IS – 610 (Database Program Development)

Project Deliverables – 4

Team Name: Gryffindor

Team Members and their Roles:

- Karan Dhamecha – Research, Information Gathering, Coding and Documentation.
- Sonal Ingle – Information Gathering, Testing, Debugging and Implementation.
- Rehan Sawant – Research, Information Gathering, Coding and Implementation.
- Dinesh Dannina – Information Gathering, Testing, Debugging and Documentation.

Topic: Restaurant Management System

Note:

1-4: Project Deliverable 1

5-9: Project Deliverable 2

10-11: Project Deliverable 3

12-13: Project Deliverable 4

1. Business Situation:

We will create a database for Restaurant management system. The database will support basic functions including order information, customer information, transactions etc. The Restaurant has multiple branches and its staff members includes Manager, Chef and Waiter. The list of food items and the beverage list can be maintained in this database. Different types of sittings are offered to the customer such as VIP, Pool side, Bar etc. The customer record and billing details can also be maintained in this system. The designed database will be helpful for the managers to manage the restaurant and customers to meet their requirements.

2. Business Rules:

Restaurant Management System – Gryffindor

- Restaurant has multiple branches.
- Restaurant have staff members which includes Manager, Chef and Waiter.
- Manager allocates the table according to the customer preferences and tracking system.
- Waiter is assigned to customer after the customer is seated on the table.
- Once the customer places the order, the order is assigned to the chef. (Customized food items are charged at a higher price).
- The chef prepares the food as per the order.
- Prepared food is served to the customer.
- After the customer is done having food, a bill is generated.
- The customer can pay the bill via cash or card payment.

3. Business Data:

- Customer
- Waiter

- Rest_Branch
- Chef
- Order
- Manager
- Bill
- Item
- Table
- Mode of payment

4. Assumptions:

We will only consider the order information, customer information and transactions not the cost and revenue i.e. Profit or Loss statements.

5. Entities:

- Customer
- Staff Members
    - Sub-entities: Waiter, Chef and Manager
- "Order"
- Item
- Order_Item
- Bill
- Rest_Branch
- "Table"
- Mode of payment

6. Relationships:

- Restaurant has multiple branches and each branch has Staff Members.
- Staff members include Waiter, Manager and Chef
- Customer selects the type of Table.
- Customer places Order.
- Manager takes Order.
- Order contains items.
- Order and Item tables have associative relationship with Order_Item.
- Order is prepared by Chef.
- Waiter brings Order.
- Customer pays Bill.
- Bill has different Mode of Payments

7. Attributes:

- Customer
    - Cust_ID: Customer ID; key identifier; required; simple; single valued
    - Cust_Name{F_Name, L_Name}: Customer Name; key identifier; required; composite; single valued
    - Cust_Contact: Customer Contact No.; required; simple; multivalued

    - Cust_Email: Customer Email Address; optional; single valued

- "Order"
    - Order_No: Order No.; key identifier; required; simple; single valued
    - Cust_ID: Customer ID.; key identifier; required; simple; single valued
    - Item_No: Item No.; key identifier; required; simple; single valued
- Item
    - Item_No: Item No.; key identifier; required; simple; single valued
    - Item_Price: Price of the Item; required; simple; single valued
    - Item_Description: Item Description; required; simple; single valued
- Bill
    - Bill_No: Bill No.; key identifier; required; simple; single valued
    - Cust_ID: Customer ID; key identifier; required; simple; single valued
    - Branch_No: Branch No.; key identifier; required; simple; single valued
    - Order_No: Order No.; key identifier; required; simple; single valued
    - Bill_Amount: Bill Amount; key identifier; required; simple; single valued
- Rest_Branch
    - Branch_No: Branch No.; key identifier; required; simple; single valued
    - Branch_Name: Branch Name; key identifier; required; simple; single valued
    - Branch _Address (street, city, state, zip): Branch Address; key identifier; composite; single valued
    - Branch _Contact: Branch Contact No.; optional; simple; multivalued
- "Table"
    - Table_No: Table No.; key identifier; required; simple; single valued
    - Cust_ID: Customer ID; key identifier; required; simple; single valued
    - Table_Type: Type of Table; optional; simple; single valued
    - Table_Capacity: Table Capacity; optional; simple; single valued
- Staff Members (Supertype Entity)
    - Staff_ID: Staff ID; key identifier; required; simple; single valued
    - Branch_No: Branch No.; key identifier; required; simple; single valued
    - Staff_Type: Type of Staff; key identifier; required; simple; single valued
    - Staff_Hours (start_time, end_time): Hours completed by each staff members; required; composite; single valued
- Waiter (Subtype Entity)
    - Waiter_Name: Waiter Name; key identifier; required; simple; single valued
- Manager (Subtype Entity)
    - Manager_Name: Manager Name; key identifier; required; simple; single valued
- Chef (Subtype Entity)
    - Chef_Name: Chef Name; key identifier; required; simple; single valued
    - Chef_Type: Type of Chef; optional; simple; single valued
- Mode of Payment
    - Bill_No: Bill No.; key identifier; required; simple; single valued
    - Bill_Amount: Bill Amount; Optional; simple; single valued
    - Payment_Type: Payment Type; Optional; simple; single valued

8. EER Diagram

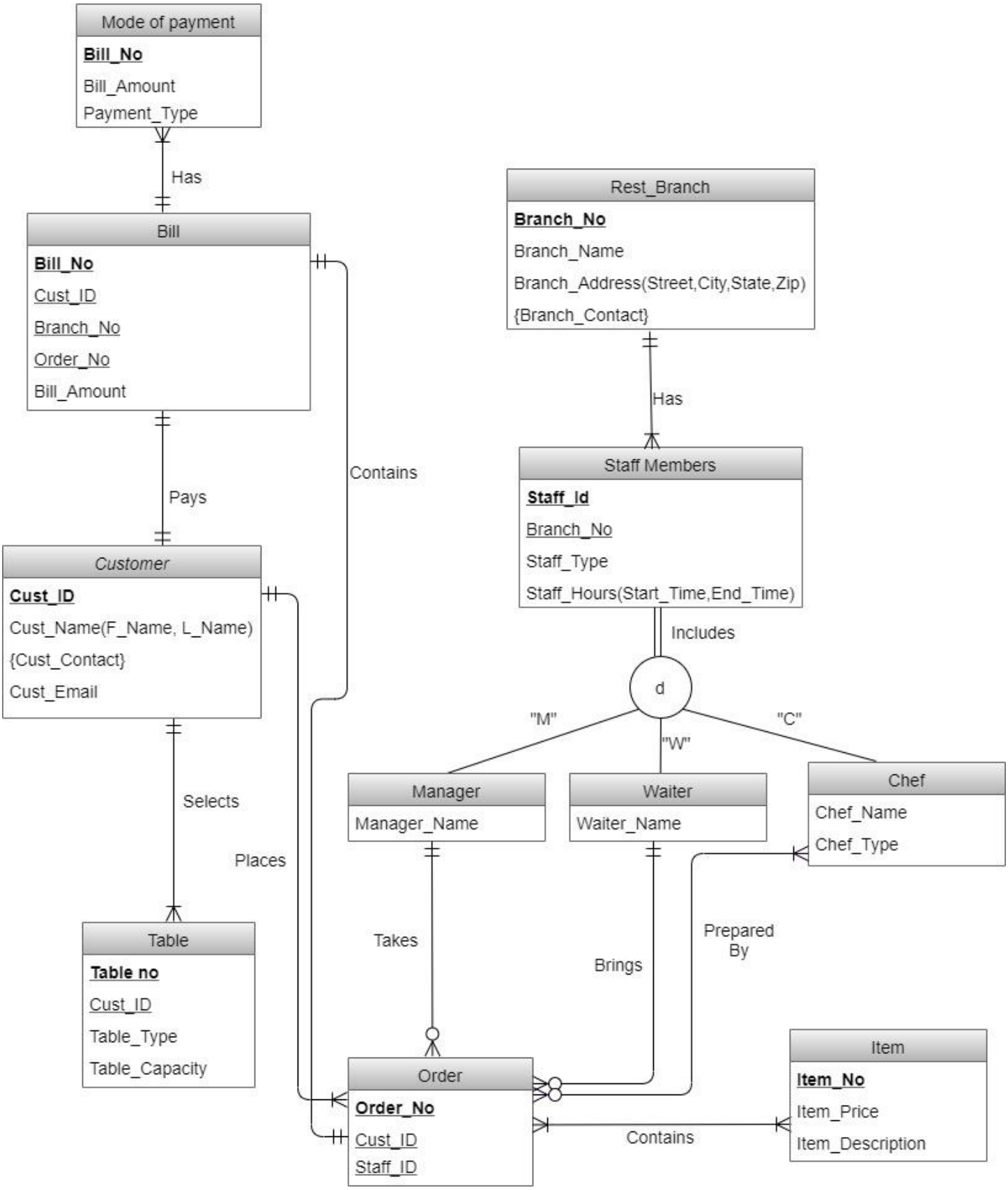


Fig 8

9. EER Diagram Explanation:

- One and only one Rest_Branch has atleast one or many staff members.
- One and only one Customer can select one or many tables.
- Staff Members (Supertype Entity) has a disjoint relationship with its subtype entities which includes Manager, Waiter and Chef.
- One and only one Customer can place atleast one or many Orders.
- One and only one Manager can take zero or many Orders.
- Zero or many orders are prepared by one or many Chefs.
- One and only one Waiter delivers zero or many Orders to the table.
- One or many Orders contain one or many items.
- Customer and Bill shares one to one relationship.
- One and only one Bill has one or many Payment Types.
- Bill and Order shares one to one relationship.

