A
Project Report
on

Online Food Delivery Management System

Developed by

RAJ SHEKHAR VAGHELA (IT-098) –
Department of IT, DDUniversity

Guided By
Internal Guide:
Prof. Vidhi B Chaudhary
Department of InformationTechnology
Faculty of Technology
DD University

Department of Information Technology Faculty of Technology, Dharmsinh Desai University College Road, Nadiad-387001
October-2020

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "**Online Food Delivery Management System** " is a bonafide report of the work carried out by **Mr. Raj Shekhar Vaghela, Student ID No : 18ITUOS131** of Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during academic year 2020-2021.

Prof. Vidhi B Chaudhary
(Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. Vipul Dabhi
Head , Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

ACKNOWLEDGEMENT

We would like to give our sincere acknowledgement to everybody responsible for the successful completion of our project **Online Food Delivery Management System**.

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of this project.

We owe our deep gratitude to our project guide Prof. Roshni M. Raval, who took been interest on our project work and guided us all along till the completion of our project work by providing all the necessary help for developing a good Database System.

We would also like to thank all our lecturers.

Finally we convoy our acknowledgement to all our friends and family members who directly or indirectly associated with us in the successful completion of the project.We thank one and all.

TABLE OF CONTENTS

1.SYSTEM OVERVIEW

1.1 Introduction

With the growing population along with advancement in technology, There is a common problem which is faced by every state of the country i.e. Unemployment.

In the light of the above situation, the food industry is among one of those industries which offer employment at a very large scale. It is therefore required for the country to have an efficient food ordering system in order to eliminate maximum problems faced by the country. Here we propose an Online Food Ordering system supporting the needs of the current society. The system can be used in any food delivery industry. This simplifies the process of food ordering for both the customer and the restaurant as the entire process is automated and digitalized.

1.2 Literature Review

In the earlier days, setting up a restaurant and managing it had been a very cumbersome task. Some of the problems faced during food deliveries are mentioned below:-

1. To place an order, the customer visits a restaurant, chooses the items to be ordered, make the payment and take the order in real time. This requires manual work and time.
2. When the customer orders in phone, he is unable to see the physical copy of the menu available at the restaurant. This lacks the verification that the order was placed for the correct food items.
3. Every restaurant needs someone or the customer himself to take the order out and make the payment.
4. If the customer is located far away or in a remote area, it is difficult for the customer as well to manually go to the restaurant and place the order.

1.3 Methodology

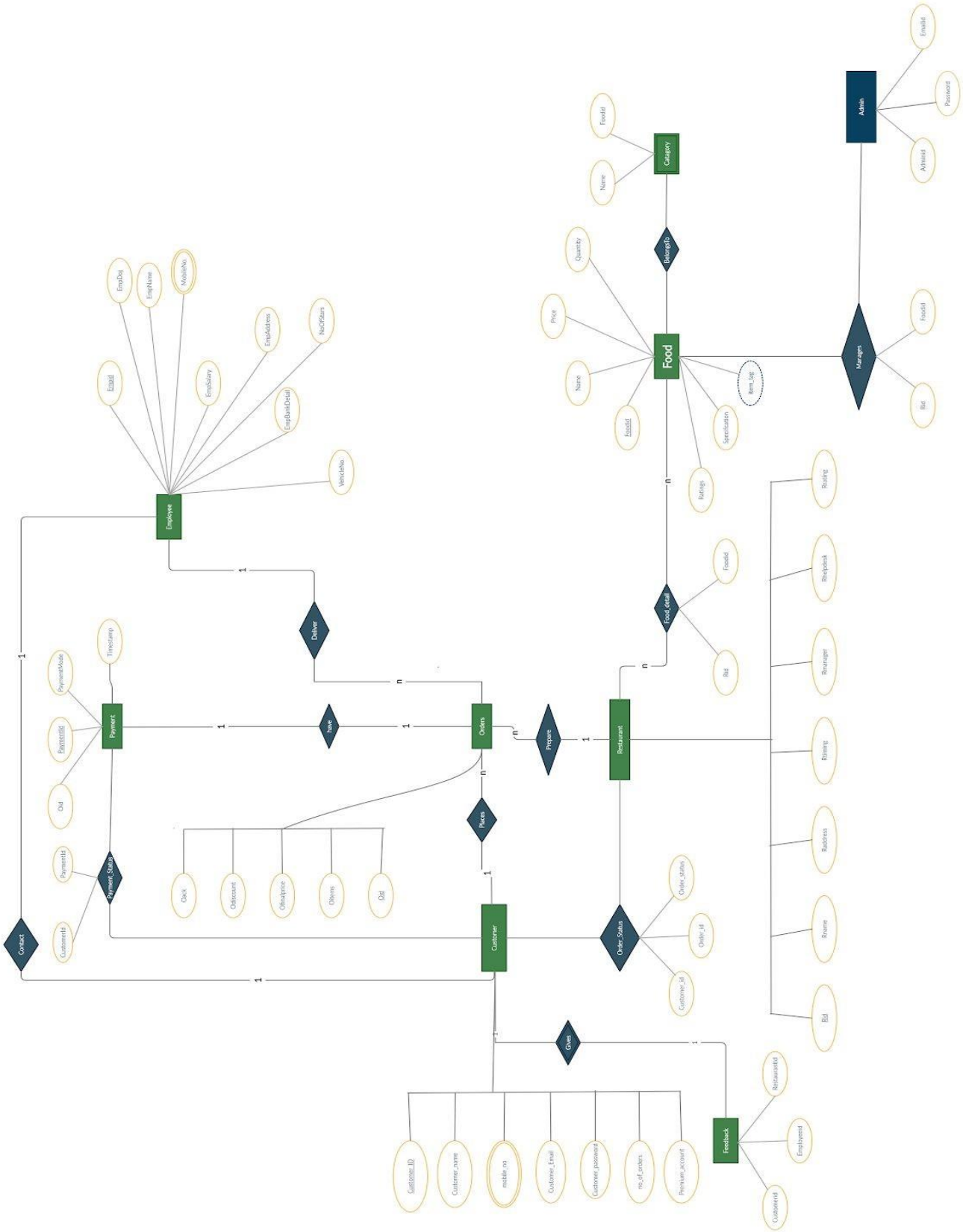
The process starts with the customer entering his/her id and password. The customer searches the food he/she wants and select the required details. Then the customer id redirected to the payment portal where he/she will fill out the required details and make the payment. On the other side, the restaurant will get to know that a person has ordered something. The administrator will give the orders, the food is processed and handed out to the delivery guy. The delivery guy details has been sent to the customer for verification. The delivery guy will reach the destination and give the order and do verification with OTP which is given to the customer.

The administrator is the main user here who gets the option of adding food, deleting food or updating food. Once the selection process is carried out, the details are reflected in the menu of that restaurant.

FOR CLEAR VIEW ER and Schema OPEN BELOW LINK :

https://drive.google.com/file/d/18ixGWu0OyAYMZqg_oclzNVVvNfRyZKSs/view?usp=sharing

2.E-R DIAGRAM


Admin
Emailid
Password
Adminid
Category
Foodid
Name
Belongsto
Quantity
Price
Food
Name
Foodid
Item_tag
Specification
Ratings
Manages
Foodid
Bid
Employee
EmpDoj
EmpName
MobileNo
EmpAddress
NoOfStars
Eaiid
EmpSalary
EmpBankDetail
VehicleNo
Deliver
Food_detail
Foodid
Bid
Payment
Timestamp
PaymentMode
Paymentid
Oid
Payment_Status
Paymentid
Customerid
have
Orders
Oack
Odiscount
Ofinalprice
Oitems
Oid
Places
Prepare
Restaurant
Rating
Rhelpdesk
Rmanager
Rtiming
Raddress
Rname
Bid
Contact
Customer
Order_Status
Order_status
Order_id
Customer_id
Gives
Feedback
Restaurantid
Employeeid
Customerid
Customer_ID
Customer_name
mobile_no
Customer_Email
Customer_password
no_of_orders
Premium_account

3. DATA DICTIONARY

3.1 ADMIN

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	AID	VARCHAR2(4 BYTE)	Yes	(null)	1	(null)
2	APSWD	VARCHAR2(8 BYTE)	Yes	(null)	2	(null)
3	ANAME	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)

3.2 CAT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CATID	VARCHAR2(4 BYTE)	No	(null)	1	(null)
2	CATNAME	VARCHAR2(20 BYTE)	No	(null)	2	(null)

3.3 CITY

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CITYID	VARCHAR2(4 BYTE)	No	(null)	1	(null)
2	CITYNAME	VARCHAR2(255 BYTE)	Yes	(null)	2	(null)
3	ZIPCODE	NUMBER(6,0)	Yes	(null)	3	(null)

3.4 CUSTOMER

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CID	VARCHAR2(4 BYTE)	No	(null)	1	(null)
2	CNAME	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	MOBILENO	NUMBER(10,0)	No	(null)	3	(null)
4	EMAIL	VARCHAR2(20 BYTE)	No	(null)	4	(null)
5	PASSWD	VARCHAR2(255 BYTE)	No	(null)	5	(null)
6	ADDRESS	VARCHAR2(255 BYTE)	No	(null)	6	(null)
7	CITYID	VARCHAR2(4 BYTE)	No	(null)	7	(null)
8	NOOFORDERS	NUMBER(38,0)	Yes	(null)	8	(null)

