



## **EE4350 : DATABASE SYSTEMS**

**MINI PROJECT – Gym Management System**

### **Group Members**

EG/2021/4676 – MIHIKALPA S.A.S.

EG/2021/4680 – MUNSSINGHE B.P.R.D.

EG/2021/4676 – MUNSIF M.F.A.

# **Contents**

1	Requirement Analysis	5
2	Conceptual Design	7
2.1	ER Diagram	7
2.2	UML Class Diagram	8
3	Implementation	9
3.1	Schema creation	9
3.2	Table definitions	9
3.3	Inserting data	19
3.4	Update and Delete operations	32
4	Transaction	38
5	Tunning	48

## List of Figures

Figure 2.1.1 : ER Diagram for the GYM Management System.....	7
Figure 2.2.1 : UML Class Diagram for the GYM Management System .....	8
Figure 3.1.1 : Creating Database .....	9
Figure 3.2.1 : Creating Manager table.....	9
Figure 3.2.2 : Creating MANAGER_PHONE table .....	10
Figure 3.2.3 : Creating BRANCH table .....	10
Figure 3.2.4 : Creating MEMBER table .....	11
Figure 3.2.5 : Creating Member_4n_Numbers table.....	11
Figure 3.2.6 : Creating MEMBERSHIP_TYPE table.....	12
Figure 3.2.7 : Creating TRAINER table .....	12
Figure 3.2.8 : Creating Trainer_4n_Numbers table .....	13
Figure 3.2.9 : Creating EQUIPMENT table.....	13
Figure 3.2.10 : Creating Equipment_branchAndUnits table.....	14
Figure 3.2.11 : Creating Equipment_repair_history table.....	14
Figure 3.2.12 : Creating SCHEDULE table .....	15
Figure 3.2.13 : Creating PAYMENT table .....	15
Figure 3.2.14 : Adding foreign keys to MANAGER,MANAGER_PHONE, MEMBER tables .....	16
Figure 3.2.15 : Adding foreign keys to Member_4n_Numbers, TRAINER, Trainer_4n_Numbers tables.....	16
Figure 3.2.16 : Adding foreign keys to Equipment_branchAndUnits, Equipment_repair_history, SCHEDULE_, PAYMENT tables .....	17
Figure 3.2.17 : Setting trigger to Update Trainer Salary .....	17
Figure 3.2.18 : Setting trigger to Enforce Minimum Salary .....	18
Figure 3.3.1 : Inserting Data to the Branch table .....	19
Figure 3.3.2 : Branch table with Inserted Data .....	19
Figure 3.3.3 : Inserting Data to the MANAGER table.....	20
Figure 3.3.4 : Manager table with Inserted Data.....	20
Figure 3.3.5 : Inserting Data to the MANAGER_PHONE table .....	21
Figure 3.3.6 : Manager_Phone table with Inserted Data.....	21
Figure 3.3.7 : Inserting Data to the EQUIPMENT table.....	22
Figure 3.3.8 : EQUIPMENT table with Inserted Data .....	22
Figure 3.3.9 : Inserting Data to the Equipment_branchAndUnits table.....	23
Figure 3.3.10 : EQUIPMENT_BranchAndUnits table with Inserted Data.....	23
Figure 3.3.11 : Inserting Data to the Equipment_repair_history table .....	24
Figure 3.3.12 : EQUIPMENT_REPAIR_HISTORY table with Inserted Data .....	24
Figure 3.3.13 : Inserting Data to the MEMBERSHIP_TYPE table.....	25
Figure 3.3.14 : MEMBERSHIP_TYPE table with Inserted Data .....	25
Figure 3.3.15 : Inserting Data to the TRAINER table .....	26
Figure 3.3.16 : Trainer table with Inserted Data .....	26
Figure 3.3.17 : Inserting Data to the Trainer_4n_Numbers table .....	27
Figure 3.3.18 : TRAINER_4n_Numbers table with Inserted Data .....	27
Figure 3.3.19 : Inserting Data to the MEMBER table .....	28
Figure 3.3.20 : Member table with Inserted Data .....	28
Figure 3.3.21 : Inserting Data to the Membr_4n_Numbers table .....	29
Figure 3.3.22 : Member_4n_Numbers table with Inserted Data.....	29
Figure 3.3.23 : Inserting Data to the SCHEDULE_ table .....	30
Figure 3.3.24 : Schedule_ table with Inserted Data .....	30
Figure 3.3.25 : Inserting Data to the PAYMENT table .....	31
Figure 3.3.26 Payment table with Inserted Data .....	31

Figure 3.4.1 : Update and Delete data from the Member table .....	32
Figure 3.4.2 : Update and Delete data from the Member_4n_Numbers table .....	32
Figure 3.4.3 : Update and Delete data from the Trainer table .....	33
Figure 3.4.4 : Update and Delete data from the Branch table .....	33
Figure 3.4.5 : Update and Delete data from the Manager table .....	34
Figure 3.4.6 : Update and Delete data from the MANAGER_PHONE table .....	34
Figure 3.4.7 : Update and Delete data from the SCHEDULE table.....	35
Figure 3.4.8 : Update and Delete data from the Payment table .....	35
Figure 3.4.9 : Update and Delete data from the Membership_Type table .....	36
Figure 3.4.10 : Update and Delete data from the Equipment table .....	36
Figure 3.4.11 : Update and Delete data from the Equiment_branchAndUnits table .....	37
Figure 3.4.12 : Update and Delete data from the Equipment_repair_history table .....	37
Figure 3.4.1 : Query to Select all members who are from Galle.....	38
Figure 3.4.2 : Simple Select query to display selected columns .....	38
Figure 3.4.3 : Cartesian Product Operation .....	39
Figure 3.4.4 : Query to Create a User View .....	39
Figure 3.4.5 : Query to Rename Operation .....	40
Figure 3.4.6 : Use of Aggregation Function.....	40
Figure 3.4.7 : Use of LIKE Keyword.....	41
Figure 3.4.8 : Union of Managers and Trainers into Employee.....	41
Figure 3.4.9 : Intersection of Member and Schedule tables .....	42
Figure 3.4.10 : Finding the members who doesn't have the workout plan yet.....	42
Figure 3.4.11 : Query for finding the average member per branch.....	43
Figure 3.4.12 : Inner Join of Tables of member and their trainer names in one table .....	43
Figure 3.4.13 : Natural join to view Members having trainers who have more than 5 years of experience and from Galle .....	44
Figure 3.4.14 : Left Outer join of Branch and Associated Managers .....	44
Figure 3.4.15 : Right Outer Join of Trainer and Branch .....	45
Figure 3.4.16 : Full Outer Join of Member and Payment .....	45
Figure 3.4.17 : Outer Union of Membership_Type and Schedule .....	46
Figure 3.4.18 : Returning details of members and trainers who are both located in Colombo using nested query.....	46
Figure 3.4.19 : The Members Who Have Membership Type As Silver And Are Assigned To A Trainer With More Than 4 Years Of Experience .....	47
Figure 3.4.20 : The Total Amount Earned from Payments Made By Members Who Have A Membership Type With A Price Higher Than The Average Price Of All Membership Types.....	47
Figure 3.4.1 : Before tunned and after tuned for query 9 .....	48
Figure 3.4.2 : Before tunned and after tuned for query 10.....	48
Figure 3.4.3 : Before tunned and after tuned for query 11 .....	49
Figure 3.4.4 : Before tunned and after tuned for query 13.....	49
Figure 3.4.5 : Before tunned and after tuned for query 14.....	50
Figure 3.4.6 : Before tunned and after tuned for query 15 .....	50
Figure 3.4.7 : Before tunned and after tuned for query 16.....	51
Figure 3.4.8 : : Before tunned and after tuned for query 17.....	51
Figure 3.4.9 : Before tunned and after tuned for query 18.....	52
Figure 3.4.10 : Before tunned and after tuned for query 19 .....	52

# 1 Requirement Analysis

Requirement analysis plays a crucial part in the development of any database. In this case, the database development of a gym management system, the requirement analysis can be divided in to 4 parts.

## 1. Stakeholder identification

Identifying all the stakeholder characters including managers, trainers and members.

## 2. Gathering requirements

Gathering all the information from the stakeholders to identify their needs and the requirements from the system.

- Functional requirements
  - Member management
  - Schedule management
  - Billing and Payment
  - Inventory management
- Data requirements
  - Entity-Manager
    - Manager\_ID
    - Phone Numbers
    - First name
    - Last name
  - Entity-Branch
    - Branch\_Num
    - City
    - Town
    - Opening hours
    - Capacity
    - Street\_Name
    - Building\_Num
  - Entity-Trainer
    - Trainer\_ID
    - First name
    - Last name
    - Salary
    - Experience\_yrs
    - Phone Numbers
    - House\_Num
    - Street\_Name
    - Town
    - City

- Entity-Member
  - Member\_ID
  - First name
  - Last name
  - Phone numbers
  - House \_Num
  - Street\_Name
  - Town
  - City
- Entity-Schedule
  - Date
  - Start\_time
  - Duration\_hrs
- Entity-Payment
  - Time
  - Date
  - Cost
- Entity-Membership\_type
  - Type\_ID
  - Name
  - Description
  - Time\_Period
- Entity-Equipment
  - Number\_of\_units
  - Name
  - Cost
  - Repair History

- Non-functional requirements
  - Security
  - Performance
  - Stability
  - Accessibility

### **3. Documentation**

Documenting all the gathered requirements and then getting reviewed and approved by the necessary parties.

### **4. Feedback and iteration**

Gather feedback during the development process and incorporating the requirements as needed.

## 2 Conceptual Design

Conceptual design is a visual representation of the database that shows the entities, relationships, attributes and data constraints. Both ER diagram and UML class diagram that has shown below is representing the database schema of a gym management system. Both diagrams have been design using Visual Paradigm.

