

Base de données

Conception, création, requêtes et administration

FORMATEUR : OLIVIER POUSSEL

Déroulé

- ▶ Qu'est-ce qu'une BDD ?
- ▶ SQL ;
- ▶ Modélisation avec Merise ;
- ▶ Création d'une base de données ;
- ▶ Requêtes ;
- ▶ Requêtes avec Java (JDBC) ;
- ▶ TPs.

Qu'est-ce qu'une BDD ?

Une base de données (BDD) constitue un ensemble organisé et structuré de données censées représenter un domaine d'application. Ces données sont exploitées au niveau informatique par un système de gestion de base de données (SGBD).

SQL

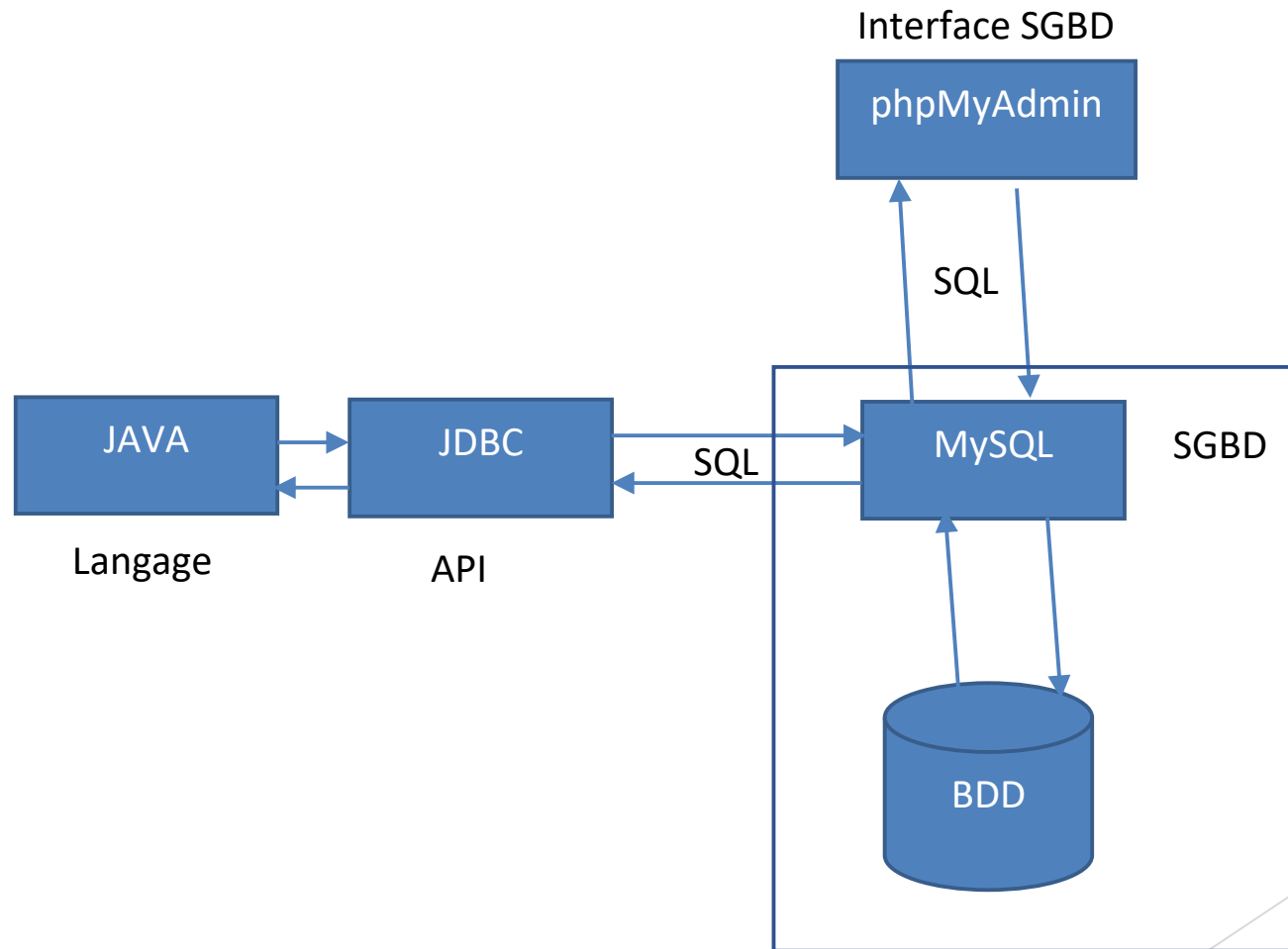
SQL (Structured Query Language ; en français : langage de requête structurée) est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

Créé en 1974, normalisé depuis 1986, le langage est reconnu par la grande majorité des Systèmes de Gestion de Bases de Données Relationnelles (SGBDR) du marché.

SGBD (en anglais : DBMS pour database management system)

- ▶ Un Système de Gestion de Base de Données (SGBD) est un logiciel système servant à stocker, à manipuler ou gérer, et à partager des données dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.
- ▶ Les SGBD les plus connus :
 - ▶ MySQL (Oracle), Maria DB, Postgres SQL, SQLite ;
 - ▶ SQL serveur (Microsoft) ;
 - ▶ Oracle Database.

FONCTIONNEMENT



Modélisation - méthode

- Merise :
 - Française ;
 - Création : 1970 ;
 - Normalisé : 1980.
- Uml :
 - Américaine ;
 - Création : 1994 ;
 - Normalisé : 1997.

Modélisation - Merise

- Cahier des charges (énoncé) ;
- Modèle Conceptuel des Données :
 - Model Entité Association (*entity/relationship* en anglais, abrégé en E/R).
- Modèle Logique de Données :
 - Model Relationnel.
- Modèle Physique de Données :
 - Implémentation dans le SGBD-R.

