



Mini Project Final Report

A report submitted to the

Department of Electrical and Information Engineering
Faculty of Engineering
University of Ruhuna
Sri Lanka

On 04th April 2024

In completing a report for the module

EE4350 Database Systems

By

GROUP NO : 54

RASANJALI A.J.U. –EG/2021/4747

RASHMI E.D.K –EG/2021/4748

RATHNAWEERA P.A.G.N. –EG/2021/4751

Table of Contents

Chapter 1 .– Requirement Analysis	5
1. Functional Requirements	5
2. Data Requirements.....	5
Chapter 2 - Conceptual Design	8
Chapter 3 – Implementation.....	10
1. Schema Creation	10
Chapter 4 - Transactions	31
1. Simple Queries.....	31
2. Complex Queries	35
Chapter 5 - Database Tuning	42
1. Union.....	42
2. Intersection.....	43
3. Set Difference	44
4. Division.....	45
5. Inner join.....	46
6. Natural Join	47
7. Right Join	48
8. Nested Queries	49
9. Full outer join.....	50
10. Nested Query 2.....	51

Table of Figures

Figure 1. 1- ER Diagram.....	8
Figure 1. 2- UML Class Diagram	9

Introduction

In the modern era, the importance of health and fitness has garnered significant attention. People are becoming increasingly conscious about their physical well-being and are actively seeking ways to monitor and improve their fitness levels. With the proliferation of technology, fitness tracking systems have emerged as popular tools for individuals to track their progress, set goals, and stay motivated towards achieving their fitness objectives.

The Fitness Tracking Management System (FTMS) is a comprehensive solution designed to address the needs of fitness enthusiasts by providing a platform for tracking and managing various aspects of their fitness journey. This system leverages the power of technology, particularly databases, to organize, store, and analyze fitness-related data efficiently.

Chapter 1 – Requirement Analysis

1. Functional Requirements

This proposed system which is for a fitness tracking system satisfy below functional requirements in order to perform the necessary tasks for the usage of the users regarding the system. So the users that consume this system can operate the system for their personal usage with these functions.

1. Simple data access and data retrieval of user data, exercise data and other related data
2. Relational schema which is in the second normal form
3. Adaptability of the database according to the user preferences
4. Simple user registration process
5. Customizable exercises, workouts
6. Effective nutritional tracking
7. Better engagement between users and the systems
8. Secured data privacy
9. Performance optimization
10. Data backup and recovery

2. Data Requirements

The detailed overview of the database about the attributes, relationships are discussed here.

Furthermore, the data constraints and the other necessary data requirements are displayed in the context of relational schema.

Each entity attributes are displayed below.

User

1. User ID
2. Name
3. Age
4. Phone number
5. Address
6. Height
7. Weight
8. BMI

Workout:

1. Workout ID
2. User ID
3. Duration
4. Start Time
5. Workout_name
6. Calories_burnt

Exercise:

1. Exercise_ID
2. Workout_ID
3. Exercise_name
4. Category

Nutrition_log:

1. User_ID
2. Calories_taken
3. Log_date_time

Exercise_log:

1. Log_ID
2. Exercise_ID
3. Sets
4. Repetitions
5. Weight

Equipment type:

1. Equipment_ID
2. Exercise_ID
3. Equipment_name
4. Duration

Location:

1. Location_ID
2. Location_name
3. Province
4. City

Location_Content:

1. Equipment ID
2. Location ID
3. Quantity

Herat_Rate_Recorded:

1. Log_ID
2. Heart_rate
3. Recorder time

In our Fitness Tracking Management System, we have identified three **weak entities**: Nutrition Log, Heart Rate Record, and Location Content. These entities are dependent on other entities and do not have their own unique identifiers. The Nutrition Log stores details of users' daily nutritional intake, the Heart Rate Record records users' heart rate measurements, and the Location Content tracks users' workout locations or routes.

Additionally, we have established a **recursive relationship** called "User Follows User," which allows users to connect with and follow other users within the system. This relationship enables social interaction and networking among users, enhancing engagement and motivation.

Furthermore, we have identified the **multivalued attribute** "Phone Number," which allows users to have multiple phone numbers associated with their account. Additionally, we have **composite attributes** for storing names (First Name and Last Name) and durations (Hours, Minutes, and Seconds) efficiently within our database schema. These attributes provide a structured approach to managing complex data and enhancing the usability of the system.

Chapter 2 - Conceptual Design

The conceptual representation of the database schema is represented by the ER diagram using the visual paradigm tool.

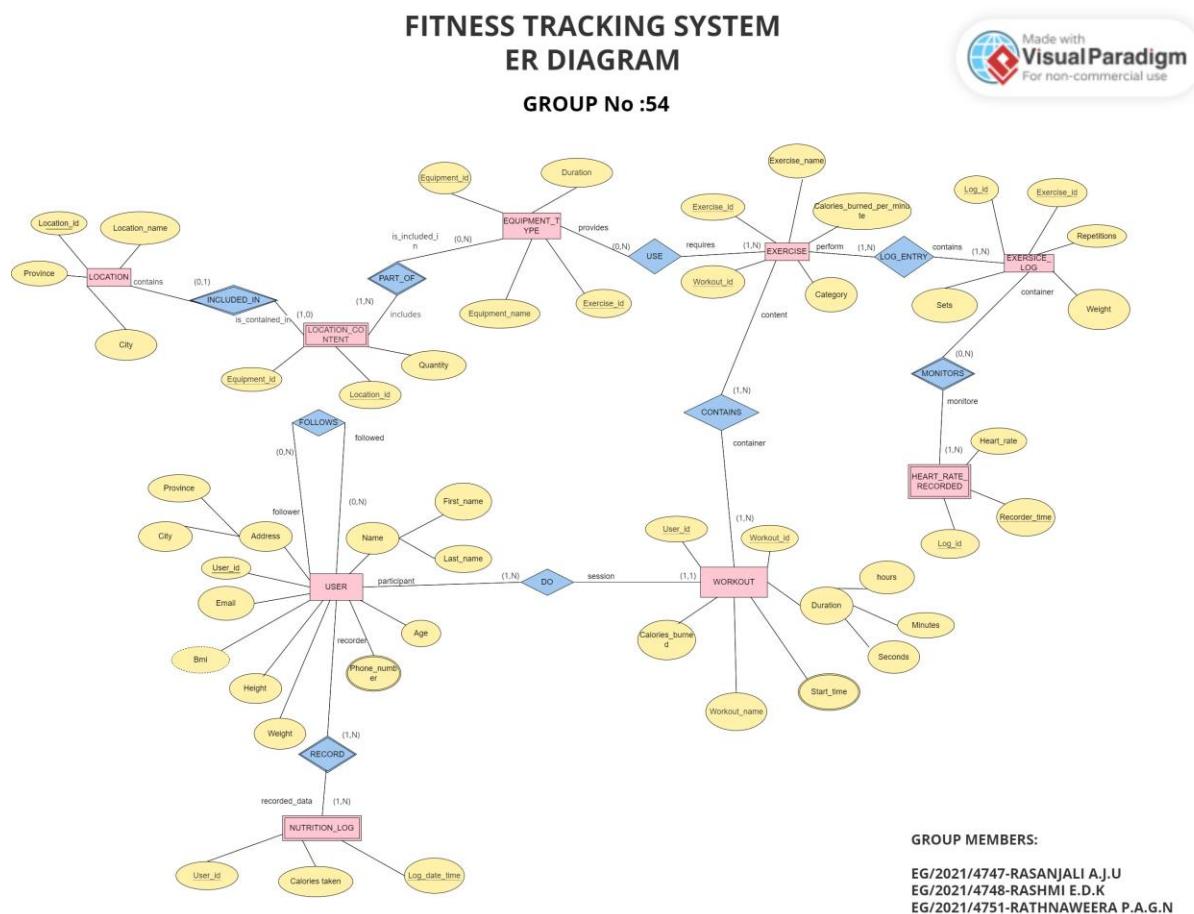


Figure 1. 1- ER Diagram

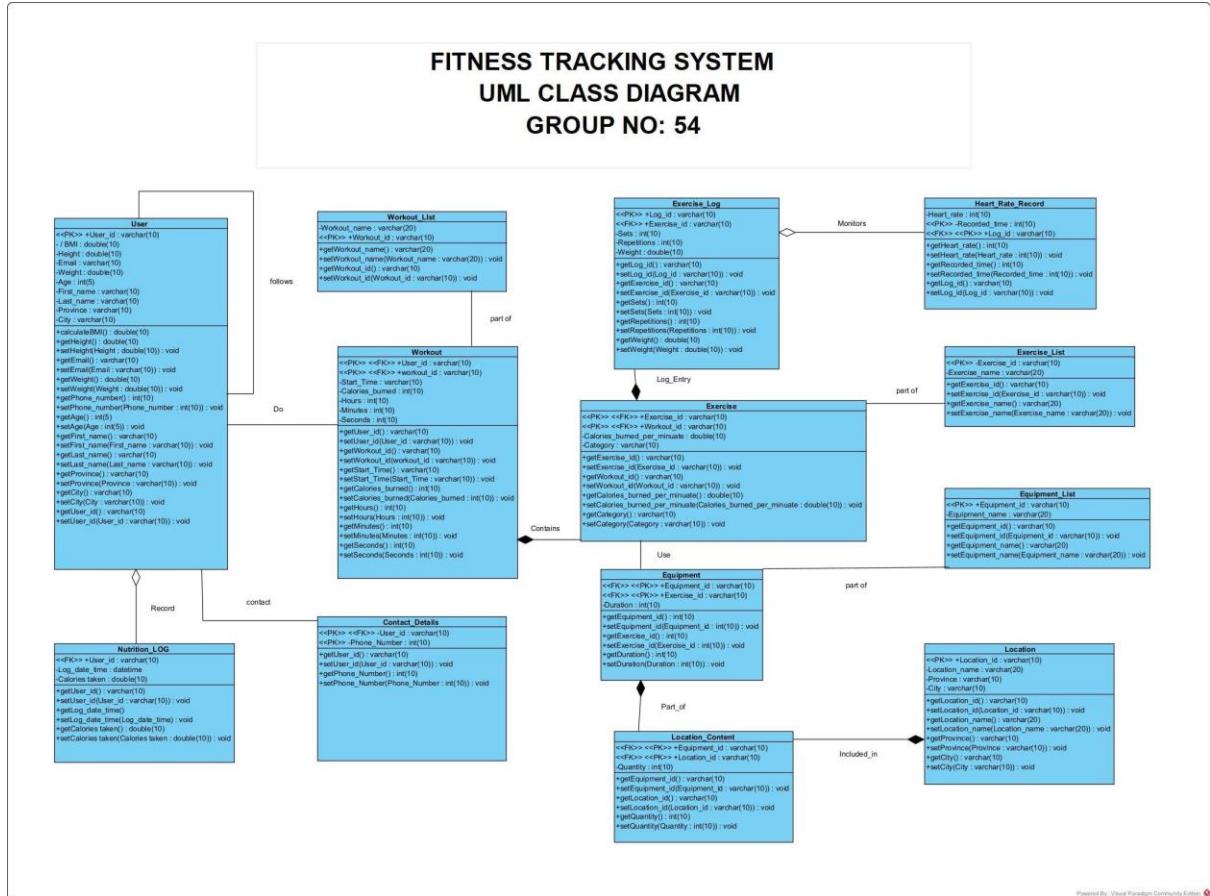


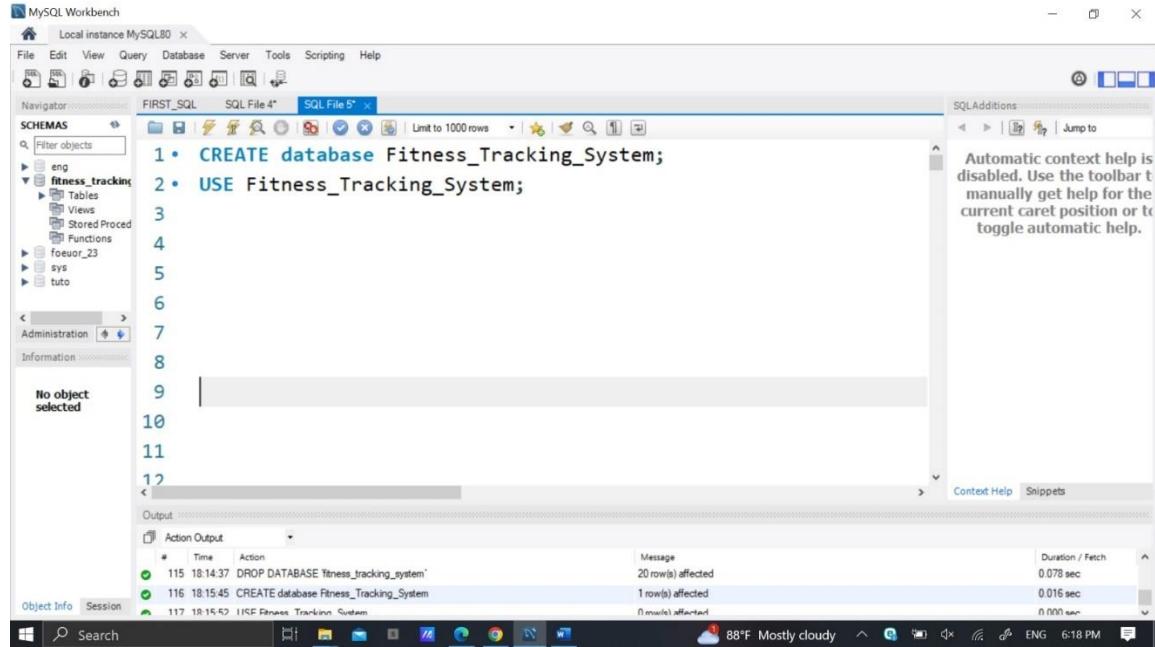
Figure 1. 2- UML Class Diagram

Chapter 3 – Implementation

After designing the ER diagram then the schema can be mapped into tabular format in the relational model which are in the second normal form.

1. Schema Creation

First we create our database name as fitness tracking system. The creation of schema as below.



The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL80' is selected. The 'Navigator' pane on the left shows the 'SCHEMAS' section with 'Fitness_tracking' expanded, revealing 'Tables', 'Views', 'Stored Proced', 'Functions', and 'foeuro_23'. The main query editor window contains the following SQL code:

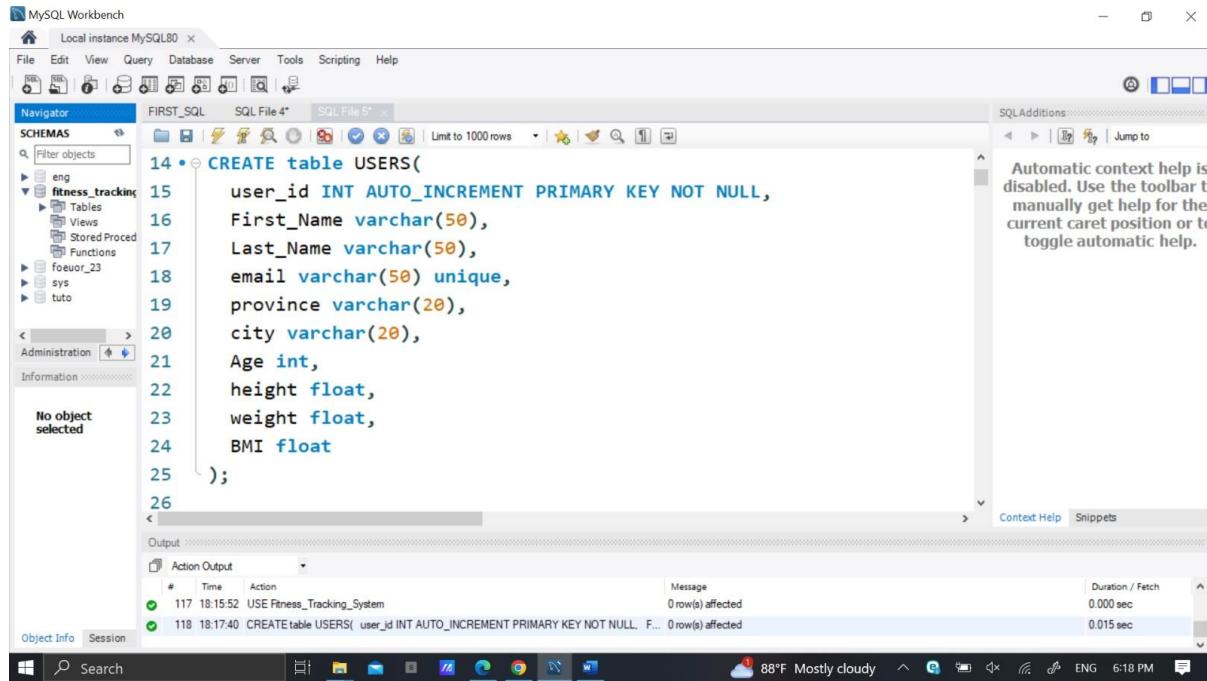
```
1 • CREATE database Fitness_Tracking_System;
2 • USE Fitness_Tracking_System;
```

The 'Output' pane at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
115	18:14:37	DROP DATABASE 'fitness_tracking_system'	20 row(s) affected	0.078 sec
116	18:14:54	CREATE database Fitness_Tracking_System	1 row(s) affected	0.016 sec
117	18:15:42	USE Fitness_Tracking_System	0 row(s) affected	0.000 sec

Users Table

• Definition



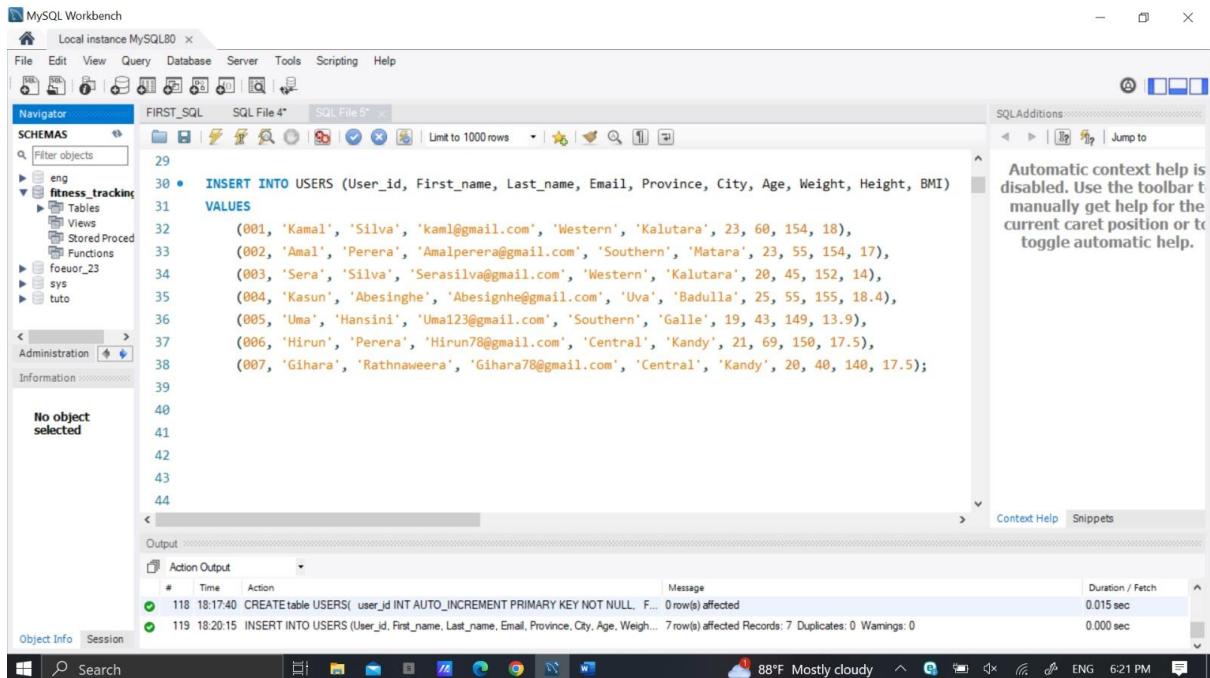
The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left shows the 'SCHEMAS' section with 'Fitness_tracking' expanded, revealing 'Tables', 'Views', 'Stored Proced', 'Functions', and 'foeuro_23'. The main query editor window contains the following SQL code:

```
14 • CREATE table USERS(
15     user_id INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
16     First_Name varchar(50),
17     Last_Name varchar(50),
18     email varchar(50) unique,
19     province varchar(20),
20     city varchar(20),
21     Age int,
22     height float,
23     weight float,
24     BMI float
25 );
```

The 'Output' pane at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
117	18:15:52	USE Fitness_Tracking_System	0 row(s) affected	0.000 sec
118	18:17:40	CREATE table USERS(user_id INT AUTO_INCREMENT PRIMARY KEY NOT NULL, F...	0 row(s) affected	0.015 sec

- Data Insertion



The screenshot shows the MySQL Workbench interface with the following details:

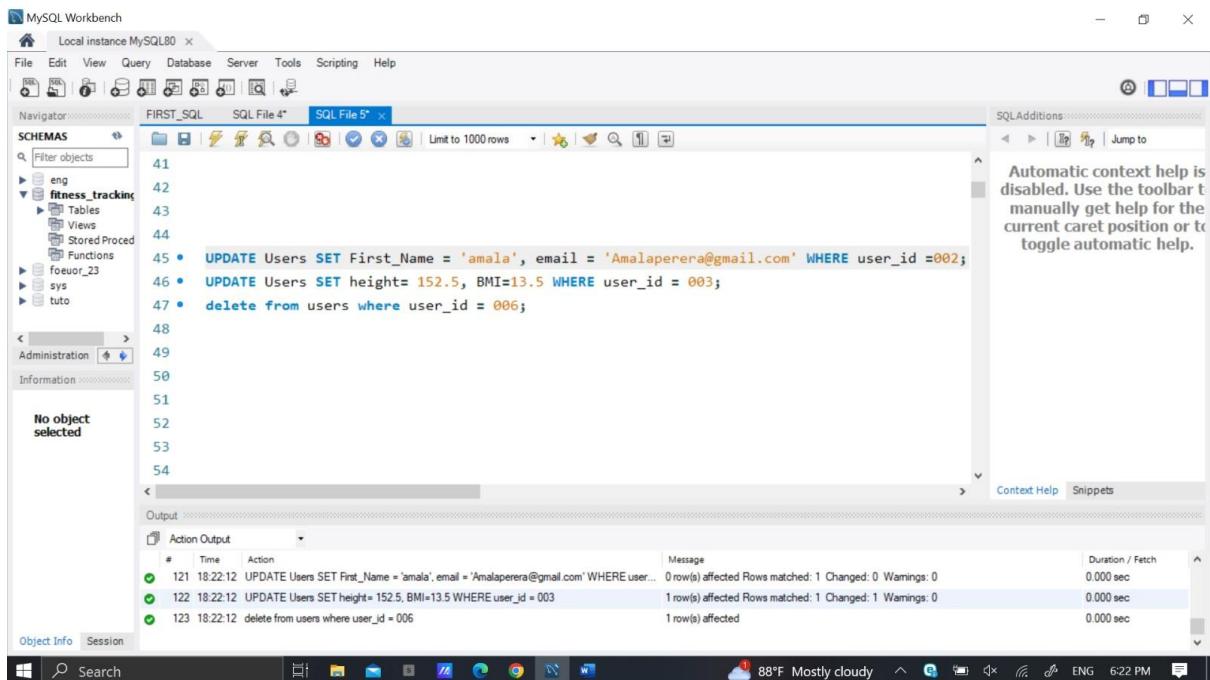
- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** eng, fitness_tracking (selected), foever_23, sys, tuto. A message "No object selected" is displayed.
- SQL Editor:** SQL File 5* containing the following SQL code:


```

29
30 • INSERT INTO USERS (User_id, First_name, Last_name, Email, Province, City, Age, Weight, Height, BMI)
31   VALUES
32     (001, 'Kamal', 'Silva', 'kamal@gmail.com', 'Western', 'Kalutara', 23, 60, 154, 18),
33     (002, 'Amal', 'Perera', 'Amalperera@gmail.com', 'Southern', 'Matara', 23, 55, 154, 17),
34     (003, 'Sera', 'Silva', 'Serasilva@gmail.com', 'Western', 'Kalutara', 20, 45, 152, 14),
35     (004, 'Kasun', 'Abesinghe', 'Abesinghe@gmail.com', 'Uva', 'Badulla', 25, 55, 155, 18.4),
36     (005, 'Uma', 'Hansini', 'Uma123@gmail.com', 'Southern', 'Galle', 19, 43, 149, 13.9),
37     (006, 'Hirun', 'Perera', 'Hirun78@gmail.com', 'Central', 'Kandy', 21, 69, 150, 17.5),
38     (007, 'Gihara', 'Rathnaweera', 'Gihara78@gmail.com', 'Central', 'Kandy', 20, 40, 140, 17.5);
39
40
41
42
43
44
      
```
- Output Window:** Shows the execution results:

#	Time	Action	Message	Duration / Fetch
118	18:17:40	CREATE table	118 18:17:40 CREATE table USERS(user_id INT AUTO_INCREMENT PRIMARY KEY NOT NULL, F... 0 rows) affected	0.015 sec
119	18:20:15	INSERT INTO	119 18:20:15 INSERT INTO USERS (User_id, First_name, Last_name, Email, Province, City, Age, Weight, Height, BMI) VALUES (001, 'Kamal', 'Silva', 'kamal@gmail.com', 'Western', 'Kalutara', 23, 60, 154, 18), (002, 'Amal', 'Perera', 'Amalperera@gmail.com', 'Southern', 'Matara', 23, 55, 154, 17), (003, 'Sera', 'Silva', 'Serasilva@gmail.com', 'Western', 'Kalutara', 20, 45, 152, 14), (004, 'Kasun', 'Abesinghe', 'Abesinghe@gmail.com', 'Uva', 'Badulla', 25, 55, 155, 18.4), (005, 'Uma', 'Hansini', 'Uma123@gmail.com', 'Southern', 'Galle', 19, 43, 149, 13.9), (006, 'Hirun', 'Perera', 'Hirun78@gmail.com', 'Central', 'Kandy', 21, 69, 150, 17.5), (007, 'Gihara', 'Rathnaweera', 'Gihara78@gmail.com', 'Central', 'Kandy', 20, 40, 140, 17.5); 7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0	0.000 sec
- System Tray:** Shows the date and time as 88°F Mostly cloudy, 8:21 PM.

