ONLINE FOOD ORDERING SYSTEM

Team no: 2
Group : COE1
Team Members:

Roll No	Name
101703002	Aadil Sachdev
101703005	Aarshi Arora
101703009	Aastha Mahajan

Submitted to: Ms. Harkiran Kaur

UCS310 Database Management System

INDEX

1. Background of the Area:

The online food ordering system is one of the latest services most fast food restaurants in the western world are adopting. With this method, food is ordered online and delivered to the customer. This is made possible through the use of electronic payment system. Customers pay with their credit cards, although credit card customers can be served even before they make payment either through cash or cheque. So, the system designed in this project will enable customers go online and place order for their food

2. Need of the Project:

Online food ordering knocks out many problems faced by the old traditional call-in-orders. Usually, miscommunication and misunderstanding can lead to order completed incorrectly & that leave the customers dissatisfied. Therefore, customer satisfaction is the key to success but, the repeated mistakes can hamper the profits.

Some of the uses and benefits of an online food ordering system would be:

1) Users can get food easily without putting in much efforts.
2) They have variety of options to choose from.
3) They get the best prices for the food.
4) They can order at their convenience and preferred time.
5) No standing in lines.

3. Objective:

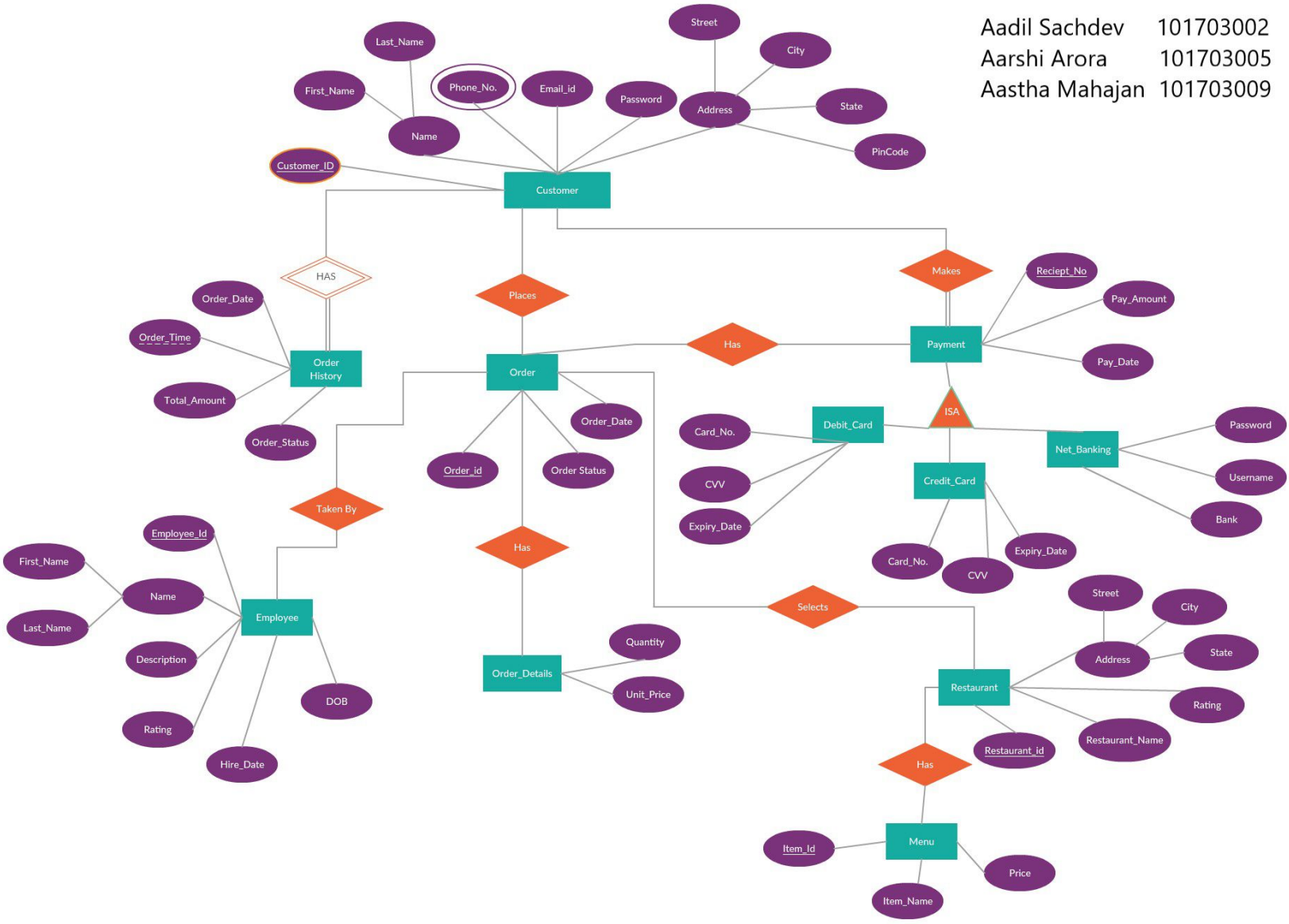
The purpose of Online Food Ordering System is to automate the existing manual system by the help of computerized equipments and full-fledged computer software.

