

# INTRODUCTION AUX BASES DE DONNÉES

Martha GONTIER  
m.gonzalez-garcia@aforp.fr

22/08/2025

1

1

## OBJECTIFS

A la fin de ce module, l'apprenti sera en capacité de :

- ✓ Expliquer les notions fondamentales de **données, bases de données et systèmes de gestion de bases de données relationnelles (SGBDR)**.
- ✓ Identifier les enjeux liés à la **qualité des données**.
- ✓ Comprendre le rôle et la structure d'un **modèle conceptuel de données (MCD)**.
- ✓ Vérifier la **validité** d'un MCD simple et proposer des ajustements.
- ✓ Modéliser un système d'information simple à l'aide de la méthode **MERISE**.
- ✓ Appliquer les règles de **conversion** d'un modèle conceptuel de données vers un modèle physique.
- ✓ Mettre en œuvre une base de données relationnelle dans un SGBDR.

22/08/2025

2

2

1

# AGENDA

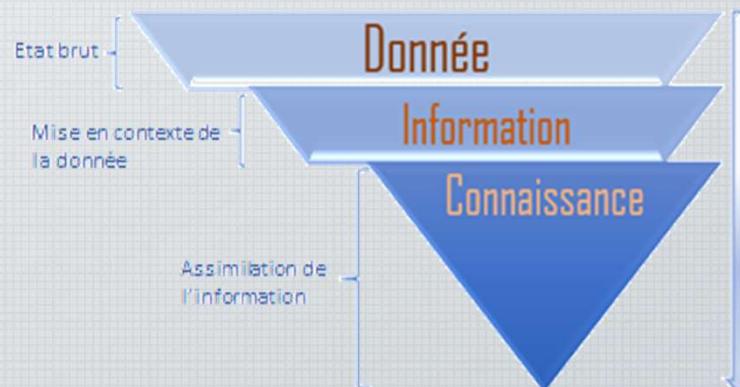
3



- 1 Introduction aux systèmes d'information
- 2 Qualité et gestion des données
- 3 Les bases de données et les systèmes de gestion de bases de données (SGBD)
- 4 Modélisation relationnelle avec MERISE

3

## Les données



4

# Donnée : un fait brut, impossible à interpréter sans un contexte

Etat brut

Donnée



Les données proviennent directement de l'observation, de capteurs, d'enquêtes ou d'autres sources, mais ne servent à rien tant qu'elles ne sont pas interprétées ou organisées.

## UNE STRUCTURE

- Simple: prix, nom, date
- Complex: personne, document, image

## UNE SÉMANTIQUE DANS UN CONTEXTE

- Le solde d'un compte courant
- Une photo de Paris réalisée par le satellite

## UN PROPRIÉTAIRE

- Responsable de la création d'une donnée
- Définit les règles pour son identification et son intégrité:
  - « La température de l'air est comprise entre -30° et +40°C »
  - « Le salaire de l'année n est supérieur au salaire de l'année n-1 »
- Accorde des droits d'utilisation

## UN COÛT

Acquisition, traitement, transport, stockage

5

5

# Information : mise en contexte d'une donnée. Elle doit être basée sur des faits vérifiables.

Les chiffres peuvent représenter

1. Dans un contexte d'une banque :
  - Le solde d'un compte bancaire
  - Le montant d'un retrait
2. Dans le contexte d'une entreprise:
  - Le salaire des employés



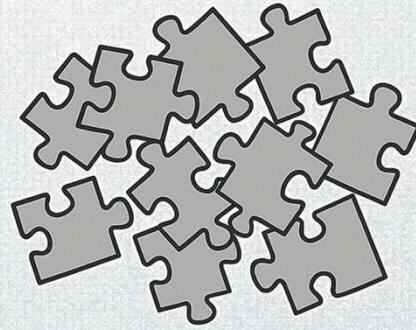
L'information donne du sens aux données et la rend utilisable pour la prise de décision ou la résolution de problèmes.

22/08/2025

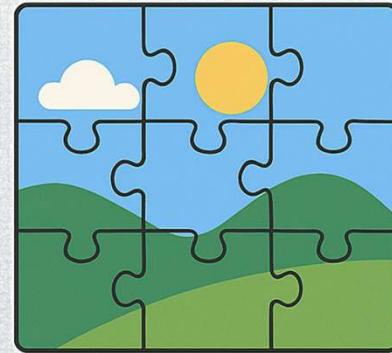
6

6

## Donnés vs Information (Exemple)



Données

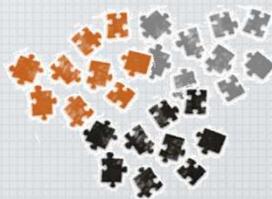


Information

22/08/2025

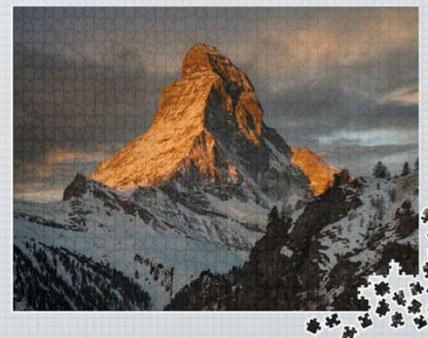
7

## Donnés vs Information (Exemple)



Données

Symbolisent les données **brutes**



Information

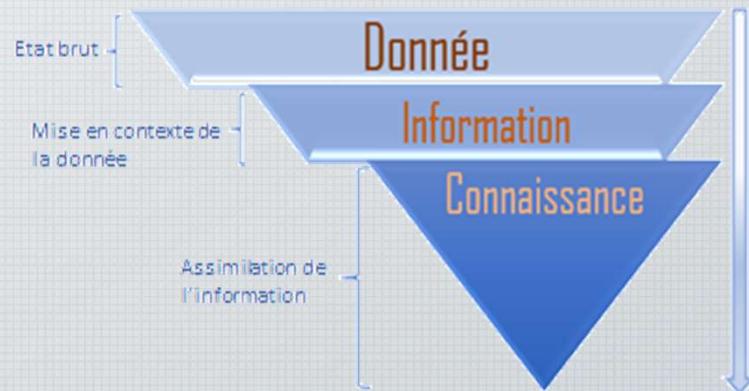
Représente l'information **structurée et compréhensible**.

22/08/2025

8

8

**Connaissance :** appropriation et à l'utilisation de l'information pour comprendre, raisonner ou agir.



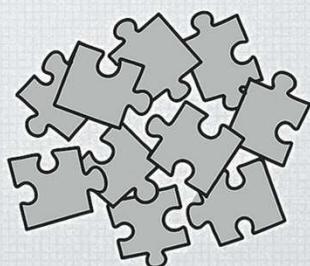
22/08/2025

**OBJECTIF : PRENDRE DE DÉCISIONS**

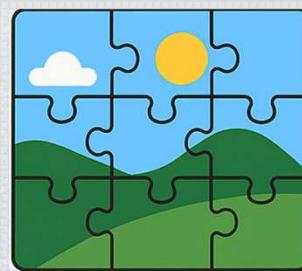
9

9

## Donnés, Information et Connaissance



**Données**



**Information**

Prise de décision

**Une balade peut être organisée, sans risque de pluie**

25/08/2025

10

10

# Donnés, Information et Connaissance



Données



Information

Pour profiter pleinement de Zermatt, il faut éviter les périodes de haute affluence comme Noël et février.

Connaissance

Prise de décision : Organiser le voyage pour le mois de janvier

25/08/2025

11

11

**Le système d'information :** ensemble organisé de ressources (humaines, matérielles, financières, ) et de procédures, permettant l'exécution des processus internes à une organisation

- UN MODÈLE DE L'ORGANISATION
  - Principaux éléments caractéristiques
  - Tâches des différents acteurs (les « métiers »)
  - Procédures régissant les activités et les interactions entre acteurs
  - Entités caractéristiques des activités (les « objets métiers »)
- ÉLÉMENTS
  - **Donnée** : un fait brut, impossible à interpréter tel que exemple : un stock à 0
  - **Information** : une donnée prenant du sens dans un contexte d'interprétation
  - **Connaissance** : une information analysée dans un contexte d'action



OBJECTIF : ASSURER LE BON FONCTIONNEMENT D'UNE ORGANISATION

25/08/2025

12

12

## Evolution des systèmes d'information

Période	Caractéristique principale	Exemple / technologie
Avant informatisation	Travail manuel (papier)	Registres, archives
1950-1970	Mainframes, automate de tâches précises	Gestion paie, comptabilité
1980-1990	PC et architectures client-serveur	Collaboration réseau, bases de données
Aujourd'hui	SI intégrés, orientés processus et données	ERP, CRM, Big Data, cloud computing

25/08/2025

13

13

## Décentralisation : la problématique dans la gestion des données

Direction des études



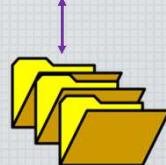
Bibliothèque



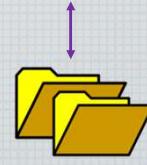
Laboratoire de recherche



Système en Cobol



Système en Fortran



Système en C

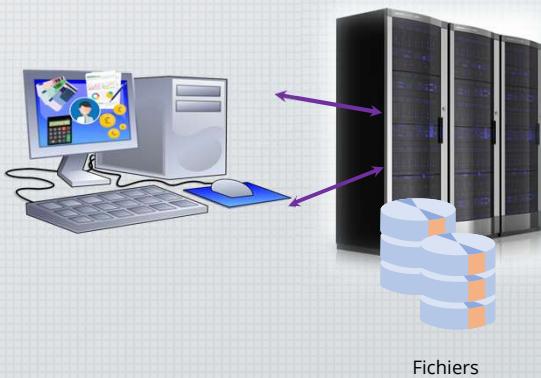


25/08/2025

14

14

## Systèmes décentralisés : problématique dans la gestion des données



- Difficulté à saisir les liens entre les données
- Pas de partage de données entre les utilisateurs
- Pas de vision globale
- Redondance
- Risque d'incohérence
- Dépendance entre les données et les traitements
- Problème de gestion de la sécurité
- Multiplicité des traitements, des langages
- Difficulté de maintenance

25/08/2025

15

15

## QUIZ

- Donner un exemple de donnée / d'information / de connaissance.
- Citer les composants d'un SI.
- Expliquer en quoi un SI est utile à une organisation.

25/08/2025

16

16



# Qualité et intégrité des données

**Intégrité des données :**

- Fait référence à l'**exactitude**, la **cohérence** et la **fiabilité** des données au cours de leur cycle de vie dans une base de données ou un système d'information.
- Elle vise à s'assurer que les données stockées, manipulées ou récupérées restent correctes, complètes et non corrompues, indépendamment des actions entreprises par les utilisateurs, les applications ou le système.

25/08/2025



17

17



# Disponibilité

**Exemple :**

- Dans une banque, les clients accèdent à leurs comptes via une application mobile pour vérifier leur solde ou effectuer des transferts. La base de données derrière l'application doit être **accessible en permanence** pour répondre à ces requêtes.

**Problème :**

- Si la base de données n'est pas disponible (en cas de panne du serveur ou de maintenance non planifiée), les clients ne pourront pas accéder à leur compte, ce qui peut entraîner des pertes financières ou une perte de confiance dans l'institution.

25/08/2025



18

18



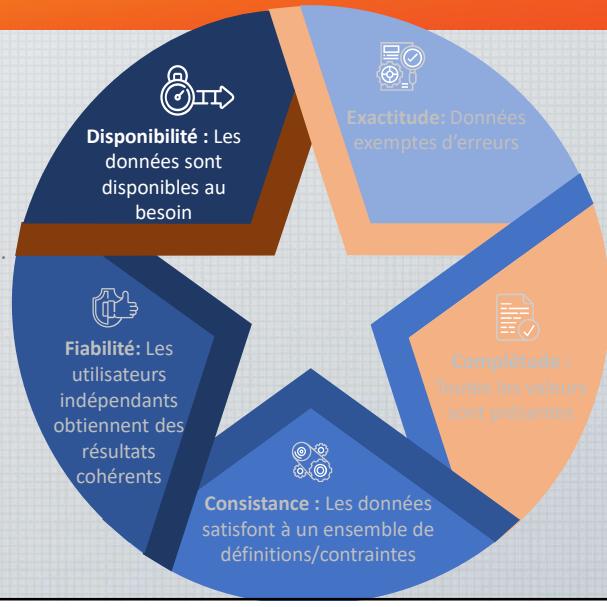
# Disponibilité

## Moyens de non-respect :

- Panne de système ou serveur** : Bases de données inaccessibles temporairement ou définitivement.
- Permissions mal configurées** : Les utilisateurs autorisés ne peuvent pas accéder aux données.
- Mauvais processus de sauvegarde/restauration** : Les données sont perdues ou non récupérables après un incident.
- Dépendance à des systèmes tiers non fiables** : Si les données proviennent de services externes, leur indisponibilité affecte tout le système.

25/08/2025

19



19



# Exactitude

## Exemple :

- Une entreprise de livraison enregistre les adresses des clients dans sa base de données. Si l'adresse est incorrecte (par exemple, une faute de frappe dans le code postal ou le nom de rue), le colis ne sera pas livré au bon endroit.

## Problème :

- Une inexactitude dans les données peut entraîner des erreurs d'expédition, des retards et des coûts supplémentaires, affectant à la fois l'entreprise et le client.

25/08/2025

20



20

10

## Exactitude

**Moyens de non-respect :**

- Saisie manuelle erronée** : Une faute de frappe ou une erreur de compréhension de l'utilisateur.
- Capteurs defectueux (dans les systèmes IoT)** : Valeurs inexactes transmises automatiquement.
- Mauvaise transformation des données** : Une erreur dans un script ETL (Extract, Transform, Load) qui tronque ou arrondit mal une valeur.
- Sources non fiables** : Importation de données provenant de sites ou systèmes peu rigoureux.

25/08/2025 21

The diagram illustrates the four dimensions of data quality: Availability, Integrity, Consistency, and Completeness. Each dimension is represented by a colored hexagon with an icon and a descriptive text. The colors follow a gradient from dark blue to light orange.

- Availability** (Dark Blue): Les données sont disponibles au besoin.
- Integrity** (Light Blue): Exactitude: Données exemptes d'erreurs.
- Consistency** (Medium Blue): Complétude : Toutes les valeurs sont présentes.
- Completeness** (Orange): Consistance : Les données satisfont à un ensemble de définitions/contraintes.

21

## Complétude

**Exemple :**

- Dans un système de gestion de la santé, les dossiers médicaux des patients doivent contenir des informations complètes telles que les antécédents médicaux, les traitements en cours, les allergies, etc.

**Problème :**

- Si les données ne sont pas complètes, un médecin pourrait prescrire un médicament auquel le patient est allergique, ce qui mettrait en danger la santé de ce dernier.

25/08/2025 22

The diagram illustrates the four dimensions of data quality: Availability, Integrity, Consistency, and Completeness. Each dimension is represented by a colored hexagon with an icon and a descriptive text. The colors follow a gradient from dark blue to light orange.

- Availability** (Dark Blue): Disponibilité : Les données sont disponibles au besoin.
- Integrity** (Light Blue): Exactitude: Données exemptes d'erreurs.
- Consistency** (Medium Blue): Fiabilité: Les utilisateurs indépendants obtiennent des résultats cohérents.
- Completeness** (Orange): Complétude : Toutes les valeurs sont présentes.

22

## Complétude

**Moyens de non-respect :**

- Champs non obligatoires ignorés par l'utilisateur (par exemple, adresse email non renseignée).
- Erreurs de migration de données : Perte de colonnes ou de lignes.
- Problèmes d'intégration : Des systèmes connectés n'échangent pas toutes les données requises.
- Modèles mal définis : Certains champs importants ne sont même pas inclus dans le modèle de données.

25/08/2025 23

23

## Consistance

**Exemple :**

- Une entreprise de vente en ligne gère les stocks de produits à travers plusieurs canaux de vente (site web, application mobile, boutique physique). Si un client achète un produit en ligne, la quantité disponible en stock doit être mise à jour dans toutes les bases de données en même temps.

**Problème :**

- Si la base de données n'est pas cohérente (elle n'est pas mise à jour correctement dans tous les canaux), un client pourrait acheter un produit en ligne qui est déjà épuisé en magasin, ce qui entraînerait des commandes impossibles à honorer et des annulations de ventes.

25/08/2025 24

24

# Consistance



**Moyens de non-respect :**

- Absence de règles de validation** (ex. : date de naissance future possible).
- Données contradictoires dans différentes tables** (ex. : un client actif dans une table, inactif dans une autre).
- Mauvais formats** (ex. : numéros de téléphone mal structurés, formats de date incohérents).
- Synchronisation imparfaite entre bases** : Deux systèmes n'appliquent pas les mêmes règles métier.



25/08/2025 25

25

# Fiabilité



**Exemple :**

- Dans une entreprise, plusieurs équipes (comptabilité, ventes, logistique) accèdent aux mêmes données clients. La base de données doit garantir que, quel que soit l'utilisateur ou l'équipe, les informations consultées (comme l'adresse du client ou l'historique des commandes) sont identiques et cohérentes.

**Problème :**

- Si la fiabilité n'est pas respectée, un utilisateur pourrait obtenir des informations différentes par rapport à un autre utilisateur. Par exemple, si l'équipe des ventes voit une adresse différente de celle vue par l'équipe de la logistique, cela pourrait entraîner des erreurs dans les livraisons ou les facturations.



25/08/2025 26

26

## Fiabilité



**Moyens de non-respect :**

- **Processus non reproductibles** : Une requête SQL retourne un résultat différent selon l'heure ou l'utilisateur.
- **Données mises à jour sans contrôle** : Des valeurs changent constamment sans logique ou historique.
- **Manque de traçabilité** : On ne sait pas qui a modifié quoi et pourquoi.
- **Accès concurrents mal gérés** : Des écritures simultanées entraînent des incohérences.



25/08/2025 27

27

## Exemple



Une entreprise collecte chaque jour des logs de connexions SSH depuis ses serveurs pour détecter les accès non autorisés. Ces logs sont traités par un script ETL (en Python avec Pandas ou PySpark) qui les centralise dans une base de données pour analyse et alertes de sécurité.



25/08/2025 28

28

## Exemple

The diagram shows a data flow from 'Serveurs' (Servers) to 'Logs de connexion réseau' (Network connection logs), then through a 'Script ETL' (Extract, Transform, Load) process, into a 'Base de données centrale' (Central database), and finally to a 'Dashboard'. Each stage has associated quality compromise scenarios:

- Serveurs:** Complétude compromise. Error: Certain logs do not contain the source IP address ('src\_ip') due to a router failure. Impact: Security reports do not reflect reality.
- Logs de connexion réseau:** Disponibilité compromise. Error: A server outage prevents the ETL script from running for two days, with no backfill mechanism. Impact: Log files for two days are missing, and analysts cannot investigate incidents.
- Script ETL:** Exactitude compromise. Error: A bug in the script converts the 'status\_code' field from 'SUCCESS'/'FAIL' to 'suCESS'/'FAIL ' (with a space). Impact: Statistics for failed attempts are inaccurate.
- Base de données centrale:** Consistance compromise. Error: Date formats in the database (e.g., YYYY-MM-DD) differ from those in the logs (e.g., DD/MM/YYYY), causing interpretation errors in joins and comparisons. Impact: Analysts get inconsistent results based on the source queried.
- Dashboard:** Fiabilité compromise. Error: Two analysts run the same query from outside, getting different results because the log table is updated continuously without versioning. Impact: Confidence in dashboards is lost.

25/08/2025 29

29

## Exercice

Identifiez quel type d'attaque peut porter atteinte à chacun des principes de qualité et intégrité

Principe de qualité	Description
Disponibilité	Les données sont disponibles au besoin
Exactitude	Données exemptes d'erreurs
Fiabilité	Les utilisateurs indépendants obtiennent des résultats cohérents
Complétude	Toutes les valeurs sont présentes
Consistance	Les données satisfont à un ensemble de définitions/contraintes
Intégrité	(Icone de cadenas)

25/08/2025 30

30



## Exercice

Principe d'intégrité des données	Type de cyberattaque	Impact
Disponibilité	DDoS (Denial of Service)	Rend les données inaccessibles aux utilisateurs légitimes.
Exactitude	Injection SQL	Introduit des erreurs ou des données frauduleuses dans la base de données.
Complétude	Ransomware	Chiffre ou supprime des données, rendant la base de données incomplète.
Consistance	Man-in-the-Middle (MITM)	Provoque des incohérences entre différentes versions des mêmes données.
Fiabilité	Altération des journaux	Empêche d'obtenir des résultats fiables et cohérents lors des audits ou des analyses.

25/08/2025

31

31

## Pourquoi une gestion structurée des données ?

Les exigences croissantes en qualité des données (Disponibilité, Exactitude, Cohérence, Complétude, Fiabilité...) rendent indispensable une organisation rigoureuse.



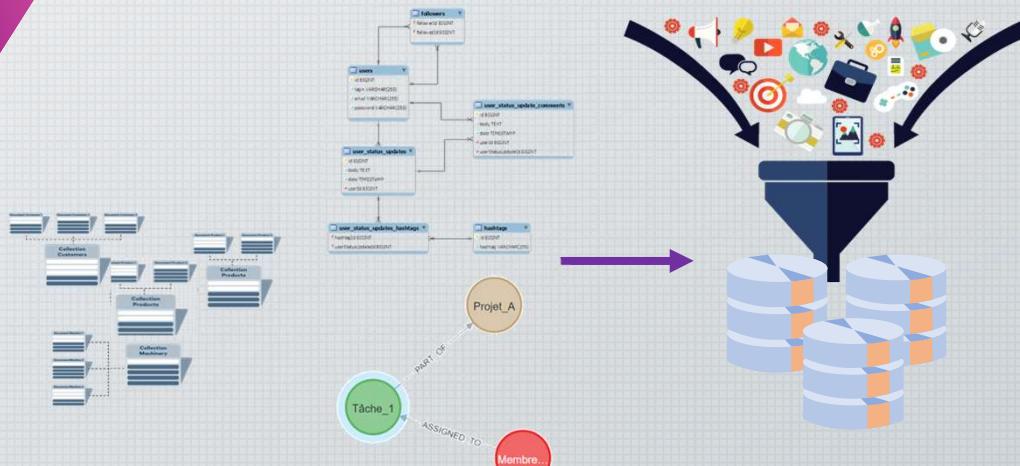
25/08/2025

32

32

## La base de données ...

.... est une collection de données décrites selon un certain modèle



25/08/2025

33

## La base de données doit satisfaire ...

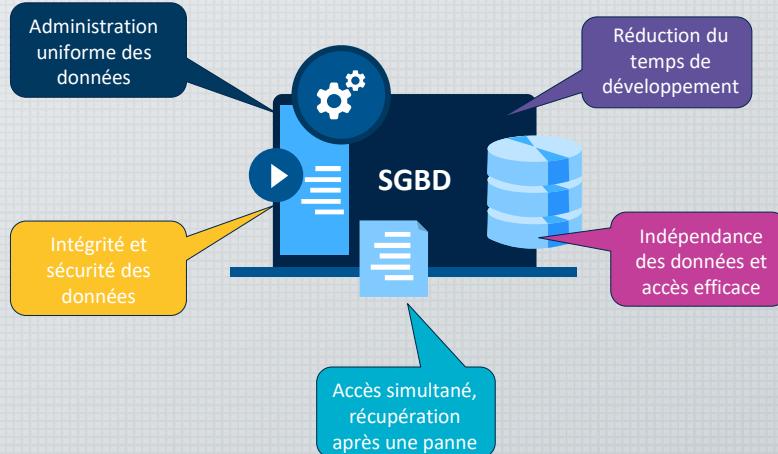
- ✓ Bonne représentation du monde réel, une image aussi fidèle que possible de la réalité à tout instant
  - ✓ Une représentation fidèle → Une information fiable et mise à jour
- ✓ Non redondance de l'information
- ✓ Indépendance entre les données et leur traitement
- ✓ Sécurité et confidentialité des données
- ✓ Partage des données
- ✓ Séparation entre les descriptions logiques et physiques des données

25/08/2025

34

# Le système de gestion de bases des données (SGBD)

Système logiciel permettant la consultation, la mise à jour, la structuration, le partage et la sécurité de données stockées dans une base.



