

American International University-Bangladesh

# Faculty of Science and Technology

# Department of Computer Science

### Introduction to Database [H]

A project in

Pastry Shop Management System

#### Prepared By

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **CONTRIBUTION** |
| 20-44206-3 | TONMOY DEY | ER Diagram, Table creation |
| 20-44346-3 | MD. HAJJAJ BIN SONOSI | Data insertion, Query writing |
| 20-44230-3 | ANAMUL HAQUE SHADIN | Introduction, conclusion |
| 20-44216-3 | SHAKIL MAHMUD | Scenario Description, Cover Page |
|  |  |  |

Supervised By

NAZIA ALFAZ

Faculty of Science and Technology American International University-Bangladesh

Dhaka, Bangladesh.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **TOPICS** | ***Page no*.** |
| **I. Title Page** | 1 |
| **II. Table of Content** | 2 |
| **1. Introduction** | 4 |
| **2. Scenario Description** | 4 |
| **3. ER diagram (MIDTERM)** | 5 |
| **4. ER diagram (FINALTERM)** | 6 |
| **5. Newly Added** | 7 |
| **6. Normalization** | 8 |
| **7. Table creation** | 9-15 |
| **8. Data insertion** | 16-29 |
| **9. Query Writing**  **10. Conclusion** | 30-40  41 |
|  |  |