3.5 EMPLOYEE

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EID	VARCHAR2(4 BYTE)	No	(null)	1	(null)
2	CITYID	VARCHAR2(4 BYTE)	Yes	(null)	2	(null)
3	ENAME	VARCHAR2(20 BYTE)	No	(null)	3	(null)
4	MOBILENO	VARCHAR2(10 BYTE)	No	(null)	4	(null)
5	ADDRESS	VARCHAR2(20 BYTE)	No	(null)	5	(null)
6	VEHICLENO	VARCHAR2(8 BYTE)	No	(null)	6	(null)
7	SALARY	NUMBER	No	(null)	7	(null)
8	DOJ	DATE	No	(null)	8	(null)
9	BANKIFSC	VARCHAR2(8 BYTE)	No	(null)	9	(null)
10	BANKACCNO	NUMBER	No	(null)	10	(null)

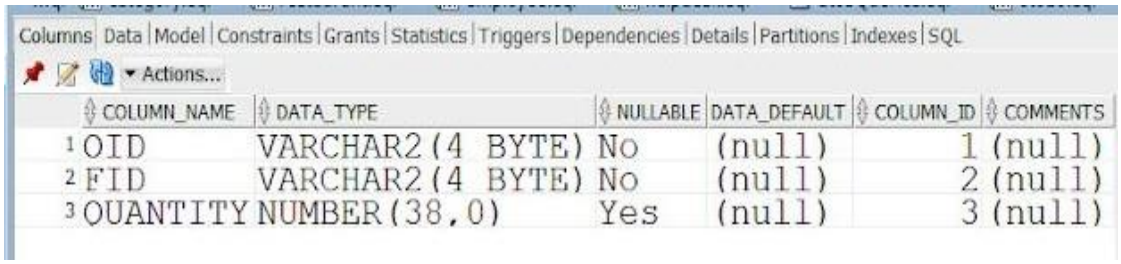
3.6 FOOD

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	FID	VARCHAR2(4 BYTE)	No	(null)	1	(null)
2	RID	VARCHAR2(4 BYTE)	No	(null)	2	(null)
3	CATID	VARCHAR2(4 BYTE)	Yes	(null)	3	(null)
4	FNAME	VARCHAR2(20 BYTE)	No	(null)	4	(null)
5	PRICE	NUMBER(38,0)	Yes	(null)	5	(null)
6	AVAILBALE	VARCHAR2(5 BYTE)	Yes	(null)	6	(null)
7	RATING	NUMBER(38,0)	Yes	(null)	7	(null)
8	OFFERS	VARCHAR2(25 BYTE)	Yes	(null)	8	(null)
9	FTAG	VARCHAR2(20 BYTE)	Yes	(null)	9	(null)

3.7 FOODINFO

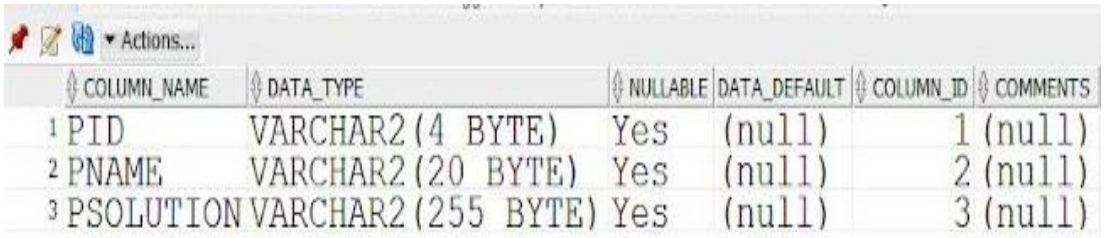


Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	OID	VARCHAR2(4 BYTE)	No	(null)	1	(null)
2	FID	VARCHAR2(4 BYTE)	No	(null)	2	(null)
3	QUANTITY	NUMBER(38,0)	Yes	(null)	3	(null)

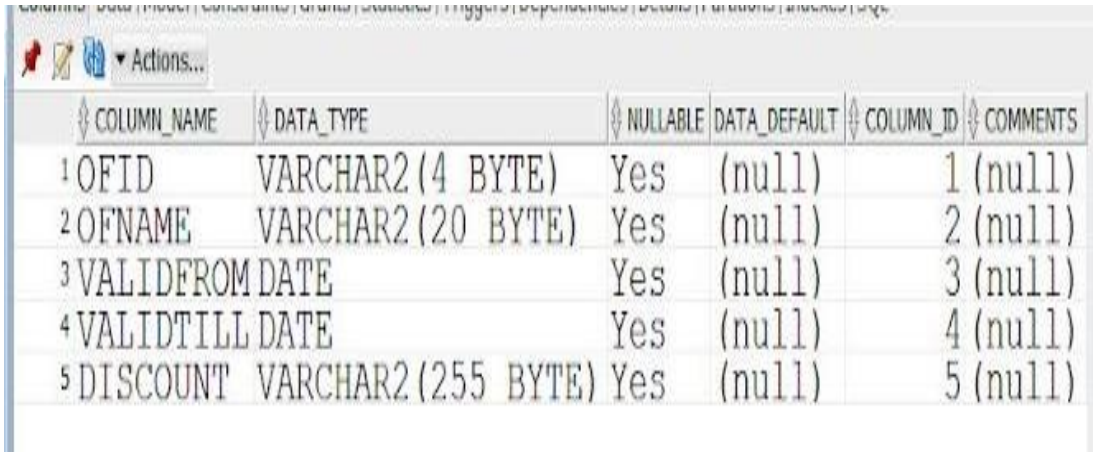
3.8 HELPDESK



Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PID	VARCHAR2(4 BYTE)	Yes	(null)	1	(null)
2	PNAME	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	PSOLUTION	VARCHAR2(255 BYTE)	Yes	(null)	3	(null)

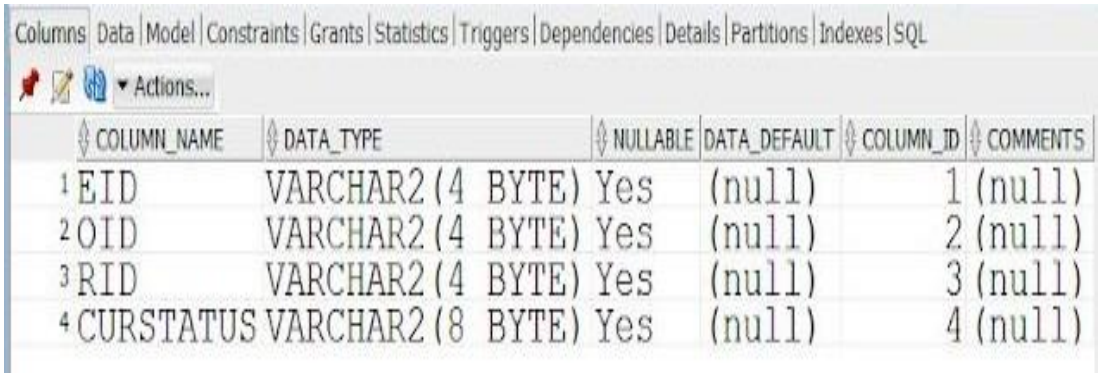
3.9 OFFERS



Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	OFID	VARCHAR2(4 BYTE)	Yes	(null)	1	(null)
2	OFNAME	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	VALIDFROM	DATE	Yes	(null)	3	(null)
4	VALIDTILL	DATE	Yes	(null)	4	(null)
5	DISCOUNT	VARCHAR2(255 BYTE)	Yes	(null)	5	(null)

3.10 ORDERINFO



Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EID	VARCHAR2(4 BYTE)	Yes	(null)	1	(null)
2	OID	VARCHAR2(4 BYTE)	Yes	(null)	2	(null)
3	RID	VARCHAR2(4 BYTE)	Yes	(null)	3	(null)
4	CURSTATUS	VARCHAR2(8 BYTE)	Yes	(null)	4	(null)

3.11 ORDERS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	OID	VARCHAR2(4 BYTE)	No	(null)	1	(null)
2	CID	VARCHAR2(4 BYTE)	No	(null)	2	(null)
3	OFID	VARCHAR2(4 BYTE)	Yes	(null)	3	(null)
4	ODATE	VARCHAR2(10 BYTE)	Yes	(null)	4	(null)
5	OTIME	VARCHAR2(8 BYTE)	Yes	(null)	5	(null)
6	OPRICE	FLOAT	Yes	(null)	6	(null)

3.12 PAYMENT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PID	VARCHAR2(4 BYTE)	No	(null)	1	(null)
2	OID	VARCHAR2(4 BYTE)	No	(null)	2	(null)
3	PMODE	VARCHAR2(20 BYTE)	No	(null)	3	(null)
4	PDATE	DATE	No	(null)	4	(null)
5	PTIME	VARCHAR2(8 BYTE)	No	(null)	5	(null)

