PROJECT FOR SQL MODULE

Swiggy food delivery database.

Name – SIDDHAY SUBHASH SHETYE

Description

1)Users:
The database would have a table to store user information. This includes details such as user ID, username, email, password and address.

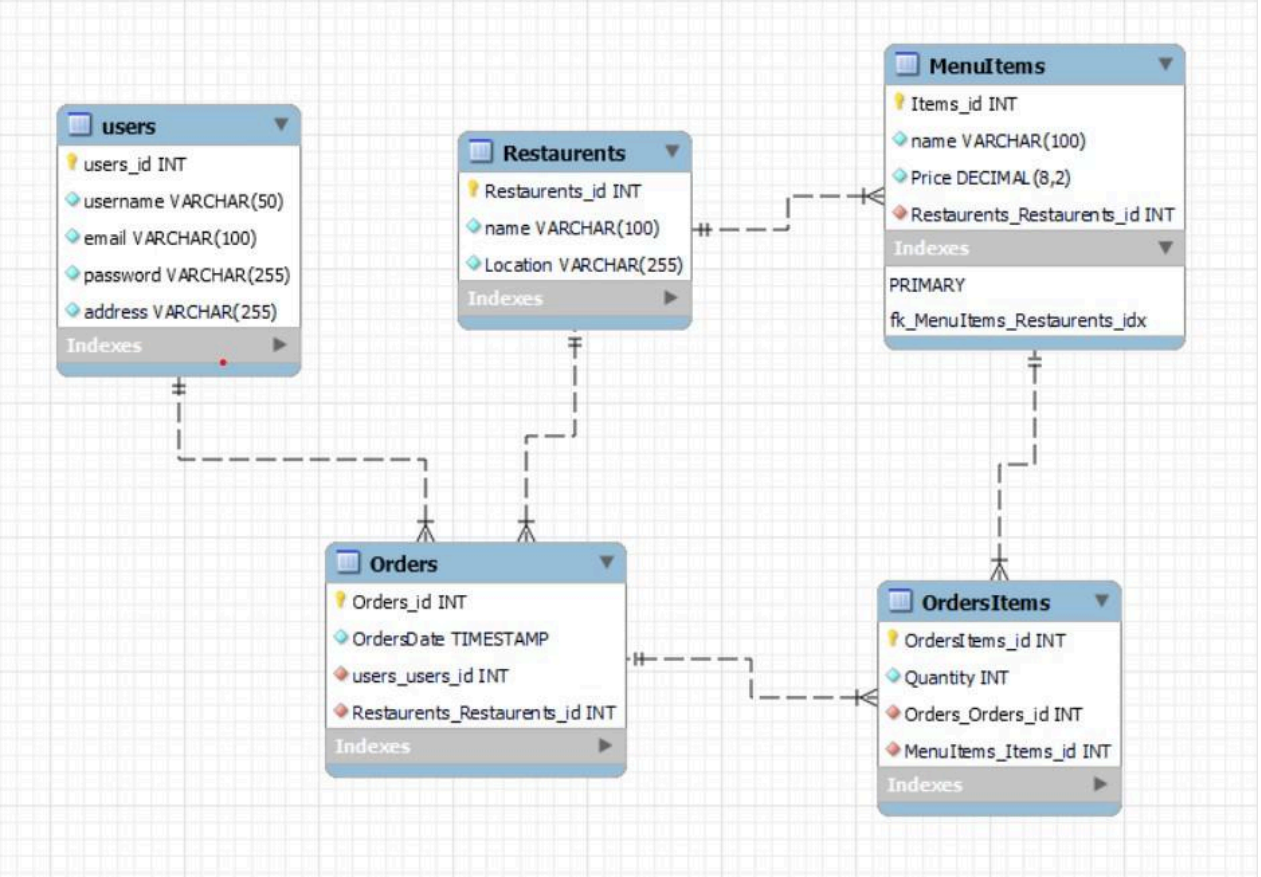
2)Restaurants:
Another table would be dedicated to storing information about restaurants. This could include Name and location.

3)Menu Items:
There would be a table to store the menu items offered by each restaurant. This table might include restaurantid, name, price.

4)Orders:
A table for managing orders would store information about each order, such as user ID (who placed the order), restaurant ID (from where the order is placed).

5)Order Items:
To capture the items within each order, there might be a table for order items. This could include details such as order ID (to associate with a specific order), item ID (to link with the menu item) and quantity.

ER-Diagram (Entity Relation-Diagram) swiggy food delivery database.