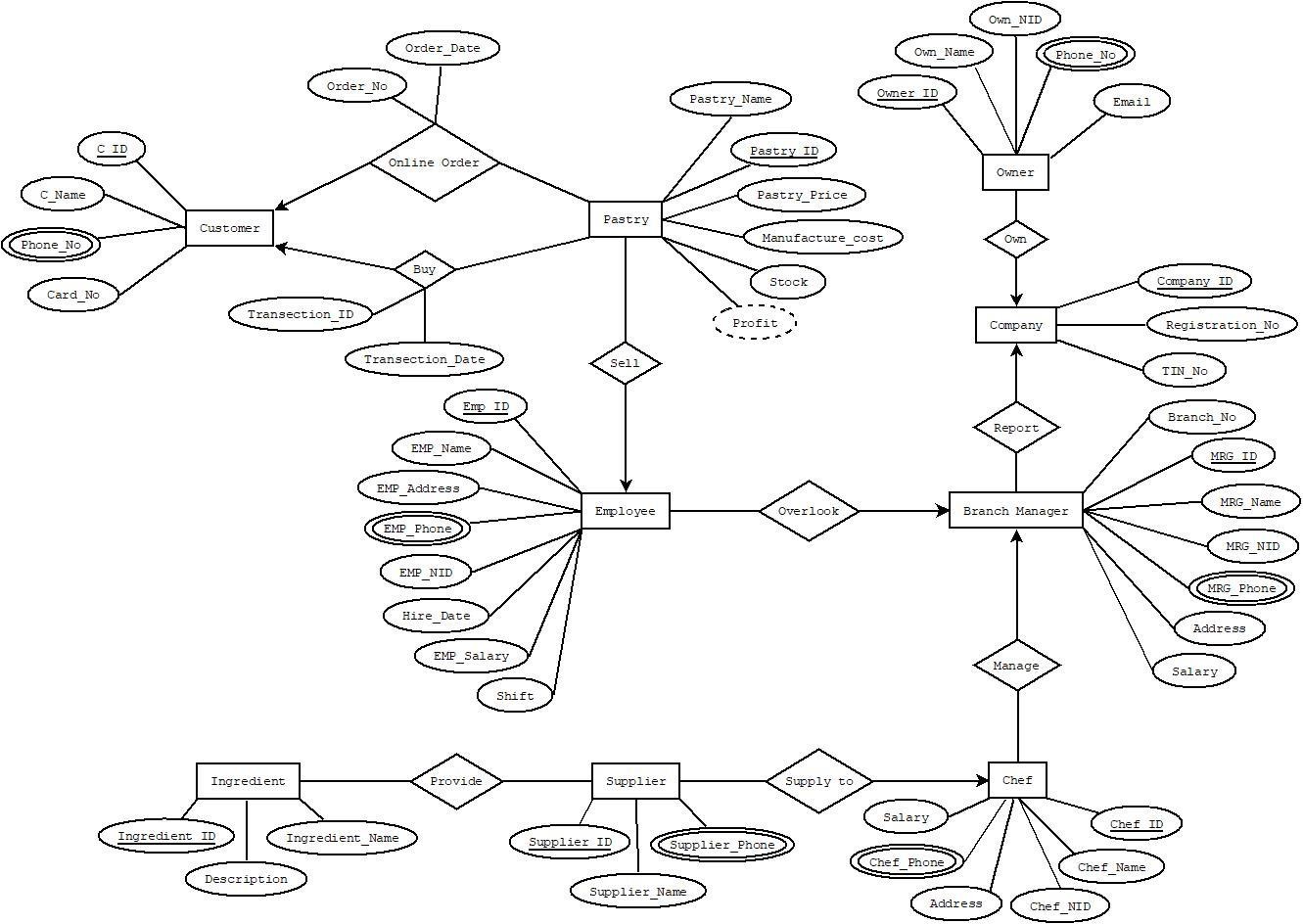
# Introduction

This project Pastry shop Management System for Pastry Company is developed by Oracle SQL. The Pastry shop Management system is an independent application which is based on ordering and selling the pastry and generating the bill. The main principle behind the need of Pastry shop Management System is easy supervision of shop. It has user friendly & modular approach. Data storing is easier. Every information will be authentic and secured in this database system. The pastry shop Management system customer can easily order a pastry. The record of each customer is stored that is customer’s name and contact details are added for reference. The employees can easily access the database and check for availability of the items. At the end of the day, report is generated to calculate the payment for each user in each day. The report will be examined by the company. This Pastry shop Management System will reduce time consumption. This project is prepared to help the company to maintain the day- to-day operations.

# Scenario Description

In a pastry shop, a customer may buy or order many pastries. Each customer has a unique customer Id. Also, customer has name, phone number, card no. A customer can have multiple phone numbers. A pastry can only be sold to the customers. While buying pastry a unique data is stored. In the pastry shop, the pastries are identified by their names. There is pastry name, pastry Id, price, stock details, profit are also stored in the management system. The profit is calculated from the cost price and sell price. Pastries are sold by employees. Where each pastry can be sold by one employee, but one employee can sell many pastries. Each employee is identified by their own unique employee Id. Also, each employee will have some personal information such as name, address, cell number, salary, join date are stored in the database in the pastry shop, some ingredient can be provided by multiple suppliers. The pastry shop is in different districts. It also has many branches over there, and the payment can be done by mobile pay (Bkash, Nagad) and with credit card. However, while ordering it also give an order number and status. Not only that there is also an attribute name as scale, which give us an idea about the salary status of employees. We have also used generalization in between person, employee, and customer, and we also established relation between the attributes and there is also many primary and foreign key.

# ER Diagram (Midterm)

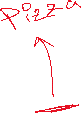
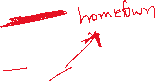


# ER Diagram (Final term)



Diagram

Description automatically generated



* 1. **Newly added:**
* The person attribute has been added which was connected to customer and employee.
* On the other hand, we added order attribute which consist of orderno and status.
* Moreover, district and branch has related to employee.
* We also added scale to check the salary status of the employee.
  1. **Normalization****:**

# Person: There is an attribute called cell which is a multi-value and can be written separately.

# Employee: This table has no transitive and on the other hand it is already in 3NF.

# Customer: This table has no functional dependency, and it is already in 3NF.

# Branch: It is in 3NF.

# Pastry: There is no transitive dependency, so it is also 3NF.

# Scale: Already in 3NF.

# 

* 1. **Table Creation:** 
     1. Create DESC BRANCH

Graphical user interface, text, application, email

Description automatically generated

1. CREATE TABLE RECEIPT

(

ORDERNO NUMBER(20) CONSTRAINT PK\_ODERNO PRIMARY KEY,

STATUS VARCHAR2(10)

