AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

Course Title

INTRODUCTION TO DATABASE (2108)

Semester: Spring 2023-2024

Section: [C]

TITLE

Restaurant Management System

Supervised By

Saeeda Sharmeen Rahman

Submitted By : Group no: 01

Name	ID
1.BURHAN UDDIN	22-49945-3
2.HASHAMEE AL SHAHRIAR	22-49941-3
3.FARHANA AHMED SHAMANTA	22-49920-3
4.HAMIM IBNE RAHMAN	22-49933-3
5.Sajid Hasan Mahir	22-49485-3

TABLE OF CONTENTS

Introduction

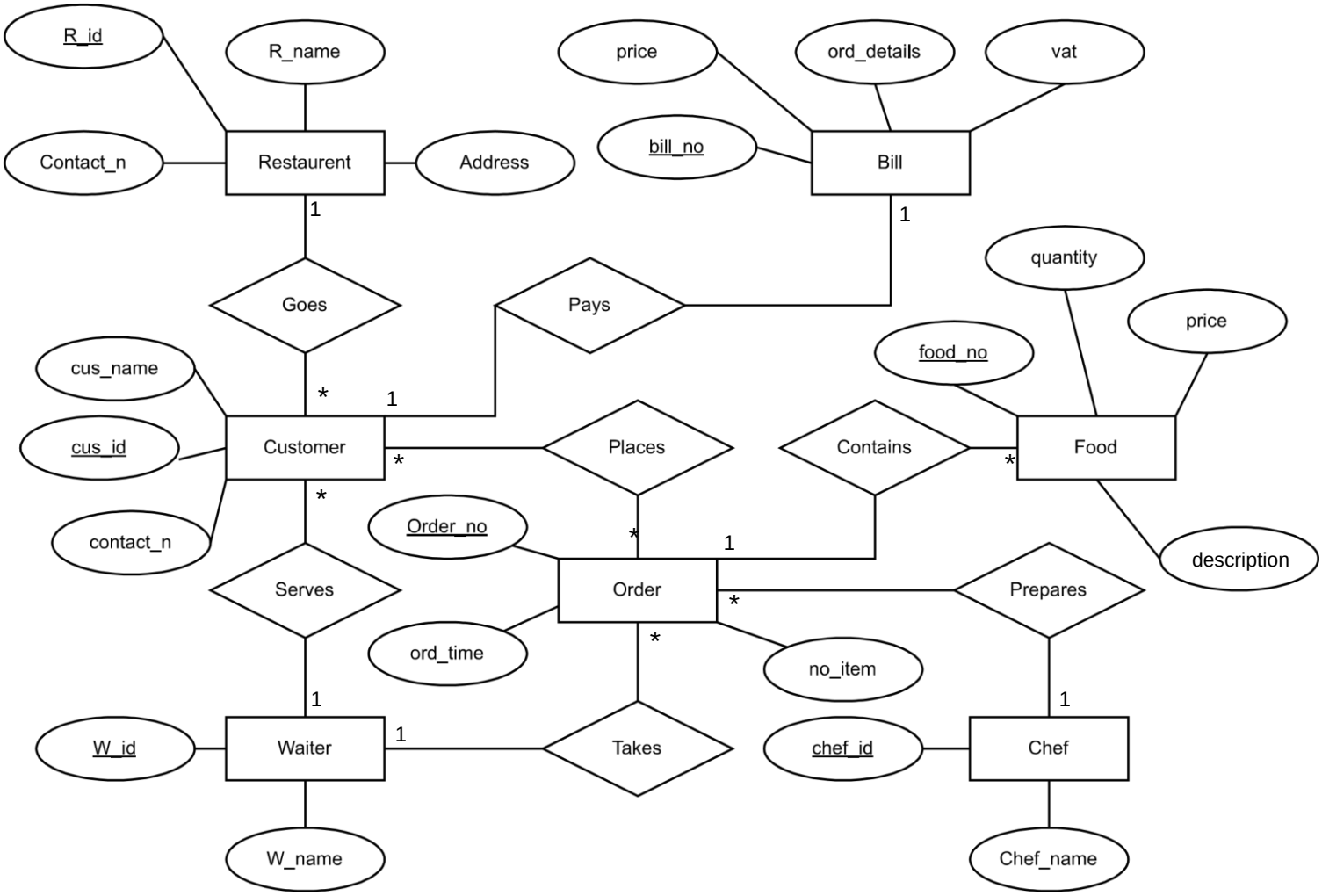
In today's fast-paced and competitive restaurant world, running things smoothly is crucial for keeping customers happy and the business thriving. As managing tasks like orders, stock, and staff becomes more complicated, having a strong Database Management System (DBMS) is vital.