25/08/2025

35

35



## Propriétés ACID d'une transaction en base de données

Garantissent la fiabilité et la cohérence des données lorsqu'une transaction est exécutée

### 01 Atomicité

En cas d'échec d'une instruction de la transaction, l'intégralité de la transaction échoue et la base de données reste inchangée.

### 02 Cohérence

La transaction doit répondre à tous les protocoles définis par le système -- aucune transaction partiellement terminée.

### 03 Isolation

Aucune transaction n'a accès à une autre transaction inachevée. Chaque transaction est indépendante.

### 04 Durabilité

Une fois qu'une transaction a été validée, elle le restera grâce à l'utilisation de journaux de transactions et de sauvegardes.

Lors d'un virement bancaire, si le débit du compte A échoue, alors le crédit du compte B n'aura pas lieu

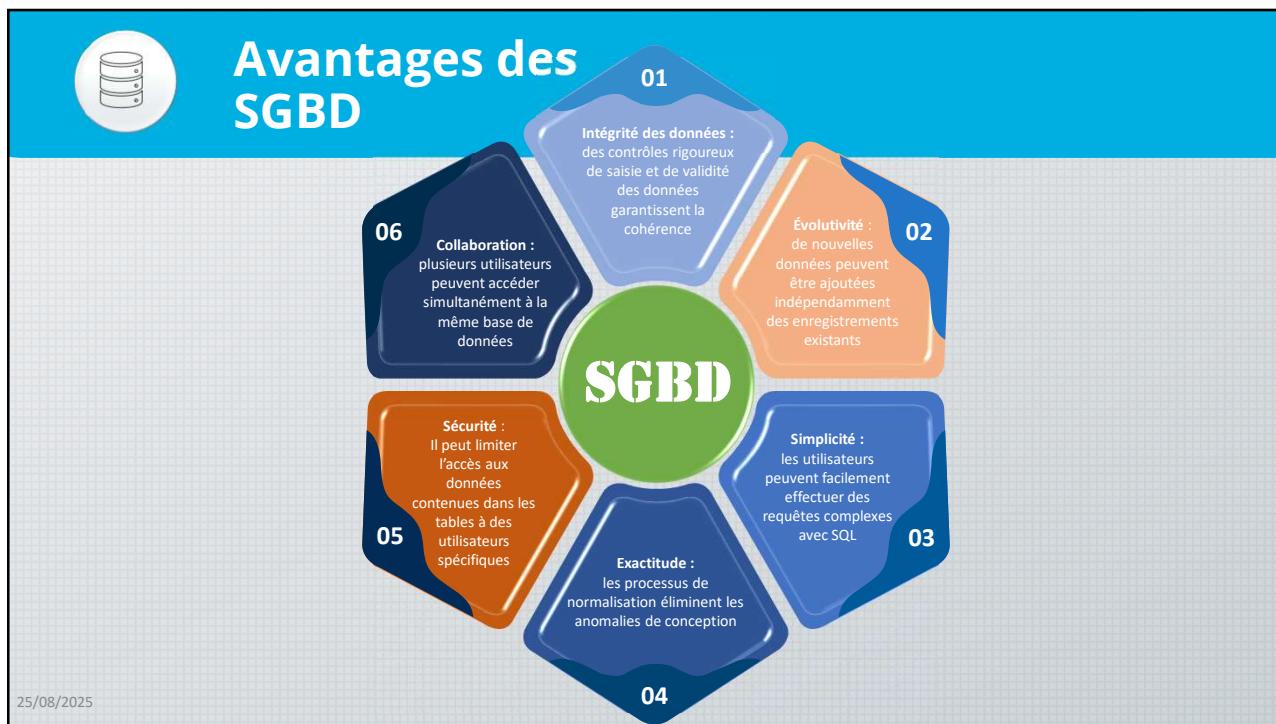
Si une règle stipule qu'un solde ne peut pas être négatif, aucune transaction ne doit violer cette règle

Deux clients retirant de l'argent simultanément ne doivent pas interférer et provoquer une incohérence dans le solde du compte

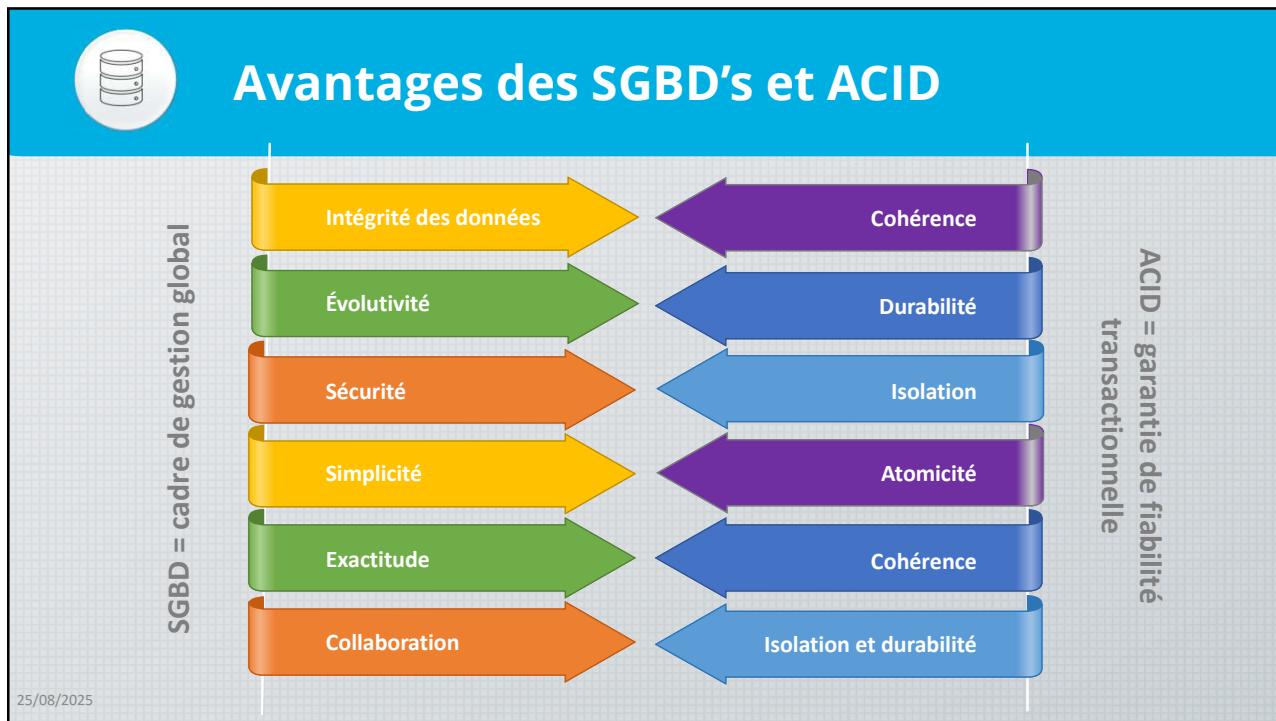
Après un virement bancaire confirmé, même si le serveur tombe en panne, l'opération doit être sauvegardée et réappliquée lors du redémarrage

25/08/2025

36



37



38

## Composants d'un SGBD

**Langages de définition et de manipulation**

- DDL (Data Definition Language) : définit la structure de la base (tables, schémas...)
- DML (Data Manipulation Language) : permet d'ajouter, modifier, supprimer ou consulter les données
- DCL (Data Control Language) : gère les droits d'accès et la sécurité

**Moteur de base de données**

Cœur du SGBD, il exécute les requêtes, gère le stockage, les accès, la sécurité et la cohérence des données. Il prend en charge la gestion des transactions et la concurrence entre utilisateurs

**Mécanisme d'intégrité**

Assurent la cohérence et la fiabilité des données, via des contraintes (unicité, clé primaire, intégrité référentielle, validation de formats) et des vérifications systématiques lors des modifications

**Gestion des transactions**

Garantit que toutes les opérations sur la base de données sont sûres, cohérentes et fiables. Il applique le principe ACID (Atomicité, Cohérence, Isolation, Durabilité)

**Catalogue système (ou dictionnaire de données)**

Stocke les métadonnées : définitions, descriptions des objets de la base, droits d'accès, règles d'intégrité,... Utilisé à chaque opération

25/08/2025

39

## SGBD et abstraction des données

Les SGBD's visent à simplifier l'accès et la manipulation des données en cachant les détails complexes de leur stockage et de leur gestion. Ce qui permet aux utilisateurs de travailler avec les données de manière plus intuitive, sans avoir à comprendre comment ces données sont réellement organisées ou stockées à un niveau physique. L'**abstraction des données** permet de dissocier l'architecture technique sous-jacente des bases de données des besoins métier.

**Le terme abstraction désigne le processus par lequel on simplifie ou masque certains détails complexes pour ne se concentrer que sur les aspects essentiels ou pertinents d'un objet, d'un concept ou d'un système. L'abstraction permet de manipuler des concepts complexes de manière plus intuitive en cachant les détails sous-jacents non nécessaires à l'utilisateur ou au concepteur.**

**L'abstraction des données vise à simplifier l'accès et la manipulation des données en cachant les détails complexes de leur stockage et de leur gestion. Ce concept permet aux utilisateurs de travailler avec les données de manière plus intuitive, sans avoir à comprendre comment ces données sont réellement organisées ou stockées à un niveau physique.**

25/08/2025

40

40



# Abstraction des données (objectifs)

**Réduction de la complexité**

En omettant les détails non pertinents, l'abstraction permet de travailler avec des concepts plus simples et de se focaliser sur les fonctionnalités principales

**Modularité**

L'abstraction encourage une séparation claire des responsabilités et des couches dans les systèmes complexes, facilitant ainsi la gestion et la maintenance

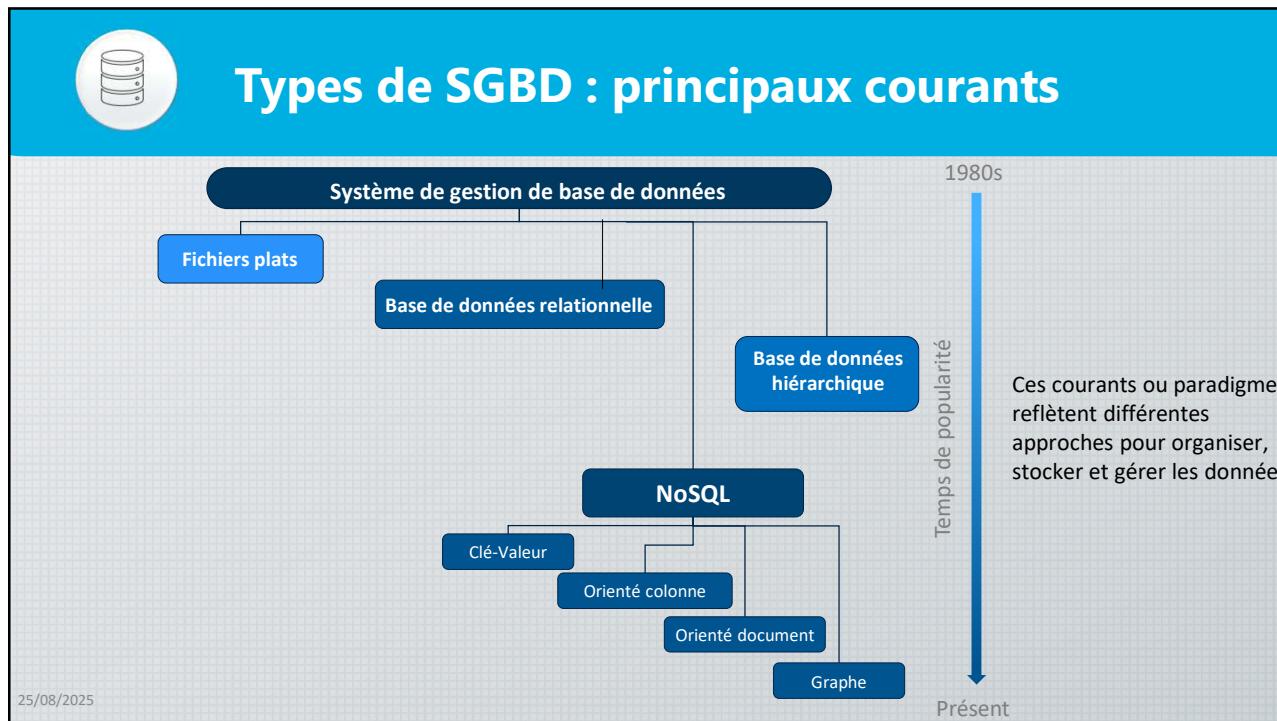
**Réutilisation**

En créant des niveaux d'abstraction, on peut réutiliser les mêmes structures ou composants dans différents contextes, car les détails spécifiques sont cachés

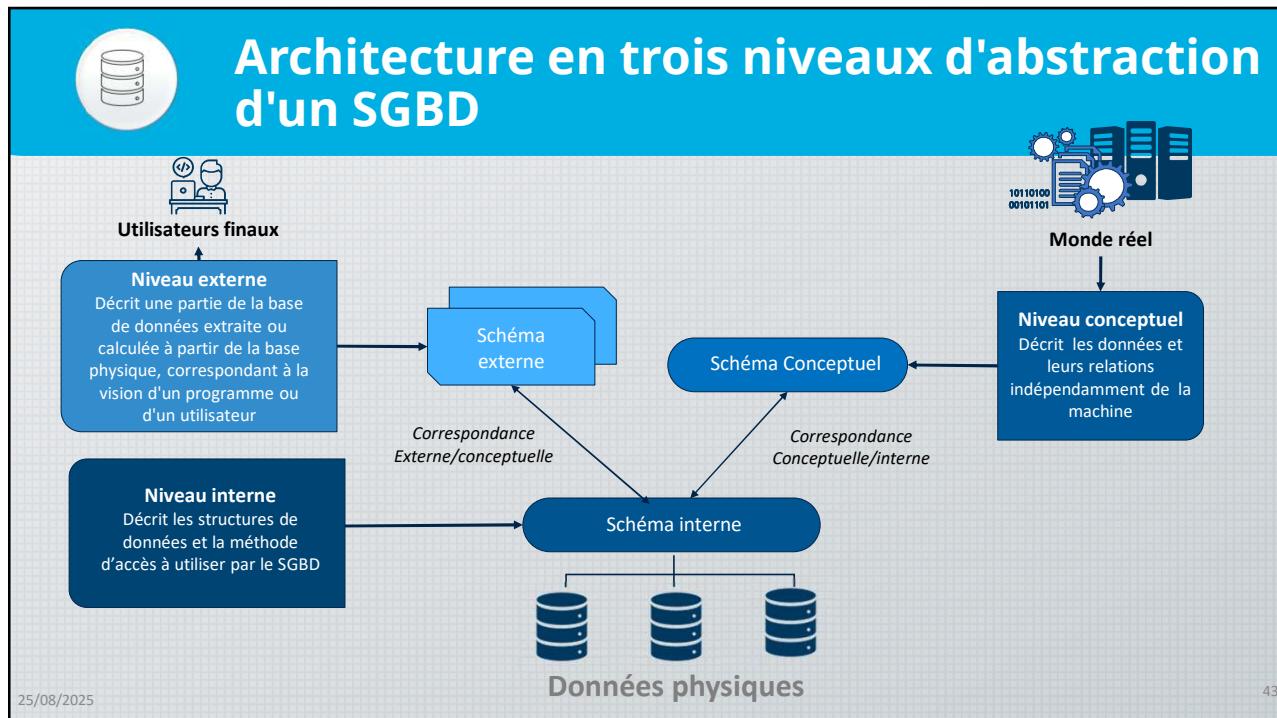
25/08/2025

41

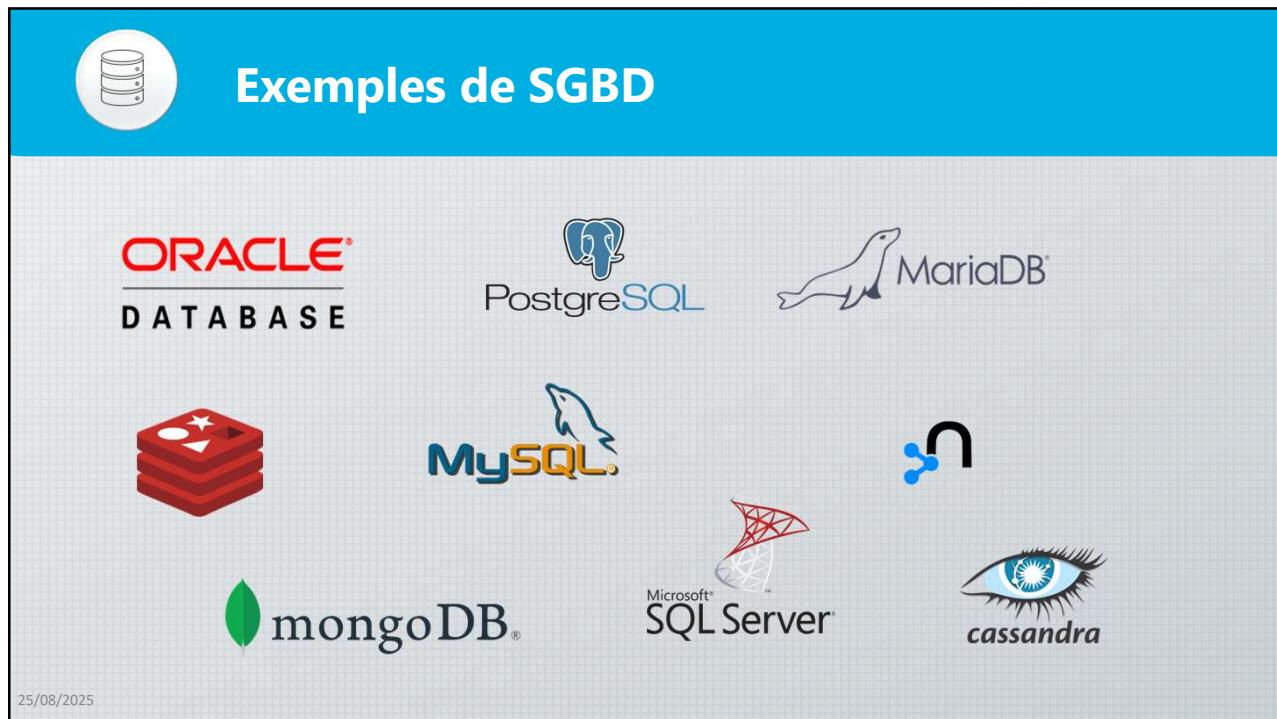
41



42



43



44



## Comparaison de SGBD

SGBD	Type	Caractéristiques principales	Cas d'usage typique
Oracle	Relationnel (propriétaire)	Très robuste, sécurisé, adapté aux très gros volumes.	Grandes entreprises (banques, assurances).
SQL Server	Relationnel (propriétaire)	Intégré à l'écosystème Microsoft, convivial, outils BI.	Entreprises utilisant Windows et Microsoft BI.
MySQL	Relationnel (open source)	Simple, rapide, très utilisé sur le web.	Sites web, applications en ligne (WordPress).
PostgreSQL	Relationnel (open source)	Respect strict des standards SQL, extensible, puissant.	Projets complexes, académiques et professionnels.
SQLite	Relationnel (open source, embarqué)	Très léger, pas besoin de serveur, utilisé localement.	Applications mobiles, petits projets, prototypage.
MongoDB	NoSQL (document)	Stocke les données sous forme de documents JSON, flexible.	Applications web modernes, données semi-structurées.
Neo4j	NoSQL (graphes)	Spécialisé dans la gestion des relations entre entités.	Réseaux sociaux, recommandations, cartographie.
Redis	NoSQL (clé-valeur, en mémoire)	Très rapide, stocke les données en RAM.	Caching, temps réel, gestion de sessions.
Cassandra	NoSQL (colonne)	Distribué, très scalable, gère des volumes massifs.	Big Data, IoT, systèmes distribués.

