**DATABASE MANAGEMENT SYSTEM PACKAGE**

**TOPIC : ONLINE FOOD ORDERING SYSTEM OF A RESTAURANT**

**DONE BY :**

1. Bharathi A - 18PD05
2. Shreenidhi N - 18PD33

CODE USING :

**MySQL**

**HTML, CSS, JAVASCRIPT, PHP**

APPLICATIONS USED :

**XAMPP, VISUAL STUDIO CODE(OR NOTEPAD++)**

**CODE :**

**CREATION ON DATABASE :**

CREATE DATABASE RESTAURANT;

**CREATION OF TABLES :**

**1 Customer :**

CREATE TABLE `customer` (

`CUSTOMER\_ID` int(11) NOT NULL AUTO\_INCREMENT,

`Customer\_name` varchar(20) NOT NULL,

`Address` varchar(30) NOT NULL,

`City` varchar(20) NOT NULL,

`Postcode` int(11) NOT NULL,

`Telephone` int(11) NOT NULL,

`Email\_id` varchar(30) NOT NULL,

`Password` varchar(20) NOT NULL,

PRIMARY KEY (`CUSTOMER\_ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb4

**2 Menu :**

CREATE TABLE `menu` (

`Product\_id` int(11) NOT NULL,

`Product\_name` varchar(20) NOT NULL,

`Category` varchar(10) NOT NULL,

`Unit\_Price` double NOT NULL,

`Number\_items` int(11) NOT NULL,

PRIMARY KEY (`Product\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4

**3 Restaurant\_Employees :**

CREATE TABLE `restaurant\_employees` (

`Employee\_id` int(11) NOT NULL,

`Employee\_name` varchar(20) NOT NULL,

`Password` varchar(20) NOT NULL,

`Address` varchar(30) NOT NULL,

`Postcode` int(11) NOT NULL,

`Telephone` int(11) NOT NULL,

PRIMARY KEY (`Employee\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4

4 Cart :

**5 Orders :**

CREATE TABLE `orders` (

`Order\_id` int(11) NOT NULL AUTO\_INCREMENT,

`Customer\_id` int(11) NOT NULL,

`Order\_date` timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

`Expected\_delivery` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',

`Order\_Status` varchar(20) DEFAULT NULL,

PRIMARY KEY (`Order\_id`),

KEY `orders\_ibfk\_1` (`Customer\_id`),

CONSTRAINT `orders\_ibfk\_1` FOREIGN KEY (`Customer\_id`) REFERENCES `customer` (`CUSTOMER\_ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=33 DEFAULT CHARSET=utf8mb4

**6 Bill :**

**(adding bill id also code at the end)**

CREATE table Bill(

Order\_id int not null,

customer\_id int not null,

Total\_Amount double,

foreign key(Order\_id) REFERENCES orders(Order\_id),

FOREIGN key(customer\_id) REFERENCES customer(Customer\_id)

);

**7 Reviews :**

CREATE TABLE `reviews` (

‘Review\_id’ int(11) PRIMARY KEY AUTO\_INCREMENT,

`Customer\_id` int(11) NOT NULL,

`name` varchar(20) DEFAULT NULL,

`Order\_id` int(11) NOT NULL,

`Feedback` varchar(50) DEFAULT NULL,

`Ratings(out of 5)` int(11) NOT NULL,

KEY `reviews\_ibfk\_1` (`Customer\_id`),

KEY `reviews\_ibfk\_2` (`Order\_id`),

CONSTRAINT `reviews\_ibfk\_1` FOREIGN KEY (`Customer\_id`) REFERENCES `customer` (`CUSTOMER\_ID`),

CONSTRAINT `reviews\_ibfk\_2` FOREIGN KEY (`Order\_id`) REFERENCES `orders` (`order\_ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4

**8 Items\_sold :**

CREATE TABLE Items\_Sold(

Sold\_id INT PRIMARY KEY AUTO\_INCREMENT,

Customer\_id INT NOT NULL,

Order\_id INT NOT NULL,

Product\_id INT NOT NULL,

Product\_name VARCHAR(20),

Order\_date TIMESTAMP,

FOREIGN KEY(Customer\_id) REFERENCES Customer(Customer\_id),

FOREIGN KEY(Order\_id) REFERENCES Orders(Order\_id),

FOREIGN KEY(Product\_id) REFERENCES Menu(Product\_id)

);

**CREATION OF PROCEDURES :**

**1 logincheck**

DELIMITER //

PROCEDURE logincheck(IN NAME VARCHAR(20),IN PWD VARCHAR(20))

BEGIN

select \* from customer where customer\_name=name AND PASSWORD= pwd;

END //

DELIMITER ;

**2 getPid**

(for returning pid n sending it to orders table)

DELIMITER //

create procedure getPid(in foodname varchar(20))

BEGIN

select Product\_id from menu where product\_name=foodname;

end //

DELIMITER;

**3 intoCart**

DELIMITER //

create PROCEDURE intoCart(in custId int,in pid int,in cost double,in qty int)

