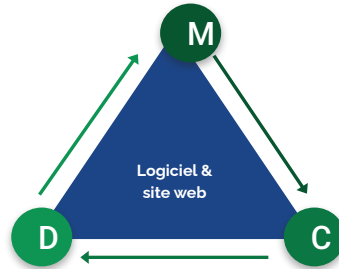


Logiciel et outils en ligne de modélisation

Logiciel



Looping



Outils en ligne

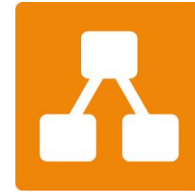




Tableau comparatif

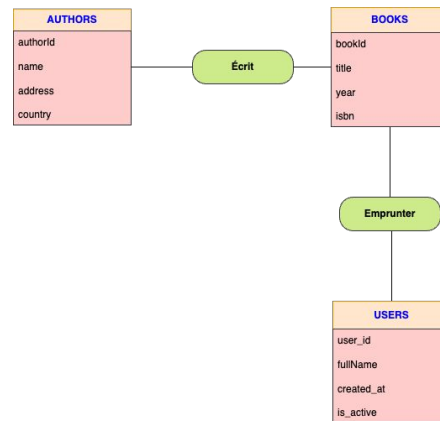
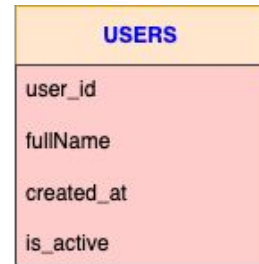
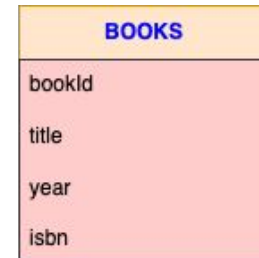
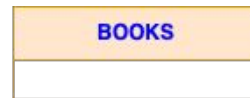
Outil	Type	Prix	Facilité d'utilisation	Langue	Fonctionnalités principales	Avantages	Inconvénients
PowerAMC (SAP PowerDesigner)	Logiciel	Payant	Moyenne	Français/Anglais	Modélisation MCD, MPD, UML, génération SQL	Très complet, reconnu en entreprise	Coût élevé, interface moins intuitive
Win'Design	Logiciel	Payant	Moyenne	Français	Modélisation MERISE complète, ergonomie avancée	Spécialisé MERISE, puissant pour projets MCD	Payant, moins connu internationalement
DB-MAIN	Logiciel	Gratuit	Moyenne	Français/Anglais	Modélisation MCD, MPD, outils d'analyse de données	Open source, adapté aux étudiants	Interface vieillissante
Looping	Logiciel	Gratuit	Facile	Français	Modélisation MCD simple pour MERISE	Léger, adapté aux débutants	Fonctionnalités limitées
draw.io (Diagrams.net)	Site web	Gratuit	Facile	Multilingue	Outil de dessin généraliste, adapté pour les MCD	Gratuit, intuitif, compatible avec les équipes	Pas spécialisé dans les bases de données
Lucidchart	Site web	Gratuit/Payant	Facile	Anglais	Collaboration en temps réel, personnalisation des diagrammes	Simple, adapté pour travail en équipe	Limitations dans la version gratuite
SQLDBM	Site web	Gratuit/Payant	Moyenne	Anglais	Modélisation de bases de données relationnelles, export SQL	Interface moderne, pensé pour les bases SQL	Fonctions limitées dans la version gratuite

Entités, attributs, associations

Entité : représente un ensemble d'objets ayant des propriétés communes et une existence propre dans le système que l'on modélise.

Attributs : propriétés qui décrivent une entité

Associations : représente un lien sémantique entre deux ou plusieurs entités.

