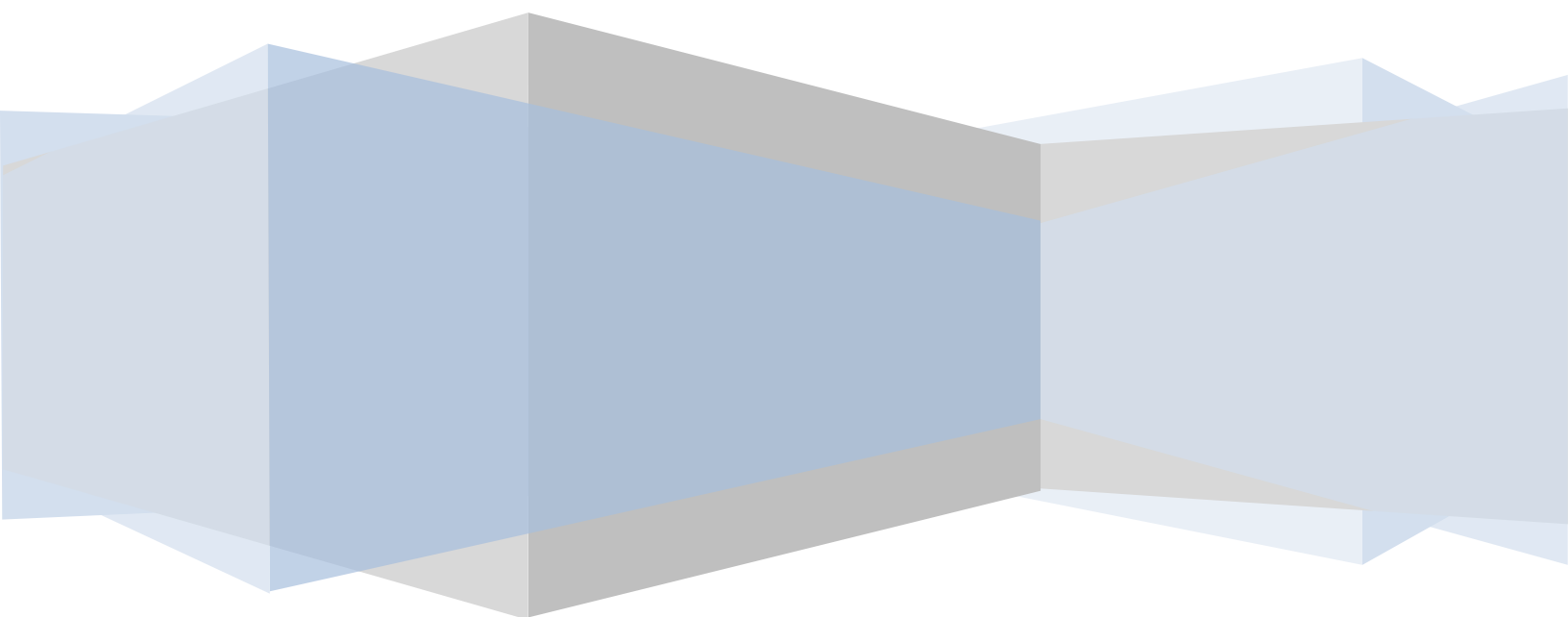


EVENT TICKETING SYSTEM DATABASE DESIGN

Database Management for Event Ticketing: An
SQL Approach

Praveen



Create Database and Tables

Create database

```
1
2  -- Create Database and Select Database --
3
4 • create database event_ticketing_system;
5 • use event_ticketing_system;
6
```

Output:

#	Time	Action	Message
1	19:08:14	use event_ticketing_system	0 row(s) affected

Create customers table

```
7  -- create customers table --
8 • create table customers
9  (
10 CustomerID int primary key auto_increment,
11 Name varchar(30) not null,
12 Age int check(age>=10),
13 Gender enum("M","F"),
14 Phone_no bigint unique not null,
15 EmailID varchar(40),
16 Address varchar(150)
17 );
18 • desc customers;
```

Result Grid

Field	Type	Null	Key	Default	Extra
CustomerID	int	NO	PRI	NULL	auto_increment
Name	varchar(30)	NO		NULL	
Age	int	YES		NULL	
Gender	enum('M','F')	YES		NULL	
Phone_no	bigint	NO	UNI	NULL	
EmailID	varchar(40)	YES		NULL	
Address	varchar(150)	YES		NULL	

Create events table

```
20 -- Create Events Table --
21 • create table events
22 (
23     EventID varchar(30) unique not null,
24     Event_Name varchar(100) unique not null,
25     Category varchar(50),
26     Date varchar(20) not null,
27     Address varchar(100) default"Chennai",
28     Description varchar(500) default"Welcome to the Event"
29 );
30 • desc events;
31
```

Field	Type	Null	Key	Default	Extra
EventID	varchar(30)	NO	PRI	NULL	
Event_Name	varchar(100)	NO	UNI	NULL	
Category	varchar(50)	YES		NULL	
Date	varchar(20)	NO		NULL	
Address	varchar(100)	YES		Chennai	
Description	varchar(500)	YES		Welcome to the Event	

Create reservations table

```
32 -- Create Reservations Table --
33 • create table Reservations
34 (
35     ReserveID int primary key auto_increment,
36     CustomerID int,
37     EventID varchar(30),
38     ReserveDATE date default(curdate()),
39     PaymentType varchar(20) default"UPI",
40     Payment_Status varchar(20) check(payment_status in ("Cancelled","Pending","Completed")),
41     constraint fr_cusn foreign key(customerID) REFERENCES customers(customerID),
42     constraint fr_evid foreign key(eventID) REFERENCES events(eventID)
43 );
44 • desc Reservations;
```

Field	Type	Null	Key	Default	Extra
ReserveID	int	NO	PRI	NULL	auto_increment
CustomerID	int	YES	MUL	NULL	
EventID	varchar(30)	YES	MUL	NULL	
ReserveDATE	date	YES		curdate()	DEFAULT_GENERATED
PaymentType	varchar(20)	YES		UPI	
Payment_Status	varchar(20)	YES		NULL	

Create tickets table

```
45
46  -- Create Tickets Table --
47 • create table Tickets
48  (
49    TicketNO int primary key auto_increment,
50    ReserveID int,
51    Category varchar(10) check(Category in ("VIP","Ordinary","FristClass")),
52    SeatNO varchar(10) not null,
53    constraint fk_rsID foreign key(ReserveID) REFERENCES Reservations(ReserveID)
54  );
55 • desc Tickets;
56
57
```

Field	Type	Null	Key	Default	Extra
TicketNO	int	NO	PRI	NULL	auto_increment
ReserveID	int	YES	MUL	NULL	
Category	varchar(10)	YES		NULL	
SeatNO	varchar(10)	NO		NULL	

