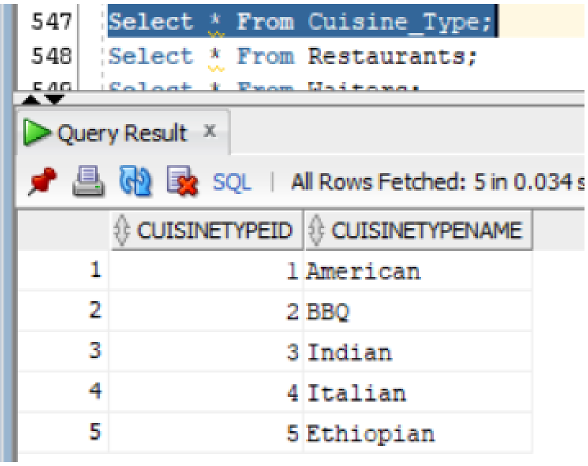
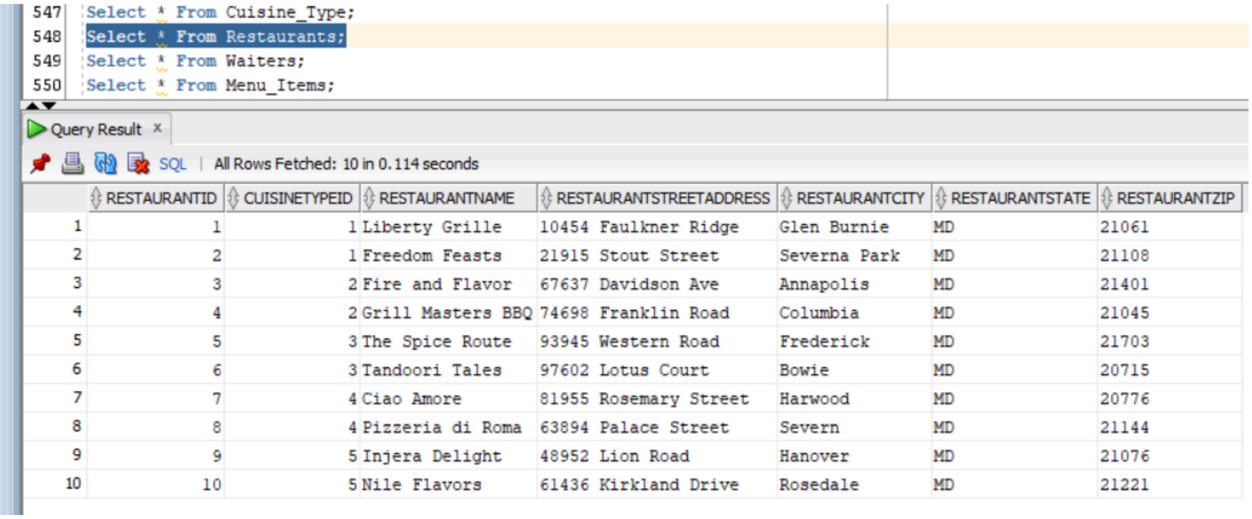
Screenshots Of Database System:

Initial Table Data Screenshots:



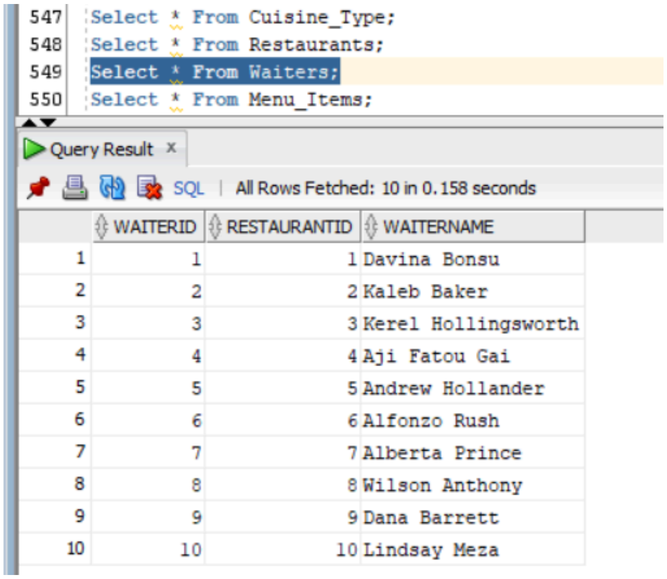
```
Select * From Cuisine_Type;
Select * From Restaurants;
Select * From Waiters;
```

	CUISINETYPEID	CUISINETYPENAME
1	1	American
2	2	BBQ
3	3	Indian
4	4	Italian
5	5	Ethiopian



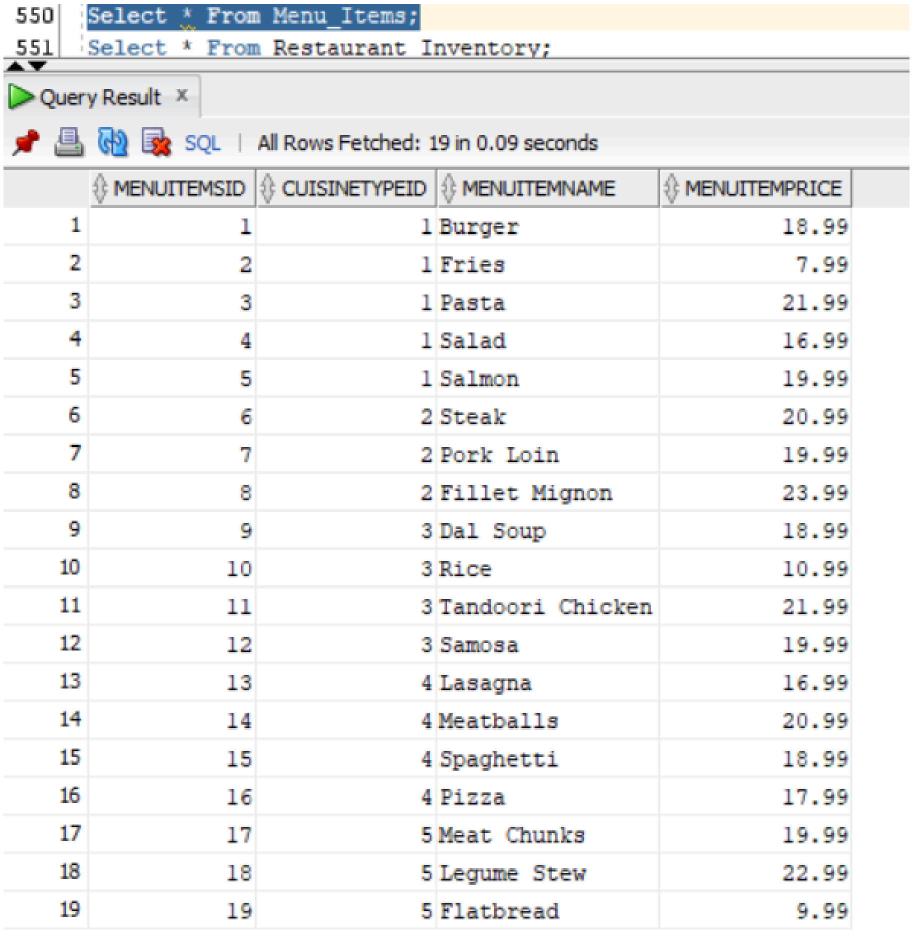
```
Select * From Cuisine_Type;
Select * From Restaurants;
Select * From Waiters;
Select * From Menu_Items;
```

	RESTAURANTID	CUISINETYPEID	RESTAURANTNAME	RESTAURANTSTREETADDRESS	RESTAURANTCITY	RESTAURANTSTATE	RESTAURANTZIP
1	1	1	Liberty Grille	10454 Faulkner Ridge	Glen Burnie	MD	21061
2	2	1	Freedom Feasts	21915 Stout Street	Severna Park	MD	21108
3	3	2	Fire and Flavor	67637 Davidson Ave	Annapolis	MD	21401
4	4	2	Grill Masters BBQ	74698 Franklin Road	Columbia	MD	21045
5	5	3	The Spice Route	93945 Western Road	Frederick	MD	21703
6	6	3	Tandoori Tales	97602 Lotus Court	Bowie	MD	20715
7	7	4	Ciao Amore	81955 Rosemary Street	Harwood	MD	20776
8	8	4	Pizzeria di Roma	63894 Palace Street	Severn	MD	21144
9	9	5	Injera Delight	48952 Lion Road	Hanover	MD	21076
10	10	5	Nile Flavors	61436 Kirkland Drive	Rosedale	MD	21221



```
Select * From Cuisine_Type;
Select * From Restaurants;
Select * From Waiters;
Select * From Menu_Items;
```

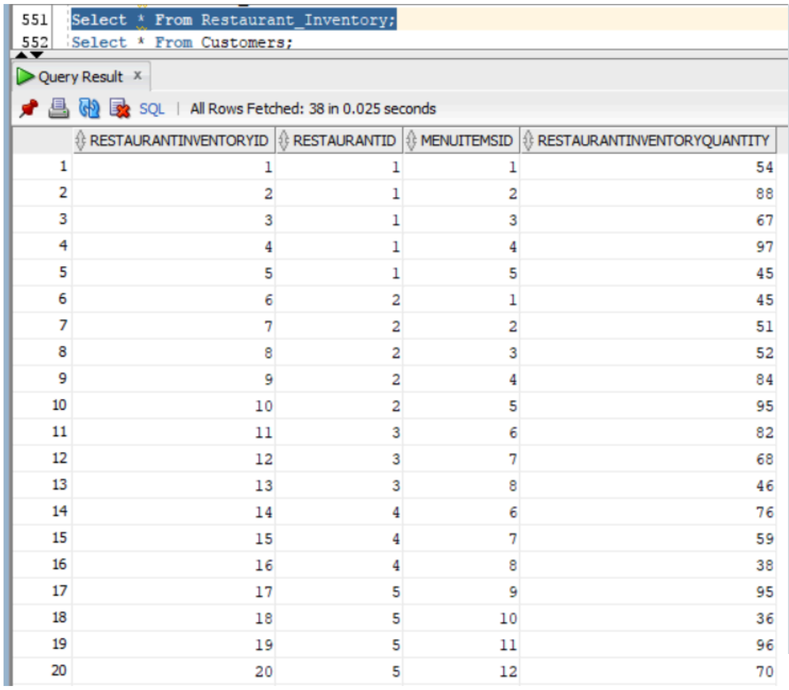
	WAITERID	RESTAURANTID	WAITERNAME
1	1	1	Davina Bonsu
2	2	2	Kaleb Baker
3	3	3	Kerel Hollingsworth
4	4	4	Aji Fatou Gai
5	5	5	Andrew Hollander
6	6	6	Alfonzo Rush
7	7	7	Alberta Prince
8	8	8	Wilson Anthony
9	9	9	Dana Barrett
10	10	10	Lindsay Meza



```
Select * From Menu_Items;
Select * From Restaurant_Inventory;
```

All Rows Fetched: 19 in 0.09 seconds

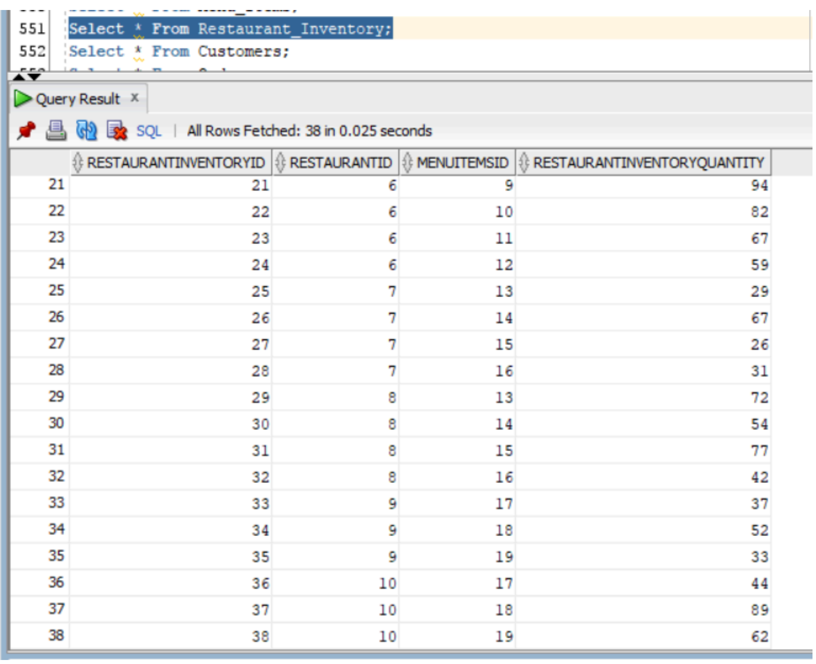
	MENUITEMSID	CUISINETYPEID	MENUITEMNAME	MENUITEMPRICE
1	1	1	Burger	18.99
2	2	1	Fries	7.99
3	3	1	Pasta	21.99
4	4	1	Salad	16.99
5	5	1	Salmon	19.99
6	6	2	Steak	20.99
7	7	2	Pork Loin	19.99
8	8	2	Fillet Mignon	23.99
9	9	3	Dal Soup	18.99
10	10	3	Rice	10.99
11	11	3	Tandoori Chicken	21.99
12	12	3	Samosa	19.99
13	13	4	Lasagna	16.99
14	14	4	Meatballs	20.99
15	15	4	Spaghetti	18.99
16	16	4	Pizza	17.99
17	17	5	Meat Chunks	19.99
18	18	5	Legume Stew	22.99
19	19	5	Flatbread	9.99



```
Select * From Restaurant_Inventory;
Select * From Customers;
```

