A Report

On the work done during 3rd semester subject

Database Management Systems

Of

B.Tech. Computer Engineering

Restaurant Management System

Prepared by: Kuldeep Gabani 22CEUOS098 CE031 Computer Engineering, Dharmsinh Desai University	Prepared by: Yash Gabani 22CEUOS137 CE032 Computer Engineering, Dharmsinh Desai University

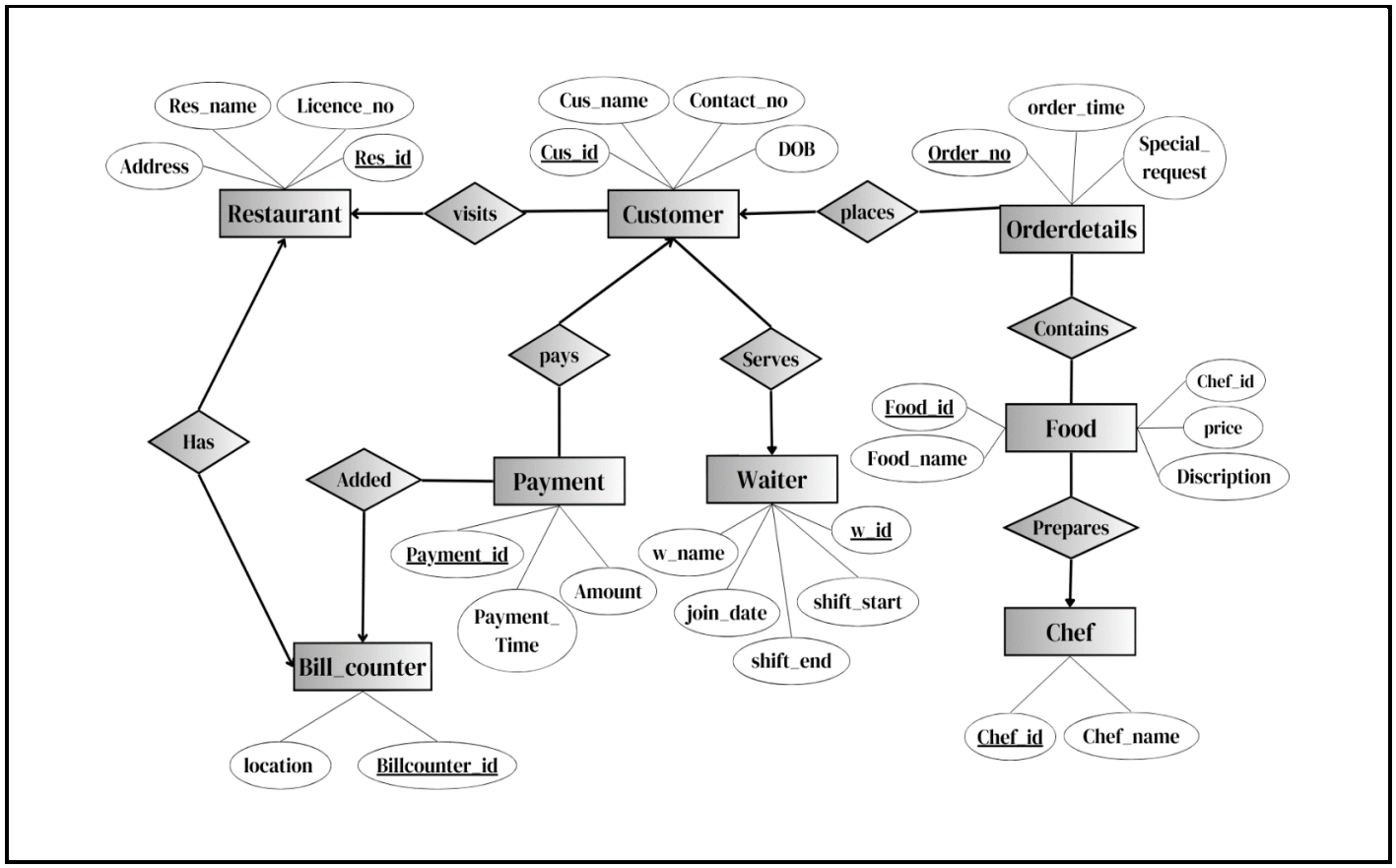
Dharmsinh Desai University
October 2023

Table of Content

Brief Information about the Restaurant Management System

- The Restaurant Management System is for efficiently handling the ordering process for various customers.
- This Database system simplifies the ordering process, making it easier for the restaurants to manage services towards customers and listing the employee/waiters' service towards their customers and also looking for less efforts at Billing Counter.
- It offers user-friendly interface for managing entire process throughout the stay of customer in their respective Restaurant including Payment process and billing the items of respective customers

