SQL File 3*

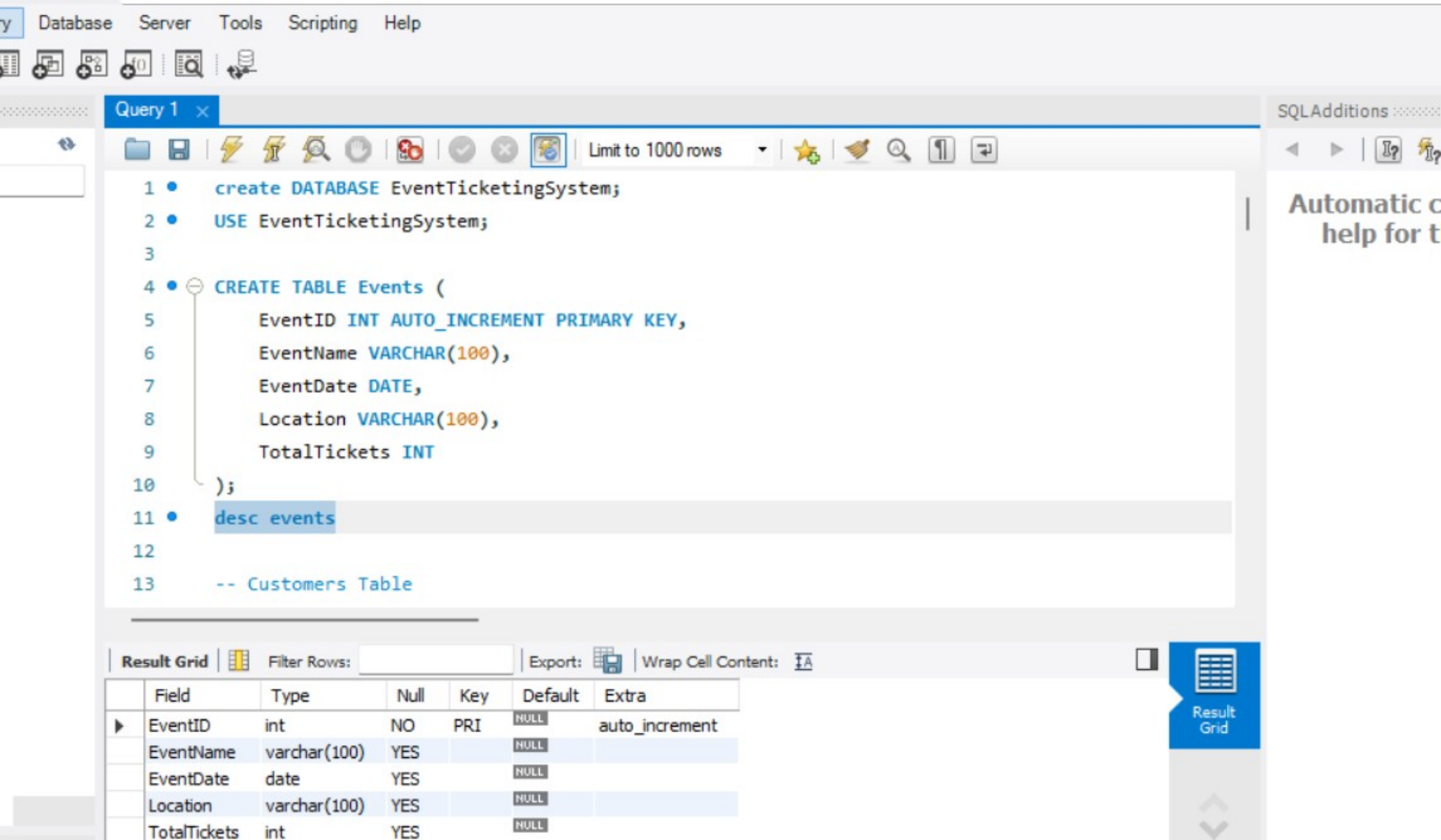
Limit to 1000 rows

```
1  create DATABASE EventTicketingSystem;
2  USE EventTicketingSystem;
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
```

