# Table Creation

1. Creating database

```
mysql> CREATE DATABASE event_portal;
Query OK, 1 row affected (0.04 sec)

mysql> USE event_portal;
Database changed
```

2. Creating Table users

```
mysql> CREATE TABLE Users (
    ->   user_id INT PRIMARY KEY AUTO_INCREMENT,
    ->   full_name VARCHAR(100) NOT NULL,
    ->   email VARCHAR(100) UNIQUE NOT NULL,
    ->   city VARCHAR(100) NOT NULL,
    ->   registration_date DATE NOT NULL
    -> );
Query OK, 0 rows affected (0.16 sec)
```

3. Creating Table Events

```
mysql> CREATE TABLE Events (
    ->   event_id INT PRIMARY KEY AUTO_INCREMENT,
    ->   title VARCHAR(200) NOT NULL,
    ->   description TEXT,
    ->   city VARCHAR(100) NOT NULL,
    ->   start_date DATETIME NOT NULL,
    ->   end_date DATETIME NOT NULL,
    ->   status ENUM('upcoming','completed','cancelled'),
    ->   organizer_id INT,
    ->   FOREIGN KEY (organizer_id) REFERENCES Users(user_id)
    -> );
Query OK, 0 rows affected (0.08 sec)
```

4. Creating Table Sessions

```
mysql> CREATE TABLE Sessions (
    ->   session_id INT PRIMARY KEY AUTO_INCREMENT,
    ->   event_id INT,
    ->   title VARCHAR(200) NOT NULL,
    ->   speaker_name VARCHAR(100) NOT NULL,
    ->   start_time DATETIME NOT NULL,
    ->   end_time DATETIME NOT NULL,
    ->   FOREIGN KEY (event_id) REFERENCES Events(event_id)
    -> );
Query OK, 0 rows affected (0.07 sec)
```

5. Creating Table Registration

```
mysql> CREATE TABLE Registrations (
    ->   registration_id INT PRIMARY KEY AUTO_INCREMENT,
    ->   user_id INT,
    ->   event_id INT,
    ->   registration_date DATE NOT NULL,
    ->   FOREIGN KEY (user_id) REFERENCES Users(user_id),
    ->   FOREIGN KEY (event_id) REFERENCES Events(event_id)
    -> );
Query OK, 0 rows affected (0.11 sec)
```

6. Creating Table Feedback

```
mysql> CREATE TABLE Feedback (
    ->   feedback_id INT PRIMARY KEY AUTO_INCREMENT,
    ->   user_id INT,
    ->   event_id INT,
    ->   rating INT CHECK (rating BETWEEN 1 AND 5),
    ->   comments TEXT,
    ->   feedback_date DATE NOT NULL,
    ->   FOREIGN KEY (user_id) REFERENCES Users(user_id),
    ->   FOREIGN KEY (event_id) REFERENCES Events(event_id)
    -> );
Query OK, 0 rows affected (0.10 sec)
```

7. Creating Table Resource

```
mysql> CREATE TABLE Resources (
    ->   resource_id INT PRIMARY KEY AUTO_INCREMENT,
    ->   event_id INT,
    ->   resource_type ENUM('pdf','image','link'),
    ->   resource_url VARCHAR(255) NOT NULL,
    ->   uploaded_at DATETIME NOT NULL,
    ->   FOREIGN KEY (event_id) REFERENCES Events(event_id)
    -> );
Query OK, 0 rows affected (0.08 sec)
```

