

EC3 / Bases de données - SQL

Master 2 - Semestre 3

INTRODUCTION

PRÉSENTATION



Fabien Pageot, Geospatial Engineer chez PUR

Contact : fabien.pageot@gmail.com

Site du cours : https://fabienpgt.github.io/cy_bdd_sql/

PROGRAMME PÉDAGOGIQUE

Séance 1	Mardi 9 septembre, 10h30 - 12h30	Concepts Clés
Séance 2	Mardi 16 septembre, 10h30 - 12h30	Modélisation Conceptuelle (MCD)
Séance 3	Mardi 23 septembre, 10h30 - 12h30	Du MCD au Modèle Logique (MLD)
Séance 4	Mardi 30 septembre, 10h30 - 12h30	Passage au SQL + Introduction à PostgreSQL et DBeaver
Séance 5	Mardi 14 octobre, 8h30 - 12h30	SQL - Requête simple et Jointure
Séance 6	Mardi 21 octobre, 10h30 - 12h30	SQL - Requêtes imbriquées
Séance 7	Mardi 4 novembre, 10h30 à 20h30	SQL - Agrégation de données
Séance 8	Mardi de 18 novembre, 10h30 - 12h30	Révisions
Séance 9	Mardi 25 Novembre, 10h30 - 12h30	Examen Final

MODALITÉS D'ÉVALUATION

- **Devoir maison (30 %)** : Exercice de modélisation conceptuelle, passage au modèle logique et traduction en SQL.
- **Examen final (70 %)** : Épreuve pratique : modélisation et requêtage d'un jeu de données.

COURS 1 - CONCEPTS CLÉS

LES TABLEURS

	A	B	C	D	E	F	G	H	I
1	Nom client	Adresse client	N° commande	Date commande	Produit	Quantité	Prix unitaire (€)	Mode de paiement	Statut livraison
2	Dupont Jean	12 rue de Paris, Lyon	CMD001	01/09/2025	Pomme	10	1,2	CB	Livré
3	Dupon J.	12 rue Paris, Lyon	CMD001	01/09/2025	Poire	5	1,5	CB	Livré
4	Martin Sophie	4 pl. Bellecour, Lyon	CMD002	01/09/2025	Banane	3	2,4	Virement	En préparation
5	Martine Sophie	4 place Bellecour, Lion	CMD002	01/09/2025	Pomme	20	1,2	Virement	En préparation
6	Nguyen Paul	85 av Jean Jaures, Marseille	CMD003	02/09/2025	Poire	15	1,5	CB	Livré
7	Dupuis Clara	14 rue Victor Hugaes, Bordeaux	CMD004	02/09/2025	Pêche	4	2,9	Espèces	Annulé
8	Durand Louis	2 impasse des Lillas, Nantes	CMD005	03/09/2025	Jus	10	3,8	CB	Livré
9	Petit Anne	11 rue Centrale, Toulouse	CMD006	04/09/2025	Banane	2	2,4	CB	En préparation
10	Garcia Maria	30 rue Natioante, Lille	CMD007	05/09/2025	Pomme	12	1,2	CB	Livré
11	Bernard Allain	7 chemin Vert, Rennes	CMD008	05/09/2025	Poire	5	1,5	Espèces	En préparation

- Redondance
- Difficulté de mise à jour
- Incohérences
- Pas de structure claire
- Pas de clés primaires/étrangères
- Pas de gestion fine des accès
- Pas de vraie gouvernance

QU'EST CE QU'UNE BASE DE DONNÉES ?

Organisation **logique** et **durable** d'informations

Permet de :

- **Stocker**
- **Retrouver**
- **Mettre à jour**
- **Partager**
- **Protéger**

Imposent un schéma et des contraintes

Gèrent les accès multiples et gros volumes



LES BASES DE DONNÉES RELATIONNELLES

client_id	nom	adresse
1	Dupont Jean	12 rue de Paris, Lyon
2	Martin Sophie	4 place Bellecour, Lyon
3	Nguyen Paul	85 av. Jean Jaurès, Marseille