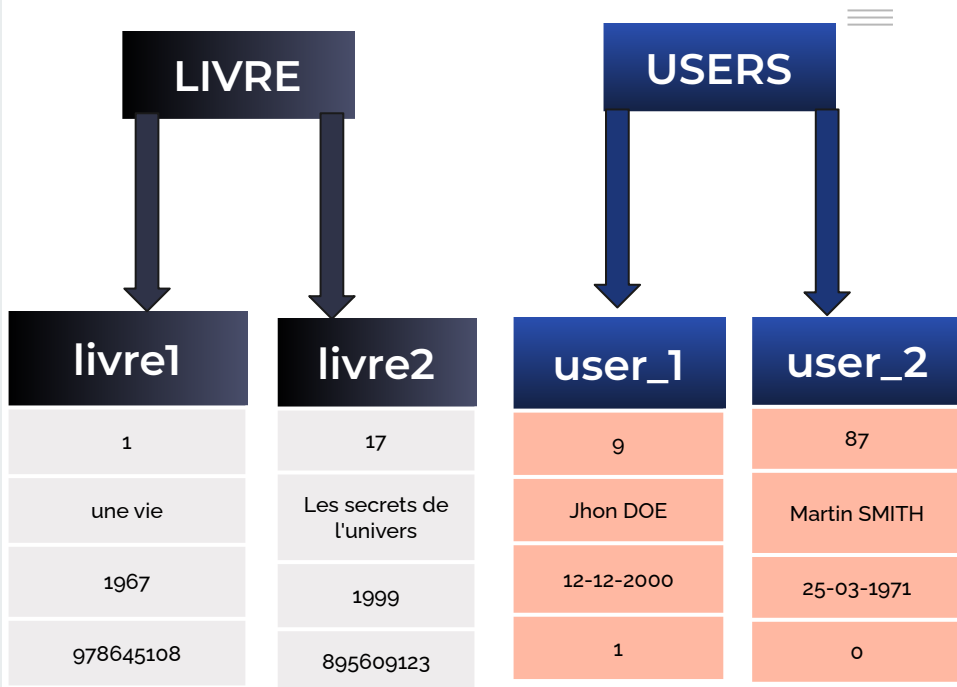


Occurrences

Occurrence : instance concrète de cette entité

Chaque occurrence est définie par :

- une valeur unique (identifiant)



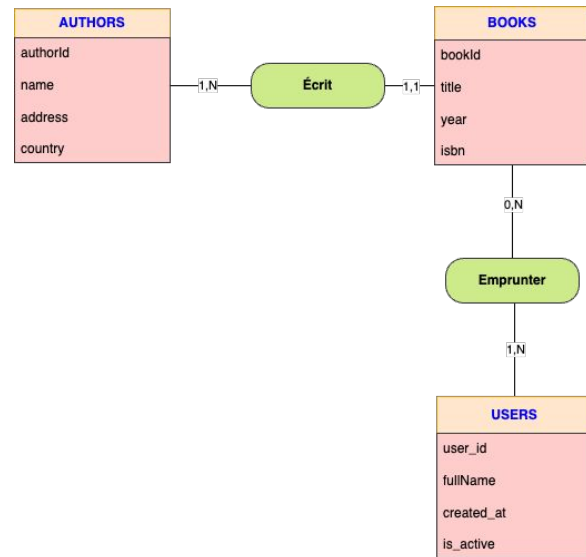
Les cardinalités et les types de relations: Relation binaire

cardinalités : nombre minimum et maximum de fois qu'une entité peut participer à une association.

Types de relations :

- **(0,1)** : 0 ou 1 occurrence
- **(1,1)** : exactement 1 occurrence
- **(0,n)** : 0 à plusieurs occurrences
- **(1,n)** : 1 à plusieurs occurrences

NB : Si une entité de cardinalité 0,1 dépend de l'entité à cardinalité 1,N ou si la relation est un cas particulier d'héritage, il pourrait être plus pertinent de fusionner les entités, mais cela reste une exception et non une règle générale.

