

Main Project  
Comp 353-CD  
Dr. Khaled Jababo  
August 10<sup>th</sup> 2025

Team ID: ntc353\_1  
Angel Asencios (40176253)  
Haaris Muhammad Khawaja (40261525)  
Ali Sher (40255236)  
Farhan Ali Khan (40287970)

AITS & GUI login  
Username: ntc353\_1  
Password: Draty@01

## **ORIGINALITY FORM**

We certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality.

**Course Number:** COMP 353 CD

**Instructor:** Professor Khaled Jababo

**Signed:** Angel Asencios (40176253)  
Haaris Muhammad Khawaja (40261525)  
Ali Sher (40255236)  
Farhan Ali Khan (40287970)

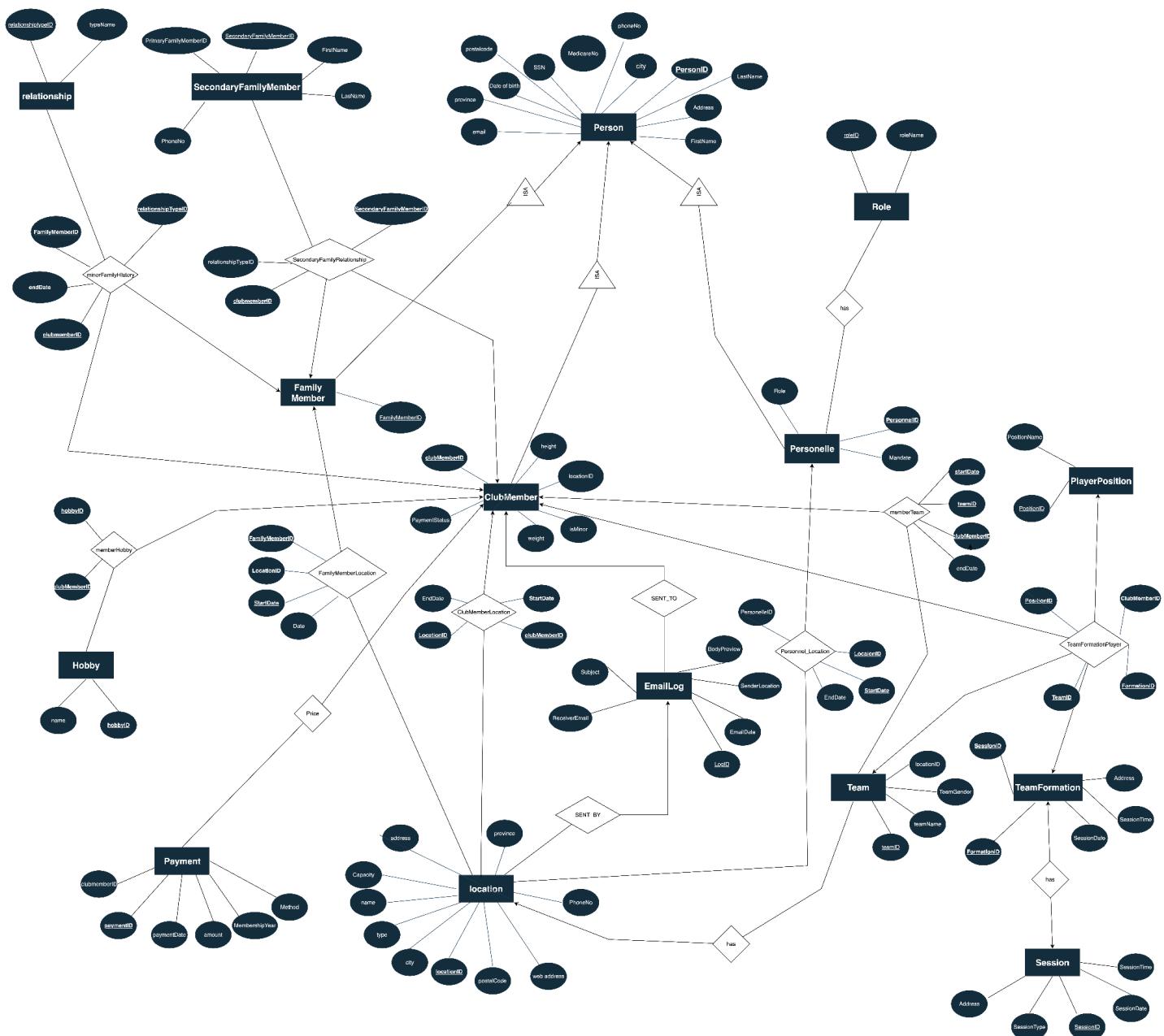
**Date:** August 10, 2025

## **Contributions:**

Angel Asencios (40176253)	<ul style="list-style-type: none"><li>● E/R database design &amp; diagram</li><li>● E/R constraints (Question 2)</li><li>● DDL/DML Statements</li><li>● E/R to Relations Conversions</li><li>● Triggers</li><li>● Project report</li><li>● GUI development, testing and deployment</li><li>● BCNF/3NF analysis (Question 4 &amp;5)</li></ul>
Haaris Muhammad Khawaja (40261525)	<ul style="list-style-type: none"><li>● E/R database design &amp; diagram</li><li>● Data population for all tables</li><li>● SQL Queries (Questions 1–7)</li><li>● Bug fixing and testing query outputs</li><li>● Creating tables and validating sample data</li><li>● Assisting in ensuring integrity constraints are met</li></ul>
Ali Sher (40255236)	<ul style="list-style-type: none"><li>● E/R database design &amp; diagram</li><li>● SQL Queries (Questions 8–15)</li><li>● Query optimization and formatting results</li><li>● Bug fixing and testing data step-by-step</li><li>● Contributed to writing the project report</li><li>● Assisted in verifying final outputs for submission</li></ul>
Farhan Ali Khan (40287970)	<ul style="list-style-type: none"><li>● E/R database design &amp; diagram</li><li>● SQL Queries (Questions 16–22)</li><li>● Triggers and stored procedures implementation</li><li>● Creating tables and validating constraints</li><li>● Bug fixing and verifying data accuracy</li></ul>

- Assisted in final review of SQL scripts and report

## 1. E/R Diagram (CONCEPTUAL MODEL)



## **2. Constraints Not Captured by E/R Diagram**

### **❖ Age Constraints:**

- Club members must be at least 11 years old at registration
- Minor status: age 11-17, Major status: over 18

### **❖ Payment Business Rules:**

- Minor members: \$100/year, Major members: \$200/year
- Maximum 4 installments per year
- Excess payments become donations

### **❖ Team Formation Rules:**

- All team players must be from same location as team
- All players in formation must be same gender
- 3-hour minimum gap between team formations for same player

### **❖ Location Capacity:**

- Number of active members must be less than or equal than location capacity

### **❖ Temporal Constraints:**

- Only one Head location allowed in system at the same time

### **❖ Email Automation:**

- Weekly Sunday email generation for upcoming sessions

## **3. E/R to relations conversion**

### **Entity sets**

1. **Person**(PersonID(PK), FirstName, LastName, DateOfBirth, SSN (Unique, Not Null), MedicareNo (Unique), PhoneNo, Address, City, Province, PostalCode, Email)

- a. Primary Key: PersonID
  - b. Unique Constraints: SSN, MedicareNo
2. **FamilyMember**(FamilyMemberID(PK, FK to Person))
- a. Primary Key: FamilyMemberID
  - b. Foreign Key: FamilyMemberID references Person.PersonID
3. **Personnel**(PersonnelID(PK, FK to Person), Role, Mandate)
- a. Primary Key: PersonnelID
  - b. Foreign Key: PersonnelID references Person.PersonID
4. **ClubMember**(ClubMemberID(PK, FK to Person), Height, Weight, isMinor, locationID(FK), PaymentStatus(Boolean))
- a. Primary Key: ClubMemberID
  - b. Foreign Keys: ClubMemberID references Person.PersonID, locationID references Location.locationID
5. **Location**(locationID(PK), name, Type(Head, Branch), Address, City, Province, PostalCode, PhoneNo, WebAddress, Capacity)
- a. Primary Key: locationID
6. **Team**(teamID(PK), teamName, team\_gender, LocationID(FK))
- a. Primary Key: teamID
  - b. Foreign Key: LocationID references Location.locationID
7. **Hobby**(hobbyID(PK), name)
- a. Primary Key: hobbyID
8. **Payment**(PaymentID(PK), ClubMemberID(FK), PaymentDate, Amount, Method, MembershipYear, InstallmentNumber (1-4))
- a. Primary Key: PaymentID
  - b. Foreign Key: ClubMemberID references ClubMember.ClubMemberID
9. **Relationship**(relationshipTypeID(PK), TypeName)
- a. Primary Key: relationshipTypeID
10. **Session**(SessionID(PK), SessionDate, SessionTime, Address, SessionType(Training,Game))
- a. Primary Key: SessionID
11. **TeamFormation**(FormationID(PK), SessionID(FK), Team1ID(FK), Team2ID(FK), Team1Score, Team2Score)

- a. Primary Key: FormationID
- b. Foreign Keys: SessionID references Session.SessionID, Team1ID references Team.teamID, Team2ID references Team.teamID

12. **PlayerPosition**(PositionID(PK), PositionName(Setter,Libero,etc))

- a. Primary Key: PositionID

13. **SecondaryFamilyMember**(SecondaryFamilyMemberID(PK), FirstName, LastName, PhoneNo, PrimaryFamilyMemberID(FK))

- a. Primary Key: SecondaryFamilyMemberID
- b. Foreign Key: PrimaryFamilyMemberID references FamilyMember.FamilyMemberID

14. **EmailLog**(LogID(PK), EmailDate, SenderLocation, ReceiverEmail, Subject, BodyPreview)

- a. Primary Key: LogID

### **Relationships as tables**

15. **ClubMemberLocation**(ClubMemberID, LocationID, StartDate, EndDate)

- a. Primary Key: (ClubMemberID, LocationID, StartDate)
- b. Foreign Keys: ClubMemberID references ClubMember.ClubMemberID, LocationID references Location.locationID

16. **Personnel\_Location**(PersonnelID(FK), LocationID(FK), StartDate(FK), EndDate)

- a. Primary Key: (PersonnelID, LocationID, StartDate)
- b. Foreign Keys: PersonnelID references Personnel.PersonnelID, LocationID references Location.locationID

17. **FamilyMemberLocation**(FamilyMemberID(FK), LocationID(FK), StartDate(FK), EndDate)

- a. Primary Key: (FamilyMemberID, LocationID, StartDate)
- b. Foreign Keys: FamilyMemberID references FamilyMember.FamilyMemberID, LocationID references Location.locationID

18. **memberHobby**(ClubMemberID, hobbyID)

- a. Primary Key: (ClubMemberID, hobbyID)
- b. Foreign Keys: ClubMemberID references ClubMember.ClubMemberID, hobbyID references Hobby.hobbyID

19. **memberTeam**(startDate, teamID, ClubMemberID, endDate)

- a. Primary Key: (ClubMemberID, teamID, startDate)
- b. Foreign Keys: ClubMemberID references ClubMember.ClubMemberID, teamID references Team.teamID

20. **minorFamilyHistory**(relationshipTypeID(FK), FamilyMemberID(FK),

ClubMemberID(FK), endDate)

- a. Primary Key: (relationshipTypeID, FamilyMemberID, ClubMemberID)
- b. Foreign Keys: FamilyMemberID references FamilyMember.FamilyMemberID, ClubMemberID references ClubMember.ClubMemberID, relationshipTypeID references Relationship.relationshipTypeID

21. **TeamFormationPlayer**(FormationID(FK), TeamID(FK), ClubMemberID(FK),  
PositionID(FK))

- a. Primary Key: (FormationID, TeamID, ClubMemberID)
- b. Foreign Keys: FormationID references TeamFormation.FormationID, TeamID references Team.teamID, ClubMemberID references ClubMember.ClubMemberID, PositionID references PlayerPosition.PositionID

22. **SecondaryFamilyRelationship**(SecondaryFamilyMemberID(FK),

ClubMemberID(FK), relationshipTypeID(FK))

- a. Primary Key: (SecondaryFamilyMemberID, ClubMemberID)
- b. Foreign Keys: SecondaryFamilyMemberID references SecondaryFamilyMember.SecondaryFamilyMemberID, ClubMemberID references ClubMember.ClubMemberID, relationshipTypeID references Relationship.relationshipTypeID

#### **4. Are all your relations in BCNF?**

No, not all relations are in BCNF. 18 out of 22 the relations are in BCNF.

## **Relations in BCNF:**

- **Person** - PersonID is the only determinant and it's a superkey
- **Location** - LocationID determines all other attributes and is a superkey
- **ClubMember** - ClubMemberID is the only determinant and it's a superkey
- **Team** - TeamID determines all other attributes and is a superkey
- **Session** - SessionID determines all other attributes and is a superkey
- **Payment** - PaymentID determines all other attributes and is a superkey
- **Hobby** - HobbyID determines Name and is a superkey
- **Relationship** - RelationshipTypeID determines TypeName and is a superkey
- **PlayerPosition** - PositionID determines PositionName and is a superkey
- **memberHobby** - Only contains key attributes (ClubMemberID, HobbyID)
- **FormationTeam** - Only contains key attributes (FormationID, TeamID)
- **EmailLog** - LogID determines all other attributes and is a superkey
- **memberHobby** - Only contains key attributes (ClubMemberID, HobbyID)
- **FormationTeam** - Only contains key attributes (FormationID, TeamID)
- **ClubMemberLocation** - Primary key (ClubMemberID, LocationID, StartDate) determines EndDate directly
- **SecondaryFamilyRelationship** - Only contains key attributes (SecondaryFamilyMemberID, ClubMemberID, RelationshipTypeID)

## **Relations NOT in BCNF:**

### **I. Personnel\_Location**

- **BCNF Violation:** The functional dependency (PersonnelID, StartDate) → LocationID exists, but (PersonnelID, StartDate) is not a superkey of the relation
- **Explanation:** A personnel member can only be at one location at any given start date, so PersonnelID and StartDate together determine LocationID, but this combination is not a superkey since it doesn't determine the full primary key

### **II. PlayerFormation**

- **BCNF Violation:** The functional dependency FormationID → TeamID exists (a formation determines which teams are involved), but FormationID is not a superkey of this relation
- **Explanation:** While FormationID determines which teams play, it doesn't determine the individual players or their positions, so it's not a superkey

### **III. minorFamilyHistory**

- **BCNF Violation:** The functional dependency (FamilyMemberID, ClubMemberID) → RelationshipTypeID exists, but (FamilyMemberID, ClubMemberID) is not a superkey
- **Explanation:** A family member has the same relationship type with a specific club member regardless of time periods, but this pair doesn't determine the StartDate component of the primary key

### **IV. FamilyMemberLocation**

- **BCNF Violation:** The functional dependency (FamilyMemberID, StartDate) → LocationID exists, but (FamilyMemberID, StartDate) is not a superkey of the relation
- **Explanation:** Similar to Personnel\_Location, a family member can only be at one location at any given start date

## **5. For any relation in your database, if it is not in BCNF, then show that it is in 3NF**

### **I. Personnel\_Location:**

- **1NF:** All attributes (PersonnelID, LocationID, StartDate, EndDate) are atomic values
- **2NF:** No partial dependencies exist, EndDate depends on the full primary key (PersonnelID, LocationID, StartDate), not just part of it
- **3NF:** No transitive dependencies exist - all non-key attributes (EndDate) depend directly on the primary key, not through other non-key attributes

### **II. PlayerFormation:**

- **1NF:** All attributes (FormationID, ClubMemberID, PositionID) are atomic values
- **2NF:** No partial dependencies exist, PositionID depends on the full primary key (FormationID, ClubMemberID) because a player's position is determined by both the specific formation and the specific player

- **3NF:** No transitive dependencies exist, PositionID depends directly on the primary key combination, not through any intermediate non-key attributes

### **III. minorFamilyHistory:**

- **1NF:** All attributes (FamilyMemberID, ClubMemberID, StartDate, RelationshipTypeID, EndDate) are atomic values
- **2NF:** No partial dependencies exist - both RelationshipTypeID and EndDate depend on the full primary key (FamilyMemberID, ClubMemberID, StartDate), as the relationship and its end date are determined by the specific family member, club member, and time period
- **3NF:** No transitive dependencies exist, RelationshipTypeID and EndDate depend directly on the primary key, not through other non-key attributes

### **IV. FamilyMemberLocation:**

- **1NF:** All attributes (FamilyMemberID, LocationID, StartDate, EndDate) are atomic values
- **2NF:** No partial dependencies exist, EndDate depends on the full primary key (FamilyMemberID, LocationID, StartDate), not just part of it
- **3NF:** No transitive dependencies exist - all non-key attributes (EndDate) depend directly on the primary key, not through other non-key attributes

## **6. TRIGGERS**

DELIMITER //

-- 1. PREVENT PLAYER TIME CONFLICTS

-- "If a player is to be assigned to two team formations on the same day,

-- then at least three hours of difference should be set between the start time"

CREATE TRIGGER PreventPlayerTimeConflict

BEFORE INSERT ON TeamFormationPlayer

FOR EACH ROW

BEGIN

```
DECLARE conflict_count INT DEFAULT 0;
DECLARE session_date DATE;
DECLARE session_time TIME;
DECLARE error_message VARCHAR(255);

SELECT s.SessionDate, s.SessionTime
INTO session_date, session_time
FROM Session s
INNER JOIN TeamFormation tf ON s.SessionID = tf.SessionID
WHERE tf.FormationID = NEW.FormationID;
```

```
SELECT COUNT(*) INTO conflict_count
FROM TeamFormationPlayer tfp
INNER JOIN TeamFormation tf ON tfp.FormationID = tf.FormationID
INNER JOIN Session s ON tf.SessionID = s.SessionID
WHERE tfp.ClubMemberID = NEW.ClubMemberID
AND s.SessionDate = session_date
AND tfp.FormationID != NEW.FormationID
AND (
    (session_time >= s.SessionTime AND
    TIMEDIFF(session_time, s.SessionTime) < '03:00:00')
    OR
    (session_time <= s.SessionTime AND
    TIMEDIFF(s.SessionTime, session_time) < '03:00:00')
);
```

```
IF conflict_count > 0 THEN  
    SET error_message = CONCAT('Player assignment rejected: Player with ID ',  
        NEW.ClubMemberID, ' already has a formation on ', session_date,  
        ' within 3 hours of ', session_time);  
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = error_message;  
END IF;  
END //
```

## -- 2. ENFORCE MINIMUM AGE REQUIREMENT

-- "A new club member must be at least 11 years old at the time of registration"

```
CREATE TRIGGER EnforceMinimumAge
```

```
BEFORE INSERT ON ClubMember
```

```
FOR EACH ROW
```

```
BEGIN
```

```
DECLARE member_age INT;
```

```
DECLARE birth_date DATE;
```

```
SELECT DateOfBirth INTO birth_date
```

```
FROM Person WHERE PersonID = NEW.ClubMemberID;
```

```
SET member_age = TIMESTAMPDIFF(YEAR, birth_date, CURDATE());
```

```
IF member_age < 11 THEN
```

```
    SIGNAL SQLSTATE '45000'
```

```
SET MESSAGE_TEXT = 'Club member must be at least 11 years old to register';

END IF;

END //
```

### -- 3. Enforce Unique Active Team Mmebership

-- Make sure that a club member is not in two teams at the same time

```
CREATE TRIGGER PreventMultipleActiveTeams
```

```
    BEFORE INSERT ON memberTeam
```

```
    FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE conflictCount INT;
```

```
    SELECT COUNT(*) INTO conflictCount
```

```
    FROM memberTeam
```

```
    WHERE ClubMemberID = NEW.ClubMemberID
```

```
    AND (
```

```
        (NEW.startDate BETWEEN startDate AND IFNULL(endDate,'9999-12-31')) OR
```

```
        (NEW.endDate IS NULL AND endDate is NULL) OR
```

```
        (startDate BETWEEN NEW.startDate AND IFNULL(NEW.endDate, '9999-12-31'))
```

```
    );
```

```
    IF conflictCount > 0 THEN
```

```
        SIGNAL SQLSTATE '45000'
```

```
        SET MESSAGE_TEXT = 'Club member already has an active team membership in that period.';
```

```
    END IF;
```

```
END//
```

-- 4.

```
CREATE TRIGGER check_capacity_before_update
BEFORE UPDATE ON Location
FOR EACH ROW
BEGIN
    DECLARE member_count INT;

    SELECT COUNT(*) INTO member_count
    FROM ClubMemberLocation
    WHERE LocationID = OLD.locationID
    AND (EndDate IS NULL OR EndDate > CURDATE());

    IF NEW.Capacity < member_count THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'New capacity cannot be less than the current number of
active members.';

        END IF;
    END//
```

DELIMITER ;

