



EE4350 – DATABASE SYSTEMS

MINI PROJECT

GROUP MEMBERS : EG/2021/4449 BULEGODAARACHCHI D. C.
EG/2021/4456 DASANAYAKA D. M. D. P. A
EG/2021/4465 DE SILVA G. B. P

GROUP NUMBER : GROUP 16

DATABASE SCHEMA : EVENT PLANANING MANAGEMENT SYSTEM

DEPARTMENT : COMPUTER ENGINEERING

Abstract

The Event Planning Management System database schema provides as the backbone for organizing and managing various aspects of events efferently. The schema includes necessary entities, relationships and attributes for smooth event coordination, meeting the different needs of customers and stakeholders.

The schema compares Entities in the database which are crucial components for event planning and execution. Relationships in between the entities are established to maintain data integrity and facilitate smooth interaction among the different elements in the database system. Attributes relevant with each entity gather necessary information for event management system such as details of the event, payment details and employee details. Then the attributes are carefully structured to ensure the accurate representation of data to facilitate efficient querying.

Normalization techniques are used to minimize redundancy and improve data consistency with in the schema, for following industry standards and best practices in database design. Primary keys, foreign keys and indexes are strategically used to enhance data retrieval speed and maintain referential integrity throughout the database.

By creating an Event Planning Management System, it provides a comprehensive framework for organizing, storing and accessing event-related information to supply efficient event planning experience.

Preface

Event Planning Management System is a groundbreaking solution designed to success events in the effective management of data through relational database. This system is not just a tool, it's surely a partner in your journey towards creating unforgettable and impactful events.

Further, we would like to encourage you to explore the capabilities of the Event Planning Management System, discover its transformative impact on event planning and management, and embark on a journey towards enhanced efficiency, effectiveness, and excellence in executing events.

Contents

Chapter 1 – Requirement Analysis	7
1.1 Functional requirements	7
1.2 Data Requirements	8
2 Chapter 2 – Conceptual design	11
3 Chapter 3 – Implementation	12
3.1 Schema Creation	12
3.2 Table Creation	13
3.3 Relationships making between tables	24
3.4 Insert Values into Tables	26
3.5 Update and delete values in Tables	31
4 Chapter 4 – Transactions	35
4.1 Simple Query	35
4.2 Complex Query	39
5 Chapter 5 - Database Tuning	46

List of Figures

Figure 2. 1: ER Diagram	11
Figure 3.1. 1: Event_Planning_Management_System Schema creation.....	12
Figure 3.2. 1: Events_TB Table creation	13
Figure 3.2. 2: Event_Coordinator Table creation.....	13
Figure 3.2. 3: Event_Coordinator_Contact_Number Table creation	14
Figure 3.2. 4: Employee Table creation	14
Figure 3.2. 5: Empl_ID_Contact_Number Table creation	15
Figure 3.2. 6: Music Table creation	15
Figure 3.2. 7: Music_Contact_Number Table creation.....	16
Figure 3.2. 8: Hotel Table creation	16
Figure 3.2. 9: Hotel_Contact_Number Table creation.....	17
Figure 3.2. 10: Transportation Table creation.....	17
Figure 3.2. 11: Transportation_Contact_Number Table creation	18
Figure 3.2. 12: Decoration Table creation.....	18
Figure 3.2. 13: Photography Table creation	19
Figure 3.2. 14: Photography_Contact_Number Table creation	19
Figure 3.2. 15: Catering Table creation.....	20
Figure 3.2. 16: Catering_Contact_Number Table creation	20
Figure 3.2. 17: Booking Table creation.....	21
Figure 3.2. 18: Clients Table creation	21
Figure 3.2. 19: Clients_Contact_Number Table creation	22
Figure 3.2. 20: Feedback Table creation	22
Figure 3.2. 21: Rating Table creation.....	23
Figure 3.4. 1: Insert values into Event_Coordinator and Event_Coordinator_Contact_Number tables	26
Figure 3.4. 2: Insert values into Employee and Empl_ID_Contact_Number tables	26
Figure 3.4. 3: Insert values into Music and Music_Contact_Number tables	27
Figure 3.4. 4: Insert values into Hotel and Hotel_Contact_Number tables	27
Figure 3.4. 5: Insert values into transportation and Transportation_Contact_Number tables	28
Figure 3.4. 6: Insert values into Decoration, Photography and Photography_Contact_Number tables	28
Figure 3.4. 7: Insert values into Catering and Catering_Contact_Number tables.....	29
Figure 3.4. 8: Insert values into Clients and Clients_Contact_Number tables	29
Figure 3.4. 9: Insert values into Booking and Feedback tables.....	30
Figure 3.4. 10: Insert values into Events_TB and Rating tables	30
Figure 3.5. 1: Update and delete in Rating, Events_TB and Event_Coordinator tables	31
Figure 3.5. 2: Update and delete in Event_Coordinator_Contact_Number, Employee and Empl_ID_Contact_Number tables	31
Figure 3.5. 3: Update and delete in Music, Music_Contact_Number and Hotel tables.....	32

Figure 3.5. 4: Update and delete in Hotel_Contact_Number, transportation and Transportation_Contact_Number tables	32
Figure 3.5. 5: Update and delete in Decoration, Photography and Photography_Contact_Number tables.....	33
Figure 3.5. 6: Update and delete in Catering, Catering_Contact_Number and Booking tables	33
Figure 3.5. 7: Update and delete in Clients, Clients_Contact_Number and Feedback tables.....	34
Figure 4.1. 1: Select Operation-Simple Query	35
Figure 4.1. 2: Project Operation-Simple Query	35
Figure 4.1. 3: Cartesian Product Operation-Simple Query	36
Figure 4.1. 4: Creating a user view-Simple Query.....	36
Figure 4.1. 5: Renaming Operation-Simple Query	37
Figure 4.1. 6: Use of an aggregation function-Simple Query	37
Figure 4.1. 7: Use of LIKE keyword-Simple Query	38
Figure 4.2. 1: Aliasing with union Operation-Complex Query.....	39
Figure 4.2. 2: Aliasing with intersection Operation-Complex Query	39
Figure 4.2. 3: Aliasing with set difference Operation-Complex Query	40
Figure 4.2. 4: Aliasing with division Operation-Complex Query.....	40
Figure 4.2. 5: Aliasing with inner join Operation-Complex Query	41
Figure 4.2. 6: Aliasing with natural join Operation-Complex Query	41
Figure 4.2. 7: Aliasing with left outer join Operation-Complex Query	42
Figure 4.2. 8: Aliasing with right outer join Operation-Complex Query.....	42
Figure 4.2. 9: Aliasing with full outer join Operation-Complex Query.....	43
Figure 4.2. 10: Aliasing with outer union Operation-Complex Query	43
Figure 4.2. 11: Nested-Complex Query	44
Figure 4.2. 12: Nested-Complex Query	44
Figure 4.2. 13: Nested-Complex Query	45
Figure 5. 1: Before Tuning.....	46
Figure 5. 2: After Tuning	46
Figure 5. 3: Before Tuning.....	47
Figure 5. 4: No more Tuning.....	47
Figure 5. 5: Before Tuning.....	48
Figure 5. 6: After Tuning	48
Figure 5. 7: Before Tuning.....	49
Figure 5. 8: Before Tuning.....	49
Figure 5. 9: Before Tuning.....	50
Figure 5. 10: After Tuning	50
Figure 5. 11: Before Tuning.....	51
Figure 5. 12: After Tuning	51
Figure 5. 13: Before Tuning.....	52
Figure 5. 14: After Tuning	52
Figure 5. 15: Before Tuning.....	53
Figure 5. 16: After Tuning	53

Chapter 1 – Requirement Analysis

1.1 Functional requirements

Functional requirements refer to describe the capabilities of the system can be performed. Functional requirements may include specifications with relevant to the types of data that need to be stored in the database. These details ensure that the database system can efficiently storing, retrieving, updating and deleting to maintain data integrity and consistency.

In the Event Planning Management System, first recognized and obtained the necessary entities, attributes and relationships relevant to the database system with the help of conceptual database model. Then through the Normalization process, the conceptual representation is transformed into the logical database structure. Subsequently, the logical database is physically established within the database management system utilizing MySQL Workbench.

- Client can consolidate all the event arrangements, including hotel selection, decor, catering, transportation, music and photography with a single, integrated event planer to facilitate a seamless and elegant event experience.
- Event coordinators can assign different tasks to employees, empowering them to contribute to various aspects of event execution and ensuring a harmonious and efficient event experience.
- Client has the option to make a single payment for the total event cost. And also, it is easier to identify regular clients and offering discounts to avoid unnecessary expenses.
- Clients are provided with the opportunity to provide feedback to upgrade the experience of using event planning service.
- Client can monitor all the details and the progress of the event from a centralized platform.
- Event coordinators can identify available employees and allocate them to upcoming events as needed.
- Booking includes comprehensive details of the event.
- The system allows for database editing as necessary, facilitating the addition of employees, event details, client information, and other relevant data as needed.

1.2 Data Requirements

In this Event Planning Management System includes few strong Entities and one weak entity and attributes

Entity : Employee

Attributes : Employee Name

Employee ID (Not Null)

Contact Number

Birthday

Entity : Event Coordinator

Attributes : Coordinator Name

Coordinator ID (Not Null)

Employee ID (Not Null)

Contact Number

Birthday

Entity : Event

Attributes : Event Name

Event ID (Not Null)

Date

Time

Number of Participants

Entity : Music

Attributes : Band Name

Band ID (Not Null)

Style of music

Contact Number

Charge per Event

Entity : Hotel

Attributes : Hotel Name

 Hotel Register Number (Not Null)

 Address

 Contact Number

 Charge per Person

Entity : Transportation

Attributes : Vehicle Number (Not Null)

 Types of Vehicle

 Contact Number

 Charge per 1km

Entity : Decoration

Attributes : Decoration ID (Not Null)

 Contact Number

Entity : Photography

Attributes : Photography Name

 Photography ID (Not Null)

 Contact Number

 Charge per Event

Entity : Catering

Attributes : Catering Name

 Catering ID (Not Null)

 Contact Number

 Charge per Person

Entity : Booking

Attributes : Client ID (Not Null)

Event ID (Not Null)

Booking Date

Payment

Entity : Client

Attributes : Client Name

Client ID (Not Null)

Address

Contact Number

Entity : Feedback

Attributes : Client ID (Not Null)

Feedback View

Entity : Rating

Attributes : Rating View

2 Chapter 2 – Conceptual design

The ER- diagram for the Event Planning Management System is drawn in below.

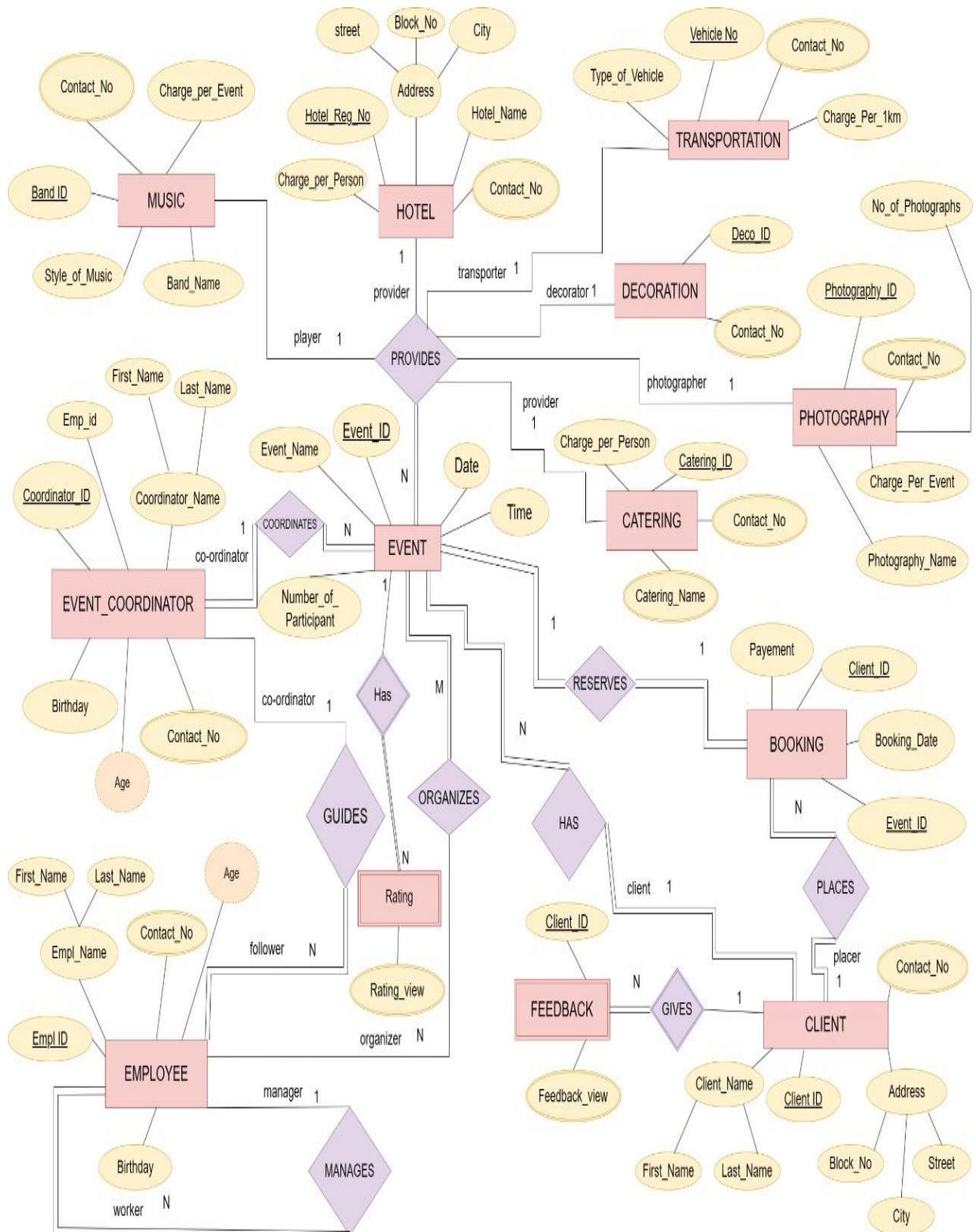


Figure 2. 1: ER Diagram

3 Chapter 3 – Implementation

3.1 Schema Creation

Use MySQL Workbench for implementation of the 2NF of the representational data model. All the schema creation, table definitions, Insert, Update, Delete Operations with screen shots are included in this chapter. And also, all the tables with 6 rows of data, updated 2 rows, deleted 1 row are included.

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

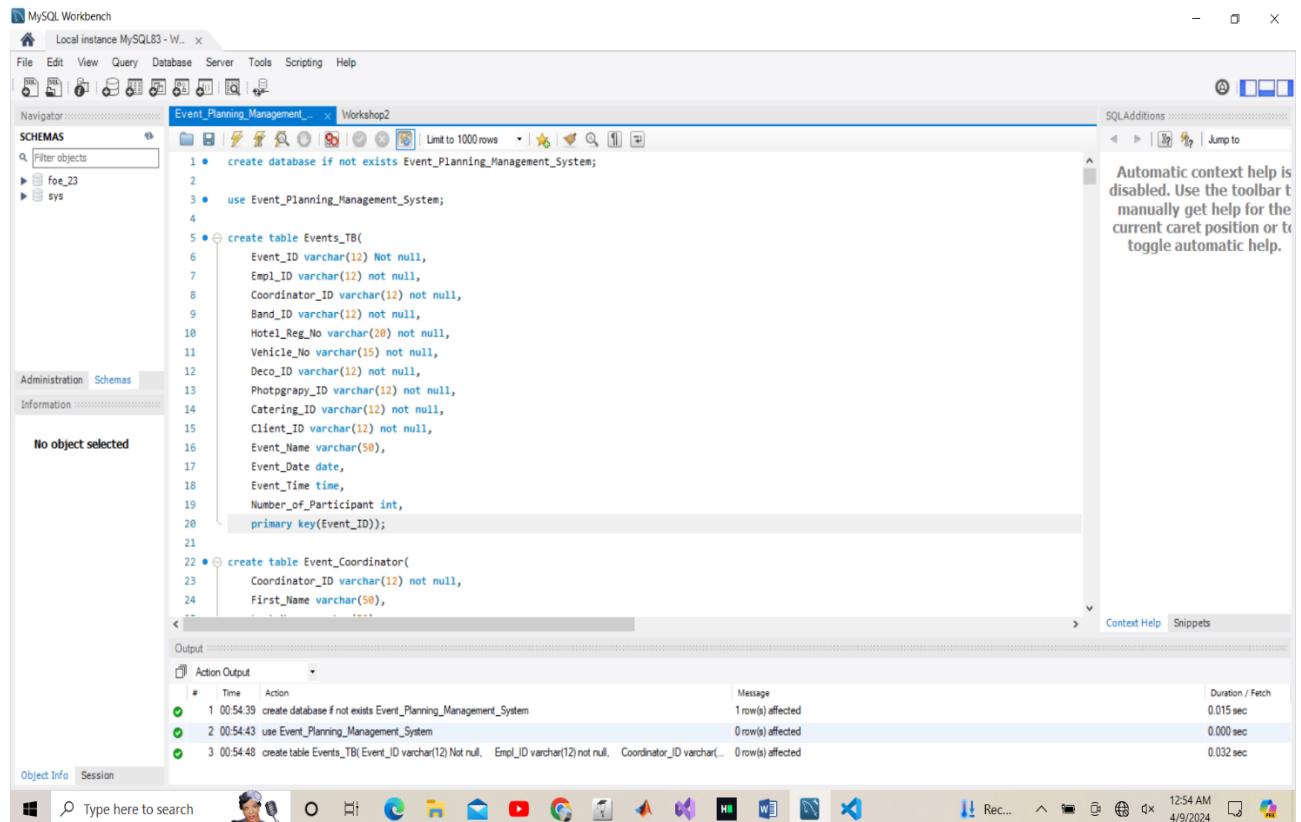
- Title Bar:** MySQL Workbench - Local instance MySQL83 - Workshop2
- File Menu:** File, Edt, View, Query, Database, Server, Tools, Scripting, Help
- Toolbar:** Includes icons for New, Open, Save, Print, Copy, Paste, Find, Replace, and others.
- Navigator:** Shows the schema "Event_Planning_Management_System" and objects "foe_23" and "sys".
- SQL Editor:** Displays the SQL code for creating the database and its tables:

```
1 • create database if not exists Event_Planning_Management_System;
2
3 • use Event_Planning_Management_System;
4
5 • create table Events_TB(
6     Event_ID varchar(12) Not null,
7     Emp1_ID varchar(12) not null,
8     Coordinator_ID varchar(12) not null,
9     Band_ID varchar(12) not null,
10    Hotel_Reg_No varchar(20) not null,
11    Vehicle_No varchar(15) not null,
12    Deco_ID varchar(12) not null,
13    Photography_ID varchar(12) not null,
14    Catering_ID varchar(12) not null,
15    Client_ID varchar(12) not null,
16    Event_Name varchar(50),
17    Event_Date date,
18    Event_Time time,
19    Number_of_Participant int,
20    primary key(Event_ID));
21
22 • create table Event_Coordinator(
23     Coordinator_ID varchar(12) not null,
24     First_Name varchar(50),
```
- Output Panel:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	22:46:32	create database if not exists Event_Planning_Management_System	1 row(s) affected	0.000 sec
- Object Info and Session Panels:** Standard MySQL Workbench panels for object information and session monitoring.
- System Tray:** Shows the Windows taskbar with various pinned icons and system status.

Figure 3.1. 1: Event_Planning_Management_System Schema creation

3.2 Table Creation



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

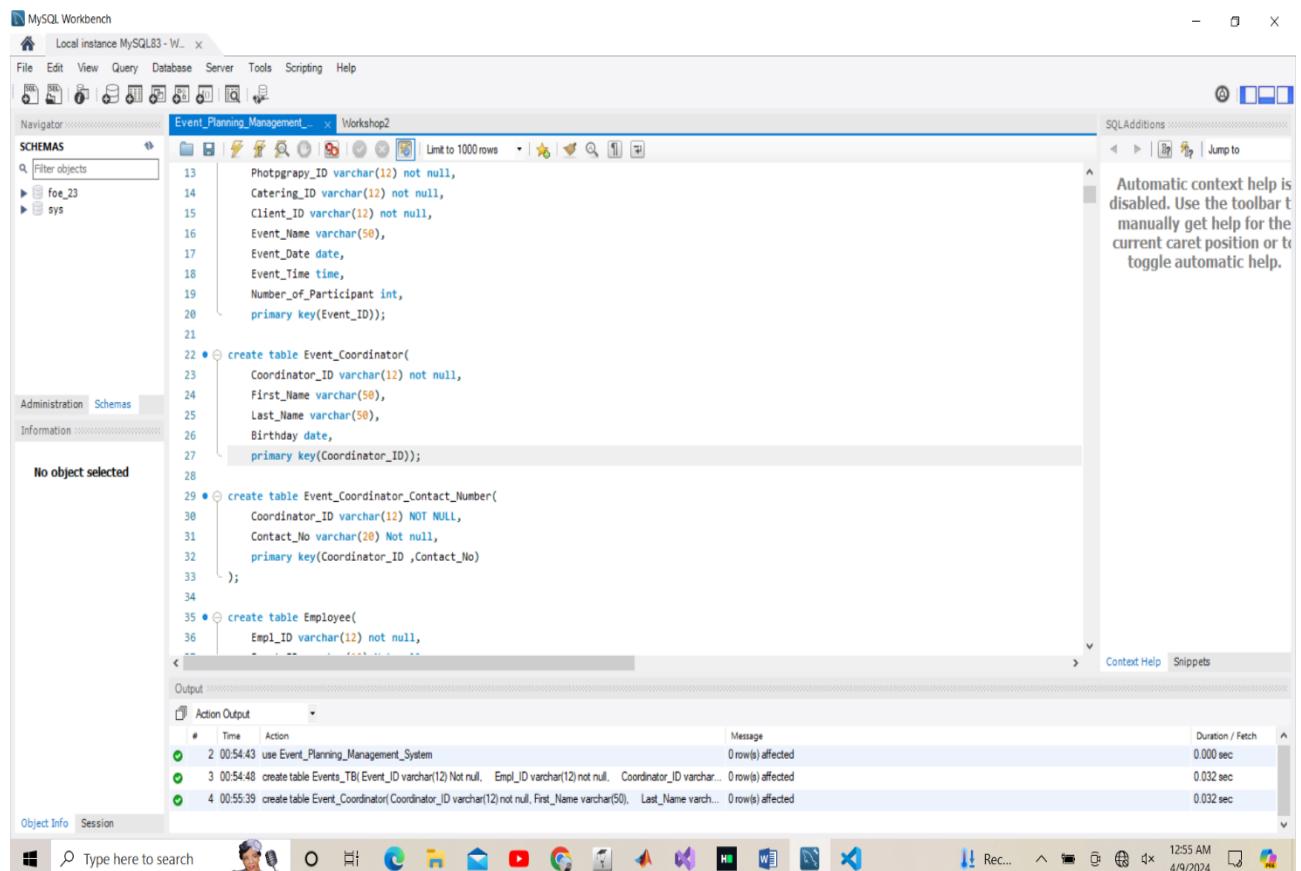
- Title Bar:** Local instance MySQL83 - Workshop2
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Toolbar:** Standard MySQL icons for database management.
- Navigator:** Schemas (Event_Planning_Management_, foe_23, sys). No object selected.
- SQL Editor:** Workshop2 tab, SQL pane showing the creation of the Events_TB table. The code is as follows:


```

1 • create database if not exists Event_Planning_Management_System;
2
3 • use Event_Planning_Management_System;
4
5 •   create table Events_TB(
6       Event_ID varchar(12) Not null,
7       Emp_ID varchar(12) not null,
8       Coordinator_ID varchar(12) not null,
9       Band_ID varchar(12) not null,
10      Hotel_Reg_No varchar(20) not null,
11      Vehicle_No varchar(15) not null,
12      Deco_ID varchar(12) not null,
13      Photograpy_ID varchar(12) not null,
14      Catering_ID varchar(12) not null,
15      Client_ID varchar(12) not null,
16      Event_Name varchar(50),
17      Event_Date date,
18      Event_Time time,
19      Number_of_Participant int,
20      primary key(Event_ID));
21
22 •   create table Event_Coordinator(
23       Coordinator_ID varchar(12) not null,
24       First_Name varchar(50),
25       Last_Name varchar(50),
26       Birthday date,
27       primary key(Coordinator_ID));
28
29 •   create table Event_Coordinator_Contact_Number(
30       Coordinator_ID varchar(12) NOT NULL,
31       Contact_No varchar(20) Not null,
32       primary key(Coordinator_ID ,Contact_No));
33
34 •   create table Employee(
35       Emp_ID varchar(12) not null,
36       ...
      
```
- Output:** Action Output pane showing the execution results:

#	Time	Action	Message	Duration / Fetch
1	00:54:39	create database if not exists Event_Planning_Management_System	1 row(s) affected	0.015 sec
2	00:54:43	use Event_Planning_Management_System	0 row(s) affected	0.000 sec
3	00:54:48	create table Events_TB(Event_ID varchar(12) Not null, Emp_ID varchar(12) not null, Coordinator_ID varchar(12) not null, Band_ID varchar(12) not null, Hotel_Reg_No varchar(20) not null, Vehicle_No varchar(15) not null, Deco_ID varchar(12) not null, Photograpy_ID varchar(12) not null, Catering_ID varchar(12) not null, Client_ID varchar(12) not null, Event_Name varchar(50), Event_Date date, Event_Time time, Number_of_Participant int, primary key(Event_ID));	0 row(s) affected	0.032 sec
- Object Info:** Session tab.
- System Bar:** Type here to search, Taskbar icons, Date/Time (12:54 AM, 4/9/2024).

