Chapitre 1 — Introduction aux bases de données

Pourquoi une base de données ? La démarche de conception

Pr. ABDELLAOUI (ENS-Meknès)

Année universitaire 2026–2027

Cours de conception de bases de données

Objectifs du chapitre

À la fin de ce chapitre, vous serez capable de :

- expliquer pourquoi une organisation doit **gérer ses informations** ;
- repérer les **limites des fichiers** et des tableurs (redondance, incohérence, anomalies) ;
- définir les notions de **donnée**, de **base de données** et de **SGBD** ;
- décrire l'architecture **ANSI/SPARC** à trois niveaux ;
- situer les **acteurs** et le **cycle de vie** d'un projet de base de données ;
- présenter la **démarche de conception** : cahier des charges $\rightarrow$ MCD $\rightarrow$ MLD $\rightarrow$ MPD.

Plan du chapitre

1. Gérer l'information dans une organisation
2. L'ère des fichiers et ses limites
3. Données, bases de données et SGBD
4. Panorama des SGBD
5. Architecture ANSI/SPARC et indépendance des données
6. Le modèle relationnel, intuitivement
7. Acteurs et cycle de vie d'un projet de base de données
8. La démarche de conception en trois niveaux
9. Merise, UML et aperçu du fil rouge
10. Mini-projet guidé : le club de sport
11. Nommage et outils du concepteur
12. Synthèse et glossaire
13. QCM d'auto-évaluation
14. Exercices supplémentaires

Gérer l’information dans une organisation

Une organisation vit d'informations

Information de gestion

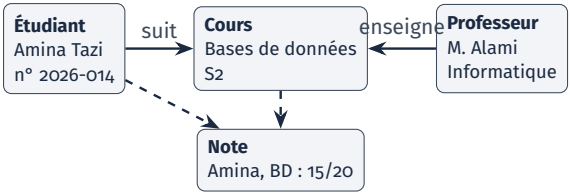
Tout ce qu'une organisation doit **connaître**, **conserver** et **retrouver** pour fonctionner : ses clients, ses produits, ses étudiants, ses livres, ses factures...



Exemple 1 : une école

Que doit savoir une école ?

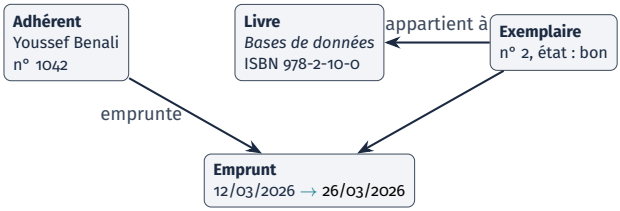
- qui sont ses **étudiants** ;
- quels **cours** sont proposés ;
- quel **professeur** enseigne quoi ;
- quelles **notes** ont été obtenues.



Exemple 2 : une bibliothèque

La bibliothèque BiblioUni

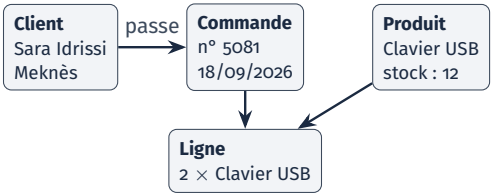
- ses **adhérents** (étudiants, enseignants) ;
- ses **livres** et leurs **exemplaires** ;
- les **emprunts** en cours ;
- les **retards** à relancer.



Exemple 3 : un commerce

Une boutique en ligne

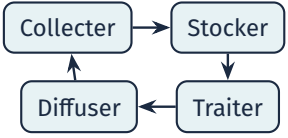
- ses **clients** et leurs adresses ;
- son **catalogue** de produits ;
- les **commandes** passées ;
- le **stock** disponible.



Ce que ces trois exemples ont en commun

Système d'information

Ensemble organisé de **moyens** (personnes, procédures, logiciels) qui permet de **collecter**, **stocker**, **traiter** et **diffuser** l'information dans une organisation.



la base de données est au cœur de « Stocker »

Question : comment faisait-on avant l'informatique ?

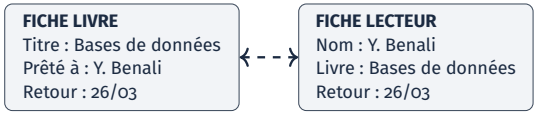
Question

Comment une bibliothèque des années 1970 savait-elle qui avait emprunté quel livre ?

Réponse

Des **fiches cartonnées** rangées dans des tiroirs : une fiche par livre, une fiche par lecteur, et une copie de chaque prêt dans les deux tiroirs.

même info × 2



déjà de la **redondance** !

Avant les bases de données : l'ère des fichiers

L’ère des fichiers et ses limites

Le fichier texte : la première solution

Fichier de données

Suite d'enregistrements stockés sur disque, lus et écrits par un **programme spécifique**. Le format est décidé par le programmeur.

```
1042;Benali;Youssef;9782100;12/03/2026
1042;Benali;Youssef;9782130;15/03/2026
1057;Tazi;Amina;9782100;16/03/2026
1061;Benali;Yousef;9782215;18/03/2026
```

Un fichier `emprunts.txt` : chaque ligne est un emprunt, les champs sont séparés par « ; ».

Le tableur : la solution « moderne » des débutants

Feuille Excel « Emprunts »

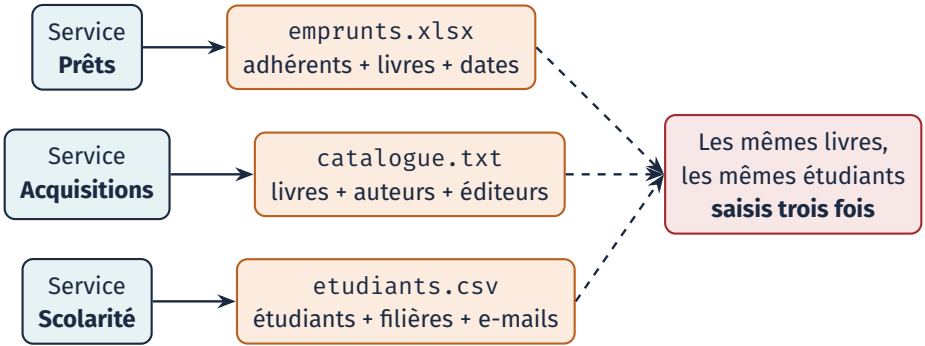
La secrétaire de BiblioUni tient une feuille de calcul : une ligne par emprunt, avec *toutes* les informations utiles.

Question

Cela semble pratique… Où est le problème ?

NumAdh	Nom	Titre	Auteur	Retour
1042	Benali	Bases de données	Gardarin	26/03
1042	Benali	Réseaux	Tanenbaum	29/03
1057	Tazi	Bases de données	Gardarin	30/03
1061	Benali	Algorithmique	Cormen	02/04
1042	Benali	Systèmes	Silberschatz	05/04

Chaque service a son propre fichier



La feuille « Emprunts » complète de BiblioUni

NumAdh	Nom	Prénom	Email	ISBN	Titre	Auteur	Emprunt	Retour
1042	Benali	Youssef	y.benali@umi.ac.ma	978-2-10	Bases de données	Gardarin	12/03/2026	26/03/2026
1042	Benali	Youssef	y.benali@umi.ac.ma	978-2-13	Réseaux	Tanenbaum	15/03/2026	29/03/2026
1057	Tazi	Amina	a.tazi@umi.ac.ma	978-2-10	Bases de données	Gardarin	16/03/2026	30/03/2026
1061	Benali	Yousef	y.benali@umi.ac.ma	978-2-21	Algorithmique	Cormen	18/03/2026	02/04/2026
1042	Benali	Youssef	youssef.b@gmail.com	978-2-40	Systèmes	Silberschatz	22/03/2026	05/04/2026

⚠ Regardez bien…

Cette feuille contient au moins **cinq** problèmes différents. Nous allons les nommer un par un.

Limite 1 : la redondance

Redondance

Une même information est **répétée** à plusieurs endroits alors qu'une seule copie suffirait.

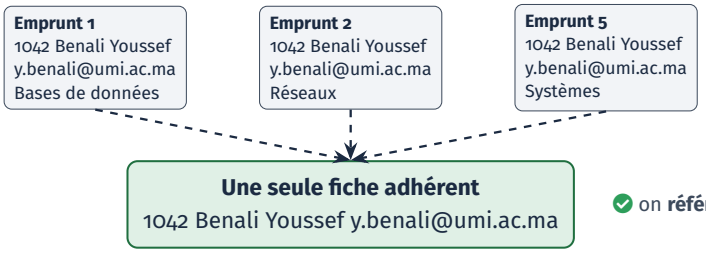
Exemple

Le nom, le prénom et l'e-mail de Youssef Benali sont saisis **trois fois** ; le titre « Bases de données » et son auteur, **deux fois**.

NumAdh	Nom	Email	Titre
1042	Benali	y.benali@umi.ac.ma	Bases de données
1042	Benali	y.benali@umi.ac.ma	Réseaux
1057	Tazi	a.tazi@umi.ac.ma	Bases de données
1042	Benali	y.benali@umi.ac.ma	Systèmes

Place perdue, saisie fastidieuse… et surtout la porte ouverte aux erreurs.

La redondance vue comme des copies d'occurrences



✅ on **référence** l'adhérent au lieu de le recopier

Limite 2 : l'incohérence

Incohérence

Deux copies d'une même information finissent par **se contredire**; on ne sait plus laquelle est vraie.

Exemple

Youssef a deux e-mails; « Yousef » est-il la même personne que « Youssef »? Deux numéros (1042, 1061) pour une seule personne?

NumAdh	Nom	Prénom	Email
1042	Benali	Youssef	y.benali@umi.ac.ma
1061	Benali	Yousef	y.benali@umi.ac.ma
1042	Benali	Youssef	youssef.b@gmail.com



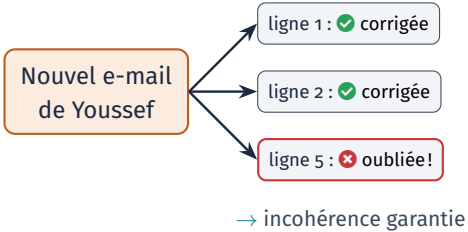
Limite 3 : l'anomalie de mise à jour

Anomalie de mise à jour

Pour modifier **une** information, il faut la corriger dans **toutes** ses copies. En oublier une crée une incohérence.

Exemple

Youssef change d'adresse e-mail : il faut retrouver et corriger ses trois lignes d'emprunt.



Limite 4 : l'anomalie d'insertion

Anomalie d'insertion

On ne peut pas enregistrer une information **tant qu'une autre n'existe pas**, parce que les deux sont mélangées dans la même ligne.

Exemple

Un livre neuf, *Compilation*, jamais emprunté : où l'enregistrer ? Il faudrait une ligne sans adhérent ni date, et le fichier mentirait.

NumAdh	Nom	Titre	Emprunt
1042	Benali	Bases de données	12/03
1057	Tazi	Bases de données	16/03
???	???	Compilation	???

Nouveau livre → Pas d'emprunt → pas de ligne

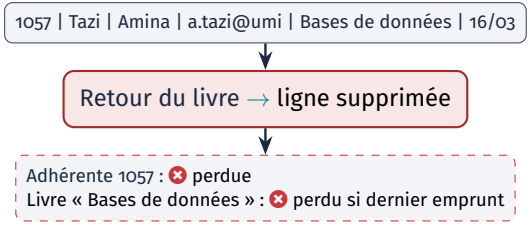
Limite 5 : l'anomalie de suppression

Anomalie de suppression

Supprimer une information en **fait disparaître une autre** qui n'avait rien demandé.

Exemple

Amina rend son livre : on efface sa ligne. Si c'était son seul emprunt, la bibliothèque **oublie qu'Amina existe**.



Limite 6 : la difficulté d'interrogation

Questions simples… en apparence

- Combien d'adhérents *distincts* ont emprunté en mars ?
- Quels livres n'ont *jamais* été empruntés ?
- Qui a plus de deux livres en retard ?

Avec un fichier ou un tableur

Il faut un **programme** ou des formules complexes pour *chaque* question ; le programme dépend du format du fichier et casse dès qu'on ajoute une colonne.

Fichier : un programme par question → Base de données : une requête SQL par question, sans programme

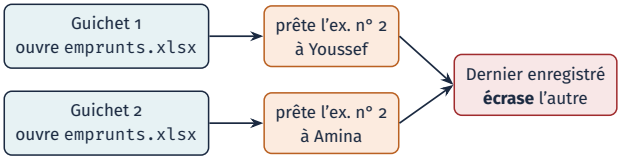
Limite 7 : le partage et les accès concurrents

Accès concurrent

Plusieurs personnes lisent ou modifient **la même donnée au même moment**.

Exemple

Deux guichets prêtent *le même* exemplaire à deux adhérents : chacun a ouvert la feuille, personne ne voit l'autre.



Limite 8 : la sécurité et la confidentialité

Sécurité des données

Chaque personne n'accède qu'aux données **qu'elle a le droit** de voir ou de modifier ; rien n'est perdu en cas de panne.

Exemple

Le fichier « Emprunts » contient les e-mails de tous : quiconque l'ouvre voit tout et peut tout effacer.



Piège fréquent : « un tableur suffit »

⚠ Le tableur n'est pas une base de données

Un tableur est parfait pour *calculer* sur une petite liste. Il devient un piège dès que :

- plusieurs personnes saisissent ;
- les informations sont liées entre elles ;
- la liste dépasse quelques centaines de lignes.

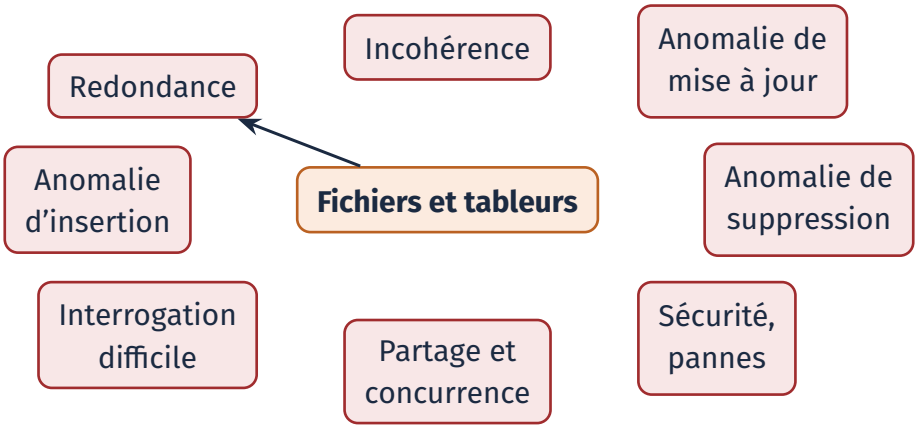
Liste de courses
✅ tableur

Budget personnel
✅ tableur

Gestion des prêts d'une bibliothèque
❌ base de données

Inscriptions de 2 000 étudiants
❌ base de données

Synthèse : les huit limites des fichiers



À vous de jouer : la feuille « Emprunts » de BiblioUni

Exercice 1.1 : une feuille « Emprunts » mal conçue

Énoncé

Voici la feuille tenue par la bibliothèque BiblioUni. Observez-la attentivement, puis répondez aux questions de la frame suivante.