- Update and Delete Operations



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** eng, fitness_tracking (selected), foever_23, sys, tuto. A message "No object selected" is displayed.
- SQL Editor:** SQL File 5* containing the following SQL code:

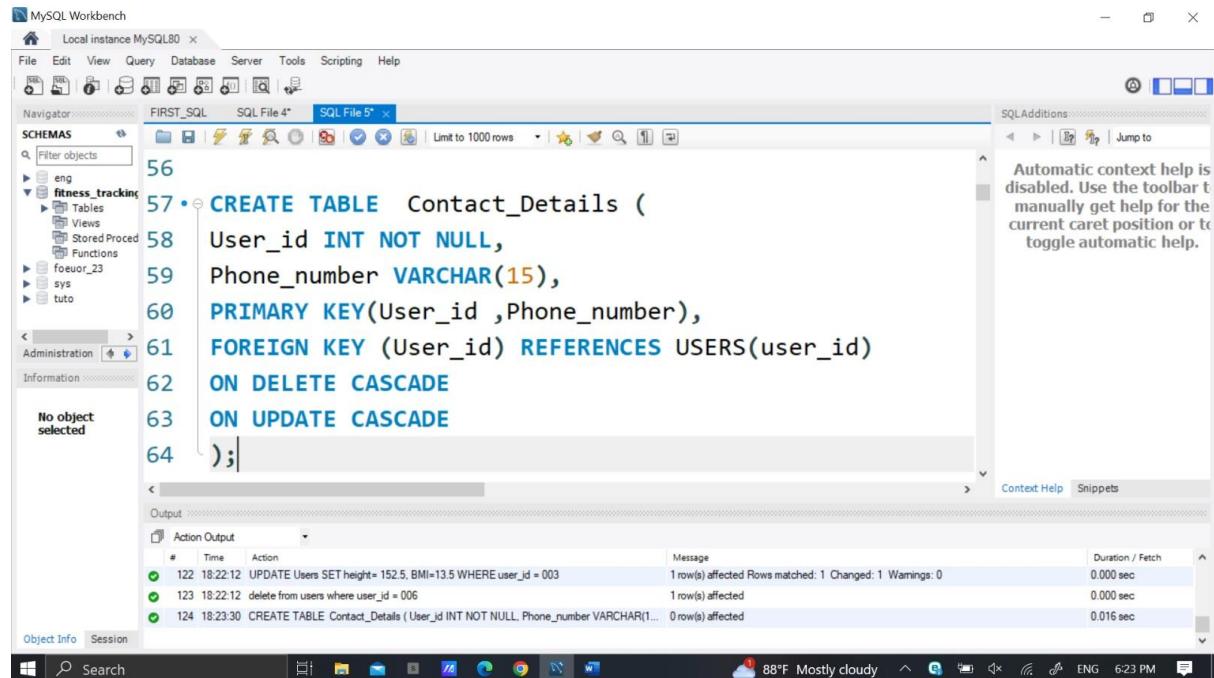

```

41
42
43
44
45 • UPDATE Users SET First_Name = 'amala', email = 'Amalaperera@gmail.com' WHERE user_id = 002;
46 • UPDATE Users SET height= 152.5, BMI=13.5 WHERE user_id = 003;
47 • delete from users where user_id = 006;
48
49
50
51
52
53
54
      
```
- Output Window:** Shows the execution results:

#	Time	Action	Message	Duration / Fetch
121	18:22:12	UPDATE Users SET First_Name = 'amala', email = 'Amalaperera@gmail.com' WHERE user_id = 002;	121 18:22:12 UPDATE Users SET First_Name = 'amala', email = 'Amalaperera@gmail.com' WHERE user_id = 002; 0 rows) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec
122	18:22:12	UPDATE Users SET height= 152.5, BMI=13.5 WHERE user_id = 003;	122 18:22:12 UPDATE Users SET height= 152.5, BMI=13.5 WHERE user_id = 003; 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
123	18:22:12	delete from users where user_id = 006	123 18:22:12 delete from users where user_id = 006; 1 row(s) affected	0.000 sec
- System Tray:** Shows the date and time as 88°F Mostly cloudy, 8:22 PM.

Contact Details Table

- Definition



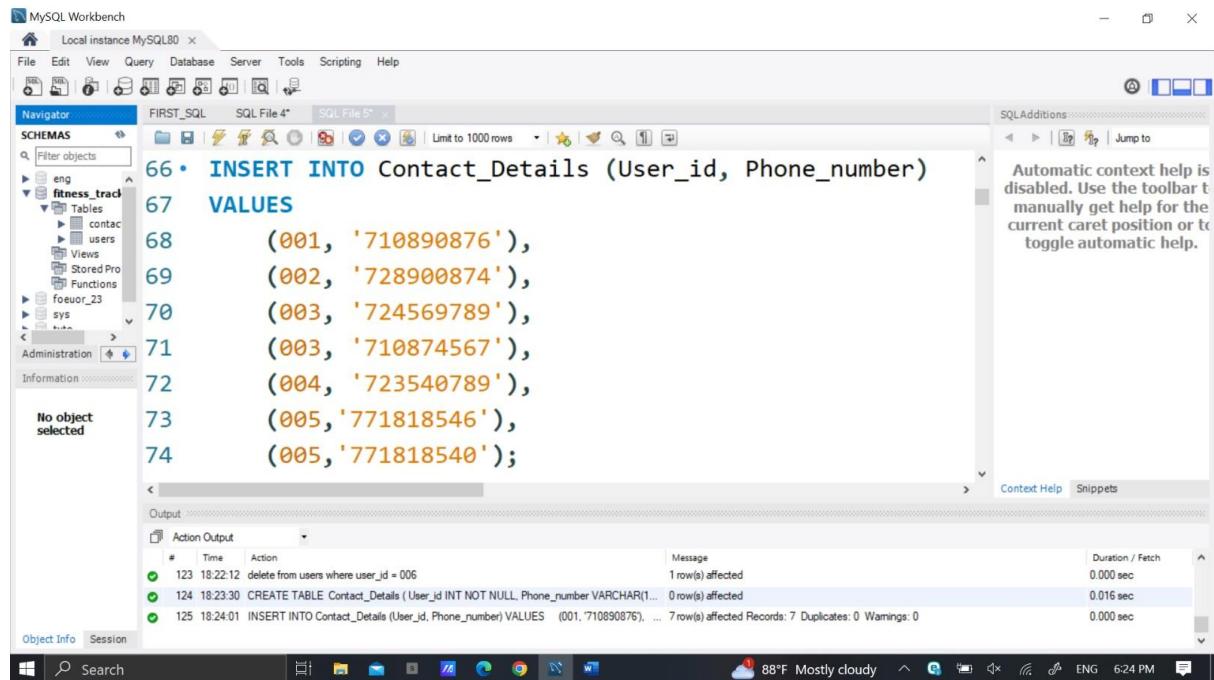
The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab active. In the SQL pane, the following SQL code is written:

```
56
57 • CREATE TABLE Contact_Details (
58     User_id INT NOT NULL,
59     Phone_number VARCHAR(15),
60     PRIMARY KEY(User_id ,Phone_number),
61     FOREIGN KEY (User_id) REFERENCES USERS(user_id)
62     ON DELETE CASCADE
63     ON UPDATE CASCADE
64 );
```

The 'Output' pane shows the results of the execution:

#	Time	Action	Message	Duration / Fetch
122	18:22:12	UPDATE Users SET height= 152.5, BMI=13.5 WHERE user_id = 003	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
123	18:22:12	delete from users where user_id = 006	1 row(s) affected	0.000 sec
124	18:23:30	CREATE TABLE Contact_Details (User_id INT NOT NULL, Phone_number VARCHAR(1...)	0 row(s) affected	0.016 sec

- Data Insertion



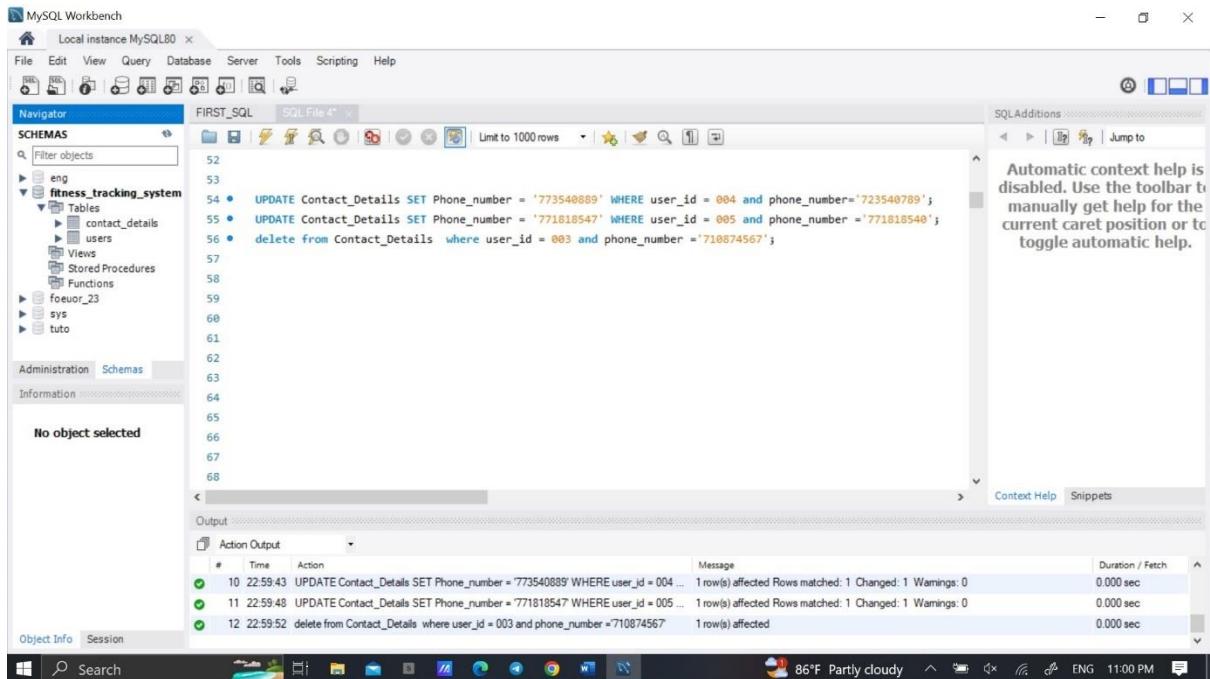
The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab active. In the SQL pane, the following SQL code is written:

```
66 • INSERT INTO Contact_Details (User_id, Phone_number)
67     VALUES
68         (001, '710890876'),
69         (002, '728900874'),
70         (003, '724569789'),
71         (003, '710874567'),
72         (004, '723540789'),
73         (005, '771818546'),
74         (005, '771818540');
```

The 'Output' pane shows the results of the execution:

#	Time	Action	Message	Duration / Fetch
123	18:22:12	delete from users where user_id = 006	1 row(s) affected	0.000 sec
124	18:23:30	CREATE TABLE Contact_Details (User_id INT NOT NULL, Phone_number VARCHAR(1...)	0 row(s) affected	0.016 sec
125	18:24:01	INSERT INTO Contact_Details (User_id, Phone_number) VALUES (001,710890876), ...	7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0	0.000 sec

- Update and Delete Operations



```

54 • UPDATE Contact_Details SET Phone_number = '773540889' WHERE user_id = 004 and phone_number='723540789';
55 • UPDATE Contact_Details SET Phone_number = '771818547' WHERE user_id = 005 and phone_number = '771818540';
56 • delete from Contact_Details where user_id = 003 and phone_number = '710874567';

62
63
64
65
66
67
68

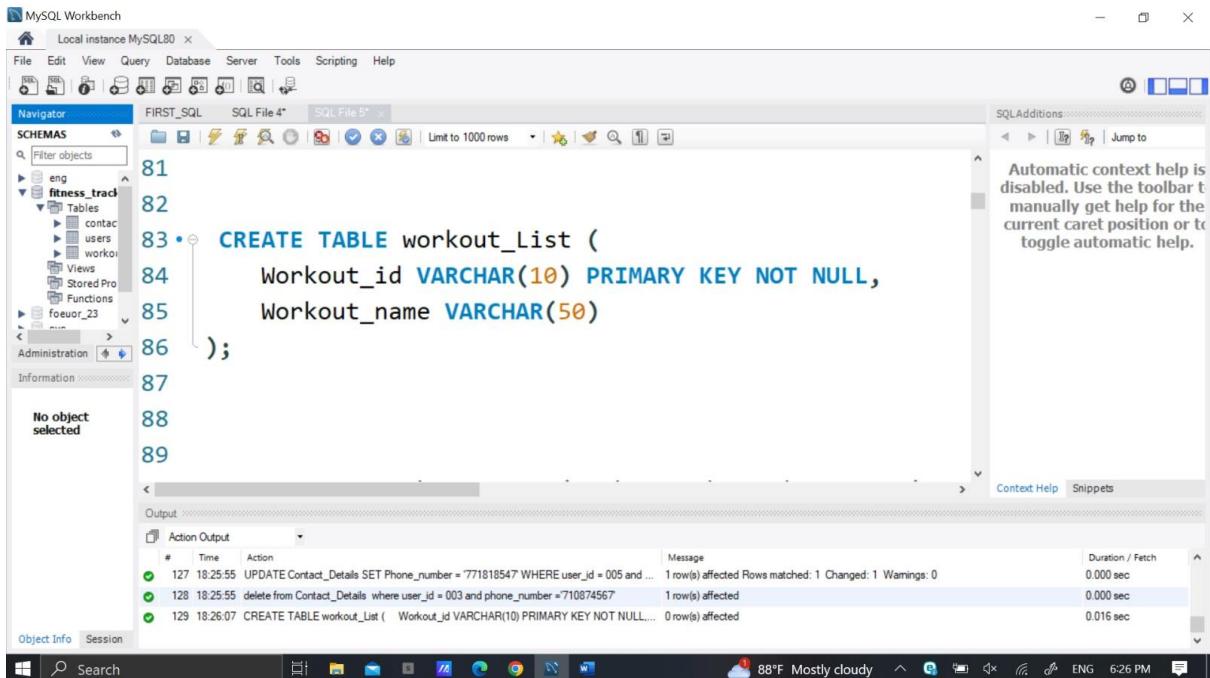
```

Action Output

#	Time	Action	Message	Duration / Fetch
10	22:59:43	UPDATE Contact_Details SET Phone_number = 773540889 WHERE user_id = 004 ...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
11	22:59:48	UPDATE Contact_Details SET Phone_number = 771818547 WHERE user_id = 005 ...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
12	22:59:52	delete from Contact_Details where user_id = 003 and phone_number = 710874567	1 row(s) affected	0.000 sec

Workout_List Table

- Definition



```

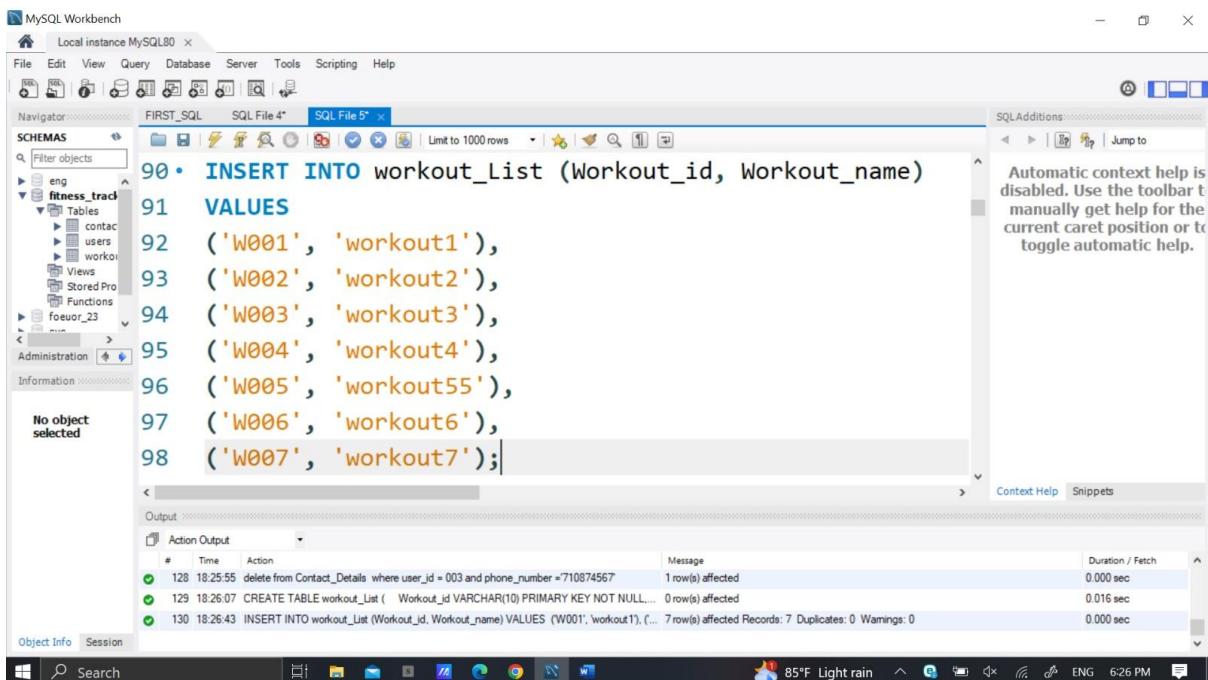
81
82
83 • CREATE TABLE workout_List (
84     Workout_id VARCHAR(10) PRIMARY KEY NOT NULL,
85     Workout_name VARCHAR(50)
86 );
87
88
89

```

Action Output

#	Time	Action	Message	Duration / Fetch
127	18:25:55	UPDATE Contact_Details SET Phone_number = 771818547 WHERE user_id = 005 and ...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
128	18:25:55	delete from Contact_Details where user_id = 003 and phone_number = 710874567	1 row(s) affected	0.000 sec
129	18:26:07	CREATE TABLE workout_List (Workout_id VARCHAR(10) PRIMARY KEY NOT NULL,...	0 row(s) affected	0.016 sec

- Data Insertion



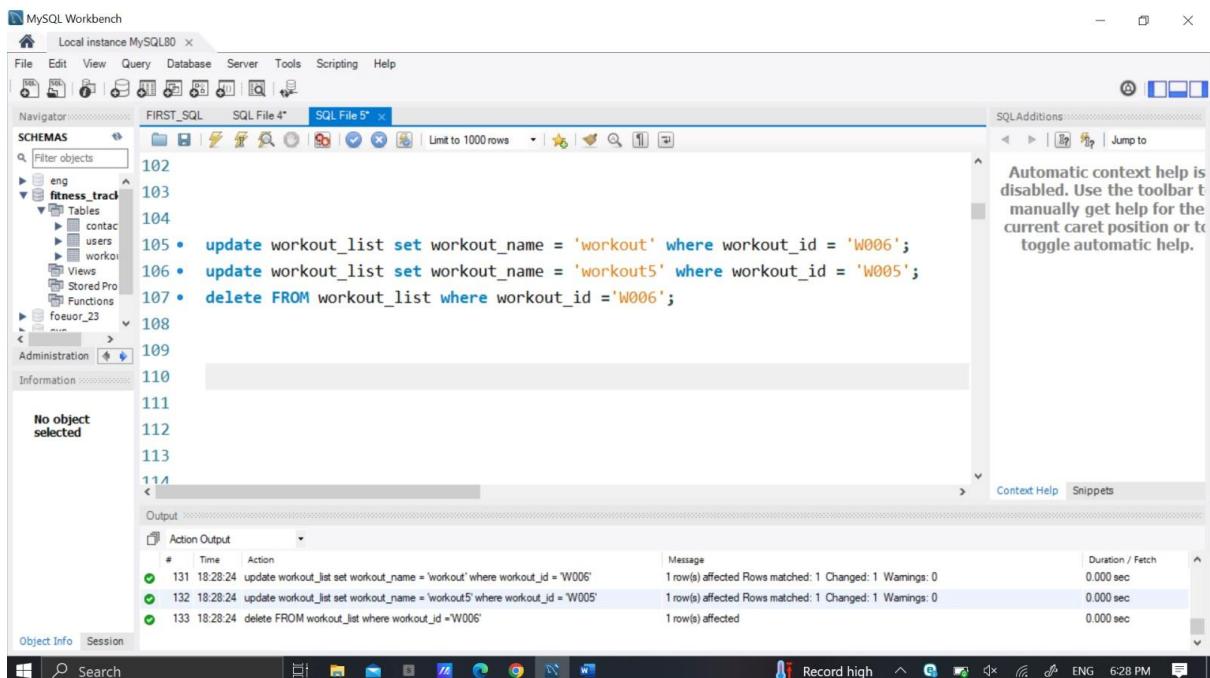
The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (eng, fitness_track), Tables (contact, users, workout), Views, Stored Procs, Functions, foetur_23.
- SQL Editor:** SQL File 5* tab, containing the following SQL code:


```
90 • INSERT INTO workout_List (Workout_id, Workout_name)
91     VALUES
92     ('W001', 'workout1'),
93     ('W002', 'workout2'),
94     ('W003', 'workout3'),
95     ('W004', 'workout4'),
96     ('W005', 'workout55'),
97     ('W006', 'workout6'),
98     ('W007', 'workout7');
```
- Output Window:** Action Output table showing the results of the insert operation:

#	Time	Action	Message	Duration / Fetch
128	18:25:55	delete from Contact_Details where user_id = 003 and phone_number > 710874567	1 row(s) affected	0.000 sec
129	18:26:07	CREATE TABLE workout_List (Workout_id VARCHAR(10) PRIMARY KEY NOT NULL,...	0 row(s) affected	0.016 sec
130	18:26:43	INSERT INTO workout_List (Workout_id, Workout_name) VALUES ('W001', 'workout1'), ('W002', 'workout2'), ('W003', 'workout3'), ('W004', 'workout4'), ('W005', 'workout55'), ('W006', 'workout6'), ('W007', 'workout7');	7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0	0.000 sec
- System Tray:** Shows weather (85°F Light rain), battery level, and system status.

- Update and Delete Operations



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (eng, fitness_track), Tables (contact, users, workout), Views, Stored Procs, Functions, foetur_23.
- SQL Editor:** SQL File 5* tab, containing the following SQL code:


```
102
103
104
105 • update workout_list set workout_name = 'workout' where workout_id = 'W006';
106 • update workout_list set workout_name = 'workouts' where workout_id = 'W005';
107 • delete FROM workout_list where workout_id = 'W006';
108
109
110
111
112
113
114
```
- Output Window:** Action Output table showing the results of the update and delete operations:

#	Time	Action	Message	Duration / Fetch
131	18:28:24	update workout_list set workout_name = 'workout' where workout_id = 'W006'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
132	18:28:24	update workout_list set workout_name = 'workouts' where workout_id = 'W005'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
133	18:28:24	delete FROM workout_list where workout_id = 'W006'	1 row(s) affected	0.000 sec
- System Tray:** Shows weather (Record high), battery level, and system status.

Workout Table

- Definition

The screenshot shows the MySQL Workbench interface with the 'Local instance MySQL80' database selected. In the Navigator pane, under the 'fitness_track' schema, the 'Tables' section is expanded, showing various tables like contact, users, workout, and workolist. A SQL editor tab titled 'SQL File 5*' contains the following SQL code:

```
L15 • CREATE TABLE Workout (
L16     User_id INT NOT NULL,
L17     Workout_id VARCHAR(10) NOT NULL,
L18     Start_time TIME,
L19     Calories_burned INT,
L20     Hours INT,
L21     Minutes INT,
L22     Seconds INT,
L23     PRIMARY KEY (User_id, Workout_id)
L24 );
L25
```

The 'Output' pane shows the results of the previous operations, including the creation of the 'Workout' table and some updates to the 'workout_list' table.

