# Gym Management System

- **Name: Rosu Liviu-Mihai**
- **Group: 1076**

**A Gym Database System** created to manage member information, handle payments, track workout programs and overall organize a gym's information

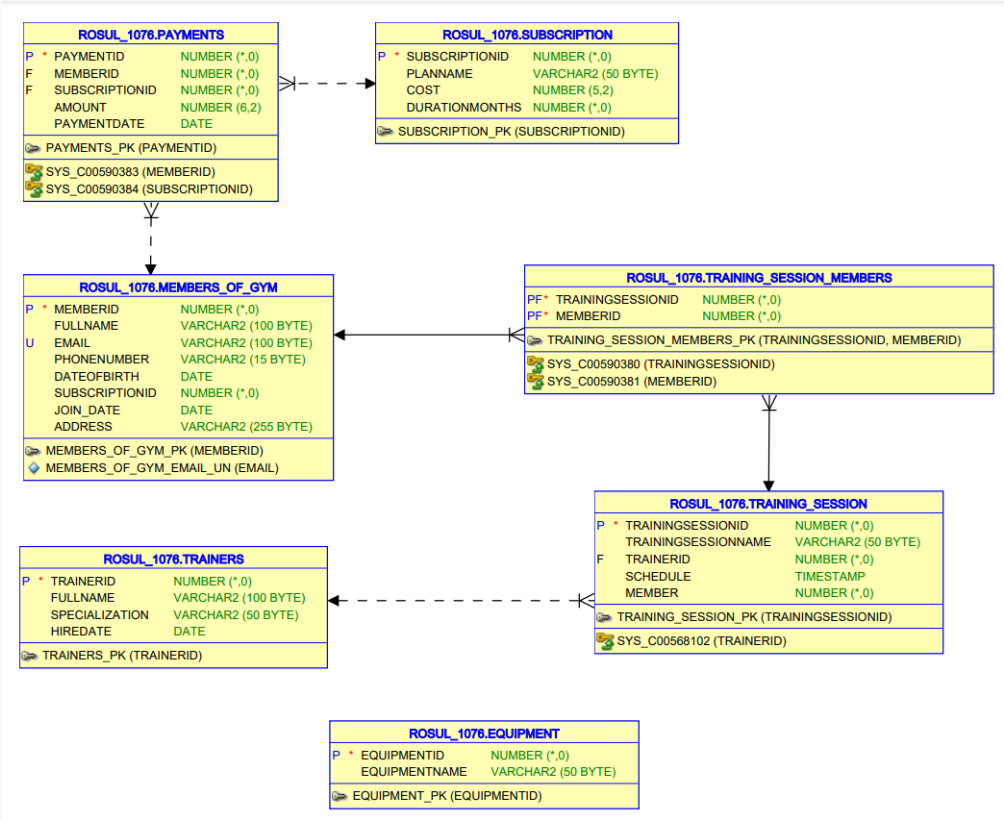
# Why this subject?

**Gyms** are essential for fostering healthier lifestyles, providing a structured and supportive environment for individuals to engage in physical activity. Regular exercise is a cornerstone of physical and mental well-being, reducing the risk of chronic illnesses such as heart disease, diabetes, and obesity while also boosting mental health by alleviating stress and improving mood. In an era where sedentary lifestyles and busy schedules are common, gyms offer accessible opportunities for people to prioritize their health through professional guidance, specialized equipment, and community-driven activities.

Beyond physical benefits, gyms also encourage social connections and personal growth, making them a vital resource for individuals striving to lead balanced and active lives. The fitness industry's focus on health, wellness, and community impact underscores its importance and relevance, making it an ideal subject for this database project.

# DATABASE SCHEMA

- The database schema of the gym is the following:



# A short description of the tables

- This project consists of 7 interconnected tables designed to manage a gym's operations efficiently. The **MEMBERS_OF_GYM** table stores member details, while **TRAINERS** holds information about gym trainers. **TRAINING_SESSION** and **TRAINING_SESSION_MEMBERS** manage scheduling and member participation in sessions. The **EQUIPMENT** table tracks gym equipment, **SUBSCRIPTION** handles membership plans, and **PAYMENTS** records financial transactions. Together, these tables provide a comprehensive framework for managing gym activities and resources.