To develop a system that will satisfy the customer service, accommodate huge amount of orders at a time, evaluate its performance and acceptability in terms of security, user-friendliness, accuracy, reliability and improve the communication between the client and the server in order to minimize the time of ordering.

4. Project Outcomes:

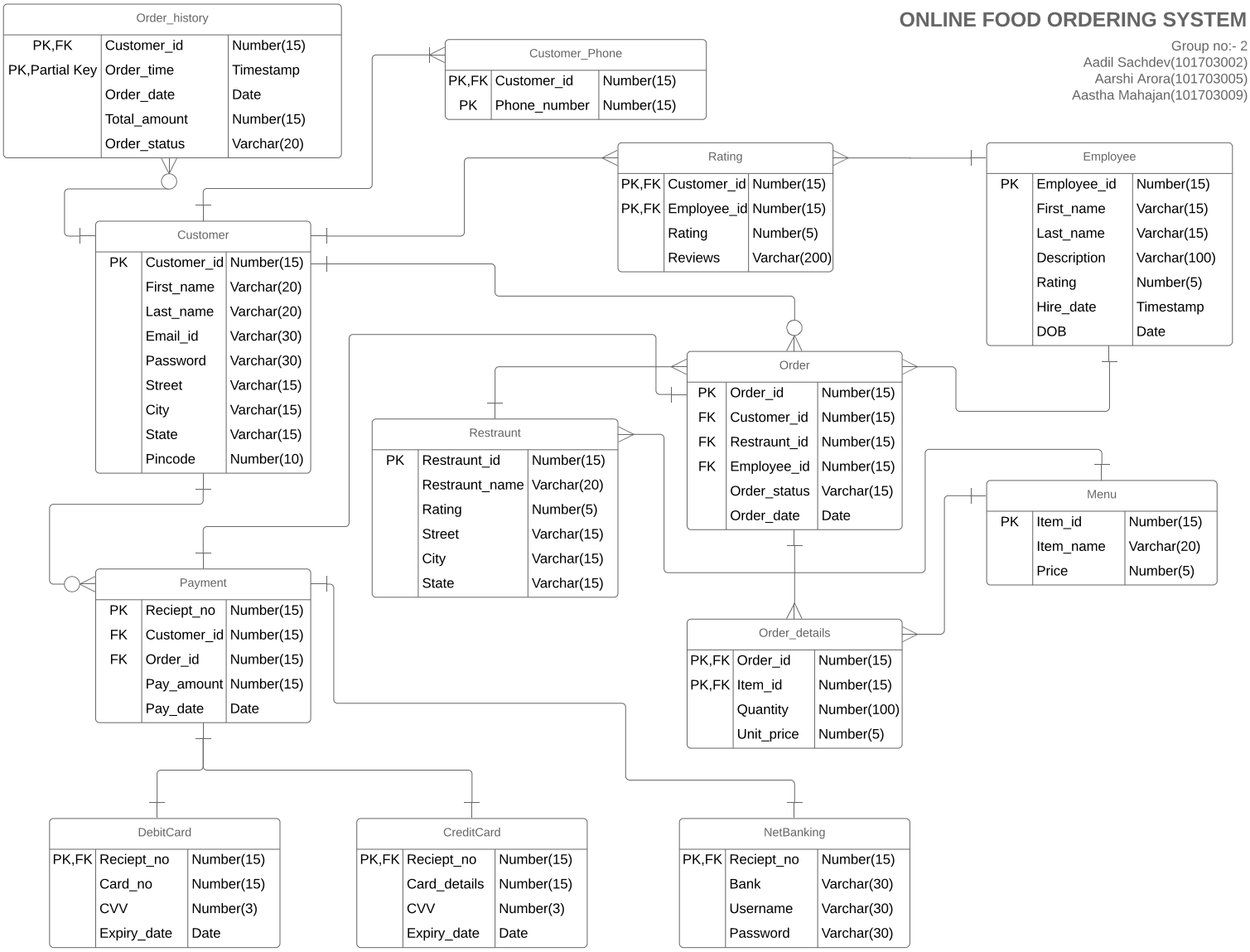
1) Provide searching facilities based on various factors such as category of the food that you want to eat, nearby restaurants, etc.
2) The consumer can have a track where the order has reached and in what time it will be received.
3) Editing, adding and updating of records is improved which results in proper resource management of food data and well as information of customers.
4) Customers get the food delivered at their doorstep within budget, even less than the actual price of the dish in the restaurant.
5) Customers get the information about the quality of the food, the type of the restaurant, the cuisines which it serves, etc. through the food rating given by the customers who previously ordered from the restaurant.