);

DESC RECEIPT;

Graphical user interface, text, application

Description automatically generated

1. CREATE TABLE DISTRICT

(

DISNO NUMBER(15) CONSTRAINT PK\_DISNO PRIMARY KEY,

DISNAME VARCHAR2(25)

);

DESC DISTRICT;

Graphical user interface, text, application, email

Description automatically generated

CREATE TABLE PASTRY

(

PASTRYNO NUMBER(30) CONSTRAINT PK\_PASTRYNO PRIMARY KEY,

PASTRYNAME VARCHAR2(30),

STOCK NUMBER(20),

PRICE NUMBER(30),

PROFIT NUMBER(20)

);

DESC PASTRY;

Graphical user interface, application, table

Description automatically generated

5) CREATE TABLE EMPLOYEE

(

EMPNO NUMBER(15) CONSTRAINT PK\_EMPNO PRIMARY KEY,

SALARY NUMBER(7),

GENDER VARCHAR2(10),

JOINDATE DATE,

DISNO NUMBER(15) CONSTRAINT FK\_DISNO REFERENCES DISTRICT

);

DESC EMPLOYEE;

Graphical user interface, application

Description automatically generated with medium confidence

6) CREATE TABLE CUSTOMER

(

CUSNO NUMBER(30) CONSTRAINT PK\_CUSNO PRIMARY KEY,

ADDRESS VARCHAR2(35),

BKASH VARCHAR2(35),

NAGAD VARCHAR2(35),

CCARD VARCHAR2(35),

EMPNO CONSTRAINT FK\_CEMPNO REFERENCES EMPLOYEE

);

DESCRIBE CUSTOMER;

Graphical user interface, application

Description automatically generated

7) CREATE TABLE PERSON

(

NAME VARCHAR2(30),

CELL VARCHAR2(20),

DOB DATE,

EMPNO NUMBER(15) CONSTRAINT FK\_EMPNO REFERENCES EMPLOYEE,

CUSNO NUMBER(15) CONSTRAINT FK\_CUSNO REFERENCES CUSTOMER

);

DESC PERSON;

A screenshot of a computer

Description automatically generated with medium confidence

8) CREATE TABLE SCALE

(

GRADE VARCHAR2(20),

HIGH NUMBER(20),

LOW NUMBER(20)

);

DESCIBE SCALE;

Application

Description automatically generated with low confidence

**Sequence**

CREATE SEQUENCE CUSTOMER\_SEQUENCE MINVALUE 10 MAXVALUE 9999 START WITH 10 NOCACHE NOCYCLE INCREMENT BY 10;

CREATE SEQUENCE EMPLOYEE\_SEQUENCE MINVALUE 10 MAXVALUE 9999 START WITH 10 NOCACHE NOCYCLE INCREMENT BY 10;

CREATE SEQUENCE BRANCH\_SEQUENCE MINVALUE 10 MAXVALUE 9999 START WITH 10 NOCACHE NOCYCLE INCREMENT BY 10;

CREATE SEQUENCE DISTRICT\_SEQUENCE MINVALUE 10 MAXVALUE 9999 START WITH 10 NOCACHE NOCYCLE INCREMENT BY 10;

CREATE SEQUENCE SCALE\_SEQUENCE MINVALUE 10 MAXVALUE 9999 START WITH 10 NOCACHE NOCYCLE INCREMENT BY 10;

CREATE SEQUENCE PERSON\_SEQUENCE MINVALUE 10 MAXVALUE 9999 START WITH 10 NOCACHE NOCYCLE INCREMENT BY 10;

CREATE SEQUENCE PASTRY\_SEQUENCE MINVALUE 10 MAXVALUE 9999 START WITH 10 NOCACHE NOCYCLE INCREMENT BY 10;

CREATE SEQUENCE RECEIPT\_SEQUENCE MINVALUE 10 MAXVALUE 9999 START WITH 10 NOCACHE NOCYCLE INCREMENT BY 10;

