|  |  |
| --- | --- |
| A picture of a winding road and trees  PRACTICAL ASSIGNMENT 2 | Oarabile Makwe Mahalefa  ADDB6311 |

QUESTION 1

CREATE TABLE CUSTOMER (

CUSTOMER\_ID NUMBER PRIMARY KEY,

FIRST\_NAME VARCHAR2(50),

SURNAME VARCHAR2(50),

ADDRESS VARCHAR2(100),

CONTACT\_NUMBER VARCHAR2(15),

EMAIL VARCHAR2(100)

);

-- Populate CUSTOMER table

INSERT INTO CUSTOMER (CUSTOMER\_ID, FIRST\_NAME, SURNAME, ADDRESS, CONTACT\_NUMBER, EMAIL)

VALUES (11011, 'Jack', 'Smith', '18 Water Rd', '0877277521', 'jsmith@isat.com');

INSERT INTO CUSTOMER (CUSTOMER\_ID, FIRST\_NAME, SURNAME, ADDRESS, CONTACT\_NUMBER, EMAIL)

VALUES (11012, 'Pat', 'Hendricks', '22 Water Rd', '0863257857', 'ph@mcom.co.za');

INSERT INTO CUSTOMER (CUSTOMER\_ID, FIRST\_NAME, SURNAME, ADDRESS, CONTACT\_NUMBER, EMAIL)

VALUES (11013, 'Andre', 'Clark', '101 Summer Lane', '0834567891', 'aclark@mcom.co.za');

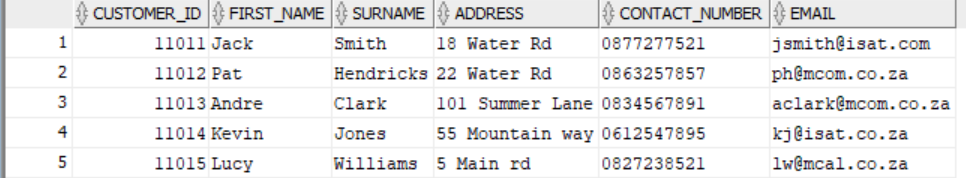
INSERT INTO CUSTOMER (CUSTOMER\_ID, FIRST\_NAME, SURNAME, ADDRESS, CONTACT\_NUMBER, EMAIL)

VALUES (11014, 'Kevin', 'Jones', '55 Mountain way', '0612547895', 'kj@isat.co.za');

INSERT INTO CUSTOMER (CUSTOMER\_ID, FIRST\_NAME, SURNAME, ADDRESS, CONTACT\_NUMBER, EMAIL)

VALUES (11015, 'Lucy', 'Williams', '5 Main rd', '0827238521', 'lw@mcal.co.za');

SELECT \* FROM Customer;



-- Create EMPLOYEE table

CREATE TABLE EMPLOYEE (

EMPLOYEE\_ID VARCHAR2(10) PRIMARY KEY,

FIRST\_NAME VARCHAR2(50),

SURNAME VARCHAR2(50),

CONTACT\_NUMBER VARCHAR2(15),

ADDRESS VARCHAR2(100),

EMAIL VARCHAR2(100)

);

-- Populate EMPLOYEE table

INSERT INTO EMPLOYEE (EMPLOYEE\_ID, FIRST\_NAME, SURNAME, CONTACT\_NUMBER, ADDRESS, EMAIL)

VALUES ('emp101', 'Jeff', 'Davis', '0877277521', '10 main road', 'jand@isat.com');

INSERT INTO EMPLOYEE (EMPLOYEE\_ID, FIRST\_NAME, SURNAME, CONTACT\_NUMBER, ADDRESS, EMAIL)

VALUES ('emp102', 'Kevin', 'Marks', '0837377522', '18 water road', 'km@isat.com');

INSERT INTO EMPLOYEE (EMPLOYEE\_ID, FIRST\_NAME, SURNAME, CONTACT\_NUMBER, ADDRESS, EMAIL)

VALUES ('emp103', 'Adanya', 'Andrews', '0817117523', '21 circle lane', 'aa@isat.com');

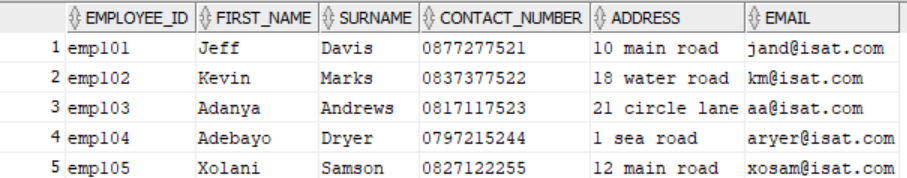
INSERT INTO EMPLOYEE (EMPLOYEE\_ID, FIRST\_NAME, SURNAME, CONTACT\_NUMBER, ADDRESS, EMAIL)