Aadil Sachdev 101703002
Aarshi Arora 101703005
Aastha Mahajan 101703009



ONLINE FOOD ORDERING SYSTEM

Group no:- 2
Aadil Sachdev(101703002)
Aarshi Arora(101703005)
Aastha Mahajan(101703009)



NORMALIZATION:

CUSTOMER			
Attributes	NULL?	Type	Key
Cust_id	no	Number(10)	PK
First_Name	no	Varchar(20)	
Last_Name		Varchar(20)	
Phone no	no	Number(20)	
Email	no	Varchar(20)	
Password	no	Varchar(20)	
Street	no	Varchar(20)	
City	no	Varchar(20)	
State	no	Varchar(20)	

Functional Dependencies:

- Cust_id->First_name
- Cust_id->Email
- Cust_id->Last_name
- Cust_id->Street
- Cust_id->City
- Cust_id->State
- City->State

Normal Forms:

- 1NF: Customer table has multivalued attribute as “PHONE_NO”, so to convert it into 1 N form, it is split as customer and Customer_phoneNo:

CUSTOMER			
Attributes	NULL?	Type	Key
Cust_id	no	Number(10)	PK
First_Name	no	Varchar(20)	
Last_Name		Varchar(20)	
Phone no	no	Number(20)	
Email	no	Varchar(20)	
Street	no	Varchar(20)	
City	no	Varchar(20)	
State	no	Varchar(20)	

CUSTOMER			
Attributes	NULL?	Type	Key

Cust_id	no	Number(10)	PK
Phone no	no	Number(20)	FK,PK

- 2NF: Customer table is already in Second normal form as there is no partial dependency
- 3NF: Transitive dependency exists so the table is split into customer and customer_address

CUSTOMER			
Attributes	NULL?	Type	Key
Cust_id	no	Number(10)	PK
First_Name	no	Varchar(20)	
Last_Name		Varchar(20)	
Phone no	no	Number(20)	
Email	no	Varchar(20)	
Password	no	Varchar(20)	
Street	no	Varchar(20)	

CUSTOMER_ADDRESS			
Attributes	NULL?	Type	Key
Street	no	Varchar(20)	PK
City	no	Varchar(20)	
State	no	Varchar(20)	

DEBIT_CARD			
Attributes	NULL?	Type	Key
Reciept_no	no	Number(10)	PK,FK
Card_no	no	Number(10)	
CVV	no	Number(3)	
Expiry_date	no	Date	

Functional Dependencies:

- Reciept_no->Card_no
- Reciept_no->CVV
- Reciept_no->Expiry_date

Normal Forms:

- 1NF: Debit_Card table is already in first normal form as there are no multivalued attributes.
- 2NF: Debit_Card table is already in second normal form due to lack of partial dependency.
- 3NF: Debit_Card is also in third normal form as there is no transitive dependency.

CREDIT_CARD			
Attributes	NULL?	Type	Key
Reciept_no	no	Number(10)	PK,FK
Card_no	no	Number(10)	
CVV	no	Number(3)	
Expiry_Date	no	Date	

Functional Dependencies:

- Reciept_no->Card_no
- Reciept_no->CVV
- Reciept_no->Expiry_date

Normal Forms:

- 1NF: Credit_Card table is already in first normal form as there are no multivalued attributes.
- 2NF: Credit_Card table is already in second normal form due to lack of partial dependency.
- 3NF: Credit_Card is also in third normal form as there is no transitive dependency.

NET_BANKING			
Attributes	NULL?	Type	Key
Reciept_no	no	Number(10)	PK,FK
Bank	no	Varchar(20)	
Username	no	Varchar(20)	
Password	no	Varchar(20)	

Functional Dependencies:

- Reciept_no->Bank
- Reciept_no->Username
- Reciept_no->Password

ormal Forms:

- 1NF: Net_Banking table is already in first normal form as there are no multivalued attributes.
- 2NF: Net_Banking table is already in second normal form due to lack of partial dependency.
- 3NF: Net_Banking is also in third normal form as there is no transitive dependency.