25/08/2025

45

45



## Modèle relationnel

### OBJECTIF :

- Garantir l'indépendance entre les niveaux logique et physique

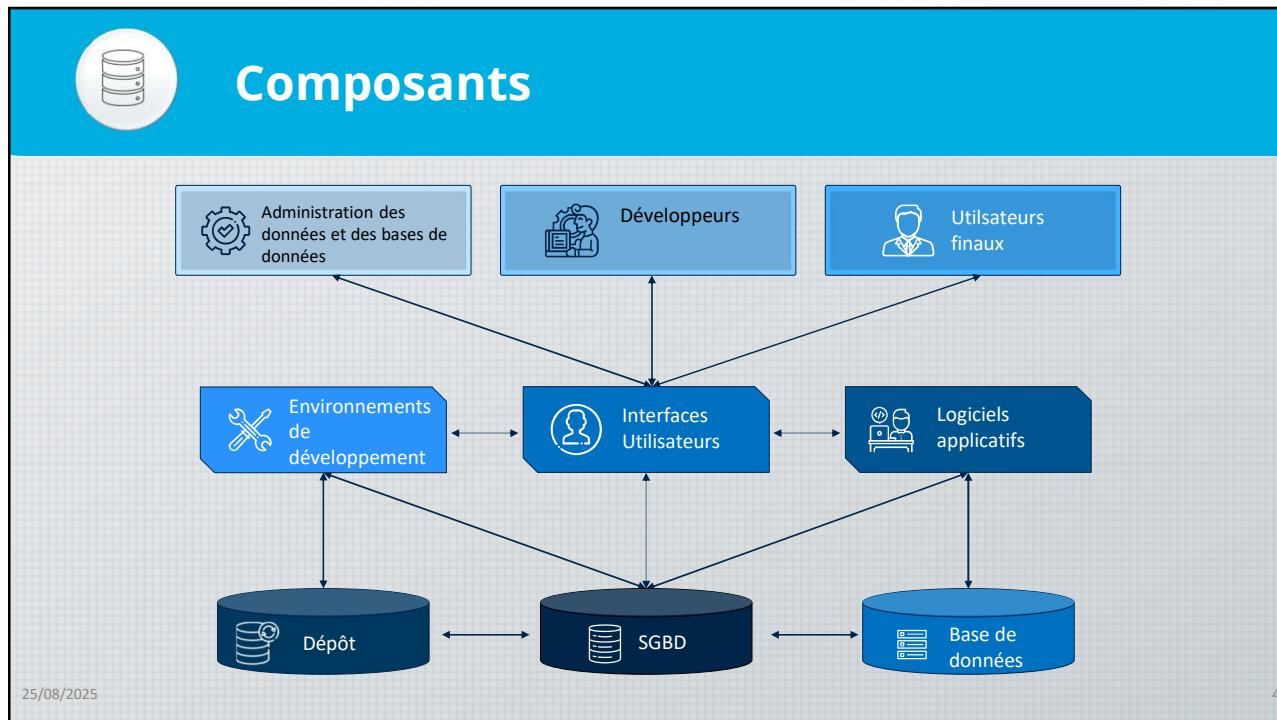
### CONCEPTS

- **Relation** regroupe un ensemble d'attributs qui caractérisent une proposition ou une combinaison de propositions.
- La **logique des prédictats** sert à exprimer des formules logiques qui permettent de combiner des relations préalablement définies pour en construire de nouvelles

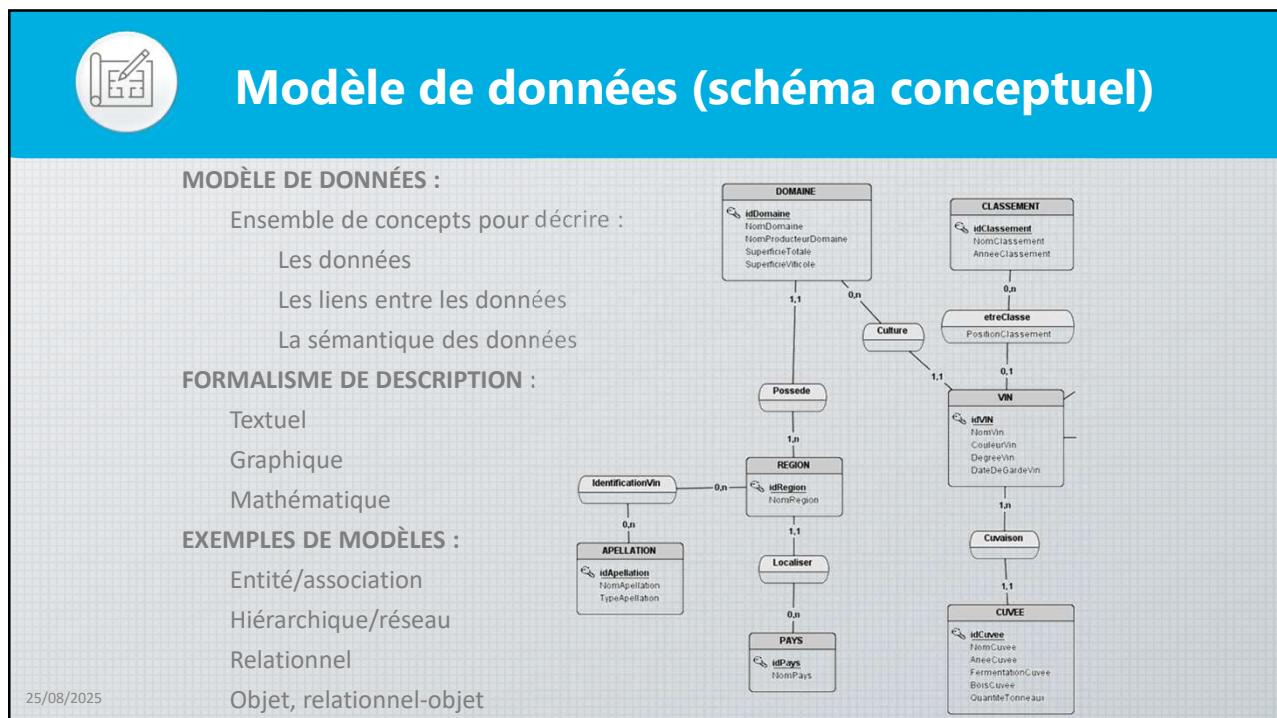
25/08/2025

46

46



47



48



## Modèle relationnel

Les relations sont représentées comme des **tables** avec **lignes** et **colonnes**

- Chaque **ligne** (enregistrement) représente une **entité**
- Chaque **colonne** (champ) représente un **attribut** de l'entité
- La valeur d'une cellule représente la **donnée**
- Les données contenues dans une table peuvent être retrouvées en utilisant la **clé primaire** (un identifiant).

Enregistrement	Clé primaire		Attribut	
	customer_id smallint	amount numeric (5,2)	count bigint	
1	10	0.99	4	
2	10	1.99	1	
3	10	2.99	7	
4	10	3.99	2	
5	10	4.99	6	
6	10	5.99	1	
7	10	6.99	1	
8	10	7.99	1	
9	10	8.99	1	

Valeur (donnée)

Colonne

25/08/2025

49



## Activité

Une petite bibliothèque souhaite suivre les emprunts de livres.

Deux solutions sont envisagées pour gérer les données des lecteurs, des livres et des emprunts :

- **Solution Excel** : tout est géré dans une seule feuille de calcul avec plusieurs colonnes (Nom du lecteur, ISBN du livre, Titre du livre, Date d'emprunt, Date de retour).

Nom lecteur	ISBN	Titre	Date emprunt	Date retour
Dupont	97812345	Le Petit Prince	01/08/2025	10/08/2025
Martin	97812346	Les Misérables	03/08/2025	15/08/2025
Dupont	97812345	Le Petit Prince	01/08/2025	10/08/2025

25/08/2025

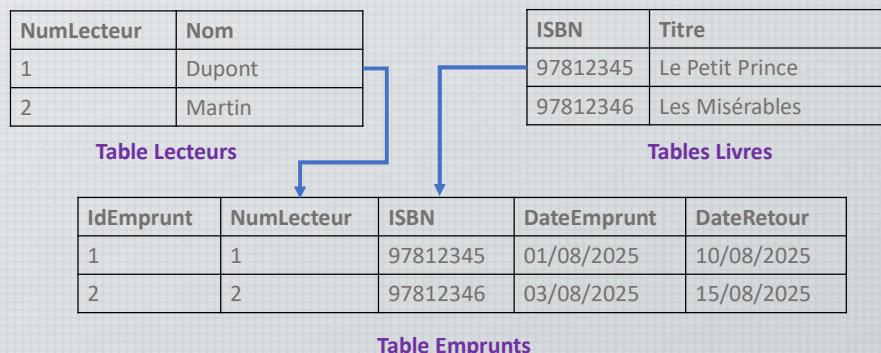
50

50



## Activité

- Solution SGBD :** les données sont réparties dans plusieurs tables reliées par des clés primaires et étrangères (ex : Lecteurs, Livres, Emprunts).



25/08/2025

51

51

## Méthode de conception et de développement de systèmes d'information informatisés

En 1977/1978, demande du "Ministère de l'Industrie" :

Constitution d'un groupe de travail pour l'élaboration d'une méthode élaborer unifiée pour la conception de systèmes d'information, avec pour premier objectif de mettre cette méthode en œuvre dans les projets de l'Administration et d'inciter les grandes entreprises à y adhérer de conception des systèmes d'information

- Équipe de J.-L. Le Moigne (Univ. d'Aix | Marseille)
- CTI (Centre Technique d'Informatique)
- CETE (Centre d'Études Techniques de l'équipement)

→ Méthode MERISE (1979) :

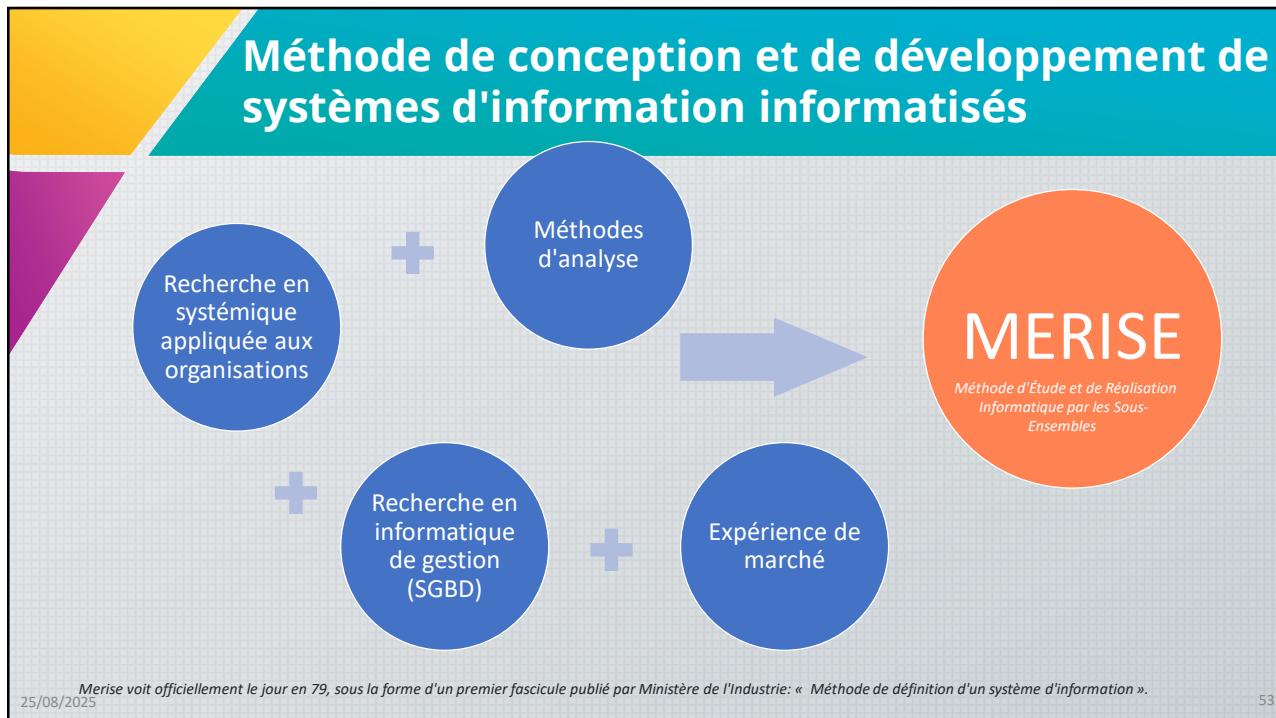
- conception du S.I. par étapes validées
- séparation des données et des traitements
- vérifier la concordance entre données et traitements
- vérifier que toutes les données nécessaires aux traitements sont présentes
- vérifier qu'il n'y a pas de données superflues

→ méthode MERISE 2<sup>ème</sup> génération en 1992

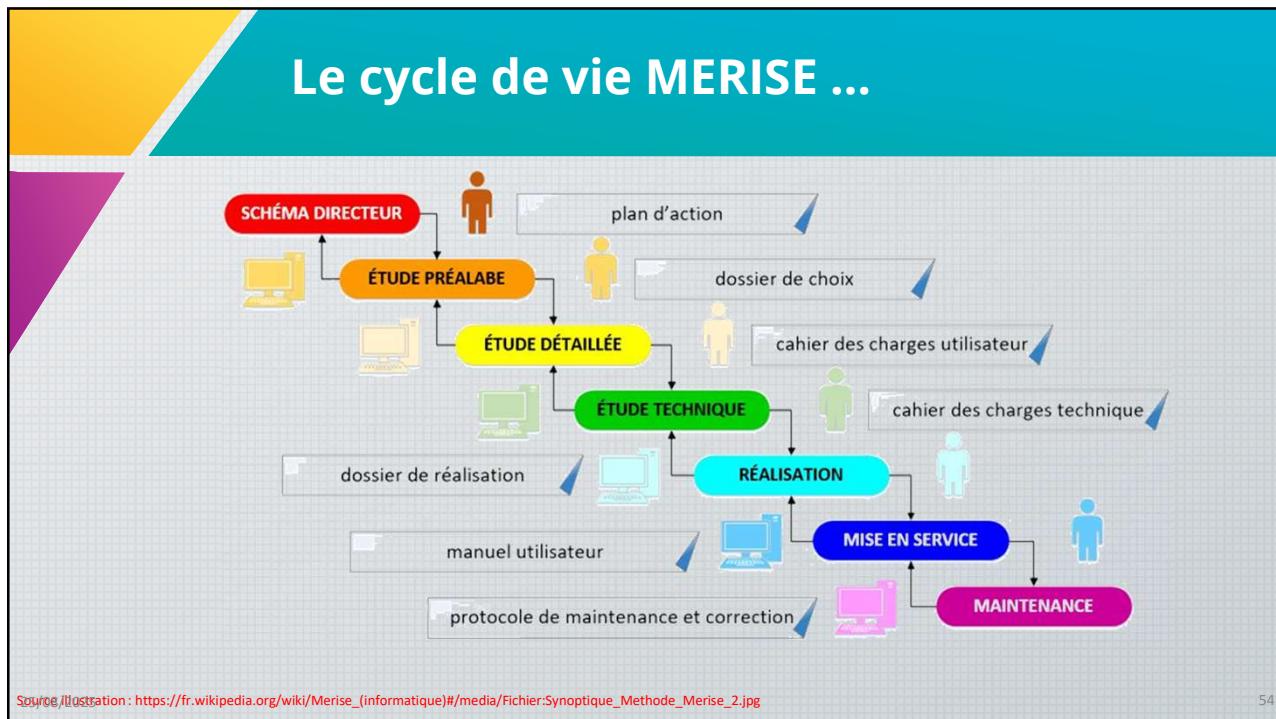
25/08/2025

52

52



53



## Méthode de conception et de développement de systèmes d'information informatisés

La démarche de développement proposée par MERISE s'inscrit dans trois dimensions :

- **Le cycle de vie** : fait référence aux différentes phases du processus de développement d'un système d'information. MERISE divise le processus de développement en trois phases principales :
  - Étude (ou Analyse) : Comprendre les besoins et les objectifs du système.
  - Spécification (ou Organisation) : Définir l'organisation des traitements et des données.
  - Conception (ou Logique) : Définir la structure technique du système.

## Méthode de conception et de développement de systèmes d'information informatisés

- **Le cycle de décision (ou Modèles Conceptuels)** : est lié à la prise de décision dans la première phase du processus de développement (Étude ou Analyse). Pendant cette phase, on utilise des modèles conceptuels, tels que le Modèle Conceptuel des Données (MCD), pour représenter les concepts et les relations du domaine métier. Les décisions prises à ce stade influencent les phases ultérieures du développement.
- **Le cycle d'abstraction** : c'est l'organisation structurelle des données et des traitements.

## Les modèles du système ...

	Données	Traitements	
Niveau conceptuel	<b>MCD</b> Modèle Conceptuel de Données	<b>MCT</b> Modèle Conceptuel de Traitements	<b>SIO</b> <i>Système d'Information Organisationnel</i>
Niveau organisationnel	<b>MOD</b> Modèle Organisationnel de Données	<b>MOT</b> Modèle Organisationnel de Traitements	
Niveau logique	<b>MLD</b> Modèle Logique de Données	<b>MLT</b> Modèle Logique de Traitements	<b>SII</b> <i>Système d'Information Informatisé</i>
Niveau physique	<b>MPD</b> Modèle Physique de Données	<b>MPT</b> Modèle Physique de Traitements	

Exprime l'activité organisée de n'importe quel système, notamment de tout système humain. Est tourné vers les utilisateurs

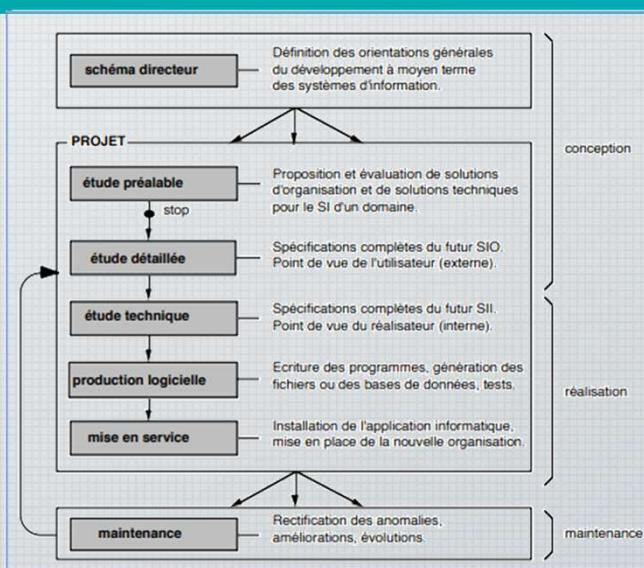
C'est la partie informatisée du SIO à laquelle les utilisateurs ont accès. Il est constitué par le ou les logiciels (les applications) utilisés dans l'entreprise, par les fichiers, la ou les bases de données, le ou les SGBD.

25/08/2025

57

57

## Le cycle de vie MERISE ...



25/08/2025

58

58

## Modèle Conceptuel de Données

Description semi-formelle des données consommées et produites par un système d'information. Cette description sous la forme d'un diagramme permet de révéler la structure conceptuelle des données du système, ce modèle ne rentre pas dans les détails techniques de la manière dont les données sont stockées ou gérées.

The diagram illustrates the Conceptual Data Model (CDM) as a central hub connected to four main components:

- Les entités qui représentent les concepts du monde réel** (Orange circle)
- Les propriétés des entités et des relations** (Purple circle)
- Les relations entre les entités** (Blue circle)
- Les conditions ou règles qui régissent la manière dont les données sont traitées** (Yellow circle)

On the left, there is a small circular diagram representing a CDM, showing entities like Entité 1, Entité 2, and Entité 3, and their associations.

59

## Les étapes pour la modélisation

The diagram shows a circular process with five sequential steps:

- Comprendre le Contexte et Définir le Périmètre** (Yellow icon)
- Identifier des règles de gestion/métier** (Orange icon)
- Créer le modèle conceptuel de données (MCD)** (Purple icon)
- Définir le dictionnaire de données** (Dark Blue icon)
- Normalisation.** (Teal icon)

In the center of the circle is a icon representing the Conceptual Data Model (CDM).

25/08/2025

60

60

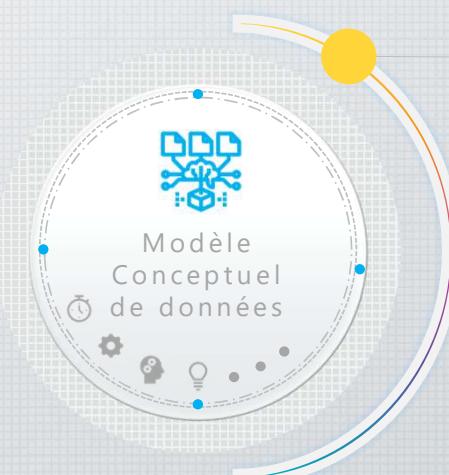
## Exemple

**Contexte :** La bibliothèque municipale souhaite informatiser la gestion de ses livres et des emprunts effectués par ses adhérents.

Elle doit pouvoir :

- ✓ Enregistrer les livres (avec leur ISBN, titre, éditeur, etc.).
- ✓ Gérer plusieurs exemplaires physiques d'un même livre.
- ✓ Suivre les auteurs des livres.
- ✓ Permettre à des adhérents d'emprunter des exemplaires avec des dates de retour prévues.
- ✓ Calculer des amendes en cas de retard.

## Les étapes pour la modélisation



### Comprendre le Contexte et Définir le Périmètre

- ✓ Lire attentivement et plusieurs fois l'expression des besoins
- ✓ Identifier les termes-clés (entités potentielles)
- ✓ Poser des questions à partir du texte
- ✓ Identifier les relations entre les termes clés
- ✓ Identifier des éléments implicites

## Exemple

**Contexte :** La bibliothèque municipale souhaite informatiser la gestion de ses livres et des emprunts effectués par ses adhérents.

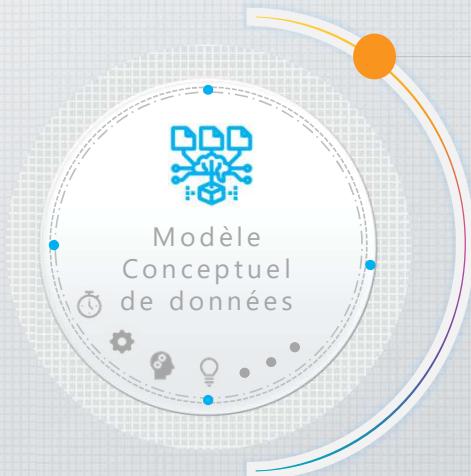
Elle doit pouvoir :

- ✓ Enregistrer les **livres** (avec leur **ISBN**, **titre**, **éditeur**, etc.).
- ✓ Gérer plusieurs **exemplaires** physiques d'un même livre.
- ✓ Suivre les **auteurs** des livres.
- ✓ Permettre à des **adhérents** d'**emprunter** des exemplaires avec des dates de retour prévues.
- ✓ Calculer des amendes en cas de retard.

## Compréhension du contexte

Terme clé	Signification	Propriétés	Relationné avec
LIVRE	Représente une œuvre ou un titre de livre	titre, isbn, éditeur, année de publication, genre	AUTEUR EXEMPLAIRE
AUTEUR	Représente une personne ayant écrit un ou plusieurs livres	nom, prénom	LIVRE
EXEMPLAIRE	Représente une copie physique d'un livre	date d'acquisition, état (neuf, usé, endommagé), disponibilité (disponible, en prêt, en réparation), prix d'achat	LIVRE, EMPRUNTER
ADHERENT	Représente un membre de la bibliothèque	nom, prénom, date d'adhésion	EMPRUNTER

## Les étapes pour la modélisation



### Identifier des règles de gestion/métier

Une **règle de gestion** est une condition ou une contrainte qui régit la manière dont les données doivent être créées, manipulées, et maintenues dans un système d'information. Les règles assurent que la structure des données reflète fidèlement le fonctionnement réel de l'organisation,

**Relever les termes clés** : Analysez les termes utilisés dans l'énoncé pour en déduire les concepts principaux et leurs relations.

#### Identifier des relations implicites

**Identifiez les quantificateurs** : Des expressions comme "chaque", "plusieurs", "au moins" ou "au maximum" suggèrent souvent des cardinalités entre les entités.

25/08/2025

65

65

## Comment les règles de gestion interviennent dans le MCD ?

### Rôle dans la modélisation de données

- **Identification des entités et des attributs** : Les règles métier aident à déterminer les entités qui doivent être représentées dans le modèle conceptuel. Elles aident également à préciser les attributs nécessaires pour chaque entité
  - **Exemple de règle** : *Chaque adhérent possède une carte d'adhérent avec son nom et son prénom.*  
→ Cela suggère une entité ADHERENT avec un attribut identifiant nom\_adherent.
- **Relations entre entités et définition des cardinalités** : Elles définissent les relations entre différentes entités.
  - **Exemple de règle** : *Un exemplaire peut faire l'objet de plusieurs emprunts successifs au cours du temps, mais à un instant donné, un exemplaire ne peut être emprunté que par un seul adhérent.*  
→ Une association EMPRUNTER entre ADHERENT et EXEMPLAIRE.
- **Contraintes fonctionnelles** : Les règles métier précisent des contraintes qui doivent être respectées dans le modèle pour garantir la cohérence des données.
  - **Exemple de règle** : *Un EXEMPLAIRE ne peut pas être emprunté s'il est déjà en prêt.*  
→ Cela peut impliquer la modélisation d'un attribut état dans EXEMPLAIRE, ou une contrainte d'intégrité sur les données.

