American International University- Bangladesh

Project Title:
Restaurant Management System

Course Name:
ADVANCE DATABASE MANAGEMENT SYSTEM

Section:
B

Student Name	Student Id
BANERJEE, PRODIPTA	15-28239-1
SAHA, MANIK	18-38965-3
MOWMITA, NAZIA AFSAN	18-37787-2
EFFAT JAHAN	18-38718-3

CONTENTS

Introduction:

Restaurant management system This system will help the users do their task easily and efficiently. It will also restrict unauthorized access to confidential data.

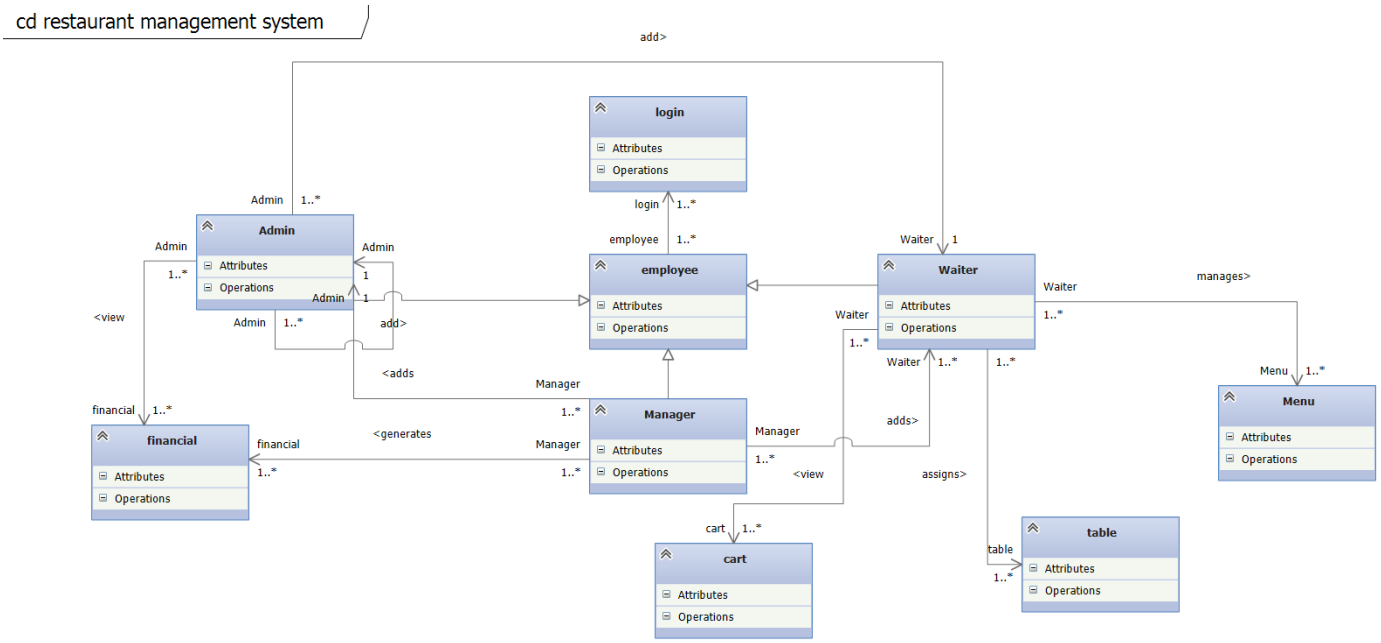
Project proposal:

Restaurant management system We designed this system for 3 types of user

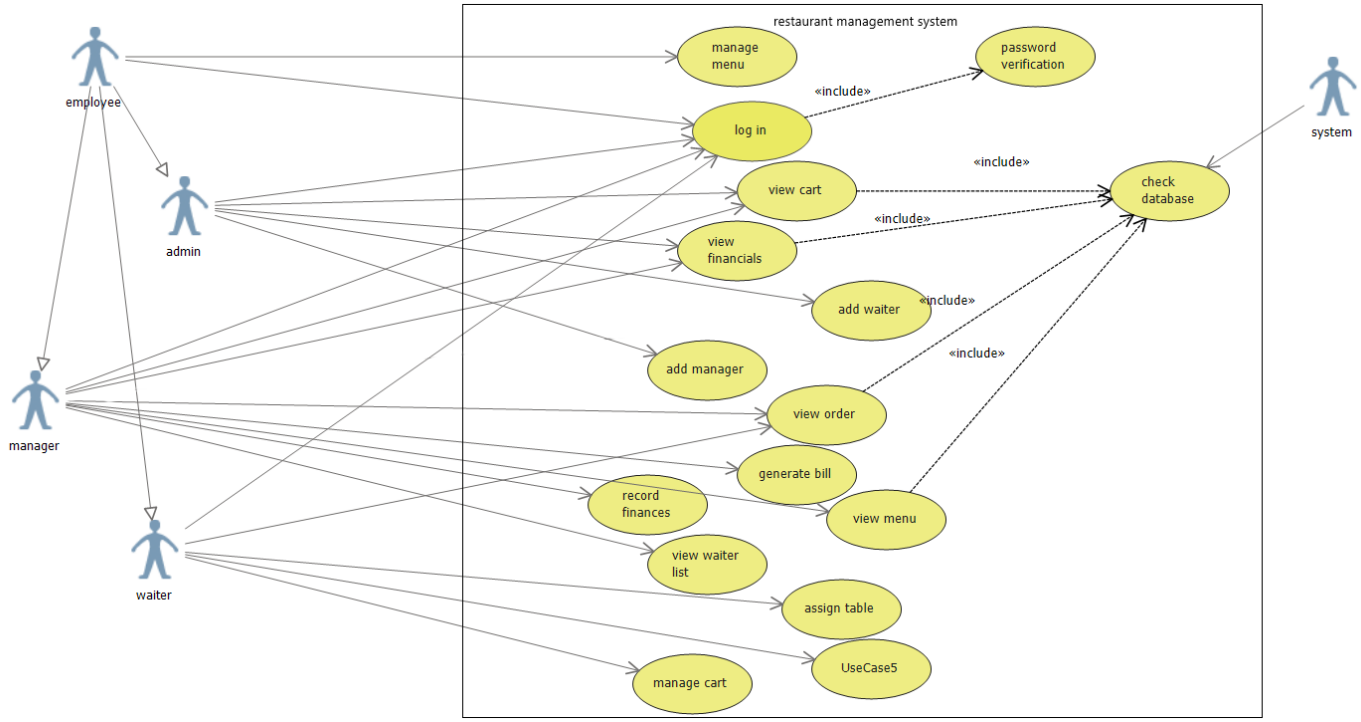
1. Admin
2. Manager
3. Waiter

The system is designed manage employees, food order, table and finances of a restaurant.

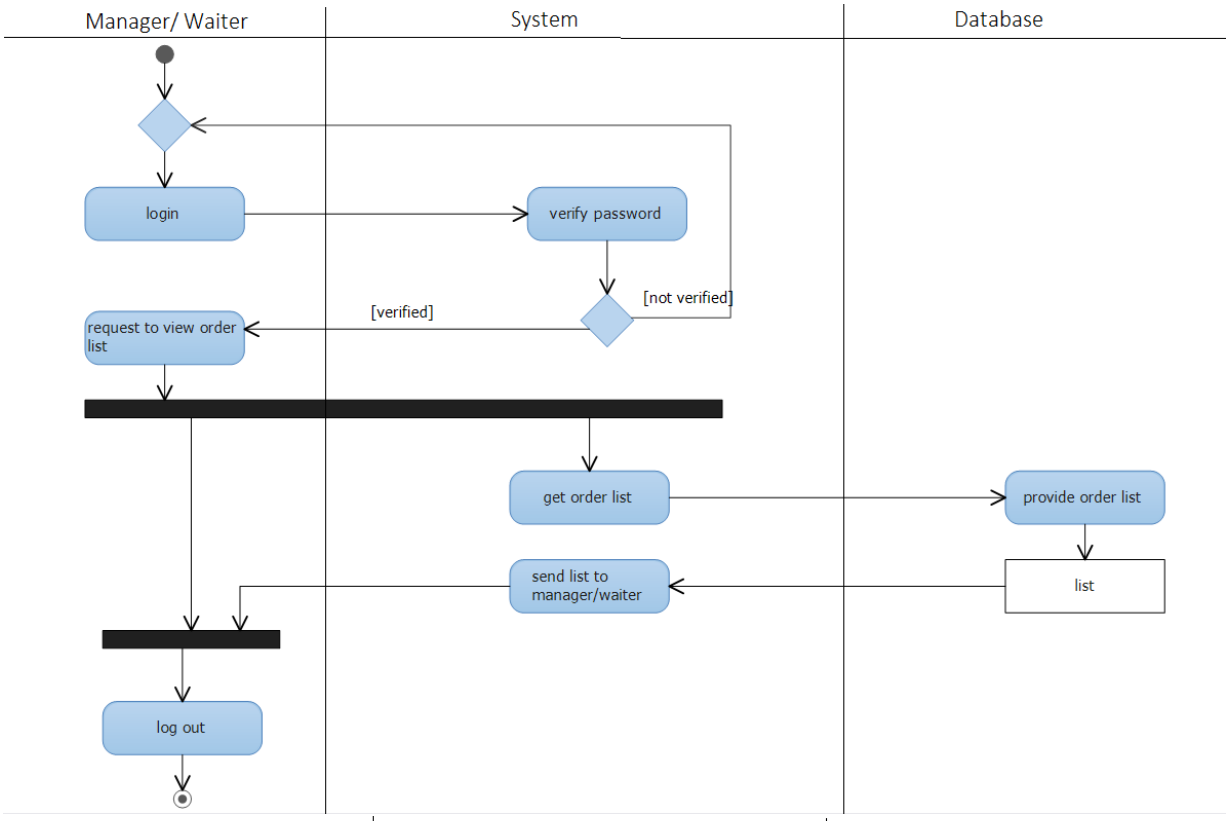
Class Diagram:

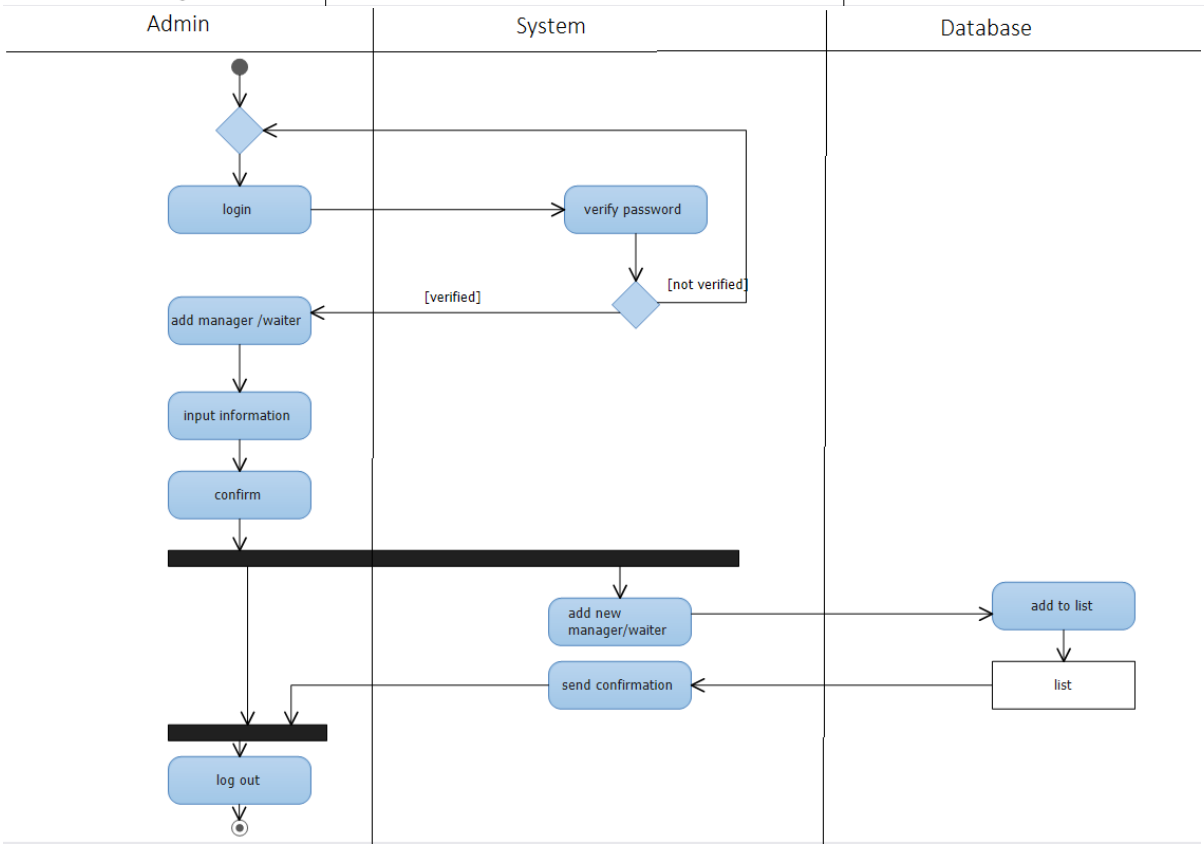


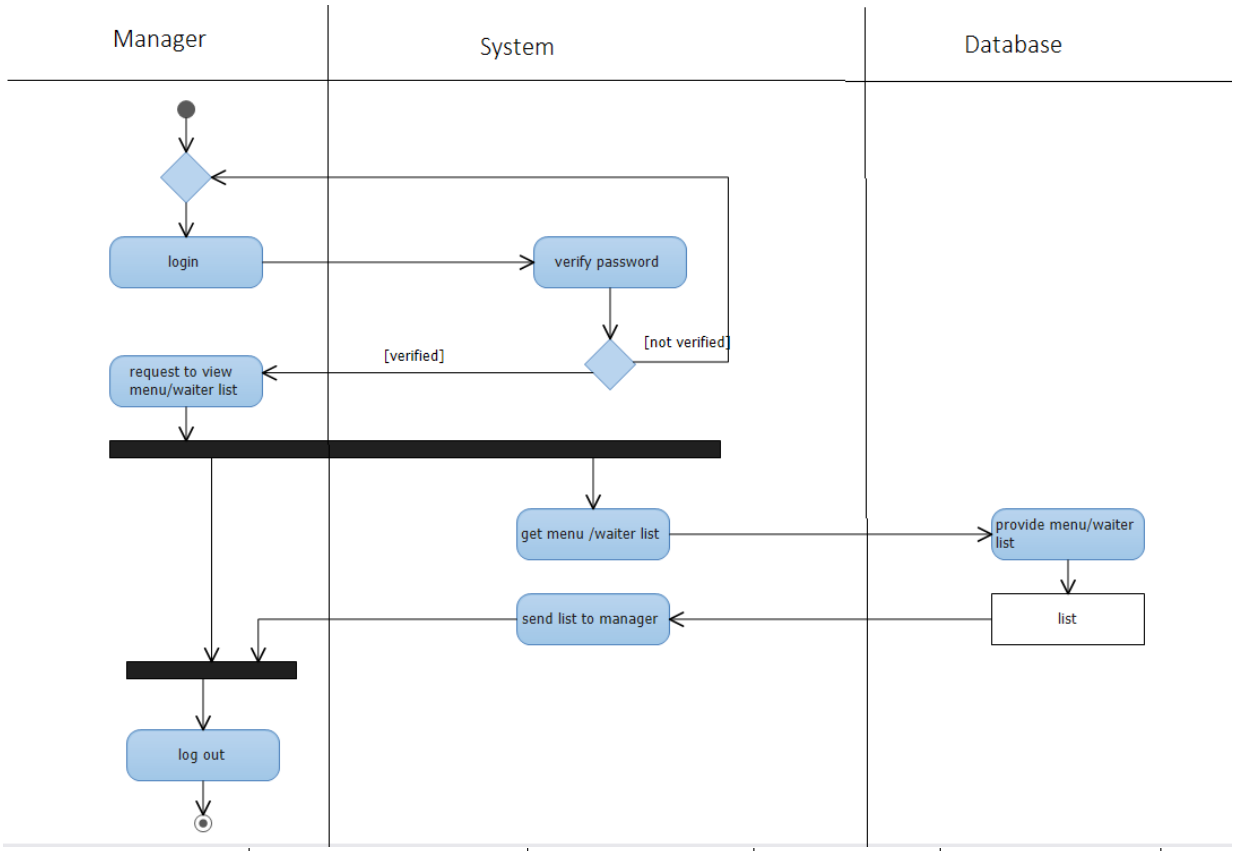
Use Case Diagram:

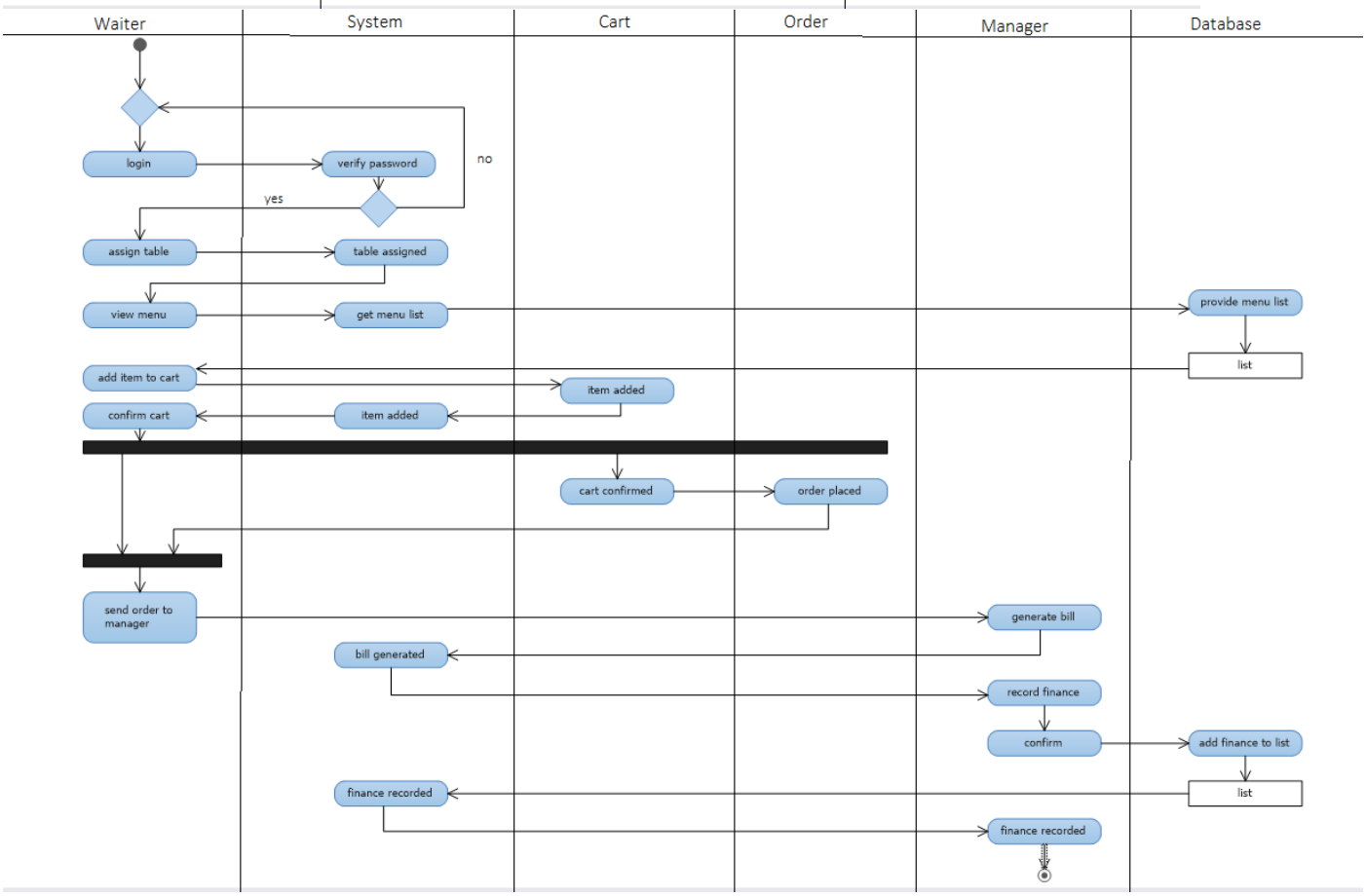


Activity Diagram:

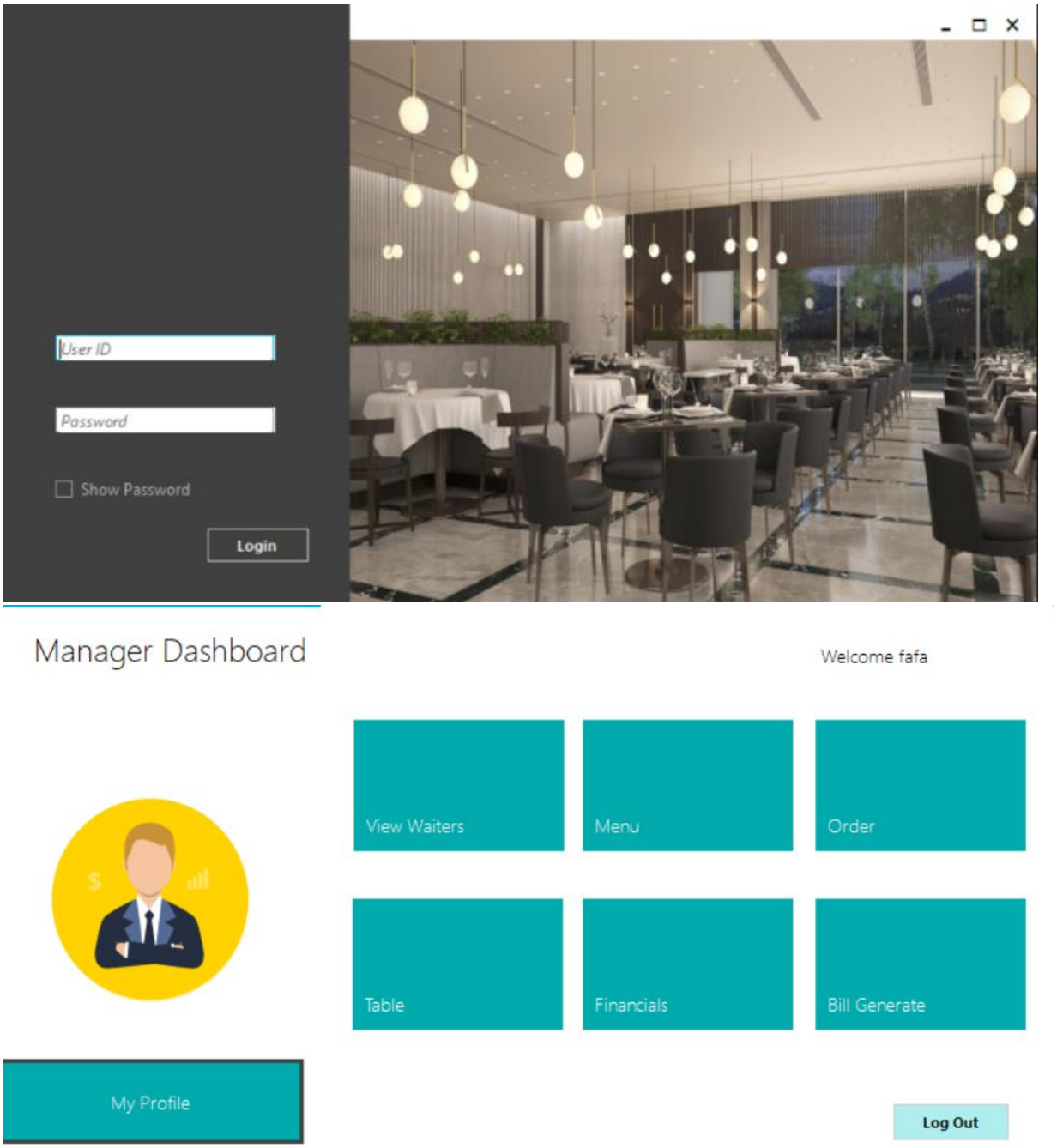





Manager
System
Database
login
verify password
[not verified]
[verified]
request to view menu/waiter list
get menu /waiter list
provide menu/waiter list
send list to manager
list
log out


Waiter
System
Cart
Order
Manager
Database
login
verify password
no
yes
assign table
table assigned
view menu
get menu list
provide menu list
list
add item to cart
item added
confirm cart
item added
cart confirmed
order placed
send order to manager
generate bill
bill generated
record finance
confirm
add finance to list
list
finance recorded
finance recorded

User Interface:



AdminDashboard

Welcome Saqif

Add Employee | Manage Admin | Manage Manager | Manage Waiter

Table | Order | Menu | Financials

My Profile

Log Out

<

Post

Id

Name

Address

Email

Phone

Gender ○ Male ○ Female

Date of Birth Tuesday, March 10, 2020

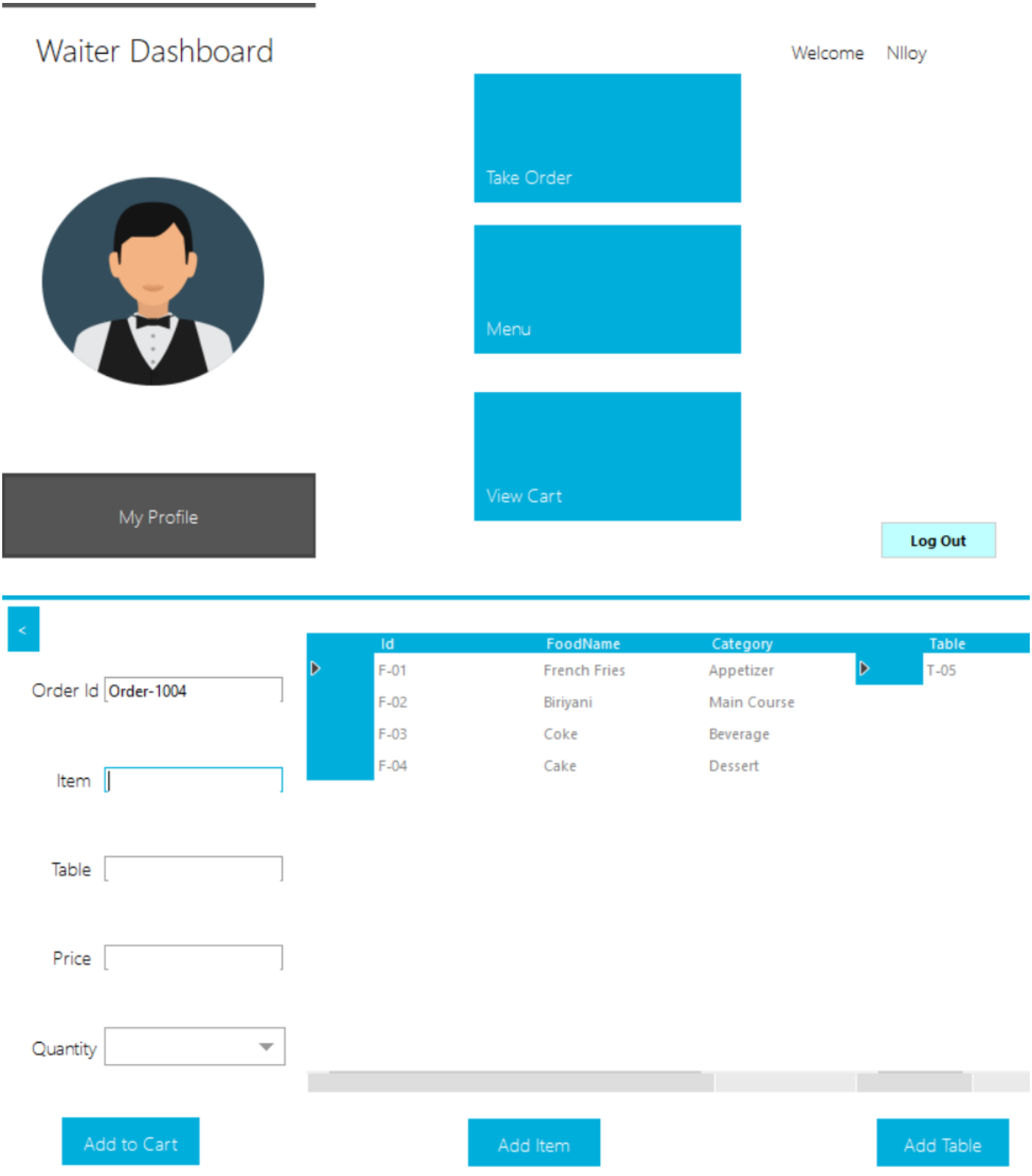
Joining Date Tuesday, March 10, 2020

Marital Status

Blood Group

Salary

Save Cancel

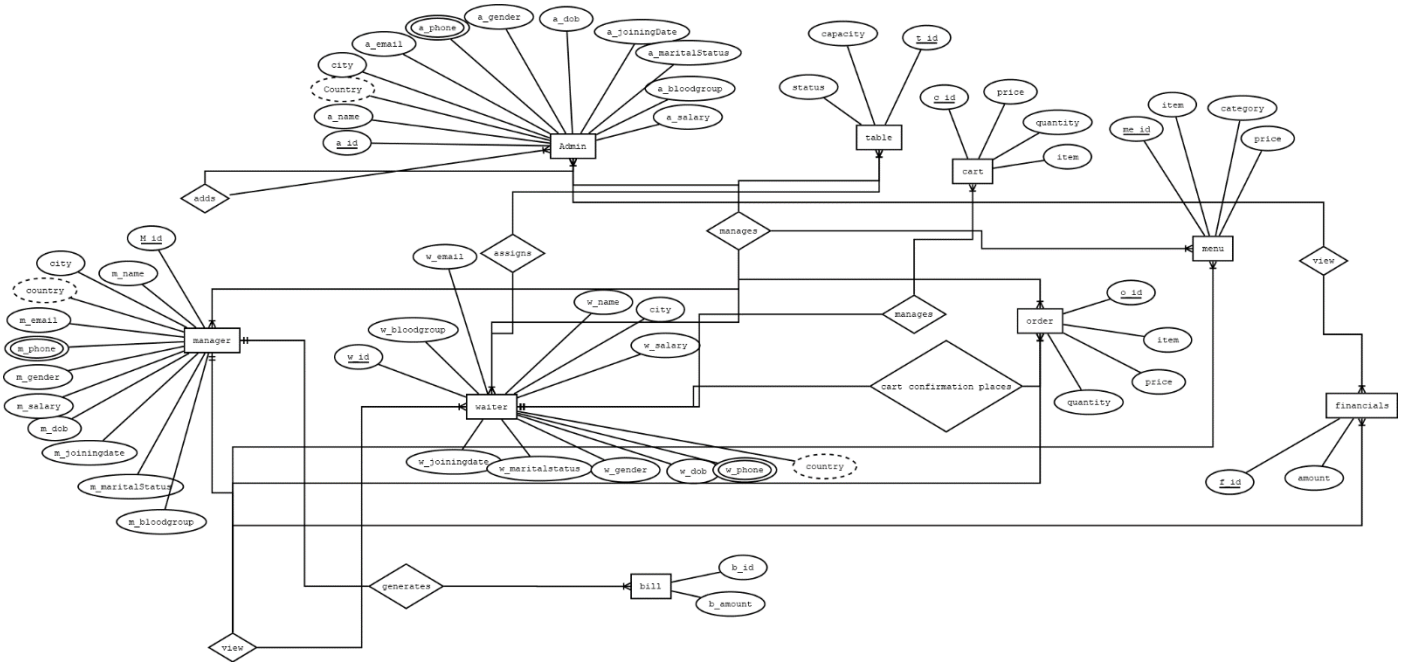


Scenario:

In a Restaurant Management System, there is an admin. Admin is identified by a unique primary key. He also has name, address, email, phone, gender, date of birth, joining date, marital status, blood group salary, city, country. An admin can have one or more phone numbers. An admin can add one or many admins. Admin can view Financial details. Financials can be viewed by one or many admin. Financials are identified by a unique id. It also has amount. Admin manages manager, waiter, table, order and menu. A manager has a unique id. He also has name, country, email, phone number, gender, date of birth, joining date, marital

status, blood group, salary and city. Manager can have one or more phone number. An admin manages one or more manager. One manager can be managed by one or more admin. Waiter is identified by a unique id. Waiter also has name, country, email, phone number, gender, date of birth, joining date, marital status, blood group, salary and city. A waiter can be managed by one or more admin. An admin can manage one or more waiter. Waiter can have one or more phone number. An admin manages one or more tables. One table can be managed by one or more admin. Tables have a status, capacity and a unique id. An admin can manage one or more order. An order can be managed by one or more admin. Order has items, price, capacity and a unique order id. An admin can manage one or more menu. One menu can be managed by one or more admin. Menu has a name, category, price and a unique menu id. A manager can view waiter list. One waiter works under exactly one manager. Manager can view order list. But one order is viewed by exactly one manager. Manager can also view menu list. Exactly one manager is responsible for one menu. A manager can view financials. One manager can generate one or more bill. One n=bill can be generated by only one manager. Bill is identified by bill id. Bill also has price and quantity. A waiter assigns one or more table. One table can be assigned by one or more waiter. This management system also has a cart. Cart is identified by a cart id. It also has price, quantity, and items. A cart can be managed by one or more waiter. One waiter can manage only one cart. One waiter can confirm cart to place one or more order. One order can be placed by only one waiter.

ER Diagram:



Normalization:

Admin Adds Admin:

UNF:

Adds (a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary)

1NF:

Here a_phone is a multivalued attribute.

1. a_id, a_name, country, city, a_email, a_phone, a_phone1, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary

2NF:

1. a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary

3NF:

1. a_id, a_name, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id**
2. c_id, country, city

Table Creation:

1. a_id, a_name, a_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary,**c_id**
2. c_id, country, city

Admin Manages Manager:

UNF:

Manages (a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary)

1NF:

Here a_phone is a multivalued attribute.

1. a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary

2NF:

1. a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary

2. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary

3NF:

1. a_id, a_namea_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id**

2. m_id, m_name, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**

3. c_id, , country, city

Table Creation:

1. a_id, a_namea_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id,am_id**

2. m_id, m_name, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id,am-id**

3. c_id, , country, city

4.am_id, **a_id, m_id**

Admin Manages Waiter:

UNF:

Manages (a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary,w_id, w_name, country, city, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary)

1NF

Here a_phone is a multivalued attribute.