10. Relational tables in 3NF:

Note: All Primary Key attributes cannot be Null and are Unique.

- **Customer:**
  - **Primary Key:** Cust_ID
  - **Non-Primary Attributes:** F_Name, L_Name, Cust_Contact, Cust_Email

- **Table:**
  - **Primary Key:** Table_No
  - **Non-Primary Attributes:** Table_Type, Table_Capacity
  - **Foreign Key:** Cust_ID

- **Item:**
  - **Primary Key:** Item_No
  - **Non-Primary Attributes:** Item_Quantity, Item_Price, Item_Description

- **Order:**
  - **Primary Key:** Order_No
  - **Foreign Key:** Item_No

- **Order_Item**
  - **Foreign Key:** Order_No, Item_No
  - **Non-Primary Attributes**: Item_Description

- **OrderChef:**
  - **Primary Key:** Oder_No, Staff_ID, Chef_ID

- **Rest_Branch:**
  - **Primary Key:** Branch_No
  - **Non-Primary Attributes:** Branch_Name, Street, City, State, Zip, Branch_Contact

- **Staff_Members:**
  - **Primary Key:** Staff_ID
  - **Non-Primary Attributes:** Staff_Type, Start_Time, End_Time
  - **Foreign_Key:** Branch_No

- **Manager:**
  - **Primary Key:** Staff_ID
  - **Non-Primary Attributes:** Manager_Name

- **Waiter:**
  - **Primary Key:** Staff_ID
  - **Non-Primary Attributes:** Waiter_Name

- **Chef:**
  - **Primary Key:** Staff_ID
  - **Non-Primary Attributes:** Chef_Name, Chef_Type

- **Bill:**
  - **Primary Key:** Bill_No
  - **Non-Primary Attributes:** Bill_amount
  - **Foreign_Key:** Cust_ID, Branch_No, Order_ID

- **Mode of Payment:**
  - **Foreign Key:** Bill_No
  - **Non-Primary Attributes:** Bill_amount, Payment_Type

Refer the Relational tables in 3NF as below:

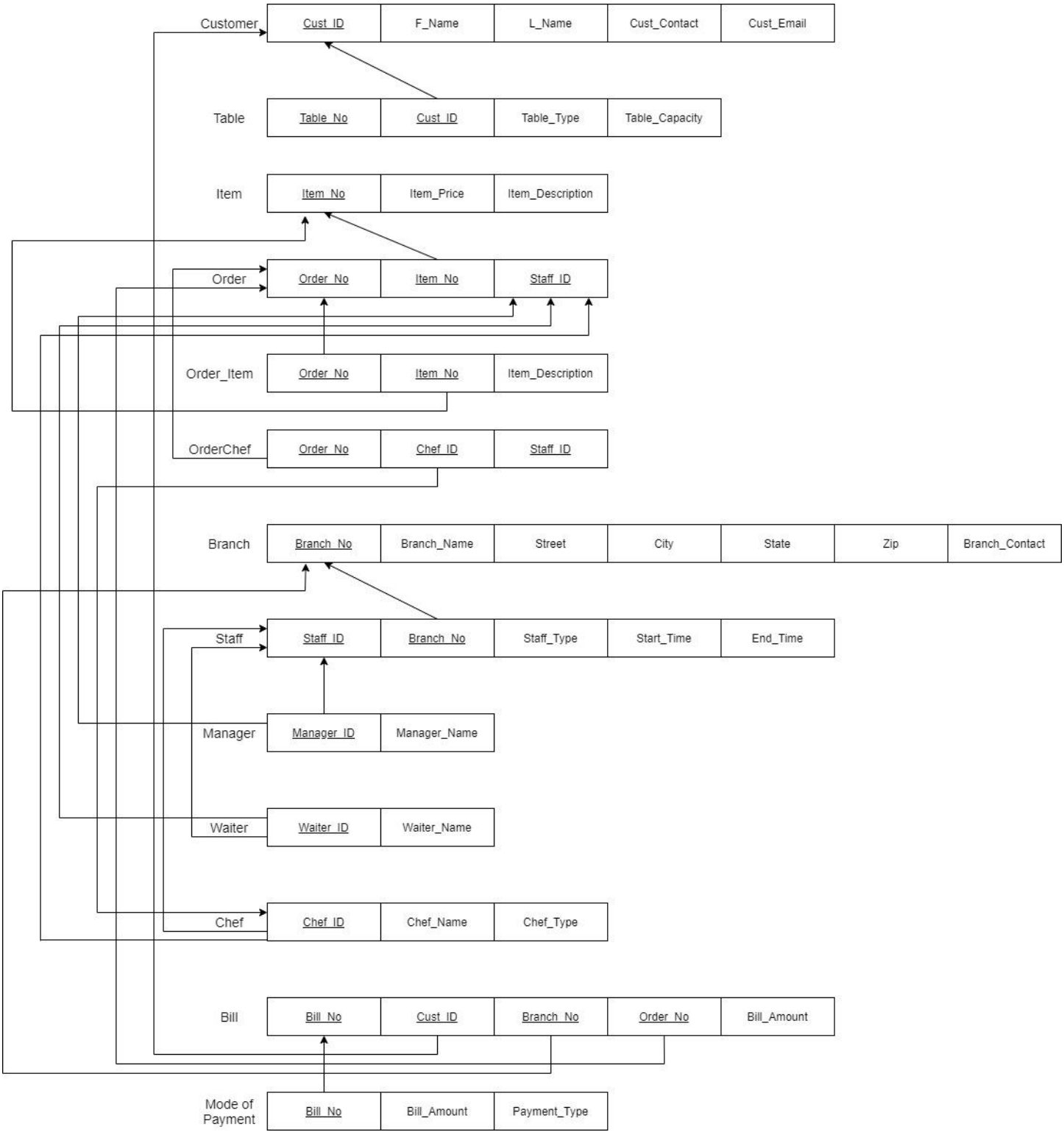


Fig 10

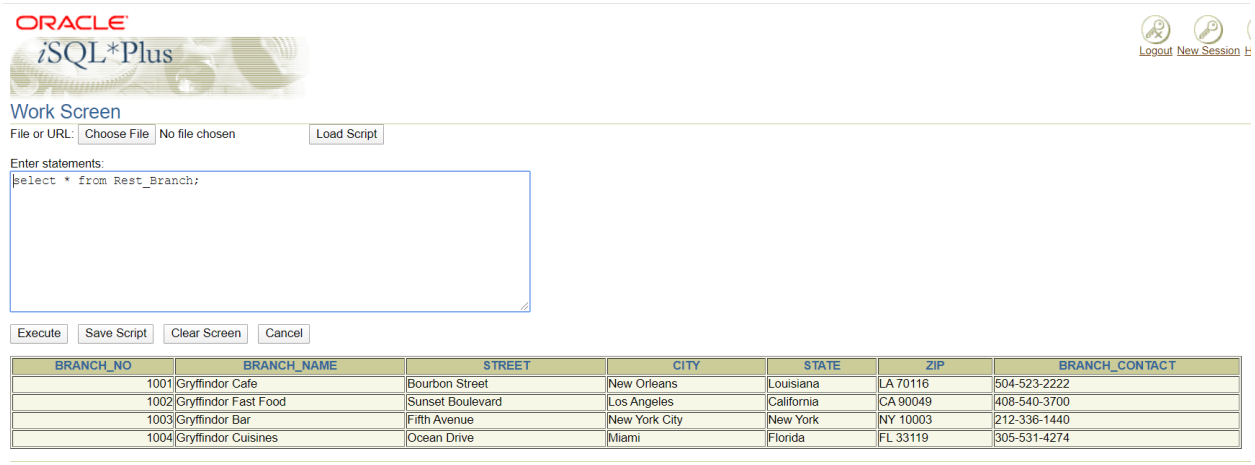
11. Table Explanation:

- **Customer:**
  - The Customer table has Primary Key Cust_ID.
  - All the non-primary attributes are functionally dependent on Primary attribute Cust_ID.
- **Table:**
  - The "Table" table has Primary Key Table_No and Foreign Key Customer_ID.
  - All the non-primary attributes are functionally dependent on Primary attribute Table_No.
- **Item:**
  - The Item table has Primary Key Item_No.
  - All the non-primary attributes are functionally dependent on Primary attribute Item_No.
- **Order:**
  - The Order table has Primary Key Order_No and Foreign Key Item_No.
  - All the non-primary attributes are functionally dependent on Primary attribute Order_No.
- **Order_Item:**
  - The Order_Item has Foreign keys Order_No, Item_No.
  - Non-primary attribute is Item_Description.
- **OrderChef:**
  - The OrderChef table has Foreign Keys Order_No, Staff_ID, Chef_ID.
- **Rest_Branch:**
  - The Branch table has Primary Key Branch_No.
    All the non-primary attributes are functionally dependent on Primary attribute Branch_No.
- **Staff:**
  - The Staff table has Primary Key Staff_ID and Foreign Key Branch_No.
  - The Staff table is the Super Type. It has disjoint relationship with Manager, Waiter and Chef.
  - All the non-primary attributes are functionally dependent on Primary attribute Staff_ID.
- **Manager:**
  - The Manager table has Primary Key Manager_ID and Foreign Key Branch_No.
  - The Manager table is the Subtype Entity of Supertype Entity Staff.
  - All the non-primary attributes are functionally dependent on Primary attribute Manager_ID.
- **Waiter:**
  - The Waiter table has Primary Key Waiter_ID and Foreign Key Branch_No.
  - The Waiter table is the Subtype Entity of Supertype Entity Staff.
  - All the non-primary attributes are functionally dependent on Primary attribute Waiter_ID.
- **Chef:**
  - The Chef table has Primary Key Chef_ID and Foreign Key Branch_No.
  - The Chef table is the Subtype Entity of Supertype Entity Staff.
  - All the non-primary attributes are functionally dependent on Primary attribute Chef_ID.
- **Bill:**
  - The Bill table has Primary Key Bill_No and Foreign Key Customer_ID , Branch_No and Order_No.
    All the non-primary attributes are functionally dependent on Primary attribute Bill_No.
- **Mode of Payment:**
  - The Mode of Payment table has Foreign Key Bill_No.
  - All the non-primary attributes are functionally dependent on Primary attribute Bill_No.

12. Creation of Tables and Insertion of Records

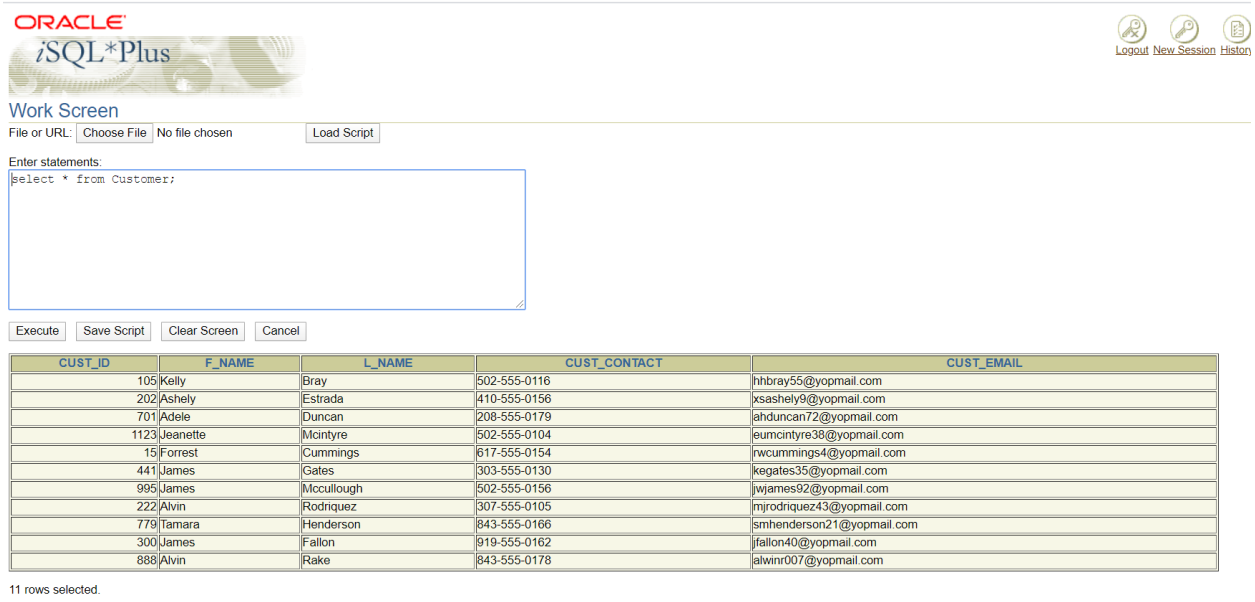
We have created the tables and inserted values in those tables as below:

- **Rest_Branch:**



BRANCH_NO	BRANCH_NAME	STREET	CITY	STATE	ZIP	BRANCH_CONTACT
1001	Gryffindor Cafe	Bourbon Street	New Orleans	Louisiana	LA 70116	504-523-2222
1002	Gryffindor Fast Food	Sunset Boulevard	Los Angeles	California	CA 90049	408-540-3700
1003	Gryffindor Bar	Fifth Avenue	New York City	New York	NY 10003	212-336-1440
1004	Gryffindor Cuisines	Ocean Drive	Miami	Florida	FL 33119	305-531-4274

- **Customer:**



CUST_ID	F_NAME	L_NAME	CUST_CONTACT	CUST_EMAIL
105	Kelly	Bray	502-555-0116	hhbray55@yopmail.com
202	Ashely	Estrada	410-555-0156	xsashely9@yopmail.com
701	Adele	Duncan	208-555-0179	ahduncan72@yopmail.com
1123	Jeanette	Mcintyre	502-555-0104	eumcintyre38@yopmail.com
15	Forrest	Cummings	617-555-0154	rwcummings4@yopmail.com
441	James	Gates	303-555-0130	kegates35@yopmail.com
995	James	Mccullough	502-555-0156	jwjames92@yopmail.com
222	Alvin	Rodriquez	307-555-0105	mjrodriquez43@yopmail.com
779	Tamara	Henderson	843-555-0166	smhenderson21@yopmail.com
300	James	Fallon	919-555-0162	jfallon40@yopmail.com
888	Alvin	Rake	843-555-0178	alwinr007@yopmail.com

11 rows selected.

- **Item:**

Enter statements:

```
select * from Item;
```

Execute | Save Script | Clear Screen | Cancel

ITEM_NO	ITEM_PRICE	ITEM_DESCRIPTION
1	12	Croque Americaine - A Breakfast Panini
2	10	Artichoke Frittata Panini
3	18	Breakfast Pizza
4	20	Avocado Toast and Poached Eggs
5	15	Spinach and Feta Quiche
6	25	Ham and Swiss Gruyere Quiche
7	22	Nectarine Mascarpone French Toast Panini
8	30	Specialty Bread Sample Plate
9	28	Smoked Salmon Platter
10	12	Steel Cut Oatmeal
11	8	Yogurt Parfait
12	15	Yogurt Cup
13	20	Yogurt with House-made Granola
14	20	Bread and Cie House-made Granola
15	22	Chicken Cobb Sandwich
16	10	Bud Light
17	25	Jack Daneils
18	18	Old Monk

18 rows selected.

- **“Table”:**

ORACLE iSQL*Plus

Logout | New Session | Histo

Work Screen

File or URL: Choose File No file chosen Load Script

Enter statements:

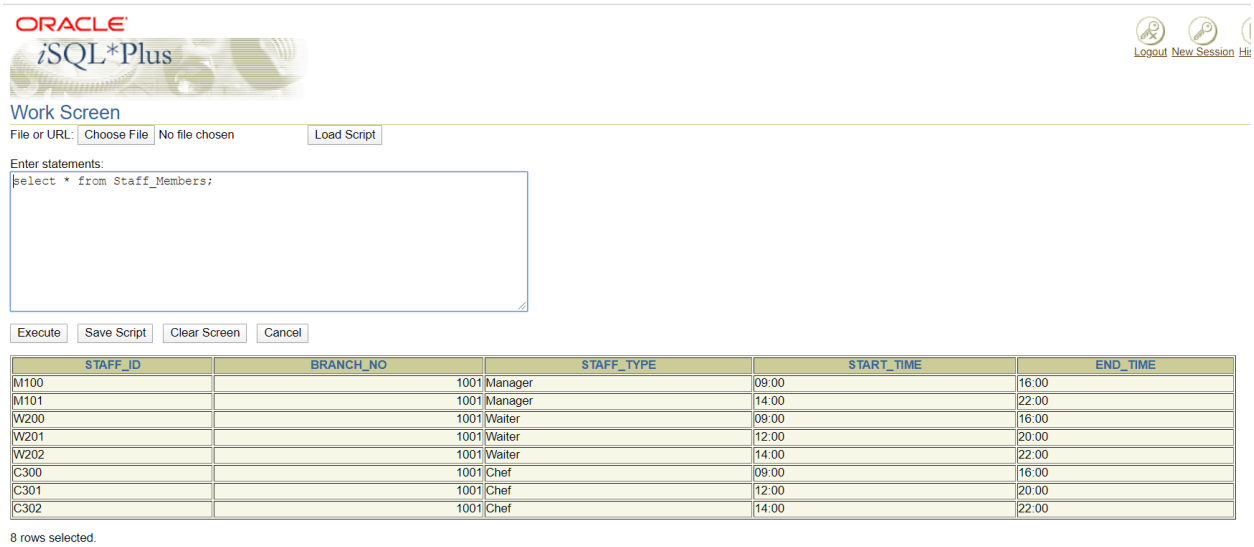
```
select * from "Table";
```

Execute | Save Script | Clear Screen | Cancel

TABLE_NO	CUST_ID	TABLE_TYPE	TABLE_CAPACITY
1	202	Bar	4
2	701	Poolside	4
3	15	Bar	8
4	300	VIP	2
5	888	Poolside	4
6	1123	VIP	4
7	995	Bar	8
8	441	Poolside	8
9	105	Poolside	4
10	779	Bar	4
11	222	Poolside	8

11 rows selected.