Graphical user interface, text

Description automatically generated

9) ALTER TABLE BRANCH DROP COLUMN LOCATION;

1. **Data insertion**

10)INSERT BRANCH

INSERT INTO BRANCH VALUES(BRANCH\_SEQUENCE.NEXTVAL,'PUTULY SWEETS DHANMONDI');

INSERT INTO BRANCH VALUES(BRANCH\_SEQUENCE.NEXTVAL,'PUTULY SWEETS DHAKA');

INSERT INTO BRANCH VALUES(BRANCH\_SEQUENCE.NEXTVAL,'PUTULY SWEETS RAJSHAI');

INSERT INTO BRANCH VALUES(BRANCH\_SEQUENCE.NEXTVAL,'PUTULY SWEETS SYLET');

INSERT INTO BRANCH VALUES(BRANCH\_SEQUENCE.NEXTVAL,'PUTULY SWEETS FARIDPUR');

INSERT INTO BRANCH VALUES(BRANCH\_SEQUENCE.NEXTVAL,'PUTULY SWEETS KHULNA');

INSERT INTO BRANCH VALUES(BRANCH\_SEQUENCE.NEXTVAL,'PUTULY SWEETS RUNGPUR');

INSERT INTO BRANCH VALUES(BRANCH\_SEQUENCE.NEXTVAL,'PUTULY SWEETS DINAJPUR');

SELECT \* FROM BRANCH;

Graphical user interface, text, application, email

Description automatically generated

11) INSERT DISTRICT:

INSERT INTO DISTRICT VALUES(DISTRICT\_SEQUENCE.NEXTVAL,'FARIDPUR');

INSERT INTO DISTRICT VALUES(DISTRICT\_SEQUENCE.NEXTVAL,'RAJSHAI');

INSERT INTO DISTRICT VALUES(DISTRICT\_SEQUENCE.NEXTVAL,'KHULNA');

INSERT INTO DISTRICT VALUES(DISTRICT\_SEQUENCE.NEXTVAL,'SYLET');

INSERT INTO DISTRICT VALUES(DISTRICT\_SEQUENCE.NEXTVAL,'RUNGPUR');

INSERT INTO DISTRICT VALUES(DISTRICT\_SEQUENCE.NEXTVAL,'DINAJPUR');

INSERT INTO DISTRICT VALUES(DISTRICT\_SEQUENCE.NEXTVAL,'KUSTIA');

INSERT INTO DISTRICT VALUES(DISTRICT\_SEQUENCE.NEXTVAL,'MAGORA');

INSERT INTO DISTRICT VALUES(DISTRICT\_SEQUENCE.NEXTVAL,'JOSSOR');

SELECT \* FROM DISTRICT;

Graphical user interface, text, application

Description automatically generated

EMPLOYEE TABLE INSERTION

ALTER TABLE EMPLOYEE ADD(BRANCHNO NUMBER(15) CONSTRAINT FK\_EBRANCH REFERENCES BRANCH);

12) INSERT INTO EMPLOYEE

VALUES(EMPLOYEE\_SEQUENCE.NEXTVAL,6000,'MALE',TO\_DATE('01-01-2017','DD-MM-YY'),10,50);

INSERT INTO EMPLOYEE VALUES(EMPLOYEE\_SEQUENCE.NEXTVAL,6000,'MALE',TO\_DATE('01-01-2009','DD-MM-YY'),10,20);

INSERT INTO EMPLOYEE VALUES(EMPLOYEE\_SEQUENCE.NEXTVAL,5000,'MALE',TO\_DATE('04-01-2017','DD-MM-YY'),20,50);

INSERT INTO EMPLOYEE VALUES(EMPLOYEE\_SEQUENCE.NEXTVAL,7000,'MALE',TO\_DATE('07-01-2018','DD-MM-YY'),30,20);

INSERT INTO EMPLOYEE VALUES(EMPLOYEE\_SEQUENCE.NEXTVAL,4000,'MALE',TO\_DATE('01-01-2016','DD-MM-YY'),100,70);