3.13 RESTAURANT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RID	VARCHAR2(4 BYTE)	No	(null)	1	(null)
2	CITYID	VARCHAR2(4 BYTE)	Yes	(null)	2	(null)
3	RNAME	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	ADDRESS	VARCHAR2(255 BYTE)	Yes	(null)	4	(null)
5	RATING	NUMBER(38,0)	Yes	(null)	5	(null)
6	TIMEING	VARCHAR2(20 BYTE)	Yes	(null)	6	(null)
7	HELPLINE	NUMBER(10,0)	Yes	(null)	7	(null)

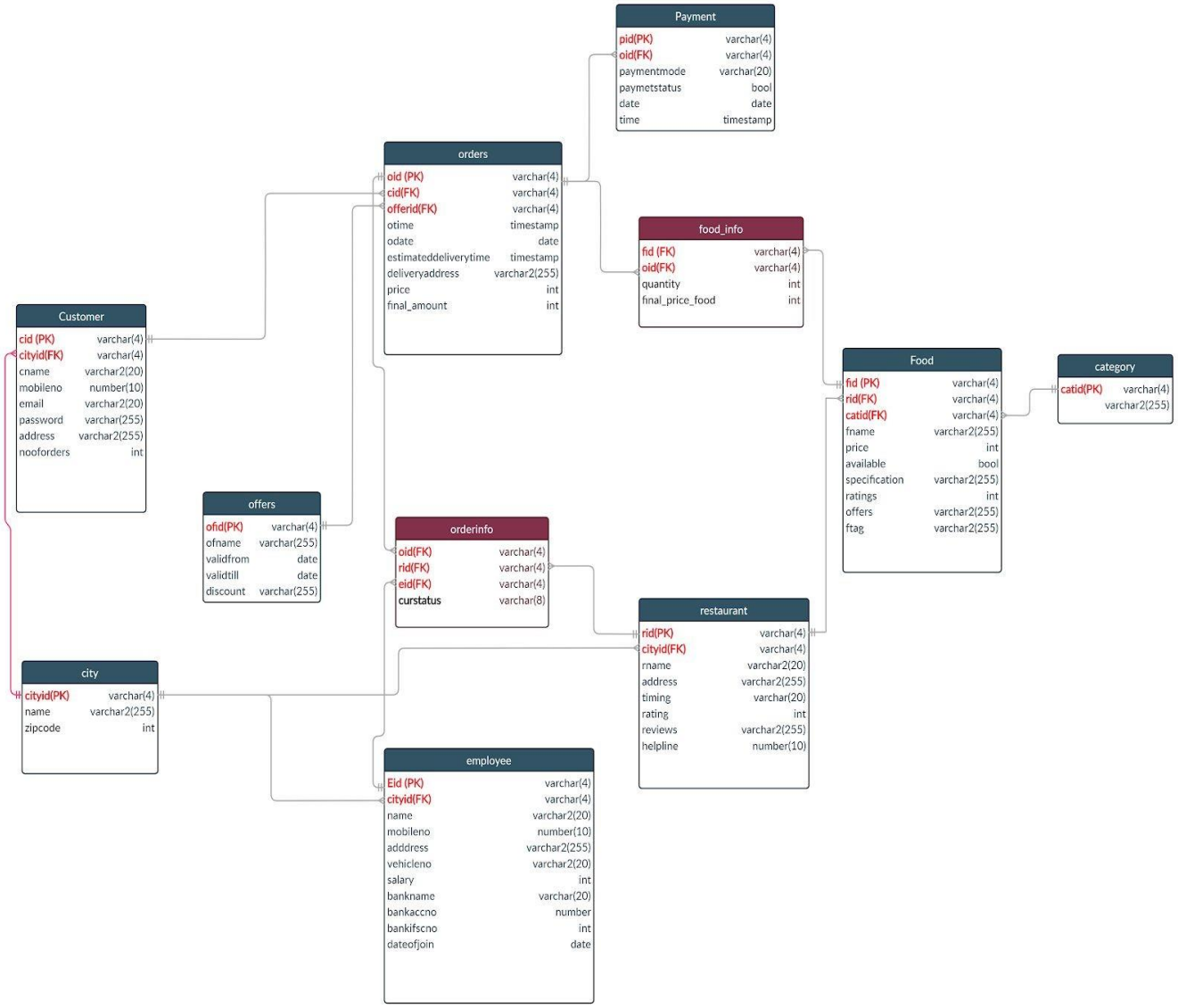
3.14 P234

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	P2NAME	VARCHAR2(20 BYTE)	Yes	(null)	1	(null)
2	P2ID	VARCHAR2(4 BYTE)	Yes	(null)	2	(null)
3	P2MODE	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)

4. SCHEMA DIAGRAM



FOR CLEAR VIEW OPEN BELOW LINK :

https://drive.google.com/file/d/18ixGWu0OyAYMZqg_oclzNVVvNfRyZKSs/view?usp=sharing

5. DATABASE IMPLEMENTATION

5.1 CREATE SCHEMA

5.1.1 ADMIN

```
create table admin(
aid varchar(4),
apswd varchar(8),
aname varchar(20));
```

5.1.2 CAT

```
create table cat(
    catid varchar(4) primary key ,
    catname varchar(20) not null
);
```

5.1.3 CITY

```
CREATE TABLE city (
    cityid    VARCHAR(4) NOT NULL PRIMARY KEY,
    cityname VARCHAR(255),
    zipcode   NUMBER(6)
);
```

5.1.4 CUSTOMER

```
CREATE TABLE customer (

        cid         VARCHAR2(4) NOT NULL PRIMARY KEY,
        cname       VARCHAR(20) NOT NULL,
        mobileno    NUMBER(10) NOT NULL,
        email       VARCHAR(20) NOT NULL,
        passwd      VARCHAR(255) NOT NULL,
        address     VARCHAR(255) NOT NULL,
        cityid      VARCHAR(4) NOT NULL,
        nooforders  INT

);
```

5.1.5 EMPLOYEE

```
CREATE TABLE employee (
   eid        VARCHAR(4) PRIMARY KEY,
   cityid     VARCHAR(4),
   ename      VARCHAR(20) NOT NULL,
   mobileno   VARCHAR(10) NOT NULL,
   address    VARCHAR(20) NOT NULL,
   vehicleno VARCHAR(8) NOT NULL,
   salary     NUMBER NOT NULL,
   doj        DATE NOT NULL,
   bankifsc   VARCHAR(8) NOT NULL,
   bankaccno NUMBER NOT NULL,
   CONSTRAINT ceid FOREIGN KEY ( cityid
   )
      REFERENCES city ( cityid )
);
```

5.1.6 FOOD

```
CREATE TABLE food (
   fid       VARCHAR(4) PRIMARY
   KEY, rid VARCHAR(4) NOT NULL,
   catid     VARCHAR(4),
   fname     VARCHAR(20) NOT NULL,
   price     INT,
   availbale VARCHAR(5),
   rating    INT,
   offers    VARCHAR(25),
   ftag      VARCHAR(20),
   CONSTRAINT frid FOREIGN KEY ( rid )
      REFERENCES restaurant ( rid ),
   CONSTRAINT fcatid FOREIGN KEY ( catid
   )
      REFERENCES cat ( catid )
);
```

5.1.7 FOODINFO

```
CREATE TABLE foodinfo (
   oid      VARCHAR(4) NOT NULL,
   fid      VARCHAR(4) NOT NULL,
   quantity  INT,
   CONSTRAINT inf FOREIGN KEY ( oid )
      REFERENCES orders ( oid ),
   CONSTRAINT inf2 FOREIGN KEY ( fid )
      REFERENCES food ( fid )
);
```

5.1.8 HELPDESK

```
create table helpdesk(
pid varchar(4) ,
pname varchar(20) ,
psolution varchar(255)
);
```

5.1.9 OFFERS

```
create table offers(
ofid varchar(4) ,
ofname varchar(20) ,
validfrom date ,
validtill date ,
discount varchar(255)
);
```

5.1.10 ORDERINFO

```
create table orderinfo (

eid varchar(4),
oid varchar(4),
rid varchar(4) ,
curstatus varchar(8),
foreign key(oid) references orders(oid),
foreign key(eid) references
employee(eid), foreign key(rid) references
restaurant(rid)
);
```

5.1.11 ORDERS

```
CREATE TABLE orders (
   oid     VARCHAR(4) NOT NULL PRIMARY KEY,
   cid     VARCHAR(4) NOT NULL,
   ofid    VARCHAR(4),
   odate   varchar(10),
   otime   VARCHAR(8),
   oprice  FLOAT
);
```

5.1.12 PAYMENT

```
CREATE TABLE payment (
   pid    VARCHAR(4) NOT NULL PRIMARY KEY,
   oid    VARCHAR(4) not null,
   pmode VARCHAR(20) not null,
   pdate DATE not null,
   ptime VARCHAR(8) not null,
   CONSTRAINT fk FOREIGN KEY ( oid
   )
      REFERENCES orders ( oid )
);
```

