DATABASE MANAGEMENT SYSTEM

Online Food Ordering Management System

**PROJECT REPORT**



**Group Members:**

* **IBTEISAM KHALID BCS203056**
* **TALAL BHATTI BCS203078**
* **ABDULLAH ASLAM QAZI BCS203263**

**Section: BCS - 3**

**Submitted To: Mam Dr. Syeda Mariam Muzammal**

**Submission Date: 31th January, 2022**

**CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**Table of Contents**

[ABSTRACT 3](#_bookmark0)

[INTRODUCTION 3](#_bookmark1)

[OBJECTIVE: 3](#_bookmark2)

[SOFTWARE TOOLS 3](#_bookmark3)

ENTITY RELATIONSHIP DIAGRAM (ERD) 19

[PERFORMANCE: 4](#_bookmark4)

[NORMALIZATION 4](#_bookmark5)

[TABLES 5](#_bookmark6)

[JOINS 6](#_bookmark7)

[VIEWS 7](#_bookmark8)

[PROCEDURES 8](#_bookmark9)

[FUNCTIONS 10](#_bookmark10)

[TRIGGERS 14](#_bookmark11)

[USERS 17](#_bookmark12)

[FUNCTIONALITY 20](#_bookmark14)

[OUTPUTS 26](#_bookmark16)

[CONCLUSION 28](#_bookmark17)

[REFERENCES 28](#_bookmark18)

[CODE 28](#_bookmark19)

**ONLINE FOOD ORDERING MANAGEMENT SYSTEM**

# ABSTRACT:

Online food ordering system provides service facility to restaurant and the customer. This service is too providing food ordering. Main objective is to provide ordering by online to the customer.in this system online, ordering management will become easier to replace the difficult system where we are still using paper.

# INTRODUCTION:

An online food ordering system is software that allows the restaurant business to accept and manage on the internet easily in a minimum time. Website or app for those customers who are hungry to view the menu and place the order at home

In this pandemic situation it is very useful for us to make our life easier and protective. We can order food and stay our home protective. This system helps restaurant to do all things more accurately and faster .it reduces manual work and improve effieciency.it stores many data base and manage the record easily. This software is helping food ordering to maintain the stock and flows and there are many more functionalities

* To maintain the records like user information, staff information, booking and food details etc.
* Control orders
* Keep record of amount
* Maintain the staff shifting information

# 

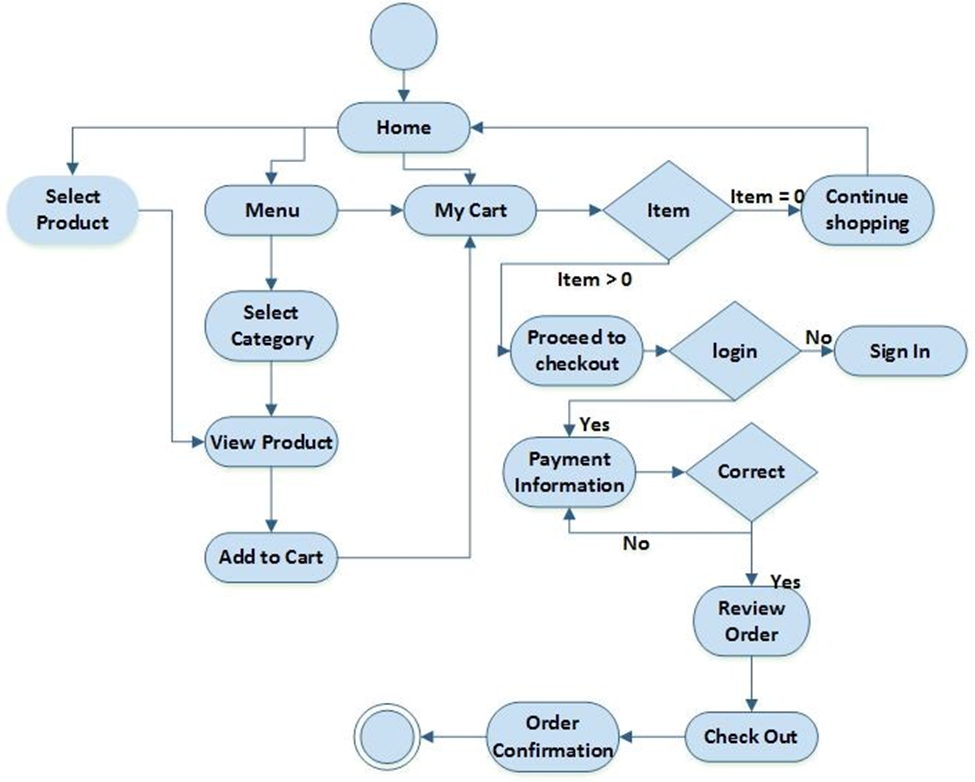
# OBJECTIVE:

Facing this pandemic situation, we all are in depression about are lives and after visiting many marts and stored we have this idea to maintain the system which store every data.

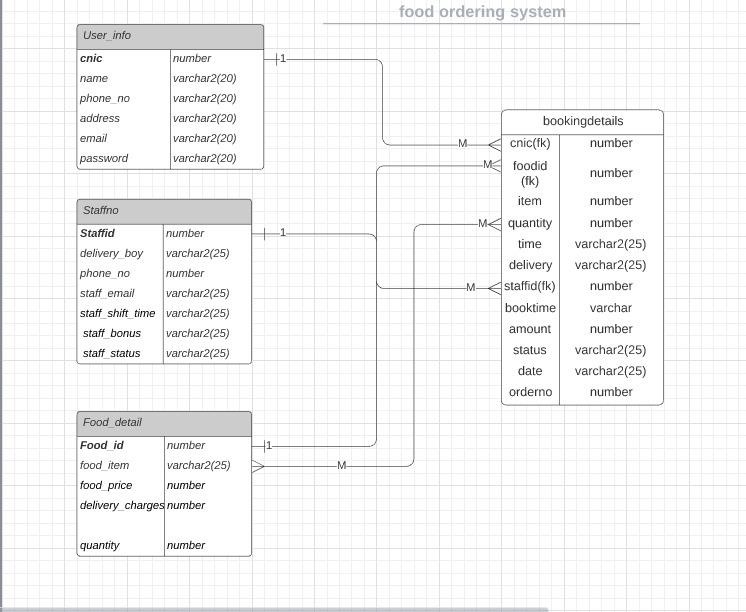
Food delivery aims to save manage your reservations on a computer and in the process let you keep the regular customer dealing. According to this system we can click one button and make your order reserve, check menu details and your total bill. There are also field for entering the customer address, phone, email.