NumAdh	Nom	Email	ISBN	Titre	Auteurs	Catégorie	Retour
1042	Benali	y.benali@umi.ac.ma	978-2-10	Bases de données	Gardarin	Informatique	26/03/2026
1042	Benali	y.benali@umi.ac.ma	978-2-13	Réseaux	Tanenbaum, Wetherall	Informatique	29/03/2026
1057	Tazi	a.tazi@umi.ac.ma	978-2-10	Bases de donnees	Gardarin	Info	30/03/2026
1080	Idrissi	s.idrissi@umi.ac.ma	978-2-70	Histoire du Maroc	Laroui	Histoire	
1042	Benali	y.benali@gmail.com	978-2-10	Bases de données	Gardarin	Informatique	05/04/2026

Exercice 1.1 : questions

Questions

1. Relevez deux exemples de **redondance**.
2. Relevez deux **incohérences**.
3. Que se passe-t-il si l'on supprime la ligne de Sara Idrissi (1080) ?
4. Pourquoi la colonne « Auteurs » pose-t-elle problème ?
5. Comment enregistrer un livre neuf, jamais emprunté ?

Question

Prenez cinq minutes, par binôme.

Corrigé 1.1 (a) : redondance et incohérence

NumAdh	Nom	Email	ISBN	Titre	Auteurs	Catégorie	Retour
1042	Benali	y.benali@umi.ac.ma	978-2-10	Bases de données	Gardarin	Informatique	26/03/2026
1042	Benali	y.benali@umi.ac.ma	978-2-13	Réseaux	Tanenbaum, Wetherall	Informatique	29/03/2026
1057	Tazi	a.tazi@umi.ac.ma	978-2-10	Bases de donnees	Gardarin	Info	30/03/2026
1080	Idrissi	s.idrissi@umi.ac.ma	978-2-70	Histoire du Maroc	Laroui	Histoire	
1042	Benali	y.benali@gmail.com	978-2-10	Bases de données	Gardarin	Informatique	05/04/2026

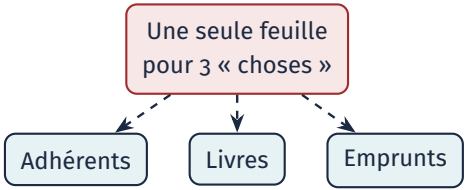
✔ Réponses 1 et 2

Redondance (orange, violet) : l'adhérent 1042 et le livre 978-2-10 sont recopiés à chaque emprunt. **Incohérences** (rouge) : « donnees » sans accent, « Info » au lieu de « Informatique », deux e-mails pour 1042.

Corrigé 1.1 (b) : les anomalies

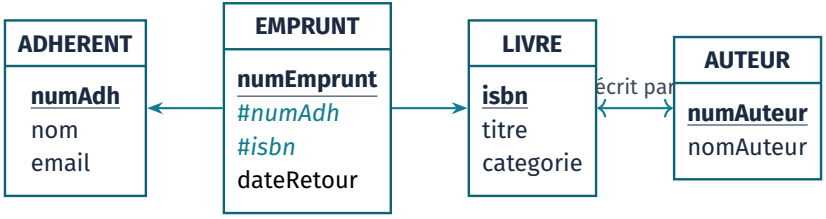
✔ Réponses 3, 4 et 5

3. **Suppression** : on perd l'adhérente 1080 *et* le livre « Histoire du Maroc ».
4. Deux auteurs dans une cellule : impossible de chercher « les livres de Wetherall » proprement.
5. **Insertion** impossible sans un faux emprunt.



→ il faut les **séparer**

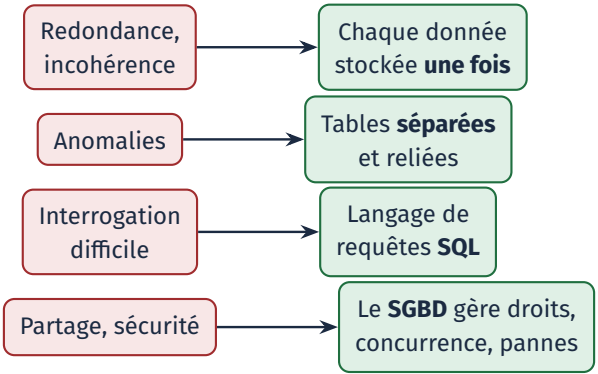
Corrigé 1.1 (c) : vers une meilleure organisation



À retenir

Chaque « chose » (adhérent, livre, auteur, emprunt) a **sa propre table**; les tables se **référencent** par des clés au lieu de recopier les données. C'est tout l'objet de ce cours.

Ce qu'apporte une base de données



À retenir

Partie 1

- Toute organisation doit **collecter, stocker, traiter et diffuser** de l'information.
- Les fichiers et les tableurs **mélangent** des informations de nature différente dans une même ligne.
- Ce mélange provoque **redondance**, **incohérence** et **anomalies** (mise à jour, insertion, suppression).
- Ils ne gèrent ni l'**interrogation**, ni le **partage**, ni la **sécurité**.
- La solution : **séparer** les données en tables reliées, gérées par un SGBD.

Partie 2 : données, bases de données et SGBD

Données, bases de données et SGBD

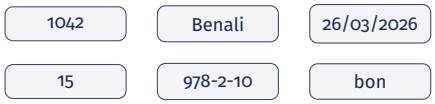
Donnée

Donnée

Élément **brut**, enregistré sous une forme conventionnelle (nombre, texte, date), *sans interprétation*.

Exemple

« 1042 », « Benali », « 26/03/2026 », « 15 » sont des données. Prises isolément, elles ne disent rien.



des valeurs… mais de quoi parle-t-on ?

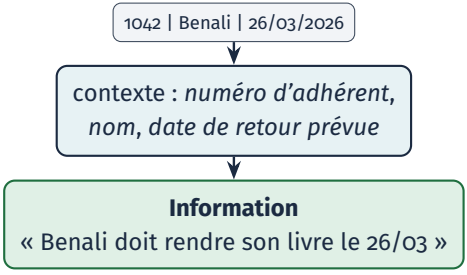
Information

Information

Une donnée **placée dans un contexte** qui lui donne un **sens** : on sait de quoi elle parle et à quoi elle sert.

Exemple

« L'adhérent n° 1042 (Benali) doit rendre son livre le 26/03/2026 » est une information : on peut agir dessus (le relancer).



Connaissance

Connaissance

Des informations **reliées, comprises et réutilisables** pour décider ou prévoir.

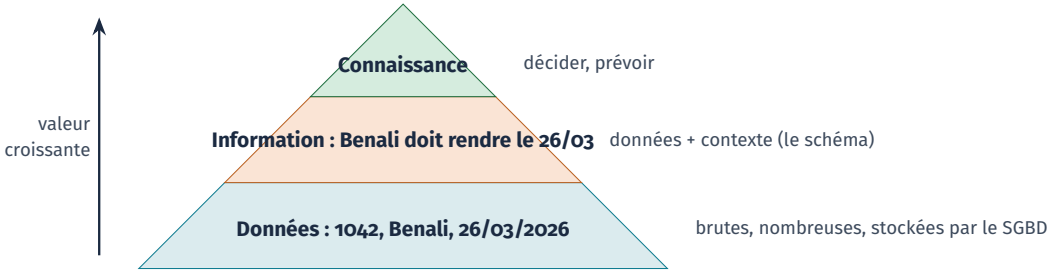
Exemple

« Les livres d'informatique sont empruntés surtout en période d'examens; il faut en acheter des exemplaires supplémentaires en mai. »

À retenir

Une base de données stocke des **données**; c'est leur **structure** (le schéma que vous concevrez) qui les transforme en informations, et l'**analyse** en connaissance.

La pyramide donnée → information → connaissance



Exercice 1.2 : donnée, information ou connaissance ?

Exercice

Classez chacun des énoncés suivants :

1. « 3 »
2. « L'exemplaire n° 3 du livre 978-2-10 est abîmé. »
3. « Les exemplaires prêtés aux première année s'abîment deux fois plus vite : il faut les couvrir. »
4. « Tazi »
5. « Amina Tazi est inscrite depuis le 05/09/2025. »

Corrigé 1.2

✔ Solution

1. « 3 » : **donnée** (un nombre sans contexte).
2. exemplaire abîmé : **information**.
3. les exemplaires s'abîment plus vite : **connaissance** (généralisation, décision).
4. « Tazi » : **donnée**.
5. Amina inscrite depuis… : **information**.

⚠ Piège

Une donnée ne devient une information que si l'on connaît **son nom** (« numExemplaire ») et **sa signification**. C'est pourquoi le concepteur documente chaque donnée dans un **dictionnaire des données** (chapitre 2).

Base de données

Base de données (BD)

Ensemble **structuré** et **cohérent** de données, **persistantes**, décrivant un domaine d'activité, **partagé** par plusieurs utilisateurs et programmes.

Structurée
tables, colonnes, clés

Cohérente
pas de contradiction

Persistante
survit à l'arrêt du programme

Partagée
plusieurs usages

Exemple

La base BiblioUni : les tables ADHERENT, LIVRE, EXEMPLAIRE, AUTEUR, EMPRUNT… et leurs liens.

Une base de données est un modèle du réel



Système de gestion de bases de données (SGBD)

SGBD

Logiciel qui sert d'intermédiaire entre les utilisateurs (ou leurs programmes) et la base de données : il **stocke**, **retrouve**, **protège** et **partage** les données.

Exemple

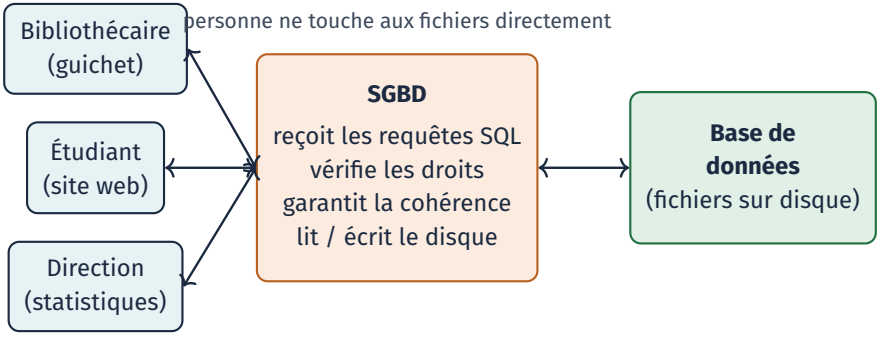
MySQL, PostgreSQL, Oracle, SQL Server, SQLite…

Ne pas confondre

- la **base** = les données (le contenu) ;
- le **SGBD** = le logiciel qui les gère (le contenant intelligent).

On dit « une base MySQL » par abus de langage.

Où se place le SGBD ?



Rôle 1 : définir les données

Langage de définition de données (LDD)

Le SGBD permet de **décrire la structure** : quelles tables, quelles colonnes, quels types, quelles clés.

C'est la traduction directe de votre travail de conception (MPD).

```sql
CREATE TABLE Adherent (
  numAdh    INT PRIMARY KEY,
  nom       VARCHAR(50) NOT NULL,
  prenom    VARCHAR(50),
  email     VARCHAR(100)
);
```

Rôle 2 : manipuler les données

Langage de manipulation de données (LMD)

Le SGBD permet d'**ajouter**, de **modifier**, de **supprimer** et surtout d'**interroger** les données, sans écrire de programme.

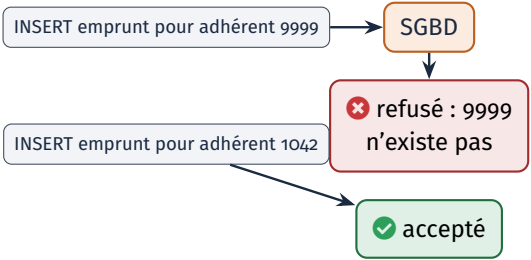
```
INSERT INTO Adherent
  VALUES (1042, 'Benali', 'Youssef',
          'y.benali@umi.ac.ma');

SELECT nom, prenom
FROM   Adherent
WHERE  numAdh = 1042;
```

Rôle 3 : contrôler l'intégrité

Intégrité

Le SGBD **refuse** toute opération qui rendrait la base incohérente : deux adhérents avec le même numéro, un emprunt pour un adhérent inexistant, une date de retour avant la date d'emprunt…



Rôle 4 : sécurité et droits d'accès

Confidentialité

Chaque utilisateur possède un **compte** et des **droits** : lire telle table, modifier telle autre, rien sur le reste.

Exemple

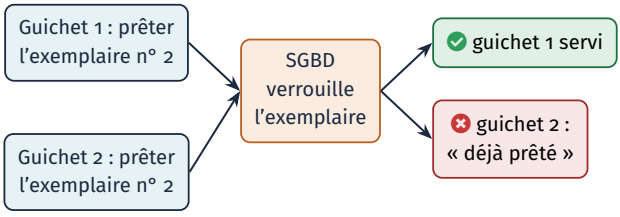
Le site web des étudiants ne peut que *lire* le catalogue ; seul le guichet peut *écrire* des emprunts.

Compte	LIVRE	EMPRUNT	ADHERENT
site web	lecture	⊗	⊗
guichet	lecture	lecture / écriture	lecture
administrateur	tout	tout	tout

Rôle 5 : gérer la concurrence

Contrôle de concurrence

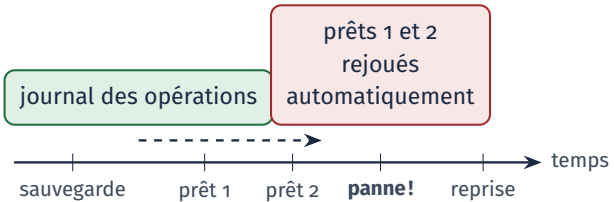
Quand deux utilisateurs modifient la même donnée en même temps, le SGBD les **sérialise** (l'un après l'autre) pour que le résultat soit cohérent.



Rôle 6 : reprise après panne

Reprise (recovery)

Après une coupure de courant ou un plantage, le SGBD **restaure** la base dans un état cohérent grâce à un **journal** des opérations et aux **sauvegardes**.



Les six rôles d'un SGBD : récapitulatif

Rôle	Question à laquelle il répond	Mot-clé
Définition	Quelle est la structure des données ?	LDD, CREATE TABLE
Manipulation	Comment lire, ajouter, modifier, supprimer ?	LMD, SELECT, INSERT
Contrôle d'intégrité	Les données sont-elles cohérentes ?	clés, contraintes
Sécurité	Qui a le droit de faire quoi ?	comptes, droits
Concurrence	Que se passe-t-il si deux personnes agissent en même temps ?	verrous, transactions
Reprise	Que se passe-t-il après une panne ?	journal, sauvegarde

Exercice 1.3 : quel rôle du SGBD ?

Exercice

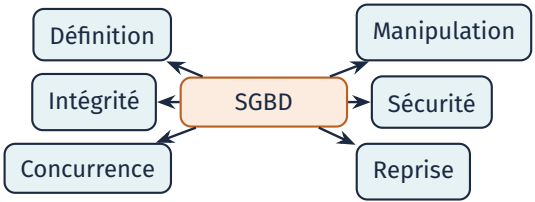
Pour chaque situation, indiquez le rôle du SGBD concerné.

