King Saud University College of Computer and Information Sciences Computer Science Department	
CSC380 Fundamentals Of Database Systems	Fall 2022

CSC 380 Project

Project Topic

Restaurant Management System

Team:5
Section:66516

Student Name	ID
Sarah Alajlan	442202314
Nouf Alrowais	442201896
Norah alguraishi	442201375
Tarfa albanyan Majd Alqahtani	442201854 442202282

Overview :

Description

Our DB system is simple and comprehensive, it serves the requirements and needs of restaurant management. The database stores the branch's data, which is managed and worked by several employees, their data is also stored, and it manages the system of reservations and orders so that it is easier for the staff to serve customers in the best possible way. Our database system provides each user with data of interest. The restaurant manager is responsible for managing staff, hiring, food costs, inventory management, and more. and he can access and manage all these things through the DB system. Staff who works in the restaurant are responsible for managing, preparing the meals, and serving the customers. Order placement and reservation by staff will be faster and easier. and will help to reduce the conflicts that happen in real-life reservations. Also, it will provide a good structure for the business.

Requirements

- The restaurant has many branches, and Each **BRANCH** has a Location, working hours, id, and phone number.
- **CUSTOMER** is defined by name, customer id, and phone number.
- **STAFF** is defined by Name, ID, phone number, birth date, work hours, role, and monthly salary.
- **ORDER** is defined by an order ID and number of items. Order ID identifies an order uniquely.
- **RECEIPT** is defined by receipt number, total amount, order type (dine-in or dine-out), date, tax, and payment method. For each order, there is a receipt associated with it. If the order is canceled, then the receipt will be deleted.
- **MENU** is defined by menu ID and menu category.
- **TABLE** is defined by table number, number of seats, and availability status.
- Each **ITEM** is defined by name, item price, and item ID.
- Each **SUPPLIER** has an ID, Name, and Phone number.
- Each table has an order. Order contains many items, one order has at least one item.
- The customer places an order. Each customer has at least one order.
- The customer reserves a table. Each customer can reserve at most one table a day and each Reservation has a date and time associated with it.
- The menu has several items, and there are at least 5 items on the menu.
- Suppliers contract with multiple branches and branches can get contracted by multiple suppliers. You must store a start date and an end date for each contract.
- The staff works in a branch, a branch has at least one staff and could handle up to 50 employees, and there's one supervisor who supervises at least one employee.Staff in charge of a table

Entity Relationship Diagram:

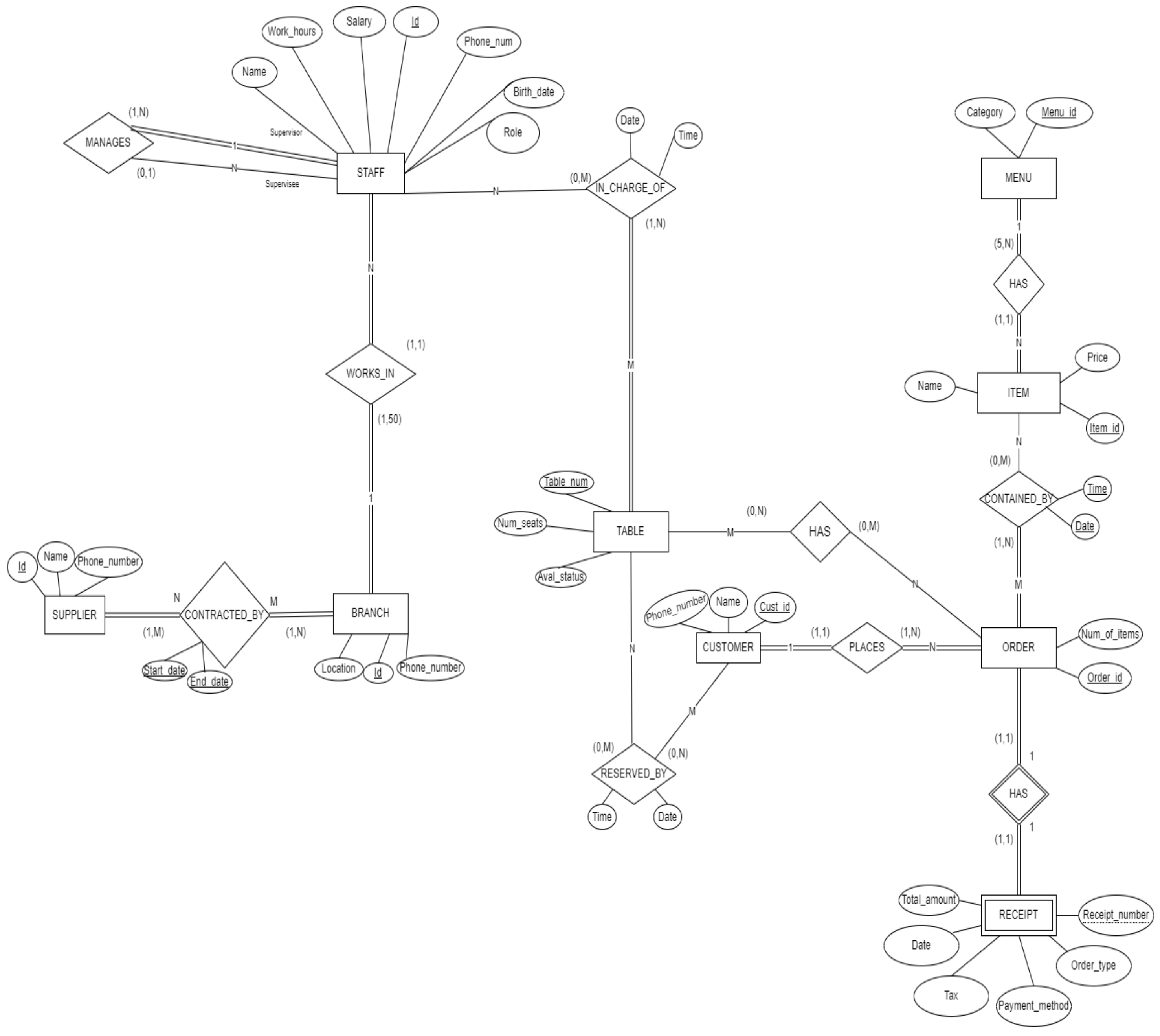


Figure 1: made by drow.io [1]

Relational Schema :

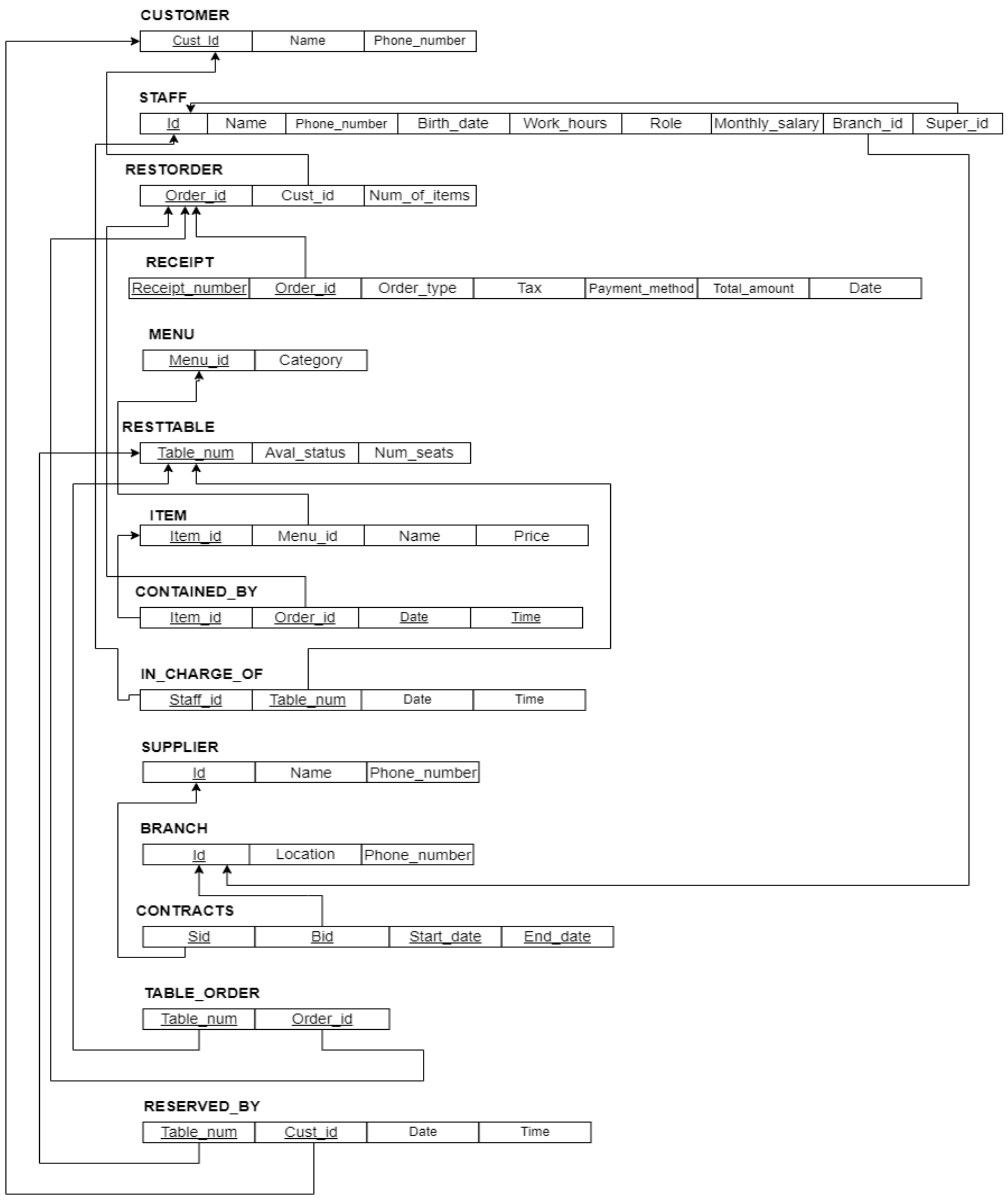


Figure 2: made by drow.io [1]