### 2.1 ER Diagram

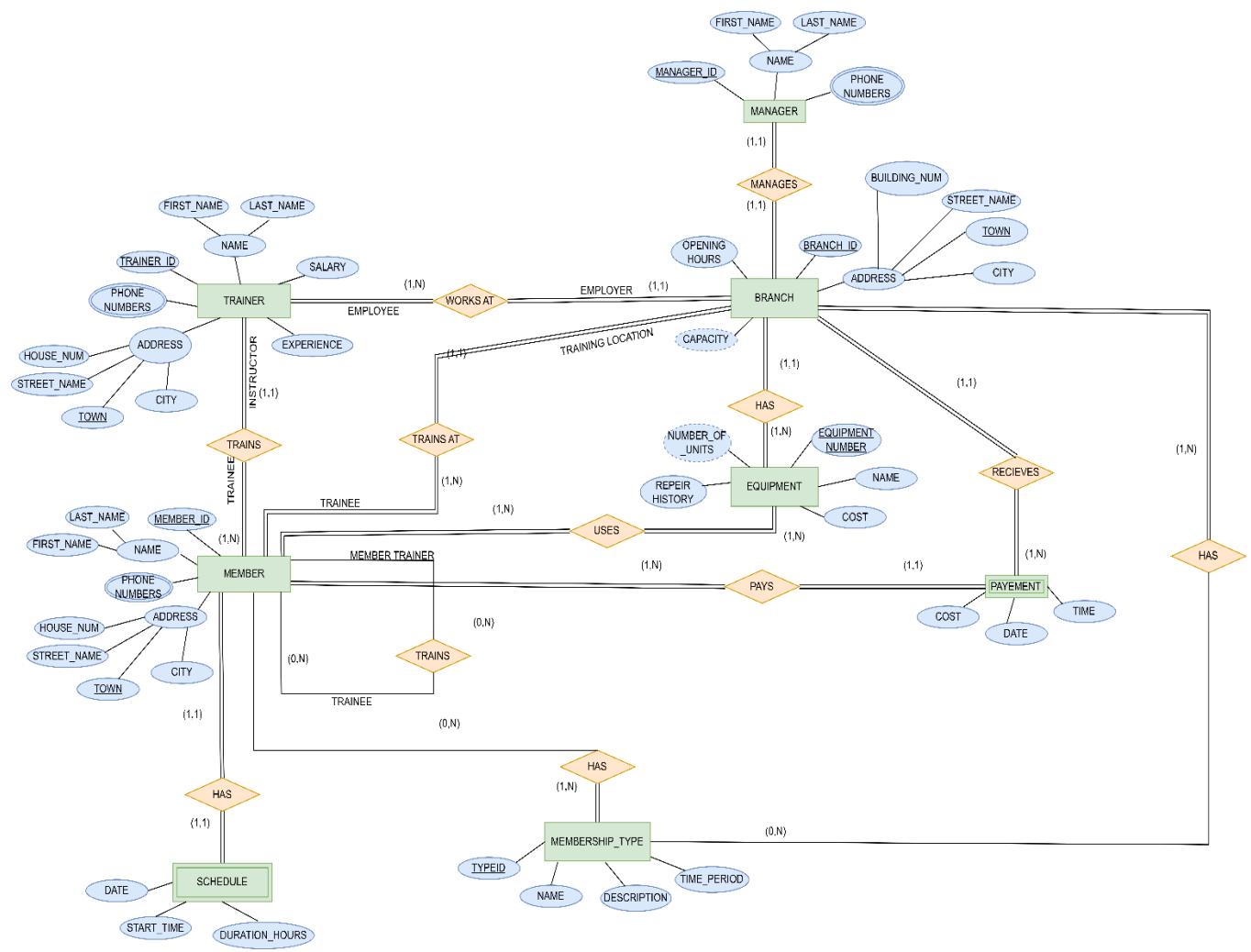


Figure 2.1.1 : ER Diagram for the GYM Management System

## 2.2 UML Class Diagram

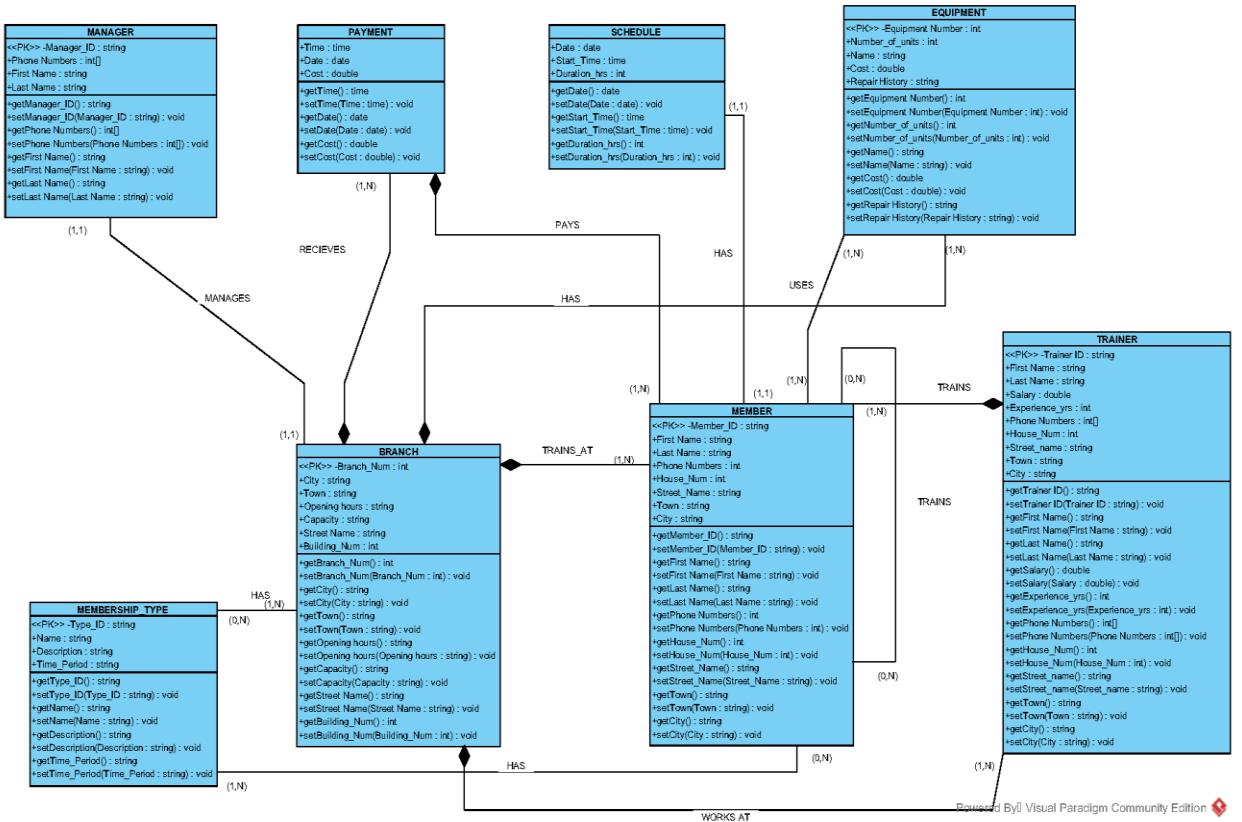
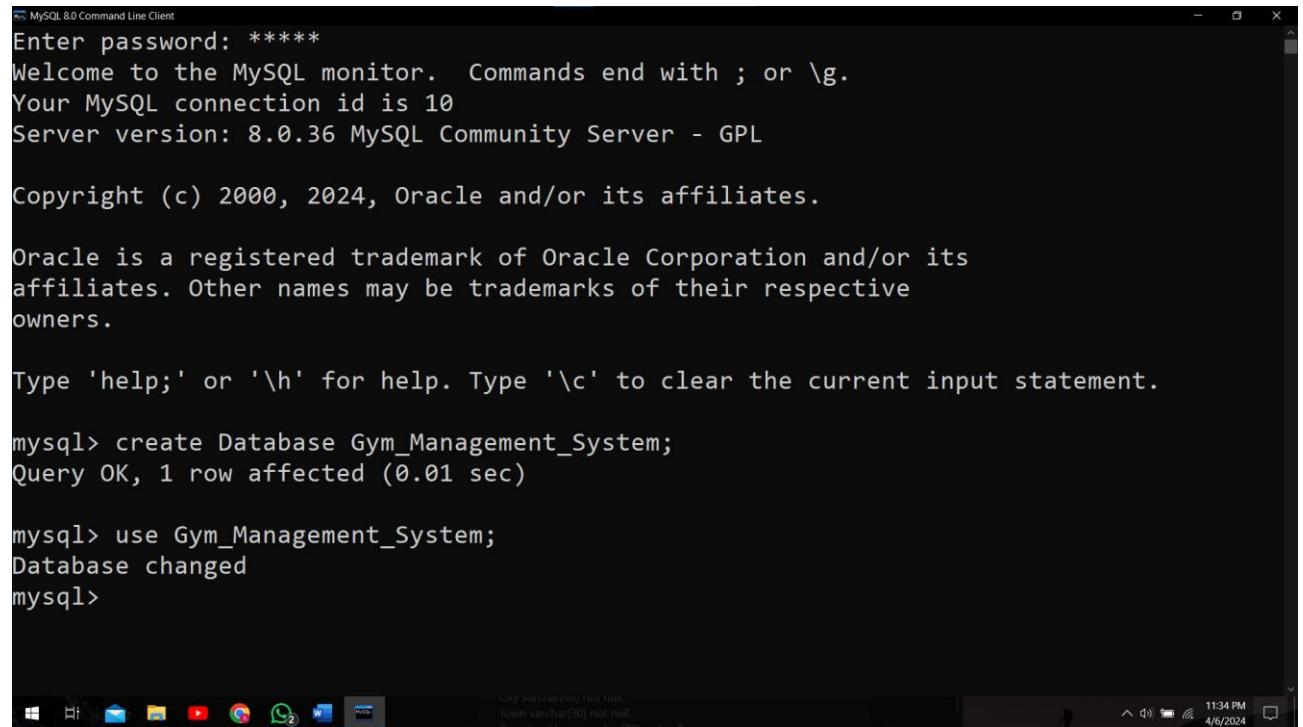


Figure 2.2.1 : UML Class Diagram for the GYM Management System

### 3 Implementation

#### 3.1 Schema creation



The screenshot shows the MySQL 8.0 Command Line Client window. It displays the following text:

```
MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

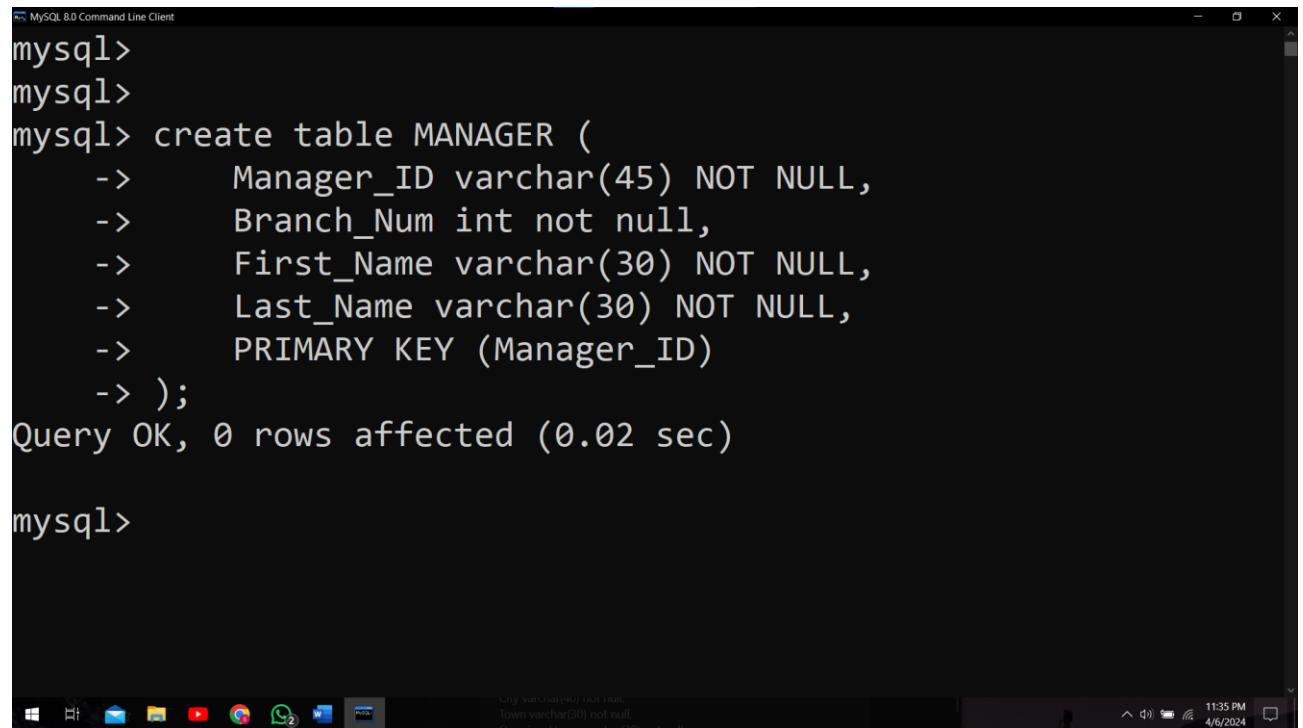
mysql> create Database Gym_Management_System;
Query OK, 1 row affected (0.01 sec)

mysql> use Gym_Management_System;
Database changed
mysql>
```

The taskbar at the bottom shows various icons, and the system tray indicates the date and time as 11:34 PM on 4/6/2024.

Figure 3.1.1 : Creating Database

#### 3.2 Table definitions



The screenshot shows the MySQL 8.0 Command Line Client window. It displays the following text:

```
MySQL 8.0 Command Line Client
mysql>
mysql>
mysql> create table MANAGER (
    ->     Manager_ID varchar(45) NOT NULL,
    ->     Branch_Num int not null,
    ->     First_Name varchar(30) NOT NULL,
    ->     Last_Name varchar(30) NOT NULL,
    ->     PRIMARY KEY (Manager_ID)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
```

The taskbar at the bottom shows various icons, and the system tray indicates the date and time as 11:35 PM on 4/6/2024.

Figure 3.2.1 : Creating Manager table

```
MySQL 8.0 Command Line Client
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql>
mysql> create table MANAGER_PHONE (
    ->     Manager_ID varchar(45) NOT NULL,
    ->     Phone_Number int NOT NULL,
    ->     FOREIGN KEY (Manager_ID) REFERENCES MANAGER(Manager_
ID)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql>
```

Figure 3.2.2 : Creating MANAGER\_PHONE table

```
MySQL 8.0 Command Line Client
mysql>
mysql> create table BRANCH(
    -> Branch_Num int not null,
    ->     City varchar(40) not null,
    ->     Town varchar(30) not null,
    ->     Opening_Hours varchar(30) not null,
    ->     Capacity varchar(30) not null,
    ->     Street_Name varchar(30) not null,
    ->     Building_Num int not null,
    ->     primary key (Branch_Num)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
```

Figure 3.2.3 : Creating BRANCH table

```
MySQL 8.0 Command Line Client
mysql>
mysql>
mysql> create table MEMBER (
->     Member_ID varchar(30) NOT NULL,
->     Trainer_ID varchar(45) not null,
-> MemType_ID varchar(45) not null,
->     First_Name varchar(45) NOT NULL,
->     Last_Name varchar(45),
->     House_Num int,
->     Street_Name varchar(45) NOT NULL,
->     City varchar(30),
->     primary key (Member_ID)
-> );
Query OK, 0 rows affected (0.06 sec)

mysql>
```

Figure 3.2.4 : Creating MEMBER table

```
MySQL 8.0 Command Line Client
mysql>
mysql>
mysql>
mysql> create table Member_4n_Numbers (
->     Member_ID VARCHAR(30) NOT NULL,
->     Phone_Number INT NOT NULL,
->     FOREIGN KEY (Member_ID) REFERENCES MEMBER(Member_ID),
->     primary key (Member_ID, Phone_Number)
-> );
Query OK, 0 rows affected (0.03 sec)

mysql>
```

Figure 3.2.5 : Creating Member\_4n\_Numbers table

```
MySQL 8.0 Command Line Client
mysql>
mysql>
mysql> create table MEMBERSHIP_TYPE(
    -> MemType_ID varchar(45) not null,
    -> Type_Name varchar(45) not null,
    -> Description_ varchar(45) not null,
    -> Time_Period varchar(45) not null,
    -> price float not null,
    -> primary key(Memtype_id)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
```

Figure 3.2.6 : Creating MEMBERSHIP\_TYPE table

```
MySQL 8.0 Command Line Client
mysql>
mysql>
mysql> create table TRAINER(
    -> Trainer_ID varchar(45) not null,
    -> Branch_Num int not null,
    -> First_Name varchar(45) not null,
    -> Last_Name varchar(45) not null,
    -> Salary float not null,
    -> Experience_yrs int not null,
    -> Street_Name varchar(45) not null,
    -> City varchar(30) not null,
    -> primary key(Trainer_ID)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
```

Figure 3.2.7 : Creating TRAINER table

```
MySQL 8.0 Command Line Client
mysql>
mysql>
mysql> create table Trainer_4n_Numbers (
    ->     Trainer_ID varchar(30) not null,
    ->     Phone_Number int not null,
    ->     FOREIGN KEY (Trainer_ID) REFERENCES TRAINER(Trainer_ID),
    ->     primary key (Trainer_ID, Phone_Number)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
```

Figure 3.2.8 : Creating Trainer\_4n\_Numbers table

```
MySQL 8.0 Command Line Client
mysql> create table EQUIPMENT(
    ->     Equipment_Number int not null,
    ->     Equipment_Name varchar(30) not null,
    ->     Cost float not null,
    ->     primary key(Equipment_Number)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
```

Figure 3.2.9 : Creating EQUIPMENT table

```
MySQL 8.0 Command Line Client
mysql>
mysql>
mysql>
mysql> create table Equipment_branchAndUnits(
    ->     Equipment_Number int not null,
    ->     Branch_Num int not null,
    ->     Number_of_Units int not null,
    ->     FOREIGN KEY (equipment_number) REFERENCES equipment(
equipment_number),
    ->     primary key(Equipment_number,branch_num,number_of_units)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
```

Figure 3.2.10 : Creating Equipment\_branchAndUnits table

```
MySQL 8.0 Command Line Client
Query OK, 0 rows affected (0.03 sec)

mysql> create table Equipment_repair_history(
    ->     Equipment_number int not null,
    ->     repair_history varchar(30) not null,
    ->     FOREIGN KEY (equipment_number) REFERENCES equipment(
equipment_number),
    ->     primary key(Equipment_number,repair_history)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
```

Figure 3.2.11 : Creating Equipment\_repair\_history table

```
MySQL 8.0 Command Line Client
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql>
mysql> create table SCHEDULE_(
    -> Date_ date not null,
    -> Start_Time time not null,
    -> Duration_hrs float not null,
    -> Member_ID varchar(30) NOT NULL,
    -> primary key(Date_,start_time,Duration_hrs)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
```

Figure 3.2.12 : Creating SCHEDULE table

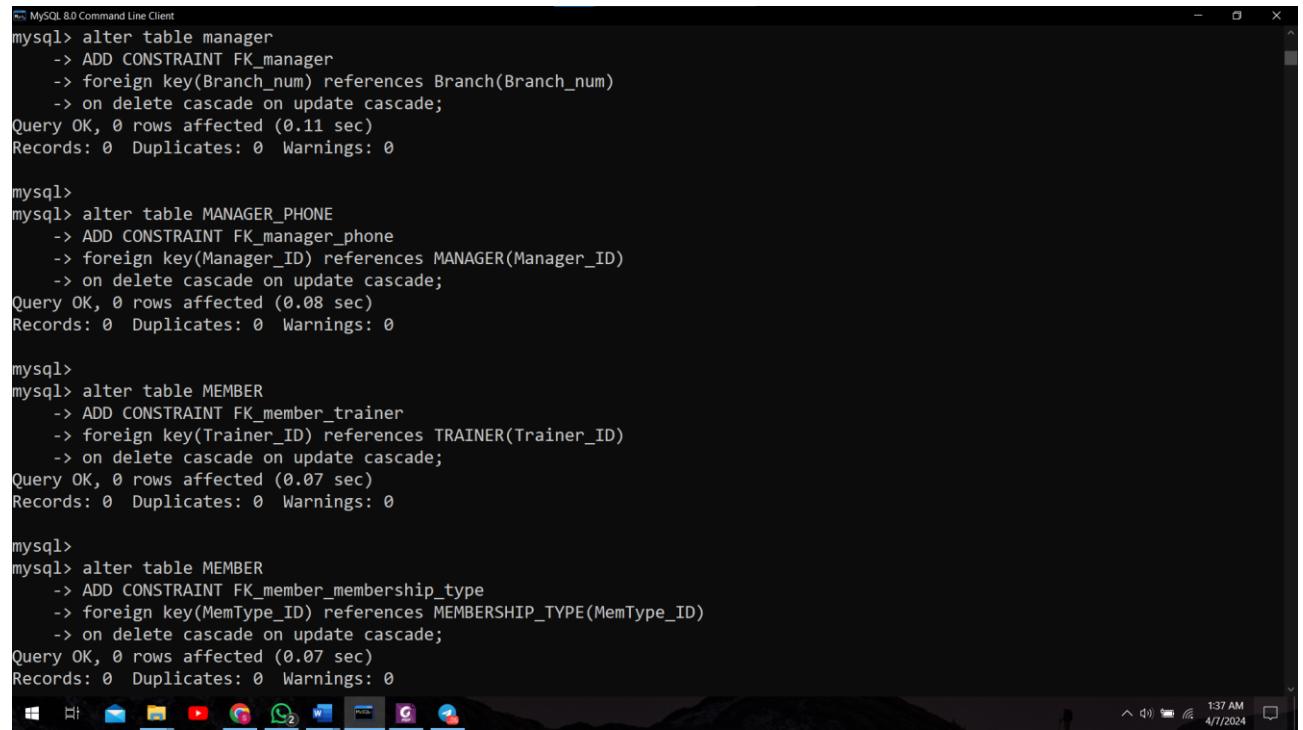
```
MySQL 8.0 Command Line Client
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql>
mysql> create table PAYMENT(
    -> Member_ID varchar(30) NOT NULL,
    -> Time_ time ,
    -> Date_ date not null,
    -> amount float not null,
    -> primary key(Time_,Date_,amount)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
```