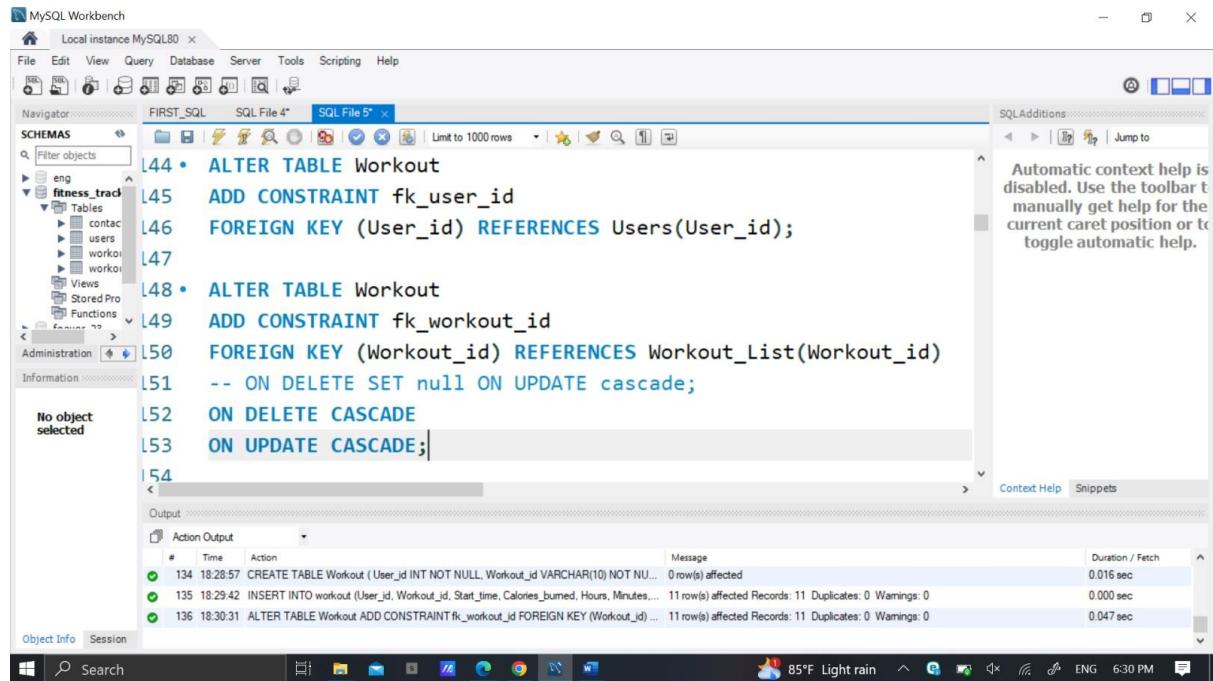
- Data Insertion

The screenshot shows the MySQL Workbench interface with the 'Local instance MySQL80' database selected. In the Navigator pane, under the 'fitness_track' schema, the 'Tables' section is expanded, showing the 'Workout' table. A SQL editor tab titled 'SQL File 5*' contains the following SQL code:

```
L127 • INSERT INTO workout (User_id, Workout_id, Start_time, Calories_burned, Hours, Minutes, Seconds)
L128     VALUES
L129     (001, 'W001', '07:00:00', 600, 1, 10, 0),
L130     (001, 'W002', '08:00:00', 250, 0, 30, 20),
L131     (002, 'W001', '07:30:00', 700, 1, 30, 10),
L132     (003, 'W003', '17:00:00', 300, 0, 30, 15),
L133     (004, 'W004', '16:30:00', 450, 2, 15, 20),
L134     (005, 'W002', '16:30:00', 400, 1, 45, 25),
L135     (005, 'W001', '15:30:00', 400, 1, 40, 30),
L136     (001, 'W003', '07:30:00', 600, 1, 10, 0),
L137     (001, 'W004', '07:50:00', 650, 1, 10, 0),
L138     (001, 'W005', '07:20:00', 300, 1, 10, 0),
L139     (001, 'W007', '06:00:00', 400, 1, 10, 0);
L140
L141
```

The 'Output' pane shows the results of the insertions, including the creation of the 'Workout' table and the successful insertion of 11 rows of data.

- Insertion Foreign key constraint



The screenshot shows the MySQL Workbench interface with the following SQL code in the editor:

```

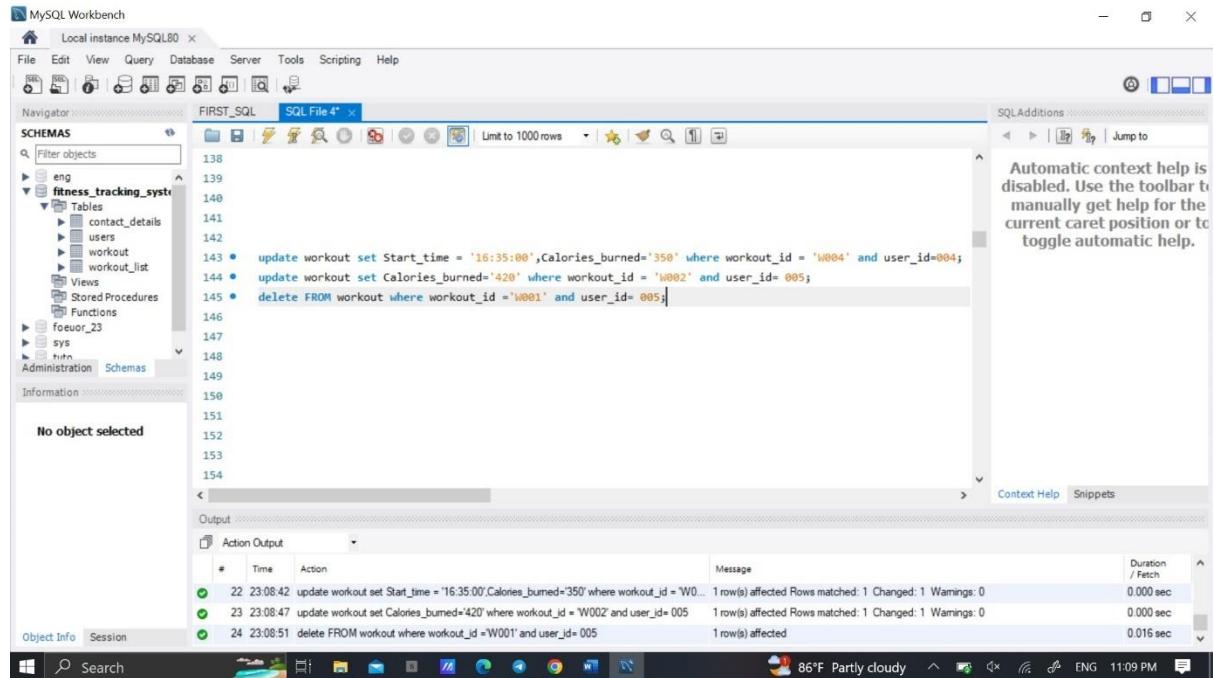
L44 • ALTER TABLE Workout
L45 ADD CONSTRAINT fk_user_id
L46 FOREIGN KEY (User_id) REFERENCES Users(User_id);
L47
L48 • ALTER TABLE Workout
L49 ADD CONSTRAINT fk_workout_id
L50 FOREIGN KEY (Workout_id) REFERENCES Workout_List(Workout_id)
L51 -- ON DELETE SET null ON UPDATE cascade;
L52 ON DELETE CASCADE
L53 ON UPDATE CASCADE;
L54

```

The Output pane shows the results of the execution:

#	Time	Action	Message	Duration / Fetch
134	18:28:57	CREATE TABLE Workout (User_id INT NOT NULL, Workout_id VARCHAR(10) NOT NU...	0 row(s) affected	0.016 sec
135	18:29:42	INSERT INTO workout (User_id, Workout_id, Start_time, Calories_burned, Hours, Minutes, ...)	11 row(s) affected Records: 11 Duplicates: 0 Warnings: 0	0.000 sec
136	18:30:31	ALTER TABLE Workout ADD CONSTRAINT fk_workout_id FOREIGN KEY (Workout_id) ...	11 row(s) affected Records: 11 Duplicates: 0 Warnings: 0	0.047 sec

- Update and Delete Operations



The screenshot shows the MySQL Workbench interface with the following SQL code in the editor:

```

138
139
140
141
142
143 • update workout set Start_time = '16:35:00',Calories_burned='350' where workout_id = 'W004' and user_id=004;
144 • update workout set Calories_burned='420' where workout_id = 'W002' and user_id= 005;
145 • delete FROM workout where workout_id ='W001' and user_id= 005;
146
147
148
149
150
151
152
153
154

```

The Output pane shows the results of the execution:

#	Time	Action	Message	Duration / Fetch
22	23:08:42	update workout set Start_time = '16:35:00',Calories_burned='350' where workout_id = 'W004' and user_id=004;	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
23	23:08:47	update workout set Calories_burned='420' where workout_id = 'W002' and user_id= 005	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
24	23:08:51	delete FROM workout where workout_id ='W001' and user_id= 005	1 row(s) affected	0.016 sec

Nutrition_Log Table

- Definition

The screenshot shows the MySQL Workbench interface with the SQL editor tab active. The code being run is:

```
CREATE TABLE Nutrition_log (
    User_id int not null,
    Log_date_time DATETIME not null,
    Calories_Taken INT,
    PRIMARY KEY (User_id, Log_date_time),
    FOREIGN KEY (User_id) REFERENCES Users(User_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

The output pane shows the results of the command execution:

#	Time	Action	Message	Duration / Fetch
138	18:32:29	update workout set Calories_burned=420' where workout_id = 'W002' and user_id= 005	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
139	18:32:29	delete FROM workout where workout_id = 'W001' and user_id= 005	1 row(s) affected	0.000 sec
140	18:32:47	CREATE TABLE Nutrition_log (User_id int not null, Log_date_time DATETIM...)	0 row(s) affected	0.031 sec

- Data Insertion

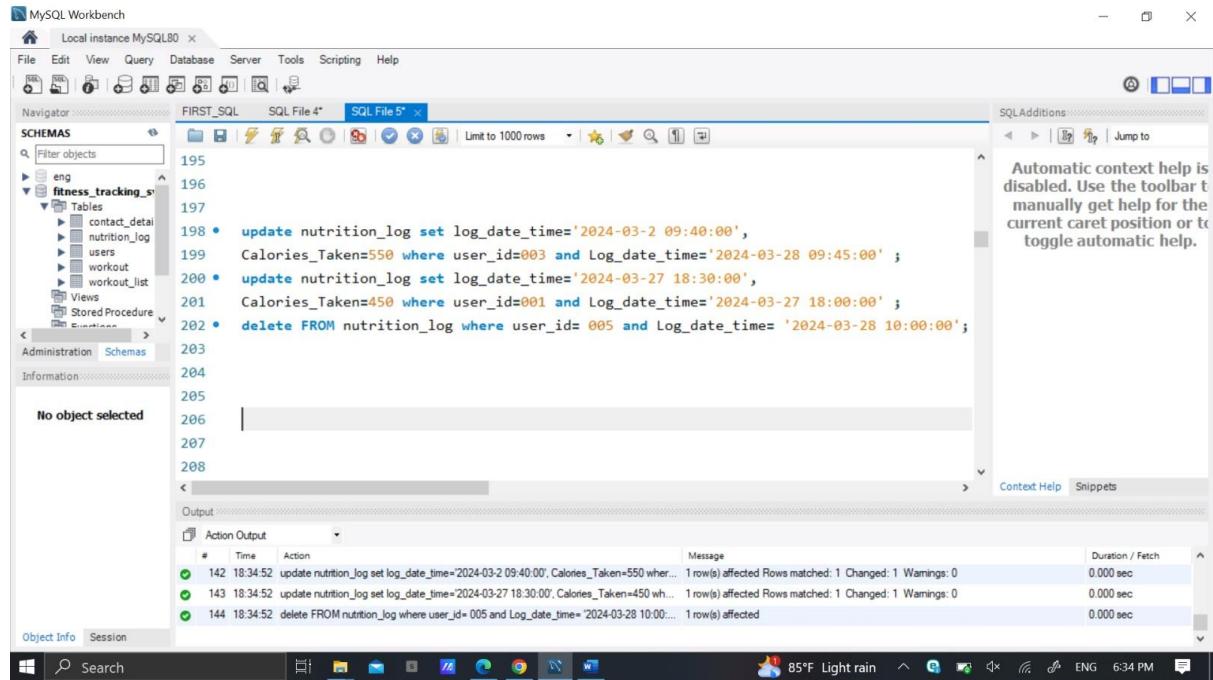
The screenshot shows the MySQL Workbench interface with the SQL editor tab active. The code being run is:

```
INSERT INTO Nutrition_log (User_id, Log_date_time, Calories_Taken)
VALUES
(001, '2024-03-28 08:00:00', 500),
(002, '2024-03-28 12:30:00', 700),
(003, '2024-03-28 09:45:00', 600),
(001, '2024-03-27 18:00:00', 800),
(004, '2024-03-28 07:15:00', 400),
(005, '2024-03-28 10:00:00', 550);
```

The output pane shows the results of the command execution:

#	Time	Action	Message	Duration / Fetch
139	18:32:29	delete FROM workout where workout_id = 'W001' and user_id= 005	1 row(s) affected	0.000 sec
140	18:32:47	CREATE TABLE Nutrition_log (User_id int not null, Log_date_time DATETIM...)	0 row(s) affected	0.031 sec
141	18:33:27	INSERT INTO Nutrition_log (User_id, Log_date_time, Calories_Taken) VALUES (0...)	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.000 sec

- Update and Delete Operations



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (eng, fitness_tracking_s), Tables (contact_detail, nutrition_log, users, workout, workout_list).
- SQL Editor:** SQL File 5* contains the following SQL code:

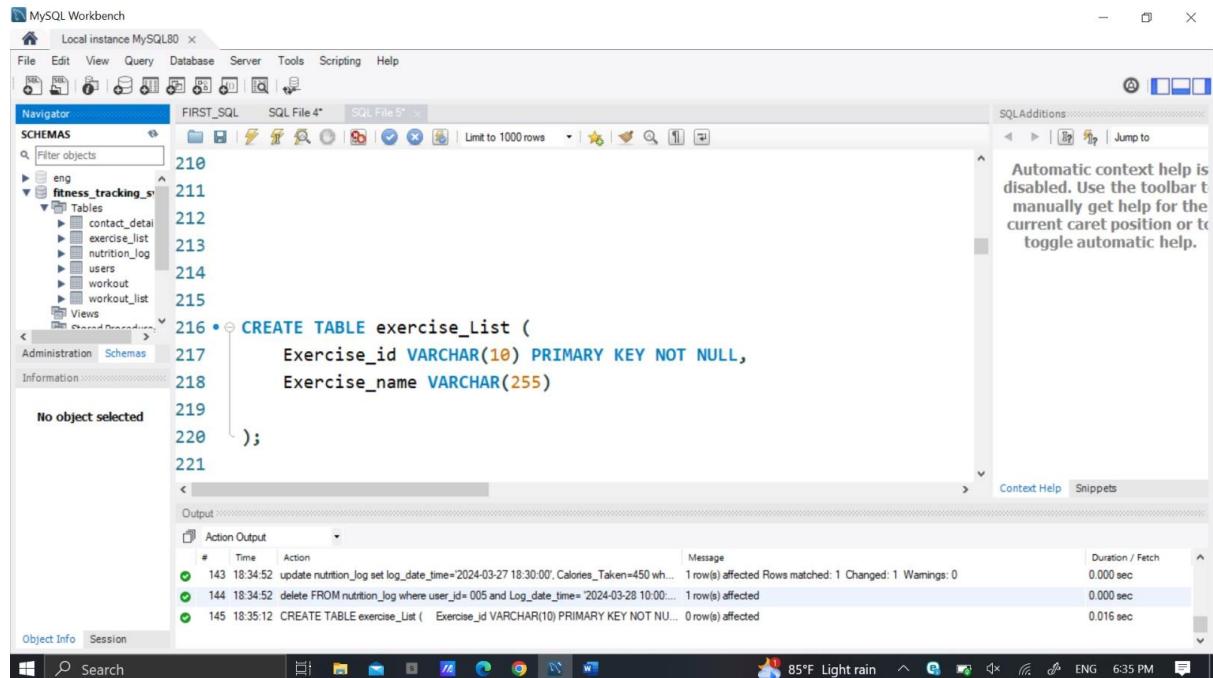

```

195
196
197
198 • update nutrition_log set log_date_time='2024-03-2 09:40:00',
199 Calories_Taken=550 where user_id=003 and Log_date_time='2024-03-28 09:45:00' ;
200 • update nutrition_log set log_date_time='2024-03-27 18:30:00',
201 Calories_Taken=450 where user_id=001 and Log_date_time='2024-03-27 18:00:00' ;
202 • delete FROM nutrition_log where user_id= 005 and Log_date_time= '2024-03-28 10:00:00';
203
204
205
206
207
208
      
```
- Output Window:** Action Output shows the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
142	18:34:52	update nutrition_log set log_date_time='2024-03-2 09:40:00', Calories_Taken=550 wher...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
143	18:34:52	update nutrition_log set log_date_time='2024-03-27 18:30:00', Calories_Taken=450 wh...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
144	18:34:52	delete FROM nutrition_log where user_id= 005 and Log_date_time= '2024-03-28 10:00:00'...	1 row(s) affected	0.000 sec

Exercise_List Table

- Definition



The screenshot shows the MySQL Workbench interface with the following details:

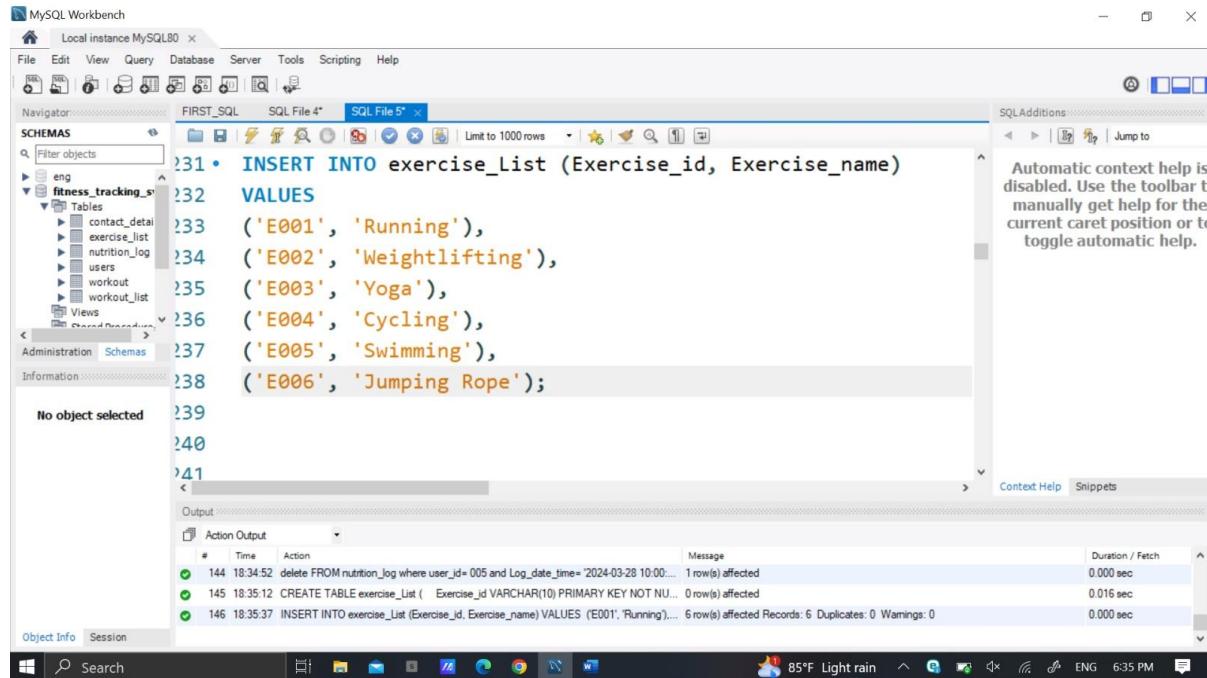
- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (eng, fitness_tracking_s), Tables (contact_detail, exercise_list, nutrition_log, users, workout, workout_list).
- SQL Editor:** SQL File 4* contains the following SQL code:


```

210
211
212
213
214
215
216 • CREATE TABLE exercise_list (
217     Exercise_id VARCHAR(10) PRIMARY KEY NOT NULL,
218     Exercise_name VARCHAR(255)
219
220 );
221
      
```
- Output Window:** Action Output shows the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
143	18:34:52	update nutrition_log set log_date_time='2024-03-27 18:30:00', Calories_Taken=450 wh...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
144	18:34:52	delete FROM nutrition_log where user_id= 005 and Log_date_time= '2024-03-28 10:00:00'...	1 row(s) affected	0.000 sec
145	18:35:12	CREATE TABLE exercise_list (Exercise_id VARCHAR(10) PRIMARY KEY NOT NU...	0 row(s) affected	0.016 sec

- Data Insertion



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the schema `fitness_tracking_s` with tables: `contact_detail`, `exercise_list`, `nutrition_log`, `users`, `workout`, and `workout_list`.
- SQL Editor:** Contains the following SQL code:

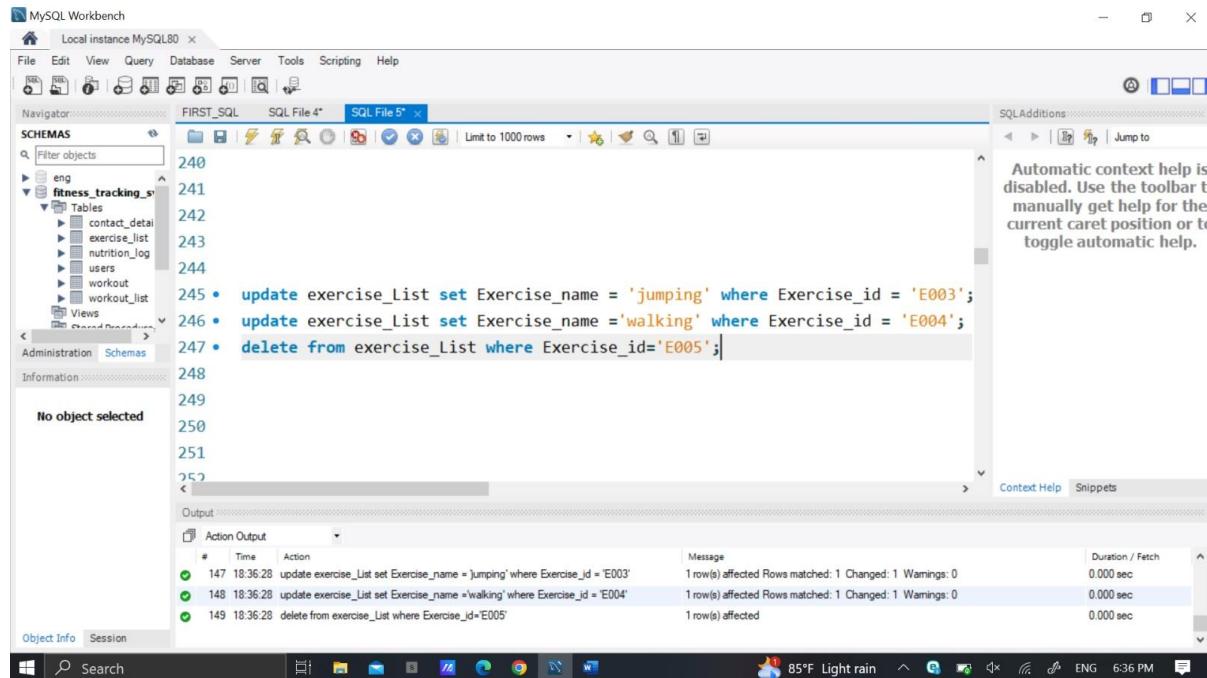
```

231 • INSERT INTO exercise_List (Exercise_id, Exercise_name)
232   VALUES
233     ('E001', 'Running'),
234     ('E002', 'Weightlifting'),
235     ('E003', 'Yoga'),
236     ('E004', 'Cycling'),
237     ('E005', 'Swimming'),
238     ('E006', 'Jumping Rope');
239
240
241

```
- Output Window:** Shows the execution results:

#	Time	Action	Message	Duration / Fetch
144	18:34:52	delete FROM nutrition_log where user_Id=005 and Log_date_time='2024-03-28 10:00:...'	1 row(s) affected	0.000 sec
145	18:35:12	CREATE TABLE exercise_List (Exercise_id VARCHAR(10) PRIMARY KEY NOT NU...)	0 row(s) affected	0.016 sec
146	18:35:37	INSERT INTO exercise_List (Exercise_id, Exercise_name) VALUES ('E001', 'Running')	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.000 sec
- System Tray:** Shows the date and time as 85°F Light rain, 6:35 PM.

- Update and Delete Operations



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the schema `fitness_tracking_s` with tables: `contact_detail`, `exercise_list`, `nutrition_log`, `users`, `workout`, and `workout_list`.
- SQL Editor:** Contains the following SQL code:

```

240
241
242
243
244
245 • update exercise_List set Exercise_name = 'jumping' where Exercise_id = 'E003';
246 • update exercise_List set Exercise_name = 'walking' where Exercise_id = 'E004';
247 • delete from exercise_List where Exercise_id='E005';
248
249
250
251

```
- Output Window:** Shows the execution results:

#	Time	Action	Message	Duration / Fetch
147	18:36:28	update exercise_List set Exercise_name = 'jumping' where Exercise_id = 'E003'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
148	18:36:28	update exercise_List set Exercise_name = 'walking' where Exercise_id = 'E004'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
149	18:36:28	delete from exercise_List where Exercise_id='E005'	1 row(s) affected	0.000 sec
- System Tray:** Shows the date and time as 85°F Light rain, 6:36 PM.

Exercise Table

- Definition

The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab selected. In the SQL pane, the following SQL code is displayed:

```
CREATE TABLE Exercise (
    Workout_id VARCHAR(10) not null,
    Exercise_id VARCHAR(10) not null,
    Category VARCHAR(50),
    Calories_burned_per_minute FLOAT,
    PRIMARY KEY (Workout_id, Exercise_id),
    FOREIGN KEY (Workout_id) REFERENCES Workout(Workout_id),
    FOREIGN KEY (Exercise_id) REFERENCES Exercise_list(Exercise_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE );
```

The 'Output' pane shows the results of the execution:

#	Time	Action	Message	Duration / Fetch
148	18:36:28	update exercise_list set Exercise_name = 'walking' where Exercise_id = 'E004'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
149	18:36:28	delete from exercise_list where Exercise_id='E005'	1 row(s) affected	0.000 sec
150	18:37:23	CREATE TABLE Exercise (Workout_id VARCHAR(10) not null, Exercise_id VARC...)	0 row(s) affected	0.031 sec

- Data Insertion

The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab selected. In the SQL pane, the following SQL code is displayed:

```
INSERT INTO Exercise (Workout_id, Exercise_id, Category, Calories_burned_per_minute)
VALUES
('W001', 'E001', 'Cardio', 10.5),
('W001', 'E002', 'Strength Training', 5.75),
('W002', 'E003', 'Flexibility', 6.25),
('W003', 'E004', 'Cardio', 8.0),
('W004', 'E004', 'Cardio', 12.3),
('W003', 'E001', 'Cardio', 12.3);
```

The 'Output' pane shows the results of the execution:

#	Time	Action	Message	Duration / Fetch
149	18:36:28	delete from exercise_list where Exercise_id='E005'	1 row(s) affected	0.000 sec
150	18:37:23	CREATE TABLE Exercise (Workout_id VARCHAR(10) not null, Exercise_id VARC...)	0 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.031 sec
151	18:37:58	INSERT INTO Exercise (Workout_id, Exercise_id, Category, Calories_burned_per_minu...)	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.000 sec

- Update and Delete Operations

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Local instance MySQL80, Schemas (eng, fitness_trac), Tables (contact, exercise, nutritic, users, worko, worko), Views.
- SQL Editor:** FIRST_SQL, SQL File 4*, SQL File 5*. The code in SQL File 5* is as follows:

```

282
283
284
285
286
287
288 • update exercise set Exercise_id= 'E006' where Exercise_id = 'E002' and workout_id='W001';
289 • update exercise set Calories_burned_per_minute=8.00 where Exercise_id = 'E003' and workout_id = 'W002';
290 • delete from exercise where Exercise_id='E005' and workout_id='W004';
291
292
293
294
295
296

```

- Output Window:** Action Output table showing the results of the executed queries.

#	Time	Action	Message	Duration / Fetch
152	18:38:48	update exercise set Exercise_id= 'E006' where Exercise_id = 'E002' and workout_id='W001';	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
153	18:38:48	update exercise set Calories_burned_per_minute=8.00 where Exercise_id = 'E003' and wor... kout_id = 'W002';	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
154	18:38:48	delete from exercise where Exercise_id='E005' and workout_id='W004';	0 row(s) affected	0.015 sec

Exercise_Log Table

- Definition

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Local instance MySQL80, Schemas (eng, fitness_trac), Tables (contact, exercise, nutritic, users, worko, worko), Views.
- SQL Editor:** FIRST_SQL, SQL File 4*, SQL File 5*. The code in SQL File 5* is as follows:

```

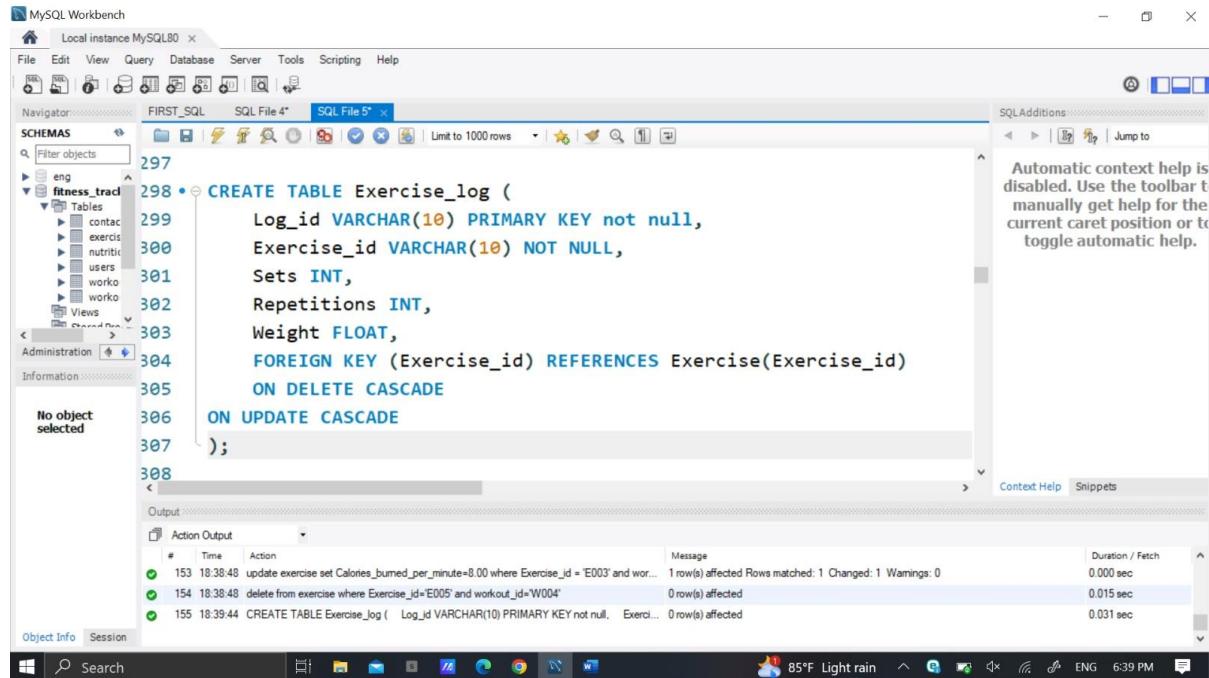
297
298 • CREATE TABLE Exercise_log (
299     Log_id VARCHAR(10) PRIMARY KEY not null,
300     Exercise_id VARCHAR(10) NOT NULL,
301     Sets INT,
302     Repetitions INT,
303     Weight FLOAT,
304     FOREIGN KEY (Exercise_id) REFERENCES Exercise(Exercise_id)
305     ON DELETE CASCADE
306     ON UPDATE CASCADE
307 );
308

```

- Output Window:** Action Output table showing the results of the executed queries.

#	Time	Action	Message	Duration / Fetch
153	18:38:48	update exercise set Calories_burned_per_minute=8.00 where Exercise_id = 'E003' and wor... kout_id = 'W004';	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
154	18:38:48	delete from exercise where Exercise_id='E005' and workout_id='W004';	0 row(s) affected	0.015 sec
155	18:39:44	CREATE TABLE Exercise_log (Log_id VARCHAR(10) PRIMARY KEY not null, Exerci...);	0 row(s) affected	0.031 sec

- Data Insertion



The screenshot shows the MySQL Workbench interface with the following details:

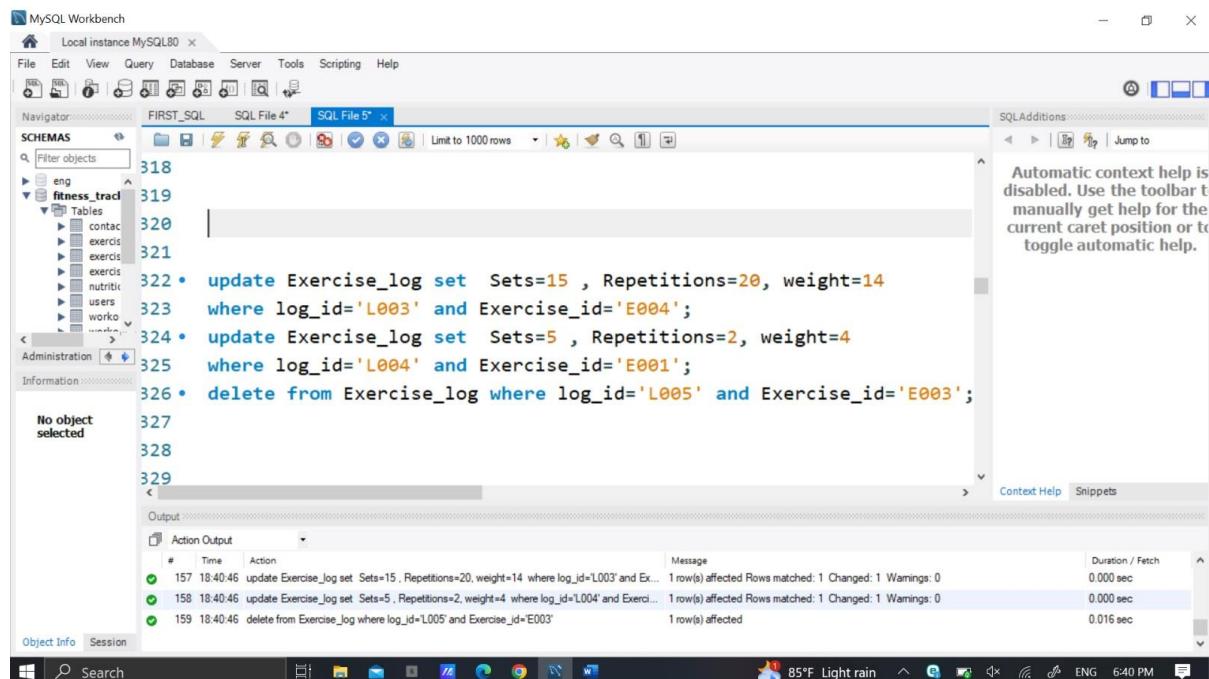
- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Local instance MySQL80, Schemas (eng, fitness_trac), Tables (contact, exercise, exercise_id, nutritic, users, worko, worko).
- SQL Editor:** SQL File 5*, containing the following SQL code:


```

297
298 • CREATE TABLE Exercise_log (
299     Log_id VARCHAR(10) PRIMARY KEY not null,
300     Exercise_id VARCHAR(10) NOT NULL,
301     Sets INT,
302     Repetitions INT,
303     Weight FLOAT,
304     FOREIGN KEY (Exercise_id) REFERENCES Exercise(Exercise_id)
305     ON DELETE CASCADE
306     ON UPDATE CASCADE
307 );
308
      
```
- Output Window:** Action Output table showing three log entries:

#	Time	Action	Message	Duration / Fetch
153	18:38:48	update exercise set Calories_burned_per_minute=8.00 where Exercise_id = 'E003' and wor...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
154	18:38:48	delete from exercise where Exercise_id='E005' and workout_id='W004'	0 row(s) affected	0.015 sec
155	18:39:44	CREATE TABLE Exercise_log (Log_id VARCHAR(10) PRIMARY KEY not null, Exerci...	0 row(s) affected	0.031 sec
- System Tray:** Shows weather (85°F Light rain), battery level, and system status.

- Update and Delete Operations



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Local instance MySQL80, Schemas (eng, fitness_trac), Tables (contact, exercise, exercise_id, nutritic, users, worko, worko).
- SQL Editor:** SQL File 5*, containing the following SQL code:


```

318
319
320
321
322 • update Exercise_log set Sets=15 , Repetitions=20, weight=14
323   where log_id='L003' and Exercise_id='E004';
324 • update Exercise_log set Sets=5 , Repetitions=2, weight=4
325   where log_id='L004' and Exercise_id='E001';
326 • delete from Exercise_log where log_id='L005' and Exercise_id='E003';
327
328
329
      
```
- Output Window:** Action Output table showing three log entries:

#	Time	Action	Message	Duration / Fetch
157	18:40:46	update Exercise_log set Sets=15 , Repetitions=20, weight=14 where log_id='L003' and Ex...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
158	18:40:46	update Exercise_log set Sets=5 , Repetitions=2, weight=4 where log_id='L004' and Exerci...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
159	18:40:46	delete from Exercise_log where log_id='L005' and Exercise_id='E003'	1 row(s) affected	0.016 sec
- System Tray:** Shows weather (85°F Light rain), battery level, and system status.

Heart_Rate_Record

- Definition

The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab selected. In the SQL pane, the following SQL code is displayed:

```
CREATE TABLE Heart_Rate_Record (
    Log_id VARCHAR(10) NOT NULL,
    Heart_Rate INT,
    Recorded_time TIME,
    PRIMARY KEY(Log_id , Recorded_time),
    FOREIGN KEY (Log_id) REFERENCES Exercise_log(Log_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

The Navigator pane shows the database schema with the 'fitness_trad' database selected, containing tables like contact, exercise, heart, and users. The Output pane displays the results of the previous operations, including the creation of the table.

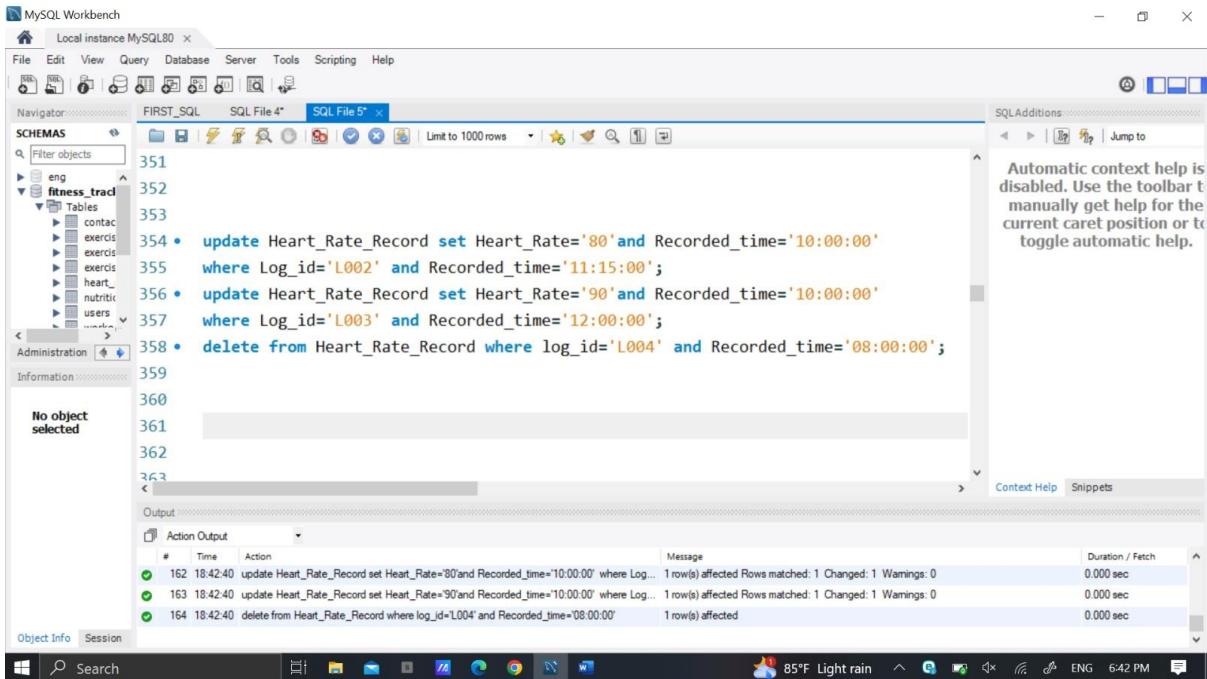
- Data Insertion

The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab selected. In the SQL pane, the following SQL code is displayed:

```
INSERT INTO Heart_Rate_Record (Log_id, Heart_Rate, Recorded_time)
VALUES
    ('L001', 100, '12:00:00'),
    ('L001', 120, '11:00:00'),
    ('L002', 100, '11:15:00'),
    ('L003', 120, '12:00:00'),
    ('L004', 125, '08:00:00');
```

The Navigator pane shows the database schema with the 'fitness_trad' database selected, containing tables like contact, exercise, heart, and users. The Output pane displays the results of the previous operations, including the insertion of data into the table.

- Update and Delete Operations



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** fitness_trad
- Tables:** contact, exercis, exercis, exercis, heart_, nutritic, users
- SQL Editor:** SQL File 5*
- Code:**

```

351
352
353
354 • update Heart_Rate_Record set Heart_Rate='80' and Recorded_time='10:00:00'
      where Log_id='L002' and Recorded_time='11:15:00';
355 • update Heart_Rate_Record set Heart_Rate='90' and Recorded_time='10:00:00'
      where Log_id='L003' and Recorded_time='12:00:00';
356 • delete from Heart_Rate_Record where log_id='L004' and Recorded_time='08:00:00';
357
358
359
360
361
362
363

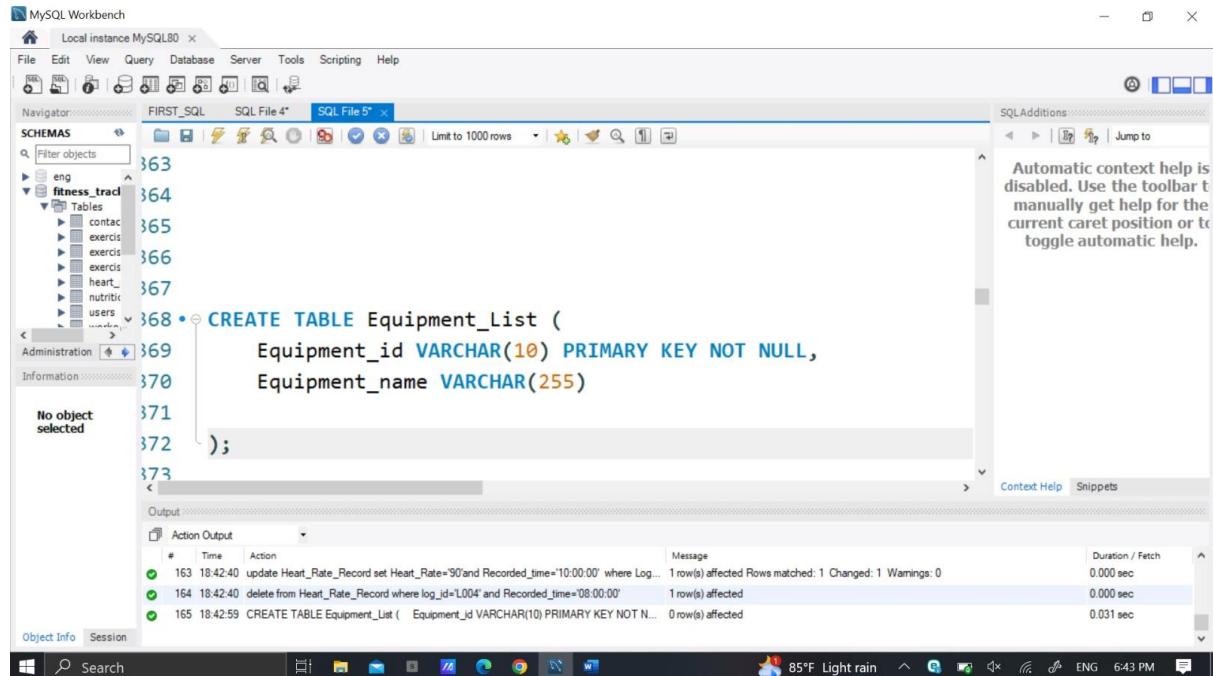
```

- Output:** Action Output table showing results of the executed statements.

#	Time	Action	Message	Duration / Fetch
162	18:42:40	update Heart_Rate_Record set Heart_Rate='80' and Recorded_time='10:00:00' where Log_id='L002' and Recorded_time='11:15:00';	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
163	18:42:40	update Heart_Rate_Record set Heart_Rate='90' and Recorded_time='10:00:00' where Log_id='L003' and Recorded_time='12:00:00';	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
164	18:42:40	delete from Heart_Rate_Record where log_id='L004' and Recorded_time='08:00:00';	1 row(s) affected	0.000 sec

Equipment_List Table

- Definition



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** fitness_trad
- Tables:** contact, exercis, exercis, exercis, heart_, nutritic, users
- SQL Editor:** SQL File 5*
- Code:**

```

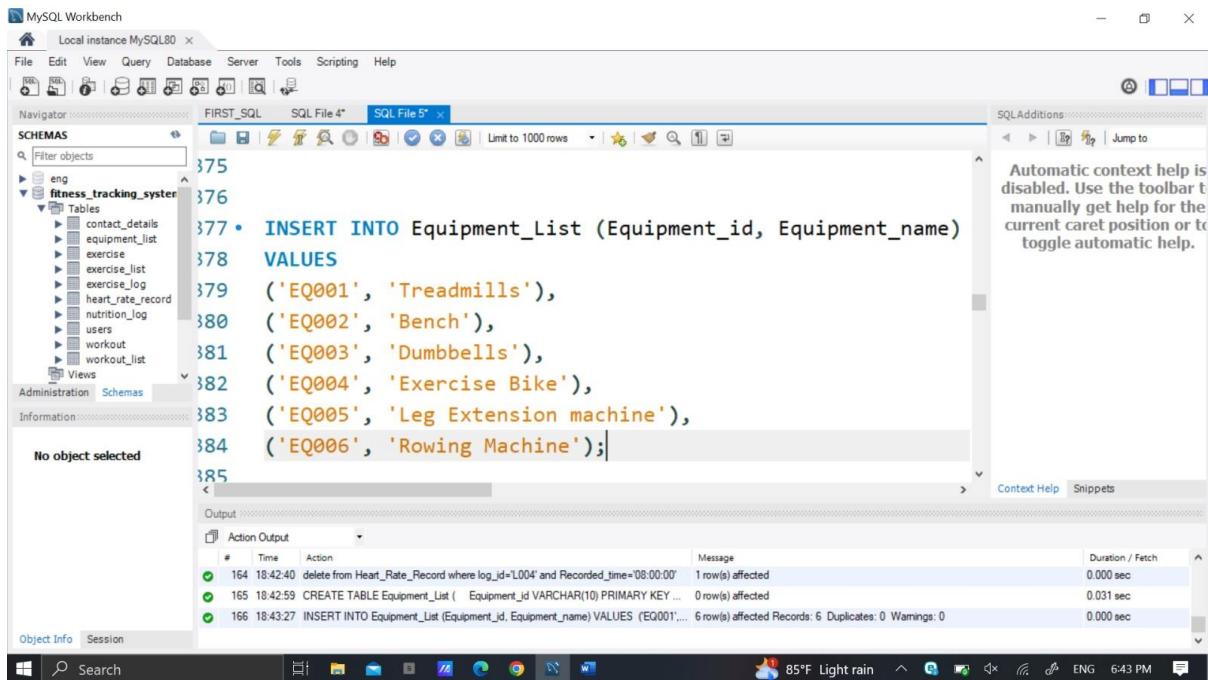
363
364
365
366
367
368 • CREATE TABLE Equipment_List (
      Equipment_id VARCHAR(10) PRIMARY KEY NOT NULL,
      Equipment_name VARCHAR(255)
    );
369
370
371
372
373

```

- Output:** Action Output table showing results of the executed statements.

