FORMATION DEV WEB WEB MOBILE
CCP 2 - DÉVELOPPER LA PARTIE BACK-END D'UNE APPLICATION WEB

LES BASES DE DONNEES RELATIONNELLES

JOUR 1 : DECOUVERTE DES BDD ET INSTALLATION

PRESENTE PAR NATACHA DESSE

AFEC

DEROULE MODULE

- **Découverte des BDD et Installation**
- Les relations et modélisation (DEA)
- Contraintes et schéma physique
- Jeu d’essai, insertion et agrégations, requêtes avancées
- Jointures et sous requêtes
- Sauvegardes SQL et gestion des rôles

DECOUVERTE DES BDD ET INSTALLATION

LES OBJECTIFS

Comprendre le rôle et l'intérêt des bases de données relationnelles dans une application

Installer et configurer un Système de Gestion de Base de Données (SGBD) open-source (MySQL)

Savoir mettre en place une BDD et faire des requêtes SQL

NUAGE DE MOTS

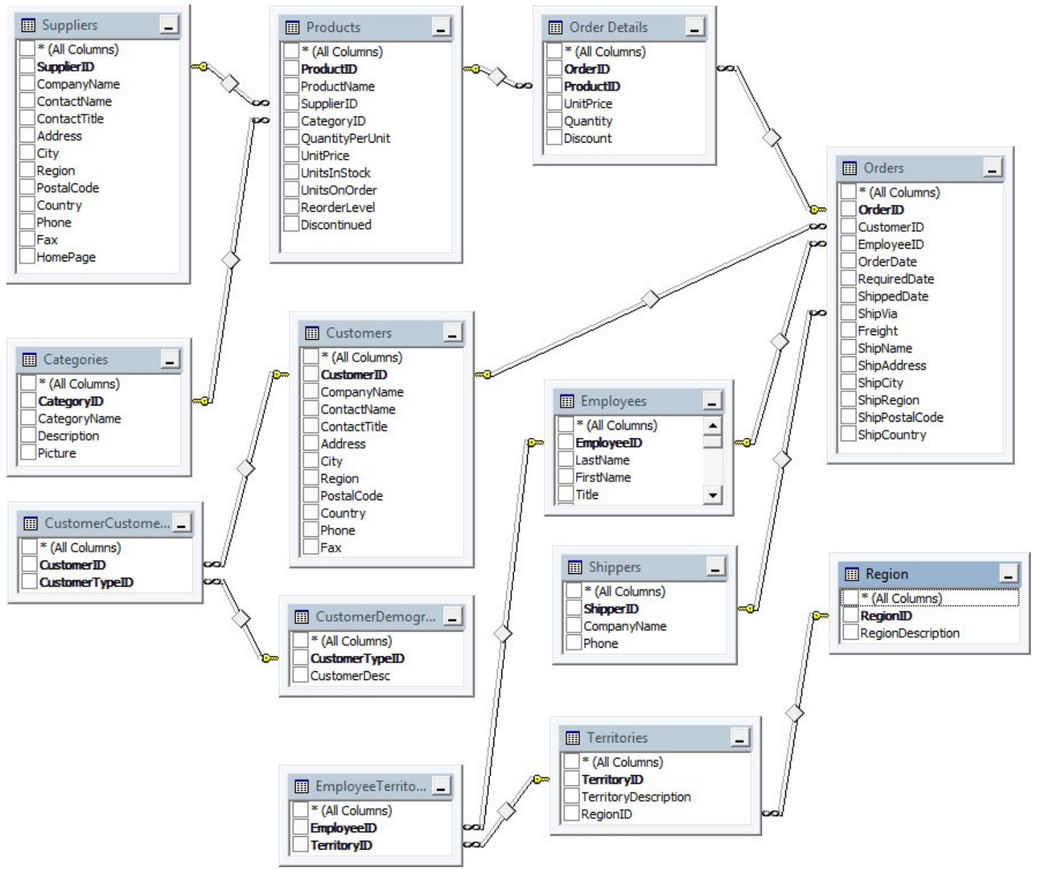
https://app.wooclap.com/BCBNPY?from=event-page

LES NOTIONS

QU'EST-CE QU'UNE BASE DE DONNÉES RELATIONNELLE ?