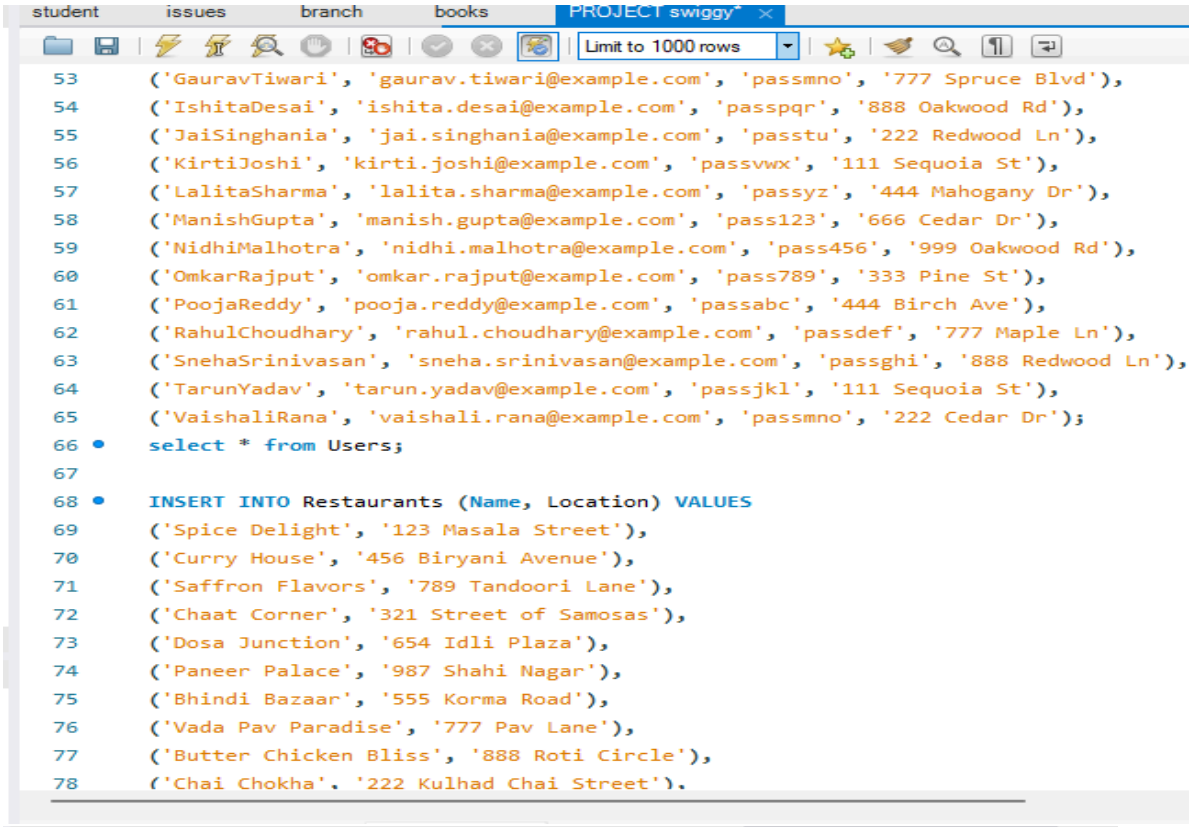
Commands

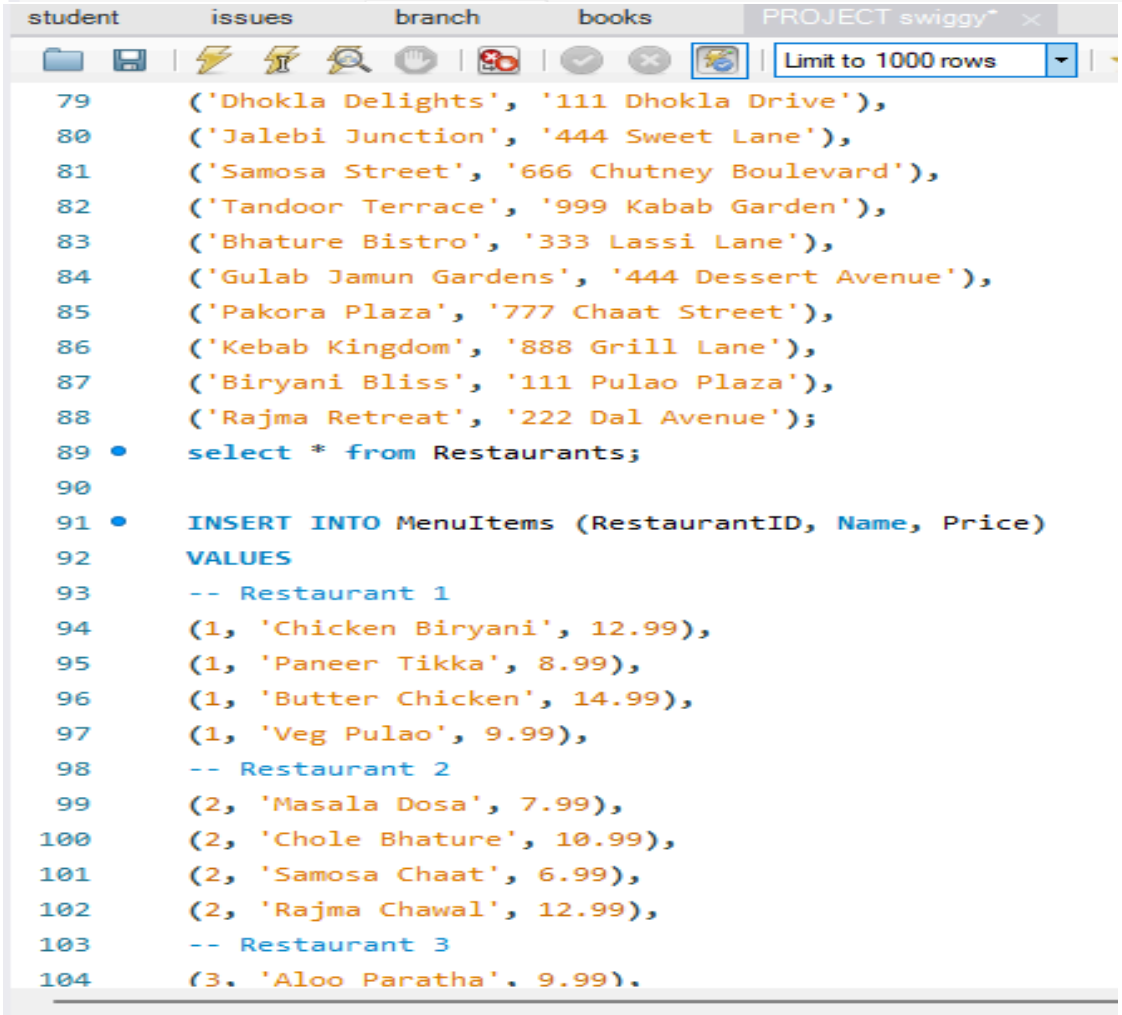
student issues branch books PROJECT swiggy* Limit to 1000 rows