Figure 3.2. 1: Events_TB Table creation



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

- Title Bar:** Local instance MySQL83 - Workshop2
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Toolbar:** Standard MySQL icons for database management.
- Navigator:** Schemas (Event_Planning_Management_, foe_23, sys). No object selected.
- SQL Editor:** Workshop2 tab, SQL pane showing the creation of the Event_Coordinator table. The code is as follows:


```

13
14
15
16
17
18
19
20
21
22 •   create table Event_Coordinator(
23       Coordinator_ID varchar(12) not null,
24       First_Name varchar(50),
25       Last_Name varchar(50),
26       Birthday date,
27       primary key(Coordinator_ID));
28
29 •   create table Event_Coordinator_Contact_Number(
30       Coordinator_ID varchar(12) NOT NULL,
31       Contact_No varchar(20) Not null,
32       primary key(Coordinator_ID ,Contact_No));
33
34 •   create table Employee(
35       Emp_ID varchar(12) not null,
36       ...
      
```
- Output:** Action Output pane showing the execution results:

#	Time	Action	Message	Duration / Fetch
2	00:54:43	use Event_Planning_Management_System	0 row(s) affected	0.000 sec
3	00:54:48	create table Events_TB(Event_ID varchar(12) Not null, Emp_ID varchar(12) not null, Coordinator_ID varchar(12) not null, Band_ID varchar(12) not null, Hotel_Reg_No varchar(20) not null, Vehicle_No varchar(15) not null, Deco_ID varchar(12) not null, Photograpy_ID varchar(12) not null, Catering_ID varchar(12) not null, Client_ID varchar(12) not null, Event_Name varchar(50), Event_Date date, Event_Time time, Number_of_Participant int, primary key(Event_ID));	0 row(s) affected	0.032 sec
4	00:55:39	create table Event_Coordinator(Coordinator_ID varchar(12) not null, First_Name varchar(50), Last_Name varchar(50), Birthday date, primary key(Coordinator_ID));	0 row(s) affected	0.032 sec
- Object Info:** Session tab.
- System Bar:** Type here to search, Taskbar icons, Date/Time (12:55 AM, 4/9/2024).

Figure 3.2. 2: Event_Coordinator Table creation

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

- Title Bar:** MySQL Workbench - Local instance MySQL83 - Workshop2
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Navigator:** Schemas (foe_23, sys)
- SQL Editor:** The code pane displays the SQL script for creating the Event_Coordinator_Contact_Number table. The table has columns Coordinator_ID (primary key), Contact_No (primary key), and Last_Name.
- Output:** The Action Output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
1	00:54:39	create database if not exists Event_Planning_Management_System	1 row(s) affected	0.015 sec
2	00:54:43	use Event_Planning_Management_System	0 row(s) affected	0.000 sec
3	00:54:48	create table Events_TB(Event_ID varchar(12) Not null, Empl_ID varchar(12) not null, Coordinator_ID varchar(12) not null, First_Name varchar(50), Last_Name varchar(50), Birthday date, primary key(Coordinator_ID))	0 row(s) affected	0.032 sec
4	00:55:39	create table Event_Coordinator(Coordinator_ID varchar(12) not null, First_Name varchar(50), Last_Name varchar(50), Birthday date, primary key(Coordinator_ID))	0 row(s) affected	0.032 sec
5	00:56:55	create table Event_Coordinator_Contact_Number(Coordinator_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Coordinator_ID ,Contact_No))	0 row(s) affected	0.032 sec
- Object Info:** Session
- Taskbar:** Shows various pinned application icons.

Figure 3.2. 3: Event_Coordinator_Contact_Number Table creation

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

- Title Bar:** MySQL Workbench - Local instance MySQL83 - Workshop2
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Navigator:** Schemas (foe_23, sys)
- SQL Editor:** The code pane displays the SQL script for creating the Employee table. The table has columns Empl_ID (primary key), Event_ID, Coordinator_ID, First_Name, Last_Name, and Birthday.
- Output:** The Action Output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
1	00:54:39	create database if not exists Event_Planning_Management_System	1 row(s) affected	0.015 sec
2	00:54:43	use Event_Planning_Management_System	0 row(s) affected	0.000 sec
3	00:54:48	create table Events_TB(Event_ID varchar(12) Not null, Empl_ID varchar(12) not null, Coordinator_ID varchar(12) not null, First_Name varchar(50), Last_Name varchar(50), Birthday date, primary key(Coordinator_ID))	0 row(s) affected	0.032 sec
4	00:55:39	create table Event_Coordinator(Coordinator_ID varchar(12) not null, First_Name varchar(50), Last_Name varchar(50), Birthday date, primary key(Coordinator_ID))	0 row(s) affected	0.032 sec
5	00:56:55	create table Event_Coordinator_Contact_Number(Coordinator_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Coordinator_ID ,Contact_No))	0 row(s) affected	0.032 sec
6	00:58:05	create table Employee(Empl_ID varchar(12) not null, Event_ID varchar(12) Not null, Coordinator_ID varchar(12) NOT NULL, First_Name varchar(50), Last_Name varchar(50), Birthday date, primary key(Empl_ID))	0 row(s) affected	0.031 sec
- Object Info:** Session
- Taskbar:** Shows various pinned application icons.

Figure 3.2. 4: Employee Table creation

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

- Title Bar:** MySQL Workbench - Local instance MySQL8.0 - Workshop2
- Toolbar:** Standard MySQL Workbench toolbar.
- Navigator:** Shows the schema 'Event_Planning_Management' and objects 'foe_23' and 'sys'.
- SQL Editor:** Displays the SQL code for creating the 'Empl_ID_Contact_Number' table:


```

34 • create table Employee(
35     Emp1_ID varchar(12) not null,
36     Event_ID varchar(12) Not null,
37     Coordinator_ID varchar(12) not null,
38     First_Name varchar(50),
39     Last_Name varchar(50),
40     Birthday date ,
41     primary key(Emp1_ID));
42
43 • create table Empl_ID_Contact_Number(
44     Emp1_ID varchar(12) NOT NULL,
45     Contact_No varchar(20) Not null,
46     primary key(Empl1_ID ,Contact_No));
47
48 • create table Music(
49     Band_ID varchar(12) not null,
50     Band_Name varchar(50),
51     Style_of_Music varchar(75) ,
52     Charge_of_Event double ,
53     primary key(Band_ID));
      
```
- Output:** Shows the execution log with 7 actions, all successful, indicating the creation of the database and tables:

#	Time	Action	Message	Duration / Fetch
1	00:54:39	create database if not exists Event_Planning_Management_System	1 row(s) affected	0.015 sec
2	00:54:43	use Event_Planning_Management_System	0 row(s) affected	0.000 sec
3	00:54:48	create table Events_TB(Event_ID varchar(12) not null, Emp1_ID varchar(12) not null, Coordinator_ID varchar(12) not null, First_Name varchar(50), Last_Name varchar(50), Birthday date, primary key(Event_ID))	0 row(s) affected	0.032 sec
4	00:55:39	create table Event_Coordinator(Coordinator_ID varchar(12) not null, First_Name varchar(50), Last_Name varchar(50), Birthdate date, primary key(Coordinator_ID))	0 row(s) affected	0.032 sec
5	00:56:55	create table Event_Coordinator_Contact_Number(Coordinator_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Coordinator_ID ,Contact_No))	0 row(s) affected	0.032 sec
6	00:58:05	create table Employee(Emp1_ID varchar(12) not null, Event_ID varchar(12) Not null, Coordinator_ID varchar(12) Not null, Contact_No varchar(20) Not null, primary key(Emp1_ID))	0 row(s) affected	0.031 sec
7	00:58:37	create table Empl_ID_Contact_Number(Empl1_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Empl1_ID ,Contact_No))	0 row(s) affected	0.031 sec
- System Bar:** Shows the system tray with icons for battery, network, volume, and clock (12:58 AM, 4/9/2024).

Figure 3.2. 5: Empl_ID_Contact_Number Table creation

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

- Title Bar:** MySQL Workbench - Local instance MySQL8.0 - Workshop2
- Toolbar:** Standard MySQL Workbench toolbar.
- Navigator:** Shows the schema 'Event_Planning_Management' and objects 'foe_23' and 'sys'.
- SQL Editor:** Displays the SQL code for creating the 'Music' table:


```

43 • create table Empl_ID_Contact_Number(
44     Emp1_ID varchar(12) NOT NULL,
45     Contact_No varchar(20) Not null,
46     primary key(Empl1_ID ,Contact_No));
47
48 • create table Music(
49     Band_ID varchar(12) not null,
50     Band_Name varchar(50),
51     Style_of_Music varchar(75) ,
52     Charge_of_Event double ,
53     primary key(Band_ID));
54
55 • create table Music_Contact_Number(
56     Band_ID varchar(12) NOT NULL,
57     Contact_No varchar(20) Not null,
58     primary key(Band_ID ,Contact_No));
59
60 • create table Hotel(
61     Hotel_Reg_No varchar(30) not null,
62     Hotel_Name varchar(30),
      
```
- Output:** Shows the execution log with 8 actions, all successful, indicating the creation of the database and tables:

#	Time	Action	Message	Duration / Fetch
2	00:54:43	use Event_Planning_Management_System	0 row(s) affected	0.000 sec
3	00:54:48	create table Events_TB(Event_ID varchar(12) not null, Emp1_ID varchar(12) not null, Coordinator_ID varchar(12) not null, First_Name varchar(50), Last_Name varchar(50), Birthday date, primary key(Event_ID))	0 row(s) affected	0.032 sec
4	00:55:39	create table Event_Coordinator(Coordinator_ID varchar(12) not null, First_Name varchar(50), Last_Name varchar(50), Birthdate date, primary key(Coordinator_ID))	0 row(s) affected	0.032 sec
5	00:56:55	create table Event_Coordinator_Contact_Number(Coordinator_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Coordinator_ID ,Contact_No))	0 row(s) affected	0.032 sec
6	00:58:05	create table Employee(Emp1_ID varchar(12) not null, Event_ID varchar(12) Not null, Coordinator_ID varchar(12) Not null, Contact_No varchar(20) Not null, primary key(Emp1_ID))	0 row(s) affected	0.031 sec
7	00:58:37	create table Empl_ID_Contact_Number(Empl1_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Empl1_ID ,Contact_No))	0 row(s) affected	0.031 sec
8	00:59:29	create table Music(Band_ID varchar(12) not null, Band_Name varchar(50), Style_of_Music varchar(75) , Charge_Event double , primary key(Band_ID))	0 row(s) affected	0.032 sec
- System Bar:** Shows the system tray with icons for battery, network, volume, and clock (12:59 AM, 4/9/2024).

Figure 3.2. 6: Music Table creation

The screenshot shows the MySQL Workbench interface with the 'Workshop2' tab selected. In the central SQL editor pane, the following SQL code is displayed:

```

46     primary key(Emp1_ID ,Contact_No));
47
48 • Ⓜ create table Music(
49     Band_ID varchar(12) not null,
50     Band_Name varchar(50),
51     Style_of_Music varchar(75),
52     Charge_of_Event double,
53     primary key(Band_ID));
54
55 • Ⓜ create table Music_Contact_Number(
56     Band_ID varchar(12) NOT NULL,
57     Contact_No varchar(20) Not null,
58     primary key(Band_ID ,Contact_No));
59
60 • Ⓜ create table Hotel(
61     Hotel_Reg_No varchar(20) not null,
62     Hotel_Name varchar(50),
63     Block_No varchar(15),
64     City varchar(50),
65     Street varchar(30),
66     Charge_per_Person double,
67     primary key (Hotel_Reg_No));
68
69 • Ⓜ create table Hotel_Contact_Number(
70     Hotel_Reg_No varchar(12) NOT NULL,
71     Contact_No varchar(20) Not null,
72     primary key(Hotel_Reg_No ,Contact_No));

```

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

#	Time	Action	Message	Duration / Fetch
3	00:54:48	create table Events_TB(Event_ID varchar(12) Not null, Emp1_ID varchar(12) not null, Coordinator_ID varchar(12) Not null, First_Name varchar(50), Last_Name varchar(50), Contact_No varchar(20) Not null, primary key(Event_ID ,Contact_No))	0 rows(s) affected	0.032 sec
4	00:55:39	create table Event_Coordinator(Coordinator_ID varchar(12) not null, First_Name varchar(50), Last_Name varchar(50), Contact_No varchar(20) Not null, primary key(Coordinator_ID ,Contact_No))	0 rows(s) affected	0.032 sec
5	00:56:55	create table Event_Coordinator_Contact_Number(Coordinator_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Coordinator_ID ,Contact_No))	0 rows(s) affected	0.032 sec
6	00:58:05	create table Employee(Emp1_ID varchar(12) not null, Event_ID varchar(12) Not null, Coordinator_ID varchar(12) Not null, primary key(Emp1_ID ,Event_ID))	0 rows(s) affected	0.031 sec
7	00:58:37	create table Empl_ID_Contact_Number(Empl_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Empl_ID ,Contact_No))	0 rows(s) affected	0.031 sec
8	00:59:29	create table Music(Band_ID varchar(12) not null, Band_Name varchar(50), Style_of_Music varchar(75), Charge_Event double, primary key(Band_ID))	0 rows(s) affected	0.032 sec
9	00:59:59	create table Music_Contact_Number(Band_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Band_ID ,Contact_No))	0 rows(s) affected	0.031 sec

Figure 3.2. 7: Music_Contact_Number Table creation

The screenshot shows the MySQL Workbench interface with the 'Workshop2' tab selected. In the central SQL editor pane, the following SQL code is displayed:

```

52     primary key(Band_ID));
53
54
55 • Ⓜ create table Music_Contact_Number(
56     Band_ID varchar(12) NOT NULL,
57     Contact_No varchar(20) Not null,
58     primary key(Band_ID ,Contact_No));
59
60 • Ⓜ create table Hotel(
61     Hotel_Reg_No varchar(20) not null,
62     Hotel_Name varchar(50),
63     Block_No varchar(15),
64     City varchar(50),
65     Street varchar(30),
66     Charge_per_Person double,
67     primary key (Hotel_Reg_No));
68
69 • Ⓜ create table Hotel_Contact_Number(
70     Hotel_Reg_No varchar(12) NOT NULL,
71     Contact_No varchar(20) Not null,
72     primary key(Hotel_Reg_No ,Contact_No));

```

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

#	Time	Action	Message	Duration / Fetch
4	00:55:39	create table Event_Coordinator(Coordinator_ID varchar(12) not null, First_Name varchar(50), Last_Name varchar(50), Contact_No varchar(20) Not null, primary key(Coordinator_ID ,Contact_No))	0 rows(s) affected	0.032 sec
5	00:56:55	create table Event_Coordinator_Contact_Number(Coordinator_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Coordinator_ID ,Contact_No))	0 rows(s) affected	0.032 sec
6	00:58:05	create table Employee(Emp1_ID varchar(12) not null, Event_ID varchar(12) Not null, Coordinator_ID varchar(12) Not null, primary key(Emp1_ID ,Event_ID))	0 rows(s) affected	0.031 sec
7	00:58:37	create table Empl_ID_Contact_Number(Empl_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Empl_ID ,Contact_No))	0 rows(s) affected	0.031 sec
8	00:59:29	create table Music(Band_ID varchar(12) not null, Band_Name varchar(50), Style_of_Music varchar(75), Charge_Event double, primary key(Band_ID))	0 rows(s) affected	0.032 sec
9	00:59:59	create table Music_Contact_Number(Band_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Band_ID ,Contact_No))	0 rows(s) affected	0.031 sec
10	01:00:39	create table Hotel(Hotel_Reg_No varchar(20) not null, Hotel_Name varchar(30), Block_No varchar(15), City varchar(50), Street varchar(30), Charge_per_Person double, primary key(Hotel_Reg_No))	0 rows(s) affected	0.031 sec

Figure 3.2. 8: Hotel Table creation

The screenshot shows the MySQL Workbench interface with the 'Event_Planning_Management' schema selected. In the central query editor, the following SQL code is displayed:

```

61     Hotel_Reg_No varchar(20) not null,
62     Hotel_Name varchar(30),
63     Block_No varchar(15),
64     City varchar(50),
65     Street varchar(30),
66     Charge_per_Person double,
67     primary key (Hotel_Reg_No));
68
69 • Ⓜ create table Hotel_Contact_Number(
70     Hotel_Reg_No varchar(12) NOT NULL,
71     Contact_No varchar(20) Not null,
72     primary key(Hotel_Reg_No ,Contact_No));
73
74 • Ⓜ create table Transportation(
75     Vehicle_No varchar(15) not null,
76     Type_of_Vehicle varchar(20) ,
77     charge_per_1km double,
78     primary key(Vehicle_No));
79 • Ⓜ create table Transportation_Contact_Number(
80     Vehicle_No varchar(15) NOT NULL,

```

The 'Output' pane below shows the execution log:

#	Time	Action	Message	Duration / Fetch
5	00:56:55	create table Event_Cordinator_Contact_Number(Cordinator_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null)	0 rows affected	0.032 sec
6	00:58:05	create table Employee(Empl_ID varchar(12) not null, Event_ID varchar(12) Not null, Coordinator_ID varchar(12) Not null)	0 rows affected	0.031 sec
7	00:58:37	create table Empl_ID_Contact_Number(Empl_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, ...)	0 rows affected	0.031 sec
8	00:59:29	create table Music_Band(Band_ID varchar(12) not null, Band_Name varchar(50), Style_of_Music varchar(75), Charge_per_1km double)	0 rows affected	0.032 sec
9	00:59:59	create table Music_Contact_Number(Band_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Band_ID ,Contact_No))	0 rows affected	0.031 sec
10	01:00:39	create table Hotel(Hotel_Reg_No varchar(20) not null, Hotel_Name varchar(30), Block_No varchar(15), City varchar(50), Street varchar(30), Charge_per_Person double)	0 rows affected	0.031 sec
11	01:01:10	create table Hotel_Contact_Number(Hotel_Reg_No varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Hotel_Reg_No ,Contact_No))	0 rows affected	0.031 sec

Figure 3.2. 9: Hotel_Contact_Number Table creation

The screenshot shows the MySQL Workbench interface with the 'Event_Planning_Management' schema selected. In the central query editor, the following SQL code is displayed:

```

67     primary key (Hotel_Reg_No));
68
69 • Ⓜ create table Hotel_Contact_Number(
70     Hotel_Reg_No varchar(12) NOT NULL,
71     Contact_No varchar(20) Not null,
72     primary key(Hotel_Reg_No ,Contact_No));
73
74 • Ⓜ create table Transportation(
75     Vehicle_No varchar(15) not null,
76     Type_of_Vehicle varchar(20) ,
77     charge_per_1km double,
78     primary key(Vehicle_No));
79
80 • Ⓜ create table Transportation_Contact_Number(
81     Vehicle_No varchar(15) NOT NULL,
82     Contact_No varchar(20) Not null,
83     primary key(Vehicle_No ,Contact_No));
84
85 • Ⓜ create table Decoration(
86     Deco_ID varchar(12) not null,

```

The 'Output' pane below shows the execution log:

#	Time	Action	Message	Duration / Fetch
6	00:58:05	create table Employee(Empl_ID varchar(12) not null, Event_ID varchar(12) Not null, Coordinator_ID varchar(12) Not null)	0 rows affected	0.031 sec
7	00:58:37	create table Empl_ID_Contact_Number(Empl_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, ...)	0 rows affected	0.031 sec
8	00:59:29	create table Music_Band(Band_ID varchar(12) not null, Band_Name varchar(50), Style_of_Music varchar(75), Charge_per_1km double)	0 rows affected	0.032 sec
9	00:59:59	create table Music_Contact_Number(Band_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Band_ID ,Contact_No))	0 rows affected	0.031 sec
10	01:00:39	create table Hotel(Hotel_Reg_No varchar(20) not null, Hotel_Name varchar(30), Block_No varchar(15), City varchar(50), Street varchar(30), Charge_per_Person double)	0 rows affected	0.031 sec
11	01:01:10	create table Hotel_Contact_Number(Hotel_Reg_No varchar(12) NOT NULL, Contact_No varchar(20) Not null, primary key(Hotel_Reg_No ,Contact_No))	0 rows affected	0.031 sec
12	01:01:47	create table Transportation(Vehicle_No varchar(15) not null, Type_of_Vehicle varchar(20), charge_per_1km double)	0 rows affected	0.016 sec

Figure 3.2. 10: Transportation Table creation

The screenshot shows the MySQL Workbench interface with the 'Event_Planning_Management' schema selected. In the central query editor, the following SQL code is being run:

```

67     primary key (Hotel_Reg_No);
68
69 • create table Hotel_Contact_Number(
70     Hotel_Reg_No varchar(12) NOT NULL,
71     Contact_No varchar(20) Not null,
72     primary key(Hotel_Reg_No ,Contact_No));
73
74 • create table Transportation(
75     Vehicle_No varchar(15) not null,
76     Type_of_Vehicle varchar(20) ,
77     charge_per_1km double,
78     primary key(Vehicle_No));
79
80 • create table Transportation_Contact_Number(
81     Vehicle_No varchar(15) NOT NULL,
82     Contact_No varchar(20) Not null,
83     primary key(Vehicle_No ,Contact_No));
84
85 • create table Decoration(
86     Deco_ID varchar(12) not null,

```

The 'Output' pane shows the results of the executed statements:

#	Time	Action	Message	Duration / Fetch
7	00:58:37	create table Emp_ID_Contact_Number(Emp_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, ... 0 row(s) affected		0.031 sec
8	00:59:29	create table Music(Band_ID varchar(12) not null, Band_Name varchar(50), Style_of_Music varchar(75) , Charge_... 0 row(s) affected		0.031 sec
9	00:59:59	create table Music_Contact_Number(Band_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, pri... 0 row(s) affected		0.031 sec
10	01:00:39	create table Hotel(Hotel_Reg_No varchar(20) not null, Hotel_Name varchar(30), Block_No varchar(15), City va... 0 row(s) affected		0.031 sec
11	01:01:10	create table Hotel_Contact_Number(Hotel_Reg_No varchar(12) NOT NULL, Contact_No varchar(20) Not null, ... 0 row(s) affected		0.031 sec
12	01:01:47	create table Transportation(Vehicle_No varchar(15) not null, Type_of_Vehicle varchar(20) , charge_per_1km dou... 0 row(s) affected		0.016 sec
13	01:02:18	create table Transportation_Contact_Number(Vehicle_No varchar(15) NOT NULL, Contact_No varchar(20) No... 0 row(s) affected		0.031 sec