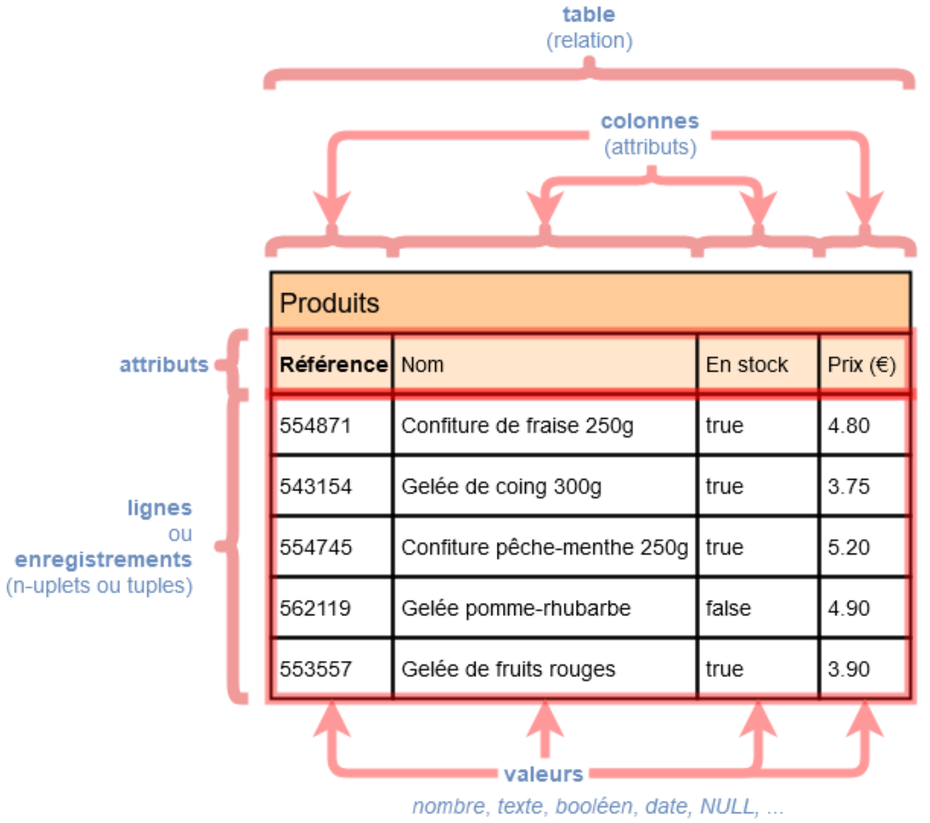
- Un Système de Gestion de Base de Données (SGBD) **stocke** et **organise** des données structurées.

- Le modèle relationnel repose sur des tables **interconnectées** par des **relations**, garantissant l'**intégrité** et la **cohérence**.



CONCEPTS CLÉS

- **Table** (entité) : structure bidimensionnelle composée de lignes (enregistrements) et de colonnes (attributs). Chaque table représente une entité métier (ex. clients, produits).
- **Colonne** (attribut) : définition d'un attribut, avec un type de données (entier, chaîne de caractères, date, booléen, etc.) et des contraintes (NOT NULL, UNIQUE, DEFAULT, CHECK).
- **Ligne** (entrée ou tuple) : instance d'une table, contenant une valeur pour chaque colonne. Chaque ligne est identifiée par une clé primaire garantissant l'unicité.
- **Valeurs** : Croisement entre une ligne et une colonne



Produits

Référence	Nom	En stock	Prix (€)
554871	Confiture de fraise 250g	true	4.80
543154	Gelée de coing 300g	true	3.75
554745	Confiture pêche-menthe 250g	true	5.20
562119	Gelée pomme-rhubarbe	false	4.90
553557	Gelée de fruits rouges	true	3.90

L'APPROCHE MERISE

Méthode d'**E**tude et de **R**ealisation **I**nformatique pour les **S**ystèmes d'**E**ntreprise

- En 1977 le Ministère de l'Industrie Français finance le développement de Merise avec des SSII, le ministère de l'équipement et des universitaires. Elle est libre de droits (open source avant l'heure).
- Elle vise les SI construits autour des bases de données relationnelles.
- Elle est encore aujourd'hui très utilisée en France, même si elle est fortement concurrencée par les approches à objets (UML). Il en existe plusieurs versions (Merise, Merise 2, Merise Objet, …). Dans la pratique beaucoup d'entreprises se limitent à un Merise de base assez restreint.
- Elle n'a jamais été exportée en dehors des pays francophones. Beaucoup de pays ont défini des méthodes nationales (ex: Structured System Analysis and Design Method – SSADM en Angleterre)

