COMP353

Databases

Project Report

Written By:

Team ID: luc353_1

Patrick Han: 40112063

Harwinder Mann: 40293187

Tonny Zhao: 40283194

Nicolas Sorescu: 40312759

Due Date: 20/07/2025

“We certify that this submission is our original work and meets the Faculty’s Expectations of Originality.”

Faculty of Engineering and Computer Science
Expectations of Originality

This form sets out the requirements for originality for work submitted by students in the Faculty of Engineering and Computer Science. Submissions such as assignments, lab reports, project reports, computer programs and take-home exams must conform to the requirements stated on this form and to the Academic Code of Conduct. The course outline may stipulate additional requirements for the course.

1. Your submissions must be your own original work. Group submissions must be the original work of the students in the group.
2. Direct quotations must not exceed 5% of the content of a report, must be enclosed in quotation marks, and must be attributed to the source by a numerical reference citation[1]. Note that engineering reports rarely contain direct quotations.
3. Material paraphrased or taken from a source must be attributed to the source by a numerical reference citation.
4. Text that is inserted from a web site must be enclosed in quotation marks and attributed to the web site by numerical reference citation.
5. Drawings, diagrams, photos, maps or other visual material taken from a source must be attributed to that source by a numerical reference citation.
6. No part of any assignment, lab report or project report submitted for this course can be submitted for any other course.
7. In preparing your submissions, the work of other past or present students cannot be consulted, used, copied, paraphrased or relied upon in any manner whatsoever.
8. Your submissions must consist entirely of your own or your group's ideas, observations, calculations, information and conclusions, except for statements attributed to sources by numerical citation.
9. Your submissions cannot be edited or revised by any other student.
10. For lab reports, the data must be obtained from your own or your lab group's experimental work.
11. For software, the code must be composed by you or by the group submitting the work, except for code that is attributed to its sources by numerical reference.

You must write one of the following statements on each piece of work that you submit:
For individual work: **"I certify that this submission is my original work and meets the Faculty's Expectations of Originality",** with your signature, I.D. #, and the date.
For group work: **"We certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality",** with the signatures and I.D. #s of all the team members and the date.

A signed copy of this form must be submitted to the instructor at the beginning of the semester in each course.

I certify that I have read the requirements set out on this form, and that I am aware of these requirements. I certify that all the work I will submit for this course will comply with these requirements and with additional requirements stated in the course outline.

Course Number: COMP 353 Instructor: Prof. Khaled Jababo
Name: Tony, Patrick, Nicolas, Harwinder I.D. # 40293187, 40112063, 40283194, 40312759
Signature: H.M, N.S, P.H, T.Z Date: July 20, 2025

[1] Rules for reference citation can be found in "Form and Style" by Patrich MacDonagh and Jack Bordan, fourth edition, May, 2000, available at http://www.encs.concordia.ca/scs/Forms/Form&Style.pdf.
Approved by the ENCS Faculty Council February 10, 2012

Table of Content

Introduction

This project focuses on expertly administering a sports club, which requires monitoring of all aspects, which includes member details and financial records. The objective of our project is to enhance record management and optimize operations for the Montréal Youth Volleyball Club (MYVC) through the creation of an organized database system. The database enhances efficiency, precision, and availability by consistently organizing data regarding club members, staff, venues, and transactions.

Project Objectives

- The database manages important administrative duties, such as:
- Member Management: Maintaining membership status, contact information, names.
- Family Records: Monitoring relatives and their connections to participants.
- Staff Allocation – Organizing staff information and assigned locations.
- Payment Tracking – Maintaining accurate membership payment records.

Development Process

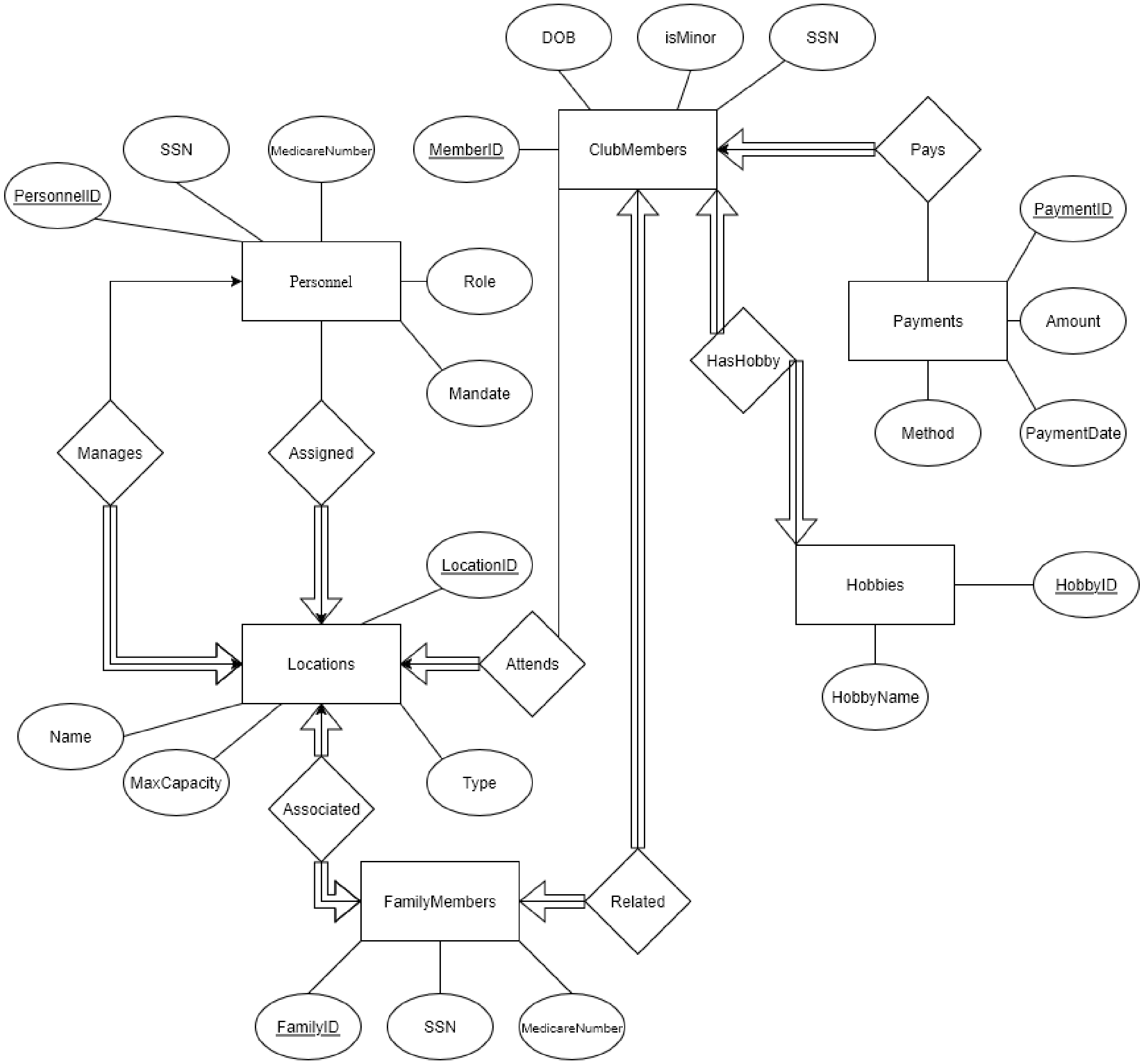
- Designed an ER diagram to visualize relationships.
- Defined relational tables with appropriate primary and foreign keys.
- Implemented the database in MySQL using SQL commands.
- Tested the system by running queries to verify functionality.

Technologies Used

- MySQL 8.0.22 – For data storage and management.
- MySQL Workbench – To execute queries and interact with the database.

Our system offers MYVC a centralized platform for handling critical information, decreasing administrative burdens, and helping logical operations. By rule of careful preparation and execution, we created a flexible and effective solution customized to the club’s requirements.

Conceptual DB Design (E/R)



E/R to Relations Conversion/DDL Statements

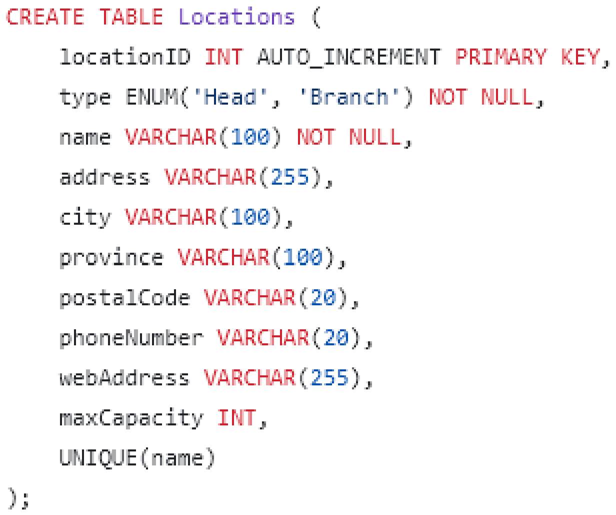
Locations

Location(locationID, type, name, address, city, province, postalCode, phoneNumber, webAddress, maxCapacity)

Primary Key: location

Functional Dependencies:

F = { locationID ➔ type, name, address, city, province, postalCode, phoneNumber, webAddress, maxCapacity }



```
CREATE TABLE Locations (
    locationID INT AUTO_INCREMENT PRIMARY KEY,
    type ENUM('Head', 'Branch') NOT NULL,
    name VARCHAR(100) NOT NULL,
    address VARCHAR(255),
    city VARCHAR(100),
    province VARCHAR(100),
    postalCode VARCHAR(20),
    phoneNumber VARCHAR(20),
    webAddress VARCHAR(255),
    maxCapacity INT,
    UNIQUE(name)
);
```

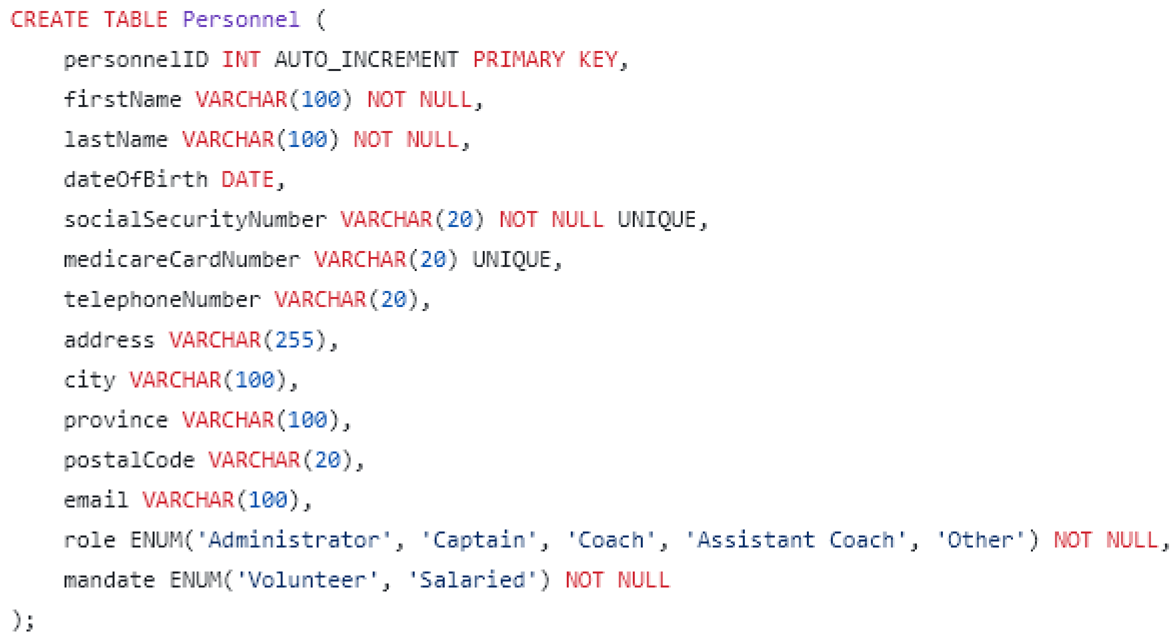
Personnel

Personnel(personnelID, firstName, lastName, dateOfBirth, socialSecurityNumber, medicareCardNumber, telephoneNumber, address, city, province, postalCode, email, role, mandate)

Primary Key: personnelID

Functional Dependencies:

F = { personnelID ➔ firstName, lastName, dateOfBirth, socialSecurityNumber, medicareCardNumber, telephoneNumber, address, city, province, postalCode, email, role, mandate }



```
CREATE TABLE Personnel (
    personnelID INT AUTO_INCREMENT PRIMARY KEY,
    firstName VARCHAR(100) NOT NULL,
    lastName VARCHAR(100) NOT NULL,
    dateOfBirth DATE,
    socialSecurityNumber VARCHAR(20) NOT NULL UNIQUE,
    medicareCardNumber VARCHAR(20) UNIQUE,
    telephoneNumber VARCHAR(20),
    address VARCHAR(255),
    city VARCHAR(100),
    province VARCHAR(100),
    postalCode VARCHAR(20),
    email VARCHAR(100),
    role ENUM('Administrator', 'Captain', 'Coach', 'Assistant Coach', 'Other') NOT NULL,
    mandate ENUM('Volunteer', 'Salaried') NOT NULL
);
```

PersonnelLocation

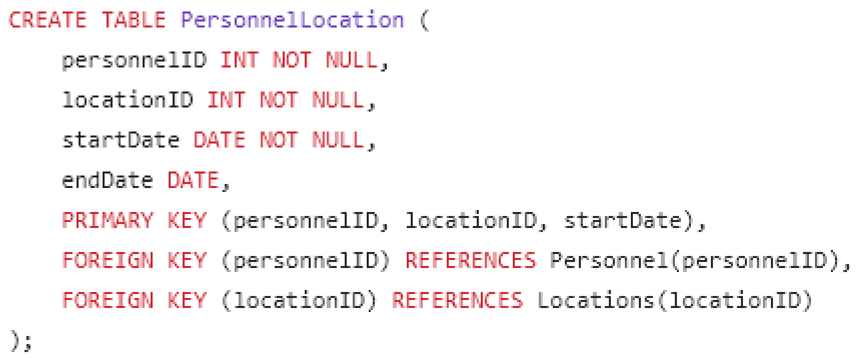
PersonnelLocation(personnelID, locationID, startDate, endDate)

Primary Key: (startDate, personnelID, locationID)

Foreign Key: personnelID, locationID

Functional Dependencies:

F = { personnelID, locationID, startDate) ➔ endDate }



```
CREATE TABLE PersonnelLocation (
    personnelID INT NOT NULL,
    locationID INT NOT NULL,
    startDate DATE NOT NULL,
    endDate DATE,
    PRIMARY KEY (personnelID, locationID, startDate),
    FOREIGN KEY (personnelID) REFERENCES Personnel(personnelID),
    FOREIGN KEY (locationID) REFERENCES Locations(locationID)
);
```

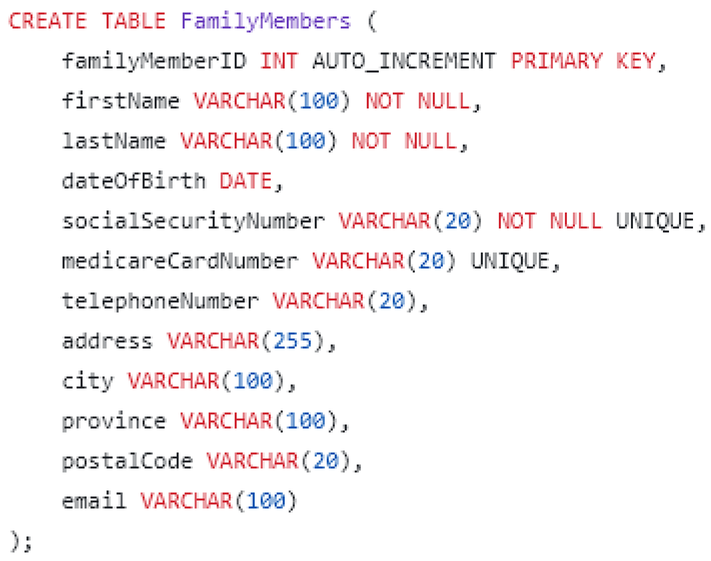
FamilyMembers

FamilyMembers(familyMemberID, firstName, lastName, dateOfBirth, socialSecurityNumber, medicareCardNumber, telephoneNumber, address, city, province, postalCode, email)

Primary Key: familyMemberID

Functional Dependencies:

F = { familyMemberID ➔ firstName, lastName, dateOfBirth, socialSecurityNumber, medicareCardNumber, telephoneNumber, address, city, province, postalCode, email }



```
CREATE TABLE FamilyMembers (
    familyMemberID INT AUTO_INCREMENT PRIMARY KEY,
    firstName VARCHAR(100) NOT NULL,
    lastName VARCHAR(100) NOT NULL,
    dateOfBirth DATE,
    socialSecurityNumber VARCHAR(20) NOT NULL UNIQUE,
    medicareCardNumber VARCHAR(20) UNIQUE,
    telephoneNumber VARCHAR(20),
    address VARCHAR(255),
    city VARCHAR(100),
    province VARCHAR(100),
    postalCode VARCHAR(20),
    email VARCHAR(100)
);
```

FamilyMemberLocation

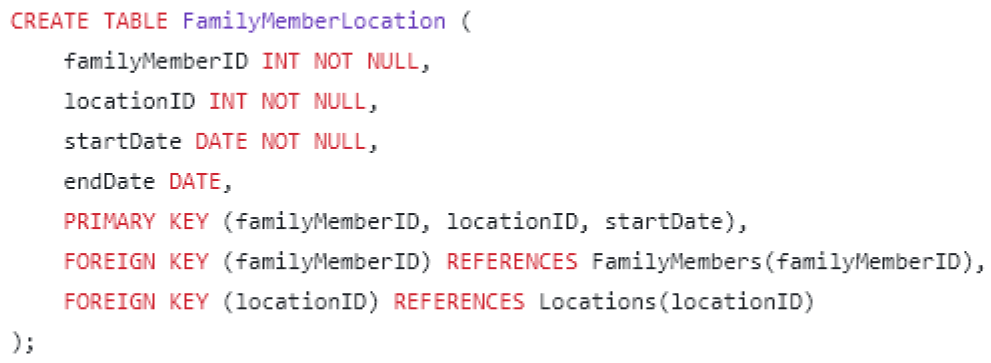
FamilyMemberLocation(familyMemberID, locationID, startDate, endDate)

Primary Key: (familyMemberID, locationID, startDate)

Foreign Key: familyMemberID, locationID

Functional Dependencies:

F = { familyMemberID, locationID, startDate ➔ endDate }



```
CREATE TABLE FamilyMemberLocation (
    familyMemberID INT NOT NULL,
    locationID INT NOT NULL,
    startDate DATE NOT NULL,
    endDate DATE,
    PRIMARY KEY (familyMemberID, locationID, startDate),
    FOREIGN KEY (familyMemberID) REFERENCES FamilyMembers(familyMemberID),
    FOREIGN KEY (locationID) REFERENCES Locations(locationID)
);
```

ClubMembers

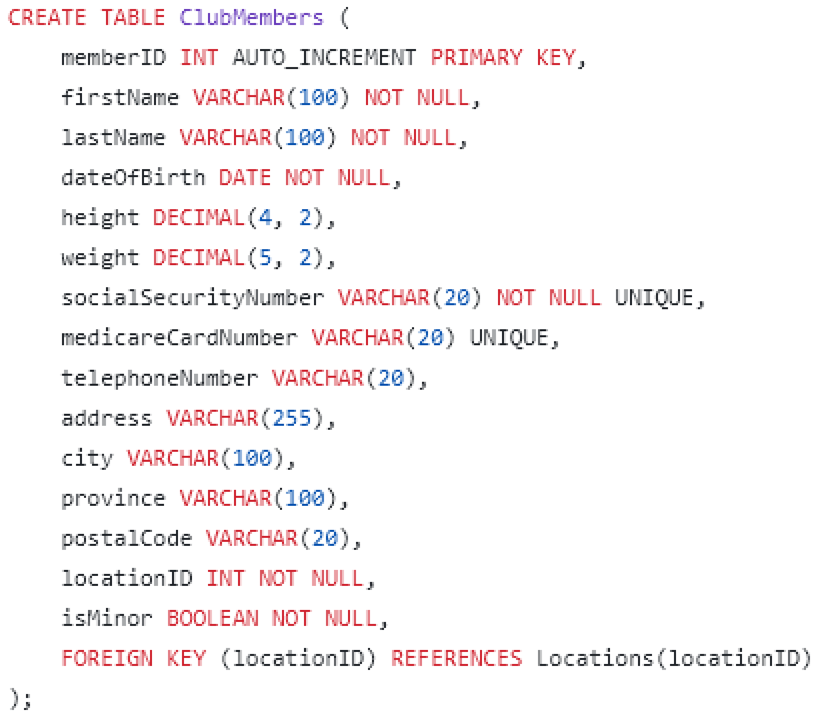
ClubMembers(memberID, firstName, lastName, dateOfBirth, height, weight, socialSecurityNumber, medicareCardNumber, telephoneNumber, address, city, province, postalCode, locationID, isMinor)

Primary Key: memberID

Foreign Key: locationID

Functional Dependencies:

F = { memberID ➔ firstName, lastName, dateOfBirth, height, weight, socialSecurityNumber, medicareCardNumber, telephoneNumber, address, city, province, postalCode, locationID, isMinor }



```
CREATE TABLE ClubMembers (
    memberID INT AUTO_INCREMENT PRIMARY KEY,
    firstName VARCHAR(100) NOT NULL,
    lastName VARCHAR(100) NOT NULL,
    dateOfBirth DATE NOT NULL,
    height DECIMAL(4, 2),
    weight DECIMAL(5, 2),
    socialSecurityNumber VARCHAR(20) NOT NULL UNIQUE,
    medicareCardNumber VARCHAR(20) UNIQUE,
    telephoneNumber VARCHAR(20),
    address VARCHAR(255),
    city VARCHAR(100),
    province VARCHAR(100),
    postalCode VARCHAR(20),
    locationID INT NOT NULL,
    isMinor BOOLEAN NOT NULL,
    FOREIGN KEY (locationID) REFERENCES Locations(locationID)
);
```

MemberFamilyRelations

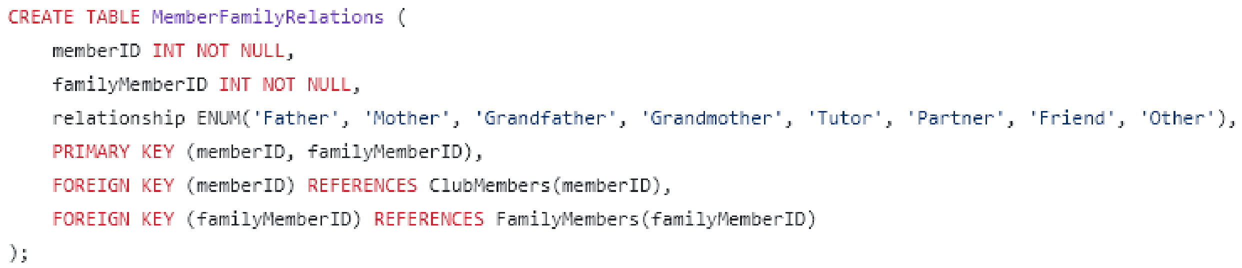
MemberFamilyRelations(memberID, familyMemberID, relationship)

Primary Key: (memberID, familyMemberID)

Foreign Key: memberID, familyMemberID

Functional Dependencies:

F = { (memberID, familyMemberID) ➔ relationship }



```
CREATE TABLE MemberFamilyRelations (
    memberID INT NOT NULL,
    familyMemberID INT NOT NULL,
    relationship ENUM('Father', 'Mother', 'Grandfather', 'Grandmother', 'Tutor', 'Partner', 'Friend', 'Other'),
    PRIMARY KEY (memberID, familyMemberID),
    FOREIGN KEY (memberID) REFERENCES ClubMembers(memberID),
    FOREIGN KEY (familyMemberID) REFERENCES FamilyMembers(familyMemberID)
);
```

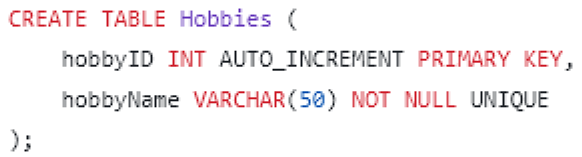
Hobbies

Hobbies(hobbyID, hobbyName)

Primary Key: Hobbies

Functional Dependencies:

F = { hobbyID ➔ hobbyName }



```
CREATE TABLE Hobbies (
    hobbyID INT AUTO_INCREMENT PRIMARY KEY,
    hobbyName VARCHAR(50) NOT NULL UNIQUE
);
```

ClubMemberHobbies

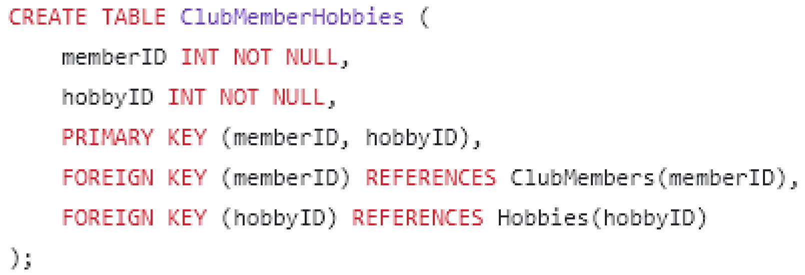
ClubMemberHobbies(memberID, hobbyID)

