**ONLINE FOOD ORDERING SYSTEM**

**Final Project For SQL Module**

**By Anuvhab Basu**

1.Descripation :

Following database schema is designed to function as a backend storage database for an online food ordering system.

An online food ordering system allows your business to accept and manage orders placed online for delivery or takeway.

Customers browse a digital menu, either on an app or website and place and pay for their order online.

* More accuracy and easy order process.
* Fast, easy and comfortable
* 24\*7 customer support
* Mobile friendly.

**This Database Contains 6 Tables:**

1)Customer

2)Menu

3)Order1

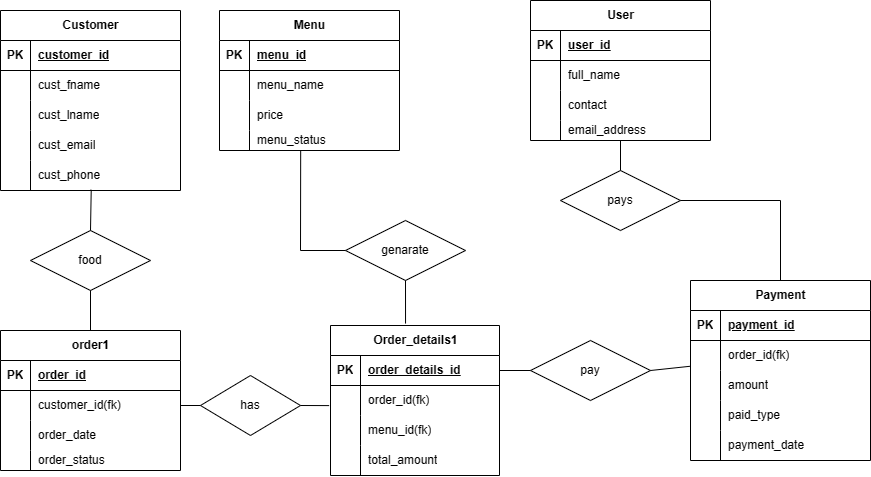
4)User

5)Order\_details1

6)Payment

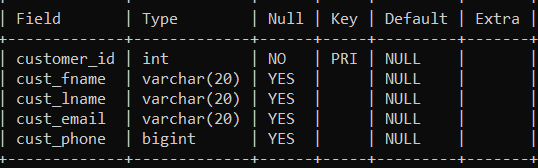
Hows these tables/entities are related to each other is shown pictorially on next page through ER diagram, i.e., Entity Relationship Diagram.

**2.E-R Diagram:**

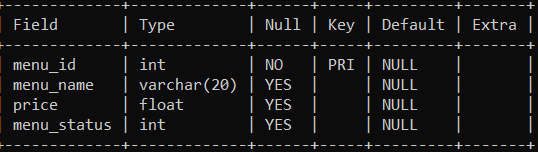
****

**3.Table description:**

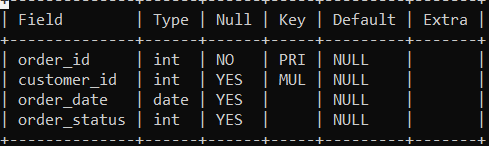
**a)Customer:**

****

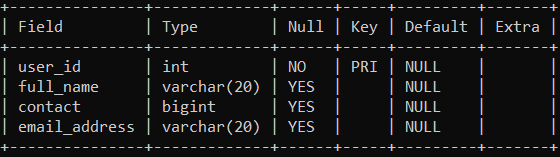
**b)Menu:**

****

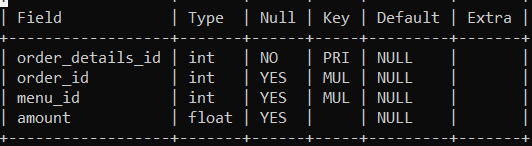
**c)Order1:**



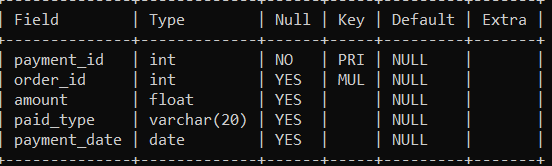
**d)User:**

****

**e) Order\_details1:**



**f) payment:**

****

**4.Commands:**

**a)Create database;**

Create database ofos;

**b)Select database;**

Use ofos;

**c)Create table named Customer:**

create table customer(customer\_id int primary key not null, cust\_fname varchar(20),cust\_lname varchar(20),cust\_email varchar(20),cust\_phone bigint);

**d)Create table named Menu:**

create table menu(menu\_id int primary key not null, menu\_name varchar(20), price float, menu\_status int);

**e)Create table named Order1:**

create table order1(order\_id int primary key not null, customer\_id int, order\_date date, order\_status int, foreign key(customer\_id) references customer(customer\_id));

**f)Create table named User:**

create table user(user\_id int primary key not null, full\_name varchar(20), contact bigint, email\_address varchar(20));