INSERT INTO EMPLOYEE VALUES(EMPLOYEE\_SEQUENCE.NEXTVAL,6500,'MALE',TO\_DATE('09-01-2017','DD-MM-YY'),50,50);

INSERT INTO EMPLOYEE VALUES(EMPLOYEE\_SEQUENCE.NEXTVAL,2500,'MALE',TO\_DATE('09-01-2019','DD-MM-YY'),70,20);

INSERT INTO EMPLOYEE VALUES(EMPLOYEE\_SEQUENCE.NEXTVAL,2000,'MALE',TO\_DATE('10-01-2015','DD-MM-YY'),10,80);

INSERT INTO EMPLOYEE VALUES(EMPLOYEE\_SEQUENCE.NEXTVAL,8000,'MALE',TO\_DATE('15-01-2019','DD-MM-YY'),80,70);

INSERT INTO EMPLOYEE VALUES(EMPLOYEE\_SEQUENCE.NEXTVAL,6500,'MALE',TO\_DATE('03-01-2016','DD-MM-YY'),80,30);

INSERT INTO EMPLOYEE VALUES(EMPLOYEE\_SEQUENCE.NEXTVAL,3000,'MALE',TO\_DATE('01-01-2015','DD-MM-YY'),100,20);

SELECT \* FROM EMPLOYEE;

Text

Description automatically generated with medium confidence

**SCALE TABLE INTERSION**

ALTER TABLE SCALE DROP COLUMN LOW;

ALTER TABLE SCALE ADD LOW NUMBER(20);

Graphical user interface, text, application, email

Description automatically generated

INSERT INTO SCALE VALUES('FIRST GRADE',500,1500);

INSERT INTO SCALE VALUES('SECOND GRADE',1501,5000);

INSERT INTO SCALE VALUES('THIRD GRADE',5001,10000);

SELECT \* FROM SCALE;

Text

Description automatically generated

**CUSTOMER**



INSERT INTO CUSTOMER VALUES(CUSTOMER\_SEQUENCE.NEXTVAL,'SAVAR DOHS,DHAKA','01878042329','01878042320','123124123',NULL);

INSERT INTO CUSTOMER VALUES(CUSTOMER\_SEQUENCE.NEXTVAL,'SAVAR DOHS,DHAKA','01778042329','01778042320','123124123',10);

INSERT INTO CUSTOMER VALUES(CUSTOMER\_SEQUENCE.NEXTVAL,'BARIDHARA,DHAKA','01578042329','01578042320','753124129',NULL);

INSERT INTO CUSTOMER VALUES(CUSTOMER\_SEQUENCE.NEXTVAL,'MIRPUR-1,DHAKA','01378042322','01878042320','623124125',30);

INSERT INTO CUSTOMER VALUES(CUSTOMER\_SEQUENCE.NEXTVAL,'BONANI,DHAKA','01678042329','01878042320','693124192',60);

INSERT INTO CUSTOMER VALUES(CUSTOMER\_SEQUENCE.NEXTVAL,'FARMGET,DHAKA','01478042329','01378042385','453124122',90);

INSERT INTO CUSTOMER VALUES(CUSTOMER\_SEQUENCE.NEXTVAL,'KHILGAON,DHAKA','01778042355','01868042342','443124122',NULL);

INSERT INTO CUSTOMER VALUES(CUSTOMER\_SEQUENCE.NEXTVAL,'AJIMPUR,DHAKA','01666042329','01978042320','243124125',NULL);

INSERT INTO CUSTOMER VALUES(CUSTOMER\_SEQUENCE.NEXTVAL,'UTTARA,DHAKA','01378042329','01378042370','863124128',NULL);

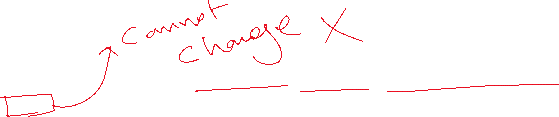
INSERT INTO CUSTOMER VALUES(CUSTOMER\_SEQUENCE.NEXTVAL,'GULSHAN,DHAKA','01677042329','01978042321','813124126',NULL);