RESTAURANT			
Attributes	NULL?	Type	Key
Restaurant_id	no	Number(10)	PK
Restaurant_Name	no	Varchar(20)	
Rating		Number(5)	

Street	no	Varchar(20)	
City	no	Varchar(20)	
State	no	Varchar(20)	

Functional Dependencies:

- Restaurant_id->Restaurant_name
- Restaurant_id->Rating
- Restaurant_id->Street
- Restaurant_id->city
- Restaurant_id->state
- Street->City
- City->State

Normal Forms:

- 1NF: Restaurant table is already in first normal form as there are no multivalued attributes.
- 2NF: Restaurant table is already in second normal form due to lack of partial dependency.
- 3NF: Transitive dependency exists and hence the table is split into restaurant1 and Restaurant2.

RESTAURANT1			
Attributes	NULL?	Type	Key
Restaurant_id	no	Number(10)	PK
Restaurant_Name	no	Varchar(20)	
Rating		Number(5)	
Street	no	Varchar(20)	FK

RESTAURANT2			
Attributes	NULL?	Type	Key
Street	no	Varchar(20)	PK
City	no	Varchar(20)	
State	no	Varchar(20)	

MENU			
Attributes	NULL?	Type	Key
Item_id	no	Number(10)	PK
Item_name	no	Number(10)	
Price	no	Number(10)	

Functional Dependencies:

- Item_id->Item_name
- Item_id->Price
- Item_name->price

Normal Forms:

- 1NF: Menu table is already in first normal form as there are no multivalued attributes.
- 2NF: Menu table is already in second normal form due to lack of partial dependency.
- 3NF: Transitive dependency exists and hence the table is split into menu1 and menu2.

MENU1			
Attributes	NULL?	Type	Key
Item_id	no	Number(10)	PK
Item_name	no	Number(10)	FK

MENU2			
Attributes	NULL?	Type	Key
Item_name	no	Number(10)	FK
Item_price	no	Number(10)	

ORDER_DETAILS			
Attributes	NULL?	Type	Key
Order_id	no	Number(10)	PK,FK
Item_id	no	Number(10)	PK,FK
Qty	no	Number(10)	
Price	no	Number	

Functional Dependencies:

- Order_id,Item_id->Item_name
- Order_id,Item_id->Price
- Order_id,Item_id->price

Normal Forms:

- 1NF: Order_details table is already in first normal form as there are no multivalued attributes.
- 2NF: Order_details table is already in second normal form as there is no partial dependency
- 3NF: Transitive dependency does not exists and hence the table is in third normal form.

ORDER			
Attributes	NULL?	Type	Key
Order_id	no	Number(10)	PK
Item_id	no	Number(10)	FK
Restaurant_id	no	Number(10)	FK
Employee_id	no	Number(10)	FK
Order_status	no	Number(10)	
Order_date	no	Number(10)	

Functional Dependencies:

- Order_id->Item_id
- Order_id->Restaurant_id
- Order_id->price
- Order_id->Employee_id
- Order_id->Order_Status
- Order_id->Order_id

Normal Forms:

- 1NF: Order table is already in first normal form as there are no multivalued attributes.
- 2NF: Order table is already in second normal form as there is no partial dependency
- 3NF: Transitive dependency does not exists and hence the table is in third normal form.

PL/SQL Code

```
create sequence cid
        start with 1
        increment by 1
        maxvalue 999
        nocycle
        nocache;
```

```
SQL> create sequence cid
  2  start with 1
  3  increment by 1
  4  maxvalue 999
  5  nocycle
  6  nocache;

Sequence created.
```

```
create or replace procedure create_account (first_name in varchar, last_name in varchar,
email_id in varchar, password in varchar, street in varchar, city in varchar, phone_number1 in
number)
is
begin
insert into customer values(cid.nextval, first_name, last_name, email_id, password, street,
city,phone_number1);
        dbms_output.put_line('Your account has been created successfully.');
        commit;
end;
/
```

```
SQL> create or replace procedure create_account
  2  (first_name in varchar, last_name in varchar, email_id in varchar, password i
n varchar, street in varchar, city in varchar, phone_number1 in number)
  3  is
  4  begin
  5  insert into customer values(cid.nextval, first_name, last_name, email_id, pas
sword, street, city,phone_number1);
  6  dbms_output.put_line('Your account has been created successfully.');
  7  commit;
  8  end;
  9  /

Procedure created.
```

```
create or replace procedure cust_address (city in varchar, state in varchar)
```