25/08/2025

66

66

## Comment les règles de gestion interviennent dans le MCD ?

### Rôle dans la modélisation de données

- **Définir les identifiants des entités:** permettent de préciser les attributs qui identifient chaque entité
  - **Exemple de règle :** Chaque adhérent possède une carte d'adhérent avec un numéro unique.  
→ Cela suggère une entité **ADHERENT** avec un attribut identifiant **num\_adherent**.
- **Clarifier les dépendances fonctionnelles:** Elles définissent les relations entre différentes entités.
  - **Exemple de règle :** La date de retour prévue d'un emprunt dépend de la date de début d'emprunt et de la durée de prêt du type de livre.  
→ Un attribut **date\_retour** n'est pas indépendant, sa valeur peut être dérivée, ce qui permet d'éviter la redondance.

25/08/2025

67

67

## Règles de gestion/métier « Gestion de bibliothèque »

### Exemples :

- ✓ **RG1** : Un livre peut avoir un ou plusieurs auteurs.
- ✓ **RG2** : Un livre peut exister en plusieurs exemplaires physiques.
- ✓ **RG3** : Un exemplaire peut être emprunté plusieurs fois dans le temps.
- ✓ **RG4** : À un instant donné, un exemplaire ne peut être emprunté que par un seul adhérent.
- ✓ **RG5** : Un adhérent peut emprunter plusieurs exemplaires (max 5) simultanément.
- ✓ **RG6** : Chaque emprunt est défini par une date de départ, une date de retour prévue et une date de retour effective.

25/08/2025

68

68

## Les étapes pour la modélisation



### Créer le modèle conceptuel de données (MCD)

À partir des informations extraites de la **compréhension du contexte et des règles de gestion**, on construit un diagramme **Entité-Association (EA)** en définissant :

- les entités (termes clés),
- les relations (les verbes ou les actions)
- les cardinalités (règles de gestion)
- es attributs (les propriétés)

25/08/2025

69

69

## Les éléments du modèle



### Entités

Représentation d'une population d'individus ayant une sémantique et des caractéristiques communes.

Une entité représente un objet du monde réel ou conceptuel qui est pertinent pour le système d'information. Ce peut être une personne, une chose, un événement ou une notion abstraite. Une entité est généralement définie par un **nom et des propriétés spécifiques**.

25/08/2025

70

70

## Entité

**Exemple :** entités identifiées lors de la compréhension du contexte

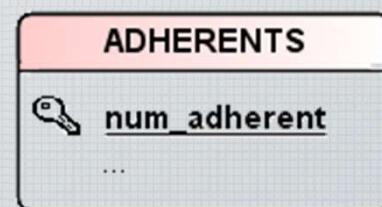


Chaque individu de la population doit être **identifiable de manière unique**

## Entité

### Règles relatives aux entités

- Une entité possède au moins une propriété : **son identifiant**
  - Permet de garantir que toutes les occurrences d'une entité sont distinctes.
  - Sa valeur est non variable dans le temps.
  - Peut éventuellement formé de plusieurs attributs.



## Les éléments du modèle



### Associations

Une **association** représente une relation entre deux ou plusieurs entités.

Elle permet de décrire comment les entités interagissent entre elles dans le système d'information et modélise les liens logiques ou fonctionnels entre les différentes entités

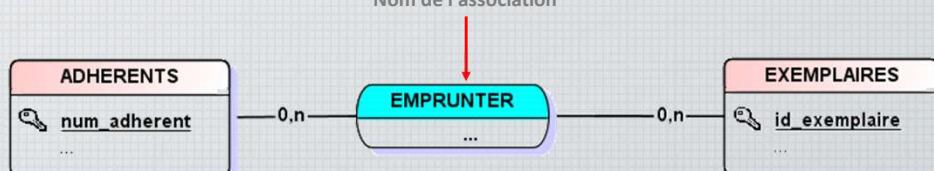
25/08/2025

73

73

## Association

Exemple :



Règles de gestion implicites :

**ADHERENT → EXEMPLAIRES :**

- un **adhérent** peut **emprunter** aucun ou plusieurs **exemplaires**.
- un **exemplaire** est **emprunté** par aucun ou plusieurs **adhérents**.

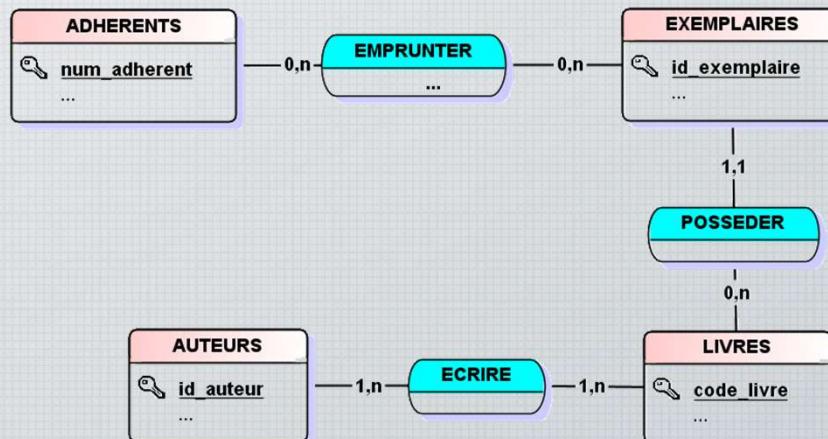
25/08/2025

74

74

## Association

Lien entre plusieurs entités



25/08/2025

75

75

## Associations Particulières

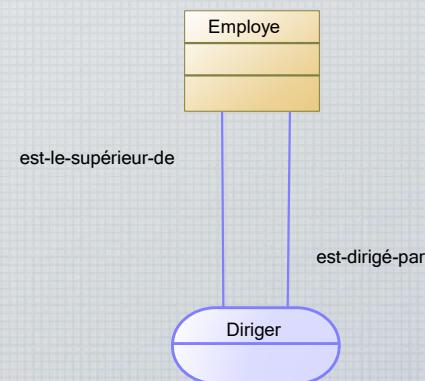
### Réflexive

- L'association lie deux individus de la même population
- Des individus de l'entité Employe sont liés par l'association Diriger.
- Chaque individu joue un rôle différent
  - Un employé *est-dirigé-par* un autre employé
  - Un employé *est-le-supérieur-des* autres employés

25/08/2025

76

76



## Exercice

### Élaborer un diagramme EA pour modéliser

L'association de paternité

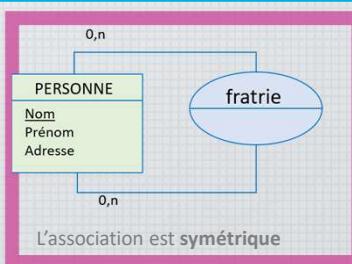
L'association de fraternité

25/08/2025

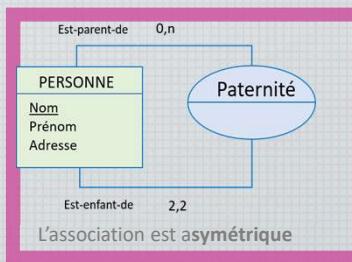
77

77

## Exercice (corrigé)



Le rôle de chaque participant de l'association est le même



Il est nécessaire de préciser le rôle de chaque participant

25/08/2025

78

78

## Les éléments du modèle



**Attributs**

Est une **caractéristique descriptive** d'une **entité** ou d'une **association**.

Il représente une propriété spécifique ou un élément d'information qui permet de décrire une **entité** ou une **association** de manière détaillée.

Chaque attribut prend **une valeur** pour chaque instance de l'entité ou de l'association.

Les attributs sont stockés par l'élément (entité ou association)

25/08/2025 79

79

## Attributs

### Règles concernant les propriétés

- Chaque propriété appartient à un **domaine** qui définit l'ensemble des valeurs possibles qui peuvent être choisies pour elle (entier, chaîne de caractères, booléen...).
- Une même propriété ne peut figurer que sur **UN SEUL** élément (entité ou association).

<b>ADHERENTS</b>	
<b>num_adherent</b>	Varchar (20)
<b>nom_adherent</b>	Varchar (50)
<b>prenom_adherent</b>	Varchar (50)
<b>date_adhesion</b>	Date

09-09-2025

80



## Attributs

### Règles concernant les propriétés

- Chacune des propriétés d'une entité doit caractériser **tout** individu de la **même manière**.
  - **Exemple :** Dans l'entreprise, on gère des marchandises fraîches et des marchandises non périssables.
  - Soit l'entité suivante, qu'en pensez-vous ?

Article		
🔑	<b>id_Article</b>	INT
	nom_Article	VARCHAR (50)
	prixUnitaire_Article	VARCHAR (100)
	description_Article	VARCHAR (100)
	dateExpiration_Article	DATETIME

25/08/2025

81

81



## Attributs

### Règles concernant les propriétés

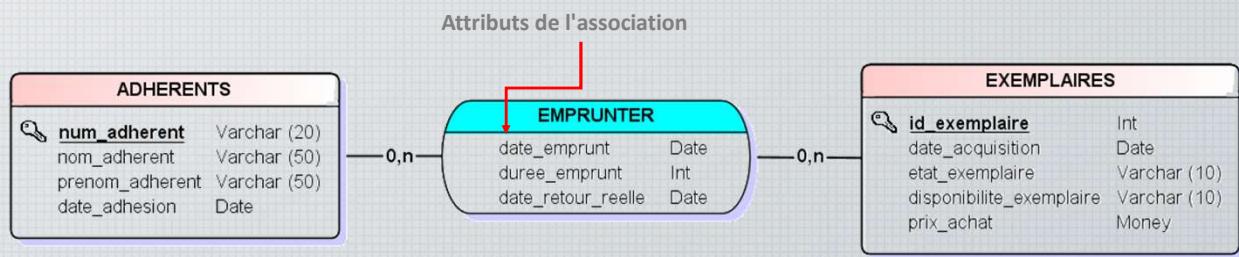
- Chaque propriété possède une valeur compatible avec son domaine
- Une propriété doit être **ELEMENTAIRE**, atomique, de telle sorte qu'on ne puisse pas la décomposer. Une propriété de n'est pas atomique lorsqu'elle :
  1. est constituée d'une agrégation de propriétés élémentaires :  
**Exemple:** Pour un compte bancaire , L'attribut **Domiciliation** est composé de :
    - Code banque,
    - Code Guichet
    - N° Compte
    - Clé
  2. admet plusieurs valeurs comme c'est le cas dans une liste :  
**Exemple :** La propriété Enfants pour une Personne est multivaluée et n'est donc pas élémentaire.
- Il n'est pas nécessaire de représenter une propriété qui peut être dérivée ou calculée

09-09-2025

82

## Attributs

Les **associations** peuvent posséder des attributs. Ces attributs décrivent des informations spécifiques à la relation entre les entités.



25/08/2025

83

83

## Attributs

### Règles concernant les propriétés des associations

- Une propriété peut être placée dans une association **uniquement lorsqu'elle dépend de toutes les entités liées par l'association.**
- Une association **N'A PAS D'IDENTIFIANT** explicite : l'association dépend des entités qu'elle relie.
- L'identifiant (implicite) **se déduit** par calcul du produit cartésien des identifiants des entités associées.

09-09-2025

84

## Les éléments du modèle



### Les cardinalités

Permettent de définir les **règles de participation** des entités dans une association.

Elles indiquent combien d'occurrences (instances) d'une entité peuvent être associées à une autre dans le cadre d'une relation spécifique.

Sont déterminées à partir des règles de gestion

25/08/2025

85

85

## Cardinalité

Le nombre de fois qu'une entité d'un ensemble d'entités participe à un ensemble de relations

- L'individu peut exister tout en n'étant impliqué dans aucune association
- Il peut être impliqué dans au maximum une association

- L'individu ne peut exister que s'il est impliqué dans exactement (au moins et au plus) une association

- L'individu ne peut exister que s'il est impliqué dans au moins une association

- L'individu peut exister tout en n'étant impliqué dans aucune association
- Il peut être impliqué, sans limitation, dans plusieurs associations

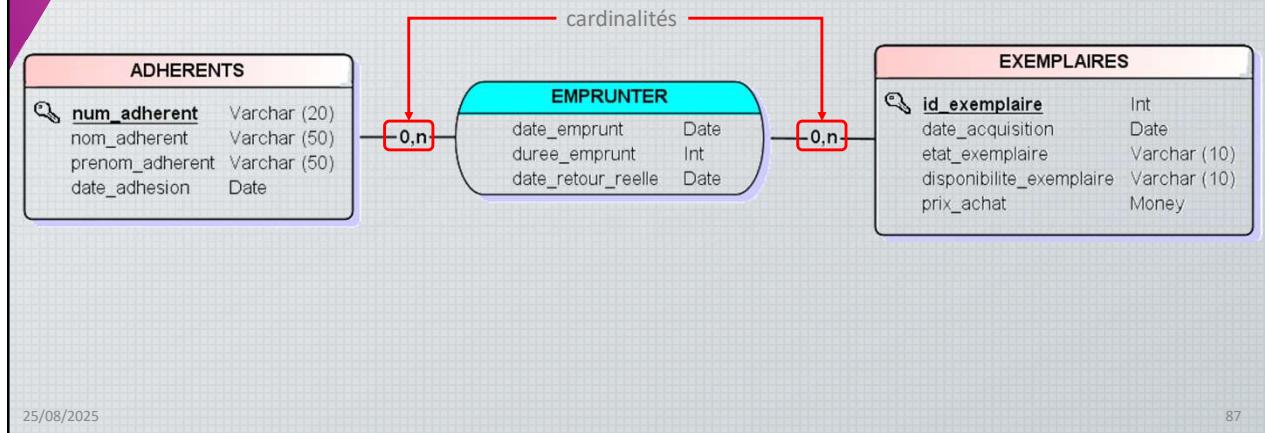
25/08/2025

86

86

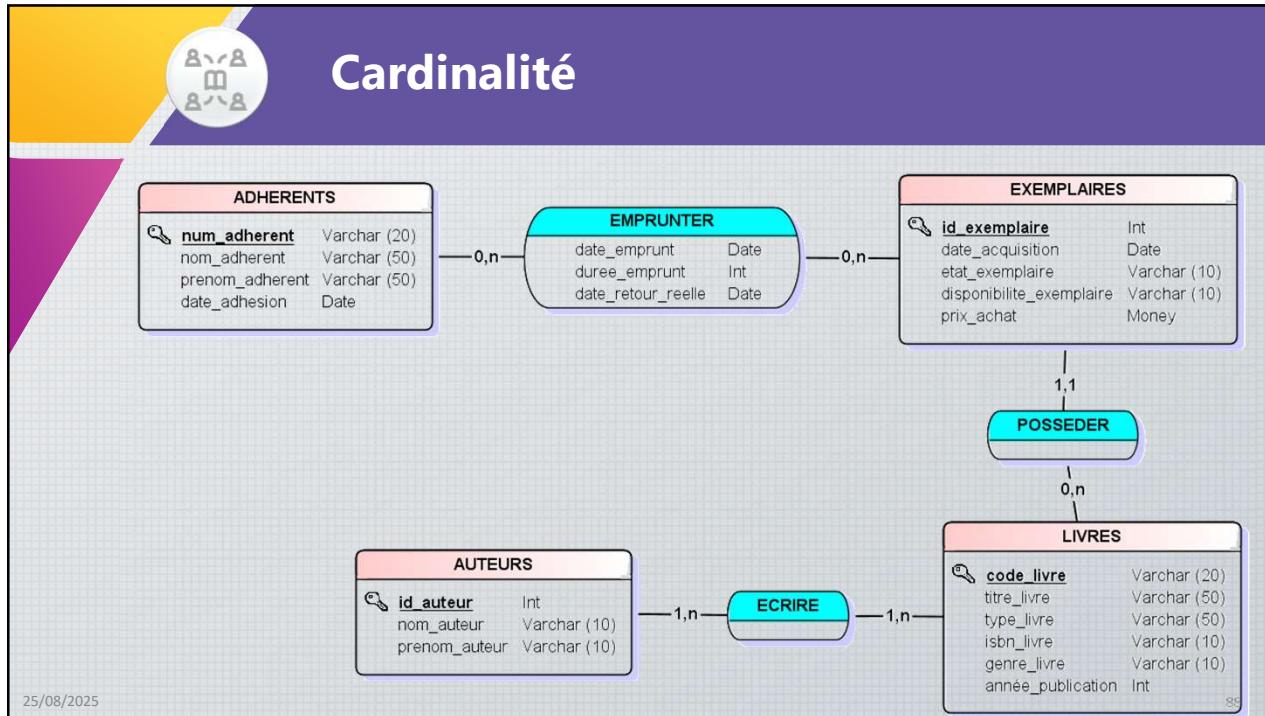
# Cardinalité

Exemple :



87

# Cardinalité

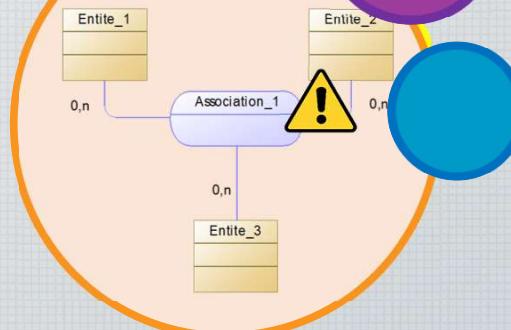


88

## Associations Particulières

Association ternaire : doit refléter une **règle métier précise** qui ne peut pas être exprimée avec des binaires

Relie trois entités



Attention aux cardinalités : il faut spécifier le sens exact des cardinalités sur chaque côté

25/08/2025

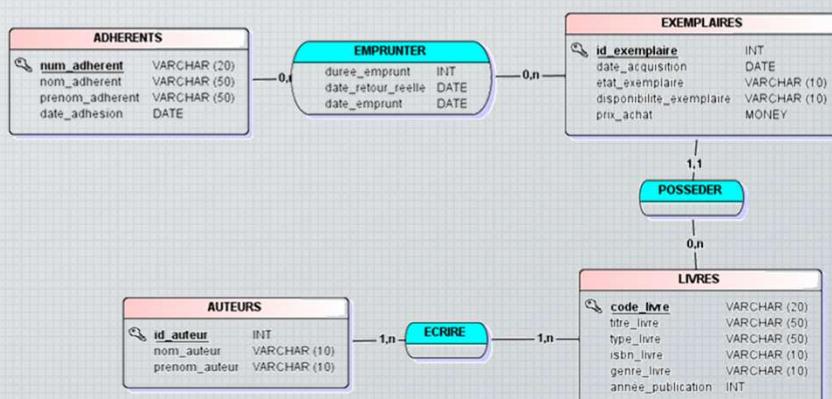
89

89

## Associations Particulières

**Exemple :** Modifiez le MCD afin de respecter la nouvelle règle de gestion énoncée ci-dessous.

**Nouvelle règle de gestion :** Un adhérent peut réservé un exemplaire d'un livre pour une période donnée auprès d'un bibliothécaire (qui valide la réservation).



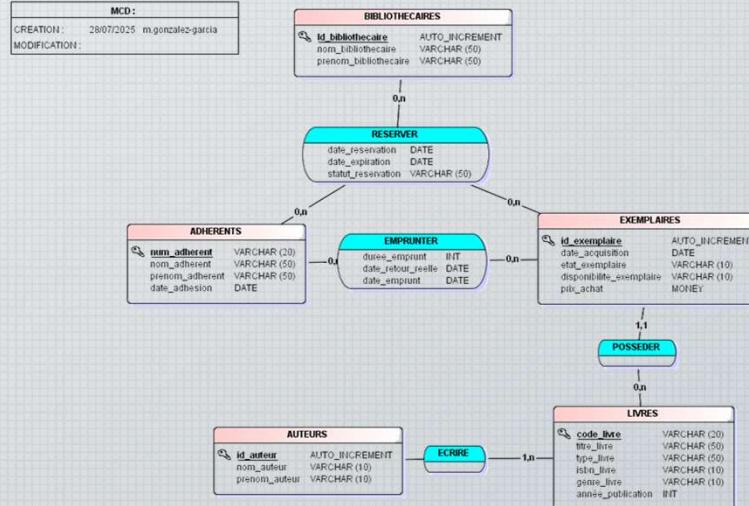
25/08/2025

90

90

## Associations Particulières

**Nouvelle règle de gestion :** Un adhérent peut résERVER un exemplaire d'un livre pour une période donnée auprès d'un bibliothécaire (qui valide la réservation).



25/08/2025

MCD modifié

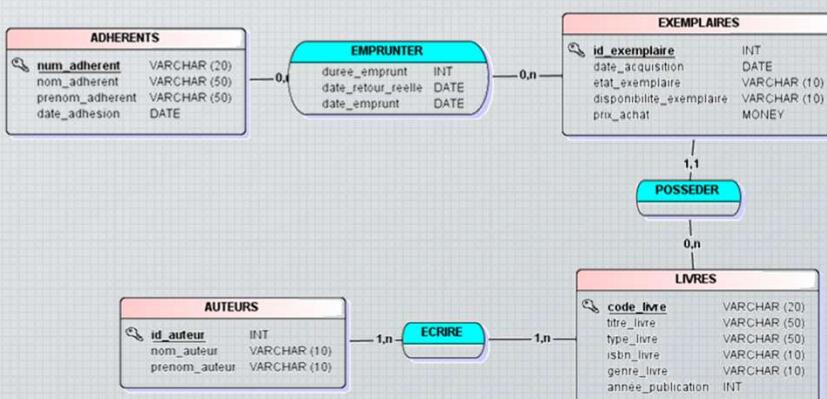
91

91

## Associations Particulières

**Exemple :** Modifiez le MCD afin de respecter la nouvelle règle de gestions énoncée ci-dessous.

**Nouvelle règle de gestion :** Un bibliothécaire attribue un exemplaire à un adhérent dans le cadre d'un emprunt.



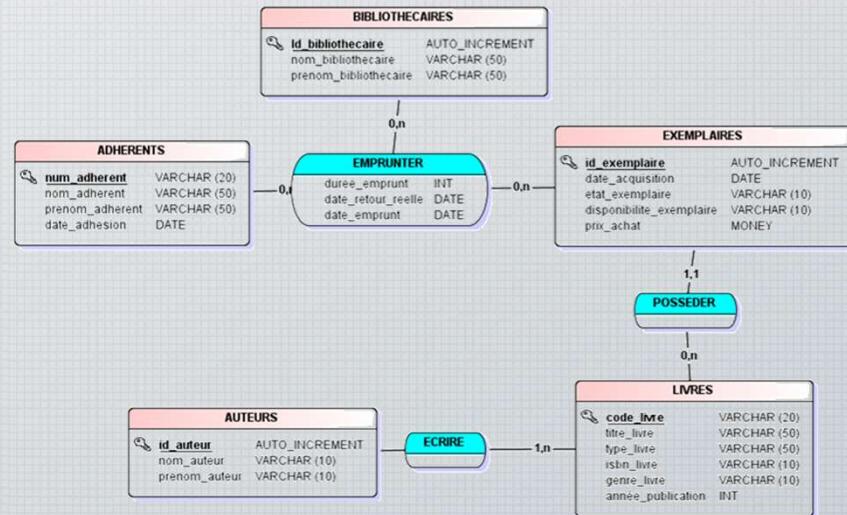
25/08/2025

92

92

# Associations Particulières

Nouvelle règle de gestion : Un bibliothécaire attribue un exemplaire à un adhérent dans le cadre d'un emprunt.