CLÉ PRIMAIRE

CLÉ ÉTRANGÈRE

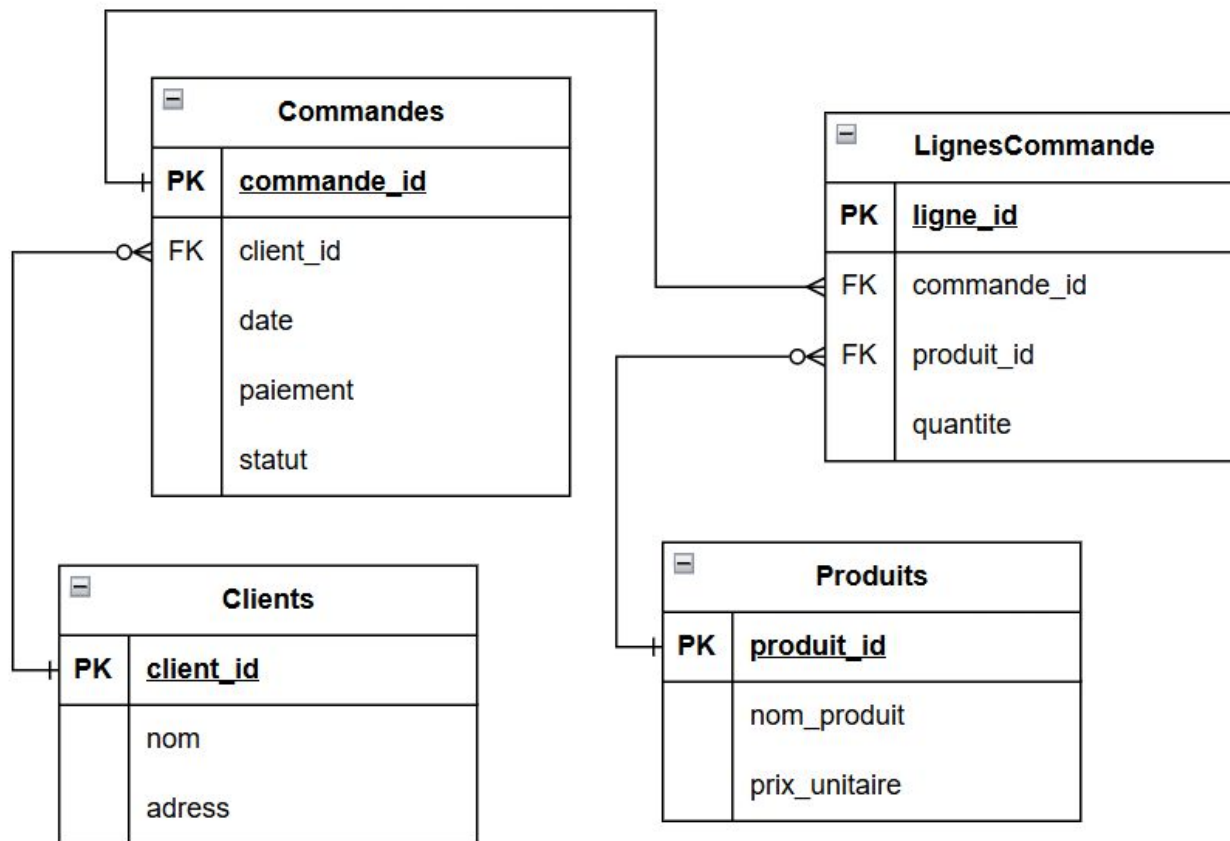
TABLE OU RELATION

ENREGISTREMENT

ATTRIBUT

commande_id	client_id	date	paiement	statut
101	1	2025-09-01	CB	Livré
102	2	2025-09-01	Virement	En préparation
103	3	2025-09-02	CB	Livré

EXEMPLE DE SCHÉMA D'UNE BASE DE DONNÉES RELATIONNELLES



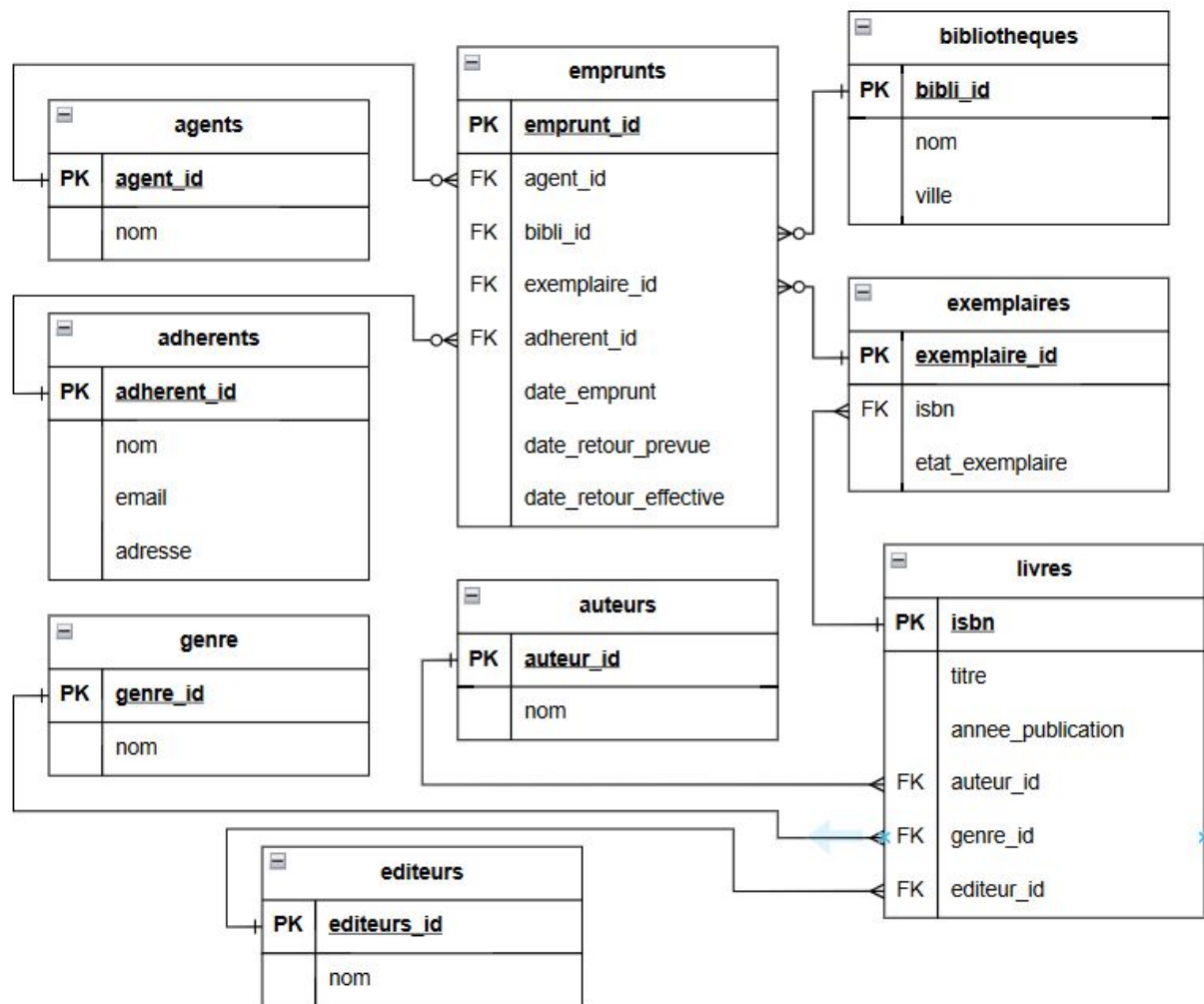
EXERCICE

Transformer un tableur en plusieurs tables

Objectif : transformer un fichier Excel « plat » en plusieurs tables relationnelles.