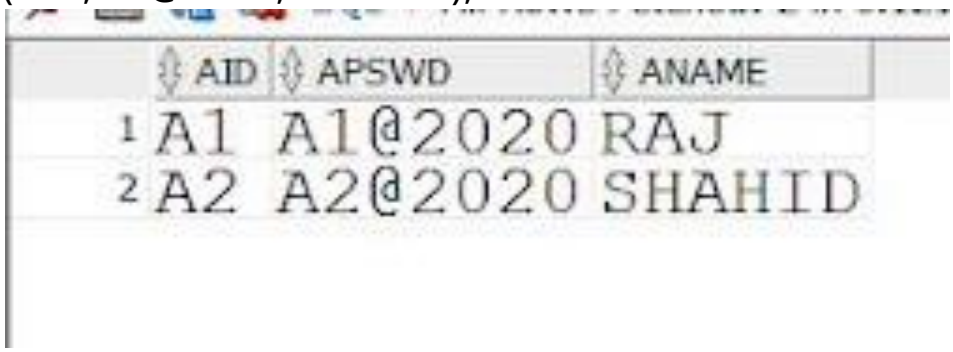
5.1.13 RESTAURANT

```
CREATE TABLE restaurant(
   rid      VARCHAR(4) PRIMARY KEY,
   cityid   VARCHAR(4),
   rname    VARCHAR(20),
   address  VARCHAR(255),
   rating   INT,
   timeing  VARCHAR(20),
   helpline NUMBER(10),
   constraint fkr foreign
   key(cityid) REFERENCES
   city(cityid)
);
```

5.2 INSERT DATA VALUE

5.2.1 ADMIN

```
REM INSERTING into SCOTT.ADMIN
SET DEFINE OFF;
Insert into SCOTT.ADMIN (AID,APSWD,ANAME) values
('A1','A1@2020','RAJ');
Insert into SCOTT.ADMIN (AID,APSWD,ANAME) values
('A2','A2@2020','SHAHID');
```



	AID	APSWD	ANAME
1	A1	A1@2020	RAJ
2	A2	A2@2020	SHAHID

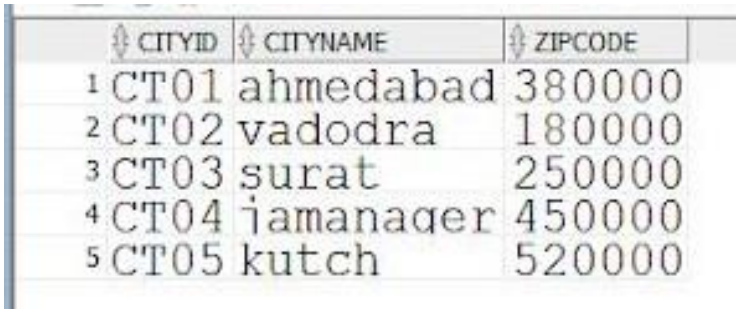
5.2.2 CAT

```
REM INSERTING into SCOTT.CAT
SET DEFINE OFF;
Insert into SCOTT.CAT (CATID,CATNAME) values ('CA1','VEG');
Insert into SCOTT.CAT (CATID,CATNAME) values ('CA2','NON-
VEG');
Insert into SCOTT.CAT (CATID,CATNAME) values
('CA3','FASTFOOD'); Insert into SCOTT.CAT (CATID,CATNAME)
values ('CA4','BAKERY');
Insert into SCOTT.CAT (CATID,CATNAME) values ('CA5','SOUTHINDIAN');
```

	CATID	CATNAME
1	CA1	VEG
2	CA2	NON-VEG
3	CA3	FASTFOOD
4	CA4	BAKERY
5	CA5	SOUTHINDIAN

5.2.3 CITY

REM INSERTING into SCOTT.CAT
SET DEFINE OFF;
Insert into SCOTT.CAT (CATID,CATNAME) values ('CA1','VEG');
Insert into SCOTT.CAT (CATID,CATNAME) values ('CA2','NON-VEG');
Insert into SCOTT.CAT (CATID,CATNAME) values ('CA3','FASTFOOD');
Insert into SCOTT.CAT (CATID,CATNAME) values ('CA4','BAKERY');
Insert into SCOTT.CAT (CATID,CATNAME) values ('CA5','SOUTHINDIAN');



	CITYID	CITYNAME	ZIPCODE
1	CT01	ahmedabad	380000
2	CT02	vadodra	180000
3	CT03	surat	250000
4	CT04	jamanagar	450000
5	CT05	kutch	520000

5.2.4 CUSTOMER

```
REM INSERTING into SCOTT.CUSTOMER
SET DEFINE OFF;
Insert into SCOTT.CUSTOMER
(CID,CNAME,MOBILENO,EMAIL,PASSWD,ADDRESS,CITYID,NOOFORDERS)
values ('C101','RAMESH',9814253689,'RAMESH@GMAIL.COM','RAMESH@101','E-
101,V RUNDAVAN8,SG HIGWAY,AHEMEABAD .','CT01',0);
Insert into SCOTT.CUSTOMER
(CID,CNAME,MOBILENO,EMAIL,PASSWD,ADDRESS,CITYID,NOOFORDERS)
values
('C102','SHAHID',7878251463,'SHAHID@GMAIL.COM','SHAHID@102','D102,VR
UNDAVAN9,MANINAGAR,AHEMEABAD .','CT04',3);
Insert into SCOTT.CUSTOMER
(CID,CNAME,MOBILENO,EMAIL,PASSWD,ADDRESS,CITYID,NOOFORDERS)
values ('C103','ROY',9512109554,'ROY@GMAIL.COM','ROY@103','D103,STAR
COMPLEX,KUBERNAGAR','CT05',9);
Insert into SCOTT.CUSTOMER
(CID,CNAME,MOBILENO,EMAIL,PASSWD,ADDRESS,CITYID,NOOFORDERS)
values
('C104','SMITH',9521748963,'SMITH@GMAIL.COM','SMITH@104','T03,GREEEN
CO.,NARSINH CHOWK','CT03',15);
Insert into SCOTT.CUSTOMER
(CID,CNAME,MOBILENO,EMAIL,PASSWD,ADDRESS,CITYID,NOOFORDERS)
values ('C105','RAJ',9924642130,'RAJ@GMAIL.COM','RAJ@105','E03,GREEEN
CO.,MANEK CHOWK','CT01',0);
```

	CID	CNAME	MOBILENO	EMAIL	PASSWD	ADDRESS	CITYID	NOOFORDERS
1	C101	RAMESH	9814253689	RAMESH@GMAIL.COM	RAMESH@101	E-101,VRUNDAVAN8,SG HIGWAY,AHEMEABAD .	CT01	0
2	C102	SHAHID	7878251463	SHAHID@GMAIL.COM	SHAHID@102	D102,VRUNDAVAN9,MANINAGAR,AHEMEABAD .	CT04	3
3	C103	ROY	9512109554	ROY@GMAIL.COM	ROY@103	D103,STAR COMPLEX,KUBERNAGAR	CT05	9
4	C104	SMITH	9521748963	SMITH@GMAIL.COM	SMITH@104	T03,GREEEN CO.,NARSINH CHOWK	CT03	15
5	C105	RAJ	9924642130	RAJ@GMAIL.COM	RAJ@105	E03,GREEEN CO.,MANEK CHOWK	CT01	0

5.2.5 EMPLOYEE

REM INSERTING into SCOTT.EMPLOYEE

SET DEFINE OFF;

Insert into SCOTT.EMPLOYEE (EID,CITYID,ENAME,MOBILENO,ADDRESS,VEHICLENO,SALARY,DOJ,BANKIFSC,BANKACCNO) values ('E101','CT05','SUNDAR','9923219291','VRUNDAVAN8','GJ053321',22000,to_date('12-APR-18','DD-MON-RR'),'SBIN0023',23482812);

Insert into SCOTT.EMPLOYEE (EID,CITYID,ENAME,MOBILENO,ADDRESS,VEHICLENO,SALARY,DOJ,BANKIFSC,BANKACCNO) values ('E102','CT01','RAM','9823249291','VRUNDAVAN11','GJ015511',44000,to_date('12-APR-18','DD-MON-RR'),'UBIN0023',33582812);

Insert into SCOTT.EMPLOYEE (EID,CITYID,ENAME,MOBILENO,ADDRESS,VEHICLENO,SALARY,DOJ,BANKIFSC,BANKACCNO) values ('E103','CT02','ARPIT','8943249291','SHAJANAND-01','GJ023231',10000,to_date('25-APR-17','DD-MON-RR'),'BOIN0023',98582812);

Insert into SCOTT.EMPLOYEE (EID,CITYID,ENAME,MOBILENO,ADDRESS,VEHICLENO,SALARY,DOJ,BANKIFSC,BANKACCNO) values ('E104','CT01','DHARMIL','7843002477','SG-01','GJ015240',32000,to_date('25-MAR-19','DD-MON-RR'),'BOBN0023',21641812);