Figure 3.2.13 : Creating PAYMENT table

## Adding Foreign Keys



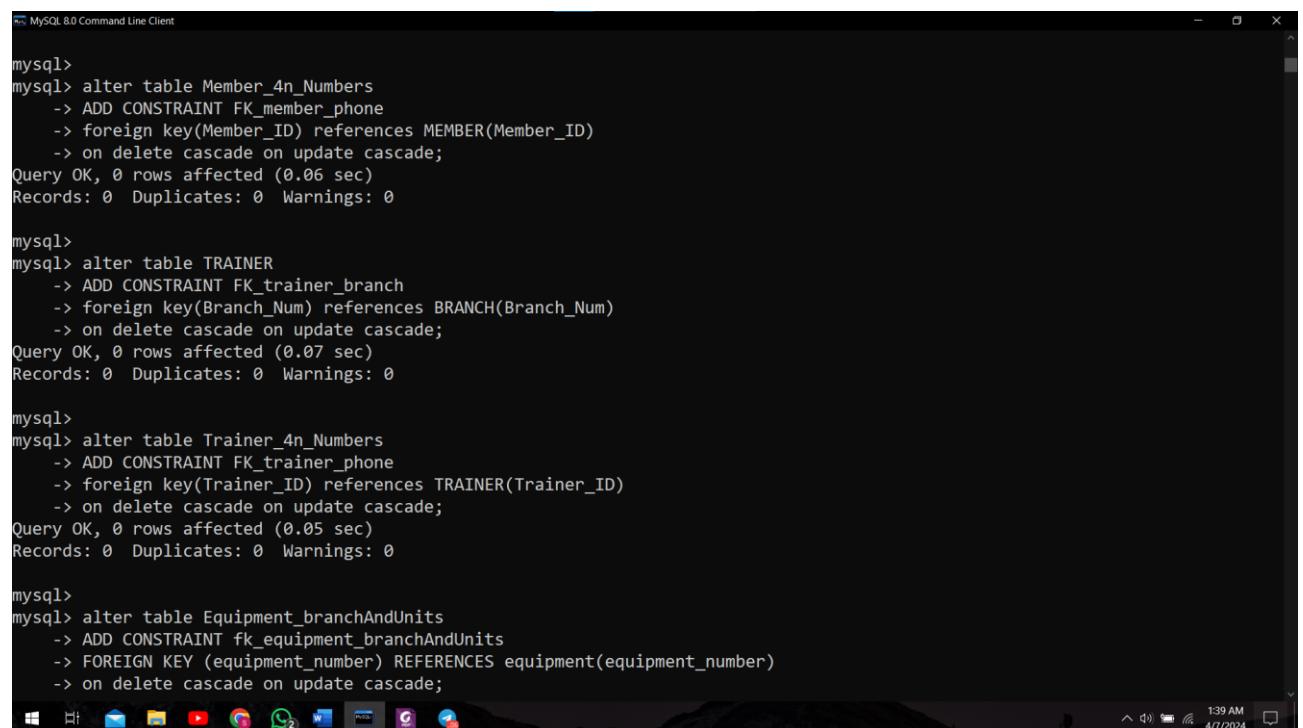
```
MySQL 8.0 Command Line Client
mysql> alter table manager
-> ADD CONSTRAINT FK_manager
-> foreign key(Branch_num) references Branch(Branch_num)
-> on delete cascade on update cascade;
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> alter table MANAGER_PHONE
-> ADD CONSTRAINT FK_manager_phone
-> foreign key(Manager_ID) references MANAGER(Manager_ID)
-> on delete cascade on update cascade;
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> alter table MEMBER
-> ADD CONSTRAINT FK_member_trainer
-> foreign key(Trainer_ID) references TRAINER(Trainer_ID)
-> on delete cascade on update cascade;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> alter table MEMBER
-> ADD CONSTRAINT FK_member_membership_type
-> foreign key(MemType_ID) references MEMBERSHIP_TYPE(MemType_ID)
-> on delete cascade on update cascade;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Figure 3.2.14 : Adding foreign keys to MANAGER,MANAGER\_PHONE, MEMBER tables



```
MySQL 8.0 Command Line Client
mysql>
mysql> alter table Member_4n_Numbers
-> ADD CONSTRAINT FK_member_phone
-> foreign key(Member_ID) references MEMBER(Member_ID)
-> on delete cascade on update cascade;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> alter table TRAINER
-> ADD CONSTRAINT FK_trainer_branch
-> foreign key(Branch_Num) references BRANCH(Branch_Num)
-> on delete cascade on update cascade;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> alter table Trainer_4n_Numbers
-> ADD CONSTRAINT FK_trainer_phone
-> foreign key(Trainer_ID) references TRAINER(Trainer_ID)
-> on delete cascade on update cascade;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> alter table Equipment_branchAndUnits
-> ADD CONSTRAINT fk_equipment_branchAndUnits
-> FOREIGN KEY (equipment_number) REFERENCES equipment(equipment_number)
-> on delete cascade on update cascade;
```

Figure 3.2.15 : Adding foreign keys to Member\_4n\_Numbers, TRAINER, Trainer\_4n\_Numbers tables

```
MySQL 8.0 Command Line Client
mysql> alter table Equipment_branchAndUnits
-> ADD CONSTRAINT fk_equipment_branchAndUnits
-> FOREIGN KEY (equipment_number) REFERENCES equipment(equipment_number)
-> on delete cascade on update cascade;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> alter table Equipment_repair_history
-> ADD CONSTRAINT fk_equipment_repair_history
-> FOREIGN KEY (equipment_number) REFERENCES equipment(equipment_number)
-> on delete cascade on update cascade;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> alter table SCHEDULE_
-> ADD CONSTRAINT FK_schedule_member
-> foreign key(Member_ID) references MEMBER(Member_ID)
-> on delete cascade on update cascade;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> alter table PAYMENT
-> ADD CONSTRAINT FK_payment_Member
-> foreign key(Member_ID) references MEMBER(Member_ID)
-> on delete cascade on update cascade;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

Figure 3.2.16 : Adding foreign keys to Equipment\_branchAndUnits, Equipment\_repair\_history, SCHEDULE\_, PAYMENT tables

## Setting triggers

```
MySQL 8.0 Command Line Client
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> DELIMITER $
mysql> CREATE TRIGGER UpdatedTrainerSalary
-> BEFORE UPDATE ON TRAINER
-> FOR EACH ROW
-> BEGIN
->     DECLARE experience_bonus INT;
->     SET experience_bonus = NEW.Experience_yrs * 1000; -- Assuming Rs.1000 bonus per year of experience
->     SET NEW.Salary = NEW.Salary + experience_bonus;
-> END;
-> $
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql>
```

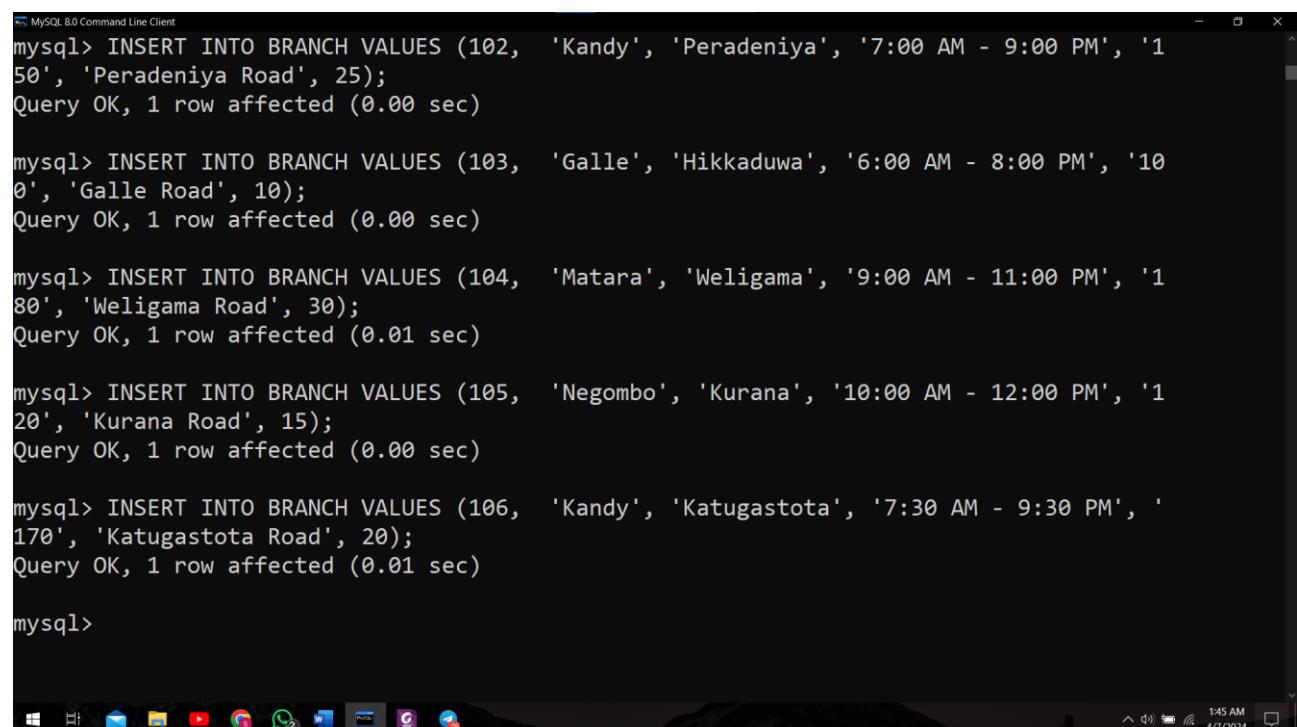
Figure 3.2.17 : Setting trigger to Update Trainer Salary

```
MySQL 8.0 Command Line Client
mysql> DELIMITER $
mysql> CREATE TRIGGER EnforceMinSalary
    -> BEFORE UPDATE ON TRAINER
    -> FOR EACH ROW
    -> BEGIN
    ->     IF NEW.Salary < OLD.Salary THEN
    ->         SIGNAL SQLSTATE '45000'
    ->         SET MESSAGE_TEXT = 'Trainer salary cannot be decreased.';
    ->     END IF;
    -> END;
    -> $
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql>
```

Figure 3.2.18 : Setting trigger to Enforce Minimum Salary

### 3.3 Inserting data



The screenshot shows the MySQL 8.0 Command Line Client window. The user has run several `INSERT INTO BRANCH` statements to add new records to the `BRANCH` table. The terminal output is as follows:

```
mysql> INSERT INTO BRANCH VALUES (102, 'Kandy', 'Peradeniya', '7:00 AM - 9:00 PM', '150', 'Peradeniya Road', 25);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO BRANCH VALUES (103, 'Galle', 'Hikkaduwa', '6:00 AM - 8:00 PM', '100', 'Galle Road', 10);
Query OK, 1 row affected (0.00 sec)

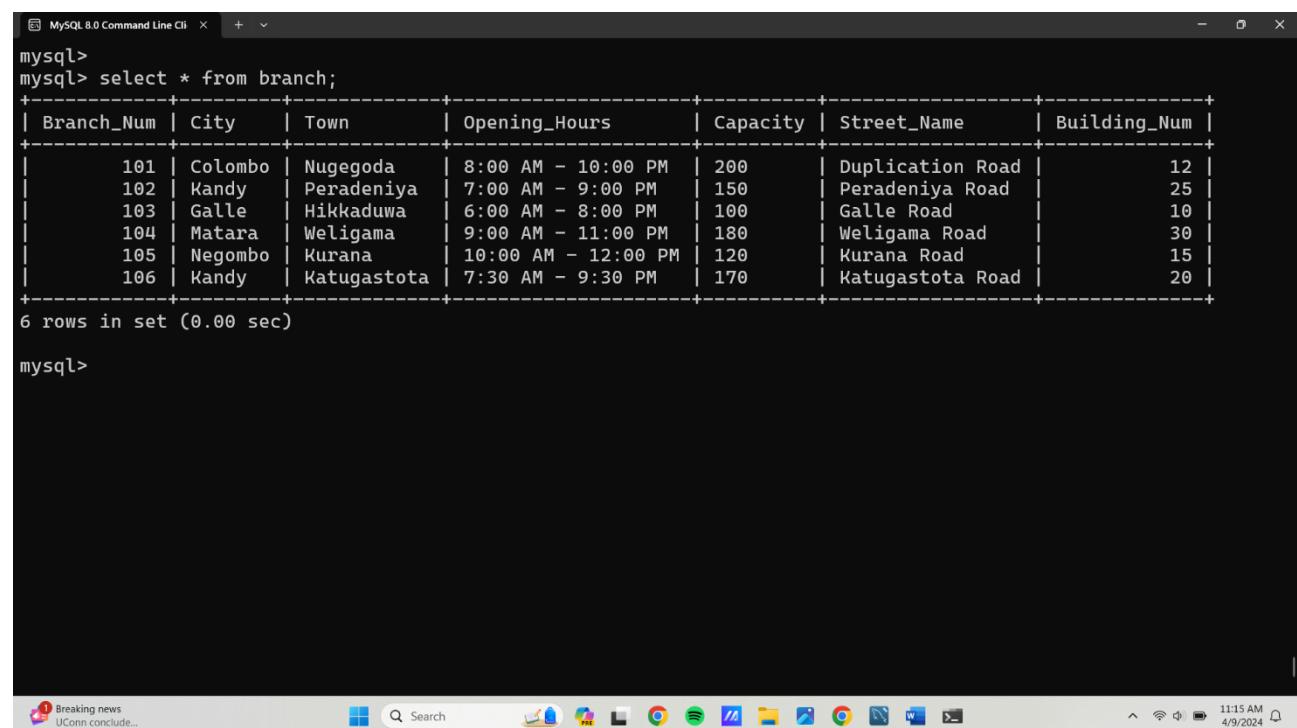
mysql> INSERT INTO BRANCH VALUES (104, 'Matara', 'Weligama', '9:00 AM - 11:00 PM', '180', 'Weligama Road', 30);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO BRANCH VALUES (105, 'Negombo', 'Kurana', '10:00 AM - 12:00 PM', '120', 'Kurana Road', 15);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO BRANCH VALUES (106, 'Kandy', 'Katugastota', '7:30 AM - 9:30 PM', '170', 'Katugastota Road', 20);
Query OK, 1 row affected (0.01 sec)

mysql>
```

Figure 3.3.1 : Inserting Data to the Branch table



The screenshot shows the MySQL 8.0 Command Line Client window. The user has run a `SELECT * FROM branch;` query to view the data inserted into the `BRANCH` table. The terminal output is as follows:

```
mysql>
mysql> select * from branch;
+-----+-----+-----+-----+-----+-----+-----+
| Branch_Num | City      | Town      | Opening_Hours | Capacity | Street_Name | Building_Num |
+-----+-----+-----+-----+-----+-----+-----+
|     101 | Colombo   | Nugegoda  | 8:00 AM - 10:00 PM |    200 | Duplication Road |        12 |
|     102 | Kandy      | Peradeniya | 7:00 AM - 9:00 PM  |    150 | Peradeniya Road |        25 |
|     103 | Galle       | Hikkaduwa | 6:00 AM - 8:00 PM  |    100 | Galle Road      |        10 |
|     104 | Matara     | Weligama   | 9:00 AM - 11:00 PM |    180 | Weligama Road   |        30 |
|     105 | Negombo    | Kurana     | 10:00 AM - 12:00 PM |    120 | Kurana Road     |        15 |
|     106 | Kandy      | Katugastota | 7:30 AM - 9:30 PM  |    170 | Katugastota Road |        20 |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 3.3.2 : Branch table with Inserted Data