Figure 3.2. 11: Transportation_Contact_Number Table creation

The screenshot shows the MySQL Workbench interface with the 'Event_Planning_Management' schema selected. In the central query editor, the following SQL code is being run:

```

73
74 • create table Transportation(
75     Vehicle_No varchar(15) not null,
76     Type_of_Vehicle varchar(20) ,
77     charge_per_1km double,
78     primary key(Vehicle_No));
79
80 • create table Transportation_Contact_Number(
81     Vehicle_No varchar(15) NOT NULL,
82     Contact_No varchar(20) Not null,
83     primary key(Vehicle_No ,Contact_No));
84
85 • create table Decoration(
86     Deco_ID varchar(12) not null,
87     Contact_No varchar(20) ,
88     primary key(Deco_ID));
89
90 • create table Photography(
91     Photograpy_ID varchar(12) not null,
92     Photography_Name varchar(30) ,

```

The 'Output' pane shows the results of the executed statements:

#	Time	Action	Message	Duration / Fetch
8	00:59:29	create table Music(Band_ID varchar(12) not null, Band_Name varchar(50), Style_of_Music varchar(75) , Charge_... 0 row(s) affected		0.032 sec
9	00:59:59	create table Music_Contact_Number(Band_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, pri... 0 row(s) affected		0.031 sec
10	01:00:39	create table Hotel(Hotel_Reg_No varchar(20) not null, Hotel_Name varchar(30), Block_No varchar(15), City va... 0 row(s) affected		0.031 sec
11	01:01:10	create table Hotel_Contact_Number(Hotel_Reg_No varchar(12) NOT NULL, Contact_No varchar(20) Not null, ... 0 row(s) affected		0.031 sec
12	01:01:47	create table Transportation(Vehicle_No varchar(15) not null, Type_of_Vehicle varchar(20) , charge_per_1km dou... 0 row(s) affected		0.016 sec
13	01:02:18	create table Transportation_Contact_Number(Vehicle_No varchar(15) NOT NULL, Contact_No varchar(20) No... 0 row(s) affected		0.031 sec
14	01:03:03	create table Decoration(Deco_ID varchar(12) not null, Contact_No varchar(20) , primary key(Deco_ID)) 0 row(s) affected		0.016 sec

Figure 3.2. 12: Decoration Table creation

MySQL Workbench

Local instance MySQL83 - Workshop2

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

No object selected

```

82     Contact_No varchar(30) Not null,
83     primary key(Vehicle_No ,Contact_No));
84
85 • create table Decoration(
86     Deco_ID varchar(12) not null,
87     Contact_No varchar(30) ,
88     primary key(Deco_ID));
89
90 • create table Photography(
91     Photgpry_ID varchar(12) not null,
92     Photography_Name varchar(30) ,
93     Charge_per_Person double,
94     primary key(Photgpry_ID));
95
96 • create table Photography_Contact_Number(
97     Photgpry_ID varchar(12) NOT NULL,
98     Contact_No varchar(30) Not null,
99     primary key(Photgpry_ID ,Contact_No));
100
101 • create table Catering(

```

Output

#	Time	Action	Message	Duration / Fetch
9	00:59:59	create table Music_Contact_Number(Band_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, pt...)	0 row(s) affected	0.031 sec
10	01:00:39	create table Hotel(Hotel_Reg_No varchar(20) not null, Hotel_Name varchar(30), Block_No varchar(15), City va...)	0 row(s) affected	0.031 sec
11	01:01:10	create table Hotel_Contact_Number(Hotel_Reg_No varchar(12) NOT NULL, Contact_No varchar(20) Not null, ...)	0 row(s) affected	0.031 sec
12	01:01:47	create table Transportation(Vehicle_No varchar(15) not null, Type_of_Vehicle varchar(20), charge_per_1km dou...	0 row(s) affected	0.016 sec
13	01:02:18	create table Transportation_Contact_Number(Vehicle_No varchar(15) NOT NULL, Contact_No varchar(20) No...)	0 row(s) affected	0.031 sec
14	01:03:03	create table Decoration(Deco_ID varchar(12) not null, Contact_No varchar(20) ,primary key(Deco_ID))	0 row(s) affected	0.016 sec
15	01:03:51	create table Photography(Photgpry_ID varchar(12) not null, Photography_Name varchar(30) ,Charge_per_Pers...	0 row(s) affected	0.015 sec

Object Info Session

Type here to search

28°C 4/9/2024

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

Figure 3.2. 13: Photography Table creation

MySQL Workbench

Local instance MySQL83 - Workshop2

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

No object selected

```

85 • create table Decoration(
86     Deco_ID varchar(12) not null,
87     Contact_No varchar(30) ,
88     primary key(Deco_ID));
89
90 • create table Photography(
91     Photgpry_ID varchar(12) not null,
92     Photography_Name varchar(30) ,
93     Charge_per_Person double,
94     primary key(Photgpry_ID));
95
96 • create table Photography_Contact_Number(
97     Photgpry_ID varchar(12) NOT NULL,
98     Contact_No varchar(30) Not null,
99     primary key(Photgpry_ID ,Contact_No));
100
101 • create table Catering(

```

Output

#	Time	Action	Message	Duration / Fetch
10	01:00:39	create table Hotel(Hotel_Reg_No varchar(20) not null, Hotel_Name varchar(30), Block_No varchar(15), City va...)	0 row(s) affected	0.031 sec
11	01:01:10	create table Hotel_Contact_Number(Hotel_Reg_No varchar(12) NOT NULL, Contact_No varchar(20) Not null, ...)	0 row(s) affected	0.031 sec
12	01:01:47	create table Transportation(Vehicle_No varchar(15) not null, Type_of_Vehicle varchar(20), charge_per_1km dou...	0 row(s) affected	0.016 sec
13	01:02:18	create table Transportation_Contact_Number(Vehicle_No varchar(15) NOT NULL, Contact_No varchar(20) No...)	0 row(s) affected	0.031 sec
14	01:03:03	create table Decoration(Deco_ID varchar(12) not null, Contact_No varchar(20) ,primary key(Deco_ID))	0 row(s) affected	0.016 sec
15	01:03:51	create table Photography(Photgpry_ID varchar(12) not null, Photography_Name varchar(30) ,Charge_per_Pers...	0 row(s) affected	0.015 sec
16	01:04:30	create table Photography_Contact_Number(Photgpry_ID varchar(12) NOT NULL, Contact_No varchar(20) N...)	0 row(s) affected	0.032 sec

Object Info Session

Type here to search

28°C 4/9/2024

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

Figure 3.2. 14: Photography_Contact_Number Table creation

The screenshot shows the MySQL Workbench interface with the 'Event_Planning_Management' schema selected. In the SQL pane, the following SQL code is being run:

```

91  Photograpy_ID varchar(12) not null,
92  Photography_Name varchar(30) ,
93  Charge_per_Person double,
94  primary key(Photograpy_ID);
95
96 • create table Photography_Contact_Number(
97  Photograpy_ID varchar(12) NOT NULL,
98  Contact_No varchar(20) Not null,
99  primary key(Photograpy_ID ,Contact_No));
100
101 • create table Catering(
102  Catering_ID varchar(12) not null,
103  Catering_Name varchar(30),
104  Charge_per_Person double ,
105  primary key(Catering_ID));
106
107 • create table Catering_Contact_Number(
108  Catering_ID varchar(12) NOT NULL,
109  Contact_No varchar(20) Not null,
110  primary key(Catering_ID ,Contact_No));

```

The 'Output' pane shows the results of the executed statements:

#	Time	Action	Message	Duration / Fetch
11	01:01:10	create table Hotel_Contact_Number(Hotel_Reg_No varchar(12) NOT NULL, Contact_No varchar(20) Not null, ...)	0 row(s) affected	0.031 sec
12	01:01:47	create table Transportation(Vehicle_No varchar(15) not null, Type_of_Vehicle varchar(20), charge_per_1km dou...	0 row(s) affected	0.016 sec
13	01:02:18	create table Transportation_Contact_Number(Vehicle_No varchar(15) NOT NULL, Contact_No varchar(20) No...	0 row(s) affected	0.031 sec
14	01:03:03	create table Decoration(Deco_ID varchar(12) not null, Contact_No varchar(20), primary key(Deco_ID))	0 row(s) affected	0.016 sec
15	01:03:51	create table Photography(Photograpy_ID varchar(12) not null, Photography_Name varchar(30), Charge_per_Pers...	0 row(s) affected	0.015 sec
16	01:04:30	create table Photography_Contact_Number(Photograpy_ID varchar(12) NOT NULL, Contact_No varchar(20) N...	0 row(s) affected	0.032 sec
17	01:05:09	create table Catering(Catering_ID varchar(12) not null, Catering_Name varchar(30), Charge_per_Person double ...)	0 row(s) affected	0.015 sec

Figure 3.2. 15: Catering Table creation

The screenshot shows the MySQL Workbench interface with the 'Event_Planning_Management' schema selected. In the SQL pane, the following SQL code is being run:

```

100
101 • create table Catering(
102  Catering_ID varchar(12) not null,
103  Catering_Name varchar(30),
104  Charge_per_Person double ,
105  primary key(Catering_ID));
106
107 • create table Catering_Contact_Number(
108  Catering_ID varchar(12) NOT NULL,
109  Contact_No varchar(20) Not null,
110  primary key(Catering_ID ,Contact_No));
111
112 • create table Booking(
113  Client_ID varchar(12) not null,
114  Event_ID varchar(12) not null,
115  Payment double ,

```

The 'Output' pane shows the results of the executed statements:

#	Time	Action	Message	Duration / Fetch
12	01:01:47	create table Transportation(Vehicle_No varchar(15) not null, Type_of_Vehicle varchar(20), charge_per_1km dou...	0 row(s) affected	0.016 sec
13	01:02:18	create table Transportation_Contact_Number(Vehicle_No varchar(15) NOT NULL, Contact_No varchar(20) No...	0 row(s) affected	0.031 sec
14	01:03:03	create table Decoration(Deco_ID varchar(12) not null, Contact_No varchar(20), primary key(Deco_ID))	0 row(s) affected	0.016 sec
15	01:03:51	create table Photography(Photograpy_ID varchar(12) not null, Photography_Name varchar(30), Charge_per_Pers...	0 row(s) affected	0.015 sec
16	01:04:30	create table Photography_Contact_Number(Photograpy_ID varchar(12) NOT NULL, Contact_No varchar(20) N...	0 row(s) affected	0.032 sec
17	01:05:09	create table Catering(Catering_ID varchar(12) not null, Catering_Name varchar(30), Charge_per_Person double ...)	0 row(s) affected	0.015 sec
18	01:05:49	create table Catering_Contact_Number(Catering_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, ...)	0 row(s) affected	0.015 sec

Figure 3.2. 16: Catering_Contact_Number Table creation

The screenshot shows the MySQL Workbench interface with the 'Workshop2' tab selected. In the central pane, the SQL editor displays the following code:

```

107 * create table Catering_Contact_Number(
108     Catering_ID varchar(12) NOT NULL,
109     Contact_No varchar(20) Not null,
110     primary key(Catering_ID ,Contact_No));
111
112 * create table Booking(
113     Client_ID varchar(12) not null,
114     Event_ID varchar(12) not null,
115     Payment double ,
116     Booking_Date date,
117     primary key(Client_ID,Event_ID));
118
119 * create table Clients(
120     Client_ID varchar(12) not null,
121     First_Name varchar(30) ,

```

The 'Output' pane below shows the execution log:

#	Time	Action	Message	Duration / Fetch
13	01:02:18	create table Transportation_Contact_Number(Vehicle_No varchar(15) NOT NULL, Contact_No varchar(20) No...)	0 row(s) affected	0.031 sec
14	01:03:03	create table Decoration(Deco_ID varchar(12) not null, Contact_No varchar(20) ,primary key(Deco_ID))	0 row(s) affected	0.016 sec
15	01:03:51	create table Photography(Photograpg_ID varchar(12)not null, Photography_Name varchar(30) ,Charge_per_Pers...)	0 row(s) affected	0.015 sec
16	01:04:30	create table Photography_Contact_Number(Photograpg_ID varchar(12) NOT NULL, Contact_No varchar(20) N...)	0 row(s) affected	0.032 sec
17	01:05:09	create table Catering(Catering_ID varchar(12) not null, Catering_Name varchar(30), Charge_per_Person double...)	0 row(s) affected	0.015 sec
18	01:05:49	create table Catering_Contact_Number(Catering_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, ...)	0 row(s) affected	0.015 sec
19	01:06:16	create table Booking(Client_ID varchar(12) not null, Event_ID varchar(12) not null, Payment double ,Booking_...)	0 row(s) affected	0.016 sec

Figure 3.2. 17: Booking Table creation

The screenshot shows the MySQL Workbench interface with the 'Workshop2' tab selected. In the central pane, the SQL editor displays the following code:

```

115
116     Payment double ,
117     Booking_Date date,
118     primary key(Client_ID,Event_ID));
119
120 * create table Clients(
121     Client_ID varchar(12) not null,
122     First_Name varchar(30) ,
123     Last_Name varchar(30) ,
124     Block_No varchar(10),
125     Street varchar(80) ,
126     City varchar(50),
127     primary key(client_ID));
128
129 * create table Clients_Contact_Number(
130     Client_ID varchar(12) NOT NULL,
131     Contact_No varchar(20) Not null,

```

The 'Output' pane below shows the execution log:

#	Time	Action	Message	Duration / Fetch
14	01:03:03	create table Decoration(Deco_ID varchar(12) not null, Contact_No varchar(20) ,primary key(Deco_ID))	0 row(s) affected	0.016 sec
15	01:03:51	create table Photography(Photograpg_ID varchar(12)not null, Photography_Name varchar(30) ,Charge_per_Pers...)	0 row(s) affected	0.015 sec
16	01:04:30	create table Photography_Contact_Number(Photograpg_ID varchar(12) NOT NULL, Contact_No varchar(20) N...)	0 row(s) affected	0.032 sec
17	01:05:09	create table Catering(Catering_ID varchar(12) not null, Catering_Name varchar(30), Charge_per_Person double...)	0 row(s) affected	0.015 sec
18	01:05:49	create table Catering_Contact_Number(Catering_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, ...)	0 row(s) affected	0.015 sec
19	01:06:16	create table Booking(Client_ID varchar(12) not null, Event_ID varchar(12) not null, Payment double ,Booking_...)	0 row(s) affected	0.016 sec
20	01:06:42	create table Clients(Client_ID varchar(12) not null, First_Name varchar(30) ,Last_Name varchar(30) ,Block_No...)	0 row(s) affected	0.015 sec

Figure 3.2. 18: Clients Table creation

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

- Title Bar:** MySQL Workbench - Local instance MySQL83 - Workshop2
- Toolbar:** Standard MySQL Workbench toolbar.
- Schemas:** Schemas pane showing 'foe_23' and 'sys'.
- Query Editor:** Workshop2 tab, code editor showing the creation of the 'Clients_Contact_Number' table and its primary key constraint.
- Output:** Action Output pane showing the execution log with 21 entries, indicating successful table creation and foreign key references.
- System Bar:** Windows taskbar at the bottom showing various pinned icons and system status.

```

121     First_Name varchar(30) ,
122     Last_Name varchar(30) ,
123     Block_No varchar(10),
124     Street varchar(80) ,
125     City varchar(50),
126     primary key(client_ID));
127
128 * create table Clients_Contact_Number(
129     Client_ID varchar(12) NOT NULL,
130     Contact_No varchar(20) Not null,
131     primary key(Client_ID ,Contact_No));
132
133 * create table Feedback(
134     Client_ID varchar(12),
135     Feedback_Comment varchar(100),
136     Foreign key(Client_ID) References Clients(Client_ID)on delete cascade on update cascade);

```

Figure 3.2. 19: Clients_Contact_Number Table creation

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

- Title Bar:** MySQL Workbench - Local instance MySQL83 - Workshop2
- Toolbar:** Standard MySQL Workbench toolbar.
- Schemas:** Schemas pane showing 'foe_23' and 'sys'.
- Query Editor:** Workshop2 tab, code editor showing the creation of the 'Feedback' table and its foreign key constraint, along with the creation of the 'Rating' table.
- Output:** Action Output pane showing the execution log with 22 entries, indicating successful table creation and foreign key references.
- System Bar:** Windows taskbar at the bottom showing various pinned icons and system status.

```

127
128 * create table Clients_Contact_Number(
129     Client_ID varchar(12) NOT NULL,
130     Contact_No varchar(20) Not null,
131     primary key(Client_ID ,Contact_No));
132
133 * create table Feedback(
134     Client_ID varchar(12),
135     Feedback_Comment varchar(100),
136     Foreign key(Client_ID) References Clients(Client_ID)on delete cascade on update cascade);
137
138 * CREATE TABLE Rating (
139     Event_ID VARCHAR(12) ,
140     Rating INT CHECK (Rating >= 0 AND Rating<= 10),
141     FOREIGN KEY (Event_ID) REFERENCES Events_TB(Event_ID)on delete cascade on update cascade) ;
142

```

Figure 3.2. 20: Feedback Table creation

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

- Title Bar:** MySQL Workbench - Local instance MySQL8.0 - Workshop2
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Navigator:** Schemas (foe_23, sys)
- SQL Editor:** Event_Planning_Management_Workshop2


```

133 • create table Feedback(
134     Client_ID varchar(12),
135     Feedback_Comment varchar(100),
136     Foreign key(Client_ID) References Clients(Client_ID)on delete cascade on update cascade);
137
138 • CREATE TABLE Rating (
139     Event_ID VARCHAR(12) ,
140     Rating INT CHECK (Rating >= 0 AND Rating<= 10),
141     FOREIGN KEY (Event_ID) REFERENCES Events_TB(Event_ID)on delete cascade on update cascade);
142
143
144 /* Relationship between Employee and Event_Coordinator */
145 • ALTER TABLE EMPLOYEE ADD CONSTRAINT FK_Dept_ID foreign key(Coordinator_ID) references Event_Coordinator(Coordinator_ID);
146
147 /* Relationship between Event_Coordinator and Event */
148 • ALTER TABLE Events_TB ADD CONSTRAINT  foreign key(Coordinator_ID) references Event_Coordinator(Coordinator_ID) on v
      
```
- Output:** Action Output

#	Time	Action	Message	Duration / Fetch
17	01:05:09	create table Catering(Catering_ID varchar(12) not null, Catering_Name varchar(30), Charge_per_Person double...)	0 row(s) affected	0.015 sec
18	01:05:49	create table Catering_Contact_Number(Catering_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null,...)	0 row(s) affected	0.015 sec
19	01:06:16	create table Booking(Event_ID varchar(12) not null, Event_ID varchar(12) not null, Payment double, Booking_...	0 row(s) affected	0.016 sec
20	01:06:42	create table Clients(Client_ID varchar(12) not null, First_Name varchar(30), Last_Name varchar(30), Block_No...	0 row(s) affected	0.015 sec
21	01:07:19	create table Clients_Contact_Number(Client_ID varchar(12) NOT NULL, Contact_No varchar(20) Not null, ph...	0 row(s) affected	0.031 sec
22	01:07:49	create table Feedback(Client_ID varchar(12), Feedback_Comment varchar(100), Foreign key(Client_ID) Ref...	0 row(s) affected	0.062 sec
23	01:08:16	CREATE TABLE Rating (Event_ID VARCHAR(12), Rating INT CHECK (Rating >= 0 AND Rating<= 10), ...)	0 row(s) affected	0.125 sec
- System Bar:** Type here to search, Taskbar icons, Date/Time (4/9/2024, 1:08 AM), Weather (28°C)

Figure 3.2. 21: Rating Table creation

3.3 Relationships making between tables

The screenshot shows the MySQL Workbench interface with the 'Event_Planning_Management' schema selected. The 'Workshop2' tab is active, displaying SQL code for creating foreign key constraints. The code includes comments for relationships between Employee, Event_Coordinator, Event, Event_Coordinator_Contact_Number, Music, Events, Hotel, Transportation, Decoration, and Event. The 'Output' pane shows the execution results with 7 successful operations and their details.

```

144 /* Relationship between Employee and Event_Coordinator */
145 • ALTER TABLE EMPLOYEE ADD CONSTRAINT FK_Dept_ID foreign key(Coordinator_ID) references Event_Coordinator(Coordinator_ID) on delete cascade on update cascade;
146
147 /* Relationship between Event_Coordinator and Event */
148 • ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Coordinator_ID) references Event_Coordinator(Coordinator_ID) on delete cascade on update cascade;
149
150 /* Relationship between Event_Coordinator and Event_Coordinator_Contact_Number */
151 • ALTER TABLE Event_Coordinator_Contact_Number ADD CONSTRAINT foreign key(Coordinator_ID) references Event_Coordinator(Coordinator_ID) on delete cascade on update cascade;
152
153 /* Relationship between Music and Event */
154 • ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Band_ID) references Music(Band_ID) on delete cascade on update cascade;
155
156 /* Relationship between Hotel and Event */
157 • ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Hotel_Reg_No) references Hotel(Hotel_Reg_No) on delete cascade on update cascade;
158
159 /* Relationship between Transportation and Event */
160 • ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Vehicle_No) references Transportation(Vehicle_No) on delete cascade on update cascade;
161
162 /* Relationship between Decoration and Event */
163 • ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Deco_ID) references Decoration(Deco_ID) on delete cascade on update cascade;
164

```

#	Time	Action	Message	Duration / Fetch
1	01:09:16	ALTER TABLE EMPLOYEE ADD CONSTRAINT FK_Dept_ID foreign key(Coordinator_ID) references Event_Coordinator(Coordinator_ID) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.109 sec
2	01:09:20	ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Coordinator_ID) references Event_Coordinator(Coordinator_ID) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.156 sec
3	01:09:34	ALTER TABLE Event_Coordinator_Contact_Number ADD CONSTRAINT foreign key(Coordinator_ID) references Event_Coordinator(Coordinator_ID) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.078 sec
4	01:09:41	ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Band_ID) references Music(Band_ID) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.125 sec
5	01:09:46	ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Hotel_Reg_No) references Hotel(Hotel_Reg_No) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.140 sec
6	01:09:51	ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Vehicle_No) references Transportation(Vehicle_No) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.125 sec
7	01:09:55	ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Deco_ID) references Decoration(Deco_ID) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.141 sec

Figure 3.3. 1: Relationships between Tables

The screenshot shows the MySQL Workbench interface with the 'Event_Planning_Management' schema selected. The 'Workshop2' tab is active, displaying SQL code for creating foreign key constraints. The code includes comments for relationships between Decoration, Photography, Catering, Client, Booking, and Music. The 'Output' pane shows the execution results with 8 successful operations and their details.

```

161 /* Relationship between Decoration and Event */
162 • ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Deco_ID) references Decoration(Deco_ID) on delete cascade on update cascade;
163
164 /* Relationship between Photography and Event */
165 • ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Photograpy_ID) references Photography(Photograpy_ID) on delete cascade on update cascade;
166
167 /* Relationship between Catering and Event */
168 • ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Catering_ID) references Catering(Catering_ID) on delete cascade on update cascade;
169
170 /* Relationship between Client and Booking */
171 • ALTER TABLE Booking ADD CONSTRAINT foreign key(Client_ID) references Clients(Client_ID) on delete cascade on update cascade;
172
173 /* Relationship between Event_TB and Booking */
174 • ALTER TABLE booking ADD INDEX idx_event_id (Event_ID);
175 • ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Event_ID) references booking(Event_ID) on delete cascade on update cascade;
176
177 /* Relationship between Client and Event */
178 • ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Client_ID) references Clients(Client_ID) on delete cascade on update cascade;
179
180 • alter table Music_Contact_Number add constraint foreign key(Band_ID) references Music(Band_ID) on delete cascade on update cascade;