1. Un stagiaire tente de supprimer la table ADHERENT ; l'opération est refusée.
2. On ajoute une colonne « téléphone » à la table ADHERENT.
3. Deux guichets enregistrent un prêt du même exemplaire à la même seconde.
4. Le courant est coupé pendant l'enregistrement d'un prêt.
5. On saisit un prêt pour un exemplaire qui n'existe pas ; le SGBD refuse.
6. On liste les adhérents en retard.

Corrigé 1.3

✔ Solution

1. Refus de suppression → **sécurité**.
2. Nouvelle colonne → **définition**.
3. Même seconde → **concurrence**.
4. Coupure de courant → **reprise**.
5. Exemplaire inexistant → **intégrité**.
6. Liste des retards → **manipulation**.



Panorama des SGBD

Les grands SGBD relationnels

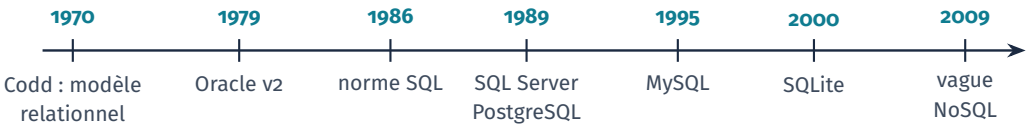
SGBD	Éditeur	Licence	Usage typique
MySQL / MariaDB	Oracle / communauté	libre	sites web, PME, enseignement
PostgreSQL	communauté	libre	applications exigeantes, données géographiques
Oracle Database	Oracle	commerciale	grandes entreprises, banques
SQL Server	Microsoft	commerciale (version gratuite)	entreprises, environnement Windows
SQLite	D. R. Hipp	domaine public	applications mobiles, navigateurs, prototypes

À retenir

Tous parlent le même langage, **SQL**, et reposent sur le même **modèle relationnel** : ce que vous concevez dans ce cours fonctionne avec chacun d'eux.

Un peu d'histoire

tout commence par un **modèle**, pas par un logiciel



MySQL et PostgreSQL : les deux libres

MySQL / MariaDB

- le plus répandu sur le web (WordPress, Facebook à ses débuts) ;
- simple à installer, rapide en lecture ;
- outil graphique : **MySQL Workbench** ;
- utilisé dans nos TP.

PostgreSQL

- très respectueux de la norme SQL ;
- riche en fonctionnalités avancées (types, géographie, JSON) ;
- réputé pour sa robustesse ;
- outil graphique : pgAdmin.

Oracle, SQL Server et SQLite

Oracle

Le SGBD historique des grandes entreprises : performances, haute disponibilité, mais coût élevé et administration complexe.

SQL Server

Le SGBD de Microsoft, bien intégré à Windows et aux outils .NET ; version Express gratuite pour apprendre.

SQLite

Pas de serveur : la base tient dans *un seul fichier*. Présent dans chaque téléphone et chaque navigateur. Idéal pour tester.



Et le NoSQL ?

NoSQL

SGBD **non relationnels** apparus vers 2009 pour les très grands volumes du web : documents (MongoDB), clé-valeur (Redis), colonnes (Cassandra), graphes (Neo4j).

Ils ne remplacent pas le relationnel : ils le *complètent* pour des besoins spécifiques.

Relationnel
tables + SQL
cohérence forte

Documents
JSON imbriqué
(MongoDB)

Clé-valeur
cache rapide
(Redis)

Graphes
réseaux sociaux
(Neo4j)

ce cours : le relationnel, socle de tous les autres

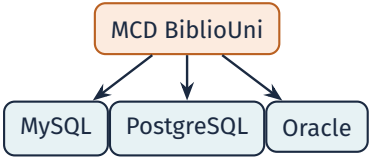
Piège : choisir le logiciel avant de concevoir

⚠ Erreur classique

« On va faire la base sous MySQL » n'est pas un début de conception. Le choix du SGBD est une décision *physique*, prise en dernier.

À retenir

Le MCD est **indépendant** du logiciel : le même MCD donne une base MySQL, PostgreSQL ou Oracle.



un modèle, plusieurs implantations

Architecture ANSI/SPARC et indépendance des données

Trois façons de voir la même base

Architecture ANSI/SPARC (1975)

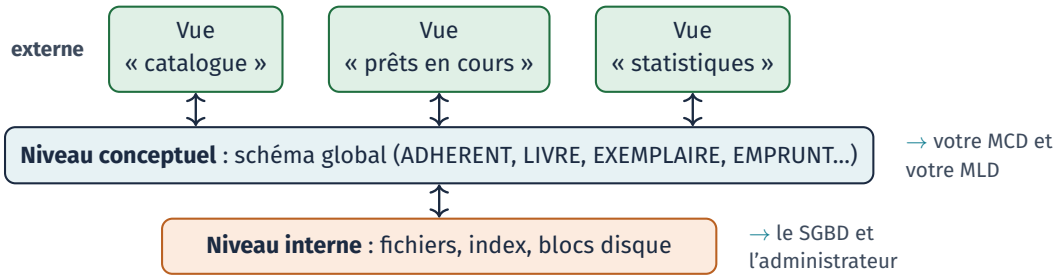
Une base de données est décrite à **trois niveaux** :

- **externe** : ce que voit chaque utilisateur (ses *vues*) ;
- **conceptuel** : la structure globale, indépendante du stockage ;
- **interne** : l'organisation physique sur disque.

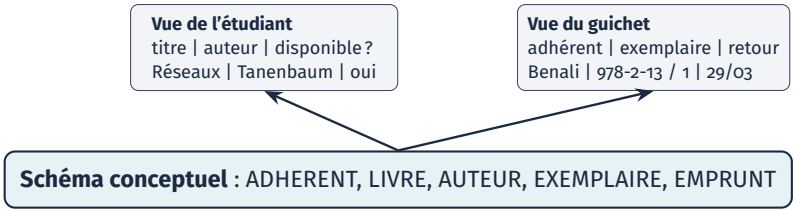
BiblioUni

L'étudiant voit « le catalogue » ; le guichet voit « les prêts en cours » ; le schéma conceptuel décrit *toutes* les tables ; le disque contient des fichiers et des index.

Le schéma ANSI/SPARC



Exemple : deux vues externes sur BiblioUni



chaque vue ne montre qu'une **partie** de la base

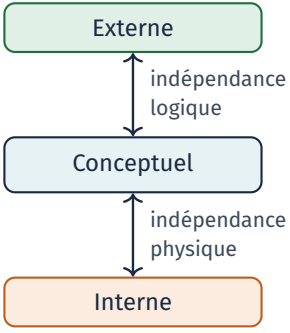
L'indépendance des données

Indépendance logique

On peut modifier le schéma **conceptuel** (ajouter une table, une colonne) sans casser les **vues** existantes.

Indépendance physique

On peut modifier le stockage **interne** (ajouter un index, changer de disque) sans toucher au schéma conceptuel ni aux programmes.



Exercice 1.4 : à quel niveau ?

Exercice

Situez chaque action dans l'architecture ANSI/SPARC :

1. créer un index sur la colonne *nom* ;
2. ajouter l'entité EDITEUR ;
3. créer un écran « mes emprunts » ;
4. déplacer la base sur un disque plus rapide.

Solution

1. index → **interne** ;
2. entité EDITEUR → **conceptuel** ;
3. écran → **externe** ;
4. disque → **interne** (indépendance physique).

À retenir

Partie 2

- **Donnée** (brute) → **information** (donnée + sens) → **connaissance** (décision).
- Une **base de données** est un ensemble structuré, cohérent, persistant et partagé de données.
- Le **SGBD** est le logiciel qui définit, manipule, protège et partage ces données (six rôles).
- MySQL, PostgreSQL, Oracle, SQL Server, SQLite parlent tous **SQL**.
- Architecture **ANSI/SPARC** : externe / conceptuel / interne, avec indépendance logique et physique.

Partie 3 : tables, acteurs et cycle de vie d'un projet

Le modèle relationnel, intuitivement

Une table

Table

Une **table** rassemble des données de **même nature** (tous les adhérents, tous les livres) sous forme de **lignes** et de **colonnes**.

Une base de données relationnelle est un ensemble de tables **reliées** entre elles.

Table ADHERENT

numAdh	nom	prenom	email
1042	Benali	Youssef	y.benali@umi.ac.ma
1057	Tazi	Amina	a.tazi@umi.ac.ma
1080	Idrissi	Sara	s.idrissi@umi.ac.ma

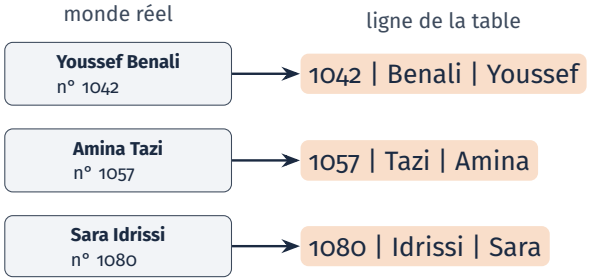
Les lignes : une ligne = une occurrence

Ligne (enregistrement, tuple)

Chaque ligne décrit **un individu** du monde réel : un adhérent précis, un livre précis.

Règle

Deux lignes ne doivent jamais décrire le même individu : **pas de doublon**.



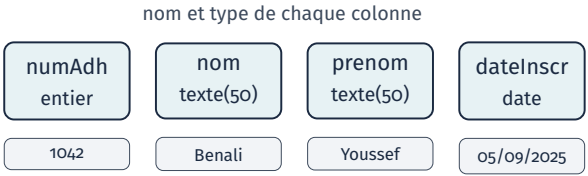
Les colonnes : une colonne = une propriété

Colonne (attribut, champ)

Chaque colonne décrit **une propriété** des individus, avec un **nom** et un **type** (entier, texte, date).

Règle

Une cellule contient **une seule valeur** : jamais « Tanenbaum, Wetherall » dans une case.



La clé : reconnaître une ligne sans ambiguïté

Clé primaire

Colonne (ou groupe de colonnes) dont la valeur est **unique** pour chaque ligne et **jamais vide** : elle identifie la ligne.

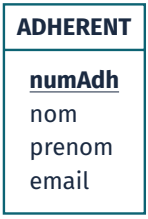
Exemple

numAdh pour ADHERENT, *isbn* pour LIVRE. Le *nom* ne convient pas : deux « Benali » existent.

numAdh	nom	prenom
1042	Benali	Youssef
1057	Tazi	Amina
1093	Benali	Karim

✓ numAdh : toujours différent ✗ nom : doublon

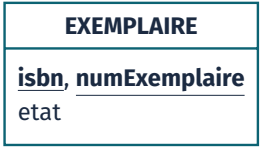
La clé primaire dans le schéma



clé simple



clé simple



clé composée de deux colonnes

À retenir

Convention : la clé primaire est **soulignée en gras**. Une clé peut être **composée** : l'exemplaire n° 2 n'a de sens que pour un livre donné.

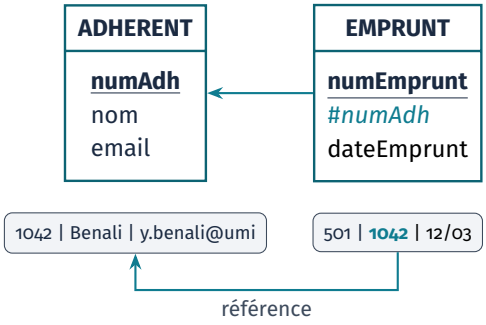
Relier deux tables : la clé étrangère

Clé étrangère

Colonne qui **recopie la clé primaire** d'une autre table pour y faire **référence**. Notée *#numAdh*.

Exemple

Dans EMPRUNT, *#numAdh* dit *quel* adhérent emprunte, sans recopier son nom ni son e-mail.



Exemple BiblioUni : trois tables reliées

numAdh	nom	email
1042	Benali	y.benali@umi.ac.ma
1057	Tazi	a.tazi@umi.ac.ma

numEmprunt	#numAdh	#isbn	dateRetour
501	1042	978-2-10	26/03/2026
502	1042	978-2-13	29/03/2026
503	1057	978-2-10	30/03/2026

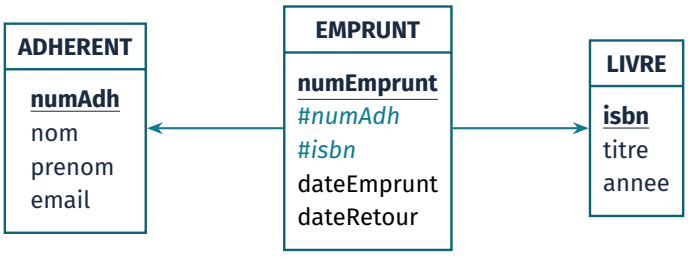
isbn	titre	auteur
978-2-10	Bases de données	Gardarin
978-2-13	Réseaux	Tanenbaum

Lecture

L'emprunt 503 concerne l'adhérent **1057** (donc Amina Tazi) et le livre **978-2-10** (donc « Bases de données »). Le nom d'Amina n'est écrit **qu'une fois** dans toute la base.

Le même exemple sous forme de schéma

les flèches vont de la clé étrangère vers la clé primaire



EMPRUNT (**numEmprunt**, *#numAdh*, *#isbn*, dateEmprunt, dateRetour)

Vocabulaire : trois mots pour la même chose

Langage courant	Théorie relationnelle	SQL / SGBD	Exemple
table	relation	table	ADHERENT
ligne	tuple, n-uplet	enregistrement, row	(1042, Benali, Youssef)
colonne	attribut	champ, column	nom
en-tête	schéma de relation	structure	ADHERENT(numAdh, nom, prenom)
nombre de lignes	cardinalité	count	3 adhérents
nombre de colonnes	degré	—	4 colonnes

⚠ Attention

Le mot « cardinalité » a un *autre* sens dans le MCD (chapitre 4) : ne les confondez pas.

Pièges fréquents sur les tables

⚠ À ne jamais faire

- plusieurs valeurs dans une cellule ;
- des colonnes numérotées : auteur1, auteur2, auteur3 ;
- compter sur l'*ordre* des lignes (« le premier est le plus ancien ») ;
- une table sans clé primaire.

isbn	titre	auteurs
978-2-13	Réseaux	Tanenbaum, Wetherall

isbn	titre	auteur1	auteur2
978-2-13	Réseaux	Tanenbaum	Wetherall
978-2-10	Bases de données	Gardarin	(vide)

✗ et si un livre a quatre auteurs ?

Exercice 1.5 : trouver la clé

Exercice

Pour chaque table, proposez une clé primaire et justifiez.

1. ETUDIANT(nom, prenom, dateNaissance, cne, filiere)
2. COURS(intitule, semestre, nbHeures, codeCours)
3. SALLE(batiment, numeroSalle, capacite)
4. NOTE(cne, codeCours, valeur, dateExamen)

Question

Indice : pour NOTE, une seule colonne suffit-elle ?

Corrigé 1.5

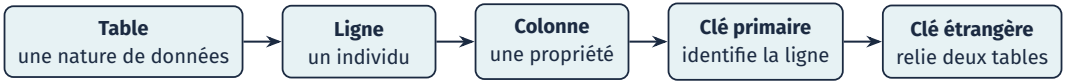
✔ Solution

1. **cne** : unique ; nom + prénom ne le sont pas.
2. **codeCours** : l'intitulé peut se répéter.
3. **batiment**, **numeroSalle** : la salle 12 existe dans chaque bâtiment (clé composée).
4. **cne**, **codeCours** : une note par étudiant et par cours (clé composée).