Create sales table

```
57
58  -- Create Sales Table --
59 • create table Sales
60  (
61    SalesID int primary key auto_increment,
62    TicketNO int,
63    Price decimal(8,2),
64    SaleDate date default(curdate()),
65    Sales_status enum("Cancelled","Sold") default "Sold",
66    constraint fr_tkno foreign key(ticketNO) REFERENCES tickets(ticketNO)
67  );
68 • desc sales;
69
```

Field	Type	Null	Key	Default	Extra
SalesID	int	NO	PRI	NULL	auto_increment
TicketNO	int	YES	MUL	NULL	
Price	decimal(8,2)	YES		NULL	
SaleDate	date	YES		curdate()	DEFAULT_GENERATED
Sales_status	enum("Cancelled","Sold")	YES		Sold	

Insert values

Insert values into events table

```
71 -- single insert statement --
72 • insert into events(eventid, event_name,category, date, Address, Description) values
73 ('01B25',"India Business Summit 2025","Business","20-June-2025","The Leela Palace, New Delhi","Join industry leaders and experts for
74
75 -- multiple insert statement --
76 • insert into events(eventid, event_name,category, date, Address,Description) values
77 ('02B25',"Startup India Expo 2025","Business","28-MAY-2025","Pragati Maidan, New Delhi","Improve your Business skills"),
78 ('03B25',"Corporate Leadership Summit 2025","Business","02-AUG-2025","The Taj Mahal Palace, Mumbai","Improve your leadership skill
79 ('04E26',"EduTech Conference 2025","Education","06-jan-2026","Anna conference hall, Adayar,Chennai","Explore the latest trends and
80 ('05E25',"Career Expo 2025","Education","04-MAY-2025","Exhibition Centre, Chidambaram","The Career Expo 2025 will feature job fair
81 ('06E26',"International Conference on Education","Education","04-MAY-2025","Jawaharlal Nehru University,Mumbai","Share knowledge i
82 ('07EN25',"Comedy Show","Entertainment","20-NOV-2025","The Comedy Club, Mumbai","Laugh out loud with India's top comedians at this
83 ('08EN25',"Dance Competition","Entertainment","22-OCT-2025","Siri Fort Auditorium, New Delhi","- Witness the best dancers in the co
84 ('09F25',"Designer Showcase 2025","Fashion","15-OCT-2025","The Oberoi, Mumbai","Get a glimpse of the latest collections from India
85 ('10F25',"Bridal Fashion Show","Fashion","26-DEC-2025","The Leela Palace, Jaipur","Witness the latest bridal fashion trends and de
86 ('11S25'," India Open Tennis Tournament","Sports","10-JUNE-2025","R.K. Khanna Tennis Stadium, New Delhi","The India Open Tennis Tou
87 ('12E25',"AGRI TEC EXPO-2025","Education","29-MAY-2025","Anandha Thirumana Mahal,Viluppuram","AGRI TEC EXPO - 2025 the objective o
88 ('13EN25',"Drink meetup Chennai","Entertainment","10-MAY-2025","Lower Deck Bar & Nightclub, Chennai","Let's meet up in Chennai for
89 ('14E25',"Automate Intelligently","Education","09-MAY-2025","Technology Services Private Limited, Chennai","Learn how to harness i
90
```

Insert values into customers table

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'event_ticketing_system' schema with tables like 'customer_details', 'CUSTOMERS_BOC', 'Specific_Event_Ti', and 'TICKETS_SOLD_C'. The main editor window contains the following SQL queries:

```
100 -- Inserting values into the Customers table --
101
102 • insert into customers (CustomerID, Name, Age, Gender, Phone_no, EmailId, Address) values
103 ('01',"Pradesh",20,"M",9824213047,"pradeshg12@gmail.com","Haraimalainagar");
104
105 # Assigning Auto_increment values --
106 • insert into customers (Name, Age, Gender, Phone_no, EmailId, Address) values
107 ("Pradhap",21,"M",9852364145,"pradhapi2@gmail.com","Tambaram"),("Pradeep",24,"M",9852647811,"pradeep14@gmail.com","Velachery"),
108 ("Vikky",15,"M",9865471236,"Vikky23@gmail.com","Chidambaram"),("Sharukan",25,"M",9645172831,"Allish@gmail.com","T.nagar"),
109 ("Praveen",21,"M",9894786424,"praveenp9894@gmail.com","Mylapore"),("Samuvel",22,"M",6748216425,"samuvels6789@gmail.com","Chidambaram"),
110 ("Srikanth",21,"M",9645781202,"srikanthar@gmail.com","Chidambaram"),("Mugesh",20,"M",9175423816,"Mugen20@gmail.com","Chidambaram"),
111 ("Gayathri",20,"F",9623174752,"gayu5689@gmail.com","KM koil"),("Priya",21,"F",9847632015,"priyavenkat@gmail.com","Porur"),
112 ("Sathana",14,"F",9841257436,"sathu3920@gmail.com","Velagiri"),("Murali",23,"M",9325648246,"murali@gmail.com","Adayar"),
113 ("Venkat",24,"M",9642105736,"venkat6789@gmail.com","Mandaveli"),("Priyanka",25,"F",9235164872,"priyal2@gmail.com","Velachery"),
114 ("Magesh",23,"F",9512046311,"magesh59@gmail.com","Kattankulathur"),("Dharshan",12,"M",9325867412,"rds245@gmail.com","Jayankondam"),
115 ("Marivel",26,"M",7356842130,"mari484@gmail.com","Valluvar kottam"),("Mani",20,"M",9142637841,"knani@gmail.com","Bhuvanigiri"),
116 ("Suthi",22,"F",9846723108,"suthi1@outlook.com","Nagarkoil"),("Maanikkam",25,"M",9864564725,"maanikam@gmail.com","Viruthachalam"),
117 ("Rohan Sharma",25,"M",9876543210,"rohan@gmail.com","New Delhi"),("Priya Singh",30,"F",8765432109,"priya@yahoo.com","Mumbai"),
118 ("Amit Kumar",22,"M",7654321098,"amit@outlook.com","Bangalore"),("Sneha Jain",28,"F",6543210987,"sneha@gmail.com","Hyderabad"),
119 ("Rajesh Patel",35,"M",5432109876,"rajesh@yahoo.com","Chennai"),("Kavita Jain",29,"F",4321098765,"kavita@outlook.com","Pune"),
120 ("Vikas Kumar",32,"M",3210987654,"vikas@gmail.com","Ahmedabad"),("Ritu Singh",26,"F",2109876543,"ritu@yahoo.com","Surat"),
121 ("Sachin Gupta",38,"M",1098765432,"sachin@outlook.com","Jaipur"),("Anjali Sharma",24,"M",9876543211,"anjali@gmail.com","Lucknow"),
```

The bottom status bar shows '4 row(s) returned' and a duration of '0.016 sec / 0.000 sec'.