```
MySQL 8.0 Command Line Client
mysql> INSERT INTO MANAGER VALUES ('MAN001', 101, 'John', 'Doe');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MANAGER VALUES ('MAN002', 102, 'Jane', 'Smith');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO MANAGER VALUES ('MAN003', 103, 'Saman', 'Perera');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO MANAGER VALUES ('MAN004', 104, 'Kumardi', 'Silva');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MANAGER VALUES ('MAN005', 105, 'Ravi', 'Fernando');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MANAGER VALUES ('MAN006', 106, 'Nishan', 'Rajapaksa');
Query OK, 1 row affected (0.01 sec)

mysql>
```

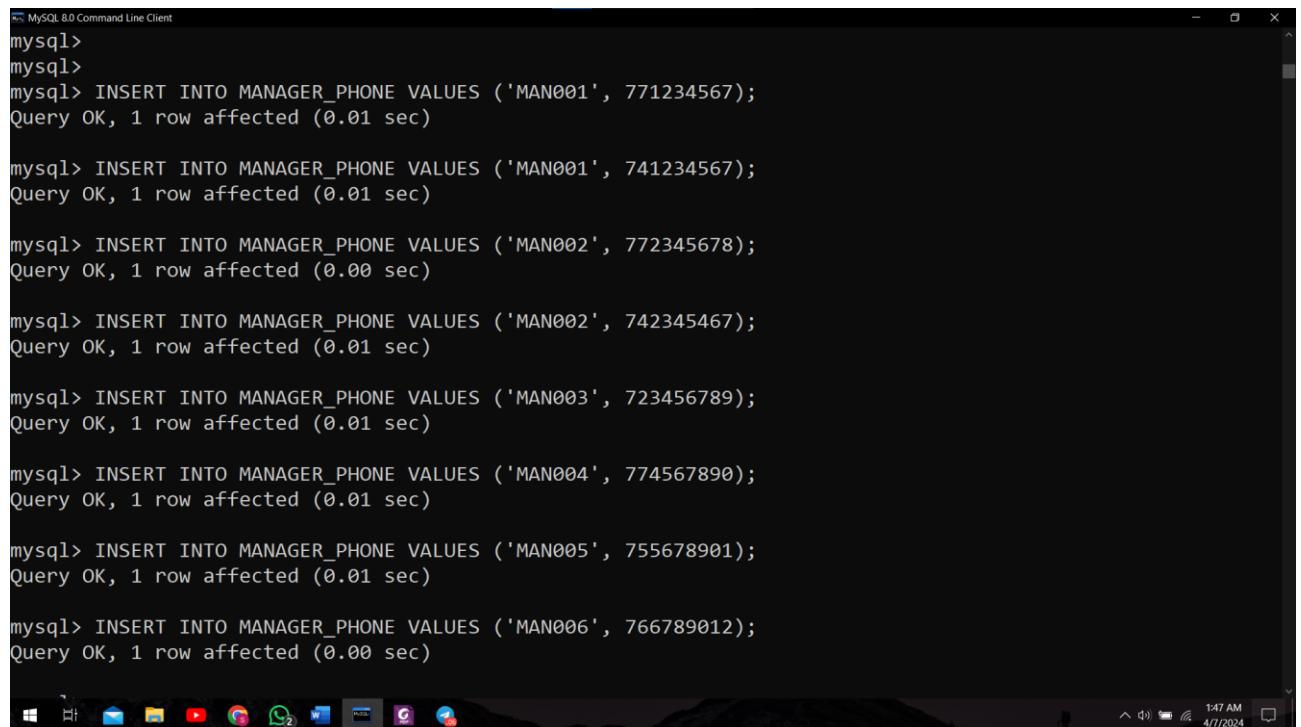
Figure 3.3.3 : Inserting Data to the MANAGER table

```
+-----+
6 rows in set (0.00 sec)

mysql> select * from MANAGER;
+-----+
| Manager_ID | Branch_Num | First_Name | Last_Name |
+-----+
| MAN001     |      101   |    John    |     Doe   |
| MAN002     |      102   |    Jane    |    Smith  |
| MAN003     |      103   |    Saman   |   Perera  |
| MAN004     |      104   |   Kumardi  |    Silva  |
| MAN005     |      105   |     Ravi   | Fernando |
| MAN006     |      106   |    Nishan   | Rajapaksa|
+-----+
6 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
```

Figure 3.3.4 : Manager table with Inserted Data



```
MySQL 8.0 Command Line Client
mysql>
mysql>
mysql> INSERT INTO MANAGER_PHONE VALUES ('MAN001', 771234567);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MANAGER_PHONE VALUES ('MAN001', 741234567);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MANAGER_PHONE VALUES ('MAN002', 772345678);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO MANAGER_PHONE VALUES ('MAN002', 742345467);
Query OK, 1 row affected (0.01 sec)

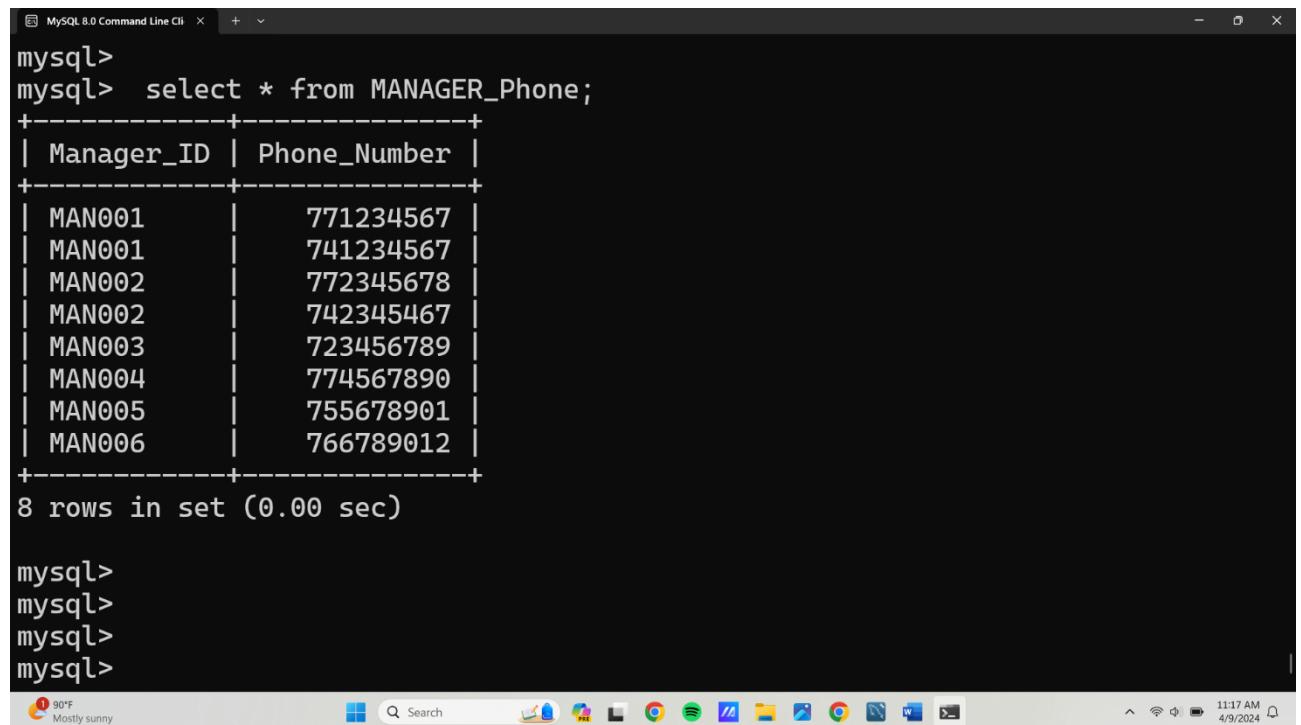
mysql> INSERT INTO MANAGER_PHONE VALUES ('MAN003', 723456789);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MANAGER_PHONE VALUES ('MAN004', 774567890);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MANAGER_PHONE VALUES ('MAN005', 755678901);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MANAGER_PHONE VALUES ('MAN006', 766789012);
Query OK, 1 row affected (0.00 sec)
```

Figure 3.3.5 : Inserting Data to the MANAGER\_PHONE table



```
MySQL 8.0 Command Line Cli ...
mysql>
mysql> select * from MANAGER_PHONE;
+-----+-----+
| Manager_ID | Phone_Number |
+-----+-----+
| MAN001     | 771234567 |
| MAN001     | 741234567 |
| MAN002     | 772345678 |
| MAN002     | 742345467 |
| MAN003     | 723456789 |
| MAN004     | 774567890 |
| MAN005     | 755678901 |
| MAN006     | 766789012 |
+-----+-----+
8 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
```

Figure 3.3.6 : Manager\_Phone table with Inserted Data

The screenshot shows the MySQL 8.0 Command Line Client window. The user has run a series of SQL INSERT statements to add 15 rows of equipment data into the EQUIPMENT table. The columns are Equipment\_Number, Equipment\_Name, and Cost. The data includes items like Treadmill, Elliptical Machine, Stationary Bike, etc., with costs ranging from 50000 to 300000. The client interface includes a toolbar at the bottom with various icons.

```
MySQL 8.0 Command Line Client
mysql>
mysql> INSERT INTO EQUIPMENT VALUES (1, 'Treadmill', 250000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EQUIPMENT VALUES (2, 'Elliptical Machine', 180000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EQUIPMENT VALUES (3, 'Stationary Bike', 150000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EQUIPMENT VALUES (4, 'Rowing Machine', 200000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EQUIPMENT VALUES (5, 'Dumbbells Set', 100000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EQUIPMENT VALUES (6, 'Bench Press', 150000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EQUIPMENT VALUES (7, 'Smith Machine', 300000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EQUIPMENT VALUES (8, 'Leg Press Machine', 250000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EQUIPMENT VALUES (9, 'Yoga Mats', 50000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EQUIPMENT VALUES (10, 'Resistance Bands', 30000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EQUIPMENT VALUES (11, 'Jump Ropes', 20000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EQUIPMENT VALUES (12, 'Exercise Balls', 60000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EQUIPMENT VALUES (13, 'Barbell Set', 120000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EQUIPMENT VALUES (14, 'Kettlebells', 90000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EQUIPMENT VALUES (15, 'Pull-Up Bar', 80000);
```

Figure 3.3.7 : Inserting Data to the EQUIPMENT table

The screenshot shows the MySQL 8.0 Command Line Client window displaying the results of a SELECT \* query on the EQUIPMENT table. The output is a table with three columns: Equipment\_Number, Equipment\_Name, and Cost. The data matches the inserted values shown in Figure 3.3.7. The client interface includes a toolbar at the bottom with various icons.

```
MySQL 8.0 Command Line Cli ...
mysql> select * from EQUIPMENT;
+-----+-----+-----+
| Equipment_Number | Equipment_Name | Cost |
+-----+-----+-----+
| 1 | Treadmill | 250000 |
| 2 | Elliptical Machine | 180000 |
| 3 | Stationary Bike | 150000 |
| 4 | Rowing Machine | 200000 |
| 5 | Dumbbells Set | 100000 |
| 6 | Bench Press | 150000 |
| 7 | Smith Machine | 300000 |
| 8 | Leg Press Machine | 250000 |
| 9 | Yoga Mats | 50000 |
| 10 | Resistance Bands | 30000 |
| 11 | Jump Ropes | 20000 |
| 12 | Exercise Balls | 60000 |
| 13 | Barbell Set | 120000 |
| 14 | Kettlebells | 90000 |
| 15 | Pull-Up Bar | 80000 |
+-----+-----+-----+
15 rows in set (0.00 sec)

mysql>
```

Figure 3.3.8 : EQUIPMENT table with Inserted Data

```
MySQL 8.0 Command Line Client

mysql> INSERT INTO Equipment_branchAndUnits VALUES (1, 101, 5);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (1, 104, 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (1, 105, 4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (2, 102, 2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (2, 103, 4);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (2, 106, 5);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (3, 106, 3);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (3, 104, 4);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (3, 101, 2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (4, 104, 5);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (5, 101, 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (6, 106, 4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (7, 101, 4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (8, 102, 5);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Equipment_branchAndUnits VALUES (9, 103, 3);


```

Figure 3.3.9 : Inserting Data to the Equipment\_branchAndUnits table

```
MySQL 8.0 Command Line Cli- x + v

mysql> Select * from Equipment_branchAndUnits;
+-----+-----+-----+
| Equipment_Number | Branch_Num | Number_of_Units |
+-----+-----+-----+
| 1 | 101 | 5 |
| 1 | 104 | 3 |
| 1 | 105 | 4 |
| 2 | 102 | 2 |
| 2 | 103 | 4 |
| 2 | 106 | 5 |
| 3 | 101 | 2 |
| 3 | 104 | 4 |
| 3 | 106 | 3 |
| 4 | 104 | 5 |
| 5 | 101 | 3 |
| 6 | 106 | 4 |
| 7 | 101 | 4 |
| 8 | 102 | 5 |
| 9 | 103 | 3 |
+-----+-----+-----+
15 rows in set (0.00 sec)

mysql>
```

Figure 3.3.10 : EQUIPMENT\_BranchAndUnits table with Inserted Data

```
MySQL 8.0 Command Line Client
mysql> INSERT INTO Equipment_repair_history VALUES (10, 'Replaced belt');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Equipment_repair_history VALUES (6, 'Fixed resistance mechanism');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Equipment_repair_history VALUES (3, 'Replaced seat');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Equipment_repair_history VALUES (8, 'Fixed handle grip');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Equipment_repair_history VALUES (5, 'Replaced damaged weights');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Equipment_repair_history VALUES (2, 'Repaired frame');
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figure 3.3.11 : Inserting Data to the Equipment\_repair\_history table

```
MySQL 8.0 Command Line Cli < + >
mysql> select * from EQUIPMENT_REPAIR_HISTORY;
+-----+-----+
| Equipment_number | repair_history |
+-----+-----+
| 2 | Repaired frame |
| 3 | Replaced seat |
| 5 | Replaced damaged weights |
| 6 | Fixed resistance mechanism |
| 8 | Fixed handle grip |
| 10 | Replaced belt |
+-----+-----+
6 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
```

Figure 3.3.12 : EQUIPMENT\_REPAIR\_HISTORY table with Inserted Data

```
MySQL 8.0 Command Line Client

mysql> INSERT INTO MEMBERSHIP_TYPE VALUES ('MET002', 'Silver', 'Limited access to facilities', '6 months', 7000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MEMBERSHIP_TYPE VALUES ('MET003', 'Bronze', 'Basic access to facilities', '3 months', 5000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MEMBERSHIP_TYPE VALUES ('MET004', 'Platinum', 'Premium access to all facilities', '2 years', 15000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MEMBERSHIP_TYPE VALUES ('MET005', 'Student', 'Discounted membership for students', '1 year', 8000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MEMBERSHIP_TYPE VALUES ('MET006', 'Corporate', 'Discounted membership for corporate employees', '1 year', 12000);
Query OK, 1 row affected (0.01 sec)

mysql>
```

Figure 3.3.13 : Inserting Data to the MEMBERSHIP\_TYPE table

```
MySQL 8.0 Command Line Cli < + >
mysql>
mysql> select * from MEMBERSHIP_Type;
+-----+-----+-----+-----+-----+
| MemType_ID | Type_Name | Description | Time_Period | price |
+-----+-----+-----+-----+-----+
| MET001 | Gold | Access to all facilities | 1 year | 10000 |
| MET002 | Silver | Limited access to facilities | 6 months | 7000 |
| MET003 | Bronze | Basic access to facilities | 3 months | 5000 |
| MET004 | Platinum | Premium access to all facilities | 2 years | 15000 |
| MET005 | Student | Discounted membership for students | 1 year | 8000 |
| MET006 | Corporate | Discounted membership for corporate employees | 1 year | 12000 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 3.3.14 : MEMBERSHIP\_TYPE table with Inserted Data

```
MySQL 8.0 Command Line Client
mysql> INSERT INTO TRAINER VALUES ('TRN001', 101, 'Kusal', 'Perera', 60000, 5, 'Galle Road', 'Colombo');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO TRAINER VALUES ('TRN002', 102, 'Chamari', 'Atapattu', 55000, 4, 'Kandy Road', 'Kandy');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO TRAINER VALUES ('TRN003', 103, 'Lasith', 'Malinga', 65000, 6, 'Matara Road', 'Galle');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO TRAINER VALUES ('TRN004', 104, 'Angelo', 'Mathews', 60000, 5, 'Weligama Road', 'Matara');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO TRAINER VALUES ('TRN005', 105, 'Muttiah', 'Muralitharan', 70000, 8, 'Katunayake Road', 'Negombo');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO TRAINER VALUES ('TRN006', 106, 'Rangana', 'Herath', 55000, 4, 'Kandy Road', 'Kandy');
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figure 3.3.15 : Inserting Data to the TRAINER table

```
MySQL 8.0 Command Line Cli < + >
mysql>
mysql>
mysql> select * from Trainer;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Trainer_ID | Branch_Num | First_Name | Last_Name | Salary | Experience_yrs | Street_Name | City |
+-----+-----+-----+-----+-----+-----+-----+-----+
| TRN001 | 101 | Kusal | Perera | 60000 | 5 | Galle Road | Colombo |
| TRN002 | 102 | Chamari | Atapattu | 55000 | 4 | Kandy Road | Kandy |
| TRN003 | 103 | Lasith | Malinga | 65000 | 6 | Matara Road | Galle |
| TRN004 | 104 | Angelo | Mathews | 60000 | 5 | Weligama Road | Matara |
| TRN005 | 105 | Muttiah | Muralitharan | 70000 | 8 | Katunayake Road | Negombo |
| TRN006 | 106 | Rangana | Herath | 55000 | 4 | Kandy Road | Kandy |
+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 3.3.16 : Trainer table with Inserted Data