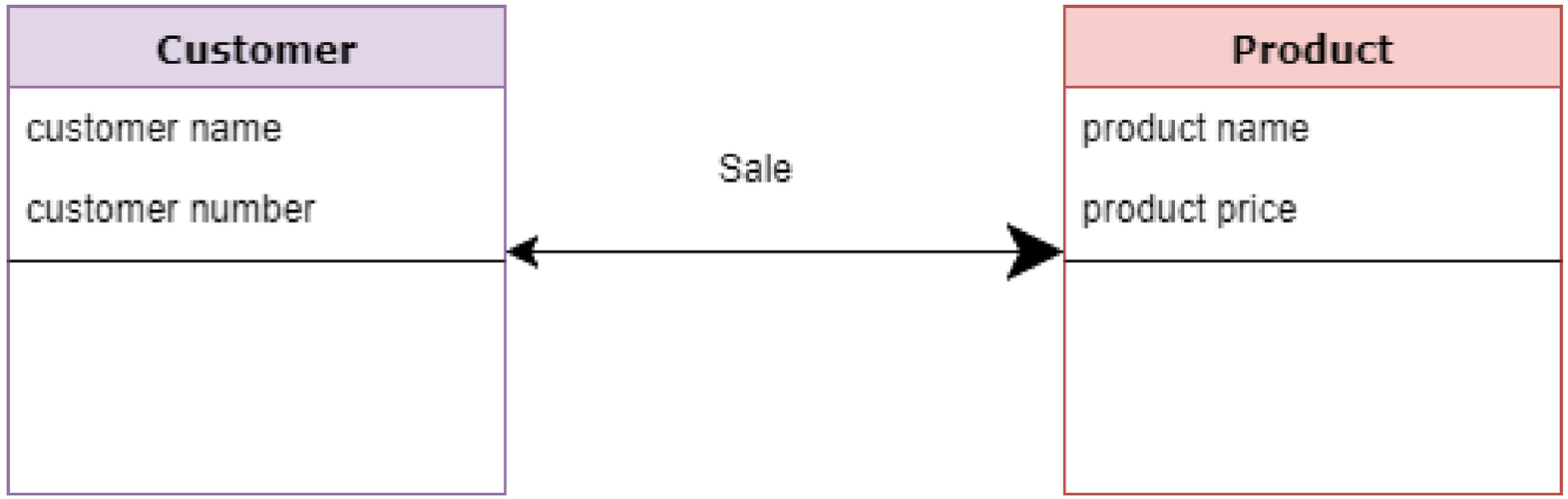
LE CYCLE D'ABSTRACTION

3 NIVEAUX

- Niveau conceptuel : Quoi ? Avec quelles données => MCD
- Niveau organisationnel : Qui ? Où ? Quand ? => MLD
- Niveau physique : Comment ? => MPD

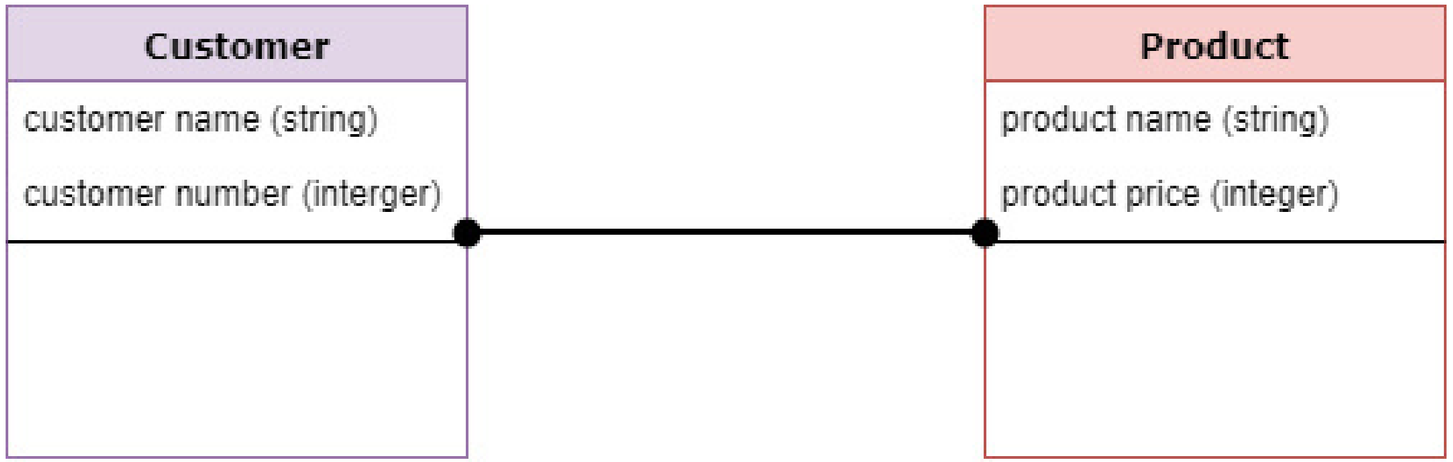
MODELE DE DONNEES CONCEPTUEL (MCD)

- Un Modèle de données conceptuel est une vue organisée des concepts de bases de données et de leurs relations.
- Le but de la création d'un modèle de données conceptuel est d'établir des entités, leurs attributs et leurs relations.
- À ce niveau de modélisation des données, il n'y a pratiquement aucun détail disponible sur la structure réelle de la base de données. Les parties prenantes commerciales et les architectes de données créent généralement un modèle de données conceptuel.



MODELE LOGIQUE DE DONNEES (MLD)