# SOFTWARE TOOLS:

* SQL Plus
* Note pad ++
* Xaamp
* Command promt

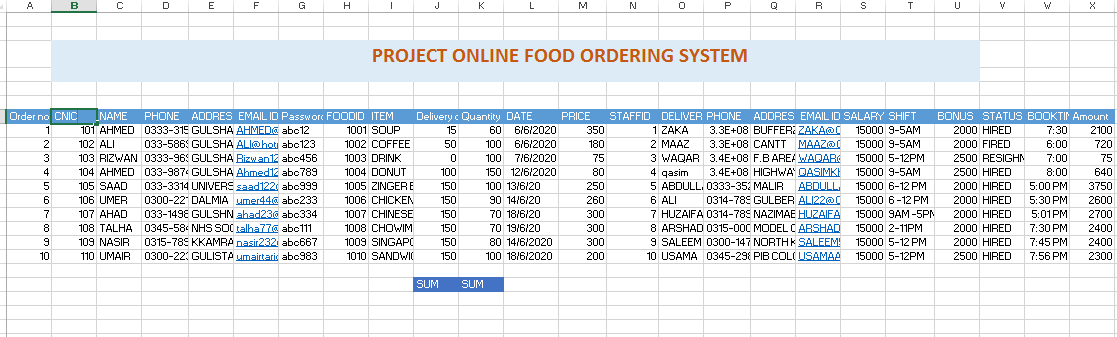


**ERD DIAGRAM:**



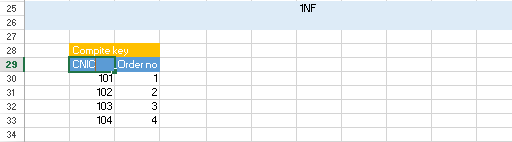
# PERFORMANCE:

## NORMALIZATION

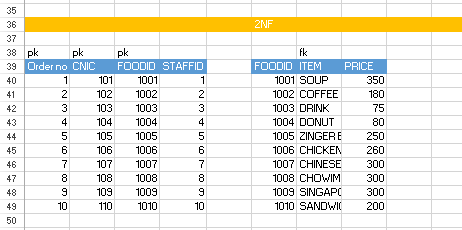


**NORMALIZED DATA:**

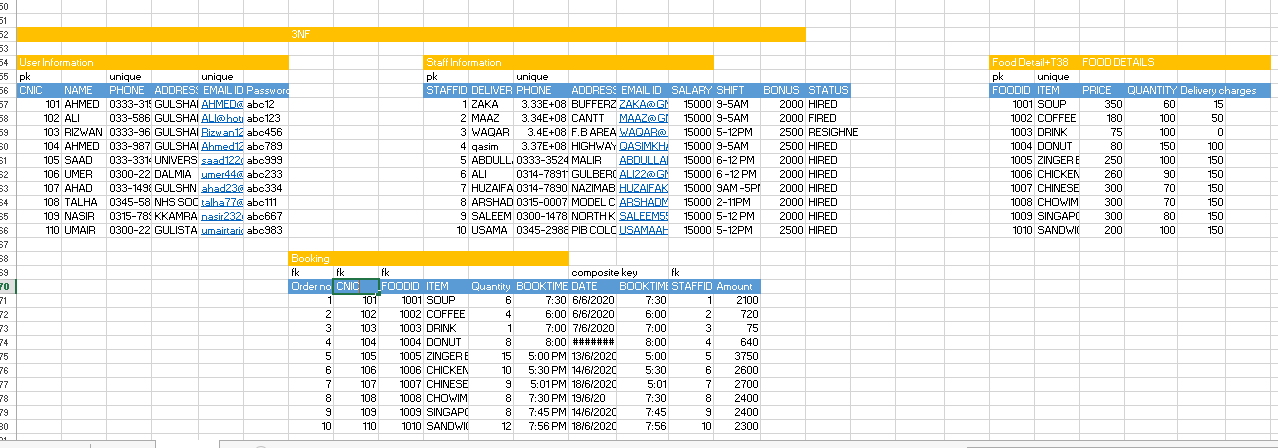
**1st NF**



**2nd NF**

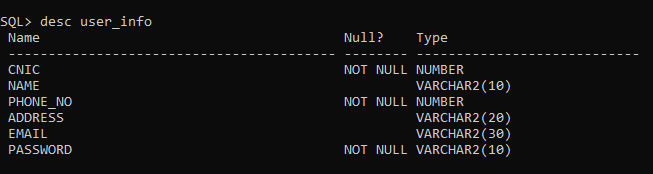


**3rd NF**

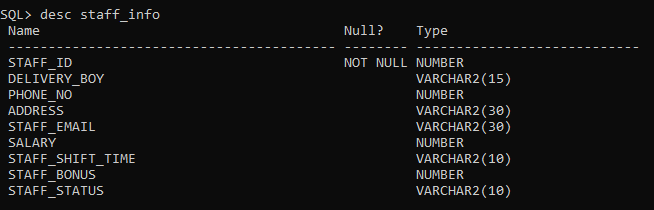


## TABLES

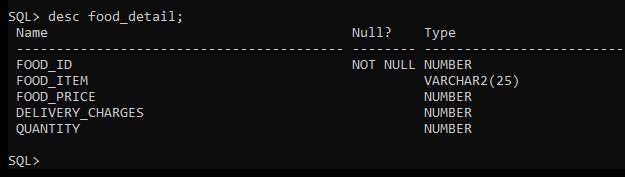
1. **User Information**



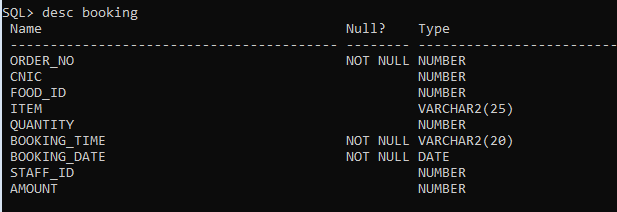
1. **Staff Information**



1. **Food details**



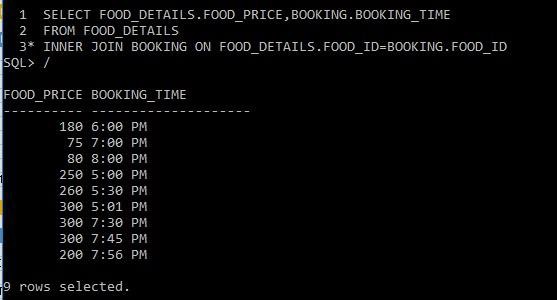
1. **Booking**



## **JOINS**

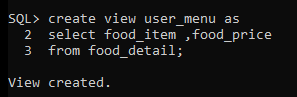
* select food\_details.food\_price, booking.booking\_time from food\_details INNER JOIN booking

on food\_details.food\_id = booking.food\_id;



## VIEWS

* create view user\_menu as select food\_item, food\_price from food\_detail;

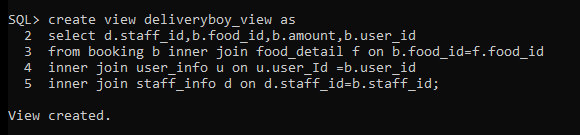


* create view deliveryboy\_view as

select d.staff\_id,b.foodid,b.totalamount,b.userid

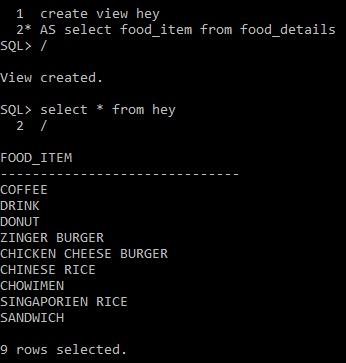
from booking b inner join food\_details f on b.foodid=f.food\_id inner join user\_info u on u.user\_Id =b.userid

inner join staff\_info d on d.staff\_id=b.staff\_id;



* create view hey

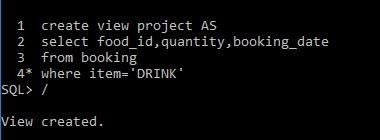
as select food\_item from food\_detail;



* create view project