Insert values into reservations table

The screenshot shows the MySQL Workbench interface with a query editor. The left sidebar displays the 'event_ticketing_system' schema with tables like 'customer_details', 'CUSTOMERS_BOOKING', 'Specific_Event_Tickets', and 'TICKETS_SOLD_OUT'. The main editor contains two SQL queries:

```
-- Insert values into Reservations table

INSERT INTO Reservations (ReserveID, customerID, EventID, ReserveDATE, PaymentType, Payment_Status) values
(5001,2, "01025", "2025-05-01", "UPI", "Completed");

INSERT INTO Reservations (customerID, EventID, ReserveDATE, PaymentType, Payment_Status) values
(1, "09F25", "2025-04-10", "Card", "Completed"),(2, "02B25", "2025-05-03", "UPI", "Pending"),
(2, "07EN25", "2025-03-08", "UPI", "Pending"),(3, "03B25", "2025-05-02", "Card", "Completed"),
(3, "08EN25", "2025-02-07", "Card", "Completed"),(3, "11S25", "2025-05-12", "UPI", "Completed"),
(4, "08EN25", "2025-05-07", "Card", "Pending"),(5, "05E25", "2025-05-06", "Card", "Completed"),
(5, "10F25", "2025-05-11", "Card", "Completed"),(5, "12E25", "2025-05-15", "UPI", "Completed"),
(6, "13EN25", "2025-03-09", "UPI", "Pending"),(7, "14E25", "2025-05-14", "Card", "Completed"),
(8, "01B25", "2025-04-01", "UPI", "Completed"),(10, "09F25", "2025-05-10", "Card", "Completed"),
(11, "02B25", "2025-05-03", "UPI", "Pending"),(12, "07EN25", "2025-05-08", "UPI", "Completed"),
(14, "08EN25", "2025-05-07", "Card", "Pending"),(15, "11S25", "2025-05-12", "UPI", "Completed"),
(16, "04E26", "2025-05-04", "UPI", "Cancelled"),(17, "05E25", "2025-05-06", "Card", "Completed"),
(18, "10F25", "2025-05-11", "Card", "Pending"),(19, "12E25", "2025-05-15", "UPI", "Completed"),
(20, "13EN25", "2025-03-09", "UPI", "Pending"),(21, "14E25", "2025-05-14", "Card", "Completed"),
(22, "01B25", "2025-04-01", "UPI", "Completed"),(13, "03B25", "2025-05-02", "Card", "Completed"),
(22, "09F25", "2025-02-10", "Card", "Completed"),(23, "02B25", "2025-05-03", "UPI", "Pending"),
(24, "07EN25", "2025-02-08", "UPI", "Pending"),(25, "03B25", "2025-05-02", "Card", "Completed"),
(25, "08EN25", "2025-04-07", "Card", "Completed"),(25, "11S25", "2025-05-12", "UPI", "Completed");
```

The output pane shows 'Action Output' with 4 rows returned, indicating successful execution of the queries.

Insert values into tickets table

The screenshot shows the MySQL Workbench interface with a query editor. The left sidebar displays the 'event_ticketing_system' schema. The main editor contains two SQL queries:

```
-- Insert values into Tickets table.

insert into Tickets(TicketNO,ReserveID,Category,SeatNO) values(2025164,5001,"VIP","V51");

insert into Tickets(ReserveID,Category,SeatNO) values
(5002,"Ordinary","OR1"),(5005,"Ordinary","OR12"),
(5006,"Ordinary","OR13"),(5007,"Ordinary","OR17"),(5009,"VIP","VIP3"),(5010,"Ordinary","OR5"),
(5013,"Ordinary","OR5"),(5014,"Ordinary","OR24"),(5015,"Ordinary","OR18"),
(5017,"VIP","VIP2"),(5019,"Ordinary","OR14"),(5021,"FristClass","F10"),(5023,"FristClass","F13"),
(5025,"Ordinary","OR16"),(5026,"FristClass","F14"),(5027,"FristClass","F5"),(5028,"VIP","VIP5"),
(5031,"FristClass","F4"),(5032,"Ordinary","OR19"),(5033,"FristClass","F3"),(5035,"FristClass","F1"),
(5036,"VIP","VIP6"),(5037,"Ordinary","OR21"),(5039,"FristClass","F2"),(5040,"Ordinary","OR8"),
(5041,"Ordinary","OR7"),(5042,"FristClass","F1"),(5045,"VIP","VIP4"),(5046,"FristClass","F15"),
(5047,"Ordinary","OR4"),(5049,"Ordinary","OR1"),(5050,"FristClass","F1"),(5051,"VIP","VIP7"),
(5053,"Ordinary","OR6"),(5054,"Ordinary","OR3"),(5057,"Ordinary","OR1"),(5058,"Ordinary","OR9"),
(5059,"Ordinary","OR11"),(5060,"Ordinary","OR12"),(5061,"VIP","VIP4"),(5062,"FristClass","F1"),
(5063,"Ordinary","OR1"),(5064,"Ordinary","OR1"),(5065,"Ordinary","OR16"),(5066,"FristClass","F1"),
(5067,"VIP","VIP2"),(5068,"FristClass","F7"),(5073,"Ordinary","OR28"),(5069,"Ordinary","OR31"),
(5078,"Ordinary","OR27"),(5071,"FristClass","F8"),(5072,"VIP","VIP6"),(5011,"Ordinary","OR25");
```

The output pane shows 'Action Output' with 4 rows returned, indicating successful execution of the queries.

Insert values into sales table

The screenshot displays the MySQL Workbench interface with a local instance of MySQL 8.0. The left sidebar shows the 'event_ticketing_system' schema with tables like 'customer_details', 'CUSTOMERS_BOC', 'Specific_Event_Ti', and 'TICKETS_SOLD_C'. The main editor shows a series of SQL queries for inserting data into the 'sales' table.