Output

Action Output

#	Time	Action	Message
1	17:24:19	create DATABASE EventTicketingSystem	1 row(s) affected
2	17:24:19	USE EventTicketingSystem	0 row(s) affected


Database
Server
Tools
Scripting
Help
Query 1
Limit to 1000 rows
SQLAdditions
Automatic c
help for t
1 create DATABASE EventTicketingSystem;
2 USE EventTicketingSystem;
3
4 CREATE TABLE Events (
5 EventID INT AUTO_INCREMENT PRIMARY KEY,
6 EventName VARCHAR(100),
7 EventDate DATE,
8 Location VARCHAR(100),
9 TotalTickets INT
10);
11 desc events
12
13 -- Customers Table
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Field Type Null Key Default Extra
EventID int NO PRI NULL auto_increment
EventName varchar(100) YES NULL
EventDate date YES NULL
Location varchar(100) YES NULL
TotalTickets int YES NULL
Result Grid

Query 1

Limit to 1000 rows

```sql
-- Customers Table
CREATE TABLE Customers (
    CustomerID INT AUTO_INCREMENT PRIMARY KEY,
    CustomerName VARCHAR(100),
    Email VARCHAR(100)
);
desc customers

-- Tickets Table
CREATE TABLE Tickets (
    TicketID INT AUTO_INCREMENT PRIMARY KEY,
    EventID INT,
    TicketType VARCHAR(50),
```

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

Field	Type	Null	Key	Default	Extra
CustomerID	int	NO	PRI	NULL	auto_increment
CustomerName	varchar(100)	YES		NULL	
Email	varchar(100)	YES		NULL	

Result Grid

SQL File 3*

Limit to 1000 rows

```
16      Email VARCHAR(100)
17  );
18
19  -- Tickets Table
20  CREATE TABLE Tickets (
21      TicketID INT AUTO_INCREMENT PRIMARY KEY,
22      EventID INT,
23      TicketType VARCHAR(50),
24      Price DECIMAL(8,2),
25      Status VARCHAR(20) DEFAULT 'Available',  -- Available, Sold, Reserved
26      FOREIGN KEY (EventID) REFERENCES Events(EventID)
27  );
28  desc Tickets
```

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

Field	Type	Null	Key	Default	Extra
TicketID	int	NO	PRI	NULL	auto_increment
EventID	int	YES	MUL	NULL	
TicketType	varchar(50)	YES		NULL	
Price	decimal(8,2)	YES		NULL	
Status	varchar(20)	YES		Available	

Result Grid

SQL File 3*

Limit to 1000 rows

SQLAdditions

Automatic cont
help for the

```
30  CREATE TABLE Reservations (
31      ReservationID INT AUTO_INCREMENT PRIMARY KEY,
32      CustomerID INT,
33      TicketID INT,
34      ReservationDate DATE,
35      FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
36      FOREIGN KEY (TicketID) REFERENCES Tickets(TicketID)
37  );
38  desc Reservations
39
40
41  -- Sales Table
42  CREATE TABLE Sales (
```

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

Field	Type	Null	Key	Default	Extra
ReservationID	int	NO	PRI	NULL	auto_increment
CustomerID	int	YES	MUL	NULL	
TicketID	int	YES	MUL	NULL	
ReservationDate	date	YES		NULL	

Result Grid

SQL File 3*

Limit to 1000 rows

```
43
44      -- Sales Table
45  CREATE TABLE Sales (
46          SaleID INT AUTO_INCREMENT PRIMARY KEY,
47          CustomerID INT,
48          TicketID INT,
49          SaleDate DATE,
50          Amount DECIMAL(8,2),
51          FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
52          FOREIGN KEY (TicketID) REFERENCES Tickets(TicketID)
53      );
54      desc sales
55      -- Insert into Events
```

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

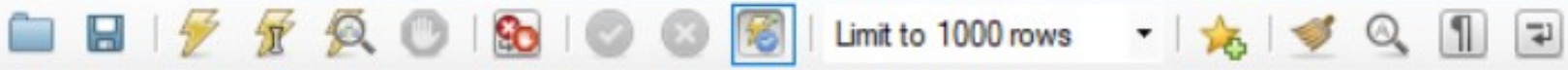
Field	Type	Null	Key	Default	Extra
SaleID	int	NO	PRI	NULL	auto_increment
CustomerID	int	YES	MUL	NULL	
TicketID	int	YES	MUL	NULL	
SaleDate	date	YES		NULL	
Amount	decimal(8,2)	YES		NULL	