## Database Schema for Demonstration of DDL

-- 1. Person

```
CREATE TABLE Person(
    PersonID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
```

```
LastName VARCHAR(50),  
DateOfBirth DATE NOT NULL,  
SSN VARCHAR(50) UNIQUE NOT NULL,  
MedicareNo VARCHAR(20) UNIQUE,  
PhoneNo VARCHAR(20),  
Address VARCHAR(200),  
City VARCHAR(50),  
Province VARCHAR(50),  
PostalCode VARCHAR(10),  
Email VARCHAR(100)  
);
```

## -- 2.FamilyMember

```
CREATE TABLE FamilyMember(  
FamilyMemberID INT PRIMARY KEY,  
FOREIGN KEY (FamilyMemberID) REFERENCES Person(PersonID)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

## -- 3.Personnel

```
CREATE TABLE Personnel(  
PersonnelID INT PRIMARY KEY,  
Role ENUM('Administrator','Captain','Coach','Assistant Coach','Other') NOT NULL,
```

```
Mandate ENUM('Volunteer','Salaried') NOT NULL,  
FOREIGN KEY (PersonnelID) REFERENCES Person(PersonID)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
ALTER TABLE Personnel  
MODIFY Role ENUM('Administrator', 'Captain', 'Coach', 'Assistant Coach', 'Head  
Coach');
```

#### -- 4.Location

```
CREATE TABLE Location(  
locationID INT AUTO_INCREMENT PRIMARY KEY,  
name VARCHAR(100),  
Type ENUM('Head', 'Branch'),  
Address VARCHAR(200),  
City VARCHAR(50),  
Province VARCHAR(50),  
PostalCode VARCHAR(10),  
PhoneNo VARCHAR(20),  
WebAddress VARCHAR(100),  
Capacity INT  
);
```

#### -- 5.ClubMember

```
CREATE TABLE ClubMember(
```

```
ClubMemberID INT PRIMARY KEY,  
Height DECIMAL(5,2),  
Weight DECIMAL(5,2),  
isMinor BOOLEAN,  
locationID INT,  
PaymentStatus BOOLEAN DEFAULT FALSE, -- Created as false but set true  
after payment  
FOREIGN KEY (ClubMemberID) REFERENCES Person(PersonID)  
ON DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (locationID) REFERENCES Location(locationID)  
);
```

## -- 6.Team

```
CREATE TABLE Team(  
teamID INT AUTO_INCREMENT PRIMARY KEY,  
teamName VARCHAR(100) UNIQUE,  
team_gender ENUM('Male', 'Female'),  
LocationID INT,  
HeadCoachID INT,  
FOREIGN KEY (LocationID) REFERENCES Location(locationID),  
FOREIGN KEY (HeadCoachID) REFERENCES Personnel(PersonnelID)  
);
```

## -- 7.Hobby

```
CREATE TABLE Hobby(
```

```
hobbyID INT AUTO_INCREMENT PRIMARY KEY,  
name VARCHAR(50) NOT NULL  
);
```

## -- 8.Payment

```
CREATE TABLE Payment(  
    PaymentID INT AUTO_INCREMENT PRIMARY KEY,  
    ClubMemberID INT,  
    PaymentDate DATE,  
    Amount DECIMAL(10,2),  
    Method ENUM('Cash','Debit','Credit'),  
    MembershipYear INT,  
    InstallmentNumber INT CHECK (InstallmentNumber BETWEEN 1 AND 4),  
    FOREIGN KEY (ClubMemberID) REFERENCES ClubMember(ClubMemberID)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

## -- 9.Relationship

```
CREATE TABLE Relationship(  
    relationshipTypeID INT AUTO_INCREMENT PRIMARY KEY,  
    TypeName VARCHAR(50) NOT NULL  
);
```

## -- 10.Session

```
CREATE TABLE Session(  
    SessionID INT AUTO_INCREMENT PRIMARY KEY,  
    SessionDate DATE,  
    SessionTime TIME,  
    Address VARCHAR(200),  
    SessionType ENUM('Training', 'Game')  
);
```

#### -- 11. TeamFormation

```
CREATE TABLE TeamFormation(  
    FormationID INT AUTO_INCREMENT PRIMARY KEY,  
    SessionID INT,  
    Team1ID INT,  
    Team2ID INT,  
    Team1Score INT,  
    Team2Score INT,  
    FOREIGN KEY (SessionID) REFERENCES Session(SessionID),  
    FOREIGN KEY (Team1ID) REFERENCES Team(teamID),  
    FOREIGN KEY (Team2ID) REFERENCES Team(teamID)  
);
```

#### -- 12. PlayerPosition

```
CREATE TABLE PlayerPosition (  
    PositionID INT AUTO_INCREMENT PRIMARY KEY,
```

```
PositionName VARCHAR(50) NOT NULL UNIQUE  
);
```

#### -- 13.SecondaryFamilyMember

```
CREATE TABLE SecondaryFamilyMember(  
    SecondaryFamilyMemberID INT AUTO_INCREMENT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    PhoneNo VARCHAR(20),  
    PrimaryFamilyMemberID INT,  
    FOREIGN KEY (PrimaryFamilyMemberID) REFERENCES  
    FamilyMember(FamilyMemberID)  
);
```

```
ALTER TABLE SecondaryFamilyMember  
DROP FOREIGN KEY SecondaryFamilyMember_ibfk_1;
```

```
ALTER TABLE SecondaryFamilyMember  
ADD CONSTRAINT fk_primary_family_member  
FOREIGN KEY (PrimaryFamilyMemberID)  
REFERENCES FamilyMember(FamilyMemberID)  
ON DELETE CASCADE;
```

#### -- 14.Email Log Table

```
CREATE TABLE EmailLog(
```

```
LogID INT AUTO_INCREMENT PRIMARY KEY,  
EmailDate DATETIME,  
SenderLocation VARCHAR(100),  
ReceiverEmail VARCHAR(100),  
Subject VARCHAR(200),  
BodyPreview TEXT  
);
```

#### -- 15.ClubMemberLocation

```
CREATE TABLE ClubMemberLocation(  
ClubMemberID INT,  
LocationID INT,  
StartDate DATE,  
EndDate DATE,  
PRIMARY KEY (ClubMemberID, LocationID, StartDate),  
FOREIGN KEY (ClubMemberID) REFERENCES ClubMember(ClubMemberID),  
FOREIGN KEY (LocationID) REFERENCES Location(LocationID)  
);
```

#### -- 16.Personnel\_Location

```
CREATE TABLE Personnel_Location(  
PersonnelID INT,  
LocationID INT,  
StartDate DATE,
```

```
        EndDate DATE,  
        PRIMARY KEY (PersonnelID, LocationID, StartDate),  
        FOREIGN KEY (PersonnelID) REFERENCES Personnel(PersonnelID),  
        FOREIGN KEY (LocationID) REFERENCES Location(LocationID)  
    );
```

#### -- 17. FamilyMemberLocation

```
CREATE TABLE FamilyMemberLocation(  
    FamilyMemberID INT,  
    LocationID INT,  
    StartDate DATE,  
    EndDate DATE,  
    PRIMARY KEY (FamilyMemberID, LocationID, StartDate),  
    FOREIGN KEY (FamilyMemberID) REFERENCES  
    FamilyMember(FamilyMemberID),  
    FOREIGN KEY (LocationID) REFERENCES Location(LocationID)  
);
```

#### -- 18. ClubMember\_Hobby

```
CREATE TABLE ClubMember_Hobby (  
    ClubMemberID INT,  
    HobbyID INT,  
    PRIMARY KEY (ClubMemberID, HobbyID),  
    FOREIGN KEY (ClubMemberID) REFERENCES ClubMember(ClubMemberID),  
    FOREIGN KEY (HobbyID) REFERENCES Hobby(HobbyID)
```

);

**-- 19.memberTeam**

```
CREATE TABLE memberTeam(  
    ClubMemberID INT,  
    teamID INT,  
    startDate DATE,  
    endDate DATE,  
    PRIMARY KEY (ClubMemberID, teamID, startDate),  
    FOREIGN KEY (ClubMemberID) REFERENCES ClubMember(ClubMemberID),  
    FOREIGN KEY (teamID) REFERENCES Team(teamID)  
);
```

**-- 20.minorFamilyHistory**

```
CREATE TABLE minorFamilyHistory(  
    relationshipTypeID INT,  
    FamilyMemberID INT,  
    ClubMemberID INT,  
    startDate DATE,  
    endDate DATE,  
    PRIMARY KEY (FamilyMemberID, ClubMemberID, startDate),  
    FOREIGN KEY (relationshipTypeID) REFERENCES  
    Relationship(relationshipTypeID),  
    FOREIGN KEY (FamilyMemberID) REFERENCES  
    FamilyMember(FamilyMemberID),
```

```
FOREIGN KEY (ClubMemberID) REFERENCES ClubMember(ClubMemberID)
);

```

### -- 21.TeaмFormationPlayer

```
CREATE TABLE TeamFormationPlayer(
    FormationID INT,
    TeamID INT,
    ClubMemberID INT,
    PositionID INT,
    PRIMARY KEY (FormationID, TeamID, ClubMemberID),
    FOREIGN KEY (FormationID) REFERENCES TeamFormation(FormationID),
    FOREIGN KEY (TeamID) REFERENCES Team(teamID),
    FOREIGN KEY (ClubMemberID) REFERENCES ClubMember(ClubMemberID),
    FOREIGN KEY (PositionID) REFERENCES PlayerPosition(PositionID)
);

```

### -- 22.SecondaryFamilyRelationship

```
CREATE TABLE SecondaryFamilyRelationship(
    SecondaryFamilyMemberID INT,
    ClubMemberID INT,
    relationshipTypeID INT,

```

```
PRIMARY KEY (SecondaryFamilyMemberID, ClubMemberID),  
FOREIGN KEY (SecondaryFamilyMemberID) REFERENCES  
SecondaryFamilyMember(SecondaryFamilyMemberID),  
FOREIGN KEY (ClubMemberID) REFERENCES ClubMember(ClubMemberID),  
FOREIGN KEY (relationshipTypeID) REFERENCES  
Relationship(relationshipTypeID)  
);
```

## **DATA INSERTED IN THE DATABASE**

### **-- 1. Person Table**

```
INSERT INTO Person (FirstName, LastName, DateOfBirth, SSN, MedicareNo, PhoneNo,  
Address, City, Province, PostalCode, Email) VALUES
```

### **-- Personnel entries**

```
('John', 'Smith', '1975-03-15', '123-456-789', 'SMIT751501', '514-555-0001', '123 Main St',  
'Montreal', 'QC', 'H1A1A1', 'john.smith@mvc.com'),  
(('Sarah', 'Johnson', '1980-07-22', '234-567-890', 'JOHN802201', '514-555-0002', '456 Oak  
Ave', 'Montreal', 'QC', 'H2B2B2', 'sarah.johnson@mvc.com'),  
(('Mike', 'Williams', '1978-11-10', '345-678-901', 'WILL781001', '450-555-0003', '789 Pine St',  
'Laval', 'QC', 'H7C3C3', 'mike.williams@mvc.com'),  
(('Lisa', 'Brown', '1982-05-18', '456-789-012', 'BROW820501', '514-555-0004', '321 Elm St',  
'Montreal', 'QC', 'H3D4D4', 'lisa.brown@mvc.com'),  
(('David', 'Davis', '1985-09-25', '567-890-123', 'DAVI850901', '450-555-0005', '654 Maple Ave',  
'Laval', 'QC', 'H7E5E5', 'david.davis@mvc.com'),  
(('Emma', 'Wilson', '1979-12-03', '678-901-234', 'WILS791201', '514-555-0006', '987 Cedar  
St', 'Montreal', 'QC', 'H4F6F6', 'emma.wilson@mvc.com'),  
(('James', 'Moore', '1983-04-14', '789-012-345', 'MOOR830401', '514-555-0007', '147 Birch  
Ave', 'Montreal', 'QC', 'H5G7G7', 'james.moore@mvc.com'),
```

('Anna', 'Taylor', '1981-08-07', '890-123-456', 'TAYL810801', '450-555-0008', '258 Spruce St', 'Laval', 'QC', 'H7H8H8', 'anna.taylor@mvc.com'),

## -- Family Members

('Robert', 'Anderson', '1970-01-20', '901-234-567', 'ANDE701201', '514-555-0009', '369 Willow Ave', 'Montreal', 'QC', 'H1I9I9', 'robert.anderson@email.com'),

('Mary', 'Anderson', '1972-06-12', '012-345-678', 'ANDE720601', '514-555-0010', '369 Willow Ave', 'Montreal', 'QC', 'H1I9I9', 'mary.anderson@email.com'),

('Thomas', 'Miller', '1975-02-28', '123-456-780', 'MILL750201', '450-555-0011', '741 Poplar St', 'Laval', 'QC', 'H7J0J0', 'thomas.miller@email.com'),

('Jennifer', 'Miller', '1977-10-15', '234-567-891', 'MILL771001', '450-555-0012', '741 Poplar St', 'Laval', 'QC', 'H7J0J0', 'jennifer.miller@email.com'),

('William', 'Garcia', '1968-12-05', '345-678-902', 'GARC681201', '514-555-0013', '852 Hickory Ave', 'Montreal', 'QC', 'H2K1K1', 'william.garcia@email.com'),

('Patricia', 'Garcia', '1970-04-18', '456-789-013', 'GARC700401', '514-555-0014', '852 Hickory Ave', 'Montreal', 'QC', 'H2K1K1', 'patricia.garcia@email.com'),

## -- Club Members (Major - 18+)

('Alex', 'Anderson', '2005-03-10', '567-890-124', 'ANDE050301', '514-555-0015', '369 Willow Ave', 'Montreal', 'QC', 'H1I9I9', 'alex.anderson@email.com'),

('Sophie', 'Miller', '2006-07-20', '678-901-235', 'MILL060701', '450-555-0016', '741 Poplar St', 'Laval', 'QC', 'H7J0J0', 'sophie.miller@email.com'),

('Lucas', 'Garcia', '2004-11-08', '789-012-346', 'GARC041101', '514-555-0017', '852 Hickory Ave', 'Montreal', 'QC', 'H2K1K1', 'lucas.garcia@email.com'),

('Emma', 'Thompson', '2003-05-25', '890-123-457', 'THOM030501', '514-555-0018', '963 Chestnut St', 'Montreal', 'QC', 'H3L2L2', 'emma.thompson@email.com'),

('Noah', 'White', '2002-09-12', '901-234-568', 'WHIT020901', '450-555-0019', '159 Walnut Ave', 'Laval', 'QC', 'H7M3M3', 'noah.white@email.com'),

('Olivia', 'Martin', '2005-01-30', '012-345-679', 'MART050101', '514-555-0020', '357 Ash St', 'Montreal', 'QC', 'H4N4N4', 'olivia.martin@email.com'),

## -- Club Members (Minor - under 18)

('Ethan', 'Anderson', '2010-08-15', '123-456-781', 'ANDE100801', '514-555-0021', '369 Willow Ave', 'Montreal', 'QC', 'H1I9I9', 'ethan.anderson@email.com'),  
(('Ava', 'Miller', '2011-12-03', '234-567-892', 'MILL111201', '450-555-0022', '741 Poplar St', 'Laval', 'QC', 'H7J0J0', 'ava.miller@email.com'),  
(('Mason', 'Garcia', '2012-04-22', '345-678-903', 'GARC120401', '514-555-0023', '852 Hickory Ave', 'Montreal', 'QC', 'H2K1K1', 'mason.garcia@email.com'),  
(('Isabella', 'Johnson', '2013-06-10', '456-789-014', 'JOHN130601', '514-555-0024', '468 Beech Ave', 'Montreal', 'QC', 'H5O5O5', 'isabella.johnson@email.com'),  
(('Logan', 'Brown', '2009-10-18', '567-890-125', 'BROW091001', '450-555-0025', '579 Sycamore St', 'Laval', 'QC', 'H7P6P6', 'logan.brown@email.com'),  
(('Mia', 'Davis', '2014-02-14', '678-901-236', 'DAVI140201', '514-555-0026', '680 Magnolia Ave', 'Montreal', 'QC', 'H6Q7Q7', 'mia.davis@email.com');

INSERT INTO Person (

PersonID, FirstName, LastName, DateOfBirth, PhoneNo, Email, Address, City, Province, PostalCode, SSN

) VALUES (

101, 'Trigger', 'Tester', '2005-01-01', '514-000-0000', 'trigger@test.com', '123 Code St',

'Montreal', 'QC', 'H1A1A1', 'SSN101'

);

INSERT INTO ClubMember (ClubMemberID, PaymentStatus, isMinor)  
VALUES (101, TRUE, FALSE);

## -- 2. FamilyMember Table

INSERT INTO FamilyMember (FamilyMemberID) VALUES

(9), -- Robert Anderson

(10), -- Mary Anderson

(11), -- Thomas Miller

(12), -- Jennifer Miller

(13), -- William Garcia

(14); -- Patricia Garcia

### -- 3. Personnel Table

INSERT INTO Personnel (PersonnelID, Role, Mandate) VALUES

(1, 'Administrator', 'Salaried'), -- John Smith (General Manager)

(2, 'Administrator', 'Salaried'), -- Sarah Johnson (Deputy Manager)

(3, 'Coach', 'Salaried'), -- Mike Williams

(4, 'Coach', 'Volunteer'), -- Lisa Brown

(5, 'Assistant Coach', 'Volunteer'), -- David Davis

(6, 'Captain', 'Volunteer'), -- Emma Wilson

(7, 'Coach', 'Salaried'), -- James Moore

(8, 'Assistant Coach', 'Volunteer'), -- Anna Taylor

(9, 'Head Coach', 'Salaried'); -- or 'Volunteer' if applicable

### -- 4. Location Table (Note: This should come before ClubMember due to foreign key dependency)

INSERT INTO Location (name, Type, Address, City, Province, PostalCode, PhoneNo, WebAddress, Capacity) VALUES

('Montreal Volleyball Club - Head Office', 'Head', '1000 Sherbrooke St W', 'Montreal', 'QC', 'H3A1G1', '514-555-1000', 'www.mvc-montreal.com', 150),

('Laval Branch', 'Branch', '2000 Boulevard Chomedey', 'Laval', 'QC', 'H7T2Z5',  
'450-555-2000', 'www.mvc-laval.com', 100),

('West Island Branch', 'Branch', '3000 Brunswick Blvd', 'Dollard-des-Ormeaux', 'QC',  
'H9B1N7', '514-555-3000', 'www.mvc-westisland.com', 80);

## -- 5. ClubMember Table

INSERT INTO ClubMember (ClubMemberID, Height, Weight, isMinor, locationID, PaymentStatus) VALUES

-- Major Members (18+)

(15, 175.5, 70.2, FALSE, 1, TRUE), -- Alex Anderson

(16, 168.3, 62.8, FALSE, 2, TRUE), -- Sophie Miller

(17, 180.1, 75.5, FALSE, 1, FALSE), -- Lucas Garcia

(18, 172.0, 65.0, FALSE, 1, TRUE), -- Emma Thompson

(19, 185.2, 80.1, FALSE, 2, TRUE), -- Noah White

(20, 165.8, 58.5, FALSE, 1, FALSE), -- Olivia Martin

## -- Minor Members (under 18)

(21, 160.0, 50.2, TRUE, 1, TRUE), -- Ethan Anderson

(22, 155.5, 48.0, TRUE, 2, TRUE), -- Ava Miller

(23, 158.2, 45.5, TRUE, 1, FALSE), -- Mason Garcia

(24, 152.0, 42.8, TRUE, 1, TRUE), -- Isabella Johnson

(25, 165.5, 55.0, TRUE, 2, TRUE), -- Logan Brown

(26, 148.5, 40.2, TRUE, 1, FALSE); -- Mia Davis

INSERT INTO Person (PersonID, FirstName, LastName, DateOfBirth, PhoneNo, Email, Address, City, Province, PostalCode, SSN, MedicareNo)

```
VALUES (
    9999,
    'Test',
    'Player',
    '2000-01-01',
    '514-123-4567',
    'test@example.com',
    '123 Test Ave',
    'Montreal',
    'QC',
    'H1X1X1',
    '999-99-9999',      -- Sample SSN
    '1234567890'        -- Sample Medicare number
);
```

```
INSERT INTO ClubMember (ClubMemberID, locationID, PaymentStatus, isMinor)
VALUES (9999, 1, TRUE, FALSE);
```

```
INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)
VALUES (9999, 1, '2024-01-01', NULL);
```

```
INSERT INTO Session (SessionID, SessionDate, SessionTime, Address, SessionType)
VALUES
```

```
(9001, '2025-03-01', '10:00:00', 'Gym A', 'Game'),
```

```
(9002, '2025-03-05', '11:00:00', 'Gym B', 'Game'),
```

```
(9003, '2025-03-10', '12:00:00', 'Gym C', 'Game'),
```

```
(9004, '2025-03-15', '13:00:00', 'Gym D', 'Game');
```

```
INSERT INTO Person (
```

```
PersonID, FirstName, LastName, DateOfBirth, PhoneNo, Email, Address, City, Province,
PostalCode, SSN

)

VALUES (

1001, 'Tommy', 'Nguyen', '2012-09-15', '514-555-1010', 'tommy@example.com',
'1234 Elm St', 'Montreal', 'QC', 'H3Z2Y7', 'SSN1001'

);
```

```
INSERT INTO ClubMember (ClubMemberID, PaymentStatus, isMinor)
VALUES (1001, TRUE, TRUE);
```

```
INSERT INTO FamilyMember (FamilyMemberID)
VALUES (4);
```

