Université Lumières Lyon 2

UFR de Sciences Économiques et de Gestion

Master 1 MBFA – Monnaie, Banque, Finance, Assurance

Gestion d'une salle de musculation

Évaluation d'initiation à SQL

Réalisé par :

RAMET Maël

FALL Moussa

I - Introduction

1. Contexte général du projet

Dans le cadre du module d'Initiation à SQL du Master 1 MBFA, ce projet consiste à concevoir et implémenter une base de données complète pour la gestion d'une salle de musculation. L'objectif est de mettre en pratique les compétences acquises en modélisation conceptuelle et logique ainsi qu'en implémentation SQL sous PostgreSQL, à travers un cas concret où de nombreuses informations doivent être suivies et organisées.

2. Choix et justification de la problématique

La gestion d'une salle de musculation est une problématique pertinente car elle implique le suivi de multiples entités et relations : adhérents, abonnements, paiements, coachs, cours et réservations. La centralisation et la cohérence des données sont essentielles pour assurer un fonctionnement efficace de la salle, planifier les cours, suivre les adhésions et gérer les transactions financières.

Cette problématique a été choisie pour plusieurs raisons :

- Elle illustre clairement la notion de relations entre entités,
- Elle permet d'introduire des contraintes métiers concrètes (ex. un adhérent ne peut avoir qu'un seul abonnement actif),
- Elle offre un équilibre entre complexité et réalisme pour un projet SQL.

3. Identification des besoins

Pour répondre à cette problématique, la base de données doit permettre de :

- Gérer les informations personnelles des adhérents,
- Suivre les abonnements et les paiements associés,
- Organiser les cours collectifs et gérer leur planning,
- Suivre la participation des adhérents aux cours,
- Identifier et suivre l'activité des coachs,
- Fournir des informations exploitables pour le suivi des revenus et de la fréquentation.

4. Contraintes spécifiques

Les contraintes prises en compte pour rendre la base fonctionnelle sont :

- **Intégrité des données** : les relations entre adhérents, abonnements, paiements, cours et réservations doivent rester cohérentes,
- **Suivi des statuts** : les réservations doivent indiquer si elles sont confirmées, annulées ou en attente,
- **Gestion temporelle** : les dates d'abonnement, de paiement et de cours doivent être correctement tracées,
- **Éviter la redondance** : les informations communes (nom, prénom, type d'abonnement, nom de cours) doivent être centralisées.

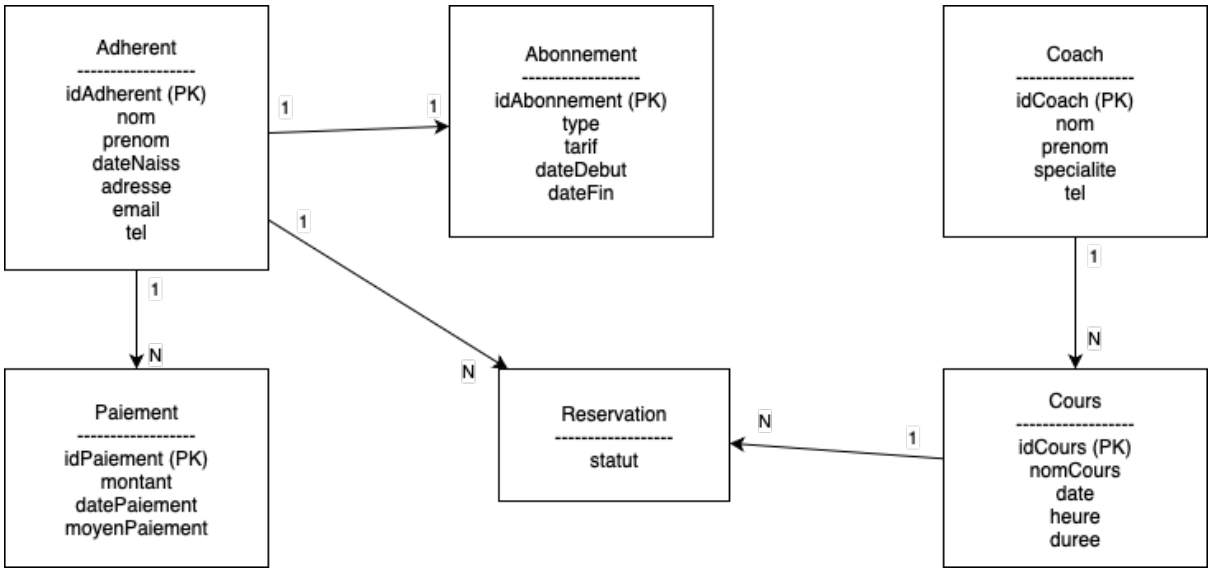
II – Modélisation conceptuelle

1. Présentation générale

Le modèle conceptuel repose sur six entités principales :

- Adhérent
- Abonnement
- Paiement
- Coach
- Cours
- Réservation

Ces entités sont reliées entre elles par des relations représentant les interactions naturelles dans une salle de musculation.