This project is all about creating and putting into action a complete database system for a made-up restaurant. This includes everything from the restaurants themselves to the customers, waiters, chefs, orders, and food items. Each part has its own special details and connections, which are the building blocks for our database setup.

Case Study / Scenario

A restaurant has a unique restaurant ID(R_id), name(R_name), contact number(contact_no) and address. Many customers can go to one restaurant. A customer has unique customer ID(Cus_id), Name (Cus_name), contact number(contact_no). One waiter can serve more than one customers. A waiter has a unique ID(w_id) and name(W_name). A customer places order via waiter and the order is prepared by Chef. An order has unique order number(order_no), Number of items(no_items) and order time (ord_time). A chef has unique ID number(chef_id) and name(chef_name). An order contains food which has unique food number(food_no), quantity, price and description. Customers pay bills that contains a unique bill number(b_no), price, order detail (ord_detail) and vat.

ER Diagram


R_id
R_name
price
ord_details
vat
Contact_n
Restaurent
Address
bill_no
Bill
1
1
quantity
Goes
Pays
price
food_no
cus_name
*
1
cus_id
Customer
*
Places
Contains
Food
*
*
contact_n
Order_no
*
1
description
Serves
Order
Prepares
*
ord_time
*
no_item
1
1
W_id
Waiter
1
Takes
chef_id
Chef
W_name
Chef_name

Normalization

Goes:

UNF: R_id, R_name, Contact_no, Address, Cus_id, Contact_no, Cus_name

1NF:

Customer table: Cus_id, Cus_name, Contact_no, R_id

Restaurant table: R_id, R_name, Contact_no, address

2NF:

Customer table: Cus_id, Cus_name, Contact_no, R_id

Restaurant table: R_id, R_name, Contact_no, address

3NF:

Customer table: Cus_id, Cus_name, Contact_no, R_id

Restaurant table: R id, R_name, contact-no, R_id

Restaurant_info table: R_name, Address

Serves:

UNF: Cus_Id, Cus_name, Contact_no, W_id, Wname

1NF:

Customer table: Cus_id, Cus_name, contact_no

Waiter table: W_id, W_name, Cus_id

2NF:

Customer table: Cus_id, Cus_name, contact_no

Waiter table: W_id, W_name, Cus_id

3NF:

Customer table: Cus_id, Cus_name, contact_no

Waiter table: W_id, W_name, Cus_id

Prepares:

UNF: chef_id, Chef_name, order_no, no_items, ord_time

1NF:

Chef table: Chef_id, Chef_name, order_no

Order table: order_no, no_items, ord_time

2NF:

Chef table: Chef_id, Chef_name, order_no

Order table: order_no, no_items, ord_time

3NF:

Chef table: Chef_id, Chef_name, order_no

Order table: order_no, no_items

Order_info table: no_items, ord_time

Takes:

UNF: W_id, W_name, order_no, no_items, ord_time

1NF:

Waiter table: W_id, W_name, order_no

Order table: order_no, no_items, ord_time

2NF:

Waiter table: W_id, W_name, order_no

Order table: order_no, no_items, ord_time

3NF:

Waiter table: W_id, W_name, order_no

Order table: order_no, no_items

Order_info table: no_items, ord_time

Places:

UNF: Cus_id, Cus_name, contact_no, order_no, no_items, ord_time

1NF:

Customer table: Cus_id, Cus_name, contact_no

Order table: order_no, no_items, ord_time

2NF:

Customer table: Cus_id, Cus_name, contact_no

Order table: order_no, no_items, ord_time

3NF:

Customer table: Cus_id, Cus_name, contact_no

Order table: order_no, no_items

Order_info table: no_items, ord_time

Contains:

UNF: Order_no, No_items, ord_date, food_no, quantity, price, description

1NF:

Order table: order_no, no_items, ord_time

Food table: food_no, quantity, price, description, order_no

2NF:

Order table: order_no, no_items, ord_time

Food table: food_no, quantity, price, description, order_no

3NF:

Order table: order_no, no_items, ord_time

Food table: food_no, price, quantity, description, order_no

Food_detail table: food_no, quantity , price

Pays:

UNF: Cus_id, Cus_name, contact_no, b_no, price, ord_detail, vat

INF: Customer table: Cus_id, Cus_name, contact_no

Bill table: b_no, price, ord_detail, vat, Cus_id

2NF: Customer table: Cus_id, Cus_name, contact_no

Bill table: b_no, price, ord_detail, vat, Cus_id

3NF: Customer table: Cus_id, Cus_name, contact_no

Bill table: b_no, price, ord_detail, Cus_id

Bill_details: price, vat

Finalization

Final Tables:

1. **Restaurant** table: r_id(pk), r_name, contact_n
2. **Customer** table: cus_id(pk), cus_name, contact_no, r_id(fk)
3. **Restaurant_info** table: r_name(pk), address
4. **Waiter** table: W_id(pk), w_name, cus_id(fk), order_no(fk)
5. **Order** table: ord_no(pk), no_item
6. **Order Info** table: no_item(pk), ord_time
7. **Food** table: food_no(pk), quantity, description, order_no(fk)
8. **Food details** table: food_no(fk), quantity, price
9. **Chef** table: chef_id(pk), chef_name, ord_no(fk)
10. **Bill** table: bill_no(pk), order_details, cus_id(fk)
11. **Bill_details**: price(pk), vat

Table Creation

Screenshots of table creation command and table description-

```
CREATE TABLE restaurant (
    r_id NUMBER(5) PRIMARY KEY,
    r_name VARCHAR2(20),
    contact_n NUMBER(11)
);
DESCRIBE restaurant;
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **RESTAURANT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RESTAURANT	R_ID	Number	-	5	0	1	-	-	-
	R_NAME	Varchar2	20	-	-	-	✓	-	-
	CONTACT_N	Number	-	11	0	-	✓	-	-

1 - 3

```
CREATE TABLE customer (
    cus_id NUMBER(5) PRIMARY KEY,
    cus_name VARCHAR2(20),
    contact_n NUMBER(11),
    r_id NUMBER,
    CONSTRAINT fk_restaurant FOREIGN KEY (r_id) REFERENCES restaurant (r_id)
);
describe customer;
```

Object Type **TABLE** Object **CUSTOMER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUS_ID	Number	-	5	0	1	-	-	-
	CUS_NAME	Varchar2	20	-	-	-	✓	-	-
	CONTACT_N	Number	-	11	0	-	✓	-	-
	R_ID	Number	-	-	-	-	✓	-	-

1 - 4

```
CREATE TABLE restaurant_info (
    r_name VARCHAR2(20) PRIMARY KEY,
    address VARCHAR2(20)
);
describe restaurant_info;
```

Object Type **TABLE** Object **RESTAURANT_INFO**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RESTAURANT_INFO	R_NAME	Varchar2	20	-	-	1	-	-	-
	ADDRESS	Varchar2	20	-	-	-	✓	-	-

1 - 2

```
CREATE TABLE "order" (
    ord_no NUMBER(5) PRIMARY KEY,
    no_item NUMBER(5)
);
describe "order";
```

Object Type **TABLE** Object **order**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
order	ORD_NO	Number	-	5	0	1	-	-	-
	NO_ITEM	Number	-	5	0	-	✓	-	-

1 - 2