Insert into SCOTT.EMPLOYEE (EID,CITYID,ENAME,MOBILENO,ADDRESS,VEHICLENO,SALARY,DOJ,BANKIFSC,BANKACCNO) values ('E105','CT03','ANAND','7462178958','CG-01','GJ038540',70000,to_date('21-APR-15','DD-MON-RR'),'BOBN0027',21641819);

	EID	CITYID	ENAME	MOBILENO	ADDRESS	VEHICLENO	SALARY	DOJ	BANKIFSC	BANKACCNO
1	E101	CT05	SUNDAR	9923219291	VRUNDAVAN8	GJ053321	22000	12-APR-18	SBIN0023	23482812
2	E102	CT01	RAM	9823249291	VRUNDAVAN11	GJ015511	44000	12-APR-18	UBIN0023	33582812
3	E103	CT02	ARPIT	8943249291	SHAJANAND-01	GJ023231	10000	25-APR-17	BOIN0023	98582812
4	E104	CT01	DHARMIL	7843002477	SG-01	GJ015240	32000	25-MAR-19	BOBN0023	21641812
5	E105	CT03	ANAND	7462178958	CG-01	GJ038540	70000	21-APR-15	BOBN0027	21641819

5.2.6 FOOD

```
REM INSERTING into SCOTT.FOOD
SET DEFINE OFF;
Insert into SCOTT.FOOD
(FID,RID,CATID,FNAME,PRICE,AVAILBALE,RATING,OFFERS,FTAG) values
('F101','R401','CA1','PANEER',150,'TRUE',7,'50%OFF','BESTSELLER');
Insert into SCOTT.FOOD
(FID,RID,CATID,FNAME,PRICE,AVAILBALE,RATING,OFFERS,FTAG) values
('F102','R401','CA2','TIKKA',300,'FALSE',7,null,null);
Insert into SCOTT.FOOD
(FID,RID,CATID,FNAME,PRICE,AVAILBALE,RATING,OFFERS,FTAG) values
('F103','R402','CA3','DABELI',30,'TRUE',10,null,'BESTSELLER');
Insert into SCOTT.FOOD
(FID,RID,CATID,FNAME,PRICE,AVAILBALE,RATING,OFFERS,FTAG) values
('F104','R403','CA3','VADAPAV',20,'TRUE',10,null,null);
Insert into SCOTT.FOOD
(FID,RID,CATID,FNAME,PRICE,AVAILBALE,RATING,OFFERS,FTAG) values
('F105','R402','CA2','MUTTONKADHAI',450,'TRUE',10,'50%OFF','BESTSELLE
R');
```

	FID	RID	CATID	FNAME	PRICE	AVAILBALE	RATING	OFFERS	FTAG
1	F101	R401	CA1	PANEER	150	TRUE	7	50%OFF	BESTSELLE
2	F102	R401	CA2	TIKKA	300	FALSE	7	(null)	(null)
3	F103	R402	CA3	DABELI	30	TRUE	10	(null)	BESTSELLE
4	F104	R403	CA3	VADAPAV	20	TRUE	10	(null)	(null)
5	F105	R402	CA2	MUTTONKADHAI	450	TRUE	10	50%OFF	BESTSELLE

5.2.7 FOODINFO

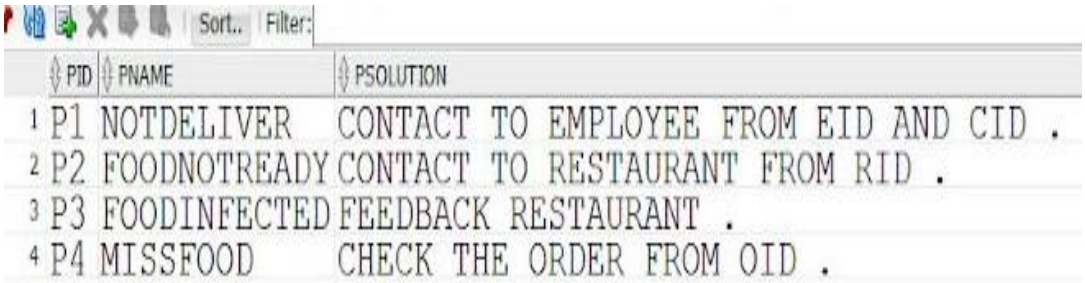
```
REM INSERTING into SCOTT.FOODINFO
SET DEFINE OFF;
Insert into SCOTT.FOODINFO (OID,FID,QUANTITY) values ('O107','F103',4);
Insert into SCOTT.FOODINFO (OID,FID,QUANTITY) values ('O109','F103',4);
Insert into SCOTT.FOODINFO (OID,FID,QUANTITY) values ('O101','F101',2);
Insert into SCOTT.FOODINFO (OID,FID,QUANTITY) values ('O101','F105',1);
Insert into SCOTT.FOODINFO (OID,FID,QUANTITY) values ('O101','F103',2);
Insert into SCOTT.FOODINFO (OID,FID,QUANTITY) values ('O102','F103',3);
Insert into SCOTT.FOODINFO (OID,FID,QUANTITY) values ('O102','F104',1);
Insert into SCOTT.FOODINFO (OID,FID,QUANTITY) values ('O103','F101',4);
Insert into SCOTT.FOODINFO (OID,FID,QUANTITY) values ('O104','F105',3);
Insert into SCOTT.FOODINFO (OID,FID,QUANTITY) values ('O105','F101',2);
Insert into SCOTT.FOODINFO (OID,FID,QUANTITY) values ('O105','F105',1);
```

	OID	FID	QUANTITY
1	O107	F103	4
2	O109	F103	4
3	O101	F101	2
4	O101	F105	1
5	O101	F103	2
6	O102	F103	3
7	O102	F104	1
8	O103	F101	4
9	O104	F105	3
10	O105	F101	2
11	O105	F105	1

5.2.8 PAYMENT

	PID	OID	PMODE	PDATE	PTIME
1	P301	O101	PAYTM	14-AUG-20	21:32:12
2	P302	O102	OFFLINE	20-AUG-20	03:32:12
3	P303	O103	GPAY	15-MAY-20	04:32:12
4	P304	O104	OFFLINE	15-MAY-20	04:32:12
5	P305	O105	OFFLINE	11-JUL-20	05:32:12

5.2.9 HELPDESK



	PID	PNAME	PSOLUTION
1	P1	NOTDELIVER	CONTACT TO EMPLOYEE FROM EID AND CID .
2	P2	FOODNOTREADY	CONTACT TO RESTAURANT FROM RID .
3	P3	FOODINFECTED	FEEDBACK RESTAURANT .
4	P4	MISSFOOD	CHECK THE ORDER FROM OID .

5.2.10 ORDERINFO

REM INSERTING into SCOTT.ORDERINFO

SET DEFINE OFF;

Insert into SCOTT.ORDERINFO (EID,OID,RID,CURSTATUS) values ('E104','O101','R401','PANDING');

Insert into SCOTT.ORDERINFO (EID,OID,RID,CURSTATUS) values ('E104','O104','R403','ONTHEWAY');

Insert into SCOTT.ORDERINFO (EID,OID,RID,CURSTATUS) values ('E101','O103','R401','COOCKING');

Insert into SCOTT.ORDERINFO (EID,OID,RID,CURSTATUS) values ('E105','O105','R403','PANDING');

Insert into SCOTT.ORDERINFO (EID,OID,RID,CURSTATUS) values ('E101','O102','R401','ONTHEWAY');

Insert into SCOTT.ORDERINFO (EID,OID,RID,CURSTATUS) values ('E102','O106','R402','PANDING');

Insert into SCOTT.ORDERINFO (EID,OID,RID,CURSTATUS) values ('E103','O107','R405','PANDING');

Insert into SCOTT.ORDERINFO (EID,OID,RID,CURSTATUS) values ('E103','O109','R405','PANDING');

	EID	OID	RID	CURSTATUS
1	E104	O101	R401	PANDING
2	E104	O104	R403	ONTHEWAY
3	E101	O103	R401	COOCKING
4	E105	O105	R403	PANDING
5	E101	O102	R401	ONTHEWAY
6	E102	O106	R402	PANDING
7	E103	O107	R405	PANDING
8	E103	O109	R405	PANDING

5.2.11 OFFERS

REM INSERTING into SCOTT.OFFERS

SET DEFINE OFF;

Insert into SCOTT.OFFERS (OFID,OFNAME,VALIDFROM,VALIDTILL,DISCOUNT) values ('OF1','SUMMERDHMAKA',to_date('14-MAY-20','DD-MON-RR'),to_date('14-JUN-20','DD-MON-RR'),'25%OFF');

Insert into SCOTT.OFFERS (OFID,OFNAME,VALIDFROM,VALIDTILL,DISCOUNT) values ('OF2','DIWALI',to_date('13-MAY-19','DD-MON-RR'),to_date('30-JUN-19','DD-MON-RR'),'55%OFF');

Insert into SCOTT.OFFERS (OFID,OFNAME,VALIDFROM,VALIDTILL,DISCOUNT) values ('OF3','CHRISTMAS',to_date('25-DEC-19','DD-MON-RR'),to_date('25-DEC-19','DD-MON-RR'),'35%OFF');