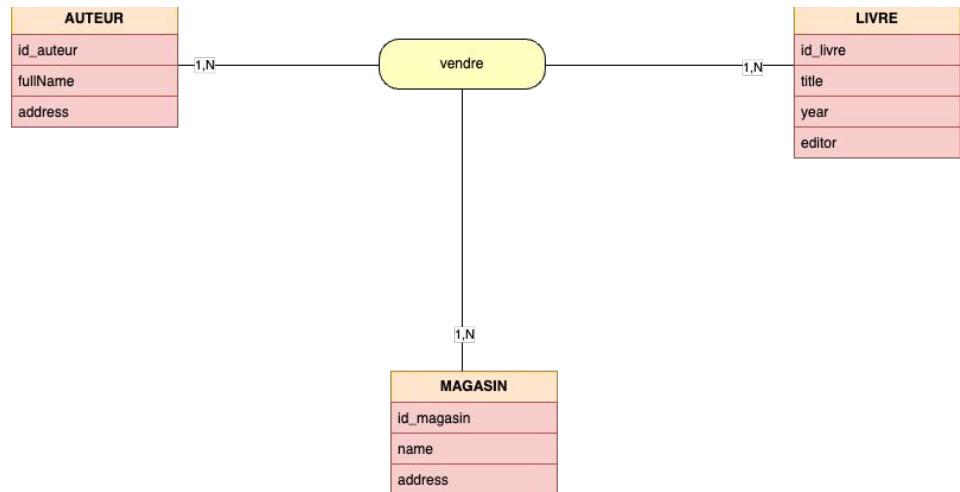


Les relations ternaires

Une relation ternaire implique trois entités.

Chaque association de **"vendre"** fait appel à un magasin, un livre et un auteur.

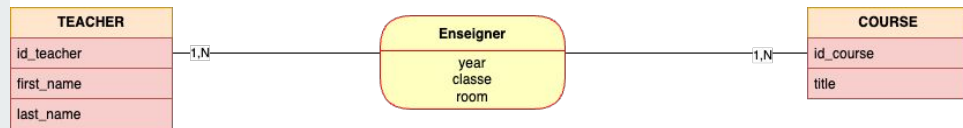
NB: Lorsqu'une relation ternaire présente une cardinalité 1,1 entre certaines entités, elle est considérée comme non optimisée. Pour améliorer l'efficacité du modèle, il est recommandé de décomposer cette relation ternaire en plusieurs relations binaires. Cela simplifie la structure tout en préservant l'intégrité des données.



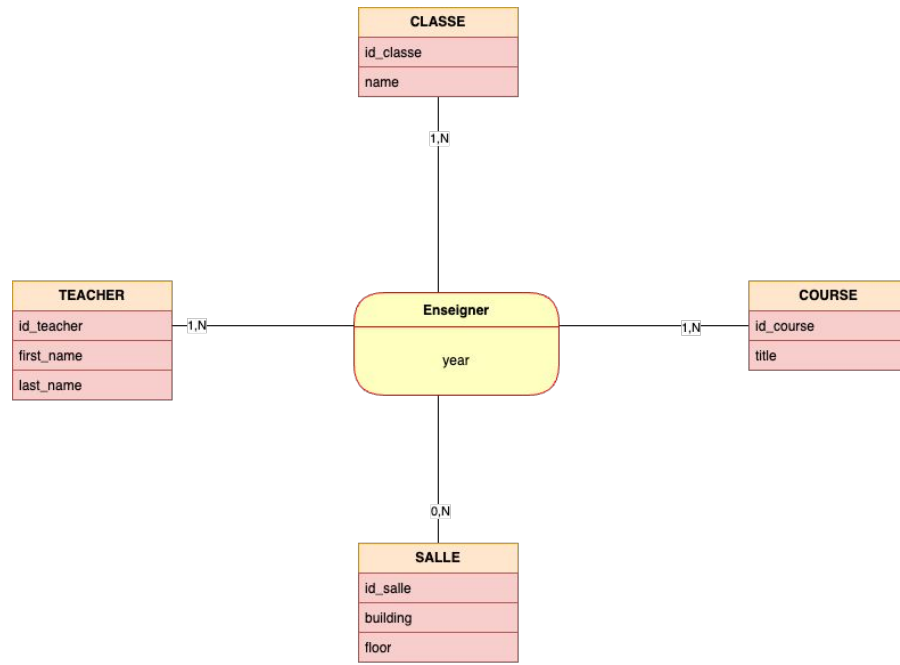
Les relations ternaires ou binaire ?

Le MCD est correct, mais il ne permet pas de détailler davantage les informations concernant year, classe, et salle.

Pour résoudre ce problème, il est nécessaire de créer deux nouvelles entités : Classe et Salle, afin de modéliser ces informations de manière plus précise et structurée.



