

Introduction

A database management system (DBMS) is a system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

Our project pastry shop management system was created by using the concept of DBMS. A pastry shop is a confectionery shop that specializes in baking pastries for its clients. Foods that are served here are not meant to satisfy the greater hunger but to satisfy the tastebuds of the foodies with a sweet tooth. A pastry shop which combines dazzling environment with wonderful cuisine, is the ideal venue for frequent hangouts and personal meetups.

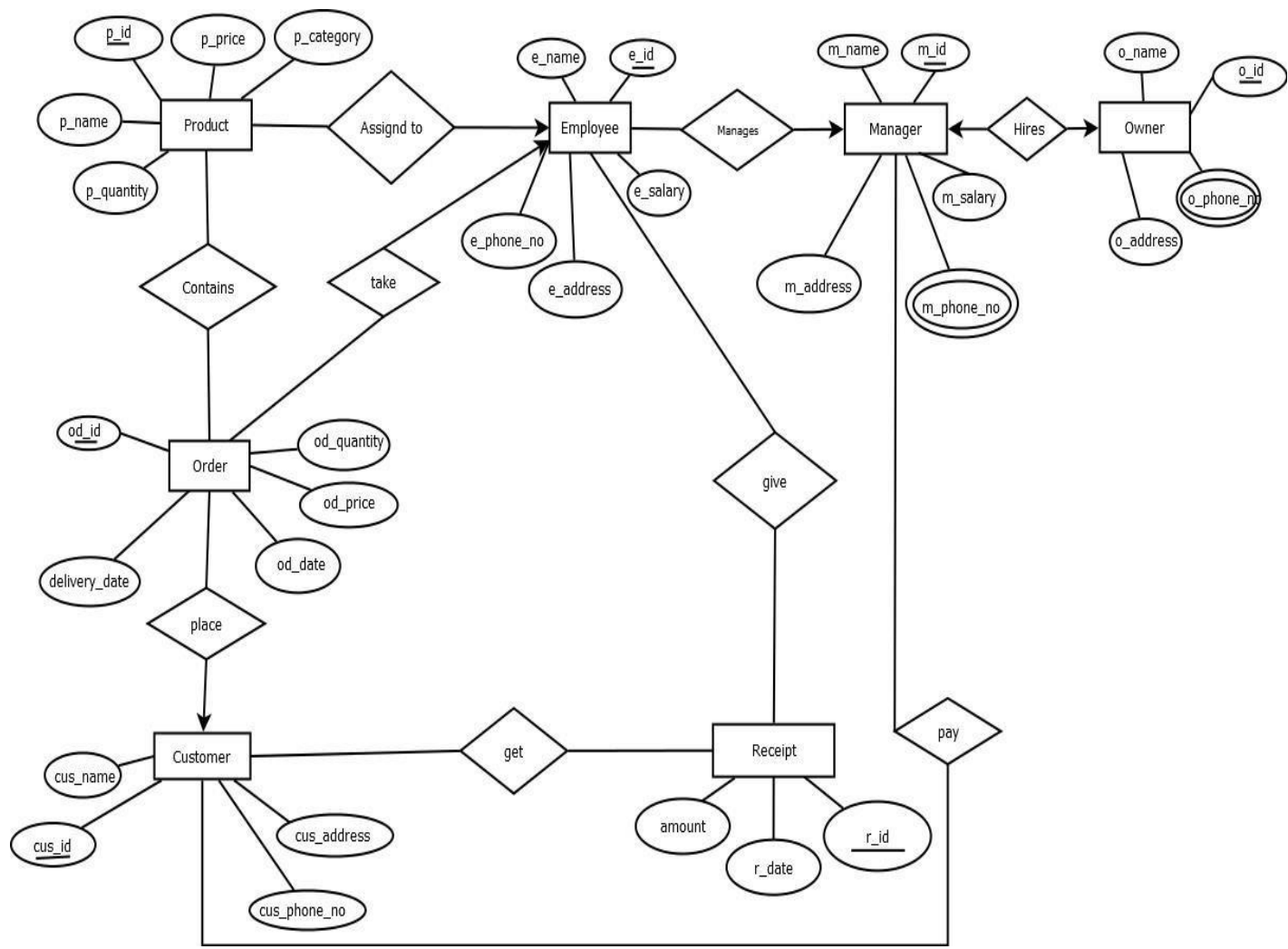
The project “Pastry Shop Management System” is designed to record the information of the manager, employees, customers and products. It is built to facilitate and to make the best use of data by keeping records in database. Through this project, we can easily deal with the daily management system of a pastry shop. For this project we created some tables. We drew an ER diagram showing the tables and their work as an overview. The basic aim of the project is to record, search, delete and insert data using SQL statements. Normalization and schema diagram are also the key elements of this project. Basically, the purpose of the project is to handle all the managements of a pastry shop.