```

#	Time	Action	Message	Duration / Fetch
1	01:12:58	ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Photograpy_ID) references Photography(Photograpy_ID) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.156 sec
2	01:13:02	ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Catering_ID) references Catering(Catering_ID) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.140 sec
3	01:13:05	ALTER TABLE Booking ADD CONSTRAINT foreign key(Client_ID) references Clients(Client_ID) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.062 sec
4	01:13:08	ALTER TABLE booking ADD INDEX idx_event_id (Event_ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
5	01:13:13	ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Event_ID) references booking(Event_ID) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.172 sec
6	01:13:13	ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Event_ID) references booking(Event_ID) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.156 sec
7	01:13:17	ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Client_ID) references Clients(Client_ID) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.156 sec
8	01:13:22	alter table Music_Contact_Number add constraint foreign key(Band_ID) references Music(Band_ID) on delete cascade on update cascade;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.062 sec

Figure 3.3. 2: Relationships between Tables

The screenshot shows the MySQL Workbench interface. The main area displays a script editor with the following SQL code:

```

179 • ALTER TABLE Events_TB ADD CONSTRAINT foreign key(Client_ID) references Clients(Client_ID) on delete cascade on update cascade;
180
181 • alter table Music_Contact_Number add constraint foreign key(Band_ID) references Music(Band_ID) on delete cascade on update cascade ;
182 • alter table Hotel_Contact_Number add constraint foreign key(hotel_Reg_No) references Hotel(Hotel_Reg_No) on delete cascade on update cascade ;
183 • alter table Catering_Contact_Number add constraint foreign key(Catering_ID) references Catering(Catering_ID) on delete cascade on update cascade ;
184 • alter table Photography_Contact_Number add constraint foreign key(Photography_ID) references Photography(Photography_ID) on delete cascade on update cascade ;
185 • alter table Transportation_Contact_Number add constraint foreign key(Vehicle_No) references transportation(Vehicle_No) on delete cascade on update cascade ;
186 • alter table Clients_Contact_Number add constraint foreign key(Client_ID) references Clients(Client_ID) on delete cascade on update cascade ;
187
188 /* Relationship between Employee and Event */
189 • CREATE TABLE EmployEventRelation(
190     Empl_ID varchar(12) not null,
191     Event_ID varchar(12) not null,
192     primary key (Event_ID,Empl_ID),
193     CONSTRAINT fk_Pk3 foreign key(Event_ID) references EVENTS_TB(Event_ID),
194     CONSTRAINT fk_emp foreign key(Empl_ID) references EMPLOYEE(Empl_ID));
195
196
197
198 /* Insert Values */
199

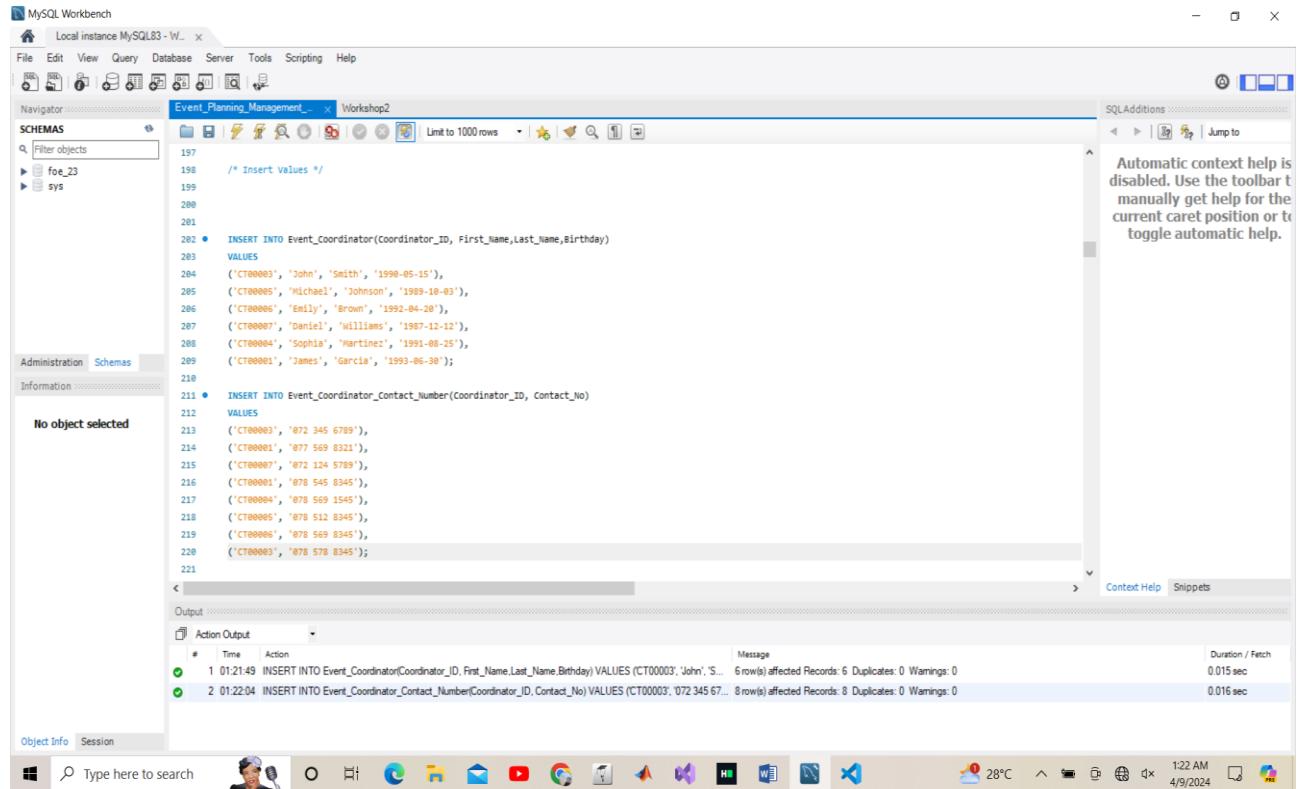
```

The output window below shows the results of the executed statements:

#	Time	Action	Message	Duration / Fetch
1	01:14:24	altertable Hotel_Contact_Number add constraint foreign key(Hotel_Reg_No) references Hotel(Hotel_Reg_No) on d...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.078 sec
2	01:14:27	altertable Catering_Contact_Number add constraint foreign key(Catering_ID) references Catering(Catering_ID) on d...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.079 sec
3	01:14:31	altertable Photography_Contact_Number add constraint foreign key(Photography_ID) references Photography(Phot...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.079 sec
4	01:14:34	altertable Transportation_Contact_Number add constraint foreign key(Vehicle_No) references transportation(Ve...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.125 sec
5	01:14:38	altertable Clients_Contact_Number add constraint foreign key(Client_ID) references Clients(Client_ID) on delete cas...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.078 sec
6	01:14:46	CREATE TABLE EmployEventRelation(Empl_ID varchar(12) not null, Event_ID varchar(12) not null, primary ke...	0 row(s) affected	0.110 sec

Figure 3.3. 3: Relationships between Tables

3.4 Insert Values into Tables



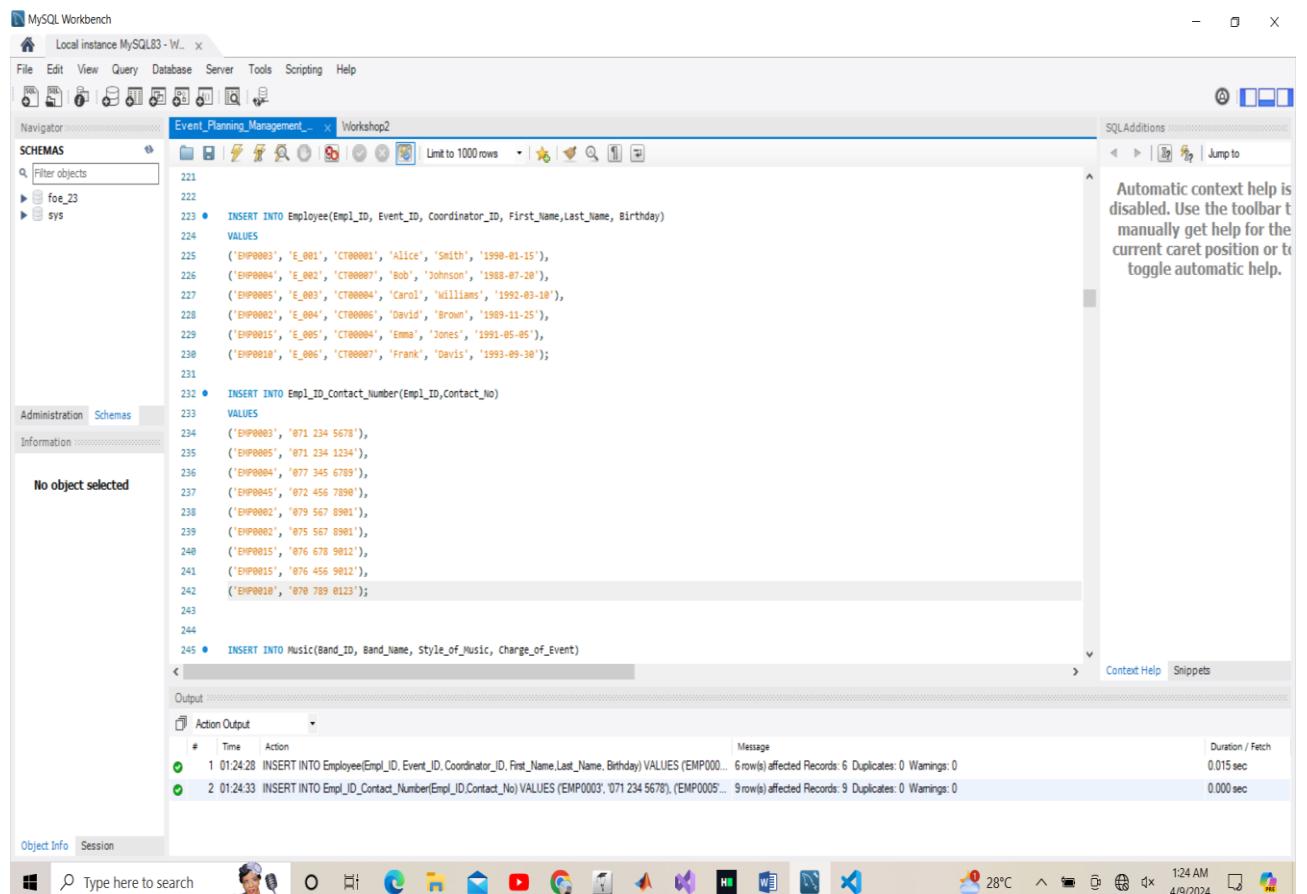
```

MySQL Workbench
Local instance MySQL83 - Workshop2

File Edit View Query Database Server Tools Scripting Help
Navigator Schemas Administration Schemas Information No object selected Object Info Session
Event_Planning_Management_... Workshop2
197
198 /* Insert Values */
199
200
201
202 • INSERT INTO Event_Coordinator(Coordinator_ID, First_Name,Last_Name,Birthday)
203     VALUES
204     ('CT00003', 'John', 'Smith', '1990-05-15'),
205     ('CT00005', 'Michael', 'Johnson', '1988-10-03'),
206     ('CT00006', 'Emily', 'Brown', '1992-04-20'),
207     ('CT00007', 'Daniel', 'Williams', '1987-12-12'),
208     ('CT00004', 'Sophia', 'Martinez', '1991-08-25'),
209     ('CT00001', 'James', 'Garcia', '1993-06-30');
210
211 • INSERT INTO Event_Coordinator_Contact_Number(Coordinator_ID, Contact_No)
212     VALUES
213     ('CT00003', '072 345 6789'),
214     ('CT00001', '077 569 8321'),
215     ('CT00007', '072 123 5789'),
216     ('CT00001', '078 545 8345'),
217     ('CT00004', '078 569 1545'),
218     ('CT00005', '078 512 8345'),
219     ('CT00006', '078 569 8345'),
220     ('CT00003', '078 578 8345');
221
Output Action Output
# Time Action Message Duration / Fetch
1 01:21:49 INSERT INTO Event_Coordinator(Coordinator_ID, First_Name,Last_Name,Birthday) VALUES (CT00003,'John','S... 6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0 0.015 sec
2 01:22:04 INSERT INTO Event_Coordinator_Contact_Number(Coordinator_ID, Contact_No) VALUES (CT00003,'072 345 67... 8 row(s) affected Records: 8 Duplicates: 0 Warnings: 0 0.016 sec
Object Info Session
Type here to search 28°C 1:22 AM 4/9/2024

```

Figure 3.4. 1: Insert values into Event_Coordinator and Event_Coordinator_Contact_Number tables



```

MySQL Workbench
Local instance MySQL83 - Workshop2

File Edit View Query Database Server Tools Scripting Help
Navigator Schemas Administration Schemas Information No object selected Object Info Session
Event_Planning_Management_... Workshop2
221
222
223 • INSERT INTO Employee(Empl_ID, Event_ID, Coordinator_ID, First_Name,Last_Name, Birthday)
224     VALUES
225     ('EMP0003', 'E_001', 'CT00001', 'Alice', 'Smith', '1990-01-15'),
226     ('EMP0004', 'E_002', 'CT00007', 'Bob', 'Johnson', '1988-10-28'),
227     ('EMP0005', 'E_003', 'CT00004', 'Carol', 'Williams', '1992-03-10'),
228     ('EMP0002', 'E_004', 'CT00006', 'David', 'Brown', '1989-11-25'),
229     ('EMP0015', 'E_005', 'CT00004', 'Emma', 'Jones', '1991-05-05'),
230     ('EMP0010', 'E_006', 'CT00007', 'Frank', 'Davis', '1993-09-30');
231
232 • INSERT INTO Empl_ID_Contact_Number(Empl_ID,Contact_No)
233     VALUES
234     ('EMP0003', '071 234 5678'),
235     ('EMP0005', '071 234 1234'),
236     ('EMP0004', '077 345 6789'),
237     ('EMP0005', '072 456 7890'),
238     ('EMP0002', '079 567 8901'),
239     ('EMP0002', '075 567 8901'),
240     ('EMP0015', '076 678 9012'),
241     ('EMP0015', '076 456 9012'),
242     ('EMP0010', '070 789 0123');
243
244
245 • INSERT INTO Music(Band_ID, Band_Name, Style_of_Music, Charge_of_Event)
Output Action Output
# Time Action Message Duration / Fetch
1 01:24:28 INSERT INTO Employee(Empl_ID, Event_ID, Coordinator_ID, First_Name,Last_Name, Birthday) VALUES (EMP0003,'E_001','CT00001','Alice','Smith','1990-01-15') 6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0 0.015 sec
2 01:24:33 INSERT INTO Empl_ID_Contact_Number(Empl_ID,Contact_No) VALUES (EMP0003,'071 234 5678'),(EMP0005,'071 234 1234'),(EMP0004,'077 345 6789'),(EMP0005,'072 456 7890'),(EMP0002,'079 567 8901'),(EMP0002,'075 567 8901'),(EMP0015,'076 678 9012'),(EMP0015,'076 456 9012'),(EMP0010,'070 789 0123') 9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0 0.000 sec
Object Info Session
Type here to search 28°C 1:24 AM 4/9/2024

```

Figure 3.4. 2: Insert values into Employee and Empl_ID_Contact_Number tables

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

- Title Bar:** MySQL Workbench - Local instance MySQL83 - Workshop2
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Schemas:** Schemas pane showing 'foe_23' and 'sys'.
- Query Editor:** Event_Planning_Management_ Workshop2 tab. The code is as follows:

```

245 • INSERT INTO Music(Band_ID, Band_Name, Style_of_Music, Charge_of_Event)
246   VALUES
247     ('BT00002', 'Phoenix', 'Rock', '30000'),
248     ('BT00003', 'Harmony', 'Pop', '35000'),
249     ('BT00004', 'Rhythmation', 'Jazz', '28000'),
250     ('BT00005', 'SoulfulSound', 'Blues', '32000'),
251     ('BT00006', 'MelodicBeats', 'Classical', '27000'),
252     ('BT00007', 'FunkyGroove', 'Funk', '71000');
253
254 • INSERT INTO Music_Contact_Number(Band_ID, Contact_No)
255   VALUES
256     ('BT00002', '071 234 5678'),
257     ('BT00002', '077 345 6789'),
258     ('BT00002', '076 456 7890'),
259     ('BT00003', '072 345 6789'),
260     ('BT00004', '075 567 8901'),
261     ('BT00004', '073 456 7890'),
262     ('BT00005', '079 567 8901'),
263     ('BT00005', '074 567 8901'),
264     ('BT00007', '078 678 9812'),
265     ('BT00005', '077 789 0123'),
266     ('BT00006', '075 678 9812'),
267     ('BT00006', '071 789 0123'),
268     ('BT00006', '078 890 1234'),
269     ('BT00007', '076 789 0123');
270

```

- Output Tab:** Action Output section showing two successful insert operations.
- System Bar:** Shows the date and time as 4/9/2024 12:26 AM.

Figure 3.4. 3: Insert values into Music and Music_Contact_Number tables

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

- Title Bar:** MySQL Workbench - Local instance MySQL83 - Workshop2
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Schemas:** Schemas pane showing 'foe_23' and 'sys'.
- Query Editor:** Event_Planning_Management_ Workshop2 tab. The code is as follows:

```

270
271
272
273 • INSERT INTO Hotel (Hotel_Reg_No, Hotel_Name, Block_No,City,Street)
274   VALUES
275     ('H0201', 'GreenOasis', 'No:22/1', 'Colombo', 'Park Avenue'),
276     ('H0201', 'SunsetView', 'No:15/9/1', 'Kandy', 'Hillside Drive'),
277     ('H0201', 'SeaBreeze', 'No:2/1A', 'Negombo', 'Beach Road'),
278     ('H0501', 'MountainTop', 'No:105/06', 'Nuwara Eliya', 'Summit Street'),
279     ('H0601', 'GoldenPalm', 'No:18/3,21/1', 'Bentota', 'Coastal Highway'),
280     ('H0701', 'Riverfront', 'No:9/1', 'Kitulgala', 'Riverside Avenue');
281
282 • INSERT INTO Hotel_Contact_Number(Hotel_Reg_No,Contact_No)
283   VALUES
284     ('H0201', '071 234 5678'),
285     ('H0201', '071 452 5679'),
286     ('H0201', '077 345 6780'),
287     ('H0301', '077 125 7853'),
288     ('H0301', '072 345 6781'),
289     ('H0401', '079 567 8902'),
290     ('H0501', '079 567 8903'),
291     ('H0501', '079 567 7854'),
292     ('H0601', '079 567 8905'),
293     ('H0701', '079 456 9658');
294
295

```

- Output Tab:** Action Output section showing two successful insert operations.
- System Bar:** Shows the date and time as 4/9/2024 1:29 AM.

Figure 3.4. 4: Insert values into Hotel and Hotel_Contact_Number tables

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

- Title Bar:** MySQL Workbench - Local instance MySQL83 - Workshop2
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Toolbar:** Standard icons for file operations.
- Navigator:** Schemas (foe_23, sys) and Administration (No object selected).
- SQL Editor:** Event_Planning_Management_Workshop2 tab, showing two SQL statements:


```

294
295
296 • INSERT INTO transportation (Vehicle_No,Type_of_Vehicle, charge_per_1km)
VALUES
297
298 ('AB-4562', 'Sedan', '50'),
299 ('GH-5678', 'Limousine', '75'),
300 ('EF-7890', 'SUV', '100'),
301 ('KL-3456', 'Van', '150'),
302 ('CD-2156', 'Limousine', '1000'),
303 ('CD-2166', 'Motorcycle', '30');
304
305 • INSERT INTO Transportation_Contact_Number(Vehicle_No,Contact_No)
VALUES
306
307 ('AB-4562', '071 234 5678'),
308 ('CD-2156', '071 452 5678'),
309 ('KL-3456', '077 345 6789'),
310 ('GH-5678', '077 125 7852'),
311 ('KL-3456', '072 345 6789'),
312 ('EF-7890', '079 567 8901'),
313 ('EF-7890', '079 125 8901'),
314 ('CD-2156', '079 567 7852'),
315 ('CD-2166', '079 567 8901');
316
317
318
319
      
```
- Output Panel:** Action Output table showing execution results:

#	Time	Action	Message	Duration / Fetch
1	01:30:29	INSERT INTO transportation (Vehicle_No,Type_of_Vehicle, charge_per_1km) VALUES (AB-4562, 'Sedan', '50')	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.015 sec
2	01:30:32	INSERT INTO Transportation_Contact_Number(Vehicle_No,Contact_No) VALUES (AB-4562, '071 234 5678')	9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0	0.000 sec
- Object Info and Session Tabs:**
- System Tray:** Shows the date and time as 4/9/2024 1:30 AM.

Figure 3.4. 5: Insert values into transportation and Transportation_Contact_Number tables

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

- Title Bar:** MySQL Workbench - Local instance MySQL83 - Workshop2
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Toolbar:** Standard icons for file operations.
- Navigator:** Schemas (foe_23, sys) and Administration (No object selected).
- SQL Editor:** Event_Planning_Management_Workshop2 tab, showing three SQL statements:


```

320 • INSERT INTO Decoration (Deco_ID, Contact_No) VALUES
321 ('DC08046', '071 234 5678'),
322 ('DC08047', '076 543 2109'),
323 ('DC08048', '072 345 6789'),
324 ('DC08049', '078 981 2345'),
325 ('DC08050', '075 674 9812'),
326 ('DC08051', '078 123 4567');
327
328 • INSERT INTO Photography (Photography_ID, Photography_Name, Charge_per_Person) VALUES
329 ('PH00002', 'FocusClick', '150.00'),
330 ('PH00003', 'SnapStudio', '100.000'),
331 ('PH00004', 'PixelPerfect', '120.000'),
332 ('PH00005', 'LensCraft', '135.000'),
333 ('PH00006', 'ShutterMasters', '160.000'),
334 ('PH00007', 'FrameFusion', '140.000');
335
336 • INSERT INTO Photography_Contact_Number(Photography_ID, Contact_No) VALUES
337 ('PH00002', '071 234 5678'),
338 ('PH00006', '071 452 5679'),
339 ('PH00002', '077 345 6789'),
340 ('PH00002', '077 125 7853'),
341 ('PH00003', '072 345 6781'),
342 ('PH00004', '079 567 8902'),
343 ('PH00005', '079 567 8903'),
344 ('PH00005', '079 567 7854'),
345 ('PH00006', '079 567 8905');
      
```
- Output Panel:** Action Output table showing execution results:

#	Time	Action	Message	Duration / Fetch
1	01:33:05	INSERT INTO Decoration (Deco_ID, Contact_No) VALUES (DC08046, '071 234 5678'), (DC08047, '076 543 2109')	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.000 sec
2	01:33:11	INSERT INTO Photography (Photography_ID, Photography_Name, Charge_per_Person) VALUES (PH00002, 'FocusClick', '150.00')	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.000 sec
3	01:33:24	INSERT INTO Photography_Contact_Number(Photography_ID, Contact_No) VALUES (PH00002, '071 234 5678'), (PH00006, '071 452 5679')	9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0	0.000 sec
- Object Info and Session Tabs:**
- System Tray:** Shows the date and time as 4/9/2024 1:33 AM.