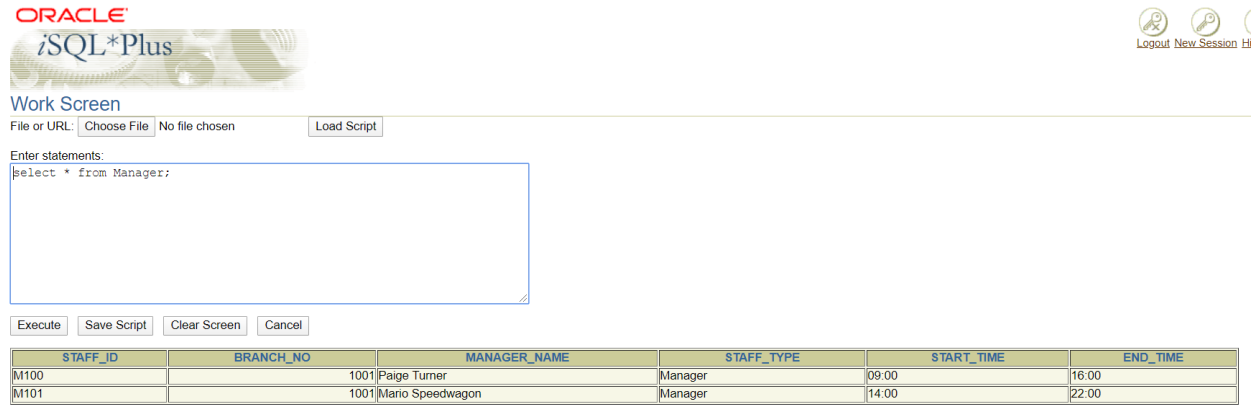
- **Staff_Members:**



STAFF_ID	BRANCH_NO	STAFF_TYPE	START_TIME	END_TIME
M100	1001	Manager	09:00	16:00
M101	1001	Manager	14:00	22:00
W200	1001	Waiter	09:00	16:00
W201	1001	Waiter	12:00	20:00
W202	1001	Waiter	14:00	22:00
C300	1001	Chef	09:00	16:00
C301	1001	Chef	12:00	20:00
C302	1001	Chef	14:00	22:00

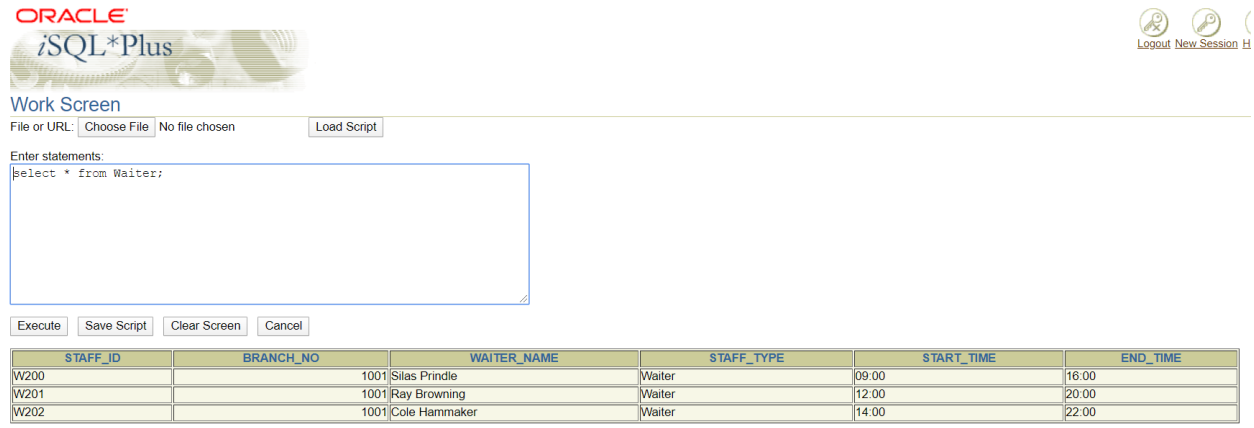
8 rows selected.

- **Manager:**



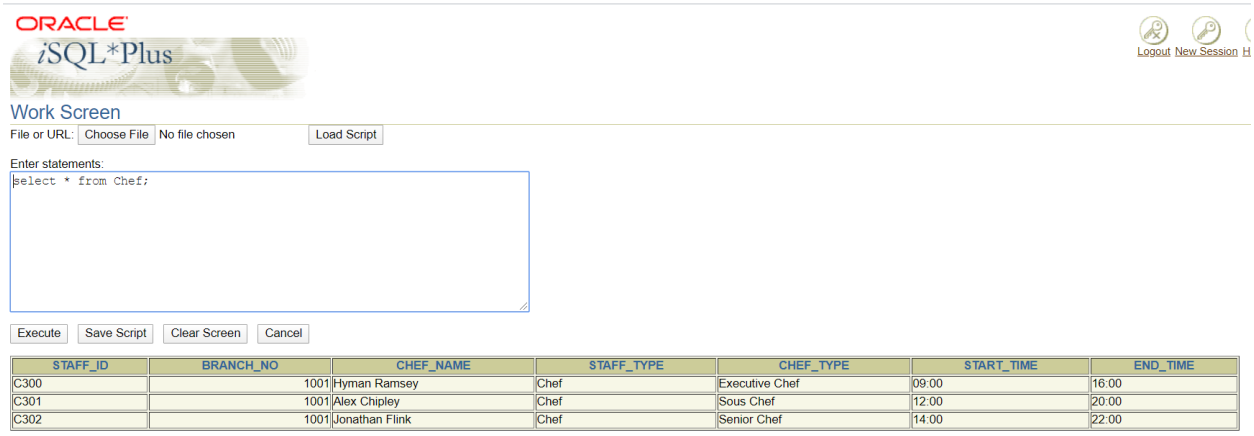
STAFF_ID	BRANCH_NO	MANAGER_NAME	STAFF_TYPE	START_TIME	END_TIME
M100	1001	Paige Turner	Manager	09:00	16:00
M101	1001	Mario Speedwagon	Manager	14:00	22:00

- **Waiter:**



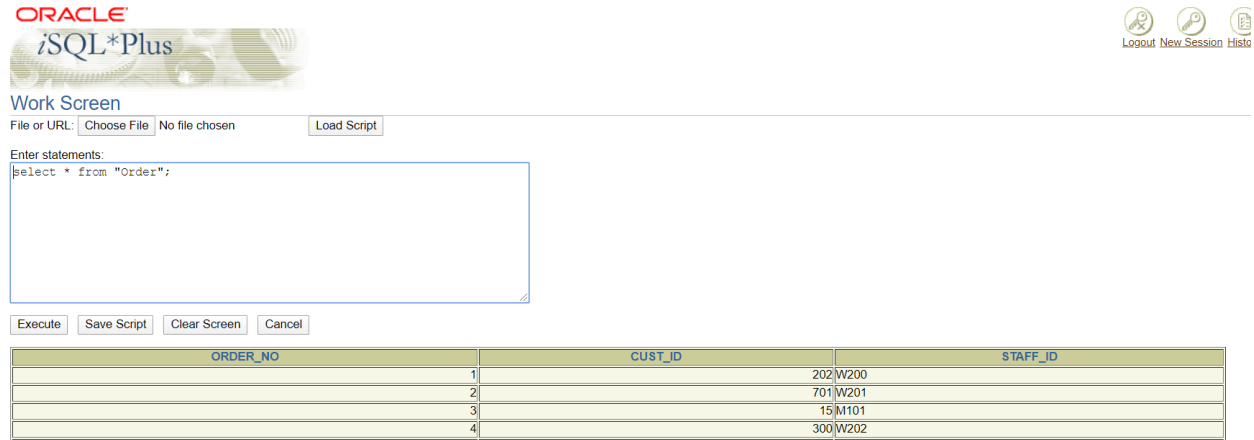
STAFF_ID	BRANCH_NO	WAITER_NAME	STAFF_TYPE	START_TIME	END_TIME
W200	1001	Silas Prindle	Waiter	09:00	16:00
W201	1001	Ray Browning	Waiter	12:00	20:00
W202	1001	Cole Hammaker	Waiter	14:00	22:00

- **Chef:**



STAFF_ID	BRANCH_NO	CHEF_NAME	STAFF_TYPE	CHEF_TYPE	START_TIME	END_TIME
C300	1001	Hyman Ramsey	Chef	Executive Chef	09:00	16:00
C301	1001	Alex Chipley	Chef	Sous Chef	12:00	20:00
C302	1001	Jonathan Flink	Chef	Senior Chef	14:00	22:00

- **“Order”:**



ORDER_NO	CUST_ID	STAFF_ID
1	202	W200
2	701	W201
3	15	M101
4	300	W202
5	888	M100
6	1123	W200
7	995	W202
8	441	M101
9	105	W200
10	779	W201
11	222	M100

11 rows selected.