Scenario Description

Baker Boys, a pastry shop located near Brooklyn Nine-Nine uses an integrated pastry shop management system which holds records of products made by the pastry shop. The management system of pastry shop stores owner’s name, address and multiple

phone numbers. The owner of pastry shop hires one manager who manages the employees. Manager can hire many employees. The manager is identified by Unique id, name, address, salary. Multiple phone number of manager is also stored. Every employee has unique id, name, phone number, salary, address. One Employee assigned to many products. To identify a product the system also stores product unique id, name, price, category, quantity. One Customer can place many orders. Customer has unique id, name, address, phone number. The Order have different id, quantity, price. Also order dates and delivery dates are stored. Many orders contain many products. One Employee can take many orders and give a receipt to the customers. Receipt has unique id, amount, date. Customers get the receipts and pay the manager. That's all the information that pastry shop management system uses on a daily basis.

ER Diagram



Normalization

Hires UNF

hires(o_id,o_name,o_address,o_phone_no,m_name,m_id,m_salary,m_address,m_phoneno) 1NF o_phone_no,m_phoneno are multiple valued attributes.

1.o_name,o_address,o_phoneno,m_name,m_id,m_salary,m_address,m_phone_no

2NF

1. o_id,o_name,o_address,o_phone_no

2.m_name,m_id,m_salary,m_address,m_phoneno

3NF

There is no transitive dependency. Relation already in 3NF form.

1. o_id,o_name,o_address,o_phone_no

2.m_name,m_id,m_salary,m_address,m_phoneno.

Table creation

1. o_id,o_name,o_address,o_phone_no,m_id

2.m_name,m_id,m_salary,m_address,m_phoneno.

Manages

UNF

Manages(m_name,m_id,m_salary,m_address,m_phone_no,e_name,e_id,e_salary,e_address,e_phoneno)

1NF m_phone_no is a multiple valued

attribute.

1.m_name,m_id,m_salary,m_address,m_phone_no,e_name,e_id,e_salary,e_address,e_phoneno

2NF

1.m_name,m_id,m_salary,m_address,m_phoneno.

2.e_name,e_id,e_salary,e_address,e_phoneno

3NF

There is no transitive dependency. Relation already in 3NF form.

1.m_name,m_id,m_salary,m_address,m_phone_no.

2.e_name,e_id,e_salary,e_address,e_phone_no

Table creation

1.m_name,m_id,m_salary,m_address,m_phoneno.

2.e_name,e_id,e_salary,e_address,e_phoneno,**m_id**

Assigned to

UNF

Assigned to(e_name,e_id,e_salary,e_address,e_phone_no,p_id,p_name,p_price,p_category, p_quantity)

1NF

There is no multivalued attribute.

1.e_name,e_id,e_salary,e_address,e_phone_no,p_id,p_name,p_price,p_category,p_quantity

2NF

1.e_name,e_id,e_salary,e_address,e_phone_no

2.p_id,p_name,p_price,p_category,p_quantity

3NF

There is no transitive dependency. Relation already in 3NF form.

1.e_name,e_id,e_salary,e_address,e_phone_no

2.p_id,p_name,p_price,p_category,p_quantity

Table creation

1.e_name,e_id,e_salary,e_address,e_phoneno

2.p_id,p_name,p_price,p_category,p_quantity,**e_id**

Contain

UNF

Contains(p_id,p_name,p_price,p_category,p_quantity,od_id,
od_quantity,od_price,od_date,delivery_date)

1NF

There is no multivalued attribute.

1. p_id,p_name,p_price,p_category,p_quantity,od_id,
od_quantity,od_price,od_date,delivery_date 2NF

1. p_id,p_name,p_price,p_category,p_quantity
2. od_id, od_quantity,od_price,od_date,delivery_date

3NF

1. p_id,p_name,p_price,p_category,p_quantity
2. od_id, od_date,delivery_date
3. od_quantity,od_price

Table Creation

1. p_id,p_name,p_price,p_category,p_quantity,
2. od_id, od_date,delivery_date,uo_id
3. od_quantity,od_price,uo_id
4. od_id,p_id

Place

UNF

Place(od_id, od_quantity,od_price,od_date,delivery_date,cus_id,
cus_name,cus_address,cus_phone_no)

1NF

There is no multivalued attribute.

1. od_id, od_quantity, od_price, od_date, delivery_date, cus_id,
cus_name, cus_address, cus_phone_no

2NF

1. od_id, od_quantity, od_price, od_date, delivery_date
2. cus_id, cus_name, cus_address, cus_phone_no

3NF

1. cus_id, cus_name, cus_address, cus_phone_no
2. od_id, od_date, delivery_date
3. od_quantity, od_price

Table Creation

1. cus_id, cus_name, cus_address, cus_phone_no
2. od_id, od_date, delivery_date, uo_id, cus_id
3. od_quantity, od_price, uo_id

Take

UNF

Take(e_name, e_id, e_salary, e_address, e_phone_no, od_id,
od_quantity, od_price, od_date, delivery_date)

1NF

There is no multivalued attribute.