SALLE
batiment, **numeroSalle** capacite

NOTE
#cne, ***#codeCours*** valeur dateExamen

À retenir : le modèle relationnel en cinq mots

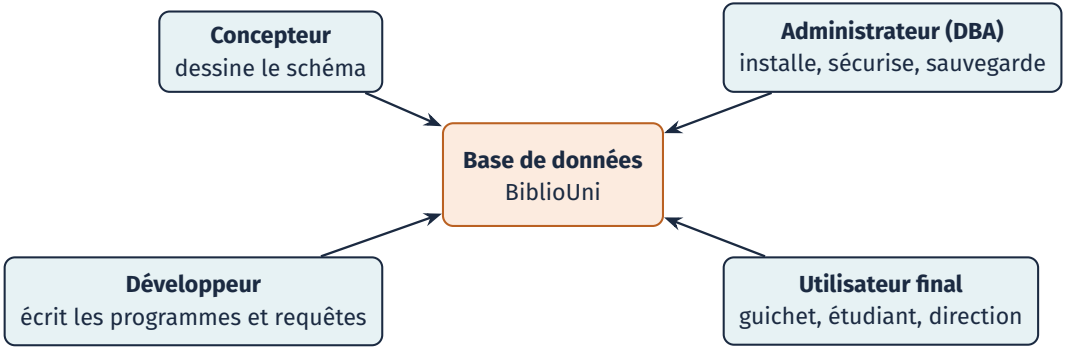


À retenir

Nous **viserons** ces tables à la fin de la démarche ; mais nous ne commencerons *jamais* par elles : nous commencerons par le MCD.

Acteurs et cycle de vie d’un projet de base de données

Qui travaille autour d'une base de données ?



L'administrateur de bases de données (DBA)

Administrateur

Responsable du **fonctionnement** de la base : installation du SGBD, comptes et droits, sauvegardes, performances, reprise après panne.

Chez BiblioUni

Il crée le compte « guichet », programme une sauvegarde chaque nuit, ajoute un index quand la recherche par titre devient lente.



niveau **interne** de l'architecture ANSI/SPARC

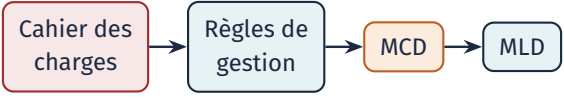
Le concepteur : c'est vous, dans ce cours

Concepteur (analyste)

Il **écoute** les utilisateurs, **analyse** le cahier des charges et produit les **modèles** (MCD, MLD) qui décrivent les données à stocker.

Chez BiblioUni

Il comprend qu'un livre a plusieurs exemplaires, qu'un exemplaire n'est prêté qu'à une personne à la fois, et le traduit en schéma.



niveau **conceptuel** : chapitres 2 à 7

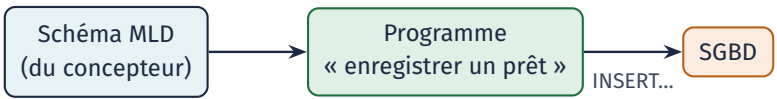
Le développeur

Développeur

Il écrit les **programmes** (site web, application de guichet) qui envoient des **requêtes SQL** au SGBD, à partir du schéma fourni par le concepteur.

Attention

Un développeur qui n'a pas de schéma clair *improvise* des tables : c'est la source n° 1 des bases mal conçues.



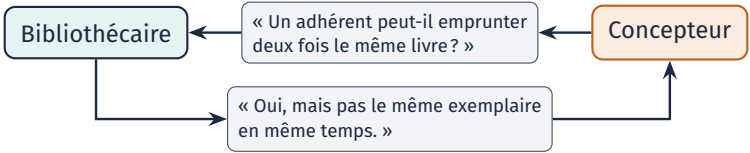
L'utilisateur final

Utilisateur final

Celui qui **se sert** de la base au quotidien à travers des écrans, *sans savoir* qu'il y a des tables derrière : bibliothécaire, étudiant, responsable.

À retenir

C'est **lui** qui connaît le métier. Le concepteur doit l'**interroger** : ses réponses deviennent les règles de gestion.



Exercice 1.6 : qui fait quoi ?

Exercice

Attribuez chaque tâche à un acteur : concepteur, administrateur, développeur ou utilisateur final.

1. Programmer la sauvegarde quotidienne.
2. Dessiner le MCD de BiblioUni.
3. Saisir le retour d'un livre au guichet.
4. Écrire la page web « rechercher un livre ».
5. Décider qu'un exemplaire ne peut être prêté qu'à une personne à la fois.
6. Créer le compte « site web » avec droit de lecture seule.

Corrigé 1.6

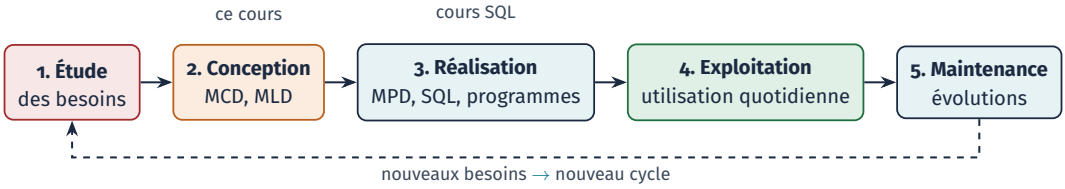
✔ Solution

1. Sauvegarde → **administrateur**.
2. MCD → **concepteur**.
3. Retour au guichet → **utilisateur final**.
4. Page web → **développeur**.
5. Règle de gestion → **utilisateur** (le concepteur la *recueille*).
6. Compte lecture seule → **administrateur**.

⚠ Dans une petite structure

Une même personne cumule souvent plusieurs rôles. Cela ne supprime pas les rôles : on doit *quand même* concevoir avant de développer.

Le cycle de vie d'un projet de base de données



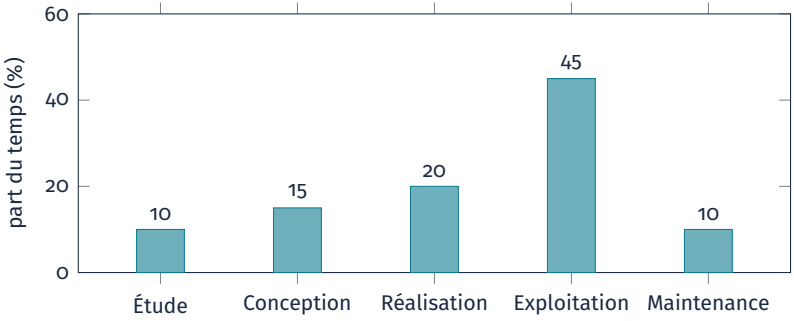
Les cinq étapes en détail

Étape	Qui ?	Entrée	Sortie
1. Étude des besoins	utilisateurs + concepteur	entretiens, documents	cahier des charges
2. Conception	concepteur	cahier des charges	MCD puis MLD
3. Réalisation	développeur + DBA	MLD	tables SQL, programmes
4. Exploitation	utilisateurs + DBA	la base en service	données réelles
5. Maintenance	tous	nouveaux besoins, bugs	schéma modifié

À retenir

Chaque étape **consomme** le résultat de la précédente : une conception bâclée se paie à toutes les étapes suivantes.

Le temps passé à chaque étape



La base sera **exploitée** pendant des années : le quart de temps consacré à l'étude et à la conception est un investissement.

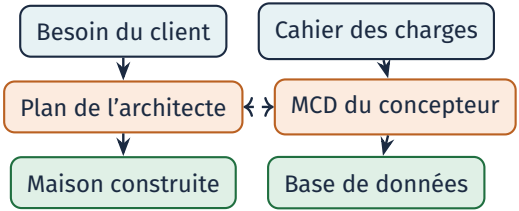
Pourquoi concevoir avant de coder : l'analogie de l'architecte

Construire une maison

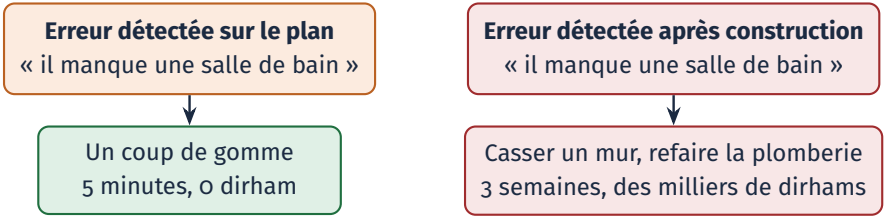
Personne ne commence par poser des briques. L'architecte dessine un **plan**, le fait **valider** par le client, puis le maçon construit.

À retenir

Le MCD est le **plan** de la base de données ; le SGBD est le chantier.

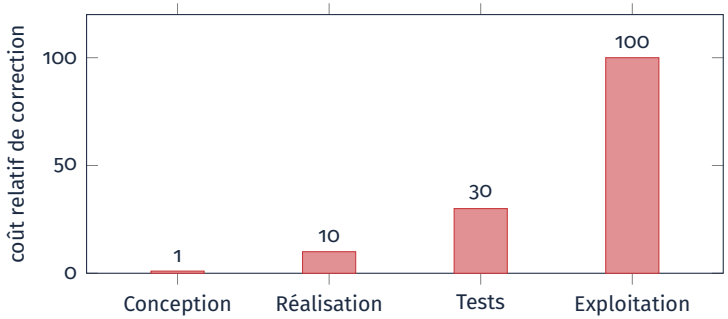


Le plan se corrige à la gomme, la maison au marteau-piqueur



Pour une base : corriger le MCD est gratuit ;
ajouter une colonne à une table remplie de 100 000 lignes et 12 programmes… ne l'est pas.

Le coût d'une erreur de conception



Une erreur qui coûte **1** à corriger sur le MCD coûte environ **100** une fois la base en service (données à migrer, programmes à réécrire, utilisateurs à reformer).

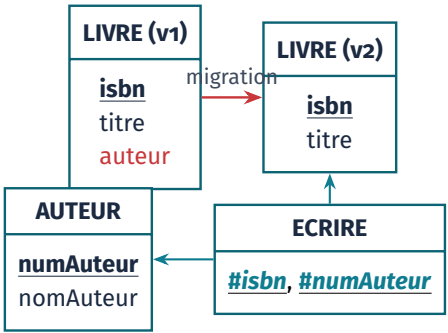
Exemple : une erreur de conception chez BiblioUni

L'erreur

Le premier développeur a créé LIVRE avec une colonne *auteur* (un seul). Six mois et 4 000 livres plus tard, on veut des livres à plusieurs auteurs.

Conséquences

Créer AUTEUR et ECRIRE, **migrer** 4 000 lignes, réécrire les écrans de saisie et de recherche.



Piège : « on verra plus tard »

⚠ Trois phrases dangereuses

- « On crée les tables maintenant, on réfléchira après. »
- « Le client ne sait pas ce qu'il veut, on devine. »
- « Un seul auteur par livre, ça suffira bien. »

La bonne attitude

- écouter et **noter** les règles de gestion ;
- dessiner le MCD et le **faire valider** ;
- se demander « et si demain… ? » avant de figer une cardinalité.

Exercice 1.7 : remettre les activités dans l'ordre

Exercice

Voici, dans le désordre, des activités d'un projet « gestion des stages » d'une école. Remettez-les dans l'ordre du cycle de vie et nommez l'étape.

a) Écrire `CREATE TABLE Stage (…)`.

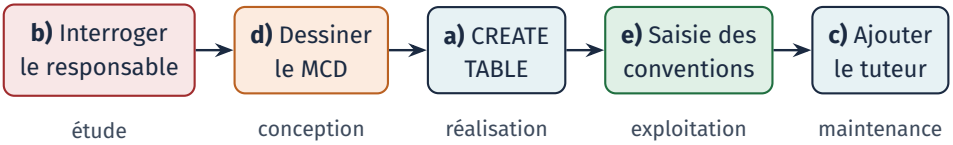
b) Interroger le responsable des stages sur ses besoins.

c) Ajouter, deux ans plus tard, la notion de « tuteur en entreprise ».

d) Dessiner le MCD ETUDIANT–EFFECTUER–STAGE–ENTREPRISE.

e) Les étudiants saisissent leurs conventions de stage en ligne.

Corrigé 1.7



✔ Solution

Ordre : **b, d, a, e, c**. Remarquez que l'activité c) (maintenance) relance un petit cycle : il faudra *re-concevoir* avant de modifier les tables.

À retenir

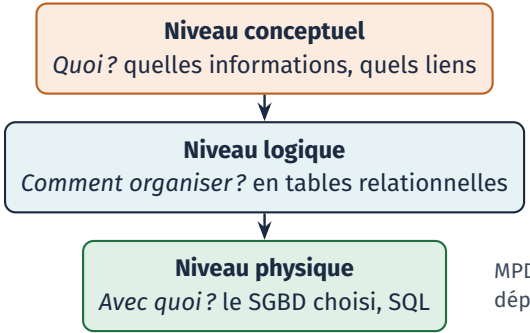
Partie 3

- Une base relationnelle = des **tables** (lignes, colonnes) reliées par des **clés**.
- Clé **primaire** : identifie une ligne ; clé **étrangère** : référence une autre table.
- Quatre acteurs : **concepteur**, **administrateur**, **développeur**, **utilisateur final**.
- Cycle de vie : étude → conception → réalisation → exploitation → maintenance.
- On conçoit **avant** de coder : une erreur de MCD coûte 1, la même erreur en exploitation coûte 100.

Partie 4 : la démarche de conception en trois niveaux

La démarche de conception en trois niveaux

Trois niveaux, trois questions



MCD : modèle conceptuel de données indépendant de toute technique

MLD : modèle logique de données dépend du type de SGBD (relationnel)

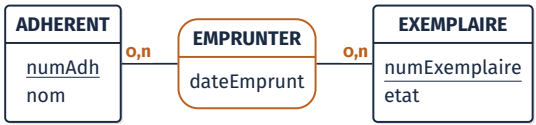
MPD : modèle physique de données dépend du SGBD précis (MySQL...)

Niveau conceptuel : le MCD

Modèle conceptuel de données

Description des informations du domaine sous forme d'**entités** (les « choses »), d'**associations** (les liens) et de **cardinalités**, *sans aucune considération technique*.

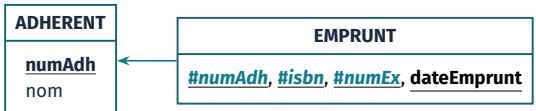
Lisible par le client : c'est le document que l'on fait **valider**.



Niveau logique : le MLD

Modèle logique de données

Traduction du MCD dans le formalisme du SGBD cible : ici, le modèle **relationnel** (tables, clés primaires, clés étrangères). Obtenu par des **règles de passage** mécaniques (chapitre 7).