```
CREATE TABLE waiter (
    w_id NUMBER(5) PRIMARY KEY,
    w_name VARCHAR2(20),
    cus_id NUMBER(5),
    ord_no NUMBER(5),
    CONSTRAINT fk_customer FOREIGN KEY (cus_id) REFERENCES customer(cus_id),
    CONSTRAINT fk_order FOREIGN KEY (ord_no) REFERENCES "order"(ord_no)
);
describe waiter;
```

Object Type **TABLE** Object **WAITER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
WAITER	W_ID	Number	-	5	0	1	-	-	-
	W_NAME	Varchar2	20	-	-	-	✓	-	-
	CUS_ID	Number	-	5	0	-	✓	-	-
	ORD_NO	Number	-	5	0	-	✓	-	-

1 - 4

```
CREATE TABLE order_info (
    no_item NUMBER(5) PRIMARY KEY,
    ord_time date
);
describe order_info;
```

Object Type **TABLE** Object **ORDER_INFO**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDER_INFO	NO_ITEM	Number	-	5	0	1	-	-	-
	ORD_TIME	Date	7	-	-	-	✓	-	-

1 - 2

```
CREATE TABLE food (
    food_no NUMBER(5) PRIMARY KEY,
    quantity NUMBER(5),
    description VARCHAR2(20),
    ord_no NUMBER(5),
    CONSTRAINT fk_orderr FOREIGN KEY (ord_no) REFERENCES "order" (ord_no)
);
describe food;
```

Object Type **TABLE** Object **FOOD**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FOOD	FOOD_NO	Number	-	5	0	1	-	-	-
	QUANTITY	Number	-	5	0	-	✓	-	-
	DESCRIPTION	Varchar2	20	-	-	-	✓	-	-
	ORD_NO	Number	-	5	0	-	✓	-	-

1 - 4

```
CREATE TABLE food_details (
    food_no NUMBER(5),
    quantity NUMBER(5),
    price NUMBER(5),
    CONSTRAINT fk_food FOREIGN KEY (food_no) REFERENCES food (food_no)
);
describe food_details;
```

Object Type **TABLE** Object **FOOD_DETAILS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FOOD_DETAILS	FOOD_NO	Number	-	5	0	-	✓	-	-
	QUANTITY	Number	-	5	0	-	✓	-	-
	PRICE	Number	-	5	0	-	✓	-	-

1 - 3

```
CREATE TABLE chef (
    chef_id NUMBER(5) PRIMARY KEY,
    chef_name VARCHAR2(20),
    ord_no NUMBER(5),
    CONSTRAINT fk_orderrr FOREIGN KEY (ord_no) REFERENCES "order" (ord_no)
);
describe chef;
```

Object Type **TABLE** Object **CHEF**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CHEF	CHEF_ID	Number	-	5	0	1	-	-	-
	CHEF_NAME	Varchar2	20	-	-	-	✓	-	-
	ORD_NO	Number	-	5	0	-	✓	-	-

1 - 3

```
CREATE TABLE bill (
    bill_no NUMBER(5) PRIMARY KEY,
    order_details VARCHAR2(20),
    price NUMBER(5),
    cus_id NUMBER(5),
    CONSTRAINT fk_customerr FOREIGN KEY (cus_id) REFERENCES customer (cus_id)
);
describe bill;
```

Object Type **TABLE** Object **BILL**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BILL	BILL_NO	Number	-	5	0	1	-	-	-
	ORDER_DETAILS	Varchar2	20	-	-	-	✓	-	-
	PRICE	Number	-	5	0	-	✓	-	-
	CUS_ID	Number	-	5	0	-	✓	-	-

1 - 4

```
CREATE TABLE bill_details (
    price NUMBER(5) PRIMARY KEY,
    vat NUMBER(5)
);
describe bill_details;
```

Object Type **TABLE** Object **BILL_DETAILS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BILL_DETAILS	PRICE	Number	-	5	0	1	-	-	-
	VAT	Number	-	5	0	-	✓	-	-

1 - 2

Inserted Values in the tables