# Adding Values to the table (using sample data)

```
mysql> INSERT INTO Users (user_id, full_name, email, city, registration_date
) VALUES
    -> (1, 'Alice Johnson', 'alice@example.com', 'New York', '2024-12-01'),
    -> (2, 'Bob Smith', 'bob@example.com', 'Los Angeles', '2024-12-05'),
    -> (3, 'Charlie Lee', 'charlie@example.com', 'Chicago', '2024-12-10'),
    -> (4, 'Diana King', 'diana@example.com', 'New York', '2025-01-15'),
    -> (5, 'Ethan Hunt', 'ethan@example.com', 'Los Angeles', '2025-02-01');
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO Events (event_id, title, description, city, start_date, e
nd_date, status, organizer_id) VALUES
    -> (1, 'Tech Innovators Meetup', 'A meetup for tech enthusiasts.', 'New
York', '2025-06-10 10:00:00', '2025-06-10 16:00:00', 'upcoming', 1),
    -> (2, 'AI & ML Conference', 'Conference on AI and ML advancements.', 'C
hicago', '2025-05-15 09:00:00', '2025-05-15 17:00:00', 'completed', 3),
    -> (3, 'Frontend Development Bootcamp', 'Hands-on training on frontend t
ech.', 'Los Angeles', '2025-07-01 10:00:00', '2025-07-03 16:00:00', 'upcomin
g', 2);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO Sessions (session_id, event_id, title, speaker_name, star
t_time, end_time) VALUES
    -> (1, 1, 'Opening Keynote', 'Dr. Tech', '2025-06-10 10:00:00', '2025-06
-10 11:00:00'),
    -> (2, 1, 'Future of Web Dev', 'Alice Johnson', '2025-06-10 11:15:00', '
2025-06-10 12:30:00'),
    -> (3, 2, 'AI in Healthcare', 'Charlie Lee', '2025-05-15 09:30:00', '202
5-05-15 11:00:00'),
    -> (4, 3, 'Intro to HTML5', 'Bob Smith', '2025-07-01 10:00:00', '2025-07
-01 12:00:00');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO Registrations (registration_id, user_id, event_id, regist
ration_date) VALUES
    -> (1, 1, 1, '2025-05-01'),
    -> (2, 2, 1, '2025-05-02'),
    -> (3, 3, 2, '2025-04-30'),
    -> (4, 4, 2, '2025-04-28'),
    -> (5, 5, 3, '2025-06-15');
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO Feedback (feedback_id, user_id, event_id, rating, comment
s, feedback_date) VALUES
    -> (1, 3, 2, 4, 'Great insights!', '2025-05-16'),
    -> (2, 4, 2, 5, 'Very informative.', '2025-05-16'),
    -> (3, 2, 1, 3, 'Could be better.', '2025-06-11');
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO Resources (resource_id, event_id, resource_type, resource
_url, uploaded_at) VALUES
    -> (1, 1, 'pdf', 'https://portal.com/resources/tech_meetup_agenda.pdf',
'2025-05-01 10:00:00'),
    -> (2, 2, 'image', 'https://portal.com/resources/ai_poster.jpg', '2025-0
4-20 09:00:00'),
    -> (3, 3, 'link', 'https://portal.com/resources/html5_docs', '2025-06-25
 15:00:00');
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

# Exercises

1. User Upcoming Events
   Show a list of all upcoming events a user is registered for in their city, sorted by date.

```
mysql> SELECT u.full_name, e.title, e.start_date
    -> FROM Users u
    -> JOIN Registrations r ON u.user_id = r.user_id
    -> JOIN Events e ON r.event_id = e.event_id
    -> WHERE e.status = 'upcoming' AND u.city = e.city
    -> ORDER BY e.start_date;
+---------------+-------------------------------+---------------------+
| full_name     | title                         | start_date          |
+---------------+-------------------------------+---------------------+
| Alice Johnson | Tech Innovators Meetup        | 2025-06-10 10:00:00 |
| Ethan Hunt    | Frontend Development Bootcamp | 2025-07-01 10:00:00 |
+---------------+-------------------------------+---------------------+
2 rows in set (0.02 sec)
```

2. Top Rated Events
   Identify events with the highest average rating, considering only those that have received at least 10 feedback submissions.

```
mysql> SELECT e.title, AVG(f.rating) AS avg_rating
    -> FROM Events e
    -> JOIN Feedback f ON e.event_id = f.event_id
    -> GROUP BY e.event_id
    -> HAVING COUNT(f.feedback_id) >= 10
    -> ORDER BY avg_rating DESC;
Empty set (0.01 sec)
```

3. Inactive Users
   Retrieve users who have not registered for any events in the last 90 days.