as select food\_id, quantity, booking\_date from booking

where item = ‘DRINK’;



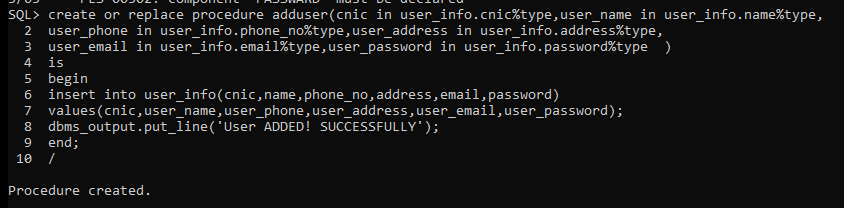
## PROCEDURES

* create or replace procedure adduser( in user\_info.cnic%type,user\_name in user\_info.name%type, user\_phone in user\_info.phone\_no%type,user\_address user\_info.address%type, user\_email in user\_info.email%type,user\_password in user\_info.passward%type )

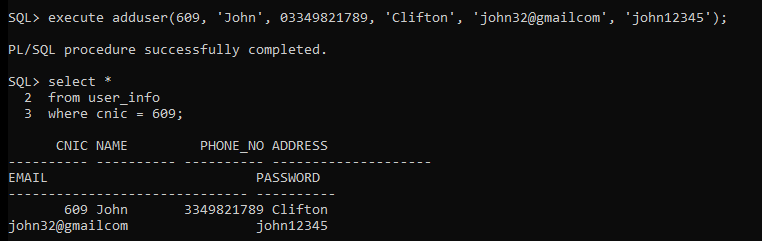
is begin

insert into user\_info(cnic,name,phone\_no,address,email,passward) values(cnic,user\_name,user\_phone,user\_address,user\_email,user\_password); dbms\_output.put\_line(' User ADDED! SUCCESSFULLY');

end;



execute adduser(609, 'John', 03349821789, 'Clifton', 'john32@gmailcom', 'john12345');

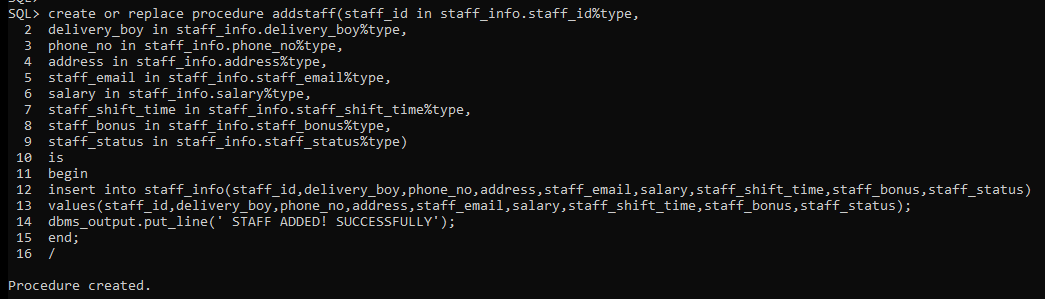


* create or replace procedure addstaff(staff\_id in staff\_info.staff\_id%type, delivery\_boy in staff\_info.delivery\_boy%type, phone\_no in staff\_info.phone\_no%type, address in staff\_info.address%type, staff\_email in staff\_info.staff\_email%type, salary in staff\_info.salary%type, staff\_shift\_time in staff\_info.staff\_shift\_time%type, staff\_bonus in staff\_info.staff\_bonus%type, staff\_status in staff\_info.staff\_status%type)