```
MySQL 8.0 Command Line Client
mysql> INSERT INTO Trainer_4n_Numbers VALUES ('TRN001', 771234567);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Trainer_4n_Numbers VALUES ('TRN001', 741234567);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Trainer_4n_Numbers VALUES ('TRN002', 772345678);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Trainer_4n_Numbers VALUES ('TRN002', 742345467);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Trainer_4n_Numbers VALUES ('TRN003', 723456789);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Trainer_4n_Numbers VALUES ('TRN004', 774567890);
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figure 3.3.17 : Inserting Data to the Trainer\_4n\_Numbers table

```
MySQL 8.0 Command Line Cli < + >
mysql>
mysql>
mysql> select * from Trainer_4n_Numbers;
+-----+-----+
| Trainer_ID | Phone_Number |
+-----+-----+
| TRN001    | 741234567 |
| TRN001    | 771234567 |
| TRN002    | 742345467 |
| TRN002    | 772345678 |
| TRN003    | 723456789 |
| TRN004    | 774567890 |
+-----+-----+
6 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
```

Figure 3.3.18 : TRAINER\_4n\_Numbers table with Inserted Data

```
MySQL 8.0 Command Line Client

mysql> INSERT INTO MEMBER VALUES ('M001', 'TRN003', 'MET003', 'Saman', 'Perera', 789, 'Third Road', 'Galle');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MEMBER VALUES ('M002', 'TRN004', 'MET004', 'Kumari', 'Silva', 1011, 'Fourth Lane', 'Matara');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MEMBER VALUES ('M003', 'TRN001', 'MET001', 'John', 'Doe', 123, 'Main Street', 'Colombo');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MEMBER VALUES ('M004', 'TRN006', 'MET006', 'Nisha', 'Rajapaksa', 1415, 'Sixth Avenue', 'Kandy');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO MEMBER VALUES ('M005', 'TRN005', 'MET005', 'Ravi', 'Fernando', 1213, 'Fifth Street', 'Negombo');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO MEMBER VALUES ('M006', 'TRN002', 'MET002', 'Jane', 'Smith', 456, 'Second Avenue', 'Kandy');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO MEMBER VALUES ('M007', 'TRN002', 'MET002', 'Samantha', 'Silva', 1819, 'Eighth Lane', 'Kandy');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MEMBER VALUES ('M008', 'TRN003', 'MET003', 'Kasun', 'Fernando', 2021, 'Ninth Street', 'Galle');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MEMBER VALUES ('M009', 'TRN001', 'MET001', 'Malith', 'Perera', 1617, 'Seventh Road', 'Colombo');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO MEMBER VALUES ('M010', 'TRN004', 'MET004', 'Nuwan', 'Rajapaksa', 2223, 'Tenth Avenue', 'Matara');
Query OK, 1 row affected (0.01 sec)

mysql>
```

Figure 3.3.19 : Inserting Data to the MEMBER table

```
MySQL 8.0 Command Line Cli X + v

mysql>
mysql>
mysql>
mysql> select * from Member;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Member_ID | Trainer_ID | MemType_ID | First_Name | Last_Name | House_Num | Street_Name | City |
+-----+-----+-----+-----+-----+-----+-----+-----+
| M001 | TRN003 | MET003 | Saman | Perera | 789 | Third Road | Galle |
| M002 | TRN004 | MET004 | Kumari | Silva | 1011 | Fourth Lane | Matara |
| M003 | TRN001 | MET001 | John | Doe | 123 | Main Street | Colombo |
| M004 | TRN006 | MET006 | Nisha | Rajapaksa | 1415 | Sixth Avenue | Kandy |
| M005 | TRN005 | MET005 | Ravi | Fernando | 1213 | Fifth Street | Negombo |
| M006 | TRN002 | MET002 | Jane | Smith | 456 | Second Avenue | Kandy |
| M007 | TRN002 | MET002 | Samantha | Silva | 1819 | Eighth Lane | Kandy |
| M008 | TRN003 | MET003 | Kasun | Fernando | 2021 | Ninth Street | Galle |
| M009 | TRN001 | MET001 | Malith | Perera | 1617 | Seventh Road | Colombo |
| M010 | TRN004 | MET004 | Nuwan | Rajapaksa | 2223 | Tenth Avenue | Matara |
+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> |
```

Figure 3.3.20 : Member table with Inserted Data

```
MySQL 8.0 Command Line Client

mysql> INSERT INTO Member_4n_Numbers VALUES ('M001', 771234567);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Member_4n_Numbers VALUES ('M001', 741234567);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Member_4n_Numbers VALUES ('M002', 772345678);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Member_4n_Numbers VALUES ('M006', 742345467);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Member_4n_Numbers VALUES ('M006', 723456789);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Member_4n_Numbers VALUES ('M005', 714567890);
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figure 3.3.21 : Inserting Data to the Membr\_4n\_Numbers table

```
MySQL 8.0 Command Line Cli - + x

mysql>
mysql> select * from Member_4n_Numbers;
+-----+-----+
| Member_ID | Phone_Number |
+-----+-----+
| M001      | 741234567   |
| M001      | 771234567   |
| M002      | 772345678   |
| M003      | 723456789   |
| M003      | 742345467   |
| M004      | 714567890   |
+-----+-----+
6 rows in set (0.00 sec)

mysql>
mysql>
mysql> |
```

Figure 3.3.22 : Member\_4n\_Numbers table with Inserted Data

```
MySQL 8.0 Command Line Client
mysql> INSERT INTO SCHEDULE_VALUES ('2024-03-01', '09:00:00', 1.5, 'M001');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SCHEDULE_VALUES ('2024-03-01', '10:00:00', 3, 'M002');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO SCHEDULE_VALUES ('2024-03-02', '08:00:00', 2, 'M003');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO SCHEDULE_VALUES ('2024-03-02', '11:00:00', 1, 'M004');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO SCHEDULE_VALUES ('2024-03-03', '10:00:00', 2.5, 'M005');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SCHEDULE_VALUES ('2024-03-03', '14:00:00', 1, 'M006');
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figure 3.3.23 : Inserting Data to the SCHEDULE\_ table

```
MySQL 8.0 Command Line Cli < + >
mysql>
mysql> select * from SCHEDULE_;
+-----+-----+-----+-----+
| Date_| Start_Time | Duration_hrs | Member_ID |
+-----+-----+-----+-----+
| 2024-03-01 | 09:00:00 | 1.5 | M001 |
| 2024-03-01 | 10:00:00 | 3 | M002 |
| 2024-03-02 | 08:00:00 | 2 | M003 |
| 2024-03-02 | 11:00:00 | 1 | M004 |
| 2024-03-03 | 10:00:00 | 2.5 | M005 |
| 2024-03-03 | 14:00:00 | 1 | M006 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
mysql>
mysql> |
```

Figure 3.3.24 : Schedule\_ table with Inserted Data

```
MySQL 8.0 Command Line Client
mysql> INSERT INTO PAYMENT VALUES ('M001', '08:30:00', '2024-02-01', 15000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PAYMENT VALUES ('M003', '09:45:00', '2024-02-01', 5000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PAYMENT VALUES ('M005', '08:15:00', '2024-02-02', 10000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PAYMENT VALUES ('M009', '11:30:00', '2024-02-02', 8000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PAYMENT VALUES ('M007', '10:30:00', '2024-02-03', 7000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PAYMENT VALUES ('M010', '14:45:00', '2024-02-03', 12000);
Query OK, 1 row affected (0.00 sec)

mysql>
```

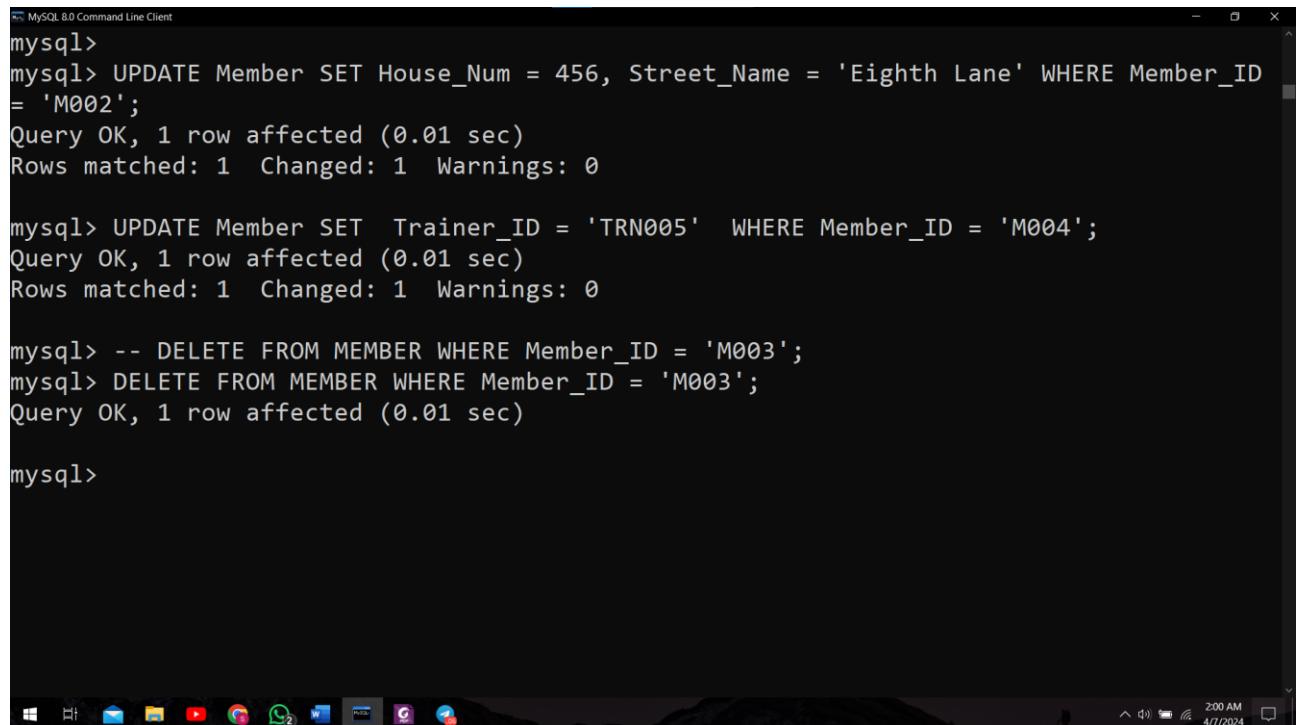
Figure 3.3.25 : Inserting Data to the PAYMENT table

```
MySQL 8.0 Command Line Cli < + >
mysql>
mysql>
mysql> select * from Payment;
+-----+-----+-----+-----+
| Member_ID | Time_ | Date_ | amount |
+-----+-----+-----+-----+
| M001      | 08:30:00 | 2024-02-01 | 15000 |
| M003      | 09:45:00 | 2024-02-01 | 5000  |
| M005      | 08:15:00 | 2024-02-02 | 10000 |
| M007      | 10:30:00 | 2024-02-03 | 7000  |
| M009      | 11:30:00 | 2024-02-02 | 8000  |
| M010      | 14:45:00 | 2024-02-03 | 12000 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 3.3.26 Payment table with Inserted Data

### 3.4 Update and Delete operations



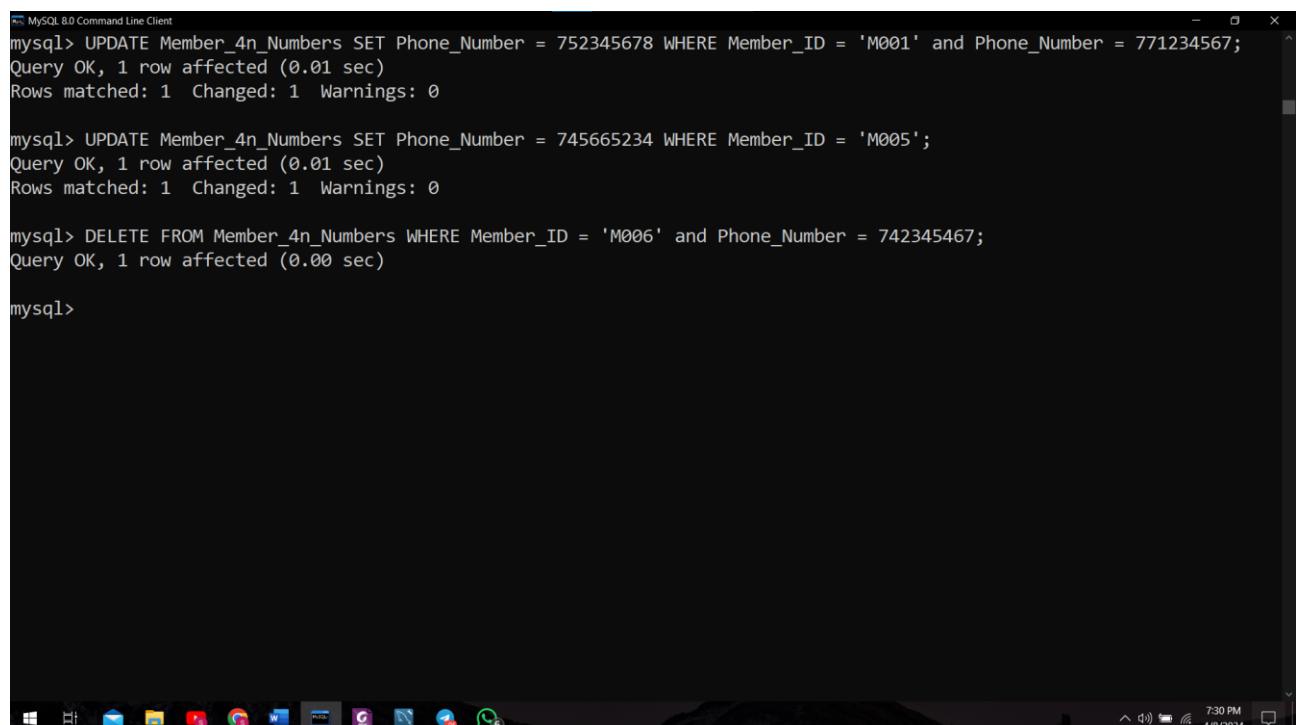
```
MySQL 8.0 Command Line Client
mysql> UPDATE Member SET House_Num = 456, Street_Name = 'Eighth Lane' WHERE Member_ID = 'M002';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Member SET Trainer_ID = 'TRN005' WHERE Member_ID = 'M004';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> -- DELETE FROM MEMBER WHERE Member_ID = 'M003';
mysql> DELETE FROM MEMBER WHERE Member_ID = 'M003';
Query OK, 1 row affected (0.01 sec)

mysql>
```

Figure 3.4.1 : Update and Delete data from the Member table



```
MySQL 8.0 Command Line Client
mysql> UPDATE Member_4n_Numbers SET Phone_Number = 752345678 WHERE Member_ID = 'M001' and Phone_Number = 771234567;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Member_4n_Numbers SET Phone_Number = 745665234 WHERE Member_ID = 'M005';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM Member_4n_Numbers WHERE Member_ID = 'M006' and Phone_Number = 742345467;
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figure 3.4.2 : Update and Delete data from the Member\_4n\_Numbers table

```
MySQL 8.0 Command Line Client
mysql> UPDATE Trainer SET Salary = 70000 WHERE Trainer_ID = 'TRN001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Trainer SET Experience_yrs = 7 WHERE Trainer_ID = 'TRN002';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM Trainer WHERE Trainer_ID = 'TRN006';
Query OK, 1 row affected (0.01 sec)

mysql>
```