```
mysql> SELECT * FROM Users
    -> WHERE user_id NOT IN (
    ->   SELECT user_id FROM Registrations
    ->   WHERE registration_date >= CURDATE() - INTERVAL 90 DAY
    -> );
Empty set (0.02 sec)
```

4. Peak Session Hours
   Count how many sessions are scheduled between 10 AM to 12 PM for each event

```
mysql> SELECT e.title, COUNT(*) AS peak_sessions
    -> FROM Events e
    -> JOIN Sessions s ON e.event_id = s.event_id
    -> WHERE TIME(s.start_time) BETWEEN '10:00:00' AND '12:00:00'
    -> GROUP BY e.title;
+-----------------------------+---------------+
| title                       | peak_sessions |
+-----------------------------+---------------+
| Tech Innovators Meetup      |             2 |
| Frontend Development Bootcamp |             1 |
+-----------------------------+---------------+
2 rows in set (0.00 sec)
```

5. Most Active Cities
   List the top 5 cities with the highest number of distinct user registrations.

```
mysql> SELECT u.city, COUNT(DISTINCT r.user_id) AS total_users
    -> FROM Users u
    -> JOIN Registrations r ON u.user_id = r.user_id
    -> GROUP BY u.city
    -> ORDER BY total_users DESC
    -> LIMIT 5;
+-------------+-------------+
| city        | total_users |
+-------------+-------------+
| Los Angeles |           2 |
| New York    |           2 |
| Chicago     |           1 |
+-------------+-------------+
3 rows in set (0.00 sec)
```

6. Event Resource Summary
   Generate a report showing the number of resources (PDFs, images, links) uploaded for each event

```
mysql> SELECT e.title,
    ->   SUM(resource_type = 'pdf') AS pdfs,
    ->   SUM(resource_type = 'image') AS images,
    ->   SUM(resource_type = 'link') AS links
    -> FROM Events e
    -> LEFT JOIN Resources r ON e.event_id = r.event_id
    -> GROUP BY e.title;
+------------------------------+------+--------+-------+
| title                        | pdfs | images | links |
+------------------------------+------+--------+-------+
| Tech Innovators Meetup       |    1 |      0 |     0 |
| AI & ML Conference           |    0 |      1 |     0 |
| Frontend Development Bootcamp |    0 |      0 |     1 |
+------------------------------+------+--------+-------+
3 rows in set (0.00 sec)
```

7. Low Feedback Alerts
   List all users who gave feedback with a rating less than 3, along with their comments and associated event names.

```
mysql> SELECT u.full_name, e.title, f.rating, f.comments
    -> FROM Feedback f
    -> JOIN Users u ON f.user_id = u.user_id
    -> JOIN Events e ON f.event_id = e.event_id
    -> WHERE f.rating < 3;
Empty set (0.00 sec)
```

8. Sessions per Upcoming Event
   Display all upcoming events with the count of sessions scheduled for them

```
mysql> SELECT e.title, COUNT(s.session_id) AS session_count
    -> FROM Events e
    -> LEFT JOIN Sessions s ON e.event_id = s.event_id
    -> WHERE e.status = 'upcoming'
    -> GROUP BY e.title;
+------------------------------+---------------+
| title                        | session_count |
+------------------------------+---------------+
| Tech Innovators Meetup       |             2 |
| Frontend Development Bootcamp |             1 |
+------------------------------+---------------+
2 rows in set (0.00 sec)
```

9. Organizer Event
   Summary For each event organizer, show the number of events created and their current status (upcoming, completed, cancelled).

```
mysql> SELECT u.full_name,
    ->   SUM(e.status = 'upcoming') AS upcoming,
    ->   SUM(e.status = 'completed') AS completed,
    ->   SUM(e.status = 'cancelled') AS cancelled
    -> FROM Users u
    -> JOIN Events e ON u.user_id = e.organizer_id
    -> GROUP BY u.full_name;
+---------------+----------+-----------+-----------+
| full_name     | upcoming | completed | cancelled |
+---------------+----------+-----------+-----------+
| Alice Johnson |        1 |         0 |         0 |
| Charlie Lee   |        0 |         1 |         0 |
| Bob Smith     |        1 |         0 |         0 |
+---------------+----------+-----------+-----------+
3 rows in set (0.00 sec)
```

10. Feedback Gap
    Identify events that had registrations but received no feedback at all.