All Rows Fetched: 38 in 0.025 seconds

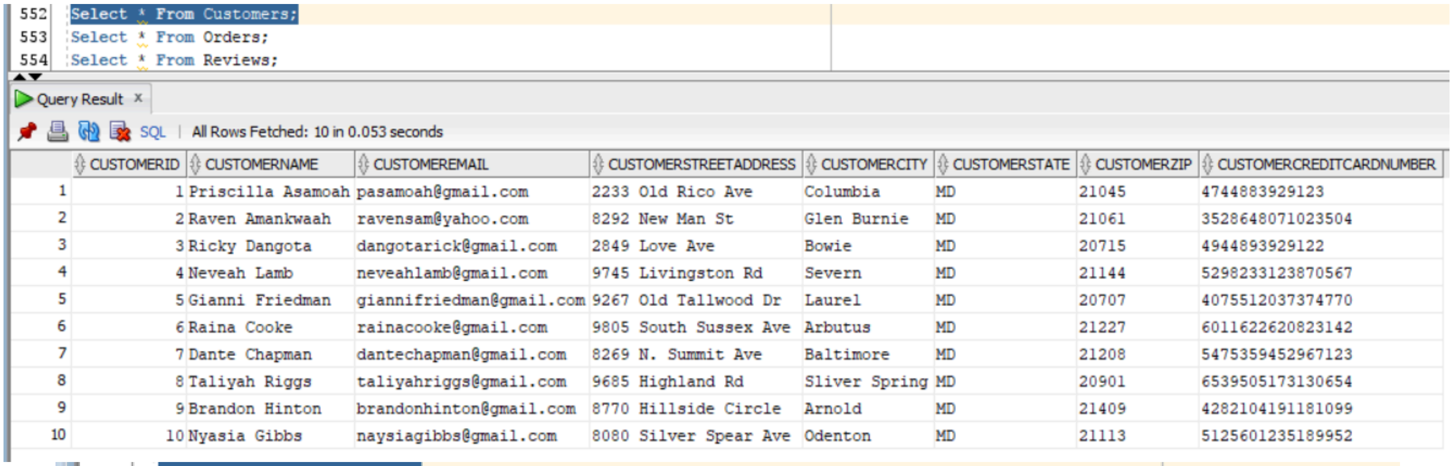
	RESTAURANTINVENTORYID	RESTAURANTID	MENUITEMSID	RESTAURANTINVENTORYQUANTITY
1	1	1	1	54
2	2	1	2	88
3	3	1	3	67
4	4	1	4	97
5	5	1	5	45
6	6	2	1	45
7	7	2	2	51
8	8	2	3	52
9	9	2	4	84
10	10	2	5	95
11	11	3	6	82
12	12	3	7	68
13	13	3	8	46
14	14	4	6	76
15	15	4	7	59
16	16	4	8	38
17	17	5	9	95
18	18	5	10	36
19	19	5	11	96
20	20	5	12	70



```
Select * From Restaurant_Inventory;
Select * From Customers;
```

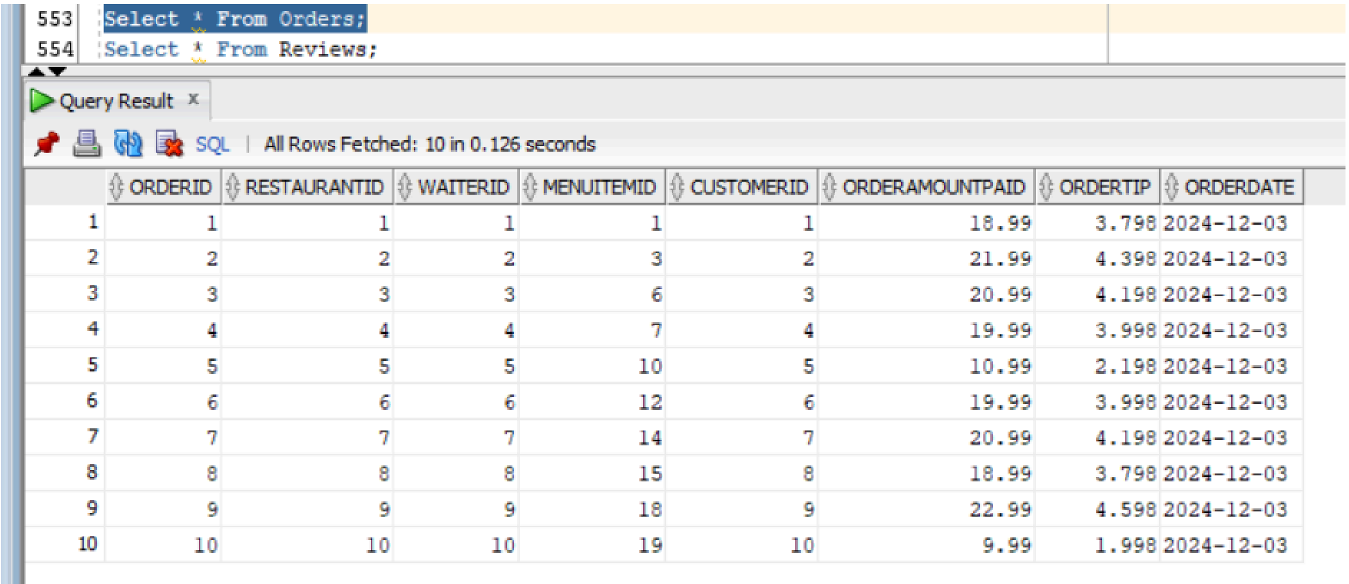
All Rows Fetched: 38 in 0.025 seconds

	RESTAURANTINVENTORYID	RESTAURANTID	MENUITEMSID	RESTAURANTINVENTORYQUANTITY
21	21	6	9	94
22	22	6	10	82
23	23	6	11	67
24	24	6	12	59
25	25	7	13	29
26	26	7	14	67
27	27	7	15	26
28	28	7	16	31
29	29	8	13	72
30	30	8	14	54
31	31	8	15	77
32	32	8	16	42
33	33	9	17	37
34	34	9	18	52
35	35	9	19	33
36	36	10	17	44
37	37	10	18	89
38	38	10	19	62



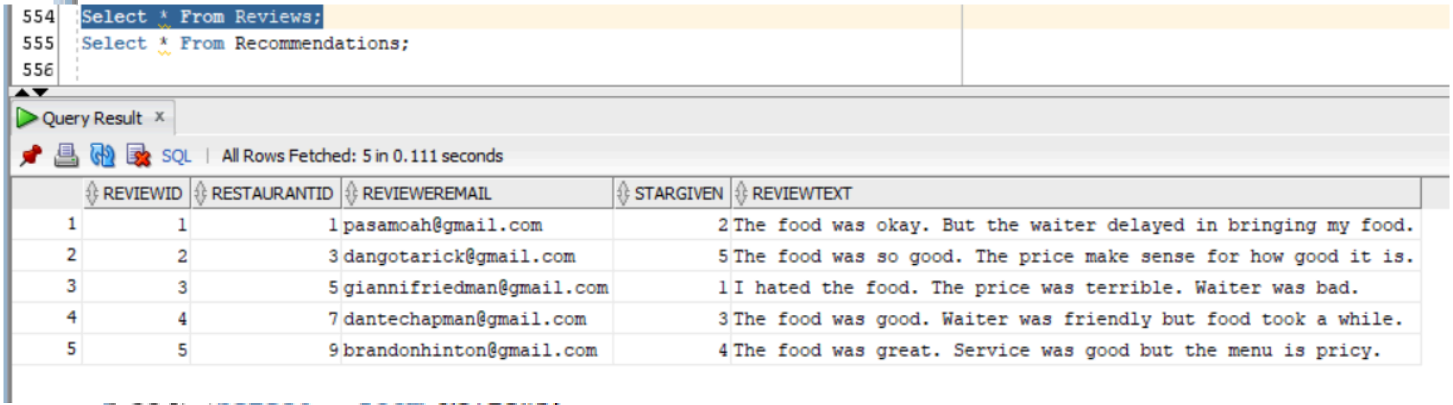
```
Select * From Customers;
Select * From Orders;
Select * From Reviews;
```

	CUSTOMERID	CUSTOMERNAME	CUSTOMEREMAIL	CUSTOMERSTREETADDRESS	CUSTOMERCITY	CUSTOMERSTATE	CUSTOMERZIP	CUSTOMERCREDITCARDNUMBER
1	1	Priscilla Asamoah	pasamoah@gmail.com	2233 Old Rico Ave	Columbia	MD	21045	4744883929123
2	2	Raven Amankwaah	ravensam@yahoo.com	8292 New Man St	Glen Burnie	MD	21061	3528648071023504
3	3	Ricky Dangota	dangotarick@gmail.com	2849 Love Ave	Bowie	MD	20715	4944893929122
4	4	Neveah Lamb	neveahlamb@gmail.com	9745 Livingston Rd	Severn	MD	21144	5298233123870567
5	5	Gianni Friedman	giannifriedman@gmail.com	9267 Old Tallwood Dr	Laurel	MD	20707	4075512037374770
6	6	Raina Cooke	rainacooke@gmail.com	9805 South Sussex Ave	Arbutus	MD	21227	6011622620823142
7	7	Dante Chapman	dantechapman@gmail.com	8269 N. Summit Ave	Baltimore	MD	21208	5475359452967123
8	8	Taliyah Riggs	taliyahriggs@gmail.com	9685 Highland Rd	Sliver Spring	MD	20901	6539505173130654
9	9	Brandon Hinton	brandonhinton@gmail.com	8770 Hillside Circle	Arnold	MD	21409	4282104191181099
10	10	Nyasia Gibbs	naysiagibbs@gmail.com	8080 Silver Spear Ave	Odenton	MD	21113	5125601235189952



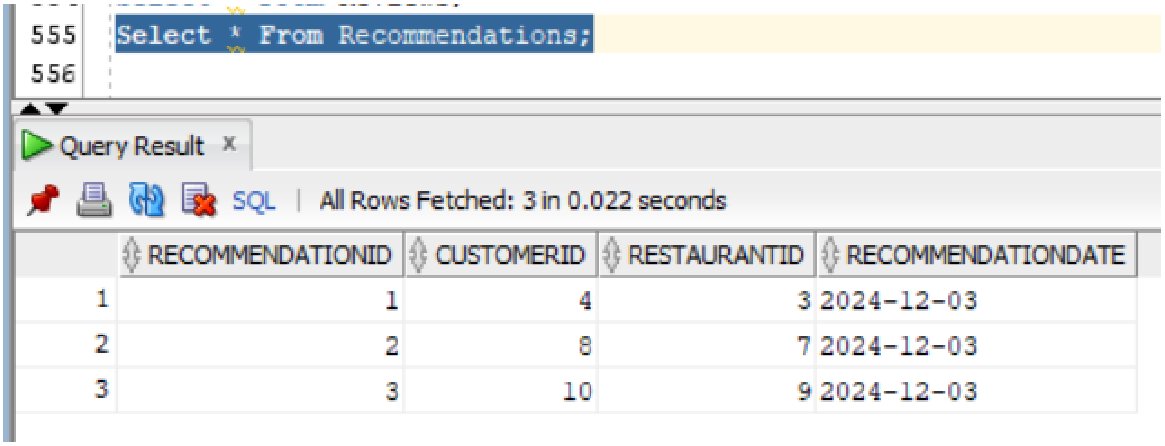
```
Select * From Orders;
Select * From Reviews;
```

	ORDERID	RESTAURANTID	WAITERID	MENUITEMID	CUSTOMERID	ORDERAMOUNTPAID	ORDERTIP	ORDERDATE
1	1	1	1	1	1	18.99	3.798	2024-12-03
2	2	2	2	3	2	21.99	4.398	2024-12-03
3	3	3	3	6	3	20.99	4.198	2024-12-03
4	4	4	4	7	4	19.99	3.998	2024-12-03
5	5	5	5	10	5	10.99	2.198	2024-12-03
6	6	6	6	12	6	19.99	3.998	2024-12-03
7	7	7	7	14	7	20.99	4.198	2024-12-03
8	8	8	8	15	8	18.99	3.798	2024-12-03
9	9	9	9	18	9	22.99	4.598	2024-12-03
10	10	10	10	19	10	9.99	1.998	2024-12-03



```
Select * From Reviews;
Select * From Recommendations;
```

	REVIEWID	RESTAURANTID	REVIEWEREMAIL	STARGIVEN	REVIEWTEXT
1	1	1	pasamoah@gmail.com	2	The food was okay. But the waiter delayed in bringing my food.
2	2	3	dangotarick@gmail.com	5	The food was so good. The price make sense for how good it is.
3	3	5	giannifriedman@gmail.com	1	I hated the food. The price was terrible. Waiter was bad.
4	4	7	dantechapman@gmail.com	3	The food was good. Waiter was friendly but food took a while.
5	5	9	brandonhinton@gmail.com	4	The food was great. Service was good but the menu is pricy.



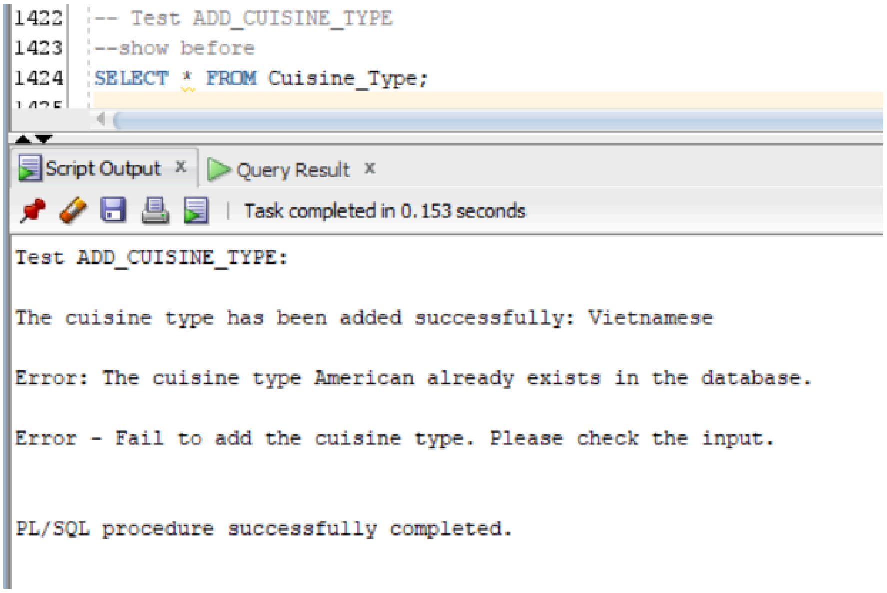
```
Select * From Recommendations;
```

	RECOMMENDATIONID	CUSTOMERID	RESTAURANTID	RECOMMENDATIONDATE
1	1	4	3	2024-12-03
2	2	8	7	2024-12-03
3	3	10	9	2024-12-03

Screenshots of Procedure Test Scripts Output:

Member 1:

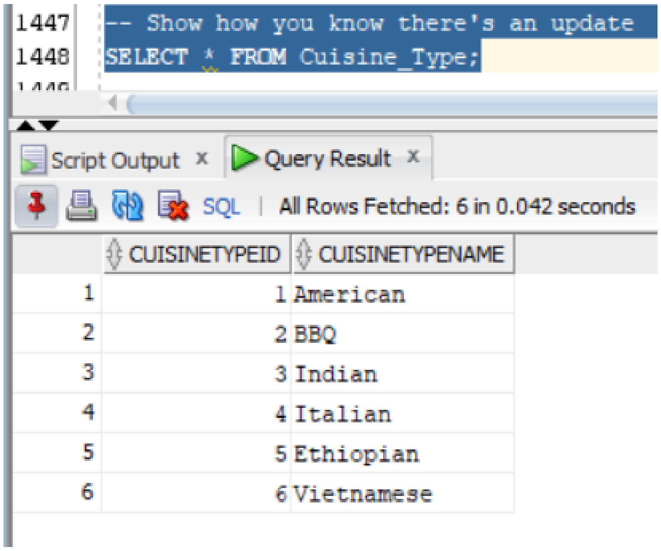
Add_Cuisine_Type:



```
-- Test ADD_CUISINE_TYPE
--show before
SELECT * FROM Cuisine_Type;
```