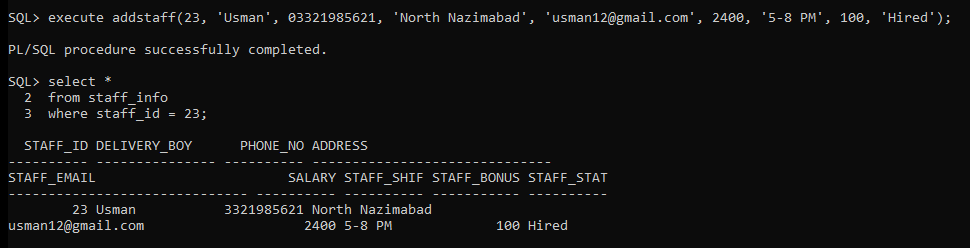
is begin

insert into staff\_info(staff\_id,delivery\_boy,phone\_no,address,staff\_email,salary,staff\_shift\_time,st aff\_bonus,staff\_status) values(staff\_id,delivery\_boy,phone\_no,address,staff\_email,salary,staff\_shift\_time,staff\_ bonus,staff\_status);

dbms\_output.put\_line(' STAFF ADDED! SUCCESSFULLY'); end;



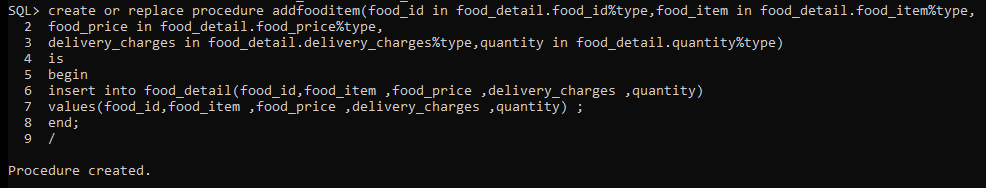
execute addstaff(23, 'Usman', 03321985621, 'North Nazimabad', 'usman12@gmail.com', 2400, '5-8 PM', 100, 'Hired');



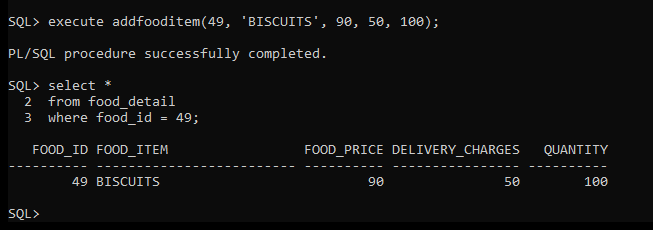
* create or replace procedure addfooditem(food\_id in food\_details.food\_id%type,food\_item in food\_details.food\_item%type, food\_price in food\_details.food\_price%type, delivery\_charges in food\_details.delivery\_charges%type,quantity in food\_details.quantity%type)

is begin

insert into food\_details(food\_id,food\_item ,food\_price ,delivery\_charges ,quantity) values(food\_id,food\_item ,food\_price ,delivery\_charges ,quantity) ;

end;

execute addfooditem(49, 'BISCUITS', 90, 50, 100);



## FUNCTIONS

* create or replace function deliver return number

is

TOTAL\_SALARY number;

begin

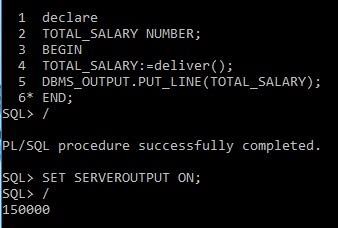
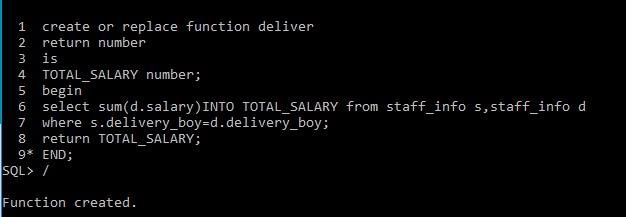
select sum(d.salary)INTO TOTAL\_SALARY from staff\_info s,staff\_info d where s.delivery\_boy=d.delivery\_boy;

return TOTAL\_SALARY; end;

declare

TOTAL\_SALARY number; begin TOTAL\_SALARY:=deliver();

dbms\_output.put\_line(TOTAL\_SALARY); end;



* create or replace function deliver return number

is

MAX\_SALARY number; begin

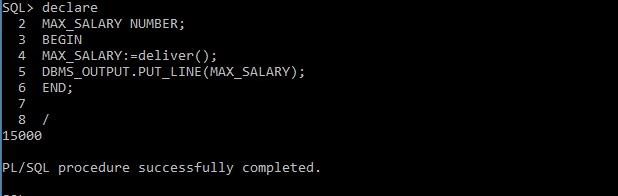
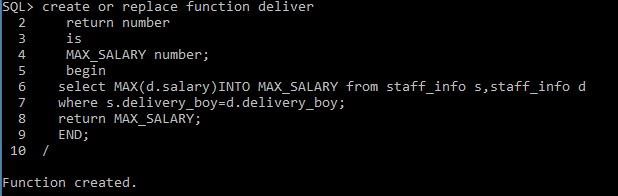
select MAX(d.salary)

INTO MAX\_SALARY from staff\_info s,staff\_info d where s.delivery\_boy=d.delivery\_boy;

return MAX\_SALARY; end;

declare

MAX\_SALARY number; begin MAX\_SALARY:=deliver();

dbms\_output.put\_line(MAX\_SALARY); end;