Étapes attendues

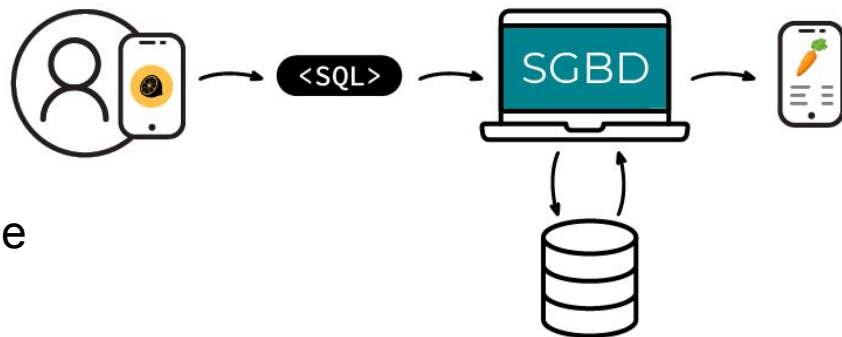
1. Identifier les entités principales
2. Créer une table pour chaque entité avec une **clé primaire (PK)**.
3. (Bonus) Déterminer les **clés étrangères (FK)** permettant de relier les tables.



QU'EST-CE QUE QU'UN SGBD?

Système de Gestion de Base de Données (SGBD)

- Moteur qui fait tourner la base
- Crée, maintient, exécute les requêtes
- Gère : sécurité, accès, logs, concurrence



Exemples de SGBD:

- Relationnels : PostgreSQL, MySQL, Oracle
- NoSQL : MongoDB, Redis, Neo4j

SGBD vs SGBDR :

- SGBD est le **terme générique** pour désigner le moteur.
- SGBDR (Système de Gestion de Base de Données Relationnelles) désigne un SGBD qui implémente le **modèle relationnel**

L'APPROCHE ACID

ACID est un acronyme qui désigne **quatre propriétés fondamentales** d'une transaction

Une **transaction** correspond à une **opération sur la base de données** : ajout d'un enregistrement, mise à jour, suppression, etc.

- **Atomicité** : une transaction est tout ou rien
- **Cohérence** : respecte toujours les contraintes d'intégrité
- **Isolation** : les transactions ne s'influencent pas entre elles
- **Durabilité** : les changements validés sont permanents, même après une panne

LE LANGAGE SQL

Le SQL (**Structured Query Language**) est un langage **déclaratif** et **standardisé**.

Il permet de :

- **décrire** la structure d'une base relationnelle
- **manipuler** les données
- **interroger** les données
- **gérer** les transactions
- **contrôler** les accès

On distingue notamment trois grandes familles de commandes SQL permettant :

- La définition des données (DDL)
- La manipulation des données (DML)
- Le requêtage de données (DQL)

Définir la structure (DDL)

Le **Data Definition Language** sert à créer/faire évoluer tables, contraintes, index, vues et schémas.

```
CREATE TABLE clients (  
    client_id SERIAL PRIMARY KEY,  
    nom        VARCHAR(100) NOT NULL,  
    adresse    VARCHAR(255) NOT NULL  
);
```


Manipuler les données (DML)

Le **Data Manipulation Language** couvre l'insertion, la mise à jour et la suppression.

```
INSERT INTO Clients (nom, adresse)  
VALUES ('Alice Dupont', '10 rue Victor Hugo, Paris');
```

```
UPDATE Clients  
SET adresse = '15 rue de Lyon, Paris'  
WHERE client_id = 1;
```

```
DELETE FROM Clients  
WHERE client_id = 999;
```

Interroger l'information (DQL)

Le **Data Query Language** s'articule autour de SELECT. On filtre, on joint, on agrège, on ordonne, on regroupe.

```
SELECT p.produit_id, p.nom_produit, SUM(lc.quantite) AS qte_vendue
FROM LignesCommande lc
JOIN Commandes c ON c.commande_id = lc.commande_id
JOIN Produits p ON p.produit_id = lc.produit_id
WHERE date_trunc('month', c.date) = date_trunc('month', CURRENT_DATE)
GROUP BY p.produit_id, p.nom_produit
ORDER BY qte_vendue DESC
LIMIT 5;
```

COURS 2 - De la modélisation à la conception d'une base de données

Partie 1 : Le Modèle Conceptuel de Données (MCD)

LA MÉTHODE MERISE

Concevoir une base de données = comme construire une maison

Nécessité de **plans avant de coder**

Les plans = la **modélisation**

Méthodologie **Merise**

- **MCD** : Modèle Conceptuel → Représentation indépendante de la technique
- **MLD** : Modèle Logique → Traduction vers le modèle relationnel
- **MPD** : Modèle Physique → Passage au SQL concret

Exemple fil rouge : **Le Festival Jazz de la Villette**

LES RÈGLES DE GESTION MÉTIER

Point de départ : **comprendre le besoin métier**

Formalisation en **règles de gestion**

Deux sources possibles :

- Utilisateurs → besoins exprimés
- Concepteur → analyse / entretiens

Questions clés :

- *Quelles infos conserver ?*
- *Quels objets relier ?*
- *Quelles contraintes respecter ?*

LES RÈGLES DE GESTION MÉTIER - Exemple du Festival

- Un **festivalier** peut assister à plusieurs **concerts**, et chaque **concert** peut accueillir plusieurs festivaliers.
- Un **concert** se déroule obligatoirement sur une seule **scène**, et chaque **scène** peut accueillir plusieurs **concerts**.
- Un **artiste** peut jouer dans plusieurs **concerts**, et chaque **concert** peut accueillir plusieurs **artistes**.
- Un **membre du personnel** peut travailler sur plusieurs **concerts**, et chaque concert mobilise plusieurs **personnels**.
- Chaque **concert** possède une **date** et une **heure de début** qui doivent être uniques pour ce **concert**.
- Un **festivalier** est identifié par son **nom**, **prénom** et **adresse e-mail**.
- Un **artiste** est identifié par son **nom**, son **style musical** et son **pays**.
- Un **membre du personnel** est identifié par son **nom**, **prénom** et **fonction**.