Task completed in 0.153 seconds

```
Test ADD_CUISINE_TYPE:

The cuisine type has been added successfully: Vietnamese

Error: The cuisine type American already exists in the database.

Error - Fail to add the cuisine type. Please check the input.


PL/SQL procedure successfully completed.
```

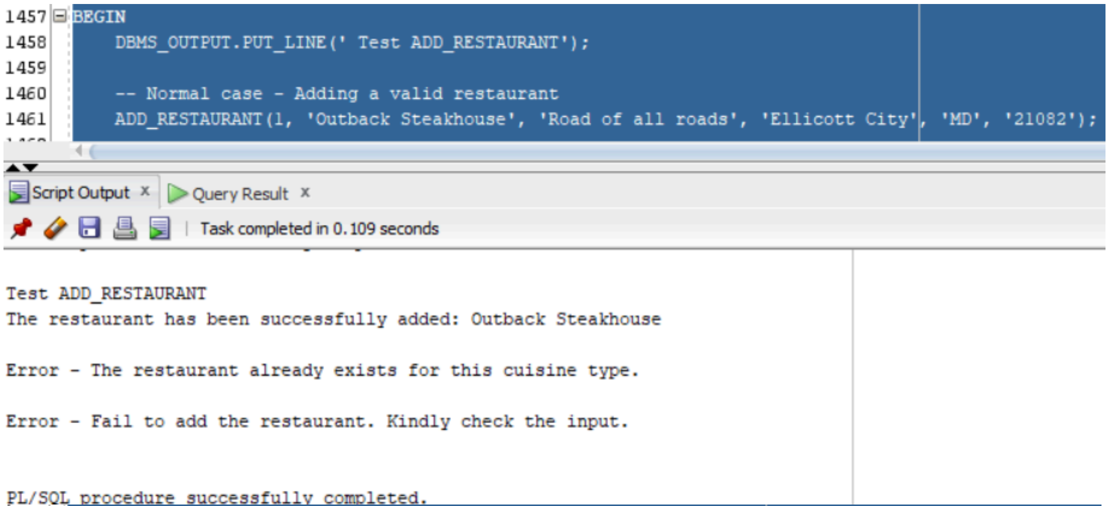


```
-- Show how you know there's an update
SELECT * FROM Cuisine_Type;
```

All Rows Fetched: 6 in 0.042 seconds

	CUISINETYPEID	CUISINETYPENAME
1	1	American
2	2	BBQ
3	3	Indian
4	4	Italian
5	5	Ethiopian
6	6	Vietnamese

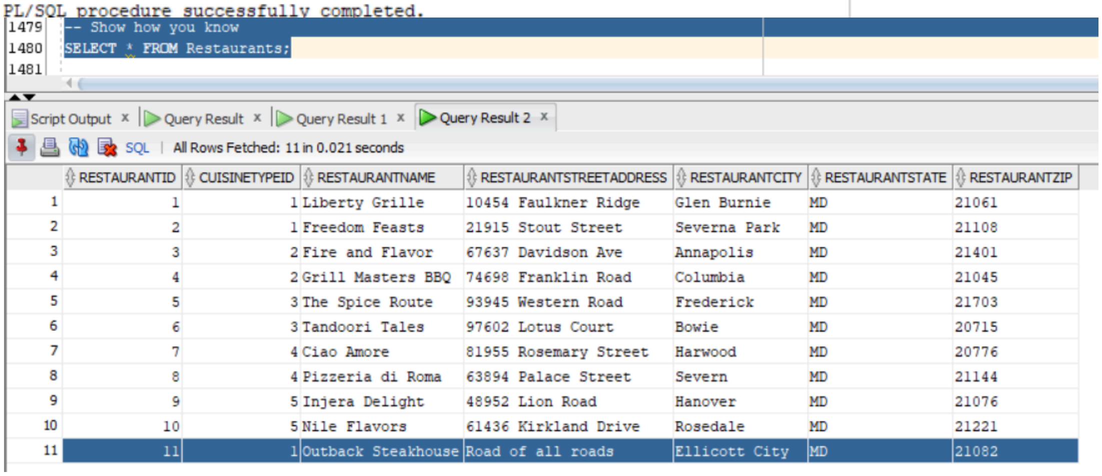
Add_Restaurant:



```
BEGIN
    DBMS_OUTPUT.PUT_LINE(' Test ADD_RESTAURANT');

    -- Normal case - Adding a valid restaurant
    ADD_RESTAURANT(1, 'Outback Steakhouse', 'Road of all roads', 'Ellicott City', 'MD', '21082');
```

Task completed in 0.109 seconds

```
Test ADD_RESTAURANT
The restaurant has been successfully added: Outback Steakhouse

Error - The restaurant already exists for this cuisine type.

Error - Fail to add the restaurant. Kindly check the input.


PL/SQL procedure successfully completed.
```

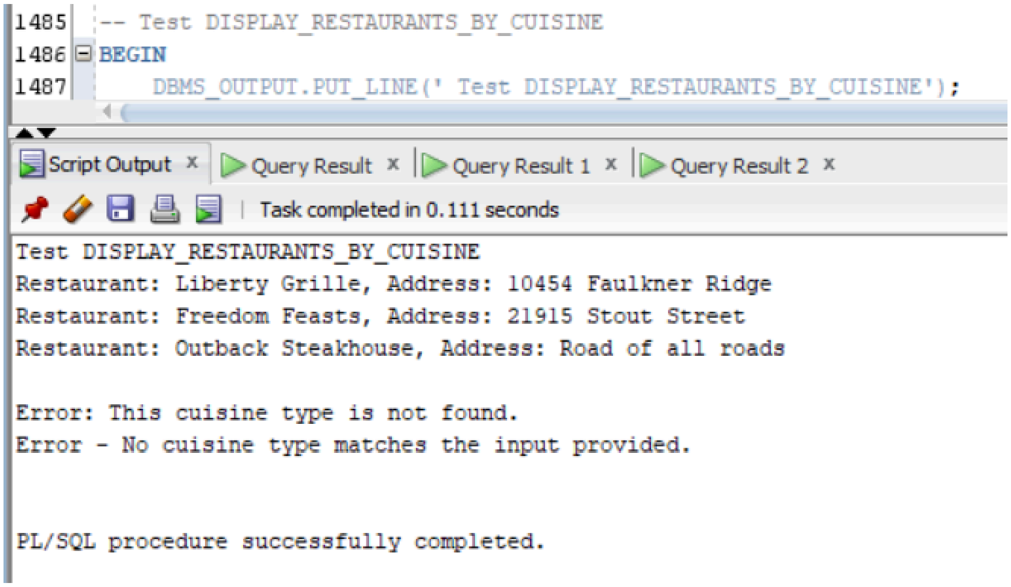


```
-- Show how you know
SELECT * FROM Restaurants;
```

All Rows Fetched: 11 in 0.021 seconds

	RESTAURANTID	CUISINETYPEID	RESTAURANTNAME	RESTAURANTSTREETADDRESS	RESTAURANTCITY	RESTAURANTSTATE	RESTAURANTZIP
1	1	1	Liberty Grille	10454 Faulkner Ridge	Glen Burnie	MD	21061
2	2	1	Freedom Feasts	21915 Stout Street	Severna Park	MD	21108
3	3	2	Fire and Flavor	67637 Davidson Ave	Annapolis	MD	21401
4	4	2	Grill Masters BBQ	74698 Franklin Road	Columbia	MD	21045
5	5	3	The Spice Route	93945 Western Road	Frederick	MD	21703
6	6	3	Tandoori Tales	97602 Lotus Court	Bowie	MD	20715
7	7	4	Ciao Amore	81955 Rosemary Street	Harwood	MD	20776
8	8	4	Pizzeria di Roma	63894 Palace Street	Severn	MD	21144
9	9	5	Injera Delight	48952 Lion Road	Hanover	MD	21076
10	10	5	Nile Flavors	61436 Kirkland Drive	Rosedale	MD	21221
11	11	1	Outback Steakhouse	Road of all roads	Ellicott City	MD	21082

Display_Restaurants_By_Cuisine:

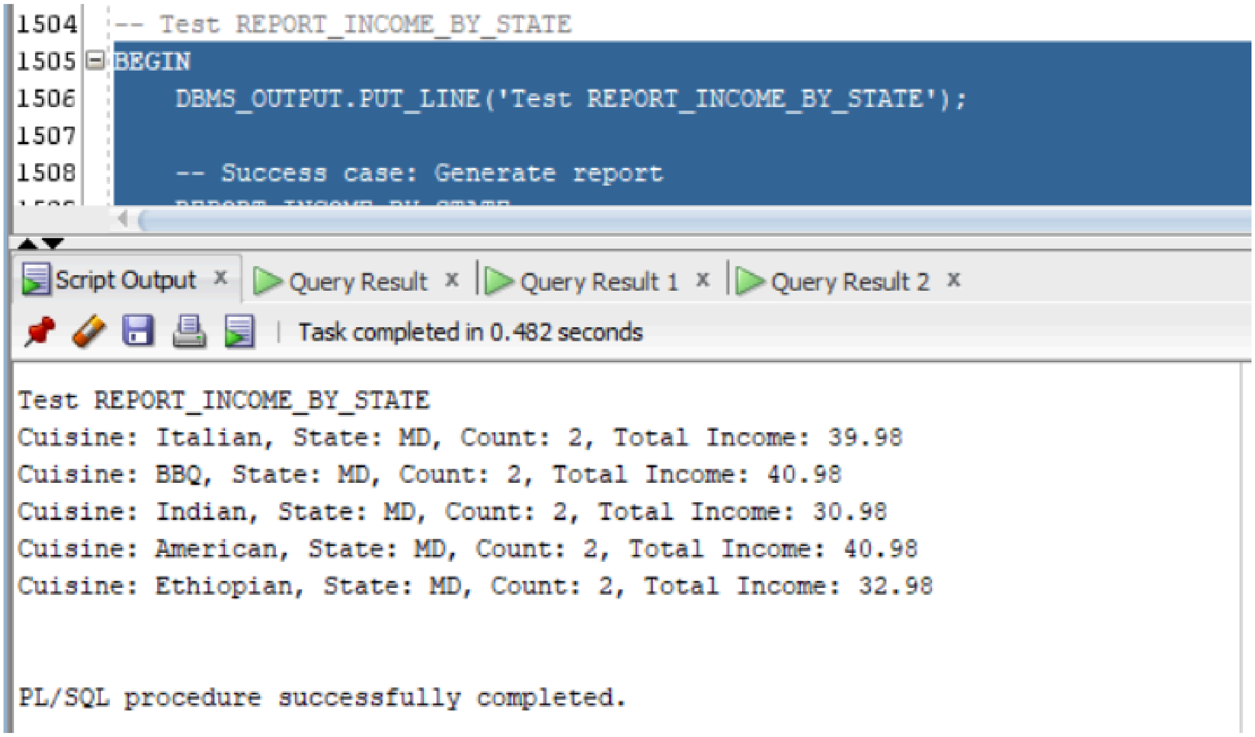


```
-- Test DISPLAY_RESTAURANTS_BY_CUISINE
BEGIN
    DBMS_OUTPUT.PUT_LINE(' Test DISPLAY_RESTAURANTS_BY_CUISINE');
```

Task completed in 0.111 seconds

```
Test DISPLAY_RESTAURANTS_BY_CUISINE
Restaurant: Liberty Grille, Address: 10454 Faulkner Ridge
Restaurant: Freedom Feasts, Address: 21915 Stout Street
Restaurant: Outback Steakhouse, Address: Road of all roads

Error: This cuisine type is not found.
Error - No cuisine type matches the input provided.


PL/SQL procedure successfully completed.
```

Report_Income_By_State:



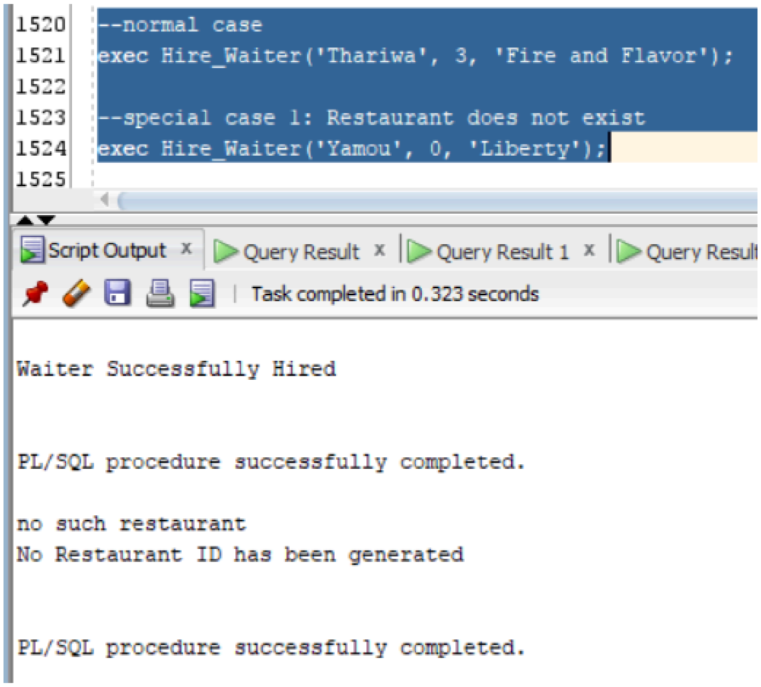
```
-- Test REPORT_INCOME_BY_STATE
BEGIN
    DBMS_OUTPUT.PUT_LINE('Test REPORT_INCOME_BY_STATE');

    -- Success case: Generate report
    REPORT_INCOME_BY_STATE
```

Task completed in 0.482 seconds

```
Test REPORT_INCOME_BY_STATE
Cuisine: Italian, State: MD, Count: 2, Total Income: 39.98
Cuisine: BBQ, State: MD, Count: 2, Total Income: 40.98
Cuisine: Indian, State: MD, Count: 2, Total Income: 30.98
Cuisine: American, State: MD, Count: 2, Total Income: 40.98
Cuisine: Ethiopian, State: MD, Count: 2, Total Income: 32.98


PL/SQL procedure successfully completed.
```

Member 2:

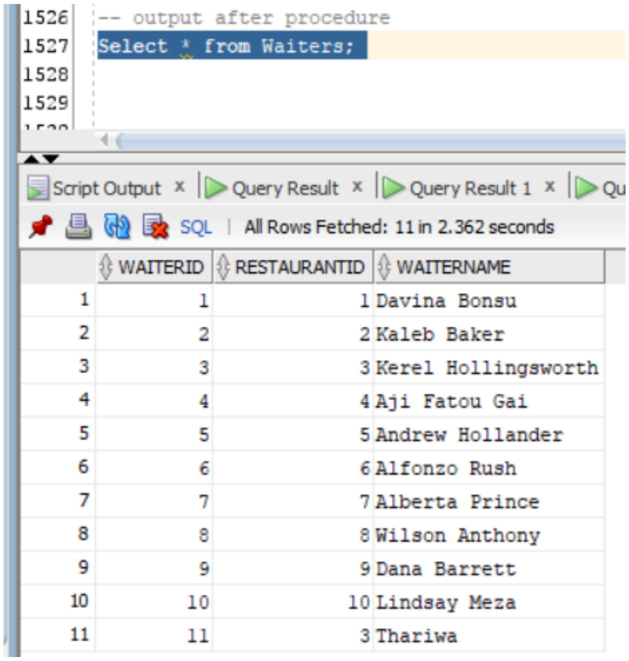
Hire_Waiter:



```
--normal case
exec Hire_Waiter('Thariwa', 3, 'Fire and Flavor');

--special case 1: Restaurant does not exist
exec Hire_Waiter('Yamou', 0, 'Liberty');
```