OPTIMISATION





Règles de modélisation

Règles de modélisation :

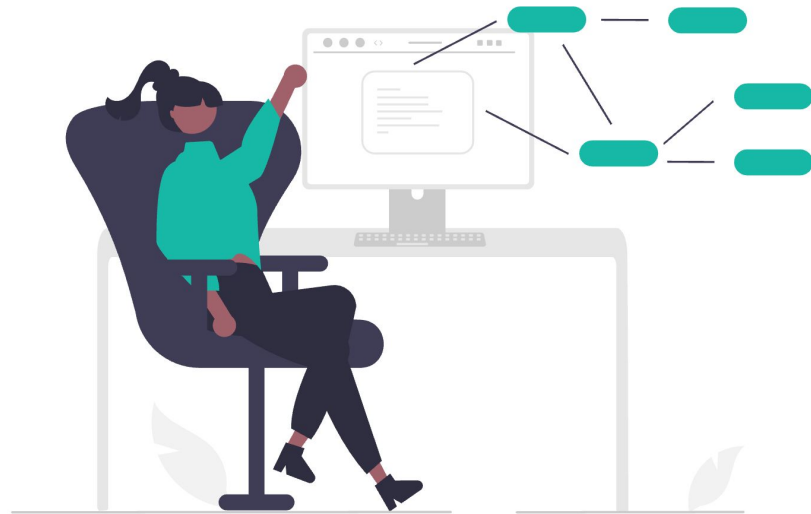
Définir clairement les cardinalités des relations : Chaque relation doit être accompagnée de cardinalités (1,1 ; 1,N ; 0,1 ; N,N) qui spécifient combien d'instances d'une entité peuvent être associées à une instance de l'autre entité.

Éviter les entités vides ou inutiles : Ne modélisez pas d'entités qui ne contiennent aucune information ou qui ne sont pas nécessaires pour le modèle de données.

Choisir des noms d'entités et d'attributs explicites : Les noms des entités et des attributs doivent être clairs et significatifs pour que l'on puisse facilement comprendre ce qu'ils représentent (par exemple, Client, Date_Commande, etc.).

Respecter l'intégrité des données : Assurez-vous que les entités respectent les contraintes d'intégrité, telles que l'intégrité référentielle, ce qui garantit que les données restent cohérentes et valides.

Utiliser des entités faibles si nécessaire : Lorsqu'une entité dépend d'une autre pour son identification, utilisez une entité faible, identifiée par une clé composée de l'entité forte et d'un attribut supplémentaire.



Méthodologie de construction du MCD

La création d'un MCD se fait en plusieurs étapes :

1. **Analyse du cahier des charges :** Lire attentivement le cahier des charges afin de relever les besoins en données essentiels.
2. **Identification des entités :** Lister les entités du domaine d'application.
3. **Définition des associations :** Déterminer les relations entre les entités.
4. **Définition des attributs :** Définir les propriétés des entités.
5. **Définition des cardinalités :** Déterminer le nombre d'occurrences des entités dans les relations.
6. **Définition des contraintes d'intégrité :** Définir les règles de modélisation et les contraintes d'intégrité.

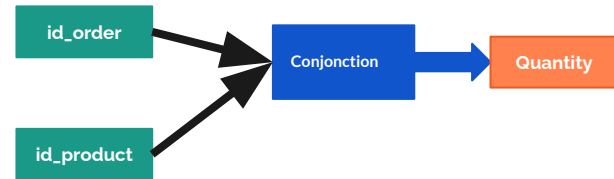
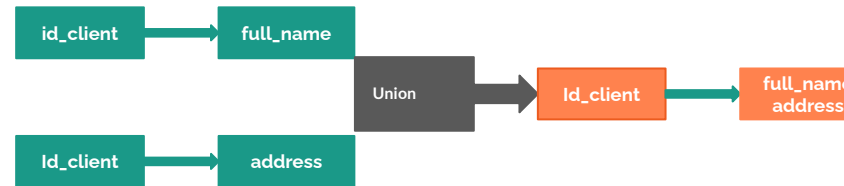
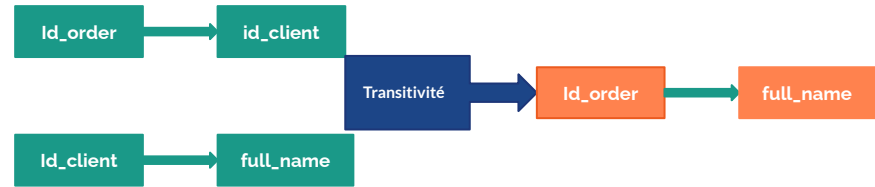