1. a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, w_id, w_name, country, city, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary

2NF:

1. a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary

2. w_id, w_name, country, city, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary

3NF:

1. a_id, a_namea_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id**

2. w_id, w_name, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id**

3. c_id, country, city

Table Creation:

1. a_id, a_namea_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id,aw_id**

2. w_id, w_name, w_email, w_phone1,w_phone2, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id,aw_id**

3. c_id, country, city

4.aw_id, **a-id, w_id**

Admin Manages Table:

UNF:

Manages (a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, t_id,*status,capacity*)

1NF:

Here a_phone is a multivalued attribute.

1. a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, t_id,status,capacity

2NF:

1. a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary

2. t_id,status,capacity

3NF:

1. a_id, a_namea_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id**

2. t_id,status,capacity

3. c_id, , country, city

Table Creation :

1. a_id, a_namea_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id,at_id**

2. t_id,status,capacity**,at_id**

3. c_id, , country, city

4.at_id, **a_id,t_id**

Admin Manages Order:

UNF:

Manages (a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, o_id,item,quantity,price)

1NF:

Here a_phone is a multivalued attribute.

1. a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, a_salary, o_id,item,quantity,price

2NF:

1. a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary

2. o_id,item,quantity,price

3NF:

1. a_id, a_name, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id**

2. o_id,item,quantity,price

3. c_id, , country, city

Table Creation:

1. a_id, a_namea_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id,ao_id**

2. o_id,item,quantity,price, **ao_id**

3. c_id, , country, city

4.ao_id, **a_id,o_id**

Admin Manages Menu:

UNF:

Manages (a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, me_id,item,catagory,price)

1NF:

Here a_phone is a multivalued attribute.

1. a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, a_salary, me_id ,item,catagory,price

2NF:

1. a_id, a_name, country, city, a_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary

2. me_id,item,catagory,price

3NF:

1. a_id, a_namea_email, a_phone, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id**

2. me_id,item,catagory,price

3. c_id, , country, city

Table Creation :

1. a_id, a_namea_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id,ame_id**

2. me_id,item,catagory,price,**ame-id**

3. c_id, , country, city

4.ame_id, **a-id,me_id**

Manager views waiter:

UNF:

Views (m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, w_id, w_name, country, city, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary)

1NF:

Here a_phone is a multivalued attribute.

1. w_id, w_name, country, city, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary

2NF:

1. w_id, w_name, country, city, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary

2. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary

3NF:

1. w_id, w_name, country, city, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id**

2. m_id, m_name, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**

3. c_id, , country, city

Table creation:

1. w_id, w_name, country, city, w_email, w_phone1,w_phone2, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id,m_id**

2. m_id, m_name, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**

3. c_id, , country, city

Manager Views Order:

UNF:

Views (m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, o_id,item,quantity,price)

1NF:

Here a_phone is a multivalued attribute.

1. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, o_id,item,quantity,price

2NF:

1. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary

2. o_id,item,quantity,price

3NF:

1. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**

2. o_id,item,quantity,price

3. c_id, country, city

Table Creation :

1. m_id, m_name, country, city, m_email, m_phone1,m_phone2, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**

2. o_id,item,quantity,price**,m_id**

3. c_id, country, city

Manager Views Menu:

UNF:

Views (m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary,me_id,item,catagory,price)

1NF:

Here a_phone is a multivalued attribute.

1. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary,me_id,item,catagory,price

2NF:

1. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary

2. me_id,item,catagory,price

3NF:

1. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**

2. me_id,item,catagory,price

3. c_id , country, city

Table Creation :

1. m_id, m_name, country, city, m_email, m_phone1,m_phone2, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**

2. me_id,item,catagory,price, **m_id**

3. c_id, country, city

Manager Views finantials:

UNF:

Views (m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary ,f_id,amount)

1NF:

Here a_phone is a multivalued attribute.

1. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary , f_id,amount

2NF:

1. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary

2. f_id,amount

3NF:

1. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**

2. f_id,amount

3. c_id, country, city

Table Creation :

1. m_id, m_name, country, city, m_email, m_phone1,m_phone2, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**

2. f_id,amount,**m_id**

3. c_id, country, city

Manager Generates Bills:

UNF:

Generates (m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary ,b_id,b_amount)

1NF:

Here a_phone is a multivalued attribute.

1. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary , b_id,b_amount

2NF:

1. m_id, m_name, country, city, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary

2. b_id,b_amount

3NF:

1. m_id, m_name, m_email, m_phone1,m_phone2, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**

2. b_id, b_amount

3. c_id, country, city

Table Creation :

1. m_id, m_name, m_email, m_phone1,m_phone2, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**

2. b_id, b_amount,**m_id**

3. c_id, country, city

Waiter Assigns Table

UNF:

Assigns (w_id, w_name, country, city, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, status, capacity, T_id)

1NF:

Here a_phone is a multivalued attribute.

1. w_id, w_name, country, city, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, status, capacity, T_id

2NF:

1. w_id, w_name, country, city, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary
2. status, capacity, T_id

3NF:

1. w_id, w_name, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id**
2. status, capacity, T_id
3. country, city

Table Creation:

1. w_id, w_name, w_email, w_phone1,w_phone2, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id,wt_id**
2. t_id, status, capacity,**wt_id**
3. c_id, country, city
4. wt_id**, w_id, t_id**

Waiter Cart Confirmation Places Order

UNF:

Confirmation places (w_id, w_name, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, O_id, quantity, item, price)

1NF:

Here a_phone is a multivalued attribute.

1. w_id, w_name, country, city, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, O_id, quantity, item, price

2NF:

1. w_id, w_name, country, city , w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary
2. O_id, quantity, item, price

3NF:

1. w_id, w_name, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id**
2. O_id, quantity, item, price
3. , c_id, country, city

Table Creation:

1. w_id, w_name, w_email, w_phone1,w_phone2, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id**

2. O_id, quantity, item, price,**w_id**
3. , c_id, country, city

Waiter manages cart

UNF:

Manages (w_id, w_name, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, city, country, c_id, c_item, c_quantity, c_price)

1NF:

Here Phone is a multivalued attribute.

1. w_id, w_name, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, c_id, c_item, c_quantity, c_price, city, country

2NF:

1. w_id, w_name, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary
2. ca_id, ca_item, ca_quantity, ca_price

3NF:

1. w_id, w_name, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id**
2. ca_id, ca_item, ca_quantity, ca_price
3. c_id, city, country

Table Creation :

1. w_id, w_name, w_email, w_phone, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id**
2. ca_id, ca_item, ca_quantity, ca_price, **w_id**
3. c_id, city, country

Total Table:

1. ~~a_id, a_name, a_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary,**c_id**~~
2. c_id, country, city

3. a_id, a_namea_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id**
4. m_id, m_name, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**
5. ~~c_id, , country, city~~
6. am_id, **a_id, m_id**
7. ~~a_id, a_namea_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id**~~
8. ~~w_id, w_name, w_email, w_phone1,w_phone2, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id**~~
9. ~~c_id, country, city~~
10. aw_id, **a-id, w_id**
11. ~~a_id, a_namea_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id**~~

12. t_id,status,capacity

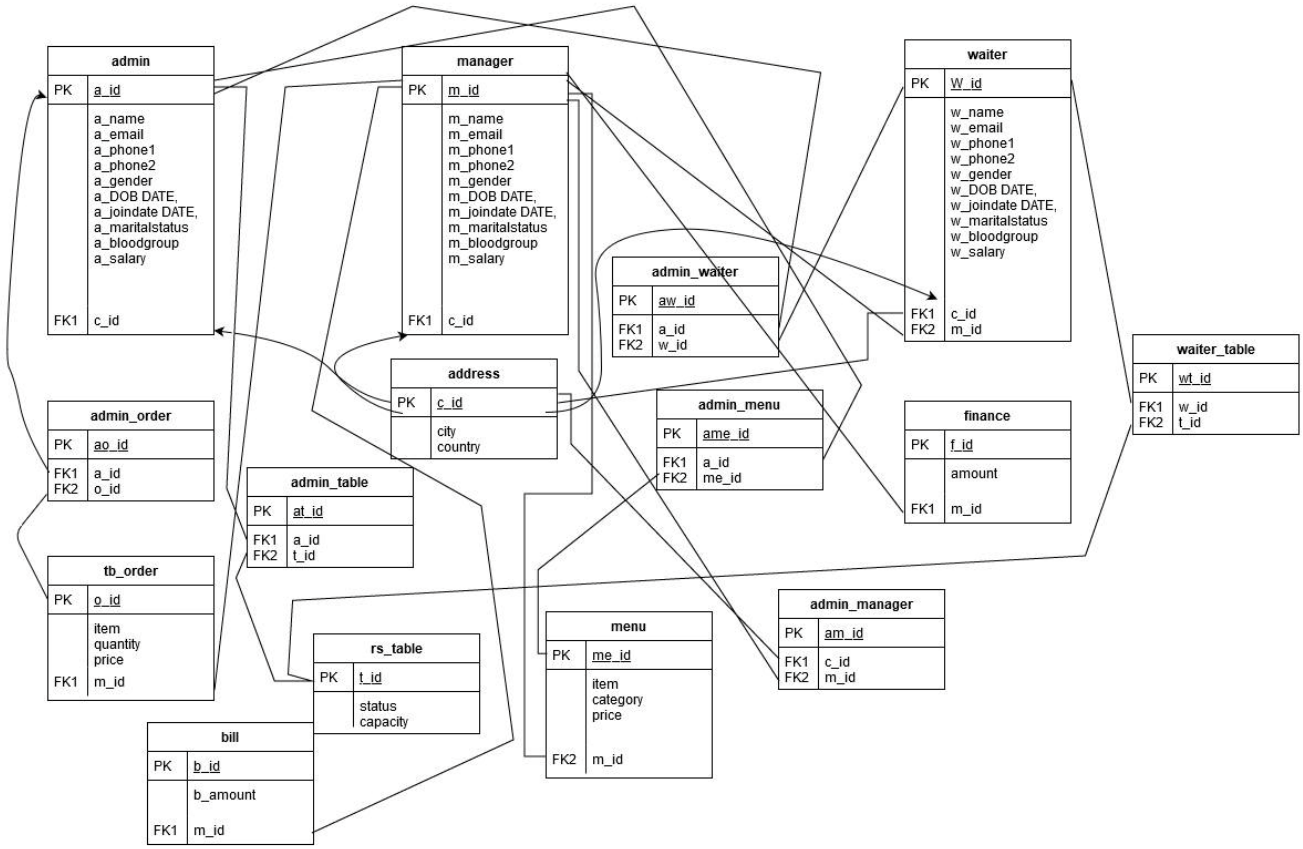
12. ~~c_id, , country, city~~
13. at_id, **a_id,t_id**
14. ~~a_id, a_name,a_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id,**~~
15. ~~o_id,item,quantity,price~~
16. ~~c_id, , country, city~~
17. ao_id, **a_id,o_id**
18. ~~a_id, a_namea_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id**~~
19. ~~me_id,item,catagory,price~~
20. ~~c_id, , country, city~~
21. ame_id, **a-id,me_id**
22. w_id, w_name, country, city, w_email, w_phone1,w_phone2, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id,m_id**
23. ~~2. m_id, m_name, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**~~
24. ~~c_id, , country, city~~
25. ~~m_id, m_name, country, city, m_email, m_phone1,m_phone2, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**~~
26. o_id,item,quantity,price **,m_id**
27. ~~c_id, country, city~~
28. ~~m_id, m_name, country, city, m_email, m_phone1,m_phone2, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**~~
29. me_id,item,catagory,price,**m_id, w_id**
30. ~~c_id, , country, city~~
31. ~~m_id, m_name, country, city, m_email, m_phone1,m_phone2, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**~~
32. f_id,amount,**m_id**
33. ~~c_id, country, city~~
34. ~~m_id, m_name, m_email, m_phone1,m_phone2, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**~~
35. b_id,b_amount,**m_id**

~~**36.** c_id, country, city~~
~~**37.** w_id, w_name, w_email, w_phone1,w_phone2, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id**~~
~~**38.** t_id, status, capacity~~
~~**39.** c_id, country, city~~
40. wt_id**, w_id, t_id**
~~**41.** w_id, w_name, w_email, w_phone1,w_phone2, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id**~~
~~**42.** O_id, quantity, item, price,**w_id**~~

Total Table :

1. c_id, country, city
2. a_id, a_namea_email, a_phone1,a_phone2, a_gender, a_dob, a_joiningdate, a_maritalstatus, a_bloodgroup, a_salary, **c_id**
3. m_id, m_name, m_email, m_phone, m_gender, m_dob, m_joiningdate, m_maritalstatus, m_bloodgroup, m_salary, **c_id**
4. am_id, **a_id, m_id**
5. aw_id, **a-id, w_id**
6. t_id,status,capacity
7. at_id, **a_id,t_id**
8. ao_id, **a_id,o_id**
9. ame_id, **a-id,me_id**
10. w_id, w_name, country, city, w_email, w_phone1,w_phone2, w_gender, w_dob, w_joiningdate, w_maritalstatus, w_bloodgroup, w_salary, **c_id,m_id**
11. o_id,item,quantity,price **,m_id**
12. me_id,item,catagory,price,**m_id, w_id**
13. f_id,amount,**m_id**
14. b_id,b_amount,**m_id**
15. wt_id**, w_id, t_id**

Schema diagram:


admin
PK a_id
a_name
a_email
a_phone1
a_phone2
a_gender
a_DOB DATE,
a_joindate DATE,
a_maritalstatus
a_bloodgroup
a_salary
FK1 c_id
manager
PK m_id
m_name
m_email
m_phone1
m_phone2
m_gender
m_DOB DATE,
m_joindate DATE,
m_maritalstatus
m_bloodgroup
m_salary
FK1 c_id
waiter
PK W_id
w_name
w_email
w_phone1
w_phone2
w_gender
w_DOB DATE,
w_joindate DATE,
w_maritalstatus
w_bloodgroup
w_salary
FK1 c_id
FK2 m_id
admin_waiter
PK aw_id
FK1 a_id
FK2 w_id
waiter_table
PK wt_id
FK1 w_id
FK2 t_id
address
PK c_id
city
country
admin_menu
PK ame_id
FK1 a_id
FK2 me_id
finance
PK f_id
amount
FK1 m_id
admin_order
PK ao_id
FK1 a_id
FK2 o_id
admin_table
PK at_id
FK1 a_id
FK2 t_id
tb_order
PK o_id
item
quantity
price
FK1 m_id
rs_table
PK t_id
status
capacity
menu
PK me_id
item
category
price
FK2 m_id
admin_manager
PK am_id
FK1 c_id
FK2 m_id
bill
PK b_id
b_amount
FK1 m_id

Screenshots:

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table WAITER
(
w_id number(10) constraint waiter_wid_pk primary key,
w_name varchar2(30),
w_email varchar2(30),
w_phone1 varchar2(30),
w_phone2 varchar2(30),
w_gender varchar2(30),
w_DOB DATE,
w_joindate DATE,
w_maritalstatus varchar2(30),
w_bloodgroup varchar2(30),
w_salary number(30) constraint waiter_salary_ck check(w_salary>0),
c_id number(10) constraint waiter_cid_fk references address(c_id),
```

Results Explain Describe Saved SQL History

Object Type TABLE Object WAITER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
WAITER	W_ID	Number	-	10	0	1	-	-	-
	W_NAME	Varchar2	30	-	-	-	✓	-	-
	W_EMAIL	Varchar2	30	-	-	-	✓	-	-
	W_PHONE1	Varchar2	30	-	-	-	✓	-	-
	W_PHONE2	Varchar2	30	-	-	-	✓	-	-
	W_GENDER	Varchar2	30	-	-	-	✓	-	-
	W_DOB	Date	7	-	-	-	✓	-	-
	W_JOINDATE	Date	7	-	-	-	✓	-	-
	W_MARITALSTATUS	Varchar2	30	-	-	-	✓	-	-
	W_BLOODGROUP	Varchar2	30	-	-	-	✓	-	-

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table table_ORDER
(
o_id number(10) constraint order_oid_pk primary key,
item varchar2(30),
quality varchar2(30),
price number(5) constraint order_price_ck check(price>0),
m_id number(10) constraint order_mid_fk references manager(m_id)
)
desc table_ORDER
```

Results Explain Describe Saved SQL History

Object Type TABLE Object TABLE_ORDER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TABLE_ORDER	O_ID	Number	-	10	0	1	-	-	-
	ITEM	Varchar2	30	-	-	-	✓	-	-
	QUALITY	Varchar2	30	-	-	-	✓	-	-
	PRICE	Number	-	5	0	-	✓	-	-
	M_ID	Number	-	10	0	-	✓	-	-

1 - 5

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table RS_TABLE
(
t_id number(10) constraint rstable_tid_pk primary key,
status varchar2(10),
capacity number(4)
)
desc rs_table
```

Results Explain Describe Saved SQL History

Object Type TABLE Object RS_TABLE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RS_TABLE	T_ID	Number	-	10	0	1	-	-	-
	STATUS	Varchar2	10	-	-	-	✓	-	-
	CAPACITY	Number	-	4	0	-	✓	-	-

1 - 3

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table MENU
(
am_id number(10) constraint menu_meid_pk primary key,
item varchar2(30),
category varchar2(30),
price number(5) constraint menu_price_ck check(price>0),
w_id number(10) constraint menu_wid_fk references waiter(w_id),
m_id number(10) constraint menu_mid_fk references manager(m_id)
)
desc menu
```

Results Explain Describe Saved SQL History

Object Type TABLE Object MENU

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MENU	AM_ID	Number	-	10	0	1	-	-	-
	ITEM	Varchar2	30	-	-	-	✓	-	-
	CATEGORY	Varchar2	30	-	-	-	✓	-	-
	PRICE	Number	-	5	0	-	✓	-	-
	W_ID	Number	-	10	0	-	✓	-	-
	M_ID	Number	-	10	0	-	✓	-	-