Result Grid

Result 5

Read Only

SQL File 3*




```
56    -- Insert into Events
57    INSERT INTO Events (EventName, EventDate, Location, TotalTickets) VALUES
58    ('Music Concert', '2025-05-10', 'palikarani', 100),
59    ('circket match', '2025-06-15', 'medavakam', 50),
60    ('dances', '2025-05-22', 'tambaram', 200);
61    select * from events;
62
63    -- Insert into Customers
64    INSERT INTO Customers (CustomerName, Email) VALUES
65    ('guru', 'guru@gmail.com'),
66    ('velu', 'velu@gmail.com'),
```

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

EventID	EventName	EventDate	Location	TotalTickets
1	Music Concert	2025-05-10	palikarani	100
2	circket match	2025-06-15	medavakam	50
3	dances	2025-05-22	tambaram	200

Result Grid

SQL File 3*

Limit to 1000 rows

```
62
63    -- Insert into Customers
64 •  INSERT INTO Customers (CustomerName, Email) VALUES
65    ('guru', 'guru@gmail.com'),
66    ('velu', 'velu@gmail.com'),
67    ('siva', 'siva@gmail.com');
68 •  select * from customers;
69
70    -- Insert into Tickets
71 •  INSERT INTO Tickets (EventID, TicketType, Price) VALUES
72    (1, 'VIP', 150.00),
```

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

CustomerID	CustomerName	Email
1	guru	guru@gmail.com
2	velu	velu@gmail.com
3	siva	siva@gmail.com

Result Grid

SQL File 3*

Limit to 1000 rows

```
70    -- Insert into Tickets
71 •  INSERT INTO Tickets (EventID, TicketType, Price) VALUES
72    (1, 'VIP', 150.00),
73    (1, 'Regular', 80.00),
74    (2, 'Regular', 50.00),
75    (3, 'VIP', 200.00),
76    (3, 'Regular', 100.00);
77
78    select * from Tickets;
79
80    -- Insert into Reservations
```

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

TicketID	EventID	TicketType	Price	Status
1	1	VIP	150.00	Available
2	1	Regular	80.00	Available
3	2	Regular	50.00	Available
4	3	VIP	200.00	Available
5	3	Regular	100.00	Available

Result Grid

SQL File 3*

Limit to 1000 rows

```
79
80    -- Insert into Reservations
81 •  INSERT INTO Reservations (CustomerID, TicketID, ReservationDate) VALUES
82    (1, 1, '2025-04-01'),
83    (2, 3, '2025-04-03');
84 •  select * from reservations
85
86    -- Insert into Sales
87    INSERT INTO Sales (CustomerID, TicketID, SaleDate, Amount) VALUES
88    (1, 2, '2025-04-05', 80.00),
89    (2, 4, '2025-04-10', 200.00),
```

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

ReservationID	CustomerID	TicketID	ReservationDate
1	1	1	2025-04-01
2	2	3	2025-04-03