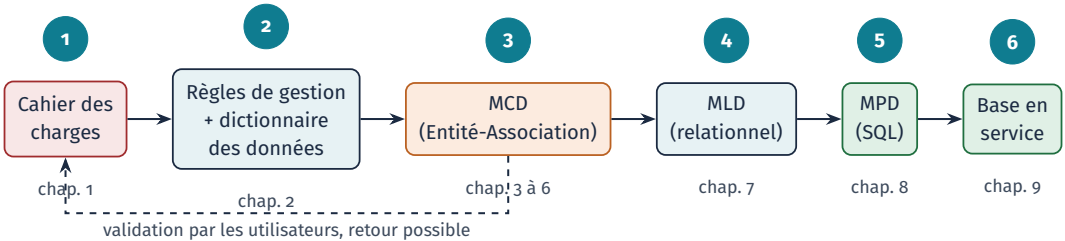
ADHERENT (**numAdh**, nom)
EMPRUNT (***#numAdh***, ***#isbn***, ***#numEx***, **dateEmprunt**)

Modèle physique de données

Le MLD complété par les **types** de colonnes, les **contraintes** et les choix propres au SGBD (index, moteur de stockage). Il s'écrit directement en **SQL**.

```sql
CREATE TABLE Adherent (
  numAdh   INT AUTO_INCREMENT
           PRIMARY KEY,
  nom      VARCHAR(50) NOT NULL
);
```

La démarche complète



Le point de départ : le cahier des charges

Extrait du cahier des charges

La bibliothèque universitaire BiblioUni souhaite informatiser la gestion de ses prêts. Chaque adhérent est identifié par un numéro, on connaît son nom, son prénom, son adresse électronique et sa date d'inscription. Un adhérent peut être étudiant ou enseignant. La bibliothèque possède des livres ; chaque livre est identifié par son ISBN, possède un titre, une année de parution et appartient à une seule catégorie (roman, science, histoire, etc.). Un livre est écrit par un ou plusieurs auteurs, et un auteur peut avoir écrit plusieurs livres. Un livre est publié par un seul éditeur. La bibliothèque peut posséder plusieurs exemplaires d'un même livre ; chaque exemplaire est numéroté à l'intérieur du livre (1, 2, 3…) et possède un état (neuf, bon, abîmé). Un adhérent emprunte des exemplaires : on enregistre la date d'emprunt et la date de retour prévue. Un exemplaire ne peut être emprunté que par un adhérent à la fois, mais il peut l'avoir été plusieurs fois au cours de son existence.

Tout ce que nous concevrons pour BiblioUni sort de ces quelques lignes. Le chapitre 2 apprendra à les **décortiquer**.

Étape 2 : règles de gestion et dictionnaire des données (aperçu)

Règles de gestion extraites

- RG1 : un livre appartient à une seule catégorie.
- RG2 : un livre est écrit par un ou plusieurs auteurs.
- RG3 : un exemplaire n'est prêté qu'à un adhérent à la fois.

Donnée	Type	Commentaire
numAdh	entier	identifiant
nom	texte(50)	
isbn	texte(13)	identifiant
titre	texte(100)	
etat	texte	neuf, bon, abîmé
dateEmprunt	date	

Récapitulatif des trois niveaux

Niveau	Modèle	Vocabulaire	Dépend de	Lecteur
Conceptuel	MCD	entité, association, cardinalité, identifiant	rien	client, concepteur
Logique	MLD	table, clé primaire, clé étrangère	type de SGBD (relationnel)	concepteur, développeur
Physique	MPD	CREATE TABLE, INT, VARCHAR, index	SGBD précis (MySQL…)	développeur, DBA

À retenir

En descendant les niveaux, on **ajoute** de la technique et l'on **perd** de l'indépendance. On ne remonte jamais : on ne « devine » pas le MCD à partir des tables.

Piège : mélanger les niveaux

⚠ Symptômes d'un mélange

- écrire « VARCHAR(50) » dans un MCD ;
- mettre une clé étrangère dans une entité ;
- parler de « table » quand on est au niveau conceptuel ;
- dessiner un MCD en copiant des tables existantes.



✗ type SQL dans un MCD
✗ clé étrangère dans une entité



✓ conceptuel : le lien vers la catégorie sera une association

Exercice 1.8 : à quel niveau appartient chaque phrase ?

Exercice

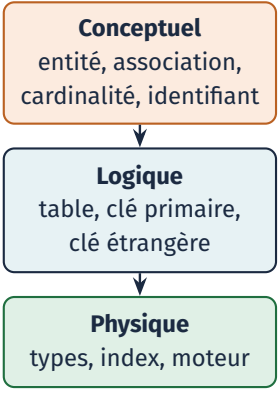
Conceptuel, logique ou physique ?

1. « Un adhérent emprunte zéro ou plusieurs exemplaires. »
2. « La table EMPRUNT contient la clé étrangère numAdh. »
3. « La colonne email est un VARCHAR(100) avec un index. »
4. « L'entité LIVRE a pour identifiant isbn. »
5. « On choisit le moteur InnoDB de MySQL. »
6. « ECRIRE(***#isbn***, ***#numAuteur***) »

Corrigé 1.8

✔ Solution

1. cardinalité → **conceptuel**.
2. clé étrangère → **logique**.
3. VARCHAR, index → **physique**.
4. entité, identifiant → **conceptuel**.
5. moteur InnoDB → **physique**.
6. notation relationnelle → **logique**.



Merise, UML et aperçu du fil rouge

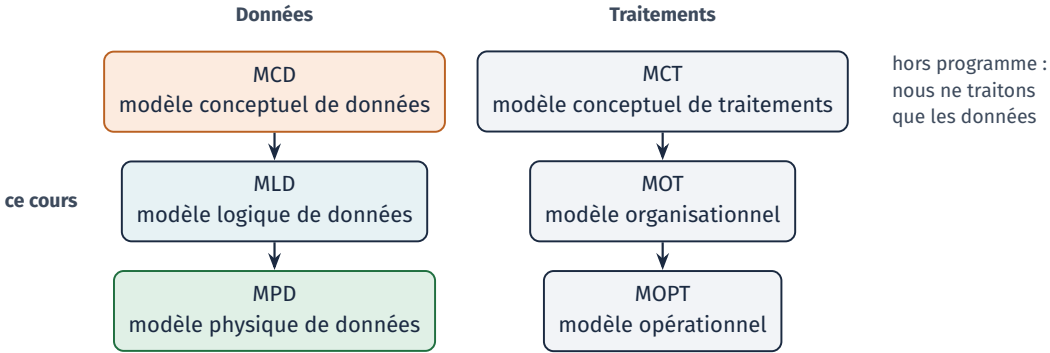
Merise : une méthode française née en 1978



Merise

Méthode d'analyse et de conception des systèmes d'information qui sépare les **données** des **traitements** et distingue les niveaux **conceptuel**, **logique** et **physique**.

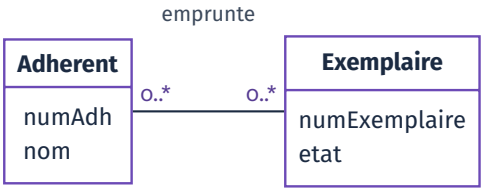
Les modèles de Merise



UML : le diagramme de classes

UML

Unified Modeling Language : notation internationale (1997) issue de la programmation objet. Son **diagramme de classes** joue le rôle du MCD : classes, attributs, associations et **multiplicités**.



Les multiplicités UML (0..*, 1) se lisent *à l'envers* des cardinalités Merise : on les écrit du côté opposé.

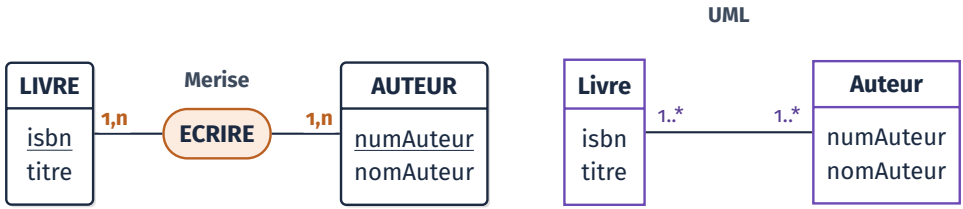
Merise et UML : correspondance

Merise (MCD)	UML (diagramme de classes)	Exemple
entité	classe	ADHERENT / Adherent
attribut	attribut	nom
identifiant (souligné)	attribut {id} ou contrainte	numAdh
association (ovale)	association (trait)	EMPRUNTER
association porteuse	classe-association	dateEmprunt
cardinalité (0,n) près de l'entité	multiplicité 0..* du côté *opposé*	un adhérent, plusieurs emprunts

À retenir

Même raisonnement, deux dessins. Qui sait lire l'un lit l'autre en dix minutes.

Le même lien dans les deux notations



Un livre a **1,n** auteurs ; en UML, le « 1..* » est écrit près d'*Auteur*, à l'autre bout du trait.

Pourquoi Merise dans ce cours ?

Arguments

- cardinalités **(min, max)** très explicites, idéales pour débuter ;
- règles de passage MCD $\rightarrow$ MLD simples et mécaniques ;
- très répandue dans les entreprises et l'enseignement francophones ;
- outils gratuits (Looping, JMerise).

À retenir

Nous **raisonnons** en Merise et nous montrons régulièrement l'**équivalent UML** pour que vous soyez à l'aise avec les deux.

Où allons-nous ? Aperçu de la destination sur BiblioUni

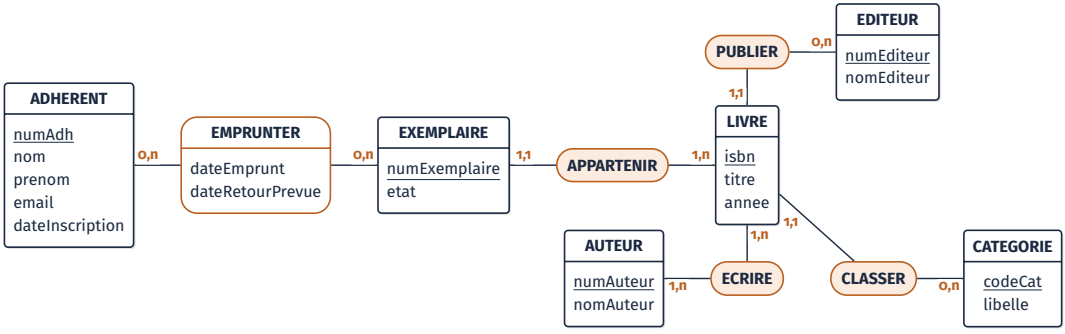
Rappel du cahier des charges BiblioUni

Extrait du cahier des charges

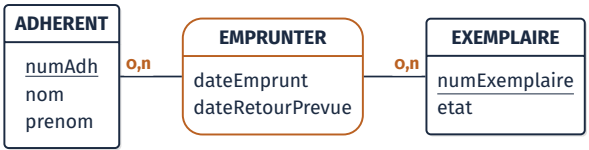
La bibliothèque universitaire BiblioUni souhaite informatiser la gestion de ses prêts. Chaque adhérent est identifié par un numéro, on connaît son nom, son prénom, son adresse électronique et sa date d'inscription. Un adhérent peut être étudiant ou enseignant. La bibliothèque possède des livres ; chaque livre est identifié par son ISBN, possède un titre, une année de parution et appartient à une seule catégorie (roman, science, histoire, etc.). Un livre est écrit par un ou plusieurs auteurs, et un auteur peut avoir écrit plusieurs livres. Un livre est publié par un seul éditeur. La bibliothèque peut posséder plusieurs exemplaires d'un même livre ; chaque exemplaire est numéroté à l'intérieur du livre (1, 2, 3…) et possède un état (neuf, bon, abîmé). Un adhérent emprunte des exemplaires : on enregistre la date d'emprunt et la date de retour prévue. Un exemplaire ne peut être emprunté que par un adhérent à la fois, mais il peut l'avoir été plusieurs fois au cours de son existence.

Nous allons voir, *sans les détailler*, le MCD, le MLD et le SQL que ce texte produira à la fin du cours.

Destination 1 : le MCD de BiblioUni (aperçu)



Lire un morceau du MCD

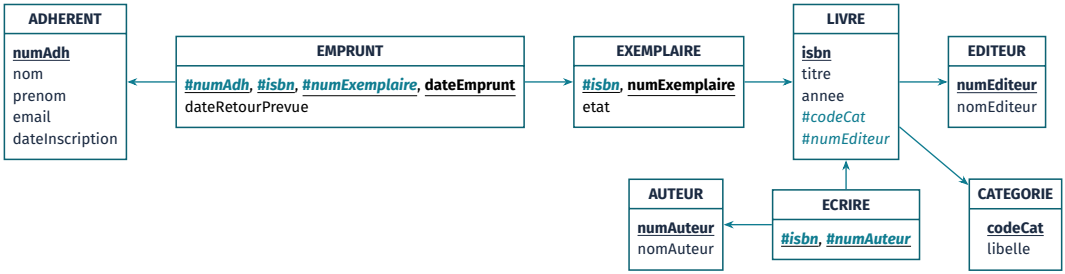


Lecture en français

- deux **entités** : ADHERENT et EXEMPLAIRE ;
- une **association** EMPRUNTER qui *porte* les dates ;
- **0,n** : un adhérent emprunte de 0 à plusieurs exemplaires ;
- **0,n** : un exemplaire est emprunté de 0 à plusieurs fois.

Chapitres 3 à 5 : vous saurez construire cela vous-même.

Destination 2 : le MLD de BiblioUni (aperçu)



Le même MLD en notation textuelle

- **ADHERENT** (**numAdh**, nom, prenom, email, dateInscription)
- **CATEGORIE** (**codeCat**, libelle)
- **EDITEUR** (**numEditeur**, nomEditeur)
- **LIVRE** (**isbn**, titre, annee, *#codeCat*, *#numEditeur*)
- **AUTEUR** (**numAuteur**, nomAuteur)
- **ECRIRE** (***#isbn***, ***#numAuteur***)
- **EXEMPLAIRE** (***#isbn***, **numExemplaire**, etat)
- **EMPRUNT** (***#numAdh***, ***#isbn***, ***#numExemplaire***, **dateEmprunt**, dateRetourPrevue)

À retenir

Huit tables, obtenues **mécaniquement** à partir du MCD grâce aux règles de passage (chapitre 7).

Destination 3 : le MPD, en SQL (aperçu)

```sql
CREATE TABLE Categorie (
  codeCat  INT PRIMARY KEY,
  libelle  VARCHAR(40) NOT NULL
);

CREATE TABLE Livre (
  isbn     VARCHAR(13) PRIMARY KEY,
  titre    VARCHAR(100) NOT NULL,
  annee    INT,
  codeCat  INT NOT NULL,
  FOREIGN KEY (codeCat)
    REFERENCES Categorie(codeCat)
);
```

Ce qu'il faut voir

- une table par relation du MLD ;
- la clé primaire devient `PRIMARY KEY` ;
- la clé étrangère devient `FOREIGN KEY … REFERENCES` ;
- chaque colonne reçoit un **type**.

Chapitre 8.