VALUES ('emp104', 'Adebayo', 'Dryer', '0797215244', '1 sea road', 'aryer@isat.com');

INSERT INTO EMPLOYEE (EMPLOYEE\_ID, FIRST\_NAME, SURNAME, CONTACT\_NUMBER, ADDRESS, EMAIL)

VALUES ('emp105', 'Xolani', 'Samson', '0827122255', '12 main road', 'xosam@isat.com');

SELECT \* FROM EMPLOYEE;



-- Create DONATOR table

CREATE TABLE DONATOR (

DONATOR\_ID NUMBER PRIMARY KEY,

FIRST\_NAME VARCHAR2(50),

SURNAME VARCHAR2(50),

CONTACT\_NUMBER VARCHAR2(15),

EMAIL VARCHAR2(100)

);

-- Populate DONATOR table

INSERT INTO DONATOR (DONATOR\_ID, FIRST\_NAME, SURNAME, CONTACT\_NUMBER, EMAIL)

VALUES (20111, 'Jeff', 'Watson', '0827172250', 'jwatson@ymail.com');

INSERT INTO DONATOR (DONATOR\_ID, FIRST\_NAME, SURNAME, CONTACT\_NUMBER, EMAIL)

VALUES (20112, 'Stephen', 'Jones', '0837865670', 'joness@ymail.com');

INSERT INTO DONATOR (DONATOR\_ID, FIRST\_NAME, SURNAME, CONTACT\_NUMBER, EMAIL)

VALUES (20113, 'James', 'Joe', '0878978650', 'jj@isat.com');

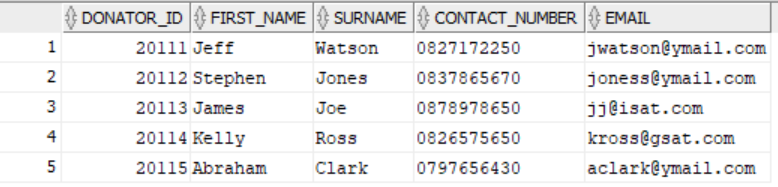
INSERT INTO DONATOR (DONATOR\_ID, FIRST\_NAME, SURNAME, CONTACT\_NUMBER, EMAIL)

VALUES (20114, 'Kelly', 'Ross', '0826575650', 'kross@gsat.com');

INSERT INTO DONATOR (DONATOR\_ID, FIRST\_NAME, SURNAME, CONTACT\_NUMBER, EMAIL)

VALUES (20115, 'Abraham', 'Clark', '0797656430', 'aclark@ymail.com');

SELECT \* FROM DONATOR;



-- Create DONATION table

CREATE TABLE DONATION (

DONATION\_ID NUMBER PRIMARY KEY,

DONATOR\_ID NUMBER,

DONATION VARCHAR2(100),

PRICE NUMBER,

DONATION\_DATE DATE,

FOREIGN KEY (DONATOR\_ID) REFERENCES DONATOR(DONATOR\_ID)

);

-- Populate DONATION table

INSERT INTO DONATION (DONATION\_ID, DONATOR\_ID, DONATION, PRICE, DONATION\_DATE)

VALUES (7111, 20111, 'KIC Fridge',599, TO\_DATE('2024-05-01', 'YYYY-MM-DD'));

INSERT INTO DONATION (DONATION\_ID, DONATOR\_ID, DONATION, PRICE, DONATION\_DATE)

VALUES (7112, 20112, 'Samsung 42inch LCD', 1299, TO\_DATE('2024-05-03', 'YYYY-MM-DD'));

INSERT INTO DONATION (DONATION\_ID, DONATOR\_ID, DONATION, PRICE, DONATION\_DATE)

VALUES (7113, 20113, 'Sharp Microwave', 1599, TO\_DATE('2024-05-03', 'YYYY-MM-DD'));

INSERT INTO DONATION (DONATION\_ID, DONATOR\_ID, DONATION, PRICE, DONATION\_DATE)

VALUES (7114, 20115, '6 Seat Dining room table',799, TO\_DATE('2024-05-05', 'YYYY-MM-DD'));