* create or replace function total return number is

total number(2):=0; begin

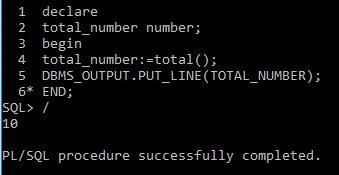
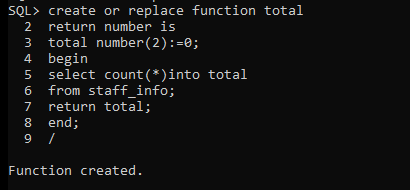
select count(\*)into total from staff\_info;

return total; end;

declare

total\_number number; begin total\_number:=total();

dbms\_output.put\_line(total\_number); end;



* create or replace function deliver return number

is

MIN\_SALARY number; begin

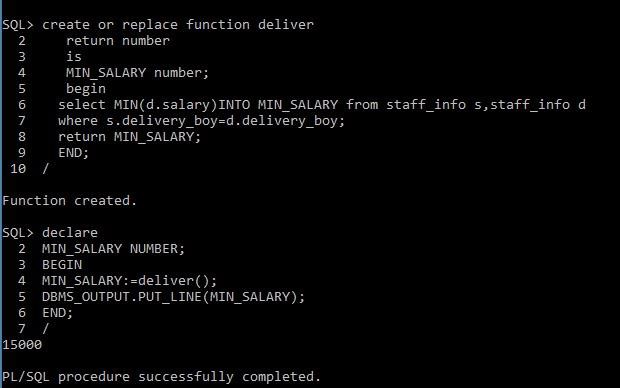
select MIN(d.salary) into MIN\_SALARY from staff\_info s, staff\_info d where s.delivery\_boy = d.delivery\_boy;

return MIN\_SALARY; end;

declare

MIN\_SALARY number; begin MIN\_SALARY:=deliver();

dbms\_output.put\_line(MIN\_SALARY); end;

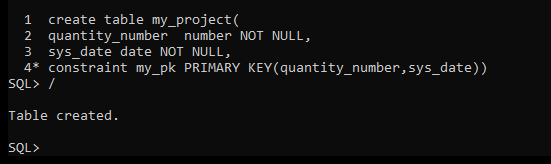


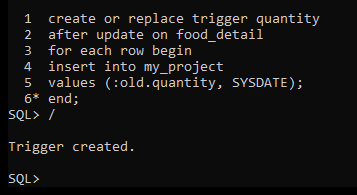
## TRIGGERS

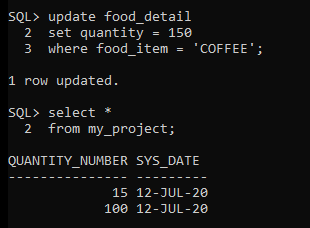
* create or replace trigger quantity after update on food\_detail

for each row begin insert into my\_project

values (:old quantity , SYSDATE); end;







* create or replace trigger display\_food\_price\_charges before delete or insert or update on food\_detail

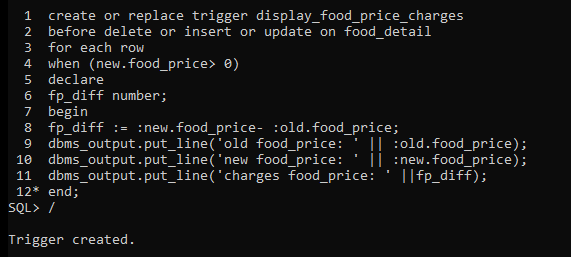
for each row

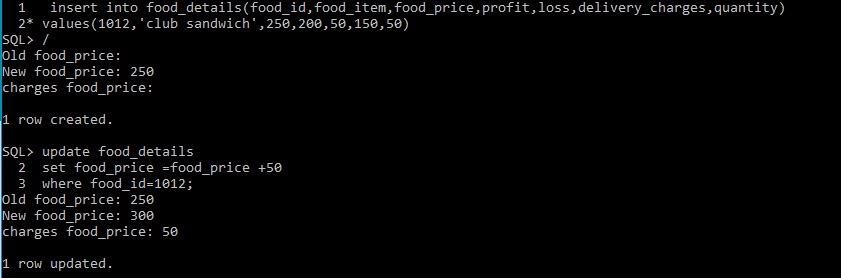
when (new.food\_price> 0) declare

fp\_diff number; begin

fp\_diff := :new.food\_price- :old.food\_price; dbms\_output.put\_line('old food\_price: ' || :old.food\_price); dbms\_output.put\_line('new food\_price: ' || :new.food\_price); dbms\_output.put\_line('charges food\_price: ' ||fp\_diff);

end;

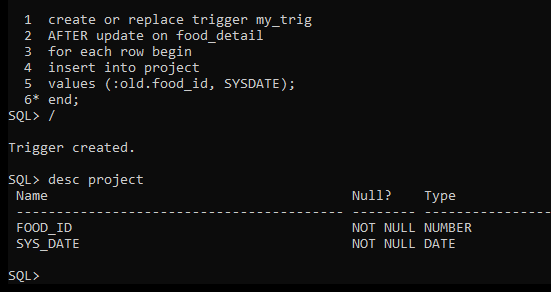




* create or replace trigger my\_trig after update on food\_detail

for each row begin insert into project

values (:old.food\_id, SYSDATE); end;



## CURSORS

* declare cursor staff is

select staff\_id,delivery\_boy from staff\_info where phone\_no=03358693345; v\_staff\_id staff\_info.staff\_id%type;

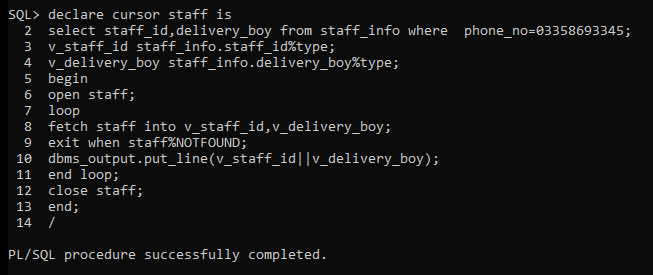
v\_delivery\_boy staff\_info.delivery\_boy%type; begin

open staff; loop

fetch staff into v\_staff\_id,v\_delivery\_boy; exit when staff%NOTFOUND;