**g)Create table named order\_details1:**

create table order\_details1(order\_details\_id int primary key not null, order\_id int, menu\_id int, amount float, foreign key(order\_id) references order1(order\_id),foreign key(menu\_id) references menu(menu\_id));

**h)Create table named payment:**

create table payment(payment\_id int primary key not null, order\_id int, amount float, paid\_type varchar(20), payment\_date date, foreign key(order\_id)references order1(order\_id));

**i)Populate Customer values:**

insert into customer values:

(1,"divya","gurav","dgurav@gmail.com","9356865016");

(2,"virat","shinde","virats@gmail.com","9987828217");

(3,"sanjay","dhamnak","sadhamnak@gmail.com","9082663617");

(4,"rani", "pawar", "rpawar@gmail.com", "7584584662");

(5,"mahi","solkar","ms@gmail.com","935682452");

(6,"sahil","gurav","sgurav@gmail.com","754854854");

(7,"shrutika","dalvi","dalvis@gmail.com","754854854");

(8,"vidya","suvre","suvrev@gmail.com","9321113673");

(9,"sona","devkad","devkads@gmail.com","9321113673");

(10,"mamata","balye","mamatab@gmail.com","885474545");

**j)Populate Menu values:**

insert into menu values:

(11,"burger",255,30);

(12,"sandwich",130,25);

(13,"french fries",129,15);

(14,"cake",600,10);

(15,"chinese",300,15);

(16,"chicken curry",400,45);

(17,"roti",20,35);

(18,"fried rice",180,60);

(19,"pepsi",60,45);

(20,"pizza",425,385);

**k)Populate Order1 values:**

insert into order1 values;

(101,1,"2023-08-30",45),

(102,5,"2023-07-12",35);

(103,2,"2023-06-25",25),

(104,6,”2023-08-21”,15);

(105,4,"2023-05-15",20);

(106,6,"2023-04-15",10),

(107,8,"2023-07-15",45),

(108,10,"2023-06-23",65),

(109,9,"2023-08-25",25);

(110,2,"2023-05-10",25);

**l)Populate User values:**

insert into user values:

(201,"virat kolhi","935686514","vk@gmail.com");

(202,"mahi dhoni","935686514","msdgmail.com");

(203,"sai gomane","85878545","saig@gmail,com");

(204,"aarti pawar","932111673","aap@gmail.com");

(205,"aashu shitap","56825454","ash@gmail.com");

(206,"shreyash gurav","8598466","sgurav@gmail.com");

(207,"sandhya shitap","8598466","sgurav@gmail.com");

(208,"sai joshi","8598466","saij@gmail.com");

**m)Populate Order\_details1:**

insert into user values:

(1001,101,12,25),

(1002,105,15,300),

(1003,104,13,450),

(1004,110,19,600);

(1005,106,14,750),

(1006,108,18,800),

(1007,101,11,120),

(1008,106,16,650),

(1009,104,17,100),

(10010,102,11,450),

(10011,103,15,200);

**n)Populate Payment:**

insert into payment values:

(201,102,250,"cash","2023-08-30");

(202,104,50,"debit card","2023-08-30");

(203,101,100,"debit card","2023-07-15");

(204,105,50,"credit card","2023-07-15");

(205,106,250,"upi payment","2023-05-20");

(206,102,250,"cash","2023-05-20");

(207,109,175,"credit card","2023-04-15");

(209,105,450,"cash","2023-08-12");

(210,108,600,"debit card","2023-08-12");

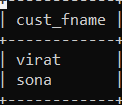
**5.Sub-queries:**

**1.Display names of customer name who have order\_status 25;**

SELECT cust\_fname FROM customer WHERE customer\_id IN

(select customer\_id from order1 where order\_status = 25);

**Result:**

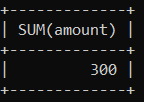
****

**2. Retrieve the total amount paid by the customer with customer id 6 for all their orders:**

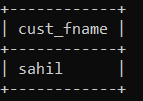
SELECT SUM(amount)

FROM Payment

WHERE order\_id IN (SELECT order\_id FROM Order1 WHERE customer\_id = 6);

****

**3. Retrieve the names of customers who have ordered the most expensive item from the menu.**

****

**6.Join:**

1. **Retrieve the details of orders along with the corresponding customer names for orders that have been paid.**

SELECT o.order\_id, c.cust\_fname

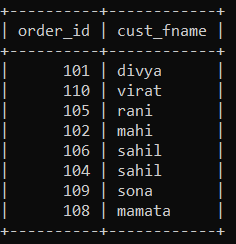
FROM Order1

INNER JOIN Customer ON o.customer\_id = c.customer\_id

WHERE o.order\_id IN (

SELECT p.order\_id

FROM Payment p);



1. **Retrieve the details of orders along with the corresponding menu items and their prices for a specific customer.**

SELECT o.order\_id, m.menu\_name, m.price

FROM Order1 o

JOIN Order\_details1 od ON o.order\_id = od.order\_id

JOIN Menu m ON od.menu\_id = m.menu\_id

WHERE o.customer\_id = 4;