25/08/2025

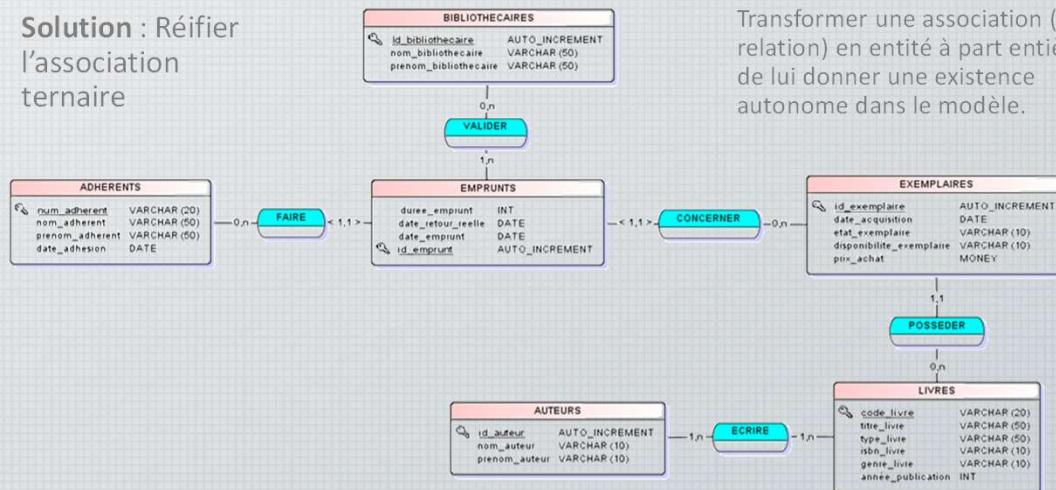
93

93

# Associations Particulières

Solution : Réifier l'association ternaire

Transformer une association (ou une relation) en entité à part entière, afin de lui donner une existence autonome dans le modèle.



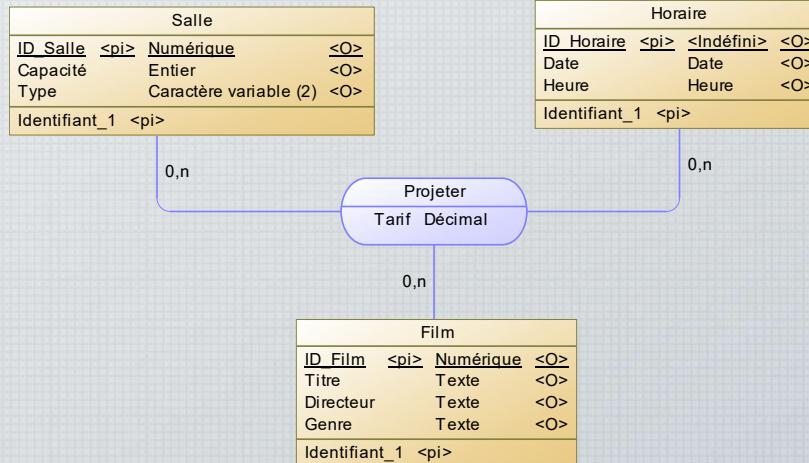
25/08/2025

94

94

## Association Ternaire

### Exemple



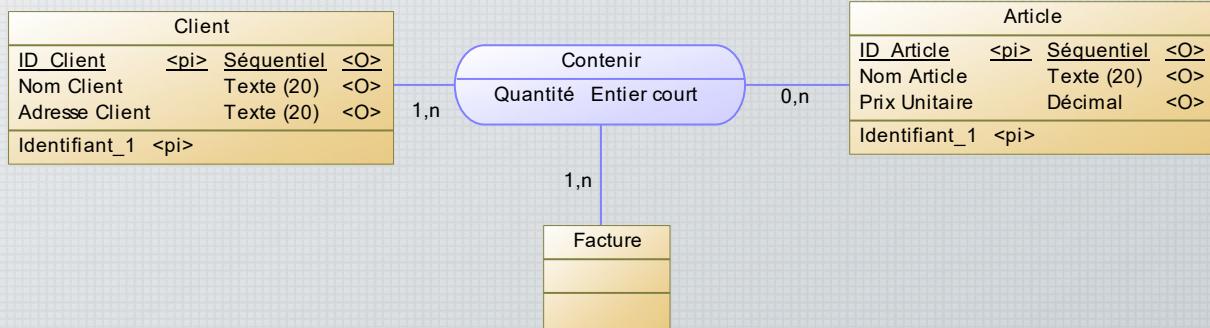
25/08/2025

95

95

## Exercice

Pour le modèle ci-joint, identifiez et corrigez les erreurs potentielles



25/08/2025

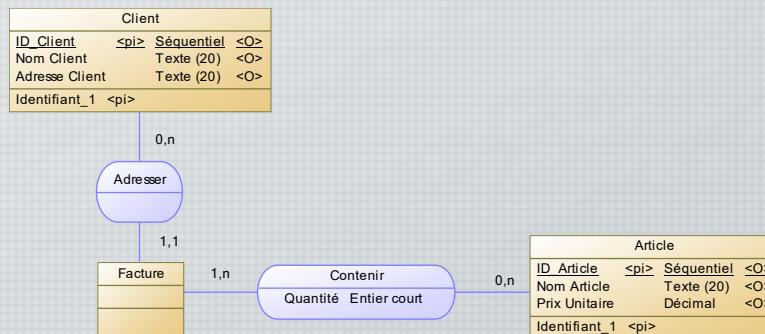
96

96

## Exercice

### Problème

- Selon le modèle une facture peut être adressée à plusieurs clients.



25/08/2025

97

### Correction

- Changer l'association ternaire par deux associations binaires
- Modifier les cardinalités



## Associations Particulières

### Réflexive

Une association réflexive (ou association unaire) est une association qui relie des occurrences d'une même entité entre elles. Elle permet de modéliser les liens de même nature qui peuvent exister entre différents objets d'un même type, comme la hiérarchie (ex : un employé encadre un autre employé) ou la symétrie (ex : deux pays sont voisins).

Les associations réflexives peuvent être :

- ✓ **Asymétriques**, lorsqu'il existe un sens ou une hiérarchie entre les deux rôles (ex : chef/subordonné),
- ✓ **Symétriques**, lorsque la relation est mutuelle sans hiérarchie (ex : jumelage, fraternité).

25/08/2025

98



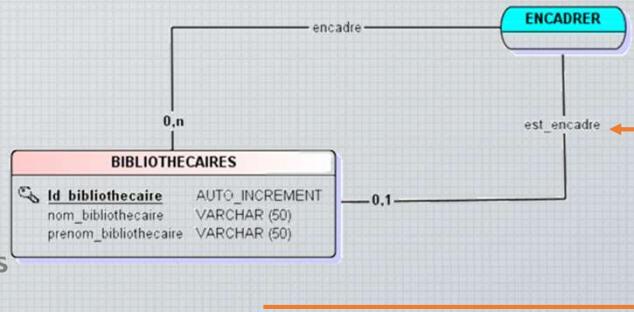
## Associations Particulières

### Exemple :

- ✓ **Asymétriques**, lorsqu'il existe un sens ou une hiérarchie entre les deux rôles (ex : chef/subordonné).

### Règle de gestion :

« Un bibliothécaire peut avoir un responsable parmi les autres bibliothécaires. Un bibliothécaire peut encadrer zéro, un ou plusieurs autres bibliothécaires. »



Utiliser la notion de *rôles* pour éliminer l'ambiguité

25/08/2025

99

99



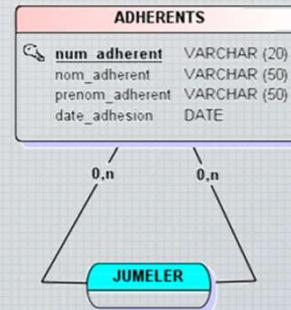
## Associations Particulières

### Réflexive

- ✓ **Symétriques**, lorsque la relation est mutuelle sans hiérarchie (ex : jumelage, fraternité).

### Règle de gestion :

« Deux adhérents peuvent se déclarer "jumelés" pour profiter d'avantages ou partager des emprunts. »



25/08/2025

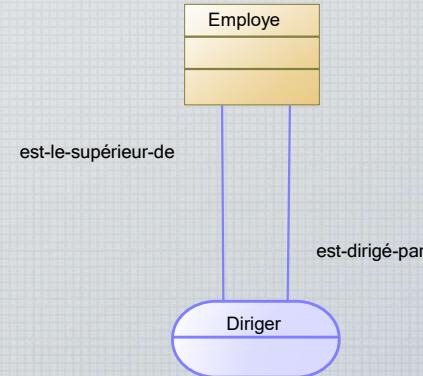
100

100

## Associations Particulières

### Réflexive

- L'association lie deux individus de la même population
- Des individus de l'entité **Employe** sont liés par l'association **Diriger**.
- Chaque individu joue un rôle différent
  - Un employé *est-dirigé-par* un autre employé
  - Un employé *est-le-supérieur-des* autres employés



25/08/2025

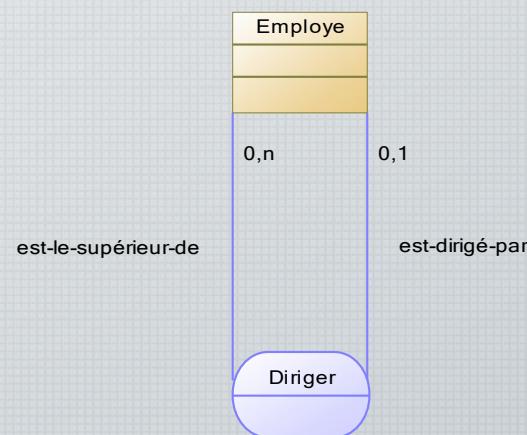
101

101

## Cardinalité

### Exemple :

- Un employé *est-dirigé-par* un autre employé à l'**exception** du directeur général (0,1)
- Un employé *est-le-supérieur-des* autres employés à l'**exception** des employés en bas de la hiérarchie (0,n)



25/08/2025

102

102



## Modélisation : une étude de cas (exercice)

**Contexte :** L'entreprise "MIG" vend des articles à des clients et achète ces articles auprès de fournisseurs. L'entreprise souhaite un système d'information pour gérer les clients, les articles, les fournisseurs, et les factures qui enregistrent les ventes d'articles aux clients.

**Exigences :**

- ✓ L'entreprise souhaite suivre ses clients, en enregistrant pour chacun leur nom, adresse et email.
- ✓ L'entreprise vend plusieurs articles, chaque article ayant un nom, une description et un prix unitaire.
- ✓ Les articles vendus sont achetés auprès de fournisseurs, pour lesquels l'entreprise enregistre le nom et le pays d'origine.
- ✓ Une facture est générée pour chaque achat réalisé par un client, incluant la date d'achat et la liste des articles achetés.
- ✓ Chaque article peut être vendu à plusieurs clients, et un client peut acheter plusieurs articles.



## Compréhension du contexte

**Contexte :** L'entreprise "MIG" vend des **articles** à des clients et achète ces articles auprès de **fournisseurs**. L'entreprise souhaite un système d'information pour gérer les **clients**, les **articles**, les **fournisseurs**, et les **factures** qui enregistrent les ventes d'articles aux clients.

**Exigences :**

- ✓ L'entreprise souhaite suivre ses **clients**, en enregistrant pour chacun leur **nom**, **adresse** et **email**.
- ✓ Les clients **commandent** plusieurs **articles**, chaque article ayant un **nom**, une **description** et un **prix unitaire**. L'entreprise souhaite également s'assurer que les commandes sont **livrées** dans les temps. Il sera nécessaire de distinguer entre le moment où une commande est passée et celui où elle est livrée.
- ✓ Les articles vendus sont **achetés** auprès de **fournisseurs**, pour lesquels l'entreprise enregistre le **nom** et le **pays d'origine**.
- ✓ Une **facture** est **générée** pour chaque **commande** passée par un client, incluant la **date d'achat**, la **date de paiement**, le **numéro de la commande** et la **liste des articles achetés**.
- ✓ Chaque **article** peut être **vendu** à plusieurs **clients**, et un client peut **acheter** plusieurs articles.

## Compréhension du contexte

Terme clé	Signification	Propriétés	Relationné avec
Client	Représente une personne qui achète des articles	Nom, Adresse, Email	Commande
Article	Un produit que la boutique vend	Nom, Description, Prix	Client Fournisseur Facture
Fournisseur	L'entreprise qui fournit les articles à la boutique	Nom, Pays, articles fournis	Article
Facture	Le document qui enregistre les paiements des commandes	Date, total à payer, date d'émission, date de paiement, commande correspondante	Commande
Commande	Représente les achats réalisés par un client	Date d'achat, date de livraison, articles achetés, client qui passe la commande	Facture Article Client

25/08/2025

105

105

## Les règles de gestion

- ✓ **RG 1** : Un client peut passer plusieurs commandes, mais une commande est passée par un seul client.
- ✓ **Règle implicite 1** : Une commande ne peut exister sans client.
- ✓ **RG2** : Une commande peut contenir plusieurs articles, et un article peut apparaître dans plusieurs commandes.
- ✓ **Règle implicite 2** : Une commande doit contenir au moins un article pour être valide.
- ✓ **RG3** : Une facture est générée pour chaque commande.
- ✓ **Règle implicite** : Une commande doit être validée avant que la facture ne soit émise.
- ✓ **RG4** : Un fournisseur peut fournir plusieurs articles, mais une commande n'est pas directement liée à un fournisseur (les fournisseurs sont liés aux articles).

25/08/2025

106

106

## Entité

**Exemple :** entités identifiées suite à l'analyse de l'expression des besoins



Chaque individu de la population doit être **identifiable de manière unique**

25/08/2025

107

107

## Entité

### Règles relatives aux entités

- Une entité possède au moins une propriété : **son identifiant**
  - Permet de garantir que toutes les occurrences d'une entité sont distinctes.
  - Il est non variable dans le temps.
  - Peut éventuellement formé de plusieurs attributs.



25/08/2025

108

108

## Les éléments du modèle



25/08/2025

### Associations

Une **association** représente une relation entre deux ou plusieurs entités.

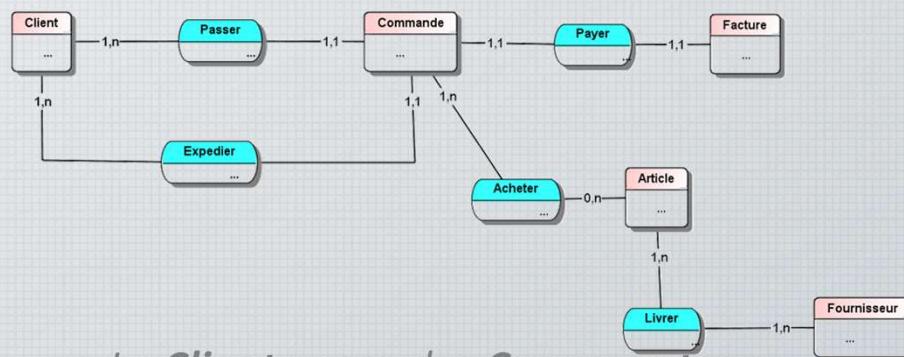
Elle permet de décrire comment les entités interagissent entre elles dans le système d'information et modélise les liens logiques ou fonctionnels entre les différentes entités

109

109

## Association

Lien entre plusieurs entités



« *Le Client passe des Commandes* »

« *Le Fournisseur livre des Articles* »

25/08/2025

110

110

## Les éléments du modèle

Modèle Conceptuel de données

Attributs

Est une **caractéristique descriptive** d'une **entité** ou d'une **association**.

Il représente une propriété spécifique ou un élément d'information qui permet de décrire une **entité** ou une **association** de manière détaillée.

Chaque attribut prend **une valeur** pour chaque instance de l'entité ou de l'association.

Les attributs sont stockés par l'élément (entité ou association)

25/08/2025 111

111

## Attributs

### Règles concernant les propriétés

- Chaque propriété appartient à un **domaine** qui définit l'ensemble des valeurs possibles qui peuvent être choisies pour elle (entier, chaîne de caractères, booléen...).
- Une même propriété ne peut figurer que sur **UN SEUL** élément (entité ou association).

Article	
	<b>id_Article</b> INT
	nom_Article VARCHAR (50)
	prixUnitaire_Article VARCHAR (100)
	description_Article VARCHAR (100)

09-09-2025

112



## Attributs

### Règles concernant les propriétés

- Chacune des propriétés d'une entité doit caractériser **tout** individu de la **même manière**.
  - **Exemple :** Dans l'entreprise, on gère des marchandises fraîches et des marchandises non périssables.
  - Soit l'entité suivante, qu'en pensez-vous ?

Article		
🔑	<b>id_Article</b>	INT
	nom_Article	VARCHAR (50)
	prixUnitaire_Article	VARCHAR (100)
	description_Article	VARCHAR (100)
	dateExpiration_Article	DATETIME

25/08/2025

113

113



## Attributs

### Règles concernant les propriétés

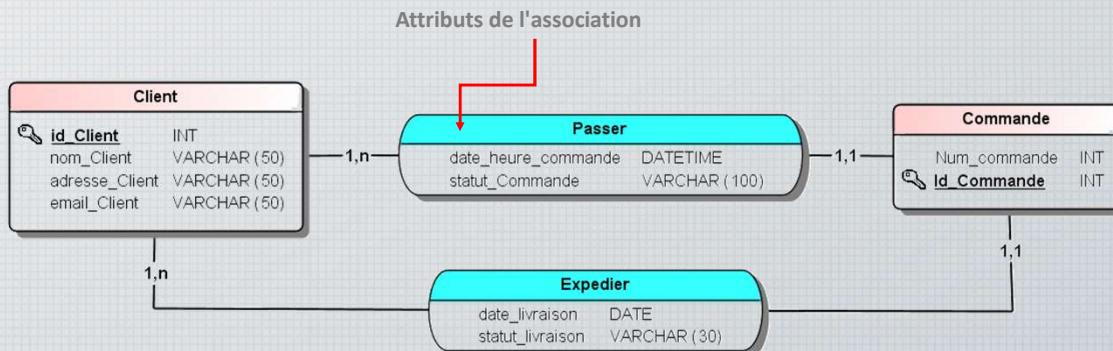
- Chaque propriété possède une valeur compatible avec son domaine
- Une propriété doit être **ELEMENTAIRE**, atomique, de telle sorte qu'on ne puisse pas la décomposer. Une propriété de **n'est pas atomique** lorsqu'elle :
  1. est constituée d'une agrégation de propriétés élémentaires (**attribut composite**) :
    - **Exemple :** Pour une adresse postale, l'attribut **Adresse** est composé de :
    - Rue : nom et numéro,
    - Ville
    - Code postal
  2. admet plusieurs valeurs comme c'est le cas dans une liste (**attribut multivalué**) :
    - **Exemple :** La propriété numéro de téléphone peut être multivalué (une personne peut avoir plusieurs numéros de téléphone)
- Il n'est pas nécessaire de représenter une propriété qui peut être **dérivée ou calculée**

09-09-2025

114

## Attributs

Les **associations** peuvent posséder des attributs. Ces attributs décrivent des informations spécifiques à la relation entre les entités.



25/08/2025

115

115

## Cardinalité

Exemple :



Règles de gestion implicites :

**Client → Commande :**

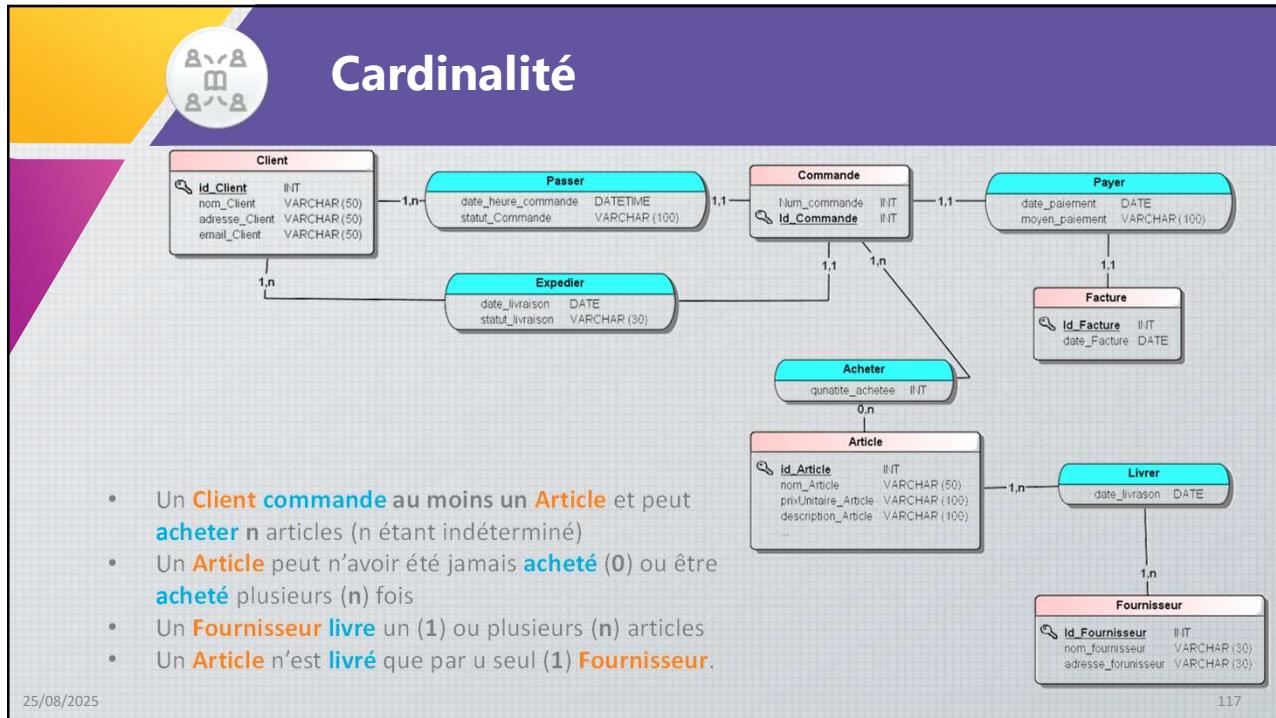
- un **client** peut **passer** plusieurs **commandes** (cardinalité 1..n).
- une **commande** est **passée** par un seul **client** (cardinalité 1..1).

Règle implicite : une commande ne peut exister sans client.

25/08/2025

116

116



117

## Modélisation : les règles de gestion

Une **règle de gestion** est une condition ou une contrainte qui régit la manière dont les données doivent être créées, manipulées, et maintenues dans un système d'information. Ces règles peuvent inclure des directives sur la structure des données, les relations entre les entités, et les opérations valides sur ces données.

- Intégrité référentielle** : "Un enregistrement dans la **table Commande** ne peut être créé que si l'ID du client référencé existe dans la **table Client**."
- Contraintes d'attribut** : "Le champ **DateNaissance** dans la **table Personne** doit contenir une date postérieure à la date actuelle."
- Unicité** : "L'email d'un utilisateur dans la **table Utilisateur** doit être **unique** et ne peut pas être partagé par plusieurs utilisateurs."
- Valider les données d'entrée** : "Le champ **Quantité** dans la **table Produit** ne peut pas être inférieur à 0."
- Classification** : "Tous les produits doivent être classés dans une catégorie définie dans la **table Catégorie**."