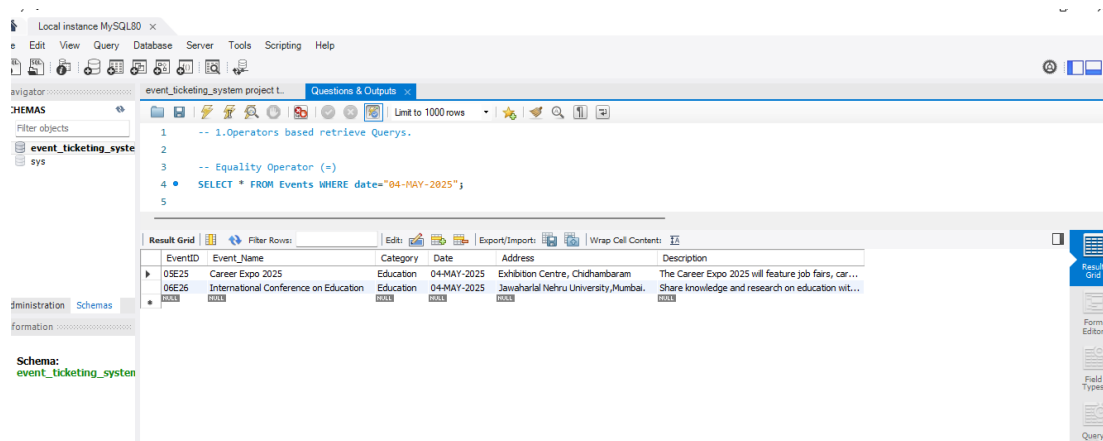
```
226 -- Insert values into Sales table
227
228 • insert into sales (SalesID,ticketno,price, saledate,sales_status) values
229 (25001,2025164,1500.00,"2025-05-01","Sold");
230
231 • insert into sales (ticketno, price, saledate) values
232 (2025165,500.00,"2025-04-28"),(2025166,500.00,"2025-04-25"),(2025167,500.00,"2025-05-18"),(2025168,500.00,"2025-05-15"),
233 (2025169,1500.00,"2025-03-26"),(2025170,500.00,"2025-03-14"),(2025171,500.00,"2025-02-01"),(2025172,500.00,"2025-03-05"),
234 (2025173,500.00,"2025-04-08"),(2025174,1500.00,"2025-03-25"),(2025175,500.00,"2025-03-15"),(2025176,1000.00,"2025-03-15"),
235 (2025177,1000.00,"2025-04-08"),(2025178,500.00,"2025-03-15"),(2025179,1000.00,"2025-04-08"),(2025180,1000.00,"2025-04-08"),
236 (2025193,1000.00,"2025-02-18"),(2025194,500.00,"2025-03-15"),(2025195,500.00,"2025-03-24"),(2025196,1000.00,"2025-02-13"),
237 (2025197,1500.00,"2025-04-01"),(2025198,500.00,"2025-02-14"),(2025199,500.00,"2025-03-13"),(2025200,500.00,"2025-03-15"),
238 (2025201,500.00,"2025-02-04"),(2025202,500.00,"2025-03-06"),(2025203,500.00,"2025-03-08"),(2025204,1500.00,"2025-04-01"),
239 (2025205,1000.00,"2025-02-03"),(2025206,500.00,"2025-03-04"),(2025207,500.00,"2025-01-19"),(2025208,500.00,"2025-03-12"),
240 (2025209,1000.00,"2025-02-18"),(2025210,1500.00,"2025-04-01");
241
242 -- Current Date default --
243 • insert into sales ( ticketno, price, sales_status) values
244 (2025211,1000.00,"Cancelled"),(2025212,500.00,"Sold"),(2025213,1500.00,"Sold"),(2025214,500.00,"Cancelled"),
245 (2025215,1000.00,"Cancelled"),(2025216,1500.00,"Sold"),(2025217,500.00,"Cancelled"),(2025181,1500.00,"Cancelled"),
246 (2025182,1000.00,"Sold"),(2025183,500.00,"Sold"),(2025184,1000.00,"Cancelled"),(2025185,1000.00,"Cancelled"),
247 (2025186,1500.00,"Cancelled"),(2025187,500.00,"Sold"),(2025188,1000.00,"Cancelled"),(2025189,500.00,"Sold"),
```

The bottom status bar shows the output of the queries, indicating that 4 rows were returned and the duration was 0.016 sec / 0.000 sec.

Querying THE DATABASE

Operators-Based Retrieve Queries:

Equality Operator (=)



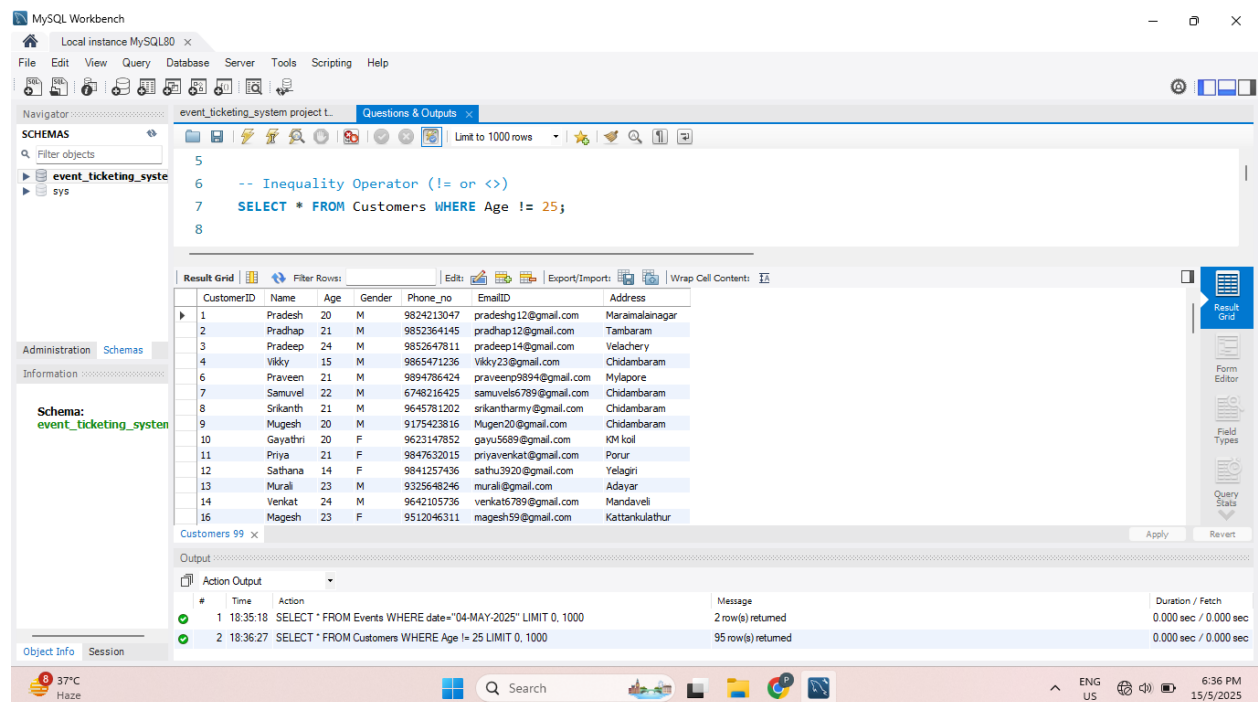
The screenshot shows the MySQL Workbench interface. The 'Questions & Outputs' tab is active, displaying a query that uses the Equality Operator (=) to filter events by date. The query is as follows:

```
-- 1.Operators based retrieve Querys.
-- Equality Operator (=)
SELECT * FROM Events WHERE date="04-MAY-2025";
```

The 'Result Grid' shows the results of the query, which are two rows of event data:

EventID	Event_Name	Category	Date	Address	Description
05E25	Career Expo 2025	Education	04-MAY-2025	Exhibition Centre, Chidambaram	The Career Expo 2025 will feature job fairs, car...
06E26	International Conference on Education	Education	04-MAY-2025	Jawaharlal Nehru University, Mumbai.	Share knowledge and research on education wit...