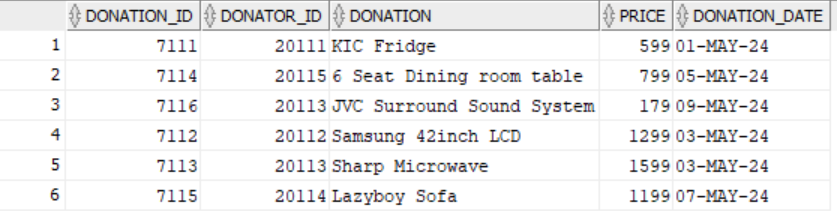
INSERT INTO DONATION (DONATION\_ID, DONATOR\_ID, DONATION, PRICE, DONATION\_DATE)

VALUES (7115, 20114, 'Lazyboy Sofa', 1199, TO\_DATE('2024-05-07', 'YYYY-MM-DD'));

INSERT INTO DONATION (DONATION\_ID, DONATOR\_ID, DONATION, PRICE, DONATION\_DATE)

VALUES (7116, 20113, 'JVC Surround Sound System', 179, TO\_DATE('2024-05-09', 'YYYY-MM-DD'));

SELECT \* FROM DONATION;



-- Create DELIVERY table

CREATE TABLE DELIVERY (

DELIVERY\_ID NUMBER PRIMARY KEY,

DELIVERY\_NOTES VARCHAR2(100),

DISPATCH\_DATE DATE,

DELIVERY\_DATE DATE

);

-- Populate DELIVERY table

INSERT INTO DELIVERY (DELIVERY\_ID, DELIVERY\_NOTES, DISPATCH\_DATE, DELIVERY\_DATE)

VALUES (511, 'Double packaging requested', TO\_DATE('2024-05-10', 'YYYY-MM-DD'), TO\_DATE('2024-05-15', 'YYYY-MM-DD'));

INSERT INTO DELIVERY (DELIVERY\_ID, DELIVERY\_NOTES, DISPATCH\_DATE, DELIVERY\_DATE)

VALUES (512, 'Delivery to work address', TO\_DATE('2024-05-12', 'YYYY-MM-DD'), TO\_DATE('2024-05-15', 'YYYY-MM-DD'));

INSERT INTO DELIVERY (DELIVERY\_ID, DELIVERY\_NOTES, DISPATCH\_DATE, DELIVERY\_DATE)

VALUES (513, 'Signature required', TO\_DATE('2024-05-12', 'YYYY-MM-DD'), TO\_DATE('2024-05-17', 'YYYY-MM-DD'));

INSERT INTO DELIVERY (DELIVERY\_ID, DELIVERY\_NOTES, DISPATCH\_DATE, DELIVERY\_DATE)

VALUES (514, 'No notes', TO\_DATE('2024-05-12', 'YYYY-MM-DD'), TO\_DATE('2024-05-15', 'YYYY-MM-DD'));

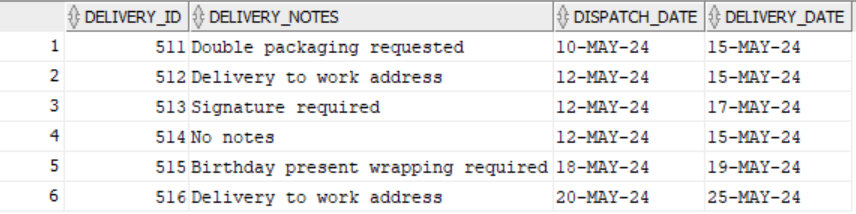
INSERT INTO DELIVERY (DELIVERY\_ID, DELIVERY\_NOTES, DISPATCH\_DATE, DELIVERY\_DATE)

VALUES (515, 'Birthday present wrapping required', TO\_DATE('2024-05-18', 'YYYY-MM-DD'), TO\_DATE('2024-05-19', 'YYYY-MM-DD'));

INSERT INTO DELIVERY (DELIVERY\_ID, DELIVERY\_NOTES, DISPATCH\_DATE, DELIVERY\_DATE)

VALUES (516, 'Delivery to work address', TO\_DATE('2024-05-20', 'YYYY-MM-DD'), TO\_DATE('2024-05-25', 'YYYY-MM-DD'));

SELECT \* FROM DELIVERY;