LE MODÈLE CONCEPTUELLE DE DONNÉES

Première étape de la modélisation

Objectif : Décrire les **objets** du monde réel et les **relations** qu'ils entretiennent entre eux

Deux notions fondamentales :

- **Entités** (objets du monde réel)
- **Associations** (liens entre entités)

LES ENTITÉS

Une entité est un ensemble homogène d'objets du monde réel.

Chaque objet de cet ensemble est appelé une **occurrence** de l'entité.

Artistes
id_artiste
Nom
Style musical
Pays

Concerts
id_concert
Date
Heure de début

Scènes
id_scène
Nom
Capacité d'accueil

Festivaliers
id_festivalier
Nom
Prénom
Adresse email
Numéro de pass

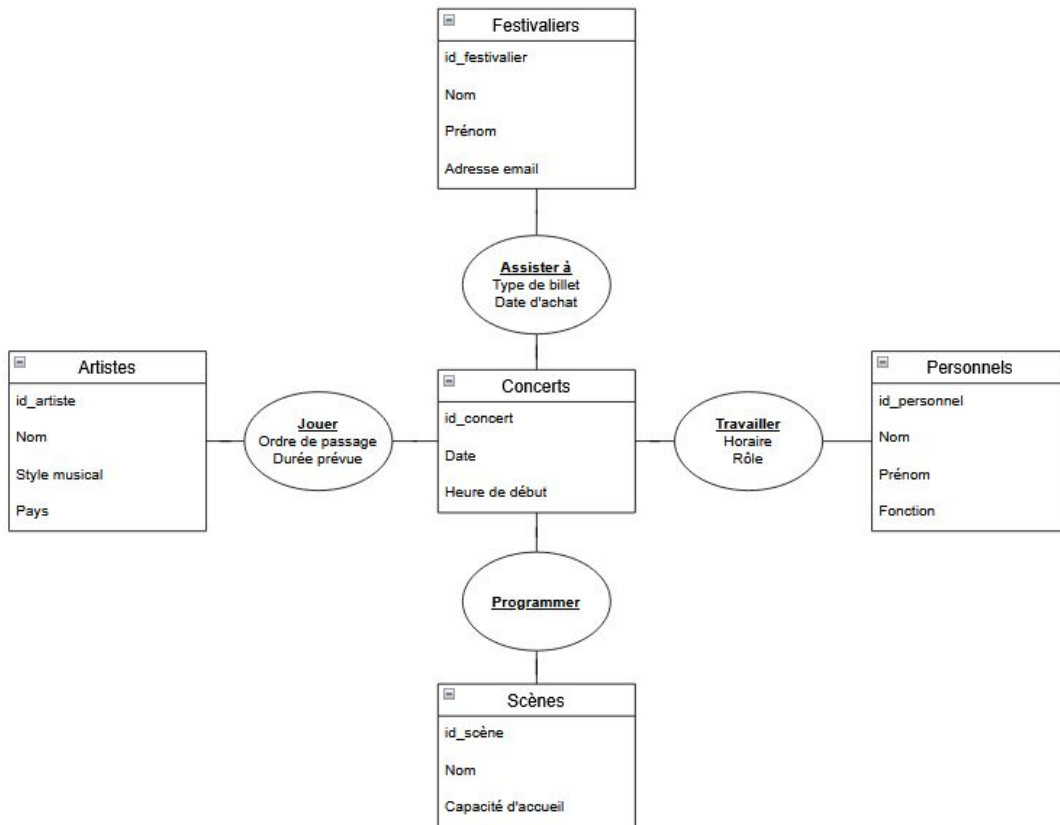
Personnels
id_personnel
Nom
Prénom
Fonction

LES ASSOCIATIONS

Une **association** établit un lien sémantique entre deux ou plusieurs entités.

Elle se nomme avec un **verbe**, à la forme active ou à la forme passive

Une association peut posséder ses propres **attributs**.

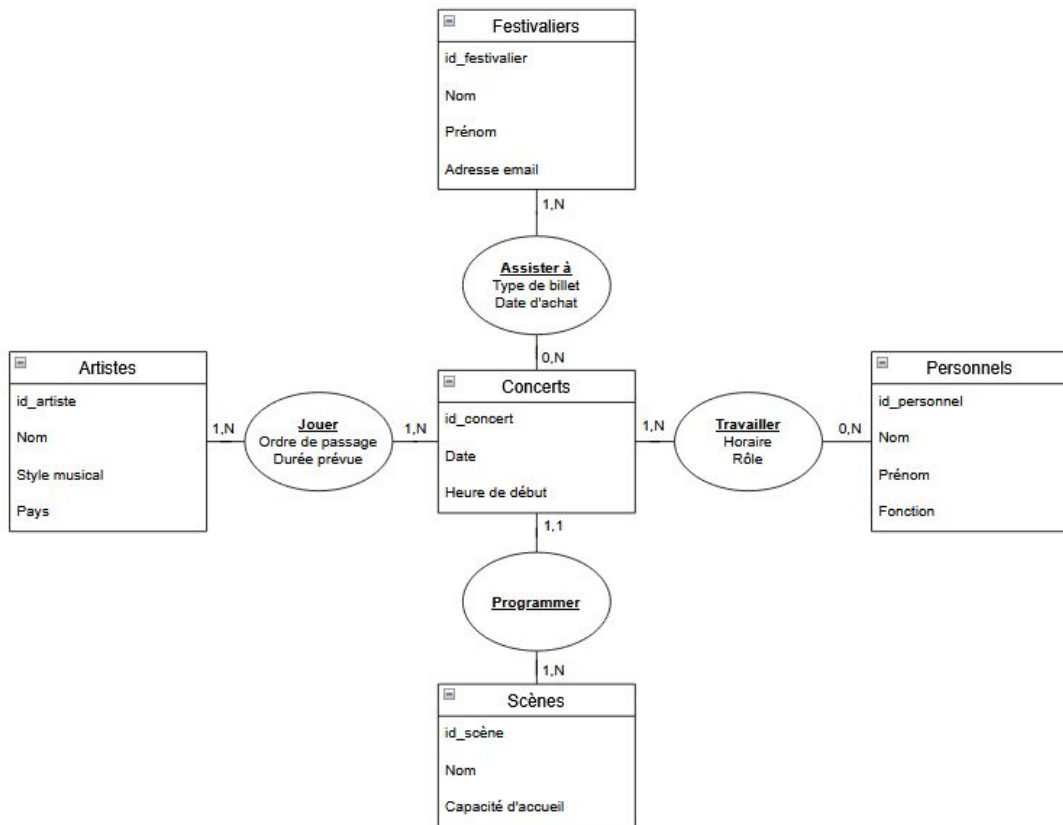


LES CARDINALITÉS

Les **cardinalités** précisent combien de fois une occurrence d'une entité peut participer à une association.