Figure 3.4. 6: Insert values into Decoration, Photography and Photography_Contact_Number tables

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Event_Planning_Management_, Workshop2
- SQL Editor:**

```

349 • INSERT INTO Catering(Catering_ID,Catering_Name, Charge_per_Person)
VALUES
350 ('CAT0001', 'ABC Catering', '500.00'),
351 ('CAT0002', 'XYZ Catering', '600.00'),
352 ('CAT0003', 'Best Caterers', '550.00'),
353 ('CAT0004', 'Delicious Catering', '700.00'),
354 ('CAT0005', 'Elite Caterers', '650.00'),
355 ('CAT0006', 'Dino Caterers', '700.00'),
356 ('CAT0007', 'Wee Caterers', '50.00');

358
359 • INSERT INTO Catering_Contact_Number(Catering_ID,Contact_No)
VALUES
360 ('CAT0001', '071 234 5678'),
361 ('CAT0002', '077 345 6789'),
362 ('CAT0003', '072 456 7890'),
363 ('CAT0004', '071 567 8901'),
364 ('CAT0005', '074 567 1234'),
365 ('CAT0006', '076 678 9012'),
366 ('CAT0007', '078 789 0123'),
367 ('CAT0008', '078 890 1234'),
368 ('CAT0009', '078 145 1234');
369 ('CAT0007', '078 145 1234');

370
371

```
- Output Tab:**

#	Time	Action	Message	Duration / Fetch
1	01:37:36	INSERT INTO Catering(Catering_ID,Catering_Name, Charge_per_Person) VALUES (CAT0001, 'ABC Catering', 500.00)	7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0	0.000 sec
2	01:37:40	INSERT INTO Catering_Contact_Number(Catering_ID,Contact_No) VALUES (CAT0001, '071 234 5678')	(CAT0001, '071 234 5678'), (CAT0002, '077 345 6789'), (CAT0003, '072 456 7890'), (CAT0004, '071 567 8901'), (CAT0005, '074 567 1234'), (CAT0006, '076 678 9012'), (CAT0007, '078 789 0123'), (CAT0008, '078 890 1234'), (CAT0009, '078 145 1234') 9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0	0.000 sec
- System Bar:** Type here to search, Taskbar icons, Date/Time: 4/9/2024, 1:37 AM.

Figure 3.4. 7: Insert values into Catering and Catering_Contact_Number tables

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Event_Planning_Management_, Workshop2
- SQL Editor:**

```

370
371
372 • INSERT INTO Clients (Client_ID, First_Name, Last_Name, Block_No, Street, City)
VALUES
373 ('C1_0002', 'Alice', 'Johnson', '22/4', 'Main Street', 'Colombo'),
374 ('C1_0003', 'Robert', 'Smith', '19/9', 'Beach Road', 'Galle'),
375 ('C1_0004', 'Emily', 'Davis', '11', 'Hillside Drive', 'Kandy'),
376 ('C1_0005', 'Michael', 'Wilson', '980/4', 'Summit Street', 'Nuwara Eliya'),
377 ('C1_0006', 'Jessica', 'Anderson', '00007', 'Coastal Highway', 'Bentota'),
378 ('C1_0007', 'David', 'Brown', '22/8/A', 'Riverside Avenue', 'Kitulgala');

380 • INSERT INTO Clients_Contact_Number(Client_ID, Contact_No)
VALUES
381 ('C1_0002', '071 234 5678'),
382 ('C1_0002', '071 456 5678'),
383 ('C1_0002', '071 789 5678'),
384 ('C1_0002', '077 345 6789'),
385 ('C1_0002', '072 456 7890'),
386 ('C1_0002', '074 567 1234'),
387 ('C1_0002', '076 678 9012'),
388 ('C1_0002', '078 789 0123'),
389 ('C1_0002', '079 567 8901'),
390 ('C1_0002', '076 123 9012'),
391 ('C1_0002', '078 789 0123'),
392 ('C1_0002', '071 234 5678');

393

```
- Output Tab:**

#	Time	Action	Message	Duration / Fetch
1	01:43:04	INSERT INTO Clients (Client_ID, First_Name, Last_Name, Block_No, Street, City) VALUES (C1_0002, 'Alice', 'Johnson', '22/4', 'Main Street', 'Colombo')	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.000 sec
2	01:43:10	INSERT INTO Clients_Contact_Number(Client_ID, Contact_No) VALUES (C1_0002, '071 234 5678')	(C1_0002, '071 234 5678'), (C1_0002, '077 345 6789'), (C1_0002, '072 456 7890'), (C1_0002, '074 567 1234'), (C1_0002, '076 678 9012'), (C1_0002, '078 789 0123'), (C1_0002, '079 567 8901'), (C1_0002, '076 123 9012'), (C1_0002, '078 789 0123') 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec
- System Bar:** Type here to search, Taskbar icons, Date/Time: 4/9/2024, 1:43 AM.

Figure 3.4. 8: Insert values into Clients and Clients_Contact_Number tables

The screenshot shows the MySQL Workbench interface with a query editor window titled 'Workshop2'. The code area contains two main sections of SQL queries:

```

394 • INSERT INTO Booking (Client_ID, Event_ID, Payment, Booking_Date)
VALUES
395     ('C1_0002', 'E_001', '15000.00', '2024-04-03'),
396     ('C1_0003', 'E_002', '20000.00', '2024-04-04'),
397     ('C1_0004', 'E_003', '18000.00', '2024-04-05'),
398     ('C1_0005', 'E_004', '25000.00', '2024-04-06'),
399     ('C1_0006', 'E_005', '22000.00', '2024-04-07'),
400     ('C1_0007', 'E_006', '19000.00', '2024-04-08'),
401     ('C1_0002', 'E_007', '30000.00', '2024-04-09');

402
403 • INSERT INTO Feedback (Client_ID, Feedback_Comment)
VALUES
404     ('C1_0007', 'Great event, loved the decorations!'),
405     ('C1_0002', 'The catering service was exceptional.'),
406     ('C1_0003', 'The transportation arrangements were smooth.'),
407     ('C1_0004', 'The band performance was fantastic!'),
408     ('C1_0005', 'The photography team captured amazing moments.'),
409     ('C1_0006', 'The coordination team did an excellent job!'),
410     ('C1_0002', 'The hotel venue was beautiful and comfortable.'),
411     ('C1_0007', 'Highly recommend this event planning team!');

412
413
414
415

```

The 'Output' pane shows the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
1	01:51:09	INSERT INTO Booking (Client_ID, Event_ID, Payment, Booking_Date) VALUES (C1_0002, 'E_001', '15000.00', '2024-04-03')	7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0	0.016 sec
2	01:51:12	INSERT INTO Feedback (Client_ID, Feedback_Comment) VALUES (C1_0007, 'Great event, loved the decorations!')	8 row(s) affected Records: 8 Duplicates: 0 Warnings: 0	0.016 sec

Figure 3.4. 9: Insert values into Booking and Feedback tables

The screenshot shows the MySQL Workbench interface with a query editor window titled 'Workshop2'. The code area contains two main sections of SQL queries:

```

415
416
417 • INSERT INTO Events_TB (Event_ID, Emp1_ID, Coordinator_ID, Band_ID, Hotel_Reg_No, Vehicle_No, Deco_ID, Photoprphy_ID, Catering_ID, Client_ID, Event_Name, Event_Date, Event_Time, Number,
VALUES
418     ('E_001', 'EMP0002', 'CT00003', 'BT00005', 'H0281', 'AB-4562', 'DC0046', 'PH00003', 'CAT0002', 'C1_0002', 'wedding', '2024-08-01', '14:00:00', '150),
419     ('E_002', 'EMP0003', 'CT00004', 'BT00006', 'H0381', 'CD-2155', 'DC00049', 'PH00003', 'CAT0003', 'C1_0007', 'birthday_Party', '2023-01-29', '18:00:00', '50),
420     ('E_003', 'EMP0045', 'CT00004', 'BT00003', 'H0581', 'EF-7890', 'DC00048', 'PH00002', 'CAT0003', 'C1_0003', 'Corporate Gala', '2024-04-18', '18:00:00', '200),
421     ('E_004', 'EMP0004', 'CT00005', 'BT00002', 'H0481', 'GH-5678', 'DC00049', 'PH00004', 'CAT0004', 'C1_0004', 'Conference', '2022-04-15', '09:00:00', '300),
422     ('E_005', 'EMP0015', 'CT00007', 'BT00004', 'H0681', 'CD-2156', 'DC00050', 'PH00005', 'CAT0005', 'C1_0005', 'Product Launch', '2021-12-20', '13:00:00', '150),
423     ('E_006', 'EMP0010', 'CT00001', 'BT00007', 'H0581', 'XL-3456', 'DC00051', 'PH00005', 'CAT0007', 'C1_0002', 'Charity Event', '2023-06-25', '17:00:00', '100);

424
425
426
427 • INSERT INTO Rating (Event_ID, Rating)
VALUES
428     ('E_001', 5),
429     ('E_002', 4),
430     ('E_003', 3),
431     ('E_005', 5),
432     ('E_004', 4),
433     ('E_006', 5);

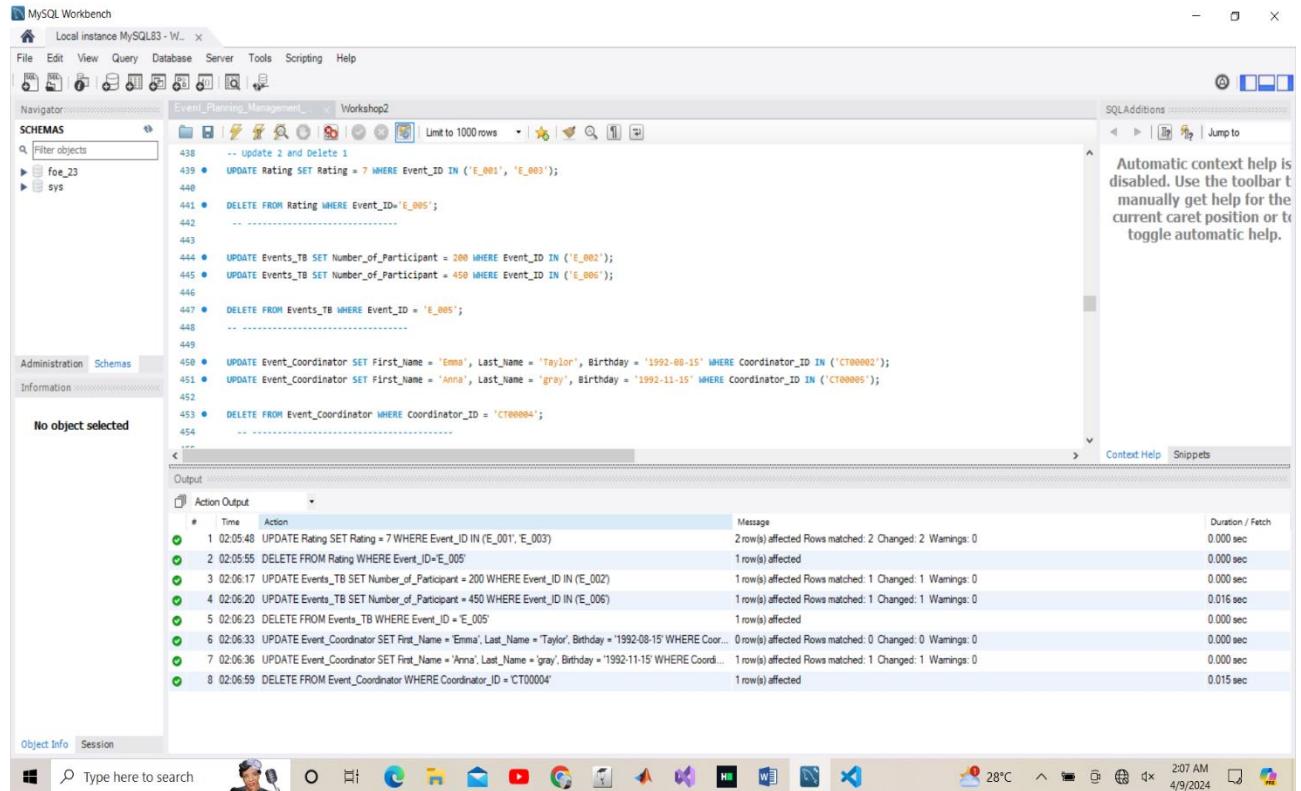
```

The 'Output' pane shows the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
1	01:52:26	INSERT INTO Events_TB (Event_ID, Emp1_ID, Coordinator_ID, Band_ID, Hotel_Reg_No, Vehicle_No, Deco_ID, Photoprphy_ID, Catering_ID, Client_ID, Event_Name, Event_Date, Event_Time, Number)	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.015 sec
2	01:52:32	INSERT INTO Rating (Event_ID, Rating) VALUES (E_001, 5), (E_002, 4), (E_003, 3), (E_005, 5), (E_004, 4), (E_006, 5);	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.000 sec

Figure 3.4. 10: Insert values into Events_TB and Rating tables

3.5 Update and delete values in Tables



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

- Schemas:** Event_Planning_Management_
- SQL Editor:**

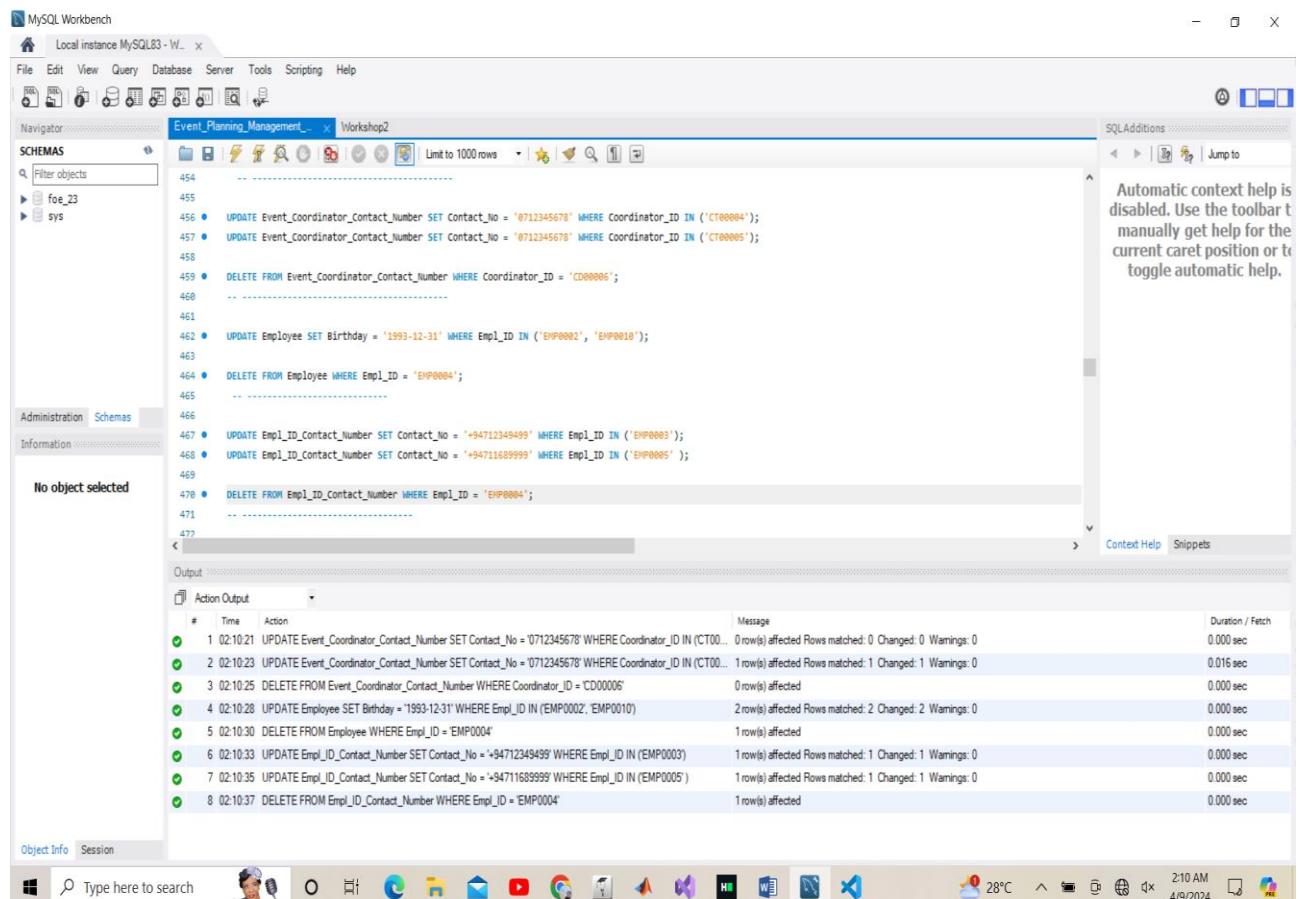
```

438 -- update 2 and Delete 1
439 • UPDATE Rating SET Rating = 7 WHERE Event_ID IN ('E_001', 'E_003');
440
441 • DELETE FROM Rating WHERE Event_ID='E_005';
442
443
444 • UPDATE Events_TB SET Number_of_Participant = 200 WHERE Event_ID IN ('E_002');
445 • UPDATE Events_TB SET Number_of_Participant = 450 WHERE Event_ID IN ('E_006');
446
447 • DELETE FROM Events_TB WHERE Event_ID = 'E_005';
448
449
450 • UPDATE Event_Coordinator SET First_Name = 'Emma', Last_Name = 'Taylor', Birthday = '1992-08-15' WHERE Coordinator_ID IN ('CT00002');
451 • UPDATE Event_Coordinator SET First_Name = 'Anna', Last_Name = 'gray', Birthday = '1992-11-15' WHERE Coordinator_ID IN ('CT00005');
452
453 • DELETE FROM Event_Coordinator WHERE Coordinator_ID = 'CT00004';
454

```
- Output:**

#	Time	Action	Message	Duration / Fetch
1	02:05:48	UPDATE Rating SET Rating = 7 WHERE Event_ID IN ('E_001', 'E_003')	2 row(s) affected Rows matched: 2 Changed: 2 Warnings: 0	0.000 sec
2	02:05:59	DELETE FROM Rating WHERE Event_ID=E_005'	1 row(s) affected	0.000 sec
3	02:06:17	UPDATE Events_TB SET Number_of_Participant = 200 WHERE Event_ID IN ('E_002')	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
4	02:06:20	UPDATE Events_TB SET Number_of_Participant = 450 WHERE Event_ID IN ('E_006')	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
5	02:06:23	DELETE FROM Events_TB WHERE Event_ID = 'E_005'	1 row(s) affected	0.000 sec
6	02:06:33	UPDATE Event_Coordinator SET First_Name = 'Emma', Last_Name = 'Taylor', Birthday = '1992-08-15' WHERE Coor...	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
7	02:06:36	UPDATE Event_Coordinator SET First_Name = 'Anna', Last_Name = 'gray', Birthday = '1992-11-15' WHERE Coord...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
8	02:06:59	DELETE FROM Event_Coordinator WHERE Coordinator_ID = 'CT00004'	1 row(s) affected	0.015 sec
- Object Info:** Session
- Session Bar:** Type here to search, Windows Start button, Taskbar icons, Date: 4/9/2024, Time: 207 AM, Weather: 28°C.

Figure 3.5. 1: Update and delete in Rating, Events_TB and Event_Coordinator tables



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

- Schemas:** Event_Planning_Management_
- SQL Editor:**

```

454
455
456 • UPDATE Event_Coordinator_Contact_Number SET Contact_No = '0712345678' WHERE Coordinator_ID IN ('CT00004');
457 • UPDATE Event_Coordinator_Contact_Number SET Contact_No = '0712345678' WHERE Coordinator_ID IN ('CT00005');
458
459 • DELETE FROM Event_Coordinator_Contact_Number WHERE Coordinator_ID = 'CD00006';
460
461
462 • UPDATE Employee SET Birthday = '1993-12-31' WHERE Empl_ID IN ('EMP0002', 'EMP0010');
463
464 • DELETE FROM Employee WHERE Empl_ID = 'EMP0004';
465
466
467 • UPDATE Empl_ID_Contact_Number SET Contact_No = '+94712349499' WHERE Empl_ID IN ('EMP0003');
468 • UPDATE Empl_ID_Contact_Number SET Contact_No = '+94711689999' WHERE Empl_ID IN ('EMP0005' );
469
470 • DELETE FROM Empl_ID_Contact_Number WHERE Empl_ID = 'EMP0004';
471
472

```
- Output:**

#	Time	Action	Message	Duration / Fetch
1	02:10:21	UPDATE Event_Coordinator_Contact_Number SET Contact_No = '0712345678' WHERE Coordinator_ID IN (CT00...)	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
2	02:10:23	UPDATE Event_Coordinator_Contact_Number SET Contact_No = '0712345678' WHERE Coordinator_ID IN (CT00...)	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
3	02:10:25	DELETE FROM Event_Coordinator_Contact_Number WHERE Coordinator_ID = 'CD00006'	0 row(s) affected	0.000 sec
4	02:10:28	UPDATE Employee SET Birthday = '1993-12-31' WHERE Empl_ID IN (EMP0002, EMP0010)	2 row(s) affected Rows matched: 2 Changed: 2 Warnings: 0	0.000 sec
5	02:10:30	DELETE FROM Employee WHERE Empl_ID = 'EMP0004'	1 row(s) affected	0.000 sec
6	02:10:33	UPDATE Empl_ID_Contact_Number SET Contact_No = '+94712349499' WHERE Empl_ID IN (EMP0003)	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
7	02:10:35	UPDATE Empl_ID_Contact_Number SET Contact_No = '+94711689999' WHERE Empl_ID IN (EMP0005)	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
8	02:10:37	DELETE FROM Empl_ID_Contact_Number WHERE Empl_ID = 'EMP0004'	1 row(s) affected	0.000 sec
- Object Info:** Session
- Session Bar:** Type here to search, Windows Start button, Taskbar icons, Date: 4/9/2024, Time: 210 AM, Weather: 28°C.