```
INSERT INTO restaurant (r_id, r_name, contact_n) VALUES (1, 'The Hungry Panda', '1234567890');
INSERT INTO restaurant (r_id, r_name, contact_n) VALUES (2, 'Italiano Ristorante', '9876543210');
INSERT INTO restaurant (r_id, r_name, contact_n) VALUES (3, 'Sushi Samurai', '5551234567');
INSERT INTO restaurant (r_id, r_name, contact_n) VALUES (4, 'Taco Town', '9998887776');
INSERT INTO restaurant (r_id, r_name, contact_n) VALUES (5, 'La Pizzeria', '4445556667');
INSERT INTO restaurant (r_id, r_name, contact_n) VALUES (6, 'Burger Haven', '7778889990');

select * from restaurant;
```

R_ID	R_NAME	CONTACT_N
1	The Hungry Panda	1234567890
2	Italiano Ristorante	9876543210
3	Sushi Samurai	5551234567
4	Taco Town	9998887776
5	La Pizzeria	4445556667
6	Burger Haven	7778889990

```
INSERT INTO restaurant_info (r_name, address) VALUES ('The Hungry Panda', '123 ABC Road, Dhaka');
INSERT INTO restaurant_info (r_name, address) VALUES ('Italiano Ristorante', '456 Road, Chittagong');
INSERT INTO restaurant_info (r_name, address) VALUES ('Sushi Samurai', '789 Road, Sylhet');
INSERT INTO restaurant_info (r_name, address) VALUES ('Taco Town', '101 Road, Rajshahi');
INSERT INTO restaurant_info (r_name, address) VALUES ('La Pizzeria', '202 Road, Khulna');
INSERT INTO restaurant_info (r_name, address) VALUES ('Burger Haven', '303 Road, Barisal');
select * from restaurant_info;
```

R_NAME	ADDRESS
The Hungry Panda	123 ABC Road, Dhaka
Italiano Ristorante	456 Road, Chittagong
Sushi Samurai	789 Road, Sylhet
Taco Town	101 Road, Rajshahi
La Pizzeria	202 Road, Khulna
Burger Haven	303 Road, Barisal

```
INSERT INTO customer (cus_id, cus_name, contact_n, r_id) VALUES (1, 'Hamim', '1234567890',1);
INSERT INTO customer (cus_id, cus_name, contact_n, r_id) VALUES (2, 'Farhana', '9876543210',1);
INSERT INTO customer (cus_id, cus_name, contact_n, r_id) VALUES (3, 'Borhan', '5551234567',1);
INSERT INTO customer (cus_id, cus_name, contact_n, r_id) VALUES (4, 'Hashamee', '9998887776',1);
INSERT INTO customer (cus_id, cus_name, contact_n, r_id) VALUES (5, 'Mahir', '4445556667',1);
select * from customer;
```

CUS_ID	CUS_NAME	CONTACT_N	R_ID
1	Hamim	1234567890	1
2	Farhana	9876543210	1
3	Borhan	5551234567	1
4	Hashamee	9998887776	1
5	Mahir	4445556667	1

```
INSERT INTO "order" (ord_no, no_item) VALUES (1, 3);
INSERT INTO "order" (ord_no, no_item) VALUES (2, 2);
INSERT INTO "order" (ord_no, no_item) VALUES (3, 4);
INSERT INTO "order" (ord_no, no_item) VALUES (4, 1);
INSERT INTO "order" (ord_no, no_item) VALUES (5, 2);
select * from "order";
```

ORD_NO	NO_ITEM
1	3
2	2
3	4
4	1
5	2

```
INSERT INTO order_info (no_item, ord_time) VALUES (3, TO_DATE('2024-05-12', 'YYYY-MM-DD'));
INSERT INTO order_info (no_item, ord_time) VALUES (2, TO_DATE('2024-05-12 ', 'YYYY-MM-DD'));
INSERT INTO order_info (no_item, ord_time) VALUES (4, TO_DATE('2024-05-11 ', 'YYYY-MM-DD '));
INSERT INTO order_info (no_item, ord_time) VALUES (1, TO_DATE('2024-05-11 ', 'YYYY-MM-DD '));
INSERT INTO order_info (no_item, ord_time) VALUES (5, TO_DATE('2024-05-11 ', 'YYYY-MM-DD '));
select * from order_info;
```

NO_ITEM	ORD_TIME
3	12-MAY-24
2	12-MAY-24
4	11-MAY-24
1	11-MAY-24
5	11-MAY-24