Elles se notent sous la forme **(min, max)**.

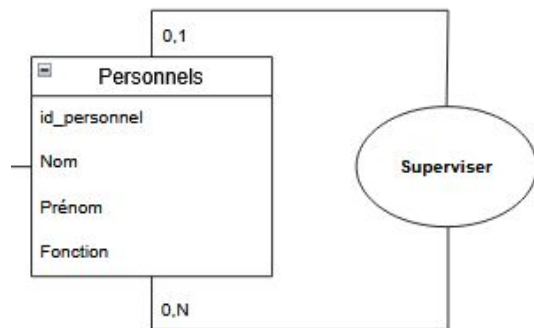
- $(0,1)$ → facultatif et unique
- $(1,1)$ → obligatoire et unique
- $(0,N)$ → facultatif et multiple
- $(1,N)$ → obligatoire et multiple



CAS PARTICULIERS

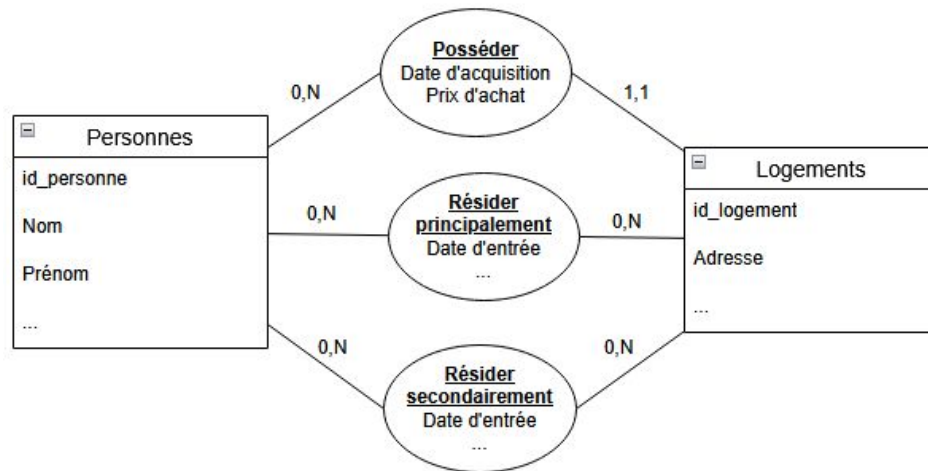
Association Réflexive

Une association reliée à elle-même.
Généralement pour exprimer une
relation hiérarchique



Association plurielle

Même paire d'entités reliée par
plusieurs associations



EXERCICE

Les propriétaires d'une salle de sport souhaitent mieux gérer leurs activités quotidiennes : organisation des cours, gestion des coachs, suivi des membres et de leurs présences. Ils font donc appel à vous, en tant que consultants en modélisation de bases de données, pour concevoir le modèle conceptuel de données (MCD) de leur futur système d'information.