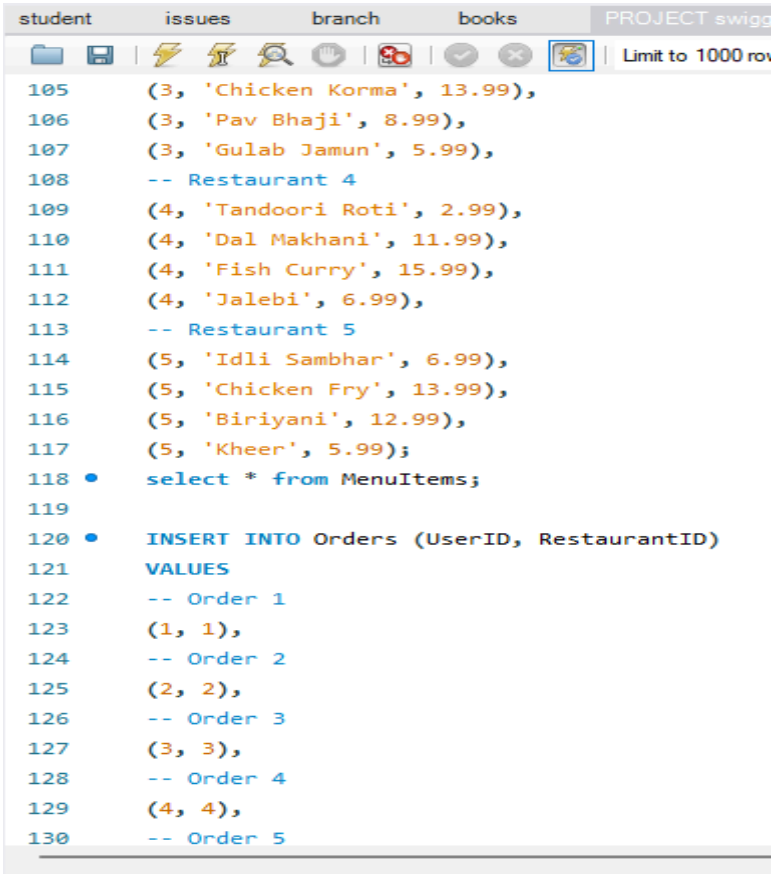
```
1  create database KA;
2  use KA;
3
4  CREATE TABLE Users (
5      UserID INT PRIMARY KEY AUTO_INCREMENT,
6      Username VARCHAR(50) NOT NULL,
7      Email VARCHAR(100) NOT NULL,
8      Password VARCHAR(255) NOT NULL,
9      Address VARCHAR(255) NOT NULL
10 );
11
12
13 CREATE TABLE Restaurants (
14     RestaurantID INT PRIMARY KEY AUTO_INCREMENT,
15     Name VARCHAR(100) NOT NULL,
16     Location VARCHAR(255) NOT NULL
17 );
18
19 CREATE TABLE MenuItems (
20     ItemID INT PRIMARY KEY AUTO_INCREMENT,
21     RestaurantID INT,
22     Name VARCHAR(100) NOT NULL,
23     Price DECIMAL(8, 2) NOT NULL,
24     FOREIGN KEY (RestaurantID) REFERENCES Restaurants(RestaurantID)
25 );
26
```

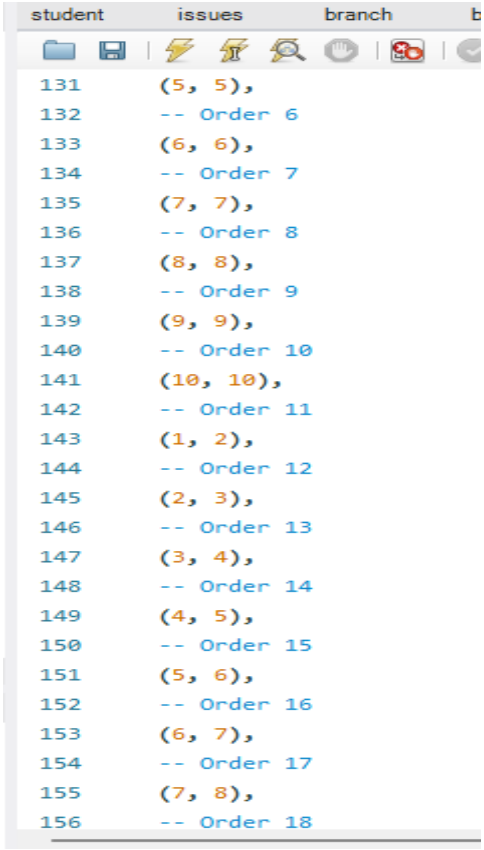
student issues branch books PROJECT swiggy* Limit to 1000 rows

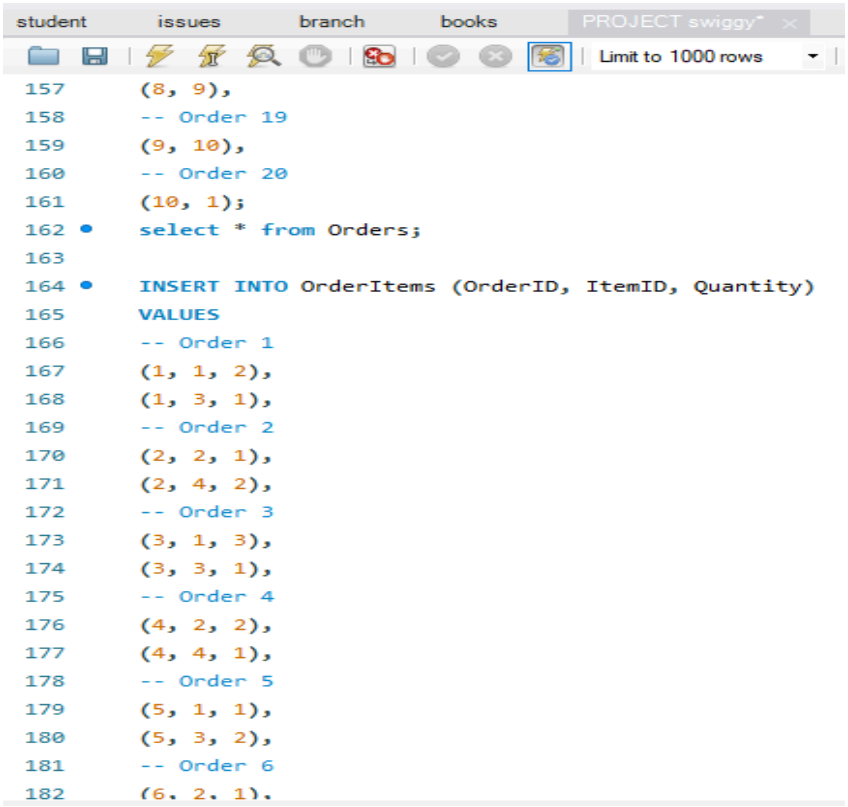
```
27 CREATE TABLE Orders (
28     OrderID INT PRIMARY KEY AUTO_INCREMENT,
29     UserID INT,
30     RestaurantID INT,
31     OrderDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
32     FOREIGN KEY (UserID) REFERENCES Users(UserID),
33     FOREIGN KEY (RestaurantID) REFERENCES Restaurants(RestaurantID)
34 );
35
36 CREATE TABLE OrderItems (
37     OrderItemID INT PRIMARY KEY AUTO_INCREMENT,
38     OrderID INT,
39     ItemID INT,
40     Quantity INT NOT NULL,
41     FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
42     FOREIGN KEY (ItemID) REFERENCES MenuItems(ItemID)
43 );
44
45 INSERT INTO Users (Username, Email, Password, Address) VALUES
46 ('AdityaSharma', 'aditya.sharma@example.com', 'pass123', '123 Main St'),
47 ('AishwaryaPatel', 'aishwarya.patel@example.com', 'pass456', '456 Oak Ave'),
48 ('ArjunKumar', 'arjun.kumar@example.com', 'pass789', '789 Maple Ln'),
49 ('BhavyaSingh', 'bhavya.singh@example.com', 'passabc', '321 Pine St'),
50 ('ChetnaMehta', 'chetna.mehta@example.com', 'passdef', '654 Elm Rd'),
51 ('DeepakVerma', 'deepak.verma@example.com', 'passghi', '987 Cedar Dr'),
52 ('EktaKapoor', 'ekta.kapoor@example.com', 'passjkl', '555 Birch Ave'),
```

