DATABASE SYSTEM

INTEGRATED EVENT CREATION & TRACKING SYSTEM

NAME	SAP ID	COURSE	SEMESTER
Alisha Noman	59859	DBS	4th
Saiqa Shabbir	58778	DBS	4th
Hoor e Ain Khattak	58797	DBS	4th

Faculty of Computing
Riphah International University, Islamabad
Fall 2025
A Dissertation Submitted To

Faculty of Computing,
Riphah International University, Islamabad
As a Partial Fulfillment of the Requirement for the Course Database Management System
Bachelors of Science in Software Engineering

Faculty of Computing Riphah International University, Islamabad
Dedication/Acknowledgment

Thanks to Allah Almighty who made us able to complete this final project report. Also our course teacher who guided us in this project. All the members of the team who worked hard and diligently to complete this project.

Alisha Noman 59859

Saiqa Shabbir 58778

Hoor e Ain Khattak 58797

Abstract

The **Integrated Event Creation & Tracking System** is a comprehensive platform designed to streamline the planning, management, and monitoring of events. It enables event creators to efficiently manage event details, ticketing, attendee registrations, payments, and venue logistics, all within a single system. By implementing a relational database, the system ensures **data consistency, integrity, and real-time updates**, while reducing redundancy and automating routine processes.

The system supports multiple stakeholders including event creators, attendees, administrators, and finance staff, allowing them to perform tasks relevant to their roles. Event creators can organize events with defined types and venues, configure ticket options, and track registrations and payments. Attendees can easily register for events, receive tickets, and view event information. Administrators maintain system integrity, oversee events, and manage venues and event types, while finance staff can monitor payment transactions and registration statuses.

This system provides **historical reporting, analytics, and dashboards** to facilitate informed decision-making. It is scalable, secure, and adaptable to various types of events, ranging from conferences and weddings to concerts and workshops. The project demonstrates how an integrated, database-driven approach enhances event management efficiency, reduces human errors, and improves stakeholder satisfaction.

Project Description

The **Integrated Event Creation & Tracking System** is a database-driven platform designed to manage events, attendees, tickets, venues, and payments efficiently. The system allows **event creators** to add and manage events with details like name, date, time, type, and venue. **Attendees** can register for events, receive tickets, and their registration is linked with payments for accurate tracking.

The relational database implements **tables for EventCreator, Event, Attendee, Ticket, Registration, Payment, Venue, EventType, and EventVenue**, ensuring data consistency and enforcing business rules such as preventing overbooking of tickets and linking each registration to a payment. **CRUD operations** are supported for all entities, allowing creation, reading, updating, and deletion of data.

Additionally, **triggers** automatically update ticket availability or stock after registration, and **views** provide quick access to critical information, such as high-priority events or top-contributing donors. With **sample data and relational constraints**, the system enables real-time updates, reporting, and analytics, supporting stakeholders like event creators, attendees, administrators, and finance staff to efficiently plan, track, and analyze events.

1. 📍 Requirement Analysis

Introduction

An Event Creator Management System is designed to streamline the planning, execution, and tracking of events. It enables event creators to manage event details, attendees, ticketing, payments, and venue logistics. The system ensures data consistency, supports real-time updates, and provides historical records for analysis. By implementing a relational database, we can achieve efficient data handling, reduce redundancy, and enforce business rules across all event-related operations.

◆ Stakeholders

Stakeholder	Role & Interaction with System
Event Creator	Creates and manages events, assigns venues, configures tickets
Attendee	Registers for events, views event details
Admin	Oversees system integrity, manages event types and venues
Finance Staff	Tracks payments and registration statuses

Functional Requirements

1. Event creators can create events with details like name, date, time, location, and type
2. Attendees can register for events and receive tickets
3. Payments are linked to registrations and tracked by method and amount
4. Events are associated with specific venues
5. Events are categorized by type (e.g., conference, wedding, concert)
6. Ticket types and quantities are managed per event
7. Registration status and payment status are tracked
8. Secure login for event creators

2. 🧩 Entity Identification

Entity	Attributes
EVENT_CREATOR	EventCreatorID (PK), FirstName, LastName, Email, Phone, Password
EVENT	EventID (PK), EventCreatorID (FK), EventName, EventDescription, EventDate, EventTime, EventLocation, EventType
ATTENDEE	AttendeeID (PK), FirstName, LastName, Email, Phone
REGISTRATION	RegistrationID (PK), EventID (FK), AttendeeID (FK), RegistrationDate, RegistrationStatus
TICKET	TicketID (PK), EventID (FK), TicketType, TicketPrice, TicketQuantity
PAYMENT	PaymentID (PK), RegistrationID (FK), PaymentDate, PaymentAmount, PaymentMethod
VENUE	VenueID (PK), VenueName, VenueLocation, VenueCapacity
EVENT_VENUE	EventVenueID (PK), EventID (FK), VenueID (FK)
EVENT_TYPE	EventTypeID (PK), EventTypeName

3. 📜 Business Rules

1. Each event must be created by one event creator
2. Each event can have one or more ticket types
3. Each attendee can register for multiple events
4. Each registration must be linked to a payment
5. Each event must be assigned to a venue via EVENT_VENUE
6. Event types must be predefined and assigned to events
7. Ticket quantities must be tracked to prevent overbooking

4.  Conceptual Design (ERD Overview)

Your ERD includes the following relationships:

EVENT_CREATOR → EVENT (1:M)

EVENT → REGISTRATION → ATTENDEE (M:N via REGISTRATION)

REGISTRATION → PAYMENT (1:1)

EVENT → TICKET (1:M)

EVENT → EVENT_VENUE → VENUE (M:N via EVENT_VENUE)

EVENT_TYPE → EVENT (1:M)

This structure supports complex event planning with multiple attendees, ticketing options, payment tracking, and venue management.

Entities and Attributes

- **InvitationResponseType**
- **Invitation**
- **Event**
- **EventStatus**
- **User**
- **Role**
- **UserRole**
- **Group**
- **AllowedEventGroup**

Each will retain its original attributes like ID, Name, Description, CreationDT, IsActive, etc.

Business Rules for Integrated Event Creation & Tracking System

Business rules define the constraints, relationships, and policies that govern how the system operates. These rules ensure data consistency, fair management, and controlled access during event planning and execution.

1. Event & Event Creator Rules

- Each Event Must Be Created by One Event Creator
  Every event is linked to a single event creator to ensure accountability and ownership. This allows tracking of who created the event and avoids duplication.
- An Event Creator Can Create Multiple Events
  One event creator can manage several events. This represents a one-to-many relationship, enabling organizers to manage multiple events efficiently.
- Each Event Must Have a Type and Assigned Venue
  Every event must be categorized (e.g., conference, wedding, concert) and linked to at least one venue. This ensures proper classification and location management.

2. Attendee & Registration Rules

- Each Attendee Can Register for Multiple Events
  Attendees may participate in more than one event, allowing flexibility in event management.
- Each Registration Must Be Linked to an Event and Attendee
  Each registration record uniquely identifies which attendee is registered for which event, preventing errors in ticketing or attendance tracking.
- Registration Status Must Be Tracked
  Registration records include statuses like Confirmed, Pending, or Cancelled to manage attendance accurately.

3. Ticket & Payment Rules

- Each Event Can Have Multiple Ticket Types
  Events can offer different ticket categories (e.g., VIP, Regular) to accommodate various pricing and privileges.
- Ticket Quantity Must Be Tracked to Prevent Overbooking
  Each ticket type has a defined quantity, ensuring that sales never exceed available capacity.
- Each Registration Must Have an Associated Payment
  Payment records are linked to registrations, capturing payment method, amount, and date to ensure financial accountability.

4. Venue & Event-Venue Rules

- A Venue Can Host Multiple Events
  Venues can host different events over time, supporting efficient resource utilization.
- Each Event Can Be Assigned to One or More Venues
  Events can span multiple locations if required, establishing a many-to-many relationship via the EventVenue table.
- Venue Capacity Must Be Respected
  The system ensures the number of tickets sold does not exceed venue capacity to prevent overcrowding.

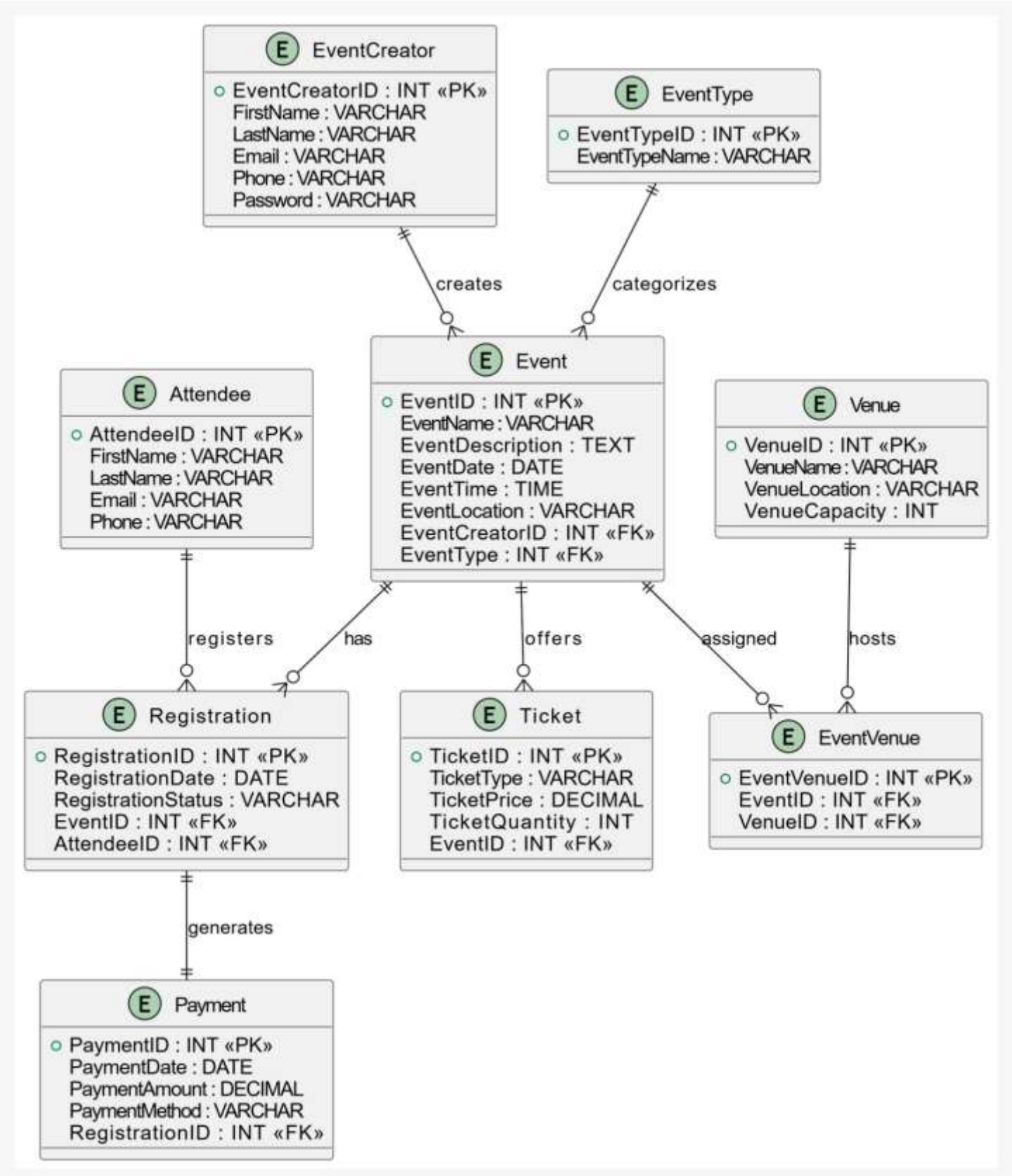
5. Admin Oversight Rules

- Only Admins Can Manage Event Types and Venues
  Admins have authority to create, update, or remove event types and venues, maintaining system integrity.
- Admins Can View Dashboards and Reports
  Admins monitor registrations, ticket sales, payments, and attendance through dashboards and analytics for decision-making.
- Admins Control Access
  Admins can manage event creator accounts, reset passwords, and oversee security to prevent unauthorized access.

6. System Integrity Rules

- Data Consistency Must Be Enforced via Relational Constraints
  Foreign keys and table relationships ensure that events, attendees, tickets, and payments are correctly linked.
- Real-Time Updates Must Reflect in All Relevant Tables
  Changes like ticket purchases, registration cancellations, or venue changes must automatically update related data to prevent conflicts.
- Historical Records Must Be Maintained
  Past events, payments, and attendance records are stored for analysis, reporting, and auditing purposes.

This set of rules directly maps to your database design, CRUD operations, triggers, and views. It covers event creators, events, attendees, registrations, tickets, payments, venues, and admin oversight, exactly like your code.


EventCreator
EventCreatorID : INT «PK»
FirstName : VARCHAR
LastName : VARCHAR
Email : VARCHAR
Phone : VARCHAR
Password : VARCHAR
EventType
EventTypeID : INT «PK»
EventTypeName : VARCHAR
creates
categorizes
Event
EventID : INT «PK»
EventName : VARCHAR
EventDescription : TEXT
EventDate : DATE
EventTime : TIME
EventLocation : VARCHAR
EventCreatorID : INT «FK»
EventType : INT «FK»
Attendee
AttendeeID : INT «PK»
FirstName : VARCHAR
LastName : VARCHAR
Email : VARCHAR
Phone : VARCHAR
Venue
VenueID : INT «PK»
VenueName : VARCHAR
VenueLocation : VARCHAR
VenueCapacity : INT
registers
has
offers
assigned
hosts
Registration
RegistrationID : INT «PK»
RegistrationDate : DATE
RegistrationStatus : VARCHAR
EventID : INT «FK»
AttendeeID : INT «FK»
Ticket
TicketID : INT «PK»
TicketType : VARCHAR
TicketPrice : DECIMAL
TicketQuantity : INT
EventID : INT «FK»
EventVenue
EventVenueID : INT «PK»
EventID : INT «FK»
VenueID : INT «FK»
generates
Payment
PaymentID : INT «PK»
PaymentDate : DATE
PaymentAmount : DECIMAL
PaymentMethod : VARCHAR
RegistrationID : INT «FK»

5. Logical Design

Table Name	Primary Key	Foreign Keys
EVENT_CREATOR	EventCreatorID	—
EVENT	EventID	EventCreatorID → EVENT_CREATOR, EventType → EVENT_TYPE
ATTENDEE	AttendeeID	—
REGISTRATION	RegistrationID	EventID → EVENT, AttendeeID → ATTENDEE
PAYMENT	PaymentID	RegistrationID → REGISTRATION
TICKET	TicketID	EventID → EVENT
VENUE	VenueID	—
EVENT_VENUE	EventVenueID	EventID → EVENT, VenueID → VENUE
EVENT_TYPE	EventTypeID	—

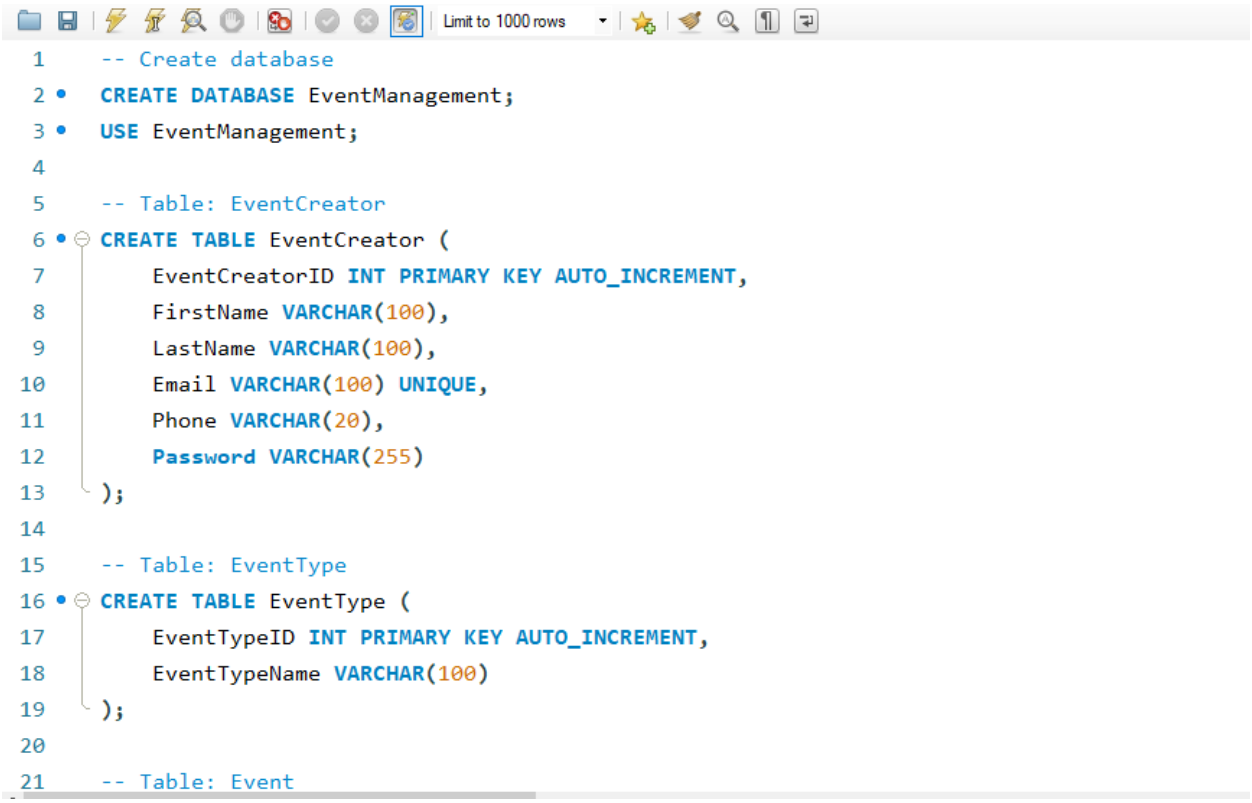
6. Final Relational Schema (SQL)

```
CREATE TABLE EventCreator (
EventCreatorID INT PRIMARY KEY AUTO_INCREMENT,
FirstName VARCHAR(100),
LastName VARCHAR(100),
Email VARCHAR(100) UNIQUE,
Phone VARCHAR(20),
Password VARCHAR(255)
);

CREATE TABLE EventType (
EventTypeID INT PRIMARY KEY AUTO_INCREMENT,
EventTypeName VARCHAR(100)
);

CREATE TABLE Event (
EventID INT PRIMARY KEY AUTO_INCREMENT,
EventCreatorID INT,
```