1 - 6

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table MANAGER
(
m_id number(10) constraint manager_mid_pk primary key,
m_name varchar2(30),
m_email varchar2(30),
m_phone1 varchar2(30),
m_phone2 varchar2(30),
m_gender varchar2(30),
m_DOB DATE,
m_joindate DATE,
m_maritalstatus varchar2(30),
m_bloodgroup varchar2(30),
m_salary number(30) constraint manager_salary_ck check(m_salary>0),
c_id number(10) constraint manager_cid_fk references address(c_id)
)
```

Results Explain Describe Saved SQL History

Object Type TABLE Object MANAGER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER	M_ID	Number	-	10	0	1	-	-	-
	M_NAME	Varchar2	30	-	-	-	✓	-	-
	M_EMAIL	Varchar2	30	-	-	-	✓	-	-
	M_PHONE1	Varchar2	30	-	-	-	✓	-	-
	M_PHONE2	Varchar2	30	-	-	-	✓	-	-
	M_GENDER	Varchar2	30	-	-	-	✓	-	-
	M_DOB	Date	7	-	-	-	✓	-	-
	M_JOINDATE	Date	7	-	-	-	✓	-	-
	M_MARITALSTATUS	Varchar2	30	-	-	-	✓	-	-
	M_BLOODGROUP	Varchar2	30	-	-	-	✓	-	-

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table FINANCE
(
f_id number(5) constraint finance_fid_pk primary key,
amount number(10),
m_id number(10) constraint finance_mid_fk references manager(m_id)
)
desc finance
```

Results Explain Describe Saved SQL History

Object Type TABLE Object FINANCE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FINANCE	F_ID	Number	-	5	0	1	-	-	-
	AMOUNT	Number	-	10	0	-	✓	-	-
	M_ID	Number	-	10	0	-	✓	-	-

1 - 3

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table BILL
(
b_id number(5) constraint bill_bid_pk primary key,
b_amount number(10) constraint bill_amount_ck check(b_amount > 0),
m_id number(10) constraint bill_mid_fk references manager(m_id)
)
desc BILL
```

Results Explain Describe Saved SQL History

Object Type TABLE Object BILL

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BILL	B_ID	Number	-	5	0	1	-	-	-
	B_AMOUNT	Number	-	10	0	-	✓	-	-
	M_ID	Number	-	10	0	-	✓	-	-

1 - 3

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table admin_waiter
(
aw_id number(10) constraint adminwaiter_awid_pk primary key,
a_id number(10) constraint adminwaiter_aid_fk references admin(a_id),
w_id number(10) constraint adminwaiter_wid_fk references waiter(w_id)
)
desc admin_waiter
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ADMIN_WAITER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADMIN_WAITER	AW_ID	Number	-	10	0	1	-	-	-
	A_ID	Number	-	10	0	-	✓	-	-
	W_ID	Number	-	10	0	-	✓	-	-

1 - 3

Language: en-us

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table admin_table
(
at_id number(10) constraint admintable_atid_pk primary key,
a_id number(10) constraint admintable_aid_fk references admin(a_id),
t_id number(10) constraint admintable_tid_fk references rs_table(t_id)
)
desc admin_table
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ADMIN_TABLE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADMIN_TABLE	AT_ID	Number	-	10	0	1	-	-	-
	A_ID	Number	-	10	0	-	✓	-	-
	T_ID	Number	-	10	0	-	✓	-	-

1 - 3

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table admin_ORDER
(
ao_id number(10) constraint adminorder_aoid_pk primary key,
a_id number(10) constraint adminorder_aid_fk references admin(a_id),
o_id number(10) constraint adminorder_oid_fk references table_order(o_id)
)
desc admin_ORDER
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ADMIN_ORDER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADMIN_ORDER	AO_ID	Number	-	10	0	1	-	-	-
	A_ID	Number	-	10	0	-	✓	-	-
	O_ID	Number	-	10	0	-	✓	-	-

1 - 3

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table admin_menu
(
ame_id number(10) constraint adminmenu_ameid_pk primary key,
a_id number(10) constraint adminmenu_aid_fk references admin(a_id),
me_id number(10) constraint adminmenu_meid_fk references menu(am_id)
)
desc admin_menu
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ADMIN_MENU

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADMIN_MENU	AME_ID	Number	-	10	0	1	-	-	-
	A_ID	Number	-	10	0	-	✓	-	-
	ME_ID	Number	-	10	0	-	✓	-	-

1 - 3

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table admin_manager
(
am_id number(10) constraint adminmanager_amid_pk primary key,
a_id number(10) constraint adminmanager_aid_fk references admin(a_id),
m_id number(10) constraint adminmanager_mid_fk references manager(m_id)
)
desc admin_manager
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ADMIN_MANAGER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADMIN_MANAGER	AM_ID	Number	-	10	0	1	-	-	-
	A_ID	Number	-	10	0	-	✓	-	-
	M_ID	Number	-	10	0	-	✓	-	-

1 - 3

User: SCOTT

lome > SQL > SQL Commands

Autocommit Display 10

```
create table ADMIN
(
a_id number(10) constraint admin_aid_pk primary key,
a_name varchar2(30),
a_email varchar2(30),
a_phone1 varchar2(30),
a_phone2 varchar2(30),
a_gender varchar2(30),
a_DOB DATE,
a_joindate DATE,
a_maritalstatus varchar2(30),
a_bloodgroup varchar2(30),
a_salary number(30) constraint admin_salary_ck check(a_salary>0),
c_id number(10) constraint admin_cid_fk references address(c_id)
)
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ADMIN

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADMIN	A_ID	Number	-	10	0	1	-	-	-
	A_NAME	Varchar2	30	-	-	-	✓	-	-
	A_EMAIL	Varchar2	30	-	-	-	✓	-	-
	A_PHONE1	Varchar2	30	-	-	-	✓	-	-
	A_PHONE2	Varchar2	30	-	-	-	✓	-	-
	A_GENDER	Varchar2	30	-	-	-	✓	-	-
	A_DOB	Date	7	-	-	-	✓	-	-
	A_JOINDATE	Date	7	-	-	-	✓	-	-
	A_MARITALSTATUS	Varchar2	30	-	-	-	✓	-	-
	A_BLOODGROUP	Varchar2	30	-	-	-	✓	-	-
	A_SALARY	Number	-	30	0	-	✓	-	-

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table address
(
c_id number(10) constraint address_cid_pk primary key,
city varchar2(30),
country varchar2(30)
)
desc address
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ADDRESS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADDRESS	C_ID	Number	-	10	0	1	-	-	-
	CITY	Varchar2	30	-	-	-	✓	-	-
	COUNTRY	Varchar2	30	-	-	-	✓	-	-

1 - 3

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create index adminmenu_aid_idx
on admin_menu(a_id)

create index adminorder_oid_idx
on admin_order(o_id)

create index admintable_tid_idx
on admin_table(t_id)

create index adminwaiter_wid_idx
on admin_waiter(w_id)

create index bill_bamount_idx
on bill(b_amount)
```

Results Explain Describe Saved SQL History

```
Index created.
```

0.09 seconds

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table waiter_table
(
wt_id number(5) constraint waitertable_wtid_pk primary key,
w_id number(10) constraint waitertable_wid_fk references waiter(w_id),
t_id number(10) constraint waitertable_tid_fk references rs_table(t_id)
)
desc waiter_table
```

Results Explain Describe Saved SQL History

Object Type TABLE Object WAITER_TABLE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
WAITER_TABLE	WT_ID	Number	-	5	0	1	-	-	-
	W_ID	Number	-	10	0	-	✓	-	-
	T_ID	Number	-	10	0	-	✓	-	-

1 - 3

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create index admin_aname_idx
on admin(a_name)

create index waiter_wname_idx
on waiter(w_name)

create index manager_mname_idx
on manager(m_name)

create index address_city_idx
on address(city)

create index adminmanager_aid_idx
on admin_manager(a_id)
```

Results Explain Describe Saved SQL History

```
Index created.
```

0.09 seconds

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
create index finance_amount_idx
on finance(amount)

create index menu_item_idx
on menu(item)

create index rstable_bamount_idx
on rs_table(status)

create index tableorder_item_idx
on table_order(item)

create index waitertable_wid_idx
on waiter_table(w_id)
```

Results Explain Describe Saved SQL History

```
Index created.
```

0.09 seconds

Home > SQL > SQL Commands

Autocommit Display 10

```
CREATE SEQUENCE oid_seq
    START WITH 1
    INCREMENT BY 1
    MAXVALUE 20000
    NOCYCLE
    NOcache;
CREATE SEQUENCE ameid_seq
    START WITH 1
    INCREMENT BY 1
    MAXVALUE 20
    NOCYCLE
    NOcache;
CREATE SEQUENCE meid_seq
    START WITH 1
    INCREMENT BY 1
    MAXVALUE 20
    NOCYCLE
    NOcache;
CREATE SEQUENCE fid_seq
    START WITH 1
    INCREMENT BY 1
    MAXVALUE 2000000
    NOCYCLE
    NOcache;

create index admin_aname_idx
on admin(a_name)

create index waiter_wname_idx
on waiter(w_name)

create index manager_mname_idx
on manager(m_name)

create index address_city_idx
on address(city)

create index adminmanager_aid_idx
on admin_manager(a_id)

create index adminmenu_aid_idx
on admin_menu(a_id)

create index adminorder_oid_idx
on admin_order(o_id)
```

ORACLE Database Express Edition

User: RMS

Home > SQL > SQL Commands

Autocommit Display 10

```
)

create table waiter_table
(
wt_id number(5) constraint waitertable_wtid_pk primary key,
w_id number(10) constraint waitertable_wid_fk references waiter(w_id),
t_id number(10) constraint waitertable_tid_fk references rs_table(t_id)
)

create table admin_manager
(
am_id number(10) constraint adminmanager_amid_pk primary key,
a_id number(10) constraint adminmanager_aid_fk references admin(a_id),
m_id number(10) constraint adminmanager_mid_fk references manager(m_id)
)
desc admin_manager

create table address
(
c_id number(10) constraint address_cid_pk primary key,
city varchar2(30),
country varchar2(30)
)
CREATE SEQUENCE mid_seq
    START WITH 3000
    INCREMENT BY 1
    MAXVALUE 3999
    NOCYCLE
    NOcache;
CREATE SEQUENCE aid_seq
    START WITH 1
    INCREMENT BY 1
    MAXVALUE 10
    NOCYCLE
    NOcache;
CREATE SEQUENCE cid_seq
    START WITH 20
    INCREMENT BY 1
    MAXVALUE 1000
    NOCYCLE
    NOcache;
CREATE SEQUENCE amid_seq
    START WITH 1000
    INCREMENT BY 1
    MAXVALUE 1999
    NOCYCLE NOcache
```

ORACLE Database Express Edition

User: RMS

Home > SQL > SQL Commands

Autocommit Display 10

```
create table admin
(
a_id number(10) constraint manager_mid_pk primary key,
a_name varchar2(30),
a_email varchar2(30),
a_phone1 varchar2(30),
a_phone2 varchar2(30),
a_gender varchar2(30),
a_DOB DATE,
a_joindate DATE,
a_maritalstatus varchar2(30),
a_bloodgroup varchar2(30),
a_salary number(30) constraint admin_salary_ck check(a_salary>0),
c_id number(10) constraint admin_cid_fk references address(c_id)
)

create table RS_TABLE
(
t_id number(10) constraint rstable_tid_pk primary key,
status varchar2(10),
capacity number(4)
)

create table BILL
(
b_id number(5) constraint bill_bid_pk primary key,
b_amount number(10) constraint bill_amount_ck check(b_amount > 0),
m_id number(10) constraint bill_mid_fk references manager(m_id)
)

create table FINANCE
(
f_id number(5) constraint finance_fid_pk primary key,
amount number(10),
m_id number(10) constraint finance_mid_fk references manager(m_id)
)

create table admin_menu
(
ame_id number(10) constraint adminmenu_ameid_pk primary key,
a_id number(10) constraint adminmenu_aid_fk references admin(a_id),
me_id number(10) constraint adminmenu_meid_fk references menu(am_id)
)
```

ORACLE Database Express Edition

User: RMS

Home > SQL > SQL Commands

Autocommit Display 10

```
create table MANAGER
(
m_id number(10) constraint manager_mid_pk primary key,
m_name varchar2(30),
m_email varchar2(30),
m_phone1 varchar2(30),
m_phone2 varchar2(30),
m_gender varchar2(30),
m_DOB DATE,
m_joindate DATE,
m_maritalstatus varchar2(30),
m_bloodgroup varchar2(30),
m_salary number(30) constraint manager_salary_ck check(m_salary>0),
c_id number(10) constraint manager_cid_fk references address(c_id)
)
desc manager

create table WAITER
(
w_id number(10) constraint waiter_wid_pk primary key,
w_name varchar2(30),
w_email varchar2(30),
w_phone1 varchar2(30),
w_phone2 varchar2(30),
w_gender varchar2(30),
w_DOB DATE,
w_joindate DATE,
w_maritalstatus varchar2(30),
w_bloodgroup varchar2(30),
w_salary number(30) constraint waiter_salary_ck check(w_salary>0),
c_id number(10) constraint waiter_cid_fk references address(c_id),
m_id number(10) constraint waiter_mid_fk references manager(m_id)
)
desc waiter

create table table_ORDER
(
o_id number(10) constraint order_oid_pk primary key,
item varchar2(30),
quality varchar2(30),
price number(5) constraint order_price_ck check(price>0),
m_id number(10) constraint order_mid_fk references manager(m_id)
)
```

Home > SQL > SQL Commands

Autocommit Display 10

```
    NOcache;

create index admin_aname_idx
on admin(a_name)

create index waiter_wname_idx
on waiter(w_name)

create index manager_mname_idx
on manager(m_name)

create index address_city_idx
on address(city)

create index adminmanager_aid_idx
on admin_manager(a_id)

create index adminmenu_aid_idx
on admin_menu(a_id)

create index adminorder_oid_idx
on admin_order(o_id)

create index admintable_tid_idx
on admin_table(t_id)

create index adminwaiter_wid_idx
on admin_waiter(w_id)

create index bill_bamount_idx
on bill(b_amount)

create index finance_amount_idx
on finance(amount)

create index menu_item_idx
on menu(item)

create index rstable_bamount_idx
on rs_table(status)

create index tb_order_item_idx
on tb_order(item)

create index waitertable_wid_idx
on waiter_table(w_id)
```

Results Explain Describe Saved SQL

ORACLE Database Express Edition

User: RMS

Home > SQL > SQL Commands

Autocommit Display 10

```
)

create table MENU
(
am_id number(10) constraint menu_meid_pk primary key,
item varchar2(30),
category varchar2(30),
price number(5) constraint menu_price_ck check(price>0),
w_id number(10) constraint menu_wid_fk references waiter(w_id),
m_id number(10) constraint menu_mid_fk references manager(m_id)
)

create table admin_ORDER
(
ao_id number(10) constraint adminorder_aoid_pk primary key,
a_id number(10) constraint adminorder_aid_fk references admin(a_id),
o_id number(10) constraint adminorder_oid_fk references table_order(o_id)
)

create table ORDER
(
o_id number(10) constraint order_oid_pk primary key,
item varchar2(30),
quality varchar2(30),
price number(5) constraint order_price_ck check(price>0),
m_id number(10) constraint order_mid_fk references manager(m_id)
)

create table admin_table
(
at_id number(10) constraint admintable_atid_pk primary key,
a_id number(10) constraint admintable_aid_fk references admin(a_id),
t_id number(10) constraint admintable_tid_fk references rs_table(t_id)
)

create table admin_waiter
(
aw_id number(10) constraint adminwaiter_awid_pk primary key,
a_id number(10) constraint adminwaiter_aid_fk references admin(a_id),
w_id number(10) constraint adminwaiter_wid_fk references waiter(w_id)
)
```

ORACLE Database Express Edition

User: RMS

Home > SQL > SQL Commands

Autocommit Display 10

```
)

create table waiter_table
(
wt_id number(5) constraint waitertable_wtid_pk primary key,
w_id number(10) constraint waitertable_wid_fk references waiter(w_id),
t_id number(10) constraint waitertable_tid_fk references rs_table(t_id)
)

create table admin_manager
(
am_id number(10) constraint adminmanager_amid_pk primary key,
a_id number(10) constraint adminmanager_aid_fk references admin(a_id),
m_id number(10) constraint adminmanager_mid_fk references manager(m_id)
)
desc admin_manager

create table address
(
c_id number(10) constraint address_cid_pk primary key,
city varchar2(30),
country varchar2(30)
)
CREATE SEQUENCE mid_seq
    START WITH 3000
    INCREMENT BY 1
    MAXVALUE 3999
    NOCYCLE
    NOcache;
CREATE SEQUENCE aid_seq
    START WITH 1
    INCREMENT BY 1
    MAXVALUE 10
    NOCYCLE
    NOcache;
CREATE SEQUENCE cid_seq
    START WITH 20
    INCREMENT BY 1
    MAXVALUE 1000
    NOCYCLE
    NOcache;
CREATE SEQUENCE amid_seq
    START WITH 1000
    INCREMENT BY 1
    MAXVALUE 1999
    NOCYCLE NOcache
```