ER Diagram


Res_name
Licence_no
Address
Res_id
Restaurant
visits
Cus_name
Contact_no
Cus_id
DOB
Customer
places
order_time
Order_no
Special_ request
Orderdetails
Contains
pays
Serves
Has
Added
Payment
Waiter
Food_id
Food_name
Food
Chef_id
price
Discription
Prepares
Payment_id
Amount
Payment_ Time
w_name
w_id
join_date
shift_start
shift_end
Chef
Bill_counter
location
Billcounter_id
Chef_id
Chef_name

Schema

➢ Restaurant:

Columns Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	RES_ID	NUMBER	22		0	No		
2	LICENCE_NO	NUMBER	22			Yes		
3	RES_NAME	VARCHAR2	255			Yes	Byte	
4	ADDRESS	VARCHAR2	255			Yes	Byte	

Constraints

Constraint	Type	Condition	Related Constraint	Related Table	Constraint Columns	On Delete	Status	Last Change	Invalid?
SYS_C00133859299	Primary Key	-	-	-	RES_ID	-	ENABLED	8 minutes ago	-

Related Constraints

Table	Constraint	Type	Column
BILLCOUNTER	SYS_C00133859359	Foreign Key	RES_ID
ORDERDETAILS	SYS_C00133859426	Foreign Key	RES_ID

➢ Customer:

Columns Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	CUS_ID	NUMBER	22		0	No		
2	CUS_NAME	VARCHAR2	60			Yes	Byte	
3	DOB	DATE	7			Yes		
4	CONTACT_NO	VARCHAR2	15			Yes	Byte	

Constraints

Constraint	Type	Condition	Related Constraint	Related Table	Constraint Columns	On Delete	Status	Last Change	Invalid?
SYS_C00133859300	Primary Key	-	-	-	CUS_ID	-	ENABLED	5 minutes ago	-

Related Constraints

Table	Constraint	Type	Column
ORDERDETAILS	SYS_C00133859425	Foreign Key	CUS_ID
PAYMENT	SYS_C00133859356	Foreign Key	CUS_ID

➢ Orderdetails:

Columns

Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	ORDER_NO	NUMBER	22		0	No		
2	ORDER_TIME	VARCHAR2	25			Yes	Byte	
3	SPECIAL_REQUEST	VARCHAR2	100			Yes	Byte	
4	CUS_ID	NUMBER	22		0	Yes		
5	RES_ID	NUMBER	22		0	Yes		
6	WAITER_ID	NUMBER	22		0	Yes		

Constraints

Constraint	Type	Condition	Related Constraint	Related Table	Constraint Columns	On Delete	Status	Last Change	Invalid?
SYS_C00133859425	Foreign Key	-	SYS_C00133859300	CUSTOMER	CUS_ID	NO ACTION	ENABLED	4 minutes ago	-
SYS_C00133859426	Foreign Key	-	SYS_C00133859299	RESTAURANT	RES_ID	NO ACTION	ENABLED	4 minutes ago	-
SYS_C00133859427	Foreign Key	-	SYS_C00133859354	WAITER	WAITER_ID	NO ACTION	ENABLED	4 minutes ago	-
SYS_C00133859424	Primary Key	-	-	-	ORDER_NO	-	ENABLED	4 minutes ago	-

Related Constraints

Table	Constraint	Type	Column
ORDERFOOD	SYS_C00133859441	Foreign Key	ORDER_NO

➢ Food:

Columns

Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	FOOD_ID	NUMBER	22		0	No		
2	FOOD_NAME	VARCHAR2	255			Yes	Byte	
3	PRICE	NUMBER	22	10	2	Yes		
4	DESCRIPTION	VARCHAR2	255			Yes	Byte	
5	CHEF_ID	NUMBER	22		0	Yes		

Constraints

Constraint	Type	Condition	Related Constraint	Related Table	Constraint Columns	On Delete	Status	Last Change	Invalid?
SYS_C00133859305	Primary Key	-	-	-	FOOD_ID	-	ENABLED	6 minutes ago	-

Related Constraints

Table	Constraint	Type	Column
ORDERFOOD	SYS_C00133859442	Foreign Key	FOOD_ID

➢ Chef:

Columns

Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	CHEF_ID	NUMBER	22		0	No		
2	CHEF_NAME	VARCHAR2	255			Yes	Byte	

Constraints

Constraint	Type	Condition	Related Constraint	Related Table	Constraint Columns	On Delete	Status	Last Change	Invalid?
SYS_C00133859330	Primary Key	-	-	-	CHEF_ID	-	ENABLED	5 minutes ago	-

Related Constraints

No related constraints defined.