- **Order_Item:**

Enter statements:

```
select * from Order_Item;
```

Execute | Save Script | Clear Screen | Cancel

ORDER_NO	ITEM_NO	ITEM_DESCRIPTION
1	3	Breakfast Pizza
1	18	Old Monk
2	6	Ham and Swiss Gruyere Quiche
3	9	Smoked Salmon Platter
3	16	Bud Light
4	15	Chicken Cobb Sandwich
4	9	Smoked Salmon Platter
4	2	Artichoke Frittata Panini
5	12	Yogurt Cup
5	4	Avocado Toast and Poached Eggs
5	5	Spinach and Feta Quiche
6	15	Chicken Cobb Sandwich
6	3	Breakfast Pizza
7	17	Jack Daneils
7	14	Bread and Cie House-made Granola
7	6	Ham and Swiss Gruyere Quiche
8	13	Yogurt with House-made Granola
8	8	Specialty Bread Sample Plate
9	9	Smoked Salmon Platter
9	4	Avocado Toast and Poached Eggs
10	18	Old Monk
10	15	Chicken Cobb Sandwich
10	7	Nectarine Mascarpone French Toast Panini
11	8	Specialty Bread Sample Plate
11	3	Ham and Swiss Gruyere Quiche

- **Bill:**

ORACLE iSQL*Plus

Logout | New Session | His

Work Screen

File or URL: Choose File | No file chosen | Load Script

Enter statements:

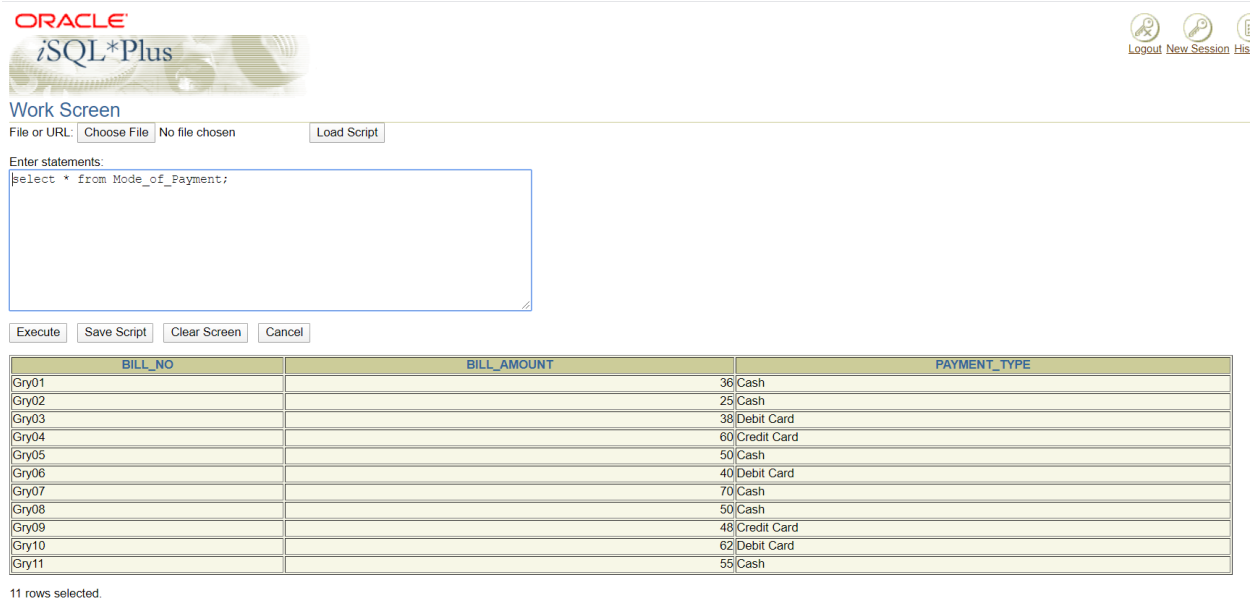
```
select * from Bill;
```

Execute | Save Script | Clear Screen | Cancel

BILL_NO	CUST_ID	BRANCH_NO	ORDER_NO	BILL_AMOUNT
Gry01	202	1001	1	36
Gry02	701	1001	2	25
Gry03	15	1001	3	38
Gry04	300	1001	4	60
Gry05	888	1001	5	50
Gry06	1123	1001	6	40
Gry07	995	1001	7	70
Gry08	441	1001	8	50
Gry09	105	1001	9	48
Gry10	779	1001	10	62
Gry11	222	1001	11	55

11 rows selected.

- **Mode_of_Payment:**

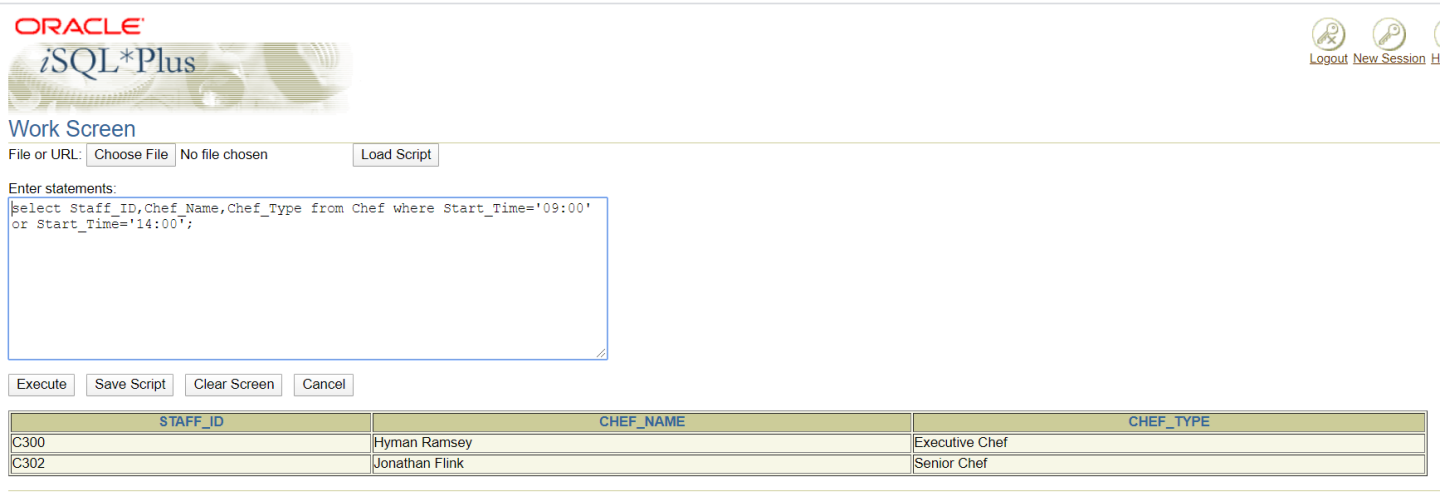


BILL_NO	BILL_AMOUNT	PAYMENT_TYPE
Gry01	36	Cash
Gry02	25	Cash
Gry03	38	Debit Card
Gry04	60	Credit Card
Gry05	50	Cash
Gry06	40	Debit Card
Gry07	70	Cash
Gry08	50	Cash
Gry09	48	Credit Card
Gry10	62	Debit Card
Gry11	55	Cash

11 rows selected.