1. e_name, e_id, e_salary, e_address, e_phone_no, od_id,
od_quantity, od_price, od_date, delivery_date

2NF

1. e_name, e_id, e_salary, e_address, e_phone_no
2. od_id, od_quantity, od_price, od_date, delivery_date

3NF

1. e_name, e_id, e_salary, e_address, e_phone_no
2. od_id, od_date, delivery_date

3. od_quantity,od_price

Table Creation

1. e_name,e_id,e_salary,e_address,e_phone_no
2. od_id,od_date,delivery_date,uo_id , e_id
3. od_quantity,od_price,uo_id

Give

UNF

Give(amount,r_id, r_date, e_name,e_id,e_salary,e_address,e_phone_no) 1NF

There is no multivalued attribute.

1. amount,r_id, r_date, e_name,e_id,e_salary,e_address,e_phone_no

2NF

1. amount,r_id,r_date
2. e_name,e_id,e_salary,e_address,e_phone_no

3NF

There is no transitive dependency. Relation already in 3NF form.

1. amount,r_id, r_date
2. e_name,e_id,e_salary,e_address,e_phone_no

Table Creation

1. amount,r_id, r_date
2. e_name,e_id,e_salary,e_address,e_phone_no
3. r_id , e_id

Get

UNF

Get(amount,r_id, r_date,cus_id , cus_name.cus_address,cus_phone_no) 1NF

There is no multivalued attribute

1. amount,r_id, r_date,cus_id , cus_name.cus_address,cus_phone_no

2NF

1. amount, r_id, r_date
2. cus_id , cus_name.cus_address, cus_phone_no

3NF

There is no transitive dependency. Relation already in 3NF form.

1. amount, r_id, r_date
2. cus_id , cus_name.cus_address, cus_phone_no Table

Creation

1. amount, r_id, r_date
2. cus_id , cus_name.cus_address, cus_phone_no
3. cus_id , r_id

Pay

UNF

Pay(m_name, m_id, m_salary, m_address, m_phone_no, cus_id ,
cus_name.cus_address, cus_phone_no)

1NF

m_phone_no is a multivalued attribute.

1. m_name, m_id, m_salary, m_address, m_phone_no, cus_id ,
cus_name.cus_address, cus_phone_no

2NF

1. m_name, m_id, m_salary, m_address, m_phone_no
2. cus_id , cus_name.cus_address, cus_phone_no

3NF

There is no transitive dependency. Relation already in 3NF form.

1. m_name,m_id,m_salary,m_address,m_phone_no
2. cus_id , cus_name.cus_address,cus_phone_no

Table Creation

1. m_name,m_id,m_salary,m_address,m_phone_no
2. cus_id , cus_name.cus_address,cus_phone_no
3. **m_id ,cus_id** Temporary Table:
 1. o_id,o_name,o_address,o_phone_no,m_id
 - 2.~~m_name,m_id,m_salary,m_address,m_phoneno.~~
 - 3.~~m_name,m_id,m_salary,m_address,m_phoneno.~~
 - 4.e_name,e_id,e_salary,e_address,e_phoneno,**m_id**
 - 5.e_name,e_id,e_salary,e_address,e_phoneno
 - 6.p_id,p_name,p_price,p_category,p_quantity,**e_id**
 7. p_id,p_name,p_price,p_category,p_quantity
 8. od_id,od_date,delivery_date,**uo_id** , **e_id**
 - 9.~~od_quantity,od_price,uo_id~~
 - 10.cus_id , cus_name.cus_address,cus_phone_no
 - 11.od_id,od_date,delivery_date,**uo_id**, **cus_id**
 - 12.~~od_quantity,od_price,uo_id~~
 - 13.~~e_name,e_id,e_salary,e_address,e_phone_no~~
 - 14.~~od_id,od_date,delivery_date,uo_id~~
 - 15.~~od_quantity,od_price,uo_id~~
 - 17.~~amount,r_id, r_date~~
 - 18.~~e_name,e_id,e_salary,e_address,e_phone_no~~
 - 19.**r_id** , **e_id**
 20. amount,r_id, r_date
 - 21.~~cus_id , cus_name.cus_address,cus_phone_no~~

22.cus_id , r_id

23.m_name,m_id,m_salary,m_address,m_phone_no

24.cus_id , cus_name.cus_address,cus_phone_no

25. m_id , cus_id

26.od_id , p_id

Final Table

1. o_id,o_name,o_address,o_phone_no, m_id

2.m_name,m_id,m_salary,m_address,m_phoneno.