Insert into SCOTT.OFFERS (OFID,OFNAME,VALIDFROM,VALIDTILL,DISCOUNT) values ('OF4','NAVRATRI',to_date('01-AUG-17','DD-MON-RR'),to_date('29-AUG-17','DD-MON-RR'),'45%OFF');

Insert into SCOTT.OFFERS (OFID,OFNAME,VALIDFROM,VALIDTILL,DISCOUNT) values ('OF5','WINTERSALE',to_date('01-JAN-19','DD-MON-RR'),to_date('14-FEB-19','DD-MON-RR'),'75%OFF');

	OFID	OFNAME	VALIDFROM	VALIDTILL	DISCOUNT
1	OF1	SUMMERDHMAKA	14-MAY-20	14-JUN-20	25%OFF
2	OF2	DIWALI	13-MAY-19	30-JUN-19	55%OFF
3	OF3	CHRISTMAS	25-DEC-19	25-DEC-19	35%OFF
4	OF4	NAVRATRI	01-AUG-17	29-AUG-17	45%OFF
5	OF5	WINTERSALE	01-JAN-19	14-FEB-19	75%OFF

5.2.12 ORDERS

REM INSERTING into SCOTT.ORDERS

SET DEFINE OFF;

Insert into SCOTT.ORDERS (OID,CID,OFID,ODATE,OTIME,OPRICE) values ('O101','C102','OF3','2020-02-01','03:14:01',278.928);

Insert into SCOTT.ORDERS (OID,CID,OFID,ODATE,OTIME,OPRICE) values ('O102','C101',null,'2020-03-11','04:35:56',1243.21);

Insert into SCOTT.ORDERS (OID,CID,OFID,ODATE,OTIME,OPRICE) values ('O103','C102','OF1','2020-01-11','06:35:56',5689.21);

Insert into SCOTT.ORDERS (OID,CID,OFID,ODATE,OTIME,OPRICE) values ('O104','C102','OF5','2020-08-04','21:35:56',235);

Insert into SCOTT.ORDERS (OID,CID,OFID,ODATE,OTIME,OPRICE) values ('O105','C103','OF1','2020-08-24','23:35:56',63.84);

Insert into SCOTT.ORDERS (OID,CID,OFID,ODATE,OTIME,OPRICE) values ('O106','C102','OF1','2020-08-24','23:35:56',567.21);

Insert into SCOTT.ORDERS (OID,CID,OFID,ODATE,OTIME,OPRICE) values ('O107','C105','OF3','2018-07-12','12:35:56',789.52);

Insert into SCOTT.ORDERS (OID,CID,OFID,ODATE,OTIME,OPRICE) values ('O108','C103','OF1','2019-11-13','13:35:56',889.52);

Insert into SCOTT.ORDERS (OID,CID,OFID,ODATE,OTIME,OPRICE) values ('O109','C103','OF1','2019-11-13','13:35:56',889.52);

Insert into SCOTT.ORDERS (OID,CID,OFID,ODATE,OTIME,OPRICE) values ('O110','C103','OF1','2019-11-13','13:35:56',889.52);

Insert into SCOTT.ORDERS (OID,CID,OFID,ODATE,OTIME,OPRICE) values

('O111','C103','OF1','2019-11-13','13:35:56',889.52);

Insert into SCOTT.ORDERS (OID,CID,OFID,ODATE,OTIME,OPRICE) values ('O112','C103','OF1','2019-11-13','13:35:56',889.52);

	OID	CID	OFID	ODATE	OTIME	OPRICE
1	O101	C102	OF3	2020-02-01	03:14:01	278.928
2	O102	C101	(null)	2020-03-11	04:35:56	1243.21
3	O103	C102	OF1	2020-01-11	06:35:56	5689.21
4	O104	C102	OF5	2020-08-04	21:35:56	235
5	O105	C103	OF1	2020-08-24	23:35:56	63.84
6	O106	C102	OF1	2020-08-24	23:35:56	567.21
7	O107	C105	OF3	2018-07-12	12:35:56	789.52
8	O108	C103	OF1	2019-11-13	13:35:56	889.52
9	O109	C103	OF1	2019-11-13	13:35:56	889.52
10	O110	C103	OF1	2019-11-13	13:35:56	889.52
11	O111	C103	OF1	2019-11-13	13:35:56	889.52
12	O112	C103	OF1	2019-11-13	13:35:56	889.52

5.3 QUERIES

5.3.1 > display all the information of customer who lives in ahmedabad .

Select * from customer where cityid = 'CT01' ;

	CID	CNAME	MOBILENO	EMAIL	PASSWD	ADDRESS	CITYID	NOOFORDERS
1	C101	RAMESH	9814253689	RAMESH@GMAIL.COM	RAMESH@101	E-101,VRUNDAVAN8,SG HIGWAY,AHEMEABAD .	CT01	0
2	C105	RAJ	9924642130	RAJ@GMAIL.COM	RAJ@105	E03,GREEEN CO.,MANEK CHOWK	CT01	0

5.3.2 > count the number of employee whose salary is greater than 30000 and also find the average salary of those employee .

```
SELECT
   AVG(salary) AS avgsal,
   COUNT(salary)
FROM
   employee
WHERE
   salary > 30000;
```

	AVGSAL	COUNT(SALARY)
1	48666.66...	3

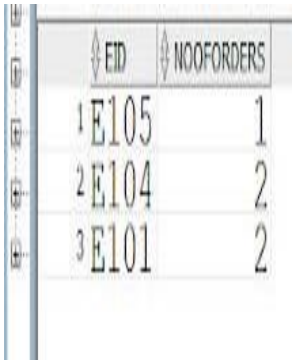
5.3.3 > display all information of restaurant which rating is between 6 to 10 .

```
SELECT
   *
FROM
   restaurant
WHERE
   rating BETWEEN 6 AND 10;
```

	RID	CITYID	RNAME	ADDRESS	RATING	TIMEING	HELPLINE
1	R401	CT01	GMB	SHAHIBAUG	7	10AM-10PM	9985412310
2	R402	CT01	BOMBAY	G-SQUARE	8	9AM-10PM	8985412321
3	R403	CT03	BURGERKING	HORIZAN	10	10AM-10PM	9825412321

5.3.4 count the number of orders assign to each employee with employee id .

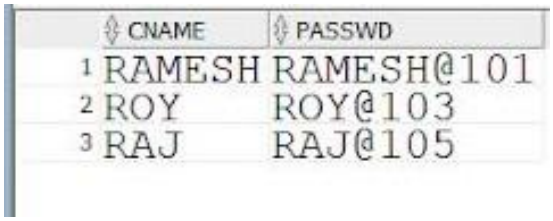
```
SELECT
   eid,
   COUNT(oid) AS
nooforders FROM
   orderinfo
GROUP BY
   eid
ORDER BY
   COUNT(oid);
```



	EID	NOOFORDERS
1	E105	1
2	E104	2
3	E101	2

5.3.5 display name and password of all customers whose name starting with the 'R'

```
SELECT
   cname,
   passwd
FROM
   customer
WHERE
   cname LIKE 'R%';
```



	CNAME	PASSWD
1	RAMESH	RAMESH@101
2	ROY	ROY@103
3	RAJ	RAJ@105

5.3.6 display the food name along with catagory name name which is belong to same category .

```
select A.fname as food1 , B.fname as food2 , (select catname as catagory from
cat where catid = A.catid)
from food A ,food B
WHERE A.catid = B.catid and A.fname<>B.fname
order by A.catid ;
```

	FOOD1	FOOD2	(SELECTCATNAMEASCATAGORYFROMCATWHERECATID=A.CATID)
1	MUTTONKADHAI	TIKKA	NON-VEG
2	TIKKA	MUTTONKADHAI	NON-VEG
3	DABELI	VADAPAV	FASTFOOD
4	VADAPAV	DABELI	FASTFOOD

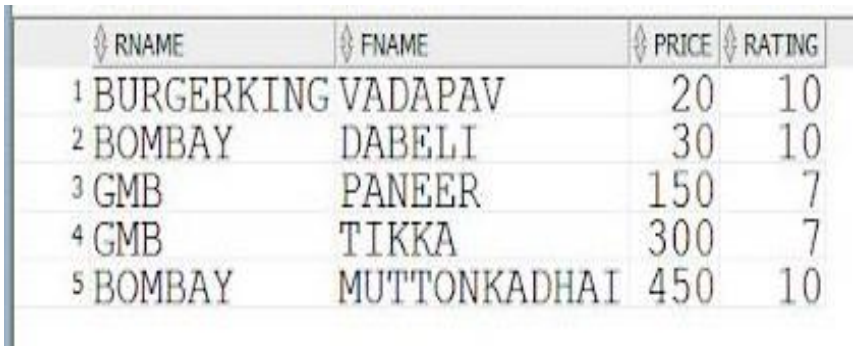
5.3.7 find the detail of an ordere i.e. order id , amount , date , which customer gives the order and which employee works for that order .

```
select O.oid , O.odate , O.oprice , employee.ename , customer.cname
from orders O
join orderinfo on O.oid = orderinfo.oid
join employee on employee.eid = orderinfo.eid
join customer on O.cid = customer.cid ;
```