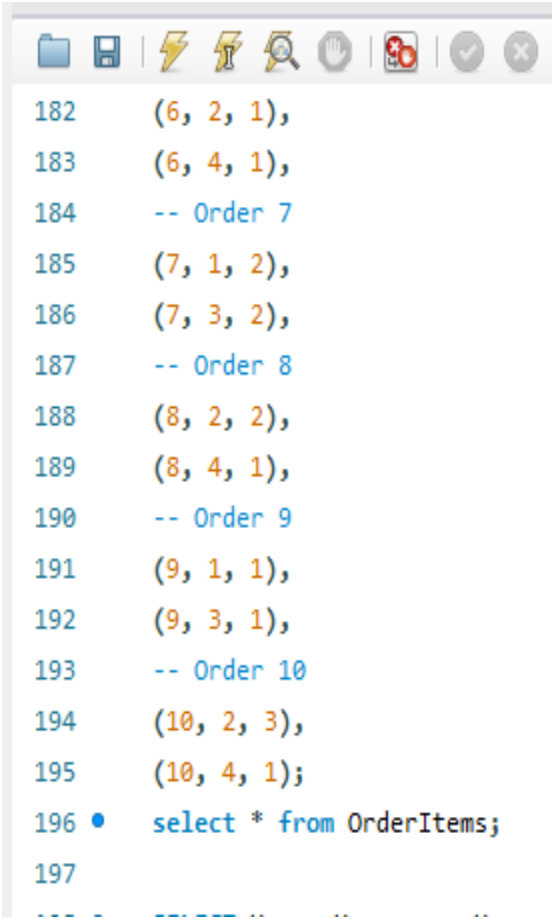

student
issues
branch
books
PROJECT swiggy*
Limit to 1000 rows
('GauravTiwari', 'gaurav.tiwari@example.com', 'passmno', '777 Spruce Blvd'),
('IshitaDesai', 'ishita.desai@example.com', 'passpqr', '888 Oakwood Rd'),
('JaiSinghania', 'jai.singhania@example.com', 'passtu', '222 Redwood Ln'),
('KirtiJoshi', 'kirti.joshi@example.com', 'passvwx', '111 Sequoia St'),
('LalitaSharma', 'lalita.sharma@example.com', 'passyz', '444 Mahogany Dr'),
('ManishGupta', 'manish.gupta@example.com', 'pass123', '666 Cedar Dr'),
('NidhiMalhotra', 'nidhi.malhotra@example.com', 'pass456', '999 Oakwood Rd'),
('OmkarRajput', 'omkar.rajput@example.com', 'pass789', '333 Pine St'),
('PoojaReddy', 'pooja.reddy@example.com', 'passabc', '444 Birch Ave'),
('RahulChoudhary', 'rahul.choudhary@example.com', 'passdef', '777 Maple Ln'),
('SnehaSrinivasan', 'sneha.srinivasan@example.com', 'passghi', '888 Redwood Ln'),
('TarunYadav', 'tarun.yadav@example.com', 'passjkl', '111 Sequoia St'),
('VaishaliRana', 'vaishali.rana@example.com', 'passmno', '222 Cedar Dr');
select * from Users;
INSERT INTO Restaurants (Name, Location) VALUES
('Spice Delight', '123 Masala Street'),
('Curry House', '456 Biryani Avenue'),
('Saffron Flavors', '789 Tandoori Lane'),
('Chaat Corner', '321 Street of Samosas'),
('Dosa Junction', '654 Idli Plaza'),
('Paneer Palace', '987 Shahi Nagar'),
('Bhindi Bazaar', '555 Korma Road'),
('Vada Pav Paradise', '777 Pav Lane'),
('Butter Chicken Bliss', '888 Roti Circle'),
('Chai Chokha', '222 Kulhad Chai Street'),


student
issues
branch
books
PROJECT swiggy*
Limit to 1000 rows
('Dhokla Delights', '111 Dhokla Drive'),
('Jalebi Junction', '444 Sweet Lane'),
('Samosa Street', '666 Chutney Boulevard'),
('Tandoor Terrace', '999 Kabab Garden'),
('Bhature Bistro', '333 Lassi Lane'),
('Gulab Jamun Gardens', '444 Dessert Avenue'),
('Pakora Plaza', '777 Chaat Street'),
('Kebab Kingdom', '888 Grill Lane'),
('Biryani Bliss', '111 Pulao Plaza'),
('Rajma Retreat', '222 Dal Avenue');
select * from Restaurants;
INSERT INTO MenuItems (RestaurantID, Name, Price)
VALUES
-- Restaurant 1
(1, 'Chicken Biryani', 12.99),
(1, 'Paneer Tikka', 8.99),
(1, 'Butter Chicken', 14.99),
(1, 'Veg Pulao', 9.99),
-- Restaurant 2
(2, 'Masala Dosa', 7.99),
(2, 'Chole Bhature', 10.99),
(2, 'Samosa Chaat', 6.99),
(2, 'Rajma Chawal', 12.99),
-- Restaurant 3
(3, 'Aloo Paratha', 9.99),