# Below are 7 images showcasing the tables (on the left) and their corresponding code (on the right)

## I will start with the first 2 tables, namely: 'MEMBERS_OF_GYM' and 'TRAINERS'

project | MEMBERS_OF_GYM

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

| | COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | MEMBERID | NUMBER(38,0) | No | (null) | 1 | (null) |
| 2 | FULLNAME | VARCHAR2(100 BYTE) | Yes | (null) | 2 | (null) |
| 3 | EMAIL | VARCHAR2(100 BYTE) | Yes | (null) | 3 | (null) |
| 4 | PHONENUMBER | VARCHAR2(15 BYTE) | Yes | (null) | 4 | (null) |
| 5 | DATEOFBIRTH | DATE | Yes | (null) | 5 | (null) |
| 6 | SUBSCRIPTIONID | NUMBER(38,0) | Yes | (null) | 6 | (null) |
| 7 | JOIN_DATE | DATE | Yes | SYSDATE | 7 | (null) |

```sql
--1. Constructing the database tables
CREATE TABLE "MEMBERS_OF_GYM" (
    MemberID INT PRIMARY KEY,
    FullName VARCHAR(100),
    Email VARCHAR(100) UNIQUE,
    PhoneNumber VARCHAR(15),
    DateOfBirth DATE,
    SubscriptionID INT,
    JOIN_DATE DATE DEFAULT SYSDATE
);
```

project | TRAINERS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

| | COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | TRAINERID | NUMBER(38,0) | No | (null) | 1 | (null) |
| 2 | FULLNAME | VARCHAR2(100 BYTE) | Yes | (null) | 2 | (null) |
| 3 | SPECIALIZATION | VARCHAR2(50 BYTE) | Yes | (null) | 3 | (null) |
| 4 | HIREDATE | DATE | Yes | SYSDATE | 4 | (null) |

```sql
CREATE TABLE "TRAINERS" (
    TrainerID INT PRIMARY KEY,
    FullName VARCHAR(100),
    Specialization VARCHAR(50),
    HireDate DATE DEFAULT SYSDATE
);
```

# The next two tables are: 'TRAINING_SESSION' and 'TRAINING_SESSION_MEMBERS'

project | TRAINING_SESSION

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

| | COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | TRAININGSESSIONID | NUMBER(38,0) | No | (null) | 1 | (null) |
| 2 | TRAININGSESSIONNAME | VARCHAR2(50 BYTE) | Yes | (null) | 2 | (null) |
| 3 | TRAINERID | NUMBER(38,0) | Yes | (null) | 3 | (null) |
| 4 | SCHEDULE | TIMESTAMP(6) | Yes | (null) | 4 | (null) |
| 5 | MEMBER | NUMBER(38,0) | Yes | (null) | 5 | (null) |

```sql
CREATE TABLE "TRAINING_SESSION" (
    TrainingSessionID INT PRIMARY KEY,
    TrainingSessionName VARCHAR(50),
    TrainerID INT,
    Schedule TIMESTAMP,
    FOREIGN KEY (TrainerID) REFERENCES Trainers(TrainerID)
);
```

project | TRAINING_SESSION_MEMBERS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

| | COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | TRAININGSESSIONID | NUMBER(38,0) | No | (null) | 1 | (null) |
| 2 | MEMBERID | NUMBER(38,0) | No | (null) | 2 | (null) |

```sql
CREATE TABLE TRAINING_SESSION_MEMBERS (
    TrainingSessionID INT,
    MemberID INT,
    PRIMARY KEY (TrainingSessionID, MemberID),
    FOREIGN KEY (TrainingSessionID) REFERENCES TRAINING_SESSION(TrainingSessionID),
    FOREIGN KEY (MemberID) REFERENCES MEMBERS_OF_GYM(MemberID)
);
```

# Next, we have: 'EQUIPMENT' and 'SUBSCRIPTION'

project | EQUIPMENT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

| | COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | EQUIPMENTID | NUMBER(38,0) | No | (null) | 1 | (null) |
| 2 | EQUIPMENTNAME | VARCHAR2(50 BYTE) | Yes | (null) | 2 | (null) |

```sql
CREATE TABLE "EQUIPMENT" (
    EquipmentID INT PRIMARY KEY,
    EquipmentName VARCHAR(50)
);
```

project | SUBSCRIPTION

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

| | COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | SUBSCRIPTIONID | NUMBER(38,0) | No | (null) | 1 | (null) |
| 2 | PLANNAME | VARCHAR2(50 BYTE) | Yes | (null) | 2 | (null) |
| 3 | COST | NUMBER(5,2) | Yes | (null) | 3 | (null) |
| 4 | DURATIONMONTHS | NUMBER(38,0) | Yes | (null) | 4 | (null) |

```sql
CREATE TABLE "SUBSCRIPTION" (
    SubscriptionID INT PRIMARY KEY,
    PlanName VARCHAR(50),
    Cost DECIMAL(5, 2), --for 5 digits with 2 digits after decimal point
    DurationMonths INT
);
```