Et la récompense : interroger la base

Question

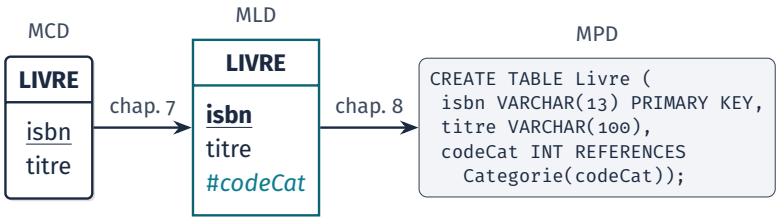
Quels adhérents ont un livre en retard aujourd'hui ?

```
SELECT DISTINCT a.nom, a.prenom
FROM   Adherent a
  JOIN Emprunt e ON e.numAdh = a.numAdh
WHERE  e.dateRetourPrevue < CURRENT_DATE;
```

À retenir

Quatre lignes de SQL répondent à une question qui demandait un programme entier avec des fichiers. C'est **possible parce que le schéma est bien conçu**.

Les trois modèles côte à côte



Même « chose » du monde réel (le livre), trois représentations de plus en plus techniques.

Ce qu'il faut savoir maintenant, et plus tard

Notion	Dès maintenant	Plus tard
Entité / association	reconnaître les rectangles et les ovales	choisir, nommer, identifier (chap. 3)
Cardinalités	lire « 0,n » comme « de zéro à plusieurs »	les déterminer (chap. 4)
Cas particuliers	savoir qu'ils existent	porteuse, réflexive, héritage (chap. 5)
Normalisation	savoir que la redondance est un problème	1FN, 2FN, 3FN (chap. 6)
Passage au MLD	savoir que c'est mécanique	les règles (chap. 7)
SQL	lire un CREATE TABLE	l'écrire (chap. 8)

Exercice 1.9 : reconnaître les modèles

Exercice

Trois documents ont été trouvés dans le dossier du projet BiblioUni. Pour chacun, dites de quel modèle il s'agit et quel acteur l'a probablement produit.

Document A

```
CREATE TABLE Auteur (
 numAuteur INT PRIMARY KEY,
 nomAuteur VARCHAR(60));
```

Document B



Document C



Corrigé 1.9

✔ Solution

- **A** : SQL, donc **MPD** (niveau physique), produit par le développeur ou le DBA.
- **B** : table avec clé primaire soulignée, donc **MLD** (niveau logique), produit par le concepteur.
- **C** : entité avec identifiant souligné, donc **MCD** (niveau conceptuel), produit par le concepteur avec les utilisateurs.

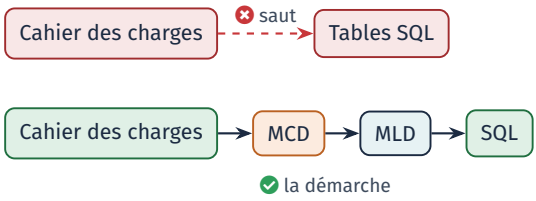
⚠ Indices visuels

- mots-clés en majuscules et parenthèses → SQL ;
- **souligné gras**, *#dièse* → MLD ;
- rectangle bleu à deux parties, ovales orange → MCD.

Piège : commencer par les tables

⚠ Le réflexe du débutant

Ouvrir MySQL Workbench et créer des tables « au feeling ». Résultat : des colonnes oubliées, des liens impossibles, des clés étrangères dans le mauvais sens.



À retenir

Partie 4

- Trois niveaux : **conceptuel** (MCD, *quoi ?*), **logique** (MLD, *comment organiser ?*), **physique** (MPD, *avec quel SGBD ?*).
- La démarche : cahier des charges $\rightarrow$ règles de gestion $\rightarrow$ MCD $\rightarrow$ MLD $\rightarrow$ MPD/SQL.
- **Merise** et **UML** expriment la même chose avec deux notations.
- Ne jamais mélanger les niveaux, ne jamais commencer par les tables.
- Le MCD de BiblioUni (7 entités) donnera 8 tables SQL : c'est la destination du cours.

Partie 5 : mini-projet guidé, nommage et outils

Mini-projet guidé : le club de sport

Le besoin

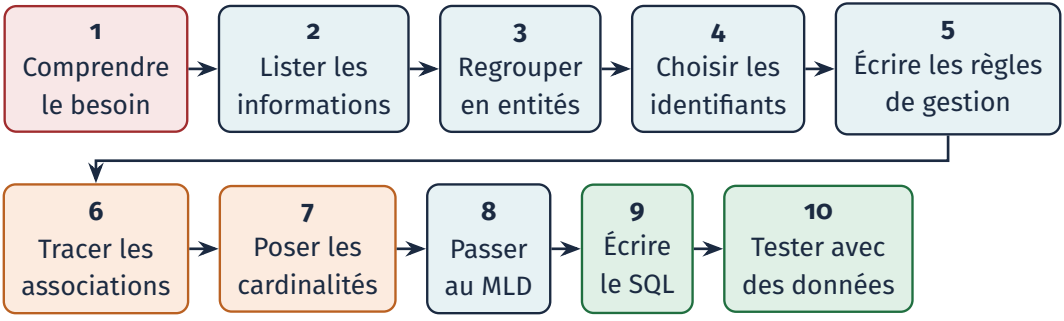
Demande du président du club

Nous voulons gérer les inscriptions à notre club de sport. Chaque membre a un nom, un prénom, une date de naissance et un téléphone. Le club propose plusieurs activités (football, natation, judo…), chacune ayant un tarif annuel. Un membre peut s'inscrire à plusieurs activités ; pour chaque inscription, nous notons la date. Chaque activité est encadrée par un seul entraîneur, mais un entraîneur peut encadrer plusieurs activités.

Question

Comment passer de ce paragraphe à des tables ? En **dix étapes**, en observant à chaque fois *ce que produit le concepteur*.

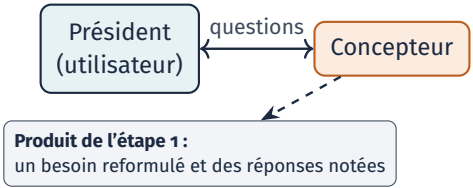
La démarche en dix étapes



Étape 1 : comprendre le besoin

Le concepteur interroge le président

- Un membre peut-il s'inscrire deux fois à la même activité ? *Non, une fois par saison.*
- Une activité peut-elle exister sans entraîneur ? *Non.*
- Deux membres peuvent-ils avoir le même nom ? *Oui, bien sûr.*



Étape 2 : lister les informations

On souligne chaque nom dans le texte

nom, prénom, date de naissance, téléphone, nom de l'activité, tarif annuel, date d'inscription, nom de l'entraîneur.

Produit de l'étape 2 : la première version du **dictionnaire des données**.

Donnée	Type	Parle de...
nom	texte	membre
prenom	texte	membre
dateNaissance	date	membre
telephone	texte	membre
nomActivite	texte	activité
tarifAnnuel	décimal	activité
dateInscription	date	inscription
nomEntraineur	texte	entraîneur

Étape 3 : regrouper en entités

Comment regrouper ?

Les données qui parlent de la **même chose** forment une entité. La colonne « parle de… » du dictionnaire donne directement les entités.

Produit de l'étape 3 : trois entités (l'inscription attendra : c'est un *lien*).







Étape 4 : choisir les identifiants

Règle

Chaque entité doit avoir un **identifiant** : unique, jamais vide, stable. Si aucune donnée naturelle ne convient (deux membres peuvent avoir le même nom), on **crée un numéro**.

Produit de l'étape 4 : trois entités identifiées.

MEMBRE
numMembre
nom
prenom
dateNaissance
telephone

ACTIVITE
codeActivite
nomActivite
tarifAnnuel

ENTRAINEUR
numEntraineur
nomEntraineur

Étape 5 : écrire les règles de gestion

Règles de gestion du club

- RG1 : un membre s'inscrit à une ou plusieurs activités.
- RG2 : une activité accueille zéro ou plusieurs membres.
- RG3 : chaque inscription a une date.
- RG4 : une activité est encadrée par *un seul* entraîneur.
- RG5 : un entraîneur encadre *une ou plusieurs* activités.

À retenir

Chaque règle de gestion est une **phrase courte** qui relie deux entités et dit *combien*. Elle sera validée par le président avant de continuer.

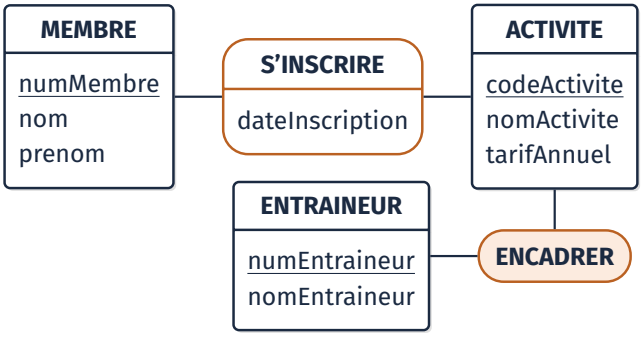
Produit de l'étape 5 : une liste de règles numérotées.

Étape 6 : tracer les associations

Règle

Chaque règle de gestion qui relie deux entités devient une **association** (verbe à l'infinitif). RG3 dit que l'inscription porte une date : l'association S'INSCRIRE **porte** *dateInscription*.

Produit de l'étape 6 : le squelette du MCD.

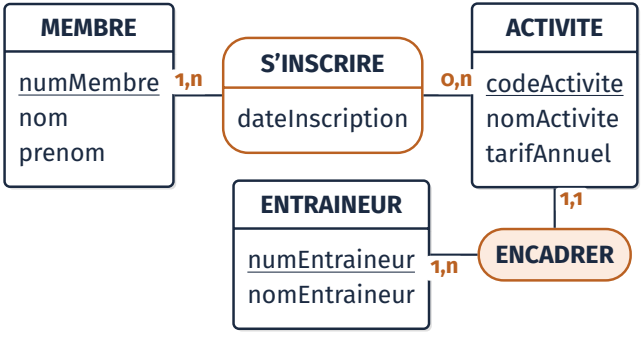


Étape 7 : poser les cardinalités

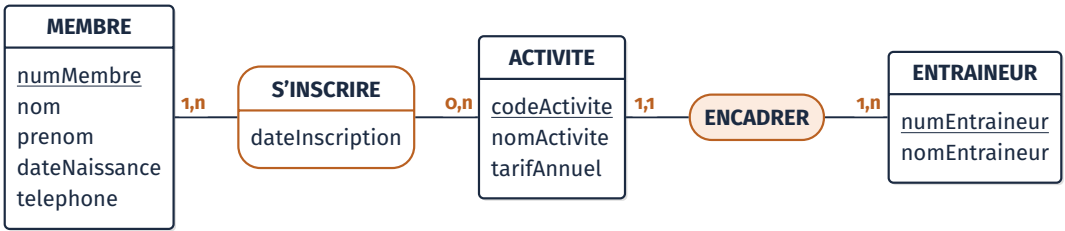
Lecture

Pour *une* occurrence de l'entité, combien de fois participe-t-elle à l'association, au minimum et au maximum ?

- un membre : **1,n** activités (RG1) ;
- une activité : **0,n** membres (RG2) ;
- une activité : **1,1** entraîneur (RG4) ;
- un entraîneur : **1,n** activités (RG5).



Le MCD complet du club de sport



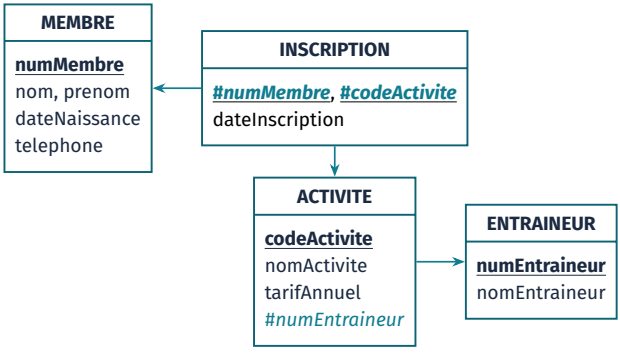
À retenir

Produit de l'étape 7 : le MCD validé par le président. C'est le document le plus important du projet.

Étape 8 : passer au MLD

Deux règles suffisent ici

- côté **1,1** (ACTIVITE–ENCADRER) : la clé de l'entraîneur **migre** dans ACTIVITE ;
- **n,n** des deux côtés (S'INSCRIRE) : **nouvelle table** avec les deux clés + la date.



Produit de l'étape 8 : quatre relations.

Étape 9 : écrire le SQL

```sql
CREATE TABLE Entraineur (
  numEntraineur INT PRIMARY KEY,
  nomEntraineur VARCHAR(60) NOT NULL
);
CREATE TABLE Activite (
  codeActivite  INT PRIMARY KEY,
  nomActivite   VARCHAR(40) NOT NULL,
  tarifAnnuel   DECIMAL(8,2),
  numEntraineur INT NOT NULL,
  FOREIGN KEY (numEntraineur)
    REFERENCES Entraineur(numEntraineur)
);
```

```sql
CREATE TABLE Membre (
  numMembre     INT PRIMARY KEY,
  nom           VARCHAR(50) NOT NULL,
  prenom        VARCHAR(50),
  dateNaissance DATE,
  telephone     VARCHAR(15)
);
CREATE TABLE Inscription (
  numMembre       INT,
  codeActivite    INT,
  dateInscription DATE,
  PRIMARY KEY (numMembre, codeActivite),
  FOREIGN KEY (numMembre)
    REFERENCES Membre(numMembre),
  FOREIGN KEY (codeActivite)
    REFERENCES Activite(codeActivite)
);
```

Étape 10 : tester avec des données

numMembre	nom	prenom
1	Alaoui	Nadia
2	Berrada	Omar

codeActivite	nomActivite	tarifAnnuel	#numEntraineur
10	Football	600	7
11	Natation	800	7
12	Judo	500	8

#numMembre	#codeActivite	dateInscription
1	10	02/09/2026
1	12	02/09/2026
2	11	05/09/2026

Vérifications

Nadia est inscrite à deux activités ✅ ; Football et Natation ont le même entraîneur (7) sans recopier son nom ✅ ; impossible d'inscrire Nadia deux fois au football (clé composée) ✅.

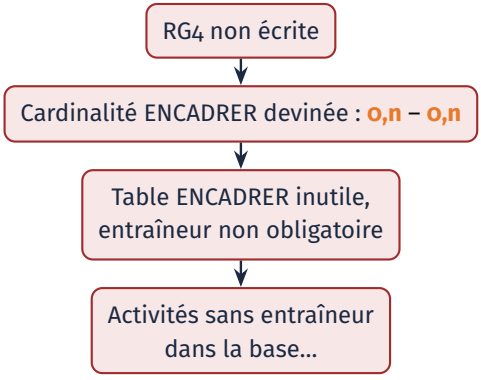
Bilan : ce que le concepteur a produit

Étape	Activité	Produit (livrable)
1	comprendre le besoin	besoin reformulé, réponses aux questions
2	lister les informations	dictionnaire des données
3–4	regrouper, identifier	entités avec identifiants
5	écrire les règles	règles de gestion numérotées
6–7	associations, cardinalités	**MCD validé**
8	passage au logique	MLD (relations)
9	SQL	script CREATE TABLE (MPD)
10	test	jeu de données, vérification des règles

Piège : sauter des étapes

⚠ Ce qui arrive quand on saute…

- l'étape 1 : on modélise ce que l'on *croit* au lieu de ce qui est ;
- l'étape 5 : les cardinalités sont devinées ;
- l'étape 7 : le MLD est faux (clé étrangère du mauvais côté) ;
- l'étape 10 : les erreurs se découvrent en exploitation.