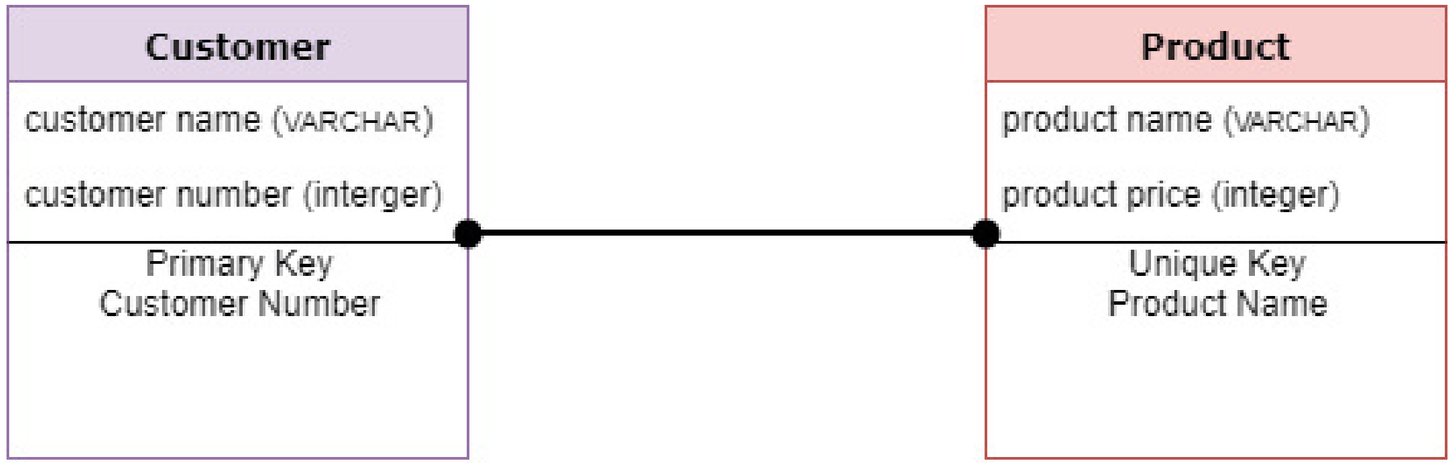
- Vue d'ensemble Modèle de données logique est utilisé pour définir la structure des éléments de données et pour établir les relations entre eux.
- Le modèle de données logique ajoute des informations supplémentaires aux éléments du modèle de données conceptuel. L'avantage d'utiliser un modèle de données logique est de fournir une base pour former la base du modèle physique. Cependant, la structure de modélisation reste générique



MODELE PHYSIQUE DE DONNEES (MPD)

- A Modèle de données physique décrit une implémentation spécifique à la base de données du modèle de données. Il offre une abstraction de base de données et aide à générer le schéma. Cela est dû à la richesse des métadonnées offertes par un modèle physique de données.

- Le modèle de données physique aide également à visualiser la structure de la base de données en répliquant les clés de colonne de la base de données, les contraintes, les index, les déclencheurs...



CLES PRIMAIRES ET SECONDAIRES

- Clé primaire : identifiant unique, souvent auto-incrémenté ou UUID.
- Clé étrangère : lien vers la clé primaire d'une autre table, permet d'assurer la référentialité.

Table "Commande"

Clé étrangère

Id commande	Produit	Prix	Id client
100	PC 16'	599	1
101	Souris	29	2
102	Lampe de bureau	30	2
103	Clavier	55	2

Clé primaire

Table "Client"

Id client	Prenom	Nom	Adresse
1	Jean	Bonneau	50 av St Marc
2	Ambre	Cerna	13 Grande Rue

Clé primaire

CLES PRIMAIRES ET SECONDAIRES

Clé primaire

Table “Commande”

Clé étrangère

Id commande	Produit	Prix	Id client	Livraison ID
100	PC 16’	599	1	200
101	Souris	29	2	201
102	Lampe de bureau	30	2	202
103	Clavier	55	2	203
104	Bureau	300	3	204
105	Chaise Gaming	159	3	205
106	PC Bureau	899	1	206

Clé primaire

Table “Client”

Id client	Prenom	Nom	Adresse
1	Jean	Bonneau	50 av St Marc
2	Ambre	Cerna	13 Grande Rue
3	Jack	Jakson	55 Mac street