# Lastly, marking the end of the 'CREATE TABLE' phase, we have: 'PAYMENTS'

project | PAYMENTS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

| | COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | PAYMENTID | NUMBER(38,0) | No | (null) | 1 | (null) |
| 2 | MEMBERID | NUMBER(38,0) | Yes | (null) | 2 | (null) |
| 3 | SUBSCRIPTIONID | NUMBER(38,0) | Yes | (null) | 3 | (null) |
| 4 | AMOUNT | NUMBER(6,2) | Yes | (null) | 4 | (null) |
| 5 | PAYMENTDATE | DATE | Yes | SYSDATE | 5 | (null) |

```sql
CREATE TABLE PAYMENTS (
    PaymentID INT PRIMARY KEY,
    MemberID INT,
    SubscriptionID INT,
    Amount DECIMAL(6, 2),
    PaymentDate DATE DEFAULT SYSDATE,
    FOREIGN KEY (MemberID) REFERENCES MEMBERS_OF_GYM(MemberID),
    FOREIGN KEY (SubscriptionID) REFERENCES SUBSCRIPTION(SubscriptionID)
);
```

## For the next step, I have inserted values into the 'MEMBERS_OF_GYM' table to populate it with relevant member data.

```oracle
--2. Using DML Statements
--inserting members of the gym
INSERT INTO MEMBERS_OF_GYM (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID)
VALUES (1, 'John Doe', 'john.doe@gmail.com', '0234567890', TO_DATE('1990-05-15', 'YYYY-MM-DD'), 101);
INSERT INTO MEMBERS_OF_GYM (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID)
VALUES (2, 'Mike Johnson', 'mike.johnson@gmail.com', '0987654321', TO_DATE('1995-07-20', 'YYYY-MM-DD'), 102);
INSERT INTO MEMBERS_OF_GYM (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID)
VALUES (3, 'Alice Matthews', 'alicematthwes@gmail.com', '0234098765', TO_DATE('1998-10-25', 'YYYY-MM-DD'), 103);
INSERT INTO MEMBERS_OF_GYM (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID)
VALUES (4, 'Andrew Joe', 'andrewjoe@gmail.com', '0432167890', TO_DATE('1999-11-04', 'YYYY-MM-DD'), 104);
INSERT INTO MEMBERS_OF_GYM (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID)
VALUES (5, 'John Smith', 'johnsmith@gmail.com', '0238495607', TO_DATE('1993-07-21', 'YYYY-MM-DD'), 105);
INSERT INTO MEMBERS_OF_GYM (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID)
VALUES (6, 'Daniel Miller', 'danielmiller@gmail.com', '0659231837', TO_DATE('2003-02-20', 'YYYY-MM-DD'), 106);
INSERT INTO MEMBERS_OF_GYM (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID)
VALUES (7, 'Emma Martinez', 'emma.martinez@gmail.com', '0436586978', TO_DATE('1998-10-25', 'YYYY-MM-DD'), 107);
INSERT INTO MEMBERS_OF_GYM (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID)
VALUES (8, 'Sophia Brown', 'sophiabrown@gmail.com', '0475697821', TO_DATE('1999-11-04', 'YYYY-MM-DD'), 108);
INSERT INTO MEMBERS_OF_GYM (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID)
VALUES (9, 'Michael Williams', 'michaelwilliams@gmail.com', '0348695869', TO_DATE('2002-06-04', 'YYYY-MM-DD'), 109);
INSERT INTO MEMBERS_OF_GYM (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID)
VALUES (10, 'Olivia Garcia', 'oliviagarcia@gmail.com', '0492182738', TO_DATE('2001-04-04', 'YYYY-MM-DD'), 110);
```