student
issues
branch
books
PROJECT swigg
Limit to 1000 ro
105 (3, 'Chicken Korma', 13.99),
106 (3, 'Pav Bhaji', 8.99),
107 (3, 'Gulab Jamun', 5.99),
108 -- Restaurant 4
109 (4, 'Tandoori Roti', 2.99),
110 (4, 'Dal Makhani', 11.99),
111 (4, 'Fish Curry', 15.99),
112 (4, 'Jalebi', 6.99),
113 -- Restaurant 5
114 (5, 'Idli Sambhar', 6.99),
115 (5, 'Chicken Fry', 13.99),
116 (5, 'Biriyani', 12.99),
117 (5, 'Kheer', 5.99);
118 select * from MenuItems;
119
120 INSERT INTO Orders (UserID, RestaurantID)
121 VALUES
122 -- Order 1
123 (1, 1),
124 -- Order 2
125 (2, 2),
126 -- Order 3
127 (3, 3),
128 -- Order 4
129 (4, 4),
130 -- Order 5


student
issues
branch
131 (5, 5),
132 -- Order 6
133 (6, 6),
134 -- Order 7
135 (7, 7),
136 -- Order 8
137 (8, 8),
138 -- Order 9
139 (9, 9),
140 -- Order 10
141 (10, 10),
142 -- Order 11
143 (1, 2),
144 -- Order 12
145 (2, 3),
146 -- Order 13
147 (3, 4),
148 -- Order 14
149 (4, 5),
150 -- Order 15
151 (5, 6),
152 -- Order 16
153 (6, 7),
154 -- Order 17
155 (7, 8),
156 -- Order 18


student
issues
branch
books
PROJECT swiggy*
Limit to 1000 rows
(8, 9),
-- Order 19
(9, 10),
-- Order 20
(10, 1);
select * from Orders;
INSERT INTO OrderItems (OrderID, ItemID, Quantity)
VALUES
-- Order 1
(1, 1, 2),
(1, 3, 1),
-- Order 2
(2, 2, 1),
(2, 4, 2),
-- Order 3
(3, 1, 3),
(3, 3, 1),
-- Order 4
(4, 2, 2),
(4, 4, 1),
-- Order 5
(5, 1, 1),
(5, 3, 2),
-- Order 6
(6, 2, 1),


(6, 2, 1),
(6, 4, 1),
-- Order 7
(7, 1, 2),
(7, 3, 2),
-- Order 8
(8, 2, 2),
(8, 4, 1),
-- Order 9
(9, 1, 1),
(9, 3, 1),
-- Order 10
(10, 2, 3),
(10, 4, 1);
select * from OrderItems;

JOINS

1. Write a query to get users orders

student issues branch books PROJECT swiggy* SQL File 7*

Limit to 1000 rows

```
1  SELECT Users.Username, Users.Email, Orders.OrderID, Orders.OrderDate
2  FROM Users
3  JOIN Orders ON Users.UserID = Orders.UserID;
```

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

Username	Email	OrderID	OrderDate
AdityaSharma	aditya.sharma@example.com	1	2024-01-10 15:29:25
AishwaryaPatel	aishwarya.patel@example.com	2	2024-01-10 15:29:25
ArjunKumar	arjun.kumar@example.com	3	2024-01-10 15:29:25
BhavyaSingh	bhavya.singh@example.com	4	2024-01-10 15:29:25
ChetnaMehta	chetna.mehta@example.com	5	2024-01-10 15:29:25
DeepakVerma	deepak.verma@example.com	6	2024-01-10 15:29:25
EktaKapoor	ekta.kapoor@example.com	7	2024-01-10 15:29:25
GauravTiwari	gaurav.tiwari@example.com	8	2024-01-10 15:29:25
IshitaDesai	ishita.desai@example.com	9	2024-01-10 15:29:25
JaiSinghania	jai.singhania@example.com	10	2024-01-10 15:29:25
AdityaSharma	aditya.sharma@example.com	11	2024-01-10 15:29:25
AishwaryaPatel	aishwarya.patel@example.com	12	2024-01-10 15:29:25
ArjunKumar	arjun.kumar@example.com	13	2024-01-10 15:29:25
BhavyaSingh	bhavya.singh@example.com	14	2024-01-10 15:29:25
ChetnaMehta	chetna.mehta@example.com	15	2024-01-10 15:29:25
DeepakVerma	deepak.verma@example.com	16	2024-01-10 15:29:25
EktaKapoor	ekta.kapoor@example.com	17	2024-01-10 15:29:25
GauravTiwari	gaurav.tiwari@example.com	18	2024-01-10 15:29:25
IshitaDesai	ishita.desai@example.com	19	2024-01-10 15:29:25
JaiSinghania	jai.singhania@example.com	20	2024-01-10 15:29:25