3.e_name,e_id,e_salary,e_address,e_phoneno, m_id

4.p_id,p_name,p_price,p_category,p_quantity, e_id

5.cus_id , cus_name.cus_address,cus_phone_no

6.od_id ,od_date,delivery_date, uo_id , cus_id , e_id

7.od_quantity,od_price,uo_id

8.r_id , e_id

9. amount,r_id, r_date

10.cus_id , r_id

11. m_id , cus_id

12.od_id , p_id

Schema Diagram

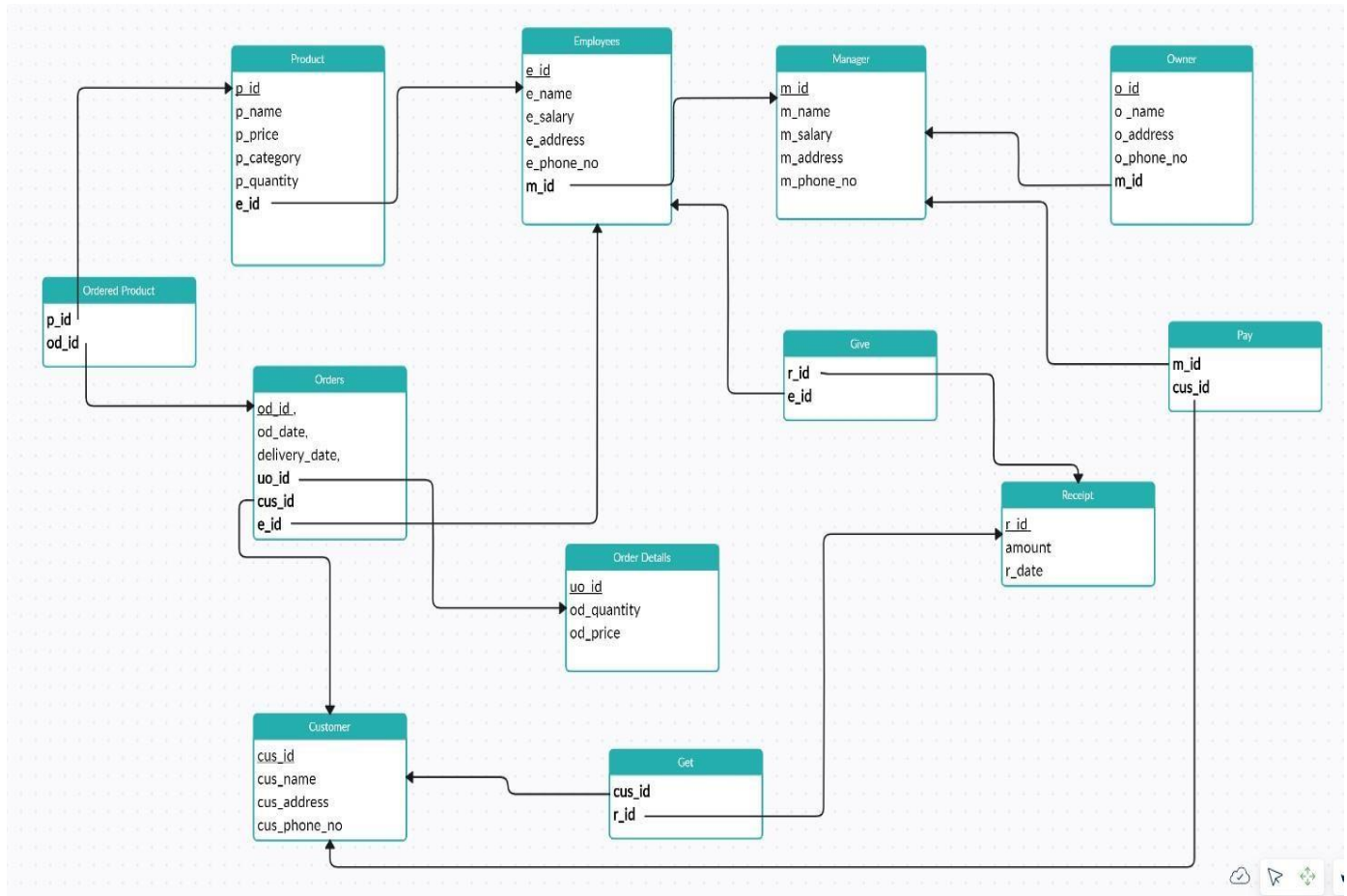


Table Creation

Create table:

- 1.create table owner(o_id number primary key, o_name varchar2(30), o_address varchar2(30), o_phone_no number(11), m_id number);
- 2.create table manager(m_id number primary key, m_name varchar2(30), m_salary number, m_address varchar2(30), m_phone_no number(11));
- 3.create table employee(e_id number primary key, e_name varchar2(30), e_salary number, e_address varchar2(30), e_phone_no number(11), m_id number, constraint chk_salary check(e_salary > 1000));
- 4.create table product(p_id number primary key, p_name varchar2(30), p_price number, p_category varchar2(30), p_quantity number, e_id number);
- 5.create table customer(cus_id number primary key, cus_name varchar2(30), cus_address varchar2(30), cus_phone_no number(11));
- 6.create table orders(od_id number primary key, od_date date, delivery_date date, uo_id number,cus_id number, e_id number);
- 7.create table order_details(uo_id number primary key, od_quantity number, od_price number);
- 8.create table give(r_id number, e_id number);
- 9.create table receipt(r_id number primary key, amount number, r_date date);
- 10.create table get(cus_id number, r_id number);
- 11.create table pay(m_id number, cus_id number);
12. create table ordered_product(p_id number, od_id number);