➢ Waiter:

Columns

Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	WAITER_ID	NUMBER	22		0	No		
2	WAITER_NAME	VARCHAR2	255			Yes	Byte	
3	JOIN_DATE	DATE	7			Yes		
4	SHIFT_START	TIMESTAMP(6)	11		6	Yes		
5	SHIFT_END	TIMESTAMP(6)	11		6	Yes		

Constraints

Constraint	Type	Condition	Related Constraint	Related Table	Constraint Columns	On Delete	Status	Last Change	Invalid?
SYS_C00133859354	Primary Key	-	-	-	WAITER_ID	-	ENABLED	8 minutes ago	-

Related Constraints

Table	Constraint	Type	Column
ORDERDETAILS	SYS_C00133859427	Foreign Key	WAITER_ID

➢ Billcounter:

Columns

Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	BILLCOUNTER_ID	NUMBER	22		0	No		
2	LOCATION	VARCHAR2	255			Yes	Byte	
3	RES_ID	NUMBER	22		0	Yes		

Constraints

Constraint	Type	Condition	Related Constraint	Related Table	Constraint Columns	On Delete	Status	Last Change	Invalid?
SYS_C00133859359	Foreign Key	-	SYS_C00133859299	RESTAURANT	RES_ID	NO ACTION	ENABLED	71 seconds ago	-
SYS_C00133859358	Primary Key	-	-	-	BILLCOUNTER_ID	-	ENABLED	71 seconds ago	-

Related Constraints

No related constraints defined.

➢ Payment:

Columns

Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	PAYMENT_ID	NUMBER	22		0	No		
2	AMOUNT	NUMBER	22	10	2	Yes		
3	PAYMENT_TIME	VARCHAR2	25			Yes	Byte	
4	CUS_ID	NUMBER	22		0	Yes		

Constraints

Constraint	Type	Condition	Related Constraint	Related Table	Constraint Columns	On Delete	Status	Last Change	Invalid?
SYS_C00133859356	Foreign Key	-	SYS_C00133859300	CUSTOMER	CUS_ID	NO ACTION	ENABLED	7 minutes ago	-
SYS_C00133859355	Primary Key	-	-	-	PAYMENT_ID	-	ENABLED	7 minutes ago	-

➢ Orderfood:

Columns

Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	ORDER_NO	NUMBER	22		0	No		
2	FOOD_ID	NUMBER	22		0	No		

Constraints

Constraint	Type	Condition	Related Constraint	Related Table	Constraint Columns	On Delete	Status	Last Change	Invalid?
SYS_C00133859441	Foreign Key	-	SYS_C00133859424	ORDERDETAILS	ORDER_NO	NO ACTION	ENABLED	5 minutes ago	-
SYS_C00133859442	Foreign Key	-	SYS_C00133859305	FOOD	FOOD_ID	NO ACTION	ENABLED	5 minutes ago	-
SYS_C00133859440	Primary Key	-	-	-	ORDER_NO, FOOD_ID	-	ENABLED	5 minutes ago	-

Related Constraints

No related constraints defined.

Normalization comments (optional)

- To bring the tables to BCNF/Higher level form, we need to ensure that for every non-trivial functional dependency, the left-hand side (LHS) of the dependency is a superkey (a candidate key). BCNF is a higher level of normalization than 3NF, and it addresses situations where there are multiple candidate keys.
- Here's how we can achieve Higher Normal Form for the tables in our system:

1) Restaurant (res_id, licence_no, res_name, address):
   - The candidate key for this table is {res_id}.
   - There is no non-trivial dependencies or transitive dependencies, hence it is in BCNF.

2) Food Table (food_id, food_name, price, description, chef_id):
   - The candidate key for this table is {food_id}.
   - There is no non-trivial dependencies or transitive dependencies, hence it is in BCNF.

3) Chef Table (chef_id, chef_name):
   - The candidate key for this table is {chef_id}.
   - There is no non-trivial dependencies or transitive dependencies, hence it is in BCNF.

4) Waiter Table (waiter_id, waiter_name, join_date, shift_start, shift_end):
   - The candidate key for this table is {waiter_id}.
   - There is no non-trivial dependencies or transitive dependencies, hence it is in BCNF.

5) Payment Table (payment_id, amount, payment_time, cus_id):

- The candidate key for this table is {payment_id}.
- There is no non-trivial dependencies or transitive dependencies, hence it is in BCNF.