Primary Key: (memberID, hobbyID)

Foreign Key: memberID, hobbyID

Functional Dependencies:

F = { (memberID, hobbyID) ➔ ∅ }



```
CREATE TABLE ClubMemberHobbies (
    memberID INT NOT NULL,
    hobbyID INT NOT NULL,
    PRIMARY KEY (memberID, hobbyID),
    FOREIGN KEY (memberID) REFERENCES ClubMembers(memberID),
    FOREIGN KEY (hobbyID) REFERENCES Hobbies(hobbyID)
);
```

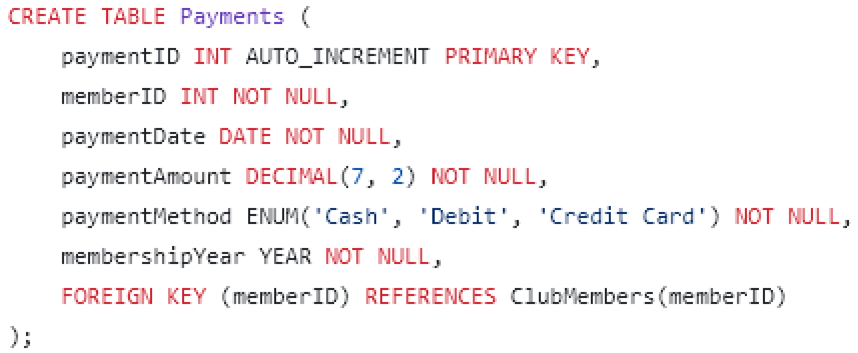
Payments

Payments(paymentID, memberID, paymentDate, paymentAmount, paymentMethod, membershipYear)

Primary Key: paymentID

Functional Dependencies:

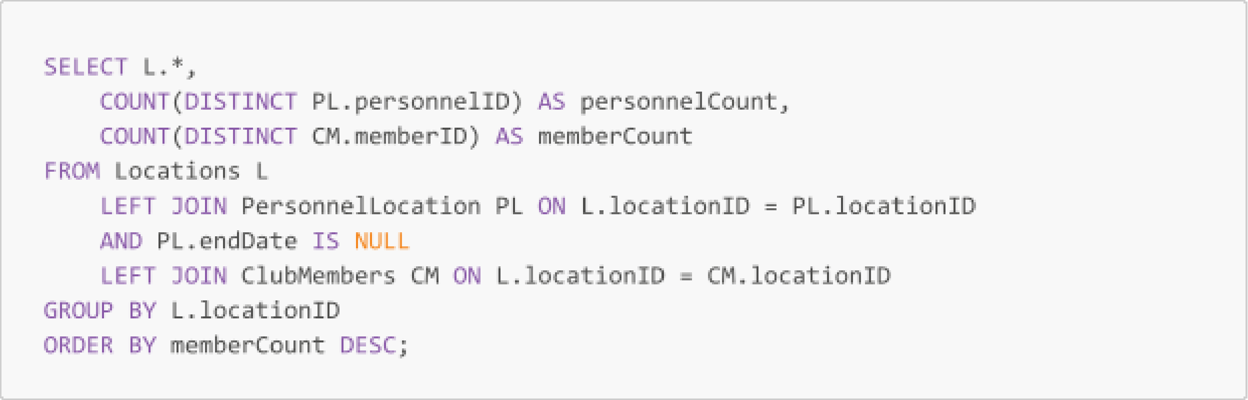
F = { paymentID ➔ memberID, paymentDate, paymentAmount, paymentMethod, membershipYear }



```sql
CREATE TABLE Payments (
    paymentID INT AUTO_INCREMENT PRIMARY KEY,
    memberID INT NOT NULL,
    paymentDate DATE NOT NULL,
    paymentAmount DECIMAL(7, 2) NOT NULL,
    paymentMethod ENUM('Cash', 'Debit', 'Credit Card') NOT NULL,
    membershipYear YEAR NOT NULL,
    FOREIGN KEY (memberID) REFERENCES ClubMembers(memberID)
);
```

DML Statement

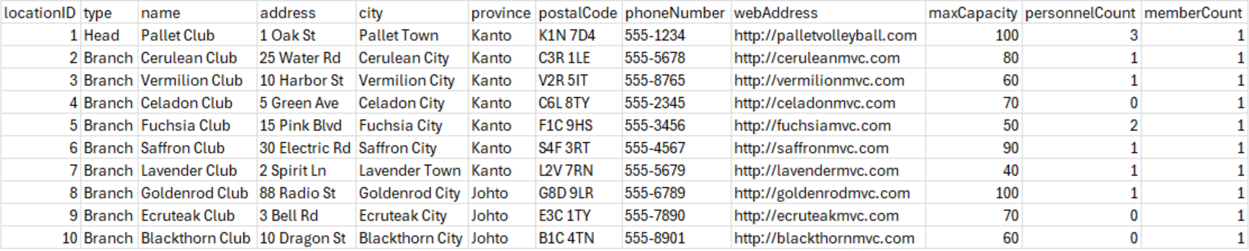
1. Location details with personnel and member counts



```
SELECT L.*,
    COUNT(DISTINCT PL.personnelID) AS personnelCount,
    COUNT(DISTINCT CM.memberID) AS memberCount
FROM Locations L
    LEFT JOIN PersonnelLocation PL ON L.locationID = PL.locationID
    AND PL.endDate IS NULL
    LEFT JOIN ClubMembers CM ON L.locationID = CM.locationID
GROUP BY L.locationID
ORDER BY memberCount DESC;
```

Expected Results:

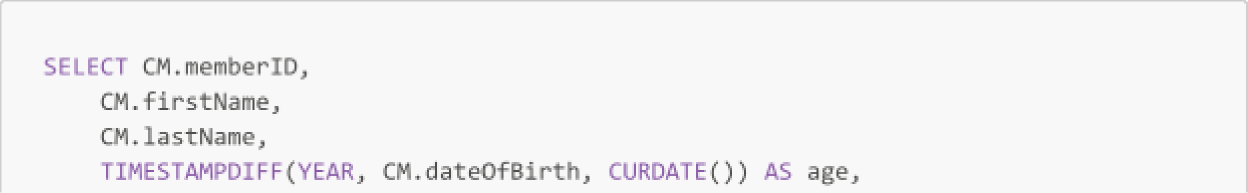
- Shows all 10 locations with their full details
- Pallet Club: 1 member, 3 personnel (Head office with Ash, Gary, Professor Oak)
- Each other club: 1 member, 1 personnel each
- Results ordered by member count (all locations have 1 member except Pallet Club with potentially multiple personnel)



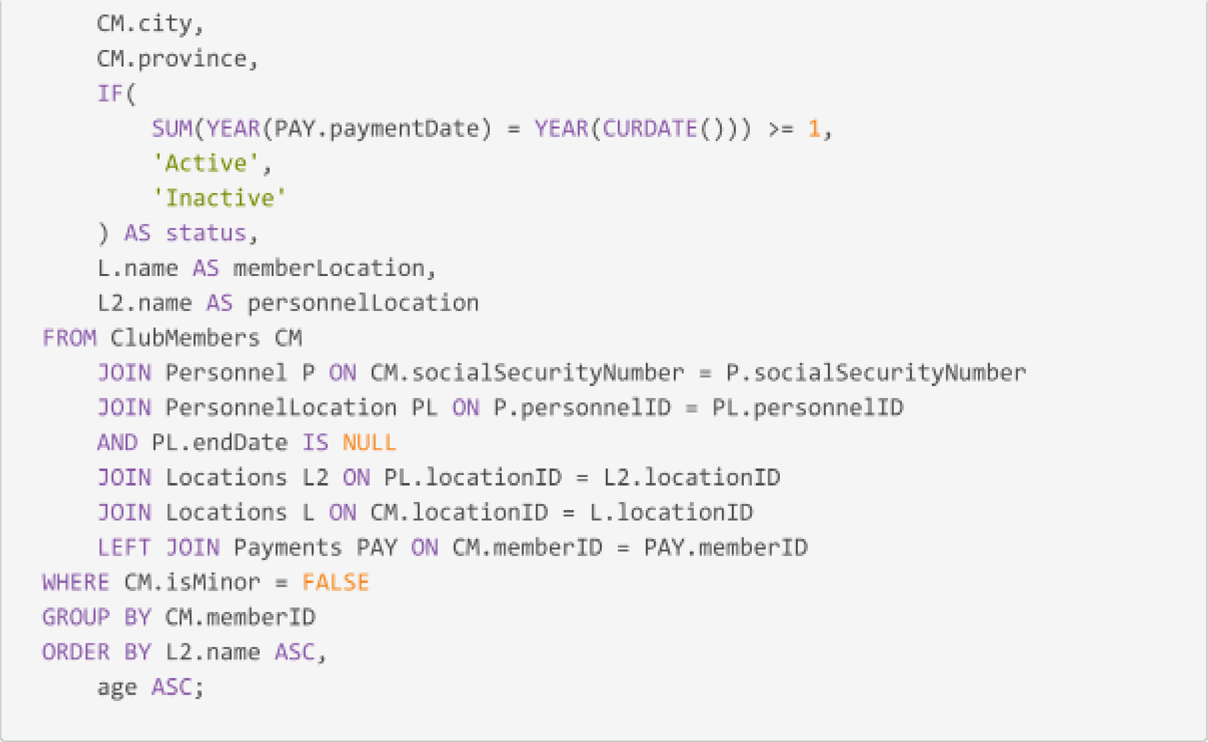
locationID	type	name	address	city	province	postalCode	phoneNumber	webAddress	maxCapacity	personnelCount	memberCount
1	Head	Pallet Club	1 Oak St	Pallet Town	Kanto	K1N 7D4	555-1234	http://palletvolleyball.com	100	3	1
2	Branch	Cerulean Club	25 Water Rd	Cerulean City	Kanto	C3R 1LE	555-5678	http://ceruleanmvc.com	80	1	1
3	Branch	Vermilion Club	10 Harbor St	Vermilion City	Kanto	V2R 5IT	555-8765	http://vermilionmvc.com	60	1	1
4	Branch	Celadon Club	5 Green Ave	Celadon City	Kanto	C6L 8TY	555-2345	http://celadonmvc.com	70	0	1
5	Branch	Fuchsia Club	15 Pink Blvd	Fuchsia City	Kanto	F1C 9HS	555-3456	http://fuchsiamvc.com	50	2	1
6	Branch	Saffron Club	30 Electric Rd	Saffron City	Kanto	S4F 3RT	555-4567	http://saffronmvc.com	90	1	1
7	Branch	Lavender Club	2 Spirit Ln	Lavender Town	Kanto	L2V 7RN	555-5679	http://lavendermvc.com	40	1	1
8	Branch	Goldenrod Club	88 Radio St	Goldenrod City	Johto	G8D 9LR	555-6789	http://goldenrodmvc.com	100	1	1
9	Branch	Ecruteak Club	3 Bell Rd	Ecruteak City	Johto	E3C 1TY	555-7890	http://ecruteakmvc.com	70	0	1
10	Branch	Blackthorn Club	10 Dragon St	Blackthorn City	Johto	B1C 4TN	555-8901	http://blackthornmvc.com	60	0	1

Figure 1: Result of Query #1

2. Major members who are also active personnel



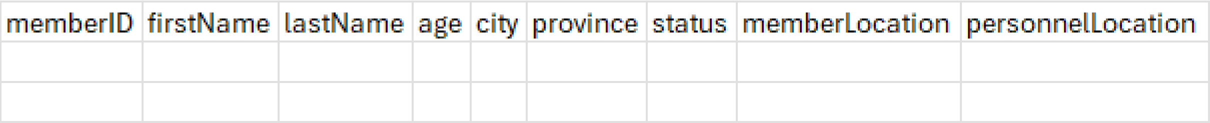
```
SELECT CM.memberID,
    CM.firstName,
    CM.lastName,
    TIMESTAMPDIFF(YEAR, CM.dateOfBirth, CURDATE()) AS age,
```



```
    CM.city,
    CM.province,
    IF(
        SUM(YEAR(PAY.paymentDate) = YEAR(CURDATE())) >= 1,
        'Active',
        'Inactive'
    ) AS status,
    L.name AS memberLocation,
    L2.name AS personnelLocation
FROM ClubMembers CM
    JOIN Personnel P ON CM.socialSecurityNumber = P.socialSecurityNumber
    JOIN PersonnelLocation PL ON P.personnelID = PL.personnelID
    AND PL.endDate IS NULL
    JOIN Locations L2 ON PL.locationID = L2.locationID
    JOIN Locations L ON CM.locationID = L.locationID
    LEFT JOIN Payments PAY ON CM.memberID = PAY.memberID
WHERE CM.isMinor = FALSE
GROUP BY CM.memberID
ORDER BY L2.name ASC,
    age ASC;
```