## -- 6. Team Table

```
INSERT INTO Team (teamName, team_gender, LocationID, HeadCoachID) VALUES
('Montreal Eagles Male', 'Male', 1, 3), -- Mike Williams as head coach
('Montreal Eagles Female', 'Female', 1, 4), -- Lisa Brown as head coach
('Laval Tigers Male', 'Male', 2, 7), -- James Moore as head coach
('Laval Tigers Female', 'Female', 2, 8), -- Anna Taylor as head coach
('West Island Sharks Male', 'Male', 3, 3), -- Mike Williams (can coach multiple teams)
('Montreal Juniors Male', 'Male', 1, 5); -- David Davis as head coach
```

```
INSERT INTO Team (TeamID, teamName, team_gender, LocationID, HeadCoachID)
VALUES
(301, 'Tigers', 'Male', 1, 9),
(302, 'Panthers', 'Female', 1, 9);
```

```
INSERT INTO Team (teamID, teamName, team_gender, LocationID, HeadCoachID)
VALUES (303, 'Wolves', 'Male', 1, 9);
```

### -- 7. Hobby Table

```
INSERT INTO Hobby (name) VALUES
('Volleyball'),
('Soccer'),
('Tennis'),
('Ping Pong'),
('Swimming'),
('Hockey'),
('Golf'),
('Basketball'),
('Baseball'),
('Cycling');
```

### -- 8. Payment Table

```
INSERT INTO Payment (ClubMemberID, PaymentDate, Amount, Method, MembershipYear,
InstallmentNumber) VALUES
```

```
-- Major members (200$ annual fee)

(15, '2024-01-15', 200.00, 'Credit', 2024, 1),
(16, '2024-02-01', 100.00, 'Debit', 2024, 1),
(16, '2024-08-01', 100.00, 'Debit', 2024, 2),
(18, '2024-01-10', 200.00, 'Cash', 2024, 1),
(19, '2024-03-15', 50.00, 'Credit', 2024, 1),
(19, '2024-06-15', 50.00, 'Credit', 2024, 2),
(19, '2024-09-15', 100.00, 'Credit', 2024, 3),
```

**-- Minor members (100\$ annual fee)**

```
(21, '2024-01-20', 100.00, 'Debit', 2024, 1),  
(22, '2024-02-15', 50.00, 'Cash', 2024, 1),  
(22, '2024-07-15', 50.00, 'Cash', 2024, 2),  
(24, '2024-01-25', 100.00, 'Credit', 2024, 1),  
(25, '2024-03-01', 25.00, 'Debit', 2024, 1),  
(25, '2024-06-01', 25.00, 'Debit', 2024, 2),  
(25, '2024-09-01', 25.00, 'Debit', 2024, 3),  
(25, '2024-12-01', 25.00, 'Debit', 2024, 4),
```

**-- Some donations (excess payments)**

```
(15, '2024-06-15', 50.00, 'Credit', 2024, 2),  
(21, '2024-08-10', 30.00, 'Cash', 2024, 2);
```

**-- 9. Relationship Table**

```
INSERT INTO Relationship (TypeName) VALUES  
    ('Father'),  
    ('Mother'),  
    ('Grandfather'),  
    ('Grandmother'),  
    ('Tutor'),  
    ('Partner'),  
    ('Friend'),  
    ('Other');
```

**-- 10. Session Table**

```
INSERT INTO Session (SessionDate, SessionTime, Address, SessionType) VALUES  
('2025-01-13', '18:00:00', '1000 Sherbrooke St W, Montreal', 'Training'),  
('2025-01-14', '19:30:00', '2000 Boulevard Chomedey, Laval', 'Training'),  
('2025-01-15', '20:00:00', '1000 Sherbrooke St W, Montreal', 'Game'),  
('2025-01-16', '18:30:00', '3000 Brunswick Blvd, DDO', 'Training'),  
('2025-01-17', '19:00:00', '2000 Boulevard Chomedey, Laval', 'Game'),  
('2025-01-20', '18:00:00', '1000 Sherbrooke St W, Montreal', 'Training'),  
('2025-01-21', '19:30:00', '2000 Boulevard Chomedey, Laval', 'Training'),  
('2025-01-22', '20:00:00', '1000 Sherbrooke St W, Montreal', 'Game');  
  
INSERT INTO Session (SessionDate, SessionTime, Address, SessionType)  
VALUES ('2025-08-15', '19:00:00', '4000 Rosemont Blvd, Montreal', 'Training');  
  
INSERT INTO Session (SessionID, SessionDate, SessionTime, Address, SessionType)  
VALUES (2001, '2025-08-10', '14:00:00', '123 Main St', 'Training');  
  
INSERT INTO Session (SessionID, SessionDate, SessionTime, Address, SessionType)  
VALUES (1010, '2025-02-15', '19:00:00', '1000 Sherbrooke St W, Montreal', 'Game');  
  
INSERT INTO Session (SessionID, SessionDate, SessionTime, Address, SessionType)  
VALUES (1011, '2025-03-10', '20:00:00', '1000 Sherbrooke St W, Montreal', 'Game');  
  
INSERT INTO Session (SessionID, SessionDate, SessionTime, SessionType)  
VALUES (3001, '2025-08-08', '11:30:00', 'Game');
```

```
INSERT INTO Session (SessionID, SessionDate, SessionTime, SessionType)  
VALUES (3002, '2025-08-08', '12:45:00', 'Game')
```

## -- 11. TeamFormation Table

```
INSERT INTO TeamFormation  
(FormationID, SessionID, Team1ID, Team2ID, Team1Score, Team2Score) VALUES
```

```
(1, 9001, 1, 6, NULL, NULL),  
(2, 9002, 3, 4, NULL, NULL),  
(3, 9003, 1, 2, 25, 23),  
(4, 9004, 5, 1, NULL, NULL),  
(5, 9005, 3, 4, 21, 25),  
(6, 9006, 2, 1, NULL, NULL),  
(7, 9007, 4, 3, NULL, NULL),  
(8, 9008, 2, 4, 25, 18);
```

```
INSERT INTO TeamFormation (FormationID, SessionID, Team1ID, Team2ID, Team1Score,  
Team2Score)
```

```
VALUES (2002, 3001, 1, 2, NULL, NULL);
```

```
INSERT INTO TeamFormation (FormationID, SessionID, Team1ID, Team2ID)
```

```
VALUES (2003, 3002, 1, 2);
```

```
INSERT INTO TeamFormation (FormationID, SessionID, Team1ID, Team2ID)
```

```
VALUES (2004, 3002, 1, 2);
```

```
INSERT INTO TeamFormation (FormationID, SessionID, Team1ID, Team2ID)
```

```
VALUES
```

```
(8001, 9001, 1, 2),
```

```
(8002, 9002, 1, 2),
```

```
(8003, 9003, 1, 2),
```

```
(8004, 9004, 1, 2);
```

### -- Insert default player positions

```
INSERT INTO PlayerPosition (PositionName) VALUES
```

```
('Setter'),
```

```
('Libero'),
```

('Outside Hitter'),  
('Opposite Hitter'),  
('Middle Blocker'),  
('Defensive Specialist');

### -- 13. SecondaryFamilyMember Table

INSERT INTO SecondaryFamilyMember (FirstName, LastName, PhoneNo, PrimaryFamilyMemberID) VALUES  
('Margaret', 'Anderson', '514-555-0101', 9), -- Secondary for Robert Anderson  
('Helen', 'Miller', '450-555-0102', 11), -- Secondary for Thomas Miller  
('Carlos', 'Garcia', '514-555-0103', 13), -- Secondary for William Garcia  
('Linda', 'Johnson', '514-555-0104', 9), -- Another secondary for Robert Anderson  
('Paul', 'Brown', '450-555-0105', 12); -- Secondary for Jennifer Miller

### -- 14. EmailLog Table

INSERT INTO EmailLog (EmailDate, SenderLocation, ReceiverEmail, Subject, BodyPreview) VALUES -- review and possibly change BodyPreview  
('2025-01-12 09:00:00', 'Montreal Volleyball Club - Head Office', 'alex.anderson@email.com', 'Montreal Eagles Male Monday 13-Jan-2025 6:00 PM training session', 'Dear Alex Anderson, You are scheduled to play as Outside Hitter in the upcoming training session...'),  
('2025-01-12 09:05:00', 'Montreal Volleyball Club - Head Office', 'lucas.garcia@email.com', 'Montreal Eagles Male Monday 13-Jan-2025 6:00 PM training session', 'Dear Lucas Garcia, You are scheduled to play as Setter in the upcoming training session...'),  
('2025-01-13 09:00:00', 'Laval Branch', 'sophie.miller@email.com', 'Laval Tigers Female Tuesday 14-Jan-2025 7:30 PM training session', 'Dear Sophie Miller, You are scheduled to play as Libero in the upcoming training session...'),  
('2025-01-14 09:00:00', 'Montreal Volleyball Club - Head Office', 'emma.thompson@email.com', 'Montreal Eagles Female Wednesday 15-Jan-2025 8:00 PM

game session', 'Dear Emma Thompson, You are scheduled to play as Middle Blocker in the upcoming game session...');

## -- 15. ClubMemberLocation Table

```
INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)  
VALUES
```

```
(15, 1, '2024-01-01', NULL), -- Alex Anderson currently at Montreal
```

```
(16, 2, '2024-01-01', NULL), -- Sophie Miller currently at Laval
```

```
(17, 1, '2024-01-01', NULL), -- Lucas Garcia currently at Montreal
```

```
(18, 1, '2024-01-01', NULL), -- Emma Thompson currently at Montreal
```

```
(19, 2, '2024-01-01', NULL), -- Noah White currently at Laval
```

```
(20, 1, '2024-01-01', '2024-06-30'), -- Olivia Martin moved
```

```
(20, 3, '2024-07-01', NULL), -- Olivia Martin moved to West Island
```

```
(21, 1, '2024-01-01', NULL), -- Ethan Anderson currently at Montreal
```

```
(22, 2, '2024-01-01', NULL), -- Ava Miller currently at Laval
```

```
(23, 1, '2024-01-01', NULL), -- Mason Garcia currently at Montreal
```

```
(24, 1, '2024-01-01', NULL), -- Isabella Johnson currently at Montreal
```

```
(25, 2, '2024-01-01', NULL), -- Logan Brown currently at Laval
```

```
(26, 1, '2024-01-01', NULL); -- Mia Davis currently at Montreal
```

```
UPDATE ClubMemberLocation
```

```
SET StartDate = '2022-07-01'
```

```
WHERE ClubMemberID = 20 AND LocationID = 1;
```

## -- 16. Personnel\_Location Table

```
INSERT INTO Personnel_Location (PersonnelID, LocationID, StartDate, EndDate) VALUES  
(1, 1, '2022-01-01', NULL), -- John Smith at Head Office  
(2, 1, '2022-06-01', NULL), -- Sarah Johnson at Head Office  
(3, 1, '2022-01-15', '2023-12-31'), -- Mike Williams at Montreal first  
(3, 2, '2024-01-01', '2024-06-30'), -- Then moved to Laval  
(3, 1, '2024-07-01', NULL), -- Then back to Montreal  
(4, 1, '2023-03-01', NULL), -- Lisa Brown at Montreal  
(5, 2, '2023-01-01', NULL), -- David Davis at Laval  
(6, 1, '2023-02-15', NULL), -- Emma Wilson at Montreal  
(7, 2, '2022-08-01', NULL), -- James Moore at Laval  
(8, 2, '2023-05-01', NULL), -- Anna Taylor at Laval  
(9, 1, '2024-01-01', NULL);
```

## -- 17. FamilyMemberLocation Table

```
INSERT INTO FamilyMemberLocation (FamilyMemberID, LocationID, StartDate, EndDate)  
VALUES  
(9, 1, '2024-01-01', NULL), -- Robert Anderson at Montreal  
(10, 1, '2024-01-01', NULL), -- Mary Anderson at Montreal  
(11, 2, '2024-01-01', NULL), -- Thomas Miller at Laval  
(12, 2, '2024-01-01', NULL), -- Jennifer Miller at Laval  
(13, 1, '2024-01-01', NULL), -- William Garcia at Montreal  
(14, 1, '2024-01-01', NULL); -- Patricia Garcia at Montreal
```

## -- 18.Inserting data into ClubMember\_Hobby

```
INSERT INTO ClubMember_Hobby (ClubMemberID, HobbyID) VALUES
```

```
(15, 1), (15, 2), (15, 5), -- Alex: Volleyball, Soccer, Swimming  
(16, 1), (16, 3), -- Sophie: Volleyball, Tennis  
(17, 1), (17, 8), (17, 9), -- Lucas: Volleyball, Basketball, Baseball  
(18, 1), (18, 4), (18, 6), -- Emma: Volleyball, Ping Pong, Hockey  
(19, 1), (19, 7), -- Noah: Volleyball, Golf  
(20, 1), (20, 3), (20, 10), -- Olivia: Volleyball, Tennis, Cycling  
(21, 1), (21, 2), -- Ethan: Volleyball, Soccer  
(22, 1), (22, 5), -- Ava: Volleyball, Swimming  
(23, 1), (23, 4), -- Mason: Volleyball, Ping Pong  
(24, 1), (24, 8), -- Isabella: Volleyball, Basketball  
(25, 1), (25, 6), (25, 9), -- Logan: Volleyball, Hockey, Baseball  
(26, 1), (26, 3); -- Mia: Volleyball, Tennis
```

## -- 19. memberTeam Table

```
INSERT INTO memberTeam (ClubMemberID, teamID, startDate, endDate) VALUES  
(15, 1, '2024-01-01', NULL), -- Alex in Montreal Eagles Male  
(17, 1, '2024-01-01', NULL), -- Lucas in Montreal Eagles Male  
(19, 3, '2024-01-01', NULL), -- Noah in Laval Tigers Male  
(25, 3, '2024-01-01', NULL), -- Logan in Laval Tigers Male  
(18, 2, '2024-01-01', NULL), -- Emma in Montreal Eagles Female  
(20, 2, '2024-01-01', '2024-06-30'), -- Olivia moved teams  
(16, 4, '2024-01-01', NULL), -- Sophie in Laval Tigers Female  
(22, 4, '2024-01-01', NULL), -- Ava in Laval Tigers Female  
(21, 6, '2024-01-01', NULL), -- Ethan in Montreal Juniors Male  
(23, 6, '2024-01-01', NULL); -- Mason in Montreal Juniors Male
```

-- 20. minorFamilyHistory Table

```
INSERT INTO minorFamilyHistory (relationshipTypeID, FamilyMemberID, ClubMemberID, startDate, endDate) VALUES  
(1, 9, 21, '2024-01-01', NULL), -- Robert Anderson (Father) -> Ethan Anderson  
(2, 10, 21, '2024-01-01', NULL), -- Mary Anderson (Mother) -> Ethan Anderson  
(1, 11, 22, '2024-01-01', NULL), -- Thomas Miller (Father) -> Ava Miller  
(2, 12, 22, '2024-01-01', NULL), -- Jennifer Miller (Mother) -> Ava Miller  
(1, 13, 23, '2024-01-01', NULL), -- William Garcia (Father) -> Mason Garcia  
(2, 14, 23, '2024-01-01', NULL), -- Patricia Garcia (Mother) -> Mason Garcia  
(1, 9, 24, '2024-01-01', NULL), -- Robert Anderson (Father) -> Isabella Johnson (guardian)  
(1, 11, 25, '2024-01-01', NULL), -- Thomas Miller (Father) -> Logan Brown  
(1, 13, 26, '2024-01-01', NULL); -- William Garcia (Father) -> Mia Davis
```

```
INSERT INTO minorFamilyHistory (
```

```
    relationshipTypeID,
```

```
    FamilyMemberID,
```

```
    ClubMemberID,
```

```
    startDate
```

```
)
```

```
VALUES (
```

```
    1, -- Father
```

```
    4, -- Lisa Brown (FamilyMember)
```

```
    1001, -- Existing ClubMember
```

```
    '2025-08-08' -- Start date
```

```
);
```

## -- 21. TeamFormationPlayer Table

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
VALUES
```

### -- Formation 1: Montreal Eagles Male vs Montreal Juniors Male (Training)

(1, 1, 15, 3), -- Alex Anderson as Outside Hitter

(1, 1, 17, 1), -- Lucas Garcia as Setter

(1, 6, 21, 2), -- Ethan Anderson as Libero

(1, 6, 23, 5), -- Mason Garcia as Middle Blocker

### -- Formation 2: Laval Tigers Male vs Laval Tigers Female (Training)

(2, 3, 19, 4), -- Noah White as Opposite Hitter

(2, 3, 25, 1), -- Logan Brown as Setter

(2, 4, 16, 2), -- Sophie Miller as Libero

(2, 4, 22, 3), -- Ava Miller as Outside Hitter

### -- Formation 3: Montreal Eagles Male vs Montreal Eagles Female (Game)

(3, 1, 15, 3), -- Alex Anderson as Outside Hitter

(3, 1, 17, 1), -- Lucas Garcia as Setter

(3, 2, 18, 5), -- Emma Thompson as Middle Blocker

### -- Formation 8: Montreal Eagles Female vs Laval Tigers Female (Game)

(8, 2, 18, 5), -- Emma Thompson as Middle Blocker

(8, 4, 16, 2), -- Sophie Miller as Libero

(8, 4, 22, 3); -- Ava Miller as Outside Hitter

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
VALUES (2002, 1, 101, 1);
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
VALUES (8001, 1, 9999, (SELECT PositionID FROM PlayerPosition WHERE PositionName
= 'Setter' LIMIT 1));
```

### -- Libero

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
VALUES (8002, 1, 9999, (SELECT PositionID FROM PlayerPosition WHERE PositionName
= 'Libero' LIMIT 1));
```

### -- Outside Hitter

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
VALUES (8003, 1, 9999, (SELECT PositionID FROM PlayerPosition WHERE PositionName
= 'Outside Hitter' LIMIT 1));
```

### -- Opposite Hitter

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
VALUES (8004, 1, 9999, (SELECT PositionID FROM PlayerPosition WHERE PositionName
= 'Opposite Hitter' LIMIT 1));
```

## -- 22. SecondaryFamilyRelationship Table

```
INSERT INTO SecondaryFamilyRelationship (SecondaryFamilyMemberID, ClubMemberID,
relationshipTypeID) VALUES
```