#	Time	Action	Message	Duration / Fetch
163	18:42:40	update Heart_Rate_Record set Heart_Rate='90' and Recorded_time='10:00:00' where Log_id='L003' and Recorded_time='12:00:00';	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
164	18:42:40	delete from Heart_Rate_Record where log_id='L004' and Recorded_time='08:00:00';	1 row(s) affected	0.000 sec
165	18:42:59	CREATE TABLE Equipment_List (Equipment_id VARCHAR(10) PRIMARY KEY NOT NULL, Equipment_name VARCHAR(255));	0 row(s) affected	0.031 sec

- Data Insertion



The screenshot shows the MySQL Workbench interface with the following details:

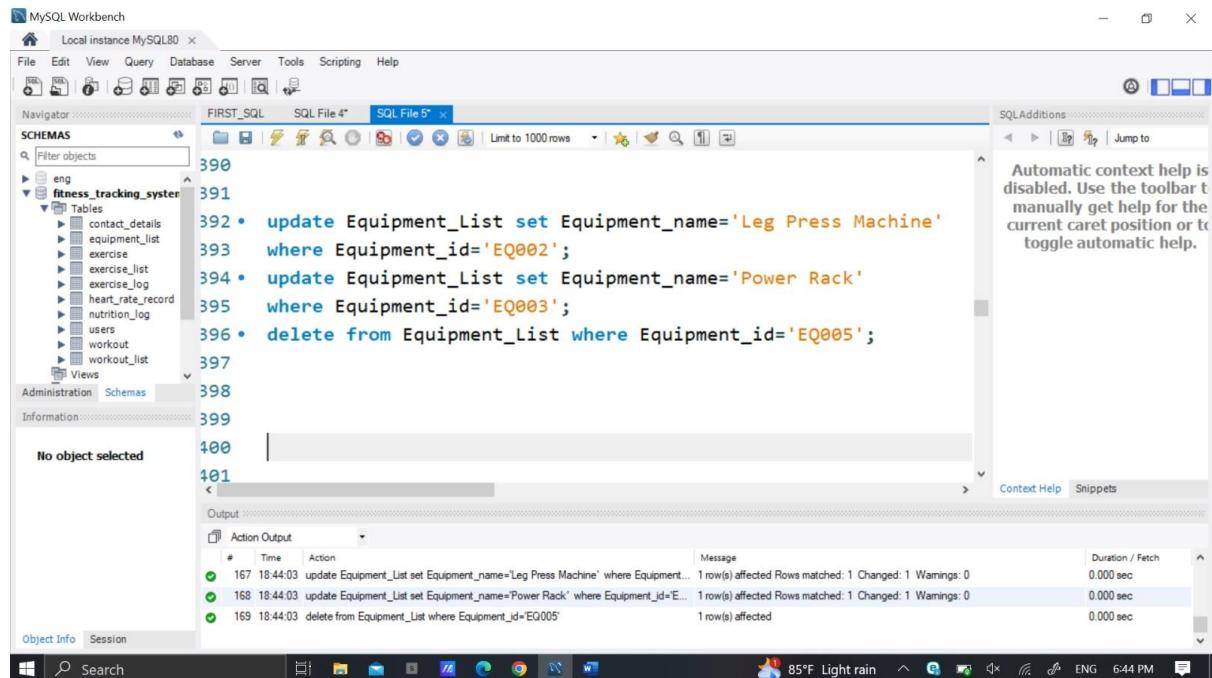
- Schemas:** Local instance MySQL80, selected schema: fitness_tracking_system
- Tables:** equipment_list
- SQL Editor:** SQL File 5* containing the following SQL code:


```

375
376
377 • INSERT INTO Equipment_List (Equipment_id, Equipment_name)
378   VALUES
379   ('EQ001', 'Treadmills'),
380   ('EQ002', 'Bench'),
381   ('EQ003', 'Dumbbells'),
382   ('EQ004', 'Exercise Bike'),
383   ('EQ005', 'Leg Extension machine'),
384   ('EQ006', 'Rowing Machine');
      
```
- Output Window:** Action Output table showing the results of the insertions:

#	Time	Action	Message	Duration / Fetch
164	18:42:40	delete from Heart_Rate_Record where log_id=L004 and Recorded_time='08:00:00'	1 row(s) affected	0.000 sec
165	18:42:59	CREATE TABLE Equipment_List (Equipment_id VARCHAR(10) PRIMARY KEY ...	0 row(s) affected	0.031 sec
166	18:43:27	INSERT INTO Equipment_List (Equipment_id, Equipment_name) VALUES (EQ001,...)	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.000 sec

- Update and Delete Operations



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** Local instance MySQL80, selected schema: fitness_tracking_system
- Tables:** equipment_list
- SQL Editor:** SQL File 5* containing the following SQL code:


```

390
391
392 • update Equipment_List set Equipment_name='Leg Press Machine'
393   where Equipment_id='EQ002';
394 • update Equipment_List set Equipment_name='Power Rack'
395   where Equipment_id='EQ003';
396 • delete from Equipment_List where Equipment_id='EQ005';
      
```
- Output Window:** Action Output table showing the results of the updates and deletion:

#	Time	Action	Message	Duration / Fetch
167	18:44:03	update Equipment_List set Equipment_name='Leg Press Machine' where Equipment_id='EQ002';	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
168	18:44:03	update Equipment_List set Equipment_name='Power Rack' where Equipment_id='EQ003';	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
169	18:44:03	delete from Equipment_List where Equipment_id='EQ005'	1 row(s) affected	0.000 sec

Equipment Table

- Definition

The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab active. In the SQL pane, the following SQL code is displayed:

```
402
403 • CREATE TABLE Equipment (
404     Equipment_id VARCHAR(10) NOT NULL,
405     Exercise_id VARCHAR(10) NOT NULL,
406     Duration INT,
407     PRIMARY KEY (Equipment_id , Exercise_id),
408     FOREIGN KEY (Exercise_id) REFERENCES Exercise(Exercise_id),
409     FOREIGN KEY (Equipment_id) REFERENCES Equipment_list(Equipment_id)
410     ON DELETE CASCADE
411     ON UPDATE CASCADE
412 );
413
```

The Navigator pane shows the 'fitness_tracking_system' schema with various tables like contact_details, equipment, etc. The Output pane at the bottom shows the results of the previous operations, including the creation of the Equipment table.

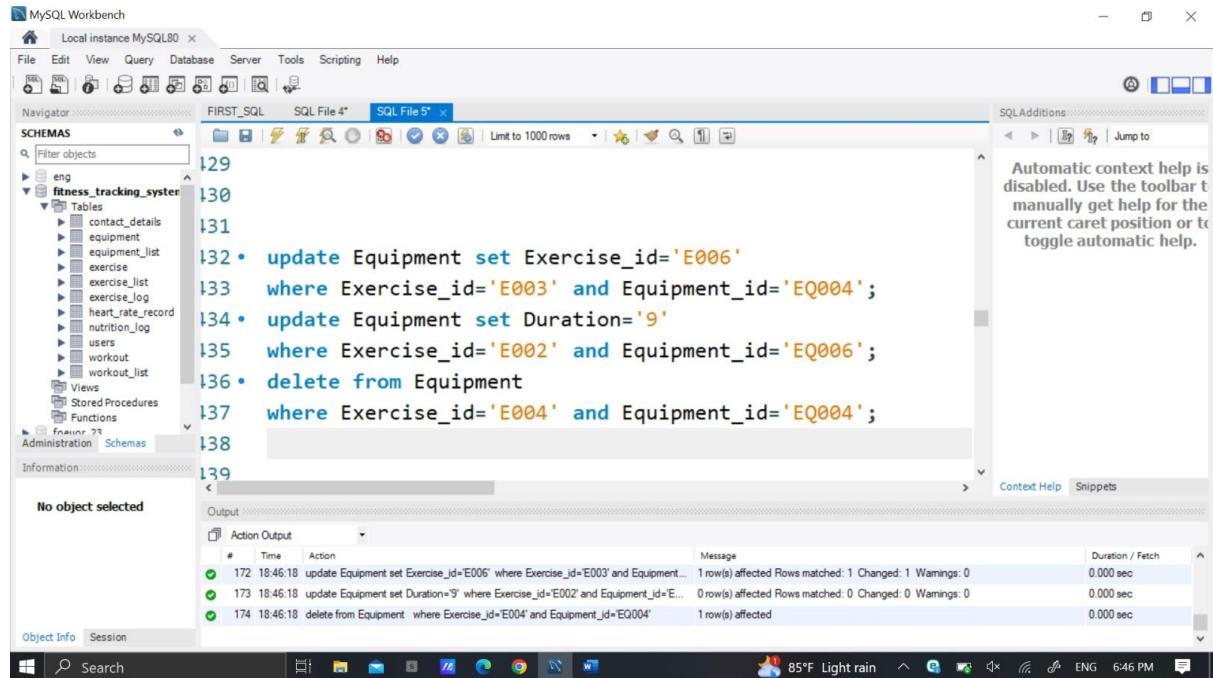
- Data Insertion

The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab active. In the SQL pane, the following SQL code is displayed:

```
417
418
419 • INSERT INTO Equipment (Equipment_id, Exercise_id, Duration)
420     VALUES
421     ('EQ001', 'E001', 3),
422     ('EQ002', 'E001', 10),
423     ('EQ002', 'E006', 2),
424     ('EQ003', 'E003', 5),
425     ('EQ004', 'E003', 7),
426     ('EQ004', 'E004', 5);
427
428
```

The Navigator pane shows the 'fitness_tracking_system' schema. The Output pane at the bottom shows the results of the previous operations, including the insertion of data into the Equipment table.

- Update and Delete Operations



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (eng, fitness_tracking_system), Tables (contact_details, equipment, equipment_list, exercise, exercise_list, exercise_log, heart_rate_record, location, nutrition_log, users, workout, workout_list).
- SQL Editor:** FIRST_SQL tab, SQL File 4*, SQL File 5*. The code executed is:

```

129
130
131
132 • update Equipment set Exercise_id='E006'
133 where Exercise_id='E003' and Equipment_id='EQ004';
134 • update Equipment set Duration='9'
135 where Exercise_id='E002' and Equipment_id='EQ006';
136 • delete from Equipment
137 where Exercise_id='E004' and Equipment_id='EQ004';
138
139

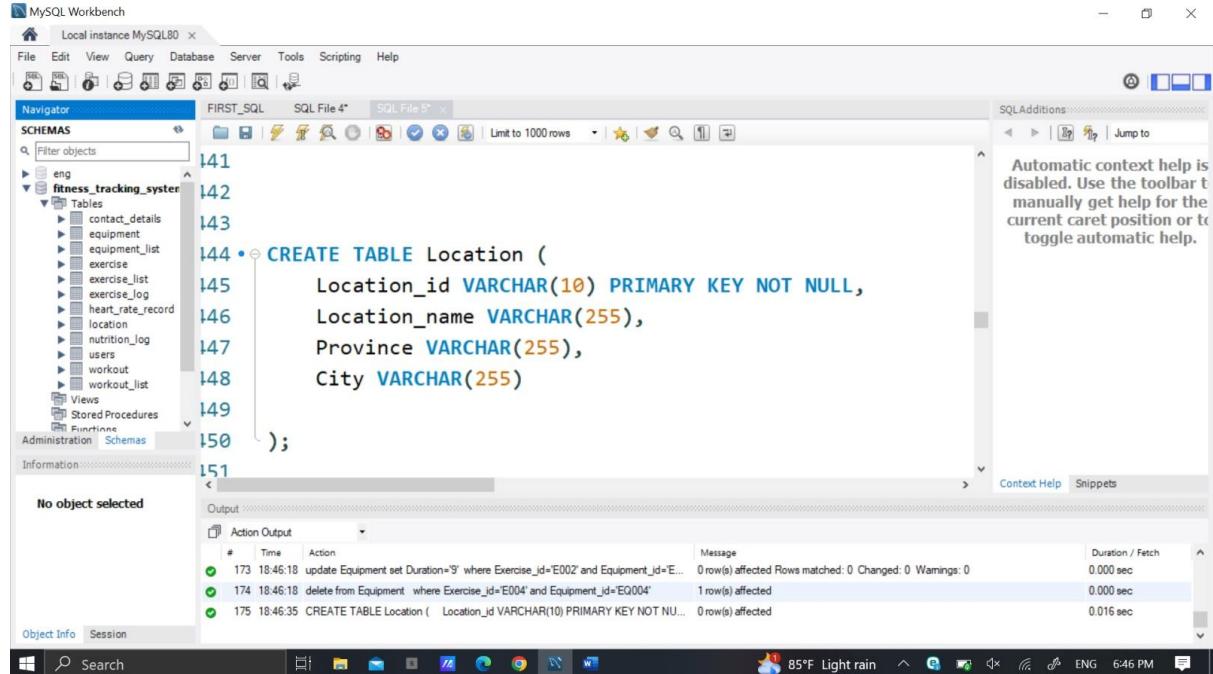
```

- Output Window:** Action Output table showing the results of the executed statements.

#	Time	Action	Message	Duration / Fetch
172	18:46:18	update Equipment set Exercise_id='E006' where Exercise_id='E003' and Equipment_id='EQ004';	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
173	18:46:18	update Equipment set Duration='9' where Exercise_id='E002' and Equipment_id='EQ006';	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
174	18:46:18	delete from Equipment where Exercise_id='E004' and Equipment_id='EQ004';	1 row(s) affected	0.000 sec

Location Table

- Definition



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (eng, fitness_tracking_system), Tables (contact_details, equipment, equipment_list, exercise, exercise_list, exercise_log, heart_rate_record, location, nutrition_log, users, workout, workout_list).
- SQL Editor:** FIRST_SQL tab, SQL File 4*, SQL File 5*. The code executed is:

```

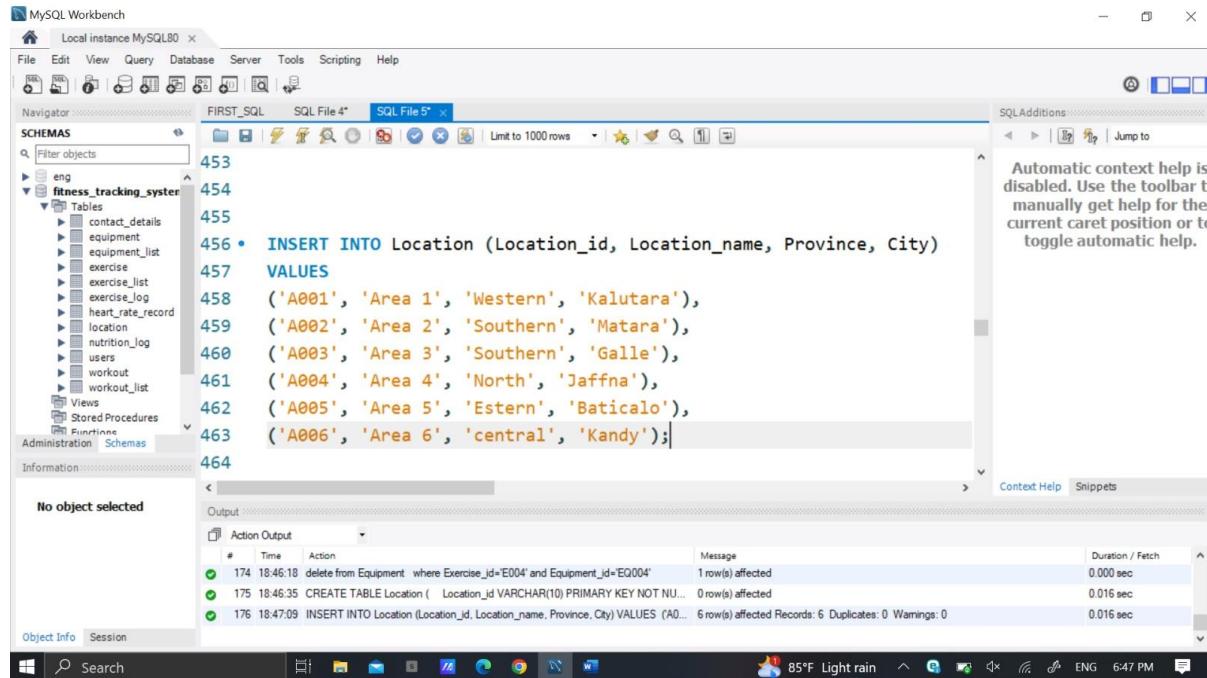
141
142
143
144 • CREATE TABLE Location (
145     Location_id VARCHAR(10) PRIMARY KEY NOT NULL,
146     Location_name VARCHAR(255),
147     Province VARCHAR(255),
148     City VARCHAR(255)
149
150 );
151

```

- Output Window:** Action Output table showing the results of the executed statements.

#	Time	Action	Message	Duration / Fetch
173	18:46:18	update Equipment set Duration='9' where Exercise_id='E002' and Equipment_id='EQ004';	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
174	18:46:18	delete from Equipment where Exercise_id='E004' and Equipment_id='EQ004';	1 row(s) affected	0.000 sec
175	18:46:35	CREATE TABLE Location (Location_id VARCHAR(10) PRIMARY KEY NOT NU...	0 row(s) affected	0.016 sec

- Data Insertion



The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following SQL code:

```

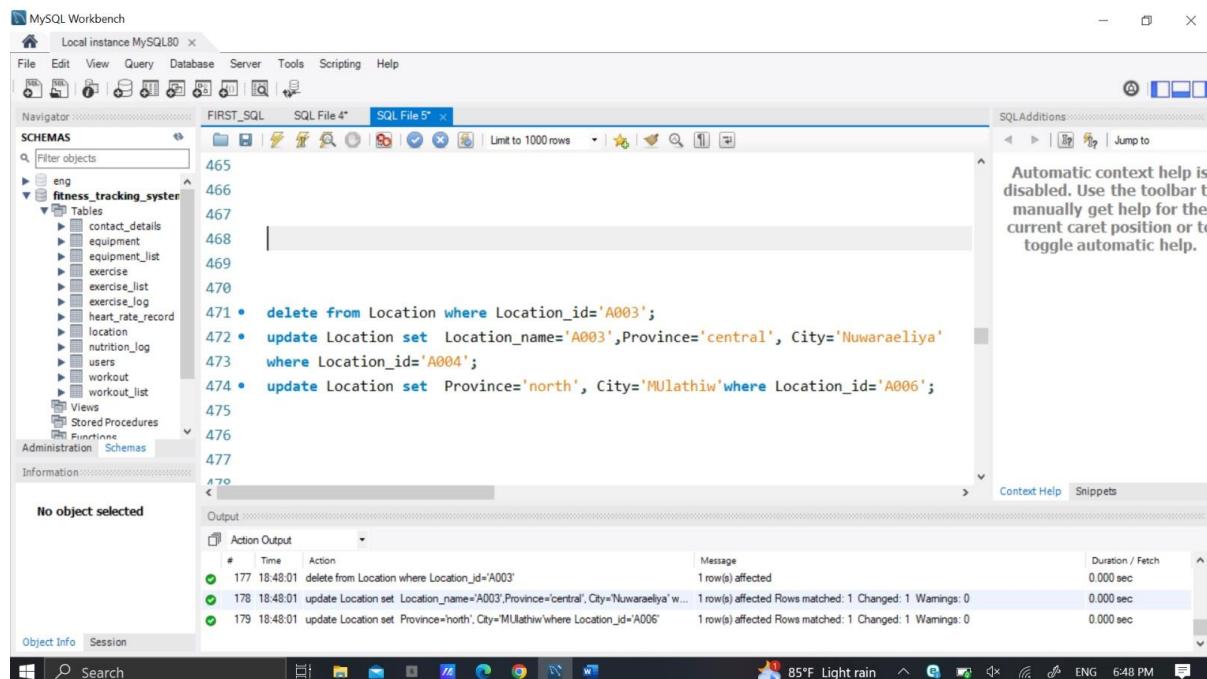
453
454
455
456 • INSERT INTO Location (Location_id, Location_name, Province, City)
457     VALUES
458     ('A001', 'Area 1', 'Western', 'Kalutara'),
459     ('A002', 'Area 2', 'Southern', 'Matara'),
460     ('A003', 'Area 3', 'Southern', 'Galle'),
461     ('A004', 'Area 4', 'North', 'Jaffna'),
462     ('A005', 'Area 5', 'Estern', 'Baticalo'),
463     ('A006', 'Area 6', 'central', 'Kandy');
464

```

The Output pane shows the results of the executed statements:

#	Time	Action	Message	Duration / Fetch
174	18:46:18	delete from Equipment where Exercise_id='E004' and Equipment_id='EQ004'	1 row(s) affected	0.000 sec
175	18:46:35	CREATE TABLE Location (Location_id VARCHAR(10) PRIMARY KEY NOT NU...	0 row(s) affected	0.016 sec
176	18:47:09	INSERT INTO Location (Location_id, Location_name, Province, City) VALUES (A0... 6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0		0.016 sec

- Update and Delete Operations



The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following SQL code:

```

465
466
467
468 |
469
470
471 • delete from Location where Location_id='A003';
472 • update Location set Location_name='A003',Province='central',City='Nuwaraeliya'
473 where Location_id='A004';
474 • update Location set Province='north',City='MULathiw'where Location_id='A006';
475
476
477

```

The Output pane shows the results of the executed statements:

#	Time	Action	Message	Duration / Fetch
177	18:48:01	delete from Location where Location_id='A003'	1 row(s) affected	0.000 sec
178	18:48:01	update Location set Location_name='A003',Province='central',City='Nuwaraeliya' w...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
179	18:48:01	update Location set Province='north',City='MULathiw'where Location_id='A006'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec

Location_Content Table

- Definition

The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab active. In the central SQL editor window, the following SQL code is displayed:

```
480 • CREATE TABLE Location_content (
    Equipment_id VARCHAR(10) NOT NULL,
    Location_id VARCHAR(10) NOT NULL,
    Quantity INT,
    PRIMARY KEY (Equipment_id ,Location_id),
    FOREIGN KEY (Equipment_id) REFERENCES Equipment(Equipment_id),
    FOREIGN KEY (Location_id) REFERENCES Location(Location_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
490
491
```

The 'Navigator' pane on the left shows the database schema, including the 'fitness_tracking_system' database and its tables: contact_details, equipment, equipment_list, exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, and workout_list.

The 'Output' pane at the bottom displays the results of the executed SQL statements, including the creation of the table and some update operations on the Location table.

- Data Insertion

The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab active. In the central SQL editor window, the following SQL code is displayed:

```
495
496
497
498 • INSERT INTO Location_content (Equipment_id, Location_id, Quantity)
499     VALUES
500     ('EQ001', 'A001', 1),
501     ('EQ001', 'A002', 2),
502     ('EQ002', 'A001', 1),
503     ('EQ003', 'A002', 3),
504     ('EQ004', 'A006', 4),
505     ('EQ004', 'A005', 1);
506
```

The 'Navigator' pane on the left shows the database schema, including the 'fitness_tracking_system' database and its tables: contact_details, equipment, equipment_list, exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, and workout_list.

The 'Output' pane at the bottom displays the results of the executed SQL statements, including the creation of the table and the insertion of data into it.

- Update and Delete Operations

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for Home, New, Open, Save, Print, Copy, Paste, Find, Replace, Refresh, and others.
- Navigator:** Shows the schema `fitness_tracking_system` with tables: contact_details, equipment, equipment_list, exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, and workout_list.
- SQL Editor:** Contains five tabs: FIRST_SQL, SQL File 4*, and SQL File 5*. The SQL File 5* tab contains the following code:

```
507
508
509
510 • update Location_content set Quantity =5
511 where Equipment_id='EQ002' and Location_id='A001';
512 • update Location_content set Quantity =7
513 where Equipment_id='EQ003' and Location_id='A002';
514 • delete from Location_content
515 where Equipment_id='EQ004' and Location_id='A005';
516
```
- Output Window:** Shows the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
182	18:49:20	update Location_content set Quantity =5 where Equipment_id='EQ002' and Locatio...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
183	18:49:20	update Location_content set Quantity =7 where Equipment_id='EQ003' and Locatio...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
184	18:49:20	delete from Location_content where Equipment_id='EQ004' and Location_id='A005'	1 row(s) affected	0.000 sec
- System Tray:** Shows the date and time as 85°F Light rain, ENG, 6:49 PM.

Chapter 4 - Transactions

1. Simple Queries

- Select Operation

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (eng, fitness_tracking_system), Tables (contact_details, equipment, equipment_list, exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, workout_list), Views, Shared Procedures, Administration, Schemas.
- SQL Editor:** SQL File 5*, containing the query: `SELECT * FROM Exercise;`
- Result Grid:** Shows the results of the query, listing rows for Exercise_Id W001 through W004, Category (Cardio, Strength Training, Flexibility), and Calories_burned_per_minute (10.5, 5.75, 8, 12.3).
- Action Output:** Displays log entries for database actions, including an update, a delete, and the execution of the SELECT query.
- System Tray:** Shows the date and time as 85°F Light rain, 8:51 PM.

- Project operation

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (eng, fitness_tracking_system), Tables (contact_details, equipment, equipment_list, exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, workout_list), Views, Shared Procedures, Administration, Schemas.
- SQL Editor:** SQL File 5*, containing the query: `SELECT Calories_burned, Hours FROM workout;`
- Result Grid:** Shows the results of the query, listing rows for Calories_burned (600, 250, 600, 650, 300, 400, 700, 300, 350, 420) and Hours (1, 0, 1, 1, 1, 1, 1, 0, 2, 1).
- Action Output:** Displays log entries for database actions, including a delete from Location_content and the execution of the SELECT query.
- System Tray:** Shows the date and time as 85°F Light rain, 8:51 PM.