-- Create RETURNS table

CREATE TABLE RETURNS (

RETURN\_ID VARCHAR2(10) PRIMARY KEY,

RETURN\_DATE DATE,

REASON VARCHAR2(100),

CUSTOMER\_ID NUMBER,

DONATION\_ID NUMBER,

EMPLOYEE\_ID VARCHAR2(10),

FOREIGN KEY (CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID),

FOREIGN KEY (DONATION\_ID) REFERENCES DONATION(DONATION\_ID),

FOREIGN KEY (EMPLOYEE\_ID) REFERENCES EMPLOYEE(EMPLOYEE\_ID)

);

-- Populate RETURNS table

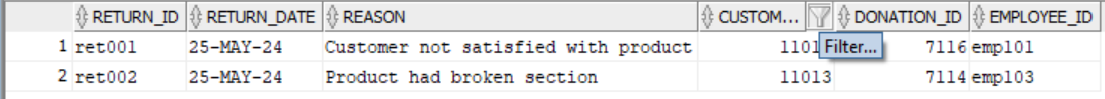
INSERT INTO RETURNS (RETURN\_ID, RETURN\_DATE, REASON, CUSTOMER\_ID, DONATION\_ID, EMPLOYEE\_ID)

VALUES ('ret001', TO\_DATE('2024-05-25', 'YYYY-MM-DD'), 'Customer not satisfied with product', 11011, 7116, 'emp101');

INSERT INTO RETURNS (RETURN\_ID, RETURN\_DATE, REASON, CUSTOMER\_ID, DONATION\_ID, EMPLOYEE\_ID)

VALUES ('ret002', TO\_DATE('2024-05-25', 'YYYY-MM-DD'), 'Product had broken section', 11013, 7114, 'emp103');

SELECT \* FROM RETURNS;



-- Create INVOICE table

CREATE TABLE INVOICE (

INVOICE\_NUM NUMBER PRIMARY KEY,

CUSTOMER\_ID NUMBER,

INVOICE\_DATE DATE,

EMPLOYEE\_ID VARCHAR2(10),

DONATION\_ID NUMBER,

DELIVERY\_ID NUMBER,

FOREIGN KEY (CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID),

FOREIGN KEY (EMPLOYEE\_ID) REFERENCES EMPLOYEE(EMPLOYEE\_ID),

FOREIGN KEY (DONATION\_ID) REFERENCES DONATION(DONATION\_ID),

FOREIGN KEY (DELIVERY\_ID) REFERENCES DELIVERY(DELIVERY\_ID)

);

-- Populate INVOICE table

INSERT INTO INVOICE (INVOICE\_NUM, CUSTOMER\_ID, INVOICE\_DATE, EMPLOYEE\_ID, DONATION\_ID, DELIVERY\_ID)

VALUES (8111, 11011, TO\_DATE('2024-05-15', 'YYYY-MM-DD'), 'emp103', 7111, 511);

INSERT INTO INVOICE (INVOICE\_NUM, CUSTOMER\_ID, INVOICE\_DATE, EMPLOYEE\_ID, DONATION\_ID, DELIVERY\_ID)

VALUES (8112, 11013, TO\_DATE('2024-05-15', 'YYYY-MM-DD'), 'emp101', 7114, 512);

INSERT INTO INVOICE (INVOICE\_NUM, CUSTOMER\_ID, INVOICE\_DATE, EMPLOYEE\_ID, DONATION\_ID, DELIVERY\_ID)

VALUES (8113, 11012, TO\_DATE('2024-05-17', 'YYYY-MM-DD'), 'emp101', 7112, 513);

INSERT INTO INVOICE (INVOICE\_NUM, CUSTOMER\_ID, INVOICE\_DATE, EMPLOYEE\_ID, DONATION\_ID, DELIVERY\_ID)

VALUES (8114, 11015, TO\_DATE('2024-05-17', 'YYYY-MM-DD'), 'emp102', 7113, 514);

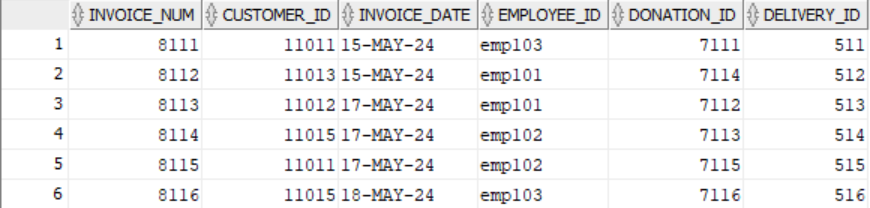
INSERT INTO INVOICE (INVOICE\_NUM, CUSTOMER\_ID, INVOICE\_DATE, EMPLOYEE\_ID, DONATION\_ID, DELIVERY\_ID)