SELECT \* FROM CUSTOMER;

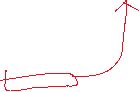
Table

Description automatically generated with medium confidence

**PERSON**



INSERT INTO PERSON VALUES('TONMOY DEY','01878042329',TO\_DATE('01-10-1997','DD-MM-YY'),10,20);



INSERT INTO PERSON VALUES('SHAKIL MOLLA','01378042329',TO\_DATE('12-05-1990','DD-MM-YY'),20,NULL);

INSERT INTO PERSON VALUES('HAJJAJ BIN SONOSI','01878042329',TO\_DATE('21-11-1957','DD-MM-YY'),30,40);

INSERT INTO PERSON VALUES('SHADIN','01578042329',TO\_DATE('01-06-1994','DD-MM-YY'),40,NULL);

INSERT INTO PERSON VALUES('RAFSAN','01678042329',TO\_DATE('01-08-1993','DD-MM-YY'),50,60);

INSERT INTO PERSON VALUES('AYMAN SADIK','01978042329',TO\_DATE('26-02-1967','DD-MM-YY'),60,90);

INSERT INTO PERSON VALUES('SULAYMAN SUKON','01368042329',TO\_DATE('19-12-1960','DD-MM-YY'),NULL,10);

INSERT INTO PERSON VALUES('AREF','01698042329',TO\_DATE('23-10-1998','DD-MM-YY'),NULL,30);

INSERT INTO PERSON VALUES('TOWFIQ-E-ELAHI','01778042329',TO\_DATE('01-05-1988','DD-MM-YY'),70,NULL);

INSERT INTO PERSON VALUES('KAZI FOYSAL','01688042329',TO\_DATE('02-08-1992','DD-MM-YY'),80,NULL);

INSERT INTO PERSON VALUES('FERDOUS AMIN LIAN','01888042329',TO\_DATE('29-10-1985','DD-MM-YY'),NULL,50);

INSERT INTO PERSON VALUES('TASNIM HASAN','01968042329',TO\_DATE('21-03-1991','DD-MM-YY'),90,NULL);

INSERT INTO PERSON VALUES('SABBIR ROHOMAN','01568042329',TO\_DATE('01-11-1990','DD-MM-YY'),NULL,70);

INSERT INTO PERSON VALUES('WASIB AHAMED','01968042329',TO\_DATE('01-07-2000','DD-MM-YY'),NULL,80);

INSERT INTO PERSON VALUES('SAIKUL ISLAM','01708042329',TO\_DATE('01-07-2003','DD-MM-YY'),NULL,100);

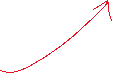
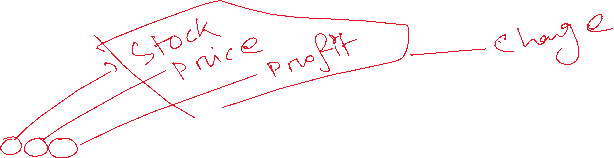
INSERT INTO PERSON VALUES('ABDULA ','01898042329',TO\_DATE('28-02-1986','DD-MM-YY'),100,NULL);

SELECT \* FROM PERSON;

Text

Description automatically generated with low confidence

**PASTRY VALUE**



INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'DARK CHOCOLET',10,120,48);



INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'VENILA STRABERRY',13,165,55);



INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'CLASSIC CUFF PASTRY',05,300,70);

INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'WHITE CHOCOLET',20,80,20);

INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'BUTTER SCOTCH ',12,120,25);

INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'FLAKY PASTRY',07,110,30);

INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'DARK FOREST',16,135,35);

INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'CHORLEY CAKE',15,70,15);

INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'RED VELVED',09,225,80);

INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'BAHULU',07,70,20);

INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'BAKLAVA',06,50,15);

INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'ALOO PIE',04,160,35);

INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'BEAVER TAIL',06,130,30);