2. Write a query to get Order Details with Menu Item Information Retrieval

student issues branch books PROJECT swiggy* SQL File 7*

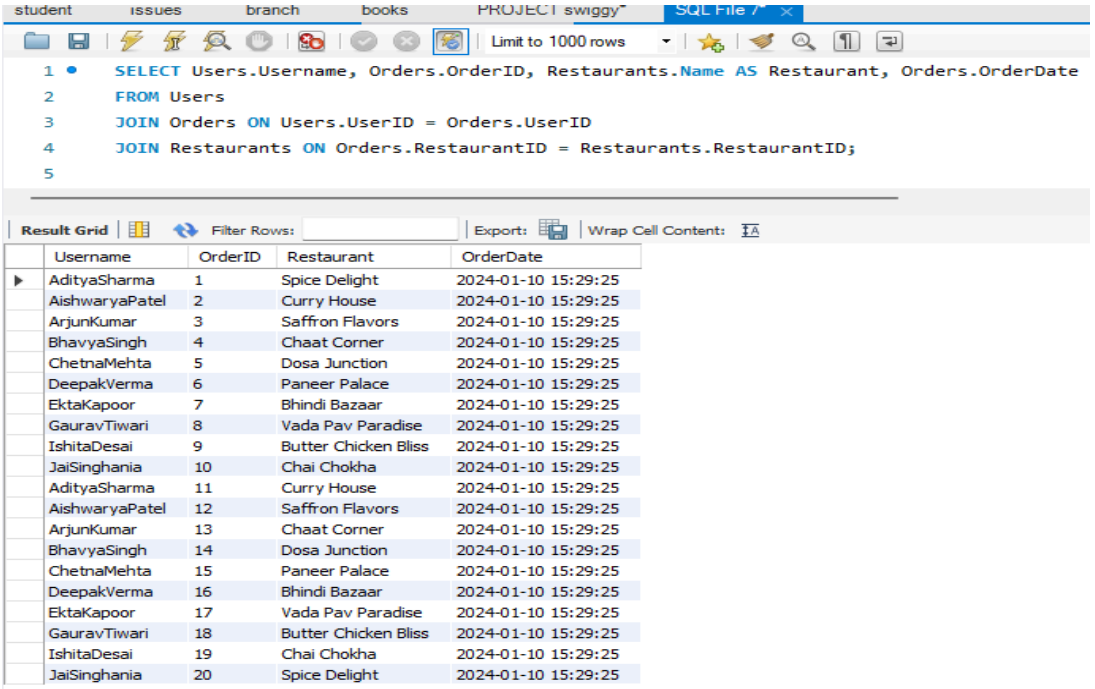
Limit to 1000 rows

```
1  SELECT Orders.OrderID, MenuItems.Name AS MenuItem, MenuItems.Price, OrderItems.Quantity
2  FROM Orders
3  JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID
4  JOIN MenuItems ON OrderItems.ItemID = MenuItems.ItemID;
5
```

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

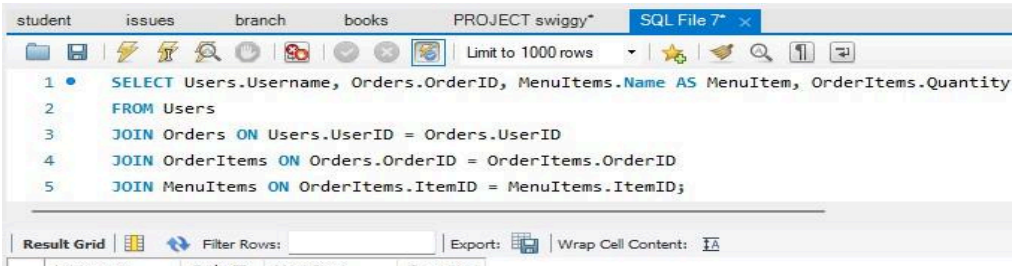
OrderID	MenuItem	Price	Quantity
1	Chicken Biryani	12.99	2
1	Butter Chicken	14.99	1
2	Paneer Tikka	8.99	1
2	Veg Pulao	9.99	2
3	Chicken Biryani	12.99	3
3	Butter Chicken	14.99	1
4	Paneer Tikka	8.99	2
4	Veg Pulao	9.99	1
5	Chicken Biryani	12.99	1
5	Butter Chicken	14.99	2
6	Paneer Tikka	8.99	1
6	Veg Pulao	9.99	1
7	Chicken Biryani	12.99	2
7	Butter Chicken	14.99	2
8	Paneer Tikka	8.99	2
8	Veg Pulao	9.99	1
9	Chicken Biryani	12.99	1
9	Butter Chicken	14.99	1
10	Paneer Tikka	8.99	3
10	Veg Pulao	9.99	1

3. Write a Query to get User Orders with Restaurant Information Retrieval Query



Username	OrderID	Restaurant	OrderDate
AdityaSharma	1	Spice Delight	2024-01-10 15:29:25
AishwaryaPatel	2	Curry House	2024-01-10 15:29:25
ArjunKumar	3	Saffron Flavors	2024-01-10 15:29:25
BhavyaSingh	4	Chaat Corner	2024-01-10 15:29:25
ChetnaMehta	5	Dosa Junction	2024-01-10 15:29:25
DeepakVerma	6	Paneer Palace	2024-01-10 15:29:25
EktaKapoor	7	Bhindi Bazaar	2024-01-10 15:29:25
GauravTiwari	8	Vada Pav Paradise	2024-01-10 15:29:25
IshitaDesai	9	Butter Chicken Bliss	2024-01-10 15:29:25
JaiSinghania	10	Chai Chokha	2024-01-10 15:29:25
AdityaSharma	11	Curry House	2024-01-10 15:29:25
AishwaryaPatel	12	Saffron Flavors	2024-01-10 15:29:25
ArjunKumar	13	Chaat Corner	2024-01-10 15:29:25
BhavyaSingh	14	Dosa Junction	2024-01-10 15:29:25
ChetnaMehta	15	Paneer Palace	2024-01-10 15:29:25
DeepakVerma	16	Bhindi Bazaar	2024-01-10 15:29:25
EktaKapoor	17	Vada Pav Paradise	2024-01-10 15:29:25
GauravTiwari	18	Butter Chicken Bliss	2024-01-10 15:29:25
IshitaDesai	19	Chai Chokha	2024-01-10 15:29:25
JaiSinghania	20	Spice Delight	2024-01-10 15:29:25