```
(1, 21, 3), -- Margaret Anderson (Grandmother) -> Ethan Anderson
```

```
(2, 22, 3), -- Helen Miller (Grandmother) -> Ava Miller
```

(3, 23, 5), -- Carlos Garcia (Tutor) -> Mason Garcia

(4, 24, 8), -- Linda Johnson (Other) -> Isabella Johnson

(5, 25, 8); -- Paul Brown (Other) -> Logan Brown

## **QUERIES**

### **QUERY 1**

-- Create Location

INSERT INTO Location

(name,Type,Address,City,Province,PostalCode,PhoneNo,WebAddress,Capacity)

VALUES

('South Shore Complex', 'Branch', '5000 Taschereau Blvd', 'Brossard', 'QC', 'J4Y1A2', '450-555-6000', 'www.mvc-southshore.com', 90),

('South Shore Complex', 'Branch', '5001 Taschereau Blvd', 'Brossard', 'QC', 'J4Z1B3', '450-555-6001', 'www.mvc-southshore.com/a', 90),

('South Shore Complex', 'Branch', '1200 Rue Saint-Charles', 'Longueuil', 'QC', 'J4K2T1', '450-555-6002', 'www.mvc-southshore.com/b', 90),

('South Shore Complex', 'Branch', '2100 Rue Victoria', 'Saint-Lambert', 'QC', 'J4P2K1', '450-555-6003', 'www.mvc-southshore.com/c', 90),

('South Shore Complex', 'Branch', '800 Rue de Montarville', 'Boucherville', 'QC', 'J4B1C3', '450-555-6004', 'www.mvc-southshore.com/d', 90);

-- Display Location

SELECT \* FROM Location;

SELECT \* FROM Location

WHERE name = 'South Shore Complex';

-- Edit Location

UPDATE Location

SET

Capacity = 100,

PhoneNo = '450-555-0000',

WebAddress = 'www.mvc-updatedsouthshore.com'

WHERE locationID = 4;

-- Delete Location

DELETE FROM Location

WHERE locationID = 4;

## RESULT

locationID	name	Type	Address	City	Province	PostalCode	PhoneNo	WebAddress	Capacity
4	South Shore Complex	Branch	5000 Taschereau Blvd	Brossard	QC	J4Y1A2	450-555-6000	www.mvc-southshore.com	90
5	South Shore Complex	Branch	5001 Taschereau Blvd	Brossard	QC	J4Z1B3	450-555-6001	www.mvc-southshore.com/a	90
6	South Shore Complex	Branch	1200 Rue Saint-Charles	Longueuil	QC	J4K2T1	450-555-6002	www.mvc-southshore.com/b	90
7	South Shore Complex	Branch	2100 Rue Victoria	Saint-Lambert	QC	J4P2K1	450-555-6003	www.mvc-southshore.com/c	90
8	South Shore Complex	Branch	800 Rue de Montarville	Boucherville	QC	J4B1C3	450-555-6004	www.mvc-southshore.com/d	90
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- Create

-- Step 1: Insert into Person table

INSERT INTO Person (PersonID, FirstName, LastName, DateOfBirth, SSN, MedicareNo, PhoneNo, Address, City, Province, PostalCode, Email) VALUES

(10000, 'Kevin', 'Hall', '1988-06-12', '999-888-770', 'HALL880612A', '514-555-7777', '123 Sunshine Rd', 'Montreal', 'QC', 'H1X2Y3', 'kevin.hall@mvc.com'),

(10001, 'Linda', 'Perez', '1990-03-04', '555-111-222', 'PERE900304', '438-555-7001', '44 Laurier Ave', 'Montreal', 'QC', 'H2X1B1', 'linda.perez@mvc.com'),

(10002, 'Omar', 'Khan', '1985-11-19', '666-222-333', 'KHAN851119', '514-555-7002', '880 St Laurent Blvd', 'Montreal', 'QC', 'H3C1C1', 'omar.khan@mvc.com'),

(10003, 'Julia', 'Chen', '1992-02-10', '777-333-444', 'CHEN920210', '514-555-7003', '1200 Peel St', 'Montreal', 'QC', 'H3B2T3', 'julia.chen@mvc.com'),

```
(10004, 'Marc', 'Leblanc', '1987-09-01', '888-444-555', 'LEBL870901', '514-555-7004', '2500  
Papineau Ave', 'Montreal', 'QC', 'H2K3P4', 'marc.leblanc@mvc.com');
```

```
select * from PERSON;
```

```
-- Step 2: Insert into Personnel table using the generated PersonID
```

```
INSERT INTO Personnel (PersonnelID, Role, Mandate) VALUES
```

```
(10000, 'Coach', 'Volunteer'),
```

```
(10001, 'Assistant Coach', 'Salaried'),
```

```
(10002, 'Coach', 'Salaried'),
```

```
(10003, 'Administrator', 'Salaried'),
```

```
(10004, 'Captain', 'Volunteer');
```

```
-- Display
```

```
-- All personnel with personal info
```

```
SELECT p.PersonID, p.FirstName, p.LastName, p.Email, per.Role, per.Mandate
```

```
FROM Person p
```

```
JOIN Personnel per ON p.PersonID = per.PersonnelID;
```

```
-- Edit
```

```
-- Change role or mandate of a personnel
```

```
UPDATE Personnel
```

```
SET Role = 'Assistant Coach',
```

```
Mandate = 'Salaried'
```

```
WHERE PersonnelID = 10000;
```

```
-- Update email or address in Person table
```

UPDATE Person

SET Email = 'k.hall@mvc.com'

WHERE PersonID = 10000;

-- Delete

-- Delete from Personnel table (first)

DELETE FROM Personnel

WHERE PersonnelID = 10000;

-- Then optionally delete from Person table

DELETE FROM Person

WHERE PersonID = 10000;

## **RESULT**

Result Grid					
Filter Rows: <input type="text"/> Export:  Wrap Cell Content: <input type="checkbox"/>					
PersonID	FirstName	LastName	Email	Role	Mandate
1	John	Smith	john.smith@mvc.com	Administrator	Salaried
2	Sarah	Johnson	sarah.johnson@mvc.com	Administrator	Salaried
3	Mike	Williams	mike.williams@mvc.com	Coach	Salaried
4	Lisa	Brown	lisa.brown@mvc.com	Coach	Volunteer
5	David	Davis	david.davis@mvc.com	Assistant Coach	Volunteer
6	Emma	Wilson	emma.wilson@mvc.com	Captain	Volunteer
7	James	Moore	james.moore@mvc.com	Coach	Salaried
8	Anna	Taylor	anna.taylor@mvc.com	Assistant Coach	Volunteer
9	Robert	Anderson	robert.anderson@email.com	Head Coach	Salaried
10000	Kevin	Hall	kevin.hall@mvc.com	Coach	Volunteer
10001	Linda	Perez	linda.perez@mvc.com	Assistant Coach	Salaried
10002	Omar	Khan	omar.khan@mvc.com	Coach	Salaried
10003	Julia	Chen	julia.chen@mvc.com	Administrator	Salaried
10004	Marc	Leblanc	marc.leblanc@mvc.com	Captain	Volunteer

PhoneNo, Address, City, Province, PostalCode, Email)

VALUES (10020, 'Carol', 'Foster', '1975-09-15', '888-777-668', 'FOST750915B', '514-555-8888', '99 Elm St', 'Montreal', 'QC', 'H2X1Y4', 'carol.foster@mvc.com');

INSERT INTO FamilyMember (FamilyMemberID) VALUES (10020);

-- READ / DISPLAY

-- Display primary family members with contact info

SELECT f.FamilyMemberID, p.FirstName, p.LastName, p.PhoneNo, p.Email

FROM FamilyMember f

```
JOIN Person p ON f.FamilyMemberID = p.PersonID;

-- Display secondary family members

SELECT s.SecondaryFamilyMemberID, s.FirstName, s.LastName, s.PhoneNo,
s.PrimaryFamilyMemberID

FROM SecondaryFamilyMember s;
```

```
-- EDIT
```

```
-- Update primary family member info
```

```
UPDATE Person

SET PhoneNo = '514-555-9999',
Email = 'carol.foster@updated.com'

WHERE PersonID = 10020;
```

```
-- Update secondary family member phone
```

```
UPDATE SecondaryFamilyMember

SET PhoneNo = '514-555-0000'

WHERE SecondaryFamilyMemberID = 1;
```

```
-- DELETE
```

```
-- Delete from SecondaryFamilyMember

DELETE FROM SecondaryFamilyMember

WHERE SecondaryFamilyMemberID = 1;
```

```
-- Delete from FamilyMember only if not referenced by ClubMember_FamilyMember
```

```

DELETE FROM FamilyMember
WHERE FamilyMemberID = 10020
AND NOT EXISTS (
    SELECT 1 FROM minorFamilyHistory
    WHERE FamilyMemberID = 10020
);

```

## RESULTS

The screenshot shows two separate result grids from a database application.

**Result Grid 1 (FamilyMember table):**

FamilyMemberID	FirstName	LastName	PhoneNo	Email
4	Lisa	Brown	514-555-0004	lisa.brown@mvc.com
9	Robert	Anderson	514-555-0009	robert.anderson@email.com
10	Mary	Anderson	514-555-0010	mary.anderson@email.com
11	Thomas	Miller	450-555-0011	thomas.miller@email.com
12	Jennifer	Miller	450-555-0012	jennifer.miller@email.com
13	William	Garcia	514-555-0013	william.garcia@email.com
14	Patricia	Garcia	514-555-0014	patricia.garcia@email.com
10020	Carol	Foster	514-555-8888	carol.foster@mvc.com

**Result Grid 2 (minorFamilyHistory table):**

SecondaryFamilyMemberID	FirstName	LastName	PhoneNo	PrimaryFamilyMemberID
1	Margaret	Anderson	514-555-0101	9
2	Helen	Miller	450-555-0102	11
3	Carlos	Garcia	514-555-0103	13
4	Linda	Johnson	514-555-0104	9
5	Paul	Brown	450-555-0105	12
<b>NUL</b>	<b>NUL</b>	<b>NUL</b>	<b>NUL</b>	<b>NUL</b>

```

INSERT INTO Person (
    FirstName, LastName, DateOfBirth, SSN, MedicareNo,
    PhoneNo, Address, City, Province, PostalCode, Email
) VALUES (
    'Omar', 'Hassani', '1998-09-12', '921-335-788', 'HAS980912',
    '438-555-0198', '2143 Papineau Ave', 'Montreal', 'QC', 'H2K4J5', 'omar.hassani@mvc.com'
);

```

-- Step 2: Insert into ClubMember (locationID = 2 → existing location)

```
INSERT INTO ClubMember (
    ClubMemberID, Height, Weight, isMinor, locationID, PaymentStatus
) VALUES (
    LAST_INSERT_ID(), 185.6, 82.3, FALSE, 2, TRUE
);
```

-- Step 1: Insert into Person

```
INSERT INTO Person (
    FirstName, LastName, DateOfBirth, SSN, MedicareNo,
    PhoneNo, Address, City, Province, PostalCode, Email
) VALUES (
    'Selena', 'Morin', '2010-03-18', '843-229-432', 'MOR100318',
    '514-555-0284', '9650 Côte-des-Neiges', 'Montreal', 'QC', 'H3V1G2',
    'selena.morin@mvc.com'
);
```

-- Step 2: Insert into ClubMember

```
INSERT INTO ClubMember (
    ClubMemberID, Height, Weight, isMinor, locationID, PaymentStatus
) VALUES (
    LAST_INSERT_ID(), 160.2, 48.5, TRUE, 1, FALSE
);
```

-- DISPLAY

```
SELECT
    cm.ClubMemberID,
```

```
p.FirstName,  
p.LastName,  
p.DateOfBirth,  
p.PhoneNo,  
cm.isMinor,  
cm.Height,  
cm.Weight,  
l.name AS Location,  
cm.PaymentStatus  
  
FROM ClubMember cm  
  
JOIN Person p ON cm.ClubMemberID = p.PersonID  
  
JOIN Location l ON cm.locationID = l.locationID;  
  
-- UPDATE  
  
UPDATE ClubMember  
  
SET Height = 162.0, PaymentStatus = TRUE  
  
WHERE ClubMemberID = (  
  
    SELECT PersonID FROM Person WHERE FirstName = 'Selena' AND LastName = 'Morin'  
);  
  
-- DELETE  
-- Delete from ClubMember (which references Person)  
  
DELETE FROM ClubMember  
  
WHERE ClubMemberID = (  
  
    SELECT PersonID FROM Person WHERE FirstName = 'Omar' AND LastName =  
    'Hassani'  
);
```

DELETE FROM Person

WHERE PersonID = 10003;

## **RESULTS**

ClubMemberID	FirstName	LastName	DateOfBirth	PhoneNo	isMinor	Height	Weight	Location	PaymentStatus
15	Alex	Anderson	2005-03-10	514-555-0015	0	175.50	70.20	Montreal Volleyball Club - Head Office	1
16	Sophie	Miller	2006-07-20	450-555-0016	0	168.30	62.80	Laval Branch	1
17	Lucas	Garcia	2004-11-08	514-555-0017	0	180.10	75.50	Montreal Volleyball Club - Head Office	0
18	Emma	Thompson	2003-05-25	514-555-0018	0	172.00	65.00	Montreal Volleyball Club - Head Office	1
19	Noah	White	2002-09-12	450-555-0019	0	185.20	80.10	Laval Branch	1
20	Olivia	Martin	2005-01-30	514-555-0020	0	165.80	58.50	Montreal Volleyball Club - Head Office	0
21	Ethan	Anderson	2010-08-18	514-555-0021	1	160.00	50.20	Montreal Volleyball Club - Head Office	1
22	Ava	Miller	2011-12-03	450-555-0022	1	155.50	48.00	Laval Branch	1
23	Mason	Garcia	2012-04-22	514-555-0023	1	158.20	45.50	Montreal Volleyball Club - Head Office	0
24	Isabella	Johnson	2013-06-10	514-555-0024	1	152.00	42.80	Montreal Volleyball Club - Head Office	1
25	Logan	Brown	2009-10-18	450-555-0025	1	165.50	55.00	Laval Branch	1
26	Mia	Davis	2014-02-14	514-555-0026	1	148.50	40.20	Montreal Volleyball Club - Head Office	0
9999	Test	Player	2000-01-01	514-123-4567	0	NULL	NULL	Montreal Volleyball Club - Head Office	1
10021	Omar	Hassani	1998-09-12	438-555-0198	0	185.60	82.30	Laval Branch	1
10022	Selena	Morin	2010-03-18	514-555-0284	1	160.20	48.50	Montreal Volleyball Club - Head Office	0

## **Query 5**

-- Create

-- Create a new training session first (if not already done)

INSERT INTO Session (SessionDate, SessionTime, Address, SessionType)

VALUES ('2025-08-10', '18:00:00', '4000 Rosemont Blvd, Montreal', 'Training');

-- Create the team formation (Team1 vs Team2)

INSERT INTO TeamFormation (SessionID, Team1ID, Team2ID, Team1Score, Team2Score)

VALUES (9003, 1, 6, NULL, NULL);

-- Display

-- List all team formations with session details

SELECT tf.FormationID, s.SessionDate, s.SessionTime, s.SessionType,

t1.teamName AS Team1, t2.teamName AS Team2,

tf.Team1Score, tf.Team2Score

```
FROM TeamFormation tf
JOIN Session s ON tf.SessionID = s.SessionID
JOIN Team t1 ON tf.Team1ID = t1.teamID
JOIN Team t2 ON tf.Team2ID = t2.teamID
ORDER BY s.SessionDate, s.SessionTime;
```

```
-- Edit
-- Update scores after a game is finished
UPDATE TeamFormation
SET Team1Score = 25,
    Team2Score = 22
WHERE FormationID = 8;
```

```
-- Or update the team match-up (only if necessary)
UPDATE TeamFormation
SET Team1ID = 2,
    Team2ID = 4
WHERE FormationID = 8;
```

```
-- Delete
--Delete from TeamFormationPlayer first (if players assigned)
DELETE FROM TeamFormationPlayer
WHERE FormationID = 8;
-- Then delete the formation
DELETE FROM TeamFormation
WHERE FormationID = 8;
```

## RESULTS

FormationID	SessionDate	SessionTime	SessionType	Team1	Team2	Team1Score	Team2Score
5	2025-01-13	18:00:00	Training	Laval Tigers Male	Laval Tigers Female	21	25
6	2025-01-14	19:30:00	Training	Montreal Eagles Female	Montreal Eagles Male	NULL	NULL
7	2025-01-15	20:00:00	Game	Laval Tigers Female	Laval Tigers Male	NULL	NULL
8	2025-01-16	18:30:00	Training	Montreal Eagles Female	Laval Tigers Female	25	18
1	2025-03-01	10:00:00	Game	Montreal Eagles Male	Montreal Juniors Male	NULL	NULL
8001	2025-03-01	10:00:00	Game	Montreal Eagles Male	Montreal Eagles Female	NULL	NULL
2	2025-03-05	11:00:00	Game	Laval Tigers Male	Laval Tigers Female	NULL	NULL
8002	2025-03-05	11:00:00	Game	Montreal Eagles Male	Montreal Eagles Female	NULL	NULL
3	2025-03-10	12:00:00	Game	Montreal Eagles Male	Montreal Eagles Female	25	23
8003	2025-03-10	12:00:00	Game	Montreal Eagles Male	Montreal Eagles Female	NULL	NULL
8005	2025-03-10	12:00:00	Game	Montreal Eagles Male	Montreal Juniors Male	NULL	NULL
8004	2025-03-15	13:00:00	Game	Montreal Eagles Male	Montreal Eagles Female	NULL	NULL
4	2025-03-15	13:00:00	Game	West Island Sharks Male	Montreal Eagles Male	NULL	NULL
2002	2025-08-08	11:30:00	Game	Montreal Eagles Male	Montreal Eagles Female	NULL	NULL
2003	2025-08-08	12:45:00	Game	Montreal Eagles Male	Montreal Eagles Female	NULL	NULL
2004	2025-08-08	12:45:00	Game	Montreal Eagles Male	Montreal Eagles Female	NULL	NULL

## Query 6

-- STEP 1: Create the Assignment Table

```
CREATE TABLE ClubMember_Formation (
    ClubMemberID INT,
    FormationID INT,
    PRIMARY KEY (ClubMemberID, FormationID),
    FOREIGN KEY (ClubMemberID) REFERENCES ClubMember(ClubMemberID),
    FOREIGN KEY (FormationID) REFERENCES TeamFormation(FormationID)
);
```

-- STEP 2: Create Trigger to Prevent Conflicting Assignments

```
DROP TRIGGER IF EXISTS prevent_conflicting_assignments;
DELIMITER //
CREATE TRIGGER prevent_conflicting_assignments
BEFORE INSERT ON ClubMember_Formation
FOR EACH ROW
BEGIN
```

```

DECLARE session_date DATE;

-- Get the session date of the target Formation
SELECT s.SessionDate
INTO session_date
FROM TeamFormation tf
JOIN Session s ON tf.SessionID = s.SessionID
WHERE tf.FormationID = NEW.FormationID;

-- Check if the club member is already assigned to any formation on the same date
IF EXISTS (
    SELECT 1
    FROM ClubMember_Formation cmf
    JOIN TeamFormation tf2 ON cmf.FORMATIONID = tf2.FORMATIONID
    JOIN Session s2 ON tf2.SessionID = s2.SessionID
    WHERE cmf.ClubMemberID = NEW.ClubMemberID
        AND s2.SessionDate = session_date
) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Conflict: Club member is already assigned to another
formation on this date';
END IF;

END;
//


DELIMITER ;

```

-- STEP 3: Assignments (Create + Conflict Test)

```
INSERT INTO TeamFormation (FormationID, SessionID, Team1ID, Team2ID, Team1Score,  
Team2Score)
```

VALUES

```
(1001, 2001, 301, 302, NULL, NULL),
```

```
(1002, 2001, 303, 1, NULL, NULL);
```

-- insert 5+ valid assignments

```
INSERT INTO ClubMember_Formation (ClubMemberID, FormationID) VALUES
```

```
(15, 1),
```

```
(16, 2),
```

```
(18, 3),
```

```
(19, 4),
```

```
(21, 1001);
```

-- Attempt conflicting assignment (should trigger error)

```
INSERT INTO ClubMember_Formation (ClubMemberID, FormationID)
```

```
VALUES (21, 1002);
```

-- STEP 4: Display

```
SELECT cmf.ClubMemberID, p.FirstName, p.LastName,  
cmf.FormationID, s.SessionDate, s.SessionTime, s.SessionType  
FROM ClubMember_Formation cmf
```

```

JOIN Person      p ON p.PersonID = cmf.ClubMemberID

JOIN TeamFormation tf ON tf.FormationID = cmf.FormationID

JOIN Session      s ON s.SessionID = tf.SessionID

ORDER BY s.SessionDate, s.SessionTime, cmf.ClubMemberID;

```

-- STEP 5: DELETE or UPDATE Assignments

-- Change assignment to a different formation

UPDATE ClubMember\_Formation

SET FormationID = 2002

WHERE ClubMemberID = 21 AND FormationID = 1001;

-- Delete assignment

DELETE FROM ClubMember\_Formation

WHERE ClubMemberID = 21 AND FormationID = 2002;

## **RESULTS**

ClubMemberID	FirstName	LastName	FormationID	SessionDate	SessionTime	SessionType
15	Alex	Anderson	1	2025-03-01	10:00:00	Game
16	Sophie	Miller	2	2025-03-05	11:00:00	Game
18	Emma	Thompson	3	2025-03-10	12:00:00	Game
19	Noah	White	4	2025-03-15	13:00:00	Game
21	Ethan	Anderson	1001	2025-08-10	14:00:00	Training

## **Query 7**

```

INSERT INTO Payment (ClubMemberID, PaymentDate, Amount, Method, MembershipYear)

VALUES (17, '2025-08-10', 40.00, 'Credit', 2025);

```

-- Installment 2

```
INSERT INTO Payment (ClubMemberID, PaymentDate, Amount, Method, MembershipYear)  
VALUES (17, '2025-08-20', 30.00, 'Credit', 2025);
```

-- Installment 3

```
INSERT INTO Payment (ClubMemberID, PaymentDate, Amount, Method, MembershipYear)  
VALUES (17, '2025-08-30', 30.00, 'Credit', 2025);
```

-- Installment 4

```
INSERT INTO Payment (ClubMemberID, PaymentDate, Amount, Method, MembershipYear)  
VALUES (17, '2025-09-01', 20.00, 'Cash', 2025);
```

-- Seed 1 payment each for four more existing members in 2025