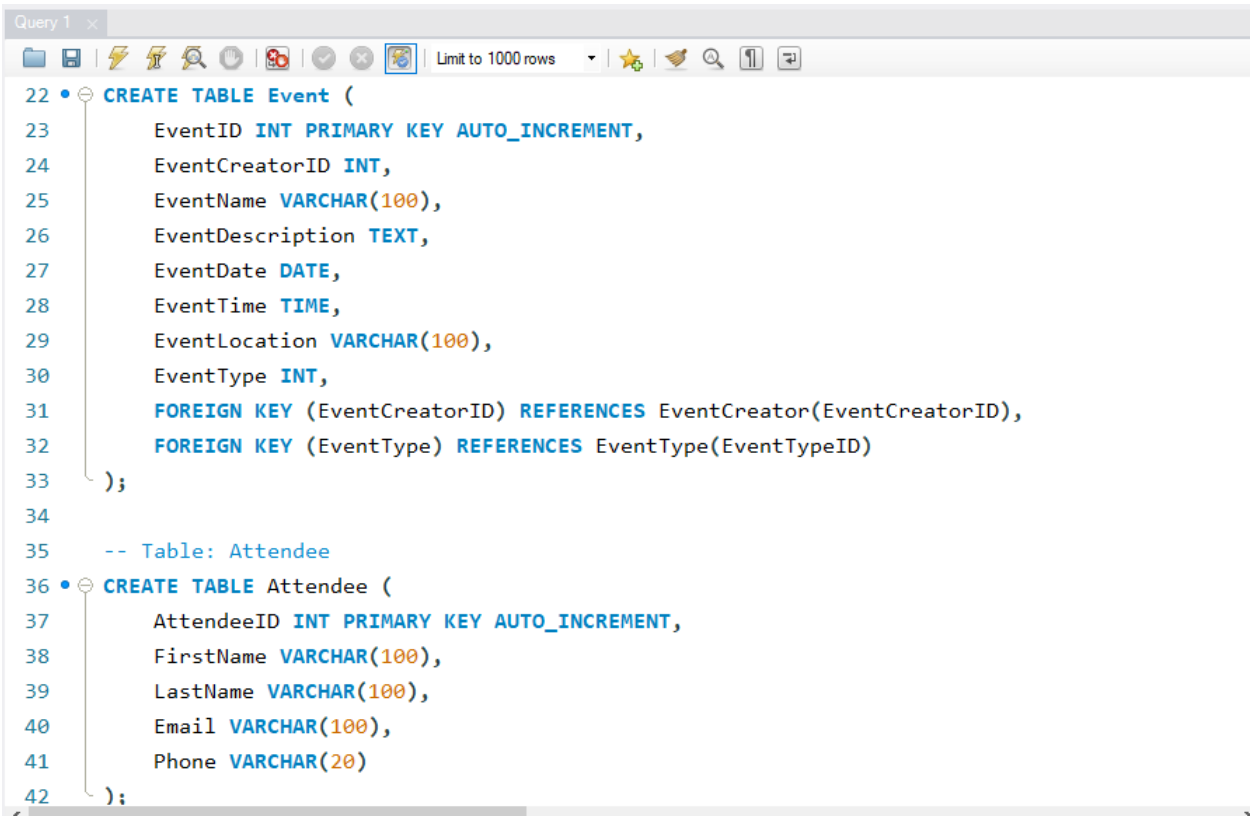
```
EventName VARCHAR(100),
EventDescription TEXT,
EventDate DATE,
EventTime TIME,
EventLocation VARCHAR(100),
EventType INT,
FOREIGN KEY (EventCreatorID) REFERENCES EventCreator(EventCreatorID),
FOREIGN KEY (EventType) REFERENCES EventType(EventTypeID)
);

CREATE TABLE Attendee (
AttendeeID INT PRIMARY KEY AUTO_INCREMENT,
FirstName VARCHAR(100),
LastName VARCHAR(100),
Email VARCHAR(100),
Phone VARCHAR(20)
);

CREATE TABLE Registration (
RegistrationID INT PRIMARY KEY AUTO_INCREMENT,
EventID INT,
AttendeeID INT,
RegistrationDate DATE,
RegistrationStatus VARCHAR(50),
FOREIGN KEY (EventID) REFERENCES Event(EventID),
FOREIGN KEY (AttendeeID) REFERENCES Attendee(AttendeeID)
);

CREATE TABLE Payment (
PaymentID INT PRIMARY KEY AUTO_INCREMENT,
RegistrationID INT,
PaymentDate DATE,
PaymentAmount DECIMAL(10,2),
PaymentMethod VARCHAR(50),
FOREIGN KEY (RegistrationID) REFERENCES Registration(RegistrationID)
);

CREATE TABLE Ticket (
TicketID INT PRIMARY KEY AUTO_INCREMENT,
EventID INT,
TicketType VARCHAR(50),
TicketPrice DECIMAL(10,2),
TicketQuantity INT,
FOREIGN KEY (EventID) REFERENCES Event(EventID)
);
```

```
CREATE TABLE Venue (
VenueID INT PRIMARY KEY AUTO_INCREMENT,
VenueName VARCHAR(100),
VenueLocation VARCHAR(100),
VenueCapacity INT
);

CREATE TABLE EventVenue (
EventVenueID INT PRIMARY KEY AUTO_INCREMENT,
EventID INT,
VenueID INT,
FOREIGN KEY (EventID) REFERENCES Event(EventID),
FOREIGN KEY (VenueID) REFERENCES Venue(VenueID)
);
```

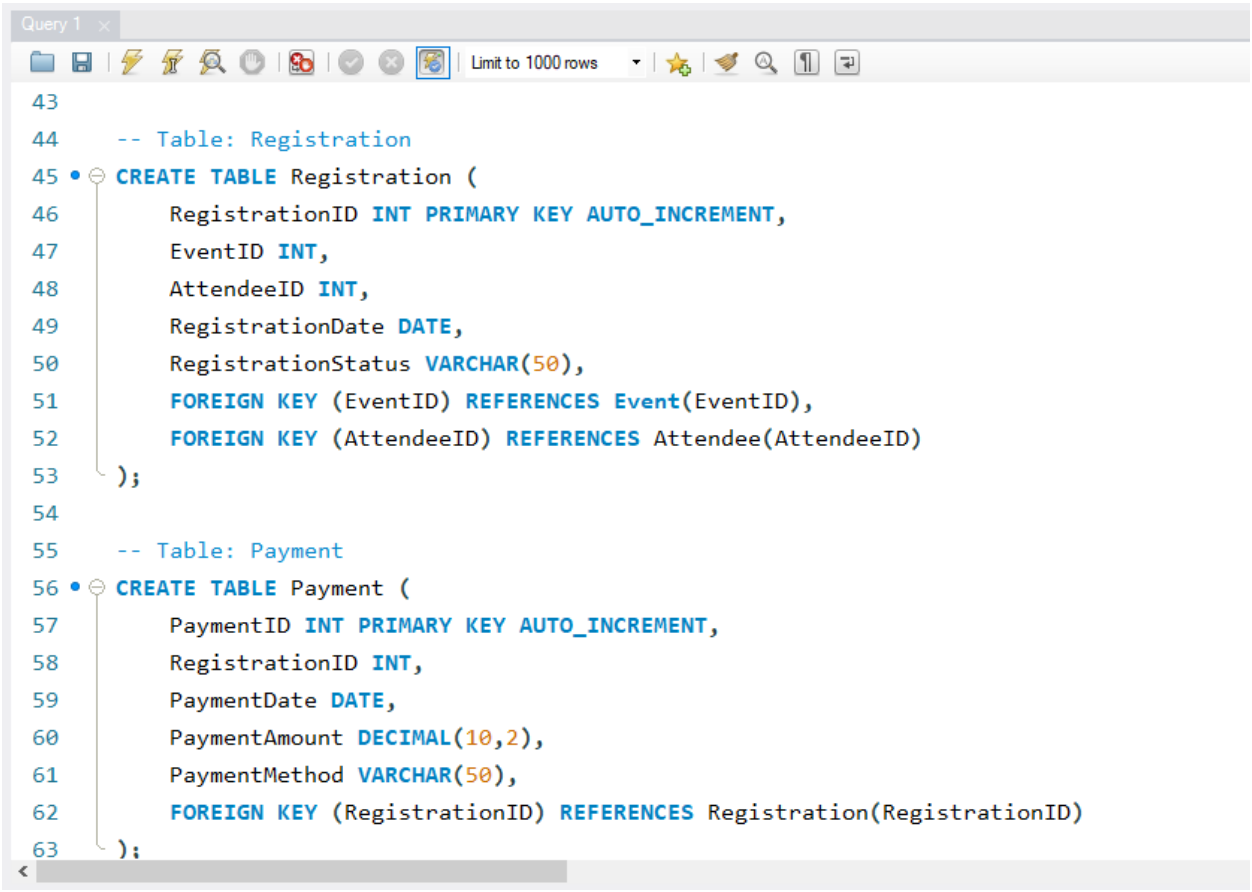
CODE:



```sql
-- Create database
CREATE DATABASE EventManagement;
USE EventManagement;

-- Table: EventCreator
CREATE TABLE EventCreator (
    EventCreatorID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(100),
    LastName VARCHAR(100),
    Email VARCHAR(100) UNIQUE,
    Phone VARCHAR(20),
    Password VARCHAR(255)
);

-- Table: EventType
CREATE TABLE EventType (
    EventTypeID INT PRIMARY KEY AUTO_INCREMENT,
    EventTypeName VARCHAR(100)
);

-- Table: Event
```



```
Query 1
Limit to 1000 rows
22  CREATE TABLE Event (
23      EventID INT PRIMARY KEY AUTO_INCREMENT,
24      EventCreatorID INT,
25      EventName VARCHAR(100),
26      EventDescription TEXT,
27      EventDate DATE,
28      EventTime TIME,
29      EventLocation VARCHAR(100),
30      EventType INT,
31      FOREIGN KEY (EventCreatorID) REFERENCES EventCreator(EventCreatorID),
32      FOREIGN KEY (EventType) REFERENCES EventType(EventTypeID)
33  );
34
35  -- Table: Attendee
36  CREATE TABLE Attendee (
37      AttendeeID INT PRIMARY KEY AUTO_INCREMENT,
38      FirstName VARCHAR(100),
39      LastName VARCHAR(100),
40      Email VARCHAR(100),
41      Phone VARCHAR(20)
42  );
```



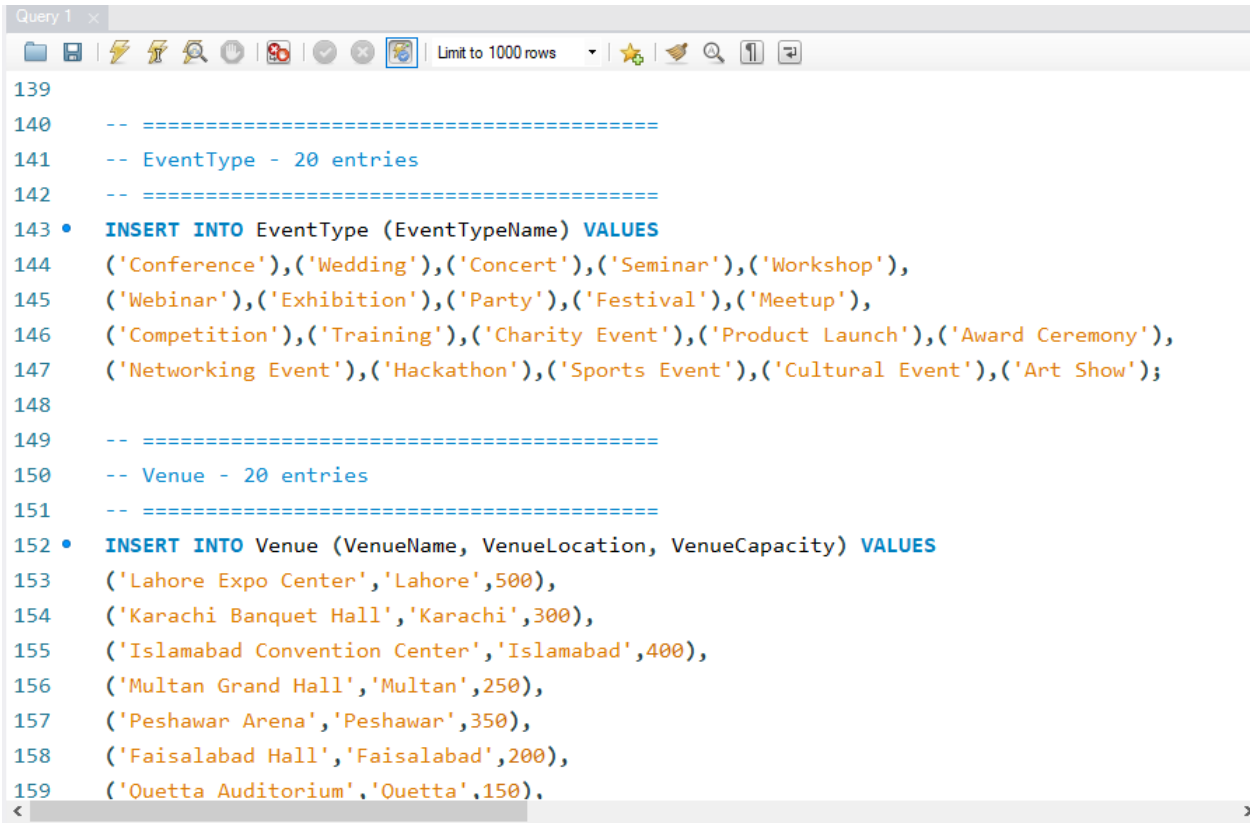
```
Query 1
Limit to 1000 rows
43
44  -- Table: Registration
45  CREATE TABLE Registration (
46      RegistrationID INT PRIMARY KEY AUTO_INCREMENT,
47      EventID INT,
48      AttendeeID INT,
49      RegistrationDate DATE,
50      RegistrationStatus VARCHAR(50),
51      FOREIGN KEY (EventID) REFERENCES Event(EventID),
52      FOREIGN KEY (AttendeeID) REFERENCES Attendee(AttendeeID)
53  );
54
55  -- Table: Payment
56  CREATE TABLE Payment (
57      PaymentID INT PRIMARY KEY AUTO_INCREMENT,
58      RegistrationID INT,
59      PaymentDate DATE,
60      PaymentAmount DECIMAL(10,2),
61      PaymentMethod VARCHAR(50),
62      FOREIGN KEY (RegistrationID) REFERENCES Registration(RegistrationID)
63  );
```

```
61      PaymentMethod VARCHAR(50),
62      FOREIGN KEY (RegistrationID) REFERENCES Registration(RegistrationID)
63  );
64
65  -- Table: Ticket
66  CREATE TABLE Ticket (
67      TicketID INT PRIMARY KEY AUTO_INCREMENT,
68      EventID INT,
69      TicketType VARCHAR(50),
70      TicketPrice DECIMAL(10,2),
71      TicketQuantity INT,
72      FOREIGN KEY (EventID) REFERENCES Event(EventID)
73  );
74
75  -- Table: Venue
76  CREATE TABLE Venue (
77      VenueID INT PRIMARY KEY AUTO_INCREMENT,
78      VenueName VARCHAR(100),
79      VenueLocation VARCHAR(100),
80      VenueCapacity INT
81  );
```

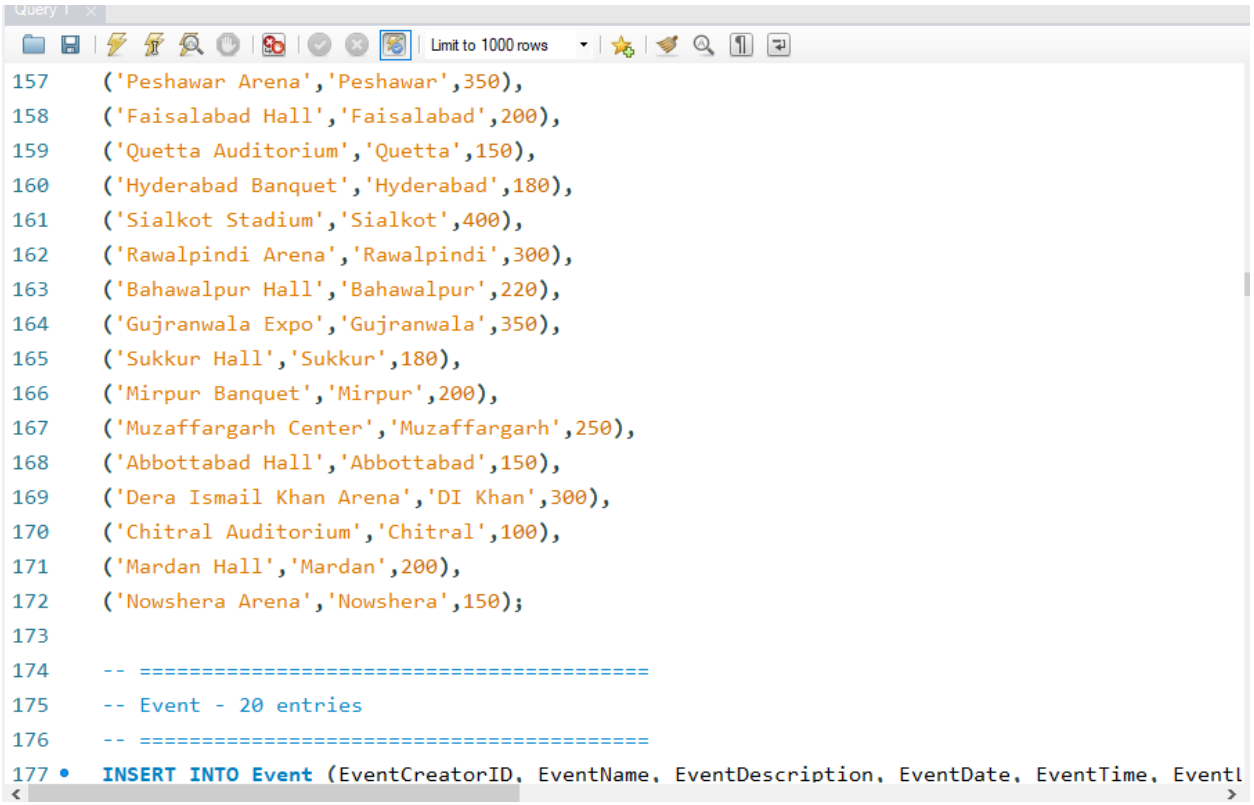
```
82
83  -- Table: EventVenue
84  CREATE TABLE EventVenue (
85      EventVenueID INT PRIMARY KEY AUTO_INCREMENT,
86      EventID INT,
87      VenueID INT,
88      FOREIGN KEY (EventID) REFERENCES Event(EventID),
89      FOREIGN KEY (VenueID) REFERENCES Venue(VenueID)
90  );
91  -- 3. Trigger: Update Ticket Quantity after Registration
92  DELIMITER //
93
94  CREATE TRIGGER UpdateTicketAfterRegistration
95  AFTER INSERT ON Registration
96  FOR EACH ROW
97  BEGIN
98      UPDATE Ticket
99      SET TicketQuantity = TicketQuantity - 1
100     WHERE EventID = NEW.EventID
101     AND TicketQuantity > 0;
102 END;
```

```
103    //
104
105    DELIMITER ;
106
107    -- 4. View: High Attendance Events
108    CREATE VIEW HighAttendanceEvents AS
109    SELECT E.EventName, E.EventDate, COUNT(R.RegistrationID) AS TotalRegistrations
110    FROM Event E
111    JOIN Registration R ON E.EventID = R.EventID
112    GROUP BY E.EventID
113    HAVING TotalRegistrations > 50;
114
115    -- ========================================
116    -- EventCreator - 20 entries
117    -- ========================================
118    INSERT INTO EventCreator (FirstName, LastName, Email, Phone, Password) VALUES
119    ('Ali','Khan','ali1@example.com','03001000001','pass123'),
120    ('Sara','Malik','sara1@example.com','03001000002','pass123'),
121    ('Ahmed','Raza','ahmed1@example.com','03001000003','pass123'),
122    ('Fatima','Noor','fatima1@example.com','03001000004','pass123'),
123    ('Hassan','Ali','hassan1@example.com','03001000005','pass123'),
```