```
mysql> SELECT e.title
    -> FROM Events e
    -> JOIN Registrations r ON e.event_id = r.event_id
    -> LEFT JOIN Feedback f ON e.event_id = f.event_id
    -> GROUP BY e.event_id
    -> HAVING COUNT(f.feedback_id) = 0;
+------------------------------+
| title                        |
+------------------------------+
| Frontend Development Bootcamp |
+------------------------------+
1 row in set (0.00 sec)
```

11. Daily New User Count
    Find the number of users who registered each day in the last 7 days.

```
mysql> SELECT registration_date, COUNT(*) AS new_users
    -> FROM Users
    -> WHERE registration_date >= CURDATE() - INTERVAL 7 DAY
    -> GROUP BY registration_date;
Empty set (0.00 sec)
```

12. Event with Maximum Sessions
    List the event(s) with the highest number of sessions.

```
mysql> SELECT e.title, COUNT(s.session_id) AS session_count
    -> FROM Events e
    -> JOIN Sessions s ON e.event_id = s.event_id
    -> GROUP BY e.event_id
    -> ORDER BY session_count DESC
    -> LIMIT 1;
+-----------------------+---------------+
| title                 | session_count |
+-----------------------+---------------+
| Tech Innovators Meetup |             2 |
+-----------------------+---------------+
1 row in set (0.01 sec)
```

13. Average Rating per City
    Calculate the average feedback rating of events conducted in each city

```
mysql> SELECT e.city, AVG(f.rating) AS avg_rating
    -> FROM Events e
    -> JOIN Feedback f ON e.event_id = f.event_id
    -> GROUP BY e.city;
+----------+------------+
| city     | avg_rating |
+----------+------------+
| New York |     3.0000 |
| Chicago  |     4.5000 |
+----------+------------+
2 rows in set (0.01 sec)
```

14. Most Registered Events
    List top 3 events based on the total number of user registrations.

```
mysql> SELECT e.title, COUNT(r.user_id) AS registrations
    -> FROM Events e
    -> JOIN Registrations r ON e.event_id = r.event_id
    -> GROUP BY e.event_id
    -> ORDER BY registrations DESC
    -> LIMIT 3;
+----------------------------+---------------+
| title                      | registrations |
+----------------------------+---------------+
| Tech Innovators Meetup     |             2 |
| AI & ML Conference         |             2 |
| Frontend Development Bootcamp |          1 |
+----------------------------+---------------+
3 rows in set (0.00 sec)
```

15. Event Session Time Conflict
    Identify overlapping sessions within the same event (i.e., session start and end times that conflict).

```
mysql> SELECT s1.event_id, s1.session_id AS session1, s2.session_id AS sessi
on2
    -> FROM Sessions s1
    -> JOIN Sessions s2 ON s1.event_id = s2.event_id AND s1.session_id < s2.
session_id
    -> WHERE s1.start_time < s2.end_time AND s1.end_time > s2.start_time;
Empty set (0.01 sec)
```

16. Unregistered Active Users
    Find users who created an account in the last 30 days but haven't registered for any events.

```
mysql> SELECT s1.event_id, s1.session_id AS session1, s2.session_id AS sessi
on2
    -> FROM Sessions s1
    -> JOIN Sessions s2 ON s1.event_id = s2.event_id AND s1.session_id < s2.
session_id
    -> WHERE s1.start_time < s2.end_time AND s1.end_time > s2.start_time;
Empty set (0.01 sec)

mysql> SELECT * FROM Users
    -> WHERE registration_date >= CURDATE() - INTERVAL 30 DAY
    -> AND user_id NOT IN (
    ->   SELECT user_id FROM Registrations
    -> );
Empty set (0.00 sec)
```

17. Multi-Session Speakers
    Identify speakers who are handling more than one session across all events.

```
mysql> SELECT speaker_name, COUNT(*) AS sessions
    -> FROM Sessions
    -> GROUP BY speaker_name
    -> HAVING COUNT(*) > 1;
Empty set (0.00 sec)
```

18. Resource Availability Check
    List all events that do not have any resources uploaded.

```
mysql> SELECT e.title
    -> FROM Events e
    -> LEFT JOIN Resources r ON e.event_id = r.event_id
    -> GROUP BY e.event_id
    -> HAVING COUNT(r.resource_id) = 0;
Empty set (0.00 sec)
```

19. Completed Events with Feedback Summary
    For completed events, show total registrations and average feedback rating.