is
begin
insert into customer_address values(city, state);
commit;
end;
/

```
SQL> create or replace procedure cust_address (city in varchar, state in varchar)
  2  is
  3  begin
  4  insert into customer_address values(city, state);
  5  commit;
  6  end;
  7  /

Procedure created.
```

create or replace procedure rating1(cust_id in number, employee_id in number, rating in number, reviews in varchar)
is
begin
insert into rating values(cust_id, employee_id, rating, reviews);
commit;
end;
/

```
SQL> create or replace procedure rating1(cust_id in number, employee_id
in number, rating in number, reviews in varchar)
  2  is
  3  begin
  4  insert into rating values(cust_id, employee_id, rating, reviews);
  5  commit;
  6  end;
  7  /

Procedure created.
```

create or replace procedure updateCust(cid in number,email in varchar, pwd in varchar)
 is
 begin
 update customer set email_id=email,password=pwd where cust_id=cid;
 end;
 /

```
SQL> create or replace procedure updateCust(cid in number,email in varchar, pwd in varchar)
  2  is
  3  begin
  4  update customer set email_id=email,password=pwd where cust_id=cid;
  5  end;
  6  /

Procedure created.
```

```
create or replace procedure cust_details(cid in number)
is
cursor c is select * from customer where cust_id=cid;
begin
for rec in c loop
dbms_output.put_line('Customer_id: '||rec.cust_id);
dbms_output.put_line('Customer_name: '||rec.first_name|| rec.last_name);
dbms_output.put_line('Address: '||rec.street||' '|| rec.city);
dbms_output.put_line('Email: '||rec.email_id);
end loop;
end;
/
```

```
SQL> create or replace procedure cust_details(cid in number)
  2  is
  3  cursor c is select * from customer where cust_id=cid;
  4  begin
  5  for rec in c loop
  6  dbms_output.put_line('Customer_id: '||rec.cust_id);
  7  dbms_output.put_line('Customer_name: '||rec.first_name|| rec.last_name);
  8  dbms_output.put_line('Address: '||rec.street||' '|| rec.city);
  9  dbms_output.put_line('Email: '||rec.email_id);
 10  end loop;
 11  end;
 12  /

Procedure created.
```

```
create or replace procedure view_order(cid in number, oid in number)
is
cursor c is select cust_id,first_name from customer where cust_id=cid;
cursor p is select restraunt_name from restaurant1 where restaurant_id=(select restraunt_id from
order1 where cust_id = cid and order_id=oid);
cursor i is select item_name from menu1 where item_id=(select item_id from order1 where
cust_id = cid and order_id=oid);
begin
```

```
for rec in c loop
dbms_output.put_line('Customer_id: '||rec.cust_id);
dbms_output.put_line('Customer_name: '||rec.first_name);
end loop;
for rec2 in p loop
dbms_output.put_line('Restaurnt_name: '||rec2.restraunt_name);
end loop;
for rec3 in i loop
dbms_output.put_line('Item_name: '||rec3.item_name);
end loop;
end;
/
```

```
SQL> create or replace procedure view_order(cid in number, oid in number)
  2  is
  3  cursor c is select cust_id,first_name from customer where cust_id=cid;
  4  cursor p is select restraunt_name from restaurant1 where restaurant_id=(select restraunt_id from order1 where
cust_id = cid and order_id=oid);
  5  cursor i is select item_name from menu1 where item_id=(select item_id from order1 where cust_id = cid and orde
r_id=oid);
  6  begin
  7  for rec in c loop
  8  dbms_output.put_line('Customer_id: '||rec.cust_id);
  9  dbms_output.put_line('Customer_name: '||rec.first_name);
 10  end loop;
 11  for rec2 in p loop
 12  dbms_output.put_line('Restaurnt_name: '||rec2.restraunt_name);
 13  end loop;
 14  for rec3 in i loop
 15  dbms_output.put_line('Item_name: '||rec3.item_name);
 16  end loop;
 17  end;
 18  /

Procedure created.
```

```
create or replace function calcPrice(i_price in number, qty in number) return number
is
tot number;
begin
tot:=i_price*qty;
return(tot);
end;
/
```

```
SQL> create or replace function calcPrice(i_price in number, qty in number) return number
  2  is
  3  tot number;
  4  begin
  5  tot:=i_price*qty;
  6  return(tot);
  7  end;
  8  /

Function created.
```

```
create or replace procedure delete_it(order_id in number)
is
ord number;
begin
dbms_output.put_line('Please enter the order_id that you want to delete');
ord:=&order_id;
delete from order_details where order_id=ord;
end;
```

```
SQL> create or replace procedure delete_it(order_id in number)
  2  is
  3  ord number;
  4  begin
  5  dbms_output.put_line('Please enter the order_id that you want to delete');
  6  ord:=&order_id;
  7  delete from order_details where order_id=ord;
  8  end;
  9  /
Enter value for order_id: 5
old   6: ord:=&order_id;
new   6: ord:=5;

Procedure created.
```