```
INSERT INTO waiter (w_id, w_name, cus_id, ord_no) VALUES (1, 'Michael', 1, 1);
INSERT INTO waiter (w_id, w_name, cus_id, ord_no) VALUES (2, 'Emily', 2, 2);
INSERT INTO waiter (w_id, w_name, cus_id, ord_no) VALUES (3, 'David', 3, 3);
INSERT INTO waiter (w_id, w_name, cus_id, ord_no) VALUES (4, 'Sophia', 4, 4);
INSERT INTO waiter (w_id, w_name, cus_id, ord_no) VALUES (5, 'Olivia', 5, 5);
select * from waiter;
```

W_ID	W_NAME	CUS_ID	ORD_NO
1	Michael	1	1
2	Emily	2	2
3	David	3	3
4	Sophia	4	4
5	Olivia	5	5

```
INSERT INTO food (food_no, quantity, description, ord_no) VALUES (1, 2, 'Chicken Curry', 1);
INSERT INTO food (food_no, quantity, description, ord_no) VALUES (2, 1, 'Vegetable Fried Rice', 1);
INSERT INTO food (food_no, quantity, description, ord_no) VALUES (3, 3, 'Sushi Platter', 2);
INSERT INTO food (food_no, quantity, description, ord_no) VALUES (4, 1, 'Taco Combo', 3);
INSERT INTO food (food_no, quantity, description, ord_no) VALUES (5, 2, 'Margherita Pizza', 4);
select * from food;
```

FOOD_NO	QUANTITY	DESCRIPTION	ORD_NO
1	2	Chicken Curry	1
2	1	Vegetable Fried Rice	1
3	3	Sushi Platter	2
4	1	Taco Combo	3
5	2	Margherita Pizza	4

```
INSERT INTO food_details (food_no, quantity, price) VALUES (1, 2, 100);
INSERT INTO food_details (food_no, quantity, price) VALUES (2, 1, 150);
INSERT INTO food_details (food_no, quantity, price) VALUES (3, 3, 200);
INSERT INTO food_details (food_no, quantity, price) VALUES (4, 1, 120);
INSERT INTO food_details (food_no, quantity, price) VALUES (5, 2, 180);
select* from food_details;
```

FOOD_NO	QUANTITY	PRICE
1	2	100
2	1	150
3	3	200
4	1	120
5	2	180

```
INSERT INTO chef (chef_id, chef_name, ord_no) VALUES (1, 'Chef John', 1);
INSERT INTO chef (chef_id, chef_name, ord_no) VALUES (2, 'Chef Maria', 2);
INSERT INTO chef (chef_id, chef_name, ord_no) VALUES (3, 'Chef David', 3);
INSERT INTO chef (chef_id, chef_name, ord_no) VALUES (4, 'Chef Emily', 4);
INSERT INTO chef (chef_id, chef_name, ord_no) VALUES (5, 'Chef Michael', 5);
select * from chef;
```

CHEF_ID	CHEF_NAME	ORD_NO
1	Chef John	1
2	Chef Maria	2
3	Chef David	3
4	Chef Emily	4
5	Chef Michael	5

```
INSERT INTO bill (bill_no, order_details, price, cus_id) VALUES (1, 'Chk-Curry,VgF-Rice', 250, 1);
INSERT INTO bill (bill_no, order_details, price, cus_id) VALUES (2, 'Sushi Platter', 300, 2);
INSERT INTO bill (bill_no, order_details, price, cus_id) VALUES (3, 'Taco Combo', 150, 3);
INSERT INTO bill (bill_no, order_details, price, cus_id) VALUES (4, 'Margherita Pizza', 200, 4);
INSERT INTO bill (bill_no, order_details, price, cus_id) VALUES (5, 'Chk-Curry,VgF-Rice', 250, 5);
select * from bill;
```

BILL_NO	ORDER_DETAILS	PRICE	CUS_ID
1	Chk-Curry,VgF-Rice	250	1
2	Sushi Platter	300	2
3	Taco Combo	150	3
4	Margherita Pizza	200	4
5	Chk-Curry,VgF-Rice	250	5