6) Billcounter Table (billcounter_id, location, res_id):

- The candidate key for this table is {billcounter_id}.
- There is no non-trivial dependencies or transitive dependencies, hence it is in BCNF.

7) Customer Table (cus_id, cus_name, dob, contact_no):

- The candidate key for this table is {cus_id}.
- Though, there is chance that this table might get into less than BCNF form because of non-trivial functional dependency: {contact_no} -> {cus_id, cus_name, dob}. The LHS (contact_no) is not a superkey.
- To bring it to BCNF, we can create a new table with contact_no as the primary key and the other attributes: (cus_id, cus_name, dob). This new table will have contact_no as a superkey and will not have partial dependencies. The original table would reference the new one through the contact_no attribute.
- Otherwise, it will be fine we don't do this as cus_id is primary key which can derive all the remaining attributes.

8) Orderdetails (order_no, order_time, special_request, cus_id, res_id, waiter_id):

- The candidate key for this table is {order_no}.

- One customer can have multiple orders. But he/she has multiple order_no.(eg., first order-timpepass, second order-pasta : Here for both orders, order numbers will be different)
- There is no non-trivial dependencies or transitive dependencies, hence it is in BCNF.

❖ Now all tables are in BCNF form as they do not have any non-trivial dependencies violating BCNF criteria. In other words, all tables in this system have atomic values, indicating that it is already in the normalized form.

CRUD

❖ CREATE:

```
CREATE TABLE Restaurant (
    res_id INT PRIMARY KEY,
    licence_no NUMBER,
    res_name VARCHAR(255),
    address VARCHAR(255)
);
CREATE TABLE Customer (
    cus_id INT PRIMARY KEY,
    cus_name VARCHAR(60),
    dob DATE,
    contact_no VARCHAR(15)
);

CREATE TABLE Food (
    food_id INT PRIMARY KEY,
    food_name VARCHAR(255),
    price DECIMAL(10, 2),
    description VARCHAR(255),
    chef_id INT
);
CREATE TABLE Chef (
    chef_id INT PRIMARY KEY,
    chef_name VARCHAR(255)
);
CREATE TABLE Waiter (
    waiter_id INT PRIMARY KEY,
    waiter_name VARCHAR(255) ,
    join_date DATE,
    shift_start TIMESTAMP,
    shift_end TIMESTAMP
);
CREATE TABLE Payment (
    payment_id INT PRIMARY KEY,
    amount DECIMAL(10, 2),
    payment_time VARCHAR(25),
    cus_id INT,
    FOREIGN KEY (cus_id) REFERENCES Customer(cus_id)
);
CREATE TABLE Billcounter (
```

```
    billcounter_id INT PRIMARY KEY,
    location VARCHAR(255),
    res_id INT,
    FOREIGN KEY (res_id) REFERENCES Restaurant(res_id)
);
CREATE TABLE Orderdetails (
    order_no INT PRIMARY KEY,
    order_time VARCHAR(25),
    special_request VARCHAR(100),
    cus_id INT,
    res_id INT,
    waiter_id INT,
    FOREIGN KEY (cus_id) REFERENCES Customer(cus_id),
    FOREIGN KEY (res_id) REFERENCES Restaurant(res_id),
    FOREIGN KEY (waiter_id) REFERENCES Waiter(waiter_id)
);
CREATE TABLE OrderFood (
    order_no INT,
    food_id INT,
    PRIMARY KEY (order_no, food_id),
    FOREIGN KEY (order_no) REFERENCES Orderdetails(order_no),
    FOREIGN KEY (food_id) REFERENCES Food(food_id)
);
```

❖ INSERT:

1 – Table: Restaurant

```
INSERT INTO Restaurant (res_id,licence_no,res_name,address) VALUES
(1,532145987,'Swad','153 College Road');

INSERT INTO Restaurant (res_id,licence_no,res_name,address) VALUES
(2,785412589,'Bluberrys','248 College Road');

INSERT INTO Restaurant (res_id,licence_no,res_name,address) VALUES
(3,324569885,'Navjivan','24 Vaniyavad');

INSERT INTO Restaurant (res_id,licence_no,res_name,address) VALUES
(4,145528836,'Ajays','59 Nadiad City');

INSERT INTO Restaurant (res_id,licence_no,res_name,address) VALUES
(5,254712699,'Madhuvan','76 College Road');
```

```
INSERT INTO Restaurant (res_id,licence_no,res_name,address) VALUES
(6,856999985,'Aquanoes','311 College Road');
```