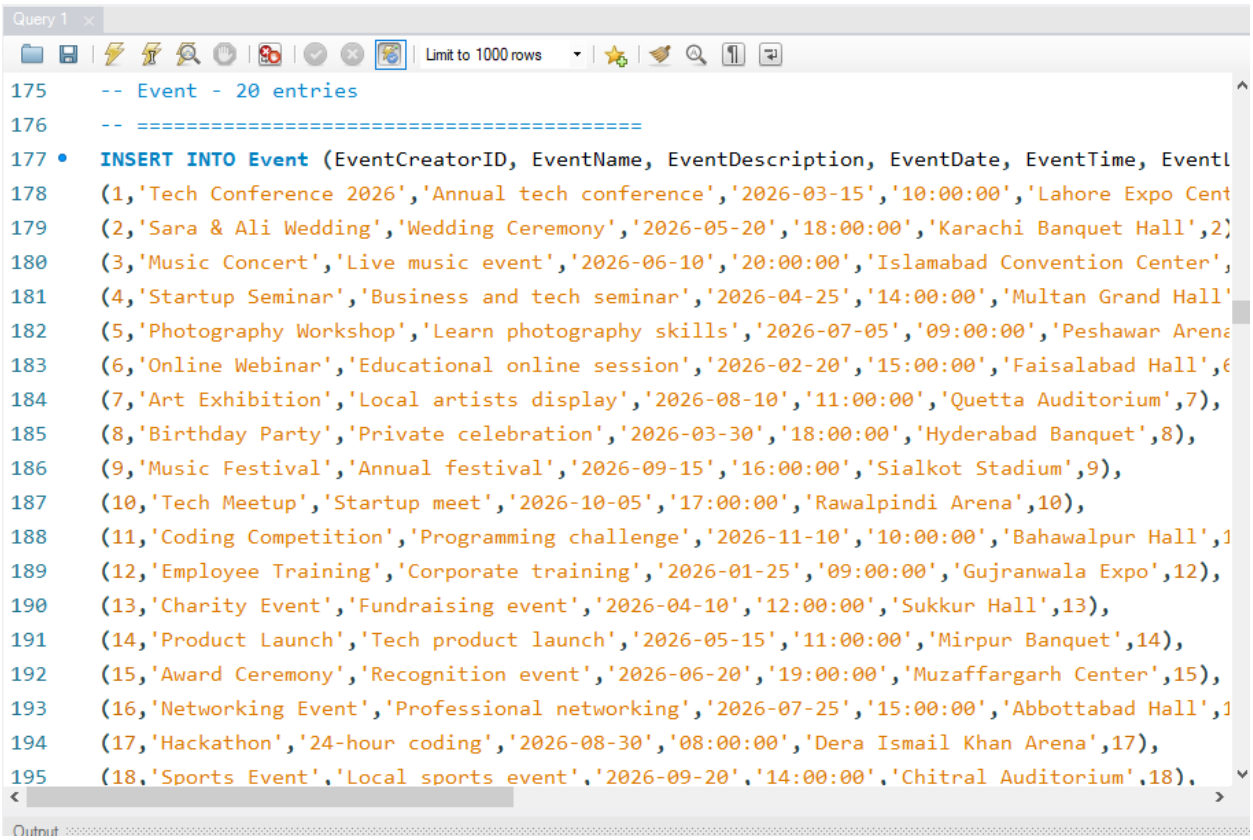
```
121    ('Ahmed','Raza','ahmed1@example.com','03001000003','pass123'),
122    ('Fatima','Noor','fatima1@example.com','03001000004','pass123'),
123    ('Hassan','Ali','hassan1@example.com','03001000005','pass123'),
124    ('Ayesha','Baloch','ayesha1@example.com','03001000006','pass123'),
125    ('Usman','Shah','usman1@example.com','03001000007','pass123'),
126    ('Zoya','Malik','zoya1@example.com','03001000008','pass123'),
127    ('Fahad','Iqbal','fahad1@example.com','03001000009','pass123'),
128    ('Noor','Fatima','noor1@example.com','03001000010','pass123'),
129    ('Kamran','Akbar','kamran1@example.com','03001000011','pass123'),
130    ('Rabia','Shaikh','rabia1@example.com','03001000012','pass123'),
131    ('Imran','Khan','imran1@example.com','03001000013','pass123'),
132    ('Sana','Malik','sana1@example.com','03001000014','pass123'),
133    ('Bilal','Hussain','bilal1@example.com','03001000015','pass123'),
134    ('Hina','Raza','hina1@example.com','03001000016','pass123'),
135    ('Tariq','Ahmed','tariq1@example.com','03001000017','pass123'),
136    ('Maryam','Ali','maryam1@example.com','03001000018','pass123'),
137    ('Shahbaz','Khan','shahbaz1@example.com','03001000019','pass123'),
138    ('Areeba','Noor','areeba1@example.com','03001000020','pass123');
139
140    -- ========================================
141    -- EventType - 20 entries
```



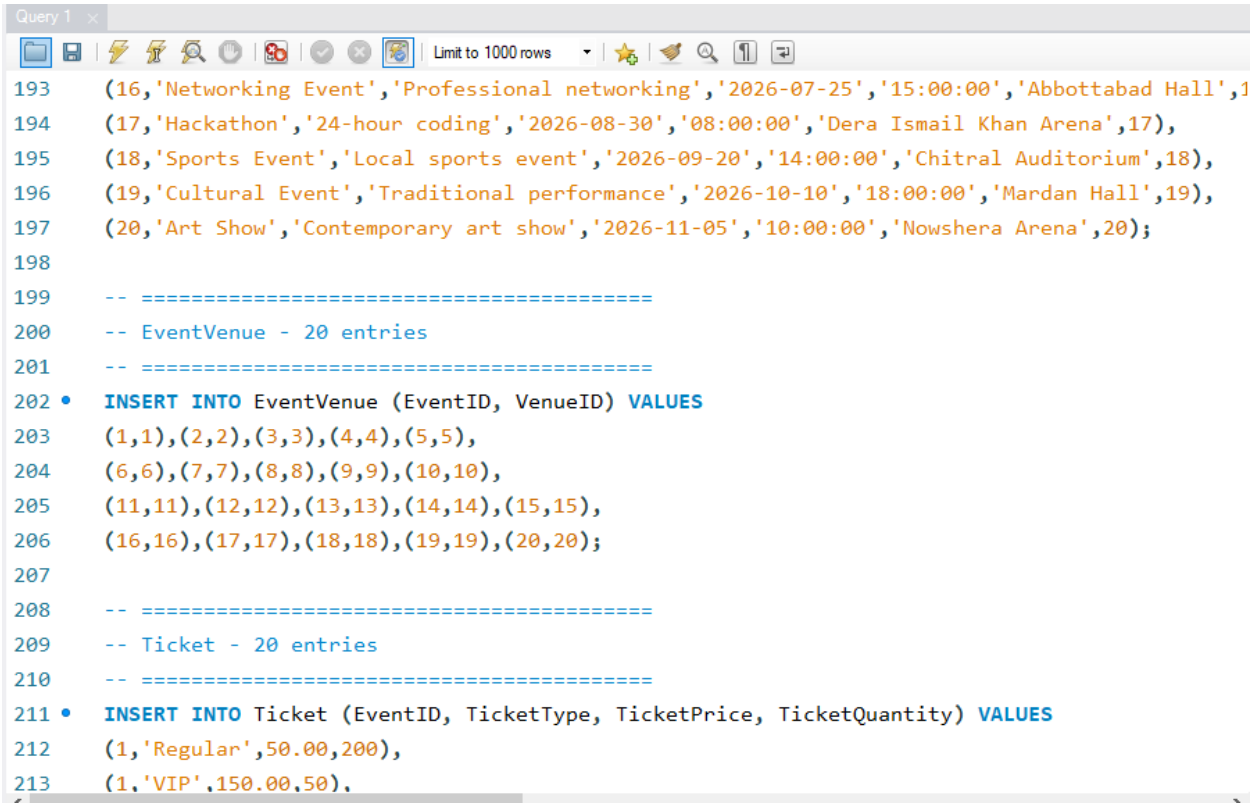
```
139
140    -- ========================================
141    -- EventType - 20 entries
142    -- ========================================
143 •  INSERT INTO EventType (EventTypeName) VALUES
144    ('Conference'),('Wedding'),('Concert'),('Seminar'),('Workshop'),
145    ('Webinar'),('Exhibition'),('Party'),('Festival'),('Meetup'),
146    ('Competition'),('Training'),('Charity Event'),('Product Launch'),('Award Ceremony'),
147    ('Networking Event'),('Hackathon'),('Sports Event'),('Cultural Event'),('Art Show');
148
149    -- ========================================
150    -- Venue - 20 entries
151    -- ========================================
152 •  INSERT INTO Venue (VenueName, VenueLocation, VenueCapacity) VALUES
153    ('Lahore Expo Center','Lahore',500),
154    ('Karachi Banquet Hall','Karachi',300),
155    ('Islamabad Convention Center','Islamabad',400),
156    ('Multan Grand Hall','Multan',250),
157    ('Peshawar Arena','Peshawar',350),
158    ('Faisalabad Hall','Faisalabad',200),
159    ('Quetta Auditorium','Quetta',150),
```



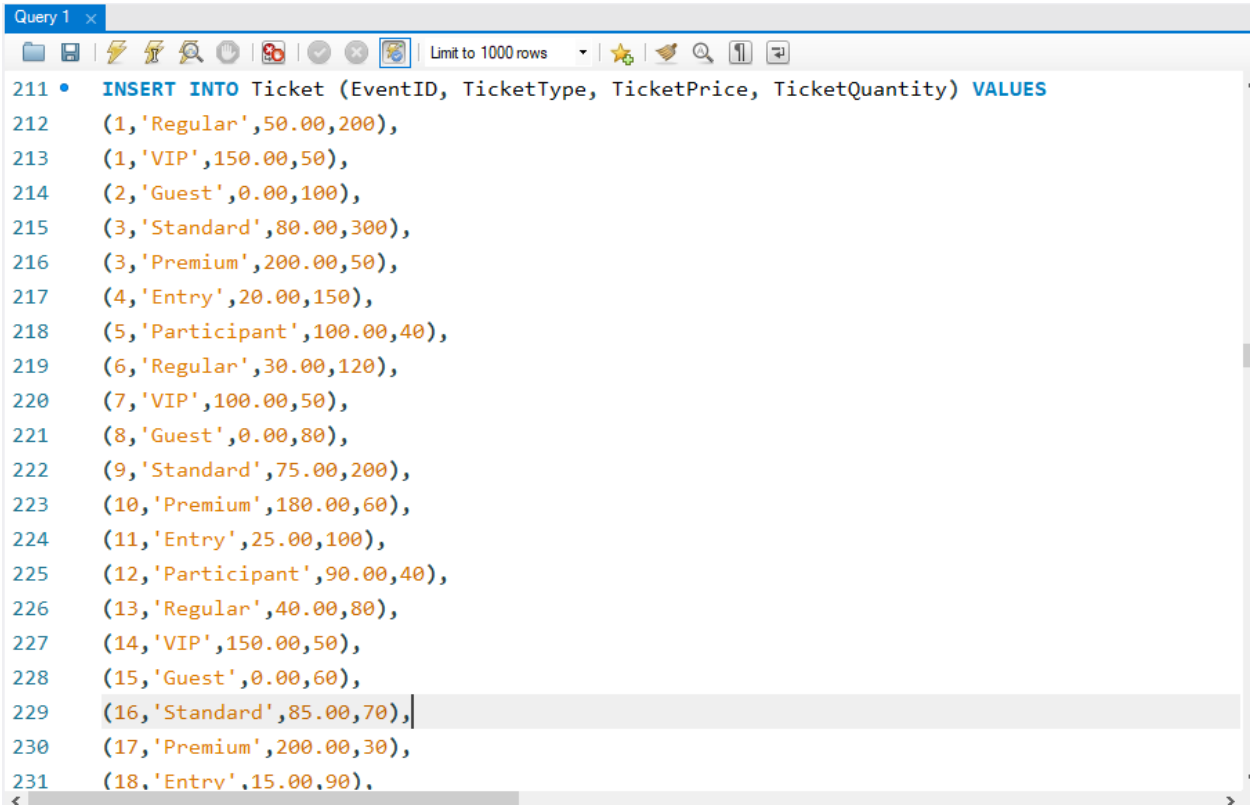
```
157    ('Peshawar Arena','Peshawar',350),
158    ('Faisalabad Hall','Faisalabad',200),
159    ('Quetta Auditorium','Quetta',150),
160    ('Hyderabad Banquet','Hyderabad',180),
161    ('Sialkot Stadium','Sialkot',400),
162    ('Rawalpindi Arena','Rawalpindi',300),
163    ('Bahawalpur Hall','Bahawalpur',220),
164    ('Gujranwala Expo','Gujranwala',350),
165    ('Sukkur Hall','Sukkur',180),
166    ('Mirpur Banquet','Mirpur',200),
167    ('Muzaffargarh Center','Muzaffargarh',250),
168    ('Abbottabad Hall','Abbottabad',150),
169    ('Dera Ismail Khan Arena','DI Khan',300),
170    ('Chitral Auditorium','Chitral',100),
171    ('Mardan Hall','Mardan',200),
172    ('Nowshera Arena','Nowshera',150);
173
174    -- ========================================
175    -- Event - 20 entries
176    -- ========================================
177 •  INSERT INTO Event (EventCreatorID, EventName, EventDescription, EventDate, EventTime, EventL
```



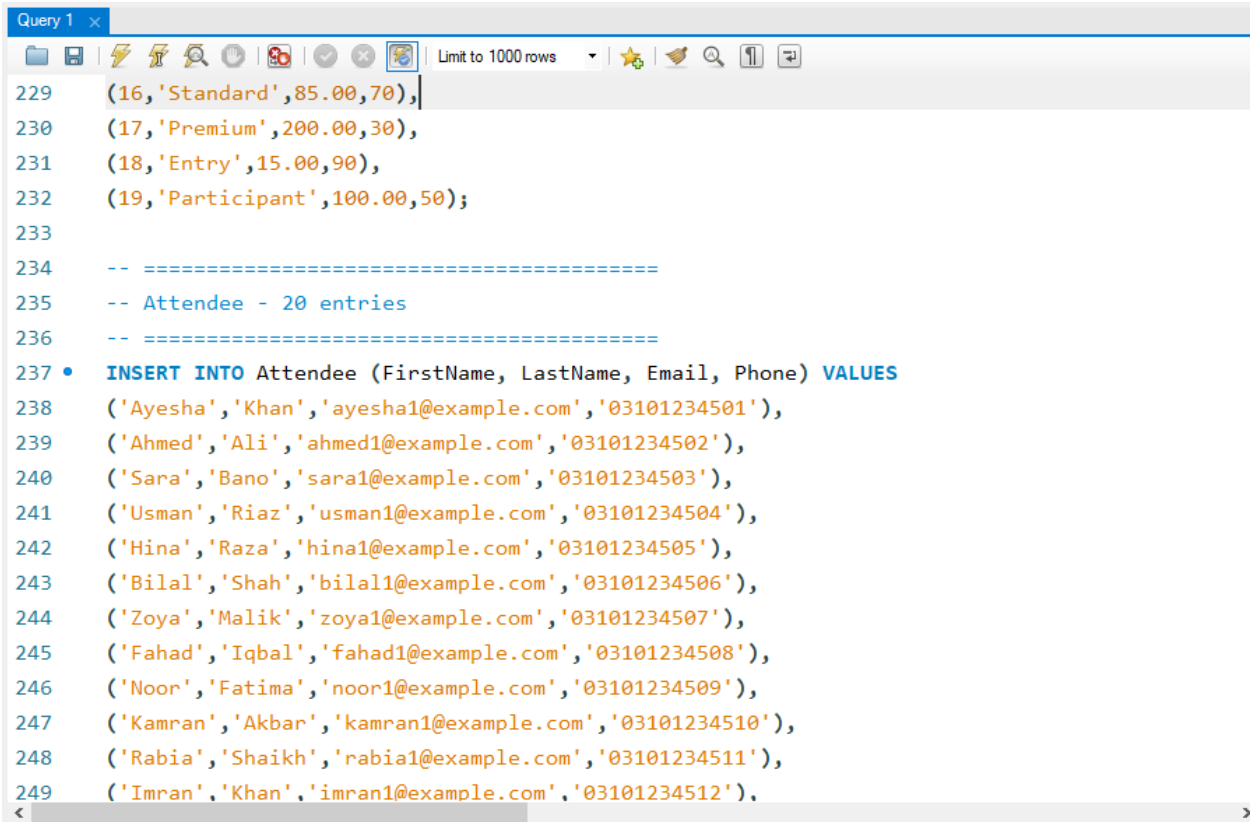
```
175    -- Event - 20 entries
176    -- ========================================
177 •  INSERT INTO Event (EventCreatorID, EventName, EventDescription, EventDate, EventTime, EventL
178    (1,'Tech Conference 2026','Annual tech conference','2026-03-15','10:00:00','Lahore Expo Cent
179    (2,'Sara & Ali Wedding','Wedding Ceremony','2026-05-20','18:00:00','Karachi Banquet Hall',2)
180    (3,'Music Concert','Live music event','2026-06-10','20:00:00','Islamabad Convention Center',
181    (4,'Startup Seminar','Business and tech seminar','2026-04-25','14:00:00','Multan Grand Hall'
182    (5,'Photography Workshop','Learn photography skills','2026-07-05','09:00:00','Peshawar Arena
183    (6,'Online Webinar','Educational online session','2026-02-20','15:00:00','Faisalabad Hall',6
184    (7,'Art Exhibition','Local artists display','2026-08-10','11:00:00','Quetta Auditorium',7),
185    (8,'Birthday Party','Private celebration','2026-03-30','18:00:00','Hyderabad Banquet',8),
186    (9,'Music Festival','Annual festival','2026-09-15','16:00:00','Sialkot Stadium',9),
187    (10,'Tech Meetup','Startup meet','2026-10-05','17:00:00','Rawalpindi Arena',10),
188    (11,'Coding Competition','Programming challenge','2026-11-10','10:00:00','Bahawalpur Hall',1
189    (12,'Employee Training','Corporate training','2026-01-25','09:00:00','Gujranwala Expo',12),
190    (13,'Charity Event','Fundraising event','2026-04-10','12:00:00','Sukkur Hall',13),
191    (14,'Product Launch','Tech product launch','2026-05-15','11:00:00','Mirpur Banquet',14),
192    (15,'Award Ceremony','Recognition event','2026-06-20','19:00:00','Muzaffargarh Center',15),
193    (16,'Networking Event','Professional networking','2026-07-25','15:00:00','Abbottabad Hall',1
194    (17,'Hackathon','24-hour coding','2026-08-30','08:00:00','Dera Ismail Khan Arena',17),
195    (18,'Sports Event','Local sports event','2026-09-20','14:00:00','Chitral Auditorium',18),
```



```
193    (16,'Networking Event','Professional networking','2026-07-25','15:00:00','Abbottabad Hall',1
194    (17,'Hackathon','24-hour coding','2026-08-30','08:00:00','Dera Ismail Khan Arena',17),
195    (18,'Sports Event','Local sports event','2026-09-20','14:00:00','Chitral Auditorium',18),
196    (19,'Cultural Event','Traditional performance','2026-10-10','18:00:00','Mardan Hall',19),
197    (20,'Art Show','Contemporary art show','2026-11-05','10:00:00','Nowshera Arena',20);
198
199    -- ========================================
200    -- EventVenue - 20 entries
201    -- ========================================
202 •  INSERT INTO EventVenue (EventID, VenueID) VALUES
203    (1,1),(2,2),(3,3),(4,4),(5,5),
204    (6,6),(7,7),(8,8),(9,9),(10,10),
205    (11,11),(12,12),(13,13),(14,14),(15,15),
206    (16,16),(17,17),(18,18),(19,19),(20,20);
207
208    -- ========================================
209    -- Ticket - 20 entries
210    -- ========================================
211 •  INSERT INTO Ticket (EventID, TicketType, TicketPrice, TicketQuantity) VALUES
212    (1,'Regular',50.00,200),
213    (1,'VIP',150.00,50),
```



```
Query 1
Limit to 1000 rows
211 • INSERT INTO Ticket (EventID, TicketType, TicketPrice, TicketQuantity) VALUES
212   (1,'Regular',50.00,200),
213   (1,'VIP',150.00,50),
214   (2,'Guest',0.00,100),
215   (3,'Standard',80.00,300),
216   (3,'Premium',200.00,50),
217   (4,'Entry',20.00,150),
218   (5,'Participant',100.00,40),
219   (6,'Regular',30.00,120),
220   (7,'VIP',100.00,50),
221   (8,'Guest',0.00,80),
222   (9,'Standard',75.00,200),
223   (10,'Premium',180.00,60),
224   (11,'Entry',25.00,100),
225   (12,'Participant',90.00,40),
226   (13,'Regular',40.00,80),
227   (14,'VIP',150.00,50),
228   (15,'Guest',0.00,60),
229   (16,'Standard',85.00,70),
230   (17,'Premium',200.00,30),
231   (18,'Entry',15.00,90),
```



```
Query 1
Limit to 1000 rows
229   (16,'Standard',85.00,70),
230   (17,'Premium',200.00,30),
231   (18,'Entry',15.00,90),
232   (19,'Participant',100.00,50);
233
234   -- ========================================
235   -- Attendee - 20 entries
236   -- ========================================
237 • INSERT INTO Attendee (FirstName, LastName, Email, Phone) VALUES
238   ('Ayesha','Khan','ayesha1@example.com','03101234501'),
239   ('Ahmed','Ali','ahmed1@example.com','03101234502'),
240   ('Sara','Bano','sara1@example.com','03101234503'),
241   ('Usman','Riaz','usman1@example.com','03101234504'),
242   ('Hina','Raza','hina1@example.com','03101234505'),
243   ('Bilal','Shah','bilal1@example.com','03101234506'),
244   ('Zoya','Malik','zoya1@example.com','03101234507'),
245   ('Fahad','Iqbal','fahad1@example.com','03101234508'),
246   ('Noor','Fatima','noor1@example.com','03101234509'),
247   ('Kamran','Akbar','kamran1@example.com','03101234510'),
248   ('Rabia','Shaikh','rabia1@example.com','03101234511'),
249   ('Imran','Khan','imran1@example.com','03101234512'),
```