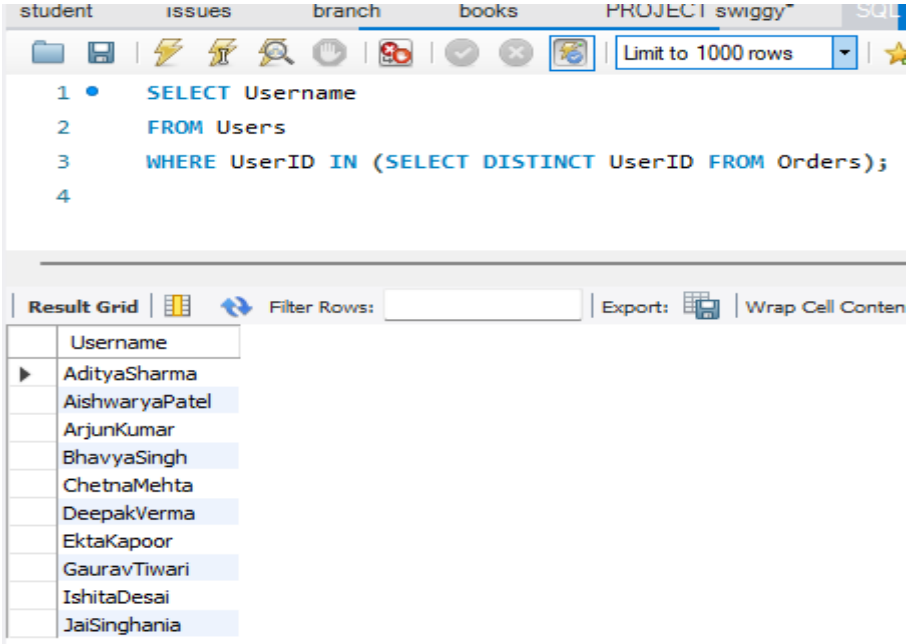
4. Write a Query to **User Order Details with Menu Item Information Retrieval Query**



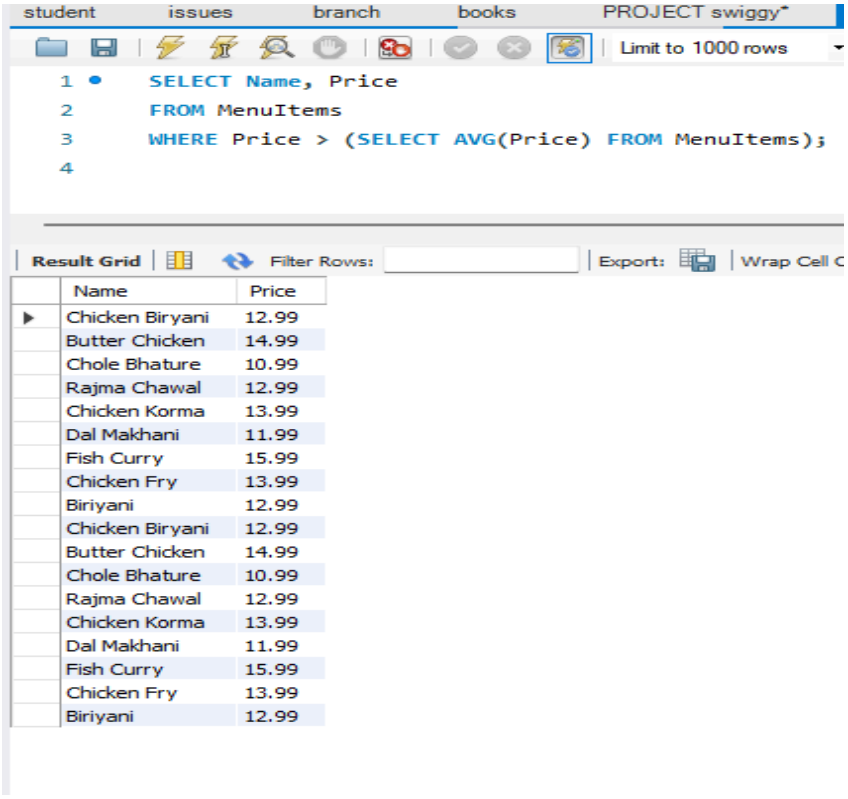
Username	OrderID	MenuItem	Quantity
AdityaSharma	1	Chicken Biryani	2
AdityaSharma	1	Butter Chicken	1
AishwaryaPatel	2	Paneer Tikka	1
AishwaryaPatel	2	Veg Pulao	2
ArjunKumar	3	Chicken Biryani	3
ArjunKumar	3	Butter Chicken	1
BhavyaSingh	4	Paneer Tikka	2
BhavyaSingh	4	Veg Pulao	1
ChetnaMehta	5	Chicken Biryani	1
ChetnaMehta	5	Butter Chicken	2
DeepakVerma	6	Paneer Tikka	1
DeepakVerma	6	Veg Pulao	1
EktaKapoor	7	Chicken Biryani	2
EktaKapoor	7	Butter Chicken	2
GauravTiwari	8	Paneer Tikka	2
GauravTiwari	8	Veg Pulao	1
IshitaDesai	9	Chicken Biryani	1
IshitaDesai	9	Butter Chicken	1
JaiSinghania	10	Paneer Tikka	3
JaiSinghania	10	Veg Pulao	1

Subquery

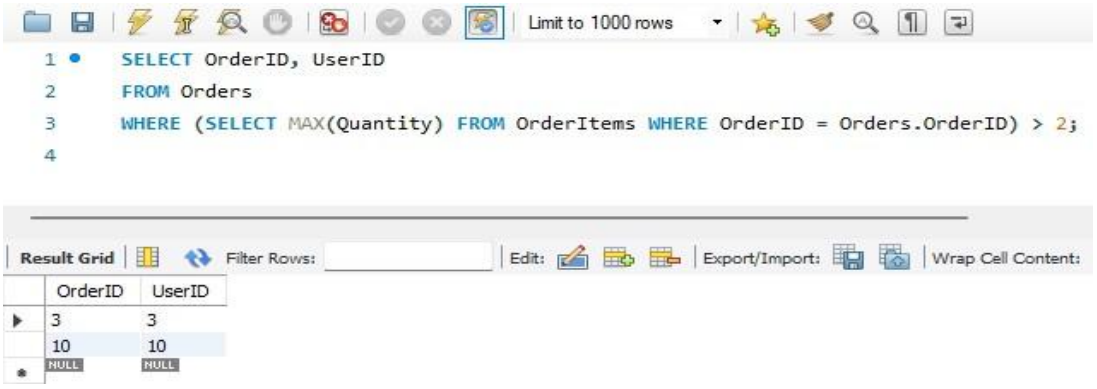
1.write a sub-query to get **Users with Orders Information Retrieval Query**