Altering tables:

alter table owner add foreign key(m_id) references manager(m_id);

alter table employee add foreign key(m_id) references manager(m_id);

alter table product add foreign key(e_id) references employee(e_id);

```
alter table orders add foreign key(uo_id) references order_details(uo_id); alter
table orders add foreign key(cus_id) references customer(cus_id); alter table orders
add foreign key(e_id) references employee(e_id);
```

```
alter table give add foreign key(r_id) references receipt(r_id); alter table
give add foreign key(e_id) references employee(e_id);
```

```
alter table get add foreign key(cus_id) references customer(cus_id); alter table
get add foreign key(r_id) references receipt(r_id);
```

```
alter table pay add foreign key(m_id) references manager(m_id); alter table
pay add foreign key(cus_id) references customer(cus_id);
```

```
alter table ordered_product add foreign key(p_id) references product(p_id);
alter table ordered_product add foreign key(od_id) references orders(od_id);
screenshots of the created table using describe command:
```

OWNER Table:

Object Type	TABLE	Object	OWNER						
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
OWNER	O_ID	Number	-	-	-	1	-	-	-
	O_NAME	Varchar2	30	-	-	-	✓	-	-
	O_ADDRESS	Varchar2	30	-	-	-	✓	-	-
	O_PHONE_NO	Number	-	11	0	-	✓	-	-
	M_ID	Number	-	-	-	-	✓	-	-
									1 - 5

MANAGER Table:

Object Type **TABLE** Object **MANAGER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>MANAGER</u>	<u>M_ID</u>	Number	-	-	-	1	-	-	-
	<u>M_NAME</u>	Varchar2	30	-	-	-	✓	-	-
	<u>M_SALARY</u>	Number	-	-	-	-	✓	-	-
	<u>M_ADDRESS</u>	Varchar2	30	-	-	-	✓	-	-
	<u>M_PHONE_NO</u>	Number	-	11	0	-	✓	-	-
1 - 5									

EMPLOYEE Table:

Object Type **TABLE** Object **EMPLOYEE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>EMPLOYEE</u>	<u>E_ID</u>	Number	-	-	-	1	-	-	-
	<u>E_NAME</u>	Varchar2	30	-	-	-	✓	-	-
	<u>E_SALARY</u>	Number	-	-	-	-	✓	-	-
	<u>E_ADDRESS</u>	Varchar2	30	-	-	-	✓	-	-
	<u>E_PHONE_NO</u>	Number	-	11	0	-	✓	-	-
	<u>M_ID</u>	Number	-	-	-	-	✓	-	-
1 - 6									

PRODUCT Table:

Object Type **TABLE** Object **PRODUCT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>PRODUCT</u>	<u>P_ID</u>	Number	-	-	-	1	-	-	-
	<u>P_NAME</u>	Varchar2	30	-	-	-	✓	-	-
	<u>P_PRICE</u>	Number	-	-	-	-	✓	-	-
	<u>P_CATEGORY</u>	Varchar2	30	-	-	-	✓	-	-
	<u>P_QUANTITY</u>	Number	-	-	-	-	✓	-	-
	<u>E_ID</u>	Number	-	-	-	-	✓	-	-
1 - 6									

CUSTOMER Table:

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

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>CUSTOMER</u>	<u>CUS_ID</u>	Number	-	-	-	1	-	-	-
	<u>CUS_NAME</u>	Varchar2	30	-	-	-	✓	-	-
	<u>CUS_ADDRESS</u>	Varchar2	30	-	-	-	✓	-	-
	<u>CUS_PHONE_NO</u>	Number	-	11	0	-	✓	-	-
1 - 4									

ORDERS Table:

Object Type TABLE Object ORDERS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>ORDERS</u>	<u>OD ID</u>	Number	-	-	-	1	-	-	-
	<u>OD DATE</u>	Date	7	-	-	-	✓	-	-
	<u>DELIVERY DATE</u>	Date	7	-	-	-	✓	-	-
	<u>UO ID</u>	Number	-	-	-	-	✓	-	-
	<u>CUS ID</u>	Number	-	-	-	-	✓	-	-
	<u>E ID</u>	Number	-	-	-	-	✓	-	-
1 - 6									

ORDER_DETAILS Table:

Object Type TABLE Object ORDER_DETAILS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>ORDER_DETAILS</u>	<u>UO ID</u>	Number	-	-	-	1	-	-	-
	<u>OD QUANTITY</u>	Number	-	-	-	-	✓	-	-
	<u>OD PRICE</u>	Number	-	-	-	-	✓	-	-
1 - 3									

RECEIPT Table:

Object Type TABLE Object RECEIPT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>RECEIPT</u>	<u>R ID</u>	Number	-	-	-	1	-	-	-
	<u>AMOUNT</u>	Number	-	-	-	-	✓	-	-
	<u>R DATE</u>	Date	7	-	-	-	✓	-	-
1 - 3									

GIVE Table:

Object Type TABLE Object GIVE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>GIVE</u>	<u>R ID</u>	Number	-	-	-	-	✓	-	-
	<u>E ID</u>	Number	-	-	-	-	✓	-	-
1 - 2									

GET Table:

Object Type TABLE Object GET

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
GET	CUS_ID	Number	-	-	-	-	✓	-	-
	R_ID	Number	-	-	-	-	✓	-	-
1 - 2									

PAY Table:

Object Type TABLE Object PAY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PAY	M_ID	Number	-	-	-	-	✓	-	-
	CUS_ID	Number	-	-	-	-	✓	-	-
1 - 2									

ORDERED PRODUCT Table:

Object Type TABLE Object ORDERED_PRODUCT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDERED_PRODUCT	P_ID	Number	-	-	-	-	✓	-	-
	OD_ID	Number	-	-	-	-	✓	-	-
1 - 2									

Create Sequence:

create sequence seq increment by 1 start with 1 nocache nocycle;

User creation:

create user Safin identified by s1a2f3i4n; grant

unlimited tablespace to Safin; **assign**

roles:

create role manager; grant create table, create view, create sequence, create user, create

role to manager; grant manager to Safin;

Data Insertion

OWNER DATA:

insert into owner(o_id, o_name, o_address, o_phone_no, m_id) values(101, 'Mr. O', 'Kuratoli, Dhaka', 01633043297, 102);

O_ID	O_NAME	O_ADDRESS	O_PHONE_NO	M_ID
101	Mr. O	Kuratoli, Dhaka	1633043297	102

MANAGER DATA:

insert into manager(m_id, m_name, m_salary, m_address, m_phone_no) values(102, 'Mr. M', 30000, 'Banani, Dhaka', 01521754978);

M_ID	M_NAME	M_SALARY	M_ADDRESS	M_PHONE_NO
102	Mr. M	30000	Banani, Dhaka	1521754978

EMPLOYEE DATA:

insert into employee(e_id, e_name, e_salary, e_address, e_phone_no, m_id) values(201, 'Abul', 7000, 'Azimpur, Dhaka', 01602700367, 102);

insert into employee(e_id, e_name, e_salary, e_address, e_phone_no, m_id) values(202, 'Babul', 9000, 'Kamlapur, Dhaka', 01802300467, 102);

insert into employee(e_id, e_name, e_salary, e_address, e_phone_no, m_id) values(203, 'Cabul', 8000, 'Shahbag, Dhaka', 01302000357, 102);

insert into employee(e_id, e_name, e_salary, e_address, e_phone_no, m_id) values(204, 'Dabul', 5000, 'Farmagate, Dhaka', 01712790367, 102);

insert into employee(e_id, e_name, e_salary, e_address, e_phone_no, m_id) values(205, 'Ebul', 6000, 'Mohammadpur, Dhaka', 01902800367, 102);

E_ID	E_NAME	E_SALARY	E_ADDRESS	E_PHONE_NO	M_ID
201	Abul	7000	Azimpur, Dhaka	1302700367	102
202	Babul	9000	Kamlapur, Dhaka	1802300467	102
203	Cabul	8000	Shahbag, Dhaka	1302000357	102
204	Dabul	5000	Farmagate, Dhaka	1712790367	102
205	Ebul	6000	Mohammadpur, Dhaka	1902800367	102

CUSTOMER DATA:

insert into customer(cus_id, cus_name, cus_address, cus_phone_no) values(301, 'Shaad', 'Mirpur, Dhaka', 01935728398);

insert into customer(cus_id, cus_name, cus_address, cus_phone_no) values(302, 'Asif', 'Rampura, Dhaka', 01843311815);

insert into customer(cus_id, cus_name, cus_address, cus_phone_no) values(303, 'Safin', 'kuratoli, Dhaka',

01876266962);

insert into customer(cus_id, cus_name, cus_address, cus_phone_no) values(304, 'Maria', 'Bashundhara, Dhaka', 01835538098);

insert into customer(cus_id, cus_name, cus_address, cus_phone_no) values(305, 'Sakib', 'Gulshan, Dhaka', 01625347735);

CUS_ID	CUS_NAME	CUS_ADDRESS	CUS_PHONE_NO
301	Shaad	Mirpur, Dhaka	1935728398
302	Asif	Rampura, Dhaka	1843311815
303	Safin	kuratoli, Dhaka	1876266962
304	Maria	Bashundhara, Dhaka	1835538098
305	Sakib	Gulshan, Dhaka	1625347735

ORDERS DATA:

insert into orders(od_id, od_date, delivery_date, uo_id, cus_id, e_id) values(601, '1-JAN-2022', '1-JAN-2022', 701, 301, 201);

insert into orders(od_id, od_date, delivery_date, uo_id, cus_id, e_id) values(602, '10-JAN-2022', '10-JAN-2022', 702, 302, 202);

insert into orders(od_id, od_date, delivery_date, uo_id, cus_id, e_id) values(603, '3-NOV-2022', '3-NOV-2022', 703, 303, 202);

insert into orders(od_id, od_date, delivery_date, uo_id, cus_id, e_id) values(604, '5-MAR-2022', '5-MAR-2022', 704, 304, 203);

insert into orders(od_id, od_date, delivery_date, uo_id, cus_id, e_id) values(605, '6-AUG-2022', '6AUG2022', 705, 305, 205);