VALUES (8115, 11011, TO\_DATE('2024-05-17', 'YYYY-MM-DD'), 'emp102', 7115, 515);

INSERT INTO INVOICE (INVOICE\_NUM, CUSTOMER\_ID, INVOICE\_DATE, EMPLOYEE\_ID, DONATION\_ID, DELIVERY\_ID)

VALUES (8116, 11015, TO\_DATE('2024-05-18', 'YYYY-MM-DD'), 'emp103', 7116, 516);

SELECT \* FROM INVOICE;



QUESTION 2

SELECT m.FIRST\_NAME || ' ' || m.SURNAME AS CUSTOMER\_NAME, i.EMPLOYEE\_ID, d.DELIVERY\_NOTES, dn.DONATION AS DONATION\_PURCHASED, i.INVOICE\_NUM, i.INVOICE\_DATE

FROM INVOICE i

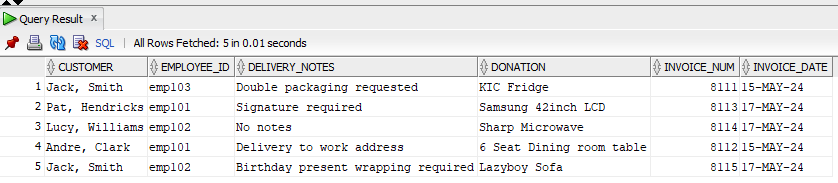
JOIN CUSTOMER m ON i.CUSTOMER\_ID = m.CUSTOMER\_ID

JOIN DELIVERY d ON i.DELIVERY\_ID = d.DELIVERY\_ID

JOIN DONATION dn ON i.DONATION\_ID = dn.DONATION\_ID

WHERE i.INVOICE\_DATE < TO\_DATE('2024-05-18', 'YYYY-MM-DD');

Output:



QUESTION 3

Create sequence for generating unique IDs

CREATE SEQUENCE funding\_seq START WITH 1 INCREMENT BY 1;

-- Create the Funding table

CREATE TABLE Funding (

funding\_id NUMBER PRIMARY KEY,

funder VARCHAR2(50),

funding\_amount NUMBER(10, 2)

);

-- Create trigger to automatically generate unique ID

CREATE OR REPLACE TRIGGER funding\_trigger

BEFORE INSERT ON Funding

FOR EACH ROW

BEGIN

SELECT funding\_seq.nextval INTO :new.funding\_id FROM dual;

END;

Output:

SELECT \* FROM FUNDING;

A screenshot of a computer

Description automatically generated

QUESTION 4

Source Code:

SET SERVEROUTPUT ON;

DECLARE

m\_customer\_name VARCHAR2(100);

m\_donation\_purchased VARCHAR2(100);

m\_donation\_price VARCHAR2(10);

m\_return\_reason VARCHAR2(100);

CURSOR INFO IS

SELECT cu.FIRST\_NAME || ' ' || cu.SURNAME AS CUSTOMER\_NAME, dn.DONATION AS DONATION\_PURCHASED, dn.PRICE AS DONATION\_PRICE, r.REASON AS RETURN\_REASON

FROM RETURNS r

INNER JOIN CUSTOMER cu ON r.CUSTOMER\_ID = cu.CUSTOMER\_ID

INNER JOIN DONATION dn ON r.DONATION\_ID = dn.DONATION\_ID;

BEGIN

FOR rec IN INFO

LOOP

m\_customer\_name := rec.CUSTOMER\_NAME;

m\_donation\_purchased := rec.DONATION\_PURCHASED;

m\_donation\_price := rec.DONATION\_PRICE;

m\_return\_reason := rec.RETURN\_REASON;

-- Display the all information

DBMS\_OUTPUT.PUT\_LINE('Customer Name: ' || m\_customer\_name);

DBMS\_OUTPUT.PUT\_LINE('Donation Purchased: ' || m\_donation\_purchased);

DBMS\_OUTPUT.PUT\_LINE('Donation Price: ' || m\_donation\_price);

DBMS\_OUTPUT.PUT\_LINE('Return Reason: ' || m\_return\_reason);

DBMS\_OUTPUT.PUT\_LINE('------------------------------------');

END LOOP;

END;

Output:

A screenshot of a computer program

Description automatically generated

QUESTION 5

SET SERVEROUTPUT ON;