	OID	ODATE	OPRICE	ENAME	CNAME
1	O102	2020-03-11	456.25	SUNDAR	RAMESH
2	O101	2020-02-01	456.25	DHARMIL	SHAHID
3	O104	2020-08-04	500.21	DHARMIL	SHAHID
4	O103	2020-01-11	1244.21	SUNDAR	SHAHID
5	O105	2020-08-24	567.21	ANAND	ROY

5.3.8 display the name of food , rating and price (low to high) along with resatuarant name .

```
SELECT(
      SELECT
         rname
      FROM
         restaurant
      WHERE
         food.rid = restaurant.rid
   ) AS rname,
   fname,price,rating
FROM
   food
ORDER BY
   Price;
```



	RNAME	FNAME	PRICE	RATING
1	BURGERKING	VADAPAV	20	10
2	BOMBAY	DABELI	30	10
3	GMB	PANEER	150	7
4	GMB	TIKKA	300	7
5	BOMBAY	MUTTONKADHAI	450	10

5.3.9 display the detail of the order for all customer with name of customer .

```
select C.cid , C.cname , O.oid , O.odate,O.otime

from customer C

left join orders O

on O.cid = C.cid

order by O.odate;
```

	CID	CNAME	OID	ODATE	OTIME
1	C102	SHAHID	O103	2020-01-11	06:35:56
2	C102	SHAHID	O101	2020-02-01	03:14:01
3	C101	RAMESH	O102	2020-03-11	04:35:56
4	C102	SHAHID	O104	2020-08-04	21:35:56
5	C103	ROY	O105	2020-08-24	23:35:56
6	C104	SMITH	(null)	(null)	(null)
7	C105	RAJ	(null)	(null)	(null)

5.3.10 find the payment information that paid by cash older to latest .

```
SELECT
   *
FROM
   payment
WHERE
   pmode = 'OFFLINE'
ORDER BY
   Pdate;
```

	PID	OID	PMODE	PDATE	PTIME
1	P304	O104	OFFLINE	15-MAY-20	04:32:12
2	P305	O105	OFFLINE	11-JUL-20	05:32:12
3	P302	O102	OFFLINE	20-AUG-20	03:32:12

5.4 PL/SQL BLOCK :

- **create a procedure that calculate the price of the order from oid including quantity .**

```
create or replace procedure amt(o_oid
VARCHAR) is
   temp foodinfo%ROWTYPE;
   val int ;
begin
   select * into temp from foodinfo where oid = o_oid ;
   select price into val from food where fid = temp.fid ;
   dbms_output.put_line(val*temp.quantity);
end ;
execute amt('O104') ;
```

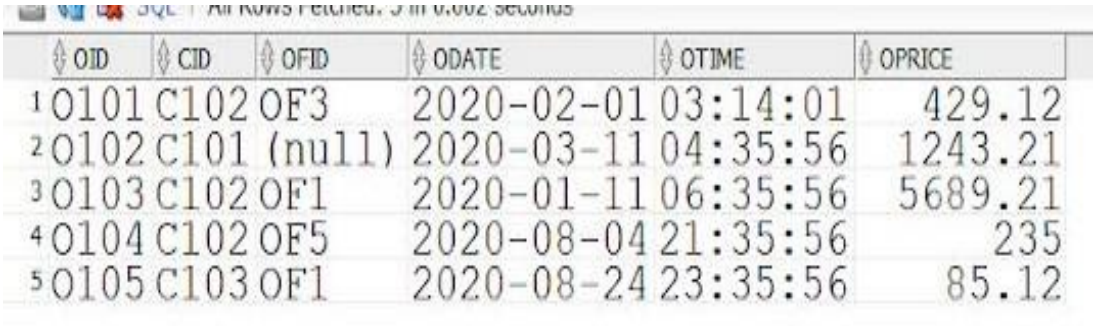
```
Procedure AMT compiled


PL/SQL procedure successfully completed.
```

5.5 FUNCTIONS

- **create the function cal that calculate the amount after applying offer .**

```
create or replace function cal (ofid varchar ,oprice float)
return float is
   total float:=0 ;
begin
   if (ofid='OF1') then
      total := oprice*25/100;
   elsif (ofid='OF2') then
      total := oprice*55/100;
   elsif (ofid='OF3') then
      total := oprice*35/100;
   elsif (ofid='OF4') then
      total := oprice*45/100;
   elsif (ofid='OF5') then
      total := oprice*75/100 ;
   else
      total := oprice ;
   end if;

   dbms_output.put_line(total);
   return total;
end ;

begin
   dbms_output.put_line(cal('OF2',123.21));
  update orders set oprice = 429.12-cal('OF3',429.12) where oid =
'O101' ;
   update orders set oprice = 85.12-cal('OF1',85.12) where oid =
'O105' ;
end ;
```

Before Executing Queries :



	OID	CID	OFID	ODATE	OTIME	OPRICE
1	O101	C102	OF3	2020-02-01	03:14:01	429.12
2	O102	C101	(null)	2020-03-11	04:35:56	1243.21
3	O103	C102	OF1	2020-01-11	06:35:56	5689.21
4	O104	C102	OF5	2020-08-04	21:35:56	235
5	O105	C103	OF1	2020-08-24	23:35:56	85.12

After Executing Queries :

	OID	CID	OFID	ODATE	OTIME	OPRICE
1	O101	C102	OF3	2020-02-01	03:14:01	278.928
2	O102	C101	(null)	2020-03-11	04:35:56	1243.21
3	O103	C102	OF1	2020-01-11	06:35:56	5689.21
4	O104	C102	OF5	2020-08-04	21:35:56	235
5	O105	C103	OF1	2020-08-24	23:35:56	63.84

5.6 CURSOR

- **cursor gives the payment information with customer name, order id , mode of payment into p234 table .**

```
create table p234(
CURnameVARCHAR(20),
CURid varchar(4),
CURmode VARCHAR(20)
);

declare
Cusname customer.cname%type ;
Orid orders.oid%type ;
Payst payment.pmode%type ;
cursor pinfo is select C.cname , O.oid , P.pmode
from orders O
inner join customer C on O.cid = C.cid
inner join payment P on O.oid = P.oid ;

begin

   open pinfo ;

   loop
   fetch pinfo into Cusname,Orid,Payst ;
   exit when pinfo%notfound ;
   insert into p234 values(Cusname,Orid,Payst) ;
   end loop ;

   close pinfo ;

end ;
```

```
SELECT * FROM P234;
commit ;
```

SQL | All Rows Fetched: 5 in 0.015 sec

	P2NAME	P2ID	P2MODE
1	RAMESH	0102	OFFLINE
2	SHAHID	0104	OFFLINE
3	SHAHID	0103	GPAY
4	SHAHID	0101	PAYTM
5	ROY	0105	OFFLINE

5.7 TRIGGERS

- **The "N_ORDER" trigger called after new order arrives in 'orders' table it will insert the reflact order information in 'orderinfo' and 'foodinfo' table .**

```
create or replace trigger n_order
after insert on orders
for each row
begin
    insert into
orderinfo(eid,oid,rid,curstatus)
values('E103',:new.oid
  ,'R405','PANDING');
    insert into
foodinfo(oid,fid,quantity)
values(:new.oid,'F103',4);
end ;
/
```

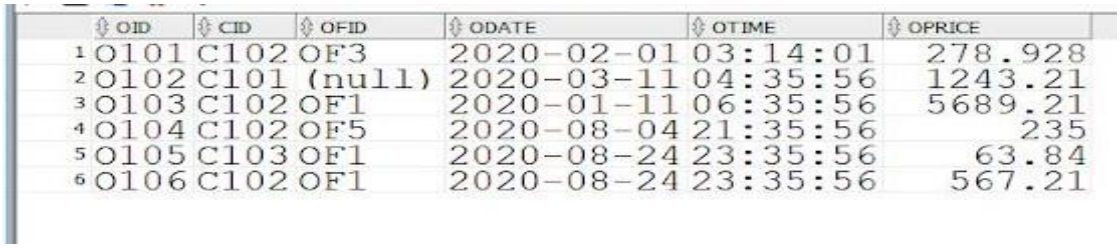
FIRING TRIGGER BY INSERTING ROW :

```
insert into orders values(
'O107','C103','OF1','2019-11-13','13:35:56',889.52
```

);

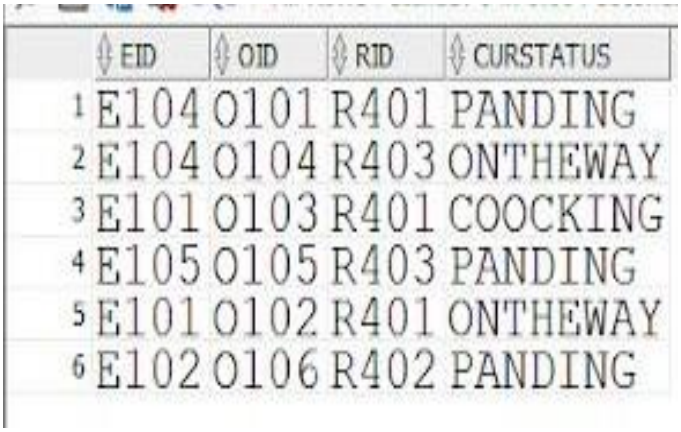
BEFORE APPLYING THE TRIGGER TABLES :