Les dépendances fonctionnelles

Les dépendances fonctionnelles aident à structurer les données de manière logique, permettant de déterminer quel attribut peut être utilisé pour en déduire un autre.

Règles:

1. Transitivité
2. Union
3. Conjonction



CIF : Contraintes d'Intégrité Fonctionnelle

CIF : règles qui garantissent que les données dans une base de données restent cohérentes et valides.

Elles sont généralement représentées par des flèches dans le MCD.

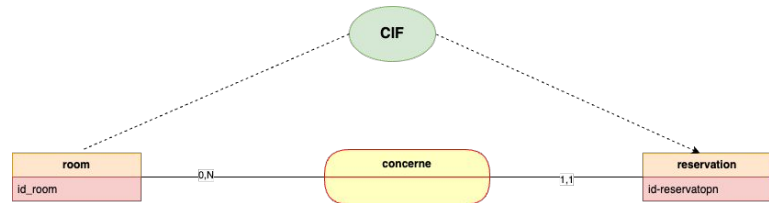
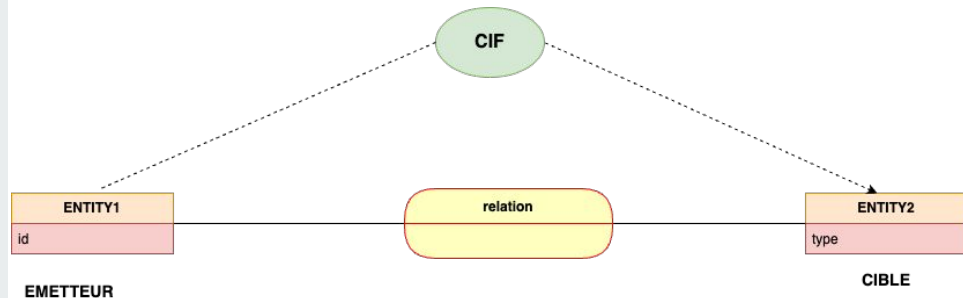
La CIF est représentée par une flèche en pointillés .
La pointe de la flèche indique l'entité dépendante.

Les CIF peuvent être de différents types, notamment :

Unicité : Assure que chaque valeur d'une colonne est unique.

Non-nullité : Assure qu'une colonne ne peut pas contenir de valeurs nulles.

Clé étrangère : Assure que les valeurs d'une colonne correspondent à des valeurs d'une autre colonne dans une autre table.



CIM : Contraintes d'Intégrité Multiples

CIM : contraintes qui impliquent plusieurs associations ou entités.

Les CIM peuvent inclure :

1. **Clé composite** : Une clé primaire composée de plusieurs attributs.
2. **Contraintes de domaine** : Assure que les valeurs d'un attribut appartiennent à un ensemble prédéfini de valeurs.
3. **Contraintes de transition** : Assure que les valeurs d'un attribut dépendent des valeurs d'autres attributs.

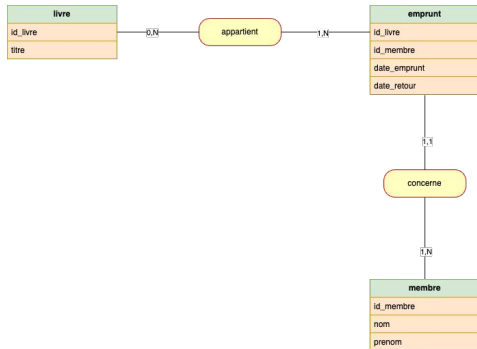
CIM dans cet exemple :

Clé composite : La clé primaire de Emprunt est composée de ID_Livre et ID_Membre.