Figure 3.5. 2: Update and delete in Event_Coordinator_Contact_Number, Employee and Empl_ID_Contact_Number tables

MySQL Workbench

Local instance MySQL83 - Workshop2

File Edit View Query Database Server Tools Scripting Help

Navigator: Event_Planning_Management

SCHEMAS: Filter objects

foe_23 sys

Event_Planning_Management

Workshop2

```

471
472
473 • UPDATE Music SET Charge_of_Event = '36000' WHERE Band_ID IN ('BT00003');
474 • UPDATE Music SET Charge_of_Event = '45000' WHERE Band_ID IN ('BT00005');
475
476 • DELETE FROM Music WHERE Band_ID = 'BT00007';
477
478
479 • UPDATE Music_Contact_Number SET Contact_No = '+94704581234' WHERE Band_ID IN ('BT00003','079 567 8901');
480 • UPDATE Music_Contact_Number SET Contact_No = '+94757894567' WHERE Band_ID IN ('BT00006','078 890 1234');
481
482 • DELETE FROM Music_Contact_Number WHERE Band_ID = 'BT00007';
483
484
485 • UPDATE Hotel SET Block_No = 'No:999/1' WHERE Hotel_Reg_No IN ('HB401');
486 • UPDATE Hotel SET Block_No = 'No:105/06' WHERE Hotel_Reg_No IN ('HB301');
487
488 • DELETE FROM Hotel WHERE Hotel_Reg_No = 'HB501';
489

```

Administration Schemas

Information

No object selected

Action Output

#	Time	Action	Message	Duration / Fetch
1	02:13:10	UPDATE Music SET Charge_of_Event = '36000' WHERE Band_ID IN ('BT00003')	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec
2	02:13:12	UPDATE Music SET Charge_of_Event = '45000' WHERE Band_ID IN ('BT00005')	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
3	02:13:15	DELETE FROM Music WHERE Band_ID = 'BT00007'	1 row(s) affected	0.015 sec
4	02:13:18	UPDATE Music_Contact_Number SET Contact_No = '+94704581234' WHERE Band_ID IN ('BT00003','079 567 8901')	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
5	02:13:21	UPDATE Music_Contact_Number SET Contact_No = '+94757894567' WHERE Band_ID IN ('BT00006','078 890 1234')	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
6	02:13:23	DELETE FROM Music_Contact_Number WHERE Band_ID = 'BT00007'	0 row(s) affected	0.000 sec
7	02:13:25	UPDATE Hotel SET Block_No = 'No:999/1' WHERE Hotel_Reg_No IN ('H0401')	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
8	02:13:28	UPDATE Hotel SET Block_No = 'No:105/06' WHERE Hotel_Reg_No IN ('H0301')	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
9	02:13:30	DELETE FROM Hotel WHERE Hotel_Reg_No = 'H0501'	1 row(s) affected	0.000 sec

Object Info Session

Type here to search

28°C 2:19 AM 4/9/2024

Figure 3.5. 3: Update and delete in Music, Music_Contact_Number and Hotel tables

MySQL Workbench

Local instance MySQL83 - Workshop2

File Edit View Query Database Server Tools Scripting Help

Navigator: Event_Planning_Management

SCHEMAS: Filter objects

foe_23 sys

Event_Planning_Management

Workshop2

```

489
490
491 • UPDATE Hotel_Contact_Number SET Contact_No = '+98 71 234 1547' WHERE Hotel_Reg_No IN ('HB301','079 567 7854');
492 • UPDATE Hotel_Contact_Number SET Contact_No = '+94 72 345 6781' WHERE Hotel_Reg_No IN ('HB501','077 345 6780');
493
494 • DELETE FROM Hotel_Contact_Number WHERE Hotel_Reg_No = 'HB401';
495
496
497 • UPDATE transportation SET charge_per_1km = '60' WHERE Vehicle_No IN ('TN00001');
498 • UPDATE transportation SET charge_per_1km = '65' WHERE Vehicle_No IN ('TN00005');
499
500 • DELETE FROM transportation WHERE Vehicle_No = 'TN00005';
501
502
503 • UPDATE Transportation_Contact_Number SET Contact_No = '+94 71 234 9999' WHERE Vehicle_No IN ('TN00001');
504 • UPDATE Transportation_Contact_Number SET Contact_No = '+94 71 234 4586' WHERE Vehicle_No IN ('TN00002');
505
506 • DELETE FROM Transportation_Contact_Number WHERE Vehicle_No = 'TN00003';
507

```

Administration Schemas

Information

No object selected

Action Output

#	Time	Action	Message	Duration / Fetch
1	02:18:39	UPDATE Hotel_Contact_Number SET Contact_No = '+98 71 234 1547' WHERE Hotel_Reg_No IN ('HB301','079 567 7854')	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
2	02:18:45	UPDATE Hotel_Contact_Number SET Contact_No = '+94 72 345 6781' WHERE Hotel_Reg_No IN ('HB501','077 345 6780')	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
3	02:18:58	DELETE FROM Hotel_Contact_Number WHERE Hotel_Reg_No = 'HB401'	1 row(s) affected	0.000 sec
4	02:19:01	UPDATE transportation SET charge_per_1km = '60' WHERE Vehicle_No IN ('TN00001')	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
5	02:19:03	UPDATE transportation SET charge_per_1km = '65' WHERE Vehicle_No IN ('TN00005')	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
6	02:19:05	DELETE FROM transportation WHERE Vehicle_No = 'TN00005'	0 row(s) affected	0.015 sec
7	02:19:09	UPDATE Transportation_Contact_Number SET Contact_No = '+94 71 234 9999' WHERE Vehicle_No IN ('TN00001')	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
8	02:19:13	UPDATE Transportation_Contact_Number SET Contact_No = '+94 71 234 4586' WHERE Vehicle_No IN ('TN00002')	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
9	02:19:15	DELETE FROM Transportation_Contact_Number WHERE Vehicle_No = 'TN00003'	0 row(s) affected	0.000 sec

Object Info Session

Type here to search

28°C 2:19 AM 4/9/2024

Figure 3.5. 4: Update and delete in Hotel_Contact_Number, transportation and Transportation_Contact_Number tables

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard icons for opening files, saving, running queries, and navigating.
- Navigator:** Shows SCHEMAS (foe_23, sys) and a list of objects under Event_Planning_Management.
- Workshop2 Tab:** Displays the following SQL code:


```

507
508
509 • UPDATE Decoration SET Contact_No = '0712345678' WHERE Deco_ID IN ('DC08047');
510 • UPDATE Decoration SET Contact_No = '0757845678' WHERE Deco_ID IN ('DC08049');
511
512 • DELETE FROM Decoration WHERE Deco_ID = 'DC08050';
513
514
515 • UPDATE Photography SET Charge_per_Person = '2000.00' WHERE Photgraphy_ID IN ('PH00003', 'PH00005');
516
517 • DELETE FROM Photography WHERE Photgraphy_ID = 'PH00006';
518
519
520 • UPDATE Photography_Contact_Number SET Contact_No = '+94 71 234 1452' WHERE Photgraphy_ID IN ('PHOT0084','079 567 8902');
521 • UPDATE Photography_Contact_Number SET Contact_No = '+94 71 785 1254' WHERE Photgraphy_ID IN ('PHOT0082','077 345 6788');
522
523 • DELETE FROM Photography_Contact_Number WHERE Photgraphy_ID = 'PHOT0085';
524
525

```
- Output Tab:** Shows the execution log with 8 entries, each with a timestamp, action, message, and duration.
- Object Info and Session Tabs:** Standard tabs for managing objects and sessions.
- System Bar:** Shows the date (4/9/2024), time (2:21 AM), and system status (28°C).

Figure 3.5. 5: Update and delete in Decoration, Photography and Photography_Contact_Number tables

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard icons for opening files, saving, running queries, and navigating.
- Navigator:** Shows SCHEMAS (foe_23, sys) and a list of objects under Event_Planning_Management.
- Workshop2 Tab:** Displays the following SQL code:


```

524
525
526 • UPDATE Catering SET Charge_per_Person = 600.00 WHERE Catering_ID IN ('CAT0003', 'CAT0005');
527
528 • DELETE FROM Catering WHERE Catering_ID = 'CAT0007';
529
530
531 • UPDATE Catering_Contact_Number SET Contact_No = '+94 71 785 7452' WHERE Catering_ID IN ('CAT0004','071 567 8901');
532 • UPDATE Catering_Contact_Number SET Contact_No = '+94 71 234 7821' WHERE Catering_ID IN ('CAT0004','074 567 1235');
533
534 • DELETE FROM Catering_Contact_Number WHERE Catering_ID = 'CAT0005','076 678 9012';
535
536
537 • UPDATE Booking SET Payment = 21000.00 WHERE Event_ID IN ('E_003');
538 • UPDATE Booking SET Payment = 21000.00 WHERE Event_ID IN ('E_006');
539
540 • DELETE FROM Booking WHERE Event_ID = 'E_007';
541
542
543

```
- Output Tab:** Shows the execution log with 16 entries, each with a timestamp, action, message, and duration.
- Object Info and Session Tabs:** Standard tabs for managing objects and sessions.
- System Bar:** Shows the date (4/9/2024), time (2:25 AM), and system status (28°C).

Figure 3.5. 6: Update and delete in Catering, Catering_Contact_Number and Booking tables

MySQL Workbench

Local instance MySQL8.3 - Workshop2

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

Filter objects

Event_Planning_Management_... Workshop2

Limit to 1000 rows

SQLAdditions

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

```

542
543
544 • UPDATE Clients SET Last_Name = 'Johnson' WHERE Client_ID IN ('C1_0005');
545 • UPDATE Clients SET Last_Name = 'Kiss' WHERE Client_ID IN ('C1_0003');
546
547 • DELETE FROM Clients WHERE Client_ID = 'C1_0006';
548 ...
549
550 • UPDATE Clients_Contact_Number SET Contact_No = '+94 71 234 7452' WHERE Client_ID IN ('C1_0007' '070 789 8123');
551 • UPDATE Clients_Contact_Number SET Contact_No = '+94 71 234 7452' WHERE Client_ID IN ('C1_0006' '076 123 9012');
552
553 • DELETE FROM Clients_Contact_Number WHERE Client_ID = 'C1_0005' '079 567 8901';
554 ...
555
556 • UPDATE Feedback SET Feedback_Comment = 'Create work' WHERE Client_ID IN ('C1_0002', 'C1_0003');
557
558 • DELETE FROM Feedback WHERE Client_ID = 'C1_0005';
559 ...
560
561

```

Administration Schemas

Information

No object selected

Action Output

#	Time	Action	Message	Duration / Fetch
1	02:26:58	UPDATE Clients SET Last_Name = 'Johnson' WHERE Client_ID IN ('C1_0005')	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
2	02:26:58	UPDATE Clients SET Last_Name = 'Kiss' WHERE Client_ID IN ('C1_0003')	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
3	02:26:58	DELETE FROM Clients WHERE Client_ID = 'C1_0006'	1 row(s) affected	0.000 sec
4	02:26:58	UPDATE Clients_Contact_Number SET Contact_No = '+94 71 234 7452' WHERE Client_ID IN ('C1_0007' '070 789 ...	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
5	02:26:58	UPDATE Clients_Contact_Number SET Contact_No = '+94 71 234 7452' WHERE Client_ID IN ('C1_0006' '076 123 ...	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
6	02:26:58	DELETE FROM Clients_Contact_Number WHERE Client_ID = 'C1_0005' '079 567 8901'	0 row(s) affected	0.000 sec
7	02:26:58	UPDATE Feedback SET Feedback_Comment = 'Create work' WHERE Client_ID IN ('C1_0002', 'C1_0003')	3 row(s) affected Rows matched: 3 Changed: 3 Warnings: 0	0.000 sec
8	02:26:58	DELETE FROM Feedback WHERE Client_ID = 'C1_0005'	0 row(s) affected	0.016 sec

Object Info Session

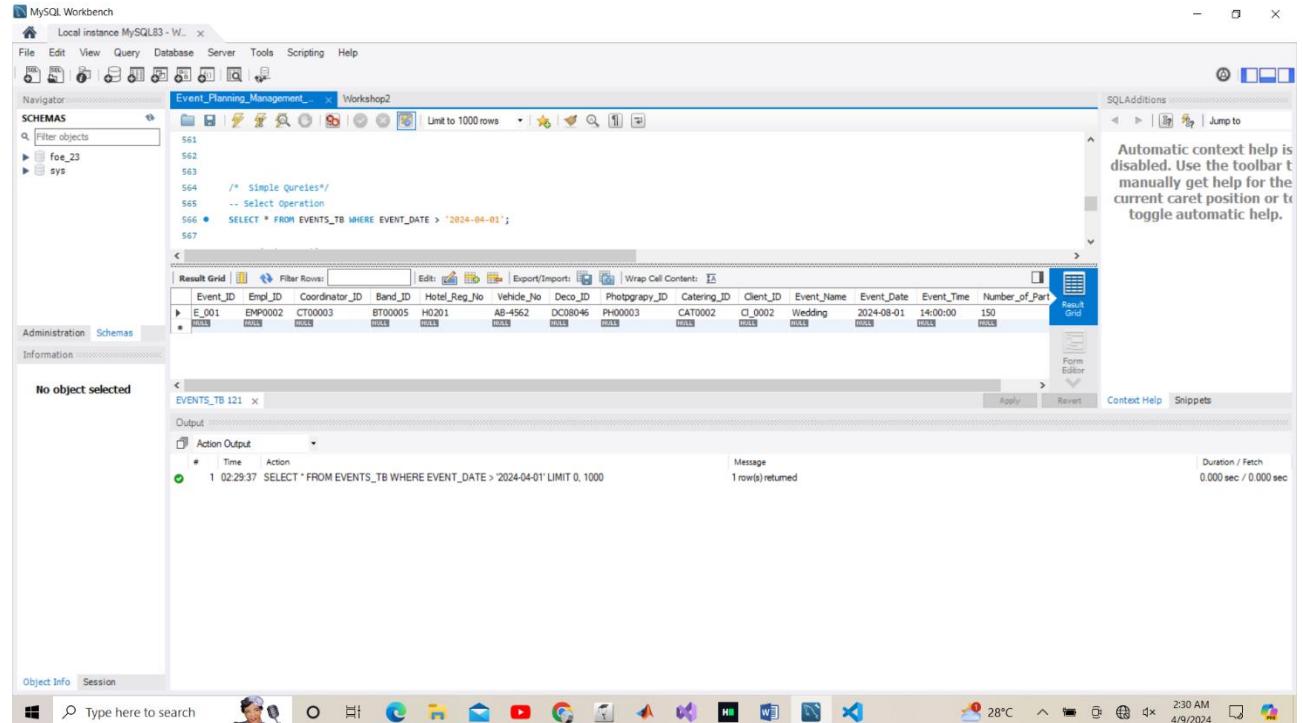
Type here to search

28°C 2:27 AM 4/9/2024

Figure 3.5. 7: Update and delete in Clients, Clients_Contact_Number and Feedback tables

4 Chapter 4 – Transactions

4.1 Simple Query



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

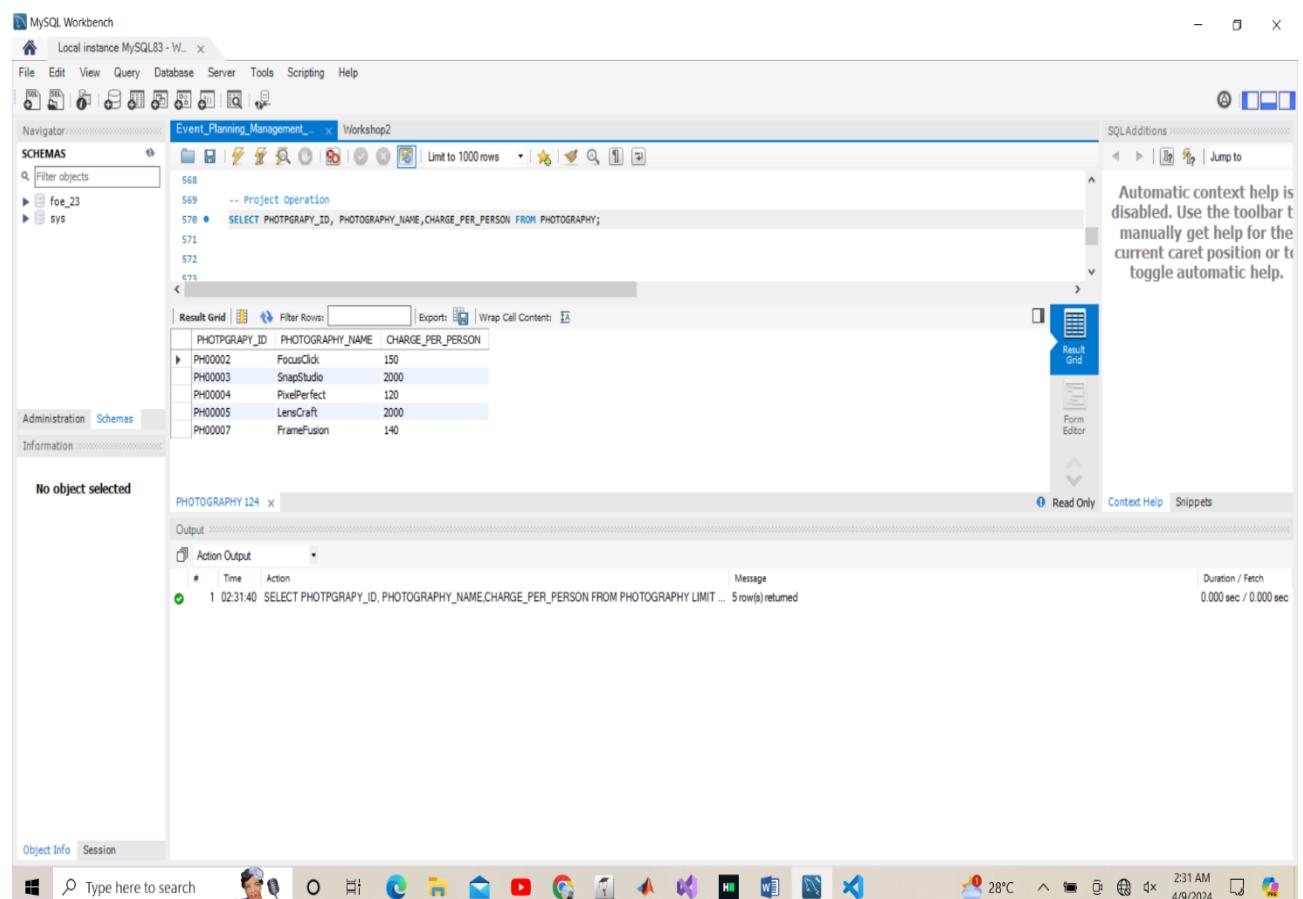
- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Event_Planning_Management_ (selected), foe_23, sys.
- SQL Editor:** Workshop2 tab, SQL code:

```
561
562
563
564    /* Simple queries*/
565    -- Select operation
566 •   SELECT * FROM EVENTS_TB WHERE EVENT_DATE > '2024-04-01';
567
```
- Result Grid:** Shows the results of the query:

Event_ID	Emp_ID	Coordinator_ID	Band_ID	Hotel_Reg_No	Vehicle_No	Deco_ID	Photography_ID	Catering_ID	Client_ID	Event_Name	Event_Date	Event_Time	Number_of_Part
E_001	EMP0002	CT00003	BT00005	H0201	AB-4562	DC08046	PH00003	CAT0002	O_0002	Wedding	2024-08-01	14:00:00	150
- Output:** Action Output table:

#	Time	Action
1	02:29:37	SELECT * FROM EVENTS_TB WHERE EVENT_DATE > '2024-04-01' LIMIT 0, 1000
- System Bar:** Type here to search, Taskbar icons, 28°C, 2:30 AM, 4/9/2024.

Figure 4.1. 1: Select Operation-Simple Query



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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Event_Planning_Management_ (selected), foe_23, sys.
- SQL Editor:** Workshop2 tab, SQL code:

```
568
569    -- Project Operation
570 •   SELECT PHOTOPGRAPHY_ID, PHOTOGRAPHY_NAME,CHARGE_PER_PERSON FROM PHOTOGRAPHY;
571
572
573
```
- Result Grid:** Shows the results of the query:

PHOTOPGRAPHY_ID	PHOTOGRAPHY_NAME	CHARGE_PER_PERSON
PH00002	FocusClick	150
PH00003	SnapStudio	2000
PH00004	PixelPerfect	120
PH00005	LensCraft	2000
PH00007	FrameFusion	140
- Output:** Action Output table:

#	Time	Action
1	02:31:40	SELECT PHOTOPGRAPHY_ID, PHOTOGRAPHY_NAME,CHARGE_PER_PERSON FROM PHOTOGRAPHY LIMIT ... 5 row(s) returned
- System Bar:** Type here to search, Taskbar icons, 28°C, 2:31 AM, 4/9/2024.

Figure 4.1. 2: Project Operation-Simple Query

MySQL Workbench

Local instance MySQL83 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator Event_Planning_Management_ Workshop2

SCHEMAS Schemas

Filter objects

foe_23 sys

571

572

573

574 -- Cartesian Product Operation

575 • SELECT * FROM CLIENTS ,EVENTS_TB;

576

Result Grid Filter Rows Export Wrap Cell Content

Client_ID	First_Name	Last_Name	Block_No	Street	City	Event_ID	Emp_ID	Coordinator_ID	Band_ID	Hotel_Reg_No	Vehicle_No	Deco_ID	Photography_ID
C1_0002	Alice	Johnson	22/4	Main Street	Colombo	E_004	EMP0004	CT00005	BT00002	H0401	GH-5678	DC08049	PH00004
C1_0002	Alice	Johnson	22/4	Main Street	Colombo	E_002	EMP0003	CT00006	BT00002	H0301	CD-2156	DC08049	PH00003
C1_0002	Alice	Johnson	22/4	Main Street	Colombo	E_001	EMP0002	CT00003	BT00005	H0201	AB-4562	DC08046	PH00003
C1_0003	Robert	Kriss	19/9	Beach Road	Galle	E_004	EMP0004	CT00005	BT00002	H0401	GH-5678	DC08049	PH00004
C1_0003	Robert	Kriss	19/9	Beach Road	Galle	E_002	EMP0003	CT00006	BT00002	H0301	CD-2156	DC08049	PH00003
C1_0003	Robert	Kriss	19/9	Beach Road	Galle	E_001	EMP0002	CT00003	BT00005	H0201	AB-4562	DC08046	PH00003
C1_0004	Emily	Davis	11	Hillside Drive	Kandy	E_004	EMP0004	CT00005	BT00002	H0401	GH-5678	DC08049	PH00004

No object selected

Result 126 x

Output Action Output

#	Time	Action	Message	Duration / Fetch
1	02:31:57	SELECT * FROM CLIENTS ,EVENTS_TB LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

28°C 232 AM 4/9/2024

Figure 4.1. 3: Cartesian Product Operation-Simple Query

MySQL Workbench

Local instance MySQL83 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator Event_Planning_Management_ Workshop2

SCHEMAS Schemas

Filter objects

foe_23 sys

576

577 -- Creating a User View

578 • CREATE VIEW BOOKING_DETAILS AS SELECT CLIENT_ID, Payment, BOOKING_DATE FROM BOOKING;

579 • SELECT * FROM BOOKING_DETAILS;

580

581

BOOKING_DETAILS 127 x

Result Grid Filter Rows Export Wrap Cell Content

CLIENT_ID	Payment	BOOKING_DATE
C1_0002	15000	2024-04-03
C1_0003	20000	2024-04-04
C1_0004	21000	2024-04-05
C1_0005	25000	2024-04-06
C1_0007	21000	2024-04-08

No object selected

Output Action Output

#	Time	Action	Message	Duration / Fetch
1	02:32:16	CREATE VIEW BOOKING_DETAILS AS SELECT CLIENT_ID, Payment, BOOKING_DATE FROM BOOKING	0 row(s) affected	0.000 sec
2	02:32:16	SELECT * FROM BOOKING_DETAILS LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

28°C 232 AM 4/9/2024

Figure 4.1. 4: Creating a user view-Simple Query

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Event_Planning_Management_
- SQL Editor:** A query window titled "Workshop2" containing the following SQL code:


```

580
581
582 -- Renaming Operation
583 • SELECT EMPL_ID AS ID , FIRST_NAME AS F_NAME , LAST_NAME AS L_NAME, Coordinator_ID AS C_ID FROM EMPLOYEE;
584
585
      
```
- Result Grid:** Shows a table with columns ID, F_NAME, L_NAME, and C_ID. The data is:

ID	F_NAME	L_NAME	C_ID
EMP0002	David	Brown	CT00006
EMP0003	Alice	Smith	CT00001
EMP0010	Frank	Davis	CT00007
- Output Window:** Titled "EMPLOYEE 128", it shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	02:32:37	SELECT EMPL_ID AS ID , FIRST_NAME AS F_NAME , LAST_NAME AS L_NAME, Coordinator_ID AS C_ID FR...	3 row(s) returned	0.000 sec / 0.000 sec
- System Tray:** Shows icons for various applications and system status (28°C, 232 AM, 4/9/2024).

Figure 4.1. 5: Renaming Operation-Simple Query

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Event_Planning_Management_
- SQL Editor:** A query window titled "Workshop2" containing the following SQL code:


```

584
585
586 -- Aggregation Function
587 • SELECT AVG(RATING) AS AVERAGE_RATING FROM RATING;
588
      
```
- Result Grid:** Shows a table with one row labeled "AVERAGE_RATING" containing the value "5.000".
- Output Window:** Titled "Result 129", it shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	02:32:55	SELECT AVG(RATING) AS AVERAGE_RATING FROM RATING LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
- System Tray:** Shows icons for various applications and system status (28°C, 233 AM, 4/9/2024).

Figure 4.1. 6: Use of an aggregation function-Simple Query

The screenshot shows the MySQL Workbench interface. In the top-left corner, it says "MySQL Workbench" and "Local instance MySQL8.3 - W...". The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons. The main workspace has a tab titled "Event_Planning_Management_ Workshop2". The left sidebar shows "Schemas" with "foe_23" and "sys" selected. The right sidebar has a "SQLAdditions" section with a note about automatic context help being disabled. The bottom right shows system status: 28°C, 233 AM, 4/9/2024.