Data Insertion:

```
INSERT INTO  addressVALUES(CID_SEQ.NEXTVAL,'MUNICH','GERMANY')

INSERT INTO address VALUES (CID_SEQ.NEXTVAL,'TORONTO','CANADA')

INSERT INTO address VALUES (CID_SEQ.NEXTVAL,'DHAKA','BANGLADESH')

INSERT INTO address VALUES (CID_SEQ.NEXTVAL,'MELBOURNE','AUSTRALLIA')

INSERT INTO address VALUES (CID_SEQ.NEXTVAL,'AMSTERDAM','NETHERLANDS')
```

```
INSERT INTO admin VALUES
(AID_SEQ.NEXTVAL,'MOWMITA','NAMOWMITA@GMAIL.COM','01402050978','01798872647','FEMALE','02-FEB-98','01-MAR-20','SINGLE','A+',50000,20)

INSERT INTO admin VALUES
(AID_SEQ.NEXTVAL,'PRODIPTA','PRODIPTA@GMAIL.COM','01402353978','017988962647','FEMALE','02-FEB-94','01-MAR-20','SINGLE','A+',50000,21)

INSERT INTO admin VALUES
(AID_SEQ.NEXTVAL,'IFFAT','IFFAT@GMAIL.COM','014023539213','017988962612','FEMALE','02-APR-99','01-MAR-20','SINGLE','O+',50000,22)

INSERT INTO admin VALUES
(AID_SEQ.NEXTVAL,'MANIK','MANIK@GMAIL.COM','01402373978','017998962647','MALE','17-DEC-98','01-MAR-20','SINGLE','B+',50000,23)

INSERT INTO admin VALUES
(AID_SEQ.NEXTVAL,'NAHRIN','NAHRIN@GMAIL.COM','01402785978','01745962647','FEMALE','24-JUN-98','01-MAR-20','SINGLE','AB+',50000,24)
```

INSERT INTO MANAGER VALUES
(MID_SEQ.NEXTVAL,'AUTHOI','AUTHOI@GMAIL.COM','014020588878','01798888847','FEMALE','22-FEB-94','04-MAR-20','SINGLE','A+',20000,22)

INSERT INTO MANAGER VALUES
(MID_SEQ.NEXTVAL,'ROSS','ROSS@GMAIL.COM','014020528878','01798338847','MALE','26-FEB-94','04-MAR-20','SINGLE','A-',20000,22)

INSERT INTO MANAGER VALUES
(MID_SEQ.NEXTVAL,'PENNY','PENNYI@GMAIL.COM','014020582378','01714888847','FEMALE','22-JUL-94','04-MAR-20','SINGLE','O+',20000,22)

INSERT INTO MANAGER VALUES
(MID_SEQ.NEXTVAL,'CHANDLER','CHANDLER@GMAIL.COM','014020588278','0179888827','MALE','21-AUG-94','04-MAR-20','SINGLE','B-',20000,22)

INSERT INTO MANAGER VALUES
(MID_SEQ.NEXTVAL,'BERNIDEETHE','BERNIDEETHE@GMAIL.COM','014020581278','01798881447','FEMALE','12-SEP-94','04-MAR-20','SINGLE','A+',20000,22)

INSERT INTO WAITER VALUES
(WID_SEQ.NEXTVAL,'MIM','MIM@GMAIL.COM','014020588878','01798888847','FEMALE','22-FEB-94','04-MAR-20','SINGLE','A+',20000,22,3000)

INSERT INTO WAITER VALUES
(WID_SEQ.NEXTVAL,'MINI','MINI@GMAIL.COM','014020528878','01798338847','FEMALE','26-FEB-94','04-MAR-20','SINGLE','A-',20000,22,3001)

INSERT INTO WAITER VALUES
(WID_SEQ.NEXTVAL,'BABUL','BABUL@GMAIL.COM','014020582378','01714888847','MALE','22-JUL-94','04-MAR-20','SINGLE','O+',20000,22,3002)

```
INSERT INTO WAITER VALUES
(WID_SEQ.NEXTVAL,'ABUL','ABUL@GMAIL.COM','014020588278','0179888827','MALE','21-AUG-94','04-MAR-20','SINGLE','B-',20000,22,3003)

INSERT INTO WAITER VALUES
(WID_SEQ.NEXTVAL,'MIMI','MIMI@GMAIL.COM','014020581278','01798881447','FEMALE','12-SEP-94','04-MAR-20','SINGLE','A+',20000,22,3004)

DELETE * FROM MANAGER WHERE M_NAME='MIM'
(WID_SEQ.NEXTVAL,'MIM','MIM@GMAIL.COM','014020588878','01798888847','FEMALE','22-FEB-94','04-MAR-20','SINGLE','A+',20000,22,3000)

INSERT INTO ADMIN_MANAGER VALUES (AMID_SEQ.NEXTVAL,2,3000)

INSERT INTO ADMIN_MANAGER VALUES (AMID_SEQ.NEXTVAL,3,3001)

INSERT INTO ADMIN_MANAGER VALUES (AMID_SEQ.NEXTVAL,4,3002)

INSERT INTO ADMIN_MANAGER VALUES (AMID_SEQ.NEXTVAL,5,3003)

INSERT INTO ADMIN_MANAGER VALUES (AMID_SEQ.NEXTVAL,6,3004)

INSERT INTO ADMIN_WAITER VALUES (AWID_SEQ.NEXTVAL,2,4000)

INSERT INTO ADMIN_WAITER VALUES (AWID_SEQ.NEXTVAL,3,4001)

INSERT INTO ADMIN_WAITER VALUES (AWID_SEQ.NEXTVAL,4,4002)

INSERT INTO ADMIN_WAITER VALUES (AWID_SEQ.NEXTVAL,5,4003)
```

```
INSERT INTO ADMIN_WAITER VALUES (AWID_SEQ.NEXTVAL,6,4000)



INSERT INTO MENU VALUES (MEID_SEQ.NEXTVAL,'MILKSHAKE','BEVERAGE',120,4000,3001)

INSERT INTO MENU VALUES (MEID_SEQ.NEXTVAL,'BURGER','MEAL',120,4000,3001)

INSERT INTO MENU VALUES (MEID_SEQ.NEXTVAL,'PIZZA','MEAL',120,4000,3001)

INSERT INTO MENU VALUES (MEID_SEQ.NEXTVAL,'CAKE','DESERT',120,4000,3002)

INSERT INTO MENU VALUES (MEID_SEQ.NEXTVAL,'FRIES','APPETIZER',120,4002,3004)



INSERT INTO TB_ORDER VALUES (OID_SEQ.NEXTVAL,'BURGER','2',240,3001)

INSERT INTO TB_ORDER VALUES (OID_SEQ.NEXTVAL,'PIZZA','2',240,3002)

INSERT INTO TB_ORDER VALUES (OID_SEQ.NEXTVAL,'PIZZA','2',240,3003)

INSERT INTO TB_ORDER VALUES (OID_SEQ.NEXTVAL,'MILKSHAKE','2',240,3001)

INSERT INTO TB_ORDER VALUES (OID_SEQ.NEXTVAL,'FRIES','2',240,3001)



INSERT INTO RS_TABLE VALUES (TID_SEQ.NEXTVAL,'OK',5)
```

```
INSERT INTO RS_TABLE VALUES (TID_SEQ.NEXTVAL,'OK',4)

INSERT INTO RS_TABLE VALUES (TID_SEQ.NEXTVAL,'OK',2)

INSERT INTO RS_TABLE VALUES (TID_SEQ.NEXTVAL,'OK',2)

INSERT INTO RS_TABLE VALUES (TID_SEQ.NEXTVAL,'OK',4)
```

```
INSERT INTO ADMIN_MENU VALUES (AMID_SEQ.NEXTVAL,2,4)

INSERT INTO ADMIN_MENU VALUES (AMID_SEQ.NEXTVAL,3,6)

INSERT INTO ADMIN_MENU VALUES (AMID_SEQ.NEXTVAL,2,3)

INSERT INTO ADMIN_MENU VALUES (AMID_SEQ.NEXTVAL,4,5)

INSERT INTO ADMIN_MENU VALUES (AMID_SEQ.NEXTVAL,2,4)
```

```
INSERT INTO ADMIN_ORDER VALUES (AOID_SEQ.NEXTVAL,2,2)

INSERT INTO ADMIN_ORDER VALUES (AOID_SEQ.NEXTVAL,2,3)

INSERT INTO ADMIN_ORDER VALUES (AOID_SEQ.NEXTVAL,2,4)

INSERT INTO ADMIN_ORDER VALUES (AOID_SEQ.NEXTVAL,2,5)

INSERT INTO ADMIN_ORDER VALUES (AOID_SEQ.NEXTVAL,2,6)
```

```
INSERT INTO ADMIN_TABLE VALUES (ATID_SEQ.NEXTVAL,2,1)

INSERT INTO ADMIN_TABLE VALUES (ATID_SEQ.NEXTVAL,2,2)

INSERT INTO ADMIN_TABLE VALUES (ATID_SEQ.NEXTVAL,2,3)

INSERT INTO ADMIN_TABLE VALUES (ATID_SEQ.NEXTVAL,2,4)

INSERT INTO ADMIN_TABLE VALUES (ATID_SEQ.NEXTVAL,2,5)



INSERT INTO BILL VALUES (BID_SEQ.NEXTVAL,240,3001)

INSERT INTO BILL VALUES (BID_SEQ.NEXTVAL,120,3001)

INSERT INTO BILL VALUES (BID_SEQ.NEXTVAL,440,3001)

INSERT INTO BILL VALUES (BID_SEQ.NEXTVAL,540,3001)

INSERT INTO BILL VALUES (BID_SEQ.NEXTVAL,640,3001)



INSERT INTO FINANCE VALUES (FID_SEQ.NEXTVAL,24000,3001)

INSERT INTO FINANCE VALUES (FID_SEQ.NEXTVAL,4000,3001)
```

INSERT INTO FINANCE VALUES (FID_SEQ.NEXTVAL,6000,3001)

INSERT INTO FINANCE VALUES (FID_SEQ.NEXTVAL,30000,3001)

INSERT INTO FINANCE VALUES (FID_SEQ.NEXTVAL,10000,3001)

INSERT INTO WAITER_TABLE VALUES (WTID_SEQ.NEXTVAL,4000,5)

INSERT INTO WAITER_TABLE VALUES (WTID_SEQ.NEXTVAL,4001,4)

INSERT INTO WAITER_TABLE VALUES (WTID_SEQ.NEXTVAL,4002,3)

INSERT INTO WAITER_TABLE VALUES (WTID_SEQ.NEXTVAL,4003,2)

INSERT INTO WAITER_TABLE VALUES (WTID_SEQ.NEXTVAL,4004,1)

Query Writing:

SINGLE ROW FUNCTION :

1. write a query to show how many month has it been since MIM joined.

SELECT MONTHS_BETWEEN(SYSDATE,W_JOINDATE) "Months" FROM WAITER WHERE W_NAME='MIM';

2. write a query to show the name and show email from the 6th character and replace the previous ones with '*'.

SELECT A_NAME,LPAD(SUBSTR(A_EMAIL, 6), 15, '*')FROM ADMIN;

3. Write a query to show the name of the admins with first letter as uppercase and rest as lowercase.

SELECT INITCAP(A_NAME) FROM ADMIN;

GROUP FUNCTION :

1. Write a query to sum of gross income.

```
SELECT SUM(AMOUNT) GROSS

FROM FINANCE
```

2. write a query to find the maximum bill amount.

```
SELECT MAX(B_AMOUNT) MAX_BILL

FROM BILL
```

3. write a query to find the minimum bill amount.

```
SELECT MIN(B_AMOUNT) MIN_BILL

FROM BILL
```

SUBQUERY :

1. Write a query to show admin names who get paid same as 'MOWMITA'

```
SELECT A_NAME FROM ADMIN WHERE A_SALARY=(SELECT A_SALARY FROM ADMIN WHERE A_NAME='MOWMITA')
```

2. Find which admin supervise the manager who is in charge of the waiter of id 4002.

```
SELECT A_NAME FROM ADMIN,ADMIN_MANAGER AM WHERE AM.M_ID=(SELECT M.M_ID FROM WAITER M WHERE W_ID=4002)
```

3. Write a Query to get price of the table with max quantity.

```
SELECT PRICE FROM TB_ORDER WHERE QUANTITY=(SELECT MAX(QUANTITY) FROM TB_ORDER)
```

JOINING :

1. Write a query to show the manager associated with the waiters.

```
SELECT M_NAME,W_NAME FROM MANAGER M,WAITER W WHERE M.M_ID=W.M_ID
```

2. Write a query to show order item with the menu category .

```
SELECT O.ITEM,M.CATEGORY FROM TB_ORDER O,MENU M WHERE O.ITEM=M.ITEM
```

3. write a query to show admin (id, name), manager (id, name) admin id and manager id is same.

SELECT A.A_ID,A.A_NAME,M.M_ID,M.M_NAME FROM ADMIN A,MANAGER M,ADMIN_MANAGER AM WHERE A.A_ID=AM.A_ID AND M.M_ID=AM.M_ID

VIEW :

1. Create a view that displays the id and the salary of every Waiter.

CREATE VIEW SAL_W_VU AS SELECT W_ID,W_SALARY FROM WAITER

2. Create a view that shows all item with its category.

CREATE VIEW CAT_VU AS SELECT O.ITEM,M.CATEGORY FROM TB_ORDER O,MENU M WHERE O.ITEM=M.ITEM

3. Create a view that shows the names of all the admins who are manager.

CREATE VIEW AMR_VU AS SELECT A.A_ID,A.A_NAME,M.M_ID,M.M_NAME FROM ADMIN A,MANAGER M,ADMIN_MANAGER AM WHERE A.A_ID=AM.A_ID AND M.M_ID=AM.M_ID

SYNONYM :

1. Create a synonym for admin table.

CREATE SYNONYM OWNERS FOR ADMIN

2. Create a synonym AMR_VU(Admin manager relation view).

CREATE SYNONYM ADMIN_MANAGER_RELATION FOR AMR_VU

3. Create a synonym for Finance table.

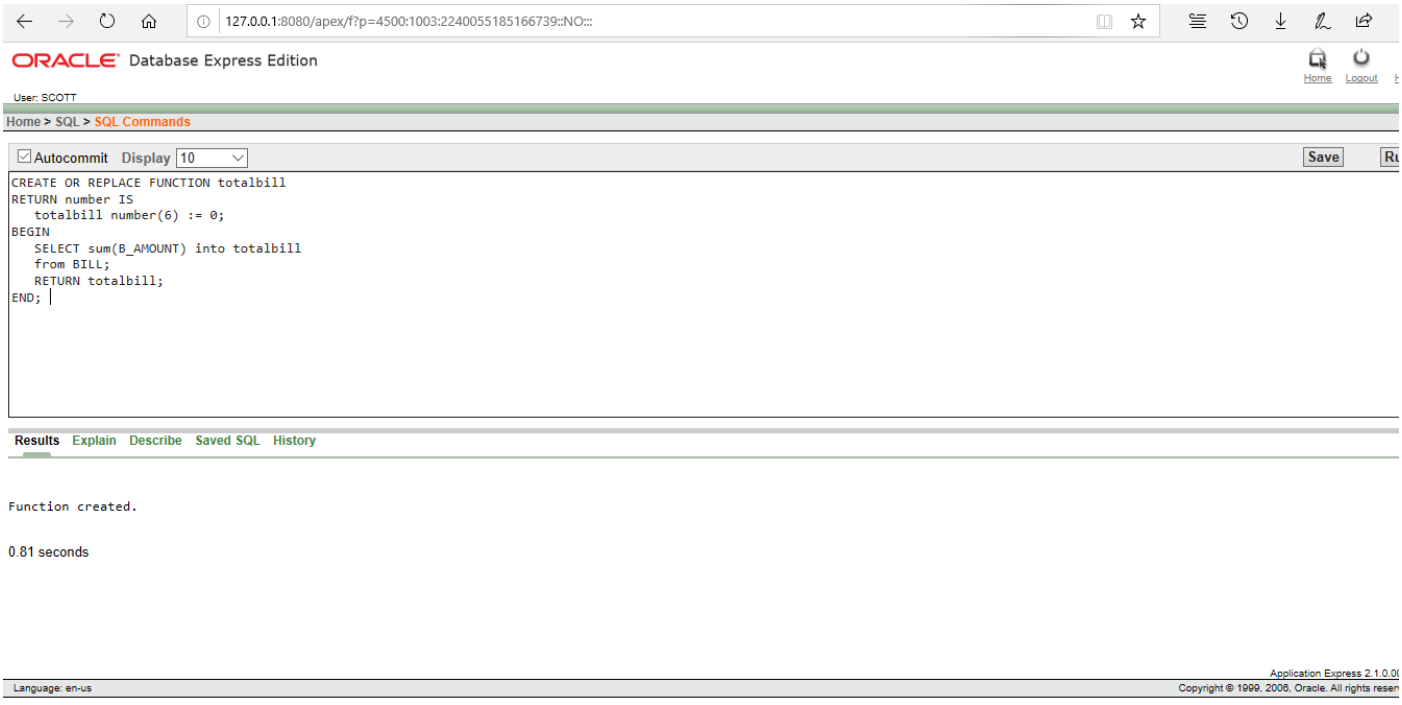
CREATE SYNONYM CALCULATION FOR FINANCE

PL/SQL:

FUNCTION:

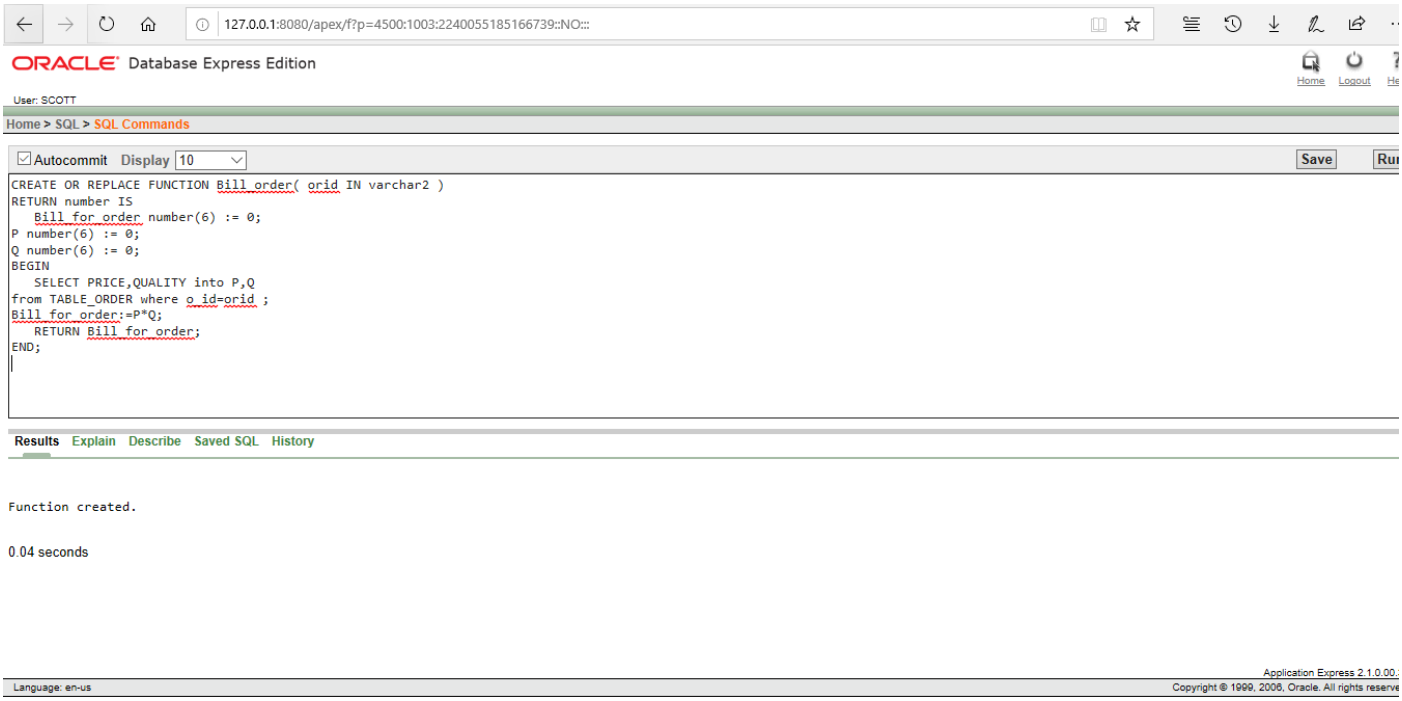
Q1:create a function for calculating totalbill

```
CREATE OR REPLACE FUNCTION totalbill
RETURN number IS
  totalbill number(6) := 0;
BEGIN
  SELECT sum(B_AMOUNT) into totalbill
  from BILL;
  RETURN totalbill;
END;
```



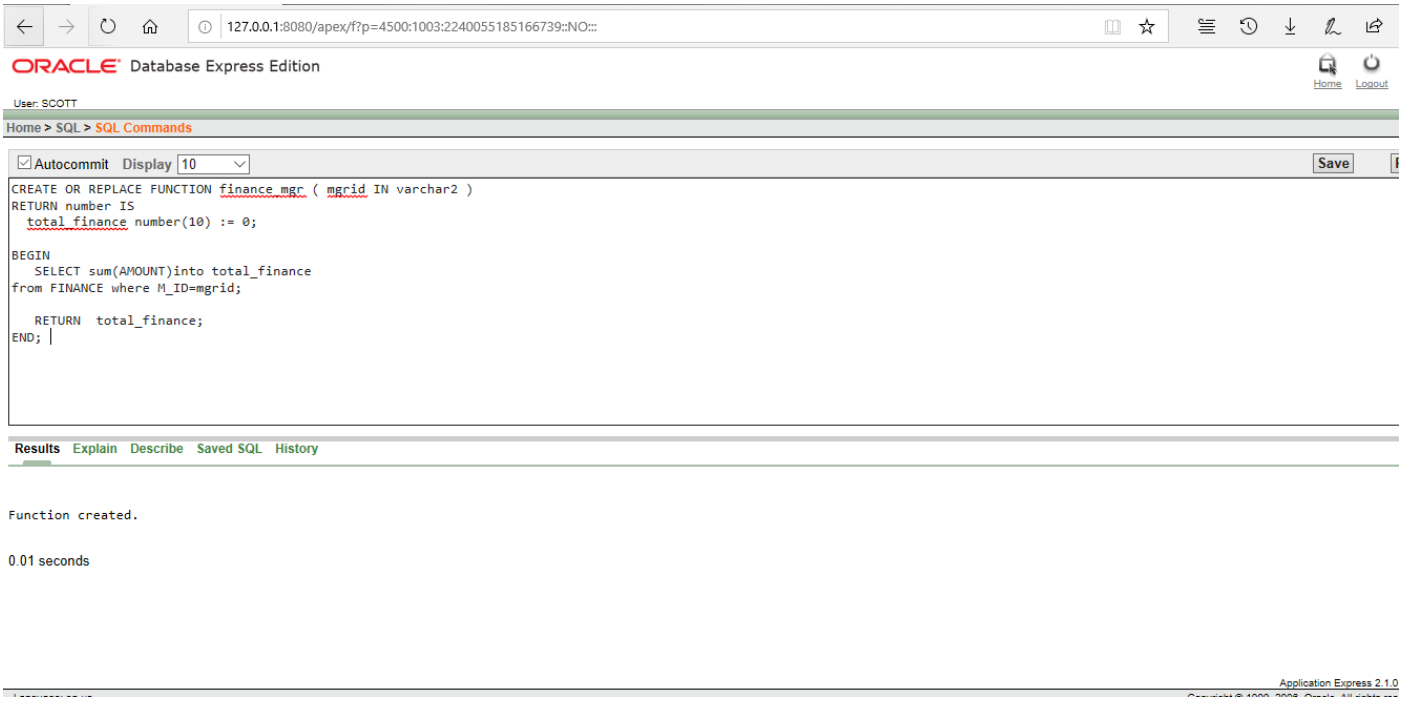
Q2:create a function for calculating bill from specific order number

```
CREATE OR REPLACE FUNCTION Bill_order( orid IN varchar2 )
RETURN number IS
  Bill_for_order number(6) := 0;
P number(6) := 0;
Q number(6) := 0;
BEGIN
  SELECT PRICE,QUANTITY into P,Q
from TB_ORDER where o_id=orid ;
Bill_for_order:=P*Q;
  RETURN Bill_for_order;
END;
```



Q3:create a function for calculating total finance from specific manager

```
CREATE OR REPLACE FUNCTION finance_mgr ( mgrid IN varchar2 )
RETURN number IS
 total_finance number(10) := 0;

BEGIN
  SELECT sum(AMOUNT)into total_finance
from FINANCE where M_ID=mgrid;

  RETURN  total_finance;
END;
```



PROCEDURE:

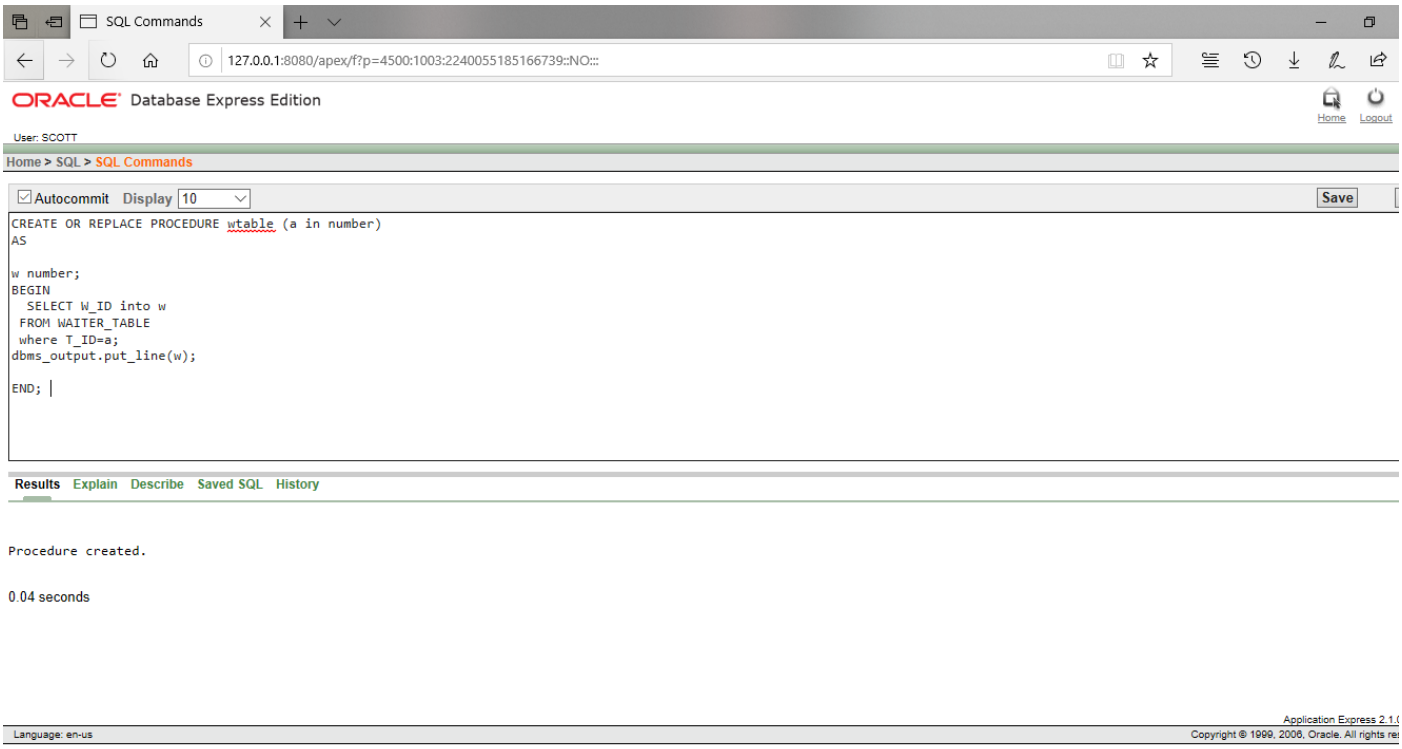
Q1:create a procedure for setting a new salary for all managers

```
CREATE OR REPLACE PROCEDURE set_new_sal (a in number)
AS
BEGIN
  UPDATE MANAGER SET M_SALARY=a;
END;
```

Q2:create a procedure for getting the assigned waiter by the table id

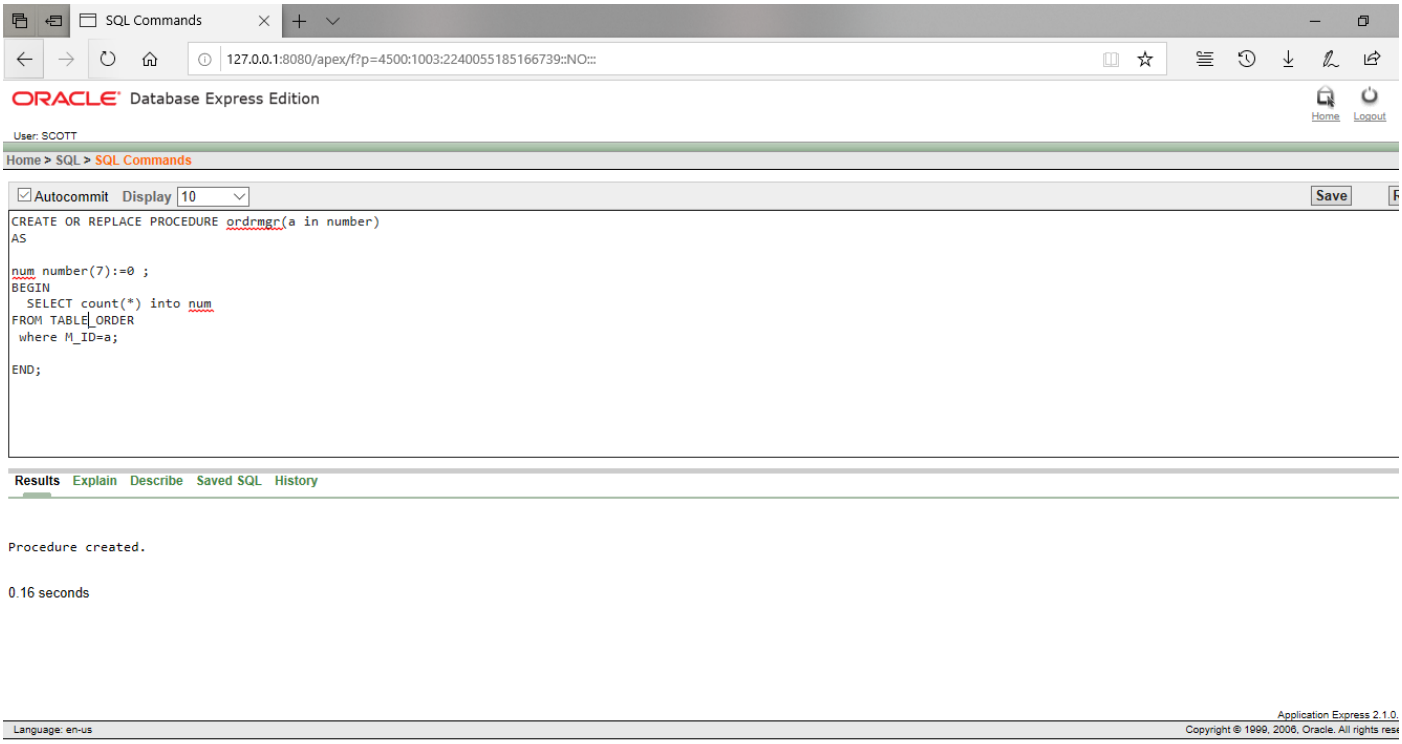
```
CREATE OR REPLACE PROCEDURE wtable (a in number)
AS

w number;
BEGIN
 SELECT W_ID into w
 FROM WAITER_TABLE
 where T_ID=a;
dbms_output.put_line(w);
```

END;



Q3:create a procedure for fetching the total number of order under a manager

```
CREATE OR REPLACE PROCEDURE ordrmgr(a in number)
AS

num number(7):=0 ;
BEGIN
 SELECT count(*) into num
FROM TB_ORDER
 where M_ID=a;

END;
```



Trigger:

QS: Write down a trigger which will fired when any new employee added like an admin, manager or a waiter.