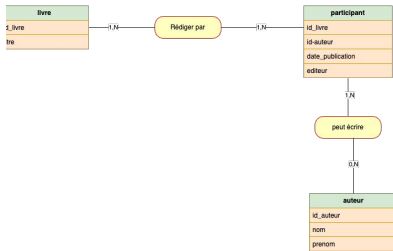
Contraintes de domaine : La Date_Emprunt doit être antérieure à la Date_Retour.

Contraintes de transition : La Date_Retour doit être postérieure à la Date_Emprunt.

NB : Les cardinalités 0,N - 1,N ou 1,N - 1,N dans un MCD peuvent indiquer la nécessité de créer une table d'association, surtout lorsque la relation est de type multiple-à-multiple ou nécessite des attributs supplémentaires pour décrire la relation. Cependant, la décision de créer une table d'association dépend des besoins spécifiques du modèle et de la nature de la relation entre les entités.



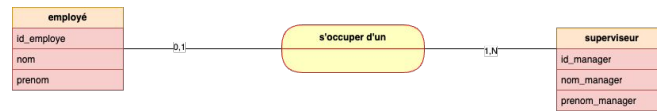
AMELIORATION



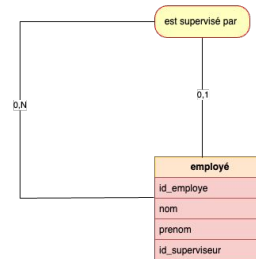
CIF : Réflexivité

Une association réflexive relie une entité à elle-même.

Cas1 : Un superviseur est une forme spécifique d'employé, ce qui peut être qualifié d'entité homogène, car les deux partagent les mêmes attributs. Cette approche peut entraîner un problème de cohérence des données, car un superviseur peut être présent à la fois dans l'entité "Employé" et dans l'entité "Superviseur". En conséquence, toute modification d'information, telle que l'adresse ou le salaire, doit être mise à jour simultanément dans ces deux entités pour éviter des incohérences et garantir l'intégrité des données.



Réflexivité



employe1

1

Jhon

DOE

employe3

2

Martin

SMITH

employe4

3

Eliza

DAVIS

1

employe4

4

Daniel

LAMMERS

2

Règles de normalisation

Les règles de normalisation aident à structurer les données de manière à minimiser la redondance et les anomalies de mise à jour.



Normalisation 1FN

Première forme normale (1FN) : Les valeurs dans chaque colonne doivent être atomiques (pas de listes, sous-ensembles, ou valeurs multiples dans une seule cellule). Chaque attribut doit être monovalué (une seule valeur par ligne).

Pour passer un MCD en 1FN, il faut :

Éliminer les attributs répétitifs : Chaque colonne doit contenir des valeurs atomiques, c'est-à-dire des valeurs indivisibles.

Créer des tables séparées pour les groupes d'attributs répétitifs : Si une table contient des groupes d'attributs répétitifs, ces groupes doivent être déplacés vers des tables séparées.