2 – Table: Customer

```
INSERT INTO Customer (cus_id,cus_name,dob,contact_no) VALUES (1,'Kuldeep
Gabani',TO_DATE('2004-10-07', 'YYYY-MM-DD'),'9879825206');

INSERT INTO Customer (cus_id,cus_name,dob,contact_no) VALUES (3,'Yash
Gabani',TO_DATE('2005-08-14', 'YYYY-MM-DD'),'7046993816');

INSERT INTO Customer (cus_id,cus_name,dob,contact_no) VALUES (2,'Vaibhav
Dhanani',TO_DATE('2004-11-27', 'YYYY-MM-DD'),'7861979302');

INSERT INTO Customer (cus_id,cus_name,dob,contact_no) VALUES (4,'Meet
Antala',TO_DATE('2005-06-13', 'YYYY-MM-DD'),'9313217743');

INSERT INTO Customer (cus_id,cus_name,dob,contact_no) VALUES (5,'Rich
Amrutiya',TO_DATE('2005-04-25', 'YYYY-MM-DD'),'6353249404');

INSERT INTO Customer (cus_id,cus_name,dob,contact_no) VALUES (6,'Mahek
Garala',TO_DATE('2005-08-07', 'YYYY-MM-DD'),'8799188894');

INSERT INTO Customer (cus_id,cus_name,dob,contact_no) VALUES (7,'Lauren
Bell',TO_DATE('1992-01-12', 'YYYY-MM-DD'),'1234567890');

INSERT INTO Customer (cus_id,cus_name,dob,contact_no) VALUES (8,'Lakhman
Patel',TO_DATE('2005-05-19', 'YYYY-MM-DD'),'9574156941');

INSERT INTO Customer (cus_id,cus_name,dob,contact_no) VALUES (9,'Marizanne
Kapp',TO_DATE('1995-10-04', 'YYYY-MM-DD'),'9876543210');
```

3 – Table: Orderdetails

```
INSERT INTO Orderdetails (order_no, order_time,special_request, cus_id,
res_id, waiter_id) VALUES (3,'2023-09-21 19:30:00', 'Spicy and More Butter'
,3,1,1);
INSERT INTO Orderdetails (order_no, order_time, special_request cus_id,
res_id, waiter_id) VALUES (1,'2023-09-21 12:30:00','Nothing',1,1,4);
INSERT INTO Orderdetails (order_no, order_time, special_request cus_id,
res_id, waiter_id) VALUES (2,'2023-09-21 13:15:00','Extra Cheeze',2,2,2);
INSERT INTO Orderdetails (order_no, order_time, special_request cus_id,
res_id, waiter_id) VALUES (4,'2023-09-21 18:45:00','Less Oily',4,3,4);
INSERT INTO Orderdetails (order_no, order_time, special_request cus_id,
res_id, waiter_id) VALUES (5,'2023-09-21 18:18:00','Nothing',5,4,5);
INSERT INTO Orderdetails (order_no, order_time, special_request cus_id,
res_id, waiter_id) VALUES (6,'2023-09-21 20:15:00','Crunchy',6,2,1);
INSERT INTO Orderdetails (order_no, order_time, special_request cus_id,
res_id, waiter_id) VALUES (7,'2023-09-21 20:30:00','Double Butter',7,1,3);
```

```
INSERT INTO Orderdetails (order_no, order_time, special_request cus_id,
res_id, waiter_id) VALUES (8,'2023-09-21 23:15:00','Nothing',8,4,6);
INSERT INTO Orderdetails (order_no, order_time, special_request cus_id,
res_id, waiter_id) VALUES (9,'2023-09-21 19:05:00','No Onion and
Garlic',9,5,6);
```

4 – Table: Chef

```
INSERT INTO Chef (chef_id,chef_name) VALUES (1,'Chef Alice');
INSERT INTO Chef (chef_id,chef_name) VALUES (2,'Chef Mark');
INSERT INTO Chef (chef_id,chef_name) VALUES (4,'Chef Jhonson');
INSERT INTO Chef (chef_id,chef_name) VALUES (3,'Chef Gill');
```