ANSWER:

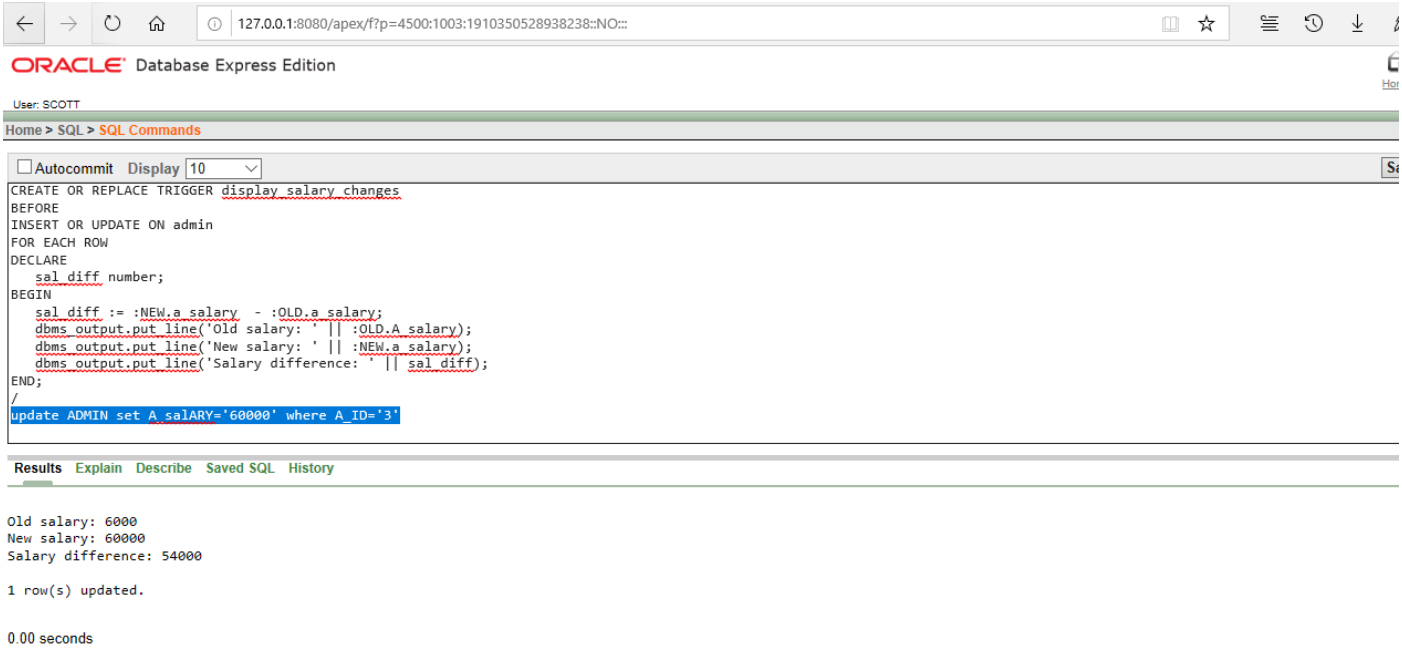
```
CREATE OR REPLACE TRIGGER employee_added
after INSERT ON MANAGER
FOR EACH ROW
BEGIN
```

```
  dbms_output.put_line('New Manager successfully Added');
END;
/
select * from manager;
INSERT INTO MANAGER VALUES
(MID_SEQ.NEXTVAL,'SHAPNA','SHAPNA@GMAIL.COM','014020599278','0179666827','FEMALE','12-AUG-96','04-MAR-20','SINGLE','AB-',20000,22)
```

QS: Write a trigger that display the salary changes of employees.

```
CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE
INSERT OR UPDATE ON admin
FOR EACH ROW
DECLARE
  sal_diff number;
BEGIN
  sal_diff := :NEW.a_salary  - :OLD.a_salary;
```

```
  dbms_output.put_line('Old salary: ' || :OLD.A_salary);
  dbms_output.put_line('New salary: ' || :NEW.a_salary);
  dbms_output.put_line('Salary difference: ' || sal_diff);
END;
/
update ADMIN set A_salARY='60000' where A_ID='3'
```

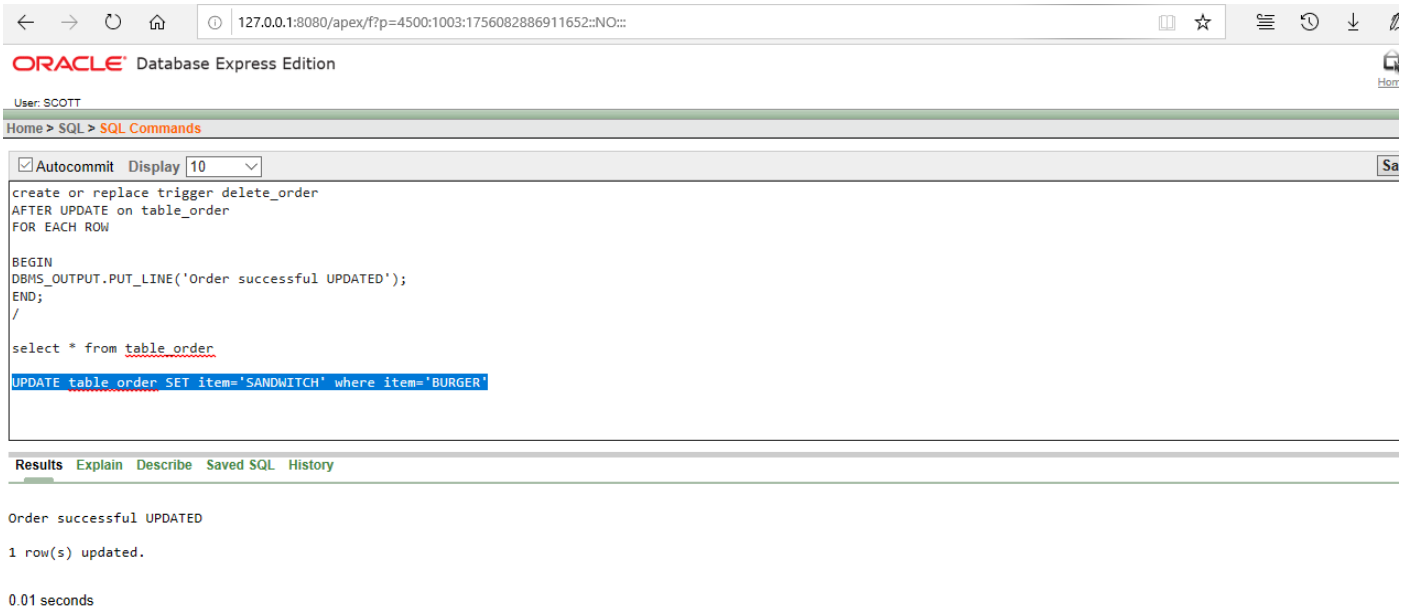


QS: Write a trigger when customer will change or update their order.

```
create or replace trigger delete_order
AFTER UPDATE on table_order
FOR EACH ROW

BEGIN
DBMS_OUTPUT.PUT_LINE('Order successful UPDATED');
END;
/
```

select * from table_order

UPDATE table_order SET item='SANDWITCH' where item='BURGER'

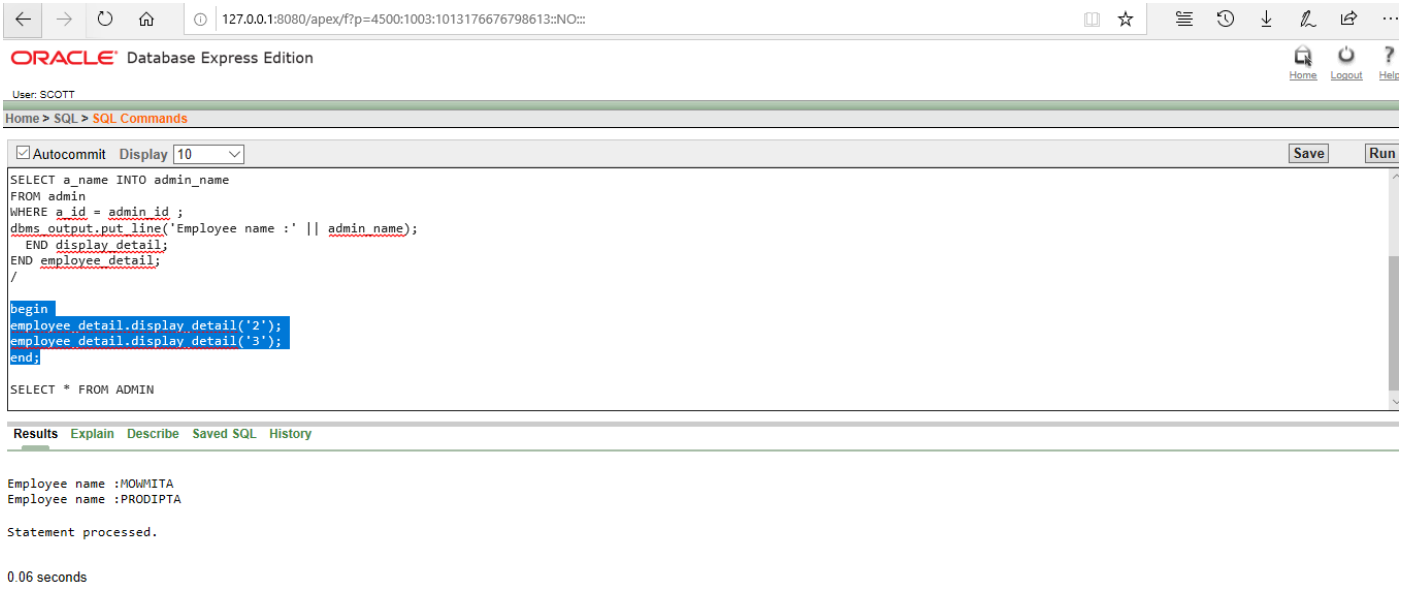


Package:

QS : Create a package that will return the employee name while passing the id of any employee.

```
CREATE OR REPLACE PACKAGE employee_detail AS
  PROCEDURE display_detail(admin_id admin.a_id%type);
END employee_detail;

CREATE OR REPLACE PACKAGE BODY employee_detail AS
  PROCEDURE display_detail(admin_id admin.a_id%type) IS
 admin_name admin.a_name%type;
BEGIN
```

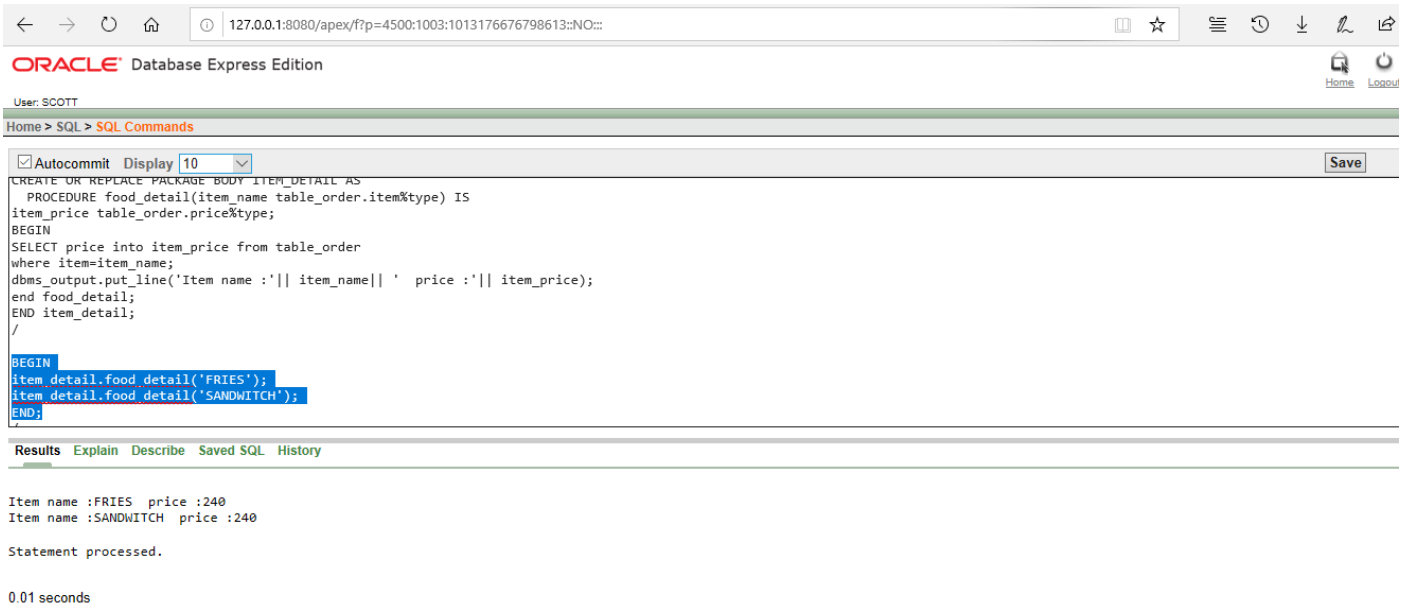
```
SELECT a_name INTO admin_name
FROM admin
WHERE a_id = admin_id ;
dbms_output.put_line('Employee name :' || admin_name);
 END display_detail;
END employee_detail;
/

begin
employee_detail.display_detail('2');
employee_detail.display_detail('3');
end;
/
```



QS: Create a package that will return the price when name will be passed through the parameter.

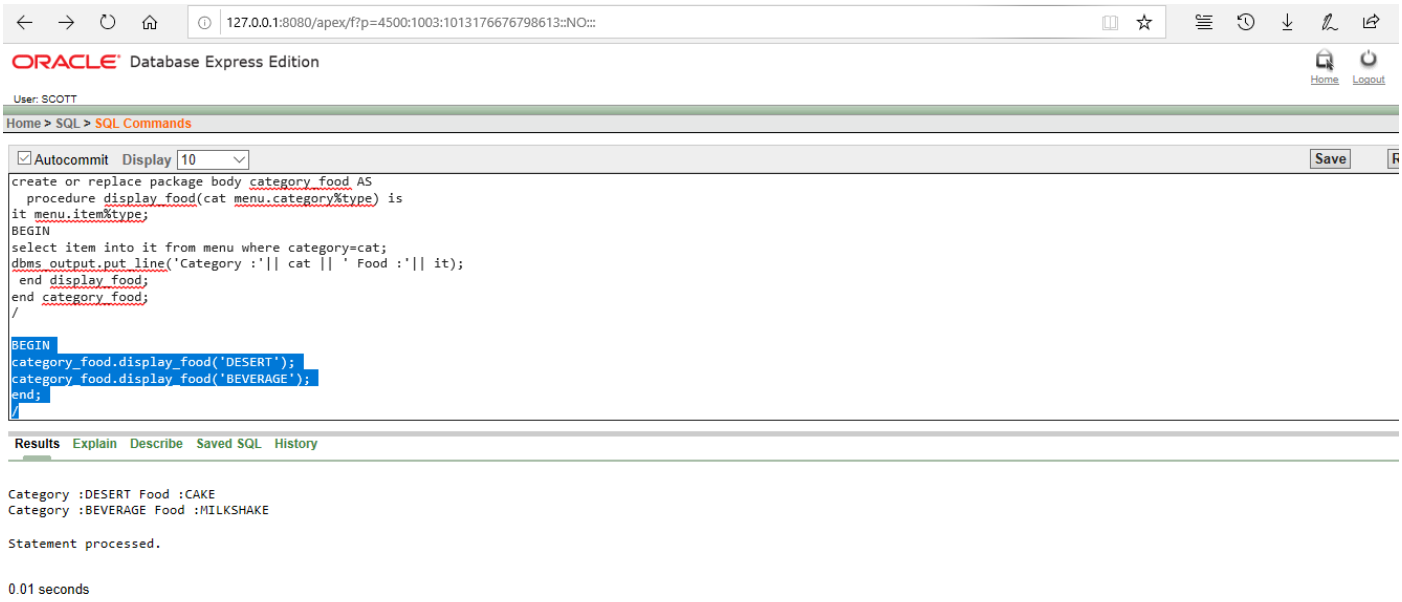
```
select * from table_order

CREATE OR REPLACE PACKAGE item_detail AS
```

```
  PROCEDURE food_detail(item_name table_order.item%type);
END item_detail;



CREATE OR REPLACE PACKAGE BODY ITEM_DETAIL AS
  PROCEDURE food_detail(item_name table_order.item%type) IS
item_price table_order.price%type;
BEGIN
SELECT price into item_price from table_order
where item=item_name;
dbms_output.put_line('Item name :'|| item_name|| '  price :'|| item_price);
end food_detail;
END item_detail;
/

BEGIN
item_detail.food_detail('FRIES');
item_detail.food_detail('SANDWITCH');
END;
```



QS : Create a package that will return an item from any category while passing the category from parameter.

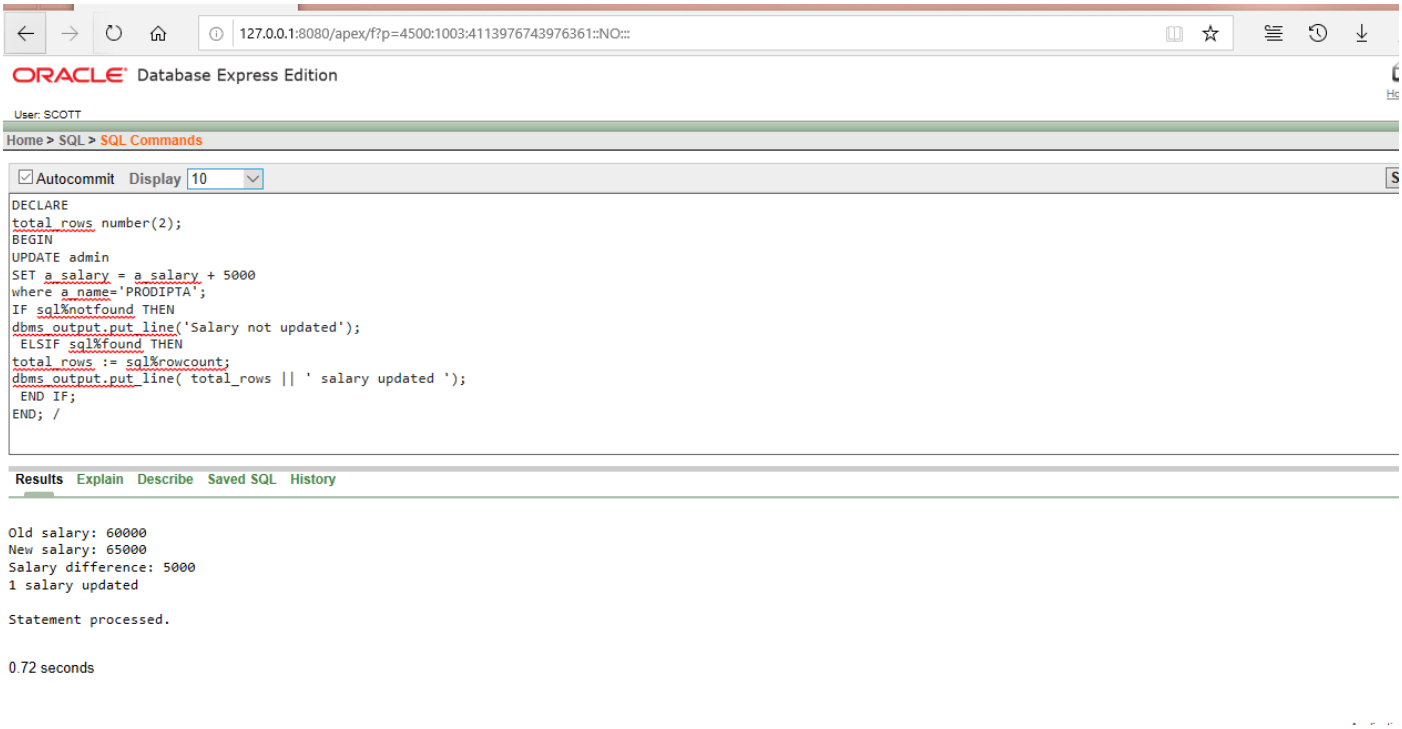
```
create or replace package category_food AS
  procedure display_food(cat menu.category%type);
end category_food;
create or replace package body category_food AS
  procedure display_food(cat menu.category%type) is
it menu.item%type;
BEGIN
select item into it from menu where category=cat;
dbms_output.put_line('Category :'|| cat || ' Food :'|| it);
 end display_food;
end category_food;
/
BEGIN
category_food.display_food('DESERT');
category_food.display_food('BEVERAGE');
end;
/
```