| | MEMBE... | FULLNAME | EMAIL | PHONENUMBER | DATEOFBIRTH | SUBSCRIPTIONID | JOIN_DATE | ADDRESS |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Michael Doe | michaeldoe@gmail.com | 0737217263 | 10-MAY-90 | 1 | 18-JAN-25 | (null) |
| 2 | 2 | Mike Johnson | mike.johnson@gmail.com | 0987654321 | 20-JUL-95 | 102 | 22-JAN-25 | (null) |
| 3 | 3 | Alice Matthews | alicematthwes@gmail.com | 1234098765 | 25-OCT-98 | 103 | 21-JAN-25 | (null) |
| 4 | 4 | Andrew Joe | andrewjoe@gmail.com | 5432167890 | 04-NOV-99 | 104 | 21-JAN-25 | (null) |
| 5 | 5 | Emily Davis | emily.davis@gmail.com | 111111111 | 18-MAR-92 | 101 | 20-JAN-25 | (null) |
| 6 | 6 | Daniel Miller | danielmiller@gmail.com | 0659231837 | 20-FEB-03 | 106 | 22-JAN-25 | (null) |
| 7 | 7 | Emma Martinez | emma.martinez@gmail.com | 0436586978 | 25-OCT-98 | 107 | 22-JAN-25 | (null) |
| 8 | 8 | Sophia Brown | sophiabrown@gmail.com | 0475697821 | 04-NOV-99 | 108 | 22-JAN-25 | (null) |
| 9 | 9 | Michael Williams | michaelwilliams@gmail.com | 0348695869 | 04-JUN-02 | 109 | 22-JAN-25 | (null) |
| 10 | 10 | Olivia Garcia | oliviagarcia@gmail.com | 0492182738 | 04-APR-01 | 110 | 22-JAN-25 | (null) |

## Next, I inserted data into the 'TRAINERS' table and the 'TRAINING_SESSION' . Since it's a small gym, we have three trainers listed and three types of training sessions in the system.

### Data corresponding to 'TRAINERS'

```
--inserting trainers
INSERT INTO TRAINERS (TrainerID, FullName, Specialization, HireDate)
VALUES (1, 'Mike Johnson', 'Weightlifting', TO_DATE('2000-03-22', 'YYYY-MM-DD'));
INSERT INTO TRAINERS (TrainerID, FullName, Specialization, HireDate)
VALUES (2, 'John Michael', 'Losing weight', TO_DATE('1998-05-27', 'YYYY-MM-DD'));
INSERT INTO TRAINERS (TrainerID, FullName, Specialization, HireDate)
VALUES (3, 'Sarah Connor', 'Yoga', TO_DATE('1999-07-17', 'YYYY-MM-DD'));
```

| | TRAINERID | FULLNAME | SPECIALIZATION | HIREDATE |
|---|---|---|---|---|
| 1 | 1 | Mike Johnson | Powerlifting | 22-MAR-00 |
| 2 | 2 | John Michael | Losing weight | 27-MAY-98 |
| 3 | 3 | Sarah Connor | Yoga | 17-JUL-99 |

### Data corresponding to 'TRAINING_SESSION

```
--inserting training sessions
INSERT INTO TRAINING_SESSION (TrainingSessionID, TrainingSessionName, TrainerID, Schedule)
VALUES (1, 'Morning Cardio', 2, TO_DATE('2025-05-10 08:00:00', 'YYYY-MM-DD HH24:MI:SS'));
INSERT INTO TRAINING_SESSION (TrainingSessionID, TrainingSessionName, TrainerID, Schedule)
VALUES (2, 'Strength training', 1, TO_DATE('2025-05-11 15:00:00', 'YYYY-MM-DD HH24:MI:SS'));
INSERT INTO TRAINING_SESSION (TrainingSessionID, TrainingSessionName, TrainerID, Schedule)
VALUES (3, 'Pilates', 3, TO_DATE('2025-05-12 10:30:00', 'YYYY-MM-DD HH24:MI:SS'));
```

| | TRAININGSESSIONID | TRAININGSESSIONNAME | TRAINERID | SCHEDULE | MEMBER |
|---|---|---|---|---|---|
| 1 | 1 | Morning Cardio | 2 | 10-MAY-25 08.00.00.000000000 AM | (null) |
| 2 | 2 | Strength training | 1 | 11-MAY-25 03.00.00.000000000 PM | (null) |
| 3 | 3 | Pilates | 3 | 12-MAY-25 10.30.00.000000000 AM | (null) |