5 – Table: Waiter

```
INSERT INTO Waiter (waiter_id, waiter_name, join_date, shift_start,
shift_end) VALUES (1, 'Ram', TO_DATE('2022-10-10', 'YYYY-MM-DD'),
TO_TIMESTAMP('2022-10-10 08:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2022-10-10 16:00:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Waiter (waiter_id, waiter_name, join_date, shift_start,
shift_end) VALUES (2, 'Smit', TO_DATE('2022-02-02', 'YYYY-MM-DD'),
TO_TIMESTAMP('2022-02-02 09:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2022-02-02 17:00:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Waiter (waiter_id, waiter_name, join_date, shift_start,
shift_end) VALUES (3, 'Suresh', TO_DATE('2022-12-22', 'YYYY-MM-DD'),
TO_TIMESTAMP('2022-12-22 10:30:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2022-12-22 18:30:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Waiter (waiter_id, waiter_name, join_date, shift_start,
shift_end) VALUES (4, 'Raju', TO_DATE('2022-11-19', 'YYYY-MM-DD'),
TO_TIMESTAMP('2022-11-19 08:30:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2022-11-19 16:30:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Waiter (waiter_id, waiter_name, join_date, shift_start,
shift_end) VALUES (5, 'Krish', TO_DATE('2022-04-14', 'YYYY-MM-DD'),
TO_TIMESTAMP('2022-04-14 11:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2022-04-14 19:00:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Waiter (waiter_id, waiter_name, join_date, shift_start,
shift_end) VALUES (6, 'Vedant', TO_DATE('2022-05-28', 'YYYY-MM-DD'),
TO_TIMESTAMP('2022-05-28 12:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2022-05-28 20:00:00', 'YYYY-MM-DD HH24:MI:SS'));
```

6 – Table: Food

```
INSERT INTO Food (food_id,food_name,price,description,chef_id) VALUES
(1,'Spaghetti Carbonara',140,'Pasta with cramy sauce',1);

INSERT INTO Food (food_id,food_name,price,description,chef_id) VALUES
(2,'Margherita Pizza',200,'Classic Pizza',2);

INSERT INTO Food (food_id,food_name,price,description,chef_id) VALUES
(4,'Vaganxe Sizzler',670,'Vegetables with Noodeles and corn',4);

INSERT INTO Food (food_id,food_name,price,description,chef_id) VALUES
(3,'Time Pass',60,'Combination of Bread and Salad',3);

INSERT INTO Food (food_id,food_name,price,description,chef_id) VALUES
(5,'Barbeque',160,'Shaked Masala Paneer',1);

INSERT INTO Food (food_id,food_name,price,description,chef_id) VALUES
(6,'Sushi Combo',180,'Assorted sushi rolls',2);
```

7 – Table: Billcounter

```
INSERT INTO Billcounter (billcounter_id,location,res_id) VALUES
(1,'Counter 1',1);

INSERT INTO Billcounter (billcounter_id,location,res_id) VALUES
(2,'Counter 2',2);

INSERT INTO Billcounter (billcounter_id,location,res_id) VALUES
(3,'Counter 3',3);

INSERT INTO Billcounter (billcounter_id,location,res_id) VALUES
(4,'Counter 4',4);

INSERT INTO Billcounter (billcounter_id,location,res_id) VALUES
(5,'Counter 5',5);

INSERT INTO Billcounter (billcounter_id,location,res_id) VALUES
(6,'Counter 6',6);
```

8 – Table: Orderfood

```
INSERT INTO Orderfood (order_no,food_id) VALUES (1,1);
INSERT INTO Orderfood (order_no,food_id) VALUES (1,3);
INSERT INTO Orderfood (order_no,food_id) VALUES (2,2);
INSERT INTO Orderfood (order_no,food_id) VALUES (2,5);
INSERT INTO Orderfood (order_no,food_id) VALUES (3,4);
INSERT INTO Orderfood (order_no,food_id) VALUES (4,1);
INSERT INTO Orderfood (order_no,food_id) VALUES (4,2);
INSERT INTO Orderfood (order_no,food_id) VALUES (5,3);
INSERT INTO Orderfood (order_no,food_id) VALUES (6,6);
```

```
INSERT INTO Orderfood (order_no,food_id) VALUES (6,4);
INSERT INTO Orderfood (order_no,food_id) VALUES (6,5);
INSERT INTO Orderfood (order_no,food_id) VALUES (7,2);
INSERT INTO Orderfood (order_no,food_id) VALUES (7,4);
INSERT INTO Orderfood (order_no,food_id) VALUES (8,1);
INSERT INTO Orderfood (order_no,food_id) VALUES (9,5);INSERT INTO
Orderfood (order_no,food_id) VALUES (8,5);
```