118      118

118

## Les étapes pour la modélisation



### Définir le dictionnaire de données

Décrit les éléments de données dans le système d'information. Il fournit des informations détaillées sur chaque élément de données

25/08/2025

119

119



## Modélisation : le dictionnaire des données

### Informations du dictionnaire des données

**Nom de l'élément de données :** Le nom que l'on donne à l'élément.

**Type de données :** Le type de données (par exemple, texte, entier, date, etc.). **Description :** Une explication de ce que l'élément de données représente.

**Contraintes :** Les règles ou restrictions qui s'appliquent à l'élément de données (comme la taille maximale, la valeur minimale, etc.).

**Source :** L'origine de l'élément de données (par exemple, s'il est saisi par un utilisateur ou calculé par le système).

**Relations :** Les liens avec d'autres éléments de données, si applicable.

25/08/2025

120

120



## Modélisation : le dictionnaire des données

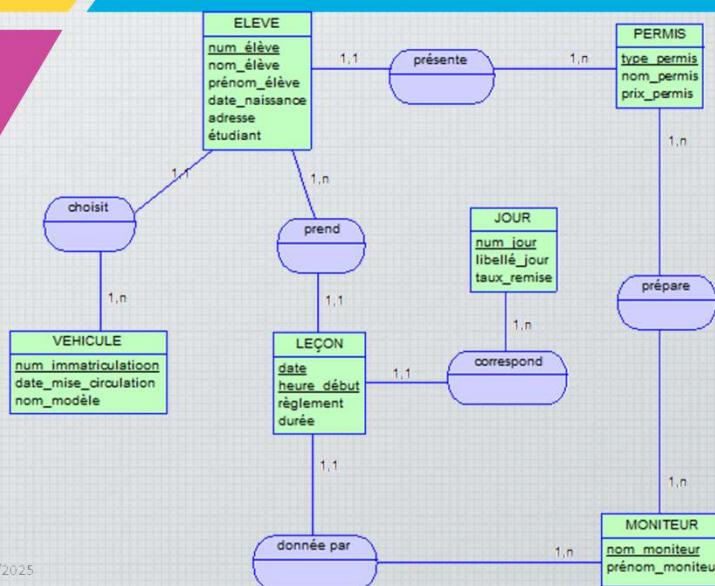
### Exemple

Nom de l'élément	Type de données	Description	Contraintes	Source
IdClient	Entier	Identifiant unique du client	Non nul	Système
NomClient	Texte	Nom du client	Non nul, max 50 caractères	Saisie utilisateur
AdresseClient	Texte	Adresse de livraison du client	Max 100 caractères	Saisie utilisateur
EmailClient	Texte	Adresse e-mail du client	Unique, non nul	Saisie utilisateur

25/08/2025

121

## Exercice : Analyser le modèle ci-dessous



- Un élève peut préparer simultanément le permis dans les deux catégories ?
- Les leçons peuvent avoir lieu avec plusieurs véhicules ?
- Il n'existe qu'un seul véhicule par modèle ?
- De quelle façon l'auto-école identifie une leçon ?
- La durée des leçons est-elle de variable?
- Un moniteur peut enseigner dans deux catégories de permis ou seulement dans l'une des deux ?

25/08/2025

122

122

## Exercice : Analyser le modèle ci-dessous

The diagram illustrates a UML Class Diagram with the following entities and their attributes:

- ELEVE**: num\_élève, nom\_élève, prénom\_élève, date\_naissance, adresse, étudiant.
- VEHICULE**: num\_immatriculation, date\_mise\_circulation, nom\_modèle.
- PERMIS**: type\_permis, nom\_permis, prix\_permis.
- JOEUR**: num\_jour, libellé\_jour, taux\_remise.
- LEÇON**: date, heure\_début, règlement, durée.
- MONITEUR**: nom\_moniteur, prénom\_moniteur.

The relationships between these entities are:

- ELEVE** has a 1,n relationship to **choisit** (which connects to **VEHICULE**), a 1,n relationship to **prend** (which connects to **LEÇON**), and a 1,n relationship to **présente** (which connects to **PERMIS**).
- VEHICULE** has a 1,n relationship to **choisit**.
- LEÇON** has a 1,1 relationship to **correspond** (which connects to **JOEUR**), a 1,1 relationship to **donnée par** (which connects to **MONITEUR**), and a 1,1 relationship to **prend**.
- JOEUR** has a 1,n relationship to **correspond**.
- PERMIS** has a 1,n relationship to **présente** and a 1,n relationship to **prépare**.
- MONITEUR** has a 1,n relationship to **donnée par**.

Log in to Poll Everywhere  
To present live activities, please log in to your Poll Everywhere account in a separate window.  
[Launch log-in window](#)

25/08/2025

123

## Exercice : Analyser le modèle ci-dessous

The diagram illustrates a UML Class Diagram with the following entities and their attributes:

- ELEVE**: num\_élève, nom\_élève, prénom\_élève, date\_naissance, adresse, étudiant.
- VEHICULE**: num\_immatriculation, date\_mise\_circulation, nom\_modèle.
- PERMIS**: type\_permis, nom\_permis, prix\_permis.
- JOEUR**: num\_jour, libellé\_jour, taux\_remise.
- LEÇON**: date, heure\_début, règlement, durée.
- MONITEUR**: nom\_moniteur, prénom\_moniteur.

The relationships between these entities are:

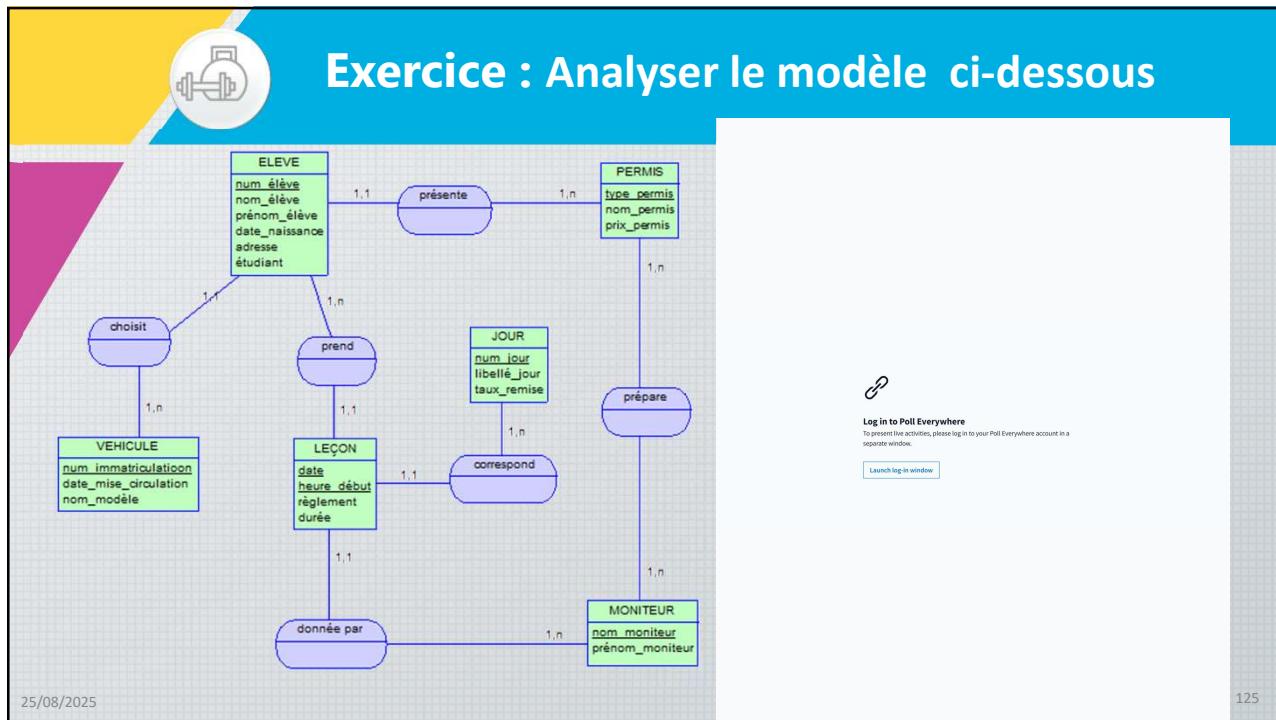
- ELEVE** has a 1,n relationship to **choisit** (which connects to **VEHICULE**), a 1,n relationship to **prend** (which connects to **LEÇON**), and a 1,n relationship to **présente** (which connects to **PERMIS**).
- VEHICULE** has a 1,n relationship to **choisit**.
- LEÇON** has a 1,1 relationship to **correspond** (which connects to **JOEUR**), a 1,1 relationship to **donnée par** (which connects to **MONITEUR**), and a 1,1 relationship to **prend**.
- JOEUR** has a 1,n relationship to **correspond**.
- PERMIS** has a 1,n relationship to **présente** and a 1,n relationship to **prépare**.
- MONITEUR** has a 1,n relationship to **donnée par**.

Log in to Poll Everywhere  
To present live activities, please log in to your Poll Everywhere account in a separate window.  
[Launch log-in window](#)

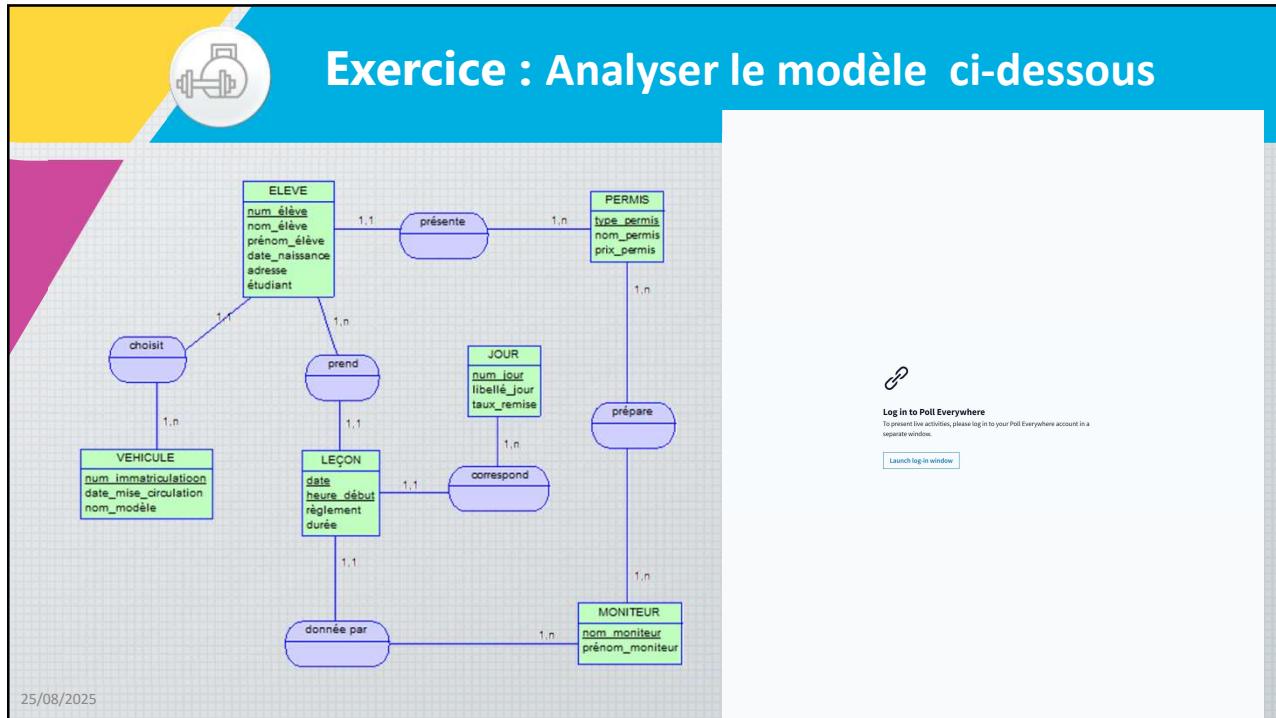
25/08/2025

124

## Exercice : Analyser le modèle ci-dessous

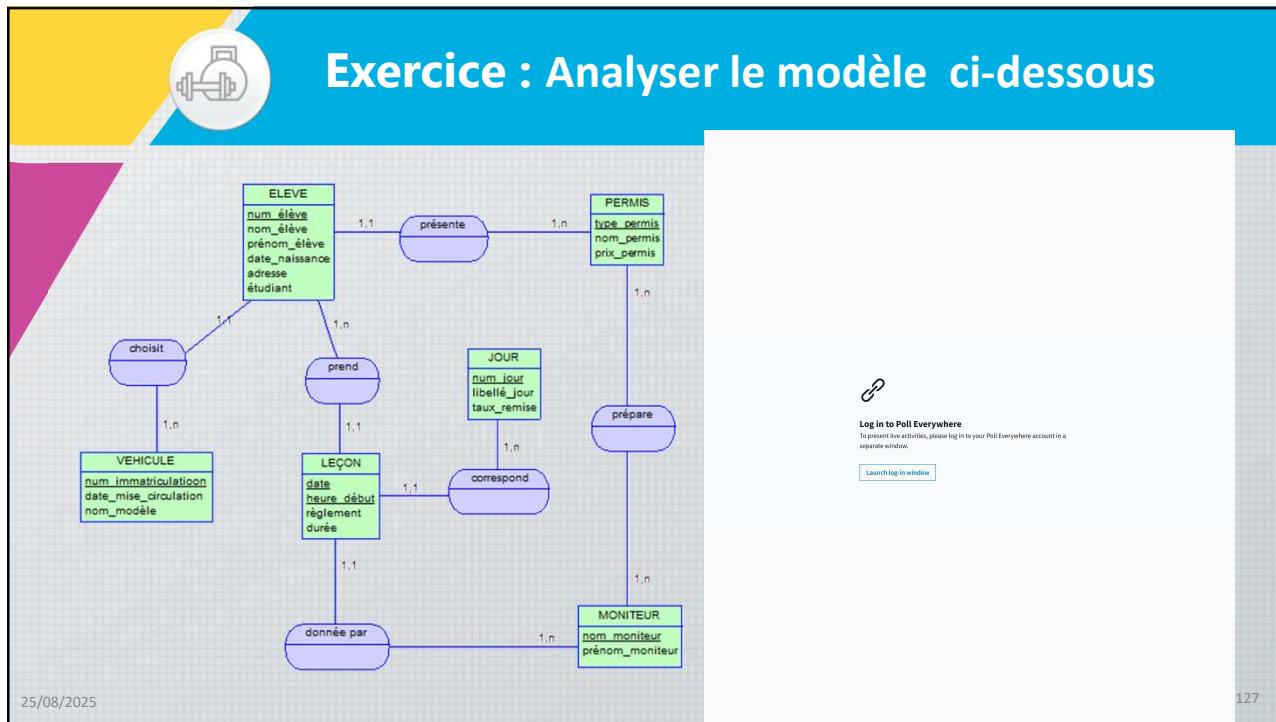


125



126

## Exercice : Analyser le modèle ci-dessous

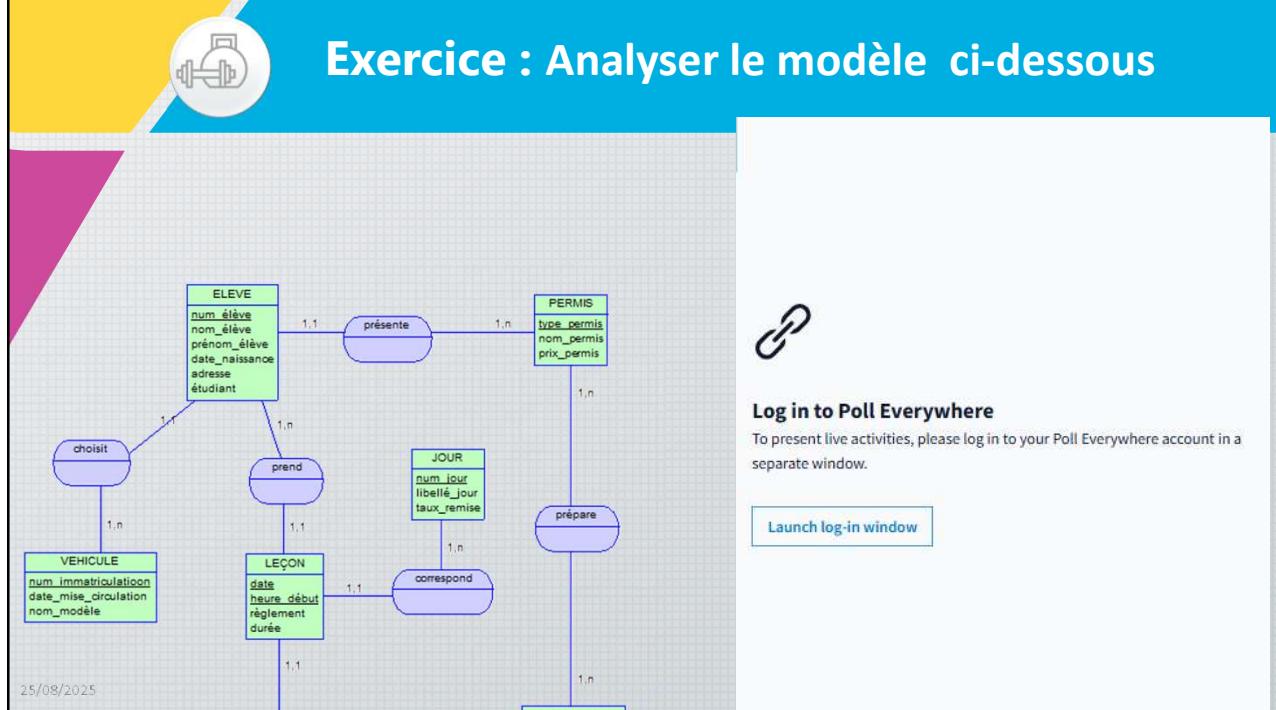


25/08/2025

127

127

## Exercice : Analyser le modèle ci-dessous



25/08/2025

128

## Exercice

Le club de judo souhaite modéliser et organiser ses données afin de faciliter la gestion de ses membres (judokas) et des tournois qu'il organise.

- **Club** : Chaque club a un nom unique et est localisé dans une ville. Un club est responsable de l'organisation de plusieurs tournois tout au long de l'année. Chaque judoka est inscrit à un seul club.
- **Judoka** : Chaque judoka est identifié par un nom, un prénom, et une date de naissance. Il est inscrit dans un seul et unique club. Un judoka peut participer à plusieurs tournois organisés par son club. La participation d'un judoka à un tournoi doit être enregistrée, en spécifiant éventuellement la date d'inscription au tournoi.

25/08/2025

129

129

## Exercice

- **Tournoi** : Chaque tournoi a un nom, une date, et un lieu où il se déroule. Un tournoi est organisé par un seul club, mais plusieurs judokas peuvent y participer. Un tournoi doit être composé d'au moins 3 judokas participants.

25/08/2025

130

130

## Exercice : règles de gestion

### 1. Règles pour l'entité "Club"

- **RG1** : Chaque club doit avoir un nom unique.
- **RG2** : Un club peut organiser plusieurs tournois, mais chaque tournoi doit être associé à un seul club.

### 2. Règles pour l'entité "Judoka"

- **RG3** : Un judoka doit être inscrit à un seul club à la fois.
- **RG4** : Le nom, le prénom et la date de naissance du judoka doivent être obligatoirement enregistrés.

25/08/2025

131

131

## Exercice : règles de gestion

### 1. Règles pour l'entité "Tournoi"

- **RG5** : Chaque tournoi doit être organisé par un seul club et avoir un nom, une date, et un lieu.
- **RG6** : Un tournoi doit compter au moins 3 judokas participants.

### 2. Règles transversales

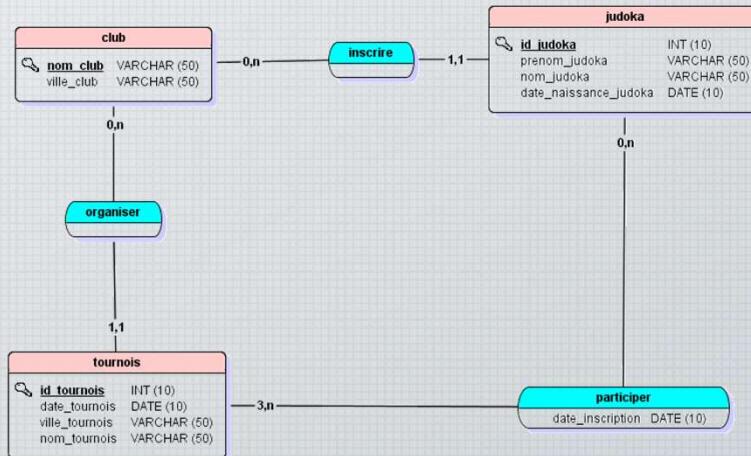
- **RG7** : Un judoka peut participer à plusieurs tournois, mais ne peut être inscrit qu'à un seul club.
- **RG8** : Lorsqu'un judoka participe à un tournoi, sa date d'inscription au tournoi doit être enregistrée.

25/08/2025

132

132

## Corrigé



25/08/2025

133

133

## Les étapes pour la modélisation



Modèle  
Conceptuel de  
données

### Normalisation.

- Procédure pour la vérification de la robustesse du modèle de données.
- S'applique à **toutes les entités et aux relations porteuses de propriétés**
- Vient après la **vérification de la cohérence** du modèle.
  - Unicité des éléments
  - Identification
- S'appuie sur les notions de:
  - dépendance fonctionnelle
  - dépendance multivaluée

Faire disparaître les redondances ou des anomalies de dépendances fonctionnelles

25/08/2025

134

134

## Normalisation : dépendance fonctionnelle

**Définition :** Un attribut Y dépend fonctionnellement d'un attribut X si et seulement si une valeur de X induit une unique valeur de Y.

- On note une dépendance fonctionnelle par :  $X \rightarrow Y$ .
- On peut également dire, **X détermine Y**

LIVRES	
clé	code_livre
	Varchar (20)
	titre_livre
	Varchar (50)
	type_livre
	Varchar (50)
	isbn_livre
	Varchar (10)
	genre_livre
	Varchar (10)
	année_publication
	Int

isbn ——————> titre\_livre

L'attribut isbn détermine un titre de livre unique

25/08/2025

135

## Normalisation : dépendance fonctionnelle

**Définition :** Un attribut Y dépend fonctionnellement d'un attribut X si et seulement si une valeur de X induit une unique valeur de Y.

- On note une dépendance fonctionnelle par :  $X \rightarrow Y$ .
- On peut également dire, **X détermine Y**

Passager	
clé	passeport_passager
	VARCHAR (50)
	nom_passager
	VARCHAR (50)
	prenom_passager
	VARCHAR (50)
	adresse_passager
	VARCHAR (50)

passeport\_passager ——————> nom\_passager

Un numéro de passeport permet d'identifier un unique passager

25/08/2025

136

136

## Normalisation : 1<sup>ère</sup> Forme Normale

- Relation (entité ou association) dont tous les attributs :
  - contiennent une valeur **atomique**.
  - contiennent des valeurs **non multiples**.
  - sont **constants dans le temps**.
- Exemple : Indiquer si l'entité **livre** est en 1<sup>ère</sup> forme normale

25/08/2025

137

LIVRES	
code_livre	Varchar (20)
titre_livre	Varchar (50)
type_livre	Varchar (50)
isbn_livre	Varchar (10)
genre_livre	Varchar (10)
année_publication	Int
auteurs_livre	Varchar (10)

Un livre peut avoir plusieurs auteurs,  
l'attribut **auteurs** peut contenir des  
valeurs multiples, donc l'entité n'est  
pas en 1<sup>ère</sup> forme normale.