Inequality Operator (!= or <>)



The screenshot shows the MySQL Workbench interface. The 'Questions & Outputs' tab is active, displaying a query that uses the Inequality Operator (!= or <>) to filter customers by age. The query is as follows:

```
-- Inequality Operator (!= or <>)
SELECT * FROM Customers WHERE Age != 25;
```

The 'Result Grid' shows the results of the query, which are 16 rows of customer data:

CustomerID	Name	Age	Gender	Phone_no	EmailID	Address
1	Pradesh	20	M	9824213047	pradeshg12@gmail.com	Maramalainagar
2	Pradhap	21	M	9852364145	pradhap12@gmail.com	Tambaram
3	Pradeep	24	M	9852647811	pradeep14@gmail.com	Velachery
4	Vikky	15	M	9865471236	Vikky23@gmail.com	Chidambaram
6	Praveen	21	M	9894786424	praveenp9894@gmail.com	Mylapore
7	Samuvel	22	M	6748216425	samuvels6789@gmail.com	Chidambaram
8	Srikanth	21	M	9645781202	srikantharmy@gmail.com	Chidambaram
9	Mugesh	20	M	9175423816	Mugen20@gmail.com	Chidambaram
10	Gayathri	20	F	9623147852	gayu5689@gmail.com	KM koi
11	Priya	21	F	9847632015	priyavenkat@gmail.com	Porur
12	Sethana	14	F	9841257436	sathu3920@gmail.com	Yelagiri
13	Murali	23	M	9325648246	murali@gmail.com	Adayer
14	Venkat	24	M	9642105736	venkat5789@gmail.com	Mandaveli
16	Magesh	23	F	9512046311	magesh59@gmail.com	Kattenkulethur

The 'Output' tab shows the execution details of the query:

#	Time	Action	Message	Duration / Fetch
1	18:35:18	SELECT * FROM Events WHERE date="04-MAY-2025" LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
2	18:36:27	SELECT * FROM Customers WHERE Age != 25 LIMIT 0, 1000	95 row(s) returned	0.000 sec / 0.000 sec

Greater Than Operator (>)

8

9 -- Greater Than Operator (>)

10 • **SELECT * FROM Customers WHERE Age > 35;**

11

CustomerID	Name	Age	Gender	Phone_no	EmailID	Address
30	Sachin Gupta	38	M	1098765432	sachin@outlook.com	Jaipur
36	Vijay Kumar	36	M	4321098766	vijay@yahoo.com	Kancheepuram, Tamil Nadu
40	Raj Kumar	37	M	9876543212	rajkumar@gmail.com	Perambalur, Tamil Nadu
42	Ramesh Kumar	39	M	7654321097	ramesh@gmail.com	Tiruvannamalai, Tamil Nadu
44	Sanjay Patel	40	M	5432109875	sanjay@outlook.com	Krishnagiri, Tamil Nadu
61	Amit Kumar	36	M	9445234569	amit@outlook.com	Kancheepuram, Tamil Nadu
67	Sachin Kumar	38	M	9445234570	sachin@outlook.com	Tiruvannamalai, Tamil Nadu
72	Rohan Singh	37	M	9445234571	rohan@yahoo.com	Chengalpattu, Tamil Nadu
79	Rohan Gupta	36	M	9745234574	rohanag@gmail.com	Thanjavur, Tamil Nadu
101	Vivek Verma	36	M	9345234577	vivekv@outlook.com	Chengalpattu, Tamil Nadu
*	NULL	NULL	NULL	NULL	NULL	NULL

Less than Operator (<)

11

12 -- Less Than Operator (<)

13 • **SELECT * FROM Customers WHERE Age < 20;**

14

CustomerID	Name	Age	Gender	Phone_no	EmailID	Address
4	Vikky	15	M	9865471236	Vikky23@gmail.com	Chidambaram
12	Sathana	14	F	9841257436	sathu3920@gmail.com	Yelagiri
17	Dharshan	12	M	9325867412	rds245@gmail.com	Jayankondam
*	NULL	NULL	NULL	NULL	NULL	NULL

Greater Than or Equal To Operator (>=)

14

15 -- Greater Than or Equal To Operator (>=)

16 • **SELECT** * **FROM** sales **WHERE** price >= 1000;

17

Result Grid

SalesID	TicketNO	Price	SaleDate	Sales_status
25001	2025164	1500.00	2025-05-01	Sold
25006	2025169	1500.00	2025-03-26	Sold
25011	2025174	1500.00	2025-03-25	Sold
25013	2025176	1000.00	2025-03-15	Sold
25014	2025177	1000.00	2025-04-08	Sold
25016	2025179	1000.00	2025-04-08	Sold
25017	2025180	1000.00	2025-04-08	Sold
25018	2025193	1000.00	2025-02-18	Sold
25021	2025196	1000.00	2025-02-13	Sold
25022	2025197	1500.00	2025-04-01	Sold
25029	2025204	1500.00	2025-04-01	Sold
25030	2025205	1000.00	2025-02-03	Sold
25034	2025209	1000.00	2025-02-18	Sold
25035	2025210	1500.00	2025-04-01	Sold

Less Than or Equal To Operator (<=)

17

18 -- Less Than or Equal To Operator (<=)

19 **SELECT** * **FROM** sales **WHERE** price <= 600;

20

Result Grid

SalesID	TicketNO	Price	SaleDate	Sales_status
25002	2025165	500.00	2025-04-20	Sold
25003	2025166	500.00	2025-04-25	Sold
25004	2025167	500.00	2025-05-18	Sold
25005	2025168	500.00	2025-05-15	Sold
25007	2025170	500.00	2025-03-14	Sold
25008	2025171	500.00	2025-02-01	Sold
25009	2025172	500.00	2025-03-05	Sold
25010	2025173	500.00	2025-04-08	Sold
25012	2025175	500.00	2025-03-15	Sold
25015	2025178	500.00	2025-03-15	Sold
25019	2025194	500.00	2025-03-15	Sold
25020	2025195	500.00	2025-03-24	Sold
25023	2025198	500.00	2025-02-14	Sold
25024	2025199	500.00	2025-03-13	Sold

“BETWEEN” Operator

```
20
21 -- BETWEEN Operator
22 • select * from reservations where reservedate between "2025-04-01" and "2025-04-30";
23
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	ReserveID	CustomerID	EventID	ReserveDATE	PaymentType	Payment_Status
▶	5002	1	09F25	2025-04-10	Card	Completed
	5008	4	04E26	2025-04-04	UPI	Cancelled
	5014	8	01B25	2025-04-01	UPI	Completed
	5026	22	01B25	2025-04-01	UPI	Completed
	5032	25	08EN25	2025-04-07	Card	Completed
	5038	28	13EN25	2025-04-09	UPI	Pending
	5044	32	07EN25	2025-04-08	UPI	Pending
	5054	38	09F25	2025-04-10	Card	Completed
✱	NULL	NULL	NULL	NULL	NULL	NULL

“IN” Operator