Figure 3.4.3 : Update and Delete data from the Trainer table

```
MySQL 8.0 Command Line Client
mysql> UPDATE Branch SET Capacity = '250' WHERE Branch_Num = 101;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Branch SET Opening_Hours = '9:00 AM - 10:00 PM' WHERE Branch_Num = 102;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM Branch WHERE Branch_Num = 106;
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figure 3.4.4 : Update and Delete data from the Branch table

MySQL 8.0 Command Line Client

```
mysql> UPDATE Manager SET Last_Name = 'Emayer' WHERE Manager_ID = 'MAN001';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Manager SET First_name = 'Perara' WHERE Manager_ID = 'MAN002';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM Manager WHERE Manager_ID = 'MAN006';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Figure 3.4.5 : Update and Delete data from the Manager table

MySQL 8.0 Command Line Client

```
mysql> UPDATE MANAGER_PHONE SET Phone_Number = 770000000 WHERE Manager_ID = 'MAN001' AND Phone_Number = 771234567;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE MANAGER_PHONE SET Phone_Number = 770000001 WHERE Manager_ID = 'MAN002' AND Phone_Number = 772345678;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM MANAGER_PHONE WHERE Manager_ID = 'MAN003' AND Phone_Number = 723456789;
Query OK, 1 row affected (0.01 sec)

mysql>
```

Figure 3.4.6 : Update and Delete data from the MANAGER\_PHONE table

```
MySQL 8.0 Command Line Client
mysql> UPDATE Schedule_ SET Start_Time = '10:30:00' WHERE Date_ = '2024-03-01'
AND Member_ID = 'M001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Schedule_ SET Duration_hrs = 2 WHERE Date_ = '2024-03-01' AND Mem
ber_ID = 'M002';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM Schedule_ WHERE Date_ = '2024-03-03' AND Member_ID = 'M006';

Query OK, 1 row affected (0.01 sec)

mysql>
```

Figure 3.4.7 : Update and Delete data from the SCHEDULE table

```
MySQL 8.0 Command Line Client
mysql> UPDATE Payment SET amount = 9000 WHERE Member_ID = 'M001' AND Time_ = '0
8:30:00' AND Date_ = '2024-02-01';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Payment SET Date_ = '2024-02-03' WHERE Member_ID = 'M007' AND Tim
e_ = '10:30:00';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> DELETE FROM Payment WHERE Member_ID = 'M009' AND Time_ = '11:30:00' AND
Date_ = '2024-02-02';
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figure 3.4.8 : Update and Delete data from the Payment table

```
MySQL 8.0 Command Line Client
mysql> UPDATE Membership_Type SET Description_ = 'Discounted Membership for School Teachers' WHERE MemType_ID = 'MET006';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Membership_Type SET price = 8500 WHERE MemType_ID = 'MET002';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM Membership_Type WHERE MemType_ID = 'MET001';
Query OK, 1 row affected (0.01 sec)

mysql>
```

Figure 3.4.9 : Update and Delete data from the Membership\_Type table

```
MySQL 8.0 Command Line Client
mysql> UPDATE Equipment SET Cost = 280000 WHERE Equipment_Number = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Equipment SET Equipment_Name = 'Bicep Machine' WHERE Equipment_Number = 2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM Equipment WHERE Equipment_Number = 6;
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figure 3.4.10 : Update and Delete data from the Equipment table

```
MySQL 8.0 Command Line Client
mysql> UPDATE Equipment_branchAndUnits SET Number_of_Units = 6 WHERE Branch_Num = 103 AND Equipment_Number = 9;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Equipment_branchAndUnits SET Number_of_Units = 4 WHERE Branch_Num = 102 AND Equipment_Number = 8;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> -- DELETE FROM Equipment_branchAndUnits WHERE Branch_Num = 105 AND Equipment_Number = 2;
mysql> DELETE FROM Equipment_branchAndUnits WHERE Branch_Num = 105 AND Equipment_Number = 2;
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figure 3.4.11 : Update and Delete data from the Equiment\_branchAndUnits table

```
MySQL 8.0 Command Line Client
mysql> UPDATE Equipment_repair_history SET repair_history = 'Replaced motor' WHERE Equipment_Number = 10;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

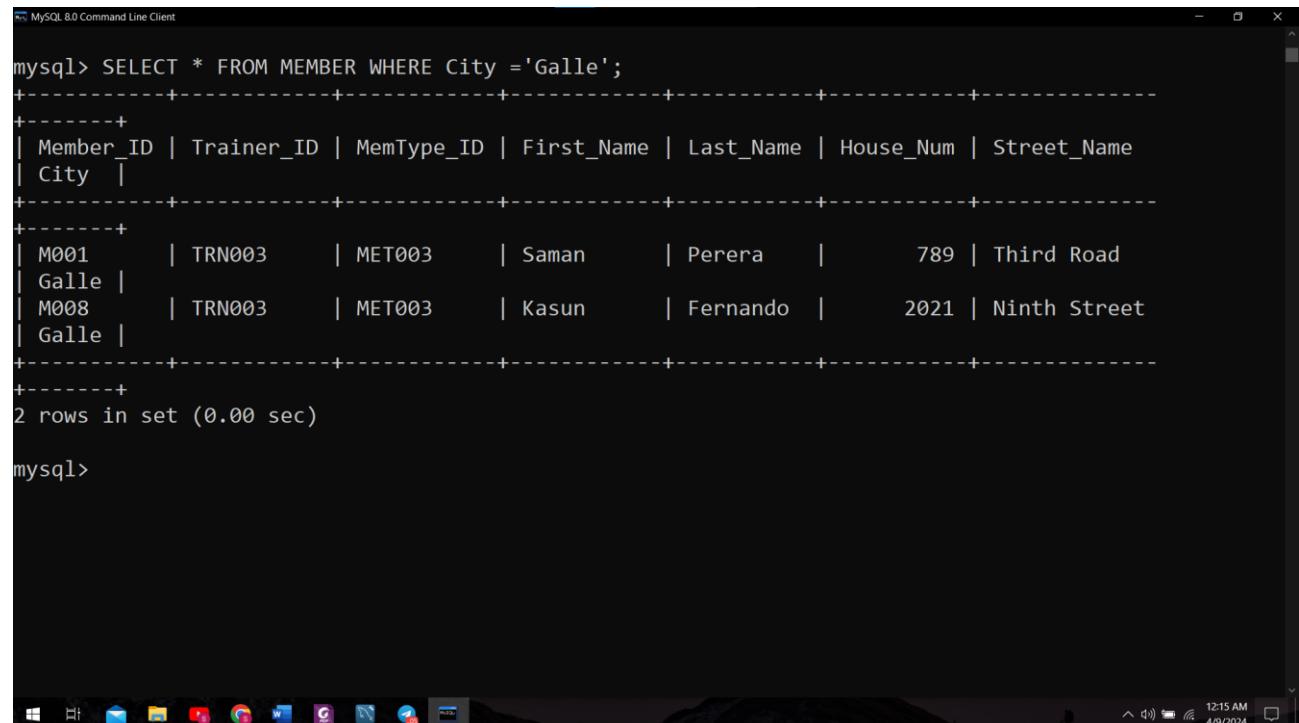
mysql> UPDATE Equipment_repair_history SET repair_history = 'Replaced display screen' WHERE Equipment_Number = 2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM Equipment_repair_history WHERE Equipment_Number = 3 AND repair_history = 'Replaced seat';
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figure 3.4.12 : Update and Delete data from the Equipment\_repair\_history table

## 4 Transaction



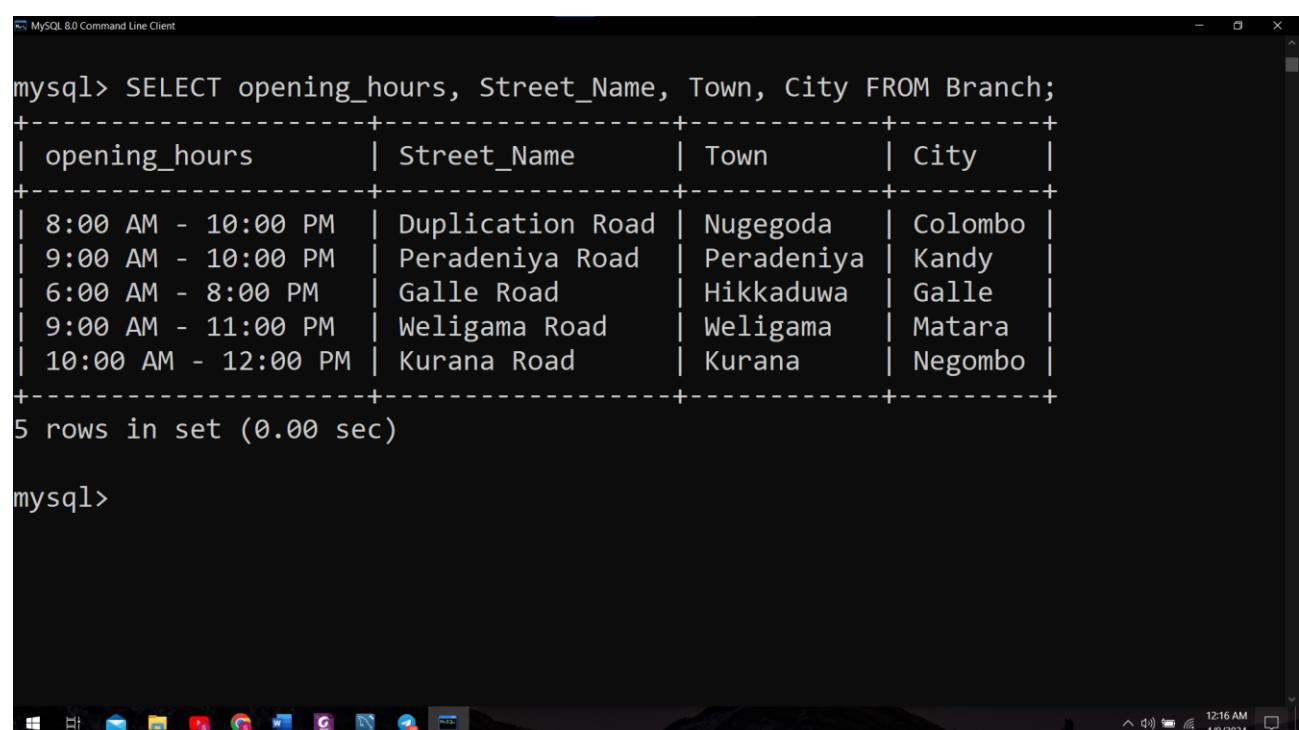
MySQL 8.0 Command Line Client

```
mysql> SELECT * FROM MEMBER WHERE City ='Galle';
+-----+-----+-----+-----+-----+-----+
| Member_ID | Trainer_ID | MemType_ID | First_Name | Last_Name | House_Num | Street_Name
| City      |
+-----+-----+-----+-----+-----+-----+
| M001      | TRN003    | MET003    | Saman      | Perera     |        789 | Third Road
| Galle     |
| M008      | TRN003    | MET003    | Kasun      | Fernando   |      2021 | Ninth Street
| Galle     |
+-----+-----+-----+-----+-----+-----+
+-----+
2 rows in set (0.00 sec)

mysql>
```

Windows taskbar at the bottom with various icons.

Figure 3.4.1 : Query to Select all members who are from Galle



MySQL 8.0 Command Line Client

```
mysql> SELECT opening_hours, Street_Name, Town, City FROM Branch;
+-----+-----+-----+-----+
| opening_hours | Street_Name | Town    | City   |
+-----+-----+-----+-----+
| 8:00 AM - 10:00 PM | Duplication Road | Nugegoda | Colombo
| 9:00 AM - 10:00 PM | Peradeniya Road | Peradeniya | Kandy
| 6:00 AM - 8:00 PM | Galle Road | Hikkaduwa | Galle
| 9:00 AM - 11:00 PM | Weligama Road | Weligama | Matara
| 10:00 AM - 12:00 PM | Kurana Road | Kurana | Negombo
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Windows taskbar at the bottom with various icons.

Figure 3.4.2 : Simple Select query to display selected columns

```
MySQL 8.0 Command Line Client

mysql> SELECT * FROM Manager NATURAL JOIN manager_phone;
+-----+-----+-----+-----+-----+
| Manager_ID | Branch_Num | First_Name | Last_Name | Phone_Number |
+-----+-----+-----+-----+-----+
| MAN001     |      101 | John       | Emayer    | 7700000000 |
| MAN001     |      101 | John       | Emayer    | 741234567  |
| MAN002     |      102 | Perara    | Smith     | 770000001  |
| MAN002     |      102 | Perara    | Smith     | 742345467  |
| MAN004     |      104 | Kumardi   | Silva    | 774567890  |
| MAN005     |      105 | Ravi      | Fernando | 755678901  |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Figure 3.4.3 : Cartesian Product Operation

```
MySQL 8.0 Command Line Client

mysql> CREATE VIEW user_view AS
-> SELECT Type_Name, Description_, price FROM Membership_type;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM user_View;
+-----+-----+-----+
| Type_Name | Description_ | price |
+-----+-----+-----+
| Silver    | Limited access to facilities | 8500 |
| Bronze    | Basic access to facilities | 5000 |
| Platinum  | Premium access to all facilities | 15000 |
| Student   | Discounted membership for students | 8000 |
| Corporate | Discounted Membership for School Teachers | 12000 |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Figure 3.4.4 : Query to Create a User View

```
MySQL 8.0 Command Line Client
mysql> ALTER TABLE equipment_repair_history
      -> RENAME COLUMN repair_history TO Recent_Repairs;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

Figure 3.4.5 : Query to Rename Operation

```
MySQL 8.0 Command Line Client
mysql>
mysql> SELECT MAX(capacity) AS max_Capacity FROM Branch;
+-----+
| max_Capacity |
+-----+
| 250          |
+-----+
1 row in set (0.00 sec)

mysql> SELECT MIN(price) AS min_price FROM membership_type;
+-----+
| min_price |
+-----+
| 5000       |
+-----+
1 row in set (0.00 sec)

mysql> SELECT AVG(cost) AS Average_Cost FROM Equipment;
+-----+
| Average_Cost    |
+-----+
| 136428.57142857142 |
+-----+
1 row in set (0.00 sec)
```

Figure 3.4.6 : Use of Aggregation Function

```
MySQL 8.0 Command Line Client
mysql> SELECT * FROM Equipment WHERE Equipment_Name LIKE '%Machine%';
+-----+-----+-----+
| Equipment_Number | Equipment_Name      | Cost   |
+-----+-----+-----+
|          2 | Bicep Machine        | 180000 |
|          4 | Rowing Machine       | 200000 |
|          7 | Smith Machine        | 300000 |
|          8 | Leg Press Machine    | 250000 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

Figure 3.4.7 : Use of LIKE Keyword

```
MySQL 8.0 Command Line Client
mysql> SELECT M.Manager_ID FROM Manager As M
      -> UNION
      -> SELECT T.Trainer_ID FROM Trainer as T;
+-----+
| Manager_ID |
+-----+
| MAN001
| MAN002
| MAN003
| MAN004
| MAN005
| TRN001
| TRN002
| TRN003
| TRN004
| TRN005
+-----+
10 rows in set (0.00 sec)

mysql>
```

Figure 3.4.8 : Union of Managers and Trainers into Employee

```
MySQL 8.0 Command Line Client
mysql> SELECT M.First_Name, M.Trainer_ID, W.Duration_hrs, W.Start_Time
   -> FROM member as M
   -> LEFT JOIN SCHEDULE_ as W ON M.Member_ID = W.Member_ID;
+-----+-----+-----+-----+
| First_Name | Trainer_ID | Duration_hrs | Start_Time |
+-----+-----+-----+-----+
| Saman      | TRN003    |      1.5     | 10:30:00
| Kumari     | TRN004    |      2        | 10:00:00
| Nisha      | TRN005    |      1        | 11:00:00
| Ravi       | TRN005    |      2.5     | 10:00:00
| Jane        | TRN002    |      NULL     | NULL
| Samantha   | TRN002    |      NULL     | NULL
| Kasun      | TRN003    |      NULL     | NULL
| Nuwan      | TRN004    |      NULL     | NULL
+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