Clé Primaire

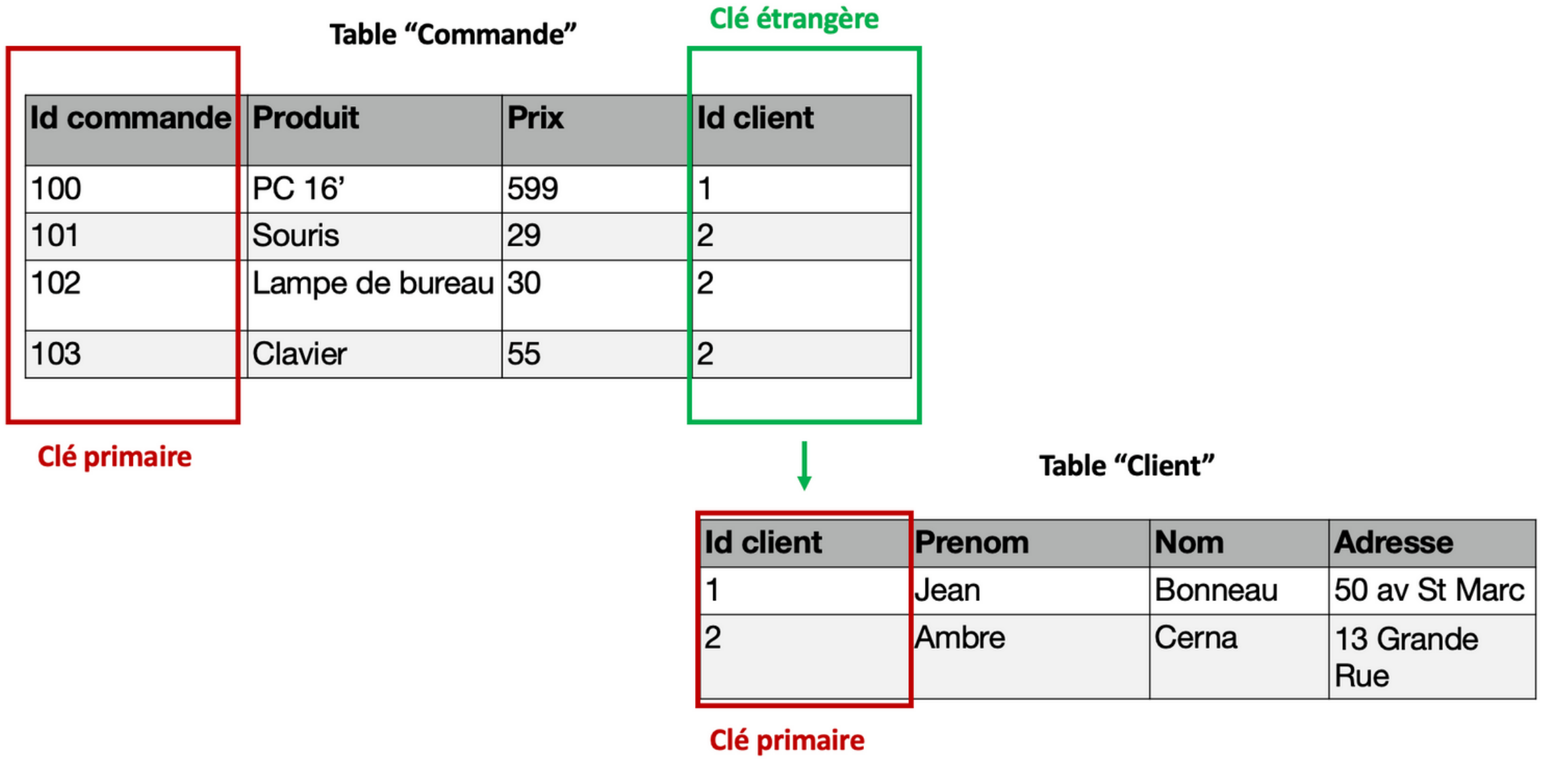
Table “Livraison”

Livraison ID	Livreur	Numéro de colis
200	Colissimo	22223958
201	DHL	44490009
202	DHL	22009877
203	Chronopost	33466771
204	FedEx	4499008
205	FedEx	557774449
206	Colissimo	112440

On peut aller encore plus loin ....

PRINCIPE D'INTEGRITE

- **Intégrité d'entité** : impossible d'avoir deux lignes avec la même clé primaire.
- **Intégrité référentielle** : suppression ou mise à jour en cascade, verrouillage des références erronées.



Id commande	Produit	Prix	Id client
100	PC 16'	599	1
101	Souris	29	2
102	Lampe de bureau	30	2
103	Clavier	55	2

Id client	Prenom	Nom	Adresse
1	Jean	Bonneau	50 av St Marc
2	Ambre	Cerna	13 Grande Rue

AVANTAGES D'UN MODÈLE RELATIONNEL

- Requêtes puissantes (SQL) pour filtrer, agréger et croiser les données.
- Multi-utilisateurs : gestion des transactions, verrous et isolation.
- Flexibilité pour faire évoluer le schéma tout en conservant l'historique des données.

ATELIER

Modélisation d’une base de données “librairie”

CONSIGNES

Voici le scénario

*Une librairie vend des **livres**.*

*Chaque livre a un **titre** , un **prix**, un **éditeur**, une **date** de parution, un **stock** disponible.*

*Un livre est écrit par un **auteur**.*

*Chaque auteur a un **nom**, un **prénom**, une **date** de naissance.*

Vous devrez constuire le MCD de cette librairie

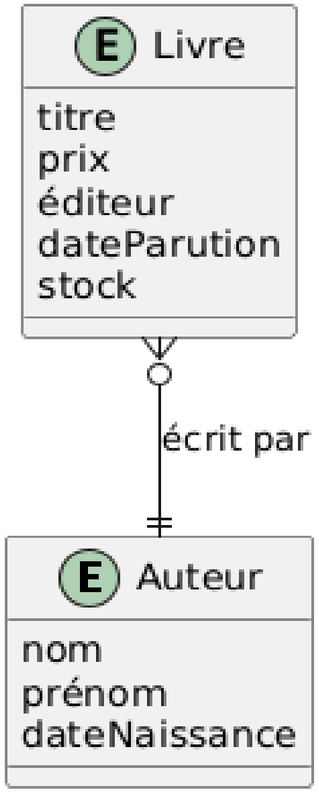
Pour cela

Utilisez Draw.io
Identifier les éléments suivants :

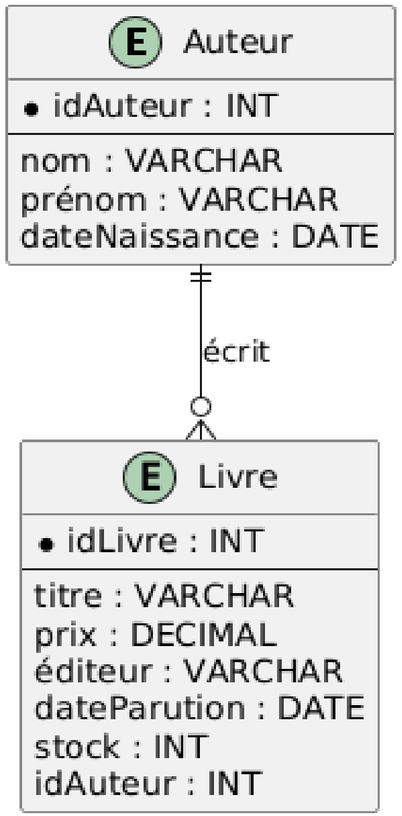
- Les entités principales (ex : Livre, Auteur…)
- Les attributs de chaque entité (ex : nom, prix…)