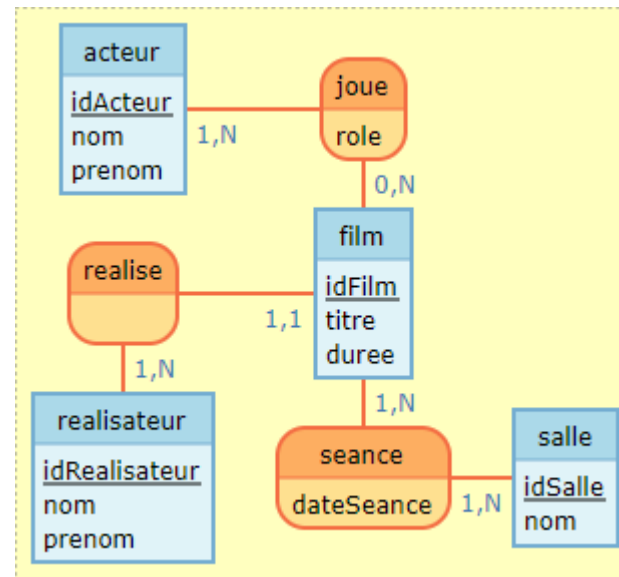
Modélisation - MCD

- **ENTITÉ** : Toute chose (concrète ou abstraite) est un type d'entité potentielle (un produit, un bâtiment, une facture, un professeur, tout animal, toute personne).
- **ATTRIBUT** : caractéristique d'une entité (nom, prénom, prix, stock, monsieur, madame, etc.). Chaque entité doit avoir un attribut identifiant qui est unique à cette entité.
- **ASSOCIATION** : on peut relier deux entités (ou plus) entre elles avec une association. Généralement les associations sont nommées avec un verbe (appartient, contient, utilise, etc.). Les associations peuvent aussi avoir des attributs.
- **CARDINALITÉ** : couple de valeurs minimales et maximales notées sous la forme **[min-max]** de chaque côté d'une association.

Modélisation - tp MCD cinéma

« Un cinéma programme des films à une date donnée. Ces films, qui n'ont qu'un seul réalisateur, ont plusieurs acteurs qui y interprètent des rôles. »

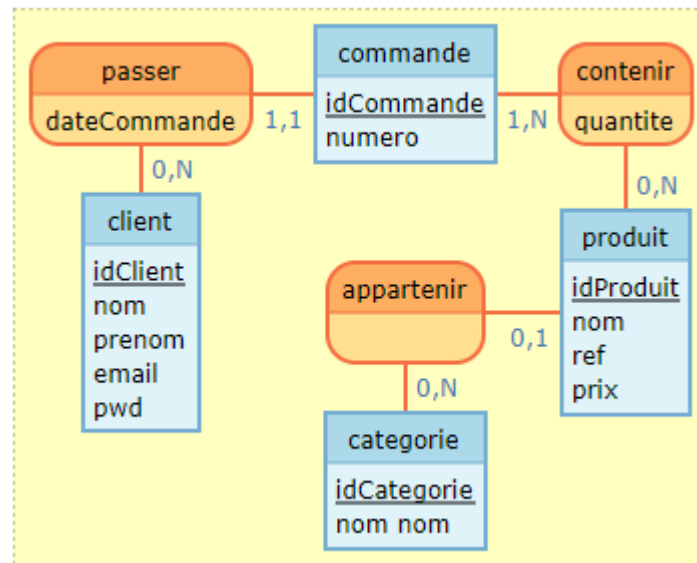
Modélisation - tp MCD Corrigé



Modélisation - tp MCD boutique

« Une boutique vend des produits, qui sont rangés par catégories. Les clients peuvent commander un ou plusieurs produits. »

Modélisation - tp MCD boutique CORRIGÉ



Modélisation - MLD

Pour passer du MCD au MLD voici les règles à suivre :

- Règle n°1 :

- Toute entité devient une table ;
- Les attributs d'une entité deviennent les propriétés de la table ;
- L'identifiant de l'entité devient la clé primaire de la table.

- Règle n°2 :

- Les associations de type N-N (cardinalité maximum) de chaque coté deviennent une table ;
- La clé primaire de la table est la concaténation des clés primaire des entités relié par l'associations. Pour cela, on référence les clés primaire concernées dans la nouvelle table créée. Cette référence s'appelle une clé étrangère.
- Si l'association avait des attributs, ils deviennent des propriétés de la table.

- Règle n°3 :

- Les associations de type 1-N se traduisent par la création d'une clé étrangère dans l'entité qui a la cardinalité 1.

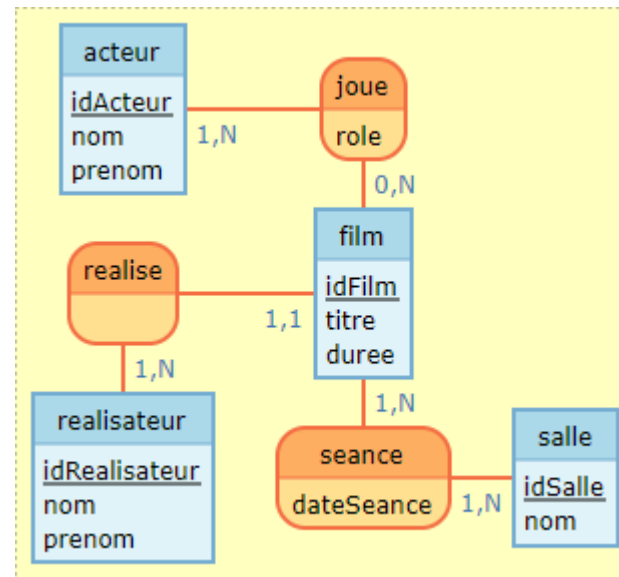
- Règle n°4 :

- Les associations de type 1-1 se traduisent par plusieurs choix :
 - Fusion des tables des types d'entités reliées ;
 - Report de clé d'une table dans l'autre (souvent) ;
 - Création d'une table reliant les deux clés des types d'entités (rare).