Exercice 1.10 : ajouter les cotisations

Exercice

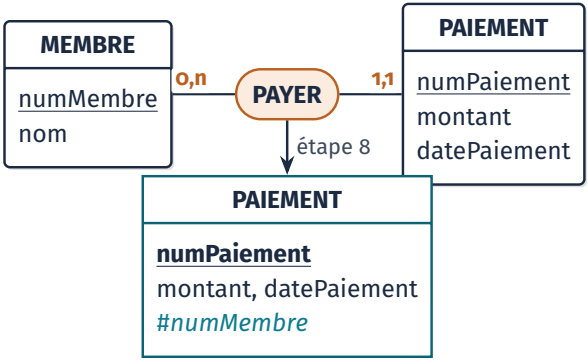
Le président ajoute : « *Chaque membre paie une cotisation annuelle au club (indépendante des activités); nous voulons conserver l'historique : montant et date de chaque paiement.* »

1. Quelles nouvelles données apparaissent ? (étape 2)
2. Faut-il une nouvelle entité ? Avec quel identifiant ? (étapes 3–4)
3. Écrivez la règle de gestion et les cardinalités. (étapes 5–7)
4. Donnez la relation MLD obtenue. (étape 8)

Corrigé 1.10

✔ Solution

1. montant, datePaiement.
2. Oui : PAIEMENT (plusieurs paiements par membre, à conserver). Identifiant créé : numPaiement.
3. RG6 : un membre effectue **0,n** paiements ; un paiement concerne **1,1** membre.
4. **PAIEMENT** (**numPaiement**, montant, datePaiement, *#numMembre*)



À retenir : le mini-projet

Dix étapes, quatre livrables

- Le concepteur **écoute** (1), **liste** (2), **regroupe** (3–4), **formule des règles** (5).
- Il **dessine** le MCD (6–7) et le fait **valider**.
- Il **traduit** mécaniquement en MLD (8) puis en SQL (9).
- Il **teste** avec des données réelles (10).
- Livrables : dictionnaire, règles de gestion, MCD, MLD/SQL.

Nommage et outils du concepteur

Pourquoi nommer proprement ?

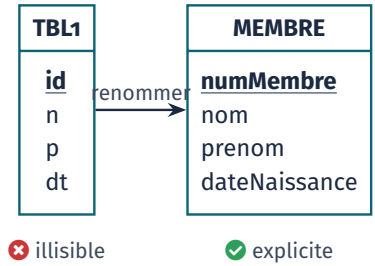
Deux versions de la même table

✗ `tbl1(id, n, p, dt, tel)`

✓ `MEMBRE(numMembre, nom, prenom, dateNaissance, telephone)`

À retenir

Un schéma est lu par le client, le développeur, le DBA… et par vous dans six mois. Les noms sont la **documentation** la moins chère qui existe.



Règles de nommage adoptées dans ce cours

Entités, associations, tables

- entité et table : nom commun **au singulier**, en MAJUSCULES : MEMBRE, ACTIVITE ;
- association : **verbe à l'infinitif** : S'INSCRIRE, ENCADRER ;
- pas d'accent, pas d'espace, pas de tiret.

Attributs, colonnes

- **camelCase** : dateNaissance, tarifAnnuel ;
- identifiant préfixé : numMembre, codeActivite, isbn ;
- nom **unique** dans tout le schéma (pas deux « nom ») : nomMembre, nomActivite ;
- pas de mot réservé SQL : *date*, *order*, *user*.

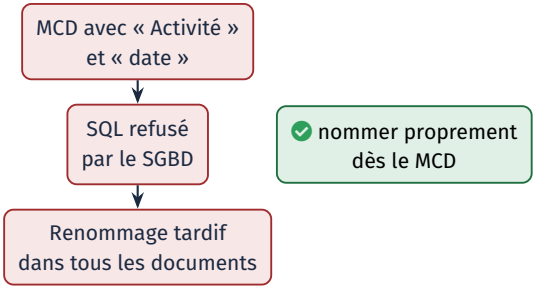
Bons et mauvais noms

Mauvais	Pourquoi	Bon
✗ Les membres	espace, pluriel	✓ MEMBRE
✗ Activité	accent	✓ ACTIVITE
✗ date	mot réservé SQL, trop vague	✓ dateInscription
✗ date_de_naissance	tirets bas (acceptés mais lourds)	✓ dateNaissance
✗ nom (dans MEMBRE et ENTRAINEUR)	ambigu lors des requêtes	✓ nomMembre, nomEntraineur
✗ MEMBRE_ACTIVITE (association)	pas un verbe	✓ S'INSCRIRE
✗ id	lequel ?	✓ numMembre

Piège : accents, espaces et mots réservés

⚠ Ce qui casse à l'étape 9

- `CREATE TABLE Activité` : fonctionne parfois, puis casse sur un autre serveur ;
- `date DATE` : erreur de syntaxe sur plusieurs SGBD ;
- `nom prenom` avec un espace : lu comme deux mots.



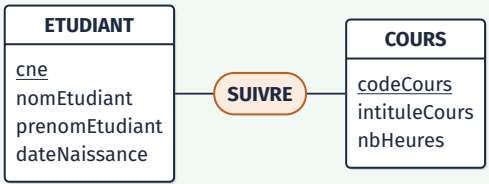
Exercice 1.11 : corriger les noms

Exercice

Corrigez ce fragment de MCD selon les règles du cours.



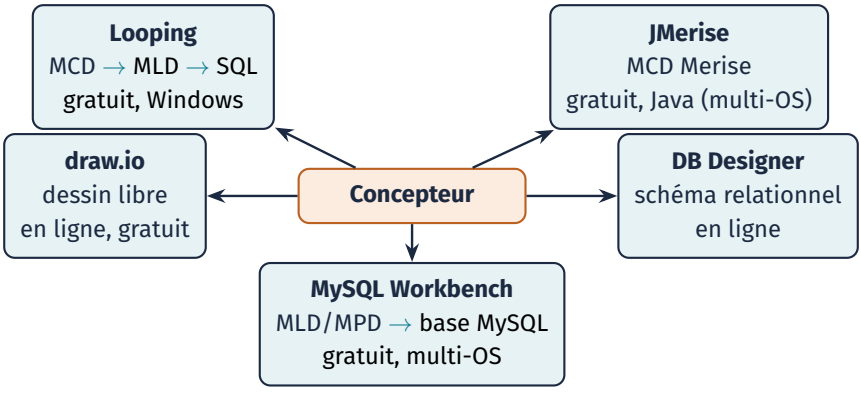
Solution



Corrigé 1.11 : les corrections une par une

Avant	Problème	Après
Les Étudiants	article, pluriel, accent, espace	ETUDIANT
id (deux fois)	même nom dans deux entités, non parlant	cne, codeCours
Nom (deux fois)	ambigu, majuscule initiale	nomEtudiant, intituleCours
Prénom	accent	prenomEtudiant
date	mot réservé, trop vague	dateNaissance
ETUDIANT_COURS	pas un verbe	SUIVRE
cours	minuscules pour une entité	COURS
nb heures	espace	nbHeures

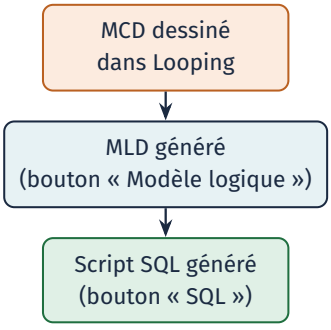
Les outils du concepteur : panorama



Looping

Looping (Patrick Bergougnoux)

- logiciel gratuit, très utilisé en enseignement ;
- on dessine le MCD en Merise (entités, associations, cardinalités) ;
- génère **automatiquement** le MLD et le script SQL pour MySQL, PostgreSQL, Oracle… ;
- vérifie la cohérence du modèle.



c'est l'outil des TP

JMerise

- gratuit, écrit en Java : Windows, macOS, Linux ;
- dédié au MCD Merise, avec le dictionnaire des données intégré ;
- passage au MLD et génération SQL ;
- gère les cas particuliers (héritage, identifiant relatif).

À retenir

Looping ou JMerise : peu importe. Ce qui compte, c'est que l'outil **respecte la notation Merise** et **génère** le MLD, pour que vous vérifiiez vos règles de passage.

draw.io et DB Designer

draw.io (diagrams.net)

- outil de dessin généraliste, en ligne ;
- bibliothèque de formes « Entity Relation » et UML ;
- aucune vérification, aucune génération : c'est un **crayon**.

DB Designer, dbdiagram.io

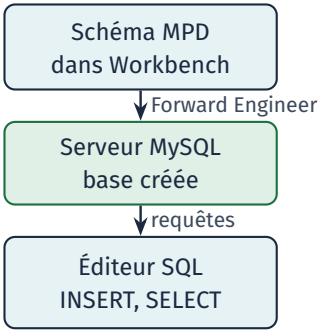
- en ligne, gratuits pour l'essentiel ;
- travaillent au niveau **logique** (tables, clés étrangères) ;
- exportent le SQL ; pratiques pour partager un MLD en équipe.

MCD : Looping, JMerise, draw.io → MLD : DB Designer, dbdiagram → MPD : MySQL Workbench

MySQL Workbench

MySQL Workbench

- l'outil officiel de MySQL, gratuit ;
- éditeur de schéma **physique** (tables, types, index) ;
- « Forward Engineer » : crée la base sur le serveur ;
- « Reverse Engineer » : dessine le schéma d'une base existante ;
- éditeur SQL pour les requêtes.



Comparatif des outils

Outil	Niveau	Merise	Génère MLD	Génère SQL	Prix
Looping	MCD	✅	✅	✅	gratuit
JMerise	MCD	✅	✅	✅	gratuit
draw.io	dessin	✅ (formes)	❌	❌	gratuit
DB Designer	MLD	❌	—	✅	gratuit / payant
MySQL Workbench	MPD	❌	—	✅	gratuit

À retenir

Dans ce cours : **Looping** (ou JMerise) pour le MCD, **MySQL Workbench** pour créer et interroger la base. Le papier et le crayon restent les meilleurs outils pour *réfléchir*.

À retenir

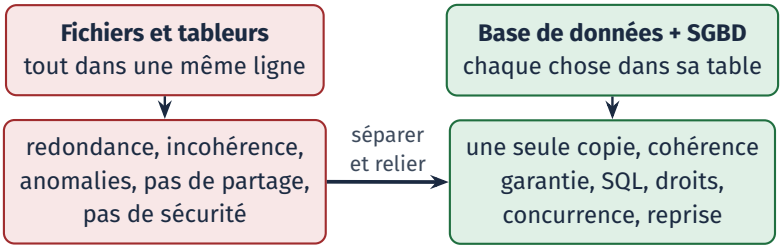
Partie 5

- Un projet, même petit, suit les **dix étapes** : du besoin aux données de test.
- À chaque étape, un **livrable** : dictionnaire, règles de gestion, MCD, MLD, SQL.
- Nommer : entités en MAJUSCULES au singulier, associations à l'**infinitif**, attributs en **camelCase**, sans accent, sans espace, sans mot réservé.
- Outils : Looping / JMerise (MCD), DB Designer (MLD), MySQL Workbench (MPD).

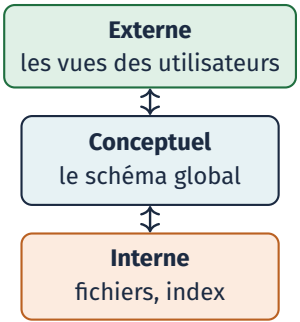
Partie 6 : synthèse, glossaire, QCM et exercices

Synthèse et glossaire

Synthèse 1 : pourquoi une base de données ?



Synthèse 2 : le SGBD et ses trois niveaux

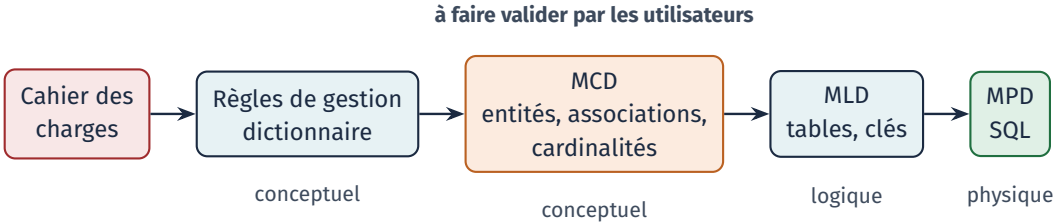


SGBD
définition
manipulation
intégrité
sécurité
concurrence
reprise

MySQL, PostgreSQL, Oracle, SQL Server, SQLite

tous parlent **SQL**

Synthèse 3 : la démarche de conception



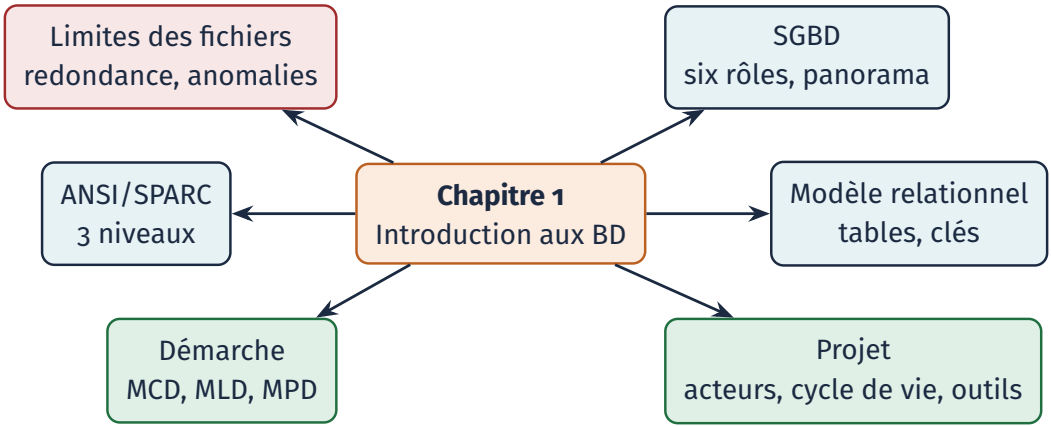
Concevoir avant de coder : une erreur corrigée sur le MCD coûte 1 ; la même erreur en exploitation coûte 100.

Fiche à retenir

Chapitre 1 en huit lignes

1. Une organisation doit **stocker, retrouver et partager** de l'information.
2. Les fichiers et tableurs créent **redondance, incohérence et anomalies**.
3. Une **base de données** est structurée, cohérente, persistante et partagée ; le **SGBD** la gère.
4. Six rôles du SGBD : définition, manipulation, intégrité, sécurité, concurrence, reprise.
5. Architecture **ANSI/SPARC** : externe, conceptuel, interne ; indépendance logique et physique.
6. Tables, lignes, colonnes, **clé primaire**, **clé étrangère**.
7. Quatre acteurs, cinq étapes de cycle de vie ; le concepteur travaille **avant** le développeur.
8. Démarche : cahier des charges $\rightarrow$ MCD $\rightarrow$ MLD $\rightarrow$ MPD (Merise ; UML équivalent).