Event_Planning_Management_ Workshop2

Schemas

No object selected

TRANSPORTATION 130

Action Output

#	Time	Action	Message	Duration / Fetch
1	02:33:22	SELECT * FROM TRANSPORTATION WHERE TYPE_OF_VEHICLE LIKE 'LIMOUSINE' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Result Grid

Form Editor

Context Help Snippets

Object Info Session

Type here to search

28°C 233 AM 4/9/2024

Figure 4.1. 7: Use of LIKE keyword-Simple Query

4.2 Complex Query

The screenshot shows the MySQL Workbench interface with a query editor window titled "Event_Planning_Management_<-- Workshop2". The code pane contains the following SQL:

```

596
597 -- Complex Queries:
598
599 -- B1 Union Operation: Combining results from two queries
600 • SELECT Event_Name AS Title FROM Events_TB
601 UNION
602 SELECT Catering_Name AS Title FROM Catering;
603

```

The result grid shows the combined results:

Title
Wedding
Birthday_Party
Conference
ABC Catering
XYZ Catering
Best Caterers
Delicious Catering
Elite Caterers
dimo Caterers

The status bar at the bottom right indicates "Duration / Fetch 0.016 sec / 0.000 sec".

Figure 4.2. 1: Aliasing with union Operation-Complex Query

The screenshot shows the MySQL Workbench interface with a query editor window titled "Event_Planning_Management_<-- Workshop2". The code pane contains the following SQL:

```

603
604 -- B2 Intersection: Finding common values between two queries
605
606 • SELECT Client_ID AS cf FROM Feedback
607 INTERSECT
608 • SELECT Client_ID AS c FROM clients;
609
610

```

The result grid shows the common client IDs:

cf
CI_0002
CI_0004
CI_0005
CI_0003
CI_0007

The status bar at the bottom right indicates "Duration / Fetch 0.000 sec / 0.000 sec".

Figure 4.2. 2: Aliasing with intersection Operation-Complex Query

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Schemas dropdown showing 'foe_23' and 'sys'.
- Query Editor:** Title: 'Event_Planning_Management - Workshop2'. The code is:

```

609
610    -- 03 Set Difference: Showing values from the first query that are not in the second query
611
612 •   SELECT Event_Name AS Title FROM Events_TB
613 ✘ EXCEPT
614     SELECT Catering_Name AS Title FROM Catering;
615
616

```
- Result Grid:** Shows results for 'Title' with rows: Birthday_Party, Wedding, Conference.
- Output Panel:** Action Output shows a single row returned: 1 row(s) returned.
- System Taskbar:** Shows the date and time as 4/9/2024 254 AM.

Figure 4.2. 3: Aliasing with set difference Operation-Complex Query

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Schemas dropdown showing 'foe_23' and 'sys'.
- Query Editor:** Title: 'Event_Planning_Management - Workshop2'. The code is:

```

616
617    -- 04 Division: Finding items in the first table that have relationships with all items in the second table
618    -- Assuming there's a relationship between Events_TB and Catering, let's find events that use all available caterers.
619 •   SELECT Event_Name FROM Events_TB AS E
620     WHERE NOT EXISTS (
621         SELECT Catering_ID FROM Catering AS C
622         WHERE NOT EXISTS (
623             SELECT * FROM Booking B
624             WHERE B.Event_ID = E.Event_ID AND B.Payment > C.Charge_per_Person
625         )
626     );
627

```
- Result Grid:** Shows results for 'Event_Name' with rows: Wedding, Birthday_Party, Conference.
- Output Panel:** Action Output shows a single row returned: 1 row(s) returned.
- System Taskbar:** Shows the date and time as 4/9/2024 255 AM.

Figure 4.2. 4: Aliasing with division Operation-Complex Query

The screenshot shows the MySQL Workbench interface with a query editor window titled "Event_Planning_Management_> Workshop2". The code is as follows:

```

627
628
629 -- 05 Inner join to create a view Event_Clients joining Events_TB and Clients
630 -- Creating a User View Event_Clients by Inner Join
631 • CREATE VIEW EventClients AS
632     SELECT ET.Event_ID, C.Client_ID, ET.Event_Name, C.First_Name, C.Last_Name
633     FROM Events_TB AS ET
634     INNER JOIN Clients C ON ET.Client_ID = C.Client_ID;
635 •     SELECT Event_Name, First_Name
636     FROM Eventclients
637     WHERE Event_Name LIKE 'C%';

```

The results grid shows one row:

Event_Name	First_Name
Conference	Emily

The output pane shows the execution log:

Action	Time	Message	Duration / Fetch
CREATE VIEW EventClients AS SELECT ET.Event_ID, C.Client_ID, ET.Event_Name, C.First_Name, C.Last_Name...	02:57:09	0 rows affected	0.016 sec
SELECT Event_Name, First_Name FROM EventClients WHERE Event_Name LIKE 'C%' LIMIT 0, 1000	02:57:09	1 row(s) returned	0.000 sec / 0.000 sec

Figure 4.2. 5: Aliasing with inner join Operation-Complex Query

The screenshot shows the MySQL Workbench interface with a query editor window titled "Event_Planning_Management_> Workshop2". The code is as follows:

```

639
640 -- 06 Natural join between Event_Clients view and Feedback table to get feedback for each event
641 -- Create the Event_Feedback view using Natural Join
642 • CREATE VIEW Event_Feedback AS
643     SELECT EC.Event_ID, EC.Event_Name, EC.Coordinator_ID, F.Feedback_Comment
644     FROM events_tb AS EC
645     NATURAL JOIN Feedback AS F;
646 -- Selecting data from the Event_Feedback view
647 •     SELECT Event_ID, Event_Name, Feedback_Comment
648     FROM Event_Feedback WHERE Event_Name LIKE 'C%';

```

The results grid shows one row:

Event_ID	Event_Name	Feedback_Comment
E_004	Conference	The band performance was fantastic!

The output pane shows the execution log:

Action	Time	Message	Duration / Fetch
CREATE VIEW Event_Feedback AS SELECT EC.Event_ID, EC.Event_Name, EC.Coordinator_ID, F.Feedback_C...	02:57:22	0 rows affected	0.000 sec
SELECT Event_ID, Event_Name, Feedback_Comment FROM Event_Feedback WHERE Event_Name LIKE 'C%'...	02:57:22	1 row(s) returned	0.015 sec / 0.000 sec

Figure 4.2. 6: Aliasing with natural join Operation-Complex Query

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL83 - Workshop2, Schemas (foe_23, sys).
- SQL Editor:** Event_Planning_Management_Workshop2 tab, displaying a complex SQL script. The script includes several CREATE VIEW statements and a LEFT OUTER JOIN clause. Lines 653-662 show the creation of a view named Events_TB_View and a Booking_View, followed by a LEFT OUTER JOIN statement.


```

651 ... left
652 ... 87 LEFT JOIN with aliases
653 ● CREATE VIEW Events_TB_View AS
654   SELECT Event_ID, Event_Name
655   FROM events_tb;
656 ● CREATE VIEW Booking_View AS
657   SELECT Event_ID, Payment, Booking_Date
658   FROM Booking;
659
660 ●   SELECT EC.Event_ID, EC.Event_Name, B.Payment, B.Booking_Date
661   FROM Events_TB_View as EC
662   LEFT OUTER JOIN Booking_View B ON EC.Event_ID = B.Event_ID where EC.Event_Name like 'c%';
663
664

```
- Result Grid:** Shows a single row of data from the query:

Event_ID	Event_Name	Payment	Booking_Date
E_004	Conference	25000	2024-04-06
- Output Tab:** Action Output table showing the execution log:

#	Time	Action	Message	Duration / Fetch
1	02:57:58	CREATE VIEW Events_TB_View AS SELECT Event_ID, Event_Name FROM events_tb	0 row(s) affected	0.016 sec
2	02:57:58	CREATE VIEW Booking_View AS SELECT Event_ID, Payment, Booking_Date FROM Booking	0 row(s) affected	0.000 sec
3	03:16:30	SELECT EC.Event_ID, EC.Event_Name, B.Payment, B.Booking_Date FROM Events_TB_View as EC LEFT OUT...	1 row(s) returned	0.000 sec / 0.000 sec
- System Bar:** Object Info, Session, Taskbar with various icons, system status (28°C), and date/time (3:17 AM 4/9/2024).

Figure 4.2. 7: Aliasing with left outer join Operation-Complex Query

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL83 - Workshop2, Schemas (foe_23, sys).
- SQL Editor:** Event_Planning_Management_Workshop2 tab, displaying a complex SQL script. The script includes several CREATE VIEW statements and a RIGHT OUTER JOIN clause. Lines 668-678 show the creation of a view named Event_Booking and a SELECT statement using a RIGHT OUTER JOIN.


```

663
664
665
666 ... right
667 ... 88 RIGHT JOIN with aliases
668 ... Create the Event_Booking view using Left Outer Join
669 ● CREATE VIEW Event_Booking AS
670   SELECT EC.Event_ID, EC.Event_Name, EC.Client_ID, B.Payment, B.Booking_Date
671   FROM events_tb EC
672   LEFT OUTER JOIN Booking B ON EC.Event_ID = B.Event_ID ;
673   -- Selecting data from the Event_Booking view
674 ●   SELECT Event_ID, Event_Name, Payment, Booking_Date
675   FROM Event_Booking where Payment like '1%';
676
677

```
- Result Grid:** Shows a single row of data from the query:

Event_ID	Event_Name	Payment	Booking_Date
E_001	Wedding	15000	2024-04-03
- Output Tab:** Action Output table showing the execution log:

#	Time	Action	Message	Duration / Fetch
1	03:17:42	CREATE VIEW Event_Booking AS SELECT EC.Event_ID, EC.Event_Name, EC.Client_ID, B.Payment, B.Booking...	0 row(s) affected	0.015 sec
2	03:17:42	SELECT Event_ID, Event_Name, Payment, Booking_Date FROM Event_Booking where Payment like '1%' LIMIT ...	1 row(s) returned	0.000 sec / 0.000 sec
- System Bar:** Object Info, Session, Taskbar with various icons, system status (28°C), and date/time (3:17 AM 4/9/2024).

Figure 4.2. 8: Aliasing with right outer join Operation-Complex Query

MySQL Workbench

Local instance MySQL83 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator Event_Planning_Management_... Workshop2

SCHEMAS Schemas

No object selected

```

677
678    -- 09 Full outer
679    -- LEFT JOIN
680    -- Create the Event_Booking_Union view using Union of Left and Right Outer Joins with Aliases
681 • CREATE VIEW Event_Booking_Union AS
682 (
683     SELECT EC.Event_ID, EC.Event_Name, EC.Client_ID, B.Payment, B.Booking_Date
684     FROM events_tb AS EC
685     LEFT OUTER JOIN Booking AS B ON EC.Event_ID = B.Event_ID
686
687     UNION
688
689     SELECT EC.Event_ID, EC.Event_Name, EC.Client_ID, B.Payment, B.Booking_Date
690     FROM Booking AS B
691     RIGHT OUTER JOIN events_tb AS EC ON EC.Event_ID = B.Event_ID
692 );
693    -- Selecting data from the Event_Booking_Union view
694 • SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date
695 FROM Event_Booking_Union where Event_Name like 'C%';
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711

```

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

Event_ID	Event_Name	Client_ID	Payment	Booking_Date
E_004	Conference	CI_0004	25000	2024-04-06

Event_Booking_Union 173 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	03:20:05	CREATE VIEW Event_Booking_Union AS (SELECT EC.Event_ID, EC.Event_Name, EC.Client_ID, B.Payment,... 0 rows affected	0.016 sec
2	03:20:05	SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date FROM Event_Booking_Union where Event_... 1 row(s) returned		0.015 sec / 0.000 sec

Object Info Session

Type here to search

Figure 4.2. 9: Aliasing with full outer join Operation-Complex Query

MySQL Workbench

Local instance MySQL83 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator Event_Planning_Management_... Workshop2

SCHEMAS Schemas

No object selected

```

698
699    -- 10 Outer Union Example
700    -- Create the Outer_Union_View view using Outer Union
701 • CREATE VIEW Outer_Union_View AS
702 (
703     SELECT Client_ID FROM Clients
704
705     UNION
706
707     SELECT Client_ID FROM Booking
708 );
709    -- Selecting data from the Outer_Union_View view
710 • SELECT * FROM Outer_Union_View ;
711

```

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

Event_ID	Event_Name	Client_ID	Payment	Booking_Date
E_004	Conference	CI_0004	25000	2024-04-06

Event_Booking_Union 173 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	03:20:05	CREATE VIEW Event_Booking_Union AS (SELECT EC.Event_ID, EC.Event_Name, EC.Client_ID, B.Payment,... 0 rows affected	0.016 sec
2	03:20:05	SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date FROM Event_Booking_Union where Event_... 1 row(s) returned		0.015 sec / 0.000 sec

Object Info Session

Type here to search

Figure 4.2. 10: Aliasing with outer union Operation-Complex Query

The screenshot shows the MySQL Workbench interface with a query editor window titled "Event_Planning_Management_> Workshop2". The query being run is:

```

711
712
713 -- 11. Nested Query with Union:
714 -- Total number of participants in events and total payment made in bookings
715 • SELECT Event_Name, Number_of_Participant AS Quantity FROM Events_TB where Number_of_Participant<100
716 UNION
717 SELECT 'Booking Total', SUM(Payment) AS Quantity FROM Booking;
718
719
    
```

The results grid shows:

Event_Name	Quantity
Booking Total	102000

The "Result 174" tab in the output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	03:21:24	SELECT Event_Name, Number_of_Participant AS Quantity FROM Events_TB where Number_of_Participant<100 ...	1 row(s) returned	0.000 sec / 0.000 sec

Figure 4.2. 11: Nested-Complex Query

The screenshot shows the MySQL Workbench interface with a query editor window titled "Event_Planning_Management_> Workshop2". The query being run is:

```

719
720 -- 12. Nested Query with Intersection:
721 -- Clients who have provided feedback and also made bookings
722 • SELECT Client_ID FROM Feedback
723 INTERSECT
724 • SELECT Client_ID FROM Booking;
725
    
```

The results grid shows:

Client_ID
C1_0002
C1_0004
C1_0005
C1_0003
C1_0007

The "Result 175" tab in the output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	03:21:50	SELECT Client_ID FROM Feedback INTERSECT SELECT Client_ID FROM Booking	5 row(s) returned	0.000 sec / 0.000 sec

Figure 4.2. 12: Nested-Complex Query

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

- Title Bar:** MySQL Workbench - Local instance MySQL8.0 - Workshop2
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help
- Toolbar:** Includes icons for Home, New, Open, Save, Print, Copy, Paste, Find, Replace, Undo, Redo, and others.
- Schemas Navigator:** Shows SCHEMAS (foe_23, sys) and a Filter objects search bar.
- Query Editor:** Displays the following SQL code:

```
726
727  -- 13. Nested Query with Set Difference:
728  -- Events where number of participants exceeds average
729  SELECT Event_Name FROM Events_TB
730  WHERE Number_of_Participant > (
731      SELECT AVG(Number_of_Participant) FROM Events_TB
732  )
733
```
- Result Grid:** Shows the output of the query with one row: Event_Name (Conference).
- Output Window:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	03:22:12	SELECT Event_Name FROM Events_TB WHERE Number_of_Participant > (SELECT AVG(Number_of_Participant) FROM Events_TB) returned 1 rows(s)	0.000 sec / 0.000 sec
- System Tray:** Shows the Windows taskbar with various pinned icons (File Explorer, Edge, Mail, YouTube, Google Chrome, etc.) and system status (28°C, 3:22 AM, 4/9/2024).

Figure 4.2. 13: Nested-Complex Query

5 Chapter 5 - Database Tuning

```

738 -- before - 01
739 • explain SELECT Event_Name, First_Name
740   FROM EventClients
741   WHERE Event_Name LIKE 'C%';

743 -- Add an index to the Event_Name column
744 • CREATE INDEX idx_event_name ON Events_TB(Event_Name);
745 • Drop index idx_event_name ON Events_TB;

747 -- after
748 • explain SELECT Event_Name, First_Name
749   FROM EventClients
750   WHERE Event_Name LIKE 'C%';

```

Result Grid:

ID	Select_Type	Table	Partitions	Type	Possible_Keys	Key	Key_Len	Ref	Rows	Filtered	Extra
1	SIMPLE	et	NULL	ALL	Client_ID	NULL	NULL	NULL	3	33.33	Using where
1	SIMPLE	c	NULL	eq_ref	PRIMARY	PRIMARY	50	event_planning_management_system.et.Client...	1	100.00	NULL

Action Output:

#	Time	Action	Message	Duration / Fetch
11	12:53:53	explain SELECT Event_Name FROM Events_TB WHERE Number_of_Participant > (SELECT AVG(Number_...	2 row(s) returned	0.000 sec / 0.000 sec
12	12:55:43	explain SELECT Event_Name, First_Name FROM EventClients WHERE Event_Name LIKE C%	2 row(s) returned	0.000 sec / 0.000 sec
13	12:56:10	explain SELECT Event_Name, First_Name FROM EventClients WHERE Event_Name LIKE C%	2 row(s) returned	0.000 sec / 0.000 sec
14	12:57:02	Drop index idx_event_name ON Events_TB;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
15	12:57:07	explain SELECT Event_Name, First_Name FROM EventClients WHERE Event_Name LIKE C%	2 row(s) returned	0.016 sec / 0.000 sec

Figure 5. 1: Before Tuning

```

738 -- before - 01
739 • explain SELECT Event_Name, First_Name
740   FROM EventClients
741   WHERE Event_Name LIKE 'C%';

743 -- Add an index to the Event_Name column
744 • CREATE INDEX idx_event_name ON Events_TB(Event_Name);
745 • Drop index idx_event_name ON Events_TB;

747 -- after
748 • explain SELECT Event_Name, First_Name
749   FROM EventClients
750   WHERE Event_Name LIKE 'C%';

```

Result Grid:

ID	Select_Type	Table	Partitions	Type	Possible_Keys	Key	Key_Len	Ref	Rows	Filtered	Extra
1	SIMPLE	et	NULL	range	Client_ID, idx_event_name	idx_event_name	203	NULL	1	100.00	Using index condition
1	SIMPLE	c	NULL	eq_ref	PRIMARY	PRIMARY	50	event_planning_management_system.et.Client...	1	100.00	NULL

Action Output:

#	Time	Action	Message	Duration / Fetch
13	12:56:10	explain SELECT Event_Name, First_Name FROM EventClients WHERE Event_Name LIKE C%	2 row(s) returned	0.000 sec / 0.000 sec
14	12:57:02	Drop index idx_event_name ON Events_TB;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
15	12:57:07	explain SELECT Event_Name, First_Name FROM EventClients WHERE Event_Name LIKE C%	2 row(s) returned	0.016 sec / 0.000 sec
16	12:57:46	CREATE INDEX idx_event_name ON Events_TB(Event_Name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.046 sec
17	12:58:05	explain SELECT Event_Name, First_Name FROM EventClients WHERE Event_Name LIKE C%	2 row(s) returned	0.000 sec / 0.000 sec

Figure 5. 2: After Tuning

```

MySQL Workbench
Local instance MySQL83 - W...
File Edit View Query Database Server Tools Scripting Help
Navigator Event_Planning_Management Workshop2
schemas
Filter objects
foe_23 sys
743 -- Add an index to the Event_Name column
744 • CREATE INDEX idx_event_name ON Events_TB(Event_Name);
745 • Drop index idx_event_name ON Events_TB;
746
747 -- after
748 • explain SELECT Event_Name, First_Name
    FROM EventClients
    WHERE Event_Name LIKE 'C%';
749
750 -- -----
751 -- before
752
753
754
755 • explain SELECT Event_ID, Event_Name, Feedback_Comment
    FROM Event_Feedback WHERE Event_Name LIKE 'C%';
756

```

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

ID	Select_Type	Table	Partitions	Type	Possible_Keys	Key	Key_Len	Ref	Rows	Filtered	Extra
1	SIMPLE	ec	NULL	ALL	Client_ID	NULL	NULL	NULL	3	33.33	Using where
1	SIMPLE	f	NULL	ref	Client_ID	Client_ID	51	event_planning_management_system.ec.Client...	1	100.00	NULL

Action Output

#	Time	Action	Message	Duration / Fetch
15	12:57:07	explain SELECT Event_Name, First_Name FROM EventClients WHERE Event_Name LIKE 'C%'	2 row(s) returned	0.016 sec / 0.000 sec
16	12:57:46	CREATE INDEX idx_event_name ON Events_TB(Event_Name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.046 sec
17	12:58:05	explain SELECT Event_Name, First_Name FROM EventClients WHERE Event_Name LIKE 'C%'	2 row(s) returned	0.000 sec / 0.000 sec
18	12:59:08	Drop index idx_event_name ON Events_TB	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
19	12:59:21	explain SELECT Event_ID, Event_Name, Feedback_Comment FROM Event_Feedback WHERE Event_Name L...	2 row(s) returned	0.000 sec / 0.000 sec

Figure 5. 3: Before Tuning

```

MySQL Workbench
Local instance MySQL83 - W...
File Edit View Query Database Server Tools Scripting Help
Navigator Event_Planning_Management Workshop2
schemas
Filter objects
foe_23 sys
749 FROM EventClients
750 WHERE Event_Name LIKE 'C%';
751
752 -- -----
753 -- before
754
755 • explain SELECT Event_ID, Event_Name, Feedback_Comment
    FROM Event_Feedback WHERE Event_Name LIKE 'C%';
756
757 -- - CREATE INDEX idx_event_name ON Event_Feedback(Event_ID,Event_Name); // this is not a base table we can't add indexes
758
759 -- -----
760 -- before
761
762 • DROP INDEX idx_event_name ON Events_TB ;

```

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

ID	Select_Type	Table	Partitions	Type	Possible_Keys	Key	Key_Len	Ref	Rows	Filtered	Extra
1	SIMPLE	ec	NULL	ALL	Client_ID	NULL	NULL	NULL	3	33.33	Using where
1	SIMPLE	f	NULL	ref	Client_ID	Client_ID	51	event_planning_management_system.ec.Client...	1	100.00	NULL

Action Output

#	Time	Action	Message	Duration / Fetch
15	12:57:07	explain SELECT Event_Name, First_Name FROM EventClients WHERE Event_Name LIKE 'C%'	2 row(s) returned	0.016 sec / 0.000 sec
16	12:57:46	CREATE INDEX idx_event_name ON Events_TB(Event_Name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.046 sec
17	12:58:05	explain SELECT Event_Name, First_Name FROM EventClients WHERE Event_Name LIKE 'C%'	2 row(s) returned	0.000 sec / 0.000 sec
18	12:59:08	Drop index idx_event_name ON Events_TB	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
19	12:59:21	explain SELECT Event_ID, Event_Name, Feedback_Comment FROM Event_Feedback WHERE Event_Name L...	2 row(s) returned	0.000 sec / 0.000 sec

Figure 5. 4: No more Tuning

MySQL Workbench

Local instance MySQL83 - Workshop2

File Edit View Query Database Server Tools Scripting Help

Navigator: Event_Planning_Management

SCHEMAS: Filter objects: fo_e23 sys

Limit to 1000 rows

```

755 • explain SELECT Event_ID, Event_Name, Feedback_Comment
756   FROM Event_Feedback WHERE Event_Name LIKE 'ck%';
757
758 -- - CREATE INDEX idx_event_name ON Event_Feedback(Event_ID,Event_Name); // this is not a base table we can't add indexes
759
760 -- -----
761 -- before
762 • DROP INDEX idx_event_name ON Events_TB ;
763 • explain SELECT EC.Event_ID, EC.Event_Name, B.Payment, B.Booking_Date
764   FROM Events_TB_View as EC
765   LEFT OUTER JOIN Booking_View B ON EC.Event_ID = B.Event_ID where EC.Event_Name like 'ck%';
766
767 -- after
768