REPONSE

MCD



MLD

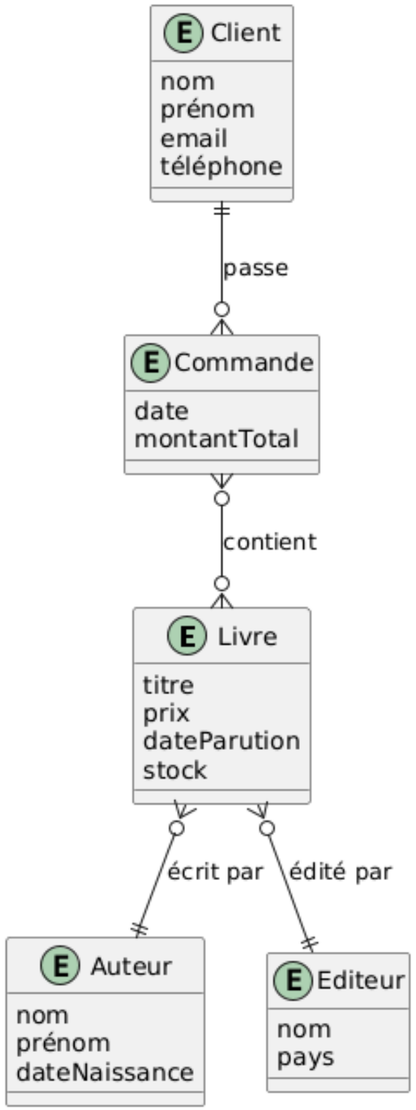


STRUCTURE PLUS REALISTE

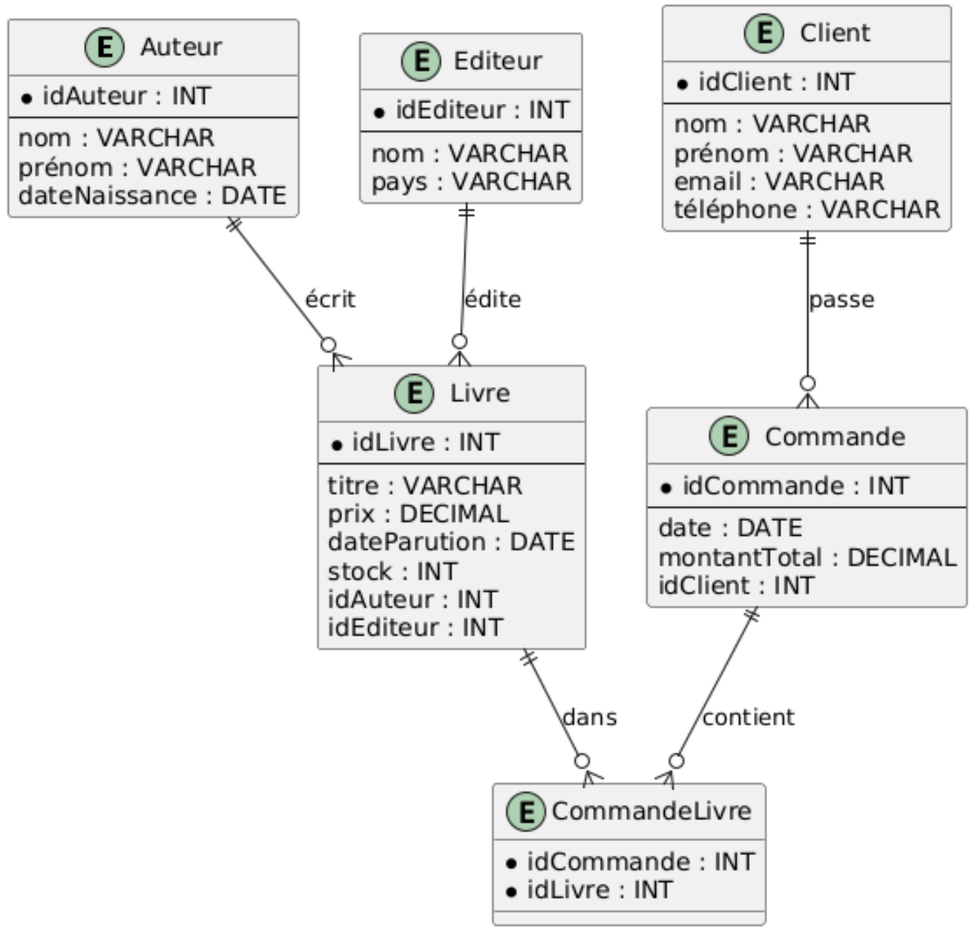
- La librairie a des **clients** qui peuvent passer des **commandes**.
- Une **commande** a une **date**, un **montant** total et peut contenir plusieurs livres.
- Chaque **client** a un **nom**, un **prénom**, un **email** et un **téléphone**.
- Un éditeur édite plusieurs **livres**, et a un **nom** et un **pays**.