Figure 3.4.9 : Intersection of Member and Schedule tables

```
MySQL 8.0 Command Line Client
mysql> SELECT M.Member_ID
   -> FROM MEMBER as M
   -> LEFT JOIN SCHEDULE_ as W ON M.Member_ID = W.Member_ID
   -> WHERE W.Member_ID IS NULL;
+-----+
| Member_ID |
+-----+
| M008      |
| M006      |
| M007      |
| M010      |
+-----+
4 rows in set (0.00 sec)

mysql>
```

Figure 3.4.10 : Finding the members who doesn't have the workout plan yet

```
MySQL 8.0 Command Line Client

mysql> SELECT
    ->     AVG(members_per_branch) as average_members_per_branch
    -> FROM
    -> (SELECT
    ->     COUNT(*) as members_per_branch
    ->     FROM
    ->         member as m
    ->     JOIN
    ->         Trainer as T on m.Trainer_ID = T.Trainer_ID
    ->     JOIN
    ->         Branch as B on T.Branch_Num = B.Branch_Num
    ->     WHERE
    ->         T.Experience_yrs >= 5
    ->         AND B.City = 'Galle'
    ->     GROUP BY
    ->         B.Branch_Num) as branch_member_counts;
+-----+
| average_members_per_branch |
+-----+
|           2.0000 |
+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 3.4.11 : Query for finding the average member per branch

```
MySQL 8.0 Command Line Client

mysql> CREATE VIEW Member_Trainer_Details_Complex AS
    -> SELECT m.First_Name AS Member_FirstName, m.Last_Name AS Member_LastName,
    ->        t.First_Name AS Trainer_FirstName, t.Last_Name AS Trainer_LastName
    -> FROM MEMBER m
    -> INNER JOIN TRAINER t ON m.Trainer_ID = t.Trainer_ID
    -> WHERE m.City = 'Galle';
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT *
    -> FROM Member_Trainer_Details_Complex;
+-----+-----+-----+-----+
| Member_FirstName | Member_LastName | Trainer_FirstName | Trainer_LastName |
+-----+-----+-----+-----+
| Saman          | Perera        | Lasith          | Malinga         |
| Kasun          | Fernando      | Lasith          | Malinga         |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Figure 3.4.12 : Inner Join of Tables of member and their trainer names in one table

```
MySQL 8.0 Command Line Client
Query OK, 0 rows affected (0.01 sec)

mysql> Create view Members_having_GOODtrainer_and_from_colombo as
-> Select
-> M.Member_ID , M.Trainer_ID, M.First_name as Member, T.First_Name as Trainer
-> from
-> Member as M
-> join
-> trainer as T
-> on M.Trainer_ID = T.Trainer_ID
-> where T.Experience_yrs >= 5 and m.city = 'Galle';
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT *
-> FROM Members_having_GOODtrainer_and_from_colombo;
+-----+-----+-----+
| Member_ID | Trainer_ID | Member | Trainer |
+-----+-----+-----+
| M001      | TRN003     | Saman   | Lasith  |
| M008      | TRN003     | Kasun   | Lasith  |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Figure 3.4.13 : Natural join to view Members having trainers who have more than 5 years of experience and from Galle

```
MySQL 8.0 Command Line Client
mysql> CREATE VIEW Branch_Manager_Details AS
-> SELECT b.Branch_Num, m.First_Name AS Manager_Name, b.City, b.Town, b.Opening_Hours, b.Capacity,
-> b.Street_Name, b.Building_Num
-> FROM BRANCH b
-> LEFT OUTER JOIN MANAGER m ON b.Branch_Num = m.Branch_Num
-> WHERE b.City LIKE 'C%' AND m.First_Name IS NOT NULL;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT *
-> FROM Branch_Manager_Details;
+-----+-----+-----+-----+-----+-----+-----+
| Branch_Num | Manager_Name | City    | Town   | Opening_Hours | Capacity | Street_Name   | Building_Num |
+-----+-----+-----+-----+-----+-----+-----+
|      101   | John        | Colombo | Nugegoda | 8:00 AM - 10:00 PM | 250     | Duplication Road |          12 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 3.4.14 : Left Outer join of Branch and Associated Managers

```
MySQL 8.0 Command Line Client
mysql> CREATE VIEW Trainer_Branch_Details AS
-> SELECT t.Trainer_ID, t.First_Name AS Trainer_name, b.branch_num,
-> t.Experience_yrs, b.Opening_Hours, b.Capacity,
-> b.City AS Branch_City
-> FROM TRAINER t
-> RIGHT OUTER JOIN BRANCH b
-> ON t.Branch_Num = b.Branch_Num
-> WHERE t.First_Name LIKE '%a%' AND b.City = 'Kandy';
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT *
-> FROM Trainer_Branch_Details;
+-----+-----+-----+-----+-----+-----+
| Trainer_ID | Trainer_name | branch_num | Experience_yrs | Opening_Hours | Capacity | Branch_City |
+-----+-----+-----+-----+-----+-----+
| TRN002     | Chamari      |      102   |          7       | 9:00 AM - 10:00 PM |    150    | Kandy
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 3.4.15 : Right Outer Join of Trainer and Branch

```
MySQL 8.0 Command Line Client
mysql> CREATE VIEW Member_Payment_Details AS
-> (SELECT m.member_ID, m.MemType_ID, m.first_name, m.city, p.Time_, p.Date_, p.amount
-> FROM member AS m
-> LEFT JOIN payment AS p
-> ON m.member_ID = p.Member_ID
-> WHERE m.first_name LIKE '%n%'
-> AND p.Date_ = '2024-02-03')
-> UNION
-> (SELECT m.member_ID, m.MemType_ID, m.first_name, m.city, p.Time_, p.Date_, p.amount
-> FROM member AS m
-> RIGHT JOIN payment AS p
-> ON m.member_ID = p.Member_ID
-> WHERE m.first_name LIKE '%n%'
-> AND p.Date_ = '2024-02-03');
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT *
-> FROM Member_Payment_Details;
+-----+-----+-----+-----+-----+-----+
| member_ID | MemType_ID | first_name | city   | Time_  | Date_  | amount |
+-----+-----+-----+-----+-----+-----+
| M007      | MET002     | Samantha   | Kandy | 10:30:00 | 2024-02-03 | 7000  |
| M010      | MET004     | Nuwan     | Matara | 14:45:00 | 2024-02-03 | 12000 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Figure 3.4.16 : Full Outer Join of Member and Payment

```
MySQL 8.0 Command Line Client
mysql> CREATE VIEW Membership_Type_Schedule AS
-> SELECT ID, Type_Name, price, Type
-> FROM (
->     SELECT MemType_ID AS ID, Type_Name, price, 'MS_Type' AS Type
->     FROM membership_type
->     WHERE Type_Name LIKE '%Gold%' OR Type_Name LIKE '%Platinum%'
->     UNION
->     SELECT Member_ID AS ID, CONVERT(Duration_hrs, CHAR), Start_Time, 'Plan' AS Type
->     FROM SCHEDULE_
->     WHERE Start_Time BETWEEN '08:00:00' AND '12:00:00'
-> ) AS Combined_Data;
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> SELECT *
-> FROM Membership_Type_Schedule;
+----+----+----+----+
| ID | Type_Name | price | Type |
+----+----+----+----+
| MET004 | Platinum | 15000 | MS_Type |
| M001 | 1.5 | 10:30:00 | Plan |
| M002 | 2 | 10:00:00 | Plan |
| M004 | 1 | 11:00:00 | Plan |
| M005 | 2.5 | 10:00:00 | Plan |
+----+----+----+----+
5 rows in set (0.00 sec)

mysql>
```

Figure 3.4.17 : Outer Union of Membership\_Type and Schedule

```
MySQL 8.0 Command Line Client
mysql> SELECT *
-> FROM (
->     SELECT Member_ID, Trainer_ID, First_Name, Last_Name
->     FROM MEMBER
->     WHERE City = 'Galle'
-> ) AS Galle_Members
-> INNER JOIN (
->     SELECT Trainer_ID, First_Name, Last_Name
->     FROM TRAINER
->     WHERE City = 'Galle'
-> ) AS Galle_Trainers ON Galle_Members.Trainer_ID = Galle_Trainers.Trainer_ID;
+----+----+----+----+----+----+
| Member_ID | Trainer_ID | First_Name | Last_Name | Trainer_ID | First_Name | Last_Name |
+----+----+----+----+----+----+
| M001 | TRN003 | Saman | Perera | TRN003 | Lasith | Malinga |
| M008 | TRN003 | Kasun | Fernando | TRN003 | Lasith | Malinga |
+----+----+----+----+----+----+
2 rows in set (0.00 sec)

mysql>
```

Figure 3.4.18 : Returning details of members and trainers who are both located in Colombo using nested query

```
MySQL 8.0 Command Line Client
mysql> SELECT *
->   FROM MEMBER
-> WHERE MemType_ID = (
->   SELECT MemType_ID
->     FROM MEMBERSHIP_TYPE
->   WHERE Type_Name = 'Silver'
->   )
-> AND Trainer_ID IN (
->   SELECT Trainer_ID
->     FROM TRAINER
->   WHERE Experience_yrs > 4
-> );
+-----+-----+-----+-----+-----+-----+-----+-----+
| Member_ID | Trainer_ID | MemType_ID | First_Name | Last_Name | House_Num | Street_Name | City |
+-----+-----+-----+-----+-----+-----+-----+-----+
| M006      | TRN002    | MET002    | Jane       | Smith     |      456  | Second Avenue | Kandy |
| M007      | TRN002    | MET002    | Samantha   | Silva    |     1819  | Eighth Lane  | Kandy |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Figure 3.4.19 : The Members Who Have Membership Type As Silver And Are Assigned To A Trainer With More Than 4 Years Of Experience

```
MySQL 8.0 Command Line Client
mysql> SELECT SUM(amount) AS Total_Earnings
->   FROM PAYMENT
-> WHERE Member_ID IN (
->   SELECT Member_ID
->     FROM MEMBER
-> WHERE MemType_ID IN (
->   SELECT MemType_ID
->     FROM MEMBERSHIP_TYPE
->   WHERE price > (
->     SELECT AVG(price)
->       FROM MEMBERSHIP_TYPE
->   )
->   )
-> );
+-----+
| Total_Earnings |
+-----+
|        12000 |
+-----+
1 row in set (0.00 sec)

mysql>
```

Figure 3.4.20 : The Total Amount Earned from Payments Made By Members Who Have A Membership Type With A Price Higher Than The Average Price Of All Membership Types

## 5 Tunning

The screenshot shows a Windows desktop environment with a MySQL 8.0 Command Line Client window open. The window displays two sets of SQL queries and their execution plans. The first set of queries is for a table named 'SCHEDULE\_'. The second set is for a table named 'WORKOUT\_PLAN'. Both sets show the creation of indexes and the execution of SELECT statements with their respective EXPLAIN outputs. The EXPLAIN output provides details about the query execution plan, including select\_type, table, partitions, type, possible\_keys, key, key\_len, ref, rows, filtered, and Extra columns. The MySQL command line interface also shows various system status icons at the bottom.

```
mysql> explain (
    -> SELECT M.First_Name, M.Trainer_ID, W.Duration_hrs, W.Start_Time
    -> FROM member as M
    -> LEFT JOIN SCHEDULE_ as W ON M.Member_ID = W.Member_ID;
+----+ | id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
| 1 | SIMPLE | M | NULL | ALL | NULL | idx_schedule_member_id | 122 | NULL | gym_management_system.M.Member_ID | 8 | 100.00 | NULL |
| 1 | SIMPLE | W | NULL | ref | idx_schedule_member_id | 122 | NULL | gym_management_system.M.Member_ID | 1 | 100.00 | Using index |
+----+
2 rows in set, 1 warning (0.00 sec)

mysql> CREATE INDEX idx_schedule_member_id ON SCHEDULE_(Member_ID);
ERROR 1061 (42000): Duplicate key name 'idx_schedule_member_id'
mysql> CREATE INDEX schedule_member_id_index ON SCHEDULE_(Member_ID);
Query OK, 0 rows affected, 1 warning (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 1

mysql> explain (
    -> SELECT M.First_Name, M.Trainer_ID, W.Duration_hrs, W.Start_Time
    -> FROM member as M
    -> LEFT JOIN SCHEDULE_ as W ON M.Member_ID = W.Member_ID;
+----+ | id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
| 1 | SIMPLE | M | NULL | ALL | NULL | idx_schedule_member_id,schedule_member_id_index | 122 | NULL | gym_management_system.M.Member_ID | 8 | 100.00 | NULL |
| 1 | SIMPLE | W | NULL | ref | idx_schedule_member_id | 122 | NULL | gym_management_system.M.Member_ID | 1 | 100.00 | Using index |
+----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

Figure 3.4.1 : Before tunned and after tuned for query 9

The screenshot shows a Windows desktop environment with a MySQL 8.0 Command Line Client window open. The window displays two sets of SQL queries and their execution plans. The first set of queries is for a table named 'MEMBER'. The second set is for a table named 'WORKOUT\_PLAN'. Both sets show the creation of indexes and the execution of SELECT statements with their respective EXPLAIN outputs. The EXPLAIN output provides details about the query execution plan, including select\_type, table, partitions, type, possible\_keys, key, key\_len, ref, rows, filtered, and Extra columns. The MySQL command line interface also shows various system status icons at the bottom.

```
mysql> explain (SELECT M.Member_ID
    -> FROM MEMBER as M
    -> LEFT JOIN WORKOUT_Plan as W ON M.Member_ID = W.Member_ID
    -> WHERE W.Member_ID IS NULL);
ERROR 1146 (42502): Table 'gym_management_system.workout_plan' doesn't exist
mysql> explain (SELECT M.Member_ID
    -> FROM MEMBER as M
    -> LEFT JOIN SCHEDULE_ as W ON M.Member_ID = W.Member_ID
    -> WHERE W.Member_ID IS NULL);
+----+ | id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
| 1 | SIMPLE | M | NULL | index | NULL | Member_index2 | 123 | NULL | gym_management_system.M.Member_ID | 8 | 100.00 | Using index |
| 1 | SIMPLE | W | NULL | ref | FK_schedule_member | 122 | NULL | gym_management_system.M.Member_ID | 1 | 100.00 | Using where; Not exists; Using index |
+----+
2 rows in set, 1 warning (0.00 sec)

mysql> CREATE INDEX idx_member_member_id ON MEMBER(Member_ID);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE INDEX idx_schedule_member_id ON SCHEDULE_(Member_ID);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain (SELECT M.Member_ID
    -> FROM MEMBER as M
    -> LEFT JOIN SCHEDULE_ as W ON M.Member_ID = W.Member_ID
    -> WHERE W.Member_ID IS NULL);
+----+ | id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
| 1 | SIMPLE | M | NULL | index | NULL | idx_member_member_id | 122 | NULL | gym_management_system.M.Member_ID | 8 | 100.00 | Using index |
| 1 | SIMPLE | W | NULL | ref | idx_schedule_member_id | 122 | NULL | gym_management_system.M.Member_ID | 1 | 100.00 | Using where; Not exists; Using index |
+----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

Figure 3.4.2 : Before tunned and after tuned for query 10