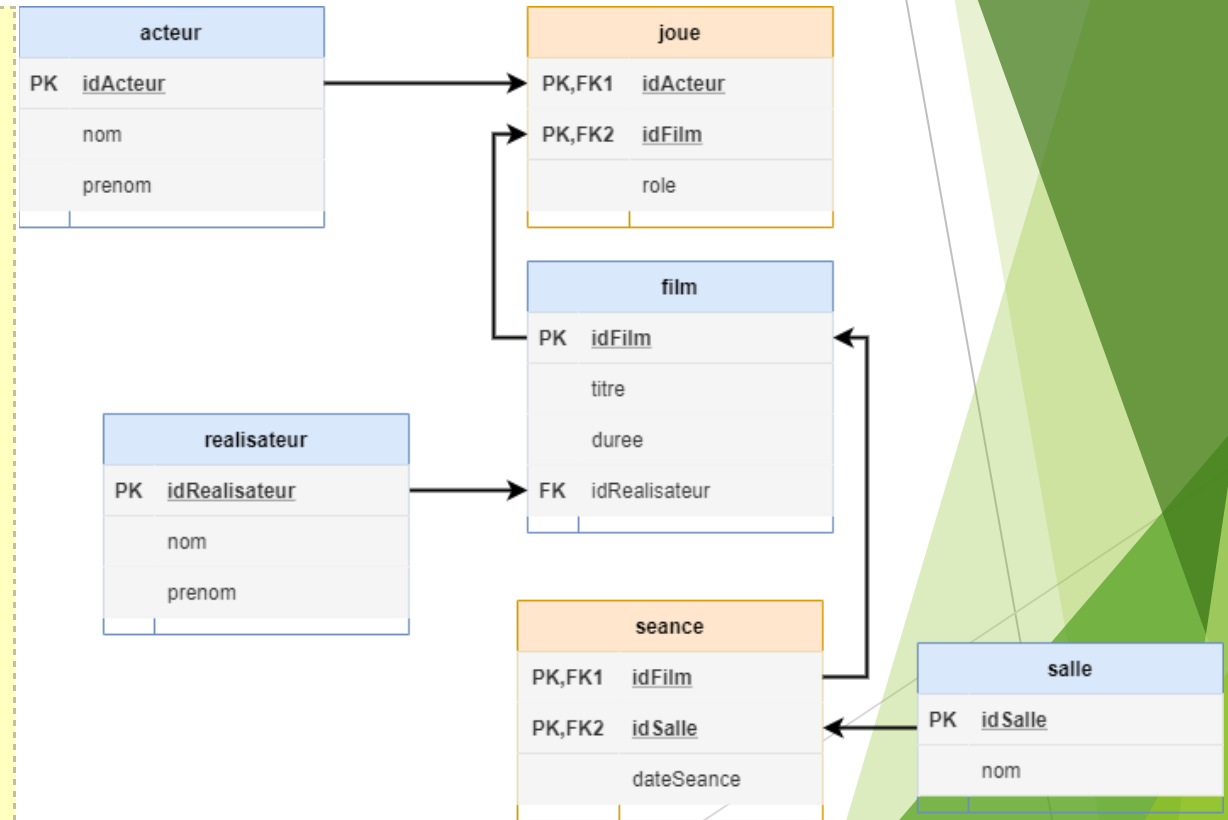
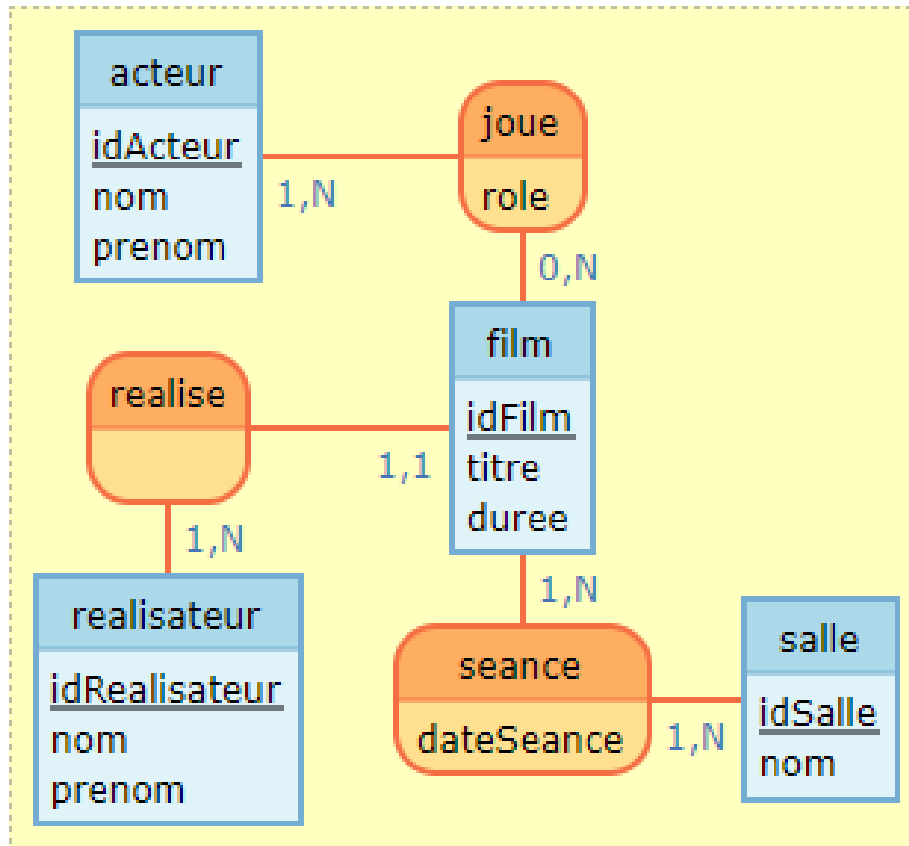
- Règle n°5 :

- Les associations ternaires (ou plus) se traduisent par la création d'une table ;
- Les clés et attributs sont gérés comme dans la règle n°2.