9 – Table: Payment

```
INSERT INTO Payment (payment_id,amount,payment_time,cus_id) VALUES
(1,200,'2023-09-21 13:30:00',1);
INSERT INTO Payment (payment_id,amount,payment_time,cus_id) VALUES
(2,260,'2023-09-21 13:45:00',2);
INSERT INTO Payment (payment_id,amount,payment_time,cus_id) VALUES
(3,180,'2023-09-21 19:15:00',3);
INSERT INTO Payment (payment_id,amount,payment_time,cus_id) VALUES
(4,670,'2023-09-21 18:45:00',4);
INSERT INTO Payment (payment_id,amount,payment_time,cus_id) VALUES
(5,180,'2023-09-21 15:00:00',5);
INSERT INTO Payment (payment_id,amount,payment_time,cus_id) VALUES
(6,850,'2023-09-21 21:00:00',6);
INSERT INTO Payment (payment_id,amount,payment_time,cus_id) VALUES
(7,870,'2023-09-21 20:30:00',7);
INSERT INTO Payment (payment_id,amount,payment_time,cus_id) VALUES
(8,300,'2023-09-21 21:15:00',8);INSERT INTO Payment
(payment_id,amount,payment_time,cus_id) VALUES (9,160,'2023-09-21
19:05:00',9);
```

❖ READ:

```
SELECT * FROM Food WHERE chef_id = 1;
```
//Retrieving food items prepared by chef1

```
SELECT * FROM Order WHERE order_no = 1;
```
//Retrieving order details

❖ UPDATE:

```
UPDATE Customer SET cus_name = 'Jane Smith' WHERE cus_id = 1;
```
// Update customer

```
UPDATE Waiter SET waiter_name = 'Bob Anderson' WHERE waiter_id = 1;
```
//Updating Waiter information

```
UPDATE Food SET price = 13.99 WHERE food_id = 1;
```
// Updating food price

❖ DELETE:

```
DELETE FROM Customer WHERE cus_id = 1;
```
// Delete customer

```
DELETE FROM Order WHERE order_no = 1;
```
// Delete an order

```
DELETE FROM Food WHERE food_id = 1;
```
// Delete food item

```
DELETE FROM Waiter WHERE waiter_id = 1;
```
// Delete waiter

```
DELETE FROM Billcounter WHERE billcounter_id = 1;
```
// Delete Bill Counter

```
DELETE FROM Chef WHERE chef_id = 1;
```
// Delete chef

❖ Queries:

1) Non-correlated Queries:-

```
1.  SELECT O.order_no, O.order_time, F.food_name, F.price FROM
Orderdetails O JOIN OrderFood Orf ON O.order_no = Orf.order_no JOIN
Food F ON Orf.food_id = F.food_id WHERE O.cus_id = cus_id;
```

// List of order for Customer

ORDER_NO	ORDER_TIME	FOOD_NAME	PRICE
3	2023-09-21 19:30:00	Vaganxe Sizzler	670
1	2023-09-21 12:30:00	Spaghetti Carbonara	140
1	2023-09-21 12:30:00	Time Pass	60
2	2023-09-21 13:15:00	Margherita Pizza	200
2	2023-09-21 13:15:00	Barbeque	160
4	2023-09-21 18:45:00	Spaghetti Carbonara	140
4	2023-09-21 18:45:00	Margherita Pizza	200
5	2023-09-21 18:18:00	Time Pass	60
6	2023-09-21 20:15:00	Vaganxe Sizzler	670
6	2023-09-21 20:15:00	Barbeque	160
6	2023-09-21 20:15:00	Sushi Combo	180
7	2023-09-21 20:30:00	Margherita Pizza	200
7	2023-09-21 20:30:00	Vaganxe Sizzler	670
8	2023-09-21 23:15:00	Spaghetti Carbonara	140
9	2023-09-21 19:05:00	Barbeque	160

```
2.  SELECT C.chef_name, LISTAGG(F.food_name, ', ') WITHIN GROUP
(ORDER BY F.food_name) AS specialties FROM Chef C JOIN Food F ON
C.chef_id = F.chef_id GROUP BY C.chef_name;
```

// List of chef's with their specialties

CHEF_NAME	SPECIALTIES
Chef Alice	Barbeque, Spaghetti Carbonara
Chef Gill	Time Pass
Chef Jhonson	Vaganxe Sizzler
Chef Mark	Margherita Pizza, Sushi Combo

```
3.  SELECT P.payment_id, P.amount, P.payment_time FROM Payment P
WHERE P.cus_id = cus_id;
```