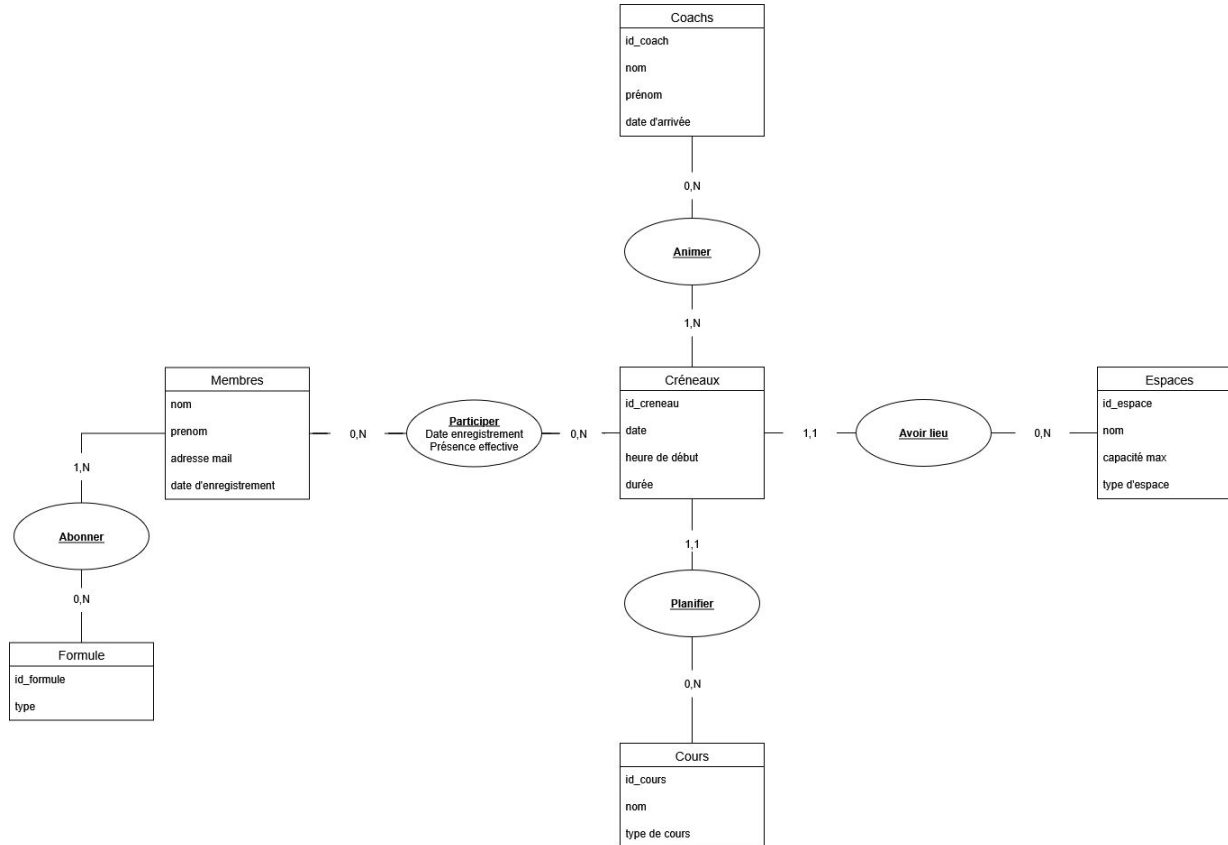
Règles de gestion métier fournies

- Une salle de sport propose plusieurs cours collectifs comme le yoga, le pilates, le crossfit ou encore des cours de self défense.
- Chaque Créneau d'un cours se déroule dans un espace dédié (salle de yoga, salle de musculation, espace extérieur, zone de tatami).
- Les créneaux d'un cours sont encadrés par des coach sportifs. On souhaite mémoriser leur nom, prénom et leur date d'arrivée dans le club.
- Les membres assistent aux créneaux d'un cours. Pour chaque membre, on enregistre : numéro d'adhésion, nom, prénom, adresse e-mail, date d'enregistrement et formule choisie (abonnement mensuel, annuel, cours individuels uniquement).
- Chaque cours est planifié sur un ou plusieurs créneaux horaires (date, heure de début, durée).
- On aimerait pouvoir savoir qui a participé réellement à un créneau (présence effective), ce qui peut différer des inscriptions prévues.

Règle de gestion supplémentaires

- Chaque créneau a lieu dans un seul espace, mais un espace peut accueillir plusieurs créneaux.
- Un créneau doit être encadré par au moins un coach, et peut être encadré par plusieurs coachs en même temps.
- Un cours peut être planifié sur plusieurs créneaux, mais chaque créneau correspond à un seul cours.
- Un coach peut encadrer plusieurs créneaux, mais peut aussi exister dans la base sans encadrer de cours (coach nouvellement recruté).
- Un cours peut être planifié sur plusieurs créneaux, mais chaque créneau correspond à un seul cours.
- Un membre peut s'inscrire à plusieurs créneaux, et un créneau peut accueillir plusieurs membres.
- Un créneau peut exister même sans inscriptions

EXERCICE - Proposition de Schéma



COURS 3 - De la modélisation à la conception d'une base de données

**Partie 2 : Du Modèle Logique de Données (MLD) au Modèle Physique de
Données (MPD)**

LE MODÈLE LOGIQUE DE DONNÉES

Rappel

- Le MCD décrit les entités, attributs et associations du monde réel.
- Le MLD traduit ce modèle en tables relationnelles.
- Objectif : préparer le passage vers le SQL concret (MPD).

Le MLD permet de traduire le MCD en structure compréhensible par un SGBD.

Deux grandes missions :

- **Transformer** les entités et associations en **tables**
- **Préparer** le passage vers le SQL concret (MPD)

CONVERSION D'UNE ENTITÉ

- Chaque entité devient une table.
- Les attributs deviennent des colonnes.
- L'identifiant devient la clé primaire (PK).

Exemple

`artistes(id_artiste, nom, style_musical, pays)`

artistes
<u>id_artiste</u>
nom
style_musicale
pays

Conversion d'associations 0/1,1 - 0/1,N

Cardinalité max 1 d'un côté et N de l'autre.

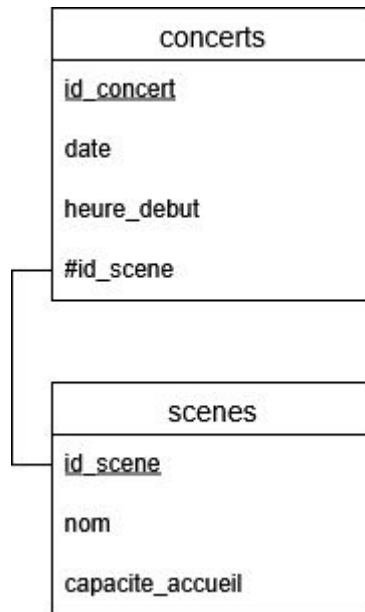
Ajout d'une **clé étrangère** dans la table côté (0,1) ou (1,1).

Exemple : Programmer (Concerts ↔ Scènes)

- Un concert est programmé sur une seule scène.
- Une scène accueille plusieurs concerts.

```
concerts(id_concert, date, heure_debut, #id_scene)
```

```
scenes(id_scene, nom, capacite_accueil)
```



Conversion d'associations (N – N)

Cardinalité N des deux côtés.

Création d'une nouvelle table associative.

PK = concaténation des clés primaires des deux entités.

Les attributs supplémentaires deviennent des colonnes.

Exemple : Jouer (Artistes ↔ Concerts)

`jouer(#id_artiste, #id_concert, ordre_passage, duree_prevue)`



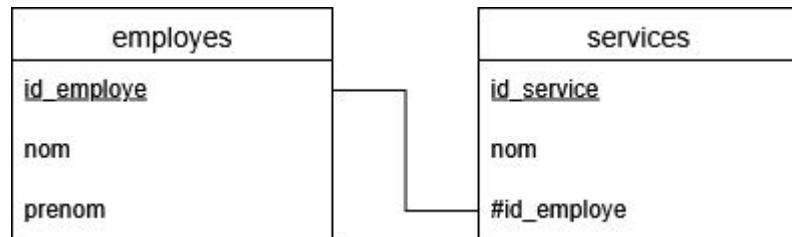
Conversion d'associations (1 – 1)

Deux options possibles :