REPONSE

MCD



MLD



INSTALLATION ET CONFIGURATION DU SGBDR

CHOIX DU SGBDR

MySQL

Adoption et écosystème

Très utilisé dans le web (WordPress, Drupal, Joomla, Magento…)
Large communauté, nombreux tutoriels et hébergeurs au support natif

Performance et simplicité

Excellente rapidité en lecture, moteur par défaut InnoDB adapté au web
Installation et configuration simplifiées

Cas d'usage

Sites à fort trafic en lecture
Applications PHP/WordPress
Projets nécessitant une mise en place rapide

PostgreSQL

Richesse fonctionnelle

- Conformité SQL avancée (CTEs, window functions, full joins)
- Types de données étendus : JSONB, UUID, arrays, géospatial (PostGIS)

Robustesse et intégrité

- Mécanismes d'indexation sophistiqués
- Transactionnel ACID complet avec MVCC performant

Cas d'usage

- Applications analytiques et reporting
- Systèmes géospatiaux
- Projets nécessitant des requêtes complexes ou du traitement de données massif

MySQL pour la rapidité et la simplicité sur des workloads web classiques.
PostgreSQL pour la puissance fonctionnelle et la scalabilité des fonctionnalités avancées.

INSTALLATION MYSQL

1. Téléchargement

- Aller sur le site de MySQL Community Server : MySQL (plus répandu)
- Choisir l'installateur Windows (Windows (x86, 64-bit), MSI Installer).

2. Installation
- Lancer le fichier MSI et suivre l'assistant.
- Sélectionner le type d'installation "Developer Default" pour installer MySQL Server, MySQL Shell, MySQL Workbench.
- Définir un mot de passe pour l'utilisateur root.

3. Configuration

- Cocher "Configure MySQL as a Windows Service" pour un démarrage automatique.
- Choisir le port par défaut 3306.

PRISE EN MAIN CLI

Vérifier SQL bien installé

```
mysql --version
```

Se connecter à sql

```
mysql -u root -p
```

Voir les databases

```
SHOW DATABASES;
```

LE LANGAGE SQL

LES TYPES EN MYSQL

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 214748-3647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LONGBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

CREER SA PREMIERE BASE DE DONNEES

Créer sa base de données

```
CREATE DATABASE librairie;
USE librairie;
```

Vérifier la database

```
SELECT DATABASE();
```

Créer la table livres

```
CREATE TABLE Livre (
 id INT AUTO_INCREMENT PRIMARY KEY,
 titre VARCHAR(255),
 auteur VARCHAR(100),
 date_publication DATE
);
```

Vérifier la structure

```
DESCRIBE Livre;
```

AJOUTER DES VALEURS

Créer la table livres

```
INSERT INTO Livre (titre, auteur, date_publication)
VALUES
 ('L\'Étranger', 'Albert Camus', '1942-05-19'),
 ('1984', 'George Orwell', '1949-06-08'),
 ('Le Petit Prince', 'Antoine de Saint-Exupéry', '1943-04-06');
```

Voir les livres

```
SELECT *
FROM Livre;
```

AJOUTER UNE COLONNE

Modifier la table

```
ALTER TABLE Livre
ADD COLUMN age_livre INT ;
```

```
UPDATE Livre
SET age_livre = YEAR(CURDATE() - YEAR(date_publication)
```

INTERROGER LES DONNEES

SELECT

```
SELECT *
FROM  Livre ;
```

```
SELECT auteur, titre
FROM Livre ;
```

```
SELECT *
FROM Livre
    WHERE date_publication > '1945-01-01'
```

TRI ET PAGINATION

```sql
SELECT titre
FROM Livre
ORDER BY date_publication DESC
LIMIT 2;
```

```sql
SELECT titre
FROM Livre
ORDER BY date_publication ASC
LIMIT 2;
```

RECHERCHE SUR CHAINE DE CARACTERES ‘LIKE’

```
SELECT *
FROM Livre
WHERE titre LIKE
'%Prince%';
```

Sous MySQL pour être insensible à la casse

```
WHERE titre COLLATE utf8_general_ci LIKE '%prince%';
```

SUPPRESSIONS

```
DELETE FROM Livre
WHERE YEAR(CURDATE()) - YEAR(date_publication) > 80;
```