```
INSERT INTO Payment (ClubMemberID, PaymentDate, Amount, Method, MembershipYear)  
VALUES  
(18, '2025-08-12', 20.00, 'Cash', 2025),  
(19, '2025-08-12', 20.00, 'Debit', 2025),  
(21, '2025-08-12', 20.00, 'Credit', 2025),  
(25, '2025-08-12', 20.00, 'Cash', 2025);
```

-- Display

```
SELECT ClubMemberID, MembershipYear, SUM(Amount) AS TotalPaid  
FROM Payment  
WHERE ClubMemberID IN (17, 18, 19, 21, 25) AND MembershipYear = 2025  
GROUP BY ClubMemberID, MembershipYear  
ORDER BY ClubMemberID;
```

## RESULTS

	ClubMemberID	MembershipYear	TotalPaid
▶	17	2025	120.00
	18	2025	20.00
	19	2025	20.00
	21	2025	20.00
	25	2025	20.00

### Query 8

SELECT

```
I.locationID,  
I.name AS LocationName,  
I.Address,  
I.City,  
I.Province,  
I.PostalCode,  
I.PhoneNo,  
I.WebAddress,  
I.Type,  
I.Capacity,
```

-- General Manager Name (Administrator with no end date)

(

```
    SELECT CONCAT(p.FirstName, ' ', p.LastName)
```

```
    FROM Personnel per
```

```
JOIN Personnel_Location pl ON per.PersonnelID = pl.PersonnelID  
JOIN Person p ON p.PersonID = per.PersonnelID  
WHERE pl.LocationID = l.locationID  
AND per.Role = 'Administrator'  
AND pl.EndDate IS NULL  
LIMIT 1  
) AS GeneralManagerName,
```

-- Count of minor members

```
COUNT(CASE WHEN cm.isMinor = TRUE THEN 1 END) AS MinorMembers,
```

-- Count of major members

```
COUNT(CASE WHEN cm.isMinor = FALSE THEN 1 END) AS MajorMembers,
```

-- Number of teams at this location

```
(  
SELECT COUNT(*)  
FROM Team t  
WHERE t.LocationID = l.locationID  
) AS TeamCount
```

```
FROM Location l
```

-- Left join to include locations even if they have no members

```
LEFT JOIN ClubMember cm ON cm.locationID = l.locationID
```

## GROUP BY

I.locationID, I.name, I.Address, I.City, I.Province, I.PostalCode,  
I.PhoneNo, I.WebAddress, I.Type, I.Capacity

## ORDER BY

I.Province ASC,  
I.City ASC;

## RESULTS

locationID	LocationName	Address	City	Province	PostalCode	PhoneNo	WebAddress	Type	Capacity	GeneralManagerName	MinorMember
8	South Shore Complex	800 Rue de Montarville	Boucherville	QC	J4B1C3	450-555-6004	www.mvc-southshore.com/d	Branch	90	NULL	0
5	South Shore Complex	5001 Taschereau Blvd	Brossard	QC	J4Z1B3	450-555-6001	www.mvc-southshore.com/a	Branch	90	NULL	0
3	West Island Branch	3000 Brunswick Blvd	Dollard-des-Ormeaux	QC	H9B1N7	514-555-3000	www.mvc-westisland.com	Branch	80	NULL	0
2	Laval Branch	2000 Boulevard Chomedey	Laval	QC	H7T2Z5	450-555-2000	www.mvc-laval.com	Branch	100	NULL	2
6	South Shore Complex	1200 Rue Saint-Charles	Longueuil	QC	J4K2T1	450-555-6002	www.mvc-southshore.com/b	Branch	90	NULL	0
1	Montreal Volleyball Club - Head Office	1000 Sherbrooke St W	Montreal	QC	H3A1G1	514-555-1000	www.mvc-montreal.com	Head	150	John Smith	5
7	South Shore Complex	2100 Rue Victoria	Saint-Lambert	QC	J4P2K1	450-555-6003	www.mvc-southshore.com/c	Branch	90	NULL	0

## Query 9

-- Adding data to get >= 5 tuples

INSERT INTO SecondaryFamilyMember (FirstName, LastName, PhoneNo, PrimaryFamilyMemberID) VALUES

('Edward', 'Anderson', '514-555-0301', 9),

('Martha', 'Anderson', '514-555-0302', 9),

('Daniel', 'Johnson', '514-555-0303', 9),

('Grace', 'Lee', '514-555-0304', 9);

-- Link each new secondary to an existing club member with a relationship type

-- (uses subqueries so you don't need to know the new IDs)

```
INSERT INTO SecondaryFamilyRelationship (SecondaryFamilyMemberID, ClubMemberID, relationshipTypeID)
```

```
SELECT s.SecondaryFamilyMemberID, 21, r.relationshipTypeID
```

```
FROM SecondaryFamilyMember s CROSS JOIN Relationship r
```

```
WHERE s.FirstName='Edward' AND s.LastName='Anderson' AND s.PrimaryFamilyMemberID=9
```

```
AND r.TypeName='Grandfather' LIMIT 1;
```

```
INSERT INTO SecondaryFamilyRelationship (SecondaryFamilyMemberID, ClubMemberID, relationshipTypeID)
```

```
SELECT s.SecondaryFamilyMemberID, 21, r.relationshipTypeID
```

```
FROM SecondaryFamilyMember s CROSS JOIN Relationship r
```

```
WHERE s.FirstName='Martha' AND s.LastName='Anderson' AND s.PrimaryFamilyMemberID=9
```

```
AND r.TypeName='Grandmother' LIMIT 1;
```

```
INSERT INTO SecondaryFamilyRelationship (SecondaryFamilyMemberID, ClubMemberID, relationshipTypeID)
```

```
SELECT s.SecondaryFamilyMemberID, 24, r.relationshipTypeID
```

```
FROM SecondaryFamilyMember s CROSS JOIN Relationship r
```

```
WHERE s.FirstName='Daniel' AND s.LastName='Johnson' AND s.PrimaryFamilyMemberID=9
```

```
AND r.TypeName='Tutor' LIMIT 1;
```

```
INSERT INTO SecondaryFamilyRelationship (SecondaryFamilyMemberID, ClubMemberID, relationshipTypeID)
```

```
SELECT s.SecondaryFamilyMemberID, 15, r.relationshipTypeID  
FROM SecondaryFamilyMember s CROSS JOIN Relationship r  
WHERE s.FirstName='Grace' AND s.LastName='Lee' AND s.PrimaryFamilyMemberID=9  
AND r.TypeName='Friend' LIMIT 1;
```

```
SELECT  
  
    sfm.FirstName AS SecondaryFirstName,  
    sfm.LastName AS SecondaryLastName,  
    sfm.PhoneNo AS SecondaryPhone,  
    cm.ClubMemberID,  
    p.FirstName AS ClubMemberFirstName,  
    p.LastName AS ClubMemberLastName,  
    p.DateOfBirth,  
    p.SSN,  
    p.MedicareNo,  
    p.PhoneNo,  
    p.Address,  
    p.City,  
    p.Province,  
    p.PostalCode,  
    r.TypeName AS RelationshipWithSecondary  
  
FROM SecondaryFamilyMember sfm  
  
JOIN SecondaryFamilyRelationship sfr ON sfm.SecondaryFamilyMemberID =  
sfr.SecondaryFamilyMemberID  
  
JOIN ClubMember cm ON sfr.ClubMemberID = cm.ClubMemberID  
  
JOIN Person p ON cm.ClubMemberID = p.PersonID
```

JOIN Relationship r ON sfr.relationshipTypeID = r.relationshipTypeID

WHERE sfm.PrimaryFamilyMemberID = 9;

## **RESULTS**

SecondaryFirstName	SecondaryLastName	SecondaryPhone	ClubMemberID	ClubMemberFirstName	ClubMemberLastName	DateOfBirth	SSN	MedicareNo	PhoneNo	Address	City	Province	PostalCode	RelationshipWithSecond
Linda	Johnson	514-555-0104	24	Isabella	Johnson	2013-06-10	456-789-014	JOHN130601	514-555-0024	468 Beech Ave	Montreal	QC	H50505	Other
Edward	Anderson	514-555-0301	21	Ethan	Anderson	2010-08-15	123-456-781	ANDE100801	514-555-0021	369 Willow Ave	Montreal	QC	H1T9T9	Grandfather
Martha	Anderson	514-555-0302	21	Ethan	Anderson	2010-08-15	123-456-781	ANDE100801	514-555-0021	369 Willow Ave	Montreal	QC	H1T9T9	Grandmother
Daniel	Johnson	514-555-0303	24	Isabella	Johnson	2013-06-10	456-789-014	JOHN130601	514-555-0024	468 Beech Ave	Montreal	QC	H50505	Tutor
Grace	Lee	514-555-0304	15	Alex	Anderson	2005-03-10	567-890-124	ANDE050301	514-555-0015	369 Willow Ave	Montreal	QC	H1T9T9	Friend

tf.FormationID,

s.SessionDate,

s.SessionTime,

s.Address AS SessionAddress,

s.SessionType,

t.teamName AS TeamName

hc.FirstName AS HeadCoachFirstName,

hc.LastName AS HeadCoachLastName,

tf.Team1Score,

tf.Team2Score,

p.FirstName AS PlayerFirstName,

p.LastName AS PlayerLastName,

pp.PositionName AS PlayerRole

```
FROM TeamFormation tf
-- Join session info
JOIN Session s ON tf.SessionID = s.SessionID

-- Join each team (Team1 or Team2) in the formation
JOIN Team t ON t.teamID = tf.Team1ID OR t.teamID = tf.Team2ID

-- Join the location of the team
JOIN Location l ON t.LocationID = l.locationID

-- Join Head Coach details (from Personnel and Person)
JOIN Personnel hc_per ON t.HeadCoachID = hc_per.PersonnelID
JOIN Person hc ON hc_per.PersonnelID = hc.PersonID

-- Join TeamFormationPlayer to get players in this formation & team
JOIN TeamFormationPlayer tfp
ON tfp.FormationID = tf.FormationID AND tfp.TeamID = t.teamID

-- Join ClubMember and their personal info
JOIN ClubMember cm ON tfp.ClubMemberID = cm.ClubMemberID
JOIN Person p ON cm.ClubMemberID = p.PersonID

-- Join Player's position info
JOIN PlayerPosition pp ON tfp.PositionID = pp.PositionID
```

-- Filter by location and session date range

WHERE l.name = 'Montreal Volleyball Club - Head Office'

AND s.SessionDate BETWEEN '2025-01-01' AND '2025-05-31'

-- Sort by session start time

ORDER BY s.SessionDate ASC, s.SessionTime ASC;

## **RESULTS**

FormationID	SessionDate	SessionTime	SessionAddress	SessionType	TeamName	HeadCoachFirstName	HeadCoachLastName	Team1Score	Team2Score	PlayerFirstName	PlayerLastName	PlayerRole
1	2025-03-01	10:00:00	Gym A	Game	Montreal Eagles Male	Mike	Williams	NULL	NULL	Alex	Anderson	Outside Hitter
1	2025-03-01	10:00:00	Gym A	Game	Montreal Eagles Male	Mike	Williams	NULL	NULL	Lucas	Garcia	Setter
1	2025-03-01	10:00:00	Gym A	Game	Montreal Juniors Male	David	Davis	NULL	NULL	Ethan	Anderson	Libero
1	2025-03-01	10:00:00	Gym A	Game	Montreal Juniors Male	David	Davis	NULL	NULL	Mason	Garcia	Middle Blocker
8001	2025-03-01	10:00:00	Gym A	Game	Montreal Eagles Male	Mike	Williams	NULL	NULL	Test	Player	Setter
8002	2025-03-05	11:00:00	Gym B	Game	Montreal Eagles Male	Mike	Williams	NULL	NULL	Test	Player	Libero
3	2025-03-10	12:00:00	Gym C	Game	Montreal Eagles Male	Mike	Williams	25	23	Alex	Anderson	Outside Hitter
3	2025-03-10	12:00:00	Gym C	Game	Montreal Eagles Male	Mike	Williams	25	23	Lucas	Garcia	Setter
3	2025-03-10	12:00:00	Gym C	Game	Montreal Eagles Female	Lisa	Brown	25	23	Emma	Thompson	Middle Blocker
8003	2025-03-10	12:00:00	Gym C	Game	Montreal Eagles Male	Mike	Williams	NULL	NULL	Test	Player	Outside Hitter
8004	2025-03-15	13:00:00	Gym D	Game	Montreal Eagles Male	Mike	Williams	NULL	NULL	Test	Player	Opposite Hitter

## **Query 11**

-- 5+ results

INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)  
VALUES

(17, 2, '2022-04-01', '2023-12-31'),

(23, 2, '2022-03-01', '2023-10-15'),

(26, 3, '2023-01-01', '2023-12-31');

SELECT PersonID, FirstName, LastName, Email FROM Person;

-- Add 1 new unpaid member with 2+ years and 2 locations

INSERT INTO Person (PersonID, FirstName, LastName, DateOfBirth, SSN, MedicareNo, PhoneNo, Address, City, Province, PostalCode, Email) VALUES

(10023, 'Bruno', 'Caron', '1989-02-14', '915-222-114', 'CARO890214', '514-555-3210', '77 St-Urbain', 'Montreal', 'QC', 'H2W1Z2', 'bruno.caron@email.com');

INSERT INTO ClubMember (ClubMemberID, Height, Weight, isMinor, locationID, PaymentStatus) VALUES

(10023, 178.0, 74.5, FALSE, 2, FALSE);

INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate) VALUES

(10023, 2, '2022-02-01', '2023-06-30'),

(10023, 1, '2023-07-01', NULL);

SELECT

cm.ClubMemberID,

p.FirstName,

p.LastName

FROM ClubMember cm

JOIN Person p ON cm.ClubMemberID = p.PersonID

JOIN (

SELECT ClubMemberID

FROM ClubMemberLocation

GROUP BY ClubMemberID

HAVING COUNT(DISTINCT LocationID) >= 2

) multi\_loc ON cm.ClubMemberID = multi\_loc.ClubMemberID

JOIN (

SELECT ClubMemberID, MIN(StartDate) AS FirstJoinDate

```

FROM ClubMemberLocation
GROUP BY ClubMemberID
) member_duration ON cm.ClubMemberID = member_duration.ClubMemberID
WHERE cm.PaymentStatus = FALSE
AND DATEDIFF(CURDATE(), member_duration.FirstJoinDate) >= 730
ORDER BY cm.ClubMemberID ASC;

```

## RESULTS

ClubMemberID	FirstName	LastName
17	Lucas	Garcia
20	Olivia	Martin
23	Mason	Garcia
26	Mia	Davis
10023	Bruno	Caron

## Query 12

```

/* 1) Create 2 South Shore teams at existing locations (Brossard + Longueuil) */

INSERT INTO Team (teamName, team_gender, LocationID, HeadCoachID)
SELECT 'South Shore Brossard Hawks','Male', l.locationID, 3
FROM Location l
WHERE l.name='South Shore Complex' AND l.Address='5001 Taschereau Blvd'
LIMIT 1;

```

```

INSERT INTO Team (teamName, team_gender, LocationID, HeadCoachID)
SELECT 'South Shore Longueuil Lynx','Male', l.locationID, 4

```

FROM Location l

WHERE l.name='South Shore Complex' AND l.Address='1200 Rue Saint-Charles'

LIMIT 1;

/\* 2) Add Game sessions within 2025-01-01 .. 2025-05-31 (dates are all within range)  
\*/

INSERT INTO Session (SessionDate, SessionTime, Address, SessionType)  
VALUES

('2025-02-01','18:00:00','Laval Gym A','Game'),

('2025-02-08','18:00:00','Laval Gym B','Game'),

('2025-03-01','18:00:00','Laval Gym C','Game'),

('2025-02-03','18:30:00','West Island Gym A','Game'),

('2025-02-10','18:30:00','West Island Gym B','Game'),

('2025-03-03','18:30:00','West Island Gym C','Game'),

('2025-02-05','19:00:00','Brossard Complex 1','Game'),

('2025-02-12','19:00:00','Brossard Complex 2','Game'),

('2025-03-05','19:00:00','Brossard Complex 3','Game'),

('2025-04-05','19:00:00','Brossard Complex 4','Game'),

('2025-02-06','19:30:00','Longueuil Complex 1','Game'),

('2025-02-13','19:30:00','Longueuil Complex 2','Game'),

('2025-03-06','19:30:00','Longueuil Complex 3','Game'),

('2025-04-06','19:30:00','Longueuil Complex 4','Game');

```
/* 3) Create formations with the home team as Team1 (so they count toward that location) */
```

```
/* Laval (needs 3 more, plus the one you already had) */
```

```
INSERT INTO TeamFormation (SessionID, Team1ID, Team2ID)
```

```
SELECT s.SessionID, 3, 1
```

```
FROM Session s
```

```
WHERE (s.SessionDate, s.SessionTime) IN ('2025-02-01','18:00:00'),
```

```
      ('2025-02-08','18:00:00'),
```

```
      ('2025-03-01','18:00:00'));
```

```
/* West Island (needs 3 more, plus the one you already had) */
```

```
INSERT INTO TeamFormation (SessionID, Team1ID, Team2ID)
```

```
SELECT s.SessionID, 5, 1
```

```
FROM Session s
```

```
WHERE (s.SessionDate, s.SessionTime) IN ('2025-02-03','18:30:00'),
```

```
      ('2025-02-10','18:30:00'),
```

```
      ('2025-03-03','18:30:00'));
```

```
/* South Shore Brossard (new team) */
```

```
INSERT INTO TeamFormation (SessionID, Team1ID, Team2ID)
```

```
SELECT s.SessionID,
```

```
      (SELECT teamID FROM Team WHERE teamName='South Shore Brossard Hawks' LIMIT 1),
```

1

```
FROM Session s  
  
WHERE (s.SessionDate, s.SessionTime) IN ('2025-02-05','19:00:00'),  
  
      ('2025-02-12','19:00:00'),  
  
      ('2025-03-05','19:00:00'),  
  
      ('2025-04-05','19:00:00'));
```

```
/* South Shore Longueuil (new team) */
```

```
INSERT INTO TeamFormation (SessionID, Team1ID, Team2ID)  
  
SELECT s.SessionID,  
  
       (SELECT teamID FROM Team WHERE teamName='South Shore Longueuil  
Lynx' LIMIT 1),
```

1

```
FROM Session s  
  
WHERE (s.SessionDate, s.SessionTime) IN ('2025-02-06','19:30:00'),  
  
      ('2025-02-13','19:30:00'),  
  
      ('2025-03-06','19:30:00'),  
  
      ('2025-04-06','19:30:00'));
```

```
/* 4) Give each new formation at least 1 player so player totals > 0 */
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID,  
PositionID)  
  
SELECT tf FormationID, tf.Team1ID, 9999,  
  
       (SELECT PositionID FROM PlayerPosition WHERE PositionName='Setter'  
LIMIT 1)
```

```
FROM TeamFormation tf
JOIN Session s ON s.SessionID = tf.SessionID
WHERE (s.SessionDate, s.SessionTime) IN ('2025-02-01','18:00:00'),
      ('2025-02-08','18:00:00'),
      ('2025-03-01','18:00:00'),
      ('2025-02-03','18:30:00'),
      ('2025-02-10','18:30:00'),
      ('2025-03-03','18:30:00'),
      ('2025-02-05','19:00:00'),
      ('2025-02-12','19:00:00'),
      ('2025-03-05','19:00:00'),
      ('2025-04-05','19:00:00'),
      ('2025-02-06','19:30:00'),
      ('2025-02-13','19:30:00'),
      ('2025-03-06','19:30:00'),
      ('2025-04-06','19:30:00'));
```

```
SELECT
```

```
I.locationID,
I.name AS LocationName,
I.City,
```

```
-- totals by session type
```

```
SUM(CASE WHEN s.SessionType = 'Training' THEN 1 ELSE 0 END) AS
TotalTrainingSessions,
```

```
SUM(CASE WHEN s.SessionType = 'Training' THEN  
COALESCE(pcnt.playerCount, 0) ELSE 0 END) AS TotalPlayersInTraining,
```

```
SUM(CASE WHEN s.SessionType = 'Game' THEN 1 ELSE 0 END) AS  
TotalGameSessions,
```

```
SUM(CASE WHEN s.SessionType = 'Game' THEN COALESCE(pcnt.playerCount,  
0) ELSE 0 END) AS TotalPlayersInGame
```