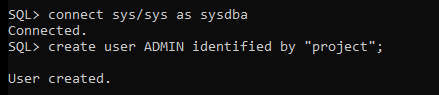
dbms\_output.put\_line(v\_staff\_id||v\_delivery\_boy); end loop;

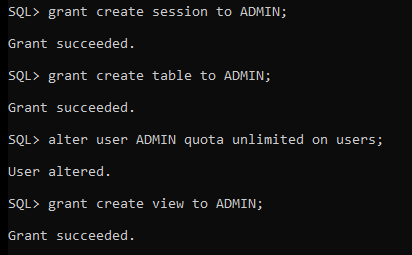
close staff; end;



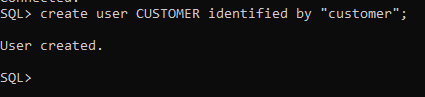
## 

## USERS

**User “ADMI****N**

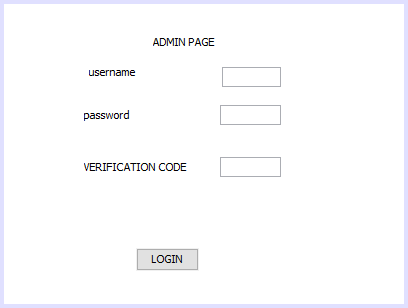


**User “CUSTOMER”**

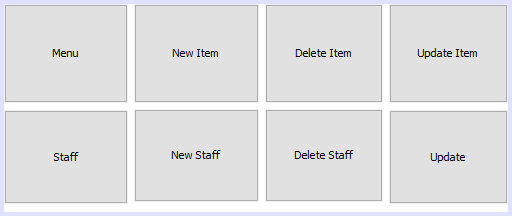


# FUNCTIONALITY

**Admin Page**



**Admin Dashboard**



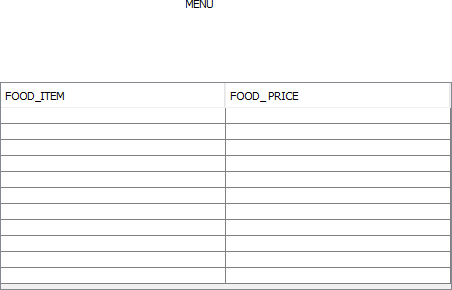
**Delete Food Item**



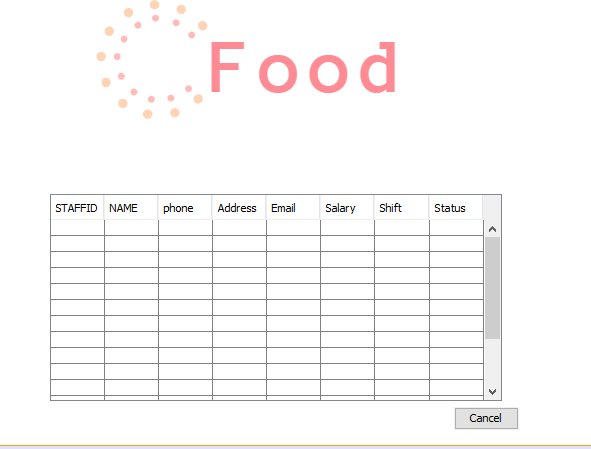
**Insert Staff**



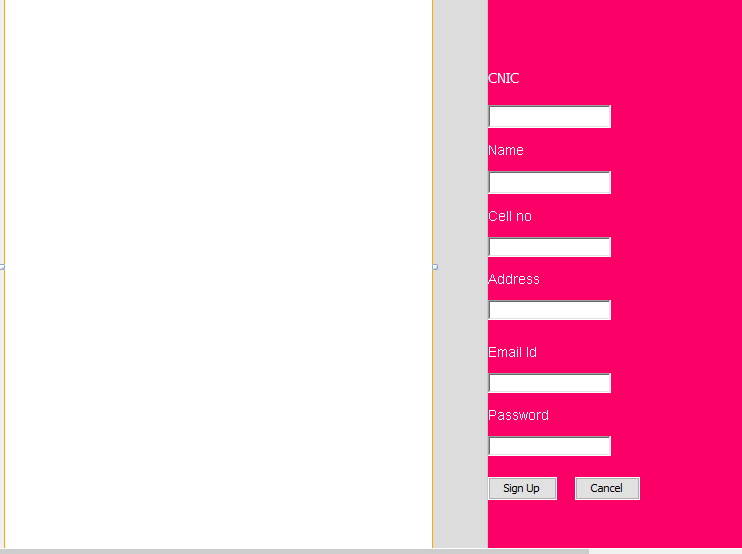
**Menu**



**Staff List**



**Signup**



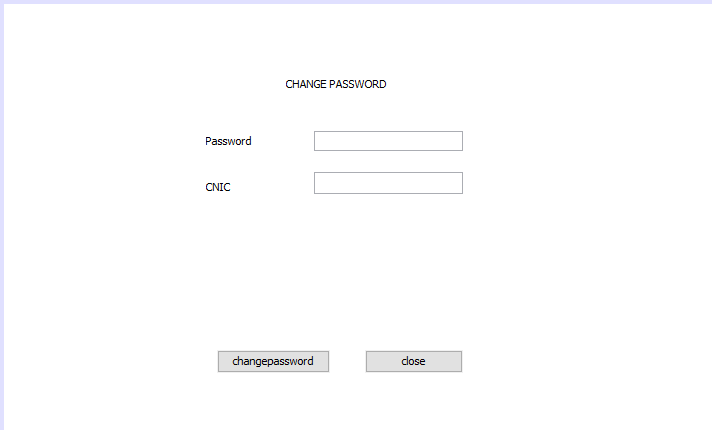
**Update Staff**



**Change Admin Password**



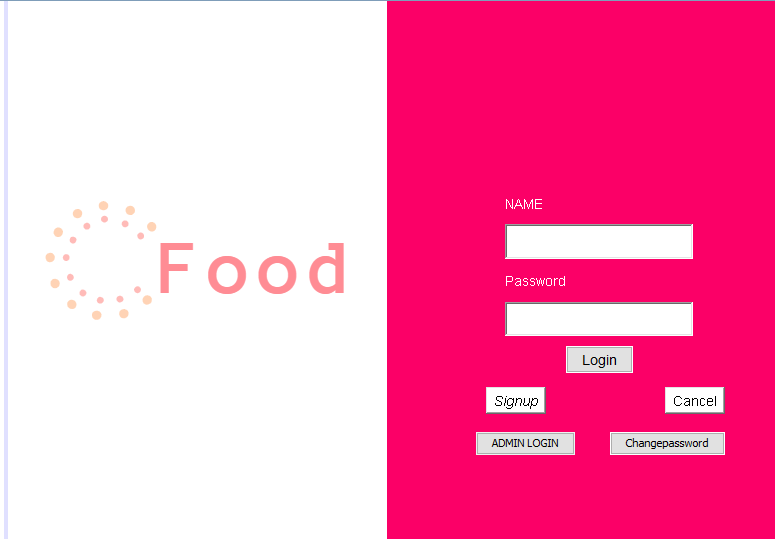
**Change Password User**



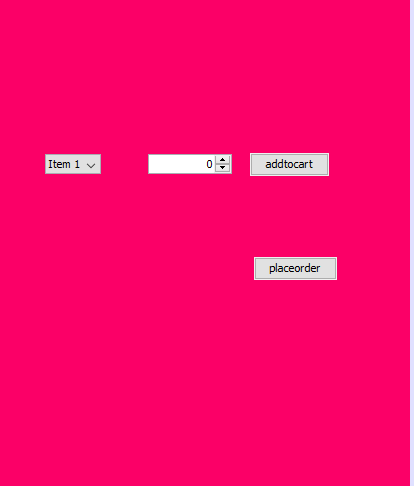
**Insert Menu Item**



**Login**



**Order**

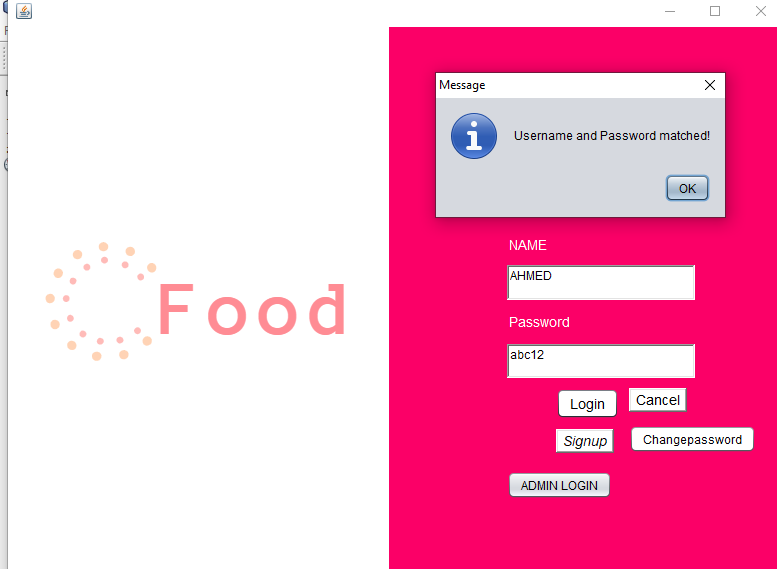
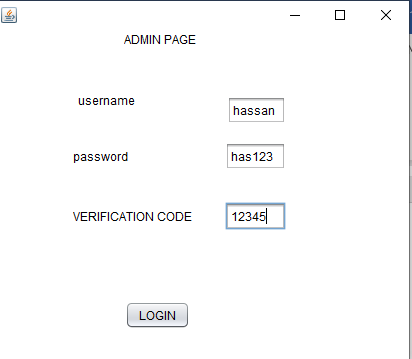