Query 1

```
247    ('Kamran','Akbar','kamran1@example.com','03101234510'),
248    ('Rabia','Shaikh','rabia1@example.com','03101234511'),
249    ('Imran','Khan','imran1@example.com','03101234512'),
250    ('Sana','Malik','sana1@example.com','03101234513'),
251    ('Bilal','Hussain','bilal1@example.com','03101234514'),
252    ('Hina','Raza','hina2@example.com','03101234515'),
253    ('Tariq','Ahmed','tariq1@example.com','03101234516'),
254    ('Maryam','Ali','maryam1@example.com','03101234517'),
255    ('Shahbaz','Khan','shahbaz1@example.com','03101234518'),
256    ('Areeba','Noor','areeba1@example.com','03101234519'),
257    ('Ali','Raza','ali2@example.com','03101234520');
258
259    -- ========================================
260    -- Registration - 20 entries
261    -- ========================================
262 •  INSERT INTO Registration (EventID, AttendeeID, RegistrationDate, RegistrationStatus) VALUES
263    (1,1,CURDATE(),'Confirmed'),
264    (1,2,CURDATE(),'Confirmed'),
265    (2,3,CURDATE(),'Confirmed'),
266    (2,4,CURDATE(),'Confirmed'),
267    (3,5,CURDATE(),'Confirmed'),
```

Query 1

```
265    (2,3,CURDATE(),'Confirmed'),
266    (2,4,CURDATE(),'Confirmed'),
267    (3,5,CURDATE(),'Confirmed'),
268    (3,6,CURDATE(),'Confirmed'),
269    (4,7,CURDATE(),'Confirmed'),
270    (4,8,CURDATE(),'Confirmed'),
271    (5,9,CURDATE(),'Confirmed'),
272    (5,10,CURDATE(),'Confirmed'),
273    (6,11,CURDATE(),'Confirmed'),
274    (6,12,CURDATE(),'Confirmed'),
275    (7,13,CURDATE(),'Confirmed'),
276    (7,14,CURDATE(),'Confirmed'),
277    (8,15,CURDATE(),'Confirmed'),
278    (8,16,CURDATE(),'Confirmed'),
279    (9,17,CURDATE(),'Confirmed'),
280    (9,18,CURDATE(),'Confirmed'),
281    (10,19,CURDATE(),'Confirmed'),
282    (10,20,CURDATE(),'Confirmed');
283
284    -- ========================================
285    -- Payment - 20 entries
```

Query 1

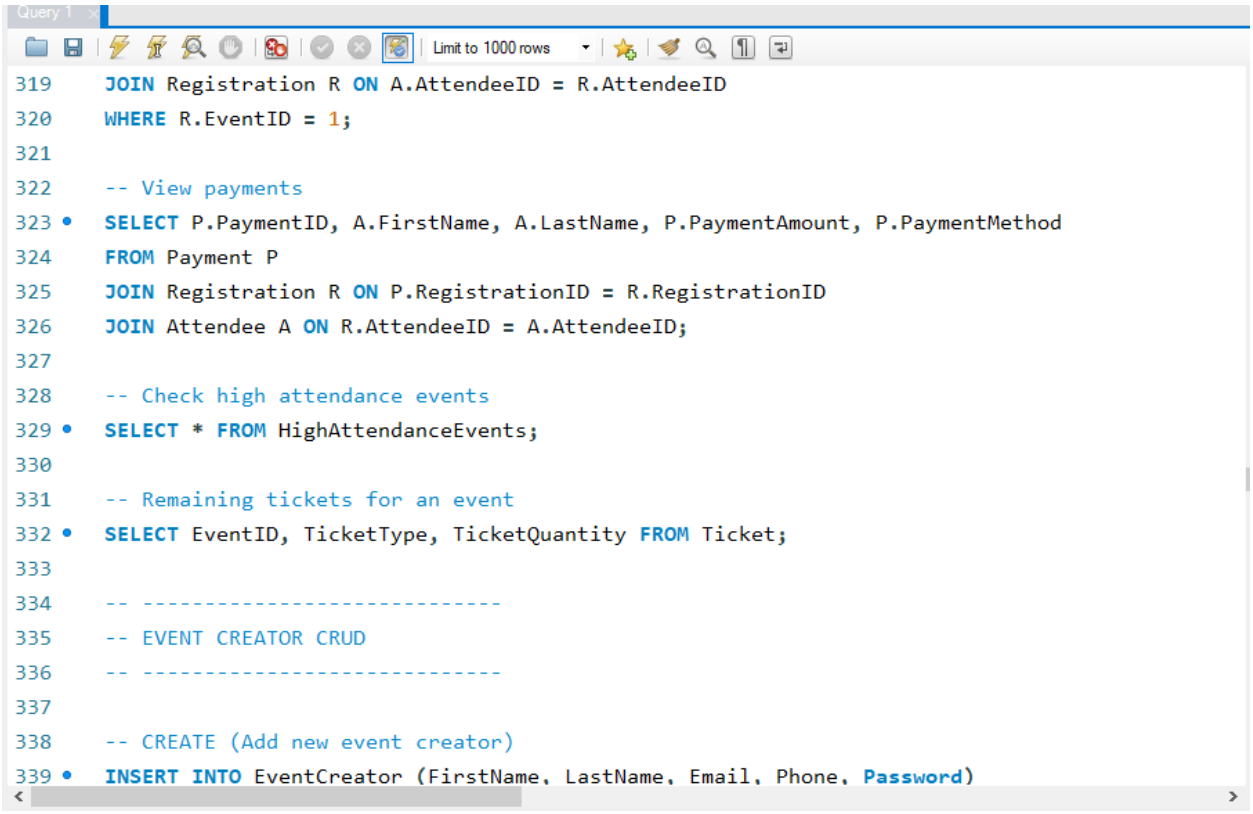
Limit to 1000 rows

```
283
284     -- ========================================
285     -- Payment - 20 entries
286     -- ========================================
287 •   INSERT INTO Payment (RegistrationID, PaymentDate, PaymentAmount, PaymentMethod) VALUES
288     (1,CURDATE(),50.00,'Card'),
289     (2,CURDATE(),150.00,'Card'),
290     (3,CURDATE(),0.00,'Free'),
291     (4,CURDATE(),0.00,'Free'),
292     (5,CURDATE(),80.00,'Card'),
293     (6,CURDATE(),200.00,'Card'),
294     (7,CURDATE(),20.00,'Cash'),
295     (8,CURDATE(),100.00,'Card'),
296     (9,CURDATE(),100.00,'Card'),
297     (10,CURDATE(),30.00,'Cash'),
298     (11,CURDATE(),25.00,'Card'),
299     (12,CURDATE(),90.00,'Card'),
300     (13,CURDATE(),40.00,'Card'),
301     (14,CURDATE(),150.00,'Card'),
302     (15,CURDATE(),0.00,'Free'),
303     (16,CURDATE(),85.00,'Card'),
```

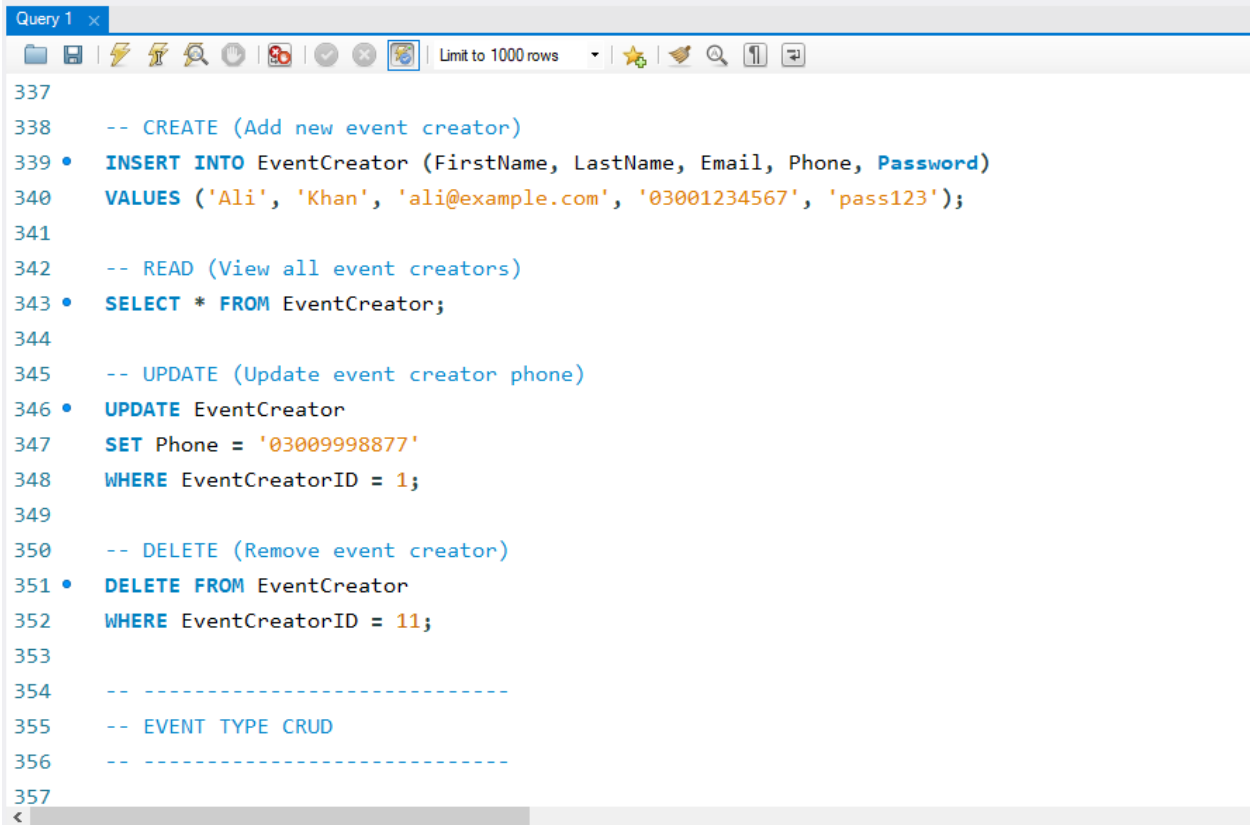
Query 1

Limit to 1000 rows

```
301     (14,CURDATE(),150.00,'Card'),
302     (15,CURDATE(),0.00,'Free'),
303     (16,CURDATE(),85.00,'Card'),
304     (17,CURDATE(),200.00,'Card'),
305     (18,CURDATE(),15.00,'Cash'),
306     (19,CURDATE(),100.00,'Card'),
307     (20,CURDATE(),100.00,'Card');
308
309
310     -- 6. Sample Queries
311     -- View all events with tickets
312 •   SELECT E.EventName, T.TicketType, T.TicketQuantity
313     FROM Event E
314     JOIN Ticket T ON E.EventID = T.EventID;
315
316     -- View attendees for an event
317 •   SELECT A.FirstName, A.LastName, R.RegistrationStatus
318     FROM Attendee A
319     JOIN Registration R ON A.AttendeeID = R.AttendeeID
320     WHERE R.EventID = 1;
321
```



```
319    JOIN Registration R ON A.AttendeeID = R.AttendeeID
320    WHERE R.EventID = 1;
321
322    -- View payments
323    SELECT P.PaymentID, A.FirstName, A.LastName, P.PaymentAmount, P.PaymentMethod
324    FROM Payment P
325    JOIN Registration R ON P.RegistrationID = R.RegistrationID
326    JOIN Attendee A ON R.AttendeeID = A.AttendeeID;
327
328    -- Check high attendance events
329    SELECT * FROM HighAttendanceEvents;
330
331    -- Remaining tickets for an event
332    SELECT EventID, TicketType, TicketQuantity FROM Ticket;
333
334    -- ----------------------------
335    -- EVENT CREATOR CRUD
336    -- ----------------------------
337
338    -- CREATE (Add new event creator)
339    INSERT INTO EventCreator (FirstName, LastName, Email, Phone, Password)
```



```
337
338    -- CREATE (Add new event creator)
339    INSERT INTO EventCreator (FirstName, LastName, Email, Phone, Password)
340    VALUES ('Ali', 'Khan', 'ali@example.com', '03001234567', 'pass123');
341
342    -- READ (View all event creators)
343    SELECT * FROM EventCreator;
344
345    -- UPDATE (Update event creator phone)
346    UPDATE EventCreator
347    SET Phone = '03009998877'
348    WHERE EventCreatorID = 1;
349
350    -- DELETE (Remove event creator)
351    DELETE FROM EventCreator
352    WHERE EventCreatorID = 11;
353
354    -- ----------------------------
355    -- EVENT TYPE CRUD
356    -- ----------------------------
357
```

Query 1

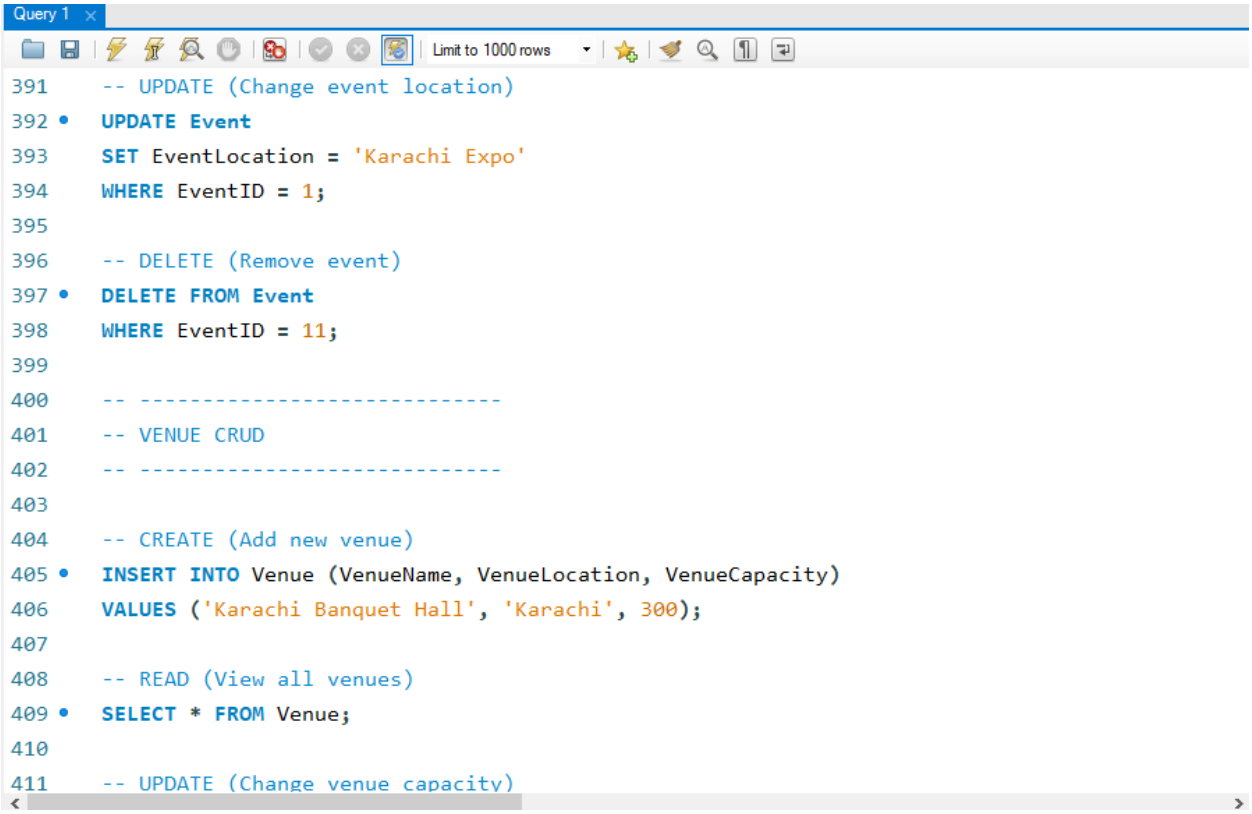
Limit to 1000 rows

```
355    -- EVENT TYPE CRUD
356    -- ------------------------------
357
358    -- CREATE (Add new event type)
359 •  INSERT INTO EventType (EventTypeName)
360    VALUES ('Seminar');
361
362    -- READ (View all event types)
363 •  SELECT * FROM EventType;
364
365    -- UPDATE (Change event type name)
366 •  UPDATE EventType
367    SET EventTypeName = 'Workshop'
368    WHERE EventTypeID = 1;
369
370    -- DELETE (Remove event type)
371 •  DELETE FROM EventType
372    WHERE EventTypeID = 11;
373
374    -- ------------------------------
375    -- EVENT CRUD
```