```
INSERT INTO bill_details (price, vat) VALUES (250, 15);
INSERT INTO bill_details (price, vat) VALUES (300, 20);
INSERT INTO bill_details (price, vat) VALUES (150, 10);
INSERT INTO bill_details (price, vat) VALUES (200, 12);
INSERT INTO bill_details (price, vat) VALUES (260, 16);
select* from bill_details;
```

PRICE	VAT
250	15
300	20
150	10
200	12
260	16

Query Test in DB

1.simple query

a) Show the customer table

```
select* from customer;
```

Fig: sql command

Results Explain Describe Saved SQL History

CUS_ID	CUS_NAME	CONTACT_N	R_ID
1	Hamim	1234567890	1
2	Farhana	9876543210	1
3	Borhan	5551234567	1
4	Hashamee	9998887776	1
5	Mahir	4445556667	1

5 rows returned in 0.00 seconds CSV Export

Fig: Result

2. Query with a single row function

a) show all customer name length

```
SELECT cus_name, LENGTH(cus_name) AS name_length
FROM customer;
```

Fig: sql command

Results Explain Describe Saved SQL History

CUS_NAME	NAME_LENGTH
Hamim	5
Farhana	7
Borhan	6
Hashamee	8
Mahir	5

5 rows returned in 0.00 seconds CSV Export

3.Query with multiple row function

a) Count the total number of orders taken.

```
SELECT COUNT(*) AS total_orders
FROM "order";
```

Fig: sql command

Results Explain Describe Saved SQL History

TOTAL_ORDERS
5

1 rows returned in 0.00 seconds CSV Export

Fig: result

4. Two single row subquery & two multiple row subquery

a) Single row subquery

1.Show the waiter id and name who serves the customer Farhana

```
select w_name, w_id
from waiter
where cus_id=(select cus_id from customer where cus_name= 'Farhana');
```

Fig: single row sub query sql command

Results Explain Describe Saved SQL History

W_NAME	W_ID
Emily	2

Fig: results

2.Show all customer name and phone number who goes to The Hungry Panda restaurant.

```
select cus_name,contact_n
from customer
where r_id =(select r_id from restaurant where r_name='The Hungry Panda');
```

Fig: single row subquery sql command

Results Explain Describe Saved SQL History

CUS_NAME	CONTACT_N
Hamim	1234567890
Farhana	9876543210
Borhan	5551234567
Hashamee	9998887776
Mahir	4445556667

5 rows returned in 0.00 seconds CSV Export

Fig: results

b) Multiple row subquery

1.Show the customer name who order Chicken Curry

```
select cus_name
from customer
 where cus_id in(select cus_id
                 from waiter
                  where ord_no= (select ord_no
                                 from food
                                  where description= 'Chicken Curry'));
```

Fig: multiple row subquery

Results Explain Describe Saved SQL History

CUS_NAME
Hamim

1 rows returned in 0.00 seconds CSV Export

Fig: result

2.Show the waiters who works in The Hungy Panda.

```
select w_id ,W_name
 from waiter
 where cus_id in(select cus_id
                 from customer
                  where r_id =(select r_id
                               from restaurant
                                where r_name='The Hungry Panda'));
```

Fig: multiple row subquery sql command

Results Explain Describe Saved SQL History

W_ID	W_NAME
1	Michael
2	Emily
3	David
4	Sophia
5	Olivia

5 rows returned in 0.02 seconds CSV Export

Fig: results

e) 4 Kinds of joining

1.show customer name, contact number and restaurant name

```
select c.cus_name, c.contact_n, r.r_name
from customer c, restaurant r
where c.r_id=r.r_id;
```

Fig: inner join query

Results Explain Describe Saved SQL History

CUS_NAME	CONTACT_N	R_NAME
Mahir	4445556667	The Hungry Panda
Hashamee	9998887776	The Hungry Panda
Borhan	5551234567	The Hungry Panda
Farhana	9876543210	The Hungry Panda
Hamim	1234567890	The Hungry Panda

5 rows returned in 0.00 seconds CSV Export

Fig: results

2. show waiter name who takes order