OD_ID	OD_DATE	DELIVERY_DATE	UO_ID	CUS_ID	E_ID
601	01-JAN-22	01-JAN-22	701	301	201
602	10-JAN-22	10-JAN-22	702	302	202
603	03-NOV-22	03-NOV-22	703	303	202
604	05-MAR-22	05-MAR-22	704	304	203
605	06-AUG-22	06-AUG-22	705	305	205

ORDER_DETAILS DATA:

insert into order_details(uo_id, od_quantity, od_price) values(701, 2, 260); insert into

order_details(uo_id, od_quantity, od_price) values(702, 1, 2400); insert into

order_details(uo_id, od_quantity, od_price) values(703, 3, 1080); insert into

order_details(uo_id, od_quantity, od_price) values(704, 1, 2700); insert into

order_details(uo_id, od_quantity, od_price) values(705, 3, 2040);

UO_ID	OD_QUANTITY	OD_PRICE
701	2	260
702	1	2400
703	3	1080
704	1	2700
705	3	2040

PRODUCT DATA:

insert into product(p_id, p_name, p_price, p_category, p_quantity, e_id) values(401, 'French toast', 130, 'Bakery item', 44, 201);

insert into product(p_id, p_name, p_price, p_category, p_quantity, e_id) values(402, 'Tiamaria cake', 2400, 'Cakes', 13, 202);

insert into product(p_id, p_name, p_price, p_category, p_quantity, e_id) values(403, 'Fudge cake', 2700, 'Cakes', 9, 203);

insert into product(p_id, p_name, p_price, p_category, p_quantity, e_id) values(404, 'Box of bon bons', 680, 'Chocolates', 17, 204);

insert into product(p_id, p_name, p_price, p_category, p_quantity, e_id) values(405, 'Savoury', 360, 'Deserts', 23, 205);

P_ID	P_NAME	P_PRICE	P_CATEGORY	P_QUANTITY	E_ID
401	French toast	130	Bakery item	44	201
402	Tiamaria cake	2400	Cakes	13	202
403	Fudge cake	2700	Cakes	9	203
404	Box of bon bons	680	Chocolates	17	204
405	Savoury	360	Deserts	23	205

RECEIPT DATA:

insert into receipt(r_id, amount, r_date) values(501, 260, '1-JAN-2022'); insert

into receipt(r_id, amount, r_date) values(502, 2400, '10-JAN-2022'); insert into

receipt(r_id, amount, r_date) values(503, 1080, '3-NOV-2022'); insert into

receipt(r_id, amount, r_date) values(504, 2700, '5-MAR-2022');

insert into receipt(r_id, amount, r_date) values(505, 2040, '6-AUG-2022');

R_ID	AMOUNT	R_DATE
501	260	01-JAN-22
502	2400	10-JAN-22
503	1080	03-NOV-22
504	2700	05-MAR-22
505	2040	06-AUG-22

GIVE Table DATA:

```
insert into give(r_id, e_id) values(501, 201); insert
into give(r_id, e_id) values(502, 202); insert into
give(r_id, e_id) values(503, 202); insert into
give(r_id, e_id) values(504, 203); insert into
give(r_id, e_id) values(505, 205);
```

R_ID	E_ID
501	201
502	202
503	202
504	203
505	205

GET Table DATA:

```
insert into get(cus_id, r_id) values(301, 501); insert
into get(cus_id, r_id) values(302, 502); insert into
get(cus_id, r_id) values(303, 503); insert into
```

```
get(cus_id, r_id) values(304, 504); insert into  
get(cus_id, r_id) values(305, 505);
```

CUS_ID	R_ID
301	501
302	502
303	503
304	504
305	505

PAY Table DATA:

```
insert into pay(m_id, cus_id) values(102, 301); insert  
into pay(m_id, cus_id) values(102, 302); insert into  
pay(m_id, cus_id) values(102, 303); insert into pay(m_id,  
cus_id) values(102, 304); insert into  
pay(m_id, cus_id) values(102, 305);
```

M_ID	CUS_ID
102	301
102	302
102	303
102	304
102	305

ORDERED PRODUCT Table DATA:

```
insert into ordered_product(p_id, od_id) values(401, 601); insert  
into ordered_product(p_id, od_id) values(402, 602); insert into  
ordered_product(p_id, od_id) values(403, 603); insert into  
ordered_product(p_id, od_id) values(404, 604); insert into  
ordered_product(p_id, od_id) values(405, 605);
```

P_ID	OD_ID
401	601
402	602
403	603
404	604
405	605

Query Writing

Single row function: Single row functions are the one who work on single row and return one output per row. For example, length and case conversion functions are single row functions.