- Cartesian product operation

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Local instance MySQL80, Schemas (eng, fitness_tracking_system), Tables (contact_details, equipment, equipment_list, exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, workout_list), Views, Stored Procedures, Administration, Schemas.
- SQL Editor:** FIRST_SQL tab, SQL File 4*, SQL File 5*. The query is:


```
533 -- 3. Cartesian product operation
534 • SELECT u.First_name, w.Workout_name
535   FROM Users As u
536   CROSS JOIN Workout_list As w;
```
- Result Grid:** Shows the result of the query, displaying 12 rows of data where each user (Ghara, Uma, Kasun, Sera, amala, Kamal) has two workout names (workout1, workout2).

First_name	Workout_name
Ghara	workout1
Uma	workout1
Kasun	workout1
Sera	workout1
amala	workout1
Kamal	workout1
Ghara	workout2
Uma	workout2
Kasun	workout2
Sera	workout2
amala	workout2
Kamal	workout2
- Output:** Action Output table showing the execution of the query and its duration.

- Creating user view

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Local instance MySQL80, Schemas (eng, fitness_tracking_system), Tables (contact_details, equipment, equipment_list, exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, workout_list), Views, Stored Procedures, Administration, Schemas.
- SQL Editor:** FIRST_SQL tab, SQL File 4*, SQL File 5*. The query is:


```
541 -- 4. Creating a user view
542 • CREATE VIEW User_Details AS
543   SELECT User_id, First_name, Last_name, Age, Weight, Height
544   FROM Users;
```
- Output:** Action Output table showing the creation of the view and its duration.

• Renaming Operation

```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator FIRST_SQL SQL File 4* SQL File 5* ...
SCHEMAS eng fitness_tracking_system Tables
552
553
554 -- 5. Renaming operation
555 • select Equipment_id as E_ID, Location_id as L_ID from Location_content;
Result Grid | Filter Rows: | Export: | Wrap Cell Content: |
E_ID L_ID
EQ001 A001
EQ002 A001
EQ001 A002
EQ003 A002
EQ004 A006
Action Output
# Time Action Message Duration / Fetch
186 18:51:11 SELECT Calories_burned, Hours FROM workout LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
187 18:52:28 SELECT u.First_name, w.Workout_name FROM Users As u CROSS JOIN Workout... 36 row(s) returned 0.016 sec / 0.000 sec
188 18:53:49 CREATE VIEW User_Details AS SELECT User_id, First_name, Last_name, Age, W... 0 row(s) affected 0.000 sec
189 18:54:50 select Equipment_id as E_ID, Location_id as L_ID from Location_content LIMIT 0, 1... 5 row(s) returned 0.000 sec / 0.000 sec

```

• Aggregation Operation

```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator FIRST_SQL SQL File 4* SQL File 5* ...
SCHEMAS eng fitness_tracking_system Tables
561 -- 6. Aggregation function (Average)
562 • SELECT AVG(Weight) AS AverageWeight
563 FROM Users;
Result Grid | Filter Rows: | Export: | Wrap Cell Content: |
AverageWeight
49.666666666666664
Action Output
# Time Action Message Duration / Fetch
187 18:52:28 SELECT u.First_name, w.Workout_name FROM Users As u CROSS JOIN Workout... 36 row(s) returned 0.016 sec / 0.000 sec
188 18:53:49 CREATE VIEW User_Details AS SELECT User_id, First_name, Last_name, Age, W... 0 row(s) affected 0.000 sec
189 18:54:50 select Equipment_id as E_ID, Location_id as L_ID from Location_content LIMIT 0, 1... 5 row(s) returned 0.000 sec / 0.000 sec
190 18:55:20 SELECT AVG(Weight) AS AverageWeight FROM Users LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec

```

- Like keyword use

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator FIRST_SQL SQL File 4* SQL File 5*

SCHEMAS

fitness_tracking_system

Tables

contact_details equipment equipment_list exercise exercise_list exercise_log heart_rate_record location location_content nutrition_log users workout workout_list

Views

Stored Procedures

Administration Schemas

No object selected

Object Info Session

-- 7. Use of LIKE keyword

```
570 -- 7. Use of LIKE keyword
571 • select *from workout where Hours like '%1%';
572
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types |

User_Id	Workout_Id	Start_time	Calories_burned	Hours	Minutes	Seconds
1	W001	07:00:00	600	1	10	0
1	W003	07:30:00	600	1	10	0
1	W004	07:50:00	650	1	10	0
1	W005	07:20:00	300	1	10	0
1	W007	06:00:00	400	1	10	0
2	W001	07:30:00	700	1	30	10
5	W002	16:30:00	420	1	45	25
*	HULL	HULL	HULL	HULL	HULL	HULL

Output

#	Time	Action	Message	Duration / Fetch
188	18:53:49	CREATE VIEW User_Details AS SELECT User_Id, First_name, Last_name, Age, W...	0 row(s) affected	0.000 sec
189	18:54:50	select Equipment_Id as E_ID, Location_Id as L_ID from Location_content LIMIT 0, 1...	5 row(s) returned	0.000 sec / 0.000 sec
190	18:55:20	SELECT AVG(Weight) AS AverageWeight FROM Users LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
191	18:55:48	select *from workout where Hours like '%1%' LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

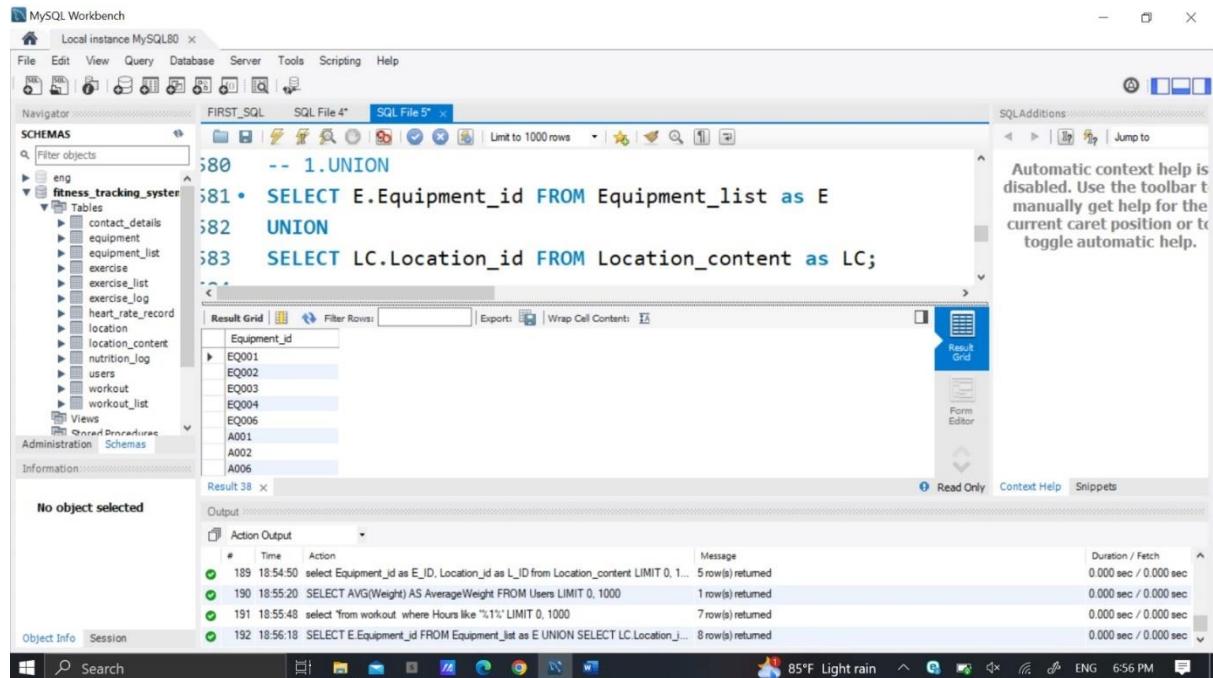
SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

85°F Light rain

2. Complex Queries

- Union



The screenshot shows the MySQL Workbench interface with a SQL editor window containing the following code:

```
580 -- 1.UNION
581 • SELECT E.Equipment_id FROM Equipment_list as E
582 UNION
583 SELECT LC.Location_id FROM Location_content as LC;
```

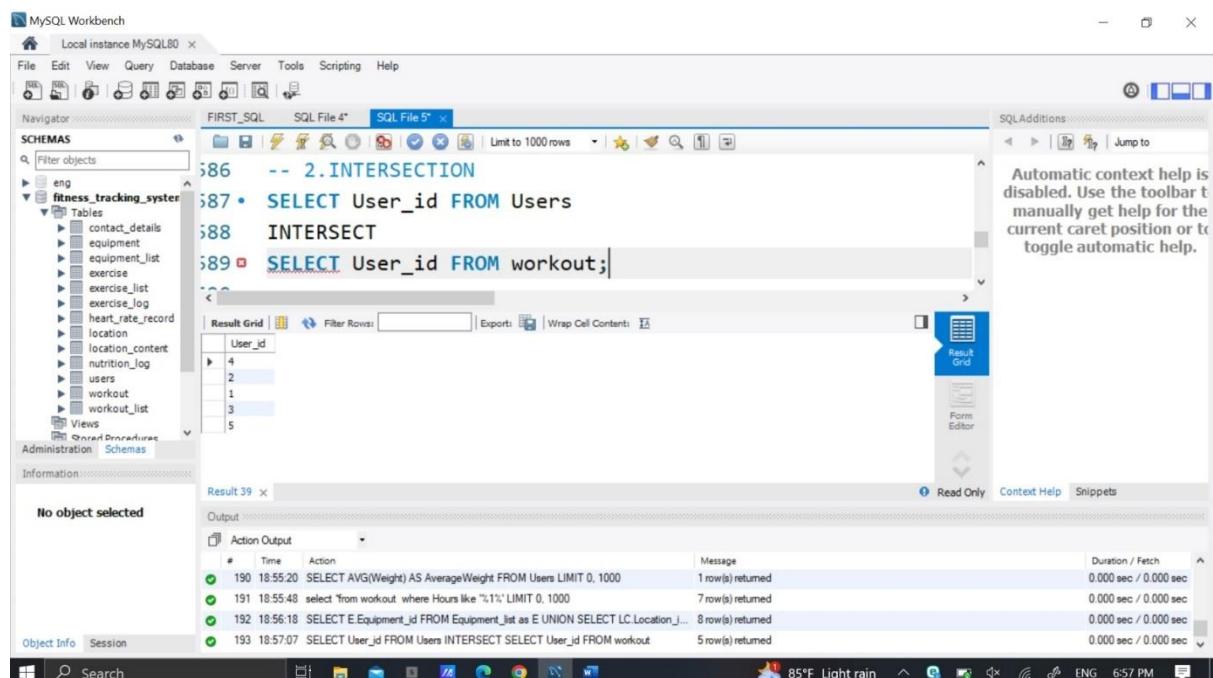
The Result Grid shows the output:

Equipment_id
EQ001
EQ002
EQ003
EQ004
EQ005
A001
A002
A006

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
189	18:54:50	selected Equipment_id as E_ID, Location_id as L_ID from Location_content LIMIT 0, 1...	5 row(s) returned	0.000 sec / 0.000 sec
190	18:55:20	SELECT AVG(Weight) AS AverageWeight FROM Users LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
191	18:55:48	select 'from workout, where Hours like "%1%" LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
192	18:56:18	SELECT E.Equipment_id FROM Equipment_list as E UNION SELECT LC.Location_id...	8 row(s) returned	0.000 sec / 0.000 sec

- Intersection



The screenshot shows the MySQL Workbench interface with a SQL editor window containing the following code:

```
586 -- 2.INTERSECTION
587 • SELECT User_id FROM Users
588 INTERSECT
589 □ SELECT User_id FROM workout;
```

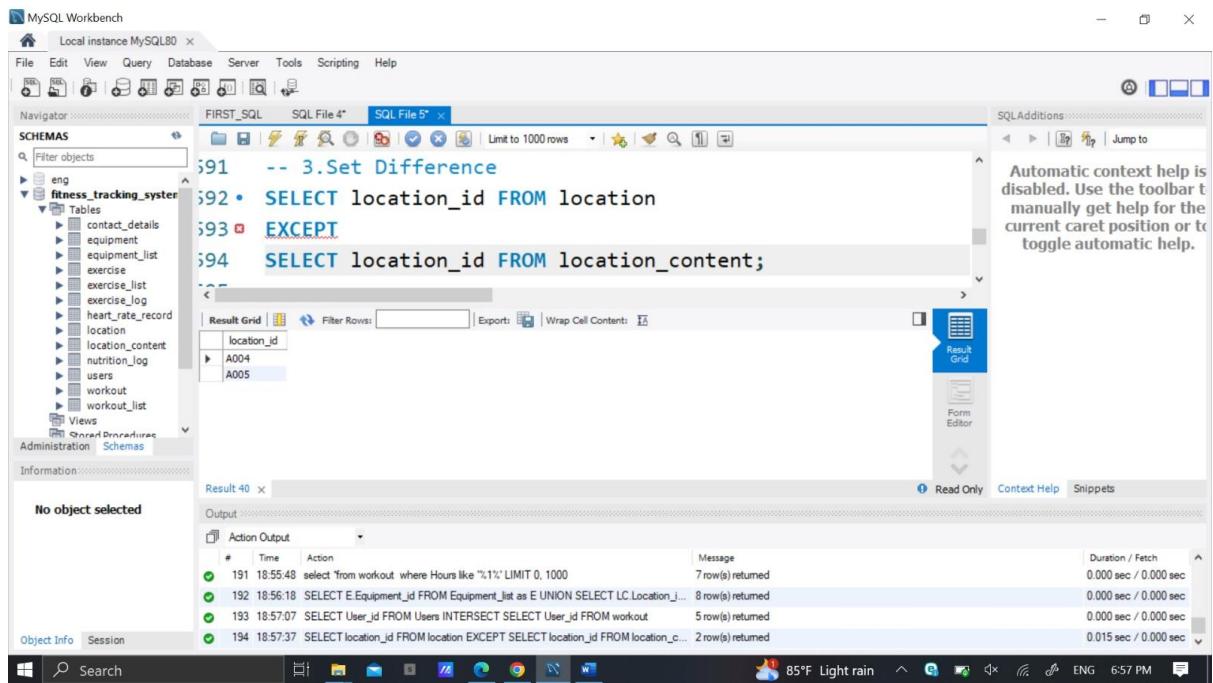
The Result Grid shows the output:

User_id
4
2
1
3
5

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
190	18:55:20	SELECT AVG(Weight) AS AverageWeight FROM Users LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
191	18:55:48	select 'from workout, where Hours like "%1%" LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
192	18:56:18	SELECT E.Equipment_id FROM Equipment_list as E UNION SELECT LC.Location_id...	8 row(s) returned	0.000 sec / 0.000 sec
193	18:57:07	SELECT User_id FROM Users INTERSECT SELECT User_id FROM workout	5 row(s) returned	0.000 sec / 0.000 sec

• Set difference



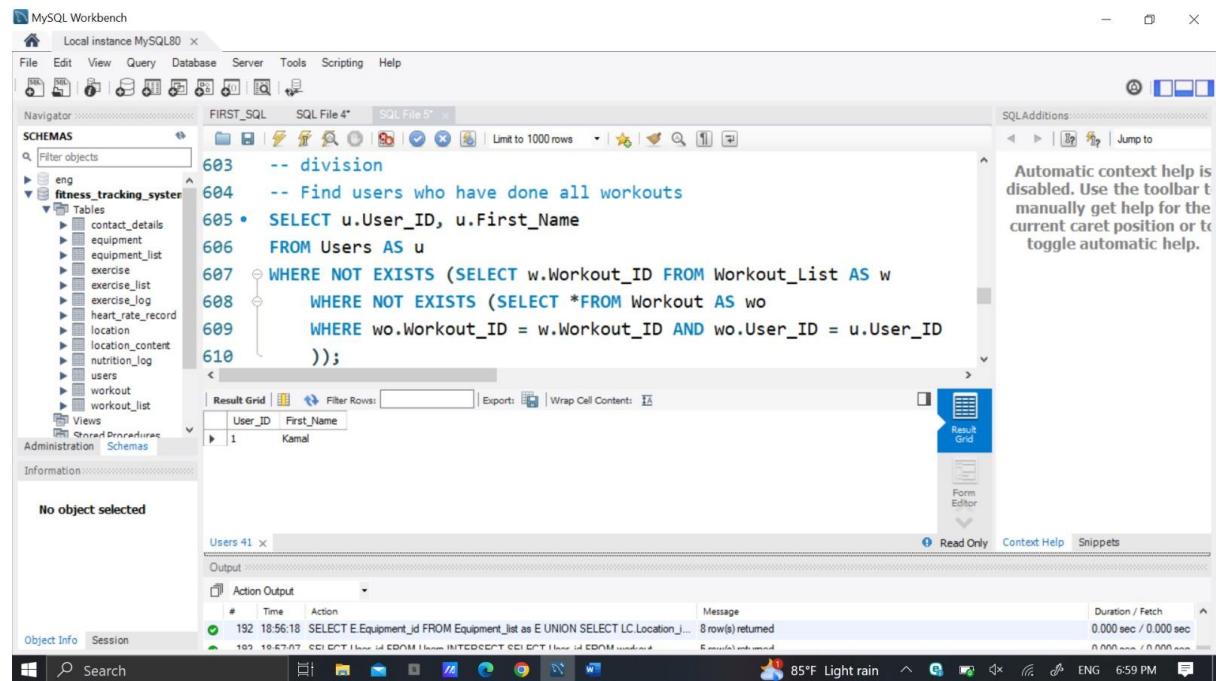
The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard database management icons.
- Navigator:** Shows the schema `fitness_tracking_system` with tables like `contact_details`, `equipment`, etc.
- SQL Editor:** Contains the following SQL code:


```

591 -- 3. Set Difference
592 • SELECT location_id FROM location
593 □ EXCEPT
594 SELECT location_id FROM location_content;
      
```
- Result Grid:** Displays the results of the query, showing two rows: A004 and A005.
- Output Window:** Shows the execution log with four entries, all marked with a green checkmark.
- System Tray:** Shows the date and time as 85°F Light rain, ENG 6:57 PM.

• Division



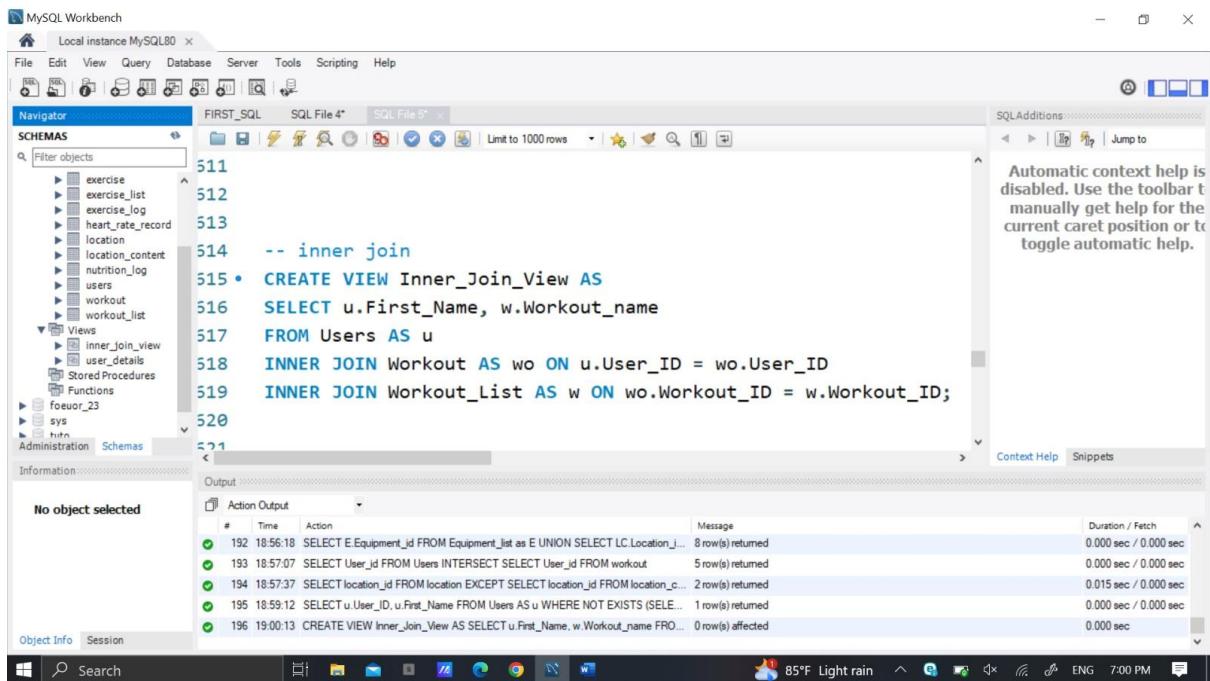
The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard database management icons.
- Navigator:** Shows the schema `fitness_tracking_system` with tables like `contact_details`, `equipment`, etc.
- SQL Editor:** Contains the following SQL code:


```

603 -- division
604 -- Find users who have done all workouts
605 • SELECT u.User_ID, u.First_Name
606   FROM Users AS u
607   WHERE NOT EXISTS (SELECT w.Workout_ID FROM Workout_List AS w
608                      WHERE NOT EXISTS (SELECT * FROM Workout AS wo
609                                WHERE wo.Workout_ID = w.Workout_ID AND wo.User_ID = u.User_ID
610                      ));
      
```
- Result Grid:** Displays the results of the query, showing one row: Kamal.
- Output Window:** Shows the execution log with two entries, both marked with a green checkmark.
- System Tray:** Shows the date and time as 85°F Light rain, ENG 6:59 PM.

- Inner join



The screenshot shows the MySQL Workbench interface with the following details:

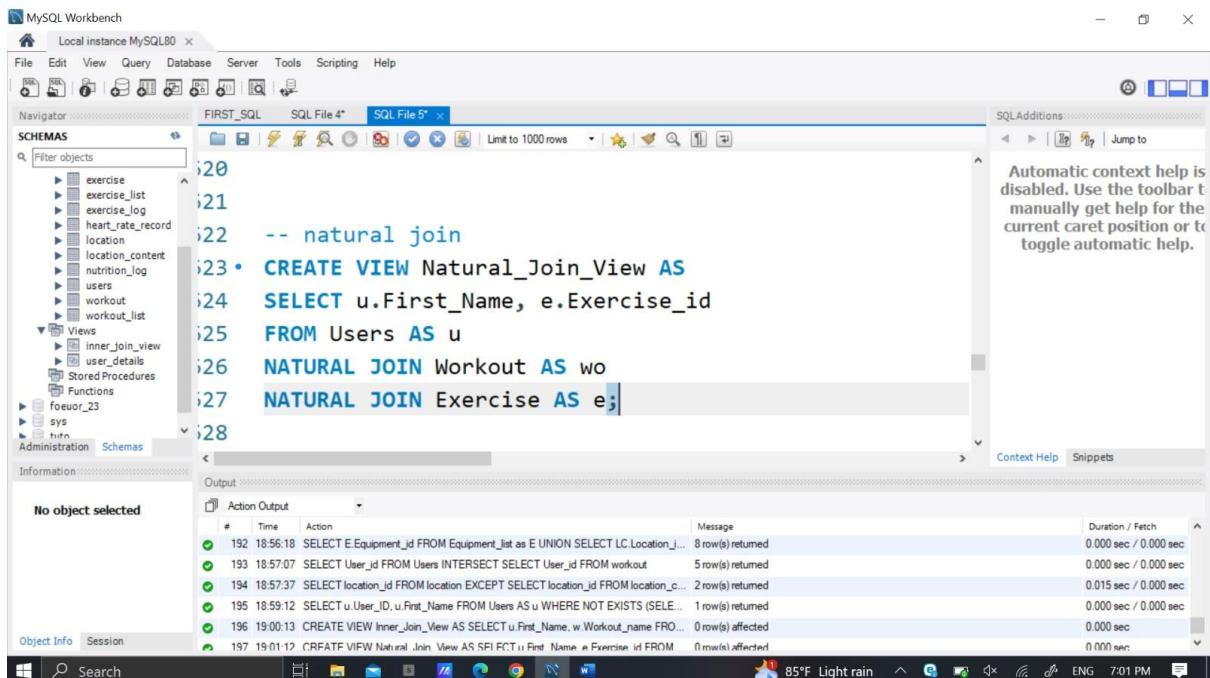
- Navigator:** Shows the database schema with tables like exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, and workout_list.
- SQL Editor:** Contains the SQL code for creating a view named Inner_Join_View:


```

511
512
513
514    -- inner join
515 • CREATE VIEW Inner_Join_View AS
516     SELECT u.First_Name, w.Workout_name
517     FROM Users AS u
518     INNER JOIN Workout AS wo ON u.User_ID = wo.User_ID
519     INNER JOIN Workout_List AS w ON wo.Workout_ID = w.Workout_ID;
520
521
      