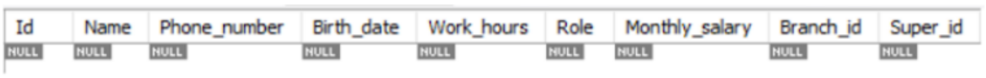
System users:

User group : Manager

This user has a broad view of the database and is responsible for commercial, financial, and administrative matters. They are also responsible for overseeing the daily operations of a restaurant. Their duties include hiring and training restaurant staff, dealing with suppliers solving problems, and creating work schedules for restaurant staff. and they can do their job by performing several operations such as inserting, updating, retrieving, and deleting to manage and maintain the system. **The manager can access the whole database.**

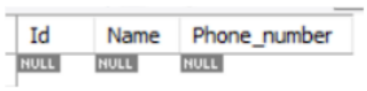
The view :

STAFF



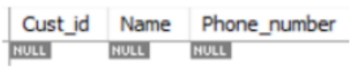
Id	Name	Phone_number	Birth_date	Work_hours	Role	Monthly_salary	Branch_id	Super_id
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SUPPLIER



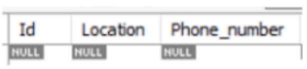
Id	Name	Phone_number
NULL	NULL	NULL

CUSTOMER



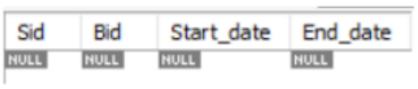
Cust_id	Name	Phone_number
NULL	NULL	NULL

BRANCH



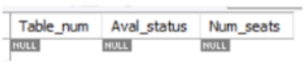
Id	Location	Phone_number
NULL	NULL	NULL

CONTRACTS



Sid	Bid	Start_date	End_date
NULL	NULL	NULL	NULL

RESTTABLE



Table_num	Aval_status	Num_seats
NULL	NULL	NULL

1/Inserting staff
INSERT INTO STAFF
VALUES(‘123498756’,’omer’,’966555555554’,’2002-09-13’,8,’ waiter’,5000,1236,
‘678954321’);

2/Inserting supplier
INSERT INTO SUPPLIER
VALUES(“200000001”,’noura’,’966555555553’);

3/Retrieve staff info
SELECT *
FROM STAFF

```
WHERE id='123498756';
```

4/ Retrieve average monthly salary

```
SELECT AVG(Monthly_salary)
FROM STAFF
WHERE Role='chef';
```

5/ Updating a particular staff salary

```
UPDATE STAFF
SET Monthly_salary=7000
WHERE id=' 987654321';
```

6/ Updating a particular staff phone number

```
UPDATE STAFF
SET Phone_number='966554455111'
WHERE id=' 123498756';
```

7/ Deleting a staff

```
DELETE FROM STAFF
WHERE Id = '123498765';
```

8/ Deleting a supplier

```
DELETE FROM SUPPLIER
WHERE Id = '200000001';
```

9/Retrieving every staff branch location

```
SELECT  S.Id, Location
FROM STAFF AS S , BRANCH AS B
WHERE S.Branch_id = B.Id;
```

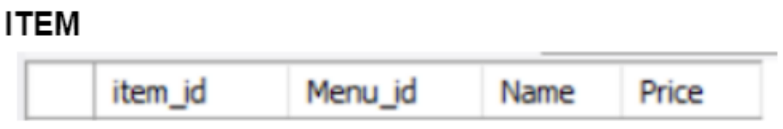
10/ Retrieving every suppliers name corresponding to the branch they supplied

```
SELECT S.Id AS Supplier_Id, B.Id AS  Branch_id, Name AS Supplier_name
FROM CONTRACTS, BRANCH AS B, SUPPLIER AS S
WHERE  B.Id = Bid AND S.Id = Sid;
```

User group: CHEF

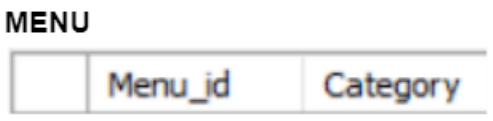
The user is responsible for preparing the order and organizing the menu. So, **he can view the order.**

The view:



ITEM

	item_id	Menu_id	Name	Price



MENU

	Menu_id	Category

1/Insert a category in menu

```
INSERT INTO MENU VALUES ('987655321','main');
```

2/Insert new item

```
INSERT INTO ITEM VALUES('123453780','987655321','burger',100);
```

3/Update: change item name

```
UPDATE  ITEM
SET Name = 'white chocolate'
WHERE item_id = '123456780';
```

4/Update the item price

```
UPDATE ITEM
SET price = 40
WHERE item_id = '123456780';
```

5/ Retrieving every items price

```
SELECT Name as item_name , price
FROM ITEM;
```

6/Retrieve menus category

```
2/ SELECT Category
FROM Menu;
```

7/Retrieve number of items in one category

```
SELECT Category,COUNT(*) as count_items_inmenu FROM ITEM NATURAL JOIN
MENU GROUP BY Category;
```

8/Retrieve items category

```
SELECT Category, Name as item_name FROM ITEM NATURAL JOIN MENU;
```

1/Delete an item
1/ DELETE FROM ITEM
WHERE item_id = '123456780';

2/ Delete a MENU
DELETE FROM MENU
WHERE Menu_id = '987654322';

Implementation:

Implementing The Database using a DBMS (MySQL)

Staff

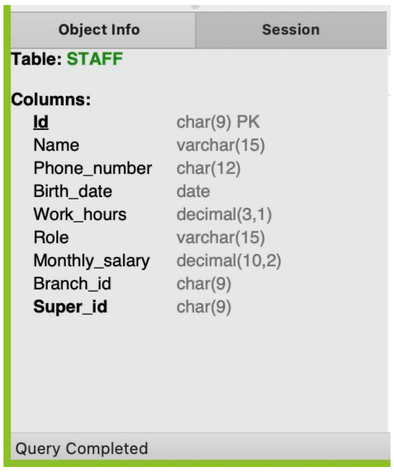

Object Info | Session

Table: STAFF

Columns:

Id	char(9) PK
Name	varchar(15)
Phone_number	char(12)
Birth_date	date
Work_hours	decimal(3,1)
Role	varchar(15)
Monthly_salary	decimal(10,2)
Branch_id	char(9)
Super_id	char(9)

Query Completed

Branch

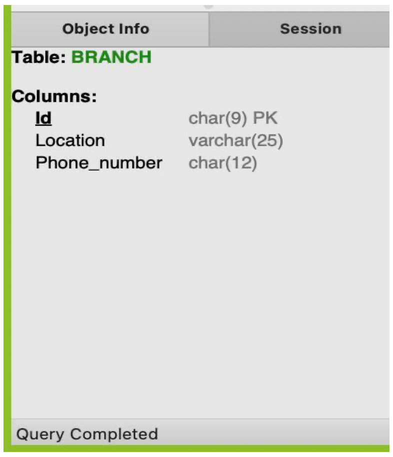

Object Info | Session

Table: BRANCH

Columns:

Id	char(9) PK
Location	varchar(25)
Phone_number	char(12)

Query Completed

Supplier

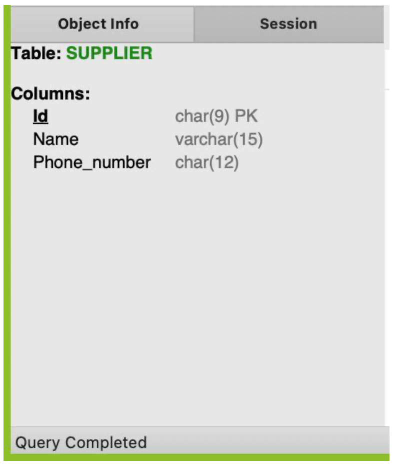

Object Info | Session

Table: SUPPLIER

Columns:

Id	char(9) PK
Name	varchar(15)
Phone_number	char(12)

