RESTAURANT MANGEMENT SERVICES

DATABASE MANAGEMENT SYSTEMS (ITE1003)

-PROJECT COMPONENT-

REVIEW REPORT

2019- NOVEMBER

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SUBMITTED BY:

ANIKET KUMAR:18BIT0214

SHANTANU DIXIT:18BIT0306

UDIT AGARWAL: 18BIT0159

Abstract:

Restaurant Management System is a service primarily designed for the need of reserving restaurants along with the items the user wanted to be ready before arriving at the restaurant. Not only it will provide convenience to the users by decreasing the waiting time, it will also serve as a better platform for the restaurants to increase the scope of business. When the user uses the service, he will be directed to the Login page where he needs to fill the credentials along with the Password. As soon as the authentication is done List of all the restaurants will be displayed along with some necessary information like Name, Address, contact details etc. the user can choose the restaurant after that he will be asked in which slot he needs to book the table. Thus, user can easily book the restaurant for the available timings. Later users can give the review of the restaurant, which helps to make some priorities for later use.

Tools:

Our project restaurant booking service is developed by using the software

Backend: SQL Plus

SQL Plus is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications.

Data Constraints:

A Restaurant is identified by a Restaurant ID(RID), and their Names, Addresses, Contact Numbers and Cities must be recorded. An Admin is allowed modify or update the Restaurant details. Each Admin is identified by its Name and a Password. Restaurant Bookings are identified by their Booking IDs(BID). For each booking, the Customer ID(CID) of the customer which is making a reservation, Restaurant ID, Date on which the booking is made and Slot must be recorded. Every Customer is identified by its Customer ID and the Name of the customer, Contact Number, Email Address is recorded. If the customer has not logged in yet, then during sign-up he has to enter a Username which is the customer ID and a password which can't be empty. A Customer can give a Feedback for a restaurant. For each feedback, the Customer ID, Restaurant ID and feedback is recorded. Different Waiters serve at Restaurants. Each waiter is identified by its Waiter ID(WID) and Names, Contact Numbers for each waiter must be recorded. Vendors provide Raw Food items to restaurants. For each Vendor Company, their Names, Addresses, Contact Numbers and Email Addresses must be recorded. For each Vendor the Restaurant ID must also be recorded to which they are providing their service. Various Accountants review the bookings. Each Accountant is identified by its Accountant ID(AID) and their Names, Contact Number, Email Addresses must be recorded. In addition, a Booking ID and a Restaurant ID must be associated with each Accountant. Every Restaurant can have multiple bookings but each booking must be associated with only one Restaurant. Customer can book multiple restaurants but each booking should be done by only one customer. A restaurant can have many Feedbacks but each feedback should be of only one restaurant. Every Customer can give multiple feedbacks but each feedback must be associated with only one customer. An Admin can add or delete multiple restaurants but there should be only one Admin. A Restaurant can have multiple waiters serving them but each waiter should be working for only one restaurant. A Vendor Company can serve raw food items to multiple restaurants but each Restaurant should be getting services from only one Vendor Company. One Accountant can review multiple bookings but one booking should be reviewed by only one accountant.

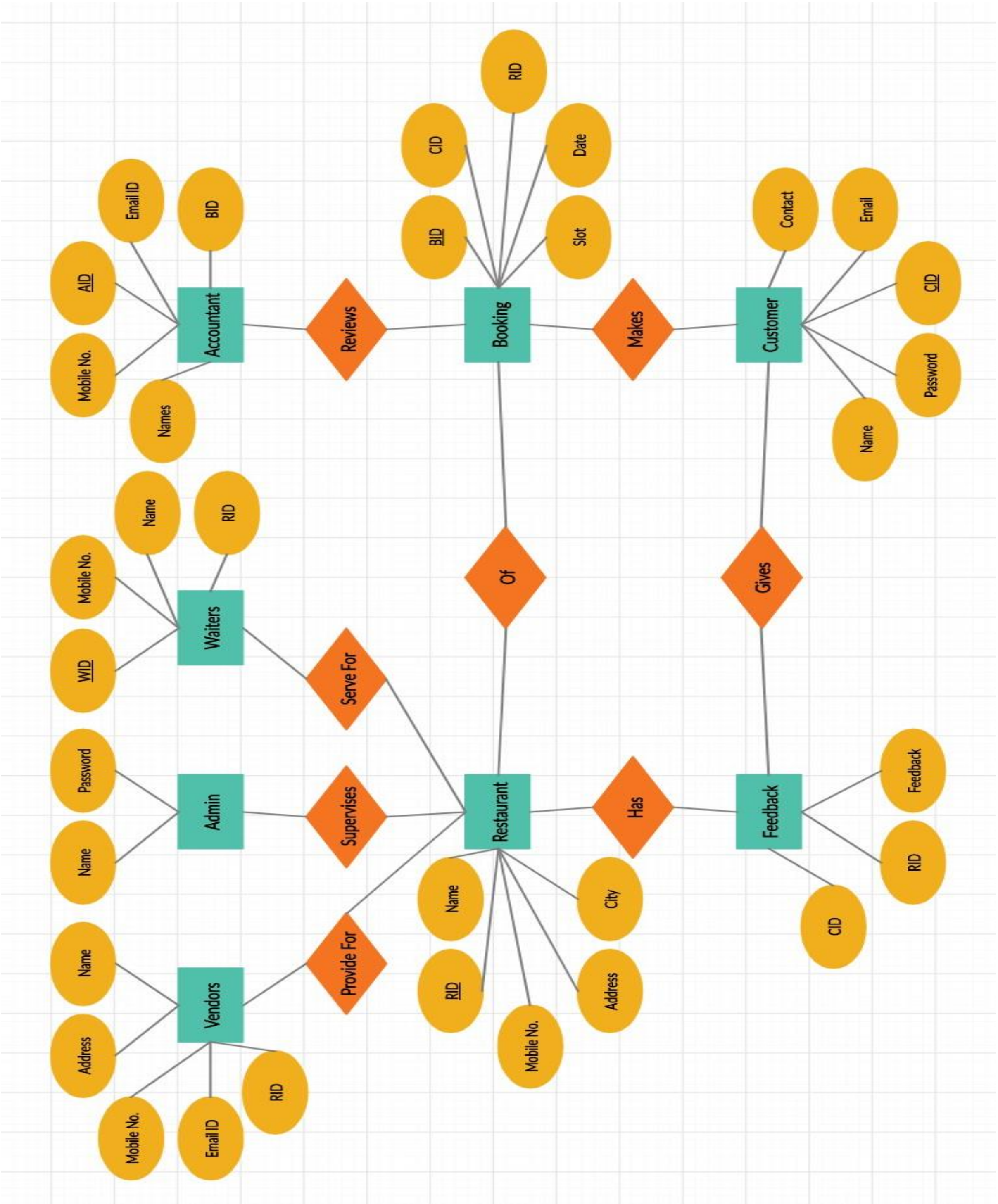
Functional Constraints:

1. **Modification of Existing data**: An Admin can add a restaurant to the database. New Customers can sign up and add up to the table. Each Customer can change their feedbacks relating to a restaurant. Customers can add new Bookings. Restaurant's details such as Contact Number can be updated. For each Booking, the slot and Date of reservation can be updated. Customers can change or modify their personal details.

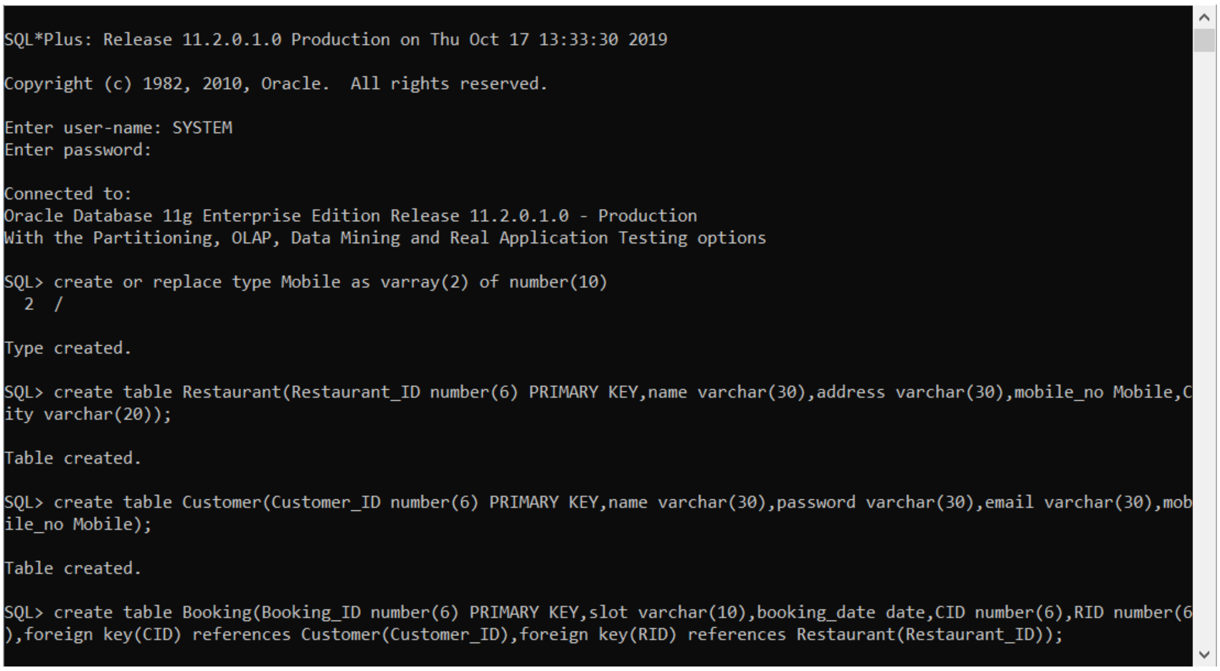
2. **Deletion of Existing Data**: An Admin can remove a restaurant. Customers can delete their accounts from the database. Customers can also delete the feedbacks that they have given for a restaurant.

3. **Retrieval of Existing Data**: An Admin can view the available restaurants. We can view the Customers that have already signed up. We can view the bookings which

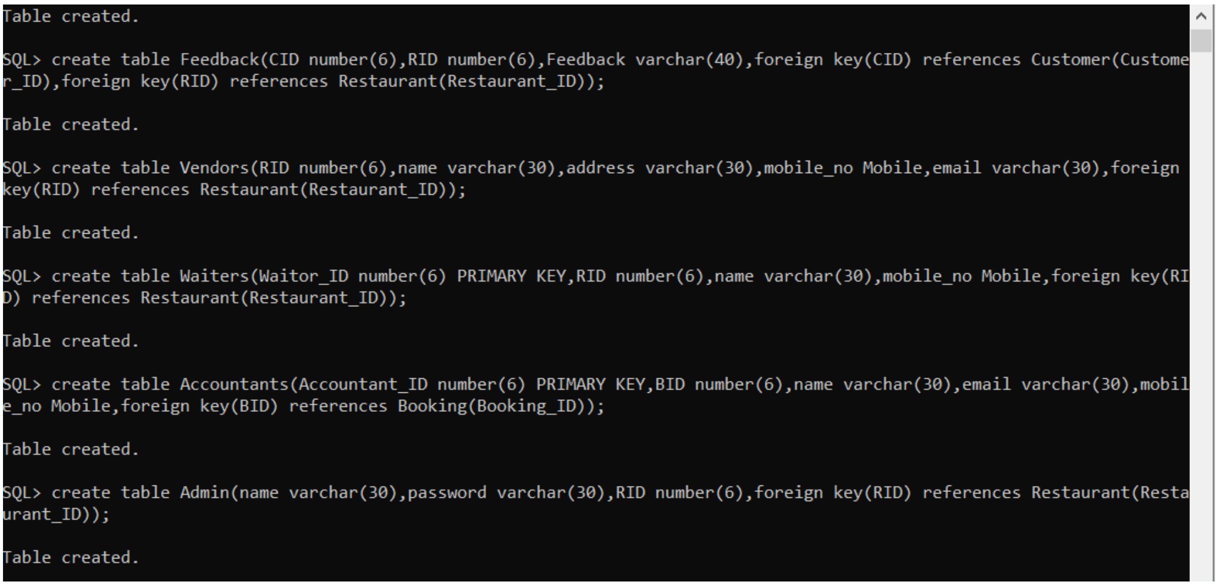
customers have made. We can view the feedbacks which the customers have given for a restaurant.