## The next set of data inserted was for the 'EQUIPMENT' table, where details of the gym's equipment were added.

```
--inserting equpiment
INSERT INTO EQUIPMENT (EquipmentID, EquipmentName) VALUES (1, 'Treadmill');
INSERT INTO EQUIPMENT (EquipmentID, EquipmentName) VALUES (2, 'Dumbbells');
INSERT INTO EQUIPMENT (EquipmentID, EquipmentName) VALUES (3, 'Barbells');
INSERT INTO EQUIPMENT (EquipmentID, EquipmentName) VALUES (4, 'Row Machine');
INSERT INTO EQUIPMENT (EquipmentID, EquipmentName) VALUES (5, 'Cardio Bike');
INSERT INTO EQUIPMENT (EquipmentID, EquipmentName) VALUES (6, 'Cardio Climbing Stairs');
INSERT INTO EQUIPMENT (EquipmentID, EquipmentName) VALUES (7, 'Squat Bar');
INSERT INTO EQUIPMENT (EquipmentID, EquipmentName) VALUES (8, 'Chest Machine');
```

| | EQUIPMENTID | EQUIPMENTNAME |
|---|---|---|
| 1 | 1 | Treadmill |
| 2 | 2 | Dumbbells |
| 3 | 3 | Barbells |
| 4 | 4 | Row Machine |
| 5 | 5 | Cardio Bike |
| 6 | 6 | Cardio Climbing Stairs |
| 7 | 7 | Squat Bar |
| 8 | 8 | Chest Machine |

## Next i added 2 types of subscriptions: 'Basic Plan' and 'Premium Plan'

## The 'Basic Plan' is cheaper and only lasts 1 month.
## The 'Premium Plan', while being more expensive lasts up to 3 months.

```
--inserting subscription types
INSERT INTO SUBSCRIPTION (SubscriptionID, PlanName, Cost, DurationMonths)
VALUES (101, 'Basic Plan', 50.00, 1);
INSERT INTO SUBSCRIPTION (SubscriptionID, PlanName, Cost, DurationMonths)
VALUES (102, 'Premium Plan', 100.00, 3);
```

| | | | |
|---|---|---|---|
| 101 | Basic Plan | 50 | 1 |
| 102 | Premium Plan | 100 | 3 |

## Going forward, I used a 'MERGE' command to check if a member exists in the table 'MEMBERS_OF_GYM.
## If they exist, their phone number is updated.
## If they do not exist, their phone number is added.

```
--using MERGE to check if a members exists in the table MEMBERS_OF_GYM
--if they exist, their phone number is updated
--if not, their phone number is added
MERGE INTO MEMBERS_OF_GYM m
USING (
    SELECT 6 AS MemberID, 'Daniel Miller' AS FullName, 'danielmiller@gmail.com' AS Email,
           '0' AS PhoneNumber, TO_DATE('2003-02-20', 'YYYY-MM-DD') AS DateOfBirth,
           106 AS SubscriptionID FROM DUAL
) src
ON (m.MemberID = src.MemberID)
WHEN MATCHED THEN
    UPDATE SET m.PhoneNumber = src.PhoneNumber
WHEN NOT MATCHED THEN
    INSERT (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID, JOIN_DATE)
    VALUES (src.MemberID, src.FullName, src.Email, src.PhoneNumber, src.DateOfBirth, src.SubscriptionID, SYSDATE);
```

## Examples on the next slide

Let's check the first case, namely if the member already exists in the database.
As we saw in the previous slide, we are looking for 'Daniel Miller'. Since he already exists in the database, we will update his phone number to '0' for testing purposes

| | MEMBE... | FULLNAME | EMAIL | PHONENUMBER | DATEOFBIRTH | SUBSCRIPTIONID | JOIN_DATE | ADDRESS |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Michael Doe | michaeldoe@gmail.com | 0737217263 | 10-MAY-90 | 1 | 18-JAN-25 | (null) |
| 2 | 2 | Mike Johnson | mike.johnson@gmail.com | 0987654321 | 20-JUL-95 | 102 | 22-JAN-25 | (null) |
| 3 | 3 | Alice Matthews | alicematthwes@gmail.com | 1234098765 | 25-OCT-98 | 103 | 21-JAN-25 | (null) |
| 4 | 4 | Andrew Joe | andrewjoe@gmail.com | 5432167890 | 04-NOV-99 | 104 | 21-JAN-25 | (null) |
| 5 | 5 | Emily Davis | emily.davis@gmail.com | 111111111 | 18-MAR-92 | 101 | 20-JAN-25 | (null) |
| 6 | 6 | Daniel Miller | danielmiller@gmail.com | 0 | 20-FEB-03 | 106 | 22-JAN-25 | (null) |
| 7 | 7 | Emma Martinez | emma.martinez@gmail.com | 0436586978 | 25-OCT-98 | 107 | 22-JAN-25 | (null) |
| 8 | 8 | Sophia Brown | sophiabrown@gmail.com | 0475697821 | 04-NOV-99 | 108 | 22-JAN-25 | (null) |
| 9 | 9 | Michael Williams | michaelwilliams@gmail.com | 0348695869 | 04-JUN-02 | 109 | 22-JAN-25 | (null) |
| 10 | 10 | Olivia Garcia | oliviagarcia@gmail.com | 0492182738 | 04-APR-01 | 110 | 22-JAN-25 | (null) |