BEGIN

insert into cart(Customer\_id,Product\_id,Unit\_Price,Quantity) values(custId,pid,cost,qty);

END //

DELIMITER ;

**4 enterFeedback**

(change apt columns in reviews table 5columns)

DELIMITER //

CREATE PROCEDURE enterFeedback(IN custid int,IN NAME VARCHAR(20),in orderid int,IN FEEDBACK VARCHAR(50),IN RATE INT)

BEGIN

INSERT INTO reviews VALUES (custid,NAME,orderid,FEEDBACK,RATE);

END //

DELIMITER ;

**5 enterEmploy**

DELIMITER //

CREATE PROCEDURE enterEmploy(in empid int,in empname varchar(20),in pwd varchar(20),in addr varchar(30),in postcode int,in tel int)

BEGIN

insert into restaurant\_employees values(empid,empname,pwd,addr,postcode,tel);

END //

DELIMITER ;

**6 getCname**

DELIMITER //

create PROCEDURE getCname(in id int)

BEGIN

SELECT Customer\_name from customer where customer\_id=id;

end //

DELIMITER ;

**7 addToOrders**

DELIMITER //

CREATE PROCEDURE addToOrders(IN Cust\_id INT,IN O\_date TIMESTAMP,IN Del\_date TIMESTAMP,IN stat VARCHAR(20))

BEGIN

INSERT INTO ORDERS(Customer\_id,Order\_date,Expected\_delivery,Order\_Status) VALUES(Cust\_id,O\_date,Del\_date,stat);

END //

DELIMITER ;

**8 getOrderID**

DELIMITER //

CREATE PROCEDURE getOrderID(IN Cust\_id INT,IN O\_date TIMESTAMP)

BEGIN

SELECT Order\_id FROM ORDERS WHERE Customer\_id = Cust\_id AND Order\_date =O\_date;

END //

DELIMITER ;

**9 insertIntoBill**

DELIMITER //

CREATE PROCEDURE insertIntoBill(IN Ord\_id INT,IN Cust\_id INT,IN total DOUBLE)

BEGIN

INSERT INTO Bill(Order\_id,Customer\_id,Total\_Amount) VALUES(Ord\_id,Cust\_id,total);

END //

DELIMITER ;

**10 Sign\_Up**

DELIMITER //

create procedure Sign\_up(IN cust\_name varchar(20),in address varchar(30),in city varchar(20),in postcode int,in telephone int,in email varchar(30),in password varchar(20))

BEGIN

insert into customer(Customer\_name,Address,City,Postcode,Telephone,Email\_Id,Password) values(cust\_name,address,city,postcode,telephone,email,password);

End //

DELIMITER ;

**11 addToItemsSold**

DELIMITER //

Create procedure addToItemsSold(IN Cust\_id int, in Ord\_id int, in Prod\_id int, in Prod\_name varchar(20),in Ord\_date timestamp)

BEGIN

INSERT INTO Items\_Sold(Customer\_id,Order\_id,Product\_id,Product\_name,Order\_date)VALUES(Cust\_id,Ord\_id,Prod\_id,Prod\_name,Ord\_date);

END //

DELIMITER ;

BEGIN

**12 removeFromCart**

Delimiter //

Create procedure removeFromCart(in pro\_id int)

BEGIN

delete from cart where Product\_id=pro\_id;

End //

Delimiter ;

**13 INSERTMENU**

DELIMITER //

CREATE PROCEDURE INSERTMENU(IN PID INT, IN PNAME VARCHAR(20), IN CAT VARCHAR(10), IN Unit\_Price INT, IN QTY INT)

BEGIN

INSERT INTO menu VALUES(PID,PNAME,CAT,Unit\_Price,QTY);

END //

DELIMITER ;

**14 addToOrders**

Delimiter //

Create procedure addToOrders(in Cust\_id int, in O\_date timestamp, in Del\_date, in stat varchar(20))

BEGIN

INSERT INTO ORDERS(Customer\_id,Order\_date,Expected\_delivery,Order\_Status) VALUES(Cust\_id,O\_date,Del\_date,stat);

END //

Delimiter ;

**15 intoCart**

Delimiter //

Create procedure intoCart(in custId int,in pid int,in p\_name varchar(20),in cost double,in qty int)

BEGIN

insert into cart(Customer\_id,Product\_id,Product\_name,Unit\_Price,Quantity) values(custId,pid,p\_name,cost,qty);

End //

Delimiter ;

**16 createGraph**

Delimiter //

CREATE PROCEDURE createGraph()

BEGIN

SELECT Product\_name,COUNT(\*) FROM Items\_Sold GROUP BY Product\_name;

END //

DELIMITER ;

**17 billsGraph**

DELIMITER //

CREATE procedure billsGraph()

BEGIN

SELECT \* FROM Bill;

END //

DELIMITER ;