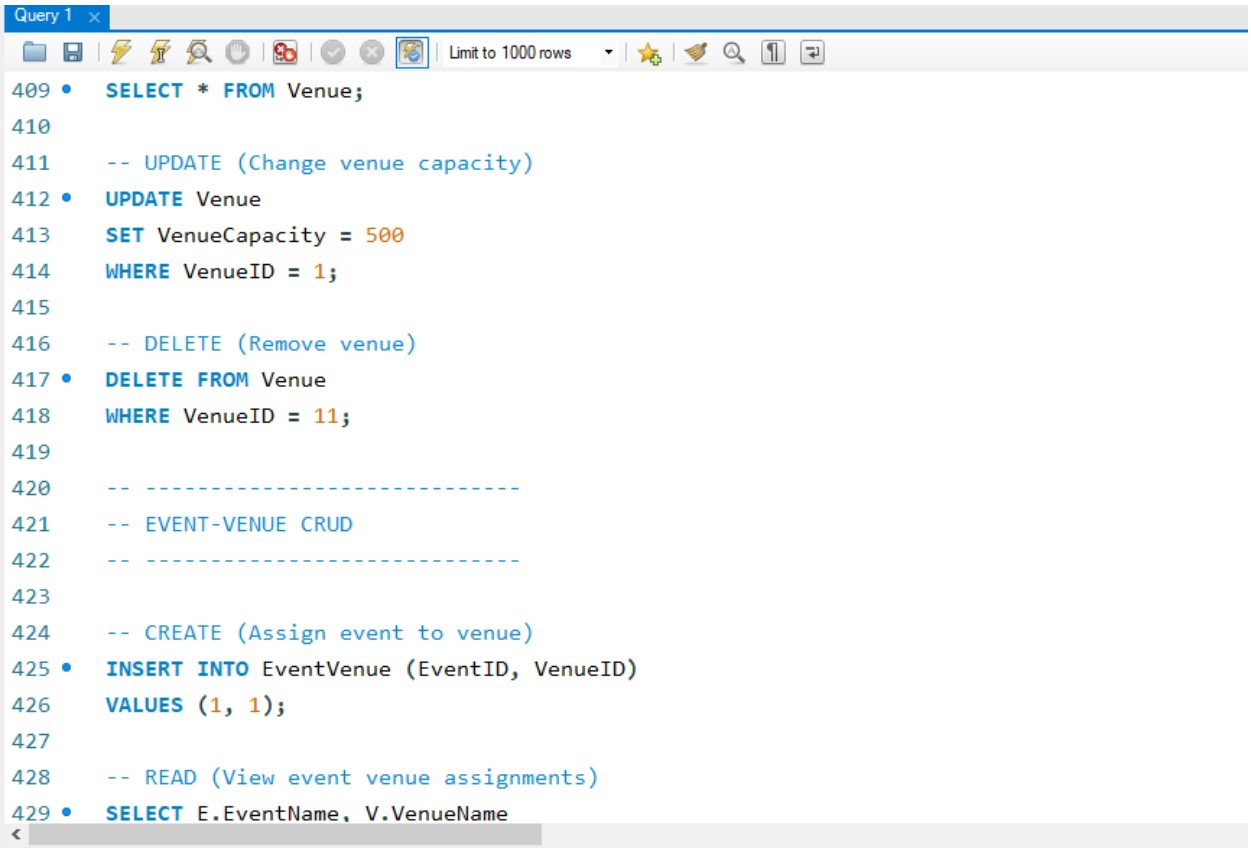
Query 1

Limit to 1000 rows

```
373
374    -- ------------------------------
375    -- EVENT CRUD
376    -- ------------------------------
377
378    -- CREATE (Add new event)
379 •  INSERT INTO Event (EventCreatorID, EventName, EventDescription, EventDate, EventTime, EventL
380    VALUES (1, 'Tech Summit 2026', 'Annual tech event', '2026-03-10', '10:00:00', 'Lahore Expo',
381
382    -- READ (View all events)
383 •  SELECT * FROM Event;
384
385    -- READ with JOIN to show creator and type
386 •  SELECT E.EventName, EC.FirstName, EC.LastName, ET.EventTypeName
387    FROM Event E
388    JOIN EventCreator EC ON E.EventCreatorID = EC.EventCreatorID
389    JOIN EventType ET ON E.EventType = ET.EventTypeID;
390
391    -- UPDATE (Change event location)
392 •  UPDATE Event
393    SET EventLocation = 'Karachi Expo'
```



```
391    -- UPDATE (Change event location)
392 •  UPDATE Event
393    SET EventLocation = 'Karachi Expo'
394    WHERE EventID = 1;
395
396    -- DELETE (Remove event)
397 •  DELETE FROM Event
398    WHERE EventID = 11;
399
400    -- ------------------------------
401    -- VENUE CRUD
402    -- ------------------------------
403
404    -- CREATE (Add new venue)
405 •  INSERT INTO Venue (VenueName, VenueLocation, VenueCapacity)
406    VALUES ('Karachi Banquet Hall', 'Karachi', 300);
407
408    -- READ (View all venues)
409 •  SELECT * FROM Venue;
410
411    -- UPDATE (Change venue capacity)
```



```
409 •  SELECT * FROM Venue;
410
411    -- UPDATE (Change venue capacity)
412 •  UPDATE Venue
413    SET VenueCapacity = 500
414    WHERE VenueID = 1;
415
416    -- DELETE (Remove venue)
417 •  DELETE FROM Venue
418    WHERE VenueID = 11;
419
420    -- ------------------------------
421    -- EVENT-VENUE CRUD
422    -- ------------------------------
423
424    -- CREATE (Assign event to venue)
425 •  INSERT INTO EventVenue (EventID, VenueID)
426    VALUES (1, 1);
427
428    -- READ (View event venue assignments)
429 •  SELECT E.EventName, V.VenueName
```

Query 1

Limit to 1000 rows

```
427
428     -- READ (View event venue assignments)
429 •   SELECT E.EventName, V.VenueName
430     FROM EventVenue EV
431     JOIN Event E ON EV.EventID = E.EventID
432     JOIN Venue V ON EV.VenueID = V.VenueID;
433
434     -- UPDATE (Change venue for an event)
435 •   UPDATE EventVenue
436     SET VenueID = 2
437     WHERE EventVenueID = 1;
438
439     -- DELETE (Remove event venue assignment)
440 •   DELETE FROM EventVenue
441     WHERE EventVenueID = 11;
442
443     -- ----------------------------
444     -- TICKET CRUD
445     -- ----------------------------
446
447     -- CREATE (Add ticket type for event)
```

Query 1

Limit to 1000 rows

```
442
443     -- ----------------------------
444     -- TICKET CRUD
445     -- ----------------------------
446
447     -- CREATE (Add ticket type for event)
448 •   INSERT INTO Ticket (EventID, TicketType, TicketPrice, TicketQuantity)
449     VALUES (1, 'VIP', 150.00, 50);
450
451     -- READ (View all tickets)
452 •   SELECT * FROM Ticket;
453
454     -- UPDATE (Change ticket quantity)
455 •   UPDATE Ticket
456     SET TicketQuantity = TicketQuantity + 20
457     WHERE TicketID = 1;
458
459     -- DELETE (Remove ticket type)
460 •   DELETE FROM Ticket
461     WHERE TicketID = 11;
462
```

Query 1

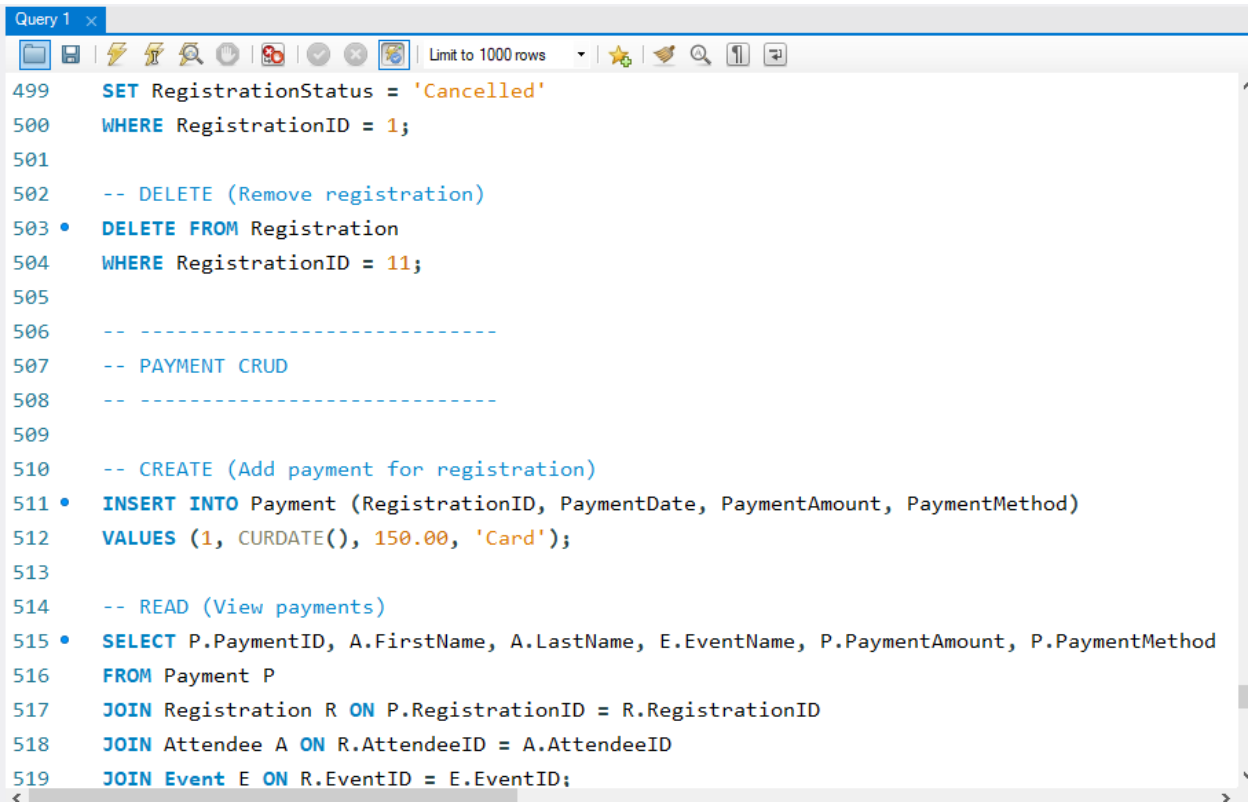
Limit to 1000 rows

```
460 •  DELETE FROM Ticket
461    WHERE TicketID = 11;
462
463    -- ----------------------------
464    -- ATTENDEE CRUD
465    -- ----------------------------
466
467    -- CREATE (Add attendee)
468 •  INSERT INTO Attendee (FirstName, LastName, Email, Phone)
469    VALUES ('Ayesha', 'Khan', 'ayesha@example.com', '03101234567');
470
471    -- READ (View all attendees)
472 •  SELECT * FROM Attendee;
473
474    -- UPDATE (Update attendee phone)
475 •  UPDATE Attendee
476    SET Phone = '03009998888'
477    WHERE AttendeeID = 1;
478
479    -- DELETE (Remove attendee)
480 •  DELETE FROM Attendee
```

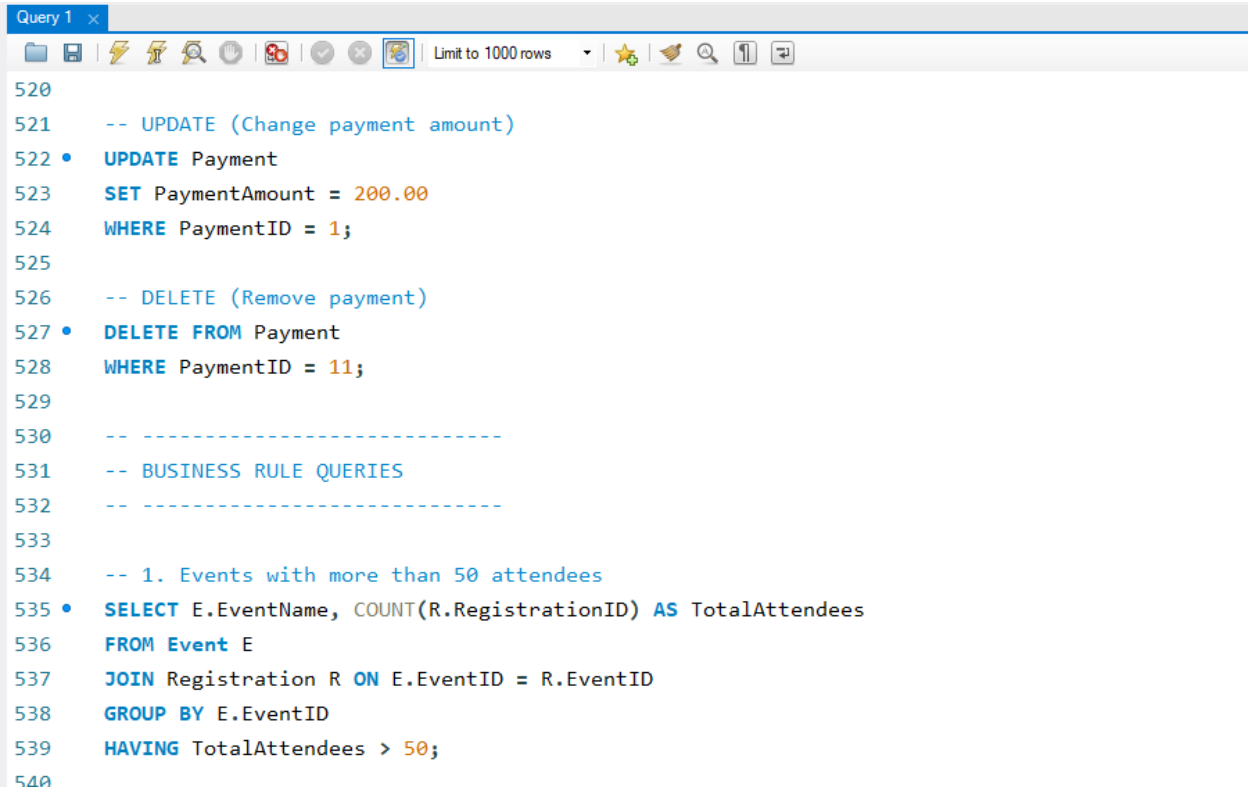
Query 1

Limit to 1000 rows

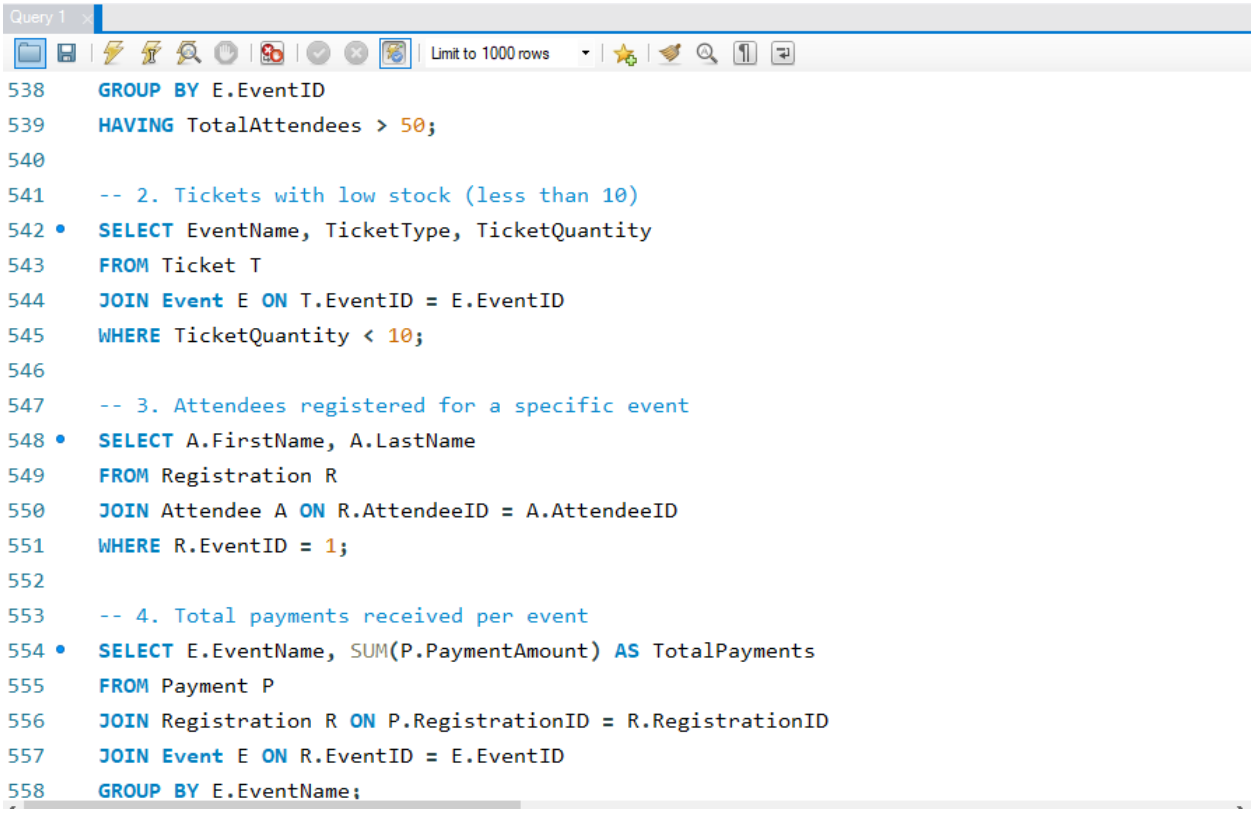
```
481    WHERE AttendeeID = 11;
482
483    -- ----------------------------
484    -- REGISTRATION CRUD
485    -- ----------------------------
486
487    -- CREATE (Register attendee for event)
488 •  INSERT INTO Registration (EventID, AttendeeID, RegistrationDate, RegistrationStatus)
489    VALUES (1, 1, CURDATE(), 'Confirmed');
490
491    -- READ (View registrations)
492 •  SELECT R.RegistrationID, A.FirstName, A.LastName, E.EventName, R.RegistrationStatus
493    FROM Registration R
494    JOIN Attendee A ON R.AttendeeID = A.AttendeeID
495    JOIN Event E ON R.EventID = E.EventID;
496
497    -- UPDATE (Change registration status)
498 •  UPDATE Registration
499    SET RegistrationStatus = 'Cancelled'
500    WHERE RegistrationID = 1;
501
```