## Now it is time to check the second case, where a member does not exist in the database, in this case I added a 'Test Member' and set his phone number to 1

```oracle
--using MERGE to check if a members exists in the table MEMBERS_OF_GYM
--if they exist, their phone number is updated
--if not, their phone number is added
MERGE INTO MEMBERS_OF_GYM m
USING (
    SELECT 11 AS MemberID, 'Test Member' AS FullName, 'testmember@gmail.com' AS Email,
           '1' AS PhoneNumber, TO_DATE('2003-02-20', 'YYYY-MM-DD') AS DateOfBirth,
           106 AS SubscriptionID FROM DUAL
) src
ON (m.MemberID = src.MemberID)
WHEN MATCHED THEN
    UPDATE SET m.PhoneNumber = src.PhoneNumber
WHEN NOT MATCHED THEN
    INSERT (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID, JOIN_DATE)
    VALUES (src.MemberID, src.FullName, src.Email, src.PhoneNumber, src.DateOfBirth, src.SubscriptionID, SYSDATE);
```

| | MEMBE... | FULLNAME | EMAIL | PHONENUMBER | DATEOFBIRTH | SUBSCRIPTIONID | JOIN_DATE | ADDRESS |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Michael Doe | michaeldoe@gmail.com | 0737217263 | 10-MAY-90 | 1 | 18-JAN-25 | (null) |
| 2 | 2 | Mike Johnson | mike.johnson@gmail.com | 0987654321 | 20-JUL-95 | 102 | 22-JAN-25 | (null) |
| 3 | 3 | Alice Matthews | alicematthwes@gmail.com | 1234098765 | 25-OCT-98 | 103 | 21-JAN-25 | (null) |
| 4 | 4 | Andrew Joe | andrewjoe@gmail.com | 5432167890 | 04-NOV-99 | 104 | 21-JAN-25 | (null) |
| 5 | 5 | Emily Davis | emily.davis@gmail.com | 111111111 | 18-MAR-92 | 101 | 20-JAN-25 | (null) |
| 6 | 6 | Daniel Miller | danielmiller@gmail.com | 0 | 20-FEB-03 | 106 | 22-JAN-25 | (null) |
| 7 | 7 | Emma Martinez | emma.martinez@gmail.com | 0436586978 | 25-OCT-98 | 107 | 22-JAN-25 | (null) |
| 8 | 8 | Sophia Brown | sophiabrown@gmail.com | 0475697821 | 04-NOV-99 | 108 | 22-JAN-25 | (null) |
| 9 | 9 | Michael Williams | michaelwilliams@gmail.com | 0348695869 | 04-JUN-02 | 109 | 22-JAN-25 | (null) |
| 10 | 10 | Olivia Garcia | oliviagarcia@gmail.com | 0492182738 | 04-APR-01 | 110 | 22-JAN-25 | (null) |
| 11 | 11 | Test Member | testmember@gmail.com | 1 | 20-FEB-03 | 106 | 22-JAN-25 | (null) |

## Also, I forgot to add my name so I used the same MERGE function to add my name into the table

```
--using MERGE to check if a members exists in the table MEMBERS_OF_GYM
--if they exist, their phone number is updated
--if not, their phone number is added
MERGE INTO MEMBERS_OF_GYM m
USING (
    SELECT 12 AS MemberID, 'Rosu Liviu-Mihai' AS FullName, 'rosuliviu23@stud.ase.ro' AS Email,
           '0767691019' AS PhoneNumber, TO_DATE('2004-05-28', 'YYYY-MM-DD') AS DateOfBirth,
           112 AS SubscriptionID FROM DUAL
) src
ON (m.MemberID = src.MemberID)
WHEN MATCHED THEN
    UPDATE SET m.PhoneNumber = src.PhoneNumber
WHEN NOT MATCHED THEN
    INSERT (MemberID, FullName, Email, PhoneNumber, DateOfBirth, SubscriptionID, JOIN_DATE)
    VALUES (src.MemberID, src.FullName, src.Email, src.PhoneNumber, src.DateOfBirth, src.SubscriptionID, SYSDATE);
```