1. Display the E_NAME and E_SALARY number by joining the columns using concatenation function.

Ans: select concat(E_NAME, E_SALARY) as Employee_info from employee;

EMPLOYEE_INFO
Abul7000
Babul9000
Cabul8000
Dabul5000
Ebul6000

2. Display Today's date and time.

Ans: select sysdate from dual;

SYSDATE
25-DEC-22

- **Group function:** Group functions are built-in SQL functions that operate on groups of rows and return one value for the entire group. These functions are: COUNT, MAX, MIN, AVG, SUM, DISTINCT

1. Display maximum salary from all employees using MAX function.

Ans: select MAX(E_SALARY) from employee;

MAX(E_SALARY)
9000

2. Display the sum of products price using the SUM function.

Ans: select SUM(p_price) from product;

SUM(P_PRICE)
6270

Subquery: A subquery is a query that is nested inside a SELECT, INSERT, UPDATE, or DELETE statement, or inside another subquery. A subquery can be used anywhere an expression is allowed. In this example, a subquery is used as a column expression named MaxUnitPrice in a SELECT statement.

1. Display the products name and price where price is greater than Box of bon bons.

Ans: select p_name, p_price from product where p_price > (select p_price from product where p_name='Box of bon bons');

P_NAME	P_PRICE
Tiamaria cake	2400
Fudge cake	2700

2. Display the name and salary of employees where salary is higher than the salary of Dabul.

Ans: select e_name, e_salary from employee where e_salary > (select e_salary from employee where e_name = 'Dabul');

E_NAME	E_SALARY
Abul	7000
Babul	9000
Cabul	8000
Ebul	6000

Joining: A SQL Join is a special form of generating a meaningful data by combining multiple tables relate to each other using a “Key”. Typically, relational tables must be designed with a unique column and this column is used to create relationships with one or more other tables. When need a result-set that includes related rows from multiple tables, need to use SQL join on this column

1. Join the table orders and order details where uo_id of both table are same.

Ans: select orders.od_date, orders.delivery_date, order_details.od_quantity, order_details.od_price, orders.uo_id, order_details.uo_id from orders, order_details where orders.uo_id = order_details.uo_id;

OD_DATE	DELIVERY_DATE	OD_QUANTITY	OD_PRICE	UO_ID	UO_ID
01-JAN-22	01-JAN-22	2	260	701	701
10-JAN-22	10-JAN-22	1	2400	702	702
03-NOV-22	03-NOV-22	3	1080	703	703
05-MAR-22	05-MAR-22	1	2700	704	704
06-AUG-22	06-AUG-22	3	2040	705	705

2. Join the table orders and employee where e_id of both table are same

Ans: select employee.e_name, employee.e_id, employee.e_address, orders.od_id from employee, orders where orders.e_id = employee.e_id;

E_NAME	E_ID	E_ADDRESS	OD_ID
Abul	201	Azimpur, Dhaka	601
Babul	202	Kamlapur, Dhaka	602
Babul	202	Kamlapur, Dhaka	603
Cabul	203	Shahbag, Dhaka	604
Ebul	205	Mohammadpur, Dhaka	605

View: In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database. Add SQL statements and functions to a view and present the data as if the data were coming from one single table.

A view is created with the **CREATE VIEW** statement.

1. Create a view of product where price is more than 1500 tk.

Ans: create view price_view as select p_name, p_price from product where p_price > 1500;

P_NAME	P_PRICE
Tiamaria cake	2400
Fudge cake	2700

2. Create a view of employee salary where salary is greater than 5000 tk;

Ans: create view emp_salary as select e_name, e_salary from employee where e_salary > 5000;

E_NAME	E_SALARY
Abul	7000
Babul	9000
Cabul	8000
Ebul	6000

Relational Algebra

1. Find the name of the customer where customer id is 302

Ans: $\pi_{\text{Cus_name}}(\sigma_{\text{cus_id} = '302'}(\text{customer}))$

2. Find the customer address where customer phone number is 1843311815

Ans: $\pi_{\text{Cus_address}}(\sigma_{\text{cus_phone_no} = '1843311815'}(\text{customer}))$

3. Find the receipt id where receipt date is '10-JAN-2022'

Ans: $\pi_{\text{r_id}}(\sigma_{\text{r_date} = '10-JAN-2022'}(\text{receipt}))$

4. Find the names of the employees who earn more than 6000

Ans: $\pi_{\text{e_name}}(\sigma_{\text{e_salary} > 6000}(\text{employee}))$

5. Find order id where delivery date is '5-MAR-2022'

Ans: $\pi_{\text{o_id}}(\sigma_{\text{delivery_date} = '5-MAR-2022'}(\text{employee}))$

Conclusion

The pastry shop management system project's scenario Gives off a brief overview of the whole project. IT gives a generic idea of what the entities are and about their attributes. The main essence of the project Is found in the ER diagram and before table creation, normalization was done to

avoid redundancy. A schema diagram was created to identify the related data sets, the predesigned tables were then created based on related data sets and finally, on the insertion part data was being stored. Some Queries and relational algebra were applied on the implemented data base to test the Management system's data integrity, which proved to be successful. By this effective database project, the organizational data accessibility will be lot quicker and efficient. Both the customers and the shop will be able to save their time and effort by using this complete pasty shop management system. The project is planned to improve by ensuring more security to protect the data. Moreover , the project is made in such a way that further customization can be added effortlessly in the already existing database.