```
FROM TeamFormation tf
```

```
JOIN Session s ON s.SessionID = tf.SessionID
```

```
JOIN Team t1 ON t1.teamID = tf.Team1ID
```

```
JOIN Location l ON l.locationID = t1.LocationID
```

```
-- pre-count players per formation
```

```
LEFT JOIN (
```

```
SELECT FormationID, COUNT(*) AS playerCount
```

```
FROM TeamFormationPlayer
```

```
GROUP BY FormationID
```

```
) pcnt ON pcnt.FormationID = tf.FormationID
```

```
WHERE s.SessionDate BETWEEN '2025-01-01' AND '2025-05-31'
```

```
GROUP BY l.locationID, l.name, l.City
```

```
HAVING TotalGameSessions >= 4
```

```
ORDER BY TotalGameSessions DESC, l.locationID;
```

## **RESULTS**

Result Grid | Filter Rows:  Export:  Wrap Cell Content:

locationID	LocationName	City	TotalTrainingSessions	TotalPlayersInTraining	TotalGameSessions	TotalPlayersInGame
1	Montreal Volleyball Club - Head Office	Montreal	1	0	7	11
2	Laval Branch	Laval	1	0	5	7
3	West Island Branch	Dollard-des-Ormeaux	0	0	4	3
5	South Shore Complex	Brossard	0	0	4	4
6	South Shore Complex	Longueuil	0	0	4	4

## Query 13

-- Ensure member 1001 has a current location (he has no formation assignments)

```
INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)
VALUES (1001, 1, '2025-01-01', NULL);
```

-- Member A: Nadia Roy (active, no formations)

```
INSERT INTO Person (FirstName, LastName, DateOfBirth, SSN, MedicareNo, PhoneNo,
Address, City, Province, PostalCode, Email)
VALUES ('Nadia', 'Roy', '2010-04-12', '915-450-101', 'ROY100412B', '514-555-4101',
'12 Parc Ave', 'Montreal', 'QC', 'H2X1Y9', 'nadia.roy@email.com');
```

```
INSERT INTO ClubMember (ClubMemberID, Height, Weight, isMinor, locationID,
PaymentStatus)
```

```
VALUES (LAST_INSERT_ID(), 162.0, 52.0, TRUE, 1, TRUE);
```

```
INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)
```

```
VALUES (LAST_INSERT_ID(), 1, '2025-01-10', NULL);
```

-- Member B: Leo Martinez (active, no formations)

```
INSERT INTO Person (FirstName, LastName, DateOfBirth, SSN, MedicareNo, PhoneNo,
Address, City, Province, PostalCode, Email)

VALUES ('Leo', 'Martinez', '2007-11-03', '915-450-102', 'MART071103B', '438-555-4102',
'88 Sainte-Catherine St', 'Montreal', 'QC', 'H3B1K1', 'leo.martinez@email.com');
```

```
INSERT INTO ClubMember (ClubMemberID, Height, Weight, isMinor, locationID,
PaymentStatus)

VALUES (LAST_INSERT_ID(), 175.0, 68.0, FALSE, 2, TRUE);
```

```
INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)

VALUES (LAST_INSERT_ID(), 2, '2025-01-12', NULL);
```

```
/* Member C: Zoe Lam (active, no formations) */
```

```
INSERT INTO Person (FirstName, LastName, DateOfBirth, SSN, MedicareNo, PhoneNo,
Address, City, Province, PostalCode, Email)

VALUES ('Zoe', 'Lam', '2012-06-25', '915-450-103', 'LAM120625B', '514-555-4103',
'300 Sherbrooke St E', 'Montreal', 'QC', 'H2X1E6', 'zoe.lam@email.com');
```

```
INSERT INTO ClubMember (ClubMemberID, Height, Weight, isMinor, locationID,
PaymentStatus)

VALUES (LAST_INSERT_ID(), 160.5, 49.0, TRUE, 3, TRUE);
```

```
INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)

VALUES (LAST_INSERT_ID(), 3, '2025-01-15', NULL);
```

```
-- REPORT: Active members with no team formation assignments
```

```
SELECT
```

```

cm.ClubMemberID,
p.FirstName,
p.LastName,
TIMESTAMPDIFF(YEAR, p.DateOfBirth, CURDATE()) AS Age,
p.PhoneNo,
p.Email,
l.name AS CurrentLocation
FROM ClubMember cm
JOIN Person p          ON cm.ClubMemberID = p.PersonID
JOIN ClubMemberLocation cml ON cm.ClubMemberID = cml.ClubMemberID
JOIN Location l        ON cml.LocationID = l.locationID
WHERE cm.PaymentStatus = TRUE
AND cml.EndDate IS NULL
AND cm.ClubMemberID NOT IN (SELECT DISTINCT ClubMemberID FROM
TeamFormationPlayer)
ORDER BY l.name ASC, Age ASC;

```

## **RESULTS**

Result Grid						
<input type="checkbox"/> Filter Rows: _____ <input type="button" value="Export:"/> Wrap Cell Content: <input type="checkbox"/>						
ClubMemberID	FirstName	LastName	Age	PhoneNo	Email	CurrentLocation
10025	Leo	Martinez	17	438-555-4102	leo.martinez@email.com	Laval Branch
24	Isabella	Johnson	12	514-555-0024	isabella.johnson@email.com	Montreal Volleyball Club - Head Office
1001	Tommy	Nguyen	12	514-555-1010	tommy@example.com	Montreal Volleyball Club - Head Office
10024	Nadia	Roy	15	514-555-4101	nadia.roy@email.com	Montreal Volleyball Club - Head Office
10026	Zoe	Lam	13	514-555-4103	zoe.lam@email.com	West Island Branch

## **Query 14**

-- members since they were minors

-- Member 1 — Gabriel Moreau

INSERT INTO Person (FirstName, LastName, DateOfBirth, SSN, MedicareNo, PhoneNo, Address, City, Province, PostalCode, Email)

VALUES

('Gabriel', 'Moreau', '2006-02-15', '915-460-201', 'MORE060215', '514-555-6101', '12 Parc Ave', 'Montreal', 'QC', 'H2X1Y9', 'gabriel.moreau@email.com');

INSERT INTO ClubMember (ClubMemberID, Height, Weight, isMinor, locationID, PaymentStatus)

VALUES (LAST\_INSERT\_ID(), 180.0, 72.0, FALSE, 1, TRUE);

INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)

VALUES (LAST\_INSERT\_ID(), 1, '2022-09-01', NULL);

-- Member 2

INSERT INTO Person (FirstName, LastName, DateOfBirth, SSN, MedicareNo, PhoneNo, Address, City, Province, PostalCode, Email)

VALUES

('Chloe', 'Dubois', '2007-06-01', '915-460-202', 'DUBO070601', '514-555-6102', '88 Ste-Catherine St', 'Montreal', 'QC', 'H3B1K1', 'chloe.dubois@email.com');

INSERT INTO ClubMember (ClubMemberID, Height, Weight, isMinor, locationID, PaymentStatus)

VALUES (LAST\_INSERT\_ID(), 168.0, 60.0, FALSE, 2, TRUE);

INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)

VALUES (LAST\_INSERT\_ID(), 2, '2024-01-10', NULL);

-- Member 3

INSERT INTO Person (FirstName, LastName, DateOfBirth, SSN, MedicareNo, PhoneNo, Address, City, Province, PostalCode, Email)

VALUES  
('Rafael','Costa','2005-10-20','915-460-203','COST051020','438-555-6103','300  
Sherbrooke St E','Montreal','QC','H2X1E6','rafael.costa@email.com');

INSERT INTO ClubMember (ClubMemberID, Height, Weight, isMinor, locationID, PaymentStatus)

VALUES (LAST\_INSERT\_ID(), 182.0, 78.0, FALSE, 3, TRUE);

INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)

VALUES (LAST\_INSERT\_ID(), 3, '2022-05-01', NULL);

-- Member 4 — Elena Petrova

INSERT INTO Person (FirstName, LastName, DateOfBirth, SSN, MedicareNo, PhoneNo, Address, City, Province, PostalCode, Email)

VALUES  
('Elena','Petrova','2006-12-05','915-460-204','PETR061205','514-555-6104','77  
St-Urbain','Montreal','QC','H2W1Z2','elena.petrova@email.com');

INSERT INTO ClubMember (ClubMemberID, Height, Weight, isMinor, locationID, PaymentStatus)

VALUES (LAST\_INSERT\_ID(), 170.0, 58.0, FALSE, 1, TRUE);

INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)

VALUES (LAST\_INSERT\_ID(), 1, '2023-06-01', NULL);

-- Member 5 — Mateo Rossi

INSERT INTO Person (FirstName, LastName, DateOfBirth, SSN, MedicareNo, PhoneNo, Address, City, Province, PostalCode, Email)

VALUES  
('Mateo','Rossi','2006-03-12','915-460-205','ROSS060312','438-555-6105','15  
Ontario St','Montreal','QC','H2X1T5','mateo.rossi@email.com');

```
INSERT INTO ClubMember (ClubMemberID, Height, Weight, isMinor, locationID, PaymentStatus)
VALUES (LAST_INSERT_ID(), 179.0, 73.0, FALSE, 2, TRUE);

INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)
VALUES (LAST_INSERT_ID(), 2, '2022-08-15', NULL);
```

```
SELECT
    cm.ClubMemberID,
    p.FirstName,
    p.LastName,
    fj.DateJoined,
    TIMESTAMPDIFF(YEAR, p.DateOfBirth, CURDATE()) AS Age,
    p.PhoneNo,
    p.Email,
    l.name AS LocationName
FROM ClubMember cm
JOIN Person p ON p.PersonID = cm.ClubMemberID

JOIN (
    SELECT ClubMemberID, MIN(StartDate) AS DateJoined
    FROM ClubMemberLocation
    GROUP BY ClubMemberID
) fj ON fj.ClubMemberID = cm.ClubMemberID
```

```

JOIN (
    SELECT ClubMemberID, LocationID
    FROM ClubMemberLocation
    WHERE EndDate IS NULL
) curr ON curr.ClubMemberID = cm.ClubMemberID
JOIN Location l ON l.locationID = curr.LocationID
WHERE cm.PaymentStatus = TRUE
AND cm.isMinor = FALSE
AND TIMESTAMPDIFF(YEAR, p.DateOfBirth, fj.DateJoined) < 18
ORDER BY l.name ASC, Age ASC;

```

## **RESULTS**

ClubMemberID	FirstName	LastName	DateJoined	Age	PhoneNo	Email	LocationName
10025	Leo	Martinez	2025-01-12	17	438-555-4102	leo.martinez@email.com	Laval Branch
10028	Chloe	Dubois	2024-01-10	18	514-555-6102	chloe.dubois@email.com	Laval Branch
16	Sophie	Miller	2024-01-01	19	450-555-0016	sophie.miller@email.com	Laval Branch
10031	Mateo	Rossi	2022-08-15	19	438-555-6105	mateo.rossi@email.com	Laval Branch
10030	Elena	Petrova	2023-06-01	19	514-555-6104	elena.petrova@email.com	Montreal Volleyball Club - Head Office
10027	Gabriel	Moreau	2022-09-01	19	514-555-6101	gabriel.moreau@email.com	Montreal Volleyball Club - Head Office
10029	Rafael	Costa	2022-05-01	19	438-555-6103	rafael.costa@email.com	West Island Branch

## **Query 15**

-- 1) Give member 101 (Trigger Tester) a current location (he's already a Setter in formation 2002)

```

INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)
VALUES (101, 1, '2025-01-01', NULL);

```

-- 2) Assign member 1001 (Tommy Nguyen, active) as a Setter (only role)

```

INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
VALUES (2002, 1, 1001, (SELECT PositionID FROM PlayerPosition WHERE
PositionName='Setter' LIMIT 1));

```

-- 3) Assign member 24 (Isabella Johnson, active) as a Setter (only role)

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
VALUES (3, 1, 24, (SELECT PositionID FROM PlayerPosition WHERE
PositionName='Setter' LIMIT 1));
```

-- 4) Create ONE new active member and assign only as Setter

```
INSERT INTO Person (FirstName, LastName, DateOfBirth, SSN, MedicareNo, PhoneNo,
Address, City, Province, PostalCode, Email)
```

```
VALUES ('Aiden', 'Clark', '2004-01-01', '915-470-301', 'CLAR040101', '514-555-7301',
'10 Peel St', 'Montreal', 'QC', 'H3B2T3', 'aiden.clark@email.com');
```

```
INSERT INTO ClubMember (ClubMemberID, Height, Weight, isMinor, locationID,
PaymentStatus)
```

```
VALUES (LAST_INSERT_ID(), 182.0, 76.0, FALSE, 1, TRUE);
```

```
INSERT INTO ClubMemberLocation (ClubMemberID, LocationID, StartDate, EndDate)
```

```
VALUES (LAST_INSERT_ID(), 1, '2025-01-20', NULL);
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
```

```
VALUES (1, 1, LAST_INSERT_ID(), (SELECT PositionID FROM PlayerPosition WHERE
PositionName='Setter' LIMIT 1));
```

```
SELECT
```

```
cm.ClubMemberID,
p.FirstName,
p.LastName,
```

```
TIMESTAMPDIFF(YEAR, p.DateOfBirth, CURDATE()) AS Age,  
p.PhoneNo,  
p.Email,  
l.name AS CurrentLocation  
FROM ClubMember cm  
JOIN Person p ON cm.ClubMemberID = p.PersonID  
JOIN ClubMemberLocation cml ON cm.ClubMemberID = cml.ClubMemberID  
JOIN Location l ON cml.LocationID = l.locationID  
WHERE cm.PaymentStatus = TRUE  
AND cml.EndDate IS NULL  
AND cm.ClubMemberID IN (  
    SELECT ClubMemberID  
    FROM TeamFormationPlayer tfp  
    JOIN PlayerPosition pp ON tfp.PositionID = pp.PositionID  
    WHERE pp.PositionName = 'Setter'  
)  
AND cm.ClubMemberID NOT IN (  
    SELECT ClubMemberID  
    FROM TeamFormationPlayer tfp  
    JOIN PlayerPosition pp ON tfp.PositionID = pp.PositionID  
    WHERE pp.PositionName <> 'Setter'  
)  
ORDER BY l.name ASC, cm.ClubMemberID ASC;
```

## **RESULTS**

ClubMemberID	FirstName	LastName	Age	PhoneNo	Email	CurrentLocation
25	Logan	Brown	15	450-555-0025	logan.brown@email.com	Laval Branch
24	Isabella	Johnson	12	514-555-0024	isabella.johnson@email.com	Montreal Volleyball Club - Head Office
101	Trigger	Tester	20	514-000-0000	trigger@test.com	Montreal Volleyball Club - Head Office
1001	Tommy	Nguyen	12	514-555-1010	tommy@example.com	Montreal Volleyball Club - Head Office
10032	Aiden	Clark	21	514-555-7301	aiden.clark@email.com	Montreal Volleyball Club - Head Office

## **QUERY 16**

-- Alex (15): add Setter on 9002, Opposite on 9004, Libero on 9028

INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)

SELECT tf.FormationID, tf.Team1ID, 15, pos.PositionID

FROM TeamFormation tf

JOIN Session s ON s.SessionID=tf.SessionID

JOIN PlayerPosition pos ON pos.PositionName='Setter'

LEFT JOIN TeamFormationPlayer tfp

ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND  
tfp.ClubMemberID=15

WHERE s.SessionType='Game' AND s.SessionID=9002 AND tfp.FORMATIONID IS NULL

LIMIT 1;

INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)

SELECT tf.FORMATIONID, tf.Team1ID, 15, pos.PositionID

FROM TeamFormation tf

JOIN Session s ON s.SessionID=tf.SessionID

JOIN PlayerPosition pos ON pos.PositionName='Opposite Hitter'

LEFT JOIN TeamFormationPlayer tfp

```
    ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND
    tfp.ClubMemberID=15

    WHERE s.SessionType='Game' AND s.SessionID=9004 AND tfp.FormationID IS NULL
    LIMIT 1;

    INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
    SELECT tf.FormationID, tf.Team1ID, 15, pos.PositionID
    FROM TeamFormation tf
    JOIN Session s ON s.SessionID=tf.SessionID
    JOIN PlayerPosition pos ON pos.PositionName='Libero'
    LEFT JOIN TeamFormationPlayer tfp
        ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND
        tfp.ClubMemberID=15

    WHERE s.SessionType='Game' AND s.SessionID=9028 AND tfp.FormationID IS NULL
    LIMIT 1;
```

-- Sophie (16): already Libero on 9002. Add Setter (9001), Outside (9003), Opposite (9004)

```
    INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
    SELECT tf.FormationID, tf.Team1ID, 16, pos.PositionID
    FROM TeamFormation tf
    JOIN Session s ON s.SessionID=tf.SessionID
    JOIN PlayerPosition pos ON pos.PositionName='Setter'
    LEFT JOIN TeamFormationPlayer tfp
        ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND
        tfp.ClubMemberID=16

    WHERE s.SessionType='Game' AND s.SessionID=9001 AND tfp.FormationID IS NULL
    LIMIT 1;
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
SELECT tf.FormationID, tf.Team1ID, 16, pos.PositionID
FROM TeamFormation tf
JOIN Session s ON s.SessionID=tf.SessionID
JOIN PlayerPosition pos ON pos.PositionName='Outside Hitter'
LEFT JOIN TeamFormationPlayer tfp
    ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND
    tfp.ClubMemberID=16
WHERE s.SessionType='Game' AND s.SessionID=9003 AND tfp.FormationID IS NULL
LIMIT 1;
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
SELECT tf.FormationID, tf.Team1ID, 16, pos.PositionID
FROM TeamFormation tf
JOIN Session s ON s.SessionID=tf.SessionID
JOIN PlayerPosition pos ON pos.PositionName='Opposite Hitter'
LEFT JOIN TeamFormationPlayer tfp
    ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND
    tfp.ClubMemberID=16
WHERE s.SessionType='Game' AND s.SessionID=9004 AND tfp.FormationID IS NULL
LIMIT 1;
```

-- Emma (18): already Middle Blocker on 9003 (OK). Add Setter (9001), Libero (9002), Opposite (9004), Outside (9028)

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
SELECT tf.FormationID, tf.Team1ID, 18, pos.PositionID
FROM TeamFormation tf
```

```
JOIN Session s ON s.SessionID=tf.SessionID  
JOIN PlayerPosition pos ON pos.PositionName='Setter'  
LEFT JOIN TeamFormationPlayer tfp  
    ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND  
    tfp.ClubMemberID=18  
WHERE s.SessionType='Game' AND s.SessionID=9001 AND tfp.FormationID IS NULL  
LIMIT 1;
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)  
SELECT tf.FormationID, tf.Team1ID, 18, pos.PositionID  
FROM TeamFormation tf  
JOIN Session s ON s.SessionID=tf.SessionID  
JOIN PlayerPosition pos ON pos.PositionName='Libero'  
LEFT JOIN TeamFormationPlayer tfp  
    ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND  
    tfp.ClubMemberID=18  
WHERE s.SessionType='Game' AND s.SessionID=9002 AND tfp.FormationID IS NULL  
LIMIT 1;
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)  
SELECT tf.FormationID, tf.Team1ID, 18, pos.PositionID  
FROM TeamFormation tf  
JOIN Session s ON s.SessionID=tf.SessionID  
JOIN PlayerPosition pos ON pos.PositionName='Opposite Hitter'  
LEFT JOIN TeamFormationPlayer tfp  
    ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND  
    tfp.ClubMemberID=18
```

```
WHERE s.SessionType='Game' AND s.SessionID=9004 AND tfp.FormationID IS NULL  
LIMIT 1;
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)  
SELECT tf.FORMATIONID, tf.TEAM1ID, 18, pos.POSITIONID  
FROM TeamFormation tf  
JOIN Session s ON s.SessionID=tf.SessionID  
JOIN PlayerPosition pos ON pos.PositionName='Outside Hitter'  
LEFT JOIN TeamFormationPlayer tfp  
ON tfp.FORMATIONID=tf.FORMATIONID AND tfp.TEAMID=tf.TEAM1ID AND  
tfp.ClubMemberID=18  
WHERE s.SessionType='Game' AND s.SessionID=9028 AND tfp.FormationID IS NULL  
LIMIT 1;
```