| | MEMBERID | FULLNAME | EMAIL | PHONENUMBER | DATEOFBIRTH | SUBSCRIPTIONID | JOIN_DATE | ADDRESS |
|---|---|---|---|---|---|---|---|---|
| 1 | 5 | Emily Davis | emily.davis@gmail.com | 0737217263 | 18-MAR-92 | 101 | 20-JAN-25 | (null) |
| 2 | 1 | Michael Doe | michaeldoe@gmail.com | 0737217263 | 10-MAY-90 | 1 | 18-JAN-25 | (null) |
| 3 | 11 | Test Member | testmember@gmail.com | 1 | 20-FEB-03 | 106 | 22-JAN-25 | (null) |
| 4 | 6 | Daniel Miller | danielmiller@gmail.com | 0 | 20-FEB-03 | 106 | 22-JAN-25 | (null) |
| 5 | 7 | Emma Martinez | emma.martinez@gmail.com | 0436586978 | 25-OCT-98 | 107 | 22-JAN-25 | (null) |
| 6 | 8 | Sophia Brown | sophiabrown@gmail.com | 0475697821 | 04-NOV-99 | 108 | 22-JAN-25 | (null) |
| 7 | 9 | Michael Williams | michaelwilliams@gmail.com | 0348695869 | 04-JUN-02 | 109 | 22-JAN-25 | (null) |
| 8 | 10 | Olivia Garcia | oliviagarcia@gmail.com | 0737217263 | 04-APR-01 | 110 | 22-JAN-25 | (null) |
| 9 | 2 | Mike Johnson | mike.johnson@gmail.com | 0987654321 | 20-JUL-95 | 2 | 22-JAN-25 | (null) |
| 10 | 3 | Alice Matthews | alicematthwes@gmail.com | 1234098765 | 25-OCT-98 | 103 | 21-JAN-25 | (null) |
| 11 | 4 | Andrew Joe | andrewjoe@gmail.com | 5432167890 | 04-NOV-99 | 104 | 21-JAN-25 | (null) |
| 12 | 12 | Rosu Liviu-Mihai | rosuliviu23@stud.ase.ro | 0767691019 | 28-MAY-04 | 112 | 23-JAN-25 | (null) |

## Next, I used 'UPDATE', to update not only the phone number of certain members of the gym but also the specialization of a trainer

```
--using UPDATE
--updating a member's phone number
UPDATE MEMBERS_OF_GYM
SET PhoneNumber = '0737217263'
WHERE MemberID = 10;

--updating trainer specialization
UPDATE TRAINERS
SET Specialization = 'Powerlifting'
WHERE TrainerID = 1;
```

| | TRAINERID | FULLNAME | SPECIALIZATION | HIREDATE |
|---|---|---|---|---|
| 1 | 1 | Mike Johnson | Powerlifting | 22-MAR-00 |
| 2 | 2 | John Michael | Losing weight | 27-MAY-98 |
| 3 | 3 | Sarah Connor | Yoga | 17-JUL-99 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 10 | 10 | Olivia Garcia | oliviagarcia@gmail.com | 0737217263 | 04-APR-01 | 110 | 22-JAN-25 | (null) |

## As we saw earlier, the member with ID 10, namely 'Olivia Garcia' had the following phone number: 0492182738 which got changed to '0737217263'.
## The same can be said about the trainer with ID 1, Mike Johnson, who was initially assigned to 'Weightlifting'. After 'UPDATE' he is now assinged to 'Powerlifting'.

TESTING UPDATE COMMAND

# DELETE COMMAND

I used some 'DELETE' commands to delete a member from the 'MEMBERS_OF_GYM' table and to also delete a subscription from the 'SUBSCRIPTION' table

MemberID = 2 on 'MEMBERS_OF_GYM' was 'Mike Johnson'
SubscriptionID = 101 on 'SUBSCRIPTION' table was 'Basic Plan', which is now deleted (I don't know why those values still appear there)

```
--using DELETE
--deleting a member of the gym from the MEMBERS_OF_GYM table
DELETE FROM MEMBERS_OF_GYM
WHERE MemberID = 2;
```