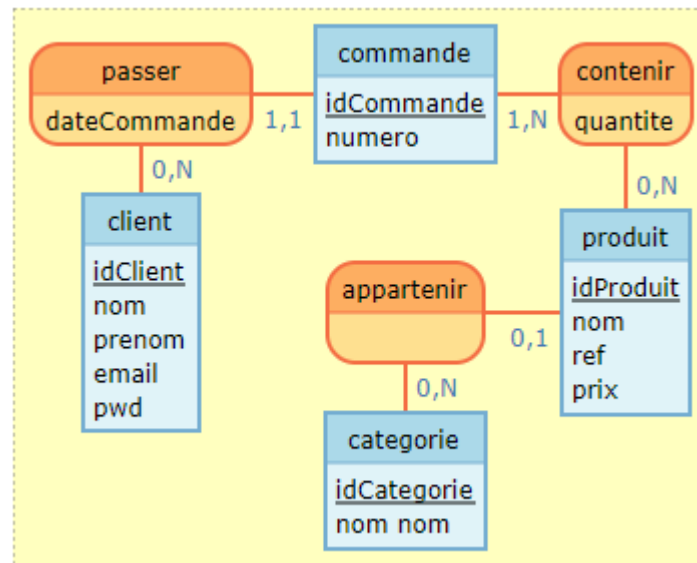
Modélisation - MLD TP cinéma



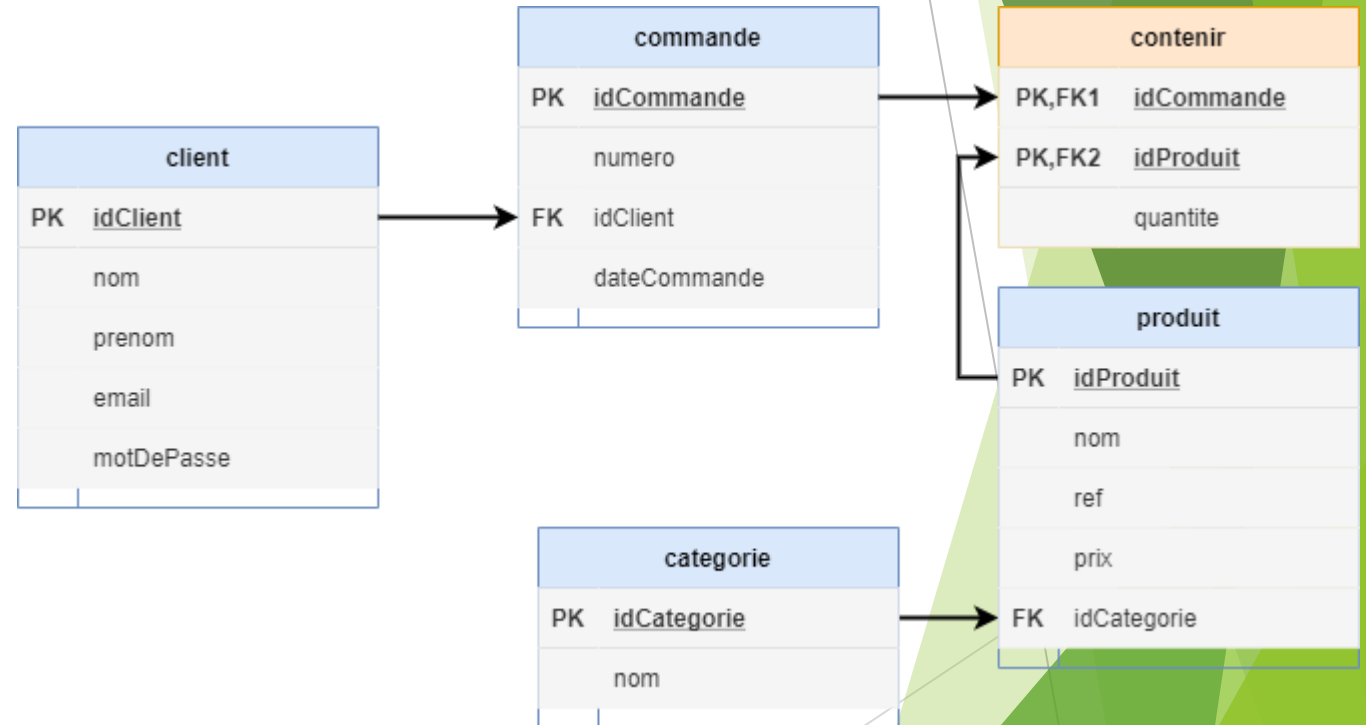
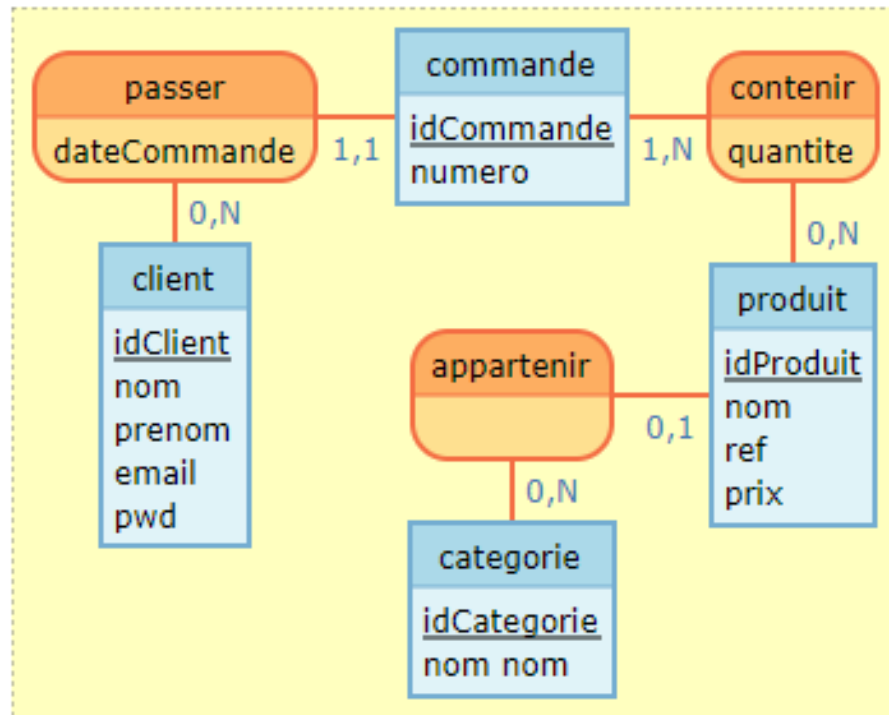
Modélisation - MLD TP cinéma corrigé