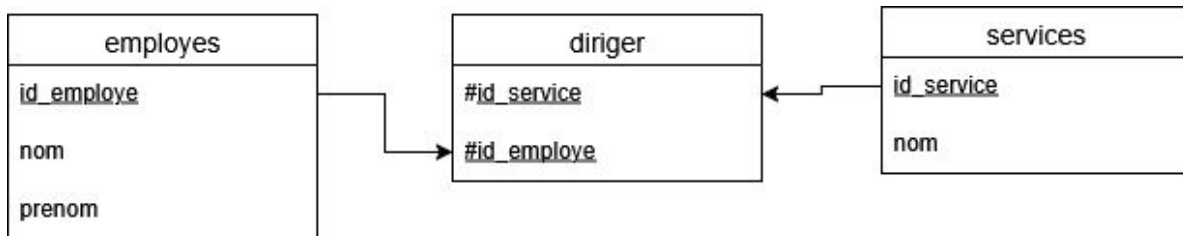
1. Ajouter une clé étrangère avec contrainte d'unicité.
2. Créer une table associative spécifique.

Exemple : Diriger (Employés ↔ Services)

Option 1 → clé étrangère `id_employe` dans `services`.



Option 2 → table `diriger(#id_service, #id_employe)`.



Exemple complet : Festival

artistes(id_artiste, nom, style_musical, pays)

scenes(id_scene, nom, capacite_accueil)

concerts(id_concert, date_concert, heure_debut, #id_scene)

festivaliers(id_festivalier, nom, prenom, adresse_email)

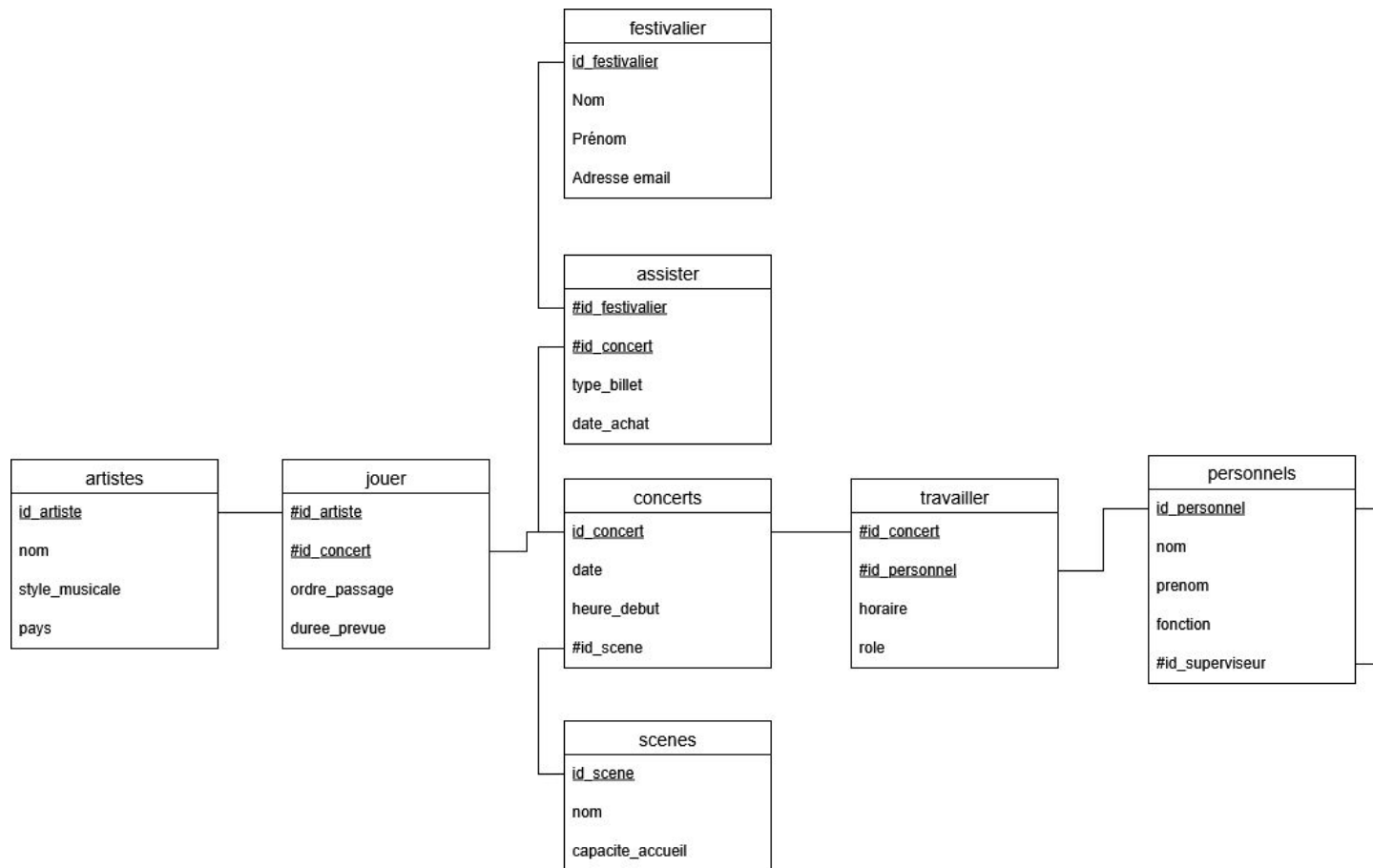
personnels(id_personnel, nom, prenom, fonction, #id_superviseur)

jouer(#id_artiste, #id_concert, ordre_passage, duree_prevue)

assister(#id_festivalier, #id_concert, type_billet, date_achat)

travailler(#id_concert, #id_personnel, horaire, role)

Exemple complet : Festival



LE MODÈLE PHYSIQUE DE DONNÉES

Traduction concrète du **MLD** dans un **SGBD** particulier (PostgreSQL, MySQL, Oracle, SQL Server...).