```
23
24 -- IN Operator
25 • SELECT * FROM Customers WHERE customerid IN (20, 25, 30);
26
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

1A

	CustomerID	Name	Age	Gender	Phone_no	EmailID	Address
▶	20	Suthi	22	F	9846723108	suthi1@outlook.com	Nagarkoil
	25	Sneha Jain	28	F	6543210987	sneha@gmail.com	Hyderabad
	30	Sachin Gupta	38	M	1098765432	sachin@outlook.com	Jaipur
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL

“LIKE” Operator

```
26
27 # LIKE Operator
28
29 • SELECT * FROM customers WHERE name LIKE 'pra%';
30
31
```

Result Grid									Filter Rows:	<input type="text"/>	Edit:				Export/Import:			Wrap Cell Content:	
CustomerID	Name	Age	Gender	Phone_no	EmailID	Address													
1	Pradesh	20	M	9824213047	pradeshg12@gmail.com	Maraimalainagar													
2	Pradhap	21	M	9852364145	pradhap12@gmail.com	Tambaram													
3	Pradeep	24	M	9852647811	pradeep14@gmail.com	Velachery													
6	Praveen	21	M	9894786424	praveenp9894@gmail.com	Mylapore													
NULL	NULL	NULL	NULL	NULL	NULL	NULL													

“NOT” Operator

```
29
30  -- NOT Operator
31  SELECT * FROM reservations WHERE NOT Payment_Status = "completed";
32
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

ReserveID	CustomerID	EventID	ReserveDATE	PaymentType	Payment_Status
5003	2	02B25	2025-05-03	UPI	Pending
5004	2	07EN25	2025-03-08	UPI	Pending
5008	4	04E26	2025-04-04	UPI	Cancelled
5012	6	13EN25	2025-03-09	UPI	Pending
5016	11	02B25	2025-05-03	UPI	Pending
5018	14	08EN25	2025-05-07	Card	Pending
5020	16	04E26	2025-05-04	UPI	Cancelled
5022	18	10F25	2025-05-11	Card	Pending
5024	20	13EN25	2025-03-09	UPI	Pending
5029	23	02B25	2025-05-03	UPI	Pending
5030	24	07EN25	2025-02-08	UPI	Pending
5034	26	04E26	2025-03-04	UPI	Cancelled
5038	28	13EN25	2025-04-09	UPI	Pending
5043	31	02B25	2025-05-03	UPI	Pending

“AND” Operator

```
32
33  -- AND Operator
34  • SELECT * FROM Customers WHERE Age > 38 AND Gender = 'M';
35
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	CustomerID	Name	Age	Gender	Phone_no	EmailID	Address
▶	42	Ramesh Kumar	39	M	7654321097	ramesh@gmail.com	Tiruvannamalai, Tamil Nadu
	44	Sanjay Patel	40	M	5432109875	sanjay@outlook.com	Krishnagiri, Tamil Nadu
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

“OR” Operator

```
35
36  # OR Operator
37  • SELECT * FROM Customers WHERE Age <=18 OR Gender = 'F';
38
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	CustomerID	Name	Age	Gender	Phone_no	EmailID	Address
▶	4	Vikky	15	M	9865471236	Vikky23@gmail.com	Chidambaram
	10	Gayathri	20	F	9623147852	gayu5689@gmail.com	KM koil
	11	Priya	21	F	9847632015	priyavenkat@gmail.com	Porur
	12	Sathana	14	F	9841257436	sathu3920@gmail.com	Yelagiri
	15	Priyanka	25	F	9235164872	priya12@gmail.com	Velachery
	16	Magesh	23	F	9512046311	magesh59@gmail.com	Kattankulathur
	17	Dharshan	12	M	9325867412	rds245@gmail.com	Jayankondam
	20	Suthi	22	F	9846723108	suthi1@outlook.com	Nagarkoil
	23	Priya Singh	30	F	8765432109	priya@yahoo.com	Mumbai
	25	Snaha Jain	28	F	65432110987	snaha@gmail.com	Hudershad

Customers 1 ×

Subquery-Based Queries:

1. Get the details of all events that have no tickets sold.

```
39 -- SUB QUERY--
40 #1.Get the details of all events that have no tickets sold.
41
42 • select * from events where eventId not in (select e.eventID from events e
43 left join reservations r on e.eventid=r.eventId
44 where r.payment_status="completed"
45 group by e.eventid);
46
```

Result Grid						
Filter Rows: <input type="text"/> Edit: Export/Import: Wrap Cell Content:						
EventID	Event_Name	Category	Date	Address	Description	
02B25	Startup India Expo 2025	Business	28-MAY-2025	Pragati Maidan, New Delhi	Improve your Business skills	
04E26	EduTech Conference 2025	Education	06-jan-2026	Anna conference hall, Adayar,Chennai.	Explore the latest trends and technologies in ed...	
13EN25	Drink meetup Chennai	Entertainment	10-MAY-2025	Lower Deck Bar & Nightclub, Chennai.	Let's meet up in Chennai for some drinks and go...	
* NULL	NULL	NULL	NULL	NULL	NULL	

2. Customers who booked tickets for more than one event.

```
49
50 # 2.Customers who Booked Tickets for More than One Event.
51
52 WITH customerid as
53 • (select customerid, count(eventId) as Count_of_events from reservations
54 group by CustomerID having Count_of_events >1)
55
56 select * from customers where customerid in (select customerid from customerid);
57
```

Result Grid						
Filter Rows: <input type="text"/> Export: Wrap Cell Content:						
CustomerID	Name	Age	Gender	Phone_no	EmailID	Address
1	Pradesh	20	M	9824213047	pradeshg12@gmail.com	Maraimalainagar
2	Pradhap	21	M	9852364145	pradhap12@gmail.com	Tambaram
3	Pradeep	24	M	9852647811	pradeep14@gmail.com	Velachery
5	Sharukan	25	M	9645172831	Allish@gmail.com	T.nagar
22	Rohan Sharma	25	M	9876543210	rohan@gmail.com	New Delhi
25	Sneha Jain	28	F	6543210987	sneha@gmail.com	Hyderabad
27	Kavita Jain	29	F	4321098765	kavita@outlook.com	Pune
30	Sachin Gupta	38	M	1098765432	sachin@outlook.com	Jaipur
33	Pooia Singh	27	F	7654321099	nnnia@vahoo.com	Cuddalore . Tamil Nadu

Result 17 x

Aggregate Function Queries:

1. Find the total number of events in each category.

```
54 -- AGGREGATE FUNCTIONS --
55
56 -- 1.Find the total number of events in each category --
57 • select Category,count(event_name) from events group by Category;
58
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Category	count(event_name)			
▶	Business	3			
	Education	5			
	Entertainment	3			
	Fashion	2			
	Sports	5			

2. Find the number of male and female customers.

```
58
59 -- 2.Find the number of male and female customers.(aggrecate functions)
60 • select gender,count(gender) No_of_customers from customers
61 group by gender;
62
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	gender	No_of_customers			
▶	M	58			
	F	47			

3. What are the top 3 days with the highest revenue for ticket sales?