Modélisation - MLD TP boutique



Modélisation - MLD TP boutique corrigé

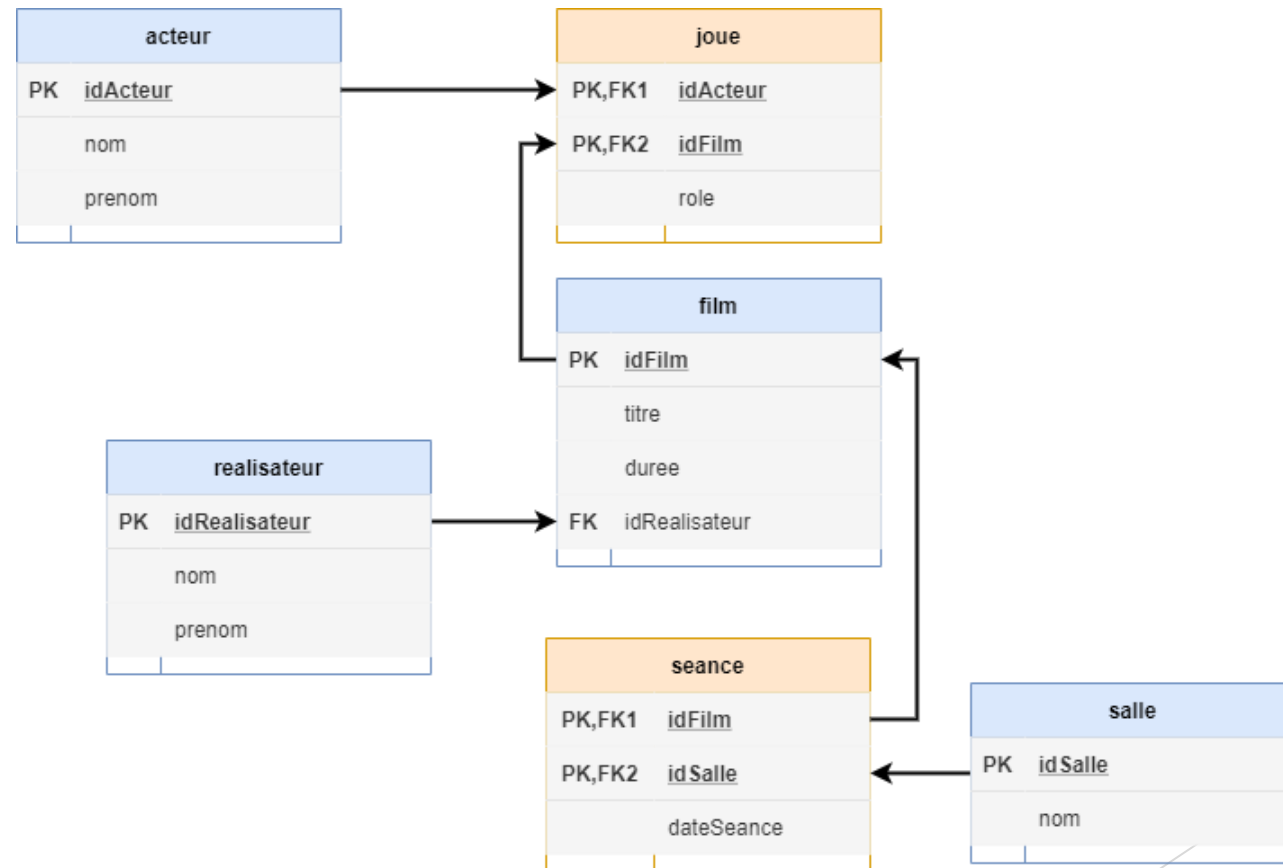


Modélisation - MPD

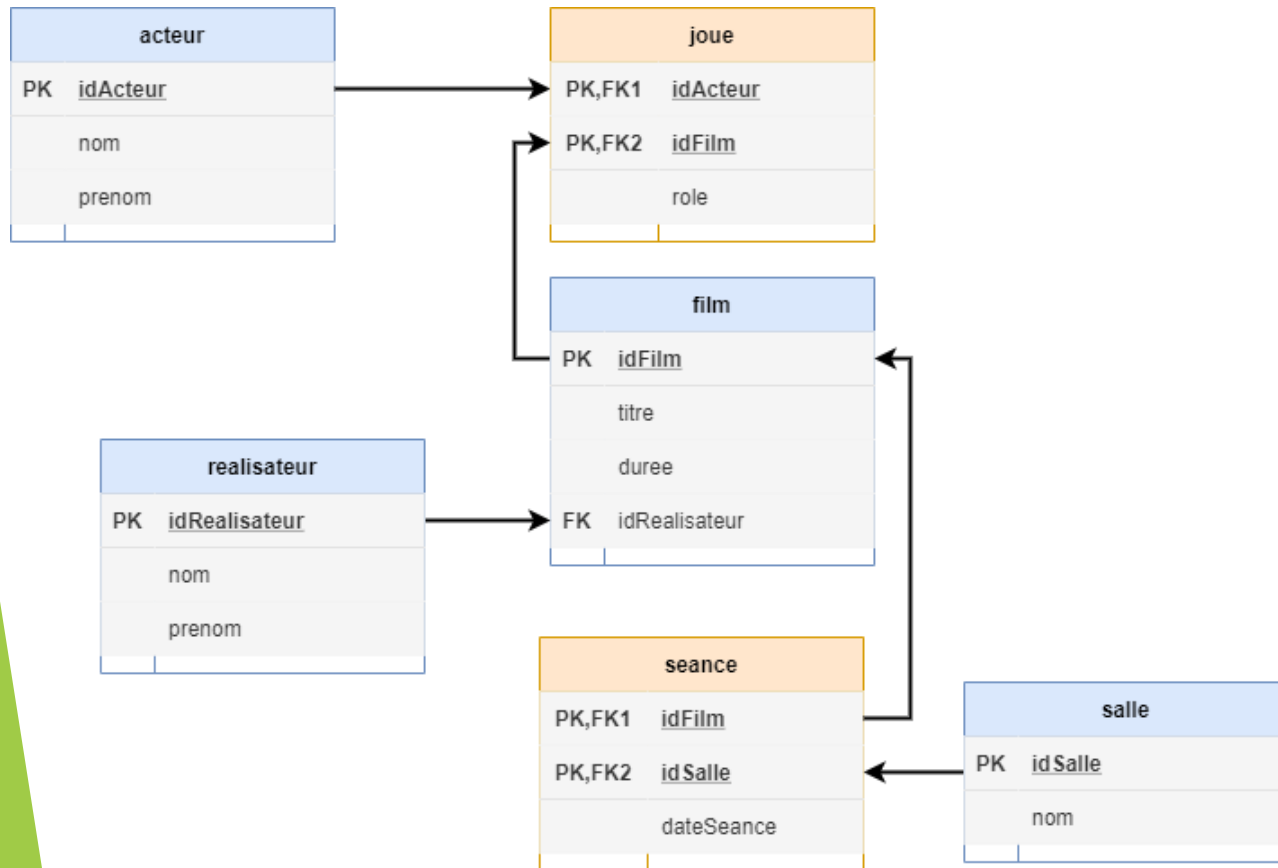
Pour passer du MLD au MPD, il faut ajouter un certain nombre d'information à notre modèle de données :

- Il faut définir le type de données des attributs (int, bool, texte, varChar, etc.) ;
- Il faut définir le jeu de caractères qu'il faut utiliser pour stocker les données (utf8, CP1252 (windows), ISO-8859-1, etc.) ;
- Il faut aussi définir l'interclassement. C'est l'algorithme de tri utilisé sur le jeu de caractère (mb4-unicode-ci, latin1_swedish_ci) ;
- Sur chaque attribut, on peut également définir s'il doit être unique, auto-incrémenté, obligatoire ou optionnel (null, not null), etc..