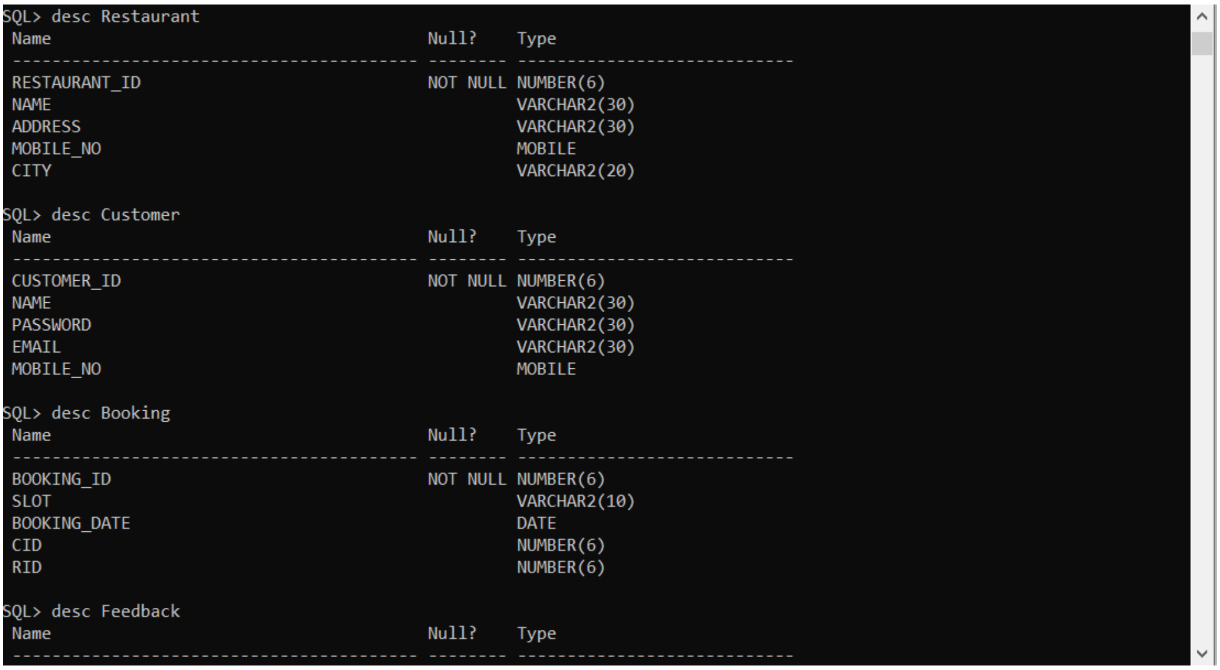
Statements:



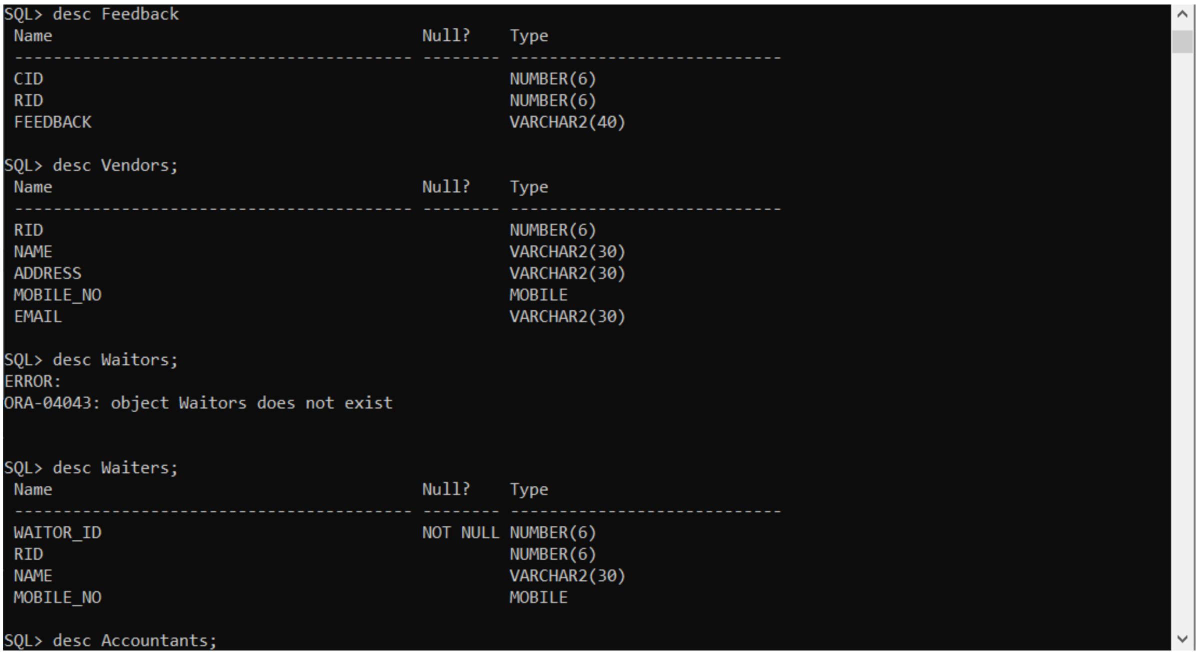
```
SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 17 13:33:30 2019

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Enter user-name: SYSTEM
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> create or replace type Mobile as varray(2) of number(10)
  2  /

Type created.

SQL> create table Restaurant(Restaurant_ID number(6) PRIMARY KEY,name varchar(30),address varchar(30),mobile_no Mobile,C
ity varchar(20));

Table created.

SQL> create table Customer(Customer_ID number(6) PRIMARY KEY,name varchar(30),password varchar(30),email varchar(30),mob
ile_no Mobile);

Table created.

SQL> create table Booking(Booking_ID number(6) PRIMARY KEY,slot varchar(10),booking_date date,CID number(6),RID number(6
),foreign key(CID) references Customer(Customer_ID),foreign key(RID) references Restaurant(Restaurant_ID));
```



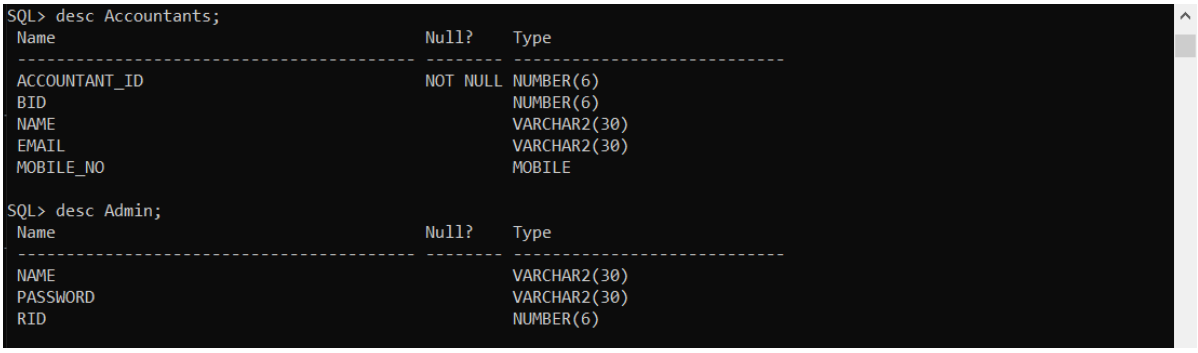
```
Table created.

SQL> create table Feedback(CID number(6),RID number(6),Feedback varchar(40),foreign key(CID) references Customer(Custome
r_ID),foreign key(RID) references Restaurant(Restaurant_ID));

Table created.

SQL> create table Vendors(RID number(6),name varchar(30),address varchar(30),mobile_no Mobile,email varchar(30),foreign
key(RID) references Restaurant(Restaurant_ID));

Table created.

SQL> create table Waiters(Waitor_ID number(6) PRIMARY KEY,RID number(6),name varchar(30),mobile_no Mobile,foreign key(RI
D) references Restaurant(Restaurant_ID));

Table created.

SQL> create table Accountants(Accountant_ID number(6) PRIMARY KEY,BID number(6),name varchar(30),email varchar(30),mobil
e_no Mobile,foreign key(BID) references Booking(Booking_ID));

Table created.

SQL> create table Admin(name varchar(30),password varchar(30),RID number(6),foreign key(RID) references Restaurant(Resta
urant_ID));

Table created.
```



```
SQL> desc Restaurant
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 RESTAURANT_ID                             NOT NULL NUMBER(6)
 NAME                                               VARCHAR2(30)
 ADDRESS                                            VARCHAR2(30)
 MOBILE_NO                                          MOBILE
 CITY                                               VARCHAR2(20)

SQL> desc Customer
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL NUMBER(6)
 NAME                                               VARCHAR2(30)
 PASSWORD                                           VARCHAR2(30)
 EMAIL                                              VARCHAR2(30)
 MOBILE_NO                                          MOBILE

SQL> desc Booking
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOKING_ID                                NOT NULL NUMBER(6)
 SLOT                                               VARCHAR2(10)
 BOOKING_DATE                                       DATE
 CID                                                NUMBER(6)
 RID                                                NUMBER(6)

SQL> desc Feedback
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
```