```
- Output:** Displays the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
192	18:56:18	SELECT E.Equipment_id FROM Equipment_list AS E UNION SELECT LC.Location_id...	8 row(s) returned	0.000 sec / 0.000 sec
193	18:57:07	SELECT User_id FROM Users INTERSECT SELECT User_id FROM workout	5 row(s) returned	0.000 sec / 0.000 sec
194	18:57:37	SELECT location_id FROM location EXCEPT SELECT location_id FROM location_c...	2 row(s) returned	0.015 sec / 0.000 sec
195	18:59:12	SELECT u.User_ID, u.First_Name FROM Users AS u WHERE NOT EXISTS (SEL...	1 row(s) returned	0.000 sec / 0.000 sec
196	19:00:13	CREATE VIEW Inner_Join_View AS SELECT u.First_Name, w.Workout_name FRO...	0 row(s) affected	0.000 sec

- Natural Join



The screenshot shows the MySQL Workbench interface with the following details:

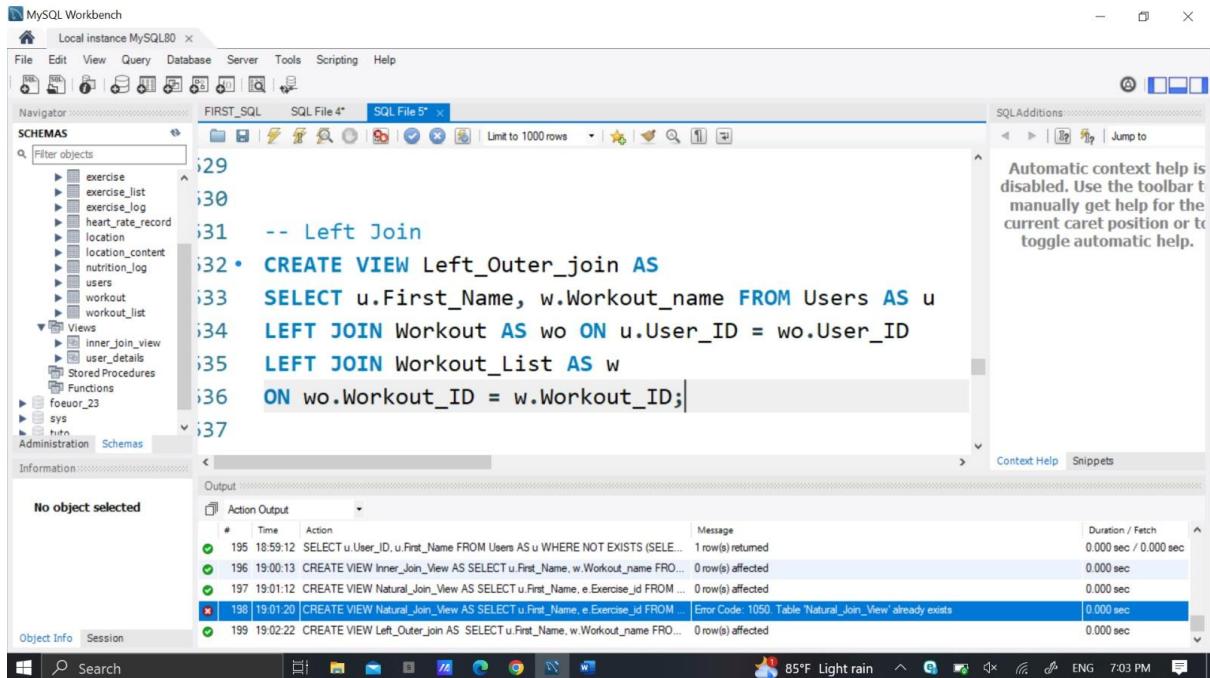
- Navigator:** Shows the database schema with tables like exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, and workout_list.
- SQL Editor:** Contains the SQL code for creating a view named Natural_Join_View:


```

520
521
522    -- natural join
523 • CREATE VIEW Natural_Join_View AS
524     SELECT u.First_Name, e.Exercise_id
525     FROM Users AS u
526     NATURAL JOIN Workout AS wo
527     NATURAL JOIN Exercise AS e;
528
      
```
- Output:** Displays the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
192	18:56:18	SELECT E.Equipment_id FROM Equipment_list AS E UNION SELECT LC.Location_id...	8 row(s) returned	0.000 sec / 0.000 sec
193	18:57:07	SELECT User_id FROM Users INTERSECT SELECT User_id FROM workout	5 row(s) returned	0.000 sec / 0.000 sec
194	18:57:37	SELECT location_id FROM location EXCEPT SELECT location_id FROM location_c...	2 row(s) returned	0.015 sec / 0.000 sec
195	18:59:12	SELECT u.User_ID, u.First_Name FROM Users AS u WHERE NOT EXISTS (SEL...	1 row(s) returned	0.000 sec / 0.000 sec
196	19:00:13	CREATE VIEW Natural_Join_View AS SELECT u.First_Name, w.Workout_name FRO...	0 row(s) affected	0.000 sec
197	19:01:12	CRFATF VIFW Natural_Join_View AS SFI FCT u First Name e Exercise_id FROM	0 row(s) affected	0.000 sec

- **Left Join**



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80 x, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, workout_list), Views (inner_join_view, user_details), Stored Procedures, Functions, fœuer_23, sys, htn.
- SQL Editor:** SQL File 5*, containing the following SQL code:

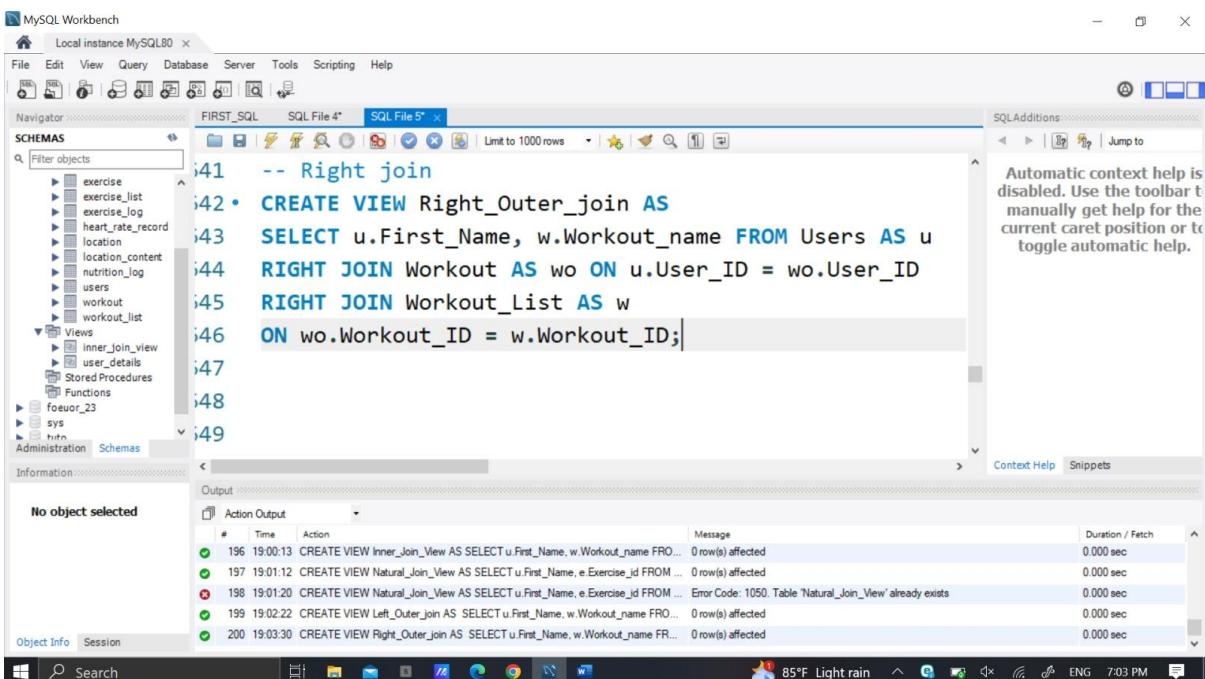
```

i29
i30
i31 -- Left Join
i32 • CREATE VIEW Left_Outer_join AS
i33     SELECT u.First_Name, w.Workout_name FROM Users AS u
i34     LEFT JOIN Workout AS wo ON u.User_ID = wo.User_ID
i35     LEFT JOIN Workout_List AS w
i36     ON wo.Workout_ID = w.Workout_ID;
i37

```
- Output:** Action Output table showing the execution of the queries:

#	Time	Action	Message	Duration / Fetch
195	18:59:12	SELECT u.User_ID, u.First_Name FROM Users AS u WHERE NOT EXISTS (SELE...	1 row(s) returned	0.000 sec / 0.000 sec
196	19:00:13	CREATE VIEW Inner_Join_View AS SELECT u.First_Name, w.Workout_name FRO...	0 row(s) affected	0.000 sec
197	19:01:12	CREATE VIEW Natural_Join_View AS SELECT u.First_Name, e.Exercise_Id FRO...	0 row(s) affected	0.000 sec
198	19:01:20	CREATE VIEW Left_Outer_join AS SELECT u.First_Name, w.Workout_name FRO...	Error Code: 1050. Table 'Natural_Join_View' already exists	0.000 sec
199	19:02:22	CREATE VIEW Left_Outer_join AS SELECT u.First_Name, w.Workout_name FRO...	0 row(s) affected	0.000 sec
- System Bar:** Context Help, Snippets, 85°F Light rain, ENG, 7:03 PM.

- **Right join**



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80 x, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, workout_list), Views (inner_join_view, user_details), Stored Procedures, Functions, fœuer_23, sys, htn.
- SQL Editor:** SQL File 5*, containing the following SQL code:

```

i41 -- Right join
i42 • CREATE VIEW Right_Outer_join AS
i43     SELECT u.First_Name, w.Workout_name FROM Users AS u
i44     RIGHT JOIN Workout AS wo ON u.User_ID = wo.User_ID
i45     RIGHT JOIN Workout_List AS w
i46     ON wo.Workout_ID = w.Workout_ID;
i47
i48
i49

```
- Output:** Action Output table showing the execution of the queries:

#	Time	Action	Message	Duration / Fetch
196	19:00:13	CREATE VIEW Inner_Join_View AS SELECT u.First_Name, w.Workout_name FRO...	0 row(s) affected	0.000 sec
197	19:01:12	CREATE VIEW Natural_Join_View AS SELECT u.First_Name, e.Exercise_Id FRO...	0 row(s) affected	0.000 sec
198	19:01:20	CREATE VIEW Natural_Join_View AS SELECT u.First_Name, e.Exercise_Id FRO...	Error Code: 1050. Table 'Natural_Join_View' already exists	0.000 sec
199	19:02:22	CREATE VIEW Left_Outer_join AS SELECT u.First_Name, w.Workout_name FRO...	0 row(s) affected	0.000 sec
200	19:03:30	CREATE VIEW Right_Outer_join AS SELECT u.First_Name, w.Workout_name FR...	0 row(s) affected	0.000 sec
- System Bar:** Context Help, Snippets, 85°F Light rain, ENG, 7:03 PM.

- Full outer join

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, workout_list), Views (inner_join_view, user_details), Stored Procedures, Functions, foever_23, sys, info.
- SQL Editor:** SQL File 4*, SQL File 5*. The code is as follows:

```

650 -- Full Outer join
651 • CREATE VIEW full_outer AS
652     SELECT u.First_Name, w.Workout_name
653     FROM Users AS u
654     LEFT OUTER JOIN Workout AS wkt ON u.User_ID = wkt.User_ID
655     LEFT OUTER JOIN Workout_List AS w ON wkt.Workout_ID = w.Workout_ID
656     UNION
657     SELECT u.First_Name, w.Workout_name
658     FROM Workout AS wkt
659     RIGHT OUTER JOIN Users AS u ON u.User_ID = wkt.User_ID
660     RIGHT OUTER JOIN Workout_List AS w ON wkt.Workout_ID = w.Workout_ID;
  
```

- Output Window:** Action Output table showing the execution of the statements. The log includes:

#	Time	Action	Message	Duration / Fetch
197	19:01:12	CREATE VIEW Natural_Join_View AS SELECT u.First_Name, e.Exercise_id FROM ...	0 row(s) affected	0.000 sec
198	19:01:20	CREATE VIEW Natural_Join_View AS SELECT u.First_Name, e.Exercise_id FROM ...	Error Code: 1050. Table 'Natural_Join_View' already exists	0.000 sec
199	19:02:22	CREATE VIEW Left_Outer_join AS SELECT u.First_Name, w.Workout_name FRO...	0 row(s) affected	0.000 sec
200	19:03:30	CREATE VIEW Right_Outer_join AS SELECT u.First_Name, w.Workout_name FR...	0 row(s) affected	0.000 sec
201	19:04:27	CREATE VIEW full_outer AS SELECT u.First_Name, w.Workout_name FROM User...	0 row(s) affected	0.015 sec
202	19:04:55	CREATE VIEW outer_union AS SELECT E.Equipment_id, E.Exercise_id FROM Eq...	0 row(s) affected	0.016 sec

- Outer union view

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, workout_list), Views (inner_join_view, user_details), Stored Procedures, Functions, foever_23, sys, info.
- SQL Editor:** SQL File 4*, SQL File 5*. The code is as follows:

```

65
66 -- CREATE VIEW Outer_Union_View AS
67 • CREATE VIEW outer_union AS
68     SELECT E.Equipment_id, E.Exercise_id
69     FROM Equipment as E
70     UNION
71     SELECT LC.Equipment_id, LC.Location_id
72     FROM Location_content as LC;
  
```

- Output Window:** Action Output table showing the execution of the statements. The log includes:

#	Time	Action	Message	Duration / Fetch
199	19:02:22	CREATE VIEW Left_Outer_join AS SELECT u.First_Name, w.Workout_name FRO...	0 row(s) affected	0.000 sec
200	19:03:30	CREATE VIEW Right_Outer_join AS SELECT u.First_Name, w.Workout_name FR...	0 row(s) affected	0.000 sec
201	19:04:27	CREATE VIEW full_outer AS SELECT u.First_Name, w.Workout_name FROM User...	0 row(s) affected	0.015 sec
202	19:04:55	CREATE VIEW outer_union AS SELECT E.Equipment_id, E.Exercise_id FROM Eq...	0 row(s) affected	0.016 sec

- Nested Query

Query 1

```
-- nested queries
578 • SELECT u.First_Name, w.Workout_Id
579 FROM (SELECT * FROM Users WHERE Age > 20) AS u
580 INNER JOIN (SELECT * FROM Workout WHERE Hours >= 1)
581 AS w ON u.User_ID = w.User_ID;
```

First_Name	Workout_Id
Kamal	W001
Kamal	W003
Kamal	W004
Kamal	W005
Kamal	W007
amala	W001
Kasun	W004

Action Output

#	Time	Action	Message	Duration / Fetch
203	19:05:45	SELECT u.First_Name, w.Workout_Id FROM (SELECT * FROM Users WHERE Age...	7 row(s) returned	0.000 sec / 0.000 sec
204	19:07:52	SELECT u.First_Name, w.Workout_Id FROM (SELECT * FROM Users WHERE Age...	7 row(s) returned	0.000 sec / 0.000 sec

Query 2

```
587 • SELECT u.First_Name, AVG(w.Calories_burned)
588 AS AvgCaloriesBurned
589 FROM Users AS u
590 INNER JOIN Workout AS w ON u.User_ID = w.User_ID
591 GROUP BY u.First_Name;
```

First_Name	AvgCaloriesBurned
Kamal	466.6667
amala	700.0000
Sera	300.0000
Kasun	350.0000
Uma	420.0000

Action Output

#	Time	Action	Message	Duration / Fetch
204	19:07:52	SELECT u.First_Name, w.Workout_Id FROM (SELECT * FROM Users WHERE Age...	7 row(s) returned	0.000 sec / 0.000 sec
205	19:08:25	SELECT u.First_Name, AVG(w.Calories_burned) AS AvgCaloriesBurned FROM User...	5 row(s) returned	0.000 sec / 0.000 sec

Query 3

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the database schema with various tables like exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, and workout_list.
- SQL Editor:** Contains the following SQL query:

```
193 • SELECT First_Name FROM Users
194 • WHERE User_ID IN (
195     SELECT User_ID FROM Workout
196     INTERSECT
197     SELECT User_ID FROM Nutrition_Log
198 );
```
- Result Grid:** Displays the results of the query, showing four rows of data: Kamal, amala, Sera, and Kasun.
- Output:** Shows the action output with two entries:

#	Time	Action	Message	Duration / Fetch
206	19:08:59	SELECT First_Name FROM Users WHERE User_ID IN (SELECT User_ID FROM... 4 row(s) returned	0.000 sec / 0.000 sec
207	19:09:11	SELECT First_Name FROM Users WHERE User_ID IN (SELECT User_ID FROM... 4 row(s) returned	0.000 sec / 0.000 sec
- System Tray:** Shows the Windows taskbar with icons for File Explorer, Mail, Task View, Start, Taskbar settings, and a system status bar indicating 85°F, Light rain, ENG, and 7:09 PM.

Chapter 5 - Database Tuning

1. Union

Before Tuning

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (fitness_tracking_system), Administration, Information.
- SQL Editor:** FIRST_SQL, SQL File 4*, SQL File 4*. The code is:

```
670 -- tuning
671 -- 1.UNION
672 -- before tuning
673 • Explain SELECT E.Equipment_id FROM Equipment_list as E
UNION
675 SELECT LC.Location_id FROM Location_content as LC;
```
- Result Grid:** Shows the execution plan for the query. The first row is highlighted in blue.
- Output:** Shows the log of actions and their durations.
- System Bar:** Shows the system tray with icons for network, battery, and time (91°F Mostly cloudy, ENG 4:52 PM).

After Tuning

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (fitness_tracking_system), Administration, Information.
- SQL Editor:** FIRST_SQL, SQL File 4*, SQL File 4*. The code is:

```
678
679 -- after tuning
680 • Explain SELECT Equipment_id FROM Equipment
UNION ALL
682 SELECT Location_id FROM Location_content;
```
- Result Grid:** Shows the execution plan for the query. The first row is highlighted in blue.
- Output:** Shows the log of actions and their durations.
- System Bar:** Shows the system tray with icons for network, battery, and time (91°F Mostly cloudy, ENG 4:53 PM).

2. Intersection

Before Tuning

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `fitness_tracking_system` containing tables, views, stored procedures, and functions.
- SQL Editor:** Contains the following SQL code:

```
684
685  -- 2. INTERSECTION
686  -- before tuning
687 • Explain SELECT User_id FROM Users
688 INTERSECT
689 • SELECT User_id FROM workout;
```
- Result Grid:** Displays the execution plan for the query, showing three rows in the table.
- Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
94	16:50:26	SELECT First_Name FROM Users WHERE User_ID IN (SELECT User_ID FRO...	0.016 sec / 0.000 sec
95	16:51:09	Explain SELECT E.Equipment_id FROM Equipment_Jet AS E UNION SELECT LC...	3 row(s) returned	0.000 sec / 0.000 sec
96	16:53:24	Explain SELECT Equipment_id FROM Equipment UNION All SELECT Location_id...	2 row(s) returned	0.016 sec / 0.000 sec
97	16:54:14	Explain SELECT User_id FROM Users INTERSECT SELECT User_id FROM work...	3 row(s) returned	0.000 sec / 0.000 sec

After Tuning

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `fitness_tracking_system` containing tables, views, stored procedures, and functions.
- SQL Editor:** Contains the following SQL code:

```
690
691  -- After Tunning
692 • explain SELECT DISTINCT u.User_id
693   FROM Users AS u
694   INNER JOIN workout AS w ON u.User_id = w.User_id;
```
- Result Grid:** Displays the execution plan for the query, showing two rows in the table.
- Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
95	16:51:09	Explain SELECT E.Equipment_id FROM Equipment_Jet AS E UNION SELECT LC...	3 row(s) returned	0.000 sec / 0.000 sec
96	16:53:24	Explain SELECT Equipment_id FROM Equipment UNION All SELECT Location_id...	2 row(s) returned	0.016 sec / 0.000 sec
97	16:54:14	Explain SELECT User_id FROM Users INTERSECT SELECT User_id FROM work...	3 row(s) returned	0.000 sec / 0.000 sec
98	16:54:36	explain SELECT DISTINCT u.User_id FROM Users AS u INNER JOIN workout A...	2 row(s) returned	0.000 sec / 0.000 sec

3. Set Difference

Before Tuning

The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab active. In the SQL Editor, the following code is present:

```
-- 3. Set Difference
-- before tuning
Explain SELECT location_id FROM location
EXCEPT
SELECT location_id FROM location_content;
```

The 'Result Grid' pane displays the execution plan for the 'Explain' command. The results are as follows:

ID	Select Type	Table	Partition	Type	Possible Keys	Key	Key_Len	Ref	Rows	Filtered	Extra
1	PRIMARY	location	NULL	index	NULL	PRIMARY	42	NULL	5	100.00	Using index
2	EXCEPT	location_content	NULL	index	NULL	Location_id	42	NULL	5	100.00	Using index
3	EXCEPT RESULT	<except...>	ALL	ALL	NULL	NULL	NULL	NULL	NULL	NULL	Using temporary

The 'Output' pane shows the history of actions taken by the system:

#	Time	Action	Message	Duration / Fetch
96	16:53:24	Explain SELECT Equipment_id FROM Equipment UNION All SELECT Location_id...	2 row(s) returned	0.016 sec / 0.000 sec
97	16:54:14	Explain SELECT User_id FROM Users INTERSECT SELECT User_id FROM work...	3 row(s) returned	0.000 sec / 0.000 sec
98	16:54:36	explain SELECT DISTINCT u.User_id FROM Users AS u INNER JOIN workout A...	2 row(s) returned	0.000 sec / 0.000 sec
99	16:55:18	Explain SELECT location_id FROM location EXCEPT SELECT location_id FROM...	3 row(s) returned	0.000 sec / 0.000 sec

After Tuning

The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab active. In the SQL Editor, the following code is present:

```
-- after tuning
Explain SELECT l.Location_id
FROM location AS l
LEFT JOIN location_content AS lc ON l.Location_id = lc.Location_id
WHERE lc.Location_id IS NULL;
```

The 'Result Grid' pane displays the execution plan for the 'Explain' command. The results are as follows:

ID	Select Type	Table	Partition	Type	Possible Keys	Key	Key_Len	Ref	Rows	Filtered	Extra
1	SIMPLE	l	NULL	index	NULL	PRIMARY	42	NULL	5	100.00	Using index
1	SIMPLE	lc	NULL	ref	Location_id	Location_id	42	fit...	1	100.00	Using where;...

The 'Output' pane shows the history of actions taken by the system:

#	Time	Action	Message	Duration / Fetch
97	16:54:14	Explain SELECT User_id FROM Users INTERSECT SELECT User_id FROM work...	3 row(s) returned	0.000 sec / 0.000 sec
98	16:54:36	explain SELECT DISTINCT u.User_id FROM Users AS u INNER JOIN workout A...	2 row(s) returned	0.000 sec / 0.000 sec
99	16:55:18	Explain SELECT location_id FROM location EXCEPT SELECT location_id FROM...	3 row(s) returned	0.000 sec / 0.000 sec
100	16:56:50	Explain SELECT l.Location_id FROM location AS l LEFT JOIN location_content A...	2 row(s) returned	0.000 sec / 0.000 sec

4. Division

Before Tuning

The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab selected. The SQL editor contains the following code:

```
741 -- division
742 -- before tuning
743 • Explain SELECT u.User_ID, u.First_Name
744   FROM Users AS u WHERE NOT EXISTS (SELECT w.Workout_ID
745     FROM Workout_List AS w
746     WHERE NOT EXISTS (SELECT * FROM Workout AS wo
747       WHERE wo.Workout_ID = w.Workout_ID AND wo.User_ID = u.User_ID
748     ));
```

The 'Result Grid' pane shows the execution plan for the 'Explain' command. The output is as follows:

ID	Select Type	Table	Partition Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra	
1	PRIMARY	u	NULL	ALL	NULL	NULL	NULL	6	100.00	Using where	
2	DEPENDENT SUBQUERY	w	NULL	index	NULL	PRIMARY	42	7	100.00	Using index	
2	DEPENDENT SUBQUERY	<subquery>	NULL	eq_ref	<auto_distinct>	<auto_distinct>	ft...	1	100.00	Using where;...	
3	MATERIALIZED	wo	NULL	ref	PRIMARY, fk...	PRIMARY	4	ft...	2	100.00	Using index

The 'Output' pane shows two log entries:

- 210 19:15:39 Explain SELECT u.First_Name, AVG(w.Calories_burned) AS AvgCaloriesBurned FROM Users AS u WHERE NOT EXISTS (SELECT w.Workout_ID FROM Workout_List AS w WHERE w.Workout_ID = u.User_ID) 2 row(s) returned 0.000 sec / 0.000 sec
- 211 19:18:06 Explain SELECT u.User_ID, u.First_Name FROM Users AS u WHERE NOT EXISTS (SELECT w.Workout_ID FROM Workout_List AS w WHERE w.Workout_ID = u.User_ID) 4 row(s) returned 0.000 sec / 0.000 sec

After Tuning

The screenshot shows the MySQL Workbench interface with the 'FIRST_SQL' tab selected. The SQL editor contains the following code:

```
749 -- after tuning
750 • Explain SELECT u.User_ID, u.First_Name
751   FROM Users AS u
752   LEFT JOIN (SELECT wo.User_ID, wo.Workout_ID FROM Workout AS wo
753     LEFT JOIN Workout_List AS wl ON wo.Workout_ID = wl.Workout_ID
754   ) AS joined ON u.User_ID = joined.User_ID
755   GROUP BY u.User_ID, u.First_Name
756   HAVING COUNT(joined.Workout_ID) = (SELECT COUNT(*) FROM Workout_List);
```

The 'Result Grid' pane shows the execution plan for the 'Explain' command. The output is as follows:

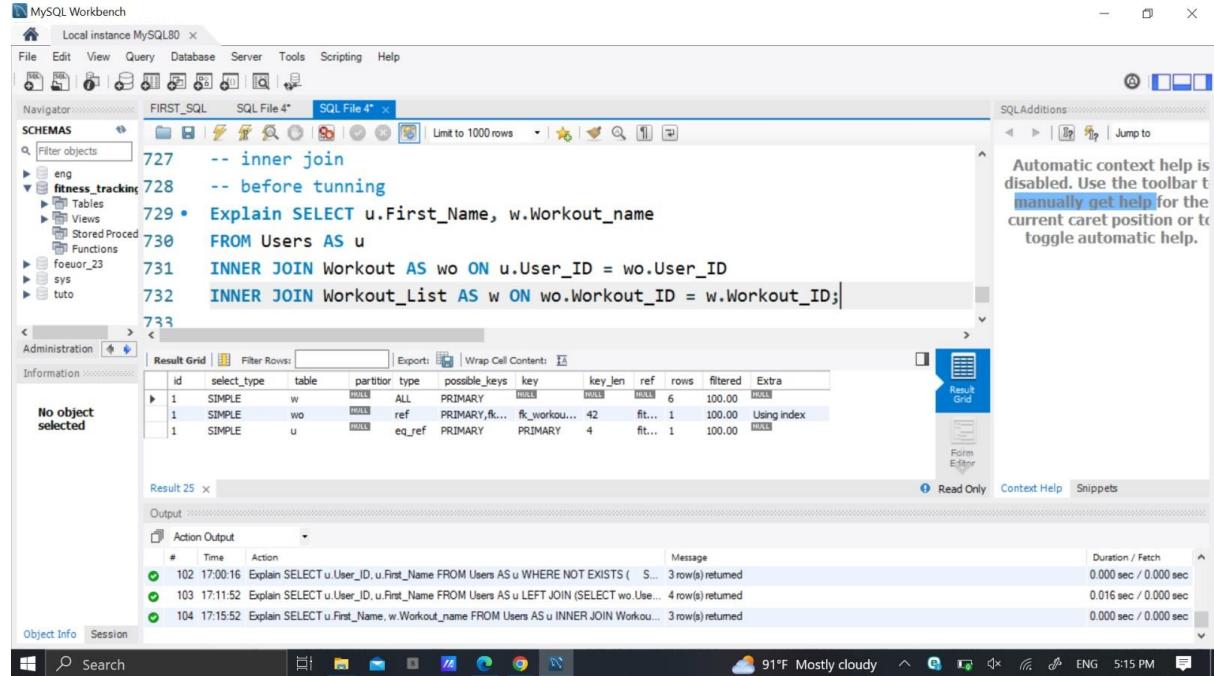
ID	Select Type	Table	Partition Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	u	NULL	ALL	NULL	NULL	NULL	6	100.00	Using temporary...
1	PRIMARY	wo	NULL	ref	PRIMARY	4	ft...	2	100.00	Using index
1	PRIMARY	wl	NULL	eq_ref	PRIMARY	42	ft...	1	100.00	Using index
3	SUBQUERY	Workout...	NULL	index	PRIMARY	42	NULL	7	100.00	Using index

The 'Output' pane shows two log entries:

- 211 19:18:06 Explain SELECT u.User_ID, u.First_Name FROM Users AS u WHERE NOT EXISTS (SELECT w.Workout_ID FROM Workout_List AS w WHERE w.Workout_ID = u.User_ID) 4 row(s) returned 0.000 sec / 0.000 sec
- 212 19:18:37 Explain SELECT u.User_ID, u.First_Name FROM Users AS u LEFT JOIN (SELECT ... 4 row(s) returned 0.000 sec / 0.000 sec

5. Inner join

Before running



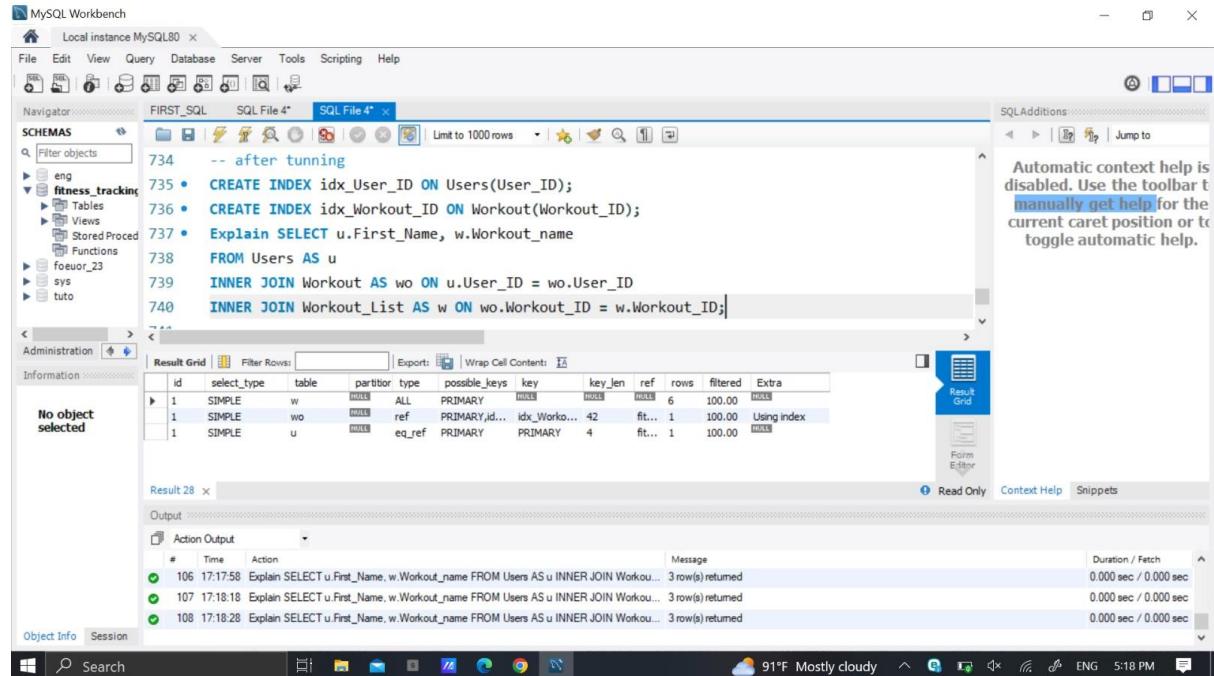
The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (eng, fitness_tracking, foefour_23, sys, tuto), Tables, Views, Stored Proced, Functions.
- SQL Editor:** SQL File 4* (Query tab) containing the following code:

```
727 -- inner join
728 -- before running
729 • Explain SELECT u.First_Name, w.Workout_name
  FROM Users AS u
  INNER JOIN Workout AS wo ON u.User_ID = wo.User_ID
  INNER JOIN Workout_List AS w ON wo.Workout_ID = w.Workout_ID;
```
- Result Grid:** Shows the execution plan for the EXPLAIN command. The output is:

ID	Select_Type	Table	Partion	Type	Possible_Keys	Key	Key_Len	Ref	Rows	Filtered	Extra
1	SIMPLE	w	HULL	ALL	PRIMARY	HULL	HULL	HULL	6	100.00	HULL
1	SIMPLE	wo	HULL	ref	PRIMARY,fk...	fk_workou...	42	ft...	1	100.00	Using index
1	SIMPLE	u	HULL	eq_ref	PRIMARY	PRIMARY	4	ft...	1	100.00	HULL
- Output:** Action Output shows three log entries related to the EXPLAIN command.
- System Bar:** Shows the system tray with icons for network, battery, and time (91°F Mostly cloudy, 5:15 PM).

After running



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (eng, fitness_tracking, foefour_23, sys, tuto), Tables, Views, Stored Proced, Functions.
- SQL Editor:** SQL File 4* (Query tab) containing the following code:

```
734 -- after running
735 • CREATE INDEX idx_User_ID ON Users(User_ID);
736 • CREATE INDEX idx_Workout_ID ON Workout(Workout_ID);
737 • Explain SELECT u.First_Name, w.Workout_name
  FROM Users AS u
  INNER JOIN Workout AS wo ON u.User_ID = wo.User_ID
  INNER JOIN Workout_List AS w ON wo.Workout_ID = w.Workout_ID;
```
- Result Grid:** Shows the execution plan for the EXPLAIN command. The output is:

ID	Select_Type	Table	Partion	Type	Possible_Keys	Key	Key_Len	Ref	Rows	Filtered	Extra
1	SIMPLE	w	HULL	ALL	PRIMARY	HULL	HULL	HULL	6	100.00	HULL
1	SIMPLE	wo	HULL	ref	PRIMARY,id...	idx_Worke...	42	ft...	1	100.00	Using index
1	SIMPLE	u	HULL	eq_ref	PRIMARY	PRIMARY	4	ft...	1	100.00	HULL
- Output:** Action Output shows three log entries related to the EXPLAIN command.
- System Bar:** Shows the system tray with icons for network, battery, and time (91°F Mostly cloudy, 5:18 PM).

6. Natural Join

Before tunning

The screenshot shows the MySQL Workbench interface with the SQL Editor tab open. The code being run is:

```
749 -- Natural join
750 • Explain SELECT u.First_Name, e.Exercise_id
    FROM Users AS u
    NATURAL JOIN Workout AS wo
    NATURAL JOIN Exercise AS e;
```

The Result Grid shows the execution plan:

ID	Select Type	Table	Partition	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra	
1	SIMPLE	e		NULL	index	PRIMARY	Exercise_id	42	NULL	6	100.00	Using index
1	SIMPLE	wo		NULL	ref	PRIMARY,id...	idx_Worke...	42	ft...	1	100.00	Using index
1	SIMPLE	u		NULL	eq_ref	PRIMARY	PRIMARY	4	ft...	1	100.00	NULL

The Action Output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
107	17:18:18	Explain SELECT u.First_Name, w.Workout_name FROM Users AS u INNER JOIN Workout AS wo ON u.User_ID = wo.User_ID INNER JOIN Exercise AS e ON wo.Workout_id = e.Workout_id;	3 row(s) returned	0.000 sec / 0.000 sec
108	17:18:28	Explain SELECT u.First_Name, w.Workout_name FROM Users AS u INNER JOIN Workout AS wo ON u.User_ID = wo.User_ID INNER JOIN Exercise AS e ON wo.Workout_id = e.Workout_id;	3 row(s) returned	0.000 sec / 0.000 sec
109	17:19:17	Explain SELECT u.First_Name, e.Exercise_id FROM Users AS u NATURAL JOIN Workout AS wo NATURAL JOIN Exercise AS e;	3 row(s) returned	0.000 sec / 0.015 sec

After tunning

The screenshot shows the MySQL Workbench interface with the SQL Editor tab open. The code being run is:

```
755 -- after tunning
756 • Explain SELECT u.First_Name, e.Exercise_id
    FROM Users AS u
    INNER JOIN Workout AS wo ON u.User_ID = wo.User_ID
    INNER JOIN Exercise AS e ON wo.Workout_id = e.Workout_id;
```

The Result Grid shows the execution plan:

ID	Select Type	Table	Partition	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra	
1	SIMPLE	e		NULL	index	PRIMARY	Exercise_id	42	NULL	6	100.00	Using index
1	SIMPLE	wo		NULL	ref	PRIMARY,id...	idx_Worke...	42	ft...	1	100.00	Using index
1	SIMPLE	u		NULL	eq_ref	PRIMARY	PRIMARY	4	ft...	1	100.00	NULL

The Action Output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
108	17:18:28	Explain SELECT u.First_Name, w.Workout_name FROM Users AS u INNER JOIN Workout AS wo ON u.User_ID = wo.User_ID INNER JOIN Exercise AS e ON wo.Workout_id = e.Workout_id;	3 row(s) returned	0.000 sec / 0.000 sec
109	17:19:17	Explain SELECT u.First_Name, e.Exercise_id FROM Users AS u NATURAL JOIN Workout AS wo NATURAL JOIN Exercise AS e;	3 row(s) returned	0.000 sec / 0.015 sec
110	17:20:17	Explain SELECT u.First_Name, e.Exercise_id FROM Users AS u INNER JOIN Workout AS wo ON u.User_ID = wo.User_ID INNER JOIN Exercise AS e ON wo.Workout_id = e.Workout_id;	3 row(s) returned	0.000 sec / 0.000 sec

7. Right Join

Before tuning

The screenshot shows the MySQL Workbench interface with the SQL Editor tab open. The code being run is:

```
762 -- Right Join
763 -- before tuning
764 • Explain SELECT u.First_Name, w.Workout_name FROM Users AS u
    RIGHT JOIN Workout AS wo ON u.User_ID = wo.User_ID
    RIGHT JOIN Workout_List AS w ON wo.Workout_ID = w.Workout_ID;
767
768
```

The Result Grid shows the execution plan:

ID	Select Type	Table	Partition	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	SIMPLE	w	HULL	ALL	HULL	HULL	HULL	HULL	6	100.00	HULL
1	SIMPLE	wo	HULL	ref	idx_Workout...	idx_Worko...	42	ft...	1	100.00	Using index
1	SIMPLE	u	HULL	eq_ref	PRIMARY	PRIMARY	4	ft...	1	100.00	HULL

The Output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
109	17:19:17	Explain SELECT u.First_Name, e.Exercise_id FROM Users AS u NATURAL JOIN Workout AS w	3 row(s) returned	0.000 sec / 0.015 sec
110	17:20:17	Explain SELECT u.First_Name, e.Exercise_id FROM Users AS u INNER JOIN Workout AS w	3 row(s) returned	0.000 sec / 0.000 sec
111	17:21:24	Explain SELECT u.First_Name, w.Workout_name FROM Users AS u RIGHT JOIN Workout AS wo	3 row(s) returned	0.000 sec / 0.000 sec

After tuning

The screenshot shows the MySQL Workbench interface with the SQL Editor tab open. The code being run is:

```
768
769     -- after tuning
770 • CREATE INDEX idx_workout_user_id ON Workout(User_id);
771 • Explain SELECT u.First_Name, w.Workout_name FROM Users AS u
    RIGHT JOIN Workout AS wo ON u.User_ID = wo.User_ID
    RIGHT JOIN Workout_List AS w ON wo.Workout_ID = w.Workout_ID;
774
```

The Result Grid shows the execution plan:

ID	Select Type	Table	Partition	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	SIMPLE	w	HULL	ALL	HULL	HULL	HULL	HULL	6	100.00	HULL
1	SIMPLE	wo	HULL	ref	idx_Workout...	idx_Worko...	42	ft...	1	100.00	Using index
1	SIMPLE	u	HULL	eq_ref	PRIMARY	PRIMARY	4	ft...	1	100.00	HULL

The Output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
110	17:20:17	Explain SELECT u.First_Name, e.Exercise_id FROM Users AS u NATURAL JOIN Workout AS w	3 row(s) returned	0.000 sec / 0.000 sec
111	17:21:24	Explain SELECT u.First_Name, w.Workout_name FROM Users AS u RIGHT JOIN Workout AS wo	3 row(s) returned	0.000 sec / 0.000 sec
112	17:22:18	Explain SELECT u.First_Name, w.Workout_name FROM Users AS u RIGHT JOIN Workout AS wo	3 row(s) returned	0.015 sec / 0.000 sec

8. Nested Queries

Query 01

Before tunning

The screenshot shows the MySQL Workbench interface with the SQL Editor tab open. The code is:

```
775 -- Nested
776 -- before tunning
777 • Explain SELECT u.First_Name, w.Workout_Id
778   FROM (SELECT * FROM Users WHERE Age > 20) AS u
779   INNER JOIN (SELECT * FROM Workout WHERE Hours >= 1
780 ) AS w ON u.User_ID = w.User_ID;
781
```

The Result Grid shows the execution plan:

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Length	Ref	Rows	Filtered	Extra
1	SIMPLE	Users	HASH	ALL	PRIMARY	HASH	HASH	6	33.33	Using where	
1	SIMPLE	Workout	HASH	ref	PRIMARY	PRIMARY	4	fit...	2	33.33	Using where

The Action Output pane shows three events:

#	Time	Action	Message	Duration / Fetch
111	17:21:24	Explain SELECT u.First_Name, w.Workout_name FROM Users AS u RIGHT JOIN Workout AS w ON u.User_ID = w.User_ID;	3 row(s) returned	0.000 sec / 0.000 sec
112	17:22:18	Explain SELECT u.First_Name, w.Workout_name FROM Users AS u RIGHT JOIN Workout AS w ON u.User_ID = w.User_ID;	3 row(s) returned	0.015 sec / 0.000 sec
113	17:24:00	Explain SELECT u.First_Name, w.Workout_Id FROM (SELECT * FROM Users WHERE Age > 20) AS u INNER JOIN Workout AS w ON u.User_ID = w.User_ID;	2 row(s) returned	0.000 sec / 0.000 sec

After tunning

The screenshot shows the MySQL Workbench interface with the SQL Editor tab open. The code is:

```
781
782 -- after tunning
783 • Explain SELECT u.First_Name, w.Workout_Id
784   FROM Users AS u
785   INNER JOIN Workout AS w ON u.User_ID = w.User_ID
786   WHERE u.Age > 20 AND w.Hours >= 1;
```

The Result Grid shows the execution plan:

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Length	Ref	Rows	Filtered	Extra
1	SIMPLE	u	HASH	ALL	PRIMARY	HASH	HASH	6	33.33	Using where	
1	SIMPLE	w	HASH	ref	PRIMARY	PRIMARY	4	fit...	2	33.33	Using where

The Action Output pane shows four events:

#	Time	Action	Message	Duration / Fetch
112	17:22:18	Explain SELECT u.First_Name, w.Workout_name FROM Users AS u RIGHT JOIN Workout AS w ON u.User_ID = w.User_ID;	3 row(s) returned	0.015 sec / 0.000 sec
113	17:24:00	Explain SELECT u.First_Name, w.Workout_Id FROM (SELECT * FROM Users WHERE Age > 20) AS u INNER JOIN Workout AS w ON u.User_ID = w.User_ID;	2 row(s) returned	0.000 sec / 0.000 sec
114	17:24:41	Explain SELECT u.First_Name, w.Workout_Id FROM Users AS u INNER JOIN Workout AS w ON u.User_ID = w.User_ID;	2 row(s) returned	0.000 sec / 0.000 sec

9. Full outer join

Before tuning

The screenshot shows the MySQL Workbench interface with the SQL Editor tab open. The query is:

```
790 • explain SELECT u.First_Name, w.Workout_name
791   FROM Users AS u
792     LEFT OUTER JOIN Workout AS wkt ON u.User_ID = wkt.User_ID
793     LEFT OUTER JOIN Workout_List AS w ON wkt.Workout_ID = w.Workout_ID
794   UNION
795   SELECT u.First_Name, w.Workout_name
796   FROM Workout AS wkt
797   RIGHT OUTER JOIN Users AS u ON u.User_ID = wkt.User_ID
798   RIGHT OUTER JOIN Workout List AS w ON wkt.Workout ID = w.Workout ID;
```

The Result Grid shows the execution plan:

ID	Select Type	Table	Partition	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	u	HASH	ALL	HASH	HASH	HASH	HASH	6	100.00	HASH
1	PRIMARY	wkt	HASH	ref	PRIMARY	PRIMARY	4	fit...	2	100.00	Using index
1	PRIMARY	w	HASH	eq_ref	PRIMARY	PRIMARY	42	fit...	1	100.00	HASH
2	UNION	w	HASH	ALL	HASH	HASH	HASH	HASH	6	100.00	HASH
2	UNION	wkt	HASH	ref	PRIMARY,id...	idx_Worke...	42	fit...	1	100.00	Using index
2	UNION	u	HASH	eq_ref	PRIMARY	PRIMARY	4	fit...	1	100.00	HASH
3	UNION RESULT	<union1,...>	HASH	ALL	HASH	HASH	HASH	HASH	HASH	11	Using temporary

The Output pane shows the query was executed successfully in 0.000 sec / 0.000 sec.

After tuning

The screenshot shows the MySQL Workbench interface with the SQL Editor tab open. The query is identical to the one above:

```
801 -- after tuning
802 • Explain SELECT u.First_Name, w.Workout_name
803   FROM Users AS u
804     LEFT JOIN Workout AS wkt ON u.User_ID = wkt.User_ID
805     LEFT JOIN Workout_List AS w ON wkt.Workout_ID = w.Workout_ID
806   UNION ALL
807   SELECT u.First_Name, w.Workout_name FROM Workout AS wkt
808   RIGHT JOIN Users AS u ON u.User_ID = wkt.User_ID
809   RIGHT JOIN Workout List AS w ON wkt.Workout ID = w.Workout ID;
```

The Result Grid shows the execution plan:

ID	Select Type	Table	Partition	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	u	HASH	ALL	HASH	HASH	HASH	HASH	6	100.00	HASH
1	PRIMARY	wkt	HASH	ref	PRIMARY	PRIMARY	4	fit...	2	100.00	Using index
1	PRIMARY	w	HASH	eq_ref	PRIMARY	PRIMARY	42	fit...	1	100.00	HASH
2	UNION	w	HASH	ALL	HASH	HASH	HASH	HASH	6	100.00	HASH
2	UNION	wkt	HASH	ref	PRIMARY,id...	idx_Worke...	42	fit...	1	100.00	Using index
2	UNION	u	HASH	eq_ref	PRIMARY	PRIMARY	4	fit...	1	100.00	HASH

The Output pane shows the query was executed successfully in 0.000 sec / 0.000 sec.

10. Nested Query 2

Before tunning

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, workout_list, Views, inner_join_view, user_details, Stored Procedures, Functions, foetur_23, sys, info, Administration, Schemas). A search bar is present.
- SQL Editor:** FIRST_SQL tab, SQL File 4*, SQL File 5*. The query is:

```
343 -- 10 tunning
344 • Explain SELECT u.First_Name,
345     AVG(w.Calories_burned) AS AvgCaloriesBurned
346     FROM Users AS u INNER JOIN
347     Workout AS w ON u.User_ID = w.User_ID WHERE
348     w.Calories_burned IS NOT NULL GROUP BY u.First_Name;
349
```
- Result Grid:** Shows the execution plan for the EXPLAIN command. The output is:

ID	Select Type	Table	Partition	Type	Possible Keys	Key	Key_len	Ref	Rows	Filtered	Extra
1	SIMPLE	w	NULL	ALL	PRIMARY	NULL	NULL	NULL	11	90.00	Using where...
1	SIMPLE	u	NULL	eq_ref	PRIMARY	PRIMARY	4	fit...	1	100.00	NULL
- Output:** Action Output table showing two rows of logs. The first row is for the EXPLAIN command and the second for the SELECT query. Both have duration and fetch times of 0.000 sec / 0.000 sec.
- System Bar:** Shows the Windows taskbar with various icons and system status (85°F Light rain, ENG, 7:14 PM).

After tunning

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (exercise, exercise_list, exercise_log, heart_rate_record, location, location_content, nutrition_log, users, workout, workout_list, Views, inner_join_view, user_details, Stored Procedures, Functions, foetur_23, sys, info, Administration, Schemas). A search bar is present.
- SQL Editor:** FIRST_SQL tab, SQL File 4*, SQL File 5*. The query is:

```
50 -- after tunning
51 • Explain SELECT u.First_Name, AVG(w.Calories_burned)
52     AS AvgCaloriesBurned
53     FROM Users AS u
54     INNER JOIN Workout AS w ON u.User_ID = w.User_ID
55     GROUP BY u.First_Name;
```
- Result Grid:** Shows the execution plan for the EXPLAIN command. The output is:

ID	Select Type	Table	Partition	Type	Possible Keys	Key	Key_len	Ref	Rows	Filtered	Extra
1	SIMPLE	w	NULL	ALL	PRIMARY	NULL	NULL	NULL	11	100.00	Using temporary...
1	SIMPLE	u	NULL	eq_ref	PRIMARY	PRIMARY	4	fit...	1	100.00	NULL
- Output:** Action Output table showing two rows of logs. The first row is for the EXPLAIN command and the second for the SELECT query. Both have duration and fetch times of 0.000 sec / 0.000 sec.
- System Bar:** Shows the Windows taskbar with various icons and system status (85°F Light rain, ENG, 7:15 PM).