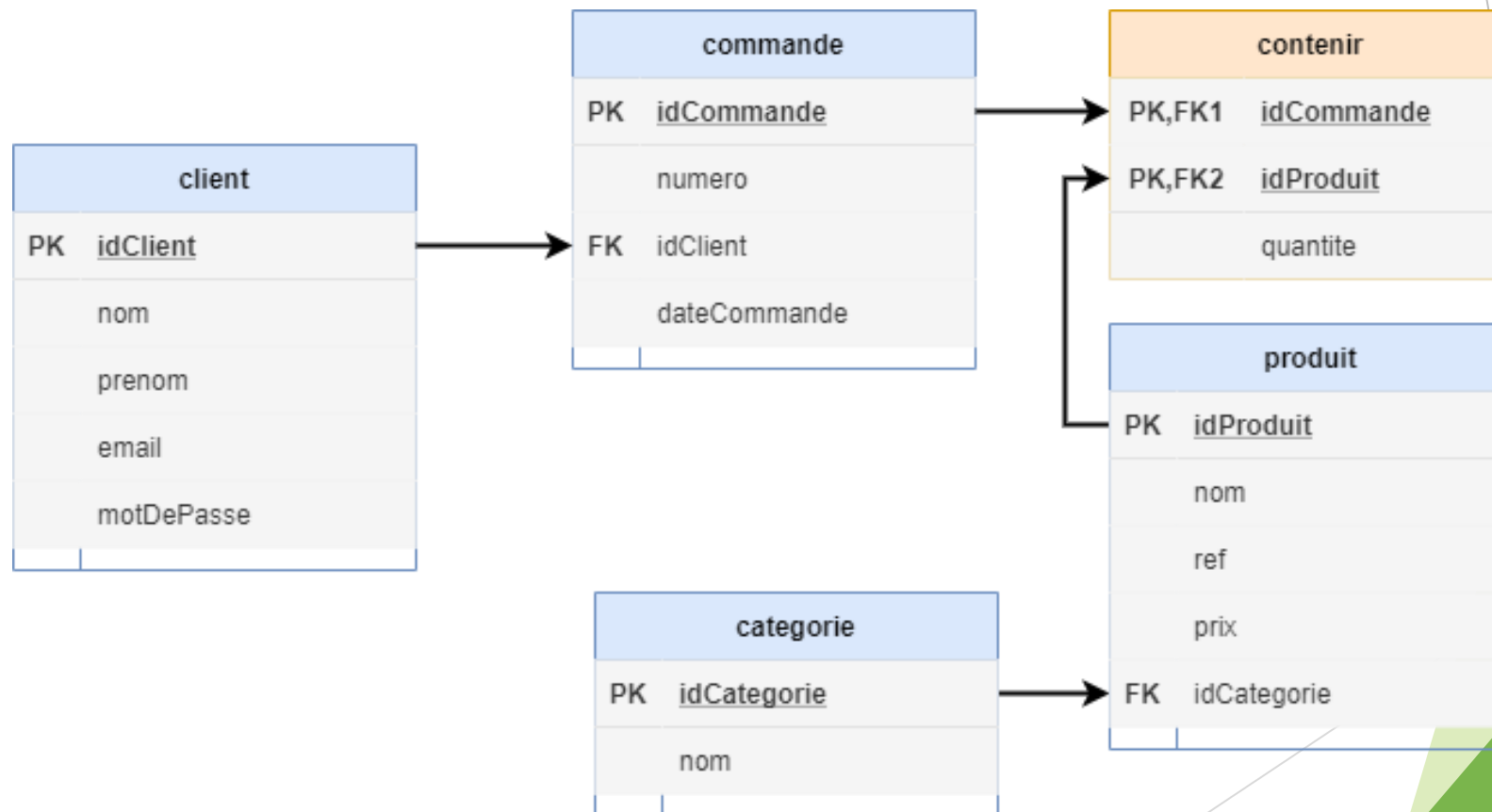
Modélisation - MPD TP cinéma



Modélisation - MPD TP cinéma corrigé



Modélisation - MPD TP boutique



Modélisation - MPD TP boutique corrigé



Modélisation - Normalisation

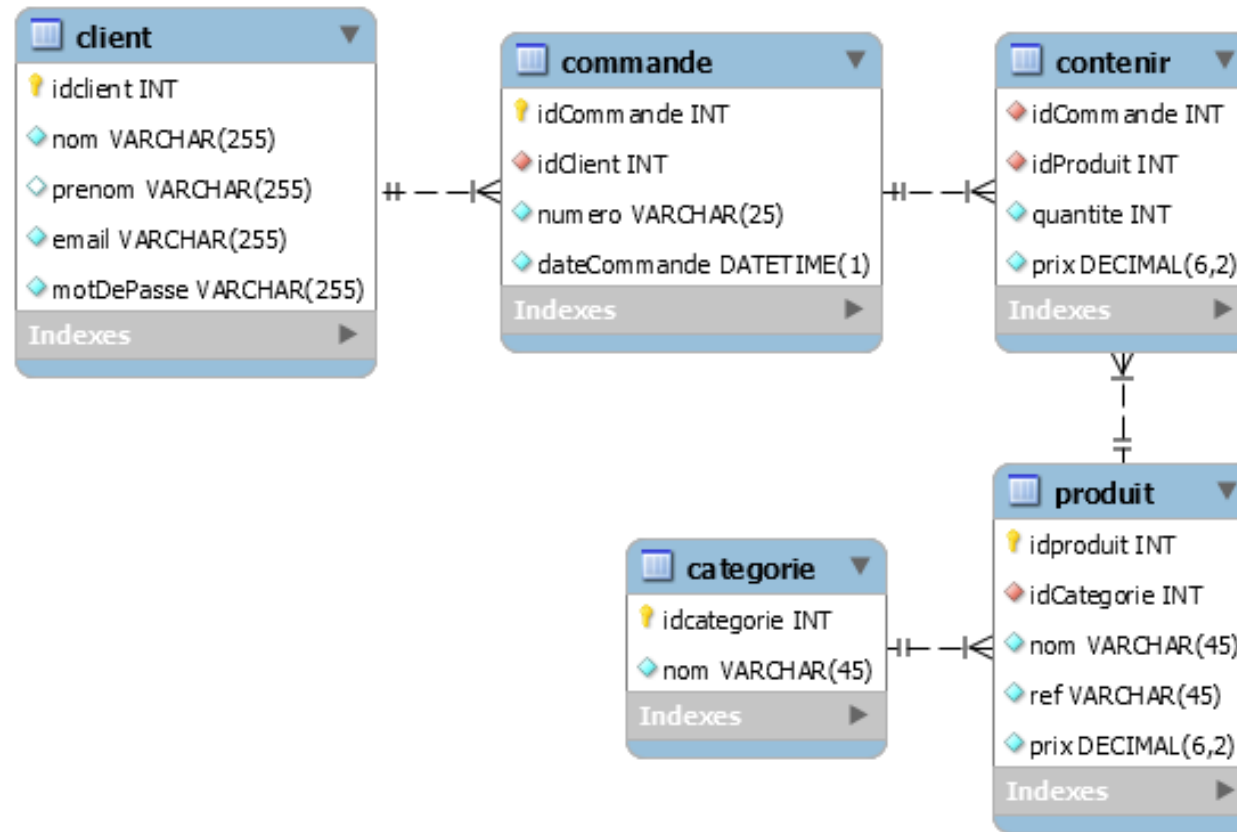
- **Clé primaire** : toute table doit avoir une clé primaire unique ;
- **Atomicité** : toute donnée dans un champ ne doit pas pouvoir être scindé en plusieurs colonnes :
 - Exemple : dans le champ **nom** il doit avoir uniquement le **nom** et pas le **nom** et le **prénom**.
- Respecter la donnée dans le temps :
 - Il est préférable de garder une date de naissance plutôt qu'un âge car l'âge augmente chaque année.
- Dépendance fonctionnelle non liée à la clé primaire :
 - Toutes les données d'une table doivent être liées à la clé primaire de la table ;
 - Exemple : une table référence les contacts des entreprises partenaires.