Snapshots

create table Customer_address(city varchar(15) primary key, state varchar(15) not null);

```
SQL> create table Customer_address(city varchar(15) primary key, state varchar(15)
 not null);

Table created.
```

create table Customer(Cust_id number primary key, first_name varchar(20) not null, last_name varchar(20), email_id varchar (30) not null, password varchar(30) not null , street varchar(15) not null,city varchar(15) references Customer_address(city) not null, phone_number_1 number(12) not null);

```
SQL> create table Customer( Cust_id number(10) primary key, first_name varchar(20)
 not null, last_name varchar(20), email_id varchar (30) unique not null, password
varchar(30) not null , street varchar(15) not null,city varchar(15) references Cus
tomer_address(city) not null);

Table created.
```

create table cust_add_phone(cust_id number references customer(cust_id)not null, phone_number number(12), constraint phone_cust_pk primary key(cust_id, phone_number));

```
SQL> create table cust_add_phone(cust_id number references customer(cust_id)not nu
ll, phone_number number(12), constraint phone_cust_pk primary key(cust_id, phone_n
umber));

Table created.
```

create table order_history (cust_id number references customer(cust_id)not null, order_time timestamp not null, order_date date not null, total_amount number not null, order_status varchar(20));

```
SQL> create table order_history (cust_id number references customer(cust_id)not nu
ll, order_time timestamp not null, order_date date not null, total_amount number n
ot null,  order_status varchar(20));

Table created.
```

create table employee (employee_id number(15) primary key, first_name varchar(15) not null, last_name varchar(15), description varchar(100), rating number(5), hire_date date default sysdate, dob date);

```
SQL> create table employee (employee_id number(15) primary key, first_name varchar
(15) not null, last_name varchar(15), description varchar(100), rating number(5),
hire_date date default sysdate, dob date);

Table created.
```

create table rating (cust_id number references customer (cust_id) not null, employee_id number references employee(employee_id) not null, rating number(5) not null, reviews varchar(200));

```
SQL> create table rating (cust_id number references customer (cust_id) not null, e
mployee_id number references employee(employee_id) not null, rating number(5) not
null, reviews varchar(200));

Table created.
```

Create Table Restaurant2(Street Varchar(15) Primary Key, City Varchar(15) NOT NULL, State Varchar(15) NOT NULL);

```
SQL> Create Table Restaurant2( Street Varchar(15) Primary Key, City Varchar(15)
 NOT NULL, State Varchar(15) NOT NULL);

Table created.
```

Create Table Restaurant1(Restaurant_id Number(15) Primary Key, Restraunt_name Varchar(20) NOT NULL, Rating Number(5), Street Varchar(15) References Restaurant2(Street));

```
SQL> Create Table Restaurant1( Restaurant_id Number(15) Primary Key, Restraunt_
name Varchar(20) NOT NULL, Rating Number(5), Street Varchar(15) References Rest
aurant2(Street));

Table created.
```

Create Table Order1(Order_id Number(15) primary key, Cust_id Number(15) References Customer(Cust_id), Restraunt_id Number(15)References Restaurant1(Restaurant_id), Employee_id Number(15) References Employee(Employee_id), Order_status Varchar(15), Order_date Date);

```
SQL> Create Table Order1(
  2  Order_id Number(15) primary key,
  3  Cust_id Number(15)  References Customer(Cust_id),
  4  Restraunt_id Number(15)References Restraunt1(Restraunt_id),
  5  Employee_id Number(15)  References Employee(Employee_id),
  6  Order_status Varchar(15),
  7  Order_date Date);

Table created.
```