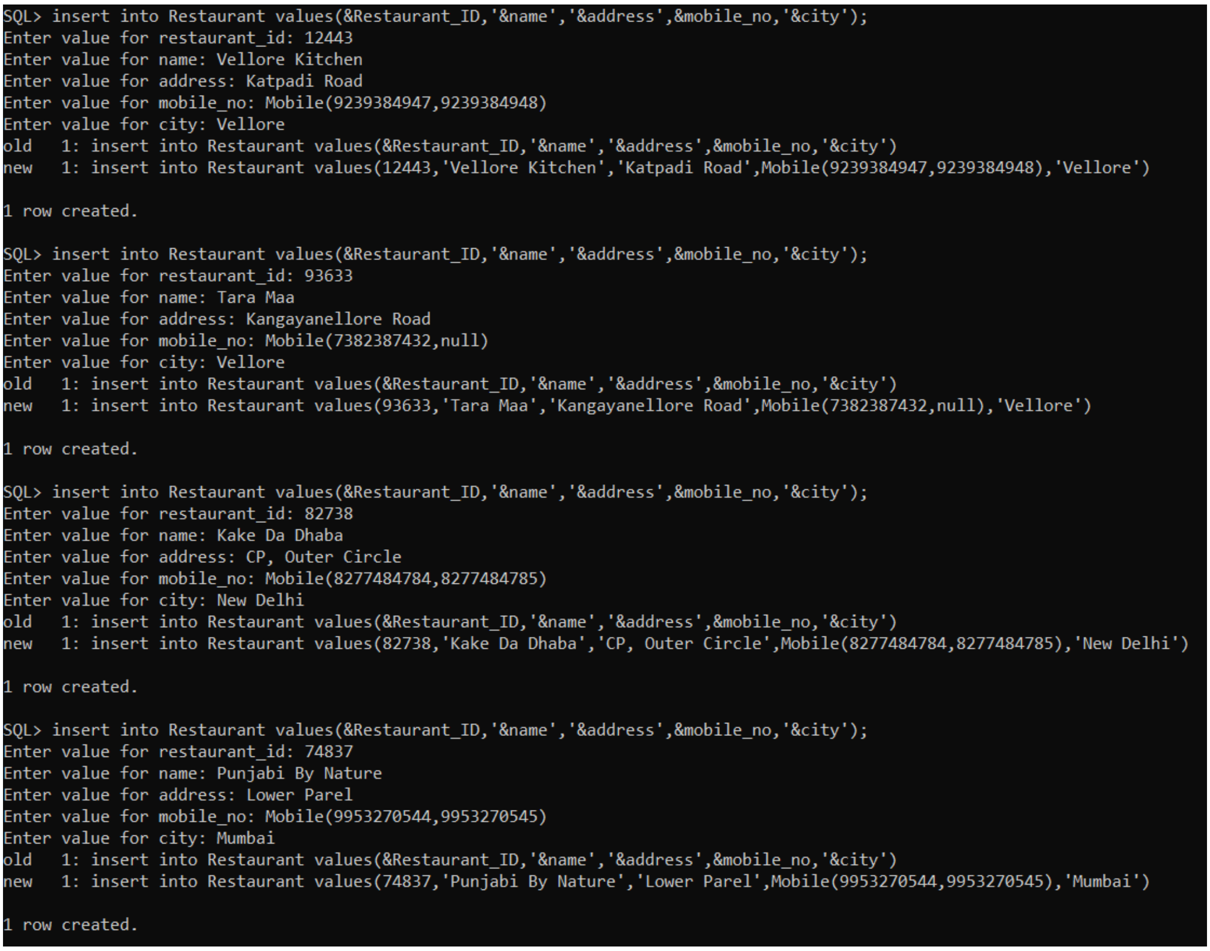
```
SQL> desc Feedback
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CID                                                NUMBER(6)
 RID                                                NUMBER(6)
 FEEDBACK                                           VARCHAR2(40)

SQL> desc Vendors;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 RID                                                NUMBER(6)
 NAME                                               VARCHAR2(30)
 ADDRESS                                            VARCHAR2(30)
 MOBILE_NO                                          MOBILE
 EMAIL                                              VARCHAR2(30)

SQL> desc Waitors;
ERROR:
ORA-04043: object Waitors does not exist


SQL> desc Waiters;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 WAITOR_ID                                 NOT NULL NUMBER(6)
 RID                                                NUMBER(6)
 NAME                                               VARCHAR2(30)
 MOBILE_NO                                          MOBILE

SQL> desc Accountants;
```



```
SQL> desc Accountants;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACCOUNTANT_ID                             NOT NULL NUMBER(6)
 BID                                                NUMBER(6)
 NAME                                               VARCHAR2(30)
 EMAIL                                              VARCHAR2(30)
 MOBILE_NO                                          MOBILE

SQL> desc Admin;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                               VARCHAR2(30)
 PASSWORD                                           VARCHAR2(30)
 RID                                                NUMBER(6)
```

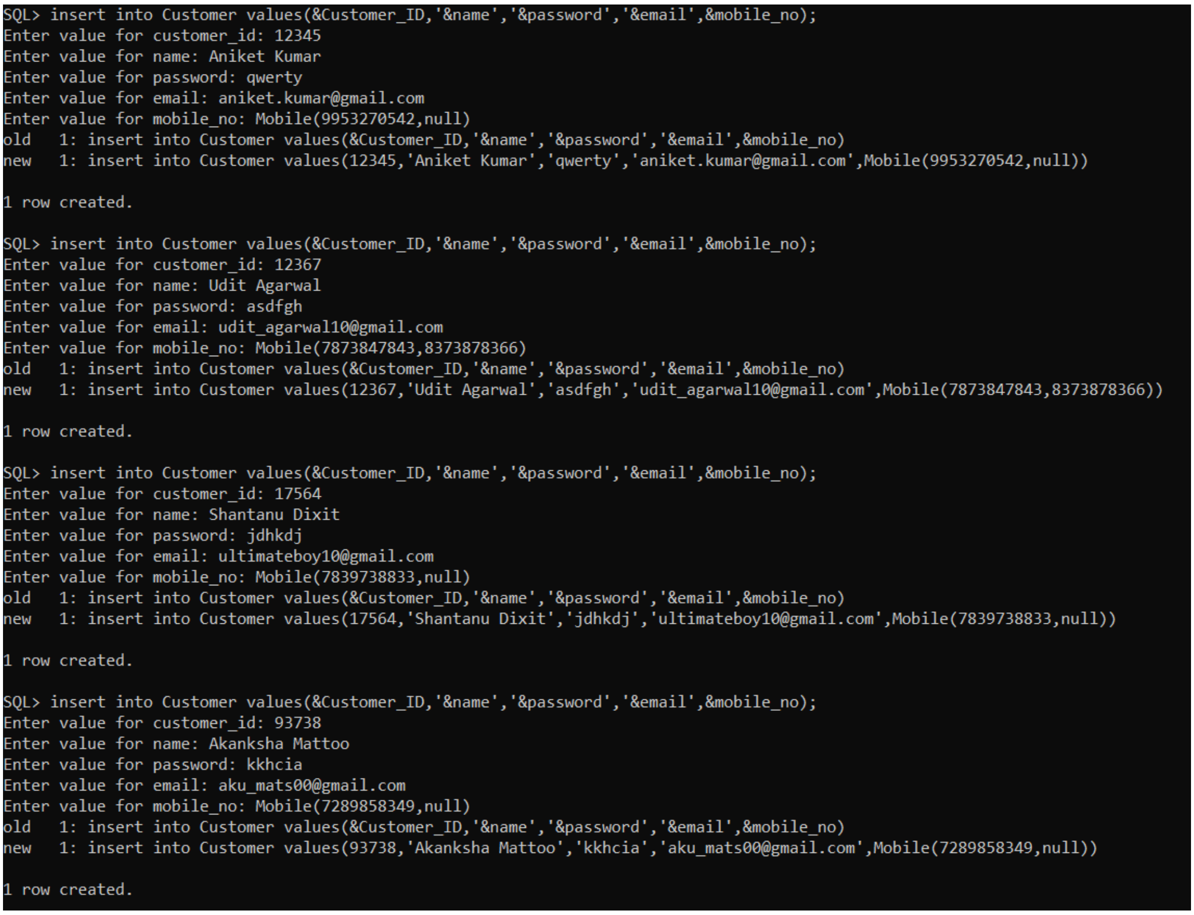
Sample Data:

1. Restaurant:



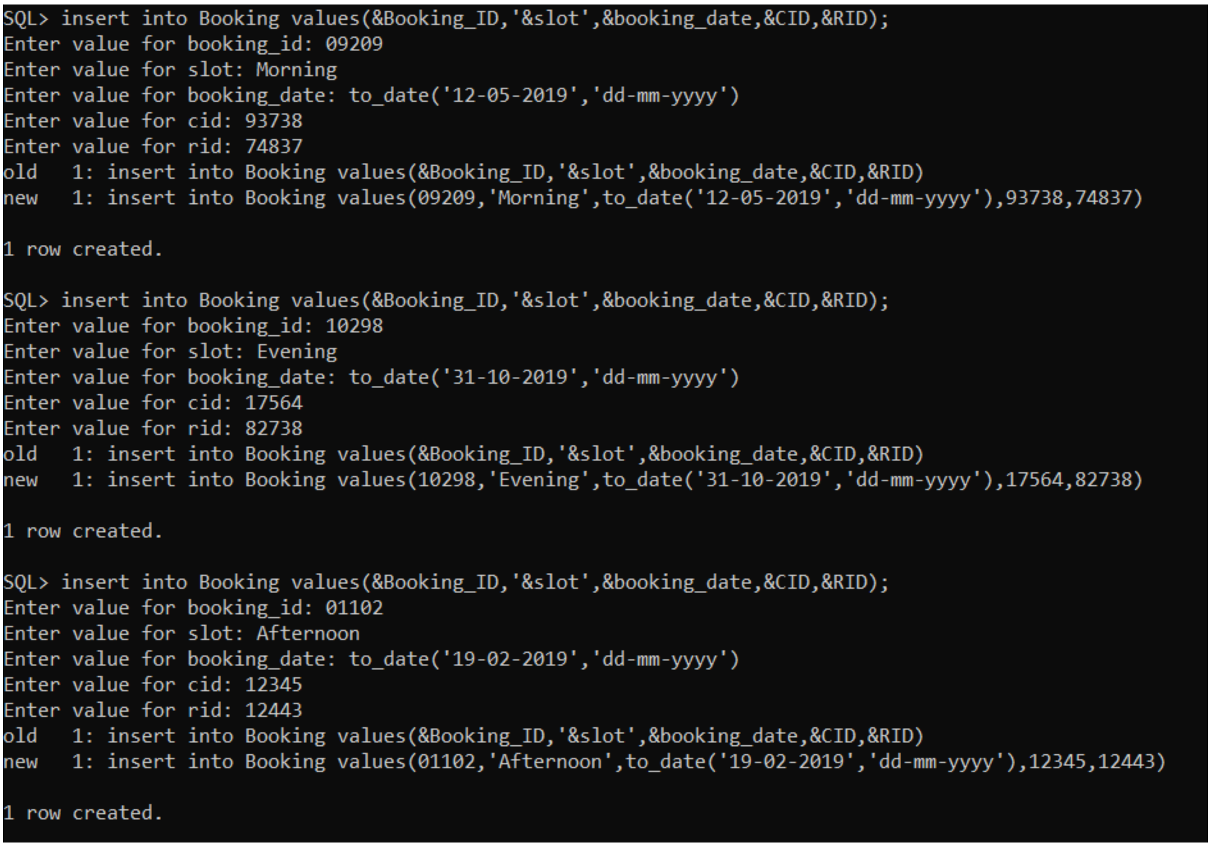
```
SQL> insert into Restaurant values(&Restaurant_ID,'&name','&address',&mobile_no,'&city');
Enter value for restaurant_id: 12443
Enter value for name: Vellore Kitchen
Enter value for address: Katpadi Road
Enter value for mobile_no: Mobile(9239384947,9239384948)
Enter value for city: Vellore
old   1: insert into Restaurant values(&Restaurant_ID,'&name','&address',&mobile_no,'&city')
new   1: insert into Restaurant values(12443,'Vellore Kitchen','Katpadi Road',Mobile(9239384947,9239384948),'Vellore')

1 row created.

SQL> insert into Restaurant values(&Restaurant_ID,'&name','&address',&mobile_no,'&city');
Enter value for restaurant_id: 93633
Enter value for name: Tara Maa
Enter value for address: Kangayanellore Road
Enter value for mobile_no: Mobile(7382387432,null)
Enter value for city: Vellore
old   1: insert into Restaurant values(&Restaurant_ID,'&name','&address',&mobile_no,'&city')
new   1: insert into Restaurant values(93633,'Tara Maa','Kangayanellore Road',Mobile(7382387432,null),'Vellore')

1 row created.

SQL> insert into Restaurant values(&Restaurant_ID,'&name','&address',&mobile_no,'&city');
Enter value for restaurant_id: 82738
Enter value for name: Kake Da Dhaba
Enter value for address: CP, Outer Circle
Enter value for mobile_no: Mobile(8277484784,8277484785)
Enter value for city: New Delhi
old   1: insert into Restaurant values(&Restaurant_ID,'&name','&address',&mobile_no,'&city')
new   1: insert into Restaurant values(82738,'Kake Da Dhaba','CP, Outer Circle',Mobile(8277484784,8277484785),'New Delhi')

1 row created.

SQL> insert into Restaurant values(&Restaurant_ID,'&name','&address',&mobile_no,'&city');
Enter value for restaurant_id: 74837
Enter value for name: Punjabi By Nature
Enter value for address: Lower Parel
Enter value for mobile_no: Mobile(9953270544,9953270545)
Enter value for city: Mumbai
old   1: insert into Restaurant values(&Restaurant_ID,'&name','&address',&mobile_no,'&city')
new   1: insert into Restaurant values(74837,'Punjabi By Nature','Lower Parel',Mobile(9953270544,9953270545),'Mumbai')

1 row created.
```