Query Completed

RestTable

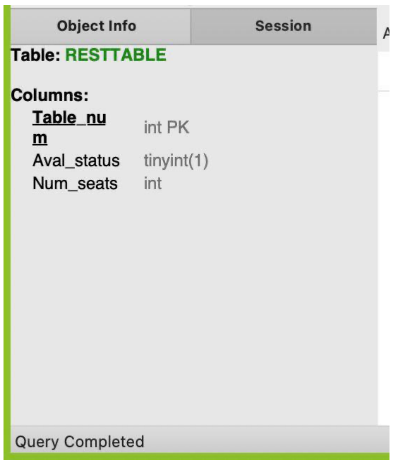

Object Info | Session

Table: RESTTABLE

Columns:

Table_num	int PK
Aval_status	tinyint(1)
Num_seats	int

Query Completed

Table_order

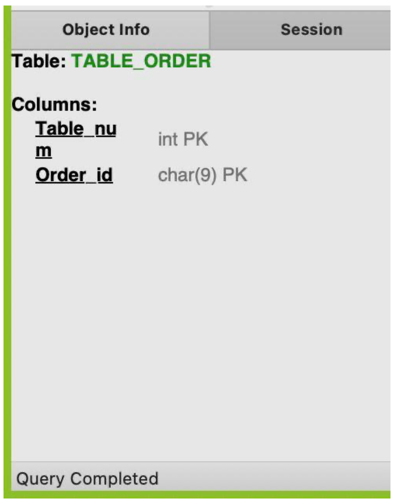

Object Info | Session

Table: TABLE_ORDER

Columns:

Table_num	int PK
Order_id	char(9) PK

Query Completed

Reserved_by

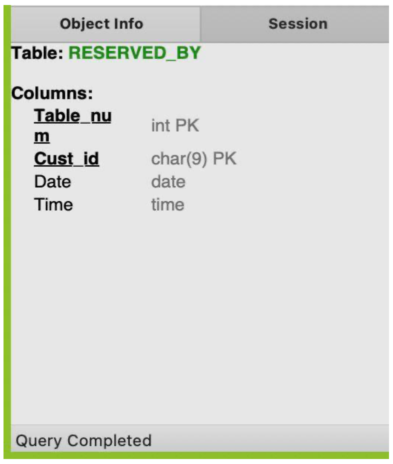

Object Info | Session

Table: RESERVED_BY

Columns:

Table_num	int PK
Cust_id	char(9) PK
Date	date
Time	time

Query Completed

Menu

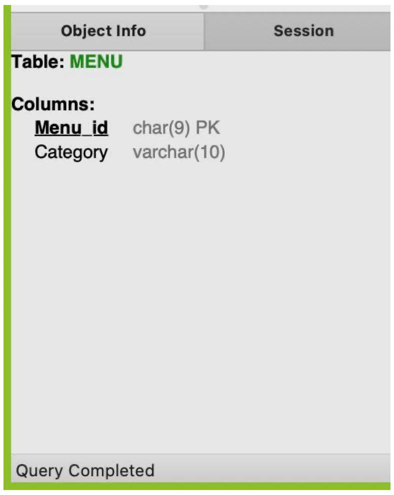

Object Info | Session

Table: MENU

Columns:

Menu_id	char(9) PK
Category	varchar(10)

Query Completed

Item

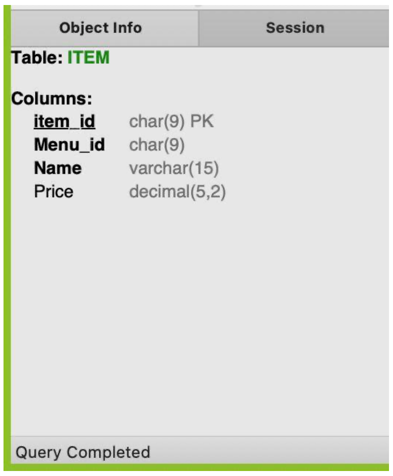

Object Info | Session

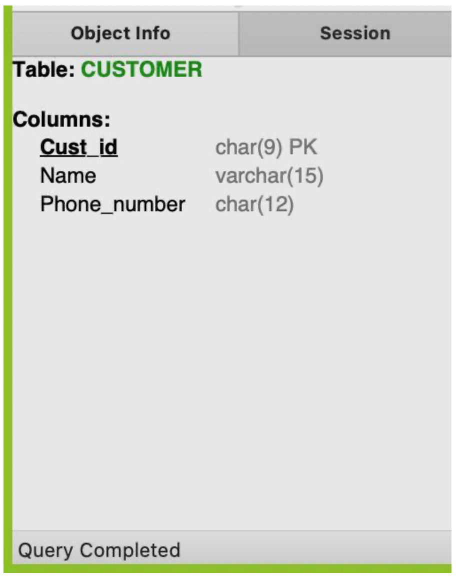
Table: ITEM

Columns:

item_id	char(9) PK
Menu_id	char(9)
Name	varchar(15)
Price	decimal(5,2)

Query Completed

Customer



Object Info | Session

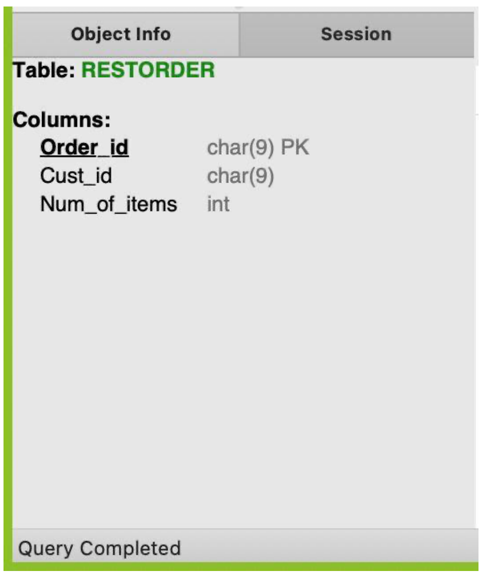
Table: CUSTOMER

Columns:

Cust_id	char(9) PK
Name	varchar(15)
Phone_number	char(12)

Query Completed

RestOrder



Object Info | Session

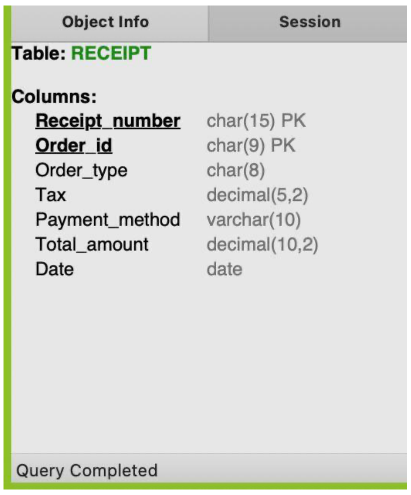
Table: RESTORDER

Columns:

Order_id	char(9) PK
Cust_id	char(9)
Num_of_items	int

Query Completed

Receipt



Object Info | Session

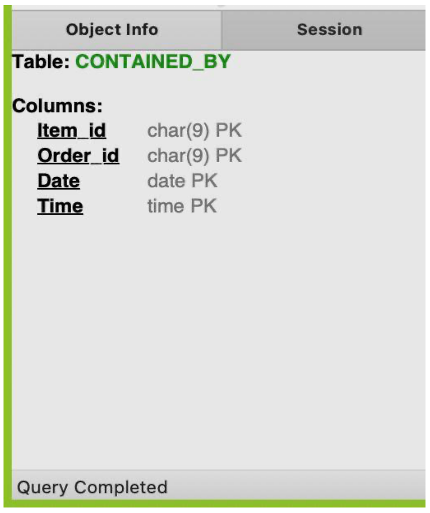
Table: RECEIPT

Columns:

Receipt_number	char(15) PK
Order_id	char(9) PK
Order_type	char(8)
Tax	decimal(5,2)
Payment_method	varchar(10)
Total_amount	decimal(10,2)
Date	date

Query Completed

Contained_by



Object Info | Session

Table: CONTAINED_BY

Columns:

Item_id	char(9) PK
Order_id	char(9) PK
Date	date PK
Time	time PK

Query Completed

Contracts

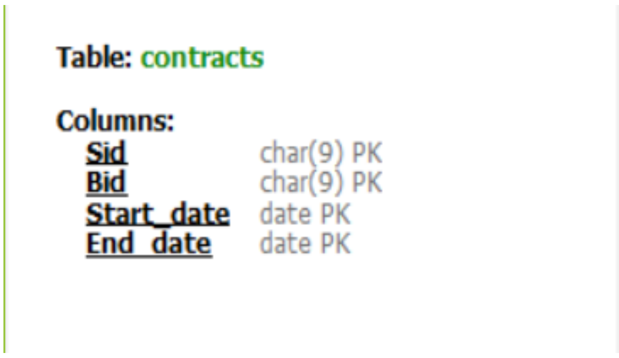


Table: contracts

Columns:

Sid	char(9) PK
Bid	char(9) PK
Start_date	date PK
End_date	date PK

In_chage_of

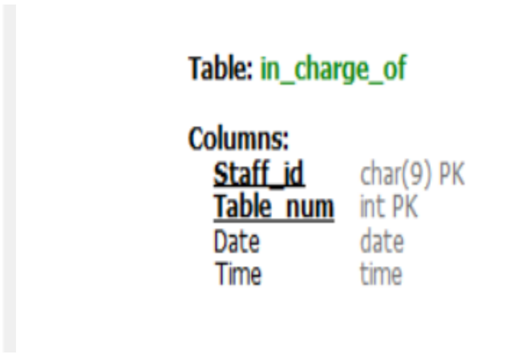


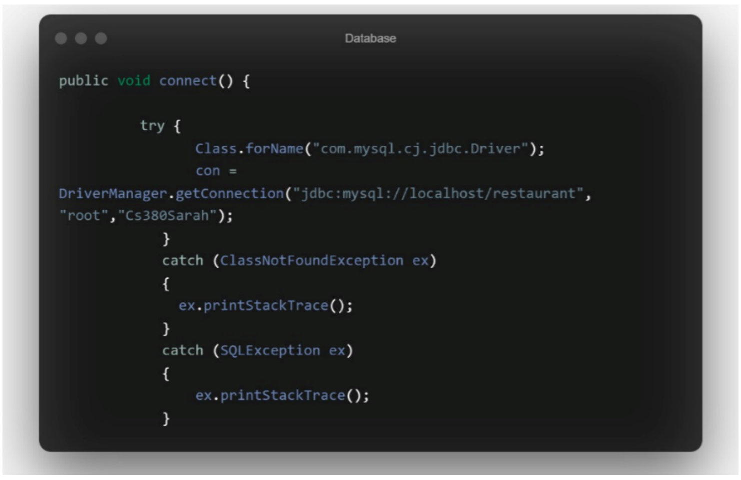
Table: in_charge_of

Columns:

Staff_id	char(9) PK
Table_num	int PK
Date	date
Time	time

Connection part:

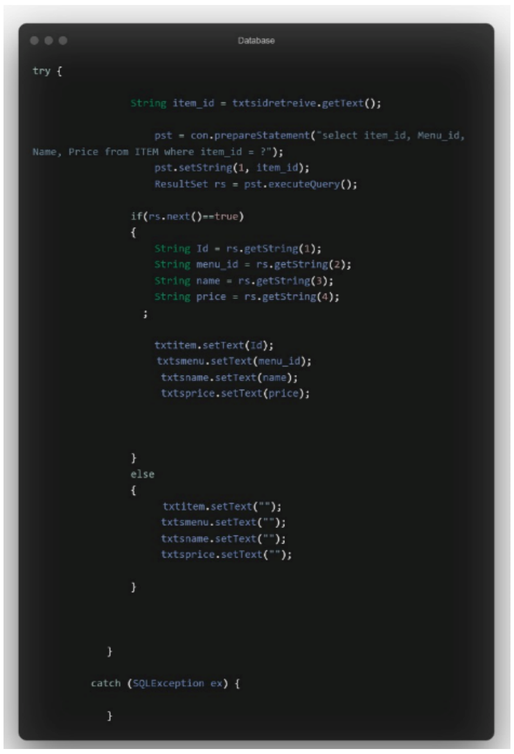
In this code, we have linked the database to eclipse



```
public void connect() {

          try {
                 Class.forName("com.mysql.cj.jdbc.Driver");
                 con =
DriverManager.getConnection("jdbc:mysql://localhost/restaurant",
"root","Cs380Sarah");
             }
             catch (ClassNotFoundException ex)
             {
               ex.printStackTrace();
             }
             catch (SQLException ex)
             {
                 ex.printStackTrace();
             }
```

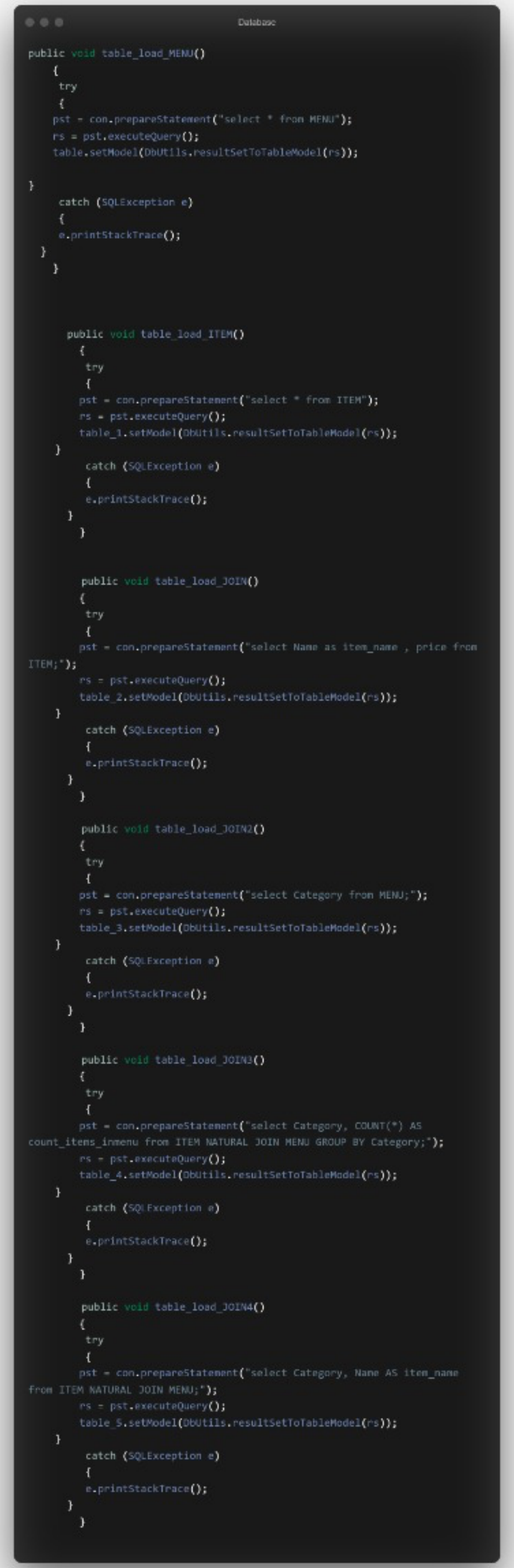
Some parts of code :

We showed some basic parts of the code that we created to be the Chef interface, which is repeated in the manager interface code.



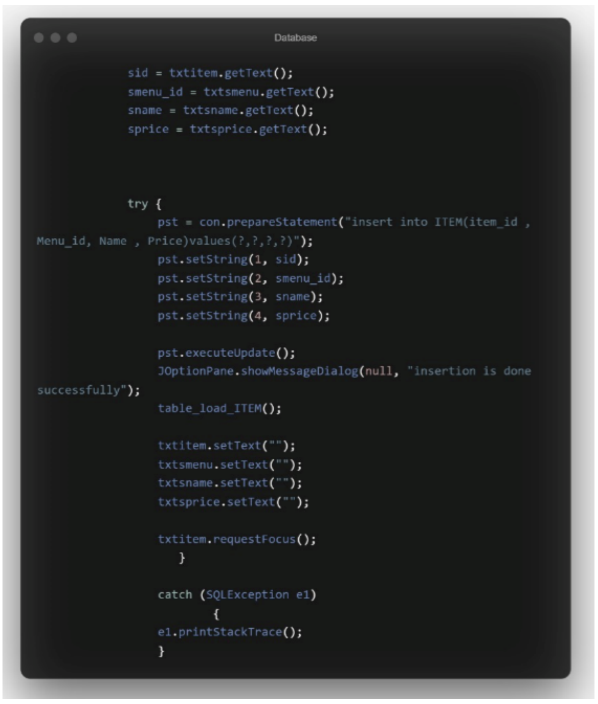
```
try {

                String item_id = txtsidretreive.getText();

                    pst = con.prepareStatement("select item_id, Menu_id,
Name, Price from ITEM where item_id = ?");
                    pst.setString(1, item_id);
                    ResultSet rs = pst.executeQuery();

                if(rs.next()==true)
                {
                    String Id = rs.getString(1);
                    String menu_id = rs.getString(2);
                    String name = rs.getString(3);
                    String price = rs.getString(4);
                  ;

                    txtitem.setText(Id);
                     txtsmenu.setText(menu_id);
                      txtsname.setText(name);
                      txtsprice.setText(price);



                }
                else
                {
                       txtitem.setText("");
                      txtsmenu.setText("");
                      txtsname.setText("");
                      txtsprice.setText("");

                }



            }

         catch (SQLException ex) {

            }
```

In this part, we did the data retrieval process using the prepareStatment method which in turn executes query.