Result Grid

SQL File 3*

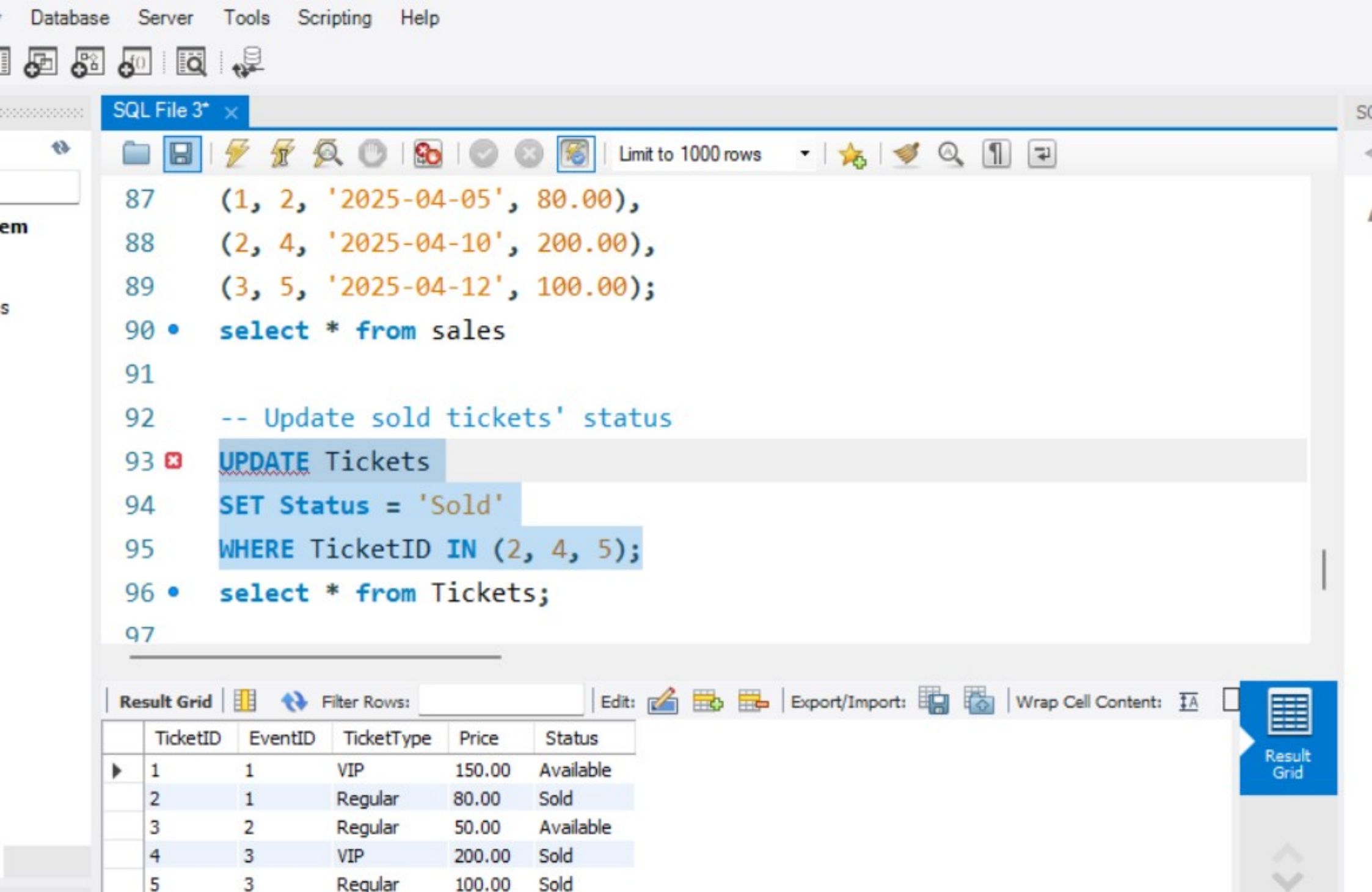
Limit to 1000 rows

```
84 •  select * from reservations
85
86    -- Insert into Sales
87    INSERT INTO Sales (CustomerID, TicketID, SaleDate, Amount) VALUES
88    (1, 2, '2025-04-05', 80.00),
89    (2, 4, '2025-04-10', 200.00),
90    (3, 5, '2025-04-12', 100.00);
91 •  select * from sales
92
93
94
```

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

SaleID	CustomerID	TicketID	SaleDate	Amount
1	1	2	2025-04-05	80.00
2	2	4	2025-04-10	200.00
3	3	5	2025-04-12	100.00

Result Grid


Database Server Tools Scripting Help
SQL File 3*
Limit to 1000 rows
87 (1, 2, '2025-04-05', 80.00),
88 (2, 4, '2025-04-10', 200.00),
89 (3, 5, '2025-04-12', 100.00);
90 select * from sales
91
92 -- Update sold tickets' status
93 UPDATE Tickets
94 SET Status = 'Sold'
95 WHERE TicketID IN (2, 4, 5);
96 select * from Tickets;
97
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
TicketID EventID TicketType Price Status
1 1 VIP 150.00 Available
2 1 Regular 80.00 Sold
3 2 Regular 50.00 Available
4 3 VIP 200.00 Sold
5 3 Regular 100.00 Sold
Result Grid