Task completed in 0.323 seconds

```
Waiter Successfully Hired


PL/SQL procedure successfully completed.

no such restaurant
No Restaurant ID has been generated


PL/SQL procedure successfully completed.
```

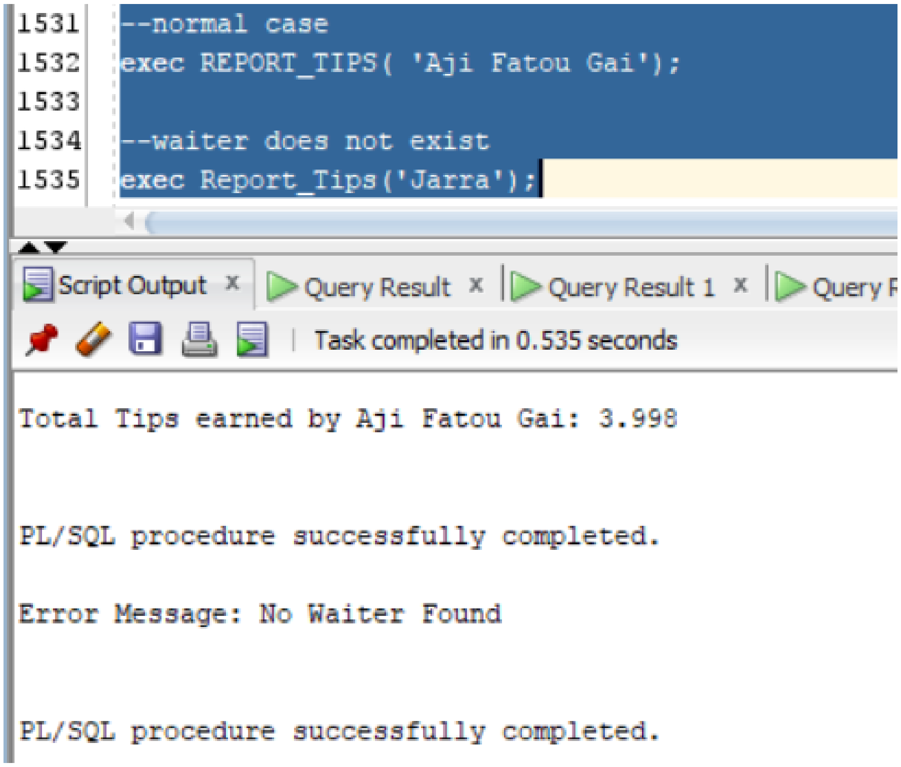


```
-- output after procedure
Select * from Waiters;
```

All Rows Fetched: 11 in 2.362 seconds

	WAITERID	RESTAURANTID	WAITERNAME
1	1	1	Davina Bonsu
2	2	2	Kaleb Baker
3	3	3	Kerel Hollingsworth
4	4	4	Aji Fatou Gai
5	5	5	Andrew Hollander
6	6	6	Alfonzo Rush
7	7	7	Alberta Prince
8	8	8	Wilson Anthony
9	9	9	Dana Barrett
10	10	10	Lindsay Meza
11	11	3	Thariwa

Report_Tips:

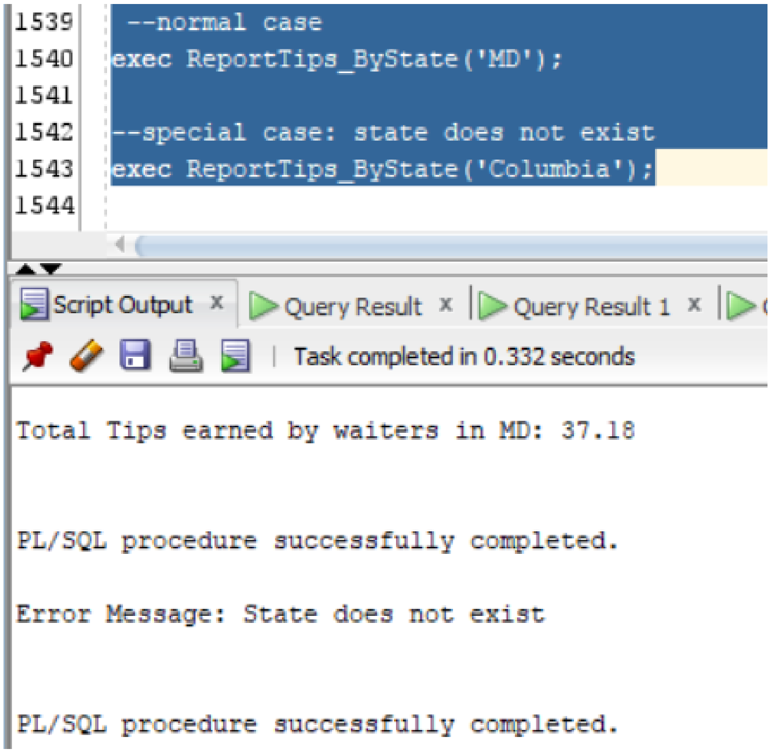


```
--normal case
exec REPORT_TIPS( 'Aji Fatou Gai');

--waiter does not exist
exec Report_Tips('Jarra');
```

Task completed in 0.535 seconds

```
Total Tips earned by Aji Fatou Gai: 3.998


PL/SQL procedure successfully completed.

Error Message: No Waiter Found


PL/SQL procedure successfully completed.
```

ReportTips_By_State:

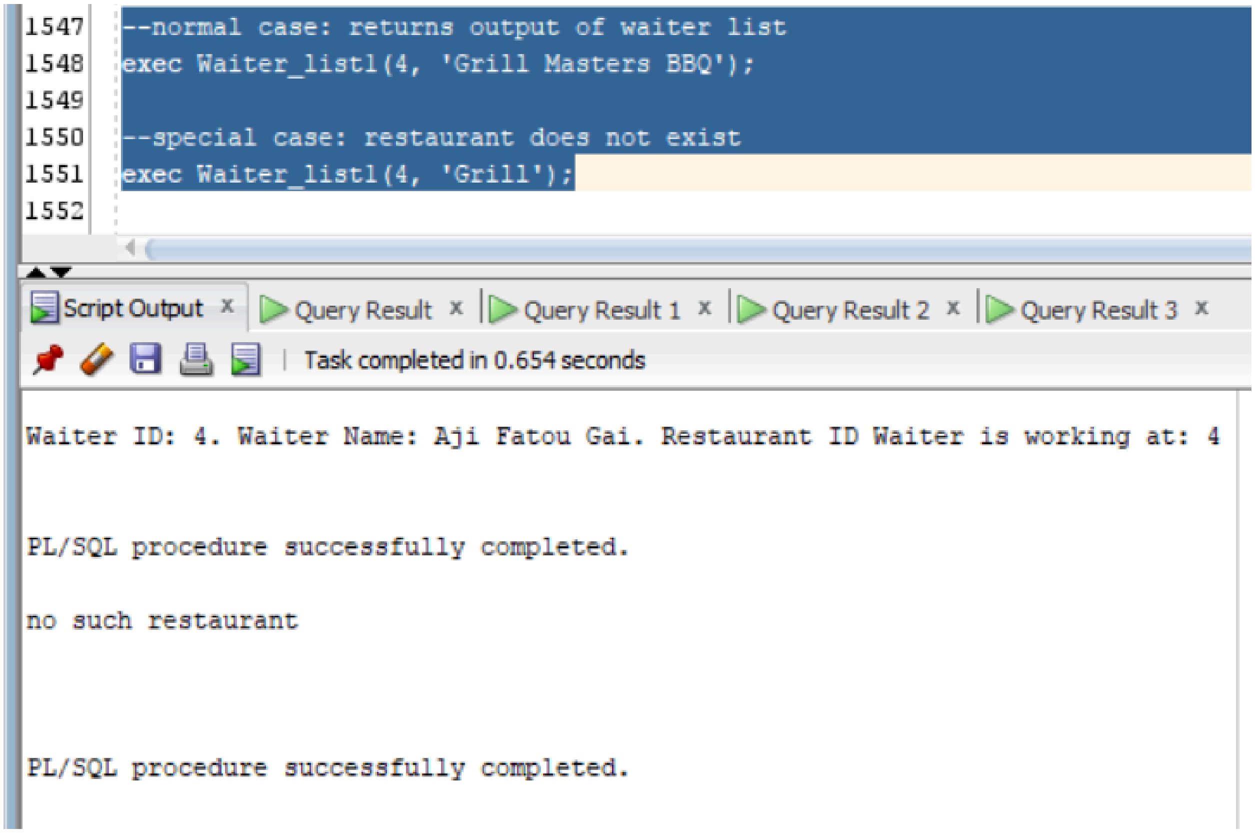


```
 --normal case
exec ReportTips_ByState('MD');

--special case: state does not exist
exec ReportTips_ByState('Columbia');
```

Task completed in 0.332 seconds

```
Total Tips earned by waiters in MD: 37.18


PL/SQL procedure successfully completed.

Error Message: State does not exist


PL/SQL procedure successfully completed.
```

Waiter_List



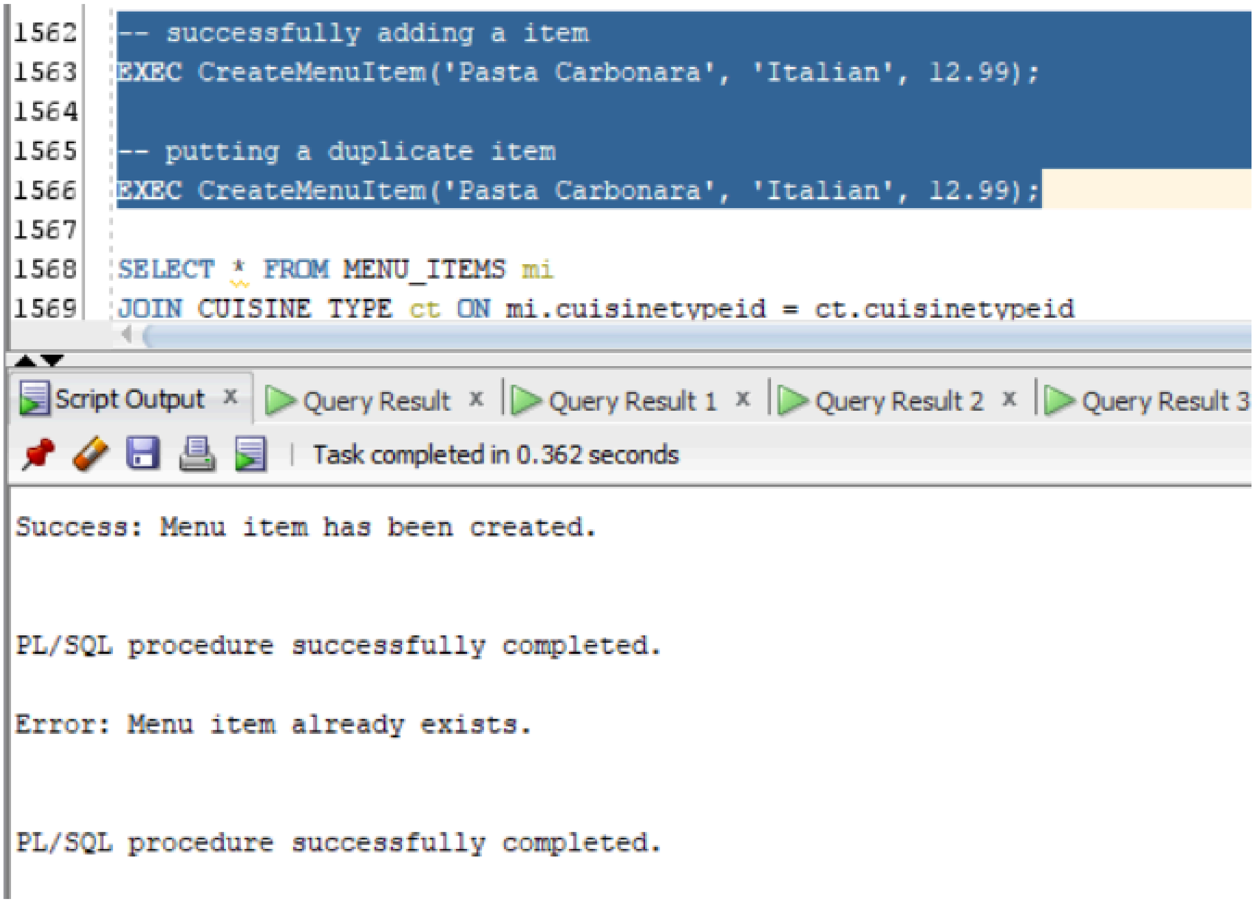
```
--normal case: returns output of waiter list
exec Waiter_listl(4, 'Grill Masters BBQ');

--special case: restaurant does not exist
exec Waiter_listl(4, 'Grill');
```

Task completed in 0.654 seconds

```
Waiter ID: 4. Waiter Name: Aji Fatou Gai. Restaurant ID Waiter is working at: 4


PL/SQL procedure successfully completed.

no such restaurant



PL/SQL procedure successfully completed.
```

Member 3:

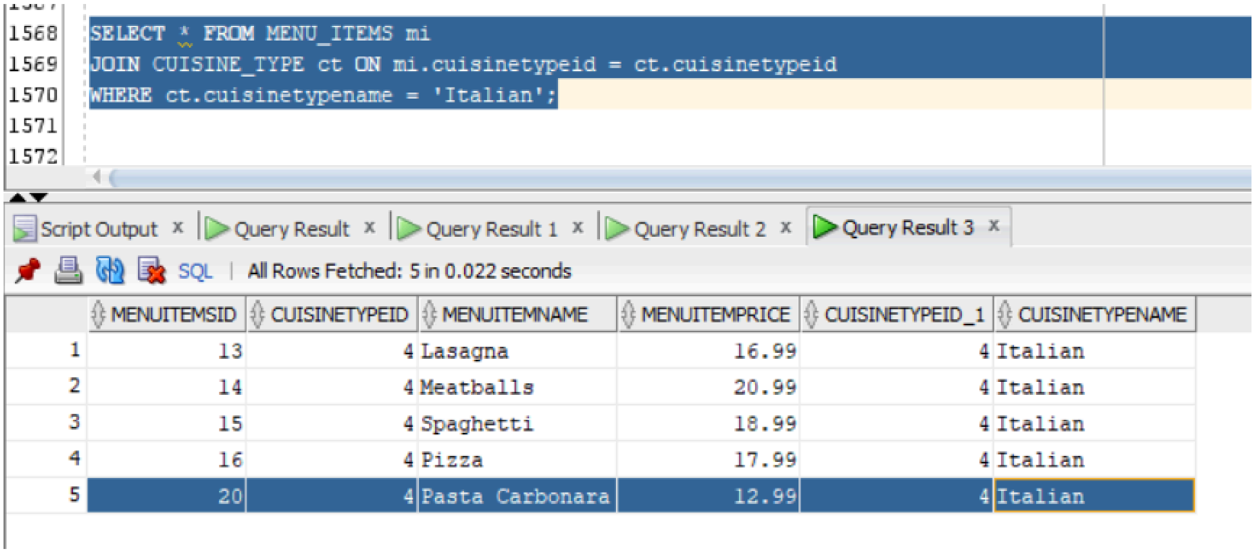
CreateMenuItem:



```
-- successfully adding a item
EXEC CreateMenuItem('Pasta Carbonara', 'Italian', 12.99);

-- putting a duplicate item
EXEC CreateMenuItem('Pasta Carbonara', 'Italian', 12.99);

SELECT * FROM MENU_ITEMS mi
JOIN CUISINE_TYPE ct ON mi.cuisinetypeid = ct.cuisinetypeid
```

Task completed in 0.362 seconds

```
Success: Menu item has been created.


PL/SQL procedure successfully completed.

Error: Menu item already exists.


PL/SQL procedure successfully completed.
```

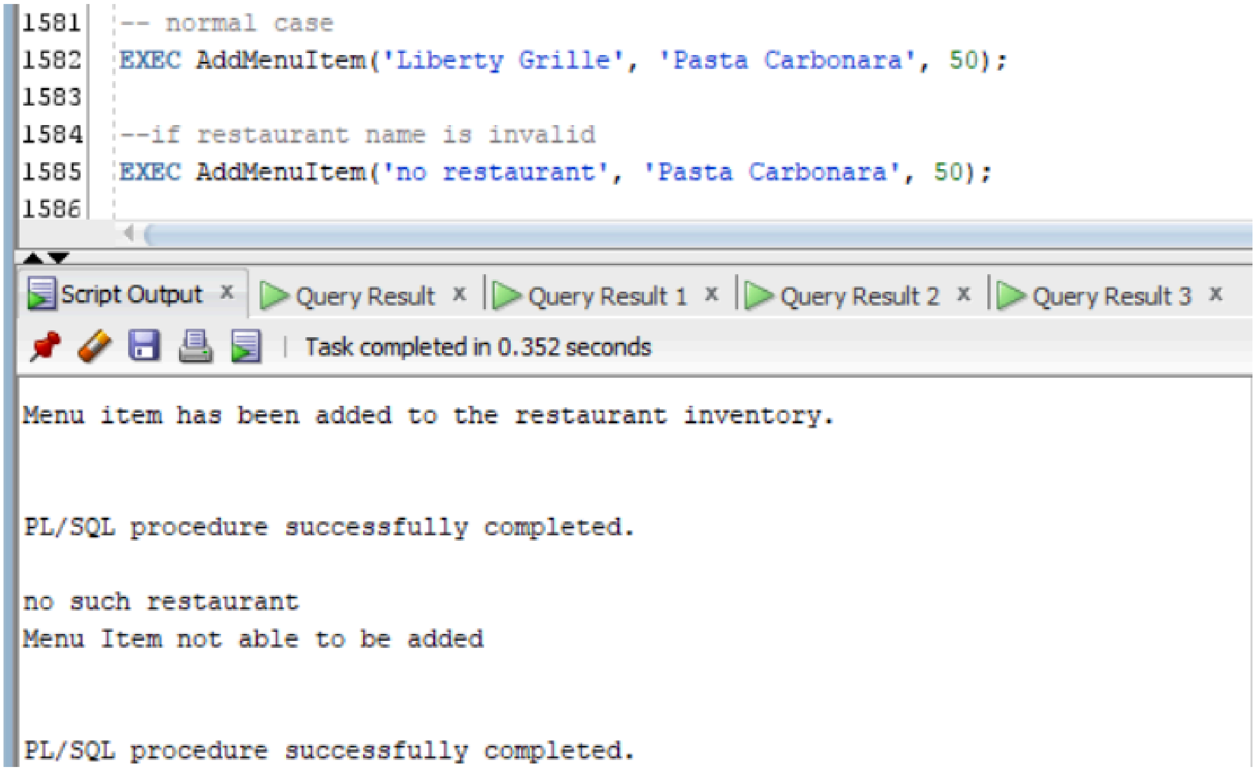


```
SELECT * FROM MENU_ITEMS mi
JOIN CUISINE_TYPE ct ON mi.cuisinetypeid = ct.cuisinetypeid
WHERE ct.cuisinetypename = 'Italian';
```

All Rows Fetched: 5 in 0.022 seconds

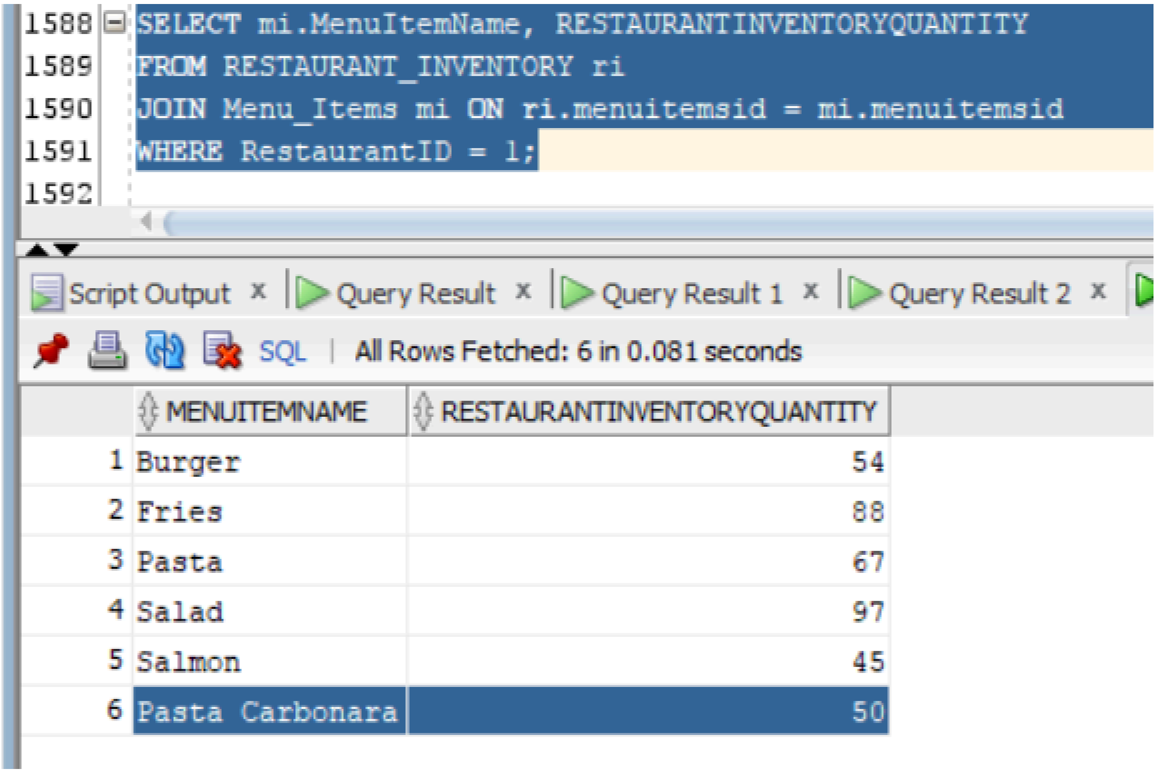
	MENUITEMSID	CUISINETYPEID	MENUITEMNAME	MENUITEMPRICE	CUISINETYPEID_1	CUISINETYPENAME
1	13	4	Lasagna	16.99	4	Italian
2	14	4	Meatballs	20.99	4	Italian
3	15	4	Spaghetti	18.99	4	Italian
4	16	4	Pizza	17.99	4	Italian
5	20	4	Pasta Carbonara	12.99	4	Italian

AddMenuItem:



```
-- normal case
EXEC AddMenuItem('Liberty Grille', 'Pasta Carbonara', 50);

--if restaurant name is invalid
EXEC AddMenuItem('no restaurant', 'Pasta Carbonara', 50);

```

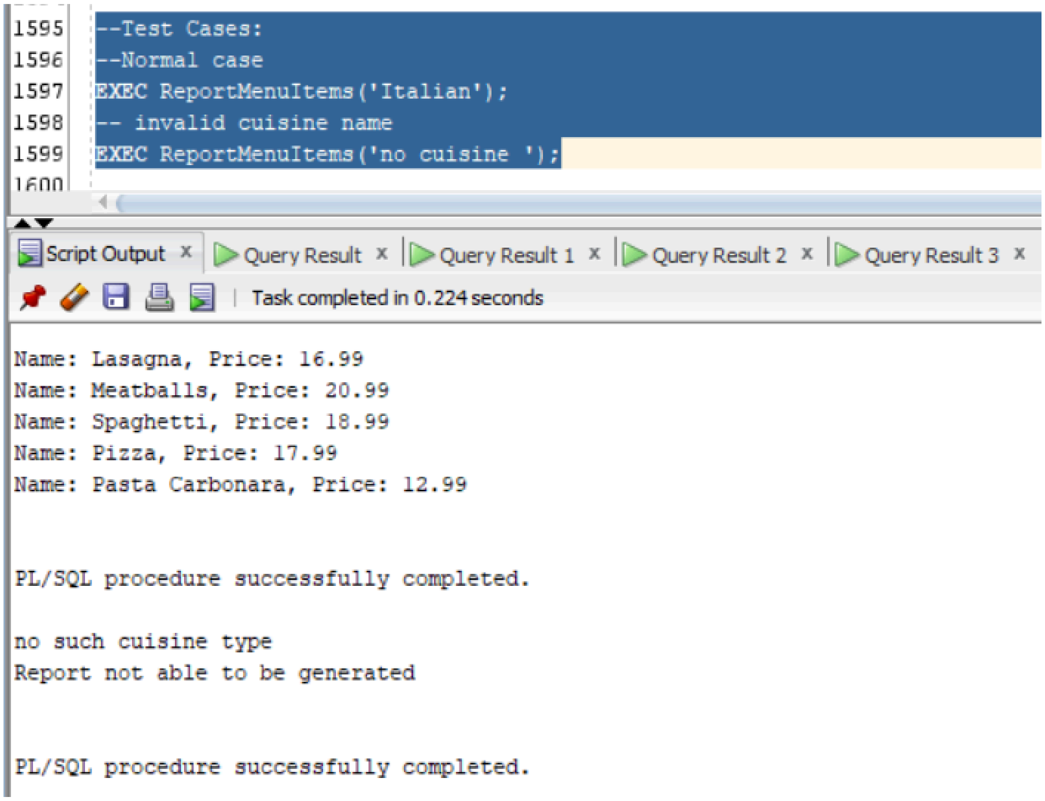
```
Menu item has been added to the restaurant inventory.


PL/SQL procedure successfully completed.

no such restaurant
Menu Item not able to be added


PL/SQL procedure successfully completed.
```



```
SELECT mi.MenuItemName, RESTAURANTINVENTORYQUANTITY
FROM RESTAURANT_INVENTORY ri
JOIN Menu_Items mi ON ri.menuitemsid = mi.menuitemsid
WHERE RestaurantID = 1;

```

	MENUITEMNAME	RESTAURANTINVENTORYQUANTITY
1	Burger	54
2	Fries	88
3	Pasta	67
4	Salad	97
5	Salmon	45
6	Pasta Carbonara	50

ReportMenuItems:

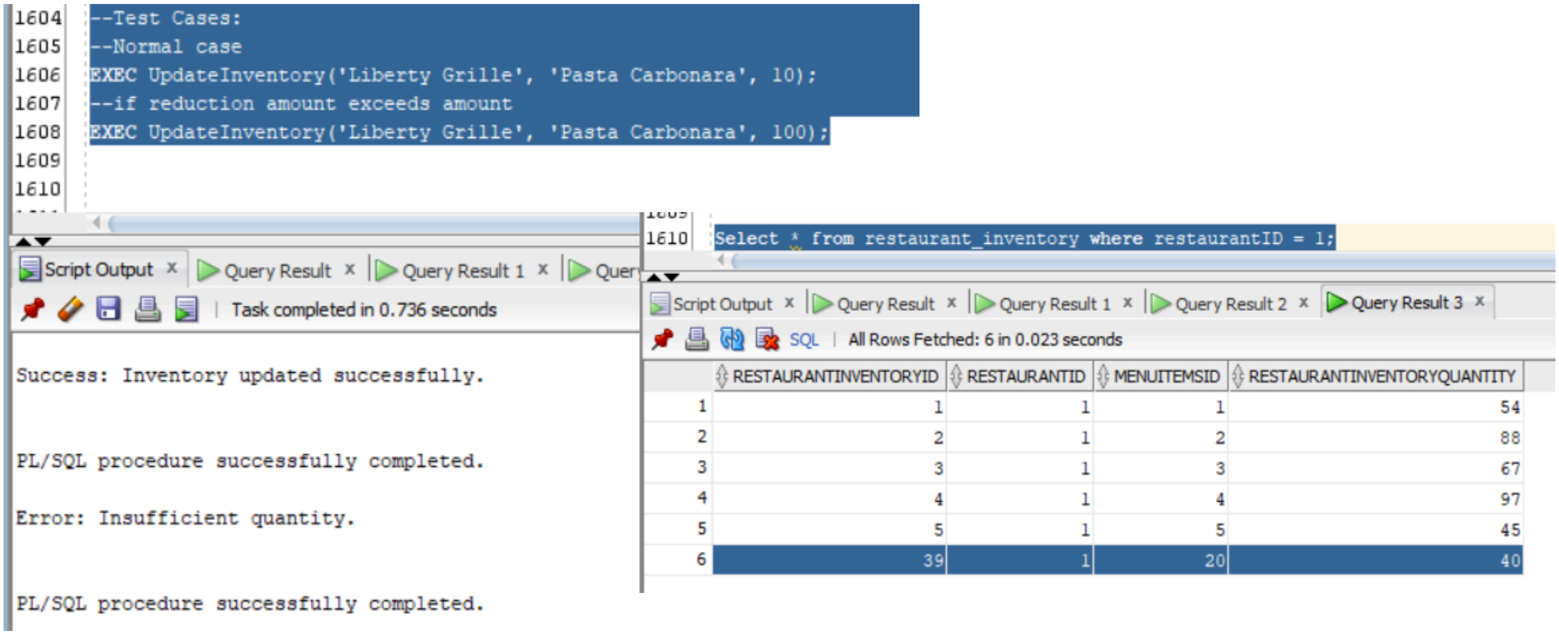


```
--Test Cases:
--Normal case
EXEC ReportMenuItems('Italian');
-- invalid cuisine name
EXEC ReportMenuItems('no cuisine ');
```