```
62
63 # 3.What are the top 3 days with the highest revenue for the ticketsales?..
64
65 • select saledate, sum(price) as total_sales_per_day from sales
66 group by SaleDate order by total_sales_per_day desc limit 3;
67
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	saledate	total_sales_per_day				
▶	2025-05-13	18000.00				
	2025-04-01	4500.00				
	2025-04-08	3500.00				

Join-Based Queries:

1. Get the details of all customers who have made a reservation.

```
71
72 # 1.Get the details of all customers who have made a reservation.(joins)
73
74 • select r.ReserveID,c.customerId, c.name, c.age, c.gender, c.phone_no, c.emailId, c.address from
75 reservations r left join customers c on r.customerId=c.customerId
76 order by r.ReserveID;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	ReserveID	customerId	name	age	gender	phone_no	emailId	address
▶	5001	2	Pradhap	21	M	9852364145	pradhap12@gmail.com	Tambaram
	5002	1	Pradesh	20	M	9824213047	pradeshg12@gmail.com	Maraimalainagar
	5003	2	Pradhap	21	M	9852364145	pradhap12@gmail.com	Tambaram
	5004	2	Pradhap	21	M	9852364145	pradhap12@gmail.com	Tambaram
	5005	3	Pradeep	24	M	9852647811	pradeep14@gmail.com	Velachery
	5006	3	Pradeep	24	M	9852647811	pradeep14@gmail.com	Velachery
	5007	3	Pradeep	24	M	9852647811	pradeep14@gmail.com	Velachery
	5008	4	Vikky	15	M	9865471236	Vikky23@gmail.com	Chidambaram
	5009	5	Sharukan	25	M	9645172831	Allish@gmail.com	T.nagar
	5010	5	Sharukan	25	M	9645172831	Allish@gmail.com	T.nagar
	5011	5	Sharukan	25	M	9645172831	Allish@gmail.com	T.nagar
	5012	6	Praveen	21	M	9804786434	praveen0804@gmail.com	Mulapeta

Result 113

Join and Aggregate Function Queries:

1. Calculate the total revenue generated from ticket sales for each event.

```
81 # 1.Calculate the total revenue generated from ticket sales for each event.
82
83 • select e.eventid, e.event_name, sum(s.price) as Total_sales_of_each_event
84 from sales s join tickets t on s.ticketno=t.ticketno
85 join reservations r on r.ReserveID=t.ReserveID
86 join events e on e.eventid=r.EventID
87 where s.Sales_status = "sold"
88 group by e.Event_Name
89 order by e.EventID;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content: [F.A](#)

eventid	event_name	Total_sales_of_each_event
01B25	India Business Summit 2025	3500.00
03B25	Corporate Leadership Summit 2025	2500.00
05E25	Career Expo 2025	3000.00
06E26	International Conference on Education	500.00
07EN25	Comedy Show	3000.00
08EN25	Dance Competition	2000.00
09F25	Designer Showcase 2025	2500.00
10F25	Bridal Fashion Show	1500.00
11S25	India Open Tennis Tournament	1500.00
12E25	AGRI TEC EXPO-2025	3000.00

Result 114

×

2. Find the average price of tickets sold for each category.

```
91 # 2.Find the average price of tickets sold for each category.
92
93 • select e.Category, avg(s.price) as Total_sales_of_each_event
94 from sales s join tickets t on s.ticketno=t.ticketno
95 join reservations r on r.ReserveID=t.ReserveID
96 join events e on e.eventid=r.EventID
97 where s.Sales_status = "sold"
98 group by e.Category;
99
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Category	Total_sales_of_each_event			
▶	Business	857.142857			
	Fashion	666.666667			
	Entertainment	1000.000000			
	Sports	714.285714			
	Education	863.636364			

3. Which events have the top two highest revenues?

```
100 # 3.Which events have the top two highest revenues?.
101
102 • select e.eventId, e.event_name, sum(s.price) Total_Income_Each_Event from sales s
103 join tickets t on t.TicketNO=s.TicketNO
104 join reservations r on r.reserveid=t.reserveid
105 join events e on r.eventid=e.eventid
106 group by e.eventid order by Total_Income_Each_Event desc limit 2;
107
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	eventId	event_name	Total_Income_Each_Event			
▶	14E25	Automate Intelligently	6000.00			
	09F25	Designer Showcase 2025	4000.00			

- Find the total number of events in each category.

```
54 -- AGGREGATE FUNCTIONS --
55
56 -- 1.Find the total number of events in each category --
57 • select Category,count(event_name) from events group by Category;
58
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Category	count(event_name)		
Business	3		
Education	5		
Entertainment	3		
Fashion	2		
Sports	5		

Advanced Queries

Stored Procedure Queries:

- Find all tickets sold for a specific event.

```
118 -- 1.Find all tickets sold for a specific event.(Stored Procedure Method & SUB QUERY)
119
120 DELIMITER $$
121 • CREATE PROCEDURE TICKETS_SOLD_DETAILS(IN EVENT_NAME VARCHAR(50))
122 BEGIN
123     select * from sales where ticketno in (select ticketno from tickets where reserveId in
124     (select r.reserveId from reservations r left join events e on e.eventId=r.eventId
125     where e.event_name=event_name and r.payment_status = "completed"))
126     and sales_status = "sold";
127 END $$
128 DELIMITER ;
129 • CALL TICKETS_SOLD_DETAILS("India Business Summit 2025");
130 • CALL TICKETS_SOLD_DETAILS("Comedy Show");
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	SalesID	TicketNO	Price	SaleDate	Sales_status
▶	25001	2025164	1500.00	2025-05-01	Sold
	25009	2025172	500.00	2025-03-05	Sold
	25016	2025179	1000.00	2025-04-08	Sold
	25051	2025189	500.00	2025-05-13	Sold

2. Get the details of all events for which a specific customer has made a reservation.

```

133 -- 2.Get the details of all events for which a specific customer has made a reservation with "STORED PROCEDURE".
134
135 DELIMITER $$
136 • CREATE PROCEDURE CUSTOMERS_BOOKED_EVENTS_DETAILS(IN CUSTOMERID INT)
137 BEGIN
138 select c.customerid, c.name, E.EventID, E.Event_Name, E.Category, E.Date, E.Address, E.Description from customers c
139     join reservations r on c.customerid=r.customerid
140     join events e on e.eventid=r.eventid
141     where c.customerid=CUSTOMERID;
142 END $$
143 DELIMITER ;
144
145 • call CUSTOMERS_BOOKED_EVENTS_DETAILS(5);

```

customerid	name	EventID	Event_Name	Category	Date	Address	Description
5	Sharukan	05E25	Career Expo 2025	Education	04-MAY-2025	Exhibition Centre, Chidhambaram	The Career Expo 2025 will feature job fairs, car...
5	Sharukan	10F25	Bridal Fashion Show	Fashion	26-DEC-2025	The Leela Palace, Jaipur	Witness the latest bridal fashion trends and des...
5	Sharukan	12E25	AGRI TEC EXPO-2025	Education	29-MAY-2025	Anandha Thirumana Mahal,Viluppuram,	AGRI TEC EXPO - 2025 the objective of this exp...