ORDERS :



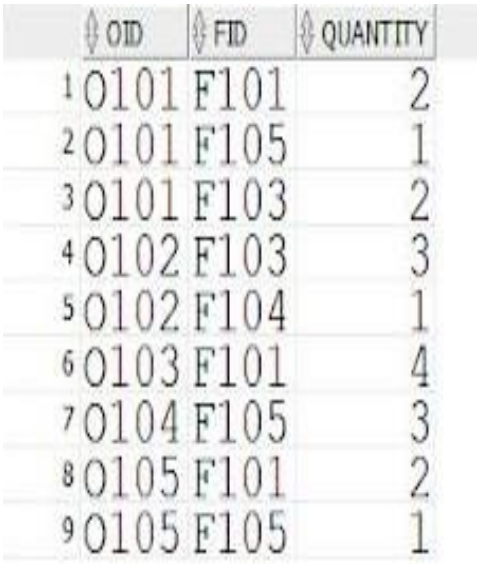
	OID	CID	OFID	ODATE	OTIME	OPRICE
1	O101	C102	OF3	2020-02-01	03:14:01	278.928
2	O102	C101	(null)	2020-03-11	04:35:56	1243.21
3	O103	C102	OF1	2020-01-11	06:35:56	5689.21
4	O104	C102	OF5	2020-08-04	21:35:56	235
5	O105	C103	OF1	2020-08-24	23:35:56	63.84
6	O106	C102	OF1	2020-08-24	23:35:56	567.21

FOODINFO :



	EID	OID	RID	CURSTATUS
1	E104	O101	R401	PANDING
2	E104	O104	R403	ONTHEWAY
3	E101	O103	R401	COOCKING
4	E105	O105	R403	PANDING
5	E101	O102	R401	ONTHEWAY
6	E102	O106	R402	PANDING

ORDERINFO :



	OID	FID	QUANTITY
1	O101	F101	2
2	O101	F105	1
3	O101	F103	2
4	O102	F103	3
5	O102	F104	1
6	O103	F101	4
7	O104	F105	3
8	O105	F101	2
9	O105	F105	1

AFTER APPLYING THE TRIGGER TABLES :

```
Trigger N_ORDER compiled


Trigger N_ORDER compiled
```

```

1 row inserted.
```

FOODINFO :

SQL | All Rows Fetched: 10 i

	OID	FID	QUANTITY
1	O107	F103	4
2	O101	F101	2
3	O101	F105	1
4	O101	F103	2
5	O102	F103	3
6	O102	F104	1
7	O103	F101	4
8	O104	F105	3
9	O105	F101	2
10	O105	F105	1

ORDERINFO :

SQL | All Rows Fetched: 7 in 0.008 seconds

	EID	OID	RID	CURSTATUS
1	E104	O101	R401	PANDING
2	E104	O104	R403	ONTHEWAY
3	E101	O103	R401	COOCKING
4	E105	O105	R403	PANDING
5	E101	O102	R401	ONTHEWAY
6	E102	O106	R402	PANDING
7	E103	O107	R405	PANDING

- **The "CNT_INTR" trigger increment the 'nooforder' in customer table for customer whose placed the order .**

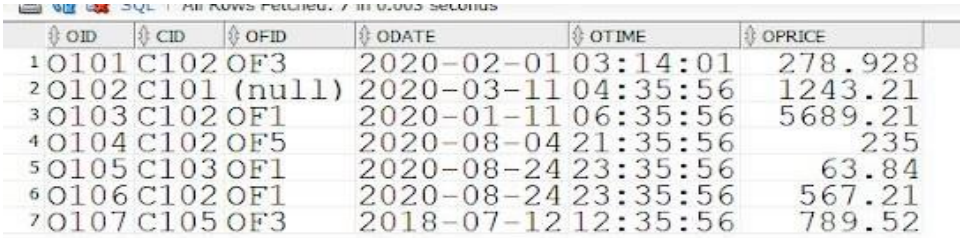
```
create or replace trigger
cnt_intr
after insert on orders
for each row
declare
   cnt int ;

begin
   select nooforders into cnt
from customer where cid =
:new.cid ;
   update customer set
nooforders=cnt+1 where cid =
:new.cid ;
end ;
/
```

TRIGGER FIRE BY THIS QUERY:

```
insert into orders values(
'O108','C103','OF1','2019-11-13',
'13:35:56',889.52
);
```

BEFORE FIRING TRIGGER
CONTENT OF TABLES :

ORDERS :



	OID	CID	OFID	ODATE	OTIME	OPRICE
1	O101	C102	OF3	2020-02-01	03:14:01	278.928
2	O102	C101	(null)	2020-03-11	04:35:56	1243.21
3	O103	C102	OF1	2020-01-11	06:35:56	5689.21
4	O104	C102	OF5	2020-08-04	21:35:56	235
5	O105	C103	OF1	2020-08-24	23:35:56	63.84
6	O106	C102	OF1	2020-08-24	23:35:56	567.21
7	O107	C105	OF3	2018-07-12	12:35:56	789.52

CUSTOMER :

	CID	CNAME	MOBILENO	EMAIL	PASSWD	ADDRESS	CITYID	NOOFORDERS
1	C101	RAMESH	9814253689	RAMESH@GMAIL.COM	RAMESH@101	E-101,VRUNDAVAN8,SG HIGWAY,AHEMEABAD .	CT01	0
2	C102	SHAHID	7878251463	SHAHID@GMAIL.COM	SHAHID@102	D102,VRUNDAVAN9,MANINAGAR,AHEMEABAD .	CT04	3
3	C103	ROY	9512109554	ROY@GMAIL.COM	ROY@103	D103,STAR COMPLEX,KUBERNAGAR	CT05	5
4	C104	SMITH	9521748963	SMITH@GMAIL.COM	SMITH@104	T03,GREEEN CO.,NARSINH CHOWK	CT03	15
5	C105	RAJ	9924642130	RAJ@GMAIL.COM	RAJ@105	E03,GREEEN CO.,MANEK CHOWK	CT01	0

AFTER FIRING TRIGGER CONTENT OF TABLES :

```
Trigger CNT_INTR compiled
```

ORDERS :

	OID	CID	OFID	ODATE	OTIME	OPRICE
1	O101	C102	OF3	2020-02-01	03:14:01	278.928
2	O102	C101	(null)	2020-03-11	04:35:56	1243.21
3	O103	C102	OF1	2020-01-11	06:35:56	5689.21
4	O104	C102	OF5	2020-08-04	21:35:56	235
5	O105	C103	OF1	2020-08-24	23:35:56	63.84
6	O106	C102	OF1	2020-08-24	23:35:56	567.21
7	O107	C105	OF3	2018-07-12	12:35:56	789.52
8	O108	C103	OF1	2019-11-13	13:35:56	889.52

CUSTOMER :

	CID	CNAME	MOBILENO	EMAIL	PASSWD	ADDRESS	CITYID	NOOFORDERS
1	C101	RAMESH	9814253689	RAMESH@GMAIL.COM	RAMESH@101	E-101,VRUNDAVAN8,SG HIGWAY,AHEMEABAD .	CT01	0
2	C102	SHAHID	7878251463	SHAHID@GMAIL.COM	SHAHID@102	D102,VRUNDAVAN9,MANINAGAR,AHEMEABAD .	CT04	3
3	C103	ROY	9512109554	ROY@GMAIL.COM	ROY@103	D103,STAR COMPLEX,KUBERNAGAR	CT05	6
4	C104	SMITH	9521748963	SMITH@GMAIL.COM	SMITH@104	T03,GREEEN CO.,NARSINH CHOWK	CT03	15
5	C105	RAJ	9924642130	RAJ@GMAIL.COM	RAJ@105	E03,GREEEN CO.,MANEK CHOWK	CT01	0

6.FUTURE ENHANCEMENTS OF THE SYSTEM

We will design Front-end Design in HTML CSS , JavaScript and Develop Back-end in Python.

For security purpose New Registration is done using OTP.

We will make database more consistent and We are making this database efficient and easy to implement with huge data capacity.

Methods and user data input will be lot easy after the implement of GUI.
We will also add some extra features so that the users can get answer for their complaints as fast as possible

7. BIBLIOGRAPHY

1 . For the successful implementation of this projectwe referred to many websites and books.

2. The schema was designed by taking ideas from **“SWIGGY”,”ZOMATO”,”UBER”** android applications .

3. We created the ER Diagram and Schema Diagram on **“Creatly.com”**.• Mostly we referred the online material for syntax of procedures, triggers, Exception and cursors.

Reference book:
Data Base System Concepts
-Henry F. Korth & A. Silberschatz 2nd Ed. McGraw-Hill 1991

Reference Websites:

- https://www.stackoverflow.com/
- https://www.w3school.com/
- https://www.tutorialspoint.com/