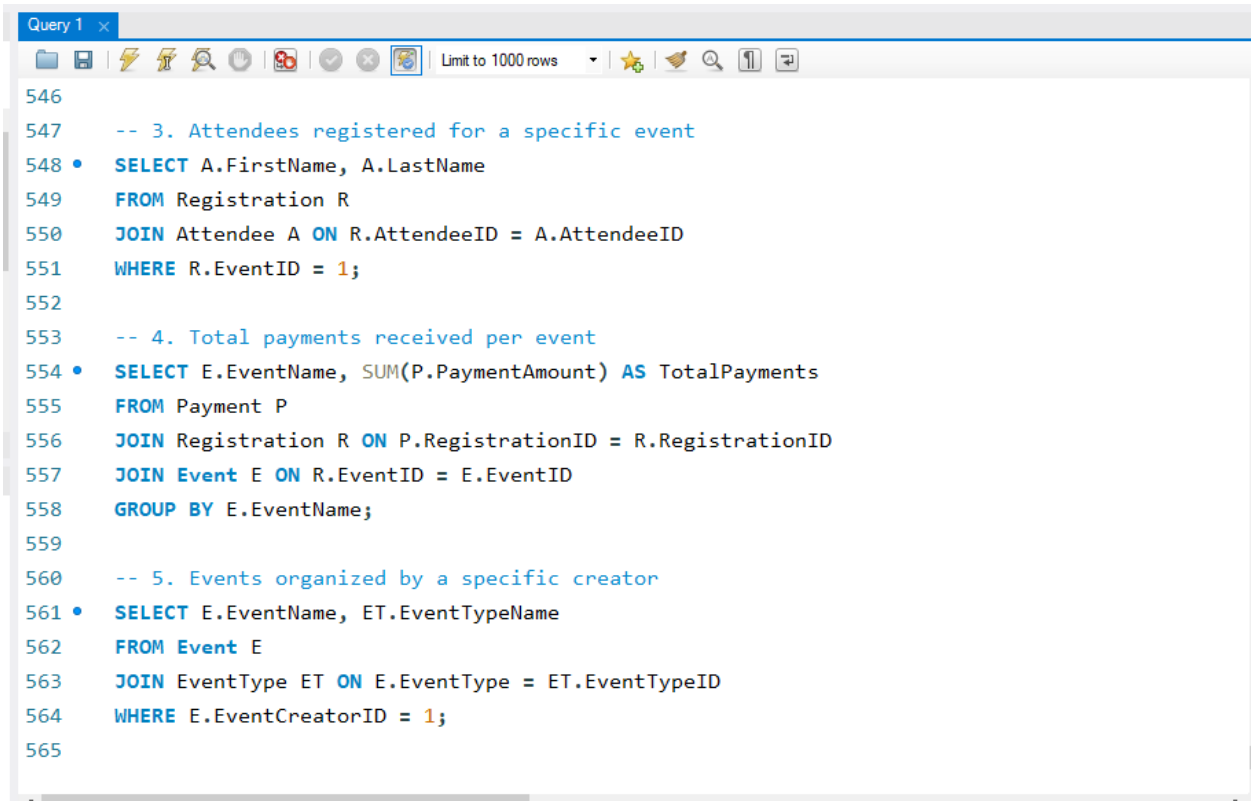
```
499     SET RegistrationStatus = 'Cancelled'
500     WHERE RegistrationID = 1;
501
502     -- DELETE (Remove registration)
503 •   DELETE FROM Registration
504     WHERE RegistrationID = 11;
505
506     -- ------------------------------
507     -- PAYMENT CRUD
508     -- ------------------------------
509
510     -- CREATE (Add payment for registration)
511 •   INSERT INTO Payment (RegistrationID, PaymentDate, PaymentAmount, PaymentMethod)
512     VALUES (1, CURDATE(), 150.00, 'Card');
513
514     -- READ (View payments)
515 •   SELECT P.PaymentID, A.FirstName, A.LastName, E.EventName, P.PaymentAmount, P.PaymentMethod
516     FROM Payment P
517     JOIN Registration R ON P.RegistrationID = R.RegistrationID
518     JOIN Attendee A ON R.AttendeeID = A.AttendeeID
519     JOIN Event E ON R.EventID = E.EventID;
```



```
520
521     -- UPDATE (Change payment amount)
522 •   UPDATE Payment
523     SET PaymentAmount = 200.00
524     WHERE PaymentID = 1;
525
526     -- DELETE (Remove payment)
527 •   DELETE FROM Payment
528     WHERE PaymentID = 11;
529
530     -- ------------------------------
531     -- BUSINESS RULE QUERIES
532     -- ------------------------------
533
534     -- 1. Events with more than 50 attendees
535 •   SELECT E.EventName, COUNT(R.RegistrationID) AS TotalAttendees
536     FROM Event E
537     JOIN Registration R ON E.EventID = R.EventID
538     GROUP BY E.EventID
539     HAVING TotalAttendees > 50;
540
```

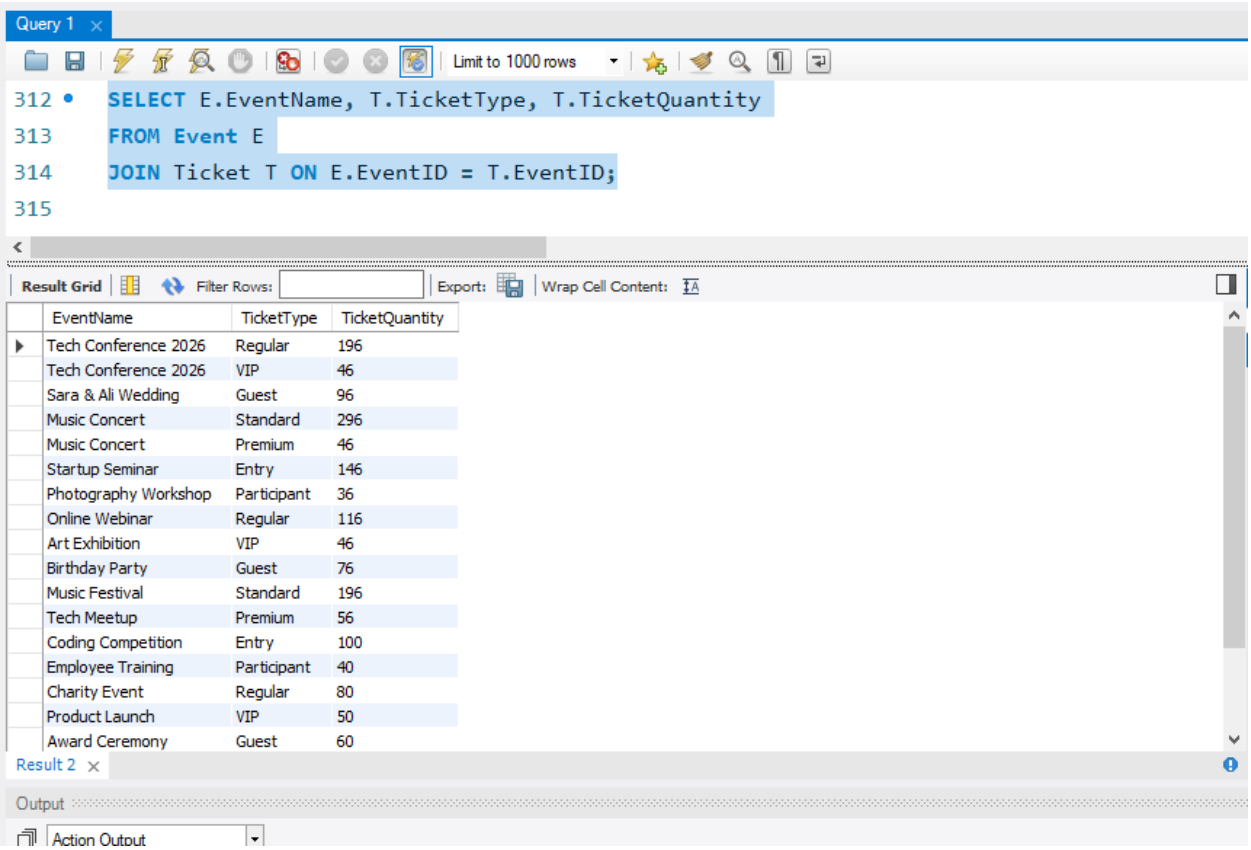


```
538     GROUP BY E.EventID
539     HAVING TotalAttendees > 50;
540
541     -- 2. Tickets with low stock (less than 10)
542 •   SELECT EventName, TicketType, TicketQuantity
543     FROM Ticket T
544     JOIN Event E ON T.EventID = E.EventID
545     WHERE TicketQuantity < 10;
546
547     -- 3. Attendees registered for a specific event
548 •   SELECT A.FirstName, A.LastName
549     FROM Registration R
550     JOIN Attendee A ON R.AttendeeID = A.AttendeeID
551     WHERE R.EventID = 1;
552
553     -- 4. Total payments received per event
554 •   SELECT E.EventName, SUM(P.PaymentAmount) AS TotalPayments
555     FROM Payment P
556     JOIN Registration R ON P.RegistrationID = R.RegistrationID
557     JOIN Event E ON R.EventID = E.EventID
558     GROUP BY E.EventName;
```



```
546
547     -- 3. Attendees registered for a specific event
548 •   SELECT A.FirstName, A.LastName
549     FROM Registration R
550     JOIN Attendee A ON R.AttendeeID = A.AttendeeID
551     WHERE R.EventID = 1;
552
553     -- 4. Total payments received per event
554 •   SELECT E.EventName, SUM(P.PaymentAmount) AS TotalPayments
555     FROM Payment P
556     JOIN Registration R ON P.RegistrationID = R.RegistrationID
557     JOIN Event E ON R.EventID = E.EventID
558     GROUP BY E.EventName;
559
560     -- 5. Events organized by a specific creator
561 •   SELECT E.EventName, ET.EventTypeName
562     FROM Event E
563     JOIN EventType ET ON E.EventType = ET.EventTypeID
564     WHERE E.EventCreatorID = 1;
565
```

OUTPUT:


Query 1
Limit to 1000 rows
312 • SELECT E.EventName, T.TicketType, T.TicketQuantity
313 FROM Event E
314 JOIN Ticket T ON E.EventID = T.EventID;
315
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
EventName TicketType TicketQuantity
Tech Conference 2026 Regular 196
Tech Conference 2026 VIP 46
Sara & Ali Wedding Guest 96
Music Concert Standard 296
Music Concert Premium 46
Startup Seminar Entry 146
Photography Workshop Participant 36
Online Webinar Regular 116
Art Exhibition VIP 46
Birthday Party Guest 76
Music Festival Standard 196
Tech Meetup Premium 56
Coding Competition Entry 100
Employee Training Participant 40
Charity Event Regular 80
Product Launch VIP 50
Award Ceremony Guest 60
Result 2
Output
Action Output

Query 1

Limit to 1000 rows

```
315
316     -- View attendees for an event
317 •   SELECT A.FirstName, A.LastName, R.RegistrationStatus
318     FROM Attendee A
319     JOIN Registration R ON A.AttendeeID = R.AttendeeID
320     WHERE R.EventID = 1;
```

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

FirstName	LastName	RegistrationStatus
Ayesha	Khan	Confirmed
Ahmed	Ali	Confirmed
Ayesha	Khan	Confirmed
Ahmed	Ali	Confirmed

Query 1

Limit to 1000 rows

```
322     -- View payments
323 •   SELECT P.PaymentID, A.FirstName, A.LastName, P.PaymentAmount, P.PaymentMethod
324     FROM Payment P
325     JOIN Registration R ON P.RegistrationID = R.RegistrationID
326     JOIN Attendee A ON R.AttendeeID = A.AttendeeID;
```

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

PaymentID	FirstName	LastName	PaymentAmount	PaymentMethod
1	Ayesha	Khan	50.00	Card
21	Ayesha	Khan	50.00	Card
2	Ahmed	Ali	150.00	Card
22	Ahmed	Ali	150.00	Card
3	Sara	Bano	0.00	Free
23	Sara	Bano	0.00	Free
4	Usman	Riaz	0.00	Free
24	Usman	Riaz	0.00	Free
5	Hina	Raza	80.00	Card
25	Hina	Raza	80.00	Card
6	Bilal	Shah	200.00	Card
26	Bilal	Shah	200.00	Card
7	Zoya	Malik	20.00	Cash
27	Zoya	Malik	20.00	Cash
8	Fahad	Iqbal	100.00	Card
28	Fahad	Iqbal	100.00	Card

Result 4

Query 1

```
325    JOIN Registration R ON P.RegistrationID = R.RegistrationID
326    JOIN Attendee A ON R.AttendeeID = A.AttendeeID;
327
328    -- Check high attendance events
329 •  SELECT * FROM HighAttendanceEvents;
330
331    -- Remaining tickets for an event
332 •  SELECT EventID, TicketType, TicketQuantity FROM Ticket;
333
334    -- ------------------------------
335    -- EVENT CREATOR CRUD
336    -- ------------------------------
```

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

EventName	EventDate	TotalRegistrations

HighAttendanceEvents 7

Query 1

```
331    -- Remaining tickets for an event
332 •  SELECT EventID, TicketType, TicketQuantity FROM Ticket;
333
334    -- ------------------------------
```

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

EventID	TicketType	TicketQuantity
1	Regular	196
1	VIP	46
2	Guest	96
3	Standard	296
3	Premium	46
4	Entry	146
5	Participant	36
6	Regular	116
7	VIP	46
8	Guest	76
9	Standard	196
10	Premium	56
11	Entry	100
12	Participant	40
13	Regular	80
14	VIP	50
15	Guest	60

Ticket 8

Query 1

```
342     -- READ (View all event creators)
343 •   SELECT * FROM EventCreator;
344
345     -- UPDATE (Update event creator phone)
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

EventCreatorID	FirstName	LastName	Email	Phone	Password
1	Ali	Khan	ali1@example.com	03001000001	pass123
2	Sara	Malik	sara1@example.com	03001000002	pass123
3	Ahmed	Raza	ahmed1@example.com	03001000003	pass123
4	Fatima	Noor	fatima1@example.com	03001000004	pass123
5	Hassan	Ali	hassan1@example.com	03001000005	pass123
6	Ayesha	Baloch	ayesha1@example.com	03001000006	pass123
7	Usman	Shah	usman1@example.com	03001000007	pass123
8	Zoya	Malik	zoya1@example.com	03001000008	pass123
9	Fahad	Iqbal	fahad1@example.com	03001000009	pass123
10	Noor	Fatima	noor1@example.com	03001000010	pass123
11	Kamran	Akbar	kamran1@example.com	03001000011	pass123
12	Rabia	Shaikh	rabia1@example.com	03001000012	pass123
13	Imran	Khan	imran1@example.com	03001000013	pass123
14	Sana	Malik	sana1@example.com	03001000014	pass123
15	Bilal	Hussain	bilal1@example.com	03001000015	pass123
16	Hina	Raza	hina1@example.com	03001000016	pass123
17	Tariq	Ahmed	tariq1@example.com	03001000017	pass123

EventCreator 9 | Apply

Query 1

```
361
362     -- READ (View all event types)
363 •   SELECT * FROM EventType;
364
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

EventTypeID	EventTypeName
1	Conference
2	Wedding
3	Concert
4	Seminar
5	Workshop
6	Webinar
7	Exhibition
8	Party
9	Festival
10	Meetup
11	Competition
12	Training
13	Charity Event
14	Product Launch
15	Award Ceremony
16	Networking Event
17	Hackathon

EventType 10 | Output

Query 1

```
381
382     -- READ (View all events)
383 •   SELECT * FROM Event;
384
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

EventID	EventCreatorID	EventName	EventDescription	EventDate	EventTime	EventLocation	EventType
1	1	Tech Conference 2026	Annual tech conference	2026-03-15	10:00:00	Lahore Expo Center	1
2	2	Sara & Ali Wedding	Wedding Ceremony	2026-05-20	18:00:00	Karachi Banquet Hall	2
3	3	Music Concert	Live music event	2026-06-10	20:00:00	Islamabad Convention Center	3
4	4	Startup Seminar	Business and tech seminar	2026-04-25	14:00:00	Multan Grand Hall	4
5	5	Photography Workshop	Learn photography skills	2026-07-05	09:00:00	Peshawar Arena	5
6	6	Online Webinar	Educational online session	2026-02-20	15:00:00	Faisalabad Hall	6
7	7	Art Exhibition	Local artists display	2026-08-10	11:00:00	Quetta Auditorium	7
8	8	Birthday Party	Private celebration	2026-03-30	18:00:00	Hyderabad Banquet	8
9	9	Music Festival	Annual festival	2026-09-15	16:00:00	Sialkot Stadium	9
10	10	Tech Meetup	Startup meet	2026-10-05	17:00:00	Rawalpindi Arena	10
11	11	Coding Competition	Programming challenge	2026-11-10	10:00:00	Bahawalpur Hall	11
12	12	Employee Training	Corporate training	2026-01-25	09:00:00	Gujranwala Expo	12
13	13	Charity Event	Fundraising event	2026-04-10	12:00:00	Sukkur Hall	13
14	14	Product Launch	Tech product launch	2026-05-15	11:00:00	Mirpur Banquet	14
15	15	Award Ceremony	Recognition event	2026-06-20	19:00:00	Muzaffargarh Center	15
16	16	Networking Event	Professional networking	2026-07-25	15:00:00	Abbottabad Hall	16
17	17	Hackathon	24-hour coding	2026-08-30	08:00:00	Dera Ismail Khan Arena	17

Event 11 | Apply

Query 1