```

MySQL 8.0 Command Line Client
mysql> explain
mysql> SELECT COUNT(*) AS average_members_per_branch
-> FROM (
->   SELECT COUNT(*) AS members_per_branch
->   FROM member
->   JOIN branch AS b ON member.Branch_Num = b.Branch_Num
->   JOIN Trainer AS t ON m.Trainer_ID = t.Trainer_ID
->   WHERE t.Experience_Yrs >= 5
->   AND t.City = 'Galle'
-> ) AS branch_member_counts;
+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys |
+-----+-----+-----+-----+
| 1 | SIMPLE | derived | 1 | ALL | NULL | |
| 2 | DERIVED | B | 1 | ALL | ref | PRIMARY,branch_branch_num_index,branch_branch_num_index |
| 2 | DERIVED | T | 1 | ALL | ref | PRIMARY,trainer_branch_num_index,ex_years_index, idx_trainer_trainer_id, trainer_experience_yrs_index, idx_trainer_branch_num |
| 2 | DERIVED | b | 1 | ALL | ref | idx_member_branch_id |
+-----+-----+-----+-----+
1 rows in set, 1 warning (0.00 sec)

mysql> CREATE INDEX branch_branch_num_index ON branch(branch_num);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX trainer_branch_idx ON trainer(branch_num);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'UM' at line 1
mysql> CREATE INDEX trainer_branch_idx ON trainer(branch_num);
Query OK, 0 rows affected, 1 warning (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> EXPLAIN
-> SELECT COUNT(*) AS average_members_per_branch
-> FROM (
->   SELECT COUNT(*) AS members_per_branch
->   FROM member
->   JOIN branch AS b ON member.Branch_Num = b.Branch_Num
->   JOIN Trainer AS t ON m.Trainer_ID = t.Trainer_ID
->   WHERE t.Experience_Yrs >= 5
->   AND t.City = 'Galle'
-> ) AS branch_member_counts;
+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys |
+-----+-----+-----+-----+
| 1 | SIMPLE | derived | 1 | ALL | NULL | |
| 2 | DERIVED | M | 1 | ALL | ref | PRIMARY,branch_branch_num_index,branch_branch_num_index |
| 2 | DERIVED | T | 1 | ALL | ref | PRIMARY,trainer_branch_num_index,ex_years_index, idx_trainer_trainer_id, trainer_experience_yrs_index, idx_trainer_branch_num, trainer_branch_num |
| 2 | DERIVED | b | 1 | ALL | ref | idx_member_branch_id |
+-----+-----+-----+-----+
1 rows in set, 1 warning (0.00 sec)

mysql>

```

Figure 3.4.3 : Before tunned and after tuned for query 11

```

MySQL 8.0 Command Line Client
mysql> explain (Select
->   M.Member_ID , M.Trainer_ID, M.First_name as Member, T.First_Name as Trainer
->   from
->   Member as M
->   join
->   trainer as T
->   on M.Trainer_ID = T.Trainer_ID
->   where t.Experience_Yrs >= 5 and m.city = 'Galle';
+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys |
+-----+-----+-----+-----+
| 1 | SIMPLE | M | 1 | ALL | ref | FK_member_trainer(Member_Index2,Member_Index1) |
| 1 | SIMPLE | T | 1 | ALL | eq_ref | PRIMARY,trainer_Index2,EX_YEARS_Index |
+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> CREATE INDEX idx_member_trainer_id ON Member(Trainer_ID);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX idx_trainer_trainer_id ON Trainer(Trainer_ID);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX idx_trainer_experience_yrs_index ON Trainer(Experience_yrs);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '_index ON Trainer(Experience_yrs)' at line 1
mysql> CREATE INDEX idx_trainer_experience_yrs_index ON Trainer(Experience_yrs);
Query OK, 0 rows affected, 1 warning (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> explain (Select
->   M.Member_ID , M.Trainer_ID, M.First_name as Member, T.First_Name as Trainer
->   from
->   Member as M
->   join
->   trainer as T
->   on M.Trainer_ID = T.Trainer_ID
->   where t.Experience_Yrs >= 5 and m.city = 'Galle';
+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys |
+-----+-----+-----+-----+
| 1 | SIMPLE | M | 1 | ALL | ref | Member_Index2,Member_Index1, idx_member_training_id |
| 1 | SIMPLE | T | 1 | ALL | eq_ref | PRIMARY,trainer_Index2,EX_YEARS_Index, idx_trainer_trainer_id, trainer_experience_yrs_index |
+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>

```

Figure 3.4.4 : Before tunned and after tuned for query 13

```

MySQL 8.0 Command Line Client

mysql> explain (SELECT b.Branch_Num, m.First_Name AS Manager_Name, b.City, b.Town, b.Opening_Hours, b.Capacity,
    ->     b.Street_Name, b.Building_Num
    -> FROM BRANCH b
    -> LEFT OUTER JOIN MANAGER m ON b.Branch_Num = m.Branch_Num
    -> WHERE b.City LIKE 'C%' AND m.First_Name IS NOT NULL;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id  | select_type | table | partitions | type | possible_keys | key   | key_len | ref
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1   | SIMPLE      | b     | NULL       | range | PRIMARY,BRANCH_CITY_index | BRANCH_CITY_index | 162   | NULL
| 1   | SIMPLE      | m     | NULL       | ref   | FK_manager        | FK_manager        | 4     | gym_management_system.b.Branch_Num
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> CREATE INDEX branch_num_index ON BRANCH(Branch_Num);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE INDEX branch_MAN_num_index ON MANAGER(Branch_Num);
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain (SELECT b.Branch_Num, m.First_Name AS Manager_Name, b.City, b.Town, b.Opening_Hours, b.Capacity,
    ->     b.Street_Name, b.Building_Num
    -> FROM BRANCH b
    -> LEFT OUTER JOIN MANAGER m ON b.Branch_Num = m.Branch_Num
    -> WHERE b.City LIKE 'C%' AND m.First_Name IS NOT NULL;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id  | select_type | table | partitions | type | possible_keys | key   | key_len | ref
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1   | SIMPLE      | b     | NULL       | range | PRIMARY,BRANCH_CITY_index,branch_num_index | BRANCH_CITY_index | 162   | NULL
| 1   | SIMPLE      | m     | NULL       | ref   | branch_MAN_num_index | branch_MAN_num_index | 4     | gym_management_system.b.Branch_Num
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>

```

Figure 3.4.5 : Before tunned and after tuned for query 14

```

MySQL 8.0 Command Line Client

mysql> explain (SELECT t.Trainer_ID, t.First_Name AS Trainer_name, b.branch_num,
    -> t.Experience_yrs, b.opening_hours, b.Capacity,
    -> b.City AS Branch_City
    -> FROM TRAINER t
    -> RIGHT OUTER JOIN BRANCH b
    -> ON t.Branch_Num = b.Branch_Num
    -> WHERE t.First_Name LIKE 'K%' AND b.City = 'Kandy';
+-----+-----+-----+-----+-----+-----+-----+-----+
| id  | select_type | table | partitions | type | possible_keys | key   | key_len | ref
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1   | SIMPLE      | b     | NULL       | ALL  | PRIMARY      | NULL  | NULL   | NULL
| 1   | SIMPLE      | t     | NULL       | ref  | FK_trainer_branch | FK_trainer_branch | 4     | gym_management_system.b.Branch_Num
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> create index BRANCH_CITY_index on Branch(city);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain (SELECT t.Trainer_ID, t.First_Name AS Trainer_name, b.branch_num,
    -> t.Experience_yrs, b.opening_hours, b.Capacity,
    -> b.City AS Branch_City
    -> FROM TRAINER t
    -> RIGHT OUTER JOIN BRANCH b
    -> ON t.Branch_Num = b.Branch_Num
    -> WHERE t.First_Name LIKE 'K%' AND b.City = 'Kandy';
+-----+-----+-----+-----+-----+-----+-----+-----+
| id  | select_type | table | partitions | type | possible_keys | key   | key_len | ref
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1   | SIMPLE      | b     | NULL       | ref  | PRIMARY,BRANCH_CITY_index | BRANCH_CITY_index | 162   | const
| 1   | SIMPLE      | t     | NULL       | ref  | FK_trainer_branch        | FK_trainer_branch | 4     | gym_management_system.b.Branch_Num
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>

```

Figure 3.4.6 : Before tunned and after tuned for query 15

```

MySQL> explain (SELECT m.member_ID, m.MemType_ID, m.first_name, m.city, p.Time_, p.Date_, p.amount
   FROM member AS m
   LEFT JOIN payment AS p
   ON m.member_ID = p.Member_ID
   WHERE m.first_name LIKE '%Ka'
   AND p.date_ = '2024-02-01')
   UNION
   (SELECT m.member_ID, m.MemType_ID, m.first_name, m.city, p.Time_, p.Date_, p.amount
   FROM member AS m
   RIGHT JOIN payment AS p
   ON m.member_ID = p.Member_ID
   WHERE m.first_name LIKE '%Ka'
   AND p.date_ = '2024-02-01');
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | p | NULL | index | FK_payment_Member | FK_payment_Member | 122 | NULL |
| 1 | PRIMARY | m | NULL | eq_ref | PRIMARY | PRIMARY | 122 | gYM_management_system.p.Member_ID |
| 2 | PRIMARY | p | NULL | index | FK_payment_Member | FK_payment_Member | 122 | NULL |
| 2 | PRIMARY | m | NULL | eq_ref | PRIMARY | PRIMARY | 122 | gYM_management_system.p.Member_ID |
| 3 | UNION | m | NULL | ALL | NULL | NULL | NULL | NULL |
| 3 | UNION RESULT | union1,> | NULL | ALL | NULL | NULL | NULL | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set, 1 warning (0.00 sec)

mysql> create index firstname_index on MEMBER(first_name);
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> create index Date_Index on Payment(Date);
ERROR 1072 (42000): Key column 'Date' doesn't exist in table
mysql> create index Date_Index on Payment(Date_);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain (SELECT m.member_ID, m.MemType_ID, m.first_name, m.city, p.Time_, p.Date_, p.amount
   FROM member AS m
   LEFT JOIN payment AS p
   ON m.member_ID = p.Member_ID
   WHERE m.first_name LIKE '%Ka'
   AND p.date_ = '2024-02-01')
   UNION
   (SELECT m.member_ID, m.MemType_ID, m.first_name, m.city, p.Time_, p.Date_, p.amount
   FROM member AS m
   RIGHT JOIN payment AS p
   ON m.member_ID = p.Member_ID
   WHERE m.first_name LIKE '%Ka'
   AND p.date_ = '2024-02-01');
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | p | NULL | ref | FK_payment_Member,Date_Index | Date_Index | 3 | const |
| 1 | PRIMARY | m | NULL | eq_ref | PRIMARY | PRIMARY | 122 | gYM_management_system.p.Member_ID |
| 2 | PRIMARY | p | NULL | ref | FK_payment_Member,Date_Index | Date_Index | 3 | const |
| 2 | PRIMARY | m | NULL | eq_ref | PRIMARY | PRIMARY | 122 | gYM_management_system.p.Member_ID |
| 3 | UNION | m | NULL | ALL | NULL | NULL | NULL | NULL |
| 3 | UNION RESULT | union1,> | NULL | ALL | NULL | NULL | NULL | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set, 1 warning (0.00 sec)

mysql>

```

Figure 3.4.7 : Before tunned and after tunned for query 16

```

MySQL> explain (SELECT ID, Type_Name, price, Type
   FROM (
   >   SELECT MemType_ID AS ID, Type_Name, price, 'MS_Type' AS Type
   >   FROM membership_type
   >   WHERE Type_Name LIKE '%Gold%' OR Type_Name LIKE '%Platinum%'
   >   UNION
   >   SELECT Member_ID AS ID, CONVERT(Duration_hrs, CHAR), Start_Time, 'Plan' AS Type
   >   FROM SCHEDULE_
   >   WHERE Start_Time BETWEEN '08:00:00' AND '12:00:00'
   > ) AS Combined_Data);
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | <derived2> | NULL | ALL | NULL | NULL | NULL | NULL |
| 2 | DERIVED | membership_type | NULL | ALL | NULL | NULL | NULL | NULL |
| 3 | UNION | SCHEDULE_| NULL | index | FK_schedule_member | FK_schedule_member | 122 | NULL |
| 4 | UNION RESULT | union2,>> | NULL | ALL | NULL | NULL | NULL | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 1 warning (0.00 sec)

mysql> create index STARTTIME_index on SCHEDULE_(START_Time);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain (SELECT ID, Type_Name, price, Type
   FROM (
   >   SELECT MemType_ID AS ID, Type_Name, price, 'MS_Type' AS Type
   >   FROM membership_type
   >   WHERE Type_Name LIKE '%Gold%' OR Type_Name LIKE '%Platinum%'
   >   UNION
   >   SELECT Member_ID AS ID, CONVERT(Duration_hrs, CHAR), Start_Time, 'Plan' AS Type
   >   FROM SCHEDULE_
   >   WHERE Start_Time BETWEEN '08:00:00' AND '12:00:00'
   > ) AS Combined_Data);
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | <derived2> | NULL | ALL | NULL | NULL | NULL | NULL |
| 2 | DERIVED | membership_type | NULL | ALL | NULL | NULL | NULL | NULL |
| 3 | UNION | SCHEDULE_| NULL | index | FK_schedule_member,STARTTIME_index | FK_schedule_member | 122 | NULL |
| 4 | UNION RESULT | union2,>3> | NULL | ALL | NULL | NULL | NULL | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 1 warning (0.00 sec)

mysql>

```

Figure 3.4.8 : : Before tunned and after tunned for query 17

```

MySQL 8.0 Command Line Client
-> WHERE T.Experience_Years > 5;
ERROR 1054 (42S22): Unknown column 'M.City' in 'where clause'
mysql> explain (SELECT *
->   FROM (
->     SELECT Member_ID, Trainer_ID, First_Name, Last_Name
->       FROM MEMBER
->      WHERE City = 'Galle'
->    ) AS Galle_Members
-> INNER JOIN (
->     SELECT Trainer_ID, First_Name, Last_Name
->       FROM TRAINER
->      WHERE City = 'Galle'
->    ) AS Galle_Trainers ON Galle_Members.Trainer_ID = Galle_Trainers.Trainer_ID;
+----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1  | SIMPLE      | TRAINER | NULL      | ALL  | PRIMARY      | NULL | NULL    | NULL  |
| 1  | SIMPLE      | MEMBER   | NULL      | ref   | FK_member_trainer,Member_index2 | FK_member_trainer | 182  | gym_management_system.TRAINER.Trainer_ID |
+----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> create index Member_index1 on Member(city);
Query OK, 0 rows affected, 1 warning (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 1

mysql> create index Member_index1 on Trainer(city);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain (SELECT *
->   FROM (
->     SELECT Member_ID, Trainer_ID, First_Name, Last_Name
->       FROM MEMBER
->      WHERE City = 'Galle'
->    ) AS Galle_Members
-> INNER JOIN (
->     SELECT Trainer_ID, First_Name, Last_Name
->       FROM TRAINER
->      WHERE City = 'Galle'
->    ) AS Galle_Trainers ON Galle_Members.Trainer_ID = Galle_Trainers.Trainer_ID;
+----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1  | SIMPLE      | TRAINER | NULL      | ALL  | PRIMARY,Member_index1 | Member_index1 | 122  | const |
| 1  | SIMPLE      | MEMBER   | NULL      | ref   | FK_member_trainer,Member_index2,Member_index1 | FK_member_trainer | 182  | gym_management_system.TRAINER.Trainer_ID |
+----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>

```

Figure 3.4.9 : Before tunned and after tuned for query 18

```

MySQL 8.0 Command Line Client
-> WHERE Mtype_ID > (
->   SELECT Mtype_ID
->     FROM MEMBERSHIP_TYPE
->    WHERE Type_Name = 'Silver'
->  )
-> AND Trainer_ID IN (
->   SELECT Trainer_ID
->     FROM TRAINER
->    WHERE Experience_Yrs > 4
->  );
+----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1  | PRIMARY    | MEMBER  | NULL      | ref  | FK_member_trainer,Mtype_ID_Index | Mtype_ID_Index | 182  | const |
| 1  | PRIMARY    | TRAINERS | NULL      | ref  | FK_member_trainer,Trainer_index | Trainer_index | 182  | const |
| 2  | SUBQUERY   |          |           |      |              |      |      |      |
| 3  |          |          |           |      |              |      |      |      |
+----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql> explain(SELECT *
->   FROM MEMBER
->  WHERE Mtype_ID > (
->    SELECT Mtype_ID
->      FROM MEMBERSHIP_TYPE
->     WHERE Type_Name = 'Silver'
->   )
->   AND Trainer_ID IN (
->     SELECT Trainer_ID
->       FROM TRAINER
->      WHERE Experience_Yrs > 4
->   ));
+----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1  | PRIMARY    | MEMBER  | NULL      | ref  | FK_member_trainer,Mtype_ID_Index | Mtype_ID_Index | 182  | const |
| 1  | PRIMARY    | TRAINERS | NULL      | ref  | FK_member_trainer,Trainer_index | Trainer_index | 182  | const |
| 2  | SUBQUERY   |          |           |      |              |      |      |      |
| 3  |          |          |           |      |              |      |      |      |
+----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql>

```

Figure 3.4.10 : Before tunned and after tuned for query 19