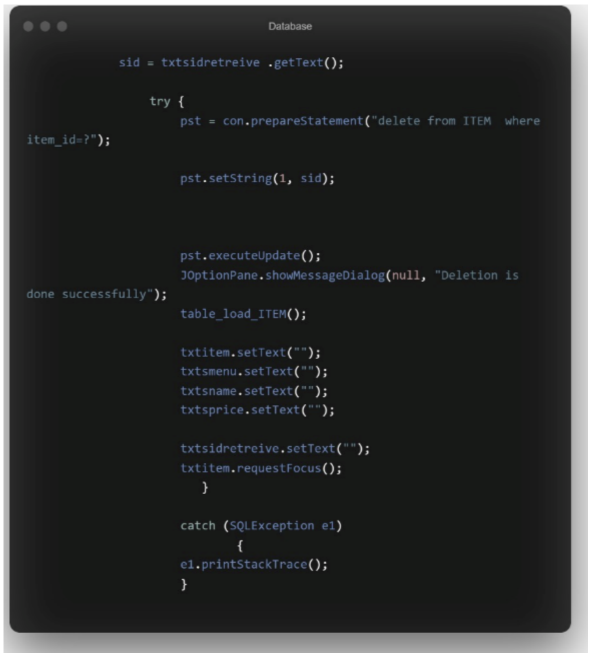
```
public void table_load_MENU()
    {
     try
     {
    pst = con.prepareStatement("select * from MENU");
    rs = pst.executeQuery();
    table.setModel(DbUtils.resultSetToTableModel(rs));


}
     catch (SQLException e)
     {
     e.printStackTrace();
  }
    }



      public void table_load_ITEM()
        {
         try
         {
        pst = con.prepareStatement("select * from ITEM");
        rs = pst.executeQuery();
        table_1.setModel(DbUtils.resultSetToTableModel(rs));
    }
         catch (SQLException e)
         {
         e.printStackTrace();
      }
        }


        public void table_load_JOIN()
        {
         try
         {
        pst = con.prepareStatement("select Name as item_name , price from
ITEM;");
        rs = pst.executeQuery();
        table_2.setModel(DbUtils.resultSetToTableModel(rs));
    }
         catch (SQLException e)
         {
         e.printStackTrace();
      }
        }

        public void table_load_JOIN2()
        {
         try
         {
        pst = con.prepareStatement("select Category from MENU;");
        rs = pst.executeQuery();
        table_3.setModel(DbUtils.resultSetToTableModel(rs));
    }
         catch (SQLException e)
         {
         e.printStackTrace();
      }
        }

        public void table_load_JOIN3()
        {
         try
         {
        pst = con.prepareStatement("select Category, COUNT(*) AS
count_items_inmenu from ITEM NATURAL JOIN MENU GROUP BY Category;");
        rs = pst.executeQuery();
        table_4.setModel(DbUtils.resultSetToTableModel(rs));
    }
         catch (SQLException e)
         {
         e.printStackTrace();
      }
        }

        public void table_load_JOIN4()
        {
         try
         {
        pst = con.prepareStatement("select Category, Name AS item_name
from ITEM NATURAL JOIN MENU;");
        rs = pst.executeQuery();
        table_5.setModel(DbUtils.resultSetToTableModel(rs));
    }
         catch (SQLException e)
         {
         e.printStackTrace();
      }
        }
```

In this part we have loaded the item and menu tables by
table_load_ITEM()
table_load_MENU()
this methods used the prepareStatment method to execute the queries and this is what we did with the rest of the tables .



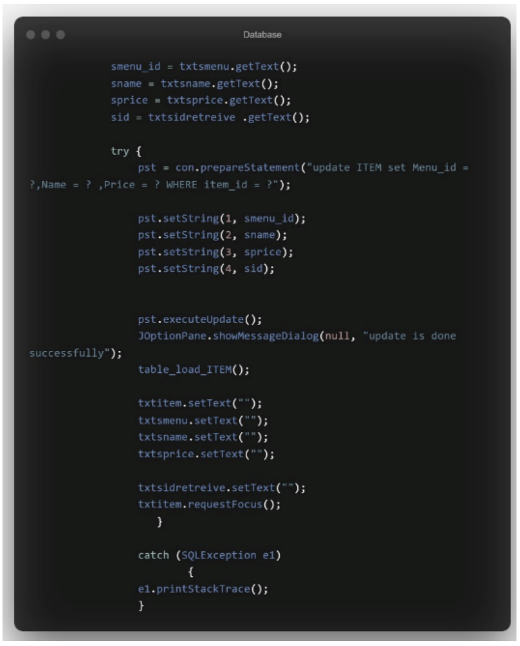
```
            sid = txtitem.getText();
            smenu_id = txtsmenu.getText();
            sname = txtsname.getText();
            sprice = txtsprice.getText();



            try {
                pst = con.prepareStatement("insert into ITEM(item_id ,
Menu_id, Name , Price)values(?,?,?,?)");
                pst.setString(1, sid);
                pst.setString(2, smenu_id);
                pst.setString(3, sname);
                pst.setString(4, sprice);

                pst.executeUpdate();
                JOptionPane.showMessageDialog(null, "insertion is done
successfully");
                table_load_ITEM();

                txtitem.setText("");
                txtsmenu.setText("");
                txtsname.setText("");
                txtsprice.setText("");

                txtitem.requestFocus();
                   }

                catch (SQLException e1)
                        {
                e1.printStackTrace();
                }
```

In this part, we did the insertion process and showed a message if the operation was successfully completed by JOptionPane.showMessageDialog()



```
            sid = txtsidretreive .getText();

                try {
                    pst = con.prepareStatement("delete from ITEM  where
item_id=?");

                    pst.setString(1, sid);



                    pst.executeUpdate();
                    JOptionPane.showMessageDialog(null, "Deletion is
done successfully");
                    table_load_ITEM();

                    txtitem.setText("");
                    txtsmenu.setText("");
                    txtsname.setText("");
                    txtsprice.setText("");

                    txtsidretreive.setText("");
                    txtitem.requestFocus();
                       }

                    catch (SQLException e1)
                            {
                    e1.printStackTrace();
                    }
```

In this part, we did the deletion process and showed a message if the operation was successfully completed by JOptionPane.showMessageDialog()

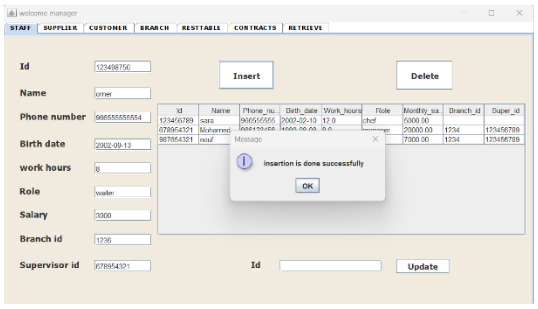


```
            smenu_id = txtsmenu.getText();
            sname = txtsname.getText();
            sprice = txtsprice.getText();
            sid = txtsidretreive .getText();

            try {
                pst = con.prepareStatement("update ITEM set Menu_id =
?,Name = ? ,Price = ? WHERE item_id = ?");

                pst.setString(1, smenu_id);
                pst.setString(2, sname);
                pst.setString(3, sprice);
                pst.setString(4, sid);


                pst.executeUpdate();
                JOptionPane.showMessageDialog(null, "update is done
successfully");
                table_load_ITEM();

                txtitem.setText("");
                txtsmenu.setText("");
                txtsname.setText("");
                txtsprice.setText("");

                txtsidretreive.setText("");
                txtitem.requestFocus();
                   }

                catch (SQLException e1)
                        {
                e1.printStackTrace();
                }
```

In this part, we did the update process and showed a message if the operation was successfully completed by JOptionPane.showMessageDialog()

Manager interface:

Inserting chef:



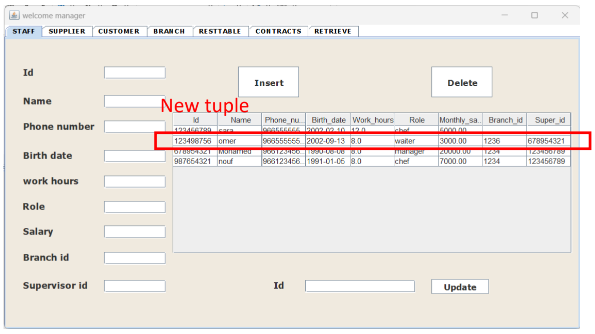
Id	123498756
Name	omer
Phone number	966555555554
Birth date	2002-09-13
work hours	8
Role	waiter
Salary	3000
Branch id	1236
Supervisor id	678954321

Id	Name	Phone_nu...	Birth_date	Work_hours	Role	Monthly_sa...	Branch_id	Super_id
123456789	sara	966555555...	2002-02-10	12.0	chef	5000.00		
678954321	Mohamed	[illegible]	[illegible]	[illegible]	[illegible]	20000.00	1234	123456789
987654321	nouf	[illegible]	[illegible]	[illegible]	[illegible]	7000.00	1234	123456789

insertion is done successfully

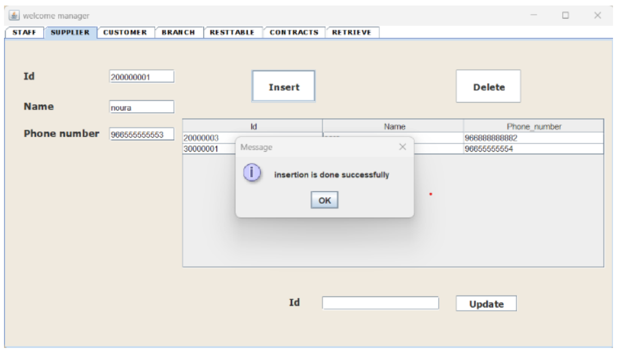
Id

After insertion:



Id	Name	Phone_nu...	Birth_date	Work_hours	Role	Monthly_sa...	Branch_id	Super_id
123456789	sara	966555555	2002-02-10	12.0	chef	5000.00		
123498756	omer	966555555...	2002-09-13	8.0	waiter	3000.00	1236	678954321
678954321	Mohamed	966123456...	1990-08-08	8.0	manager	20000.00	1234	123456789
987654321	nouf	966123456...	1991-01-05	8.0	chef	7000.00	1234	123456789

Inserting supplier:



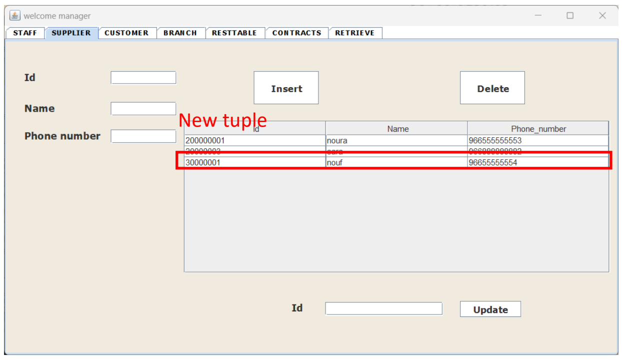
Id	200000001
Name	noura
Phone number	966555555553

Id	Name	Phone_number
20000003	[illegible]	966888888882
30000001	[illegible]	96655555554

insertion is done successfully

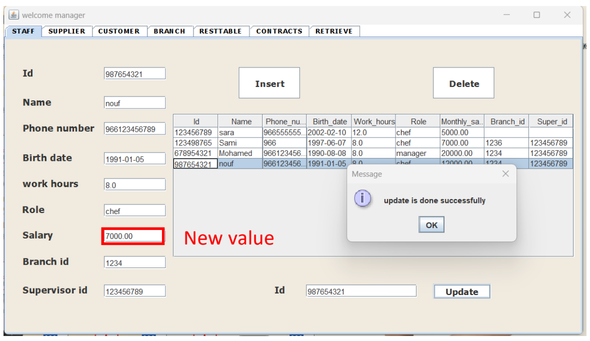
Id

After insertion:



Id	Name	Phone_number
200000001	noura	966555555553
20000003	sara	966888888882
30000001	nouf	96655555554

Updating a particular staff salary:



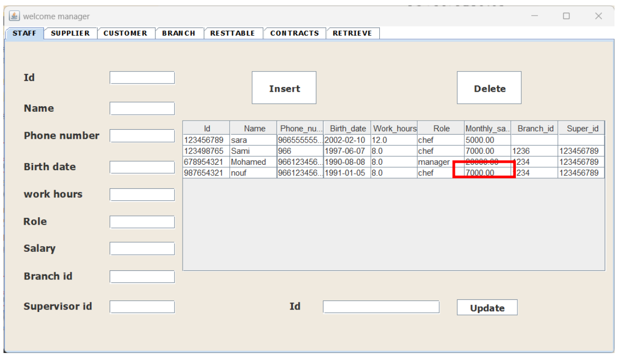
Id	987654321
Name	nouf
Phone number	966123456789
Birth date	1991-01-05
work hours	8.0
Role	chef
Salary	7000.00
Branch id	1234
Supervisor id	123456789

Id	Name	Phone_nu...	Birth_date	Work_hours	Role	Monthly_sa...	Branch_id	Super_id
123456789	sara	966555555...	2002-02-10	12.0	chef	5000.00		
123498765	Sami	966	1997-06-07	8.0	chef	7000.00	1236	123456789
678954321	Mohamed	966123456...	1990-08-08	8.0	manager	20000.00	1234	123456789
987654321	nouf	966123456...	1991-01-05	8.0	chef	12000.00	1234	123456789

update is done successfully

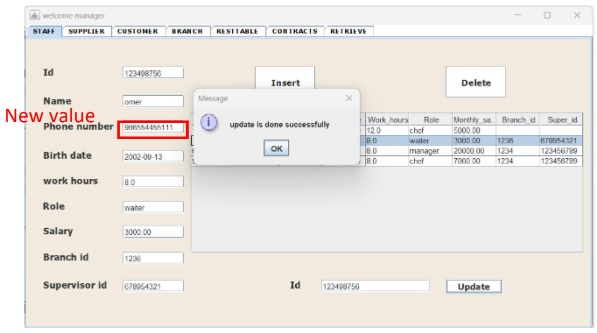
Id 987654321

After the update:



Id	Name	Phone_nu...	Birth_date	Work_hours	Role	Monthly_sa...	Branch_id	Super_id
123456789	sara	966555555...	2002-02-10	12.0	chef	5000.00		
123498765	Sami	966	1997-06-07	8.0	chef	7000.00	1236	123456789
678954321	Mohamed	966123456...	1990-08-08	8.0	manager	20000.00	1234	123456789
987654321	nouf	966123456...	1991-01-05	8.0	chef	7000.00	1234	123456789

Updating a particular staff phone number :



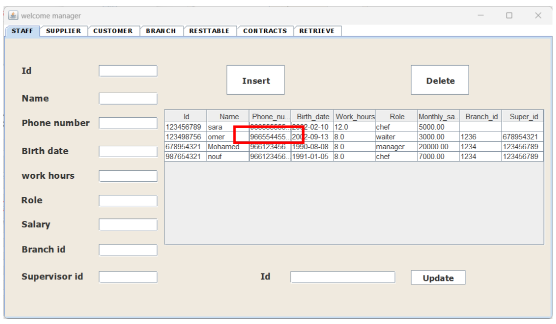
Id	123498756
Name	omer
Phone number	966554455111
Birth date	2002-09-13
work hours	8.0
Role	waiter
Salary	3000.00
Branch id	1236
Supervisor id	678954321

Work_hours	Role	Monthly_sa...	Branch_id	Super_id
12.0	chef	5000.00		
8.0	waiter	3000.00	1236	678954321
8.0	manager	20000.00	1234	123456789
8.0	chef	7000.00	1234	123456789

update is done successfully

Id 123498756

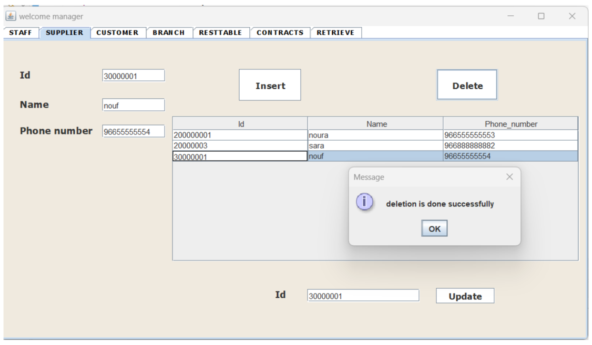
After the update:



Id	Name	Phone_nu...	Birth_date	Work_hours	Role	Monthly_sa...	Branch_id	Super_id
123456789	sara	966555555...	2002-02-10	12.0	chef	5000.00		
123498756	omer	966554455...	2002-09-13	8.0	waiter	3000.00	1236	678954321
678954321	Mohamed	966123456...	1990-08-08	8.0	manager	20000.00	1234	123456789
987654321	nouf	966123456...	1991-01-05	8.0	chef	7000.00	1234	123456789

Deleting a supplier:

After Deletion:

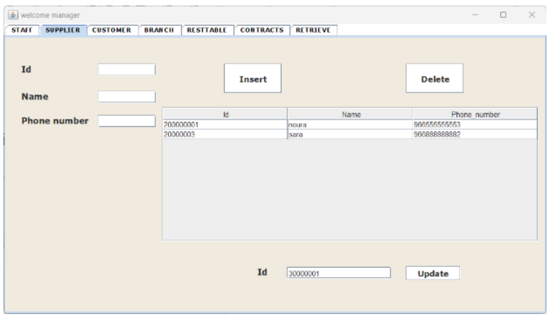


Id	30000001
Name	nouf
Phone number	96655555554

Id	Name	Phone_number
200000001	noura	966555555553
20000003	sara	966888888882
30000001	nouf	96655555554

deletion is done successfully

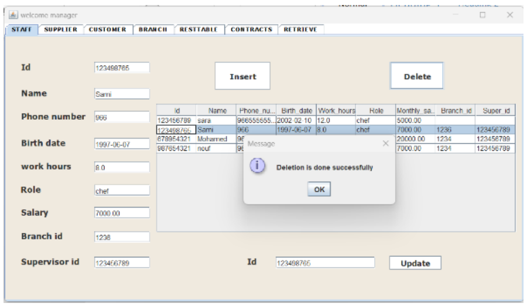
Id 30000001



Id	Name	Phone_number
200000001	noura	966555555553
20000003	sara	966888888882

Id 30000001

Deleting a staff:



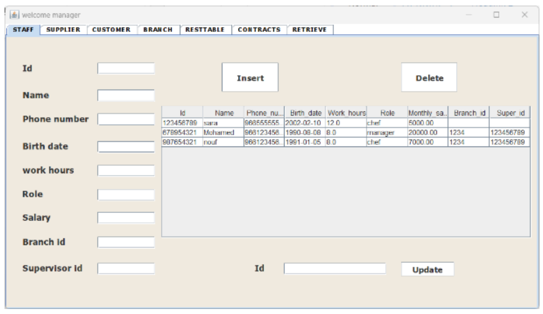
Id	123498765
Name	Sami
Phone number	966
Birth date	1997-06-07
work hours	8.0
Role	chef
Salary	7000.00
Branch id	1236
Supervisor id	123456789

Id	Name	Phone_nu...	Birth_date	Work_hours	Role	Monthly_sa...	Branch_id	Super_id
123456789	sara	966555555...	2002-02-10	12.0	chef	5000.00		
123498765	Sami	966	1997-06-07	8.0	chef	7000.00	1236	123456789
678954321	Mohamed	96				20000.00	1234	123456789
987654321	nouf	96				7000.00	1234	123456789

Deletion is done successfully

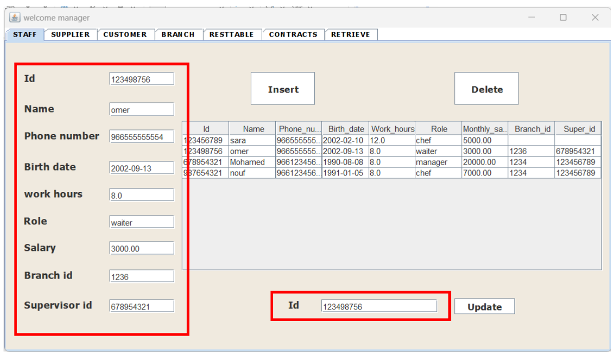
Id 123498765

After Deletion:



Id	Name	Phone_nu...	Birth_date	Work_hours	Role	Monthly_sa...	Branch_id	Super_id
123456789	sara	966555555...	2002-02-10	12.0	chef	5000.00		
678954321	Mohamed	966123456...	1990-08-08	8.0	manager	20000.00	1234	123456789
987654321	nouf	966123456...	1991-01-05	8.0	chef	7000.00	1234	123456789

Retrieve staff info:



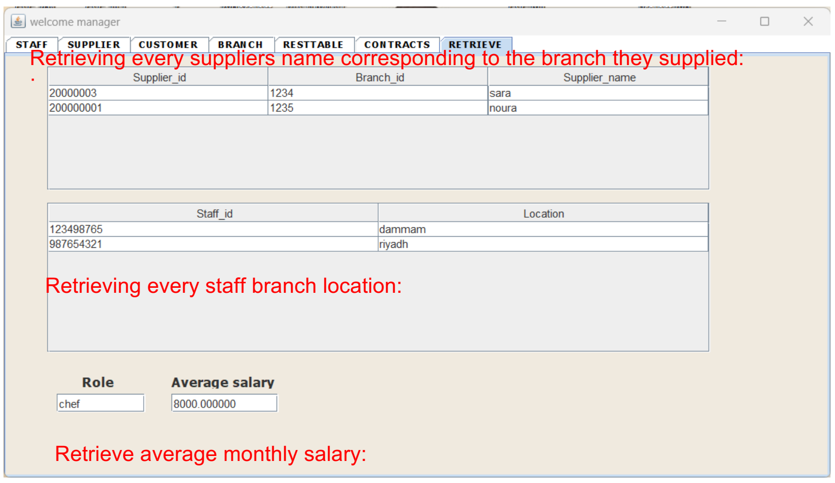
Id	123498756
Name	omer
Phone number	966555555554
Birth date	2002-09-13
work hours	8.0
Role	waiter
Salary	3000.00
Branch id	1236
Supervisor id	678954321

Id	Name	Phone_nu...	Birth_date	Work_hours	Role	Monthly_sa...	Branch_id	Super_id
123456789	sara	966555555...	2002-02-10	12.0	chef	5000.00		
123498756	omer	966555555...	2002-09-13	8.0	waiter	3000.00	1236	678954321
678954321	Mohamed	966123456...	1990-08-08	8.0	manager	20000.00	1234	123456789
987654321	nouf	966123456...	1991-01-05	8.0	chef	7000.00	1234	123456789

Id 123498756

By entering the id, the employee's information is automatically output on the other side.

Retrieve average monthly salary
Retrieving every staff branch location
Retrieving every suppliers name corresponding to the branch they supplied:



Retrieving every suppliers name corresponding to the branch they supplied:

Supplier_id	Branch_id	Supplier_name
20000003	1234	sara
200000001	1235	noura

Staff_id	Location
123498765	dammam
987654321	riyadh

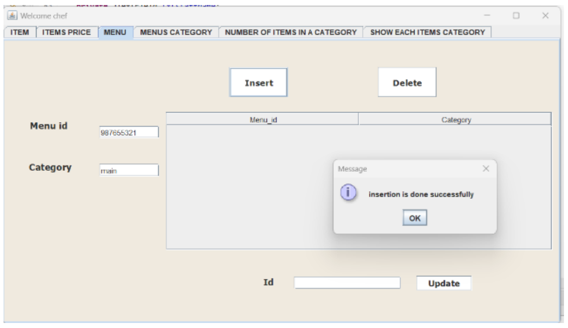
Retrieving every staff branch location:

Role	Average salary
chef	8000.000000

Retrieve average monthly salary:

Chef interface:

Insert a category in menu: **After insertion:**



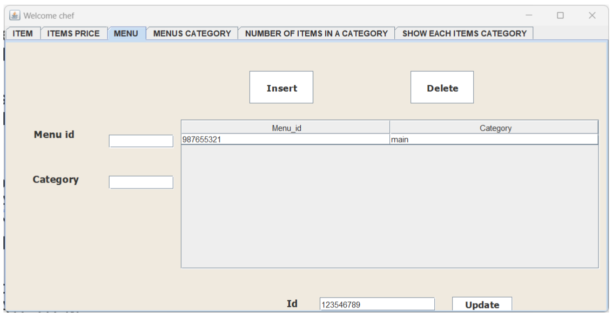
Menu id: 987655321

Category: main

Menu_id	Category

Message: insertion is done successfully

Id:



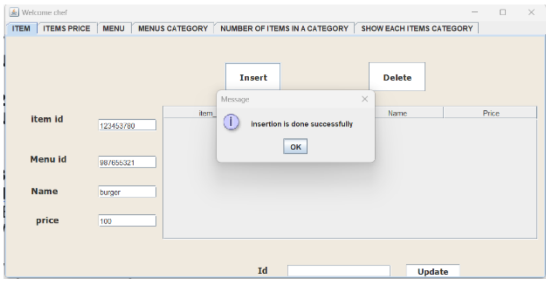
Menu id:

Category:

Menu_id	Category
987655321	main

Id: 123546789

Insert new item:



item id: 123453780

Menu id: 987655321

Name: burger

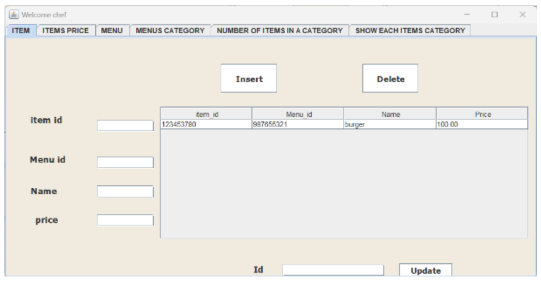
price: 100

item_id	Menu_id	Name	Price

Message: insertion is done successfully

Id:

After insertion:



item id:

Menu id:

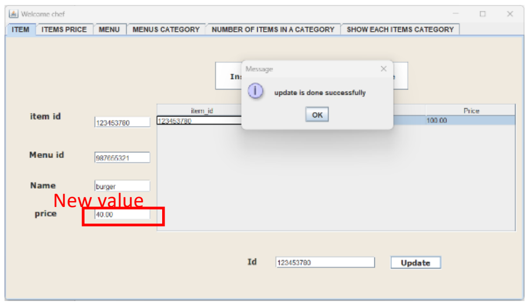
Name:

price:

item_id	Menu_id	Name	Price
123453780	987655321	burger	100.00

Id:

Update the item price:



item id: 123453780

Menu id: 987655321

Name: burger

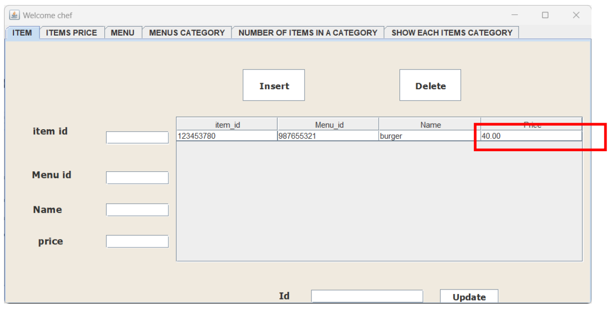
price: 40.00

item_id			Price
123453780			100.00

Message: update is done successfully

Id: 123453780

After the update:



item id:

Menu id:

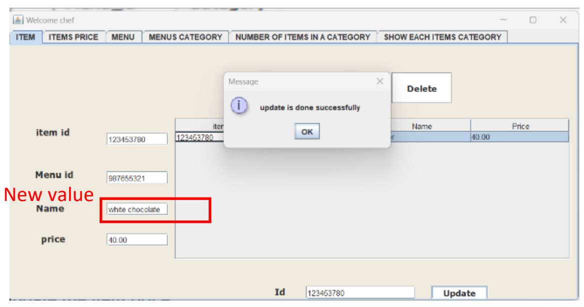
Name:

price:

item_id	Menu_id	Name	Price
123453780	987655321	burger	40.00

Id:

Update: change item name:



Message

update is done successfully

item id: 123453780

Menu id: 987655321

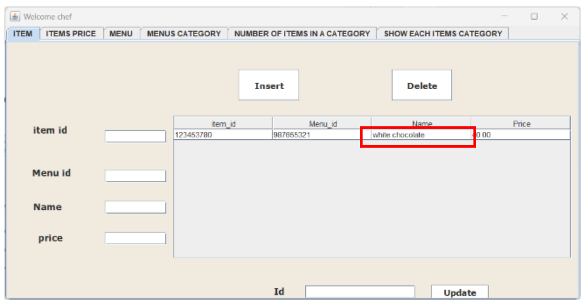
Name: white chocolate

price: 40.00

Id: 123453780

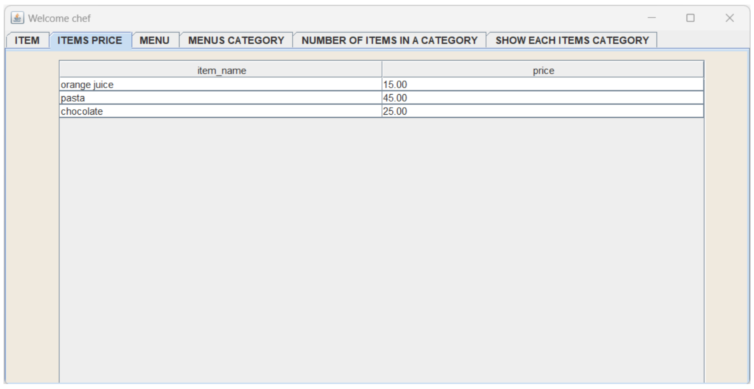
iter	Name	Price
123453780	r	40.00

After the update:



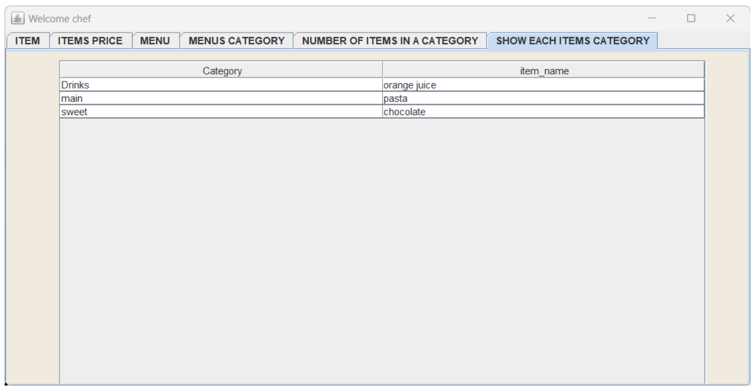
item_id	Menu_id	Name	Price
123453780	987655321	white chocolate	40.00

Retrieving every items price:



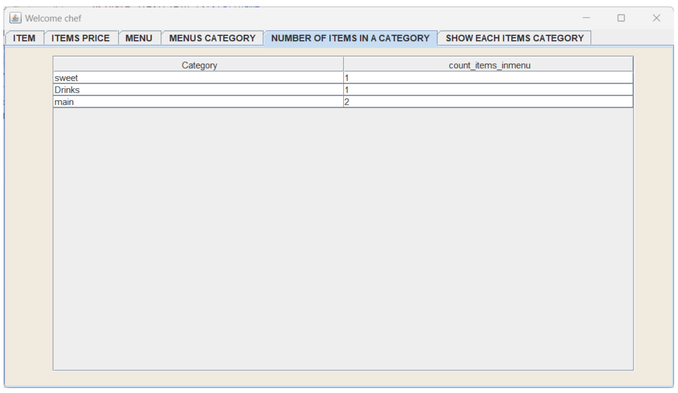
item_name	price
orange juice	15.00
pasta	45.00
chocolate	25.00

Retrieve items category:



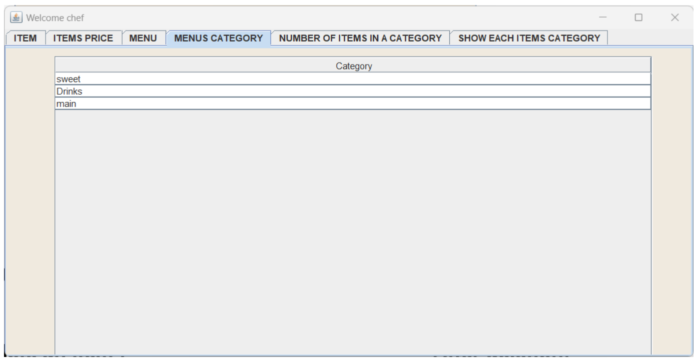
Category	item_name
Drinks	orange juice
main	pasta
sweet	chocolate

Retrieve number of items in one category:



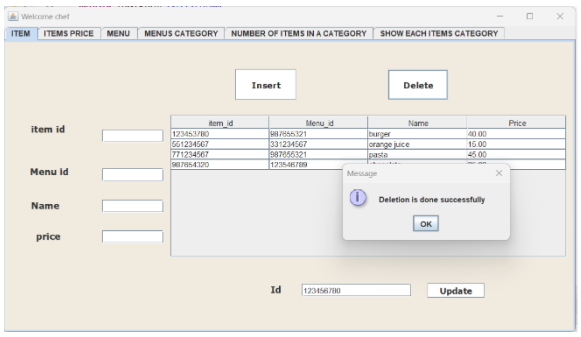
Category	count_items_inmenu
sweet	1
Drinks	1
main	2

Retrieve menus category:



Category
sweet
Drinks
main

Delete an item :

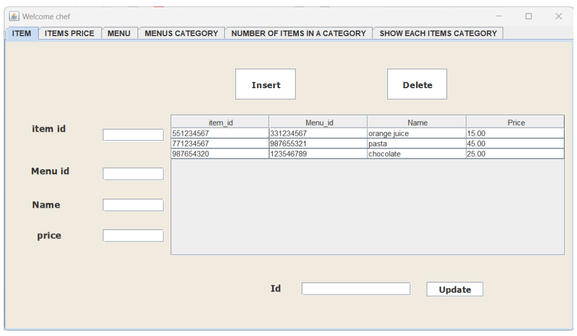


item_id	Menu_id	Name	Price
123453780	987655321	burger	40.00
551234567	331234567	orange juice	15.00
771234567	987655321	pasta	45.00
987654320	123546789	[illegible]	[illegible]

Deletion is done successfully

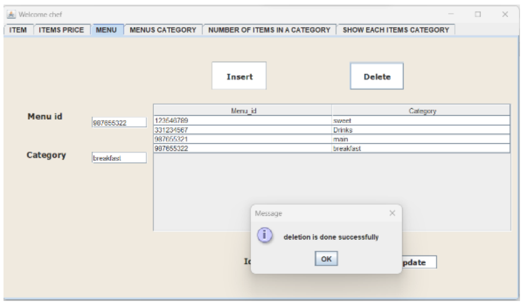
Id: 123456780

After deletion:



item_id	Menu_id	Name	Price
551234567	331234567	orange juice	15.00
771234567	987655321	pasta	45.00
987654320	123546789	chocolate	25.00

Delete a menu:



Menu id 987655322

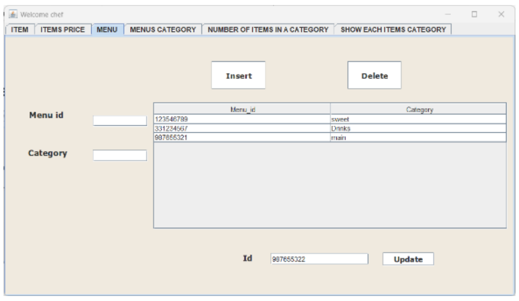
Category breakfast

Menu_id	Category
123546789	sweet
331234567	Drinks
987655321	main
987655322	breakfast

Message

deletion is done successfully

After deletion:



Menu id

Category

Menu_id	Category
123546789	sweet
331234567	Drinks
987655321	main

Id 987655322

Difficulties:

We faced some difficulties such as lack of time and self-learning in creating interfaces, but we did the work in the end.

References:

[1] https://drawio-app.com/