```
384
385     -- READ with JOIN to show creator and type
386 •   SELECT E.EventName, EC.FirstName, EC.LastName, ET.EventTypeName
387     FROM Event E
388     JOIN EventCreator EC ON E.EventCreatorID = EC.EventCreatorID
389     JOIN EventType ET ON E.EventType = ET.EventTypeID;
390
```

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

EventName	FirstName	LastName	EventTypeName
Tech Conference 2026	Ali	Khan	Workshop
Tech Summit 2026	Ali	Khan	Workshop
Tech Summit 2026	Ali	Khan	Workshop
Sara & Ali Wedding	Sara	Malik	Wedding
Music Concert	Ahmed	Raza	Concert
Startup Seminar	Fatima	Noor	Seminar
Photography Workshop	Hassan	Ali	Workshop
Online Webinar	Ayesha	Baloch	Webinar
Art Exhibition	Usman	Shah	Exhibition
Birthday Party	Zoya	Malik	Party
Music Festival	Fahad	Iqbal	Festival
Tech Meetup	Noor	Fatima	Meetup
Coding Competition	Kamran	Akbar	Competition

Result 12

Output

Action Output

Query 1

```
409 •  SELECT * FROM Venue;
410
411    -- UPDATE (Change venue capacity)
412 •  UPDATE Venue
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

VenueID	VenueName	VenueLocation	VenueCapacity
1	Lahore Expo Center	Lahore	500
2	Karachi Banquet Hall	Karachi	300
3	Islamabad Convention Center	Islamabad	400
4	Multan Grand Hall	Multan	250
5	Peshawar Arena	Peshawar	350
6	Faisalabad Hall	Faisalabad	200
7	Quetta Auditorium	Quetta	150
8	Hyderabad Banquet	Hyderabad	180
9	Sialkot Stadium	Sialkot	400
10	Rawalpindi Arena	Rawalpindi	300
11	Bahawalpur Hall	Bahawalpur	220
12	Gujranwala Expo	Gujranwala	350
13	Sukkur Hall	Sukkur	180
14	Mirpur Banquet	Mirpur	200
15	Muzaffargarh Center	Muzaffargarh	250
16	Abbottabad Hall	Abbottabad	150
17	Dera Ismail Khan Arena	DI Khan	300

Venue 13 | Apply

Query 1

```
427
428    -- READ (View event venue assignments)
429 •  SELECT E.EventName, V.VenueName
430    FROM EventVenue EV
431    JOIN Event E ON EV.EventID = E.EventID
432    JOIN Venue V ON EV.VenueID = V.VenueID;
```

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

EventName	VenueName
Tech Conference 2026	Lahore Expo Center
Tech Conference 2026	Lahore Expo Center
Tech Conference 2026	Lahore Expo Center
Sara & Ali Wedding	Karachi Banquet Hall
Sara & Ali Wedding	Karachi Banquet Hall
Music Concert	Islamabad Convention Center
Music Concert	Islamabad Convention Center
Startup Seminar	Multan Grand Hall
Startup Seminar	Multan Grand Hall
Photography Workshop	Peshawar Arena
Photography Workshop	Peshawar Arena
Online Webinar	Faisalabad Hall
Online Webinar	Faisalabad Hall
Art Exhibition	Quetta Auditorium

Result 14

Query 1

```
451     -- READ (View all tickets)
452 •   SELECT * FROM Ticket;
453
454     -- UPDATE (Change ticket quantity)
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

TicketID	EventID	TicketType	TicketPrice	TicketQuantity
1	1	Regular	50.00	196
2	1	VIP	150.00	46
3	2	Guest	0.00	96
4	3	Standard	80.00	296
5	3	Premium	200.00	46
6	4	Entry	20.00	146
7	5	Participant	100.00	36
8	6	Regular	30.00	116
9	7	VIP	100.00	46
10	8	Guest	0.00	76
11	9	Standard	75.00	196
12	10	Premium	180.00	56
13	11	Entry	25.00	100
14	12	Participant	90.00	40
15	13	Regular	40.00	80
16	14	VIP	150.00	50
17	15	Guest	0.00	60

Ticket 15 | Apply

Output

Query 1

```
469     VALUES ('Ayesha', 'Khan', 'ayesha@example.com', '03101234567');
470
471     -- READ (View all attendees)
472 •   SELECT * FROM Attendee;
473
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

AttendeeID	FirstName	LastName	Email	Phone
1	Ayesha	Khan	ayesha1@example.com	03101234501
2	Ahmed	Ali	ahmed1@example.com	03101234502
3	Sara	Bano	sara1@example.com	03101234503
4	Usman	Riaz	usman1@example.com	03101234504
5	Hina	Raza	hina1@example.com	03101234505
6	Bilal	Shah	bilal1@example.com	03101234506
7	Zoya	Malik	zoya1@example.com	03101234507
8	Fahad	Iqbal	fahad1@example.com	03101234508
9	Noor	Fatima	noor1@example.com	03101234509
10	Kamran	Akbar	kamran1@example.com	03101234510
11	Rabia	Shaikh	rabia1@example.com	03101234511
12	Imran	Khan	imran1@example.com	03101234512
13	Sana	Malik	sana1@example.com	03101234513
14	Bilal	Hussain	bilal1@example.com	03101234514
15	Hina	Raza	hina2@example.com	03101234515
16	Tariq	Ahmed	tariq1@example.com	03101234516

Attendee 16 | Apply

Query 1

Limit to 1000 rows

```
490
491     -- READ (View registrations)
492 •   SELECT R.RegistrationID, A.FirstName, A.LastName, E.EventName, R.RegistrationStatus
493     FROM Registration R
494     JOIN Attendee A ON R.AttendeeID = A.AttendeeID
495     JOIN Event E ON R.EventID = E.EventID;
```

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

RegistrationID	FirstName	LastName	EventName	RegistrationStatus
1	Ayesha	Khan	Tech Conference 2026	Confirmed
21	Ayesha	Khan	Tech Conference 2026	Confirmed
41	Ayesha	Khan	Tech Conference 2026	Confirmed
2	Ahmed	Ali	Tech Conference 2026	Confirmed
22	Ahmed	Ali	Tech Conference 2026	Confirmed
3	Sara	Bano	Sara & Ali Wedding	Confirmed
23	Sara	Bano	Sara & Ali Wedding	Confirmed
4	Usman	Riaz	Sara & Ali Wedding	Confirmed
24	Usman	Riaz	Sara & Ali Wedding	Confirmed
5	Hina	Raza	Music Concert	Confirmed
25	Hina	Raza	Music Concert	Confirmed
6	Bilal	Shah	Music Concert	Confirmed
26	Bilal	Shah	Music Concert	Confirmed
7	Zoya	Malik	Startup Seminar	Confirmed

Result 17

Query 1

Limit to 1000 rows

```
514     -- READ (View payments)
515 •   SELECT P.PaymentID, A.FirstName, A.LastName, E.EventName, P.PaymentAmount, P.PaymentMethod
516     FROM Payment P
517     JOIN Registration R ON P.RegistrationID = R.RegistrationID
518     JOIN Attendee A ON R.AttendeeID = A.AttendeeID
519     JOIN Event E ON R.EventID = E.EventID;
```

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

PaymentID	FirstName	LastName	EventName	PaymentAmount	PaymentMethod
1	Ayesha	Khan	Tech Conference 2026	50.00	Card
21	Ayesha	Khan	Tech Conference 2026	50.00	Card
41	Ayesha	Khan	Tech Conference 2026	150.00	Card
42	Ayesha	Khan	Tech Conference 2026	150.00	Card
2	Ahmed	Ali	Tech Conference 2026	150.00	Card
22	Ahmed	Ali	Tech Conference 2026	150.00	Card
3	Sara	Bano	Sara & Ali Wedding	0.00	Free
23	Sara	Bano	Sara & Ali Wedding	0.00	Free
4	Usman	Riaz	Sara & Ali Wedding	0.00	Free
24	Usman	Riaz	Sara & Ali Wedding	0.00	Free
5	Hina	Raza	Music Concert	80.00	Card
25	Hina	Raza	Music Concert	80.00	Card
6	Bilal	Shah	Music Concert	200.00	Card
26	Bilal	Shah	Music Concert	200.00	Card

Result 18

Output

Query 1

Limit to 1000 rows

```
535 •  SELECT E.EventName, COUNT(R.RegistrationID) AS TotalAttendees
536    FROM Event E
537    JOIN Registration R ON E.EventID = R.EventID
538    GROUP BY E.EventID
539    HAVING TotalAttendees > 50;
540
541    -- 2. Tickets with low stock (less than 10)
542 •  SELECT EventName, TicketType, TicketQuantity
543    FROM Ticket T
544    JOIN Event E ON T.EventID = E.EventID
```

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

EventName	TotalAttendees

Query 1

Limit to 1000 rows

```
538    GROUP BY E.EventID
539    HAVING TotalAttendees > 50;
540
541    -- 2. Tickets with low stock (less than 10)
542 •  SELECT EventName, TicketType, TicketQuantity
543    FROM Ticket T
544    JOIN Event E ON T.EventID = E.EventID
545    WHERE TicketQuantity < 10;
546
547    -- 3. Attendees registered for a specific event
```

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

EventName	TicketType	TicketQuantity

Query 1

```
544     JOIN Event E ON T.EventID = E.EventID
545     WHERE TicketQuantity < 10;
546
547     -- 3. Attendees registered for a specific event
548     SELECT A.FirstName, A.LastName
549     FROM Registration R
550     JOIN Attendee A ON R.AttendeeID = A.AttendeeID
551     WHERE R.EventID = 1;
552
553     -- 4. Total payments received per event
```

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

FirstName	LastName
Ayesha	Khan
Ahmed	Ali
Ayesha	Khan
Ahmed	Ali
Ayesha	Khan

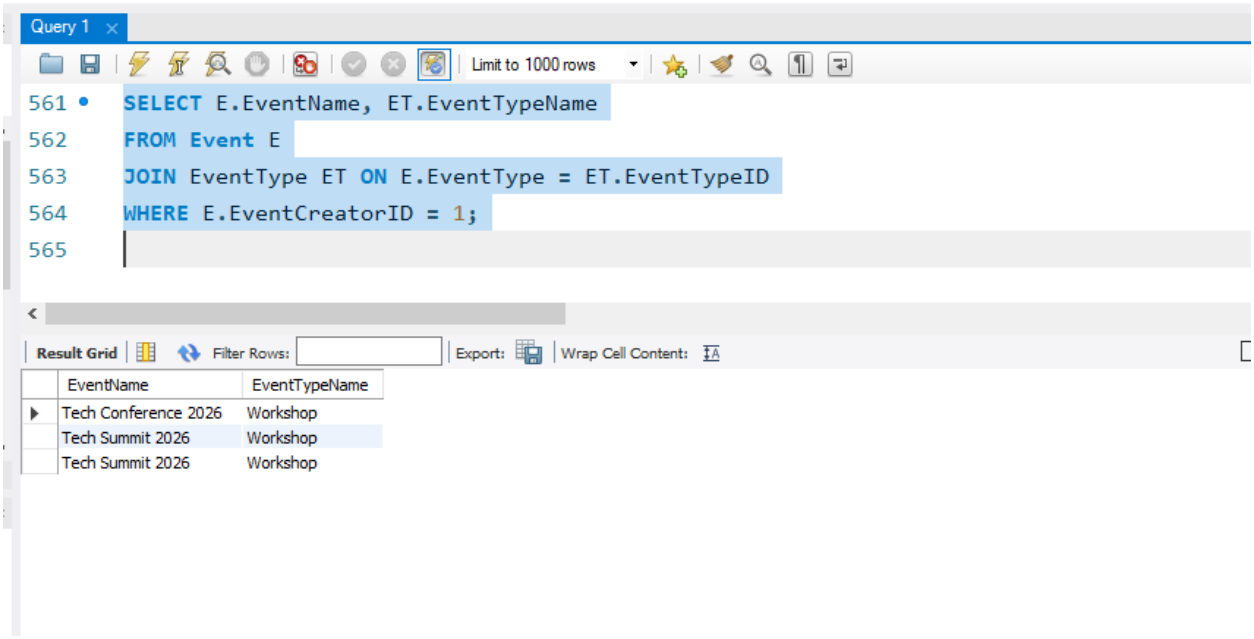
Query 1

```
554     SELECT E.EventName, SUM(P.PaymentAmount) AS TotalPayments
555     FROM Payment P
556     JOIN Registration R ON P.RegistrationID = R.RegistrationID
557     JOIN Event E ON R.EventID = E.EventID
558     GROUP BY E.EventName;
559
```

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

EventName	TotalPayments
Tech Conference 2026	850.00
Sara & Ali Wedding	0.00
Music Concert	560.00
Startup Seminar	240.00
Photography Workshop	260.00
Online Webinar	205.00
Art Exhibition	380.00
Birthday Party	170.00
Music Festival	430.00
Tech Meetup	400.00

Result 22


Query 1
Limit to 1000 rows
561 SELECT E.EventName, ET.EventTypeName
562 FROM Event E
563 JOIN EventType ET ON E.EventType = ET.EventTypeID
564 WHERE E.EventCreatorID = 1;
565
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
EventName
EventTypeName
Tech Conference 2026
Workshop
Tech Summit 2026
Workshop
Tech Summit 2026
Workshop

SOURCE CODE:

```
-- Create database
CREATE DATABASE EventManagement;
USE EventManagement;

-- Table: EventCreator
CREATE TABLE EventCreator (
    EventCreatorID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(100),
    LastName VARCHAR(100),
    Email VARCHAR(100) UNIQUE,
    Phone VARCHAR(20),
    Password VARCHAR(255)
);

-- Table: EventType
CREATE TABLE EventType (
    EventTypeID INT PRIMARY KEY AUTO_INCREMENT,
    EventTypeName VARCHAR(100)
);

-- Table: Event
CREATE TABLE Event (
    EventID INT PRIMARY KEY AUTO_INCREMENT,
    EventCreatorID INT,
    EventName VARCHAR(100),
    EventDescription TEXT,
    EventDate DATE,
    EventTime TIME,
    EventLocation VARCHAR(100),
```

```
    EventType INT,
    FOREIGN KEY (EventCreatorID) REFERENCES EventCreator(EventCreatorID),
    FOREIGN KEY (EventType) REFERENCES EventType(EventTypeID)
);

-- Table: Attendee
CREATE TABLE Attendee (
    AttendeeID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(100),
    LastName VARCHAR(100),
    Email VARCHAR(100),
    Phone VARCHAR(20)
);

-- Table: Registration
CREATE TABLE Registration (
    RegistrationID INT PRIMARY KEY AUTO_INCREMENT,
    EventID INT,
    AttendeeID INT,
    RegistrationDate DATE,
    RegistrationStatus VARCHAR(50),
    FOREIGN KEY (EventID) REFERENCES Event(EventID),
    FOREIGN KEY (AttendeeID) REFERENCES Attendee(AttendeeID)
);

-- Table: Payment
CREATE TABLE Payment (
    PaymentID INT PRIMARY KEY AUTO_INCREMENT,
    RegistrationID INT,
    PaymentDate DATE,
    PaymentAmount DECIMAL(10,2),
```

```
    PaymentMethod VARCHAR(50),
    FOREIGN KEY (RegistrationID) REFERENCES Registration(RegistrationID)
);

-- Table: Ticket
CREATE TABLE Ticket (
    TicketID INT PRIMARY KEY AUTO_INCREMENT,
    EventID INT,
    TicketType VARCHAR(50),
    TicketPrice DECIMAL(10,2),
    TicketQuantity INT,
    FOREIGN KEY (EventID) REFERENCES Event(EventID)
);

-- Table: Venue
CREATE TABLE Venue (
    VenueID INT PRIMARY KEY AUTO_INCREMENT,
    VenueName VARCHAR(100),
    VenueLocation VARCHAR(100),
    VenueCapacity INT
);

-- Table: EventVenue
CREATE TABLE EventVenue (
    EventVenueID INT PRIMARY KEY AUTO_INCREMENT,
    EventID INT,
    VenueID INT,
    FOREIGN KEY (EventID) REFERENCES Event(EventID),
    FOREIGN KEY (VenueID) REFERENCES Venue(VenueID)
);
-- 3. Trigger: Update Ticket Quantity after Registration
```

```
DELIMITER //

CREATE TRIGGER UpdateTicketAfterRegistration
AFTER INSERT ON Registration
FOR EACH ROW
BEGIN
   UPDATE Ticket
   SET TicketQuantity = TicketQuantity - 1
   WHERE EventID = NEW.EventID
   AND TicketQuantity > 0;
END;
//

DELIMITER ;

-- 4. View: High Attendance Events
CREATE VIEW HighAttendanceEvents AS
SELECT E.EventName, E.EventDate, COUNT(R.RegistrationID) AS TotalRegistrations
FROM Event E
JOIN Registration R ON E.EventID = R.EventID
GROUP BY E.EventID
HAVING TotalRegistrations > 50;

-- ========================================
-- EventCreator - 20 entries
-- ========================================
INSERT INTO EventCreator (FirstName, LastName, Email, Phone, Password) VALUES
('Ali','Khan','ali1@example.com','03001000001','pass123'),
('Sara','Malik','sara1@example.com','03001000002','pass123'),
('Ahmed','Raza','ahmed1@example.com','03001000003','pass123'),
('Fatima','Noor','fatima1@example.com','03001000004','pass123'),
```

```
('Hassan','Ali','hassan1@example.com','03001000005','pass123'),
('Ayesha','Baloch','ayesha1@example.com','03001000006','pass123'),
('Usman','Shah','usman1@example.com','03001000007','pass123'),
('Zoya','Malik','zoya1@example.com','03001000008','pass123'),
('Fahad','Iqbal','fahad1@example.com','03001000009','pass123'),
('Noor','Fatima','noor1@example.com','03001000010','pass123'),
('Kamran','Akbar','kamran1@example.com','03001000011','pass123'),
('Rabia','Shaikh','rabia1@example.com','03001000012','pass123'),
('Imran','Khan','imran1@example.com','03001000013','pass123'),
('Sana','Malik','sana1@example.com','03001000014','pass123'),
('Bilal','Hussain','bilal1@example.com','03001000015','pass123'),
('Hina','Raza','hina1@example.com','03001000016','pass123'),
('Tariq','Ahmed','tariq1@example.com','03001000017','pass123'),
('Maryam','Ali','maryam1@example.com','03001000018','pass123'),
('Shahbaz','Khan','shahbaz1@example.com','03001000019','pass123'),
('Areeba','Noor','areeba1@example.com','03001000020','pass123');

-- ========================================
-- EventType - 20 entries
-- ========================================
INSERT INTO EventType (EventTypeName) VALUES
('Conference'),('Wedding'),('Concert'),('Seminar'),('Workshop'),
('Webinar'),('Exhibition'),('Party'),('Festival'),('Meetup'),
('Competition'),('Training'),('Charity Event'),('Product Launch'),('Award Ceremony'),
('Networking Event'),('Hackathon'),('Sports Event'),('Cultural Event'),('Art Show');

-- ========================================
-- Venue - 20 entries
-- ========================================
INSERT INTO Venue (VenueName, VenueLocation, VenueCapacity) VALUES
('Lahore Expo Center','Lahore',500),
```