2. Customer:



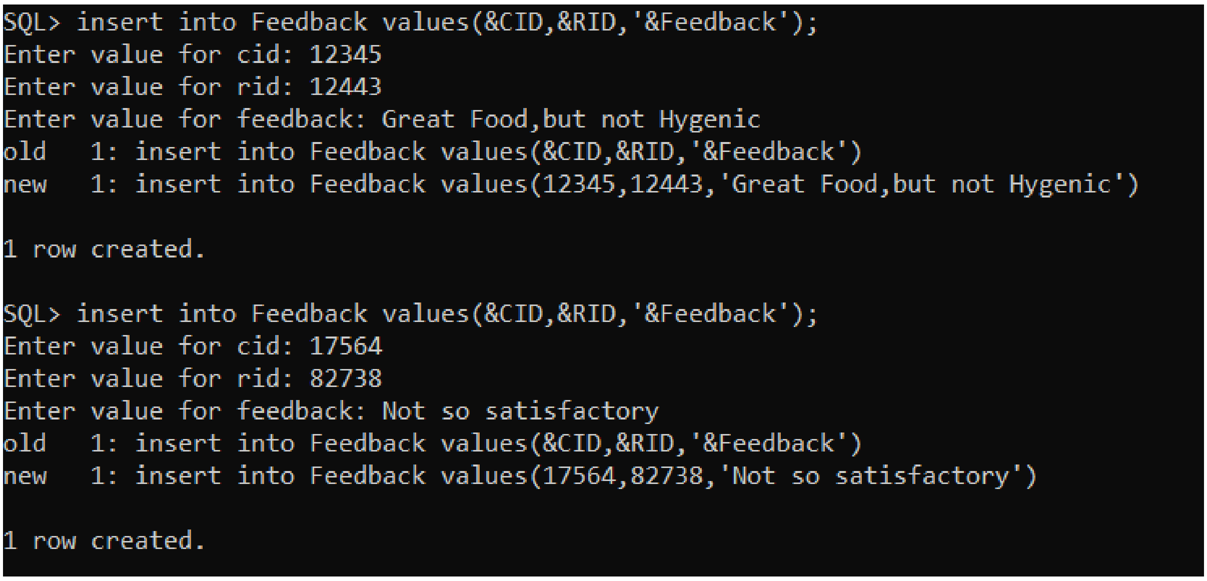
```
SQL> insert into Customer values(&Customer_ID,'&name','&password','&email',&mobile_no);
Enter value for customer_id: 12345
Enter value for name: Aniket Kumar
Enter value for password: qwerty
Enter value for email: aniket.kumar@gmail.com
Enter value for mobile_no: Mobile(9953270542,null)
old   1: insert into Customer values(&Customer_ID,'&name','&password','&email',&mobile_no)
new   1: insert into Customer values(12345,'Aniket Kumar','qwerty','aniket.kumar@gmail.com',Mobile(9953270542,null))

1 row created.

SQL> insert into Customer values(&Customer_ID,'&name','&password','&email',&mobile_no);
Enter value for customer_id: 12367
Enter value for name: Udit Agarwal
Enter value for password: asdfgh
Enter value for email: udit_agarwal10@gmail.com
Enter value for mobile_no: Mobile(7873847843,8373878366)
old   1: insert into Customer values(&Customer_ID,'&name','&password','&email',&mobile_no)
new   1: insert into Customer values(12367,'Udit Agarwal','asdfgh','udit_agarwal10@gmail.com',Mobile(7873847843,8373878366))

1 row created.

SQL> insert into Customer values(&Customer_ID,'&name','&password','&email',&mobile_no);
Enter value for customer_id: 17564
Enter value for name: Shantanu Dixit
Enter value for password: jdhkdj
Enter value for email: ultimateboy10@gmail.com
Enter value for mobile_no: Mobile(7839738833,null)
old   1: insert into Customer values(&Customer_ID,'&name','&password','&email',&mobile_no)
new   1: insert into Customer values(17564,'Shantanu Dixit','jdhkdj','ultimateboy10@gmail.com',Mobile(7839738833,null))

1 row created.

SQL> insert into Customer values(&Customer_ID,'&name','&password','&email',&mobile_no);
Enter value for customer_id: 93738
Enter value for name: Akanksha Mattoo
Enter value for password: kkhcia
Enter value for email: aku_mats00@gmail.com
Enter value for mobile_no: Mobile(7289858349,null)
old   1: insert into Customer values(&Customer_ID,'&name','&password','&email',&mobile_no)
new   1: insert into Customer values(93738,'Akanksha Mattoo','kkhcia','aku_mats00@gmail.com',Mobile(7289858349,null))

1 row created.
```

3. Booking:



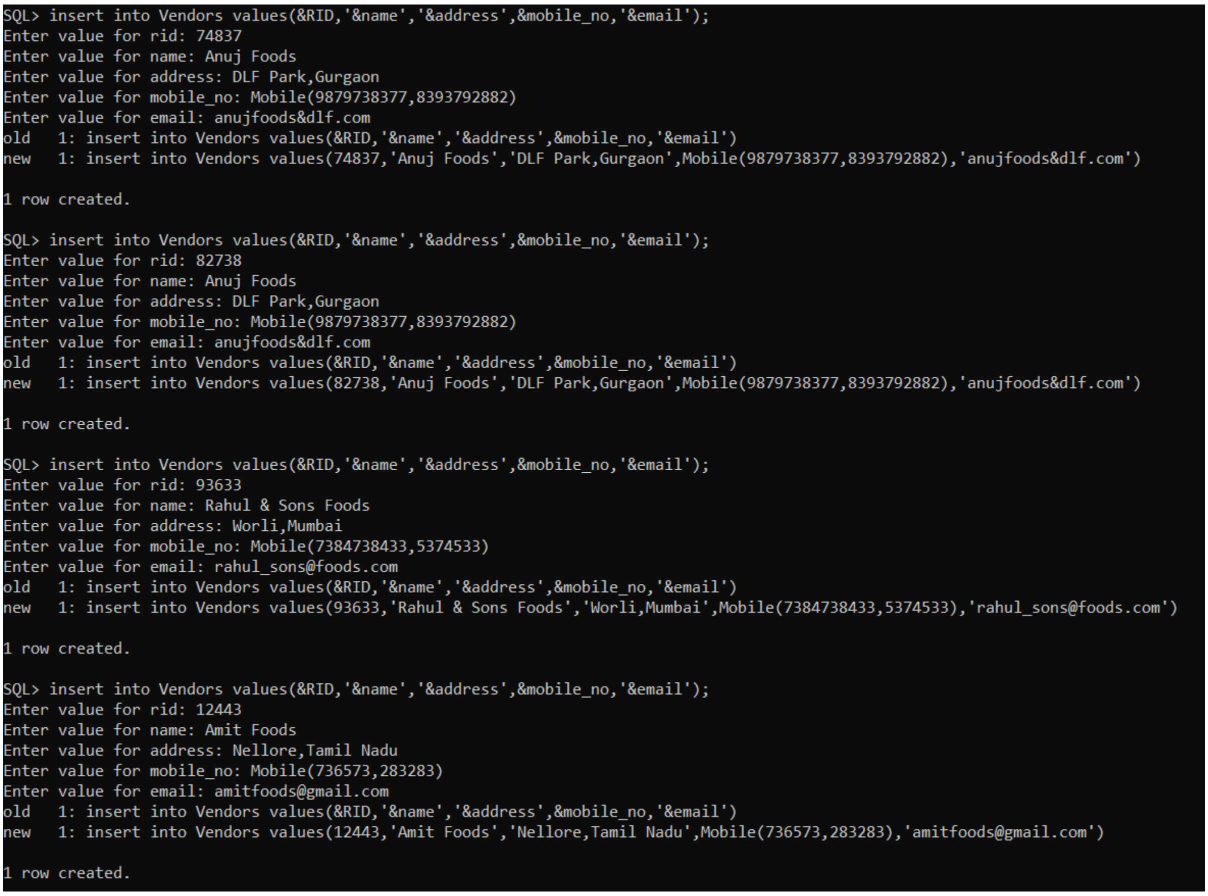
```
SQL> insert into Booking values(&Booking_ID,'&slot',&booking_date,&CID,&RID);
Enter value for booking_id: 09209
Enter value for slot: Morning
Enter value for booking_date: to_date('12-05-2019','dd-mm-yyyy')
Enter value for cid: 93738
Enter value for rid: 74837
old   1: insert into Booking values(&Booking_ID,'&slot',&booking_date,&CID,&RID)
new   1: insert into Booking values(09209,'Morning',to_date('12-05-2019','dd-mm-yyyy'),93738,74837)

1 row created.

SQL> insert into Booking values(&Booking_ID,'&slot',&booking_date,&CID,&RID);
Enter value for booking_id: 10298
Enter value for slot: Evening
Enter value for booking_date: to_date('31-10-2019','dd-mm-yyyy')
Enter value for cid: 17564
Enter value for rid: 82738
old   1: insert into Booking values(&Booking_ID,'&slot',&booking_date,&CID,&RID)
new   1: insert into Booking values(10298,'Evening',to_date('31-10-2019','dd-mm-yyyy'),17564,82738)

1 row created.

SQL> insert into Booking values(&Booking_ID,'&slot',&booking_date,&CID,&RID);
Enter value for booking_id: 01102
Enter value for slot: Afternoon
Enter value for booking_date: to_date('19-02-2019','dd-mm-yyyy')
Enter value for cid: 12345
Enter value for rid: 12443
old   1: insert into Booking values(&Booking_ID,'&slot',&booking_date,&CID,&RID)
new   1: insert into Booking values(01102,'Afternoon',to_date('19-02-2019','dd-mm-yyyy'),12345,12443)

1 row created.
```