INSERT INTO PASTRY VALUES(PASTRY\_SEQUENCE.NEXTVAL,'APPLE PIE',13,85,25);

SELECT \* FROM PASTRY;

Text

Description automatically generated

**RECEIPT**

ALTER TABLE RECEIPT RENAME COLUMN ORDERNO TO RECEIPTNO;

ALTER TABLE RECEIPT ADD(

PASTRYNO NUMBER(20) CONSTRAINT FK\_RPASTRYNO REFERENCES PASTRY,

CUSNO NUMBER(20) CONSTRAINT FK\_RCUSNO REFERENCES CUSTOMER

);

INSERT INTO RECEIPT VALUES(RECEIPT\_SEQUENCE.NEXTVAL,'PAID',10,10);

INSERT INTO RECEIPT VALUES(RECEIPT\_SEQUENCE.NEXTVAL,'PAID',20,10);

INSERT INTO RECEIPT VALUES(RECEIPT\_SEQUENCE.NEXTVAL,NULL,30,10);

INSERT INTO RECEIPT VALUES(RECEIPT\_SEQUENCE.NEXTVAL,NULL,50,30);

INSERT INTO RECEIPT VALUES(RECEIPT\_SEQUENCE.NEXTVAL,'RETURED',40,20);

SELECT \* FROM RECEIPT;

Text

Description automatically generated with low confidence

1. **Query Writing**

**2 SIMPLE QUERIES WITH SIMPLE CONDITIONS**

1)-- FIND ALL PERSON WHO'S NAME HAS FIRST WORD 'T'

SELECT \* FROM PERSON WHERE NAME LIKE 'T%';

**Table

Description automatically generated**

2)-- SHOW ALL PASTRY PRICES GREATER THAN 200

SELECT \* FROM PASTRY WHERE PRICE > 200;

**Table

Description automatically generated**

**1 SINGLE ROW FUNCTION QUERY**

--1) FIND THE REMAINDER WHEN PASTRY PRICE IS DEVIDED BY 2

SSSELECT PASTRYNAME,PRICE,MOD(PRICE,2) FROM PASTRY;

**Table

Description automatically generated**

**1 MULTIPLE ROW FUNCTION QUERY**

--1) FIND THE MOST EXPENSIVE PASTRY FROM THE PASTRY SHOP

SELECT MAX(PRICE) FROM PASTRY;

**Graphical user interface, text, application

Description automatically generated**

**2 SINGLE ROW SUB QUERY**

--1) FIND ALL EMPLOYEES WHO'S HOMETOWN IS IN FARIDPUR

SELECT \* FROM EMPLOYEE WHERE DISNO = (SELECT DISNO FROM DISTRICT WHERE DISNAME = 'FARIDPUR');

**Graphical user interface

Description automatically generated with low confidence**

**-- 2) FIND THE CUSTOMER INFORMATION FROM**

PERSONAL TABLE WHERE EMPLOYEE IS 10

SELECT \* FROM PERSON WHERE CUSNO = (SELECT CUSNO FROM CUSTOMER WHERE EMPNO = 10);

**Table

Description automatically generated**

**1 MULTI ROW SUB QUERY**

SELECT \* FROM PERSON WHERE EMPNO IN (SELECT EMPNO FROM EMPLOYEE WHERE SALARY > 5000);

**Table

Description automatically generated**

**2 COMPLEX SUB QUERIES**

---- 1)RETRIVE ALL EMPLOYEE INFORMATION FROM PERSON TABLE WHERE BRANCH FROM DHAKA

SELECT \* FROM PERSON WHERE EMPNO IN (SELECT EMPNO FROM EMPLOYEE WHERE BRANCHNO IN (SELECT BRANCHNO FROM BRANCH WHERE BRANCHNAME LIKE '%DHAKA'));

**Table

Description automatically generated**

---2) FIND ALL CUSTOMER INFORMATION WHO IS FROM DHAKA AND PURCHASED ATLEAST ONE PASTRY

SELECT \* FROM PERSON WHERE CUSNO IN (SELECT CUSNO FROM CUSTOMER WHERE CUSNO IN (SELECT CUSNO FROM RECEIPT) AND ADDRESS LIKE '%DHAKA');