('Karachi Banquet Hall','Karachi',300),

('Islamabad Convention Center','Islamabad',400),

('Multan Grand Hall','Multan',250),

('Peshawar Arena','Peshawar',350),

('Faisalabad Hall','Faisalabad',200),

('Quetta Auditorium','Quetta',150),

('Hyderabad Banquet','Hyderabad',180),

('Sialkot Stadium','Sialkot',400),

('Rawalpindi Arena','Rawalpindi',300),

('Bahawalpur Hall','Bahawalpur',220),

('Gujranwala Expo','Gujranwala',350),

('Sukkur Hall','Sukkur',180),

('Mirpur Banquet','Mirpur',200),

('Muzaffargarh Center','Muzaffargarh',250),

('Abbottabad Hall','Abbottabad',150),

('Dera Ismail Khan Arena','DI Khan',300),

('Chitral Auditorium','Chitral',100),

('Mardan Hall','Mardan',200),

('Nowshera Arena','Nowshera',150);

-- ==

-- Event - 20 entries

-- ==

INSERT INTO Event (EventCreatorID, EventName, EventDescription, EventDate, EventTime, EventLocation, EventType) VALUES

(1,'Tech Conference 2026','Annual tech conference','2026-03-15','10:00:00','Lahore Expo Center',1),

(2,'Sara & Ali Wedding','Wedding Ceremony','2026-05-20','18:00:00','Karachi Banquet Hall',2),

(3,'Music Concert','Live music event','2026-06-10','20:00:00','Islamabad Convention Center',3),

(4,'Startup Seminar','Business and tech seminar','2026-04-25','14:00:00','Multan Grand Hall',4),

(5,'Photography Workshop','Learn photography skills','2026-07-05','09:00:00','Peshawar Arena',5),

(6,'Online Webinar','Educational online session','2026-02-20','15:00:00','Faisalabad Hall',6),

(7,'Art Exhibition','Local artists display','2026-08-10','11:00:00','Quetta Auditorium',7),

(8,'Birthday Party','Private celebration','2026-03-30','18:00:00','Hyderabad Banquet',8),

(9,'Music Festival','Annual festival','2026-09-15','16:00:00','Sialkot Stadium',9),

(10,'Tech Meetup','Startup meet','2026-10-05','17:00:00','Rawalpindi Arena',10),

(11,'Coding Competition','Programming challenge','2026-11-10','10:00:00','Bahawalpur Hall',11),

(12,'Employee Training','Corporate training','2026-01-25','09:00:00','Gujranwala Expo',12),

(13,'Charity Event','Fundraising event','2026-04-10','12:00:00','Sukkur Hall',13),

(14,'Product Launch','Tech product launch','2026-05-15','11:00:00','Mirpur Banquet',14),

(15,'Award Ceremony','Recognition event','2026-06-20','19:00:00','Muzaffargarh Center',15),

(16,'Networking Event','Professional networking','2026-07-25','15:00:00','Abbottabad Hall',16),

(17,'Hackathon','24-hour coding','2026-08-30','08:00:00','Dera Ismail Khan Arena',17),

(18,'Sports Event','Local sports event','2026-09-20','14:00:00','Chitral Auditorium',18),

(19,'Cultural Event','Traditional performance','2026-10-10','18:00:00','Mardan Hall',19),

(20,'Art Show','Contemporary art show','2026-11-05','10:00:00','Nowshera Arena',20);

-- ==

-- EventVenue - 20 entries

-- ==

INSERT INTO EventVenue (EventID, VenueID) VALUES

(1,1),(2,2),(3,3),(4,4),(5,5),

(6,6),(7,7),(8,8),(9,9),(10,10),

(11,11),(12,12),(13,13),(14,14),(15,15),

(16,16),(17,17),(18,18),(19,19),(20,20);

-- ==

-- Ticket - 20 entries

-- ==

INSERT INTO Ticket (EventID, TicketType, TicketPrice, TicketQuantity) VALUES

(1,'Regular',50.00,200),

(1,'VIP',150.00,50),

(2,'Guest',0.00,100),

```
(3,'Standard',80.00,300),
(3,'Premium',200.00,50),
(4,'Entry',20.00,150),
(5,'Participant',100.00,40),
(6,'Regular',30.00,120),
(7,'VIP',100.00,50),
(8,'Guest',0.00,80),
(9,'Standard',75.00,200),
(10,'Premium',180.00,60),
(11,'Entry',25.00,100),
(12,'Participant',90.00,40),
(13,'Regular',40.00,80),
(14,'VIP',150.00,50),
(15,'Guest',0.00,60),
(16,'Standard',85.00,70),
(17,'Premium',200.00,30),
(18,'Entry',15.00,90),
(19,'Participant',100.00,50);

-- ========================================
-- Attendee - 20 entries
-- ========================================
INSERT INTO Attendee (FirstName, LastName, Email, Phone) VALUES
('Ayesha','Khan','ayesha1@example.com','03101234501'),
('Ahmed','Ali','ahmed1@example.com','03101234502'),
('Sara','Bano','sara1@example.com','03101234503'),
('Usman','Riaz','usman1@example.com','03101234504'),
('Hina','Raza','hina1@example.com','03101234505'),
('Bilal','Shah','bilal1@example.com','03101234506'),
('Zoya','Malik','zoya1@example.com','03101234507'),
('Fahad','Iqbal','fahad1@example.com','03101234508'),
```

('Noor','Fatima','noor1@example.com','03101234509'),

('Kamran','Akbar','kamran1@example.com','03101234510'),

('Rabia','Shaikh','rabia1@example.com','03101234511'),

('Imran','Khan','imran1@example.com','03101234512'),

('Sana','Malik','sana1@example.com','03101234513'),

('Bilal','Hussain','bilal1@example.com','03101234514'),

('Hina','Raza','hina2@example.com','03101234515'),

('Tariq','Ahmed','tariq1@example.com','03101234516'),

('Maryam','Ali','maryam1@example.com','03101234517'),

('Shahbaz','Khan','shahbaz1@example.com','03101234518'),

('Areeba','Noor','areeba1@example.com','03101234519'),

('Ali','Raza','ali2@example.com','03101234520');

-- ==

-- Registration - 20 entries

-- ==

INSERT INTO Registration (EventID, AttendeeID, RegistrationDate, RegistrationStatus) VALUES

(1,1,CURDATE(),'Confirmed'),

(1,2,CURDATE(),'Confirmed'),

(2,3,CURDATE(),'Confirmed'),

(2,4,CURDATE(),'Confirmed'),

(3,5,CURDATE(),'Confirmed'),

(3,6,CURDATE(),'Confirmed'),

(4,7,CURDATE(),'Confirmed'),

(4,8,CURDATE(),'Confirmed'),

(5,9,CURDATE(),'Confirmed'),

(5,10,CURDATE(),'Confirmed'),

(6,11,CURDATE(),'Confirmed'),

(6,12,CURDATE(),'Confirmed'),

(7,13,CURDATE(),'Confirmed'),

(7,14,CURDATE(),'Confirmed'),

```
(8,15,CURDATE(),'Confirmed'),
(8,16,CURDATE(),'Confirmed'),
(9,17,CURDATE(),'Confirmed'),
(9,18,CURDATE(),'Confirmed'),
(10,19,CURDATE(),'Confirmed'),
(10,20,CURDATE(),'Confirmed');

-- ========================================
-- Payment - 20 entries
-- ========================================
INSERT INTO Payment (RegistrationID, PaymentDate, PaymentAmount, PaymentMethod) VALUES
(1,CURDATE(),50.00,'Card'),
(2,CURDATE(),150.00,'Card'),
(3,CURDATE(),0.00,'Free'),
(4,CURDATE(),0.00,'Free'),
(5,CURDATE(),80.00,'Card'),
(6,CURDATE(),200.00,'Card'),
(7,CURDATE(),20.00,'Cash'),
(8,CURDATE(),100.00,'Card'),
(9,CURDATE(),100.00,'Card'),
(10,CURDATE(),30.00,'Cash'),
(11,CURDATE(),25.00,'Card'),
(12,CURDATE(),90.00,'Card'),
(13,CURDATE(),40.00,'Card'),
(14,CURDATE(),150.00,'Card'),
(15,CURDATE(),0.00,'Free'),
(16,CURDATE(),85.00,'Card'),
(17,CURDATE(),200.00,'Card'),
(18,CURDATE(),15.00,'Cash'),
(19,CURDATE(),100.00,'Card'),
(20,CURDATE(),100.00,'Card');
```

```
-- 6. Sample Queries
-- View all events with tickets
SELECT E.EventName, T.TicketType, T.TicketQuantity
FROM Event E
JOIN Ticket T ON E.EventID = T.EventID;

-- View attendees for an event
SELECT A.FirstName, A.LastName, R.RegistrationStatus
FROM Attendee A
JOIN Registration R ON A.AttendeeID = R.AttendeeID
WHERE R.EventID = 1;

-- View payments
SELECT P.PaymentID, A.FirstName, A.LastName, P.PaymentAmount, P.PaymentMethod
FROM Payment P
JOIN Registration R ON P.RegistrationID = R.RegistrationID
JOIN Attendee A ON R.AttendeeID = A.AttendeeID;

-- Check high attendance events
SELECT * FROM HighAttendanceEvents;

-- Remaining tickets for an event
SELECT EventID, TicketType, TicketQuantity FROM Ticket;

-- ------------------------------
-- EVENT CREATOR CRUD
-- ------------------------------

-- CREATE (Add new event creator)
```

```
INSERT INTO EventCreator (FirstName, LastName, Email, Phone, Password)
VALUES ('Ali', 'Khan', 'ali@example.com', '03001234567', 'pass123');

-- READ (View all event creators)
SELECT * FROM EventCreator;

-- UPDATE (Update event creator phone)
UPDATE EventCreator
SET Phone = '03009998877'
WHERE EventCreatorID = 1;

-- DELETE (Remove event creator)
DELETE FROM EventCreator
WHERE EventCreatorID = 11;

-- ----------------------------
-- EVENT TYPE CRUD
-- ----------------------------

-- CREATE (Add new event type)
INSERT INTO EventType (EventTypeName)
VALUES ('Seminar');

-- READ (View all event types)
SELECT * FROM EventType;

-- UPDATE (Change event type name)
UPDATE EventType
SET EventTypeName = 'Workshop'
WHERE EventTypeID = 1;
```

```
-- DELETE (Remove event type)
DELETE FROM EventType
WHERE EventTypeID = 11;

-- ----------------------------
-- EVENT CRUD
-- ----------------------------

-- CREATE (Add new event)
INSERT INTO Event (EventCreatorID, EventName, EventDescription, EventDate, EventTime, EventLocation, EventType)
VALUES (1, 'Tech Summit 2026', 'Annual tech event', '2026-03-10', '10:00:00', 'Lahore Expo', 1);

-- READ (View all events)
SELECT * FROM Event;

-- READ with JOIN to show creator and type
SELECT E.EventName, EC.FirstName, EC.LastName, ET.EventTypeName
FROM Event E
JOIN EventCreator EC ON E.EventCreatorID = EC.EventCreatorID
JOIN EventType ET ON E.EventType = ET.EventTypeID;

-- UPDATE (Change event location)
UPDATE Event
SET EventLocation = 'Karachi Expo'
WHERE EventID = 1;

-- DELETE (Remove event)
DELETE FROM Event
WHERE EventID = 11;

-- ----------------------------
```

```
-- VENUE CRUD
-- ------------------------------

-- CREATE (Add new venue)
INSERT INTO Venue (VenueName, VenueLocation, VenueCapacity)
VALUES ('Karachi Banquet Hall', 'Karachi', 300);

-- READ (View all venues)
SELECT * FROM Venue;

-- UPDATE (Change venue capacity)
UPDATE Venue
SET VenueCapacity = 500
WHERE VenueID = 1;

-- DELETE (Remove venue)
DELETE FROM Venue
WHERE VenueID = 11;

-- ------------------------------
-- EVENT-VENUE CRUD
-- ------------------------------

-- CREATE (Assign event to venue)
INSERT INTO EventVenue (EventID, VenueID)
VALUES (1, 1);

-- READ (View event venue assignments)
SELECT E.EventName, V.VenueName
FROM EventVenue EV
JOIN Event E ON EV.EventID = E.EventID
```

```
JOIN Venue V ON EV.VenueID = V.VenueID;

-- UPDATE (Change venue for an event)
UPDATE EventVenue
SET VenueID = 2
WHERE EventVenueID = 1;

-- DELETE (Remove event venue assignment)
DELETE FROM EventVenue
WHERE EventVenueID = 11;

-- ----------------------------
-- TICKET CRUD
-- ----------------------------

-- CREATE (Add ticket type for event)
INSERT INTO Ticket (EventID, TicketType, TicketPrice, TicketQuantity)
VALUES (1, 'VIP', 150.00, 50);

-- READ (View all tickets)
SELECT * FROM Ticket;

-- UPDATE (Change ticket quantity)
UPDATE Ticket
SET TicketQuantity = TicketQuantity + 20
WHERE TicketID = 1;

-- DELETE (Remove ticket type)
DELETE FROM Ticket
WHERE TicketID = 11;
```

```
-- ----------------------------
-- ATTENDEE CRUD
-- ----------------------------

-- CREATE (Add attendee)
INSERT INTO Attendee (FirstName, LastName, Email, Phone)
VALUES ('Ayesha', 'Khan', 'ayesha@example.com', '03101234567');

-- READ (View all attendees)
SELECT * FROM Attendee;

-- UPDATE (Update attendee phone)
UPDATE Attendee
SET Phone = '03009998888'
WHERE AttendeeID = 1;

-- DELETE (Remove attendee)
DELETE FROM Attendee
WHERE AttendeeID = 11;

-- ----------------------------
-- REGISTRATION CRUD
-- ----------------------------

-- CREATE (Register attendee for event)
INSERT INTO Registration (EventID, AttendeeID, RegistrationDate, RegistrationStatus)
VALUES (1, 1, CURDATE(), 'Confirmed');

-- READ (View registrations)
SELECT R.RegistrationID, A.FirstName, A.LastName, E.EventName, R.RegistrationStatus
FROM Registration R
```

```
JOIN Attendee A ON R.AttendeeID = A.AttendeeID
JOIN Event E ON R.EventID = E.EventID;

-- UPDATE (Change registration status)
UPDATE Registration
SET RegistrationStatus = 'Cancelled'
WHERE RegistrationID = 1;

-- DELETE (Remove registration)
DELETE FROM Registration
WHERE RegistrationID = 11;

-- ------------------------------
-- PAYMENT CRUD
-- ------------------------------

-- CREATE (Add payment for registration)
INSERT INTO Payment (RegistrationID, PaymentDate, PaymentAmount, PaymentMethod)
VALUES (1, CURDATE(), 150.00, 'Card');

-- READ (View payments)
SELECT P.PaymentID, A.FirstName, A.LastName, E.EventName, P.PaymentAmount, P.PaymentMethod
FROM Payment P
JOIN Registration R ON P.RegistrationID = R.RegistrationID
JOIN Attendee A ON R.AttendeeID = A.AttendeeID
JOIN Event E ON R.EventID = E.EventID;

-- UPDATE (Change payment amount)
UPDATE Payment
SET PaymentAmount = 200.00
WHERE PaymentID = 1;
```

```
-- DELETE (Remove payment)
DELETE FROM Payment
WHERE PaymentID = 11;

-- ------------------------------
-- BUSINESS RULE QUERIES
-- ------------------------------

-- 1. Events with more than 50 attendees
SELECT E.EventName, COUNT(R.RegistrationID) AS TotalAttendees
FROM Event E
JOIN Registration R ON E.EventID = R.EventID
GROUP BY E.EventID
HAVING TotalAttendees > 50;

-- 2. Tickets with low stock (less than 10)
SELECT EventName, TicketType, TicketQuantity
FROM Ticket T
JOIN Event E ON T.EventID = E.EventID
WHERE TicketQuantity < 10;

-- 3. Attendees registered for a specific event
SELECT A.FirstName, A.LastName
FROM Registration R
JOIN Attendee A ON R.AttendeeID = A.AttendeeID
WHERE R.EventID = 1;

-- 4. Total payments received per event
SELECT E.EventName, SUM(P.PaymentAmount) AS TotalPayments
FROM Payment P
```

```
JOIN Registration R ON P.RegistrationID = R.RegistrationID
JOIN Event E ON R.EventID = E.EventID
GROUP BY E.EventName;

-- 5. Events organized by a specific creator
SELECT E.EventName, ET.EventTypeName
FROM Event E
JOIN EventType ET ON E.EventType = ET.EventTypeID
WHERE E.EventCreatorID = 1;
```

Lessons Learned

1. **Importance of Relational Database Design**
    - Proper table design, primary and foreign keys, and relationships (1:M, M:N, 1:1) are essential for **data consistency, integrity, and scalability**.
    - Linking entities like Event → Registration → Attendee ensures accurate tracking of registrations and payments.
2. **Enforcing Business Rules via Constraints and Triggers**
    - Using **foreign keys, unique constraints, and triggers** helped enforce rules automatically, such as:
        - Preventing overbooking of tickets.
        - Maintaining accurate ticket quantities after each registration.
    - This reduces human errors and increases system reliability.
3. **Data Consistency Across Multiple Tables**
    - CRUD operations demonstrated how creating, updating, and deleting records affects related tables.
    - Realizing the need to carefully manage dependencies (e.g., deleting an Event requires attention to registrations, payments, tickets) was a key insight.
4. **Handling Complex Relationships**
    - Implementing **many-to-many relationships** using junction tables (EventVenue, Registration) helped understand complex real-world scenarios like:
        - Assigning multiple venues to a single event.
        - Allowing attendees to register for multiple events.
5. **Scalability and Sample Data**
    - Populating **20 entries for each table** highlighted potential issues with bulk inserts, foreign key dependencies, and query performance.
    - It reinforced the need for efficient indexing and query optimization for larger datasets.
6. **Integration of Stakeholders**
    - Understanding roles (Event Creators, Attendees, Admin, Finance Staff) showed how **system design must account for different access levels and responsibilities**.
    - Admin oversight ensures security, reporting, and controlled access.
7. **Practical Use of SQL Views and Queries**
    - Creating views like HighAttendanceEvents and writing queries for ticket stock, payments, and registrations helped in generating **real-time reports and dashboards**.
    - This reinforced the idea that a database can do more than store data—it can actively support decision-making.
8. **Triggers and Automation**
    - Learning to use triggers helped automate repetitive tasks, like updating ticket quantity after registration, demonstrating **how business logic can be embedded in the database**.

9. **Attention to Detail in Data Entry and Testing**
    - While inserting sample data, maintaining **consistent foreign key references** and **accurate event-date logic** was critical.
    - Errors in IDs or dates could break queries or reports, highlighting the importance of careful testing.
10. **Bridging Theory and Practical Application**
    - This project illustrated how database concepts learned in theory—normalization, entity relationships, constraints, triggers—**directly translate to real-world event management systems**.

Summary:
This project reinforced the importance of careful database design, enforcing business rules, automating processes, and considering multiple stakeholders. It also provided hands-on experience in creating a **robust, scalable, and integrated event management system**, highlighting both the challenges and best practices for real-world database applications.