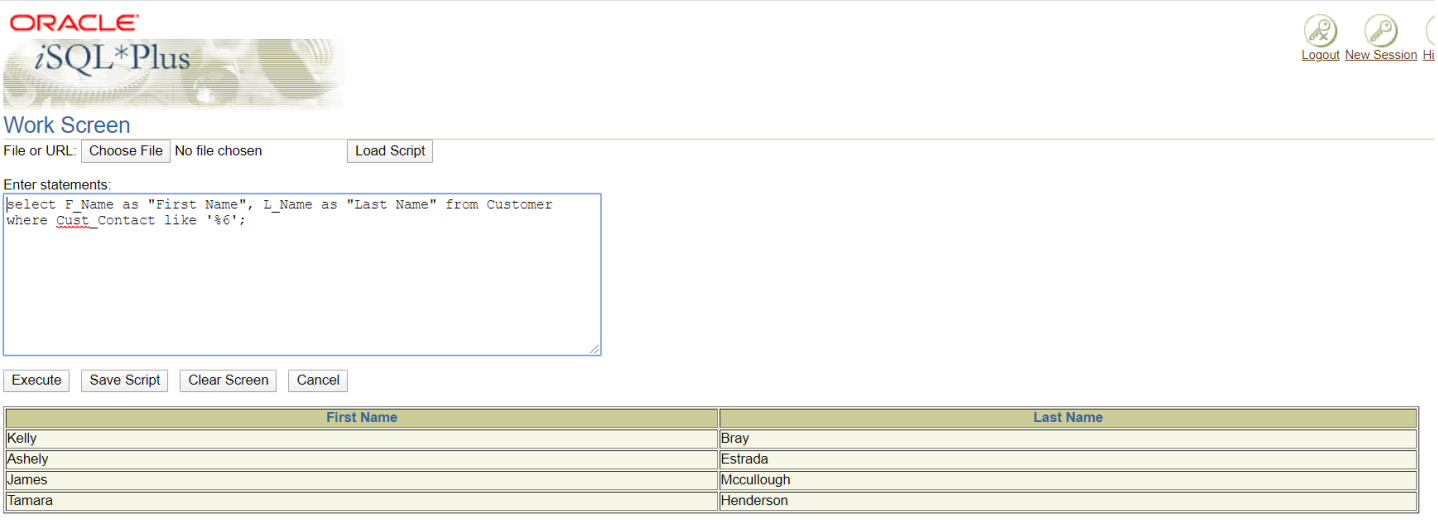
13. Querying of data in the tables:

- **SQL select queries:**
  - **Purpose of Query:** This query displays the details of all the chef whose Start Time is 09:00 or Start Time is 14:00 as below:
  - **Query:** select Staff_ID,Chef_Name,Chef_Type from Chef where Start_Time='09:00' or Start_Time='14:00';

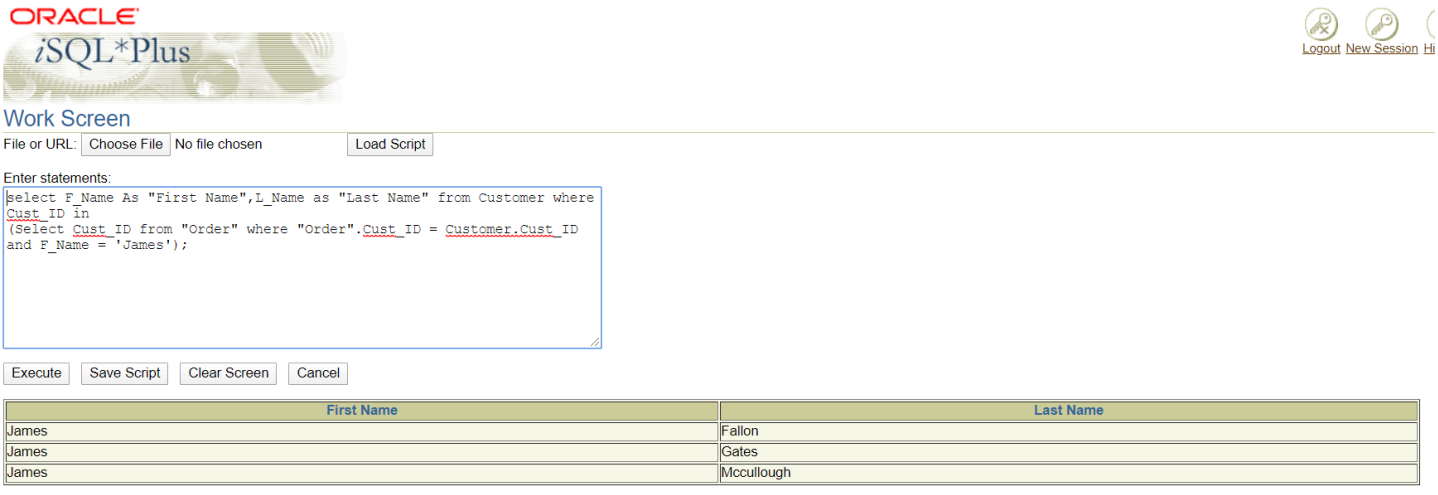


STAFF_ID	CHEF_NAME	CHEF_TYPE
C300	Hyman Ramsey	Executive Chef
C302	Jonathan Flink	Senior Chef

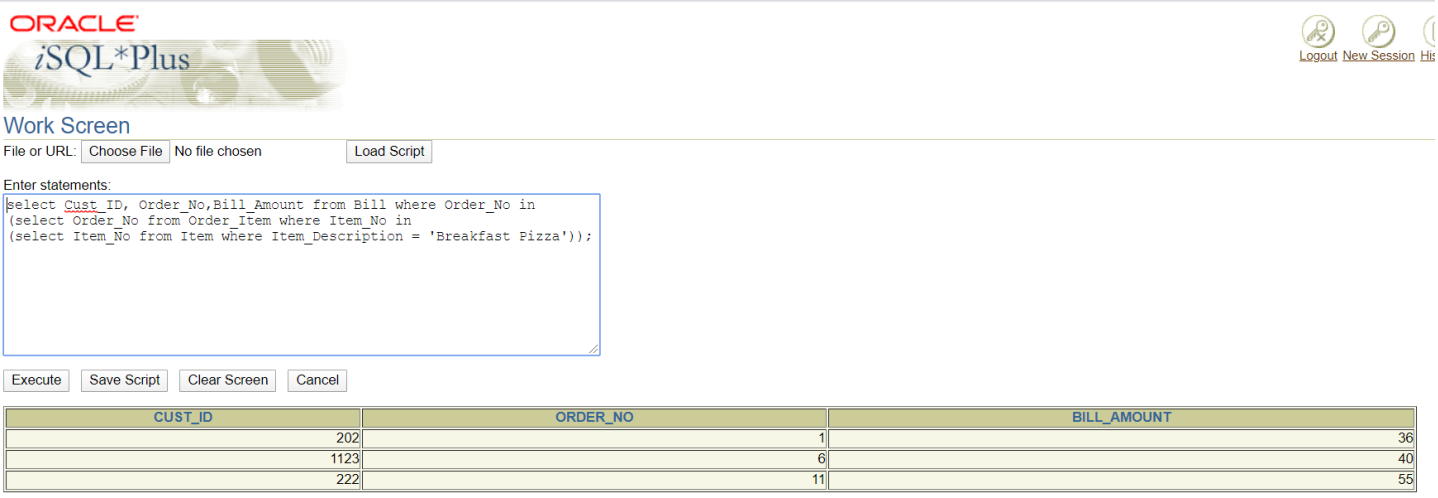
  - **Purpose of Query:** This query displays the First Name and Last Name of all the Customers whose Customer Contact contains Any of the digit containing 6 except the first Digit as below:
  - **Query:** select F_Name as "First Name", L_Name as "Last Name" from Customer where Cust_Contact like '%6';



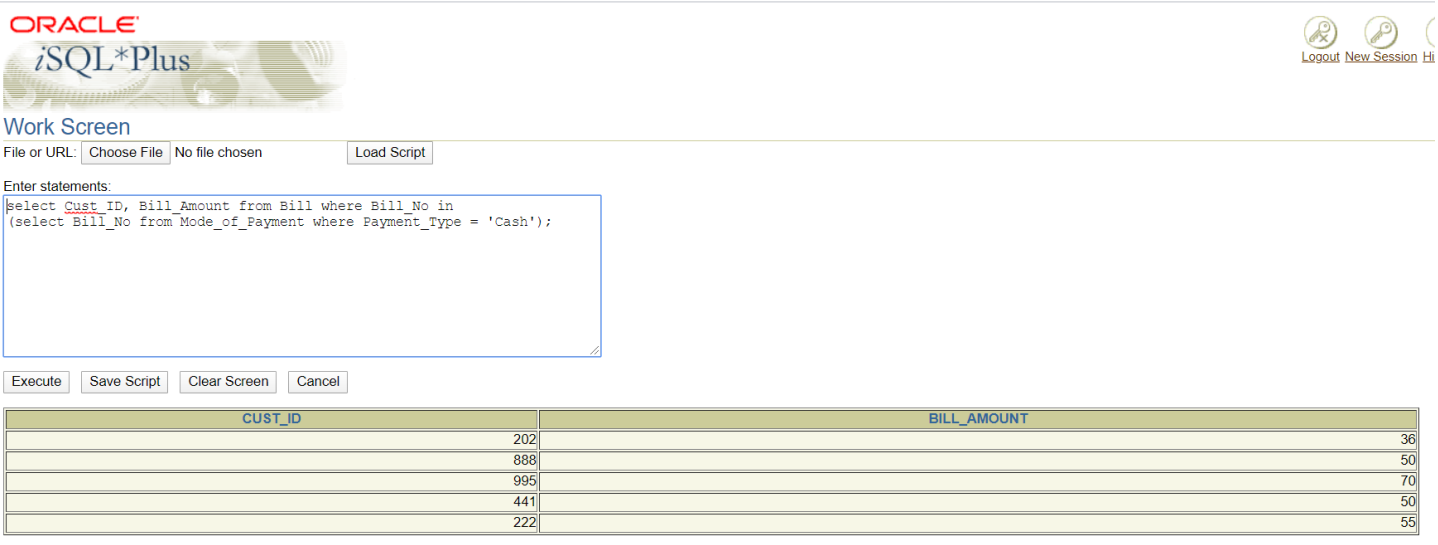
- **Purpose of Query (Join Query):** This query displays the First Name and Last Name who have placed an order and whose First Name is 'James' as below:
- **Query:** select F_Name As "First Name",L_Name as "Last Name" from Customer where Cust_ID in (Select Cust_ID from "Order" where "Order".Cust_ID = Customer.Cust_ID
  and F_Name = 'James');