4. Feedback:



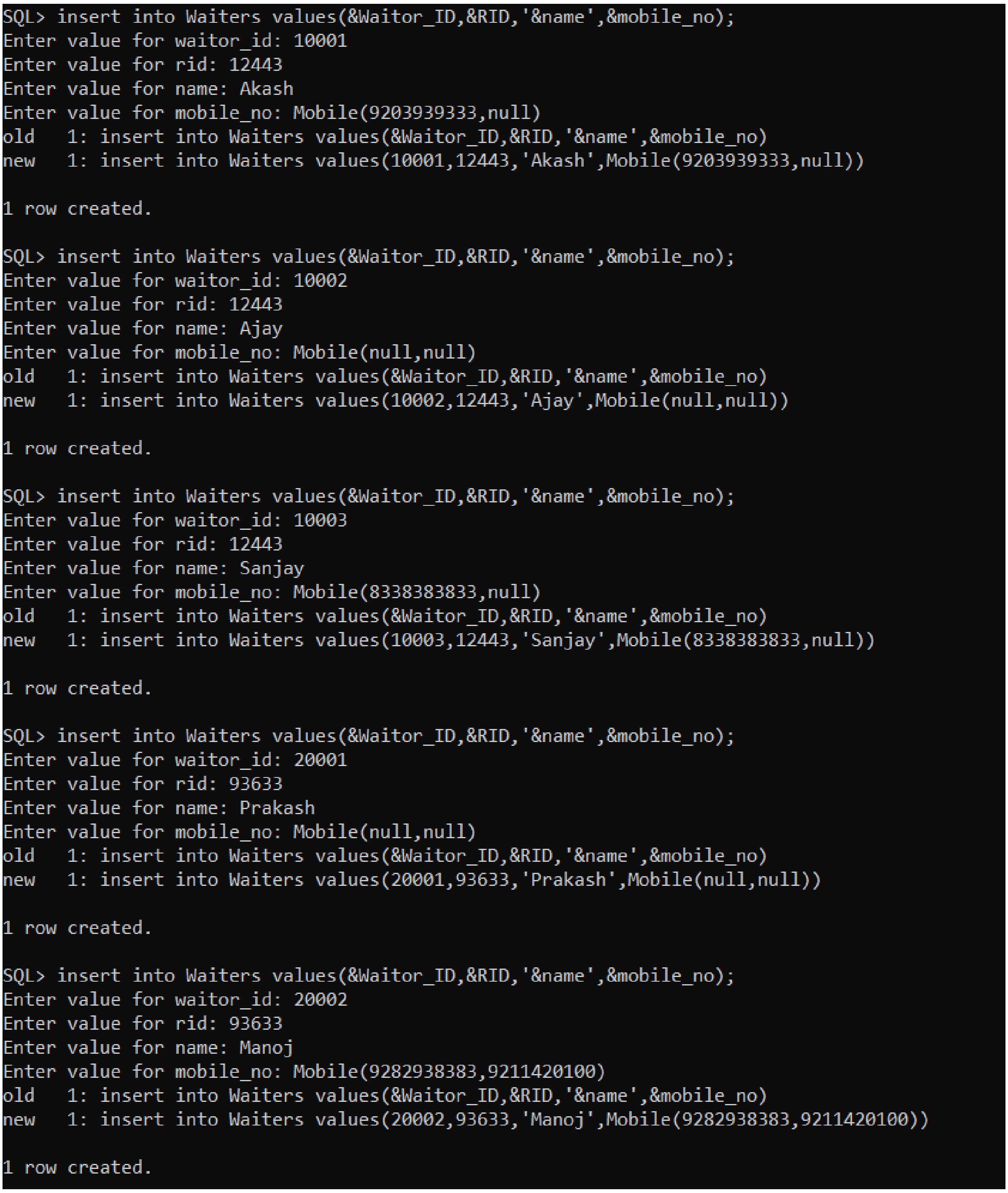
```
SQL> insert into Feedback values(&CID,&RID,'&Feedback');
Enter value for cid: 12345
Enter value for rid: 12443
Enter value for feedback: Great Food,but not Hygenic
old   1: insert into Feedback values(&CID,&RID,'&Feedback')
new   1: insert into Feedback values(12345,12443,'Great Food,but not Hygenic')

1 row created.

SQL> insert into Feedback values(&CID,&RID,'&Feedback');
Enter value for cid: 17564
Enter value for rid: 82738
Enter value for feedback: Not so satisfactory
old   1: insert into Feedback values(&CID,&RID,'&Feedback')
new   1: insert into Feedback values(17564,82738,'Not so satisfactory')

1 row created.
```

5. Vendors:

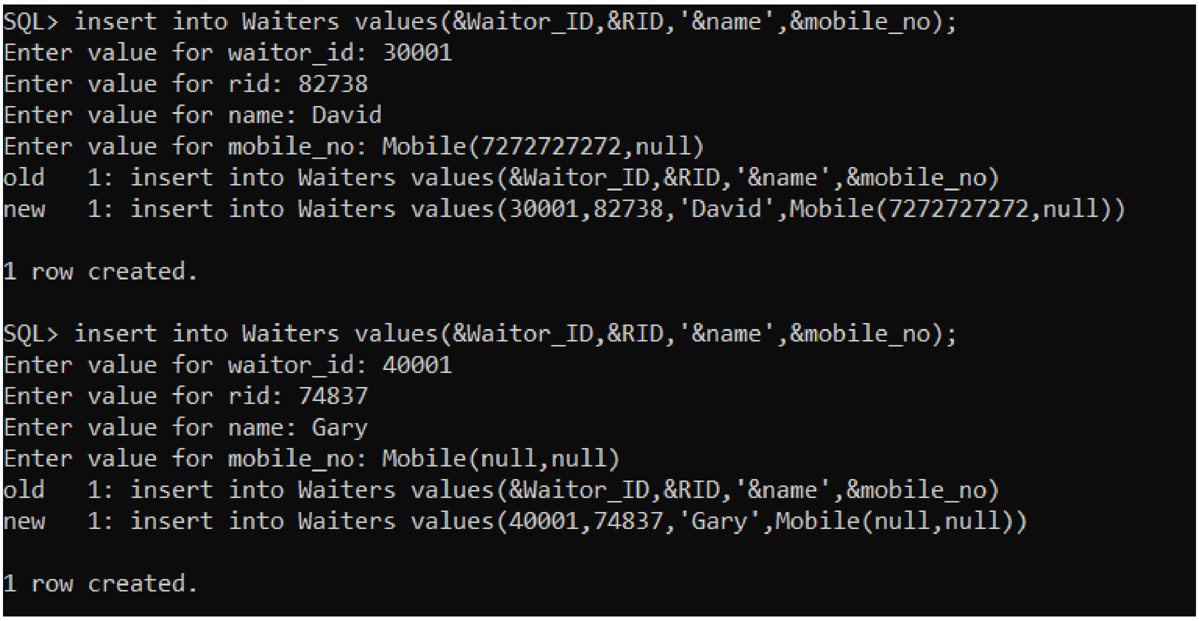


```
SQL> insert into Vendors values(&RID,'&name','&address',&mobile_no,'&email');
Enter value for rid: 74837
Enter value for name: Anuj Foods
Enter value for address: DLF Park,Gurgaon
Enter value for mobile_no: Mobile(9879738377,8393792882)
Enter value for email: anujfoods&dlf.com
old   1: insert into Vendors values(&RID,'&name','&address',&mobile_no,'&email')
new   1: insert into Vendors values(74837,'Anuj Foods','DLF Park,Gurgaon',Mobile(9879738377,8393792882),'anujfoods&dlf.com')

1 row created.

SQL> insert into Vendors values(&RID,'&name','&address',&mobile_no,'&email');
Enter value for rid: 82738
Enter value for name: Anuj Foods
Enter value for address: DLF Park,Gurgaon
Enter value for mobile_no: Mobile(9879738377,8393792882)
Enter value for email: anujfoods&dlf.com
old   1: insert into Vendors values(&RID,'&name','&address',&mobile_no,'&email')
new   1: insert into Vendors values(82738,'Anuj Foods','DLF Park,Gurgaon',Mobile(9879738377,8393792882),'anujfoods&dlf.com')

1 row created.

SQL> insert into Vendors values(&RID,'&name','&address',&mobile_no,'&email');
Enter value for rid: 93633
Enter value for name: Rahul & Sons Foods
Enter value for address: Worli,Mumbai
Enter value for mobile_no: Mobile(7384738433,5374533)
Enter value for email: rahul_sons@foods.com
old   1: insert into Vendors values(&RID,'&name','&address',&mobile_no,'&email')
new   1: insert into Vendors values(93633,'Rahul & Sons Foods','Worli,Mumbai',Mobile(7384738433,5374533),'rahul_sons@foods.com')

1 row created.

SQL> insert into Vendors values(&RID,'&name','&address',&mobile_no,'&email');
Enter value for rid: 12443
Enter value for name: Amit Foods
Enter value for address: Nellore,Tamil Nadu
Enter value for mobile_no: Mobile(736573,283283)
Enter value for email: amitfoods@gmail.com
old   1: insert into Vendors values(&RID,'&name','&address',&mobile_no,'&email')
new   1: insert into Vendors values(12443,'Amit Foods','Nellore,Tamil Nadu',Mobile(736573,283283),'amitfoods@gmail.com')

1 row created.
```