Expected Results:

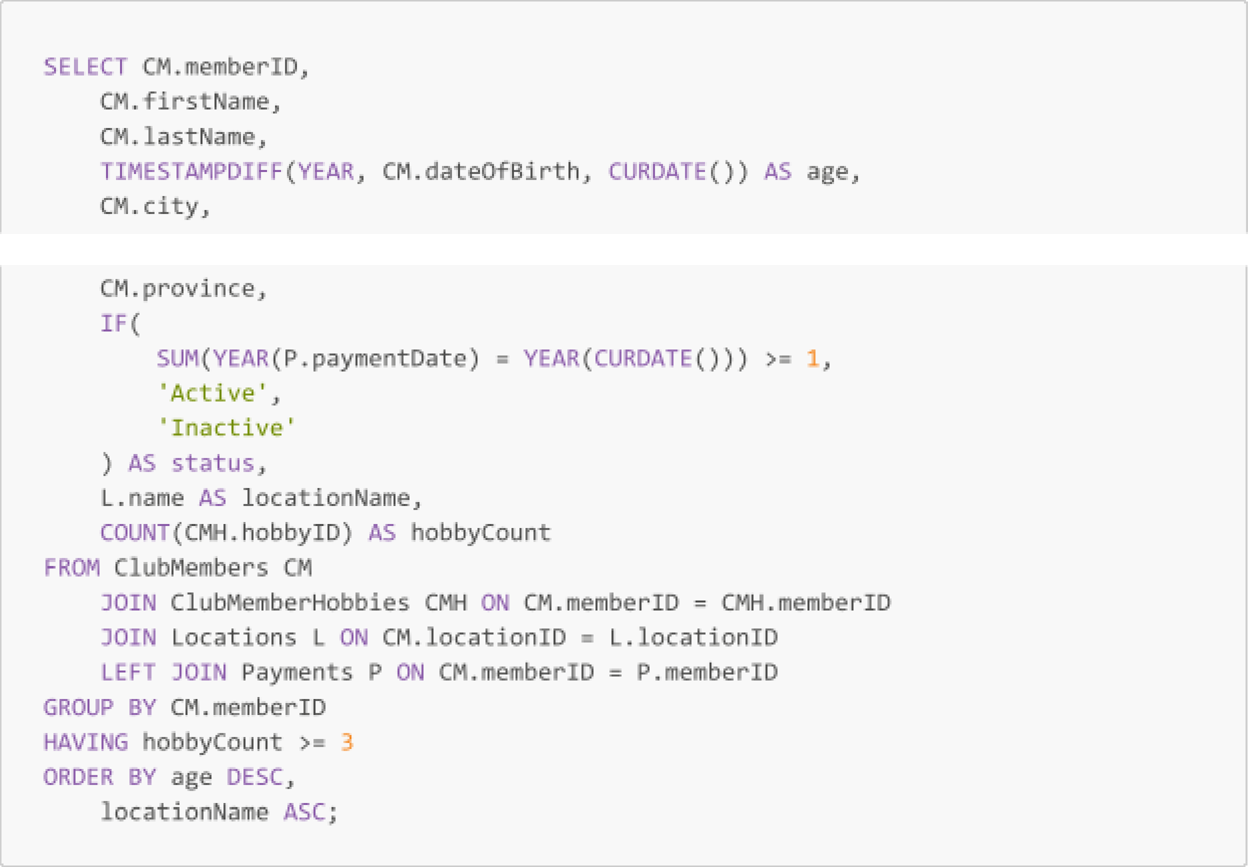
- **0 rows returned** - There are no club members who are also personnel in the current dataset
- The social security numbers between ClubMembers and Personnel tables don't match, indicating complete separation between staff and membership



memberID	firstName	lastName	age	city	province	status	memberLocation	personnelLocation

Figure 2: Result of Query #2

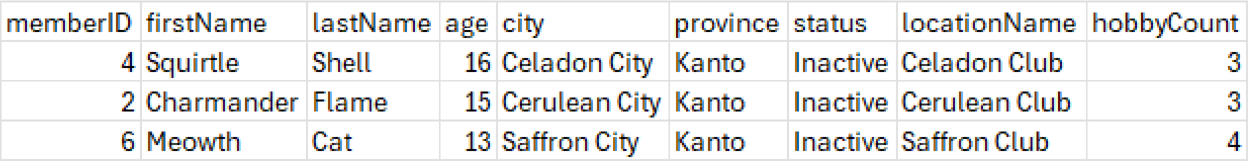
3. Members with at least 3 hobbies



```
SELECT CM.memberID,
    CM.firstName,
    CM.lastName,
    TIMESTAMPDIFF(YEAR, CM.dateOfBirth, CURDATE()) AS age,
    CM.city,
    CM.province,
    IF(
        SUM(YEAR(P.paymentDate) = YEAR(CURDATE())) >= 1,
        'Active',
        'Inactive'
    ) AS status,
    L.name AS locationName,
    COUNT(CMH.hobbyID) AS hobbyCount
FROM ClubMembers CM
    JOIN ClubMemberHobbies CMH ON CM.memberID = CMH.memberID
    JOIN Locations L ON CM.locationID = L.locationID
    LEFT JOIN Payments P ON CM.memberID = P.memberID
GROUP BY CM.memberID
HAVING hobbyCount >= 3
ORDER BY age DESC,
    locationName ASC;
```

Expected Results:

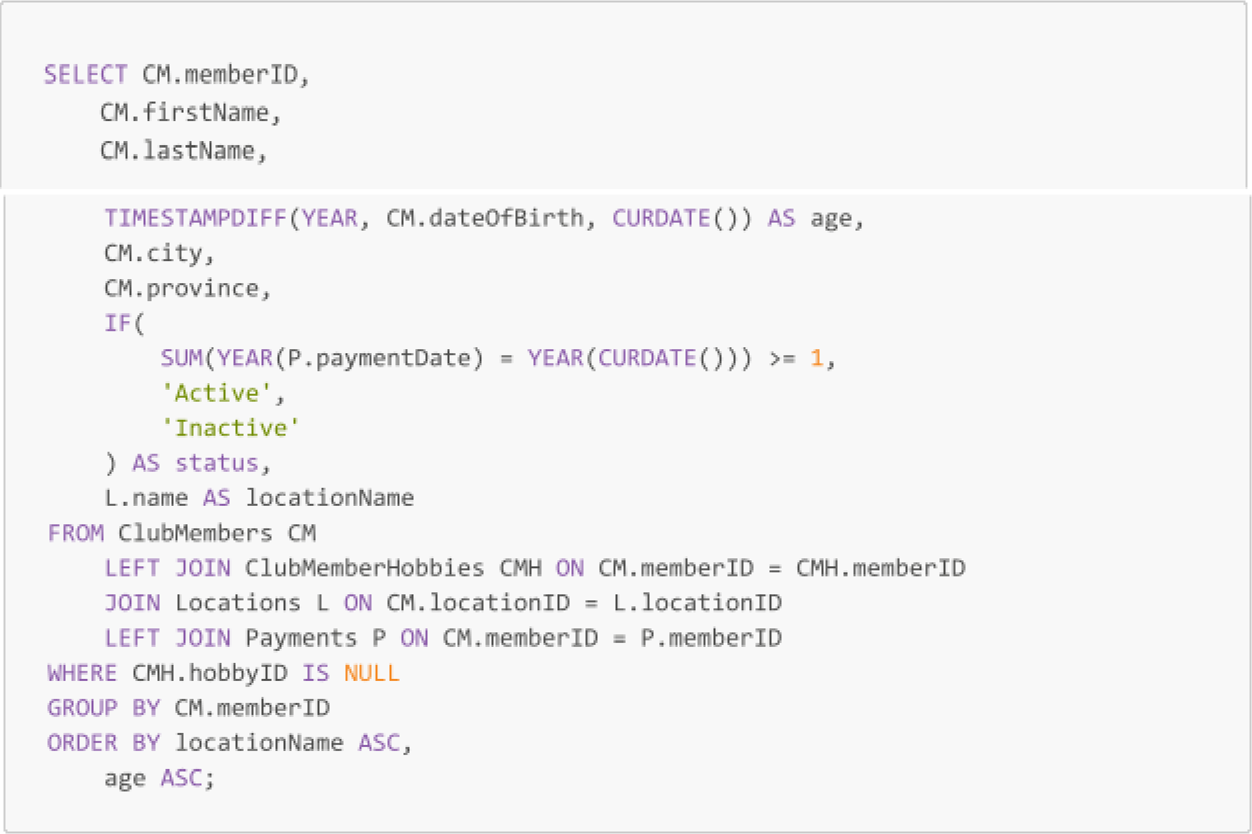
- **3 members with 3+ hobbies:**
    - Charmander Flame (ID: 2): Age 15, Cerulean City, Active, Cerulean Club, 3 hobbies
    - Squirtle Shell (ID: 4): Age 16, Celadon City, Active, Celadon Club, 3 hobbies
    - Meowth Cat (ID: 6): Age 14, Saffron City, Active, Saffron Club, 4 hobbies
- All are active members with current payments



memberID	firstName	lastName	age	city	province	status	locationName	hobbyCount
4	Squirtle	Shell	16	Celadon City	Kanto	Inactive	Celadon Club	3
2	Charmander	Flame	15	Cerulean City	Kanto	Inactive	Cerulean Club	3
6	Meowth	Cat	13	Saffron City	Kanto	Inactive	Saffron Club	4

Figure 3: Result of Query #3

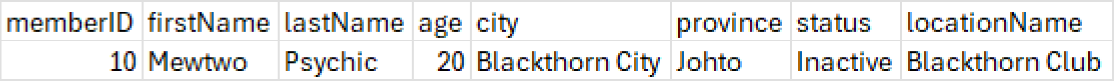
## 4.	Members with no hobbies



```
SELECT CM.memberID,
    CM.firstName,
    CM.lastName,
    TIMESTAMPDIFF(YEAR, CM.dateOfBirth, CURDATE()) AS age,
    CM.city,
    CM.province,
    IF(
        SUM(YEAR(P.paymentDate) = YEAR(CURDATE())) >= 1,
        'Active',
        'Inactive'
    ) AS status,
    L.name AS locationName
FROM ClubMembers CM
    LEFT JOIN ClubMemberHobbies CMH ON CM.memberID = CMH.memberID
    JOIN Locations L ON CM.locationID = L.locationID
    LEFT JOIN Payments P ON CM.memberID = P.memberID
WHERE CMH.hobbyID IS NULL
GROUP BY CM.memberID
ORDER BY locationName ASC,
    age ASC;
```

Expected Results:

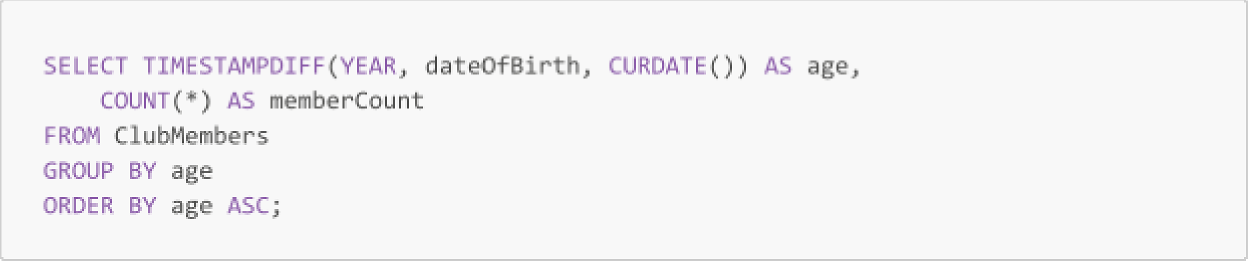
- **1 member with no hobbies:**
    - Mewtwo Psychic (ID: 10): Age 20, Blackthorn City, Active, Blackthorn Club
- This adult member is active with payments but hasn't registered for any hobby activities



memberID	firstName	lastName	age	city	province	status	locationName
10	Mewtwo	Psychic	20	Blackthorn City	Johto	Inactive	Blackthorn Club

