Restaurant Management System

1. E-R diagram

1

m

1

m

1

1

m

1

m

1

bill

employee

serves

payment

Paid through

costs

orderss

menu

from

Menu\_

items

customer

2.

CREATE TABLE customer`(

`id` INT NOT NULL AUTO\_INCREMENT ,

`name` VARCHAR(30) DEFAULT NULL ,

`email` VARCHAR(30) DEFAULT NULL ,

`contact\_number` INT(15) DEFAULT NULL ,

`feedback` VARCHAR(100) DEFAULT NULL ,

`address` VARCHAR(100) DEFAULT NULL ,

PRIMARY KEY (`id`));

CREATE TABLE orders` (

`order\_id` INT NOT NULL ,

`type` VARCHAR(10) NOT NULL DEFAULT 'dineInn' ,

‘time’ TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP(),

PRIMARY KEY (`order\_id`),

FOREIGN KEY(order\_id) REFERENCES customer(id)

);

CREATE TABLE ‘menu\_items` (

`item\_name` VARCHAR(30) NOT NULL ,

`order\_id` INT(10) NOT NULL ,

`qty` INT(3) NOT NULL DEFAULT '1',

FOREIGN KEY(order\_id) REFERENCES orders(order\_id),

FOREIGN KEY(item\_name) REFERENCES menu(item\_name)

);

CREATE TABLE `bill` (

`bill\_id` INT NOT NULL AUTO\_INCREMENT ,

`order\_id` INT NOT NULL ,

`amount` DOUBLE(5) NOT NULL ,

PRIMARY KEY (`bill\_id`),

FOREIGN KEY(order\_id) REFERENCES orders(order\_id)

);

CREATE TABLE `payment` (

`bill\_id` INT NOT NULL ,

`txn\_id` INT NOT NULL AUTO\_INCREMENT ,

`type` INT NOT NULL ,

`bank\_name` VARCHAR(30) NOT NULL ,

PRIMARY KEY (`txn\_id`),

FOREIGN KEY(bill\_id) REFERENCES bill(bill\_id)

);

CREATE TABLE `employee` (

`emp\_id` INT NOT NULL ,

`name` VARCHAR(30) NOT NULL ,

`email` VARCHAR(30) NOT NULL ,

`phone` INT(15) NOT NULL ,

`address` VARCHAR(50) NOT NULL ,

`designation` VARCHAR(20) NOT NULL

);

CREATE TABLE `serves` (

`emp\_id` INT NOT NULL ,

`order\_id` INT NOT NULL ,

FOREIGN KEY(emp\_id) REFERENCES employee(emp\_id)

FOREIGN KEY(order\_id) REFERENCES orders(order\_id)

);

CREATE TABLE `menu` (

`item\_name` VARCHAR(30) NOT NULL ,

`price` INT(3) NOT NULL

);

3.

insert into customer

values(1,'anupam',’anu@gmail.com ,'12345667',’good food’,jabalpur),

insert into menu

values(‘chapati’,67),

insert into employee

values(1,harshal',’harshal@123’,12345678','iiitb','waiter');

5.

i)

**create view vw\_CustomerSnapshot as**

**select max(emails) as visits,tbl1.email,name,contact\_number**

**from**

**(select count(email) as emails,**

**email from customer group by email) as tbl1,**

**customer**

**where tbl1.email=customer.email.**

***Note:- the above query counts the no. of visits by a customer by the email id provided by the customer in every visit, ‘id’ field in customer is different for each visit, so email is used to count the visits by each customer.***

ii)

**create view vw\_OrderSnapshot as**

**select max(counts) as max\_ordered,tbl1.item\_name,price from**

**(**

**Select count(item\_name) as counts, item\_name from menu\_item group by item\_name**

**) as tbl1,**

**menu where tbl1.item\_name=menu.item\_name;**

6.

**delimiter $$**

**create function fn\_getOrderTimeElapsed(CustomerId)**

**returns timestamp**

**begin**

**declare time\_elapsed timestamp;**

**declare order\_time timestamp;**

**select time into order\_time from orders where order\_id = CustomerId;**

**set time\_elapsed = temp – now();**

**return time\_elapsed;**

**end**

**$$**

**delimiter** ;

7.

**delimiter $$**

**create sp\_Procedure GetOrder(IN CustomerId)**

**begin**

**select o.order\_id, mt.item\_name,mt.qty from orders as o,menu\_items as mt where o.order\_id = CustomerId and mt.order\_id=o.order\_id;**

**end**

**$$**

**delimiter ;**

8.

**delimiter $$**

**create procedure sp\_GenerateBill(IN OrderId, out BillAMT)**

**begin**

**declare temp float;**

**select sum(price\*qty) into temp from menu\_items,menu where order\_id = OrderId and menu\_items.item\_name=menu.item\_name;**

**set BillAMT = temp;**

**end**

**$$**

**delimitter ;**