2. Description du Modèle Conceptuel :

Le modèle conceptuel de données présenté repose sur six entités principales qui traduisent les besoins identifiés dans la gestion d'une salle de musculation : suivi des adhérents, de leurs abonnements, des paiements, ainsi que de l'organisation des cours collectifs et des réservations.

1. Adhérent

Cette entité regroupe les informations personnelles de chaque membre inscrit à la salle : nom, prénom, date de naissance, adresse, adresse e-mail et numéro de téléphone.
Chaque adhérent est identifié par un **idAdherent**, qui constitue la clé primaire. Ces données permettent d'assurer un suivi individuel et de relier l'adhérent à ses abonnements, paiements et réservations.

2. Abonnement

L'entité Abonnement définit le type d'abonnement souscrit (mensuel, trimestriel ou annuel), son tarif, ainsi que les dates de début et de fin.
Chaque abonnement est identifié par un idAbonnement. La relation entre Adhérent et Abonnement est de type 1 à 1, car un adhérent ne peut posséder qu'un seul abonnement actif à la fois.

3. Paiement

L'entité Paiement enregistre les règlements effectués par les adhérents pour leur abonnement.
Elle contient le montant, la date du paiement et le mode utilisé (espèces, carte bancaire, virement, etc.).
Chaque paiement est identifié par un idPaiement, et la relation avec Adhérent est de type 1 à N, car un même adhérent peut effectuer plusieurs paiements pour son abonnement.

4. Coach

L'entité Coach regroupe les informations relatives aux entraîneurs : nom, prénom, spécialité (musculation, cardio, yoga, etc.) et coordonnées téléphoniques.
Chaque coach possède un idCoach unique. La relation avec Cours est de type 1 à N, puisqu'un coach peut animer plusieurs cours collectifs.

5. Cours

L'entité Cours décrit les différentes séances proposées aux adhérents : intitulé, date, heure et durée.
Chaque cours est associé à un coach responsable (relation 1-N entre Coach et Cours). L'identifiant principal est idCours.
Affiche tous les cours auxquels un adhérent spécifique est inscrit, avec le statut de sa réservation (confirmée, annulée ou en attente). Utile pour vérifier la participation de chaque adhérent.

6. Réservation

L'entité Réservation joue le rôle de table d'association entre les entités Adhérent et Cours.
Elle permet de gérer les inscriptions des adhérents aux différents cours collectifs.
Cette relation est de type N-N, car un adhérent peut réserver plusieurs cours, et un même cours peut accueillir plusieurs participants.
La table Réservation contient également un champ statut, qui permet de suivre l'état de la réservation (confirmée, annulée, en attente, etc.).

III – Modélisation logique

ADHERENT (idAdherent PK, nom, prenom, dateNaiss, adresse, email, tel, idAbonnement#)

La table ADHERENT contient toutes les informations personnelles des membres : nom, prénom, date de naissance, adresse, email et téléphone.

- idAdherent est la clé primaire et identifie de manière unique chaque adhérent.
- idAbonnement est une clé étrangère vers la table ABONNEMENT et représente la relation 1-1 : chaque adhérent possède un seul abonnement actif.

- Cette séparation permet de gérer facilement les adhérents et de relier chaque membre à son abonnement, sans duplication des informations.

ABONNEMENT (idAbonnement PK, type, tarif, dateDebut, dateFin)

La table ABONNEMENT contient les informations sur les différents types d'abonnements proposés : type (mensuel, trimestriel, annuel), tarif et dates de début et de fin.

- idAbonnement est la clé primaire qui identifie chaque abonnement.
- Cette table est séparée d'ADHERENT pour éviter la redondance des informations et faciliter la gestion des abonnements.
- Grâce à cette structure, il est facile de modifier un abonnement ou d'en ajouter un nouveau sans toucher aux données des adhérents.

PAIEMENT (idPaiement PK, montant, datePaiement, moyenPaiement, idAdherent#)

La table PAIEMENT enregistre les paiements effectués par les adhérents.

- idPaiement est la clé primaire.
- idAdherent est une clé étrangère vers ADHERENT, reflétant la relation 1-N : un adhérent peut effectuer plusieurs paiements, mais chaque paiement est associé à un seul adhérent.
- Cette table permet de suivre l'historique des paiements et les différents moyens de règlement utilisés.

COACH (idCoach PK, nom, prenom, specialite, tel)

La table COACH contient les informations des coachs : nom, prénom, spécialité et téléphone.

- idCoach est la clé primaire qui identifie chaque coach.

Les coachs peuvent animer plusieurs cours, ce qui sera représenté par la clé étrangère idCoach dans la table COURS.