**CREATE FUNCTIONS :**

**1 calculateTotal**

DELIMITER //

CREATE FUNCTION calculateTotal(PRO\_ID INT) RETURNS DOUBLE DETERMINISTIC

BEGIN

DECLARE DONE INT DEFAULT FALSE;

DECLARE TOTAL\_PRICE DOUBLE;

DECLARE U\_PRICE DOUBLE;

DECLARE QUANT INT;

DECLARE CURPRICE CURSOR FOR SELECT UNIT\_PRICE,QUANTITY FROM CART WHERE PRODUCT\_ID = PRO\_ID;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN CURPRICE;

FETCH CURPRICE INTO U\_PRICE,QUANT;

SET TOTAL\_PRICE = U\_PRICE\*QUANT;

CLOSE CURPRICE;

RETURN TOTAL\_PRICE;

END; //

DELIMITER ;\*/

**OTHER CODES :**

SET SQL\_MODE='ALLOW\_INVALID\_DATES';

**INSERTION OF RECORDS**

call insertmenu(1,'Pizza Margherita','Pizza',119,100);

call insertmenu(3,'Double Cheese','Pizza',200,100);

call insertmenu(4,'Farm House','Pizza',150,100);

call insertmenu(5,'Peppy Paneer','Pizza',140,100);

call insertmenu(6,'Mexican Green Wave','Pizza',120,100);

call insertmenu(7,'Paneer Makhani','Pizza',120,100);

call insertmenu(8,'Deluxe Veggie','Pizza',120,100);

call insertmenu(9,'Veg Extravaganza','Pizza',120,100);

call insertmenu(11,'Cheese Garlic','Pizza',200,100);

call insertmenu(12,'Brocolli Pizza','Pizza',60,100);

call insertmenu(13,'Veg Paradise','Pizza',200,100);

call insertmenu(14,'Indi Tandoori','Pizza',60,100);

call insertmenu(2,'Capellini Pasta','Pasta',120,100);

call insertmenu(10,'Fusilli Pasta','Pasta',120,100);

call insertmenu(15,'Macaroni','Pasta',120,100);

call insertmenu(16,'Italian Pasta','Pasta',200,100);

call insertmenu(17,'Penne Pasta','Pasta',120,100);

call insertmenu(18,'Ziti Pasta','Pasta',60,100);

**CUSTOMER TABLE : ( ADDING AUTO INCREMENT CONSTRAINT):**

ALTER TABLE orders DROP FOREIGN KEY orders\_ibfk\_1;

ALTER TABLE reviews DROP FOREIGN KEY reviews\_ibfk\_1;

ALTER TABLE cart DROP FOREIGN KEY cart\_ibfk\_1;

ALTER TABLE bill DROP FOREIGN KEY bill\_ibfk\_2;

ALTER TABLE CUSTOMER MODIFY CUSTOMER\_ID INT AUTO\_INCREMENT;

ALTER TABLE ORDERS ADD CONSTRAINT orders\_ibfk\_1 FOREIGN KEY(CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID);

ALTER TABLE REVIEWS ADD CONSTRAINT reviews\_ibfk\_1 FOREIGN KEY(CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID);

ALTER TABLE CART ADD CONSTRAINT cart\_ibfk\_1 FOREIGN KEY(CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID);

ALTER TABLE BILL ADD CONSTRAINT bill\_ibfk\_2 FOREIGN KEY(CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID);

ALTER TABLE REVIEWS DROP FOREIGN KEY reviews\_ibfk\_2;

ALTER TABLE BILL DROP FOREIGN KEY bill\_ibfk\_2;

ALTER TABLE orders MODIFY order\_ID INT AUTO\_INCREMENT;

**MODIFICATION OF ORDERS ORDER\_ID :**

SET SQL\_MODE = 'ALLOW\_INVALID\_DATES';

ALTER TABLE REVIEWS DROP FOREIGN KEY reviews\_ibfk\_2;

ALTER TABLE ORDERS MODIFY Order\_id INT AUTO\_INCREMENT;

ALTER TABLE BILL ADD CONSTRAINT bill\_ibfk\_1 FOREIGN KEY(Order\_id) REFERENCES ORDERS(Order\_id);

ALTER TABLE REVIEWS ADD CONSTRAINT reviews\_ibfk\_2 FOREIGN KEY(Order\_id) REFERENCES ORDERS(Order\_id);

**CHANGE**

ALTER TABLE BILL ADD CONSTRAINT bill\_ibfk\_1 FOREIGN KEY(Order\_id) REFERENCES ORDERS(Order\_id);

ALTER TABLE REVIEWS ADD CONSTRAINT reviews\_ibfk\_2 FOREIGN KEY(Order\_id) REFERENCES ORDERS(Order\_id);

ALTER TABLE BILL ADD COLUMN Bill\_id INT AUTO\_INCREMENT PRIMARY KEY FIRST;

**REVIEW TABLE ( AUTO INCREMENT CONSTRAINT )**

ALTER TABLE Reviews ADD COLUMN Review\_id INT PRIMARY KEY AUTO\_INCREMENT FIRST;