```
-- Logan (25): already Setter on 9002. Add Libero (9001), Outside (9003), Opposite (9004)  
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)  
SELECT tf.FORMATIONID, tf.TEAM1ID, 25, pos.POSITIONID  
FROM TeamFormation tf  
JOIN Session s ON s.SessionID=tf.SessionID  
JOIN PlayerPosition pos ON pos.PositionName='Libero'  
LEFT JOIN TeamFormationPlayer tfp  
ON tfp.FORMATIONID=tf.FORMATIONID AND tfp.TEAMID=tf.TEAM1ID AND  
tfp.ClubMemberID=25  
WHERE s.SessionType='Game' AND s.SessionID=9001 AND tfp.FormationID IS NULL  
LIMIT 1;
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
SELECT tf.FormationID, tf.Team1ID, 25, pos.PositionID
FROM TeamFormation tf
JOIN Session s ON s.SessionID=tf.SessionID
JOIN PlayerPosition pos ON pos.PositionName='Outside Hitter'
LEFT JOIN TeamFormationPlayer tfp
    ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND
    tfp.ClubMemberID=25
WHERE s.SessionType='Game' AND s.SessionID=9003 AND tfp.FormationID IS NULL
LIMIT 1;
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
SELECT tf.FormationID, tf.Team1ID, 25, pos.PositionID
FROM TeamFormation tf
JOIN Session s ON s.SessionID=tf.SessionID
JOIN PlayerPosition pos ON pos.PositionName='Opposite Hitter'
LEFT JOIN TeamFormationPlayer tfp
    ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND
    tfp.ClubMemberID=25
WHERE s.SessionType='Game' AND s.SessionID=9004 AND tfp.FormationID IS NULL
LIMIT 1;
```

-- Trigger (101): has Setter on 3001. Add Libero (9001), Outside (9003), Opposite (9004)

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
SELECT tf.FormationID, tf.Team1ID, 101, pos.PositionID
FROM TeamFormation tf
```

```
JOIN Session s ON s.SessionID=tf.SessionID  
JOIN PlayerPosition pos ON pos.PositionName='Libero'  
LEFT JOIN TeamFormationPlayer tfp  
    ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND  
    tfp.ClubMemberID=101  
WHERE s.SessionType='Game' AND s.SessionID=9001 AND tfp.FormationID IS NULL  
LIMIT 1;
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)  
SELECT tf.FormationID, tf.Team1ID, 101, pos.PositionID  
FROM TeamFormation tf  
JOIN Session s ON s.SessionID=tf.SessionID  
JOIN PlayerPosition pos ON pos.PositionName='Outside Hitter'  
LEFT JOIN TeamFormationPlayer tfp  
    ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND  
    tfp.ClubMemberID=101  
WHERE s.SessionType='Game' AND s.SessionID=9003 AND tfp.FormationID IS NULL  
LIMIT 1;
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)  
SELECT tf.FormationID, tf.Team1ID, 101, pos.PositionID  
FROM TeamFormation tf  
JOIN Session s ON s.SessionID=tf.SessionID  
JOIN PlayerPosition pos ON pos.PositionName='Opposite Hitter'  
LEFT JOIN TeamFormationPlayer tfp  
    ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND  
    tfp.ClubMemberID=101
```

```
WHERE s.SessionType='Game' AND s.SessionID=9004 AND tfp.FormationID IS NULL  
LIMIT 1;
```

-- Tommy (1001): has Setter on 3001. Add Libero (9002), Outside (9001), Opposite (9004)

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
```

```
SELECT tf FormationID, tf.Team1ID, 1001, pos.PositionID
```

```
FROM TeamFormation tf
```

```
JOIN Session s ON s.SessionID=tf.SessionID
```

```
JOIN PlayerPosition pos ON pos.PositionName='Outside Hitter'
```

```
LEFT JOIN TeamFormationPlayer tfp
```

```
ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND  
tfp.ClubMemberID=1001
```

```
WHERE s.SessionType='Game' AND s.SessionID=9001 AND tfp.FormationID IS NULL
```

```
LIMIT 1;
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
```

```
SELECT tf FormationID, tf.Team1ID, 1001, pos.PositionID
```

```
FROM TeamFormation tf
```

```
JOIN Session s ON s.SessionID=tf.SessionID
```

```
JOIN PlayerPosition pos ON pos.PositionName='Libero'
```

```
LEFT JOIN TeamFormationPlayer tfp
```

```
ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND  
tfp.ClubMemberID=1001
```

```
WHERE s.SessionType='Game' AND s.SessionID=9002 AND tfp.FormationID IS NULL
```

```
LIMIT 1;
```

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
SELECT tf.FormationID, tf.Team1ID, 1001, pos.PositionID
FROM TeamFormation tf
JOIN Session s ON s.SessionID=tf.SessionID
JOIN PlayerPosition pos ON pos.PositionName='Opposite Hitter'
LEFT JOIN TeamFormationPlayer tfp
ON tfp.FormationID=tf.FormationID AND tfp.TeamID=tf.Team1ID AND
tfp.ClubMemberID=1001
WHERE s.SessionType='Game' AND s.SessionID=9004 AND tfp.FormationID IS NULL
LIMIT 1;
```

-- (9999 already has Setter, Libero, Outside, Opposite on game sessions—no inserts needed)

```
SELECT
cm.ClubMemberID,
p.FirstName,
p.LastName,
TIMESTAMPDIFF(YEAR, p.DateOfBirth, CURDATE()) AS Age,
p.PhoneNo,
p.Email,
l.name AS CurrentLocation
FROM ClubMember cm
JOIN Person p ON cm.ClubMemberID = p.PersonID
JOIN ClubMemberLocation cml ON cm.ClubMemberID = cml.ClubMemberID
JOIN Location l ON cml.LocationID = l.locationID
```

```
WHERE cm.PaymentStatus = TRUE  
AND cml.EndDate IS NULL  
AND cm.ClubMemberID IN (  
    SELECT tfp.ClubMemberID  
    FROM TeamFormationPlayer tfp  
    JOIN TeamFormation tf ON tfp.FormationID = tf.FormationID  
    JOIN Session s ON tf.SessionID = s.SessionID  
    JOIN PlayerPosition pp ON tfp.PositionID = pp.PositionID  
    WHERE s.SessionType = 'Game' AND pp.PositionName = 'Setter'  
)  
AND cm.ClubMemberID IN (  
    SELECT tfp.ClubMemberID  
    FROM TeamFormationPlayer tfp  
    JOIN TeamFormation tf ON tfp.FormationID = tf.FormationID  
    JOIN Session s ON tf.SessionID = s.SessionID  
    JOIN PlayerPosition pp ON tfp.PositionID = pp.PositionID  
    WHERE s.SessionType = 'Game' AND pp.PositionName = 'Libero'  
)  
AND cm.ClubMemberID IN (  
    SELECT tfp.ClubMemberID  
    FROM TeamFormationPlayer tfp  
    JOIN TeamFormation tf ON tfp.FormationID = tf.FormationID  
    JOIN Session s ON tf.SessionID = s.SessionID  
    JOIN PlayerPosition pp ON tfp.PositionID = pp.PositionID  
    WHERE s.SessionType = 'Game' AND pp.PositionName = 'Outside Hitter'
```

```

)
AND cm.ClubMemberID IN (
    SELECT tfp.ClubMemberID
    FROM TeamFormationPlayer tfp
    JOIN TeamFormation tf ON tfp.FormationID = tf.FormationID
    JOIN Session s ON tf.SessionID = s.SessionID
    JOIN PlayerPosition pp ON tfp.PositionID = pp.PositionID
    WHERE s.SessionType = 'Game' AND pp.PositionName = 'Opposite Hitter'
)
ORDER BY l.name ASC, cm.ClubMemberID ASC;

```

## **RESULTS**

ClubMemberID	Firstname	Lastname	Age	PhoneNo	Email	CurrentLocation
16	Sophie	Miller	19	450-555-0016	sophie.miller@email.com	Laval Branch
25	Logan	Brown	15	450-555-0025	logan.brown@email.com	Laval Branch
15	Alex	Anderson	20	514-555-0015	alex.anderson@email.com	Montreal Volleyball Club - Head Office
18	Emma	Thompson	22	514-555-0018	emma.thompson@email.com	Montreal Volleyball Club - Head Office
101	Trigger	Tester	20	514-000-0000	trigger@test.com	Montreal Volleyball Club - Head Office
1001	Tommy	Nguyen	12	514-555-1010	tommy@example.com	Montreal Volleyball Club - Head Office
9999	Test	Player	25	514-123-4567	test@example.com	Montreal Volleyball Club - Head Office

## **QUERY 17**

-- Make sure Mary (10), William (13), Patricia (14) can be head coaches (needed for Team.HeadCoachID FK)

```

INSERT INTO Personnel (PersonnelID, Role, Mandate) VALUES
(10, 'Head Coach', 'Volunteer'),
(13, 'Head Coach', 'Volunteer'),
(14, 'Head Coach', 'Volunteer');

```

-- Ensure William's and Patricia's associated club members are active at Location 1

-- (William -> Mason 23, Mia 26; Patricia -> Mason 23). Both 23/26 are already at LocationID=1.

```
UPDATE ClubMember SET PaymentStatus = TRUE WHERE ClubMemberID IN (23, 26);
```

-- Create teams at Location 1 with those family members as head coaches

```
INSERT INTO Team (teamName, team_gender, LocationID, HeadCoachID) VALUES  
('Head Office Mary Lions', 'Female', 1, 10),  
('Head Office Garcia Bears', 'Male', 1, 13),  
('Head Office Patricia Panthers','Female',1, 14);
```

-- Create a few sessions at Head Office timeframe

```
INSERT INTO Session (SessionDate, SessionTime, Address, SessionType) VALUES  
('2025-02-20','17:00:00','1000 Sherbrooke St W, Montreal','Game'),  
('2025-02-24','17:30:00','1000 Sherbrooke St W, Montreal','Training'),  
('2025-02-26','18:00:00','1000 Sherbrooke St W, Montreal','Game');
```

-- Make a formation for each new team (Team1 is the head coach's team; opponent can be team 1)

```
INSERT INTO TeamFormation (SessionID, Team1ID, Team2ID)
```

```
SELECT s.SessionID, t.teamID, 1
```

```
FROM Session s
```

```
JOIN Team t ON t.teamName = 'Head Office Mary Lions'
```

```
WHERE s.SessionDate='2025-02-20' AND s.SessionTime='17:00:00' LIMIT 1;
```

```
INSERT INTO TeamFormation (SessionID, Team1ID, Team2ID)
```

```
SELECT s.SessionID, t.teamID, 1
FROM Session s
JOIN Team t ON t.teamName = 'Head Office Garcia Bears'
WHERE s.SessionDate='2025-02-24' AND s.SessionTime='17:30:00' LIMIT 1;
```

```
INSERT INTO TeamFormation (SessionID, Team1ID, Team2ID)
```

```
SELECT s.SessionID, t.teamID, 1
FROM Session s
JOIN Team t ON t.teamName = 'Head Office Patricia Panthers'
```

```
WHERE s.SessionDate='2025-02-26' AND s.SessionTime='18:00:00' LIMIT 1;
```

-- Family members who:

- (a) are listed in FamilyMember,
  - (b) HEAD-COACH at least one team that appears in a TeamFormation at the given location,
  - (c) have at least one CURRENTLY ACTIVE club member associated to them at the SAME location.
- Replace the literal 1 with your desired LocationID if needed.

```
SELECT DISTINCT
```

```
p.FirstName,
p.LastName,
p.PhoneNo
```

```
FROM FamilyMember fm
```

```
JOIN Person p ON p.PersonID = fm.FamilyMemberID
```

-- Be a head coach by assignment on a team that appeared in at least one formation at this location

```

JOIN Team t
  ON t.HeadCoachID = fm.FamilyMemberID
  AND t.LocationID = 1
JOIN TeamFormation tf
  ON tf.Team1ID = t.teamID OR tf.Team2ID = t.teamID
-- Must have a currently ACTIVE associated club member at the same location
WHERE EXISTS (
  SELECT 1
  FROM minorFamilyHistory mfh
  JOIN ClubMember cm      ON cm.ClubMemberID = mfh.ClubMemberID
  JOIN ClubMemberLocation cml
    ON cml.ClubMemberID = cm.ClubMemberID
    AND cml.EndDate IS NULL
  WHERE mfh.FamilyMemberID = fm.FamilyMemberID
    AND cm.PaymentStatus = TRUE
    AND cml.LocationID = t.LocationID
)
ORDER BY p.LastName, p.FirstName;

```

## **RESULTS**

FirstName	LastName	PhoneNo
Mary	Anderson	514-555-0010
Robert	Anderson	514-555-0009
Lisa	Brown	514-555-0004
Patricia	Garcia	514-555-0014
William	Garcia	514-555-0013

## **QUERY 18**

```
SELECT  
    cm.ClubMemberID,  
    p.FirstName,  
    p.LastName,  
    TIMESTAMPDIFF(YEAR, p.DateOfBirth, CURDATE()) AS Age,  
    p.PhoneNo,  
    p.Email,  
    l.name AS CurrentLocation  
  
FROM ClubMember cm  
  
JOIN Person p  
    ON p.PersonID = cm.ClubMemberID  
  
JOIN ClubMemberLocation cml  
    ON cml.ClubMemberID = cm.ClubMemberID  
    AND cml.EndDate IS NULL      -- current location only  
  
JOIN Location l  
    ON l.locationID = cml.LocationID  
  
WHERE cm.PaymentStatus = TRUE      -- active members  
    -- played at least one GAME session (scores may be NULL or set)  
  
    AND EXISTS (  
        SELECT 1  
        FROM TeamFormationPlayer tfp  
        JOIN TeamFormation tf ON tf.FormationID = tfp.FormationID  
        JOIN Session s      ON s.SessionID = tf.SessionID  
        WHERE tfp.ClubMemberID = cm.ClubMemberID
```

```

AND s.SessionType = 'Game'

)

-- never on the losing side in any COMPLETED game (both scores set)

AND NOT EXISTS (

    SELECT 1

    FROM TeamFormationPlayer tfp

    JOIN TeamFormation tf ON tf.FormationID = tfp.FormationID

    JOIN Session s      ON s.SessionID = tf.SessionID

    WHERE tfp.ClubMemberID = cm.ClubMemberID

    AND s.SessionType = 'Game'

    AND tf.Team1Score IS NOT NULL

    AND tf.Team2Score IS NOT NULL

    AND (

        (tfp.TeamID = tf.Team1ID AND tf.Team1Score < tf.Team2Score) OR

        (tfp.TeamID = tf.Team2ID AND tf.Team2Score < tf.Team1Score)

    )

)

ORDER BY

l.name ASC,

cm.ClubMemberID ASC;

```

## **RESULTS**

ClubMemberID	FirstName	LastName	Age	PhoneNo	Email	CurrentLocation
16	Sophie	Miller	19	450-555-0016	sophie.miller@email.com	Laval Branch
19	Noah	White	22	450-555-0019	noah.white@email.com	Laval Branch
22	Ava	Miller	13	450-555-0022	ava.miller@email.com	Laval Branch
25	Logan	Brown	15	450-555-0025	logan.brown@email.com	Laval Branch
15	Alex	Anderson	20	514-555-0015	alex.anderson@email.com	Montreal Volleyball Club - Head Office
21	Ethan	Anderson	14	514-555-0021	ethan.anderson@email.com	Montreal Volleyball Club - Head Office
23	Mason	Garcia	13	514-555-0023	mason.garcia@email.com	Montreal Volleyball Club - Head Office
24	Isabella	Johnson	12	514-555-0024	isabella.johnson@email.com	Montreal Volleyball Club - Head Office
101	Trigger	Tester	20	514-000-0000	trigger@test.com	Montreal Volleyball Club - Head Office
1001	Tommy	Nguyen	12	514-555-1010	tommy@example.com	Montreal Volleyball Club - Head Office
9999	Test	Player	25	514-123-4567	test@example.com	Montreal Volleyball Club - Head Office
10032	Aiden	Clark	21	514-555-7301	aiden.clark@email.com	Montreal Volleyball Club - Head Office

## **QUERY 19**

-- Ensure these family members are volunteers (if not already)

UPDATE Personnel

SET Role='Head Coach', Mandate='Volunteer'

WHERE PersonnelID IN (10,13,14); -- Mary (10), William (13), Patricia (14)

-- Give them a current personnel location (unique start dates avoid PK clashes)

INSERT INTO Personnel\_Location (PersonnelID, LocationID, StartDate, EndDate)  
VALUES

(10, 1, '2025-02-20', NULL),

(13, 1, '2025-02-24', NULL),

(14, 1, '2025-02-26', NULL);

-- Make two existing volunteer coaches also family members of minors

INSERT INTO FamilyMember (FamilyMemberID) VALUES

(5), -- David Davis (Assistant Coach, Volunteer)

(8); -- Anna Taylor (Assistant Coach, Volunteer)

-- Link them to minors (uses existing minors 26 = Mia Davis, 24 = Isabella Johnson)

INSERT INTO minorFamilyHistory (relationshipTypeID, FamilyMemberID, ClubMemberID, startDate)

SELECT r.relationshipTypeID, 5, 26, '2025-02-01' -- David Davis -> Mia Davis  
(Tutor)

FROM Relationship r WHERE r.TypeName='Tutor' LIMIT 1;

```
INSERT INTO minorFamilyHistory (relationshipTypeID, FamilyMemberID,  
ClubMemberID, startDate)
```

```
SELECT r.relationshipTypeID, 8, 24, '2025-02-01' -- Anna Taylor -> Isabella Johnson  
(Other)
```

```
FROM Relationship r WHERE r.TypeName='Other' LIMIT 1;
```

```
SELECT
```

```
    per.PersonnelID,
```

```
    p.FirstName,
```

```
    p.LastName,
```

```
    COUNT(DISTINCT mfh.ClubMemberID) AS NumberOfMinors,
```

```
    p.PhoneNo,
```

```
    p.Email,
```

```
    l.name AS CurrentLocation,
```

```
    per.Role
```

```
FROM Personnel per
```

```
JOIN Person p ON per.PersonnelID = p.PersonID
```

```
JOIN Personnel_Location pl ON per.PersonnelID = pl.PersonnelID
```

```
JOIN Location l ON pl.LocationID = l.locationID
```

```
JOIN minorFamilyHistory mfh ON per.PersonnelID = mfh.FamilyMemberID
```

```
JOIN ClubMember cm ON mfh.ClubMemberID = cm.ClubMemberID
```

```
WHERE per.Mandate = 'Volunteer'
```

```
    AND pl.EndDate IS NULL
```

```
    AND cm.isMinor = TRUE
```

GROUP BY per.PersonnelID, p.FirstName, p.LastName, p.PhoneNo, p.Email, l.name, per.Role

ORDER BY l.name ASC, per.Role ASC, p.FirstName ASC, p.LastName ASC;

## **RESULTS**



A screenshot of a database result grid titled "Result Grid". The grid has columns: PersonnelID, FirstName, LastName, NumberOfMinors, PhoneNo, Email, CurrentLocation, and Role. There are 7 rows of data:

PersonnelID	FirstName	LastName	NumberOfMinors	PhoneNo	Email	CurrentLocation	Role
8	Anna	Taylor	1	450-555-0008	anna.taylor@mvc.com	Laval Branch	Assistant Coach
5	David	Davis	1	450-555-0005	david.davis@mvc.com	Laval Branch	Assistant Coach
4	Lisa	Brown	1	514-555-0004	lisa.brown@mvc.com	Montreal Volleyball Club - Head Office	Coach
10	Mary	Anderson	1	514-555-0010	mary.anderson@email.com	Montreal Volleyball Club - Head Office	Head Coach
14	Patricia	Garcia	1	514-555-0014	patricia.garcia@email.com	Montreal Volleyball Club - Head Office	Head Coach
13	William	Garcia	2	514-555-0013	william.garcia@email.com	Montreal Volleyball Club - Head Office	Head Coach

The system includes the following **four triggers** to enforce business constraints, ensure data integrity, and prevent scheduling conflicts automatically.

### **1. PreventPlayerTimeConflict**

#### **Purpose:**

Ensures that a player is not scheduled for multiple formation sessions **less than 3 hours apart** on the same day.

#### **Trigger Type:**

BEFORE INSERT on TeamFormationPlayer

#### **Business Rule Enforced:**

If a player is assigned to two team formations on the same day, then at least three hours of time difference must exist between their session start times.

#### **Benefit:**

- Prevents over-scheduling and physical strain on players.
- Maintains realistic and conflict-free team assignments.

### **2. EnforceMinimumAge**

#### **Purpose:**

Ensures that new club members are at least 11 years old at the time of registration.

#### **Trigger Type:**

BEFORE INSERT on ClubMember

#### **Business Rule Enforced:**

A new club member must be at least 11 years old to register.

**Benefit:**

- Maintains age eligibility rules.
- Avoids enrollment of members who are too young to participate.

### **3. PreventMultipleActiveTeams**

**Purpose:**

Prevents a club member from being actively assigned to **multiple teams during overlapping periods**.

**Trigger Type:**

BEFORE INSERT on memberTeam

**Business Rule Enforced:**

A club member should not be assigned to two teams with overlapping active periods.

**Benefit:**

- Ensures exclusive team participation per member per period.
- Avoids logistical and coaching conflicts.

### **4. check capacity before update**

**Purpose:**

Prevents updating a location's capacity to a value lower than its current active member count.

**Trigger Type:**

BEFORE UPDATE on Location

**Business Rule Enforced:**

New capacity must not be less than the current number of active members assigned to the location.

**Benefit:**

- Maintains physical and regulatory constraints.
- Ensures realistic adjustments to location capacity.

## QUERY 21

### **1) Preventing Player Time Conflicts:**

A player should not be assigned to two different team formation sessions that take place on the **same day** and **less than three hours apart**.