Carte mentale du chapitre



Glossaire (1/3) : les fondations

Terme	Définition
Donnée	Élément brut (nombre, texte, date) sans interprétation.
Information	Donnée placée dans un contexte qui lui donne un sens.
Système d'information	Moyens humains et techniques pour collecter, stocker, traiter et diffuser l'information.
Base de données	Ensemble structuré, cohérent, persistant et partagé de données décrivant un domaine.
SGBD	Logiciel qui gère la base : définition, manipulation, intégrité, sécurité, concurrence, reprise.
Redondance	Répétition inutile d'une même information.
Incohérence	Contradiction entre deux copies d'une même information.
Anomalie	Effet indésirable d'une insertion, d'une mise à jour ou d'une suppression, dû à une mauvaise structure.

Glossaire (2/3) : le modèle relationnel

Terme	Définition
Table (relation)	Ensemble de lignes de même structure décrivant des individus de même nature.
Ligne (tuple)	Description d'un individu : une occurrence.
Colonne (attribut)	Propriété des individus, avec un nom et un type.
Clé primaire	Colonne(s) identifiant chaque ligne de façon unique et non vide.
Clé étrangère	Colonne qui référence la clé primaire d'une autre table.
SQL	Langage standard pour définir et manipuler les bases relationnelles.
Niveau externe / conceptuel / interne	Les trois niveaux de description d'une base (ANSI/SPARC).
Indépendance des données	Possibilité de modifier un niveau sans toucher aux autres.

Glossaire (3/3) : la conception

Terme	Définition
Cahier des charges	Document décrivant les besoins des utilisateurs en langage courant.
Règle de gestion	Phrase courte exprimant une contrainte du domaine (« un livre a une seule catégorie »).
Dictionnaire des données	Liste de toutes les données avec leur type et leur signification.
MCD	Modèle conceptuel de données : entités, associations, cardinalités (Merise).
MLD	Modèle logique de données : tables, clés primaires et étrangères.
MPD	Modèle physique de données : le script SQL adapté au SGBD choisi.
Merise	Méthode française (1978) de conception des systèmes d'information.
UML	Notation internationale (1997); le diagramme de classes équivaut au MCD.

QCM d’auto-évaluation

QCM : huit questions, une seule bonne réponse par question

Questions 1 et 2

Q1. Une même adresse e-mail saisie dans dix lignes d'emprunt, c'est…

a) une anomalie de suppression ;
b) de la redondance ;
c) de l'indépendance physique.

Q2. Le SGBD, c'est…

a) l'ensemble des données de l'organisation ;
b) le logiciel qui gère les données ;
c) le schéma dessiné par le concepteur.

Questions 3 et 4

Q3. Ajouter un index sur la colonne *titre* relève du niveau…

a) externe ;
b) conceptuel ;
c) interne.

Q4. Dans la table EMPRUNT, la colonne #numAdh est…

a) une clé primaire ;
b) une clé étrangère vers ADHERENT ;
c) une cardinalité.

Questions 5 et 6

Q5. Le premier document produit par le concepteur est…

a) le script CREATE TABLE ;
b) le MLD ;
c) les règles de gestion et le MCD.

Q6. Un MCD contenant « VARCHAR(50) » est…

a) correct : il faut préciser les types ;
b) incorrect : il mélange les niveaux conceptuel et physique ;
c) correct uniquement pour MySQL.

Questions 7 et 8

Q7. Deux guichets prêtent le même exemplaire à la même seconde ; le SGBD …

a) enregistre les deux prêts ;
b) sérialise les opérations et refuse la seconde (contrôle de concurrence) ;
c) supprime l'exemplaire.

Q8. La multiplicité UML « 1..* » correspond, en Merise, à la cardinalité…

a) **1,n** écrite du côté opposé du trait ;
b) **0,1** ;
c) **1,1**.

Réponses aux questions 1 à 4

✔ Solution

- **Q1 : b)** redondance. L'e-mail devrait être stocké une seule fois dans ADHERENT.
- **Q2 : b)** le SGBD est le logiciel ; les données forment la base ; le schéma est le MCD/MLD.
- **Q3 : c)** interne : l'index est un choix de stockage, invisible au niveau conceptuel (indépendance physique).
- **Q4 : b)** clé étrangère : elle référence la clé primaire numAdh de la table ADHERENT.

Réponses aux questions 5 à 8

✔ Solution

- **Q5 : c)** on commence par l'analyse (règles de gestion) et le MCD ; le SQL vient en dernier.
- **Q6 : b)** les types SQL appartiennent au MPD ; le MCD reste indépendant de toute technique.
- **Q7 : b)** contrôle de concurrence : verrou sur l'exemplaire, la seconde opération est refusée.
- **Q8 : a)** `1,n` ; en UML la multiplicité s'écrit près de l'entité opposée.

À retenir

Moins de 6 bonnes réponses ? Relisez la fiche « à retenir » et refaites les exercices 1.1, 1.4 et 1.8.

Exercices supplémentaires

Exercice S1 : la feuille « Commandes » d'une boutique

Exercice

Une boutique tient la feuille ci-dessous. Relevez : (a) deux redondances ; (b) une incohérence ; (c) une anomalie de suppression ; (d) proposez des tables séparées.

NumCde	Date	Client	Ville	Produit	PrixUnit	Qté	Total
5081	18/09/2026	Sara Idrissi	Meknès	Clavier USB	150	2	300
5081	18/09/2026	Sara Idrissi	Meknès	Souris	80	1	80
5082	19/09/2026	Omar Berrada	Fès	Clavier USB	150	1	150
5083	20/09/2026	Sara Idrissi	Meknes	Écran 24 pouces	1 500	1	1 500
5084	21/09/2026	Nadia Alaoui	Rabat	Souris	85	2	170

Corrigé S1 (a, b, c)

✔ Solution

(a) **Redondances** : Sara Idrissi et sa ville sont recopiées à chaque ligne ; la date de la commande 5081 est écrite deux fois ; le prix du clavier est recopié.

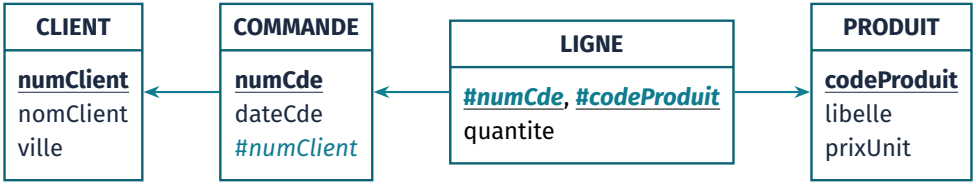
(b) **Incohérences** : « Meknès » / « Meknes » ; la souris coûte 80 puis 85.

(c) **Suppression** : si l'on annule la commande 5084, on perd la cliente Nadia Alaoui et la ville Rabat.

⚠ Et la colonne Total ?

Elle se *calcule* (PrixUnit $\times$ Qté) : la stocker crée une redondance supplémentaire et un risque d'incohérence. On la supprimera du modèle (chapitre 2).

Corrigé S1 (d) : des tables séparées



✔ Solution

Quatre tables : chaque client, chaque produit et chaque commande n'existent qu'**une fois** ; LIGNE relie une commande à ses produits avec la quantité. Le total se calcule par une requête.

Exercice S2 : l'architecture ANSI/SPARC chez BiblioUni

Exercice

La direction de BiblioUni demande trois choses :

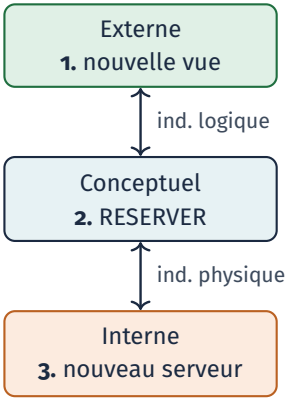
1. un écran « livres les plus empruntés » pour la direction ;
2. l'ajout de la notion de « réservation » d'un livre par un adhérent ;
3. le déplacement de la base sur un nouveau serveur plus rapide.

Pour chaque demande : quel niveau est modifié ? Les autres niveaux sont-ils touchés ? Quelle indépendance est en jeu ?

Corrigé S2

✔ Solution

1. Écran direction → nouvelle **vue externe** ; le conceptuel et l'interne ne changent pas.
2. Réservation → le **conceptuel** change (nouvelle association) ; les vues existantes continuent de fonctionner : *indépendance logique*.
3. Nouveau serveur → **interne** seulement ; ni le schéma ni les écrans ne changent : *indépendance physique*.



Exercice S3 : acteurs et cycle de vie

Exercice

Un hôpital lance un projet « gestion des consultations ». Voici des événements du projet, dans le désordre. Pour chacun, donnez l'**acteur** et l'**étape** du cycle de vie.

a) Le MCD PATIENT–CONSULTER–MEDECIN est présenté au chef de service, qui le corrige.

b) Une sauvegarde nocturne est programmée sur le serveur.

c) Un an après, on ajoute la notion d'« ordonnance ».

d) Les secrétaires saisissent les rendez-vous.

e) L'application « prise de rendez-vous » est programmée.

f) Les médecins expliquent qu'une consultation concerne un seul patient.

Corrigé S3

Événement	Acteur	Étape
f) un seul patient par consultation	utilisateurs (médecins) + concepteur	1. étude des besoins
a) MCD présenté et corrigé	concepteur + utilisateur	2. conception
e) application programmée	développeur	3. réalisation
b) sauvegarde nocturne	administrateur	3. réalisation / 4. exploitation
d) saisie des rendez-vous	utilisateurs finaux	4. exploitation
c) ajout de l'ordonnance	concepteur puis développeur	5. maintenance

✔ Ordre chronologique

f → a → e → b → d → c. Remarquez que la règle « un seul patient » (f) est recueillie *avant* le MCD (a) : c'est elle qui fixera la cardinalité **1,1**.

Exercice S4 : mini-conception « école »

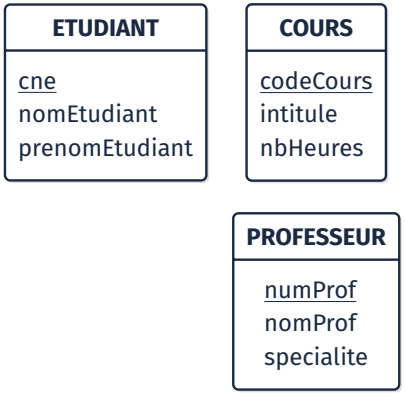
Besoin

Une école veut suivre ses cours. Chaque étudiant (CNE, nom, prénom) suit plusieurs cours ; chaque cours (code, intitulé, nombre d'heures) est suivi par plusieurs étudiants. Un cours est assuré par un seul professeur (numéro, nom, spécialité) ; un professeur peut assurer plusieurs cours. Pour chaque étudiant et chaque cours, on enregistre une note.

Exercice

En suivant les dix étapes : (a) listez les entités et leurs identifiants ; (b) écrivez les règles de gestion ; (c) dessinez le MCD ; (d) donnez le MLD.

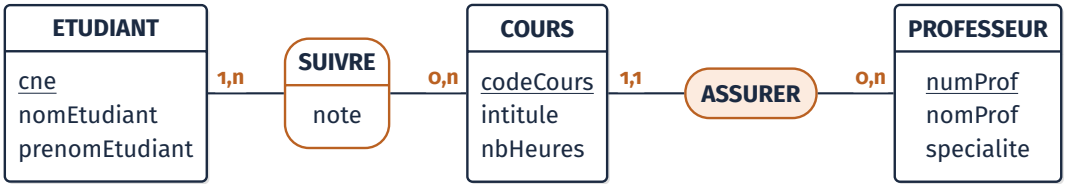
Corrigé S4 (a, b) : entités et règles



✔ Règles de gestion

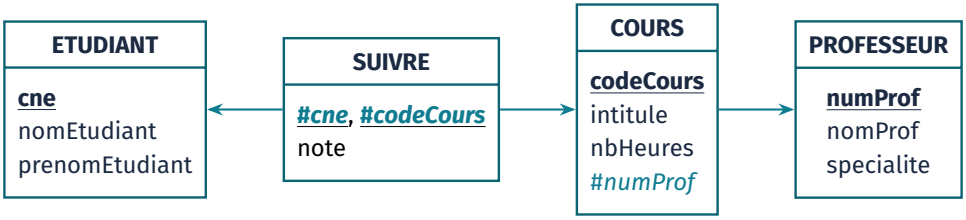
- RG1 : un étudiant suit **1,n** cours ; un cours est suivi par **0,n** étudiants.
- RG2 : la note dépend *à la fois* de l'étudiant et du cours : elle sera **portée** par l'association SUIVRE.
- RG3 : un cours est assuré par **1,1** professeur ; un professeur assure **0,n** cours.

Corrigé S4 (c) : le MCD



La note est portée par SUIVRE : elle n'appartient ni à l'étudiant seul, ni au cours seul.

Corrigé S4 (d) : le MLD



ETUDIANT (**cne**, nomEtudiant, prenomEtudiant) **COURS** (**codeCours**, intitule, nbHeures, *#numProf*)

SUIVRE (***#cne***, ***#codeCours***, note) **PROFESSEUR** (**numProf**, nomProf, specialite)

Erreurs fréquentes des débutants (bilan du chapitre)

⚠ À ne pas faire

- confondre base de données et SGBD ;
- commencer par créer des tables dans MySQL ;
- écrire des types SQL dans le MCD ;
- mettre plusieurs valeurs dans une cellule ;
- inventer les règles de gestion sans interroger les utilisateurs.

À faire

- nommer les niveaux : conceptuel, logique, physique ;
- suivre la démarche dans l'ordre ;
- une table par « chose », une clé par table ;
- faire valider le MCD avant d'aller plus loin ;
- nommer proprement dès le début.

Êtes-vous prêt pour le chapitre 2 ?

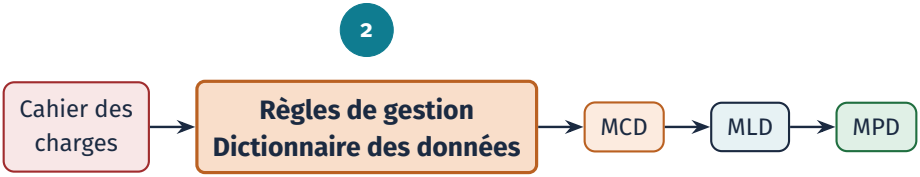
Cochez mentalement

- ☐ Je sais citer trois limites des fichiers.
- ☐ Je distingue base de données et SGBD.
- ☐ Je sais reconnaître un MCD, un MLD et un script SQL.
- ☐ Je sais lire **0,n** et **1,1**.
- ☐ Je connais les dix étapes du mini-projet.

Pour aller plus loin

- G. Gardarin, *Bases de données*, Eyrolles.
- J. Akoka, I. Comyn-Wattiau, *Conception des bases de données relationnelles*, Vuibert.
- Documentation Looping et MySQL Workbench (en ligne).

Prochain chapitre : analyser le cahier des charges



Chapitre 2 : lire un cahier des charges, extraire les règles de gestion, construire le dictionnaire des données, épurer

À préparer

Relisez le cahier des charges de BiblioUni et soulignez chaque nom de donnée et chaque phrase qui exprime une contrainte.

Merci de votre attention. Questions ?