SQL File 3*

Limit to 1000 rows

```
92    -- Update sold tickets' status
93    UPDATE Tickets
94    SET Status = 'Sold'
95    WHERE TicketID IN (2, 4, 5);
96
97    -- Update reserved tickets' status
98    UPDATE Tickets
99    SET Status = 'Reserved'
100   WHERE TicketID IN (1, 3);
101
102
```

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

TicketID	EventID	TicketType	Price	Status
1	1	VIP	150.00	Reserved
2	1	Regular	80.00	Sold
3	2	Regular	50.00	Reserved
4	3	VIP	200.00	Sold
5	3	Regular	100.00	Sold

Result Grid

Database Server Tools Scripting Help

SQL File 3*

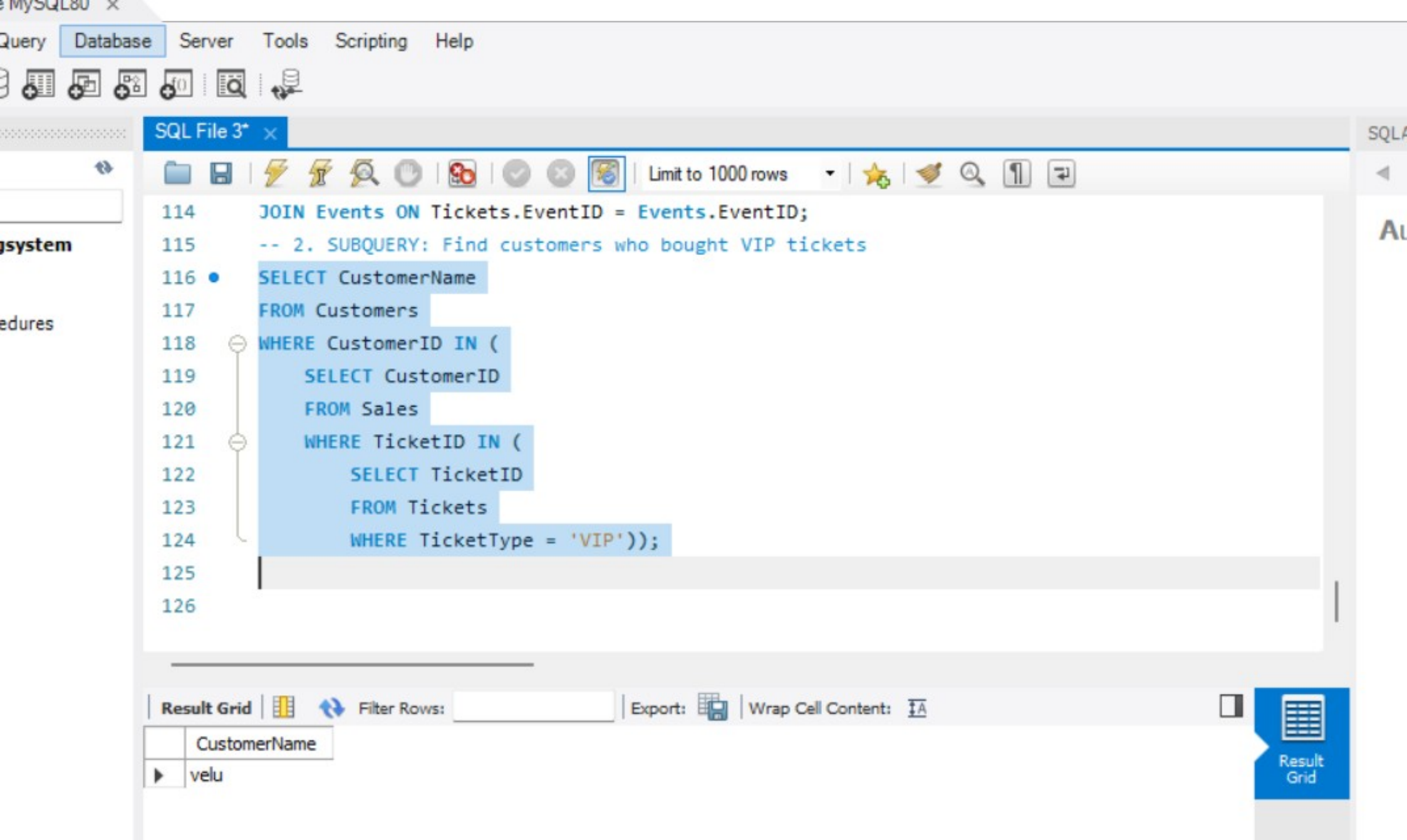
Limit to 1000 rows

```
100     WHERE TicketID IN (1, 3);
101     -- 1. JOIN: List customer names and the events they reserved or bought tickets
102     SELECT Customers.CustomerName, Events.EventName, Tickets.TicketType, Tickets.Price
103     FROM Customers
104     JOIN Reservations ON Customers.CustomerID = Reservations.CustomerID
105     JOIN Tickets ON Reservations.TicketID = Tickets.TicketID
106     JOIN Events ON Tickets.EventID = Events.EventID
107     UNION ALL
108     SELECT Customers.CustomerName, Events.EventName, Tickets.TicketType, Tickets.Price
109     FROM Customers
110     JOIN Sales ON Customers.CustomerID = Sales.CustomerID
111     JOIN Tickets ON Sales.TicketID = Tickets.TicketID
112     JOIN Events ON Tickets.EventID = Events.EventID;
113
```