Modélisation - Normalisation

contact									
idContact	entreprise	nom	prenom	poste	dateNaissance	tel	tel_ent	ville_ent	pays_ent
1	TF1	LEJEUNE	Max	DRH	1970-03-17	06 78 65 45 67	01 42 24 70 00	Paris	France
2	HBO	LEGRAND	Bob	CEO	1960-12-10	06 98 67 98 98	123 456 789	Dallas	USA
3	TF1	LEFORT	Tom	JRI	1990-01-04	06 34 54 28 12	01 42 24 70 00	Paris	France

contact						
idContact	idEntreprise	nom	prenom	poste	dateNaissance	tel
1	1	LEJEUNE	Max	DRH	1970-03-17	06 78 65 45 67
2	2	LEGRAND	Bob	CEO	1960-12-10	06 98 67 98 98
3	1	LEFORT	Tom	JRI	1990-01-04	06 34 54 28 12
	entreprise					
	idEntreprise	nom	tel	ville	pays	
	1	TF1	01 42 24 70 00	Paris	France	
	2	HBO	123 456 789	Dallas	USA	

Modélisation - Dénormalisation



SQL

- **DCL** : Data Control Language / Gestion des droits d'accès au données ;
- **DDL** : Data Definition Language / Administrer la structure des données ;
- **DML** : Data Manipulation Language / Langage de manipulation des données (Requête) ;
- **TCL** : Transaction Control Language / Gestion des transactions.

SQL - DDL

- Création, modification et suppression de schémas, de tables et d'attributs (champs ou colonnes) ;
- Création des clés primaires et étrangères ;
- Création des index (de recherche).

```
CREATE SCHEMA IF NOT EXISTS `cinema` DEFAULT CHARACTER SET utf8mb4 ;  
USE `cinema` ;
```

```
-----  
-- Table `cinema`.`acteur`  
-----
```

```
CREATE TABLE IF NOT EXISTS `cinema`.`acteur` (  
  `idacteur` INT NOT NULL AUTO_INCREMENT,  
  `nom` VARCHAR(45) NOT NULL,  
  `prenom` VARCHAR(45) NULL,  
  PRIMARY KEY (`idacteur`))  
ENGINE = InnoDB;
```

SQL - DDL

- Création et modification du schéma de base de données ;
- Syntaxe :
 - Create : Création schéma ou table ;
 - CREATE TABLE <nom de la table>
 - Alter : Modification schéma ou table ;
 - Drop : Suppression schéma ou table.

SQL - DML

- Récupérer des données depuis la base de données ;
- Syntaxe :
 - `SELECT` colonne `FROM` table ;
 - `SELECT` colonne1, colonne2 `FROM` table ;
 - `SELECT DISTINCT` colonne1 `FROM` table ;
 - `SELECT` colonne `FROM` table `LIMIT` nombre / `SELECT` colonne `FROM` table `LIMIT` 3 ;
 - `SELECT` colonne `FROM` table `LIMIT` index, nombre/ `SELECT` colonne `FROM` table `LIMIT` 3, 3 ;
 - `SELECT` colonne `FROM` table `WHERE` colonne = valeur / `SELECT` colonne `FROM` table `WHERE` id = 1.

SQL - DML tp entreprise & boutique

- Créer les bases ;
- Import des données ;
- Données à récupérer.

SQL - DML - filtre, comparaison et tri

- Null ;
- Between and ;
- In ;
- Like ;
- AND ;
- OR ;
- ORDER BY ;
- AS (Alias).

SQL - DML - agrégation

- Count, max, min, avg, sum() ;
- Group By ;
- Having.

SQL - DML - jointure

- Jointure ;
- Inner Join ;
- Outer Join (left et right) ;
- ON ou Using.

SQL - DML tp entreprise & boutique jointure

Récupérer des données agrégées avec des jointures.

SQL - DML - Insert & update

- Ajout de données :
 - `INSERT INTO` table `VALUES` (value, value, value, value) // ordre important ;
 - `INSERT INTO` table (colonne, colonne) `VALUES` (value, value).
- Mise à jour des données :
 - `UPDATE` table `SET` colonne = value // attention pas de where, toutes les lignes de la table sont affectées ;
 - `UPDATE` table `SET` colonne = value `WHERE` id = 1 // mise à jour de la donnée uniquement pour l'id 1 ;
 - `UPDATE` table `SET` colonne1 = value1, colonne2 = value1, colonne3 = value1 `WHERE` id = 1.
- Suppression de données :
 - `DELETE FROM` table // attention pas de close where, toutes les lignes de la table vont être supprimées ;
 - `DELETE FROM` table `WHERE` id = 1 ;
 - `TRUNCATE TABLE` table // supprime toutes les lignes de la table et réinitialise l'index.

SQL - TCL

- Transaction de requêtes ;
- Une transaction est un ensemble de requêtes qui sont exécutées en un seul bloc. Ainsi, si une des requêtes du bloc échoue, on peut décider d'annuler tout le bloc de requêtes. Cela permet de revenir à l'état initial de la base avant la transaction ;
- **BEGIN / COMMIT / ROLLBACK.**

SQL - JDBC

- Qu'est-ce que Java DataBase Connectivity ?
- Cette API est constituée d'un ensemble d'interfaces et de classes qui permettent l'accès, à partir de programmes Java, à des données tabulaires :
 - L'établissement d'une connexion avec le SGBD ;
 - L'envoi de requêtes SQL au SGBD, à partir du programme Java ;
 - Le traitement, au niveau du programme, des données retournées par le SGBD ;
 - Le traitement des métadonnées de la connexion, de l'instruction ou des résultats ;
 - Le traitement des erreurs retournées par le SGBD lors de l'exécution d'une instruction.

SQL - JDBC

Comment ça fonctionne ?

- Connexion :
 - URL ;
 - Datasource.
- Statement vs Preparestatement ;
- ExecuteQuery & ExecuteUpdate ;
- ResultSet ;
- Import de masse :
 - Csv ;
 - SQL.

SQL - JDBC

Model Entity/Repository

- E/R (DTO/DAO) ;
- CRUD en POO avec le model E/R.

SQL - JDBC - TP

TP cinéma

- Installation des outils ;
- TP.