137

## Normalisation : 1<sup>ère</sup> Forme Normale

### • Normaliser l'entité **livres**

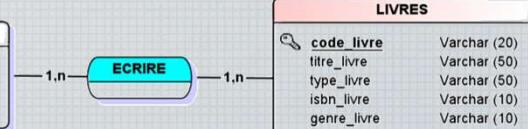
- Créer une nouvelle entité « **auteurs** » pour représenter les auteurs
- Définir une association entre les entités « **livres** » et « **auteurs** »

LIVRES	
code_livre	Varchar (20)
titre_livre	Varchar (50)
type_livre	Varchar (50)
isbn_livre	Varchar (10)
genre_livre	Varchar (10)
année_publication	Int
auteurs_livre	Varchar (10)

1<sup>ère</sup> Forme  
Normale

AUTEURS	
id_auteur	Int
nom_auteur	Varchar (10)
prenom_auteur	Varchar (10)

LIVRES	
code_livre	Varchar (20)
titre_livre	Varchar (50)
type_livre	Varchar (50)
isbn_livre	Varchar (10)
genre_livre	Varchar (10)
année_publication	Int



Aucune répétition de groupes de valeurs (attributs atomiques)

25/08/2025

138

138

## Normalisation : 2<sup>ème</sup> Forme Normale

- Relation qui est en 1<sup>ère</sup> forme normale et :
  - Tout attribut ne composant pas un identifiant est en **dépendance fonctionnelle pleine élémentaire** avec l'identifiant.
  - Dans le cas où l'identifiant est composé de plusieurs attributs les autres attributs de la relation doivent dépendre de l'identifiant en entier (et pas seulement d'une partie de cet identifiant).
  - **Exemple :**

ADHERENTS	
	Pk_Primary <b>num_adherent</b> Varchar (20)
Attributs	nom_adherent Varchar (50) prenom_adherent Varchar (50) date_adhesion Date

- L'identifiant est **num\_adherent**
- Il existe des Dépendances Fonctionnelles :
  - **num\_adherent → nom\_adherent, prenom\_adherent, date\_adhesion**

25/08/2025

139

139

## Normalisation : 2<sup>ème</sup> Forme Normale

- Relation qui est en 1<sup>ère</sup> forme normale et :
  - Tout attribut ne composant pas un identifiant est en **dépendance fonctionnelle pleine élémentaire** avec l'identifiant.
  - Dans le cas où l'identifiant est composé de plusieurs attributs les autres attributs de la relation doivent dépendre de l'identifiant en entier (et pas seulement d'une partie de cet identifiant).
  - **Exemple :**

Facture	
	<b>numero_facture</b>
	<b>numero_article</b>
Attributs	description_article prix_unitaire quantite_article

- L'identifiant est composé des attributs **numero\_facture**, **numero\_article**
- Il existe des Dépendances Fonctionnelles :
  - **numero\_article → description\_article, prix\_unitaire**
  - **numero\_facture → quantité\_article**



L'entité FACTURE n'est pas en 2<sup>ème</sup> forme normale : l'attribut « **quantité\_article** » ne dépend pas pleinement de l'identifiant (**numero\_article** ne détermine pas la quantité)

25/08/2025

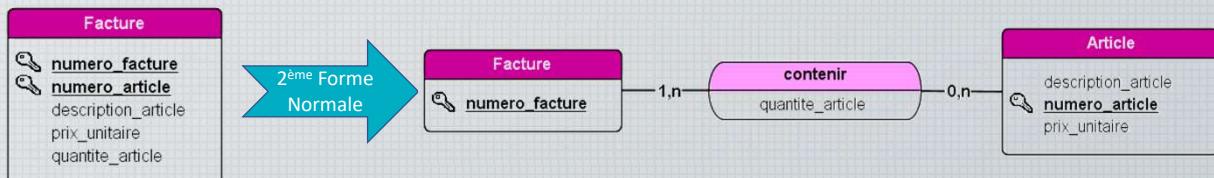
140

140

## Normalisation : 2<sup>ème</sup> Forme Normale

- Exemple :

- Le non-respect de la 2FN entraîne une redondance des données
  - L'attribut « Description Article » est dupliqué pour chaque « numero\_facture »
  - L'attribut « Prix Unitaire » est dupliqué pour chaque « numero\_facture »
  - La quantite\_article dépend de « numero\_facture » et « numero\_article »
- Il faut normaliser.
  - Création d'une nouvelle entité ARTICLE
  - Définition de l'association contient entre les entités FACTURE et ARTICLE
- Le résultat du passage en 2FN est le suivant :



25/08/2025

141

141

## Normalisation : 2<sup>ème</sup> Forme Normale

- Exercice

- Analysez les dépendances fonctionnelles de la table ci-dessous

EMPRUNTER	
Pfk_Prim_Forei	
<b>id_exemplaire</b>	Int
<b>num_adherent</b> Varchar (20)	
Attributs	
date_emprunt	Date
duree_emprunt	Int
date_retour_reelle	Date

- La clé primaire est composée des attributs id\_exemplaire, num\_adherent

25/08/2025

142

142

## Normalisation : 2<sup>ème</sup> Forme Normale

- Corrigé

- Analysez les dépendances fonctionnelles de la table ci-dessous

EMPRUNTER		
PK_Prim_Forei		
	<b>id_exemplaire</b>	Int
	<b>num_adherent</b>	Varchar (20)
Attributs		
	date_emprunt	Date
	duree_emprunt	Int
	date_retour_reelle	Date



L'entité **EMPRUNTER** n'est pas en 2<sup>ème</sup> forme normale : La clé composée ne tient pas compte de la date de l'emprunt, or un même adhérent peut emprunter plusieurs fois le même exemplaire à des dates différentes.

- La clé primaire est composée des attributs **id\_exemplaire**, **num\_adherent**
- Dépendance incorrecte :
  - **id\_exemplaire** → **duree\_emprunt**
  - Un exemplaire peut être emprunté à différentes dates, avec des durées différentes
- La vraie clé primaire devrait être : (**id\_exemplaire**, **num\_adherent**, **date\_emprunt**)

(**id\_exemplaire**, **num\_adherent**, **date\_emprunt**) → **duree\_emprunt**, **date\_retour\_reelle**

25/08/2025

143

## Normalisation : 3<sup>ème</sup> Forme Normale

- Relation qui est en 2<sup>ème</sup> forme normale et :

- Tout attribut ne composant pas un identifiant dépend **directement** de l'identifiant. Aucune dépendance fonctionnelle transitive (pas d'intermédiaire entre clé et attribut)

Exemple :

ADHERENTS		
Pk_Primary		
	<b>num_adherent</b>	Varchar (20)
Attributs		
	nom_adherent	Varchar (50)
	prenom_adherent	Varchar (50)
	date_adhesion	Date

- Il existe des Dépendances Fonctionnelles

num_adherent	→	nom_adherent
num_adherent	→	prenom_adherent
num_adherent	→	date_adhesion

25/08/2025

144

144

## Normalisation : 3<sup>ème</sup> Forme Normale

- Exemple :

Supposons que l'on ajoute à l'entité ADHERENTS des attributs code\_postale, ville et adresse\_adherent.

- Est cette entité en troisième forme normale ?
- Comment normaliser ?

ADHERENTS	
🔍	num_adherent
VARCHAR (20)	
nom_adherent	VARCHAR (50)
prenom_adherent	VARCHAR (50)
date_adhesion	DATE
adresse_adherent	VARCHAR (50)
code_postale	INT
ville	VARCHAR (50)

25/08/2025

145

145

## Normalisation : 3<sup>ème</sup> Forme Normale

- Corrigé :

Supposons que l'on ajoute à l'entité ADHERENTS des attributs code\_postale, ville et adresse. Est cette nouvelle entité en troisième forme normale ?

ADHERENTS	
🔍	num_adherent
VARCHAR (20)	
nom_adherent	VARCHAR (50)
prenom_adherent	VARCHAR (50)
date_adhesion	DATE
adresse_adherent	VARCHAR (50)
code_postale	INT
ville	VARCHAR (50)

Clé primaire : num\_adherent

Attributs non-clés :

ville → code\_postale dépendance entre attributs non-clés

num\_adherent → ville dépendance directe

num\_adherent → code\_postale est transitive via ville

Cela viole la 3<sup>e</sup> forme normale, car code\_postale dépend de ville, qui dépend de la clé.

25/08/2025

146

146

## Normalisation : 3<sup>ème</sup> Forme Normale

- Relation qui est en 2<sup>ème</sup> forme normale et :
  - Tout attribut ne composant pas un identifiant dépend **directement** de l'identifiant. Aucune dépendance fonctionnelle transitive (pas d'intermédiaire entre clé et attribut)

**Exemple :**

Attribut
numero_commande
date_commande
total_commande
nom_client
adresse_client
numero_client

- Il existe des Dépendances Fonctionnelles
 
$$\begin{array}{l} \text{numero_client} \rightarrow \text{nom_client, adresse_client} \\ \text{numero_commande} \rightarrow \text{numero_client, date_commande,} \\ \text{total_commande} \end{array}$$

25/08/2025

147

147

## Normalisation : 3<sup>ème</sup> Forme Normale

- Cette entité n'est pas en 3<sup>ème</sup> forme normale :
  - Les attributs **nom\_client, adresse\_client** sont déterminés par l'identifiant **numero\_commande** mais pas **directement**. La dépendance est par l'**intermédiaire** de l'attribut **numero\_client**.
  - Le non-respect de la 3FN entraîne une redondance des données
    - Les informations du client vont se dupliquer pour chaque facture
- Il faut **normaliser**
  - Conserver dans l'entité initiale (**COMMANDE**) les attributs dépendant directement de la clé.
  - Créer une nouvelle entité (**CLIENT**) pour regrouper les attributs indépendants, l'**attribut de transition** devient l'identifiant (**numero\_client**)
  - Créer une nouvelle association entre les deux entités
    - Le passage en 3FN donne le résultat suivant :



25/08/2025

148

148

## Normalisation : dépendance multivalués

### Définition :

Soit R(A1, A2,...An) un schéma de relation et X et Y des sous-ensembles de A1, A2,...An

- Une **Dépendance Multivaluée** caractérise une indépendance entre deux ensembles d'attributs (Y et Z) corrélés par un même X.
- On dit que X multi-détermine Y (ou Z) :  $X \rightarrow\!\!> Y$  (ou  $X \rightarrow\!\!> Z$ )

### Exemple :

VOL (NV, AVION, PILOTE)

- Tout PILOTE conduit tout AVION sur n'importe quel vol
- AVION et PILOTE sont indépendants
  - NV  $\rightarrow\!\!>$  AVION
  - NV  $\rightarrow\!\!>$  PILOTE

25/08/2025

149

## Normalisation : 4<sup>ème</sup> Forme Normale

### Définition :

- Une relation est en 4NF si et seulement si les seules DME sont celles où un identifiant détermine un attribut

### Exemple :

La relation VOL (NV, AVION, PILOTE) n'est pas en 4FN

- NV  $\rightarrow\!\!>$  AVION
- NV  $\rightarrow\!\!>$  PILOTE



Décomposer les relations ayant  
des DM

25/08/2025

150

150

## Modélisation conceptuelle des données (Étapes)

### Identifier :

- Les concepts qui font partie du vocabulaire du système
- Déterminer les identifiants 

### Déterminer les relations entre les concepts



### Identifier les propriétés de ces concepts



### Définir les cardinalités



### Élaborer le dictionnaire de données



### Vérifier la cohérence et la pertinence du modèle

151

## Le Modèle Logique des Données MLD

Le **Modèle Logique de Données** est une traduction du Modèle Conceptuel (MCD) en un schéma plus formel, adapté au système de gestion de base de données (SGBD), mais encore indépendant des considérations physiques spécifiques. Il se concentre sur la structure des données et leur organisation logique.

**Objectif :** Définir la structure des données de manière qu'elles soient prêtes pour une implémentation dans un SGBD particulier.

### Contenu :

- ✓ **Tables ou relations** : Les entités deviennent des tables dans une base de données relationnelle.
- ✓ **Colonnes ou attributs** : Chaque attribut de l'entité devient une colonne de la table.
- ✓ **Clés primaires et étrangères** : Identification des colonnes uniques (clés primaires) et des relations entre les tables (clés étrangères).
- ✓ **Types de données** : Définition des types de données (entiers, chaînes de caractères, etc.) pour chaque colonne.

# Définitions

- **Clé primaire (Primary Key)** : est un ou plusieurs attributs (colonnes) d'une table qui permettent d'**identifier de manière unique chaque enregistrement** (ligne) dans cette table. Elle ne peut pas contenir de **valeurs nulles, ni doublons**.

**Rôle** : Garantir l'unicité des enregistrements dans une table.

## Critères de qualité de données garantis

- **Unicité** : Chaque enregistrement est unique.
- **Complétude** : Chaque enregistrement est bien identifié (pas de valeurs manquantes dans les attributs qui composent la clé)
- **Exactitude et cohérence** : Les doublons et les ambiguïtés sont évités.

## Exemple :

- **id\_adherent** dans la table ADHERENTS
- **isbn** dans la table LIVRES (si chaque édition est unique)
- **(id\_adherent, id\_exemplaire)** dans la table qui traduit l'association EMPRUNTER

25/08/2025

153

# Définitions

- **Clé étrangère (Foreign Key)** : est un ou plusieurs attributs d'une table qui font référence à la clé primaire d'une autre table (ou parfois de la même table). Elle sert à créer un lien logique entre deux tables.

**Rôle** : Maintenir l'**intégrité référentielle** entre les tables.

## Critères de qualité de données garantis

- **Cohérence** : Assure que les données liées existent vraiment (pas d'orphelins).
- **Intégrité référentielle** : Chaque enregistrement est bien identifié (pas de valeurs manquantes dans les attributs qui composent la clé)

## Exemple :

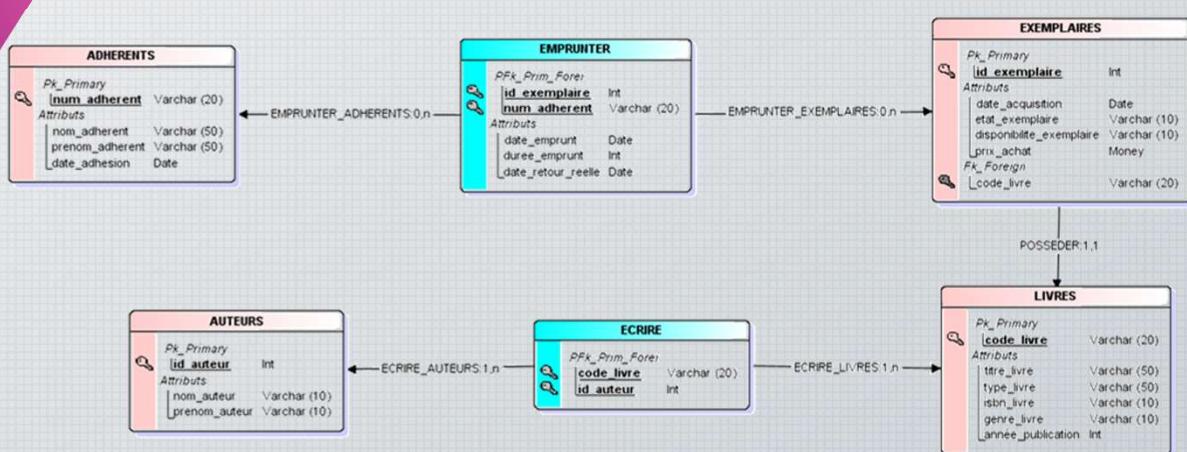
- **code\_livre** dans la table EXEMPLAIRE est une clé étrangère vers LIVRE(code\_livre)
- **id\_adherent** dans EMPRUNTER est une clé étrangère vers ADHERENT(id\_adherent)

L'**intégrité référentielle** garantit qu'on ne peut pas insérer un exemplaire pour un code\_livre qui n'existe pas dans la table LIVRES.

25/08/2025

154

## Le Modèle Logique des Données MLD (Exemple)



25/08/2025

155

155

## Le modèle logique de données (MLD)

Notation Crow's Foot

	Zero or More
	One or More
	One and only One
	Zero or One

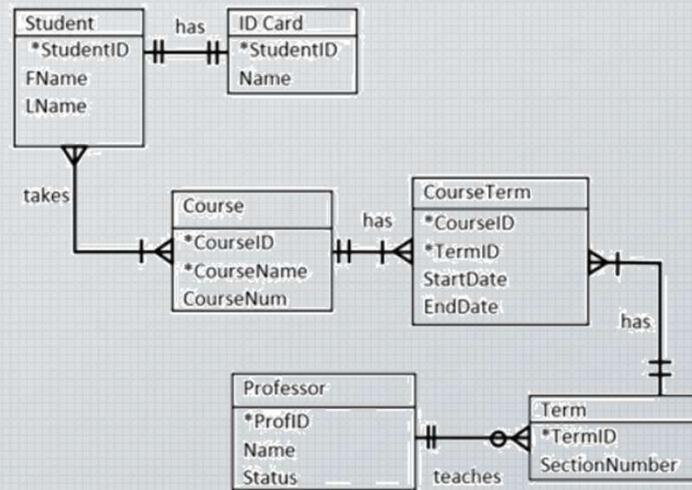
25/08/2025

156

156

## Le modèle logique de données (MLD)

Notation Crow's Foot : Exercice



25/08/2025

157

157

## Les étapes pour la traduction



Modèle Logique de données



Traduire les entités en tables



Traduire les identifiants en clés primaires et les propriétés en colonnes



Traduction des associations



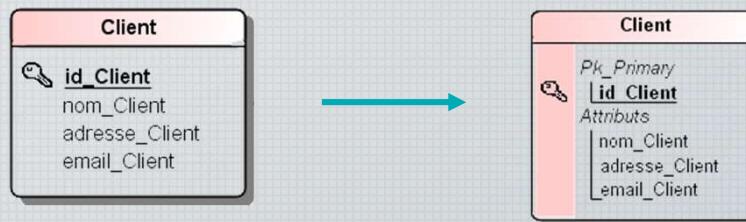
Définir les clés étrangères

25/08/2025

158

## MCD vers MLD

1. Une **entité** du MCD devient une **relation**, c'est à dire une table.
2. Son **identifiant** devient la **clé primaire de la relation**.
  1. La clé primaire permet d'identifier de façon unique un enregistrement dans la table.
  2. Les valeurs de la clé primaire sont uniques.
  3. Les valeurs de la clé primaire sont obligatoirement non nulles.
3. Les autres **propriétés** deviennent les **attributs** de la relation.

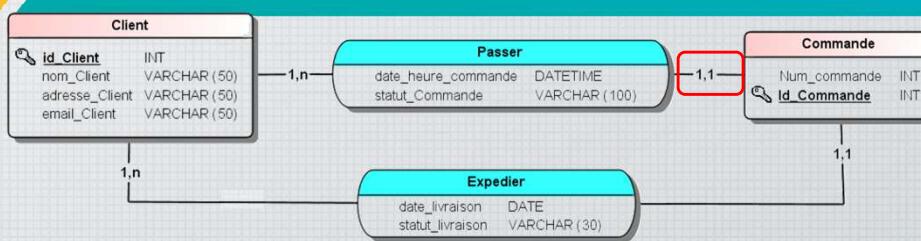


25/08/2025

159

159

## MCD vers MLD



Pour les Associations Binaires 1,n (1,1 → 1,n ou 0,n), les cardinalités maximales positionnées à « 1 » d'un côté de l'association et à « n » de l'autre côté)

La relation issue de l'entité côté cardinalité 1,1 reçoit comme **clé étrangère** la clé primaire de l'entité liée.

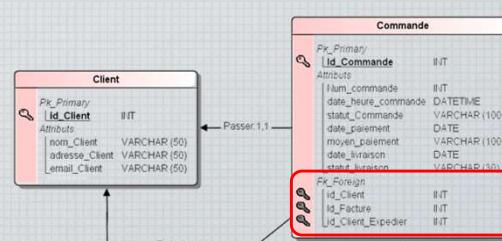
25/08/2025

160

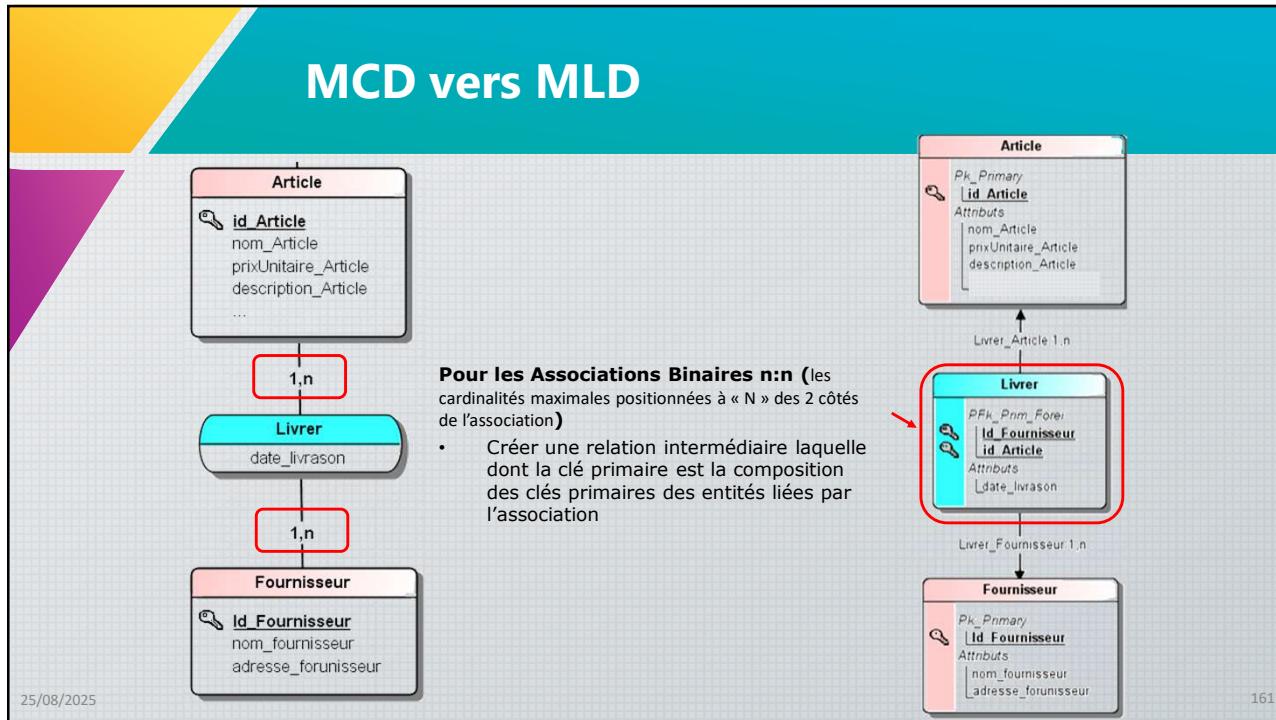
160

### Les clés étrangères ajoutées

- Id\_Client
- Id\_Client\_Expedier

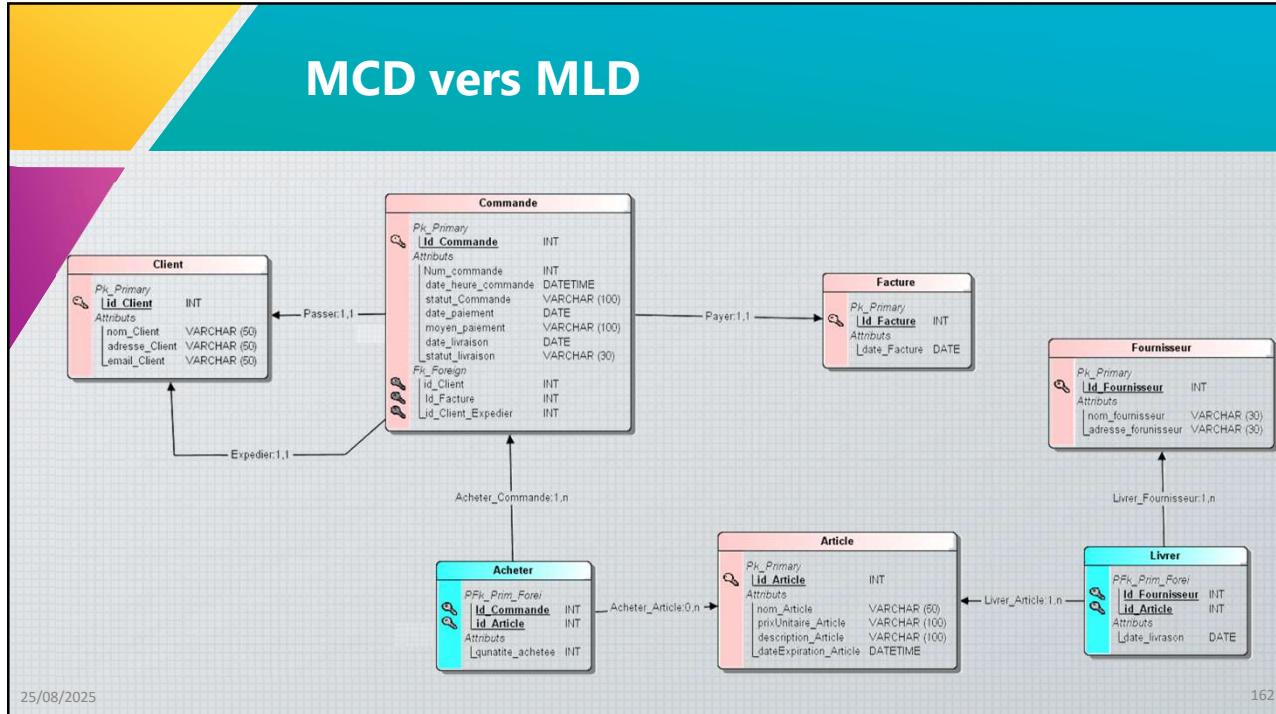


## MCD vers MLD



161

## MCD vers MLD



162

## Le Modèle Physique de Données

Le Modèle Physique de Données (MPD) est la **représentation informatique** concrète de la structure des données, telle qu'elle sera implémentée dans un **système de gestion de base de données (SGBD)** spécifique.