```
mysql> SELECT e.title, COUNT(DISTINCT r.user_id) AS total_regs, AVG(f.rating
) AS avg_rating
    -> FROM Events e
    -> LEFT JOIN Registrations r ON e.event_id = r.event_id
    -> LEFT JOIN Feedback f ON e.event_id = f.event_id
    -> WHERE e.status = 'completed'
    -> GROUP BY e.event_id;
+-------------------+------------+------------+
| title             | total_regs | avg_rating |
+-------------------+------------+------------+
| AI & ML Conference |          2 |     4.5000 |
+-------------------+------------+------------+
1 row in set (0.00 sec)
```

20. User Engagement Index
    For each user, calculate how many events they attended and how many feedbacks they submitted.

```
mysql> SELECT u.full_name,
    ->   COUNT(DISTINCT r.event_id) AS events_attended,
    ->   COUNT(DISTINCT f.feedback_id) AS feedbacks_given
    -> FROM Users u
    -> LEFT JOIN Registrations r ON u.user_id = r.user_id
    -> LEFT JOIN Feedback f ON u.user_id = f.user_id
    -> GROUP BY u.user_id;
+---------------+-----------------+-----------------+
| full_name     | events_attended | feedbacks_given |
+---------------+-----------------+-----------------+
| Alice Johnson |               1 |               0 |
| Bob Smith     |               1 |               1 |
| Charlie Lee   |               1 |               1 |
| Diana King    |               1 |               1 |
| Ethan Hunt    |               1 |               0 |
+---------------+-----------------+-----------------+
5 rows in set (0.00 sec)
```

21. Top Feedback Providers
    List top 5 users who have submitted the most feedback entries.

```
mysql> SELECT u.full_name, COUNT(f.feedback_id) AS feedbacks
    -> FROM Users u
    -> JOIN Feedback f ON u.user_id = f.user_id
    -> GROUP BY u.user_id
    -> ORDER BY feedbacks DESC
    -> LIMIT 5;
+-------------+-----------+
| full_name   | feedbacks |
+-------------+-----------+
| Bob Smith   |         1 |
| Charlie Lee |         1 |
| Diana King  |         1 |
+-------------+-----------+
3 rows in set (0.00 sec)
```

22. Duplicate Registrations
    Check Detect if a user has been registered more than once for the same event

```
mysql> SELECT user_id, event_id, COUNT(*) AS times_registered
    -> FROM Registrations
    -> GROUP BY user_id, event_id
    -> HAVING COUNT(*) > 1;
Empty set (0.00 sec)
```

23. Registration Trends
    Show a month-wise registration count trend over the past 12 months.

```
mysql> SELECT DATE_FORMAT(registration_date, '%Y-%m') AS month, COUNT(*) AS
total
    -> FROM Registrations
    -> WHERE registration_date >= CURDATE() - INTERVAL 12 MONTH
    -> GROUP BY month
    -> ORDER BY month;
+---------+-------+
| month   | total |
+---------+-------+
| 2025-04 |     2 |
| 2025-05 |     2 |
| 2025-06 |     1 |
+---------+-------+
3 rows in set (0.01 sec)
```

24. Average Session Duration per Event
    Compute the average duration (in minutes) of sessions in each event.

```
mysql> SELECT e.title, AVG(TIMESTAMPDIFF(MINUTE, s.start_time, s.end_time))
AS avg_duration
    -> FROM Events e
    -> JOIN Sessions s ON e.event_id = s.event_id
    -> GROUP BY e.event_id;
+------------------------------+--------------+
| title                        | avg_duration |
+------------------------------+--------------+
| Tech Innovators Meetup       |      67.5000 |
| AI & ML Conference           |      90.0000 |
| Frontend Development Bootcamp |     120.0000 |
+------------------------------+--------------+
3 rows in set (0.00 sec)
```

25. Events Without Sessions
    List all events that currently have no sessions scheduled under them.

```
mysql> SELECT e.title
    -> FROM Events e
    -> LEFT JOIN Sessions s ON e.event_id = s.event_id
    -> WHERE s.session_id IS NULL;
Empty set (0.00 sec)
```