Task completed in 0.224 seconds

```
Name: Lasagna, Price: 16.99
Name: Meatballs, Price: 20.99
Name: Spaghetti, Price: 18.99
Name: Pizza, Price: 17.99
Name: Pasta Carbonara, Price: 12.99


PL/SQL procedure successfully completed.

no such cuisine type
Report not able to be generated


PL/SQL procedure successfully completed.
```

UpdateInventory:



```
--Test Cases:
--Normal case
EXEC UpdateInventory('Liberty Grille', 'Pasta Carbonara', 10);
--if reduction amount exceeds amount
EXEC UpdateInventory('Liberty Grille', 'Pasta Carbonara', 100);
```

Task completed in 0.736 seconds

```
Success: Inventory updated successfully.


PL/SQL procedure successfully completed.

Error: Insufficient quantity.


PL/SQL procedure successfully completed.
```

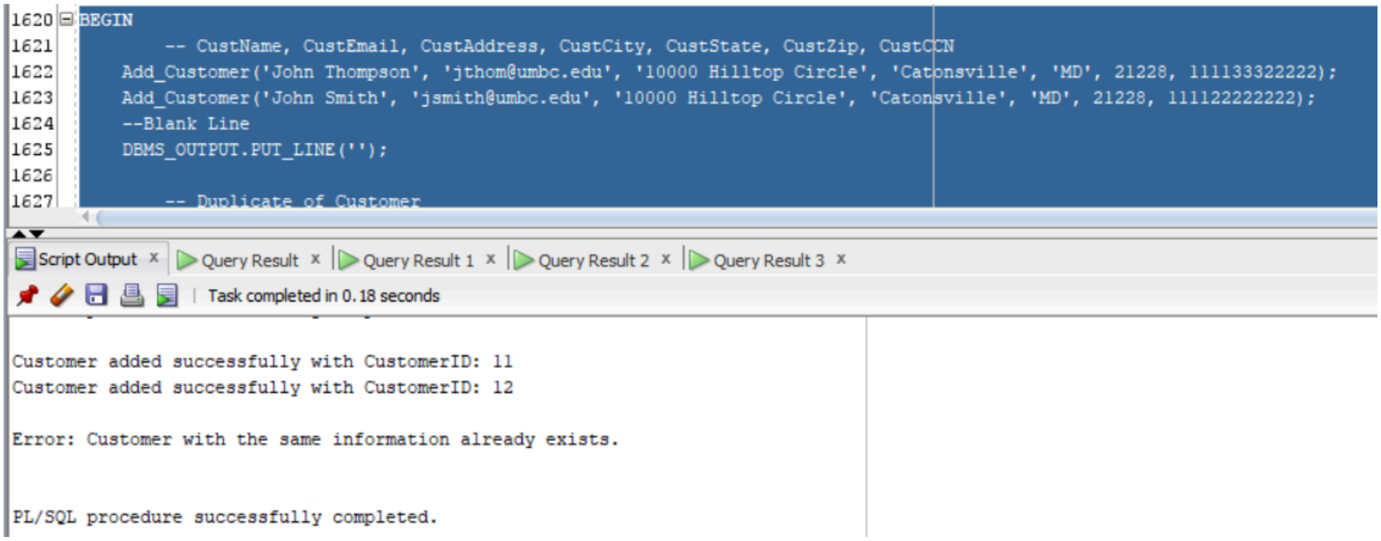
```
Select * from restaurant_inventory where restaurantID = 1;
```

All Rows Fetched: 6 in 0.023 seconds

	RESTAURANTINVENTORYID	RESTAURANTID	MENUITEMSID	RESTAURANTINVENTORYQUANTITY
1	1	1	1	54
2	2	1	2	88
3	3	1	3	67
4	4	1	4	97
5	5	1	5	45
6	39	1	20	40

Member 4 (Andrew Hollander):

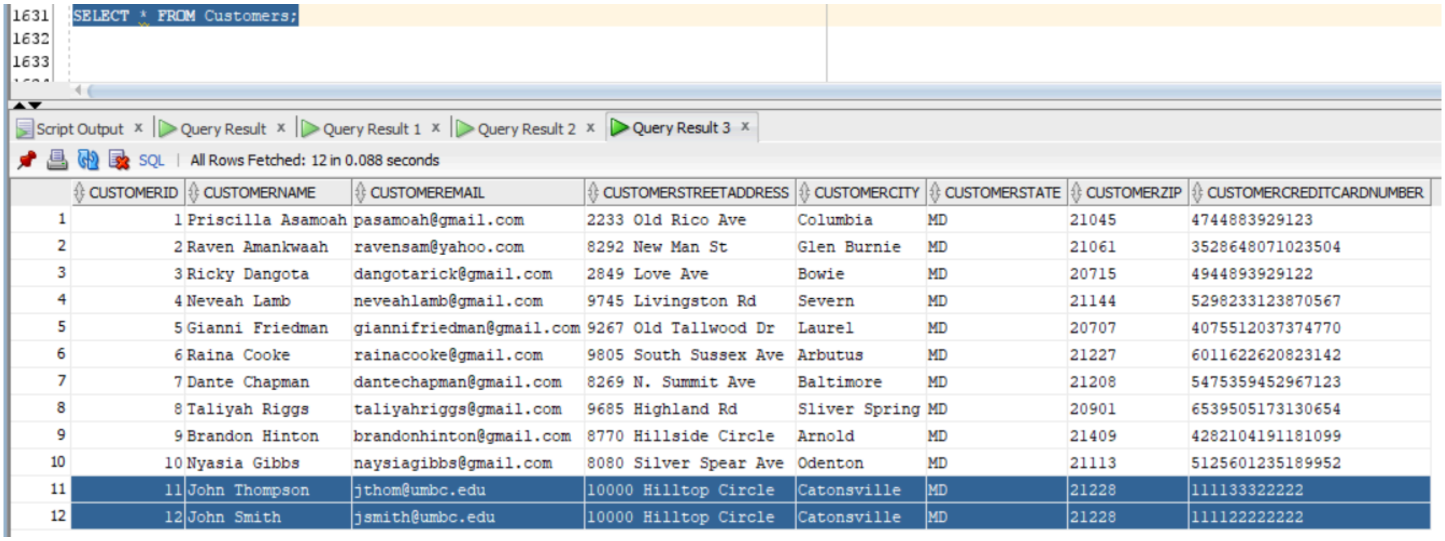
Add_Customer:



```
BEGIN
    -- CustName, CustEmail, CustAddress, CustCity, CustState, CustZip, CustCCN
    Add_Customer('John Thompson', 'jthom@umbc.edu', '10000 Hilltop Circle', 'Catonsville', 'MD', 21228, 111133322222);
    Add_Customer('John Smith', 'jsmith@umbc.edu', '10000 Hilltop Circle', 'Catonsville', 'MD', 21228, 111122222222);
    --Blank Line
    DBMS_OUTPUT.PUT_LINE('');

        -- Duplicate of Customer
```

Task completed in 0.18 seconds

```
Customer added successfully with CustomerID: 11
Customer added successfully with CustomerID: 12

Error: Customer with the same information already exists.


PL/SQL procedure successfully completed.
```

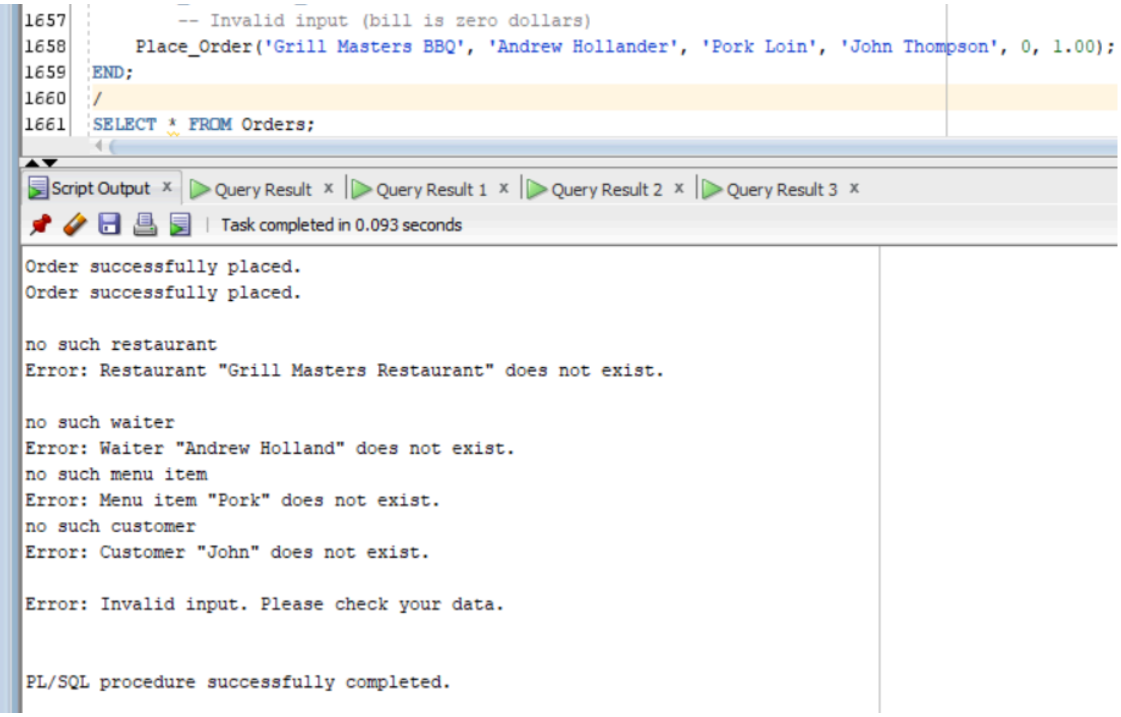


```
SELECT * FROM Customers;
```

All Rows Fetched: 12 in 0.088 seconds

	CUSTOMERID	CUSTOMERNAME	CUSTOMEREMAIL	CUSTOMERSTREETADDRESS	CUSTOMERCITY	CUSTOMERSTATE	CUSTOMERZIP	CUSTOMERCREDITCARDNUMBER
1	1	Priscilla Asamoah	pasamoah@gmail.com	2233 Old Rico Ave	Columbia	MD	21045	4744883929123
2	2	Raven Amankwaah	ravensam@yahoo.com	8292 New Man St	Glen Burnie	MD	21061	3528648071023504
3	3	Ricky Dangota	dangotarick@gmail.com	2849 Love Ave	Bowie	MD	20715	4944893929122
4	4	Neveah Lamb	neveahlamb@gmail.com	9745 Livingston Rd	Severn	MD	21144	5298233123870567
5	5	Gianni Friedman	giannifriedman@gmail.com	9267 Old Tallwood Dr	Laurel	MD	20707	4075512037374770
6	6	Raina Cooke	rainacooke@gmail.com	9805 South Sussex Ave	Arbutus	MD	21227	6011622620823142
7	7	Dante Chapman	dantechapman@gmail.com	8269 N. Summit Ave	Baltimore	MD	21208	5475359452967123
8	8	Taliyah Riggs	taliyahriggs@gmail.com	9685 Highland Rd	Sliver Spring	MD	20901	6539505173130654
9	9	Brandon Hinton	brandonhinton@gmail.com	8770 Hillside Circle	Arnold	MD	21409	4282104191181099
10	10	Nyasia Gibbs	naysiagibbs@gmail.com	8080 Silver Spear Ave	Odenton	MD	21113	5125601235189952
11	11	John Thompson	jthom@umbc.edu	10000 Hilltop Circle	Catonsville	MD	21228	111133322222
12	12	John Smith	jsmith@umbc.edu	10000 Hilltop Circle	Catonsville	MD	21228	111122222222

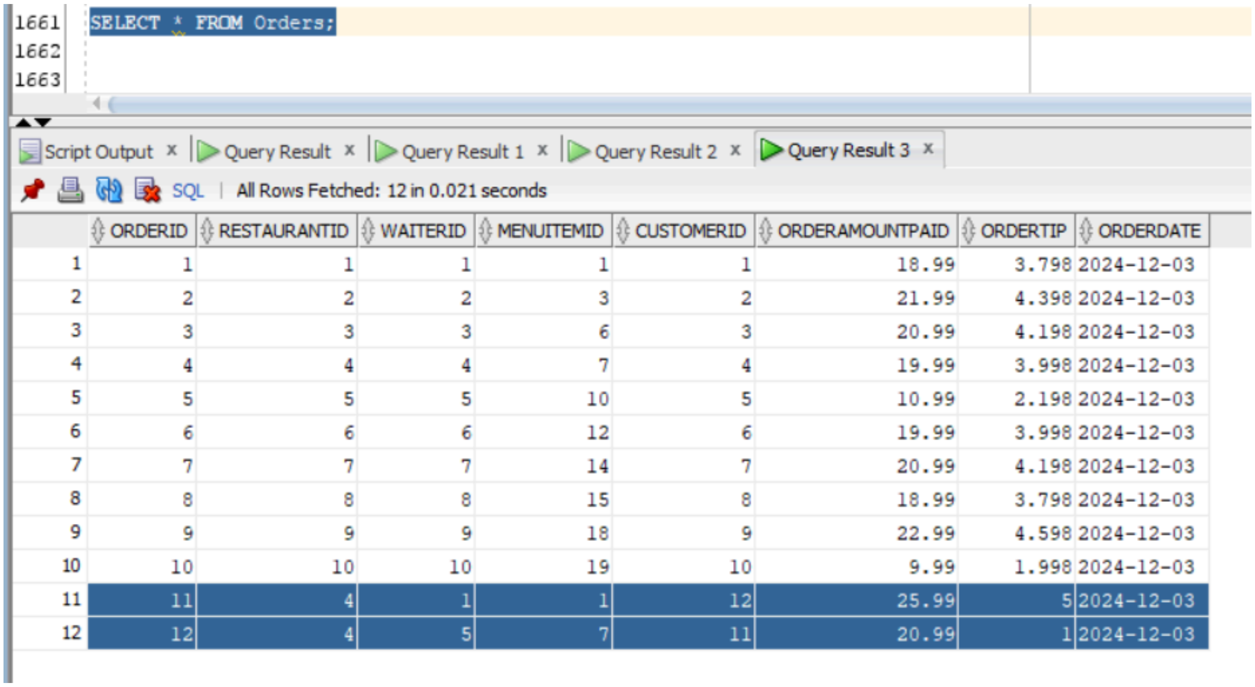
Place_Order:



```
        -- Invalid input (bill is zero dollars)
    Place_Order('Grill Masters BBQ', 'Andrew Hollander', 'Pork Loin', 'John Thompson', 0, 1.00);
END;
/
SELECT * FROM Orders;
```