6. Waiters:

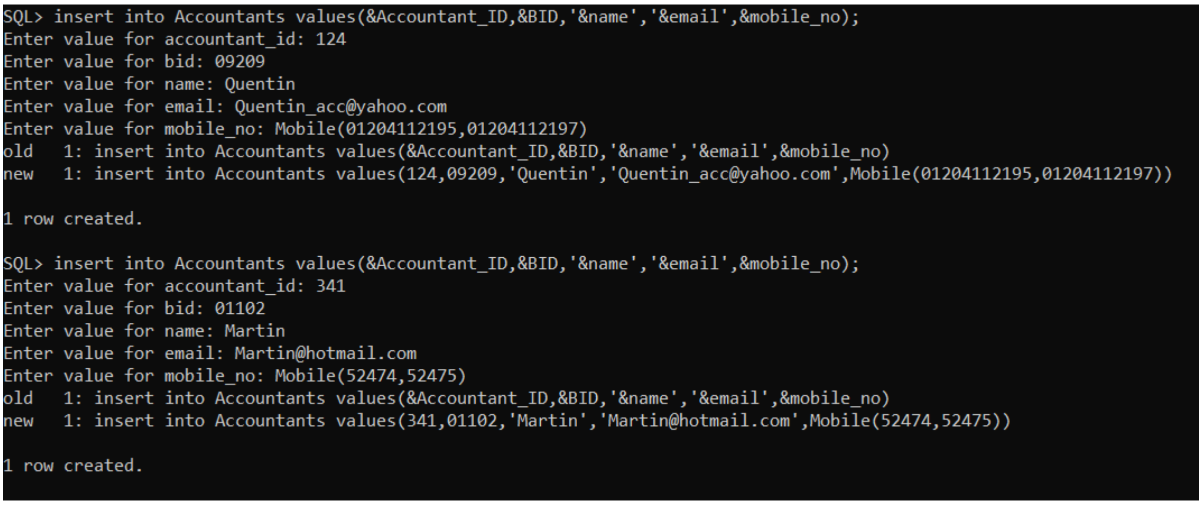


```
SQL> insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no);
Enter value for waitor_id: 10001
Enter value for rid: 12443
Enter value for name: Akash
Enter value for mobile_no: Mobile(9203939333,null)
old   1: insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no)
new   1: insert into Waiters values(10001,12443,'Akash',Mobile(9203939333,null))

1 row created.

SQL> insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no);
Enter value for waitor_id: 10002
Enter value for rid: 12443
Enter value for name: Ajay
Enter value for mobile_no: Mobile(null,null)
old   1: insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no)
new   1: insert into Waiters values(10002,12443,'Ajay',Mobile(null,null))

1 row created.

SQL> insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no);
Enter value for waitor_id: 10003
Enter value for rid: 12443
Enter value for name: Sanjay
Enter value for mobile_no: Mobile(8338383833,null)
old   1: insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no)
new   1: insert into Waiters values(10003,12443,'Sanjay',Mobile(8338383833,null))

1 row created.

SQL> insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no);
Enter value for waitor_id: 20001
Enter value for rid: 93633
Enter value for name: Prakash
Enter value for mobile_no: Mobile(null,null)
old   1: insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no)
new   1: insert into Waiters values(20001,93633,'Prakash',Mobile(null,null))

1 row created.

SQL> insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no);
Enter value for waitor_id: 20002
Enter value for rid: 93633
Enter value for name: Manoj
Enter value for mobile_no: Mobile(9282938383,9211420100)
old   1: insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no)
new   1: insert into Waiters values(20002,93633,'Manoj',Mobile(9282938383,9211420100))

1 row created.
```

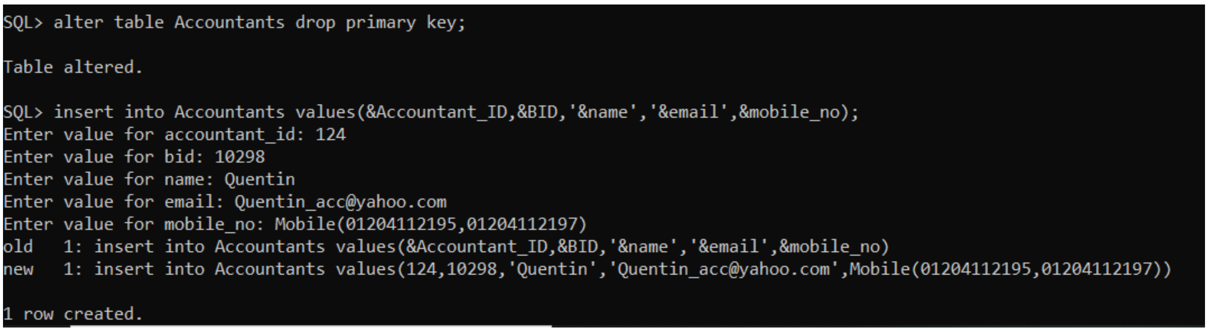


```
SQL> insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no);
Enter value for waitor_id: 30001
Enter value for rid: 82738
Enter value for name: David
Enter value for mobile_no: Mobile(7272727272,null)
old   1: insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no)
new   1: insert into Waiters values(30001,82738,'David',Mobile(7272727272,null))

1 row created.

SQL> insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no);
Enter value for waitor_id: 40001
Enter value for rid: 74837
Enter value for name: Gary
Enter value for mobile_no: Mobile(null,null)
old   1: insert into Waiters values(&Waitor_ID,&RID,'&name',&mobile_no)
new   1: insert into Waiters values(40001,74837,'Gary',Mobile(null,null))

1 row created.
```

7. Accountants:

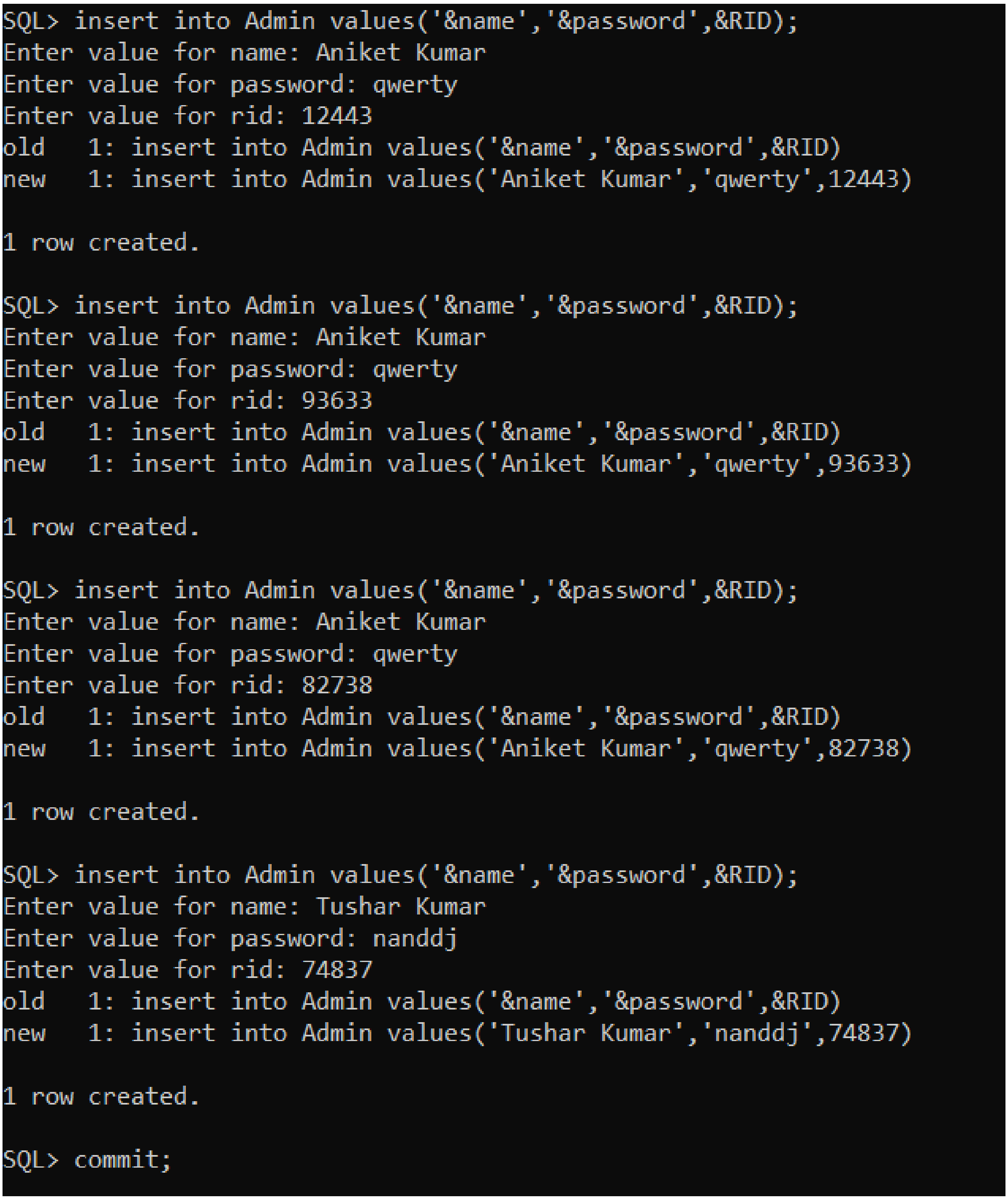


```
SQL> insert into Accountants values(&Accountant_ID,&BID,'&name','&email',&mobile_no);
Enter value for accountant_id: 124
Enter value for bid: 09209
Enter value for name: Quentin
Enter value for email: Quentin_acc@yahoo.com
Enter value for mobile_no: Mobile(01204112195,01204112197)
old   1: insert into Accountants values(&Accountant_ID,&BID,'&name','&email',&mobile_no)
new   1: insert into Accountants values(124,09209,'Quentin','Quentin_acc@yahoo.com',Mobile(01204112195,01204112197))

1 row created.

SQL> insert into Accountants values(&Accountant_ID,&BID,'&name','&email',&mobile_no);
Enter value for accountant_id: 341
Enter value for bid: 01102
Enter value for name: Martin
Enter value for email: Martin@hotmail.com
Enter value for mobile_no: Mobile(52474,52475)
old   1: insert into Accountants values(&Accountant_ID,&BID,'&name','&email',&mobile_no)
new   1: insert into Accountants values(341,01102,'Martin','Martin@hotmail.com',Mobile(52474,52475))

1 row created.
```



```
SQL> alter table Accountants drop primary key;

Table altered.

SQL> insert into Accountants values(&Accountant_ID,&BID,'&name','&email',&mobile_no);
Enter value for accountant_id: 124
Enter value for bid: 10298
Enter value for name: Quentin
Enter value for email: Quentin_acc@yahoo.com
Enter value for mobile_no: Mobile(01204112195,01204112197)
old   1: insert into Accountants values(&Accountant_ID,&BID,'&name','&email',&mobile_no)
new   1: insert into Accountants values(124,10298,'Quentin','Quentin_acc@yahoo.com',Mobile(01204112195,01204112197))

1 row created.
```

8. Admin:



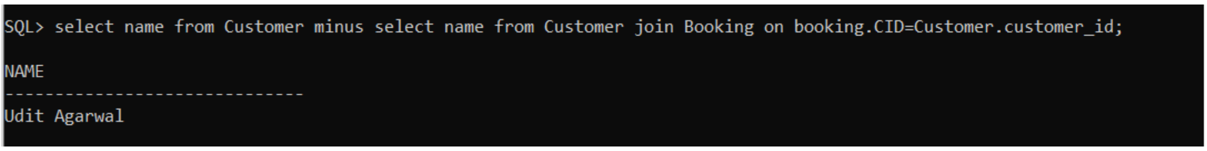
```
SQL> insert into Admin values('&name','&password',&RID);
Enter value for name: Aniket Kumar
Enter value for password: qwerty
Enter value for rid: 12443
old   1: insert into Admin values('&name','&password',&RID)
new   1: insert into Admin values('Aniket Kumar','qwerty',12443)

1 row created.

SQL> insert into Admin values('&name','&password',&RID);
Enter value for name: Aniket Kumar
Enter value for password: qwerty
Enter value for rid: 93633
old   1: insert into Admin values('&name','&password',&RID)
new   1: insert into Admin values('Aniket Kumar','qwerty',93633)

1 row created.

SQL> insert into Admin values('&name','&password',&RID);
Enter value for name: Aniket Kumar
Enter value for password: qwerty
Enter value for rid: 82738
old   1: insert into Admin values('&name','&password',&RID)
new   1: insert into Admin values('Aniket Kumar','qwerty',82738)

1 row created.

SQL> insert into Admin values('&name','&password',&RID);
Enter value for name: Tushar Kumar
Enter value for password: nanddj
Enter value for rid: 74837
old   1: insert into Admin values('&name','&password',&RID)
new   1: insert into Admin values('Tushar Kumar','nanddj',74837)

1 row created.

SQL> commit;
```

Statements related to Restaurant Management Service:

1. Data Retreival:

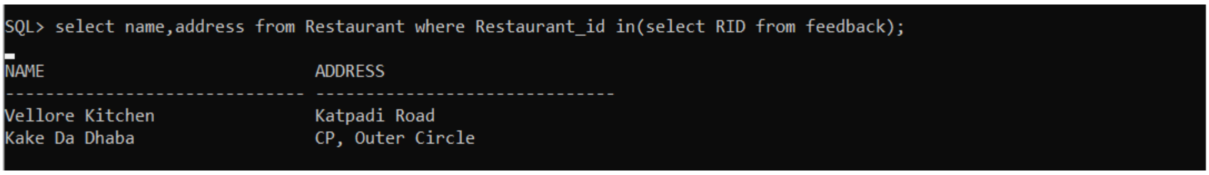
Select Statements:

1.