Figure 4: Result of Query #4

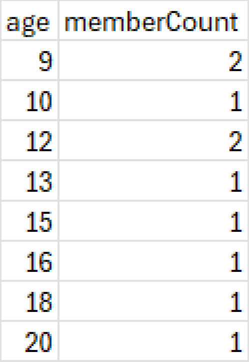
5. Total number of members by age



```
SELECT TIMESTAMPDIFF(YEAR, dateOfBirth, CURDATE()) AS age,
    COUNT(*) AS memberCount
FROM ClubMembers
GROUP BY age
ORDER BY age ASC;
```

Expected Results (as of July 2025):

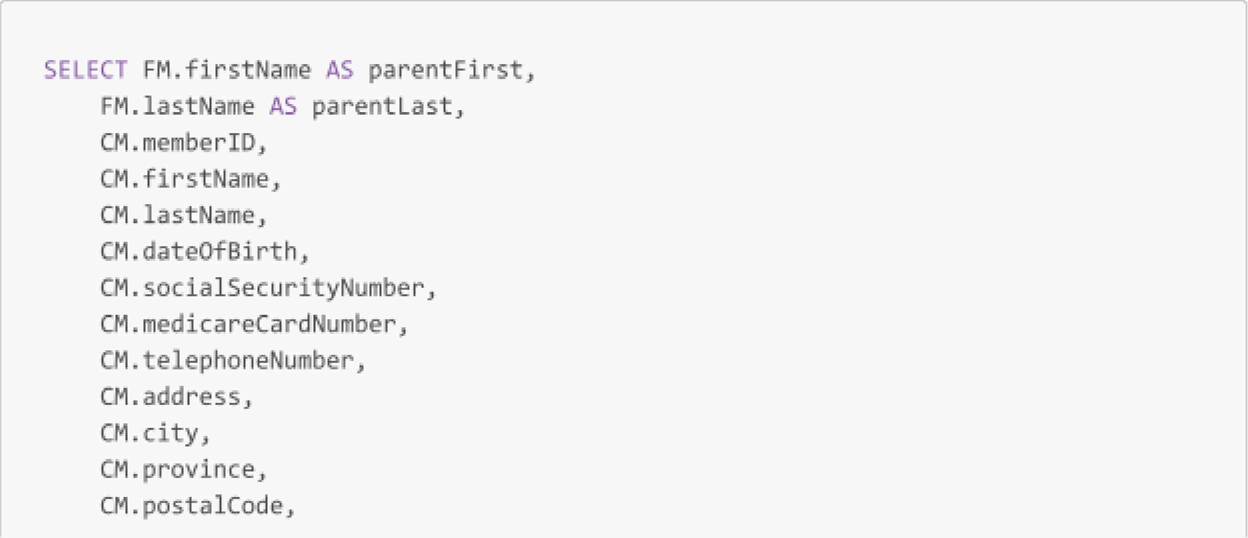
- Age 9: 2 members (Pikachu Electric, Jigglypuff Song)
- Age 11: 1 member (Psyduck Duck)
- Age 12: 2 members (Bulbasaur Leaf, Eevee Evolution)
- Age 14: 1 member (Meowth Cat)
- Age 15: 1 member (Charmander Flame)
- Age 16: 1 member (Squirtle Shell)
- Age 18: 1 member (Snorlax Sleep)
- Age 20: 1 member (Mewtwo Psychic)
- Shows a young membership base with most members being minors



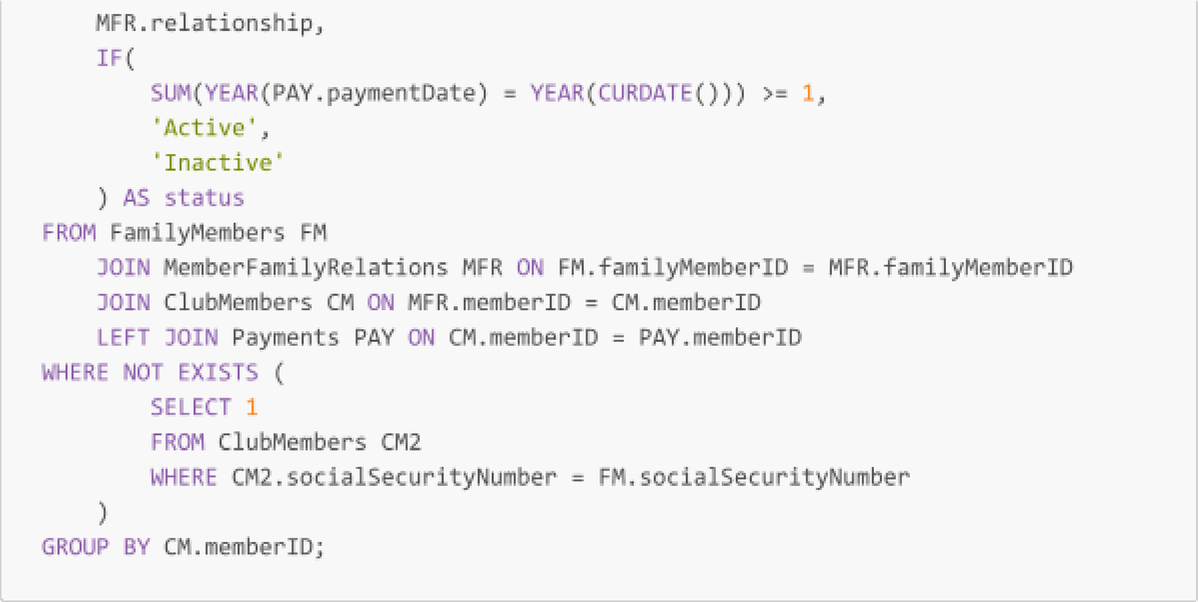
age	memberCount
9	2
10	1
12	2
13	1
15	1
16	1
18	1
20	1

Figure 5: Result of Query #5

6. Major family members and their children



```sql
SELECT FM.firstName AS parentFirst,
    FM.lastName AS parentLast,
    CM.memberID,
    CM.firstName,
    CM.lastName,
    CM.dateOfBirth,
    CM.socialSecurityNumber,
    CM.medicareCardNumber,
    CM.telephoneNumber,
    CM.address,
    CM.city,
    CM.province,
    CM.postalCode,
```

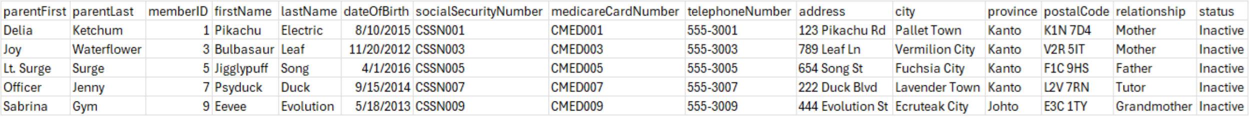


```sql
    MFR.relationship,
    IF(
        SUM(YEAR(PAY.paymentDate) = YEAR(CURDATE())) >= 1,
        'Active',
        'Inactive'
    ) AS status
FROM FamilyMembers FM
    JOIN MemberFamilyRelations MFR ON FM.familyMemberID = MFR.familyMemberID
    JOIN ClubMembers CM ON MFR.memberID = CM.memberID
    LEFT JOIN Payments PAY ON CM.memberID = PAY.memberID
WHERE NOT EXISTS (
        SELECT 1
        FROM ClubMembers CM2
        WHERE CM2.socialSecurityNumber = FM.socialSecurityNumber
    )
GROUP BY CM.memberID;
```

Expected Results:

- **5 family relationships:**
  - Delia Ketchum → Pikachu Electric (Mother relationship)
  - Joy Waterflower → Bulbasaur Leaf (Mother relationship)
  - Lt. Surge → Jigglypuff Song (Father relationship)
  - Officer Jenny → Psyduck Duck (Tutor relationship)
  - Sabrina Gym → Eevee Evolution (Grandmother relationship)

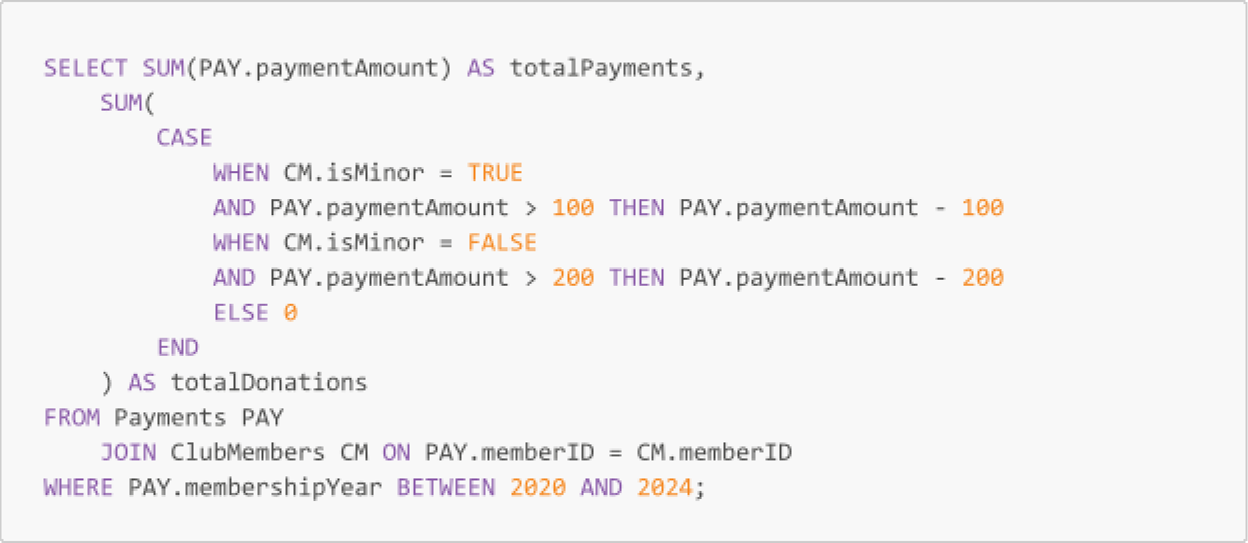
- All related club members are active with current payments
- Shows diverse family relationship types beyond just parents



parentFirst	parentLast	memberID	firstName	lastName	dateOfBirth	socialSecurityNumber	medicareCardNumber	telephoneNumber	address	city	province	postalCode	relationship	status
Delia	Ketchum	1	Pikachu	Electric	8/10/2015	CSSN001	CMED001	555-3001	123 Pikachu Rd	Pallet Town	Kanto	K1N 7D4	Mother	Inactive
Joy	Waterflower	3	Bulbasaur	Leaf	11/20/2012	CSSN003	CMED003	555-3003	789 Leaf Ln	Vermilion City	Kanto	V2R 5IT	Mother	Inactive
Lt. Surge	Surge	5	Jigglypuff	Song	4/1/2016	CSSN005	CMED005	555-3005	654 Song St	Fuchsia City	Kanto	F1C 9HS	Father	Inactive
Officer	Jenny	7	Psyduck	Duck	9/15/2014	CSSN007	CMED007	555-3007	222 Duck Blvd	Lavender Town	Kanto	L2V 7RN	Tutor	Inactive
Sabrina	Gym	9	Eevee	Evolution	5/18/2013	CSSN009	CMED009	555-3009	444 Evolution St	Ecruteak City	Johto	E3C 1TY	Grandmother	Inactive

Figure 6; Result of Query #6

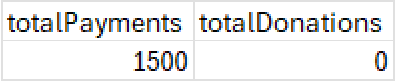
7. Total membership fees and donations by major members (2020–2024)



```
SELECT SUM(PAY.paymentAmount) AS totalPayments,
    SUM(
        CASE
            WHEN CM.isMinor = TRUE
            AND PAY.paymentAmount > 100 THEN PAY.paymentAmount - 100
            WHEN CM.isMinor = FALSE
            AND PAY.paymentAmount > 200 THEN PAY.paymentAmount - 200
            ELSE 0
        END
    ) AS totalDonations
FROM Payments PAY
    JOIN ClubMembers CM ON PAY.memberID = CM.memberID
WHERE PAY.membershipYear BETWEEN 2020 AND 2024;
```

Expected Results:

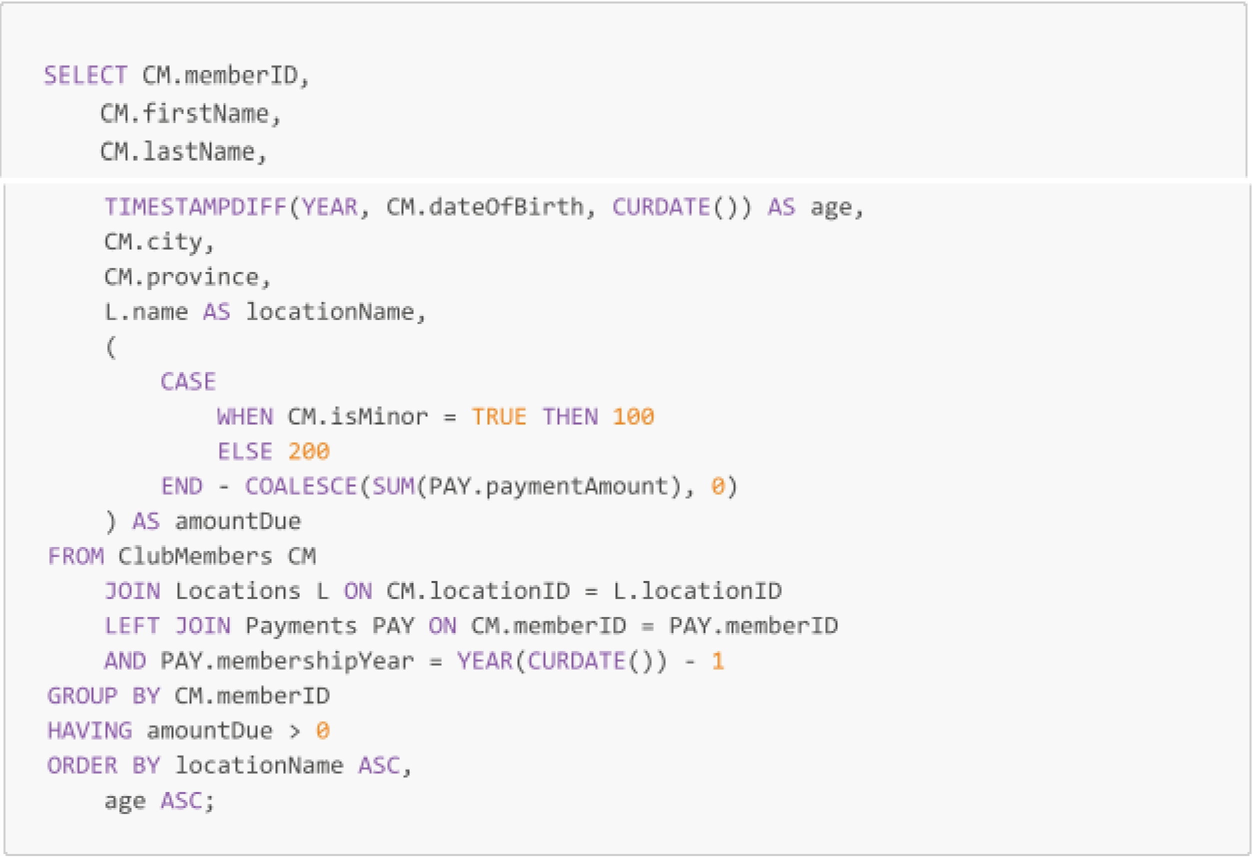
- **Total Payments:** $1,500.00 (sum of all 10 payments from 2024)
    - 6 minor payments: $600.00 (6 × $100)
    - 4 adult payments: $900.00 (4 × $200 + varying amounts)
- **Total Donations:** $0.00 (no payments exceed the standard membership fees)
- All payments exactly match the expected fees: $100 for minors, $200 for adults Shows consistent fee collection but no additional donations received



totalPayments	totalDonations
1500	0

Figure 7: Result of Query #7

8. Inactive members and amount due

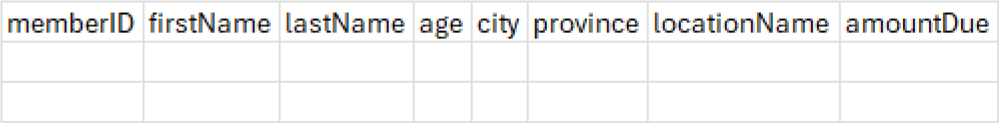


```
SELECT CM.memberID,
    CM.firstName,
    CM.lastName,
    TIMESTAMPDIFF(YEAR, CM.dateOfBirth, CURDATE()) AS age,
    CM.city,
    CM.province,
    L.name AS locationName,
    (
        CASE
            WHEN CM.isMinor = TRUE THEN 100
            ELSE 200
        END - COALESCE(SUM(PAY.paymentAmount), 0)
    ) AS amountDue
FROM ClubMembers CM
    JOIN Locations L ON CM.locationID = L.locationID
    LEFT JOIN Payments PAY ON CM.memberID = PAY.memberID
    AND PAY.membershipYear = YEAR(CURDATE()) - 1
GROUP BY CM.memberID
HAVING amountDue > 0
ORDER BY locationName ASC,
    age ASC;
```

Expected Results:

- **0 rows returned** - No members have outstanding balances
- All members made their required payments for 2024 (the previous year from current date July 2025) Query looks for 2024 payments since current year is 2025, and all payment data shows complete 2024 payments
- This indicates excellent payment compliance across all locations

Note: The query logic checks for payments in YEAR(CURDATE()) - 1 (2024), and since all members have 2024 payment records matching their required fees, no outstanding balances exist.



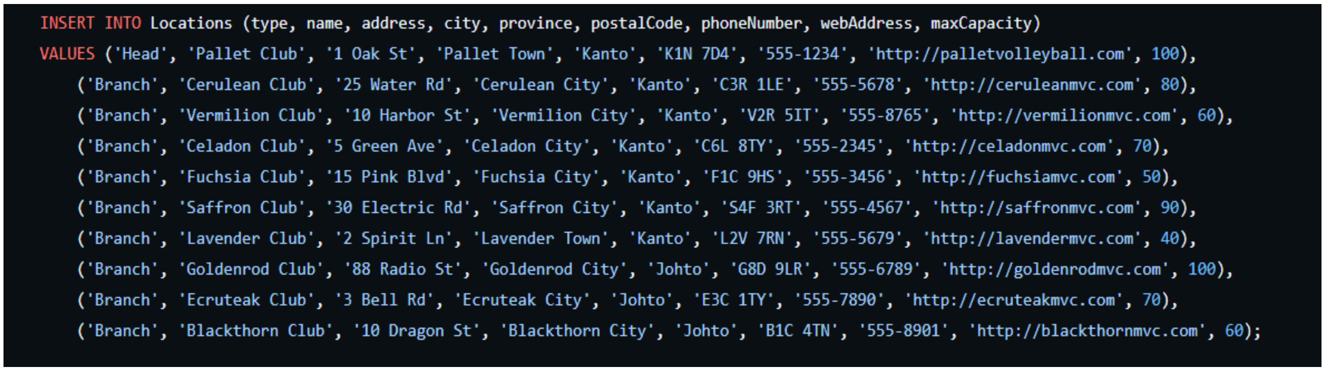
memberID	firstName	lastName	age	city	province	locationName	amountDue

Figure 8: Result of Query #8

Populating the tables in the Databases

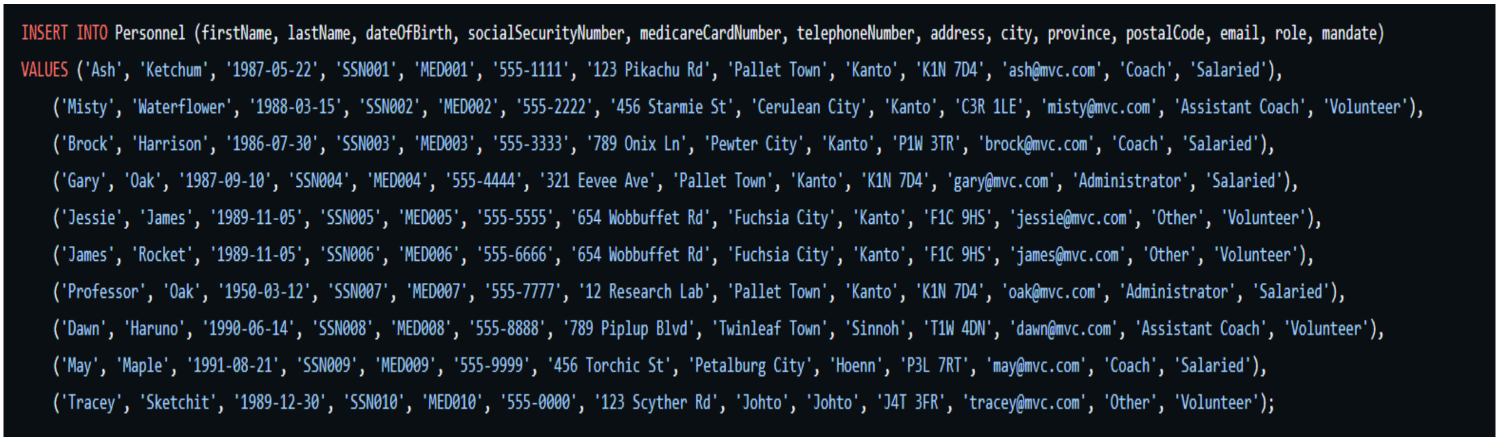
The following section demonstrates the insertion of sample data into the created tables. The data is used in the subsequent queries.

Locations:



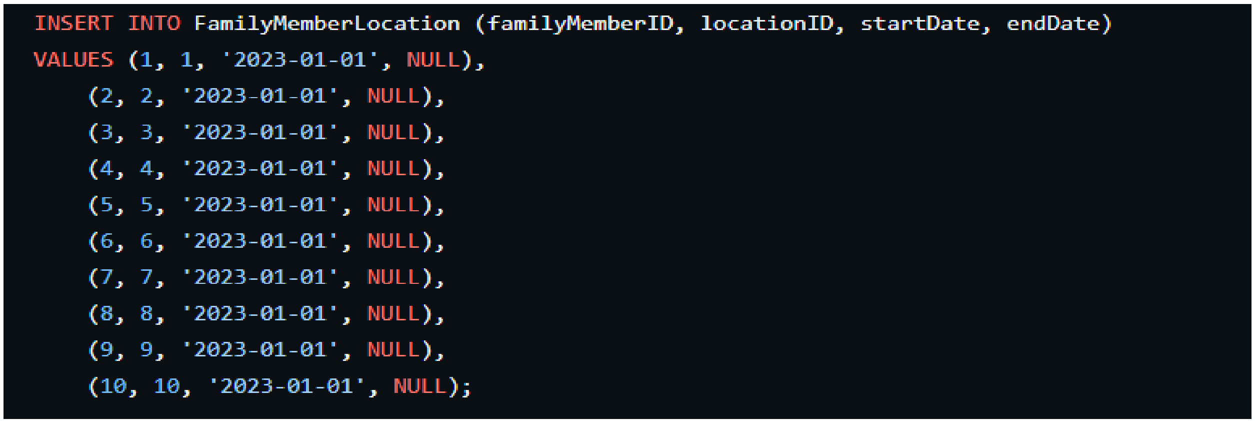
```
INSERT INTO Locations (type, name, address, city, province, postalCode, phoneNumber, webAddress, maxCapacity)
VALUES ('Head', 'Pallet Club', '1 Oak St', 'Pallet Town', 'Kanto', 'K1N 7D4', '555-1234', 'http://palletvolleyball.com', 100),
    ('Branch', 'Cerulean Club', '25 Water Rd', 'Cerulean City', 'Kanto', 'C3R 1LE', '555-5678', 'http://ceruleanmvc.com', 80),
    ('Branch', 'Vermilion Club', '10 Harbor St', 'Vermilion City', 'Kanto', 'V2R 5IT', '555-8765', 'http://vermilionmvc.com', 60),
    ('Branch', 'Celadon Club', '5 Green Ave', 'Celadon City', 'Kanto', 'C6L 8TY', '555-2345', 'http://celadonmvc.com', 70),
    ('Branch', 'Fuchsia Club', '15 Pink Blvd', 'Fuchsia City', 'Kanto', 'F1C 9HS', '555-3456', 'http://fuchsiamvc.com', 50),
    ('Branch', 'Saffron Club', '30 Electric Rd', 'Saffron City', 'Kanto', 'S4F 3RT', '555-4567', 'http://saffronmvc.com', 90),
    ('Branch', 'Lavender Club', '2 Spirit Ln', 'Lavender Town', 'Kanto', 'L2V 7RN', '555-5679', 'http://lavendermvc.com', 40),
    ('Branch', 'Goldenrod Club', '88 Radio St', 'Goldenrod City', 'Johto', 'G8D 9LR', '555-6789', 'http://goldenrodmvc.com', 100),
    ('Branch', 'Ecruteak Club', '3 Bell Rd', 'Ecruteak City', 'Johto', 'E3C 1TY', '555-7890', 'http://ecruteakmvc.com', 70),
    ('Branch', 'Blackthorn Club', '10 Dragon St', 'Blackthorn City', 'Johto', 'B1C 4TN', '555-8901', 'http://blackthornmvc.com', 60);
```

Personnel:



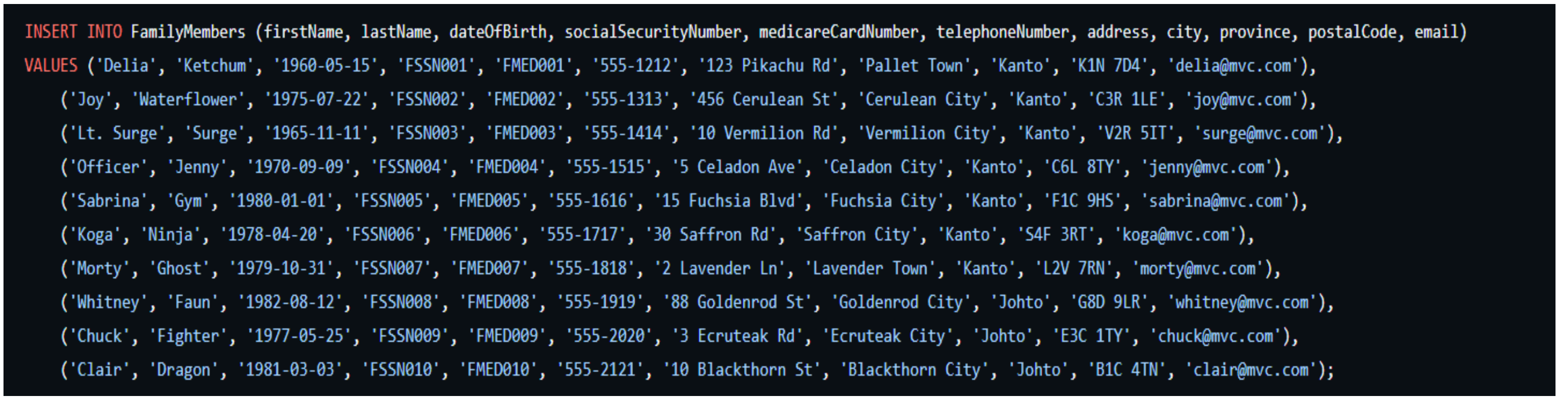
```
INSERT INTO Personnel (firstName, lastName, dateOfBirth, socialSecurityNumber, medicareCardNumber, telephoneNumber, address, city, province, postalCode, email, role, mandate)
VALUES ('Ash', 'Ketchum', '1987-05-22', 'SSN001', 'MED001', '555-1111', '123 Pikachu Rd', 'Pallet Town', 'Kanto', 'K1N 7D4', 'ash@mvc.com', 'Coach', 'Salaried'),
    ('Misty', 'Waterflower', '1988-03-15', 'SSN002', 'MED002', '555-2222', '456 Starmie St', 'Cerulean City', 'Kanto', 'C3R 1LE', 'misty@mvc.com', 'Assistant Coach', 'Volunteer'),
    ('Brock', 'Harrison', '1986-07-30', 'SSN003', 'MED003', '555-3333', '789 Onix Ln', 'Pewter City', 'Kanto', 'P1W 3TR', 'brock@mvc.com', 'Coach', 'Salaried'),
    ('Gary', 'Oak', '1987-09-10', 'SSN004', 'MED004', '555-4444', '321 Eevee Ave', 'Pallet Town', 'Kanto', 'K1N 7D4', 'gary@mvc.com', 'Administrator', 'Salaried'),
    ('Jessie', 'James', '1989-11-05', 'SSN005', 'MED005', '555-5555', '654 Wobbuffet Rd', 'Fuchsia City', 'Kanto', 'F1C 9HS', 'jessie@mvc.com', 'Other', 'Volunteer'),
    ('James', 'Rocket', '1989-11-05', 'SSN006', 'MED006', '555-6666', '654 Wobbuffet Rd', 'Fuchsia City', 'Kanto', 'F1C 9HS', 'james@mvc.com', 'Other', 'Volunteer'),
    ('Professor', 'Oak', '1950-03-12', 'SSN007', 'MED007', '555-7777', '12 Research Lab', 'Pallet Town', 'Kanto', 'K1N 7D4', 'oak@mvc.com', 'Administrator', 'Salaried'),
    ('Dawn', 'Haruno', '1990-06-14', 'SSN008', 'MED008', '555-8888', '789 Piplup Blvd', 'Twinleaf Town', 'Sinnoh', 'T1W 4DN', 'dawn@mvc.com', 'Assistant Coach', 'Volunteer'),
    ('May', 'Maple', '1991-08-21', 'SSN009', 'MED009', '555-9999', '456 Torchic St', 'Petalburg City', 'Hoenn', 'P3L 7RT', 'may@mvc.com', 'Coach', 'Salaried'),
    ('Tracey', 'Sketchit', '1989-12-30', 'SSN010', 'MED010', '555-0000', '123 Scyther Rd', 'Johto', 'Johto', 'J4T 3FR', 'tracey@mvc.com', 'Other', 'Volunteer');
```

Family MemberLocation:



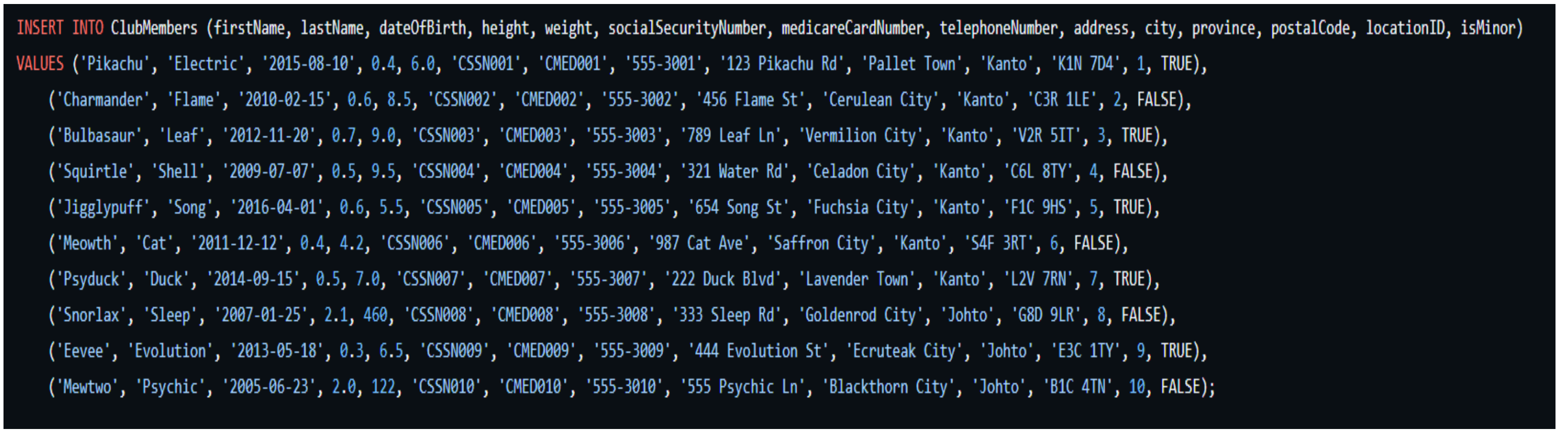
```
INSERT INTO FamilyMemberLocation (familyMemberID, locationID, startDate, endDate)
VALUES (1, 1, '2023-01-01', NULL),
    (2, 2, '2023-01-01', NULL),
    (3, 3, '2023-01-01', NULL),
    (4, 4, '2023-01-01', NULL),
    (5, 5, '2023-01-01', NULL),
    (6, 6, '2023-01-01', NULL),
    (7, 7, '2023-01-01', NULL),
    (8, 8, '2023-01-01', NULL),
    (9, 9, '2023-01-01', NULL),
    (10, 10, '2023-01-01', NULL);
```

Family Members:



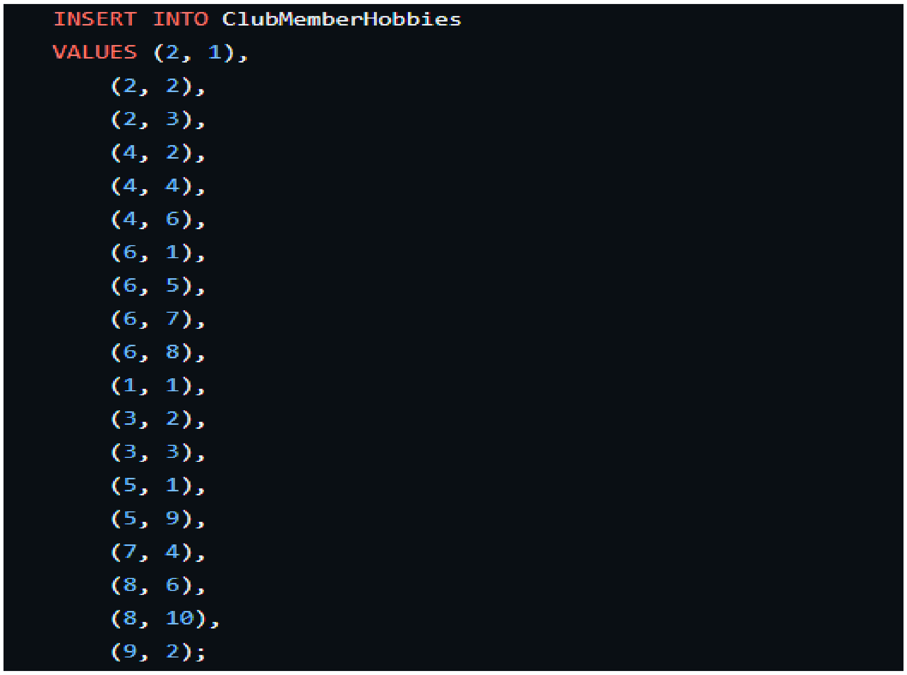
```
INSERT INTO FamilyMembers (firstName, lastName, dateOfBirth, socialSecurityNumber, medicareCardNumber, telephoneNumber, address, city, province, postalCode, email)
VALUES ('Delia', 'Ketchum', '1960-05-15', 'FSSN001', 'FMED001', '555-1212', '123 Pikachu Rd', 'Pallet Town', 'Kanto', 'K1N 7D4', 'delia@mvc.com'),
    ('Joy', 'Waterflower', '1975-07-22', 'FSSN002', 'FMED002', '555-1313', '456 Cerulean St', 'Cerulean City', 'Kanto', 'C3R 1LE', 'joy@mvc.com'),
    ('Lt. Surge', 'Surge', '1965-11-11', 'FSSN003', 'FMED003', '555-1414', '10 Vermilion Rd', 'Vermilion City', 'Kanto', 'V2R 5IT', 'surge@mvc.com'),
    ('Officer', 'Jenny', '1970-09-09', 'FSSN004', 'FMED004', '555-1515', '5 Celadon Ave', 'Celadon City', 'Kanto', 'C6L 8TY', 'jenny@mvc.com'),
    ('Sabrina', 'Gym', '1980-01-01', 'FSSN005', 'FMED005', '555-1616', '15 Fuchsia Blvd', 'Fuchsia City', 'Kanto', 'F1C 9HS', 'sabrina@mvc.com'),
    ('Koga', 'Ninja', '1978-04-20', 'FSSN006', 'FMED006', '555-1717', '30 Saffron Rd', 'Saffron City', 'Kanto', 'S4F 3RT', 'koga@mvc.com'),
    ('Morty', 'Ghost', '1979-10-31', 'FSSN007', 'FMED007', '555-1818', '2 Lavender Ln', 'Lavender Town', 'Kanto', 'L2V 7RN', 'morty@mvc.com'),
    ('Whitney', 'Faun', '1982-08-12', 'FSSN008', 'FMED008', '555-1919', '88 Goldenrod St', 'Goldenrod City', 'Johto', 'G8D 9LR', 'whitney@mvc.com'),
    ('Chuck', 'Fighter', '1977-05-25', 'FSSN009', 'FMED009', '555-2020', '3 Ecruteak Rd', 'Ecruteak City', 'Johto', 'E3C 1TY', 'chuck@mvc.com'),
    ('Clair', 'Dragon', '1981-03-03', 'FSSN010', 'FMED010', '555-2121', '10 Blackthorn St', 'Blackthorn City', 'Johto', 'B1C 4TN', 'clair@mvc.com');
```

ClubMembers:



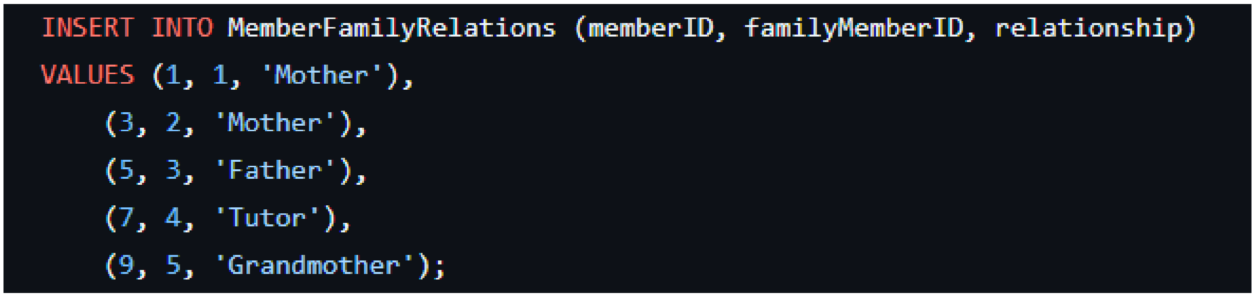
```
INSERT INTO ClubMembers (firstName, lastName, dateOfBirth, height, weight, socialSecurityNumber, medicareCardNumber, telephoneNumber, address, city, province, postalCode, locationID, isMinor)
VALUES ('Pikachu', 'Electric', '2015-08-10', 0.4, 6.0, 'CSSN001', 'CMED001', '555-3001', '123 Pikachu Rd', 'Pallet Town', 'Kanto', 'K1N 7D4', 1, TRUE),
   ('Charmander', 'Flame', '2010-02-15', 0.6, 8.5, 'CSSN002', 'CMED002', '555-3002', '456 Flame St', 'Cerulean City', 'Kanto', 'C3R 1LE', 2, FALSE),
   ('Bulbasaur', 'Leaf', '2012-11-20', 0.7, 9.0, 'CSSN003', 'CMED003', '555-3003', '789 Leaf Ln', 'Vermilion City', 'Kanto', 'V2R 5IT', 3, TRUE),
   ('Squirtle', 'Shell', '2009-07-07', 0.5, 9.5, 'CSSN004', 'CMED004', '555-3004', '321 Water Rd', 'Celadon City', 'Kanto', 'C6L 8TY', 4, FALSE),
   ('Jigglypuff', 'Song', '2016-04-01', 0.6, 5.5, 'CSSN005', 'CMED005', '555-3005', '654 Song St', 'Fuchsia City', 'Kanto', 'F1C 9HS', 5, TRUE),
   ('Meowth', 'Cat', '2011-12-12', 0.4, 4.2, 'CSSN006', 'CMED006', '555-3006', '987 Cat Ave', 'Saffron City', 'Kanto', 'S4F 3RT', 6, FALSE),
   ('Psyduck', 'Duck', '2014-09-15', 0.5, 7.0, 'CSSN007', 'CMED007', '555-3007', '222 Duck Blvd', 'Lavender Town', 'Kanto', 'L2V 7RN', 7, TRUE),
   ('Snorlax', 'Sleep', '2007-01-25', 2.1, 460, 'CSSN008', 'CMED008', '555-3008', '333 Sleep Rd', 'Goldenrod City', 'Johto', 'G8D 9LR', 8, FALSE),
   ('Eevee', 'Evolution', '2013-05-18', 0.3, 6.5, 'CSSN009', 'CMED009', '555-3009', '444 Evolution St', 'Ecruteak City', 'Johto', 'E3C 1TY', 9, TRUE),
   ('Mewtwo', 'Psychic', '2005-06-23', 2.0, 122, 'CSSN010', 'CMED010', '555-3010', '555 Psychic Ln', 'Blackthorn City', 'Johto', 'B1C 4TN', 10, FALSE);
```

ClubMemberHobbies:



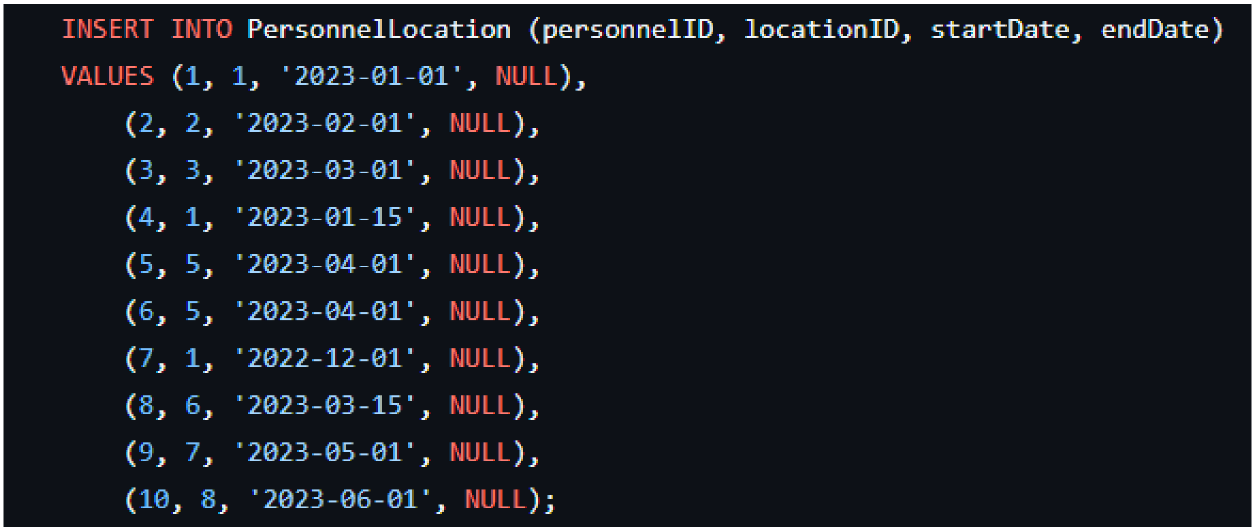
```
INSERT INTO ClubMemberHobbies
VALUES (2, 1),
    (2, 2),
    (2, 3),
    (4, 2),
    (4, 4),
    (4, 6),
    (6, 1),
    (6, 5),
    (6, 7),
    (6, 8),
    (1, 1),
    (3, 2),
    (3, 3),
    (5, 1),
    (5, 9),
    (7, 4),
    (8, 6),
    (8, 10),
    (9, 2);
```

MemberFamilyRelations:



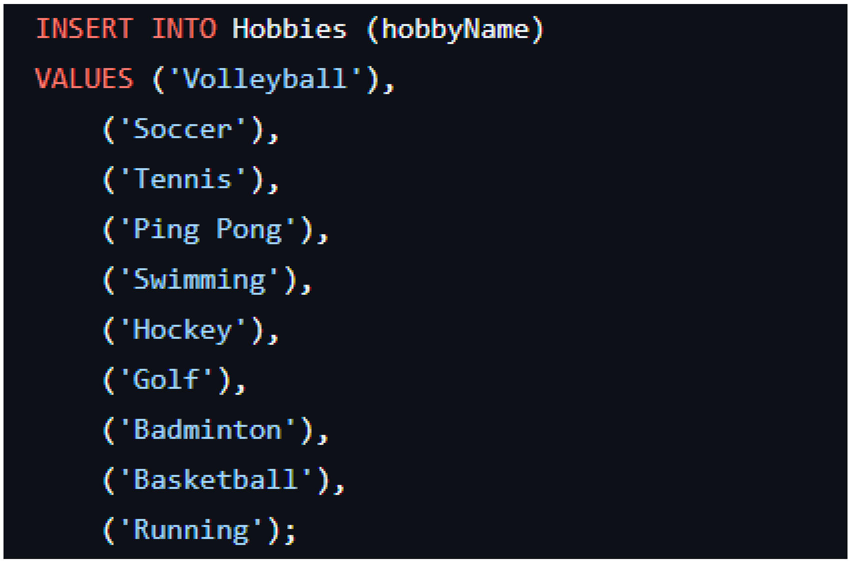
```
INSERT INTO MemberFamilyRelations (memberID, familyMemberID, relationship)
VALUES (1, 1, 'Mother'),
    (3, 2, 'Mother'),
    (5, 3, 'Father'),
    (7, 4, 'Tutor'),
    (9, 5, 'Grandmother');
```

PersonnelLocation:



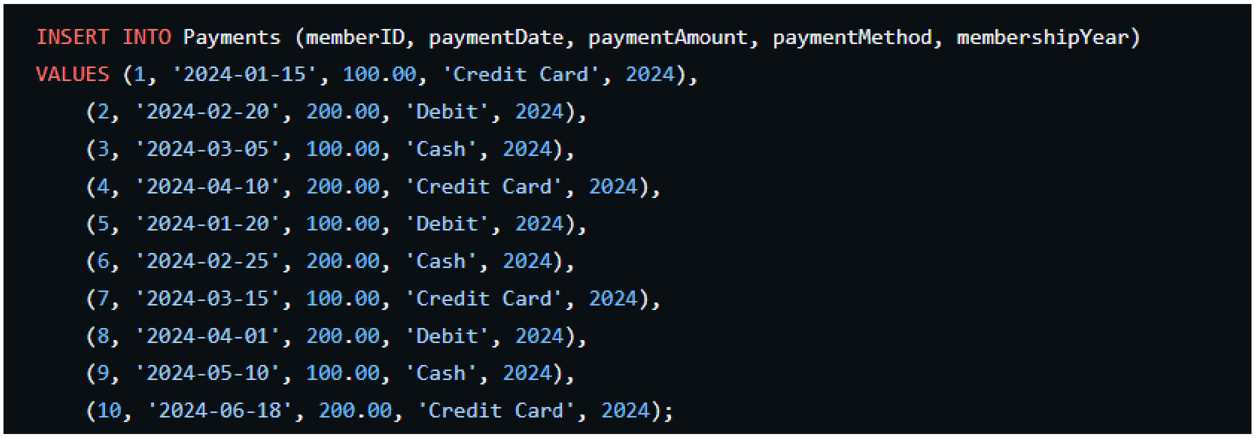
```
INSERT INTO PersonnelLocation (personnelID, locationID, startDate, endDate)
VALUES (1, 1, '2023-01-01', NULL),
    (2, 2, '2023-02-01', NULL),
    (3, 3, '2023-03-01', NULL),
    (4, 1, '2023-01-15', NULL),
    (5, 5, '2023-04-01', NULL),
    (6, 5, '2023-04-01', NULL),
    (7, 1, '2022-12-01', NULL),
    (8, 6, '2023-03-15', NULL),
    (9, 7, '2023-05-01', NULL),
    (10, 8, '2023-06-01', NULL);
```

Hobbies:



```
INSERT INTO Hobbies (hobbyName)
VALUES ('Volleyball'),
    ('Soccer'),
    ('Tennis'),
    ('Ping Pong'),
    ('Swimming'),
    ('Hockey'),
    ('Golf'),
    ('Badminton'),
    ('Basketball'),
    ('Running');
```

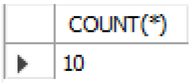
Payments:



```
INSERT INTO Payments (memberID, paymentDate, paymentAmount, paymentMethod, membershipYear)
VALUES (1, '2024-01-15', 100.00, 'Credit Card', 2024),
    (2, '2024-02-20', 200.00, 'Debit', 2024),
    (3, '2024-03-05', 100.00, 'Cash', 2024),
    (4, '2024-04-10', 200.00, 'Credit Card', 2024),
    (5, '2024-01-20', 100.00, 'Debit', 2024),
    (6, '2024-02-25', 200.00, 'Cash', 2024),
    (7, '2024-03-15', 100.00, 'Credit Card', 2024),
    (8, '2024-04-01', 200.00, 'Debit', 2024),
    (9, '2024-05-10', 100.00, 'Cash', 2024),
    (10, '2024-06-18', 200.00, 'Credit Card', 2024);
```

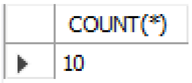
Count * Statements

SELECT COUNT(*) FROM Locations;



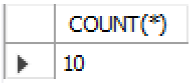
COUNT(*)
10

SELECT COUNT(*) FROM Personnel;



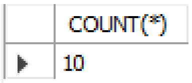
COUNT(*)
10

SELECT COUNT(*) FROM PersonnelLocations;



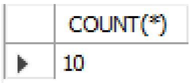
COUNT(*)
10

SELECT COUNT(*) FROM FamilyMembers;



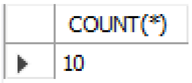
COUNT(*)
10

SELECT COUNT(*) FROM FamilyMemberLocation;



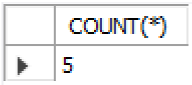
COUNT(*)
10

SELECT COUNT(*) FROM ClubMembers;



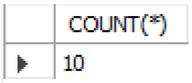
COUNT(*)
10

SELECT COUNT(*) FROM MemberFamilyRelations;



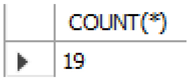
COUNT(*)
5

SELECT COUNT(*) FROM Hobbies;



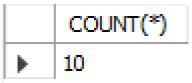
COUNT(*)
10

SELECT COUNT(*) FROM ClubMemberHobbies;



COUNT(*)
19

SELECT COUNT(*) FROM Payments;



COUNT(*)
10