Le MPD consiste à **écrire les requêtes SQL** qui créent les tables avec :

- Colonnes
- Types de données
- Contraintes

Les types de données et leur importance

Bien choisir un type de données permet de :

- Assurer la **cohérence**
- Optimiser la **mémoire**.
- Améliorer les **performances**.
- Renforcer la **fiabilité**

Les types de données les plus courants

Numériques : `smallint`, `integer`, `bigint`, `numeric`, `real`, `double precision`, `serial`

Texte : `char(n)`, `varchar(n)`, `text`

Temporels : `date`, `time`, `timestamp`, `interval`

Booléens : `boolean`

Spécialisés : `uuid`, `json/jsonb`, `geometry`

Bonnes pratiques de nommage SQL

Règles à respecter dans le cadre du cours :

- Nom de tables au pluriel (`concerts`, `artistes`, `festivaliers`)
- Colonnes au singulier (`nom`, `date_concert`, `id_scene`)
- Pas d'accents, espaces ou majuscules (`snake_case`)
- Clés primaires explicites (`id_artiste`, `id_concert`)
- Clés étrangères reprenant le nom référencé (`id_scene` dans `concerts`)

CREATE TABLE

```
CREATE TABLE nom_table (  
    nom_colonne type_donnee [contrainte],  
    nom_colonne type_donnee [contrainte],  
    ...  
    [contraintes_table]  
);
```

- `nom_table` : nom de la table
- `nom_colonne` : nom de chaque colonne
- `type_donnee` : choix du type
- `contrainte` : règle appliquée à la colonne
- `contraintes_table` : contraintes globales

Contraintes les plus courantes

PRIMARY KEY : identifie chaque ligne de manière unique

FOREIGN KEY : relation avec une autre table

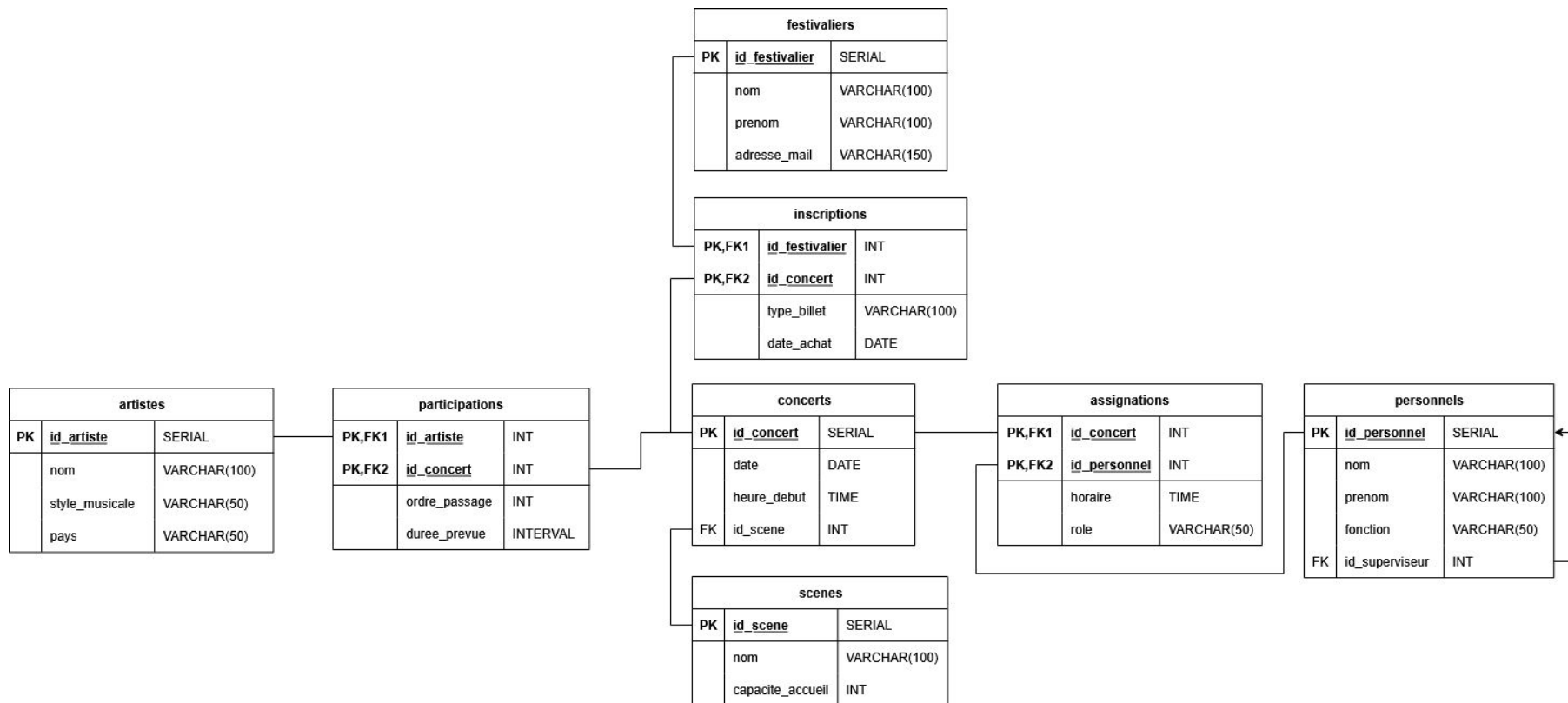
NOT NULL : valeur obligatoire

UNIQUE : empêche les doublons

DEFAULT : valeur par défaut

CHECK : impose une condition (ex. `capacite > 0`)

Exemple du Festival



DEVOIR MAISON

Vous devez réaliser, à partir d'un **exemple concret** (tiré de votre travail, de votre alternance, de votre stage, ou d'un autre contexte professionnel ou personnel de votre choix), une démarche complète de **modélisation et mise en œuvre d'une base de données**.

Ce devoir a pour but de vérifier votre capacité à :

- recueillir les règles de gestion d'un métier,
- les traduire dans un **modèle conceptuel de données (MCD)**,
- transformer ce modèle en **modèle logique de données (MLD)**,
- produire le **modèle physique de données (MPD)**,
- et enfin écrire les requêtes **SQL de création des tables**.