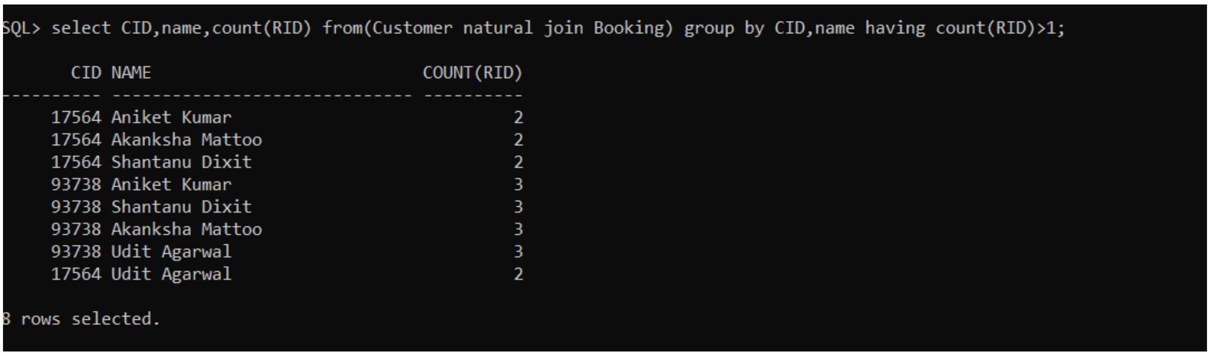
```
SQL> select name from Customer minus select name from Customer join Booking on booking.CID=Customer.customer_id;

NAME
------------------------------
Udit Agarwal
```

2.



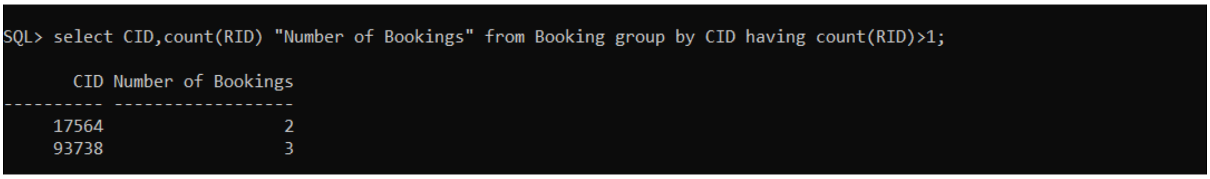
```
SQL> select name,address from Restaurant where Restaurant_id in(select RID from feedback);

NAME                           ADDRESS
------------------------------ ------------------------------
Vellore Kitchen                Katpadi Road
Kake Da Dhaba                  CP, Outer Circle
```

3.



```
SQL> select CID,name,count(RID) from(Customer natural join Booking) group by CID,name having count(RID)>1;

       CID NAME                           COUNT(RID)
---------- ------------------------------ ----------
     17564 Aniket Kumar                            2
     17564 Akanksha Mattoo                         2
     17564 Shantanu Dixit                          2
     93738 Aniket Kumar                            3
     93738 Shantanu Dixit                          3
     93738 Akanksha Mattoo                         3
     93738 Udit Agarwal                            3
     17564 Udit Agarwal                            2

8 rows selected.
```

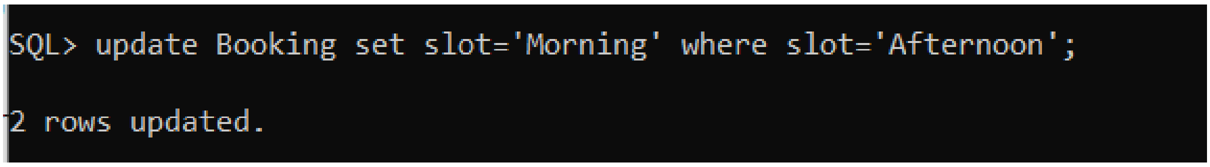
4.



```
SQL> select CID,count(RID) "Number of Bookings" from Booking group by CID having count(RID)>1;

       CID Number of Bookings
---------- ------------------
     17564                  2
     93738                  3
```

2. Data Modification:

Update Statements:

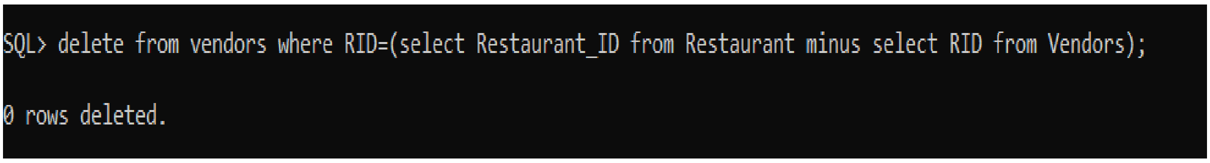


```
SQL> update Booking set slot='Morning' where slot='Afternoon';

2 rows updated.
```

3. Data Removal:

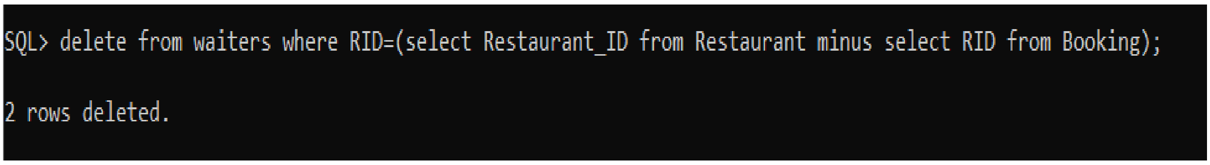
Delete Statements:

1.



```
SQL> delete from vendors where RID=(select Restaurant_ID from Restaurant minus select RID from Vendors);

0 rows deleted.
```

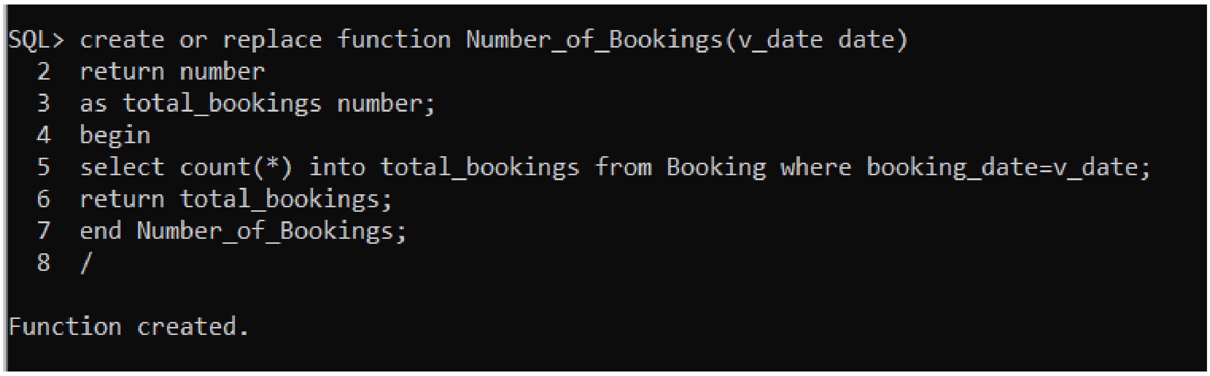
2.



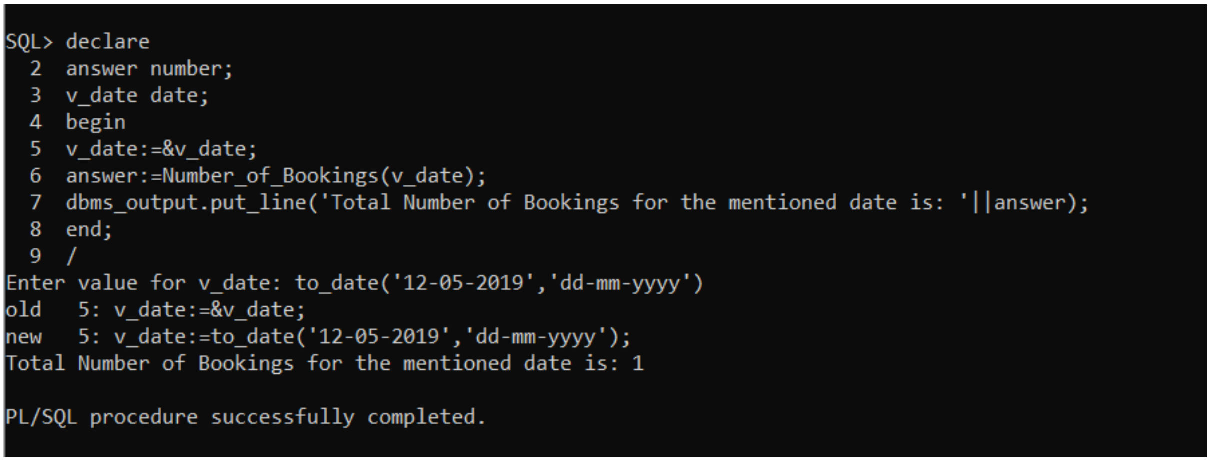
```
SQL> delete from waiters where RID=(select Restaurant_ID from Restaurant minus select RID from Booking);

2 rows deleted.
```

PL/SQL Programming:

1. PL/SQL Function:

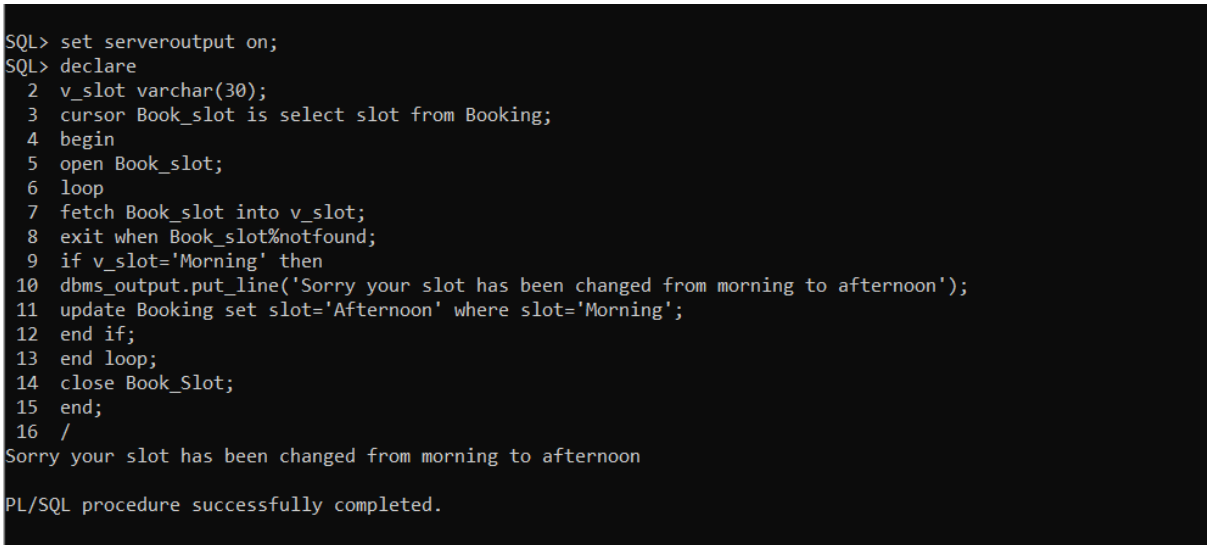


```
SQL> create or replace function Number_of_Bookings(v_date date)
  2  return number
  3  as total_bookings number;
  4  begin
  5  select count(*) into total_bookings from Booking where booking_date=v_date;
  6  return total_bookings;
  7  end Number_of_Bookings;
  8  /

Function created.
```



```
SQL> declare
  2  answer number;
  3  v_date date;
  4  begin
  5  v_date:=&v_date;
  6  answer:=Number_of_Bookings(v_date);
  7  dbms_output.put_line('Total Number of Bookings for the mentioned date is: '||answer);
  8  end;
  9  /
Enter value for v_date: to_date('12-05-2019','dd-mm-yyyy')
old   5: v_date:=&v_date;
new   5: v_date:=to_date('12-05-2019','dd-mm-yyyy');
Total Number of Bookings for the mentioned date is: 1

PL/SQL procedure successfully completed.
```

2. PL/SQL Procedure:

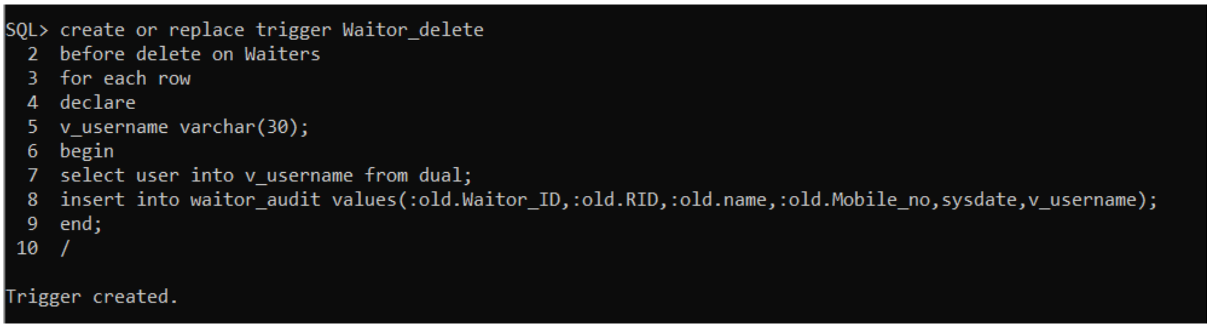


```
SQL> set serveroutput on;
SQL> declare
  2  v_slot varchar(30);
  3  cursor Book_slot is select slot from Booking;
  4  begin
  5  open Book_slot;
  6  loop
  7  fetch Book_slot into v_slot;
  8  exit when Book_slot%notfound;
  9  if v_slot='Morning' then
 10  dbms_output.put_line('Sorry your slot has been changed from morning to afternoon');
 11  update Booking set slot='Afternoon' where slot='Morning';
 12  end if;
 13  end loop;
 14  close Book_Slot;
 15  end;
 16  /
Sorry your slot has been changed from morning to afternoon

PL/SQL procedure successfully completed.
```

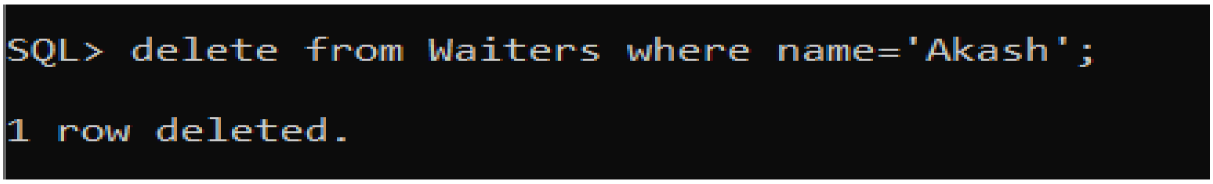
Triggers:

Rule 1:

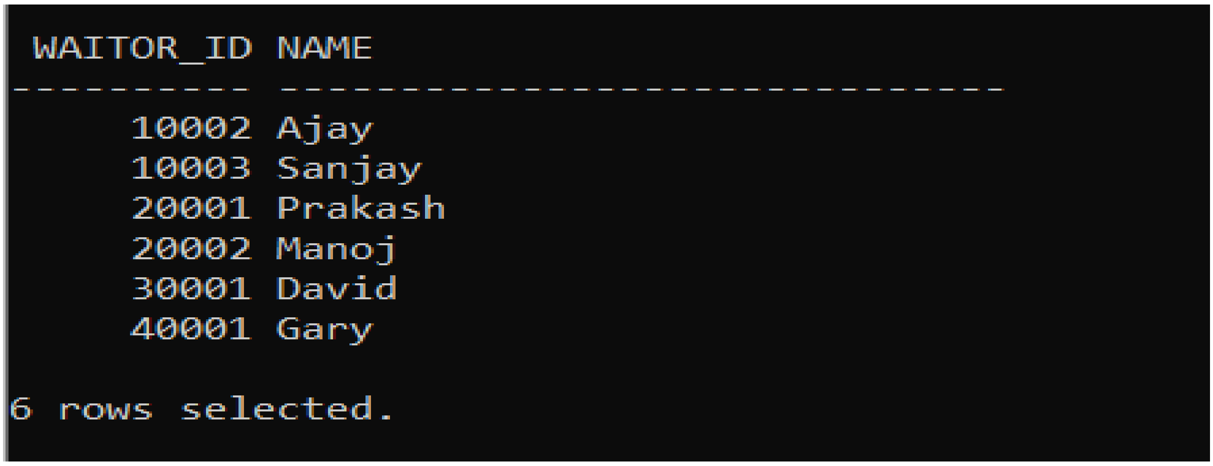
Firing of Waiters from any Restaurant creates a trigger and their names get added to Waiters_Audit Table.



```
SQL> create or replace trigger Waitor_delete
before delete on Waiters
for each row
declare
v_username varchar(30);
begin
select user into v_username from dual;
insert into waitor_audit values(:old.Waitor_ID,:old.RID,:old.name,:old.Mobile_no,sysdate,v_username);
end;
/

Trigger created.
```

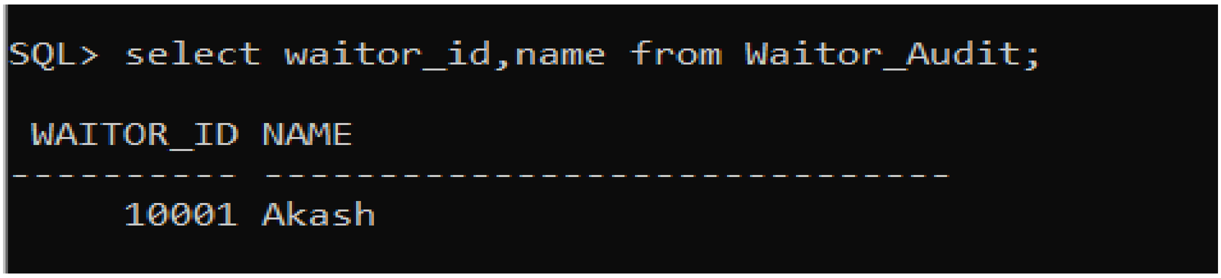


```
SQL> delete from Waiters where name='Akash';

1 row deleted.
```



```
 WAITOR_ID NAME
---------- ------------------------------
     10002 Ajay
     10003 Sanjay
     20001 Prakash
     20002 Manoj
     30001 David
     40001 Gary

6 rows selected.
```

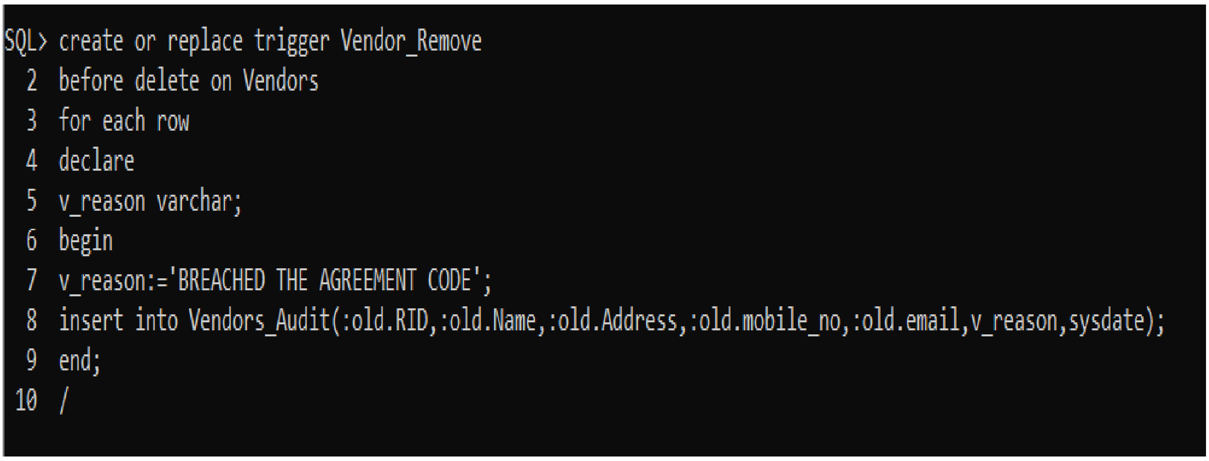
Entry into Waitor_Audit Table:



```
SQL> select waitor_id,name from Waitor_Audit;

 WAITOR_ID NAME
---------- ------------------------------
     10001 Akash
```

Rule 2:

Removal of Vendors from any Restaurant creates a trigger and their names get added to Vendors_Audit Table.



```
SQL> create or replace trigger Vendor_Remove
before delete on Vendors
for each row
declare
v_reason varchar;
begin
v_reason:='BREACHED THE AGREEMENT CODE';
insert into Vendors_Audit(:old.RID,:old.Name,:old.Address,:old.mobile_no,:old.email,v_reason,sysdate);
end;
/
```