### Objectifs du MPD

- Décrire les tables, les colonnes, les types de données (VARCHAR, INT, DATE, etc.)
- Préciser les contraintes : clés primaires, clés étrangères, contraintes d'unicité, NOT NULL, etc.
- Optimiser le stockage : index, partitions, performances, etc.
- Être adapté au SGBD cible (ex. : PostgreSQL, MySQL, Oracle...)

25/08/2025

163

163

## MLD vers MPD

### Exemple

ADHERENTS	
<i>Pk_Primary</i>	<i>num_adherent</i> VARCHAR (20)
<i>Attributs</i>	nom_adherent VARCHAR (50)
	prenom_adherent VARCHAR (50)
	date_adhesion DATE

```
#-----
# Table: ADHERENTS
#-----

CREATE TABLE ADHERENTS (
    num_adherent Varchar (20) NOT NULL ,
    nom_adherent Varchar (50) NOT NULL ,
    prenom_adherent Varchar (50) NOT NULL ,
    date_adhesion Date NOT NULL ,
    CONSTRAINT ADHERENTS_PK PRIMARY KEY
    (num_adherent)
) ENGINE=InnoDB;
```

25/08/2025

164

164

## MLD vers le modèle physique

Est la création, par des requêtes SQL de type **CREATE TABLE** dans la création tous les **attributs** et leur **type** sont ajoutés. Les **clés primaires** doivent également être définies

```
#-----
# Table: Client
#-----

CREATE TABLE Client(
    id_Client  Int NOT NULL ,
    nom_Client  Varchar (50) NOT NULL ,
    adresse_Client Varchar (50) NOT NULL ,
    email_Client Varchar (50) NOT NULL ,
    CONSTRAINT Client_PK PRIMARY KEY (id_Client)
);
```

25/08/2025

165

165

## Exercice

Réalisez la conversion de le MCD vers le MLD du contexte "Club de Judo"

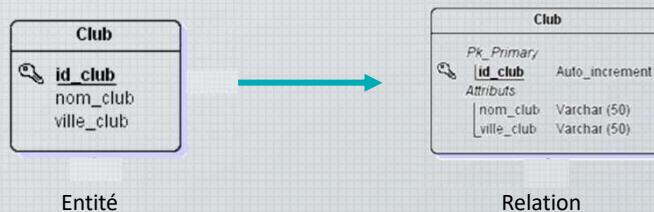
25/08/2025

166\_166

166

# MCD vers MLD

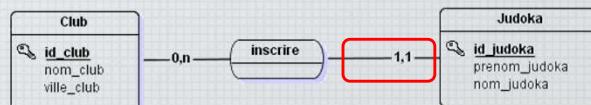
1. Une **entité** du MCD devient une **relation**, c'est à dire une table.
  2. Son **identifiant** devient la **clé primaire de la relation**.
    1. La clé primaire permet d'identifier de façon unique un enregistrement dans la table.
    2. Les valeurs de la clé primaire sont uniques.
    3. Les valeurs de la clé primaire sont obligatoirement non nulles.
  3. Les autres **propriétés** deviennent les **attributs** de la relation.



25/08/2025

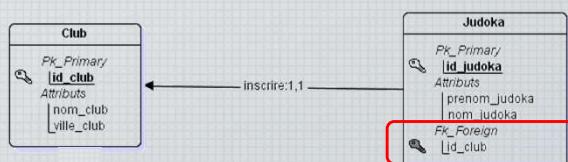
167

## MCD vers MLD



**Pour les Associations Binaires 1,n (1,1 → 1,n ou 0,n, les cardinalités maximales positionnées à « 1 » d'un côté de l'association et à « n » de l'autre côté)**

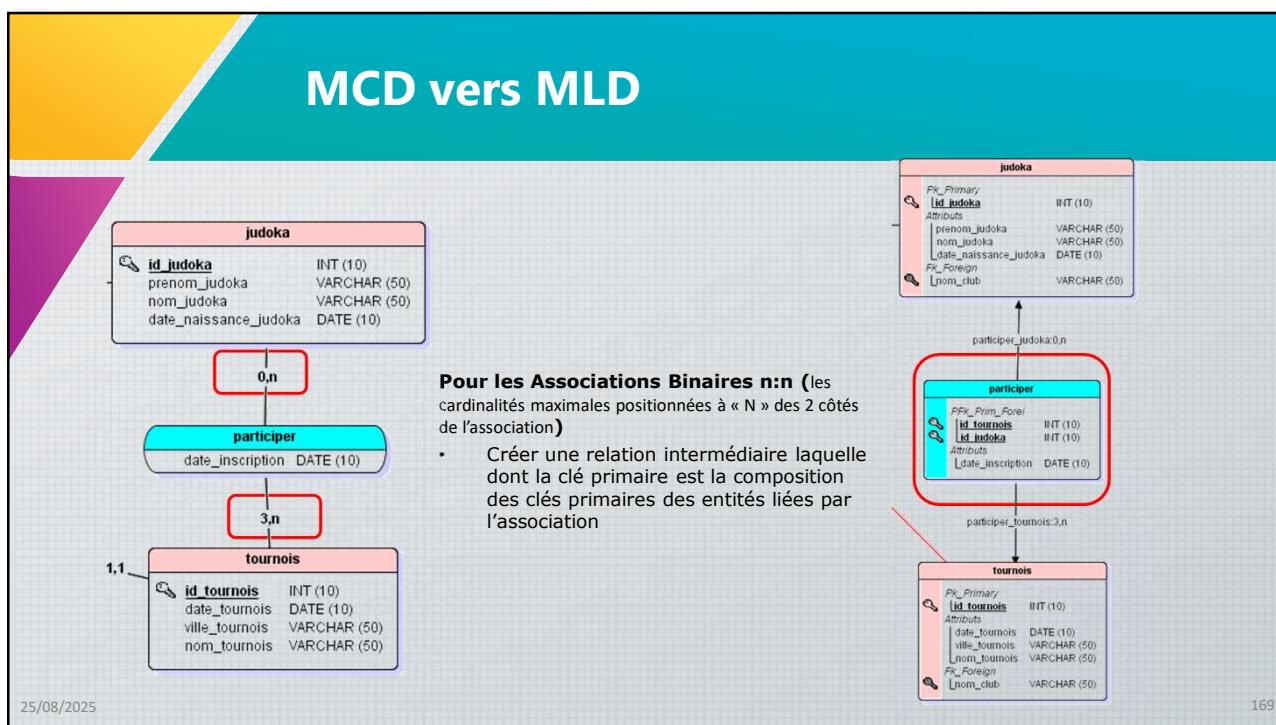
La relation issue de l'entité coté cardinalité 1,1 reçoit comme clé étrangère la clé primaire de l'entité liée.



25/08/2025

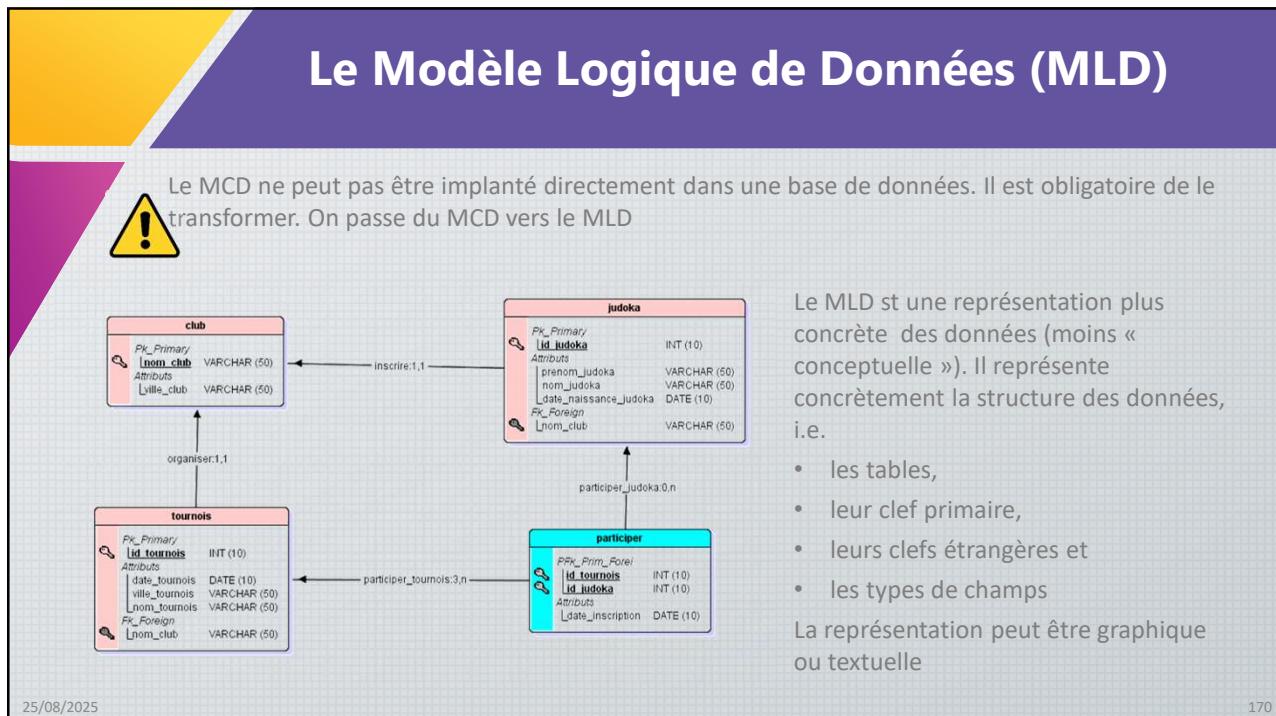
168

## MCD vers MLD



169

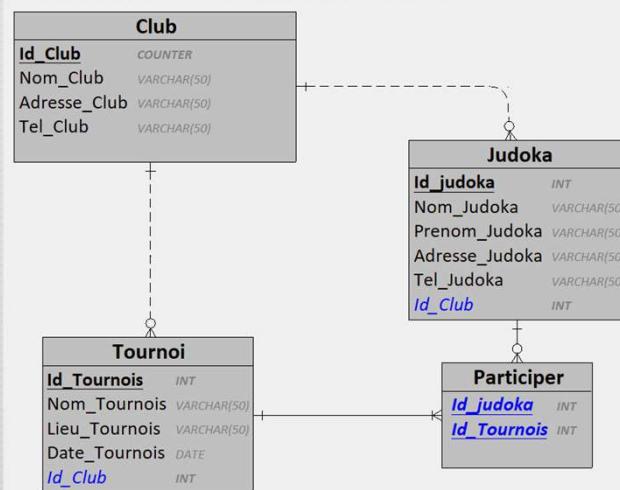
## Le Modèle Logique de Données (MLD)



## Le modèle logique de données (MLD)

Notation Crow's Foot

- Zero or More
- | One or More
- ||---| One and only One
- +---○ Zero or One



25/08/2025

171

171

## MLD vers le modèle physique

```

#-----
# Table: Club
#-----
```

```

CREATE TABLE Club(
    id_club Int Auto_increment NOT NULL,
    nom_club Varchar (50) NOT NULL,
    ville_club Varchar (50) NOT NULL,
    CONSTRAINT Club_PK PRIMARY KEY (id_club)
)ENGINE=InnoDB;
```

```

CREATE TABLE Judoka(
    id_judoka Int Auto_increment NOT NULL,
    prenom_judoka Varchar (50) NOT NULL,
    nom_judoka Varchar (50) NOT NULL,
    date_naissance_judoka Date NOT NULL,
    id_club Int NOT NULL,
    CONSTRAINT Judoka_PK PRIMARY KEY (id_judoka),
    CONSTRAINT Judoka_Club_FK FOREIGN KEY (id_club) REFERENCES Club(id_club)
)ENGINE=InnoDB;
```

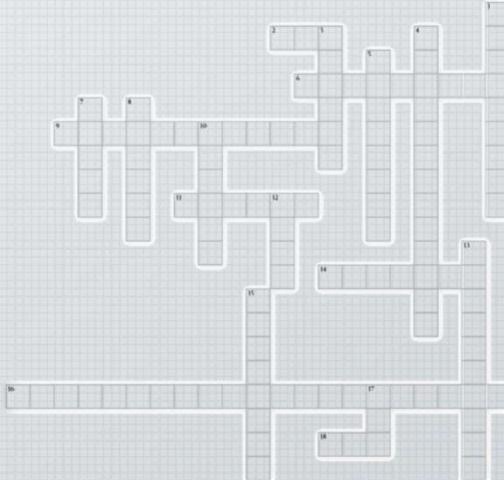
25/08/2025

172

172

## Exercice

Complétez la grille suivante – Révision de notions fondamentales



25/08/2025

173

173

## Exercice

Réaliser un diagramme Entité-Association pour modéliser les données sur les films diffusés par une chaîne de salles de cinéma:

- Pour les films on connaît, le titre, les acteurs, le réalisateur et la salle où le film sera diffusé.
- Pour les acteurs et les réalisateurs on connaît leur nom, prénom ainsi que leurs adresses.
- Un acteur joue dans au moins un film et peut jouer dans plusieurs films.
- Dans un film peuvent jouer plusieurs acteurs, mais dans le cas de films documentaires aucun acteur ne joue.
- Un film a un unique réalisateur

25/08/2025

174

174



## Exercice

### • Corrigé

- Liste des entités
  - Acteur
  - Réalisateur
  - Film
  - Salle
- Liste de propriétés des objets
  - Acteur
    - nom, prénom, adresse
  - Réalisateur
    - nom, prénom, adresse
  - Film
    - titre, format, trailer, durée, résumé, type
  - Salle
    - numéro, adresse, capacité, type (3D, Imax, etc.)
- Déterminer les relations entre les objets
  - Un acteur joue dans des films
  - Un réalisateur réalise des films
  - Un film est projeté dans une salle de cinéma
- Déterminer les identifiants
- Définir les cardinalités

25/08/2025

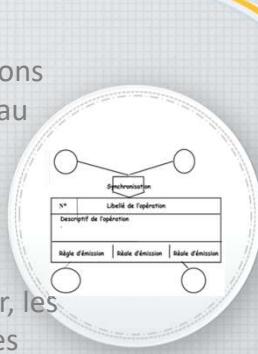
175

175

## Modèle Conceptuel de Traitements

Identifie les grandes fonctions et les interactions. Au niveau logique, le MOT spécifie l'organisation détaillée des traitements.

Les informations traitées incluent les activités métier, les flux d'informations entre les traitements, et les interactions entre les acteurs du système.



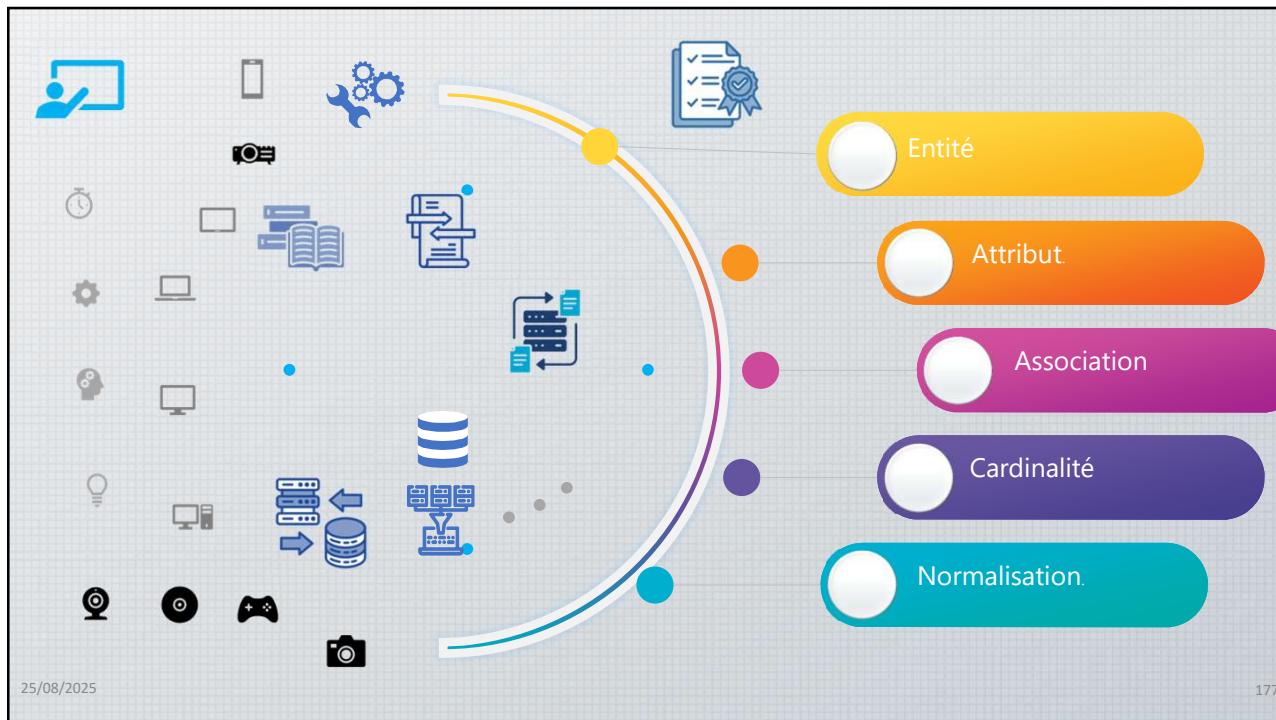
les entités  
= concepts du monde réel

les relations entre les entités

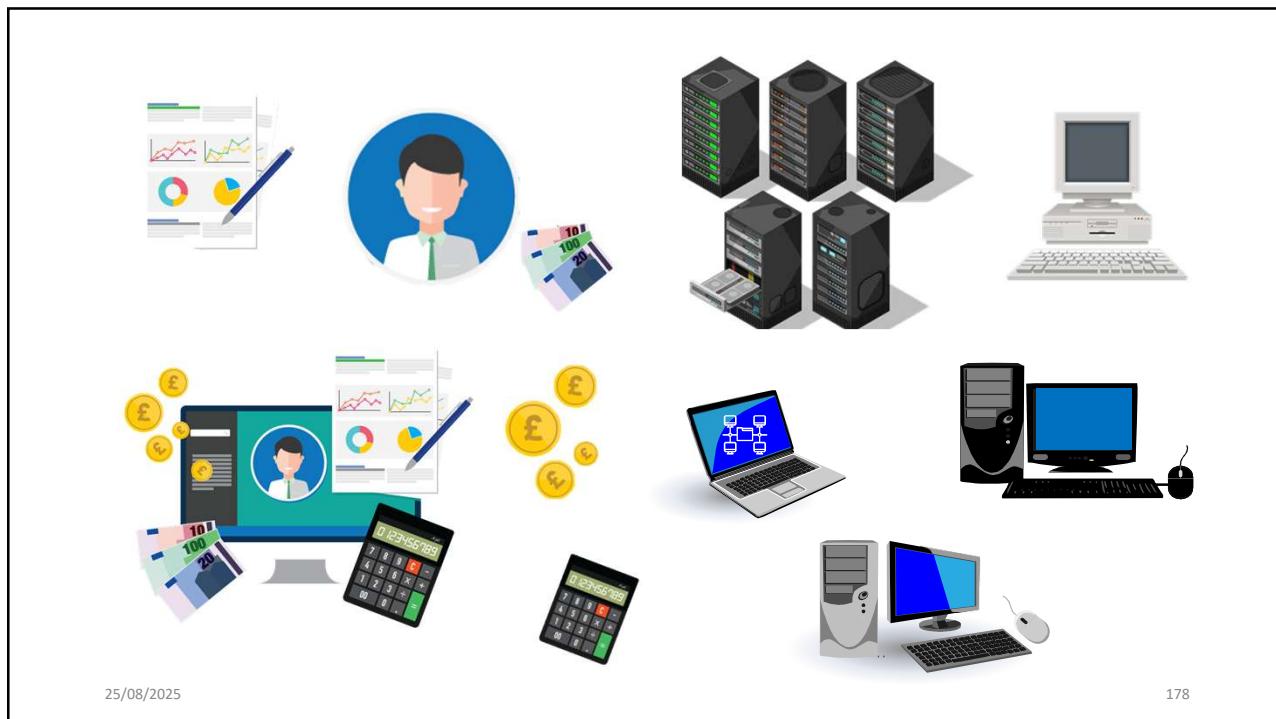
Les propriétés des entités et des relations

176

88



177



178