```
select w.w_name
from waiter w, food f
where w.ord_no=f.ord_no;
```

Results Explain Describe Saved SQL History

W_NAME
Michael
Michael
Emily
David
Sophia

5 rows returned in 0.00 seconds CSV Export

Fig: results

3.Show the foods which is order by customers

```
select f.description,f.food_no, w.cus_id
from food f, waiter w
where f.ord_no=w.ord_no;
```

Results Explain Describe Saved SQL History

DESCRIPTION	FOOD_NO	CUS_ID
Vegetable Fried Rice	2	1
Chicken Curry	1	1
Sushi Platter	3	2
Taco Combo	4	3
Margherita Pizza	5	4

5 rows returned in 0.00 seconds CSV Export

4.Right join

```
SELECT c.cus_name, r.r_name
FROM customer c
RIGHT JOIN restaurant r ON c.r_id = r.r_id;
```

Results | Explain | Describe | Saved SQL | History

CUS_NAME	R_NAME
Hamim	The Hungry Panda
Farhana	The Hungry Panda
Borhan	The Hungry Panda
Hashamee	The Hungry Panda
Mahir	The Hungry Panda
-	Burger Haven
-	La Pizzeria
-	Taco Town
-	Sushi Samurai
-	Italiano Ristorante

10 rows returned in 0.00 seconds CSV Export

Simple View—

1.Create a simple view to display names and contact number of customer

```
CREATE VIEW customer_view AS
SELECT cus_name, contact_n
FROM customer;
```

Fig: simple view creation commant

Object Type **VIEW** Object **CUSTOMER_VIEW**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_VIEW	CUS_NAME	Varchar2	20	-	-	-	✓	-	-
	CONTACT_N	Number	-	11	0	-	✓	-	-

1 - 2

Fig: description of the simple view

Results | Explain | Describe | Saved SQL | History

CUS_NAME	CONTACT_N
Hamim	1234567890
Farhana	9876543210
Borhan	5551234567
Hashamee	9998887776
Mahir	4445556667

5 rows returned in 0.00 seconds CSV Export

Fig: result of simple view

Complex view—

1. Create a complex view named order_detailsview tha show order number, customer name, restaurant name, food description and quantity

```
CREATE VIEW order_detailsview AS
SELECT w.ord_no, c.cus_name, r.r_name, f.description, f.quantity
FROM waiter w
JOIN customer c ON w.cus_id = c.cus_id
JOIN restaurant r ON c.r_id = r.r_id
JOIN food f ON w.ord_no = f.ord_no;
```

Fig: complex view creation command

Object Type **VIEW** Object **ORDER_DETAILSVIEW**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDER_DETAILSVIEW	ORD_NO	Number	-	5	0	-	✓	-	-
	CUS_NAME	Varchar2	20	-	-	-	✓	-	-
	R_NAME	Varchar2	20	-	-	-	✓	-	-
	DESCRIPTION	Varchar2	20	-	-	-	✓	-	-
	QUANTITY	Number	-	5	0	-	✓	-	-

1 - 5

Fig: description of the complex view.

Results Explain Describe Saved SQL History

ORD_NO	CUS_NAME	R_NAME	DESCRIPTION	QUANTITY
4	Hashamee	The Hungry Panda	Margherita Pizza	2
3	Borhan	The Hungry Panda	Taco Combo	1
2	Farhana	The Hungry Panda	Sushi Platter	3
1	Hamim	The Hungry Panda	Vegetable Fried Rice	1
1	Hamim	The Hungry Panda	Chicken Curry	2

5 rows returned in 0.00 seconds CSV Export

Fig: Compex view results

Conclusion

In conclusion, the project has successfully developed a relational database schema for restaurant management, facilitating efficient operations through tables like customer, restaurant, order, food, and waiter, and views such as customer_info and order_detailsview. Leveraging SQL queries, data manipulation tasks were executed seamlessly. Looking forward, enhancements include advanced reporting for sales trends and customer preferences, integration with external systems for streamlined operations, adoption of data analysis tools for informed decision-making, and continuous performance optimization efforts for scalability and efficiency. These developments aim to empower the restaurant with greater adaptability and improved customer experiences.