COURS (idCours PK, nomCours, date, heure, duree, idCoach#)

La table COURS contient les informations sur les cours collectifs : nom du cours, date, heure, durée et le coach responsable.

- idCours est la clé primaire.
- idCoach est une clé étrangère vers COACH, reflétant la relation 1-N : chaque cours est animé par un seul coach, mais un coach peut animer plusieurs cours.

- Cette structure permet de planifier les cours et d'identifier le coach responsable pour chaque cours.

RESERVATION (idAdherent# PK, idCours# PK, statut)

La table RESERVATION est une table d'association qui gère la participation des adhérents aux cours, car il existe une relation M-N entre ADHERENT et COURS.

- Les clés primaires composées (idAdherent, idCours) garantissent qu'un adhérent ne peut pas réserver deux fois le même cours.
- Statut indique si la réservation est confirmée, annulée ou en attente.
- Cette table permet de suivre la participation des adhérents et de connaître le nombre de participants pour chaque cours.

IV – Implémentation SQL

1. Présentation

La base a été implémentée sous PostgreSQL via DBeaver. Chaque table correspond à une entité ou relation du modèle conceptuel. Les contraintes d'intégrité, les types de données et les relations entre les tables ont été respectés.

2. Création des tables

CREATE TABLE Abonnement (idAbonnement SERIAL PRIMARY KEY, type VARCHAR(50) NOT NULL, tarif NUMERIC(6,2) NOT NULL, dateDebut DATE NOT NULL, dateFin DATE);

CREATE TABLE Adherent (idAdherent SERIAL PRIMARY KEY, nom VARCHAR(100), prenom VARCHAR(100), dateNaiss DATE, adresse VARCHAR(255), email VARCHAR(150) UNIQUE, tel VARCHAR(20), idAbonnement INT REFERENCES Abonnement(idAbonnement));

CREATE TABLE Paiement (idPaiement SERIAL PRIMARY KEY, montant NUMERIC(6,2), datePaiement DATE, moyenPaiement VARCHAR(50), idAdherent INT REFERENCES Adherent(idAdherent));

CREATE TABLE Coach (idCoach SERIAL PRIMARY KEY, nom VARCHAR(100), prenom VARCHAR(100), specialite VARCHAR(100), tel VARCHAR(20));

CREATE TABLE Cours (idCours SERIAL PRIMARY KEY, nomCours VARCHAR(100), date DATE, heure TIME, duree INT, idCoach INT REFERENCES Coach(idCoach));

CREATE TABLE Reservation (idAdherent INT REFERENCES Adherent(idAdherent), idCours INT REFERENCES Cours(idCours), statut VARCHAR(50), PRIMARY KEY (idAdherent, idCours));

3. Peuplement des tables

```
-- ABONNEMENT
INSERT INTO Abonnement (type, tarif, dateDebut, dateFin) VALUES
('Mensuel', 50.00, '2025-09-01', '2025-09-30'),
('Trimestriel', 135.00, '2025-07-01', '2025-09-30'),
('Annuel', 480.00, '2025-01-01', '2025-12-31');

-- ADHERENT
INSERT INTO Adherent (nom, prenom, dateNaiss, adresse, email, tel, idAbonnement)
VALUES
('Martin', 'Alice', '1990-05-12', '12 rue des Lilas', 'alice.martin@email.com',
'0601020304', 1),
('Dupont', 'Jean', '1985-03-22', '45 avenue Victor Hugo', 'jean.dupont@email.com',
'0605060708', 2),
('Durand', 'Claire', '1992-11-05', '7 boulevard de la République',
'claire.durand@email.com', '0611121314', 3),
('Lemoine', 'Paul', '1988-02-14', '23 rue des Fleurs', 'paul.lemoine@email.com',
'0622334455', 1),
('Morel', 'Sophie', '1995-08-30', '89 avenue de Paris', 'sophie.morel@email.com',
'0677889900', 2);

-- COACH
INSERT INTO Coach (nom, prenom, specialite, tel) VALUES
('Leclerc', 'Marc', 'Musculation', '0622334455'),
('Moreau', 'Sophie', 'Yoga', '0677889900'),
('Bernard', 'Lucas', 'Cardio', '0611223344');

-- COURS
INSERT INTO Cours (nomCours, date, heure, duree, idCoach) VALUES
('Renforcement musculaire', '2025-10-08', '18:00', 60, 1),
('Yoga détente', '2025-10-08', '19:30', 45, 2),
('Cardio intense', '2025-10-09', '17:00', 50, 3),
('Musculation débutant', '2025-10-10', '18:00', 60, 1),
('Yoga avancé', '2025-10-11', '19:30', 60, 2);

-- PAIEMENT
INSERT INTO Paiement (montant, datePaiement, moyenPaiement, idAdherent)
VALUES
(50.00, '2025-09-01', 'Carte bancaire', 1),
(135.00, '2025-07-01', 'Virement', 2),
(480.00, '2025-01-01', 'Espèces', 3),
(50.00, '2025-09-01', 'Espèces', 4),
(135.00, '2025-07-01', 'Carte bancaire', 5);

-- RESERVATION
INSERT INTO Reservation (idAdherent, idCours, statut) VALUES
(1, 1, 'Confirmé'),
(2, 1, 'Confirmé'),
(3, 2, 'Annulé'),
(4, 3, 'Confirmé'),
(5, 2, 'Confirmé'),
```