Task completed in 0.093 seconds

```
Order successfully placed.
Order successfully placed.

no such restaurant
Error: Restaurant "Grill Masters Restaurant" does not exist.

no such waiter
Error: Waiter "Andrew Holland" does not exist.
no such menu item
Error: Menu item "Pork" does not exist.
no such customer
Error: Customer "John" does not exist.

Error: Invalid input. Please check your data.


PL/SQL procedure successfully completed.
```

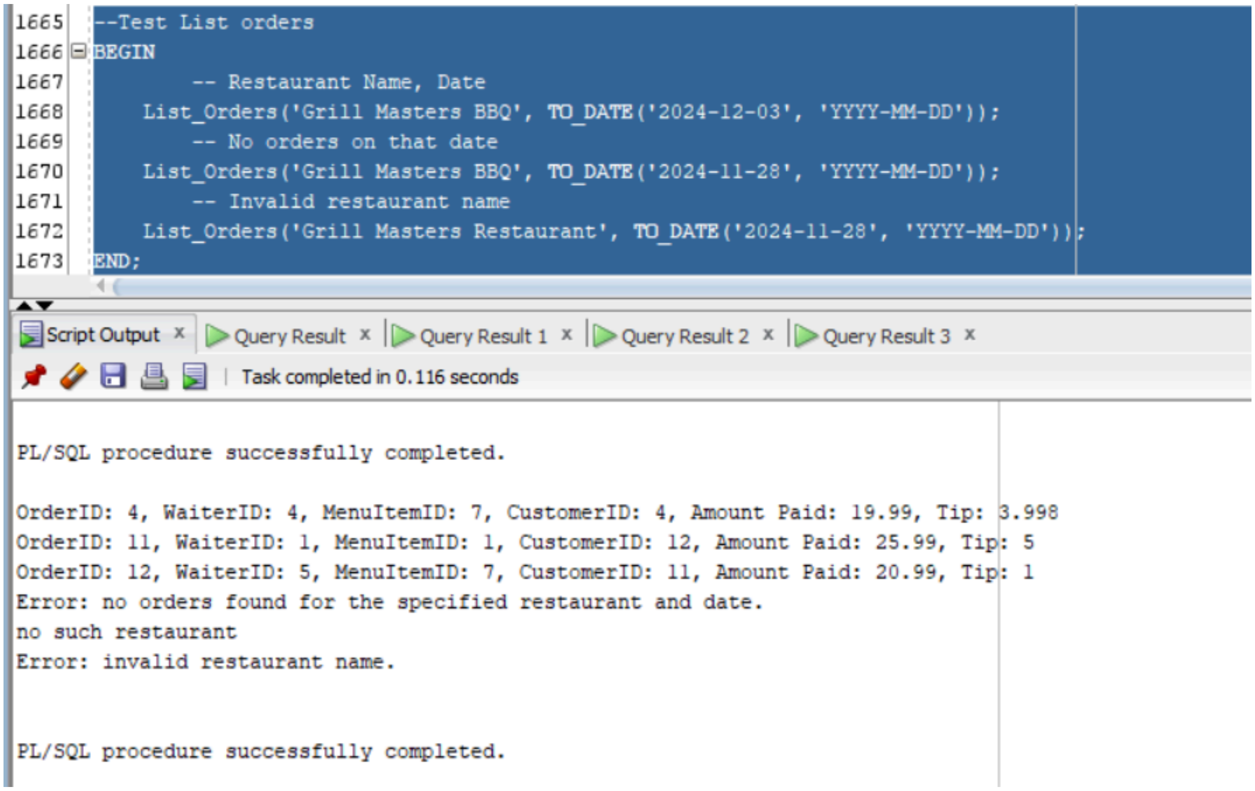


```
SELECT * FROM Orders;
```

All Rows Fetched: 12 in 0.021 seconds

	ORDERID	RESTAURANTID	WAITERID	MENUITEMID	CUSTOMERID	ORDERAMOUNTPAID	ORDERTIP	ORDERDATE
1	1	1	1	1	1	18.99	3.798	2024-12-03
2	2	2	2	3	2	21.99	4.398	2024-12-03
3	3	3	3	6	3	20.99	4.198	2024-12-03
4	4	4	4	7	4	19.99	3.998	2024-12-03
5	5	5	5	10	5	10.99	2.198	2024-12-03
6	6	6	6	12	6	19.99	3.998	2024-12-03
7	7	7	7	14	7	20.99	4.198	2024-12-03
8	8	8	8	15	8	18.99	3.798	2024-12-03
9	9	9	9	18	9	22.99	4.598	2024-12-03
10	10	10	10	19	10	9.99	1.998	2024-12-03
11	11	4	1	1	12	25.99	5	2024-12-03
12	12	4	5	7	11	20.99	1	2024-12-03

List_Orders:

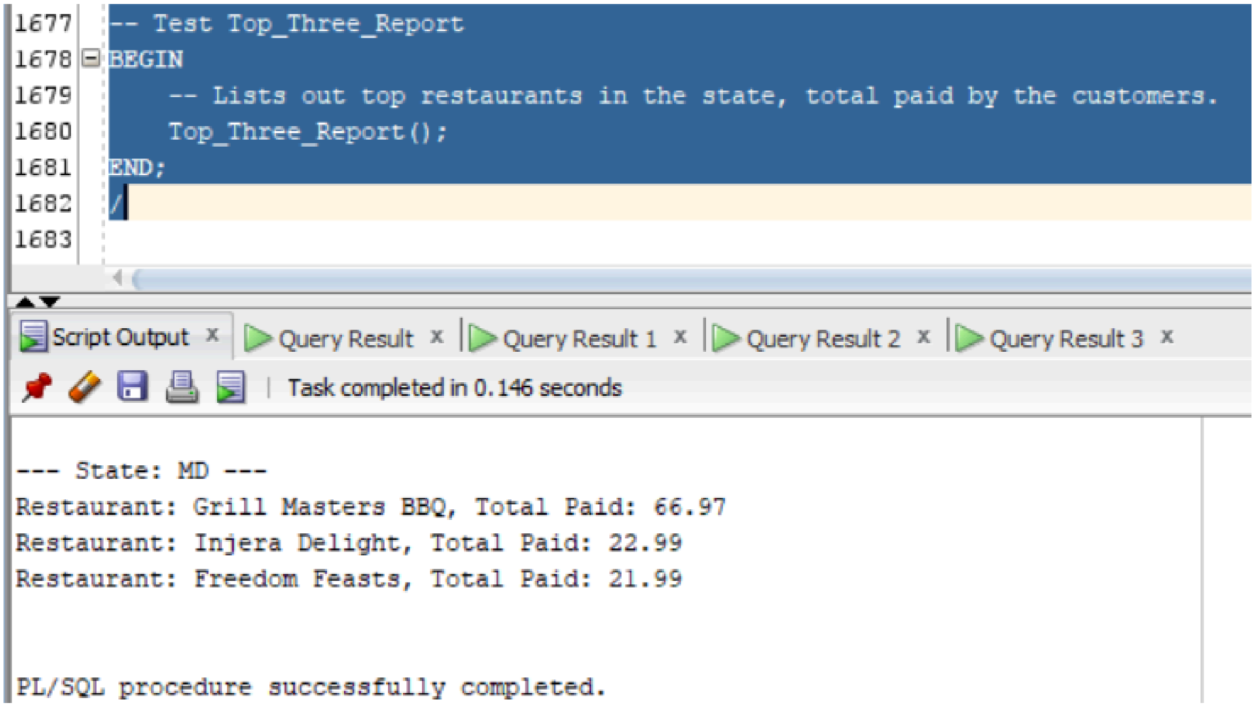


```
--Test List orders
BEGIN
        -- Restaurant Name, Date
    List_Orders('Grill Masters BBQ', TO_DATE('2024-12-03', 'YYYY-MM-DD'));
        -- No orders on that date
    List_Orders('Grill Masters BBQ', TO_DATE('2024-11-28', 'YYYY-MM-DD'));
        -- Invalid restaurant name
    List_Orders('Grill Masters Restaurant', TO_DATE('2024-11-28', 'YYYY-MM-DD'));
END;
```

Task completed in 0.116 seconds

```
PL/SQL procedure successfully completed.

OrderID: 4, WaiterID: 4, MenuItemID: 7, CustomerID: 4, Amount Paid: 19.99, Tip: 3.998
OrderID: 11, WaiterID: 1, MenuItemID: 1, CustomerID: 12, Amount Paid: 25.99, Tip: 5
OrderID: 12, WaiterID: 5, MenuItemID: 7, CustomerID: 11, Amount Paid: 20.99, Tip: 1
Error: no orders found for the specified restaurant and date.
no such restaurant
Error: invalid restaurant name.


PL/SQL procedure successfully completed.
```

Top_Three_Report:



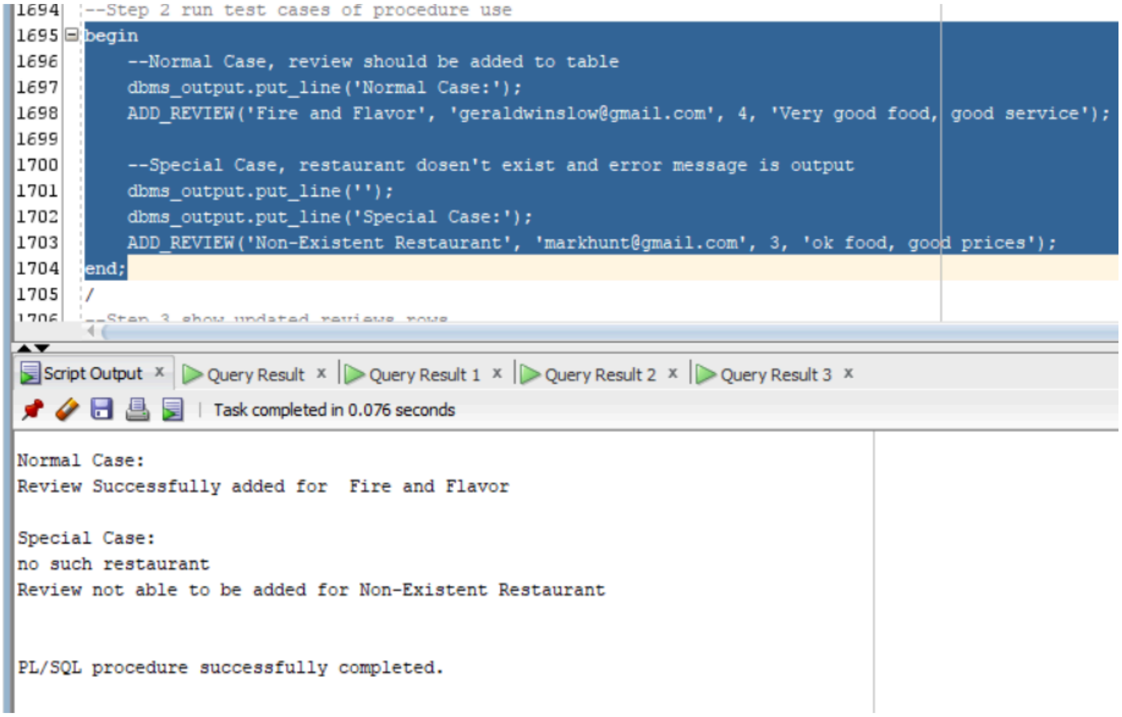
```
-- Test Top_Three_Report
BEGIN
    -- Lists out top restaurants in the state, total paid by the customers.
    Top_Three_Report();
END;
/
```

Task completed in 0.146 seconds

```
--- State: MD ---
Restaurant: Grill Masters BBQ, Total Paid: 66.97
Restaurant: Injera Delight, Total Paid: 22.99
Restaurant: Freedom Feasts, Total Paid: 21.99


PL/SQL procedure successfully completed.
```

Member 5:

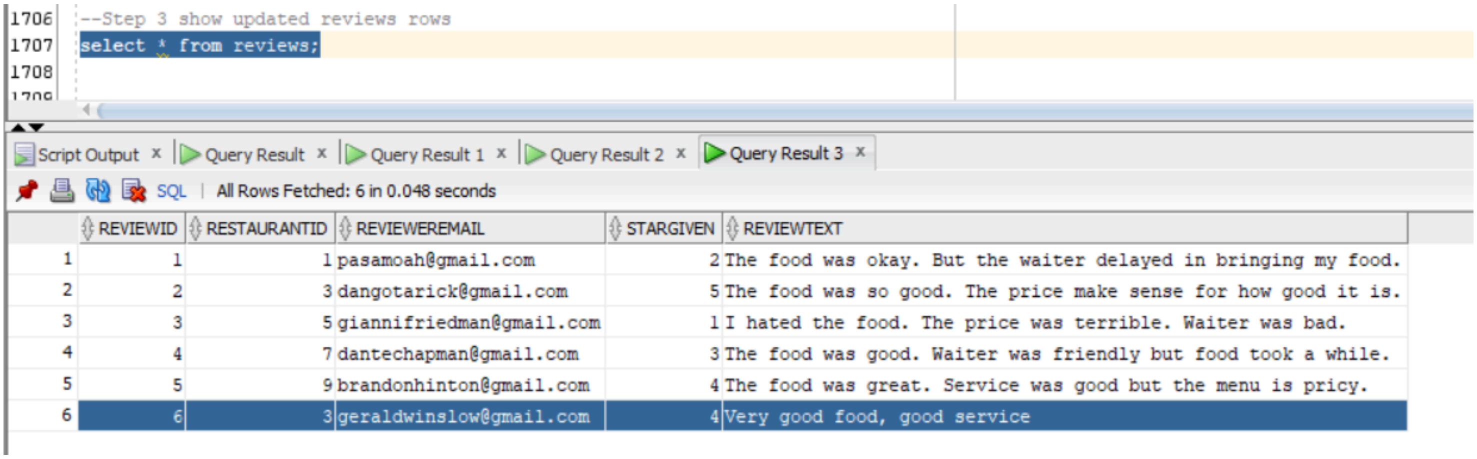
Add_Review:



```
--Step 2 run test cases of procedure use
begin
    --Normal Case, review should be added to table
    dbms_output.put_line('Normal Case:');
    ADD_REVIEW('Fire and Flavor', 'geraldwinslow@gmail.com', 4, 'Very good food, good service');

    --Special Case, restaurant dosen't exist and error message is output
    dbms_output.put_line('');
    dbms_output.put_line('Special Case:');
    ADD_REVIEW('Non-Existent Restaurant', 'markhunt@gmail.com', 3, 'ok food, good prices');
end;
/
--Step 3 show updated reviews rows
```

Task completed in 0.076 seconds

```
Normal Case:
Review Successfully added for  Fire and Flavor

Special Case:
no such restaurant
Review not able to be added for Non-Existent Restaurant


PL/SQL procedure successfully completed.
```