| | MEMBE... | FULLNAME | EMAIL | PHONENUMBER | DATEOFBIRTH | SUBSCRIPTIONID | JOIN_DATE | ADDRESS |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Michael Doe | michaeldoe@gmail.com | 0737217263 | 10-MAY-90 | 1 | 18-JAN-25 | (null) |
| 2 | 3 | Alice Matthews | alicematthwes@gmail.com | 1234098765 | 25-OCT-98 | 103 | 21-JAN-25 | (null) |
| 3 | 4 | Andrew Joe | andrewjoe@gmail.com | 5432167890 | 04-NOV-99 | 104 | 21-JAN-25 | (null) |
| 4 | 5 | Emily Davis | emily.davis@gmail.com | 0737217263 | 18-MAR-92 | 101 | 20-JAN-25 | (null) |
| 5 | 6 | Daniel Miller | danielmiller@gmail.com | 0 | 20-FEB-03 | 106 | 22-JAN-25 | (null) |
| 6 | 7 | Emma Martinez | emma.martinez@gmail.com | 0436586978 | 25-OCT-98 | 107 | 22-JAN-25 | (null) |
| 7 | 8 | Sophia Brown | sophiabrown@gmail.com | 0475697821 | 04-NOV-99 | 108 | 22-JAN-25 | (null) |
| 8 | 9 | Michael Williams | michaelwilliams@gmail.com | 0348695869 | 04-JUN-02 | 109 | 22-JAN-25 | (null) |
| 9 | 10 | Olivia Garcia | oliviagarcia@gmail.com | 0737217263 | 04-APR-01 | 110 | 22-JAN-25 | (null) |
| 10 | 11 | Test Member | testmember@gmail.com | 1 | 20-FEB-03 | 106 | 22-JAN-25 | (null) |

```
--deleting a subscription from the SUBSCRIPTION table
DELETE FROM SUBSCRIPTION
WHERE SubscriptionID = 101;
```

| | SUBSCRIPTIO... | PLANNAME | COST | DURATIONMONTHS |
|---|---|---|---|---|
| 1 | 1 | Basic Plan | 50 | 1 |
| 2 | 2 | Premium Plan | 100 | 3 |
| 3 | 102 | Premium Plan | 100 | 3 |
| 4 | 103 | Basic Plan | 50 | 1 |
| 5 | 104 | Premium Plan | 100 | 3 |

## I used some 'SELECT * FROM' to display the information that exists in the tables

```
SELECT * FROM MEMBERS_OF_GYM;
SELECT * FROM TRAINERS;
SELECT * FROM TRAINING_SESSION;
SELECT * FROM EQUIPMENT;
SELECT * FROM SUBSCRIPTION;
```

## We display all training sessions scheduled after a specific date

```
SELECT TrainingSessionName, Schedule
FROM TRAINING_SESSION
WHERE Schedule > TO_DATE('2025-01-01', 'YYYY-MM-DD')
```

| | TRAININGSESSIONNAME | SCHEDULE |
|---|---|---|
| 1 | Morning Cardio | 10-MAY-25 |
| 2 | Strength training | 11-MAY-25 |
| 3 | Pilates | 12-MAY-25 |

```
SELECT FullName, HireDate
FROM Trainers
WHERE HireDate <= TO_DATE('2000-01-01', 'YYYY-MM-DD');
```

| | FULLNAME | HIREDATE |
|---|---|---|
| 1 | John Michael | 27-MAY-98 |
| 2 | Sarah Connor | 17-JUL-99 |

```
SELECT FullName
FROM TRAINERS
WHERE Specialization = 'Powerlifting';
```

| | FULLNAME |
|---|---|
| 1 | Mike Johnson |

This SQL query retrieves gym members' full names and their subscription status, classifying them as "Basic," "Premium," or "No subscription" based on cost. It uses a `LEFT JOIN` to combine data from the `MEMBERS_OF_GYM` and `SUBSCRIPTION` tables.

```sql
SELECT FullName,
       CASE
           WHEN Cost <= 50 THEN 'Basic'
           WHEN Cost > 50 THEN 'Premium'
           ELSE 'No subscription'
       END AS SubscriptionStatus
FROM MEMBERS_OF_GYM m
LEFT JOIN SUBSCRIPTION s ON m.SubscriptionID = s.SubscriptionID;
```

| | FULLNAME | SUBSCRIPTIONSTATUS |
|---|---|---|
| 1 | Mike Johnson | Premium |
| 2 | Michael Doe | Basic |
| 3 | Alice Matthews | Basic |
| 4 | Andrew Joe | Premium |
| 5 | Emma Martinez | No subscription |
| 6 | Sophia Brown | No subscription |
| 7 | Michael Williams | No subscription |
| 8 | Olivia Garcia | No subscription |
| 9 | Emily Davis | No subscription |
| 10 | Rosu Liviu-Mihai | No subscription |
| 11 | Test Member | No subscription |
| 12 | Daniel Miller | No subscription |

# Thank you for your time!

ENDING