Create Table Order_details(Order_id Number(15) References Order1(Order_id), Item_id Number(15) References Menu1(Item_id), Quantity Number NOT NULL, Unit_price Number(5) NOT NULL, Constraint order_pk Primary Key(Order_id,Item_id));

```
SQL> Create Table Order_details(
  2  Order_id Number(15)  References Order1(Order_id),
  3  Item_id Number(15)  References Menu1(Item_id),
  4  Quantity Number NOT NULL,
  5  Unit_price Number(5) NOT NULL,
  6  Constraint order_pk Primary Key(Order_id,Item_id));

Table created.
```

Create Table Menu2(Item_name Varchar(20) Primary Key, Price Number(5) NOT NULL);

```
SQL> Create Table Menu2(
  2  Item_name Varchar(20) Primary Key,
  3  Price Number(5) NOT NULL);

Table created.
```

Create Table Menu1(Item_id Number(15) Primary Key, Item_name Varchar(20) References Menu2(Item_name));

```
SQL> Create Table Menu1(
  2  Item_id Number(15) Primary Key,
  3  Item_name Varchar(20) References Menu2(Item_name));

Table created.
```

create table Payment(Reciept_no number(15) primary key,Cust_id number(15) references customer(cust_id),Order_id number(15) references order1(order_id),Pay_amount number(15) not null,Pay_Date date default sysdate);

```
SQL> create table Payment(Reciept_no number(15) primary key,Cust_id number(15) ref
erences customer(cust_id),Order_id number(15) references order1(order_id),Pay_amou
nt number(15) not null,Pay_Date date default sysdate);

Table created.
```

create table DebitCard(Receipt_no number(15) primary key,Card_no number(15) unique,CVV number(3) not null,Expiry_Date date not null);

```
SQL> create table DebitCard(Receipt_no number(15) primary key,Card_no number(15) u
nique,CVV number(3) not null,Expiry_Date date not null);

Table created.
```

create table CreditCard(Receipt_no number(15) primary key,Card_no number(15) unique,CVV number(3) not null,Expiry_Date date not null);

```
SQL> create table CreditCard(Receipt_no number(15) primary key,Card_no number(15)
unique,CVV number(3) not null,Expiry_Date date not null);

Table created.
```

create table NetBanking(Receipt_no number(15) primary key,Bank varchar(30) not null,Username varchar(30) unique,Password varchar(30) not null);

```
SQL> create table NetBanking(Receipt_no number(15) primary key,Bank varchar(30) no
t null,Username varchar(30) unique,Password varchar(30) not null);

Table created.
```

insert into customer_address values('Patiala', 'Punjab');

```
SQL> insert into customer_address values('Patiala', 'Punjab');

1 row created.
```

insert into customer values(1,'abc','def','abc@gmail.com','abc','22 Baker Street','Patiala',7508118810);

```
SQL> insert into customer values(1,'abc','def','abc@gmail.com','abc','22 Baker Street','Patiala',7508118810);

1 row created.
```

insert into restaurant2 values('Bhupindra Road', 'Patiala', 'Punjab');
insert into restaurant2 values('Sirhind Road', 'Patiala', 'Punjab');
insert into restaurant2 values('Leela Bhawan', 'Patiala', 'Punjab');

```
SQL> insert into restaurant2 values('Bhupindra Road', 'Patiala', 'Punjab');

1 row created.

SQL> insert into restaurant2 values('Sirhind Road', 'Patiala', 'Punjab');

1 row created.

SQL> insert into restaurant2 values('Leela Bhawan', 'Patiala', 'Punjab');

1 row created.
```

insert into restaurant1 values(1,'Chawlas',4.5,'Bhupindra Road');
insert into restaurant1 values(2,'Cafe Yorker',4.2,'Bhupindra Road');
insert into restaurant1 values(3,'Bhupindra Plaza',3.9,'Sirhind Road');
insert into restaurant1 values(4,'Kokos Kitchen',4.1,'Leela Bhawan');
insert into restaurant1 values(5,'HC Burger',3.7,'Leela Bhawan');

```
SQL> insert into restaurant1 values(1,'Chawlas',4.5,'Bhupindra Road');

1 row created.

SQL> insert into restaurant1 values(2,'Cafe Yorker',4.2,'Bhupindra Road');

1 row created.

SQL> insert into restaurant1 values(3,'Bhupindra Plaza',3.9,'Sirhind Road');

1 row created.

SQL> insert into restaurant1 values(4,'Kokos Kitchen',4.1,'Leela Bhawan');

1 row created.

SQL> insert into restaurant1 values(5,'HC Burger',3.7,'Leela Bhawan');

1 row created.
```