2.write a sub-query to get **Menu Items Above Average Price Retrieval**



3.write a sub-query to **Orders with Maximum Quantity Exceeding 2 Retrieval Query**

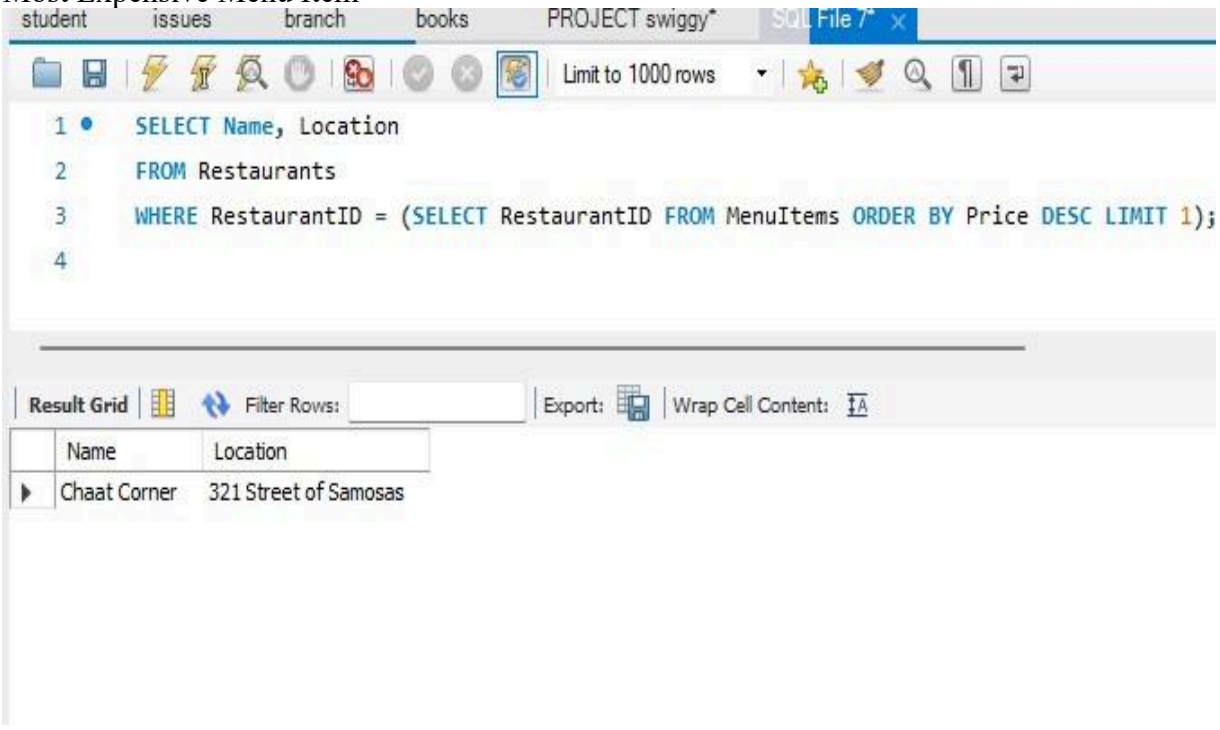


```
1 •  SELECT OrderID, UserID
2    FROM Orders
3    WHERE (SELECT MAX(Quantity) FROM OrderItems WHERE OrderID = Orders.OrderID) > 2;
4
```

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

OrderID	UserID
3	3
10	10
NULL	NULL

4. Write a sub-Query to Retrieve Name and Location of the Restaurant with the Most Expensive Menu Item



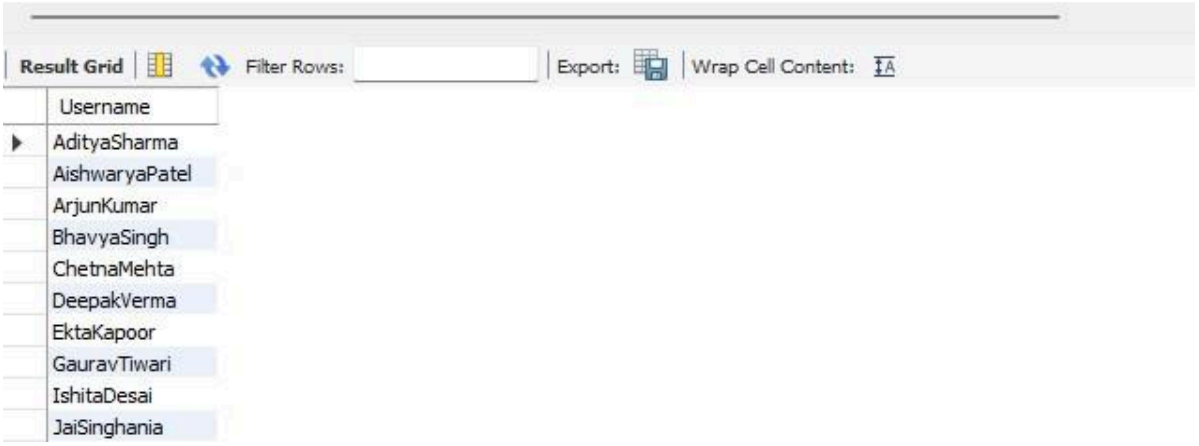
student | issues | branch | books | PROJECT swiggy* | File 7*

```
1 •  SELECT Name, Location
2    FROM Restaurants
3    WHERE RestaurantID = (SELECT RestaurantID FROM MenuItems ORDER BY Price DESC LIMIT 1);
4
```

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

Name	Location
Chaat Corner	321 Street of Samosas

5. write a subquery to Retrieve Usernames of Users with Multiple Orders



Conclusion

In conclusion, the Swiggy food delivery database project in SQL has been successfully designed and implemented to meet the requirements of a dynamic and efficient food delivery system. The project focused on key entities such as Users, Restaurants, Orders, and Menus, ensuring a comprehensive and well-organized database structure.

The primary goals of the project were to provide a seamless experience for users, efficient management for restaurants, and streamlined order processing. The SQL database schema effectively captures the relationships between different entities, facilitating data integrity and enabling smooth interactions within the system.