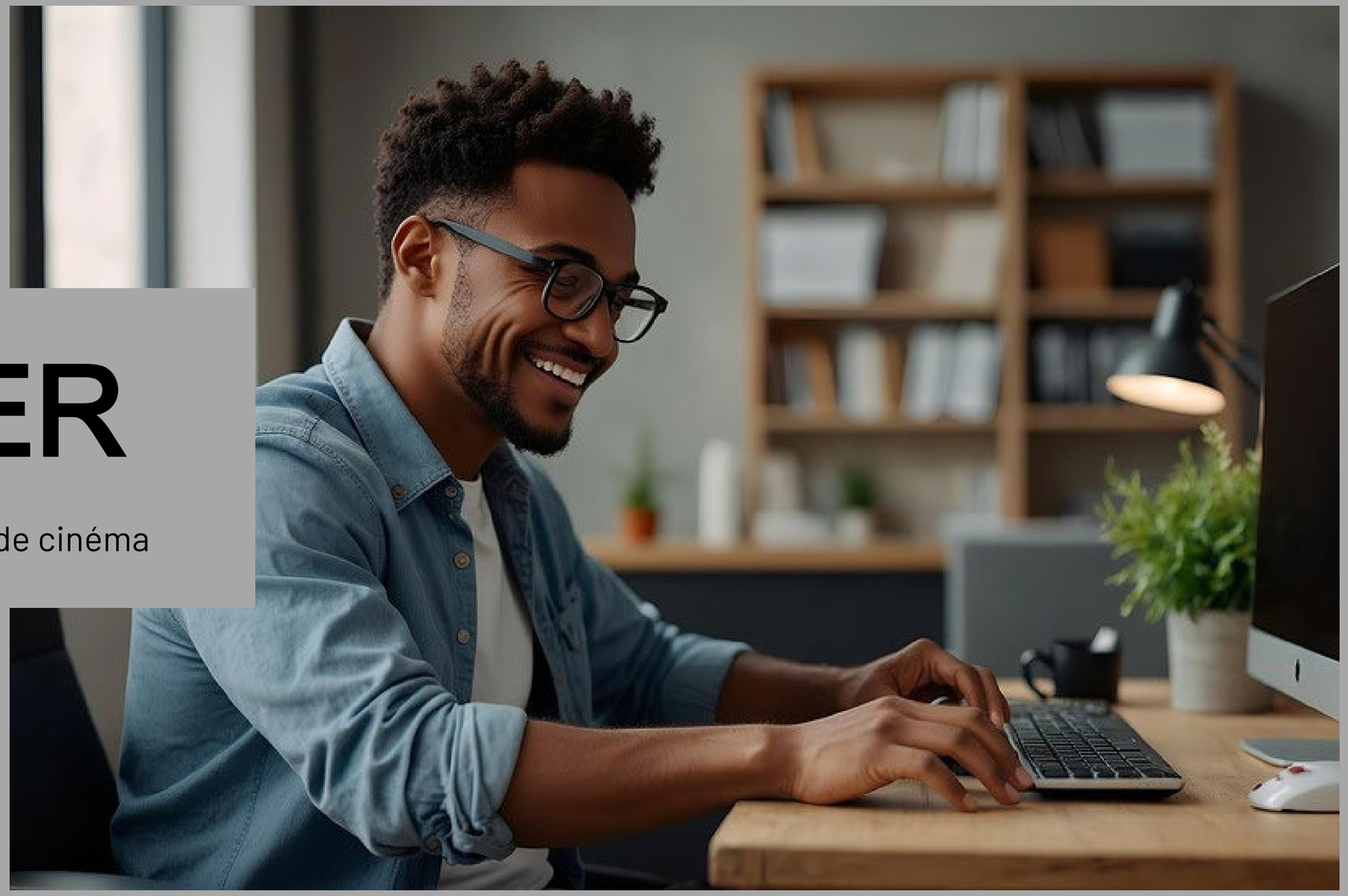
```
ALTER TABLE Livre
DROP COLUMN age_livre;
```

```
DELETE FROM Livre
WHERE id=3;
```

```
DROP TABLE IF EXISTS Livre;
```

ATELIER

Exercices Gestion d’une salle de cinéma



CONSIGNES

Voici le scénario

Une petite salle dœcinéma souhaite gérer sa programmation via une application simple. Le personnel abesoin de stocker, consulter et analyserles films projetés. Chaquefilm doit être caractérisé par :

- *Titulaire d'unidentifiant unique*
- *Titre*
- *Annéede sortie*
- *Duréeen minutes*
- *Genre*
- *Note critique (sur 10, avec un decimal)*

EXERCICE 1 : CREATION DE LA BASE

***Objectif** : Mettre en place la base de données cinema_simple et la table films correspondant au scénario.*

***Consigne**: Rédigez et exécutez les commandes SQL permettant :*

1. *Decréer la base cinema_simple*
2. *Desélectionner la base cinema_simple*
3. *Decréer la table films avec les colonnes:*
   - *id INT AUTO_INCREMENT PRIMARY KEY*
   - *titre VARCHAR(255)*
   - *annee_sortie INT*
   - *duree_min INT*
   - *genre VARCHAR(50)*
   - *note DECIMAL(3,1)*

***Livrable** : Votre script SQL reponses.sql*

CORRIGE EXERCICE 1

```sql
CREATE DATABASE cinema_simple;
USE cinema_simple;

CREATE TABLE films (
  id INT AUTO_INCREMENT PRIMARY KEY,
  titre VARCHAR(255) NOT NULL,
  annee_sortie INT NOT NULL,
  duree_min INT NOT NULL,
  note DECIMAL(2,1),
  pays VARCHAR(50) NOT NULL
);
```

EXERCICE 2 : REQUETES

https://github.com/dessna12/01-Decouverte-SQL

QUIZZ INTERACTI