insert into menu2 values('Burger Combo',180);
insert into menu2 values('Paneer Crispy Rollo',129);
insert into menu2 values('Chicken Cheese Rollo',139);
insert into menu2 values('Honey Chilli Potato',129);
insert into menu2 values('Veggie Corn Salad',149);
insert into menu2 values('Chicken Salad',169);
insert into menu2 values('Chilli Paneer',179);

insert into menu2 values('Mexican Rice Burrito',99);

```
SQL> insert into menu2 values('Burger Combo',180);

1 row created.

SQL> insert into menu2 values('Paneer Crispy Rollo',129);

1 row created.

SQL> insert into menu2 values('Chicken Cheese Rollo',139);

1 row created.

SQL> insert into menu2 values('Honey Chilli Potato',129);

1 row created.

SQL> insert into menu2 values('Veggie Corn Salad',149);

1 row created.

SQL> insert into menu2 values('Chicken Salad',169);

1 row created.

SQL> insert into menu2 values('Chilli Paneer',179);

1 row created.

SQL> insert into menu2 values('Mexican Rice Burrito',99);

1 row created.
```

insert into menu1 values(1,'Burger Combo');
insert into menu1 values(2,'Paneer Crispy Rollo');
insert into menu1 values(3,'Chicken Cheese Rollo');
insert into menu1 values(4,'Honey Chilli Potato');
insert into menu1 values(5,'Veggie Corn Salad');
insert into menu1 values(6,'Chicken Salad');
insert into menu1 values(7,'Chilli Paneer');
insert into menu1 values(8,'Mexican Rice Burrito');

```
SQL> insert into menu1 values(3,'Chicken Cheese Rollo');

1 row created.

SQL> insert into menu1 values(4,'Honey Chilli Potato');

1 row created.

SQL> insert into menu1 values(5,'Veggie Corn Salad');

1 row created.

SQL> insert into menu1 values(6,'Chicken Salad');

1 row created.

SQL> insert into menu1 values(7,'Chilli Paneer');

1 row created.

SQL> insert into menu1 values(8,'Mexican Rice Burrito');

1 row created.

SQL> insert into menu1 values(2,'Paneer Crispy Rollo');

1 row created.
```

insert into employee values(1,'Rakesh','Singh',NULL,4.2,'25-APR-2019','29-MAY-1990');
insert into employee values(2,'Rahul',NULL,NULL,4.1,'05-MAR-2018','09-OCT-1991');
insert into employee values(3,'Sneha','Sharma',NULL,4.7,'17-JUN-2018','15-DEC-1990');
insert into employee values(4,'Ankur',NULL,NULL,3.8,'15-JAN-2017','27-APR-1992');
insert into employee values(5,'Rhea','Gerewal',NULL,4.2,'02-APR-2019','14-JAN-1993');

```
SQL> insert into employee values(1,'Rakesh','Singh',NULL,4.2,'25-APR-2019','29-MAY-1990');

1 row created.

SQL> insert into employee values(2,'Rahul',NULL,NULL,4.1,'05-MAR-2018','09-OCT-1991');

1 row created.

SQL> insert into employee values(3,'Sneha','Sharma',NULL,4.7,'17-JUN-2018','15-DEC-1990');

1 row created.

SQL> insert into employee values(4,'Ankur',NULL,NULL,3.8,'15-JAN-2017','27-APR-1992');

1 row created.

SQL> insert into employee values(5,'Rhea','Gerewal',NULL,4.2,'02-APR-2019','14-JAN-1993');

1 row created.
```

insert into order1 values(1,1,1,1,'Delivered','05-MAY-2019');
insert into order1 values(2,1,2,3,'Delivered','06-MAY-2019');

```
SQL> insert into order1 values(1,1,1,1,'Delivered','05-MAY-2019');

1 row created.

SQL> insert into order1 values(2,1,2,3,'Delivered','06-MAY-2019');

1 row created.
```

insert into order_details values(1,2,1,129);
insert into order_details values(2,4,1,129);

```
SQL> insert into order_details values(1,2,1,129);

1 row created.

SQL> insert into order_details values(2,4,1,129);

1 row created.
```

Insert into rating values(1,2,4.2,NULL);
Insert into rating values(1,3,4.5,NULL);

```
SQL> Insert into rating values(1,2,4.2,NULL);

1 row created.

SQL> Insert into rating values(1,3,4.5,NULL);

1 row created.
```

References

- Lecture Notes- https://sites.google.com/site/dbms310cse/lecture-notes
- Normalisation- https://www.studytonight.com/dbms/database-normalization.php
- Triggers- https://dev.mysql.com/doc/refman/5.5/en/trigger-syntax.html