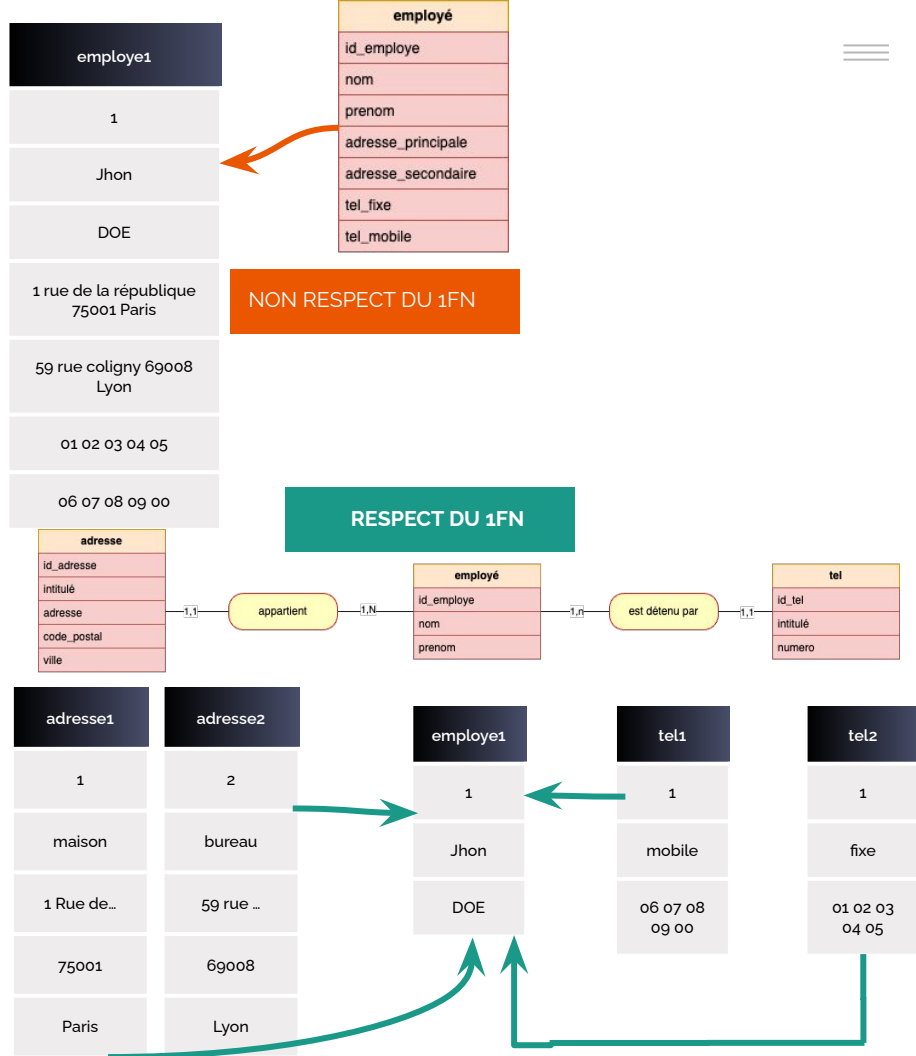
Assurer l'unicité des colonnes : Chaque colonne doit avoir un nom unique et des valeurs uniques.

Avantages de la 1FN

Réduction de la redondance : En éliminant les attributs répétitifs, on réduit la redondance des données.

Simplicité de la structure : Les tables sont plus simples et plus faciles à comprendre.

Facilité de manipulation : Les opérations de mise à jour et de suppression sont plus faciles à gérer.



Normalisation: 2FN

Deuxième forme normale (2FN) : La table doit être en 1FN, et tous les attributs non-clés doivent dépendre entièrement de la clé primaire (pas de dépendances partielles).

Pour passer un MCD en 2FN, il faut :

- 1. Identifier les clés primaires composées : Ce sont des clés primaires qui sont constituées de plusieurs attributs.
- 2. Éliminer les dépendances partielles : Si un attribut non-clé dépend seulement d'une partie de la clé primaire, il doit être déplacé dans une nouvelle entité.
- 3. Revoir les relations : Créer de nouvelles entités pour séparer les dépendances partielles.

Avantages de la 2FN :

Réduction des redondances : En éliminant les dépendances partielles, on évite de répéter des informations à chaque enregistrement.

Meilleure intégrité des données : Les anomalies de mise à jour sont moins probables.

Table avant normalisation (avant 2FN)

ID_réservation	ID_client	Nom_client	ID_chambre	Nom_employé
101	1	Alice	201	M. Lefevre
101	2	Bob	202	M. Lefevre
102	3	Claire	203	Mme Durand
103	4	David	201	M. Lefevre

Après normalisation

Table des Clients

ID_client	Nom_client
1	Alice
2	Bob
3	Claire
4	David

Table des Chambres

ID_chambre	Type_chambre
201	Simple
202	Double
203	Suite

Table des Employés

ID_employé	Nom_employé
1	M. Lefevre
2	Mme Durand

Table des Réservations et Employés

ID_réservation	ID_employé
101	1
102	2
103	1

Table des Réservations

ID_réservation	ID_client	ID_chambre
101	1	201
101	2	202
102	3	203
103	4	201

Normalisation: 3FN

Troisième forme normale (3FN) : La table doit être en 2FN, et il ne doit pas y avoir de dépendances transitives (c'est-à-dire qu'un attribut non-clé ne doit pas dépendre d'un autre attribut non-clé).