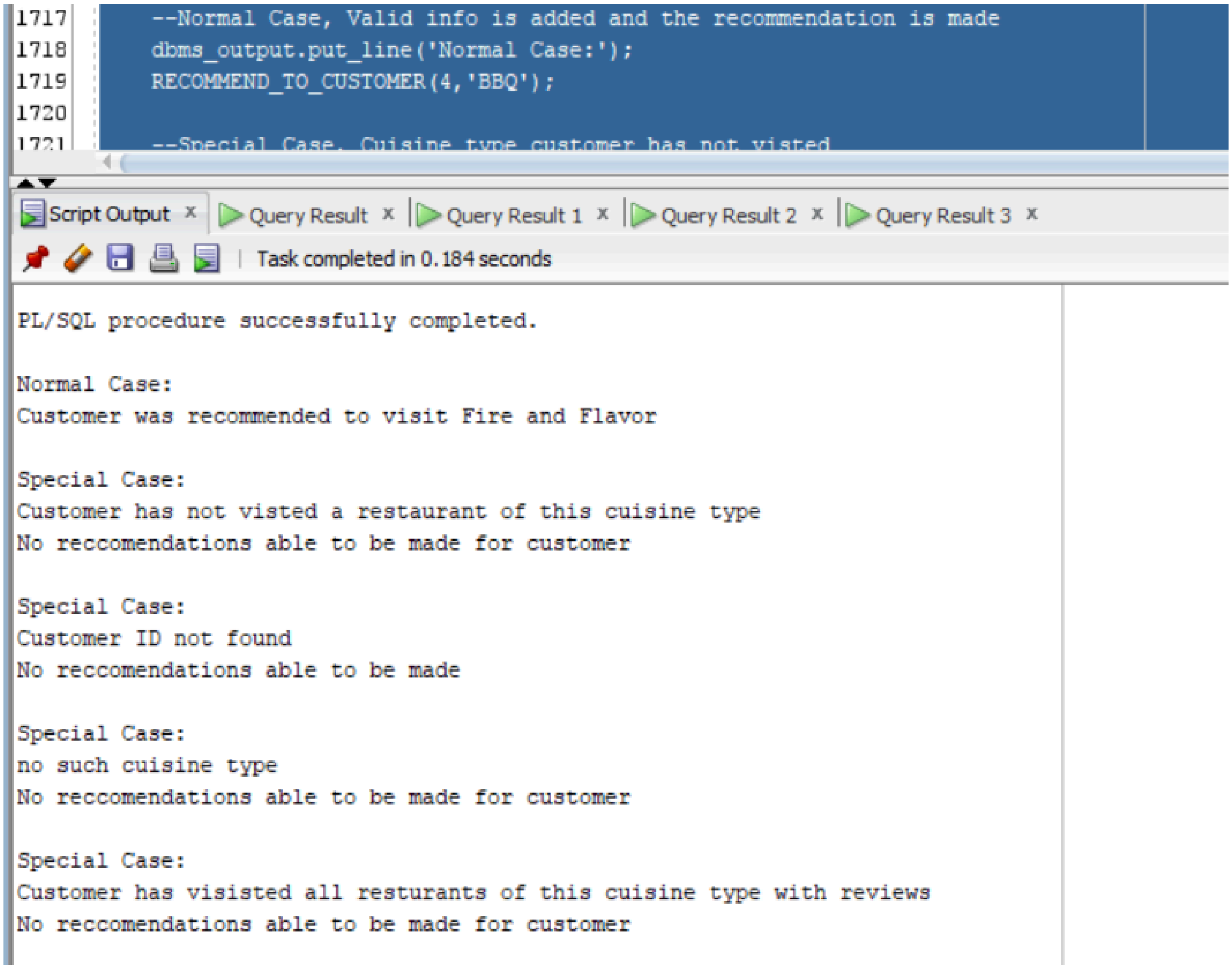


```
--Step 3 show updated reviews rows
select * from reviews;
```

All Rows Fetched: 6 in 0.048 seconds

	REVIEWID	RESTAURANTID	REVIEWEREMAIL	STARGIVEN	REVIEWTEXT
1	1	1	pasamoah@gmail.com	2	The food was okay. But the waiter delayed in bringing my food.
2	2	3	dangotarick@gmail.com	5	The food was so good. The price make sense for how good it is.
3	3	5	giannifriedman@gmail.com	1	I hated the food. The price was terrible. Waiter was bad.
4	4	7	dantechapman@gmail.com	3	The food was good. Waiter was friendly but food took a while.
5	5	9	brandonhinton@gmail.com	4	The food was great. Service was good but the menu is pricy.
6	6	3	geraldwinslow@gmail.com	4	Very good food, good service

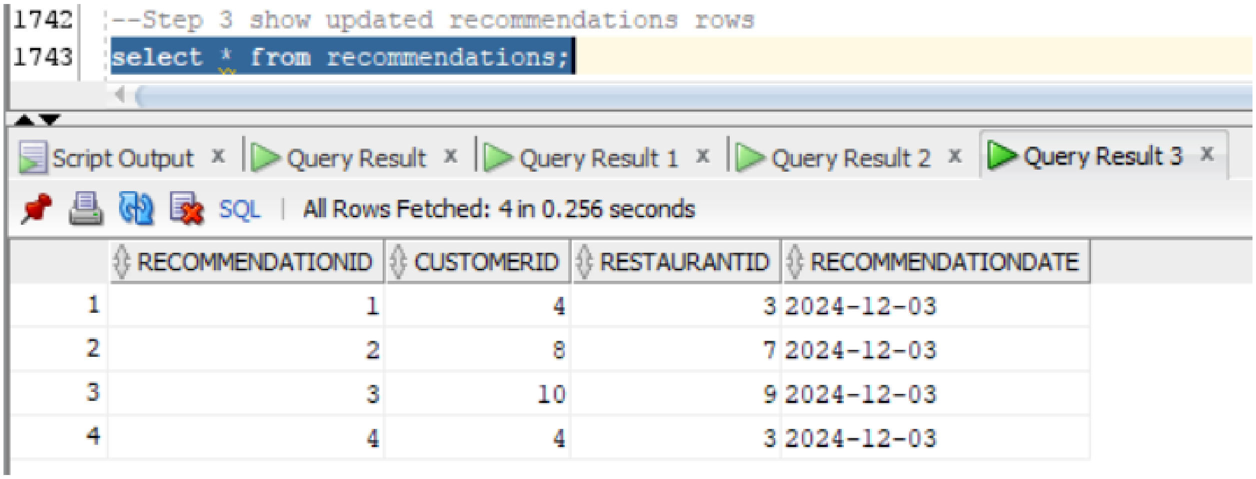
Recommend_To_Customer:



```
    --Normal Case, Valid info is added and the recommendation is made
    dbms_output.put_line('Normal Case:');
    RECOMMEND_TO_CUSTOMER(4,'BBQ');

    --Special Case. Cuisine type customer has not visted
```

Task completed in 0.184 seconds

```
PL/SQL procedure successfully completed.

Normal Case:
Customer was recommended to visit Fire and Flavor

Special Case:
Customer has not visted a restaurant of this cuisine type
No reccomendations able to be made for customer

Special Case:
Customer ID not found
No reccomendations able to be made

Special Case:
no such cuisine type
No reccomendations able to be made for customer

Special Case:
Customer has visisted all resturants of this cuisine type with reviews
No reccomendations able to be made for customer
```

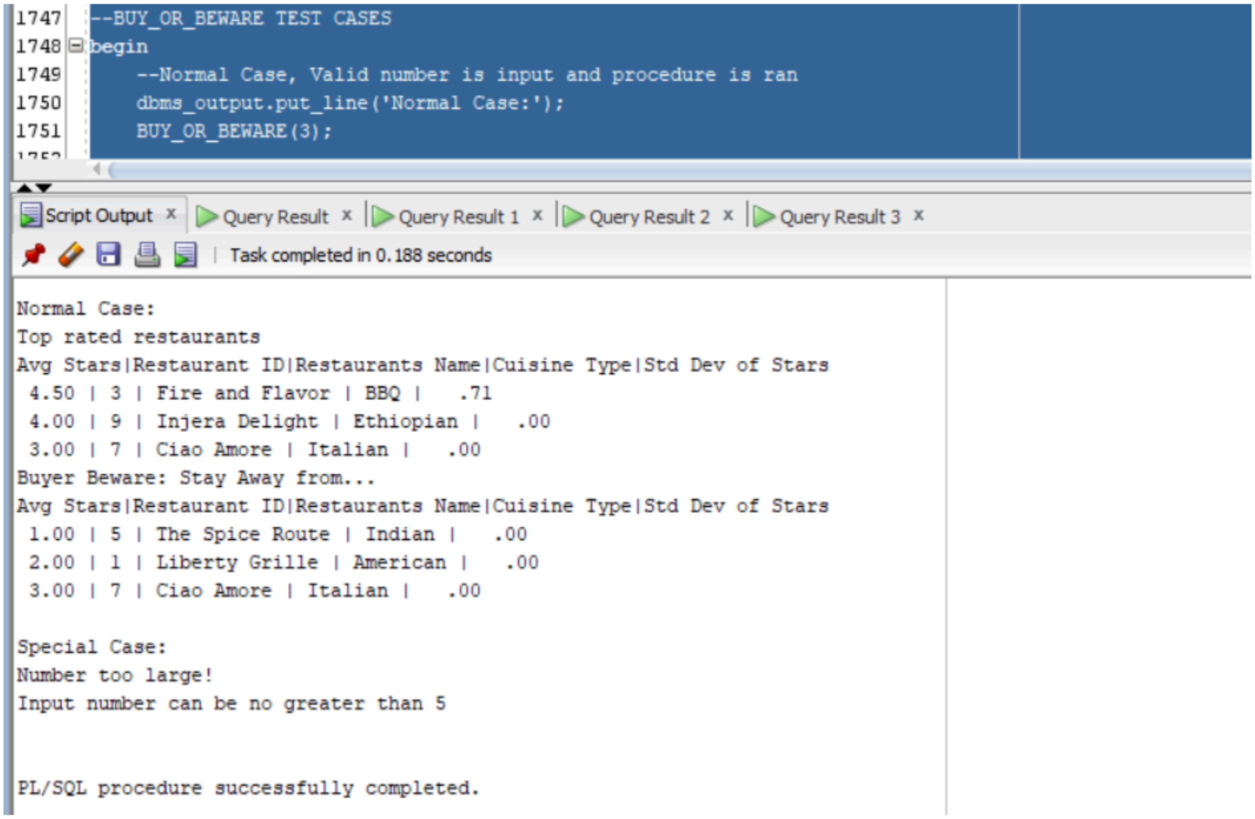


```
--Step 3 show updated recommendations rows
select * from recommendations;
```

All Rows Fetched: 4 in 0.256 seconds

	RECOMMENDATIONID	CUSTOMERID	RESTAURANTID	RECOMMENDATIONDATE
1	1	4	3	2024-12-03
2	2	8	7	2024-12-03
3	3	10	9	2024-12-03
4	4	4	3	2024-12-03

Buy_Or_Beware:



```
--BUY_OR_BEWARE TEST CASES
begin
    --Normal Case, Valid number is input and procedure is ran
    dbms_output.put_line('Normal Case:');
    BUY_OR_BEWARE(3);
```

```
Normal Case:
Top rated restaurants
Avg Stars|Restaurant ID|Restaurants Name|Cuisine Type|Std Dev of Stars
 4.50 | 3 | Fire and Flavor | BBQ |   .71
 4.00 | 9 | Injera Delight | Ethiopian |   .00
 3.00 | 7 | Ciao Amore | Italian |   .00
Buyer Beware: Stay Away from...
Avg Stars|Restaurant ID|Restaurants Name|Cuisine Type|Std Dev of Stars
 1.00 | 5 | The Spice Route | Indian |   .00
 2.00 | 1 | Liberty Grille | American |   .00
 3.00 | 7 | Ciao Amore | Italian |   .00

Special Case:
Number too large!
Input number can be no greater than 5


PL/SQL procedure successfully completed.
```

List_Recommendations:

```
create or replace function FIND_CUISINE_TYPE_ID (Cuisine_Name in
varchar) return number
is Cuisine_ID number;
begin
   select CuisineTypeID into Cuisine_ID from Cuisine_Type where CuisineTypeName LIKE
Cuisine_Name;
   return Cuisine_ID;
exception
   when no_data_found then
   dbms_output.put_line('no such cuisine type');
   return -1;
end;
```

```
Select FIND_CUISINE_TYPE_ID('Italian') from dual;
```

```
create or replace function FIND_RESTARANT_ID (Restaurant_Name in
varchar) return number
is Restaurant_ID number;
begin
   select RestaurantID into Restaurant_ID from Restaurants where RestaurantName LIKE
Restaurant_Name;
   return Restaurant_ID;
exception
   when no_data_found then
   dbms_output.put_line('no such restaurant');
   return -1;
end;

Select FIND_RESTARANT_ID('Liberty Grille') from dual;

create or replace function FIND_MENU_ITEM_ID (Menu_Item_Name in
varchar) return number
is Menu_Item_ID number;
begin
   select MenuItemsID into Menu_Item_ID from Menu_Items where MenuItemName LIKE
Menu_Item_Name;
   return Menu_Item_ID;
exception
   when no_data_found then
   dbms_output.put_line('no such menu item');
   return -1;
end;


Select FIND_MENU_ITEM_ID('Pizza') from dual;


create or replace function FIND_CUSTOMER_ID (Customer_Name in
varchar) return number
is Customer_ID number;
begin
   select CustomerID into Customer_ID from Customers where CustomerName LIKE
Customer_Name;
   return Customer_ID;
exception
   when no_data_found then
   dbms_output.put_line('no such customer');
   return -1;
end;


Select FIND_CUSTOMER_ID('Neveah Lamb') from dual;


create or replace function FIND_WAITER_ID (Waiter_Name in
varchar) return number
is Waiter_ID number;
begin
   select WaiterID into Waiter_ID from Waiters where WaiterName LIKE Waiter_Name;
   return Waiter_ID;
exception
   when no_data_found then
   dbms_output.put_line('no such waiter');
   return -1;
end;


Select FIND_WAITER_ID('Kaleb Baker') from dual;


create or replace function FIND_ORDER_ID (Restuarant_Name in
varchar, Waiter_Name in varchar, Menu_Item_Name in varchar, Customer_Name in varchar)
return number
is Order_ID number;
begin
   select OrderID into Order_ID from Orders
   where RestaurantID = (Select FIND_RESTARANT_ID(Restuarant_Name) from dual)
   and WaiterID = (Select FIND_WAITER_ID(Waiter_Name) from dual)
   and MenuItemID = (Select FIND_MENU_ITEM_ID(Menu_Item_Name) from dual)
   and CustomerID = (Select FIND_CUSTOMER_ID(Customer_Name) from dual);
   return Order_ID;
exception
   when no_data_found then
   dbms_output.put_line('no such order');
   return -1;
end;


Select FIND_ORDER_ID('Liberty Grille', 'Davina Bonsu', 'Burger', 'Priscilla Asamoah') from dual;
```