3. Retrieve the details of all customers who have purchased tickets for a specific event.

```

149 -- 3.Retrieve the details of all customers who have purchased tickets for a specific event with "STORED PROCEDURE".
150
151 DELIMITER $$
152 • CREATE PROCEDURE Specific_Event_Ticket_Purchase_Customer_Information(IN EVENTID INT)
153 BEGIN
154 select e.eventID,C.CustomerID, C.Name, C.Age, C.Gender, C.Phone_no, C.EmailID, C.Address from customers c
155     join reservations r on c.customerid=r.customerid
156     join events e on e.eventid=r.eventid where e.eventid=EVENTID;
157 END $$
158 DELIMITER ;
159
160 • CALL Specific_Event_Ticket_Purchase_Customer_Information("07EN25");
161 • CALL Specific_Event_Ticket_Purchase_Customer_Information("17S25");

```

eventID	CustomerID	Name	Age	Gender	Phone_no	EmailID	Address
07EN25	2	Pradhap	21	M	9852364145	pradhap12@gmail.com	Tambaram
07EN25	12	Sathana	14	F	9841257436	sathu3920@gmail.com	Yelagiri
07EN25	24	Amit Kumar	22	M	7654321098	amit@outlook.com	Bangalore
07EN25	32	Suresh Kumar	31	M	8765432108	suresh@gmail.com	Erode, Tamil Nadu
07EN25	40	Raj Kumar	37	M	9876543212	rajkumar@gmail.com	Perambalur, Tamil Nadu

Result 125 x

CTE and Window Function Queries:

1. Who was the first customer to purchase a ticket for each event?

```
166 -- 1.Who was the first customer to purchase a ticket for each event (using CTE & WINDOW function).
167
168 • WITH event_first_booked_date as (
169     select c.CustomerID, c.Name, c.Age, c.Gender, c.Phone_no, c.EmailID, c.Address, R.ReserveID,
170     e.EventID, r.ReserveDATE, r.PaymentType, r.Payment_Status,
171     row_number() over(partition by EventID order by ReserveDATE) as rdate from reservations r
172     join customers c on r.CustomerID=c.CustomerID
173     join events e on e.EventID=r.eventid)
174
175     select EventID, CustomerID, Name, Age, Gender, Phone_no, EmailID, Address
176     from event_first_booked_date where rdate=1 and payment_status ="completed";
```

	EventID	CustomerID	Name	Age	Gender	Phone_no	EmailID	Address
▶	01B25	30	Sachin Gupta	38	M	1098765432	sachin@outlook.com	Jaipur
	03B25	33	Pooja Singh	27	F	7654321099	pooja@yahoo.com	Cuddalore, Tamil Nadu
	05E25	5	Sharukan	25	M	9645172831	Alish@gmail.com	T.nagar
	06E26	65	Vikas Singh	34	M	9945234569	vikas@gmail.com	Perambalur, Tamil Nadu
	08EN25	33	Pooja Singh	27	F	7654321099	pooja@yahoo.com	Cuddalore, Tamil Nadu
	09F25	22	Rohan Sharma	25	M	9876543210	rohan@gmail.com	New Delhi
	10F25	27	Kavita Jain	29	F	4321098765	kavita@outlook.com	Pune
	11S25	25	Sneha Jain	28	F	6543210987	sneha@gmail.com	Hyderabad
	12E25	5	Sharukan	25	M	9645172831	Alish@gmail.com	T.nagar
	15S25	61	Amit Kumar	36	M	9445234569	amit@outlook.com	Kancheepuram, Tamil Nadu

2. How many tickets were sold for each event?

```
179 -- 2.How many tickets were sold for each event? (using CTE & WINDOW Function).
180
181 • WITH total_count as(
182     select e.EventID, e.Event_Name, s.SaleDate, count(r.ReserveID)over(partition by EventId) as Tickets_sales_count,
183     row_number() over(partition by Event_Name order by SaleDate desc) as Rownum from sales s
184     join tickets t on t.TicketNO=s.TicketNO
185     join reservations r on r.reserveid=t.reserveid
186     join events e on r.eventid=e.eventid)
187
188     select Eventid, Event_name, Tickets_Sales_Count from total_count where rownum=1;
189
```

	Eventid	Event_name	Tickets_Sales_Count
▶	11S25	India Open Tennis Tournament	4
	12E25	AGRI TEC EXPO-2025	4
	14E25	Automate Intelligently	7
	10F25	Bridal Fashion Show	3
	05E25	Career Expo 2025	4
	07EN25	Comedy Show	2
	03B25	Corporate Leadership Summit 2025	4
	08EN25	Dance Competition	3
	09F25	Designer Showcase 2025	5
	01B25	India Business Summit 2025	5

View-Based Queries:

1. Create a view for ticket booking customer details.

```
190
191 -- USING VIEW --
192
193 -- 1.Create view for Ticket Booking Customer Details.(view)
194 • create view Customer_Details_for_Booked_Tickets as
195 select r.ReserveID,c.customerId, c.name, c.age, c.gender, c.phone_no, c.emailId, c.address from
196 reservations r left join customers c on r.customerId=c.customerId
197 order by r.ReserveID;
198
199 • select * from Customer_Details_for_Booked_Tickets;
200
```

Result Grid								
Filter Rows: <input type="text"/> Export: Wrap Cell Content:								
	ReserveID	customerId	name	age	gender	phone_no	emailId	address
▶	5001	2	Pradhap	21	M	9852364145	pradhap12@gmail.com	Tambaram
	5002	1	Pradesh	20	M	9824213047	pradeshg12@gmail.com	Maraimalaiagar
	5003	2	Pradhap	21	M	9852364145	pradhap12@gmail.com	Tambaram
	5004	2	Pradhap	21	M	9852364145	pradhap12@gmail.com	Tambaram
	5005	3	Pradeep	24	M	9852647811	pradeep14@gmail.com	Velachery
	5006	3	Pradeep	24	M	9852647811	pradeep14@gmail.com	Velachery
	5007	3	Pradeep	24	M	9852647811	pradeep14@gmail.com	Velachery
	5008	4	Vikky	15	M	9865471236	Vikky23@gmail.com	Chidambaram
	5009	5	Sharukan	25	M	9645172831	Allish@gmail.com	T.nagar
	5010	5	Sharukan	25	M	9645172831	Allish@gmail.com	T.nagar