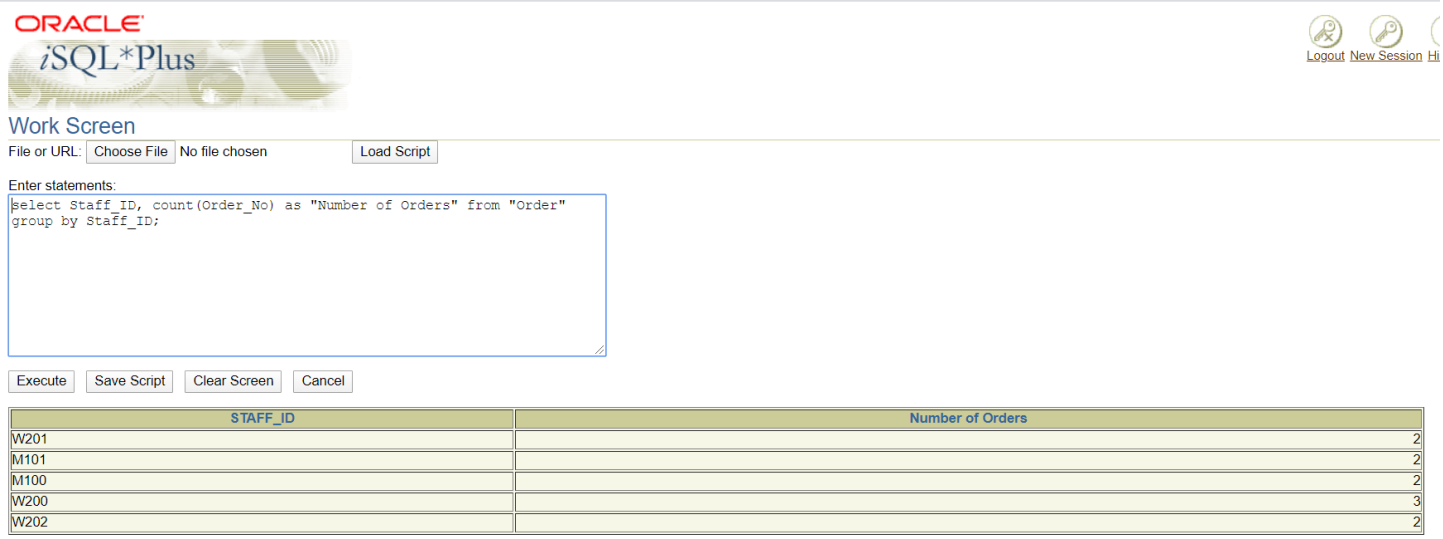
- **Purpose of Query (Join Query):** This query displays the Bill Details when we provide Item Description as below:
- **Query:** select Cust_ID, Order_No,Bill_Amount from Bill where Order_No in
  (select Order_No from Order_Item where Item_No in
  (select Item_No from Item where Item_Description = 'Breakfast Pizza'));



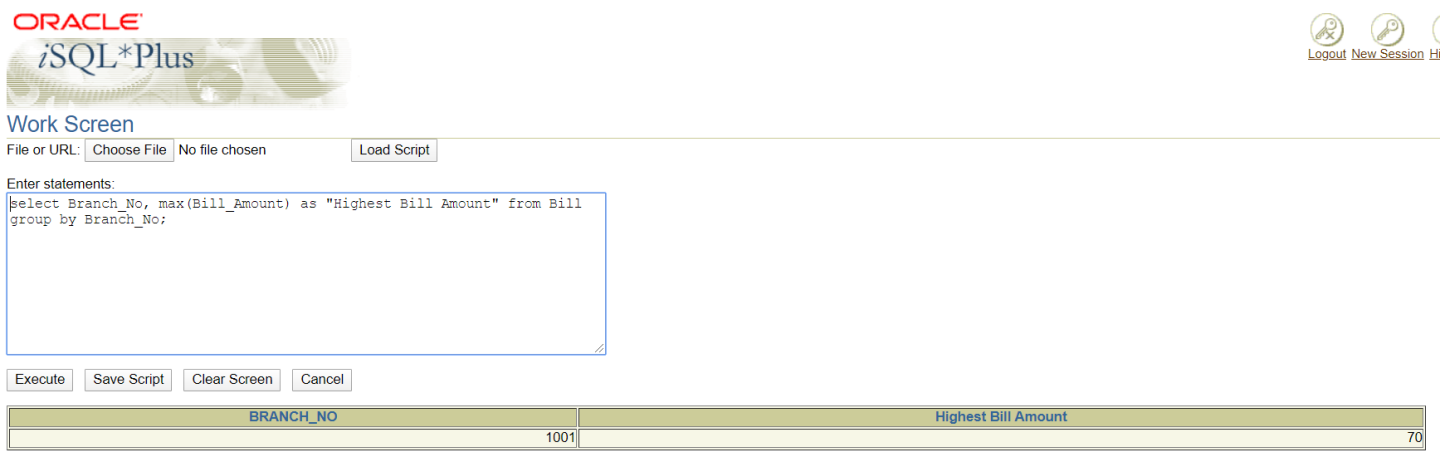
- **Purpose of Query (Join Query):** This query displays the Customer ID and Bill Amount of all the customers whose Payment Type is 'Cash' as below:
- **Query:** select Cust_ID, Bill_Amount from Bill where Bill_No in
(select Bill_No from Mode_of_Payment where Payment_Type = 'Cash');



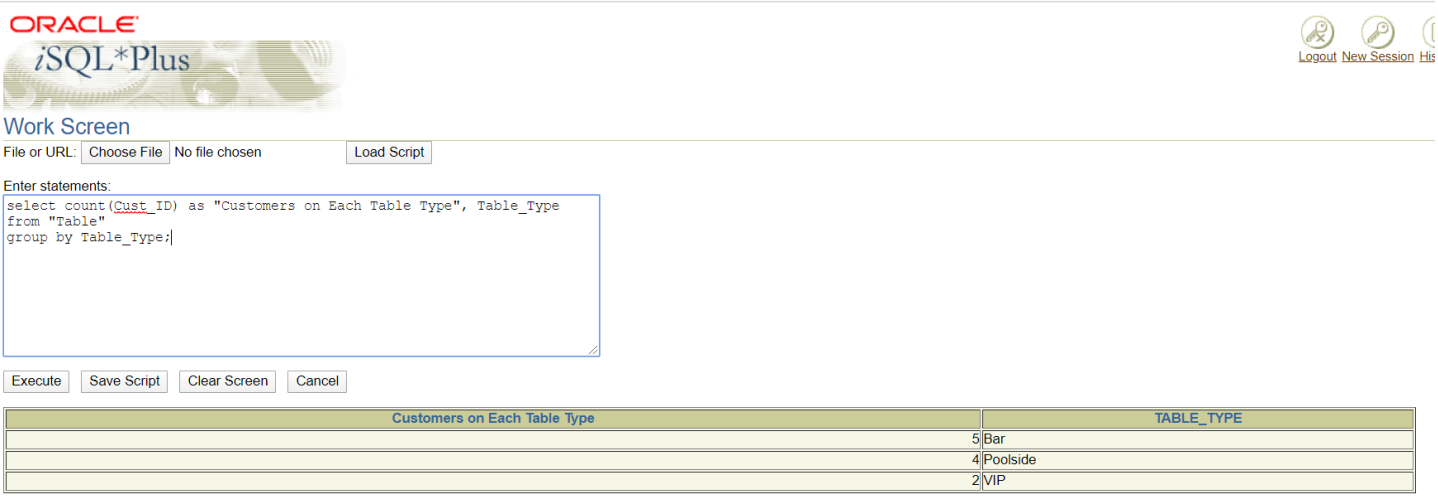
- **Purpose of Query (Group By Query):** This query gives the count of the orders for each Staff ID (How many orders does each Staff ID have) as below:
- **Query:** select Staff_ID, count(Order_No) as "Number of Orders" from "Order"
group by Staff_ID;