```

No object selected

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

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	SIMPLE	events_tb	NULL	ALL	NULL	NULL	NULL	NULL	3	33.33	Using where
1	SIMPLE	booking	NULL	ref	idx_event_id	idx_event_id	50	event_planning_management_system.events_t...	1	100.00	NULL

Result 62 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
18	12:59:08	Drop index idx_event_name ON Events_TB	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
19	12:59:21	explain SELECT Event_ID, Event_Name, Feedback_Comment FROM Event_Feedback WHERE Event_Name L...	2 row(s) returned	0.000 sec / 0.000 sec
20	13:01:10	DROP INDEX idx_event_name ON Events_TB	Error Code: 1091. Can't DROP 'idx_event_name'; check that column/key exists	0.015 sec
21	13:01:19	DROP INDEX idx_event_name ON Events_TB	Error Code: 1091. Can't DROP 'idx_event_name'; check that column/key exists	0.000 sec
22	13:01:30	explain SELECT EC.Event_ID, EC.Event_Name, B.Payment, B.Booking_Date FROM Events_TB_View as EC L...	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

Figure 5. 5: Before Tuning

MySQL Workbench

Local instance MySQL83 - Workshop2

File Edit View Query Database Server Tools Scripting Help

Navigator: Event_Planning_Management

SCHEMAS: Filter objects: fo_e23 sys

Limit to 1000 rows

```

-- before
762 • D Execute the selected portion of the script or everything, if there is no selection
763 • explain SELECT EC.Event_ID, EC.Event_Name, B.Payment, B.Booking_Date
764   FROM Events_TB_View as EC
765   LEFT OUTER JOIN Booking_View B ON EC.Event_ID = B.Event_ID where EC.Event_Name like 'ck%';
766
767 -- after
768
769 • CREATE INDEX idx_event_name ON Events_TB(Event_Name);
770 • explain SELECT EC.Event_ID, EC.Event_Name, B.Payment, B.Booking_Date
771   FROM Events_TB_View as EC
772   LEFT OUTER JOIN Booking_View B ON EC.Event_ID = B.Event_ID where EC.Event_Name like 'ck%';
773
774 -----

```

No object selected

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

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	SIMPLE	events_tb	NULL	range	idx_event_name	idx_event_name	203	NULL	1	100.00	Using where; Using index
1	SIMPLE	booking	NULL	ref	idx_event_id	idx_event_id	50	event_planning_management_system.events_t...	1	100.00	NULL

Result 63 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
20	13:01:10	DROP INDEX idx_event_name on Events_TB	Error Code: 1091. Can't DROP 'idx_event_name'; check that column/key exists	0.015 sec
21	13:01:19	DROP INDEX idx_event_name on Events_TB	Error Code: 1091. Can't DROP 'idx_event_name'; check that column/key exists	0.000 sec
22	13:01:30	explain SELECT EC.Event_ID, EC.Event_Name, B.Payment, B.Booking_Date FROM Events_TB_View as EC L...	2 row(s) returned	0.000 sec / 0.000 sec
23	13:02:14	CREATE INDEX idx_event_name ON Events_TB(Event_Name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.046 sec
24	13:02:21	explain SELECT EC.Event_ID, EC.Event_Name, B.Payment, B.Booking_Date FROM Events_TB_View as EC L...	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

Figure 5. 6: After Tuning

```

File Edit View Query Database Server Tools Scripting Help
Navigator: Event_Planning_Management x Workshop2
SCHEMAS Filter objects
foe_23 sys
Limit to 1000 rows
770 • explain SELECT EC.Event_ID, EC.Event_Name, B.Payment, B.Booking_Date
771 FROM Events_TB_View as EC
772 LEFT OUTER JOIN Booking_View B ON EC.Event_ID = B.Event_ID where EC.Event_Name like 'ck';
773
774 -- -----
775 -- before
776 • explain SELECT Event_ID, Event_Name, Payment, Booking_Date
777 FROM Event_Booking where Payment like '1%';
778
779 • create index Bookinh_price on booking(Payment);
780 • drop index Bookinh_price on booking ;
781
782 • explain SELECT Event_ID, Event_Name, Payment, Booking_Date
783 FROM Event_Booking where Payment like '1%';
784
785 -- -----
786 -- before

```

No object selected

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows			filtered	Extra
									rows	filtered	Using where		
1	SIMPLE	b	NULL	ALL	idx_event_id	NULL	NULL	NULL	5	20.00	Using where		
1	SIMPLE	ec	NULL	eq_ref	PRIMARY	PRIMARY	50	event_planning_management_system.b.Event_ID	1	100.00	NULL		

Result 65 x

Action Output

#	Time	Action	Message	Duration / Fetch
24	13:02:21	explain SELECT EC.Event_ID, EC.Event_Name, B.Payment, B.Booking_Date FROM Events_TB_View as EC L...	2 row(s) returned	0.000 sec / 0.000 sec
25	13:02:55	explain SELECT Event_ID, Event_Name, Payment, Booking_Date FROM Event_Booking where Payment like '1%'	2 row(s) returned	0.000 sec / 0.000 sec
26	13:03:19	create index Bookinh_price on booking(Payment)	Error Code: 1061. Duplicate key name 'Bookinh_price'	0.000 sec
27	13:03:26	drop index Bookinh_price on booking	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
28	13:03:32	explain SELECT Event_ID, Event_Name, Payment, Booking_Date FROM Event_Booking where Payment like '1%'	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search 34°C 1:03 PM 4/9/2024

Figure 5. 7: Before Tuning

```

File Edit View Query Database Server Tools Scripting Help
Navigator: Event_Planning_Management x Workshop2
SCHEMAS Filter objects
foe_23 sys
Limit to 1000 rows
773
774 -- -----
775 -- before
776 • explain SELECT Event_ID, Event_Name, Payment, Booking_Date
777 FROM Event_Booking where Payment like '1%';
778
779 • create index Bookinh_price on booking(Payment);
780 • drop index Bookinh_price on booking ;
781
782 • explain SELECT Event_ID, Event_Name, Payment, Booking_Date
783 FROM Event_Booking where Payment like '1%';
784
785 -- -----
786 -- before

```

No object selected

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows			filtered	Extra
									rows	filtered	Using where		
1	SIMPLE	b	NULL	ALL	idx_event_id,Bookinh_price	NULL	NULL	NULL	5	20.00	Using where		
1	SIMPLE	ec	NULL	eq_ref	PRIMARY	PRIMARY	50	event_planning_management_system.b.Event_ID	1	100.00	NULL		

Result 67 x

Action Output

#	Time	Action	Message	Duration / Fetch
28	13:03:32	explain SELECT Event_ID, Event_Name, Payment, Booking_Date FROM Event_Booking where Payment like '1%'	2 row(s) returned	0.000 sec / 0.000 sec
29	13:04:15	drop index Bookinh_price on booking	Error Code: 1091. Can't DROP 'Bookinh_price'; check that column/key exists	0.016 sec
30	13:04:22	explain SELECT Event_ID, Event_Name, Payment, Booking_Date FROM Event_Booking where Payment like '1%'	2 row(s) returned	0.000 sec / 0.000 sec
31	13:04:28	create index Bookinh_price on booking(Payment)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
32	13:04:34	explain SELECT Event_ID, Event_Name, Payment, Booking_Date FROM Event_Booking where Payment like '1%'	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search 34°C 1:04 PM 4/9/2024

Figure 5. 8: Before Tuning

MySQL Workbench

Local instance MySQL8.3 - W...

File Edit View Query Database Server Tools Scripting Help

Navigators: SCHEMAS, Filter objects, fo_23, sys

Event_Planning_Management_... Workshop2

782 • explain SELECT Event_ID, Event_Name, Payment, Booking_Date
FR Execute the selected portion of the script or everything, if there is no selection

783
784
785 -- before
786
787 • explain SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date
FROM Event_Booking_Union where Event_Name like 'ck';

788
789
790 • drop INDEX idx_event_name ON Events_TB;
791 • CREATE INDEX idx_event_name ON Events_TB(Event_Name);
792 • create index Bookinh_price on booking(Payment);

793
794 -- after
795 • explain SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date

No object selected

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

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	<derived2>		ALL	NULL	NULL	NULL	NULL	4	100.00	NULL
2	DERIVED	ec		ALL	NULL	NULL	NULL	NULL	3	33.33	Using where
2	DERIVED	b		ref	idx_event_id	idx_event_id	50	event_planning_management_system.ec.Event...	1	100.00	NULL
3	UNION	ec		ALL	NULL	NULL	NULL	NULL	3	33.33	Using where
3	UNION	b		ref	idx_event_id	idx_event_id	50	event_planning_management_system.ec.Event...	1	100.00	NULL

Result 69 x

Output:

#	Time	Action	Message	Duration / Fetch
34	13:05:24	use Event_Planning_Management_System	0 row(s) affected	0.000 sec
35	13:05:24	create table Events_TB(Event_ID varchar(12) Not null, Emp_ID varchar(12) not null, Coordinator_ID varchar...	Error Code: 1050. Table 'events_tb' already exists	0.015 sec
36	13:05:29	explain SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date FROM Event_Booking_Union wher...	6 row(s) returned	0.000 sec / 0.000 sec
37	13:05:37	drop INDEX idx_event_name ON Events_TB	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
38	13:05:43	explain SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date FROM Event_Booking_Union wher...	6 row(s) returned	0.016 sec / 0.000 sec

Object Info Session

Type here to search

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

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	<derived2>		ALL	NULL	NULL	NULL	NULL	4	100.00	NULL
2	DERIVED	ec		range	idx_event_name	idx_event_name	203	NULL	1	100.00	Using index condition
2	DERIVED	b		ref	idx_event_id	idx_event_id	50	event_planning_management_system.ec.Event...	1	100.00	NULL
3	UNION	ec		range	idx_event_name	idx_event_name	203	NULL	1	100.00	Using index condition

Result 70 x

Output:

#	Time	Action	Message	Duration / Fetch
40	13:06:05	use Event_Planning_Management_System	0 row(s) affected	0.000 sec
41	13:06:05	create table Events_TB(Event_ID varchar(12) Not null, Emp_ID varchar(12) not null, Coordinator_ID varchar...	Error Code: 1050. Table 'events_tb' already exists	0.016 sec
42	13:06:14	CREATE INDEX idx_event_name ON Events_TB(Event_Name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
43	13:06:21	create index Bookinh_price on booking(Payment)	Error Code: 1061. Duplicate key name 'Bookinh_price'	0.000 sec
44	13:06:28	explain SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date FROM Event_Booking_Union wher...	6 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

Figure 5. 9: Before Tuning

MySQL Workbench

Local instance MySQL8.3 - W...

File Edit View Query Database Server Tools Scripting Help

Navigators: SCHEMAS, Filter objects, fo_23, sys

Event_Planning_Management_... Workshop2

782 • explain SELECT Event_ID, Event_Name, Payment, Booking_Date
FR Execute the selected portion of the script or everything, if there is no selection

783
784
785 -- before
786
787 • explain SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date
FROM Event_Booking_Union where Event_Name like 'ck';

788
789
790 • drop INDEX idx_event_name ON Events_TB;
791 • CREATE INDEX idx_event_name ON Events_TB(Event_Name);
792 • create index Bookinh_price on booking(Payment);

793
794 -- after
795 • explain SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date

No object selected

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

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	<derived2>		ALL	NULL	NULL	NULL	NULL	4	100.00	NULL
2	DERIVED	ec		range	idx_event_name	idx_event_name	203	NULL	1	100.00	Using index condition
2	DERIVED	b		ref	idx_event_id	idx_event_id	50	event_planning_management_system.ec.Event...	1	100.00	NULL
3	UNION	ec		range	idx_event_name	idx_event_name	203	NULL	1	100.00	Using index condition

Result 70 x

Output:

#	Time	Action	Message	Duration / Fetch
40	13:06:05	use Event_Planning_Management_System	0 row(s) affected	0.000 sec
41	13:06:05	create table Events_TB(Event_ID varchar(12) Not null, Emp_ID varchar(12) not null, Coordinator_ID varchar...	Error Code: 1050. Table 'events_tb' already exists	0.016 sec
42	13:06:14	CREATE INDEX idx_event_name ON Events_TB(Event_Name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
43	13:06:21	create index Bookinh_price on booking(Payment)	Error Code: 1061. Duplicate key name 'Bookinh_price'	0.000 sec
44	13:06:28	explain SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date FROM Event_Booking_Union wher...	6 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

Figure 5. 10: After Tuning

MySQL Workbench

Local instance MySQL83 - Workshop2

File Edit View Query Database Server Tools Scripting Help

Navigator: Event_Planning_Management_

SCHEMAS: Schemas

Administration Schemas Information No object selected

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

```

791 • CREATE INDEX idx_event_name ON Events_TB(Event_Name);
792 • create index Bookinh_price on booking(Payment);
793
794 -- after
795 • explain SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date
    FROM Event_Booking_Union where Event_Name like 'ck';
796
797
798 -- -----
799 -- Selecting data from the Outer_Union_View view
800 • explain SELECT * FROM Outer_Union_View where client_id='%2';
801
802 • create index clientId_index on clients(client_id);
803 • drop index clientId_index on clients;
804 • create index Booking_Id_index on booking(client_id);

```

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

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	<derived2>	HASH	ALL	HASH	HASH	HASH	HASH	3	100.00	HASH
2	DERIVED		HASH	HASH	HASH	HASH	HASH	HASH			no matching row in const table
3	UNION	booking	HASH	ref	PRIMARY, idx_event_id, Booking_Id_Index, Book... PRIMARY	50	const	1	100.00	Using index	
4	UNION RESULT	<union2,3>	HASH	ALL	HASH	HASH	HASH	HASH			Using temporary

Result 71 x Read Only Context Help Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
42	13:06:14	CREATE INDEX idx_event_name ON Events_TB(Event_Name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
43	13:06:21	create index Bookinh_price on booking(Payment)	Error Code: 1061. Duplicate key name 'Bookinh_price'	0.000 sec
44	13:06:28	explain SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date FROM Event_Booking_Union where Event_Name like 'ck';	6 row(s) returned	0.000 sec / 0.000 sec
45	13:06:59	drop INDEX idx_event_name ON Events_TB	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.032 sec
46	13:07:07	explain SELECT * FROM Outer_Union_View where client_id='%2';	4 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

Figure 5. 11: Before Tuning

MySQL Workbench

Local instance MySQL83 - Workshop2

File Edit View Query Database Server Tools Scripting Help

Navigator: Event_Planning_Management_

SCHEMAS: Schemas

Administration Schemas Information No object selected

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

```

794 -- after
795 • ex Execute the selected portion of the script or everything, if there is no selection g_Date
    FROM Event_Booking_Union where Event_Name like 'ck';
796
797
798 -- -----
799 -- Selecting data from the Outer_Union_View view
800 • explain SELECT * FROM Outer_Union_View where client_id='%2';
801
802 • create index clientId_index on clients(client_id);
803 • drop index clientId_index on clients;
804 • create index Booking_Id_index on booking(client_id);
805
806 • explain SELECT * FROM Outer_Union_View where client_id='%2';

```

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

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	<derived2>	HASH	ALL	HASH	HASH	HASH	HASH	3	100.00	HASH
2	DERIVED		HASH	HASH	HASH	HASH	HASH	HASH			no matching row in const table
3	UNION	booking	HASH	ref	PRIMARY, idx_event_id, Booking_Id_Index, Book... PRIMARY	50	const	1	100.00	Using index	
4	UNION RESULT	<union2,3>	HASH	ALL	HASH	HASH	HASH	HASH			Using temporary

Result 72 x Read Only Context Help Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
44	13:06:28	explain SELECT Event_ID, Event_Name, Client_ID, Payment, Booking_Date FROM Event_Booking_Union where Event_Name like 'ck';	6 row(s) returned	0.000 sec / 0.000 sec
45	13:06:59	drop INDEX idx_event_name ON Events_TB	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.032 sec
46	13:07:07	explain SELECT * FROM Outer_Union_View where client_id='%2';	4 row(s) returned	0.000 sec / 0.000 sec
47	13:07:57	create index Booking_Id_index on booking(client_id)	Error Code: 1061. Duplicate key name 'Booking_Id_Index'	0.000 sec
48	13:08:02	explain SELECT * FROM Outer_Union_View where client_id='%2';	4 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

Figure 5. 12: After Tuning

MySQL Workbench

Local instance MySQL83 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

Event_Planning_Management Workshop2

```

807
808 -- no more tuning
809
810 -----
811 -- before
812
813
814 • explain SELECT Event_Name, Number_of_Participant AS Quantity FROM Events_TB where Number_of_Participant<100
815 UNION
816     SELECT 'Booking Total', SUM(Payment) AS Quantity FROM Bookings;
817
818 • drop index participation_index on events_Tb;
819 • create index participation_index on Events_TB (Number_of_Participant);
820

```

No object selected

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

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	Events_TB	NULL	ALL	NULL	NULL	NULL	NULL	3	33.33	Using where
2	UNION	Booking	NULL	index	NULL	Bookinh_price	9	NULL	5	100.00	Using index
3	UNION RESULT	<union1,2>	NULL	ALL	NULL	NULL	NULL	NULL	NULL	NULL	Using temporary

Result 74 x

Action Output

#	Time	Action	Message	Duration / Fetch
47	13:07:57	create index Booking_Id_Index on booking(client_id)	Error Code: 1061. Duplicate key name 'Booking_Id_Index'	0.000 sec
48	13:08:02	explain SELECT * FROM Outer_Union_View where client_id=%2%	4 row(s) returned	0.000 sec / 0.000 sec
49	13:08:23	explain SELECT Event_Name, Number_of_Participant AS Quantity FROM Events_TB where Number_of_Participant<100	3 row(s) returned	0.000 sec / 0.000 sec
50	13:08:49	drop index participation_index on events_Tb;	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
51	13:08:55	explain SELECT Event_Name, Number_of_Participant AS Quantity FROM Events_TB where Number_of_Participant<100	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

10:09 PM 4/9/2024

Figure 5. 13: Before Tuning

MySQL Workbench

Local instance MySQL83 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

Event_Planning_Management Workshop2

```

806 • explain SELECT * FROM Outer_Union_View where client_id=%2%;
807 Execute the selected portion of the script or everything, if there is no selection
808 -- no more tuning
809
810 -----
811 -- before
812
813
814 • explain SELECT Event_Name, Number_of_Participant AS Quantity FROM Events_TB where Number_of_Participant<100
815 UNION
816     SELECT 'Booking Total', SUM(Payment) AS Quantity FROM Bookings;
817
818 • drop index participation_index on events_Tb;
819 • create index participation_index on Events_TB (Number_of_Participant);
820

```

No object selected

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

ID	Select Type	Table	Partitions	Type	Possible Keys	Key	Key Len	Ref	Rows	Filtered	Extra
1	PRIMARY	Events_TB	NULL	range	participation_index	participation_index	5	NULL	1	100.00	Using index condition
2	UNION	Booking	NULL	index	NULL	Bookinh_price	9	NULL	5	100.00	Using index
3	UNION RESULT	<union1,2>	NULL	ALL	NULL	NULL	NULL	NULL	NULL	NULL	Using temporary

Result 73 x

Action Output

#	Time	Action	Message	Duration / Fetch
45	13:06:59	drop INDEX idx_event_name ON Events_TB	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.032 sec
46	13:07:07	explain SELECT * FROM Outer_Union_View where client_id=%2%	4 row(s) returned	0.000 sec / 0.000 sec
47	13:07:57	create index Booking_Id_Index on booking(client_id)	Error Code: 1061. Duplicate key name 'Booking_Id_Index'	0.000 sec
48	13:08:02	explain SELECT * FROM Outer_Union_View where client_id=%2%	4 row(s) returned	0.000 sec / 0.000 sec
49	13:08:23	explain SELECT Event_Name, Number_of_Participant AS Quantity FROM Events_TB where Number_of_Participant<100	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

10:08 PM 4/9/2024

Figure 5. 14: After Tuning

MySQL Workbench

Local instance MySQL83 - Workshop2

File Edit View Query Database Server Tools Scripting Help

Navigators: Schemas, Information

No object selected

Event_Planning_Management_

Workshop2

```

810
811 -- before
812
813
814 • explain SELECT Event_Name, Number_of_Participant AS Quantity FROM Events_TB where Number_of_Participant<100
815 UNION
816 SELECT 'Booking Total', SUM(Payment) AS Quantity FROM Booking;
817
818 • drop index participation_index on events_Tb;
819 • create index participation_index on Events_TB (Number_of_Participant);
820
821 --
822 -- after
823 • explain SELECT Event_Name FROM Events_TB

```

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

ID	Select_Type	Table	Partitions	Type	Possible_Keys	Key	Key_Len	Ref	Rows	Filtered	Extra
1	PRIMARY	Events_TB	NULL	ALL	NULL	NULL	NULL	NULL	3	33.33	Using where
2	SUBQUERY	Events_TB	NULL	ALL	NULL	NULL	NULL	NULL	3	100.00	

Action Output

#	Time	Action	Message	Duration / Fetch
49	13:08:23	explain SELECT Event_Name, Number_of_Participant AS Quantity FROM Events_TB where Number_of_Participant < 100	3 row(s) returned	0.000 sec / 0.000 sec
50	13:08:49	drop index participation_index on events_Tb;	0 rows affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
51	13:08:55	explain SELECT Event_Name, Number_of_Participant AS Quantity FROM Events_TB where Number_of_Participant < 100	3 row(s) returned	0.000 sec / 0.000 sec
52	13:09:27	drop index participation_index on events_Tb;	Error Code: 1091. Can't DROP 'participation_index'; check that column/key exists	0.000 sec
53	13:09:33	explain SELECT Event_Name FROM Events_TB WHERE Number_of_Participant > (SELECT AVG(Number_of_Participant))	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Result Grid | Read Only | Context Help | Snippets

Type here to search

Figure 5. 15: Before Tuning

MySQL Workbench

Local instance MySQL83 - Workshop2

File Edit View Query Database Server Tools Scripting Help

Navigators: Schemas, Information

No object selected

Event_Planning_Management_

Workshop2

```

810
811 Execute the selected portion of the script or everything, if there is no selection
812
813
814 • explain SELECT Event_Name, Number_of_Participant AS Quantity FROM Events_TB where Number_of_Participant<100
815 UNION
816 SELECT 'Booking Total', SUM(Payment) AS Quantity FROM Booking;
817
818 • drop index participation_index on events_Tb;
819 • create index participation_index on Events_TB (Number_of_Participant);
820
821 --
822 -- after
823 • explain SELECT Event_Name FROM Events_TB

```

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

ID	Select_Type	Table	Partitions	Type	Possible_Keys	Key	Key_Len	Ref	Rows	Filtered	Extra
1	PRIMARY	Events_TB	NULL	range	participation_index	participation_index	5	NULL	1	100.00	Using where
2	SUBQUERY	Events_TB	NULL	index	NULL	participation_index	5	NULL	3	100.00	Using index

Action Output

#	Time	Action	Message	Duration / Fetch
51	13:08:55	explain SELECT Event_Name, Number_of_Participant AS Quantity FROM Events_TB where Number_of_Participant < 100	3 row(s) returned	0.000 sec / 0.000 sec
52	13:09:27	drop index participation_index on events_Tb;	Error Code: 1091. Can't DROP 'participation_index'; check that column/key exists	0.000 sec
53	13:09:33	explain SELECT Event_Name FROM Events_TB WHERE Number_of_Participant > (SELECT AVG(Number_of_Participant))	2 row(s) returned	0.000 sec / 0.000 sec
54	13:10:00	create index participation_index on Events_TB (Number_of_Participant)	0 rows affected Records: 0 Duplicates: 0 Warnings: 0	0.062 sec
55	13:10:03	explain SELECT Event_Name FROM Events_TB WHERE Number_of_Participant > (SELECT AVG(Number_of_Participant))	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Result Grid | Read Only | Context Help | Snippets

Type here to search

Figure 5. 16: After Tuning