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

CustomerName	EventName	TicketType	Price
guru	Music Concert	VIP	150.00
velu	circket match	Regular	50.00
guru	Music Concert	Regular	80.00
velu	dances	VIP	200.00
siva	dances	Regular	100.00

Result Grid


Query
Database
Server
Tools
Scripting
Help
SQL File 3*
Limit to 1000 rows
114 JOIN Events ON Tickets.EventID = Events.EventID;
115 -- 2. SUBQUERY: Find customers who bought VIP tickets
116 SELECT CustomerName
117 FROM Customers
118 WHERE CustomerID IN (
119 SELECT CustomerID
120 FROM Sales
121 WHERE TicketID IN (
122 SELECT TicketID
123 FROM Tickets
124 WHERE TicketType = 'VIP'));
125
126
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
CustomerName
velu
Result Grid

SQL File 3*

Limit to 1000 rows

```
123         WHERE TicketType = 'VIP'
124     )
125 );
126 -- 3. VIEW: Create a view to see all ticket sales with event names
127 -- 4. Using the VIEW: Display all ticket sales
128 CREATE VIEW TicketSalesView AS
129 SELECT Sales.SaleID, Customers.CustomerName, Events.EventName, Tickets.TicketType, Sales.Amou
130 FROM Sales
131 JOIN Customers ON Sales.CustomerID = Customers.CustomerID
132 JOIN Tickets ON Sales.TicketID = Tickets.TicketID
133 JOIN Events ON Tickets.EventID = Events.EventID;
134
135 SELECT * FROM TicketSalesView;
136
```

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

SaleID	CustomerName	EventName	TicketType	Amount	SaleDate
1	guru	Music Concert	Regular	80.00	2025-04-05
2	velu	dances	VIP	200.00	2025-04-10
3	siva	dances	Regular	100.00	2025-04-12

Result Grid

ce MySQL80 ×

Query Database Server Tools Scripting Help

SQL File 3* ×

Limit to 1000 rows

```
135 •  SELECT * FROM TicketSalesView;
136
137    -- 5. STORED PROCEDURE: Count number of tickets sold for an event
138    DELIMITER //
139 •  CREATE PROCEDURE GetSoldTicketCount(IN inputEventID INT)
140    BEGIN
141        SELECT COUNT(*) AS SoldTickets
142        FROM Tickets
143        WHERE EventID = inputEventID AND Status = 'Sold';
144    END //
145    DELIMITER ;
146 •  -- Execute Stored Procedure Example
147    call GetSoldTicketCount(1)
```

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

SoldTickets
1

Result Grid

SQL Additions

Automatic
help for