**Table

Description automatically generated**

**6 JOINING QUERIES –**

**1 ) Equijoin**

--1) EQUIJOIN : SHOW ALL EMPLOYEE WHO IS EARNING MORE THAN 4000

SELECT P.\* FROM PERSON P,EMPLOYEE E WHERE P.EMPNO = E.EMPNO AND E.SALARY > 4000;

**Table

Description automatically generated**

**-- 2) NON-EQUIJOIN :**

FIND THE PAYSCALE OF EVERY EMPLOYEE

SELECT E.EMPNO,E.SALARY,S.GRADE FROM EMPLOYEE E,SCALE S WHERE E.SALARY BETWEEN S.HIGH AND S.LOW;

**Table

Description automatically generated**

**1 LEFT OUTER JOIN**

SHOW ALL BRANCH INFORMATION AND EMPLOYEE NUMBER FROM BRANCH TABLE

SELECT BRANCH.\*,EMPLOYEE.EMPNO FROM BRANCH LEFT OUTER JOIN EMPLOYEE ON (BRANCH.BRANCHNO = EMPLOYEE.BRANCHNO);

**Table

Description automatically generated**

**1) RIGHT OUTER JOIN**

SHOW THE DISTRICT INFORMATION OF ALL EMPLOYEES AND ALL THE DISTRICT INFORMATIONS

SELECT DISTRICT.\*,EMPLOYEE.\* FROM EMPLOYEE RIGHT OUTER JOIN DISTRICT ON (EMPLOYEE.DISNO = DISTRICT.DISNO) ORDER BY DISTRICT.DISNO DESC;

**Table

Description automatically generated**

**1 FULL OUTER JOIN**

RETRIVE ALL DATAS FROM FOLLOWING TABLES USING ONE QUERY : PERSON & EMPLOYEE

SELECT PERSON.\*,EMPLOYEE.\* FROM PERSON FULL OUTER JOIN EMPLOYEE ON (EMPLOYEE.EMPNO = PERSON.EMPNO);

**Table

Description automatically generated with medium confidence**

**1) SELF-JOIN**

SHOW ALL THE WRONG INFORMATION WHERE TWO EMPLOYEE HAS SAME PHONE NUMBER

SELECT P1.NAME,P1.CELL, P2.CELL,P2.NAME FROM PERSON P1,PERSON P2 WHERE P1.CELL = P2.CELL AND P1.EMPNO <> P2.EMPNO;

**Text, table

Description automatically generated**

**1 VIEW**

--SHOW RECEIPT USING A VIEW

CREATE OR REPLACE VIEW PASTRYVIEW (CUSTOMER\_NAME, PHONE, CUSTOMER\_NUMBER, PASTRY\_NAME,ORDER\_STATUS) AS SELECT PERSON.NAME,PERSON.CELL,RECEIPT.CUSNO,PASTRY.PASTRYNAME,NVL2(RECEIPT.STATUS,RECEIPT.STATUS,'STATUS PENDING') FROM PERSON,RECEIPT,PASTRY WHERE PERSON.CUSNO = RECEIPT.CUSNO AND PASTRY.PASTRYNO = RECEIPT.PASTRYNO;

SELECT \* FROM PASTRYVIEW;

**Table

Description automatically generated**

1. **Conclusion**

We have completed the project titled Pastry Shop Management System successfully. The purpose of this project was to develop a relational database management system to manage all aspects of a Pastry Shop company from employee management to inventory management to sales management. We have tried to implement all topics taught to us in the course. We have taken care to develop the system free of errors and make it user friendly. Although we are happy with how much we have done in our project, there is scope for also development. Various features can be implemented for the customers such as the customer being able to see history of all their previous purchases. Some features that can be implemented which would help the business are daily stock update, record of sale of a specific Pastry over a month to determine which Pastry is more in demand.