// Customer Payment

PAYMENT_ID	AMOUNT	PAYMENT_TIME
1	200	2023-09-21 13:30:00
2	260	2023-09-21 13:45:00
3	180	2023-09-21 19:15:00
4	670	2023-09-21 18:45:00
5	180	2023-09-21 15:00:00
6	850	2023-09-21 21:00:00
7	870	2023-09-21 20:30:00
8	300	2023-09-21 21:15:00

```
4.  SELECT R.res_name, B.location FROM Restaurant R LEFT JOIN
Billcounter B ON R.res_id = B.res_id;
```

// Bill Counters for each restaurant

RES_NAME	LOCATION
Swad	Counter 1
Bluberrys	Counter 2
Navjivan	Counter 3
Ajays	Counter 4
Madhuvan	Counter 5
Aquanoes	Counter 6

```
5. SELECT cus_name, MAX(amount) AS highest_amount FROM Customer C
JOIN Payment P ON C.cus_id = P.cus_id GROUP BY cus_name ORDER BY
highest_amount DESC FETCH FIRST 1 ROW ONLY;
```

// Customer with highest Payment

CUS_NAME	HIGHEST_AMOUNT
Lauren Bell	870

2) Correlated Queries:-

```
1. SELECT C.cus_id, C.cus_name, OrF.order_no, F.food_name FROM
Customer C JOIN Orderdetails OD ON C.cus_id = OD.cus_id JOIN
OrderFood OrF ON OD.order_no = OrF.order_no JOIN Food F ON
OrF.food_id = F.food_id;
```

// Retrieving customer name and ordered food items

CUS_ID	CUS_NAME	ORDER_NO	FOOD_NAME
1	Kuldeep Gabani	1	Spaghetti Carbonara
4	Meet Antala	4	Spaghetti Carbonara
8	Lakhman Patel	8	Spaghetti Carbonara
2	Vaibhav Dhanani	2	Margherita Pizza
4	Meet Antala	4	Margherita Pizza
7	Lauren Bell	7	Margherita Pizza
3	Yash Gabani	3	Vaganxe Sizzler
6	Mahek Garala	6	Vaganxe Sizzler
7	Lauren Bell	7	Vaganxe Sizzler
1	Kuldeep Gabani	1	Time Pass
5	Rich Amrutiya	5	Time Pass
2	Vaibhav Dhanani	2	Barbeque
6	Mahek Garala	6	Barbeque
9	Marizanne Kapp	9	Barbeque
6	Mahek Garala	6	Sushi Combo

```
2. SELECT cus_id, cus_name,(SELECT COUNT(*) FROM Orderdetails WHERE
Orderdetails.cus_id = Customer.cus_id) AS order_count FROM Customer;
```

// Customer’s total orders

CUS_ID	CUS_NAME	ORDER_COUNT
1	Kuldeep Gabani	1
3	Yash Gabani	1
2	Vaibhav Dhanani	1
4	Meet Antala	1
5	Rich Amrutiya	1
6	Mahek Garala	1
7	Lauren Bell	1
8	Lakhman Patel	1
9	Marizanne Kapp	1

```
3. SELECT cus_name FROM Customer C WHERE EXISTS(SELECT 1 FROM Payment
P WHERE P.cus_id = C.cus_id);
```

// Retrieving customer who made payment

CUS_NAME
Kuldeep Gabani
Vaibhav Dhanani
Yash Gabani
Meet Antala
Rich Amrutiya
Mahek Garala
Lauren Bell
Lakhman Patel

```
4. SELECT res_name, COUNT(order_no) AS total_orders FROM Restaurant R
LEFT JOIN Orderdetails O ON R.res_id = O.res_id GROUP BY res_name;
```

// Retrieving restaurants with total number of orders

RES_NAME	TOTAL_ORDERS
Aquanoes	0
Bluberrys	2
Swad	3
Madhuvan	1
Navjivan	1
Ajays	2

```
5. SELECT C.cus_id, C.cus_name, SUM(P.amount) AS total_amount FROM
Customer C JOIN Payment P ON C.cus_id = P.cus_id GROUP BY C.cus_id,
C.cus_name;
```

// total amount spent by each customer

CUS_ID	CUS_NAME	TOTAL_AMOUNT
4	Meet Antala	670
8	Lakhman Patel	300
2	Vaibhav Dhanani	260
5	Rich Amrutiya	180
3	Yash Gabani	180
6	Mahek Garala	850
1	Kuldeep Gabani	200
7	Lauren Bell	870

References

- Henry Korth Book for designing ER Diagram
- Canva to create the ER diagram digitally
- Advice from Faculty to reconstruct the ER Diagram
- Oracle live SQL
- Stack Overflow for Query Questions
- Simplilearn.com
- JavaTPoint for SQL Syntax
- Great Learning