(1, 4, 'En attente'),
(2, 5, 'Confirmé'),
(3, 5, 'Confirmé'),
(4, 1, 'Confirmé'),
(5, 3, 'Confirmé');

4. Requêtes SQL

1. **Lister les adhérents actifs**

```
SELECT A.nom, A.prenom
FROM Adherent A
JOIN Abonnement B ON A.idAbonnement = B.idAbonnement
WHERE B.dateFin IS NULL OR B.dateFin >= CURRENT_DATE;
```

Cette requête affiche les noms et prénoms des adhérents dont l'abonnement est toujours valide.
On utilise un JOIN pour relier chaque adhérent à son abonnement.
La condition dateFin IS NULL OR dateFin >= CURRENT_DATE permet de filtrer uniquement les abonnements actifs.

2. **Planning des cours d'un coach donné**

```
SELECT C.nomCours, C.date, C.heure
FROM Cours C
JOIN Coach Co ON C.idCoach = Co.idCoach
WHERE Co.nom = 'Leclerc';
```

Affiche tous les cours animés par le coach "Leclerc". Le JOIN permet de relier chaque cours à son coach. Utile pour consulter rapidement le planning d'un coach.

3. **Nombre de participants par cours**

```
SELECT C.nomCours, COUNT(R.idAdherent) AS nbParticipants
FROM Cours C
LEFT JOIN Reservation R ON C.idCours = R.idCours
GROUP BY C.nomCours;
```

Compte combien d'adhérents participent à chaque cours. Le LEFT JOIN permet d'inclure tous les cours, même ceux sans réservation. Le GROUP BY regroupe les résultats par cours.

4. **Historique des paiements d'un adhérent**

```
SELECT P.datePaiement, P.montant, P.moyenPaiement
FROM Paiement P
JOIN Adherent A ON P.idAdherent = A.idAdherent
WHERE A.nom = 'Martin';
```

Montre tous les paiements effectués par l'adhérent "Martin". Permet de suivre l'historique des paiements et le mode de règlement utilisé.

5. **Voir les cours réservés par un adhérent**

```
SELECT C.nomCours, C.date, C.heure, R.statut
FROM Cours C
JOIN Reservation R ON C.idCours = R.idCours
WHERE R.idAdherent = 1;
```

Affiche tous les cours auxquels un adhérent spécifique est inscrit, avec le statut de sa réservation (confirmée, annulée ou en attente). Utile pour vérifier la participation de chaque adhérent.

6. **Revenu total sur l'année 2025**

```
SELECT SUM(montant) AS revenu_annuel
FROM Paiement
WHERE EXTRACT(YEAR FROM datePaiement) = 2025;
```

La requête additionne tous les paiements effectués en 2025 et retourne le total sous le nom revenu_annuel.

7. **Voir tous les cours avec le nom du coach**

```
SELECT c.nomCours, c.date, c.heure, c.duree, co.nom AS coach_nom, co.prenom
AS coach_prenom
FROM Cours c
JOIN Coach co ON c.idCoach = co.idCoach
ORDER BY c.date, c.heure;
```

Cette requête affiche tous les cours avec leur date, heure, durée, et le nom/prénom du coach, triés dans l'ordre chronologique.

Conclusion

Ce projet nous a permis de concevoir et d'implémenter une base de données relationnelle destinée à la gestion complète d'une salle de musculation. La base prend en compte l'ensemble des besoins essentiels : suivi des adhérents, gestion des abonnements et des paiements, organisation des cours collectifs et suivi des réservations, avec des relations entre les tables clairement définies. Elle offre ainsi un outil efficace pour suivre les adhésions, les paiements, le planning des cours et la participation des membres.

Le modèle conceptuel et logique a été conçu de manière normalisée et cohérente. Les tables sont reliées avec intégrité référentielle grâce aux clés primaires et étrangères, et les requêtes SQL développées permettent d'extraire des informations clés, telles

que le planning des cours, le nombre de participants, l'historique des paiements ou encore le revenu généré par la salle.

En définitive, ce projet montre qu'une base de données bien structurée constitue un véritable levier pour la gestion quotidienne d'une salle de sport, en fournissant des informations fiables et exploitables pour la prise de décision.