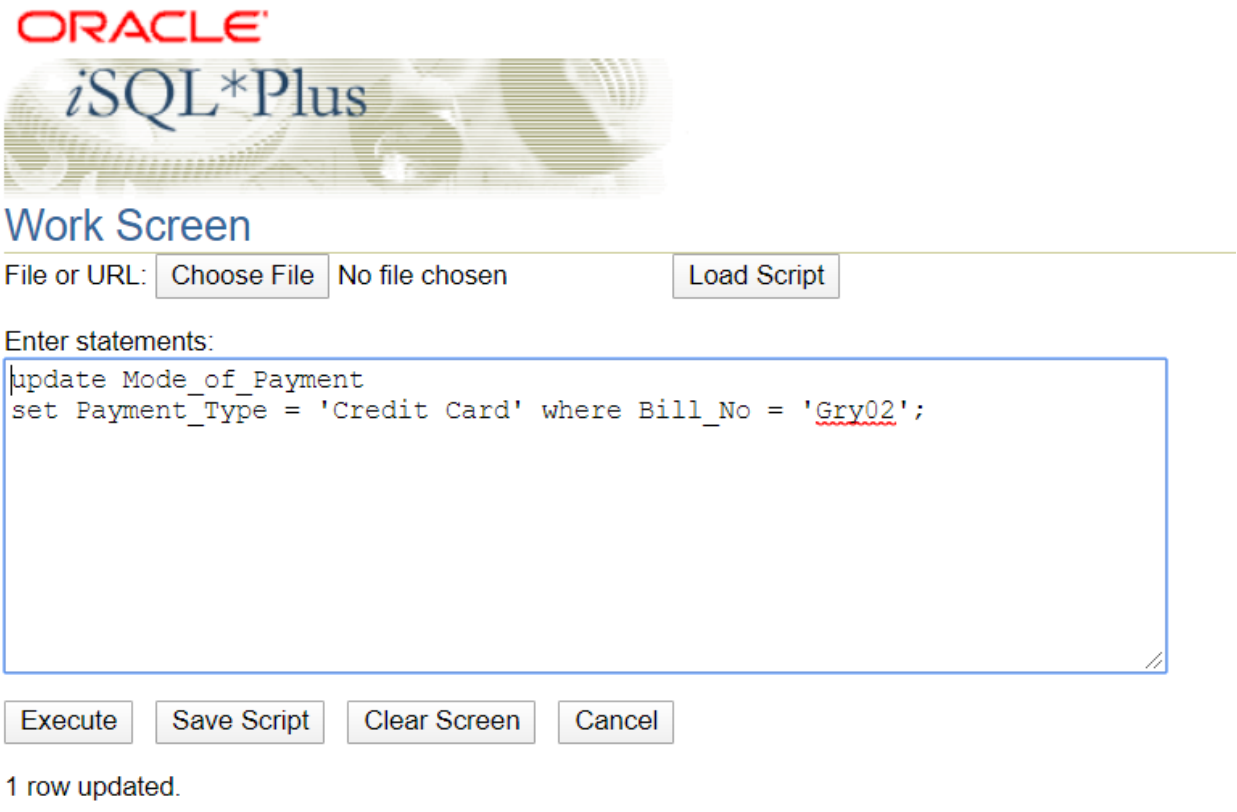
- **Purpose of Query (Group By Query):** This query displays the Highest Bill Amount for the branch as below:
- **Query:** select Branch_No, max(Bill_Amount) as "Highest Bill Amount" from Bill group by Branch_No;



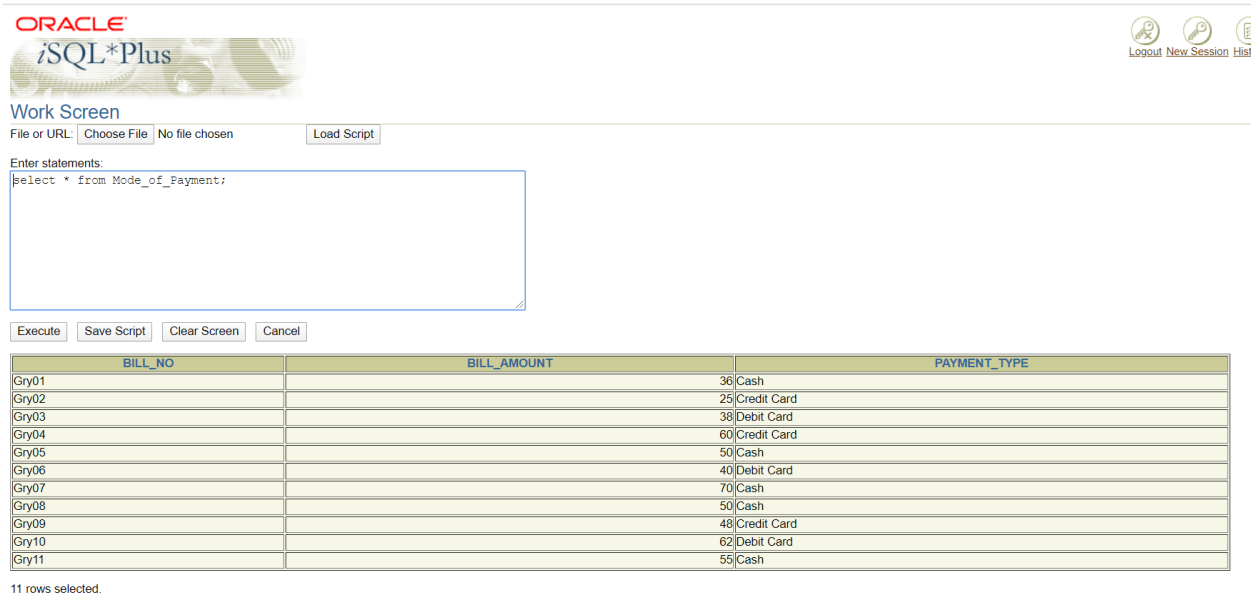
- **Purpose of Query (Group By Query):** This query displays how many Customers are seating for each Table Type as below:
- **Query:** select count(Cust_ID) as "Customers on Each Table Type", Table_Type from "Table" group by Table_Type;



- **SQL Update Queries:**
  - **Purpose of Query:** This query updates the Mode of Payment of a Customer to Credit Card whose Bill No. is 'Gry02' as below:
  - **Query:** update Mode_of_Payment
    set Payment_Type = 'Credit Card' where Bill_No = 'Gry02';

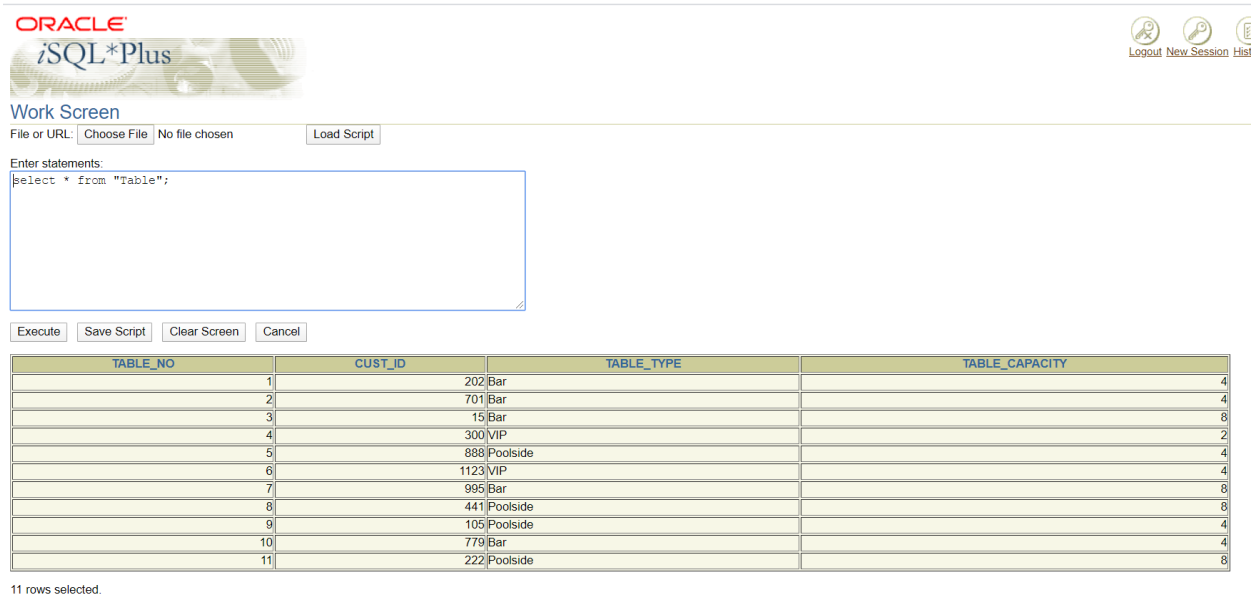


- **Output after Update:**



- **Purpose of Query:** This query updates Table type to 'Bar' whose Table No. is 2 and Table Capacity is 4 as below:
- **Query:** update "Table"
  set Table_Type = 'Bar' where Table_No = 2 and Table_Capacity = 4;

- **Output after Update:**



TABLE_NO	CUST_ID	TABLE_TYPE	TABLE_CAPACITY
1	202	Bar	4
2	701	Bar	4
3	15	Bar	8
4	300	VIP	2
5	888	Poolside	4
6	1123	VIP	4
7	995	Bar	8
8	441	Poolside	8
9	105	Poolside	4
10	779	Bar	4
11	222	Poolside	8

11 rows selected.