**Update Menu Item**

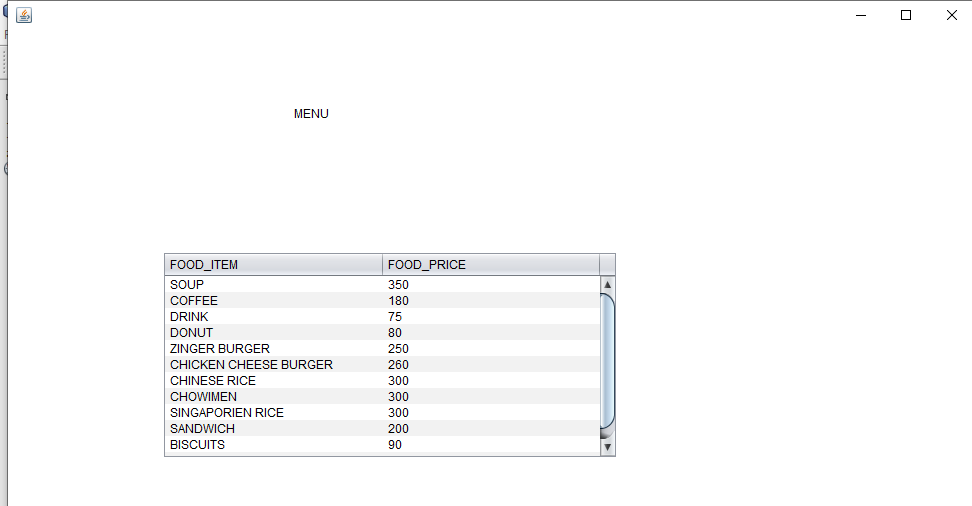


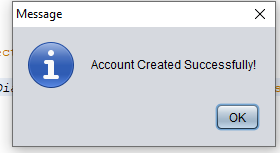
# OUTPUTS

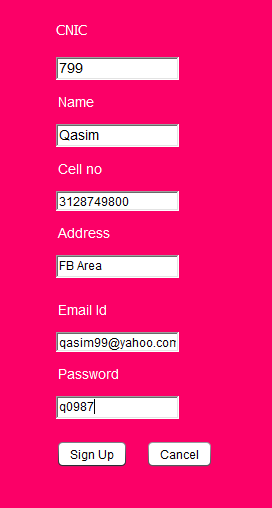
**Admin Login User Login**



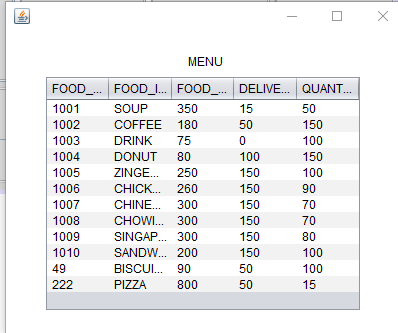
**Menu**

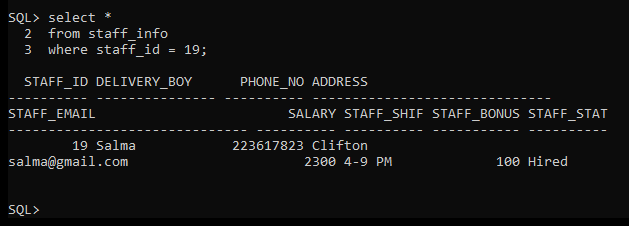
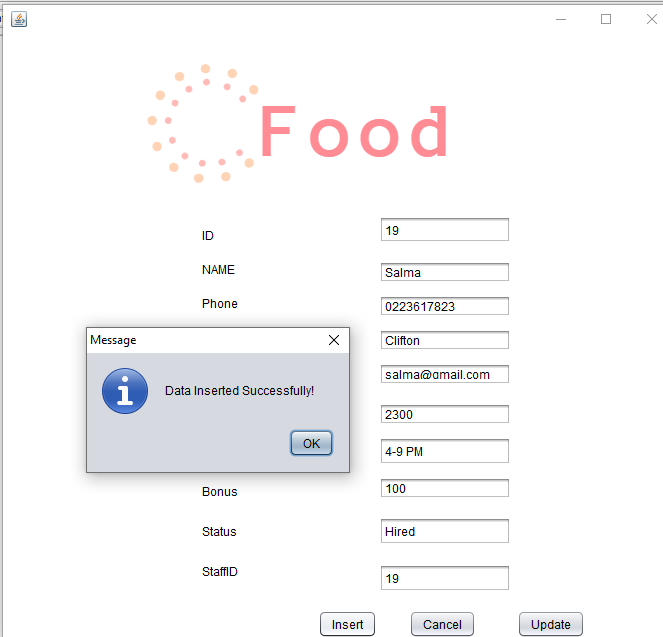


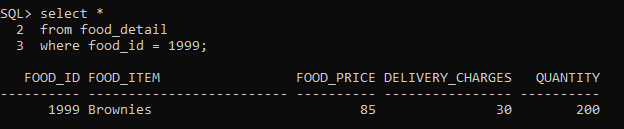
**Signup**

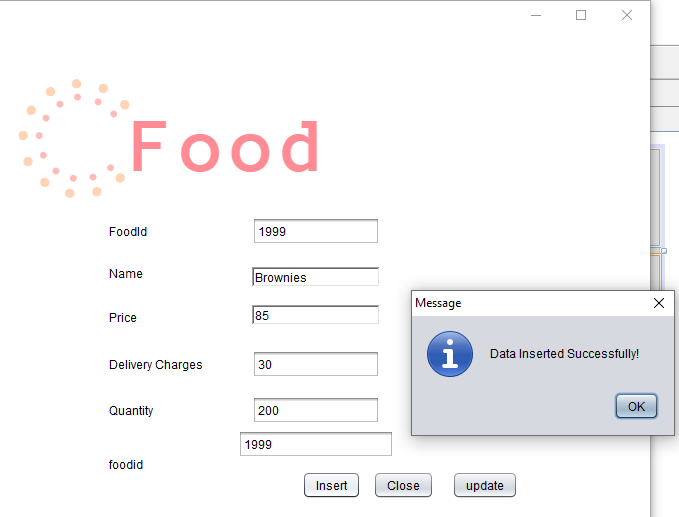


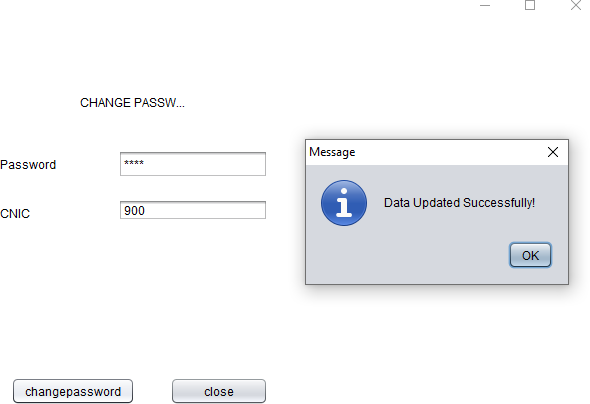
**Admin Dashboard Admin Menu**

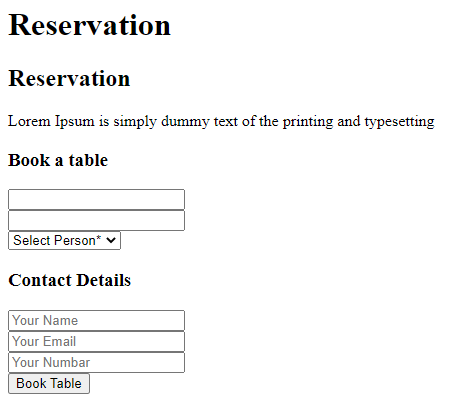
**Insert Staff **



**Insert Menu Item**





****

# 

# **CONCLUSION**

* All food order details like order type are stored daily.
* All the amounts updated daily in database.
* This system reduces manual works.
* All the shift timings of staff, delivery staff data, charges and bonus will be stored in the database for easy access.
* All data will be saved in database making it easier for the admin to view all the data.
* Data related to users, staff or food items can be inserted, updated or deleted.

# REFERENCES

* Stack Overflow
* Database Management Systems books and websites