**Demonstration:**

**Insert this in your sql:**

-- Assume PlayerID = 101 is already assigned to a session at 10:00 AM on 2025-08-08.

-- This insert will be rejected if the new session is at 11:30 AM on the same day.

```
INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID)
VALUES (2002, 2, 101, 3);
```

## RESULTS

Action	Output	Time	Action	Response	Duration / Fetch Time
			300 10:00:29 INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID) VALUES (2003, 2, 101, 4)	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails	0.000 sec
		386 16:38:25	INSERT INTO Session (SessionID, SessionDate, SessionTime, SessionType) VALUES (3002, '2025-08-08', '12:45:00', 'Game')	1 row(s) affected	0.018 sec
		387 16:38:43	INSERT INTO TeamFormation (FormationID, SessionID, TeamID, Team2ID) VALUES (2003, 3002, 1, 2)	1 row(s) affected	0.0054 sec
		388 16:39:13	INSERT INTO TeamFormation (FormationID, SessionID, TeamID, Team2ID) VALUES (2003, 3002, 1, 2)	Error Code: 1062. Duplicate entry '2003' for key 'team...'	0.0021 sec
		389 16:40:33	INSERT INTO TeamFormation (FormationID, SessionID, TeamID, Team2ID) VALUES (2004, 3002, 1, 2)	1 row(s) affected	0.0053 sec
		390 16:40:51	INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID) VALUES (2004, 2, 101, 2)	Error Code: 1644. Player assignment rejected: Player...	0.0085 sec

### **2) Enforcing Minimum Age for Registration**

**New members must be at least 11 years old when registering.**

**Test case:**

```
INSERT INTO Person (
```

PersonID, FirstName, LastName, DateOfBirth, PhoneNo, Email, Address, City, Province,  
PostalCode, SSN

```
) VALUES (
```

1201, 'Young', 'Player', '2015-01-01', '514-555-1212', 'young@example.com',

'999 Young St', 'Montreal', 'QC', 'H3Z2Y7', 'YNG001'

);

INSERT INTO ClubMember (ClubMemberID, PaymentStatus, isMinor)

VALUES (1201, TRUE, TRUE);

Action Output	Time	Action	Response	Duration / Fetch Time
	300 16:39:13	INSERT INTO TeamFormation (FormationID, SessionID, TeamID) VALUES (2003, 3002, 1, 2)	Error Code: 1002. Duplicate entry '2003' for key 'idx...	0.0021 sec
	389 16:40:33	INSERT INTO TeamFormation (FormationID, SessionID, TeamID, Team2ID) VALUES (2004, 3002, 1, 2)	1 row(s) affected	0.0053 sec
	390 16:40:51	INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID) VALUES (2004, 2, 101, 2)	Error Code: 1644. Player assignment rejected: Player...	0.0085 sec
	391 16:42:31	INSERT INTO Person (PersonID, FirstName, ..., DateOfBirth, SSN) VALUES (1201, 'Young', ..., '2015-01-01', 'YNG001')	Error Code: 1064. You have an error in your SQL syntax...	0.0022 sec
	392 16:43:53	INSERT INTO Person (PersonID, FirstName, LastName, DateOfBirth, PhoneNo, Email, Address, City, Province, PostalCode, SSN) VALUES (1201, 'Young', 'Player', '20...', 1 row(s) affected)	0.01 sec	
	393 16:44:06	INSERT INTO ClubMember (ClubMemberID, PaymentStatus, isMinor) VALUES (1201, TRUE, TRUE)	Error Code: 1644. Club member must be at least 11 ye...	0.0050 sec

### 3) Preventing Overlapping Team Memberships

A club member should not be assigned to multiple teams with overlapping membership periods.

**Test:**

INSERT INTO memberTeam (TeamID, ClubMemberID, StartDate, EndDate)

VALUES (2, 1001, '2025-06-01', '2025-09-01');

-- Now, try adding another overlapping team assignment for the same member:

INSERT INTO memberTeam (TeamID, ClubMemberID, StartDate, EndDate)

VALUES (3, 1001, '2025-07-15', '2025-10-01');

Action Output	Time	Action	Response	Duration / Fetch Time
	300 16:40:51	INSERT INTO TeamFormationPlayer (FormationID, TeamID, ClubMemberID, PositionID) VALUES (2004, 2, 101, 2)	Error Code: 1644. Player assignment rejected: Player...	0.0085 sec
	391 16:42:31	INSERT INTO Person (PersonID, FirstName, ..., DateOfBirth, SSN) VALUES (1201, 'Young', ..., '2015-01-01', 'YNG001')	Error Code: 1064. You have an error in your SQL syntax...	0.0022 sec
	392 16:43:53	INSERT INTO Person (PersonID, FirstName, LastName, DateOfBirth, PhoneNo, Email, Address, City, Province, PostalCode, SSN) VALUES (1201, 'Young', 'Player', '20...', 1 row(s) affected)	0.01 sec	
	393 16:44:06	INSERT INTO ClubMember (ClubMemberID, PaymentStatus, isMinor) VALUES (1201, TRUE, TRUE)	Error Code: 1644. Club member must be at least 11 ye...	0.0050 sec
	394 16:47:26	INSERT INTO memberTeam (TeamID, ClubMemberID, StartDate, EndDate) VALUES (2, 1001, '2025-06-01', '2025-09-01')	1 row(s) affected	0.051 sec
	395 16:48:26	INSERT INTO memberTeam (TeamID, ClubMemberID, StartDate, EndDate) VALUES (3, 1001, '2025-07-15', '2025-10-01')	Error Code: 1644. Club member already has an active...	0.0055 sec

#### **4. Preventing Capacity Overflow**

**Location capacity should not be updated to a value less than its current number of active members.**

LogID	EmailDate	SenderLocation	ReceiverEmail	Subject	BodyPreview
8	HULL	HULL	james.moore@mvc.com	Upcoming Practice Session – Reminder	Dear James, you have an upcoming practice se...
7	HULL	HULL	lisa.brown@mvc.com	Upcoming Practice Session – Reminder	Dear Lisa, you have an upcoming practice sessi...
6	HULL	HULL	mike.williams@mvc.com	Upcoming Practice Session – Reminder	Dear Mike, you have an upcoming practice sess...
5	HULL	HULL	example@domain.com	Welcome to the MVC Club!	Thank you for registering. Your membership is n...
4	2025-01-14 09:00:00	Montreal Volleyball Club - Head Office	emma.thompson@email.com	Montreal Eagles Female Wednesday 15-Jan-20...	Dear Emma Thompson, You are scheduled to pl...
3	2025-01-13 09:00:00	Laval Branch	sophie.miller@email.com	Laval Tigers Female Tuesday 14-Jan-2025 7:30...	Dear Sophie Miller, You are scheduled to play a...
2	2025-01-12 09:05:00	Montreal Volleyball Club - Head Office	lucas.garcia@email.com	Montreal Eagles Male Monday 13-Jan-2025 6:0...	Dear Lucas Garcia, You are scheduled to play a...
1	2025-01-12 09:00:00	Montreal Volleyball Club - Head Office	alex.anderson@email.com	Montreal Eagles Male Monday 13-Jan-2025 6:0...	Dear Alex Anderson, You are scheduled to play...
	HULL	HULL	HULL	HULL	HULL

#### **Query 22**

INSERT INTO EmailLog (ReceiverEmail, Subject, BodyPreview)

VALUES (

'example@domain.com',

"Welcome to the MVC Club!",

'Thank you for registering. Your membership is now active.'

);

-- notifying about event

INSERT INTO EmailLog (ReceiverEmail, Subject, BodyPreview)

SELECT

p.Email,

'Upcoming Practice Session – Reminder',

CONCAT('Dear ', p.FirstName, ', you have an upcoming practice session scheduled.  
Please check the portal for details.')

FROM

Personnel per

JOIN

Person p ON p.PersonID = per.PersonnelID

WHERE

per.Role = 'Coach';

SELECT \* FROM EmailLog ORDER BY LogID DESC;

LogID	EmailDate	SenderLocation	ReceiverEmail	Subject	BodyPreview
8	NULL	NULL	james.moore@mvc.com	Upcoming Practice Session – Reminder	Dear James, you have an upcoming practice se...
7	NULL	NULL	lisa.brown@mvc.com	Upcoming Practice Session – Reminder	Dear Lisa, you have an upcoming practice sessi...
6	NULL	NULL	mike.williams@mvc.com	Upcoming Practice Session – Reminder	Dear Mike, you have an upcoming practice sess...
5	NULL	NULL	example@domain.com	Welcome to the MVC Club!	Thank you for registering. Your membership is n...
4	2025-01-14 09:00:00	Montreal Volleyball Club - Head Office	emma.thompson@email.com	Montreal Eagles Female Wednesday 15-Jan-20...	Dear Emma Thompson, You are scheduled to pl...
3	2025-01-13 09:00:00	Laval Branch	sophie.miller@email.com	Laval Tigers Female Tuesday 14-Jan-2025 7:30...	Dear Sophie Miller, You are scheduled to play a...
2	2025-01-12 09:05:00	Montreal Volleyball Club - Head Office	lucas.garcia@email.com	Montreal Eagles Male Monday 13-Jan-2025 6:0...	Dear Lucas Garcia, You are scheduled to play a...
1	2025-01-12 09:00:00	Montreal Volleyball Club - Head Office	alex.anderson@email.com	Montreal Eagles Male Monday 13-Jan-2025 6:0...	Dear Alex Anderson, You are scheduled to play...
NULL	NULL	NULL	NULL	NULL	NULL

## GUI

Main page/Dashboard:

Montréal Volleyball Club (MVC)

Connected as: ntc353\_1

Dashboard View Tables Query 8 Query 21 Query 22 Logout

Montréal Volleyball Club (MVC) - Dashboard Overview

Welcome to the MVC Database Management System

Total Members: 15 Active Teams: 9 Club Locations: 7 Upcoming Sessions: 1

**Member Statistics**

Minor Members (Under 18):	7
Major Members (18+):	8
Paid Members:	11
Payment Pending:	4

**Club Activity**

Total Personnel:	9
Active Coaches:	6
Recent Payments:	\$0.00
Total Revenue (2024):	\$1280.00

**Recent Member Registrations**

Name	Age	Location	Payment Status
Test Player	25	Montreal Volleyball Club - Head Office	✓ Paid
Tommy Nguyen	12	None	✓ Paid
Trigger Tester	20	None	✓ Paid
Mia Davis	11	Montreal Volleyball Club - Head Office	⚠ Pending
Logan Brown	15	Laval Branch	✓ Paid

Quick Actions:

Manage Tables Location Report Test Triggers Email Logs

View tables page:

Montréal Volleyball Club (MVC)

Connected as: ntc353\_1

Dashboard View Tables Query 8 Query 21 Query 22 Logout

**Database Tables**

Table Name	Actions
ClubMember	<a href="#">View Data</a> <a href="#">Create</a>
ClubMemberLocation	<a href="#">View Data</a> <a href="#">Create</a>
ClubMember_Hobby	<a href="#">View Data</a> <a href="#">Create</a>
EmailLog	<a href="#">View Data</a> <a href="#">Create</a>
FamilyMember	<a href="#">View Data</a> <a href="#">Create</a>
FamilyMemberLocation	<a href="#">View Data</a> <a href="#">Create</a>
Hobby	<a href="#">View Data</a> <a href="#">Create</a>
Location	<a href="#">View Data</a> <a href="#">Create</a>
Payment	<a href="#">View Data</a> <a href="#">Create</a>
Person	<a href="#">View Data</a> <a href="#">Create</a>
Personnel	<a href="#">View Data</a> <a href="#">Create</a>
Personnel_Location	<a href="#">View Data</a> <a href="#">Create</a>
PlayerPosition	<a href="#">View Data</a> <a href="#">Create</a>
Relationship	<a href="#">View Data</a> <a href="#">Create</a>
SecondaryFamilyMember	<a href="#">View Data</a> <a href="#">Create</a>
SecondaryFamilyRelationship	<a href="#">View Data</a> <a href="#">Create</a>
Session	<a href="#">View Data</a> <a href="#">Create</a>
Team	<a href="#">View Data</a> <a href="#">Create</a>
TeamFormation	<a href="#">View Data</a> <a href="#">Create</a>
TeamFormationPlayer	<a href="#">View Data</a> <a href="#">Create</a>
memberTeam	<a href="#">View Data</a> <a href="#">Create</a>
minorFamilyHistory	<a href="#">View Data</a> <a href="#">Create</a>

## Location page:

**Montréal Volleyball Club (MVC)** Connected as: ntc353\_1

[Dashboard](#) [View Tables](#) [Query 8](#) [Query 21](#) [Query 22](#) [Logout](#)

**Table: Location**

[Add New Record](#)

locationID	name	Type	Address	City	Province	PostalCode	PhoneNo	WebAddress	Capacity	Actions
1	Montreal Volleyball Club - Head Office	Head	1000 Sherbrooke St W	Montreal	QC	H3A1G1	514-555-1000	www.mvc-montreal.com	30	<a href="#">Edit</a> <a href="#">Delete</a>
2	Laval Branch	Branch	2000 Boulevard Chomedey	Laval	QC	H7T2Z5	450-555-2000	www.mvc-laval.com	100	<a href="#">Edit</a> <a href="#">Delete</a>
3	West Island Branch	Branch	3000 Brunswick Blvd	Dollard-des-Ormeaux	QC	H9B1N7	514-555-3000	www.mvc-westisland.com	80	<a href="#">Edit</a> <a href="#">Delete</a>
5	South Shore Complex	Branch	5001 Taschereau Blvd	Brossard	QC	J4Z1B3	450-555-6001	www.mvc-southshore.com/a	90	<a href="#">Edit</a> <a href="#">Delete</a>
6	South Shore Complex	Branch	1200 Rue Saint-Charles	Longueuil	QC	J4K2T1	450-555-6002	www.mvc-southshore.com/b	90	<a href="#">Edit</a> <a href="#">Delete</a>
7	South Shore Complex	Branch	2100 Rue Victoria	Saint-Lambert	QC	J4P2K1	450-555-6003	www.mvc-southshore.com/c	90	<a href="#">Edit</a> <a href="#">Delete</a>
8	South Shore Complex	Branch	800 Rue de Montarville	Boucherville	QC	J4B1C3	450-555-6004	www.mvc-southshore.com/d	90	<a href="#">Edit</a> <a href="#">Delete</a>

7 rows displayed

## Add new location page:

**Montréal Volleyball Club (MVC)** Connected as: ntc353\_1

[Dashboard](#) [View Tables](#) [Query 8](#) [Query 21](#) [Query 22](#) [Logout](#)

**Create New Record in Location**

**locationID (Primary Key)**  
  
Type: int | auto\_increment

**name**  
  
Type: varchar(100)

**Type**  
  
Type: enum('Head', 'Branch')

**Address**  
  
Type: varchar(200)

**City**  
  
Type: varchar(50)

**Province**  
  
Type: varchar(50)

**PostalCode**  
  
Type: varchar(10)

**PhoneNo**  
  
Type: varchar(20)

**WebAddress**  
  
Type: varchar(100)

**Capacity**  
  
Type: int

[Create Record](#) [Cancel](#)

## Query 8 - Location Summary Report page:

**Montréal Volleyball Club (MVC)**

Connected as: ntc353\_1

Dashboard View Tables Query 8 Query 21 Query 22 Logout

Query 8 executed successfully! Found 7 locations.

**Query 8 - Location Summary Report**

This query provides a comprehensive overview of all locations including general managers, member counts, and team information.

Location ID	Location Name	Address	City	Province	Postal Code	Phone	Website	Type	Capacity	General Manager	Minor Members	Major Members	Team Count
8	South Shore Complex	800 Rue de Montarville	Boucherville	QC	J4B1C3	450-555-6004	www.mvc-southshore.com/d	Branch	90	No Manager Assigned	0	0	0
5	South Shore Complex	5001 Taschereau Blvd	Brossard	QC	J4Z1B3	450-555-6001	www.mvc-southshore.com/a	Branch	90	No Manager Assigned	0	0	0
3	West Island Branch	3000 Brunswick Blvd	Dollard-des-Ormeaux	QC	H9B1N7	514-555-3000	www.mvc-westisland.com	Branch	80	No Manager Assigned	0	0	1
2	Laval Branch	2000 Boulevard Chomedey	Laval	QC	H7T2Z5	450-555-2000	www.mvc-laval.com	Branch	100	No Manager Assigned	2	2	2
6	South Shore Complex	1200 Rue Saint-Charles	Longueuil	QC	J4K2T1	450-555-6002	www.mvc-southshore.com/b	Branch	90	No Manager Assigned	0	0	0
1	Montreal Volleyball Club - Head Office	1000 Sherbrooke St W	Montreal	QC	H3A1G1	514-555-1000	www.mvc-montreal.com	Head	30	John Smith	4	5	6
7	South Shore Complex	2100 Rue Victoria	Saint-Lambert	QC	J4P2K1	450-555-6003	www.mvc-southshore.com/c	Branch	90	No Manager Assigned	0	0	0

7 location(s) found

Execute Query 8

## Query 21 - Trigger Testing Operations page:

**Montréal Volleyball Club (MVC)**

Connected as: ntc353\_1

Dashboard View Tables Query 8 Query 21 Query 22 Logout

Test 3 execution completed

**Query 21 - Trigger Testing Operations**

This section tests various database triggers and constraints including time conflicts, age requirements, team memberships, and capacity limits.

**Available Test Operations:**

- Test 1: Team Formation Player Assignment**  
Tests player time conflict prevention trigger
- Test 2: Add Young Player (Under 11)**  
Tests minimum age requirement trigger
- Test 3: Overlapping Team Membership**  
Tests unique active team membership trigger
- Test 4: Capacity Limit Update**  
Tests location capacity constraint trigger

**Test Results:**

**Test 3: Overlapping Team Membership**

Correctly blocked: 1644 (45000): Club member already has an active team membership in that period.  
Unique active team membership trigger working properly

## Query 22 - Email Log Operations page:

**Montréal Volleyball Club (MVC)**

Connected as: ntc353\_1

[Dashboard](#) [View Tables](#) [Query 8](#) [Query 21](#) **Query 22** [Logout](#)

Recent email logs retrieved

**Query 22 - Email Log Operations**

This section manages email logging functionality including welcome messages and coach notifications.

**Available Email Operations:**

- [Send Welcome Email](#)  
Log a welcome email for new members
- [Notify All Coaches](#)  
Send practice session reminders to all coaches
- [View Recent Email Logs](#)  
Display the most recent email log entries

Log ID	Date/Time	Sender Location	Receiver Email	Subject	Body Preview
10	2025-08-09 22:20:44	System	lisa.brown@mvc.com	Upcoming Practice Session – Reminder	Dear Lisa, you have an upcoming practice session scheduled. Please check the portal for details.
9	2025-08-09 22:20:44	System	mike.williams@mvc.com	Upcoming Practice Session – Reminder	Dear Mike, you have an upcoming practice session scheduled. Please check the portal for details.
8	2025-08-09 22:20:38	System	james.moore@mvc.com	Upcoming Practice Session – Reminder	Dear James, you have an upcoming practice session scheduled. Please check the portal for details.
7	2025-08-09 22:20:38	System	lisa.brown@mvc.com	Upcoming Practice Session – Reminder	Dear Lisa, you have an upcoming practice session scheduled. Please check the portal for details.
6	2025-08-09 22:20:38	System	mike.williams@mvc.com	Upcoming Practice Session – Reminder	Dear Mike, you have an upcoming practice session scheduled. Please check the portal for details.
5	2025-08-09 22:20:35	System	example@domain.com	Welcome to the MVC Club!	Thank you for registering. Your membership is now active.
4	2025-01-14 09:00:00	Montreal Volleyball Club - Head Office	emma.thompson@email.com	Montreal Eagles Female Wednesday 15-Jan-2025 8:00 PM game session	Dear Emma Thompson, You are scheduled to play as Middle Blocker in the upcoming game session...
3	2025-01-13 09:00:00	Laval Branch	sophie.miller@email.com	Laval Tigers Female Tuesday 14-Jan-2025 7:30 PM training session	Dear Sophie Miller, You are scheduled to play as Libero in the upcoming training session...
2	2025-01-12 09:05:00	Montreal Volleyball Club - Head Office	lucas.garcia@email.com	Montreal Eagles Male Monday 13-Jan-2025 6:00 PM training session	Dear Lucas Garcia, You are scheduled to play as Setter in the upcoming training session...
1	2025-01-12 09:00:00	Montreal Volleyball Club - Head Office	alex.anderson@email.com	Montreal Eagles Male Monday 13-Jan-2025 6:00 PM training session	Dear Alex Anderson, You are scheduled to play as Outside Hitter in the upcoming training session...

10 email log(s) displayed