Elle a pour but d'éliminer les dépendances transitives et de garantir que les attributs sont uniquement dépendants de la clé primaire.

Pour passer un MCD en 3FN, il faut :

- 1. Identifier les dépendances transitives.
- 2. Créer des entités supplémentaires pour isoler les attributs transitifs.
- 3. Revoir les associations et cardinalités pour préserver la cohérence

Avantages de la 3FN

Réduction des redondances : L'Adresse est enregistrée une seule fois dans la table des Propriétaires.

Cohérence : Une modification de l'Adresse d'un Propriétaire ne nécessite qu'une seule mise à jour.

Facilité de maintenance : La structure est plus simple à gérer et évite les anomalies de mise à jour, de suppression et d'insertion.



Table des Taxis avant Normalisation

Immatriculation	Marque	Modèle	Propriétaire	Adresse	Puissance	Carburant	Couleur
1234AB56	Peugeot	308	M. Dupont	12 rue de Paris	120 CV	Essence	Bleu
5678CD90	Renault	Clio	Mme Martin	45 avenue des Champs	90 CV	Diesel	Rouge
1357EF24	Peugeot	208	M. Dupont	12 rue de Paris	100 CV	Essence	Vert

Après normalisation

Table des Taxis

Taxi_ID	Immatriculation	Marque	Modèle
1	1234AB56	Peugeot	308
2	5678CD90	Renault	Clio
3	1357EF24	Peugeot	208

Table des Propriétaires

Propriétaire_ID	Nom	Adresse
1	M. Dupont	12 rue de Paris
2	Mme Martin	45 avenue des Champs

Table des Détails des Taxis

Detail_ID	Taxi_ID	Puissance	Carburant	Couleur
1	1	120 CV	Essence	Bleu
2	2	90 CV	Diesel	Rouge
3	3	100 CV	Essence	Vert



Dictionnaire des Données

Le Dictionnaire des Données est un répertoire qui contient toutes les définitions des données utilisées dans le système d'information.

Il permet de garantir la cohérence et la compréhension des données par tous les acteurs du projet.

Chaque élément, qu'il s'agisse d'une entité, d'un attribut ou d'une association, doit être clairement défini dans ce dictionnaire afin de garantir la cohérence et la précision de la modélisation.

- Objectifs :**
- Clarification des concepts :** Chaque terme est défini de manière précise.
 - Communication :** Il facilite la compréhension et la discussion entre les membres de l'équipe de développement et avec les utilisateurs finaux.
 - Garantie de cohérence :** Il aide à maintenir l'intégrité du modèle en assurant qu'il n'y a pas de redondance ou d'ambiguïté.
 - Base pour la mise en œuvre :** Le dictionnaire des données est un support essentiel pour la phase de conception de la base de données, notamment lors de la création du modèle logique et physique.
 - Maintenance :** Aider à la maintenance et à l'évolution du système d'information.

Table : Users

Attribut	Type	Description	Contraintes
id	Int	Identifiant unique de l'utilisateur	Clé primaire, Non nul, Auto-incrémenté
nom	String	Nom de famille de l'utilisateur	Non nul
email	String	Email unique de l'utilisateur	Unique, Non nul
mot_de_passe	String	Mot de passe crypté de l'utilisateur	Non nul
date_naissance	Date	Date de naissance de l'utilisateur	Optionnel
is_admin	Boolean	Indique si l'utilisateur est un administrateur	Non nul, Par défaut : false
date_creation	Date	Date de création du compte utilisateur	Non nul, Par défaut : date actuelle
presentation	Text	Description personnelle ou biographie de l'utilisateur	Optionnel
salaire	Decimal	Salaire annuel de l'utilisateur	Optionnel
longitude	Float	Coordonnée géographique longitude	Optionnel

Table : Users_Groups

Attribut	Type	Description	Contraintes
user_id	Int	Identifiant de l'utilisateur (Clé étrangère)	Clé étrangère, Non nul
group_id	Int	Identifiant du groupe (Clé étrangère)	Clé étrangère, Non nul
date_ajout	Date	Date à laquelle l'utilisateur a été ajouté au groupe	Non nul, Par défaut : date actuelle