CURSOR :

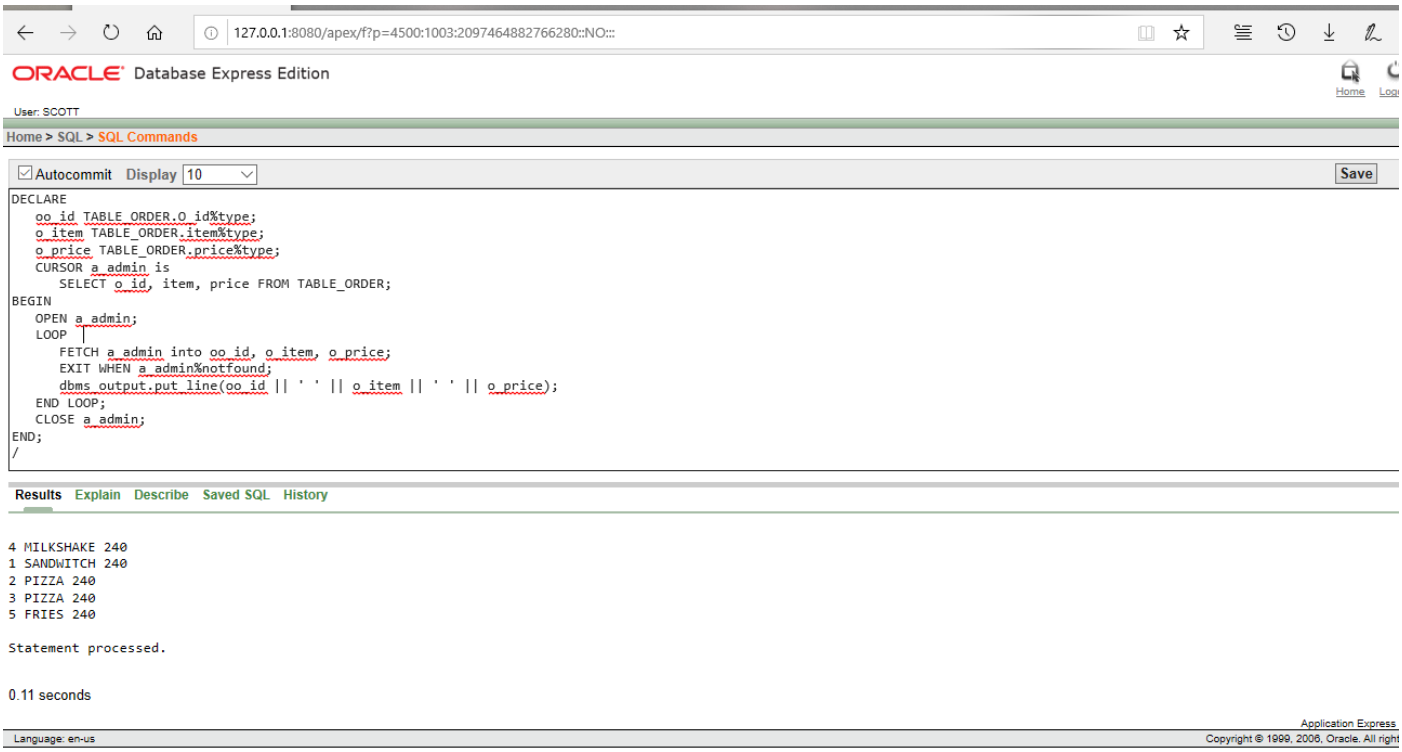
1. write a query to update the table and increase salary of admins by 5000.

```
DECLARE
total_rows number(2);
BEGIN
UPDATE admin
SET a_salary = a_salary + 5000
where a_name='PRODIPTA';
IF sql%notfound THEN
dbms_output.put_line('Salary not updated');
 ELSIF sql%found THEN
total_rows := sql%rowcount;
dbms_output.put_line( total_rows || ' salary updated ');
 END IF;
END;
/
```



2. write a query to retrive order item and price.

```
DECLARE
  oo_id TABLE_ORDER.O_id%type;
  o_item TABLE_ORDER.item%type;
  o_price TABLE_ORDER.price%type;
  CURSOR a_admin is
    SELECT o_id, item, price FROM TABLE_ORDER;
BEGIN
  OPEN a_admin;
  LOOP
    FETCH a_admin into oo_id, o_item, o_price;
    EXIT WHEN a_admin%notfound;
    dbms_output.put_line(oo_id || ' ' || o_item || ' ' || o_price);
  END LOOP;
  CLOSE a_admin;
END;
/
```



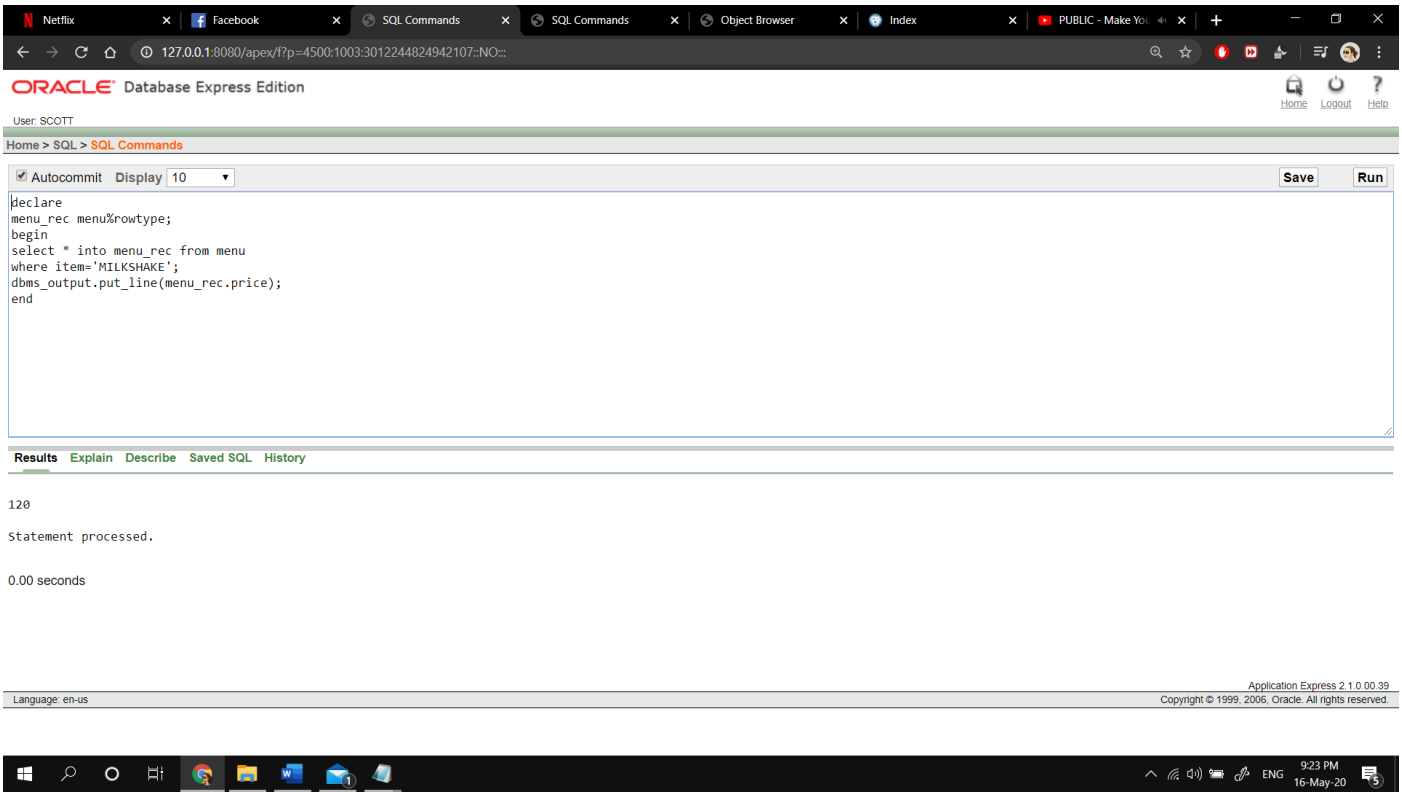
3. write a query to update the table and increase salary of waiter by 25%.

```
DECLARE
total_rows number(2);
BEGIN
UPDATE waiter
SET w_salary = w_salary *.25 + w_salary;
IF sql%notfound THEN
dbms_output.put_line('Salary not updated');
 ELSIF sql%found THEN
total_rows := sql%rowcount;
dbms_output.put_line( total_rows || ' salary updated ');
 END IF;
END;
/
```

RECORD :

Question 1 : Retrieve the price Milkshake from table menu using record.

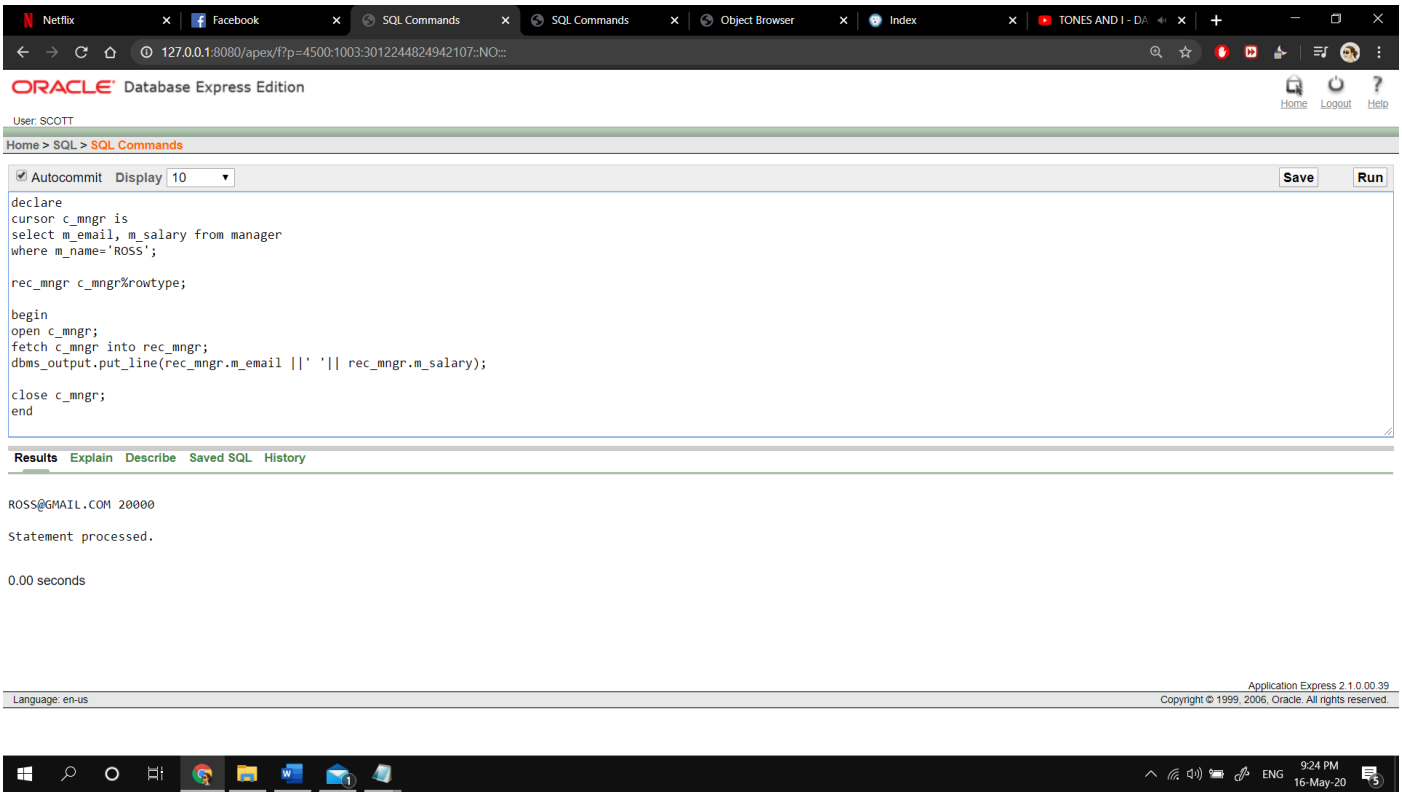
```
declare
menu_rec menu%rowtype;
begin
select * into menu_rec from menu
where item='MILKSHAKE';
dbms_output.put_line(menu_rec.price);
end
```



Question 2 : Retrieve email and salary using cursor based record where manager is ross.

```
declare
cursor c_mngr is
select m_email, m_salary from manager
```

```
where m_name='ROSS';


rec_mngr c_mngr%rowtype;


begin
open c_mngr;
fetch c_mngr into rec_mngr;
dbms_output.put_line(rec_mngr.m_email ||' '|| rec_mngr.m_salary);


close c_mngr;
end
```



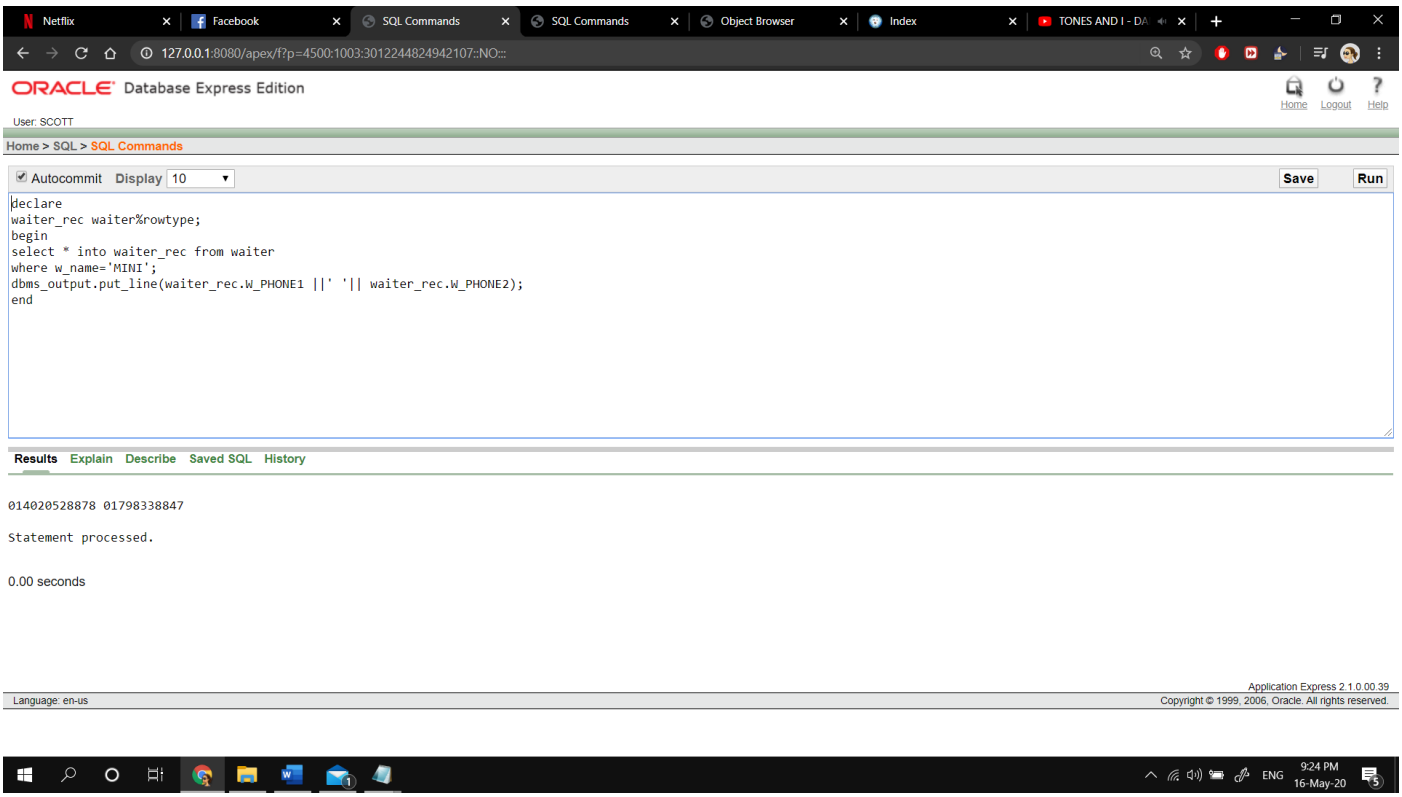
Question 3 : Retrieve phone numbers of waiter mini.

```
declare
waiter_rec waiter%rowtype;
begin
select * into waiter_rec from waiter
```

where w_name='MINI';

dbms_output.put_line(waiter_rec.W_PHONE1 ||' '|| waiter_rec.W_PHONE2);

end



Conclusion:

While doing the project we learnt about how we can manage a restaurant using database and how we can make tasks easier for the users with the help of database. For the future we would like to add more features to the system. For Final term we would develop the application and database for the designed system.