DECLARE

m\_customer\_name VARCHAR2(100);

m\_employee\_name VARCHAR2(100);

m\_donation VARCHAR2(100);

m\_dispatch\_date DATE;

m\_delivery\_date DATE;

m\_days\_between NUMBER;

CURSOR INFO IS

SELECT c.FIRST\_NAME || ' ' || c.SURNAME AS CUSTOMER\_NAME, e.FIRST\_NAME || ' ' || e.SURNAME AS EMPLOYEE\_NAME,dn.DONATION,me.DISPATCH\_DATE,me.DELIVERY\_DATE,

CASE

WHEN c.CUSTOMER\_ID = 11013 THEN me.DELIVERY\_DATE - me.DISPATCH\_DATE

ELSE NULL

END AS DAYS\_BETWEEN

FROM INVOICE i, CUSTOMER c, DELIVERY me, EMPLOYEE e, DONATION dn

WHERE i.CUSTOMER\_ID = c.CUSTOMER\_ID

AND i.DELIVERY\_ID = me.DELIVERY\_ID

AND i.EMPLOYEE\_ID = e.EMPLOYEE\_ID

AND i.DONATION\_ID = dn.DONATION\_ID

AND i.CUSTOMER\_ID = 11013;

BEGIN

FOR rec IN INFO

LOOP

m\_customer\_name := rec.CUSTOMER\_NAME;

m\_employee\_name := rec.EMPLOYEE\_NAME;

m\_donation := rec.DONATION;

m\_dispatch\_date := rec.DISPATCH\_DATE;

m\_delivery\_date := rec.DELIVERY\_DATE;

m\_days\_between := rec.DAYS\_BETWEEN;

-- Display the information

DBMS\_OUTPUT.PUT\_LINE('Customer: ' || m\_customer\_name);

DBMS\_OUTPUT.PUT\_LINE('Employee: ' || m\_employee\_name);

DBMS\_OUTPUT.PUT\_LINE('Donation: ' || m\_donation);

DBMS\_OUTPUT.PUT\_LINE('Dispatch Date: ' || TO\_CHAR(m\_dispatch\_date, 'YYYY-MM-DD'));

DBMS\_OUTPUT.PUT\_LINE('Delivery Date: ' || TO\_CHAR(m\_delivery\_date, 'YYYY-MM-DD'));

-- Display the number of days that separate customer 11013's dispatch and delivery.

IF m\_customer\_name = 'Andre Clark' THEN

DBMS\_OUTPUT.PUT\_LINE('Days To Delivery: ' || m\_days\_between);

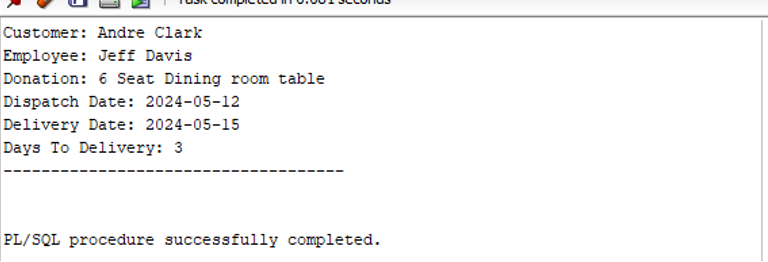
END IF;

DBMS\_OUTPUT.PUT\_LINE('------------------------------------');

END LOOP;

END;

Output:



QUESTION 6

SET SERVEROUTPUT ON;

DECLARE

k\_customer\_name VARCHAR2(100);

k\_customer\_surname VARCHAR2(100);

k\_total\_spent NUMBER;

k\_customer\_rating VARCHAR2(20);

CURSOR INFO IS

SELECT c.FIRST\_NAME, c.SURNAME, NVL(SUM(REPLACE(dn.PRICE, 'R ', '') \* 1), 0) AS TOTAL\_AMOUNT\_SPENT

FROM CUSTOMER c

LEFT JOIN INVOICE i ON c.CUSTOMER\_ID = i.CUSTOMER\_ID

LEFT JOIN DONATION dn ON i.DONATION\_ID = dn.DONATION\_ID

GROUP BY c.CUSTOMER\_ID, c.FIRST\_NAME, c.SURNAME

HAVING NVL(SUM(REPLACE(dn.PRICE, 'R ', '') \* 1), 0) >= 799;

BEGIN

FOR rec IN INFO

LOOP

k\_customer\_name := rec.FIRST\_NAME;

k\_customer\_surname := rec.SURNAME;

k\_total\_spent := rec.TOTAL\_AMOUNT\_SPENT;

-- get the customer rating

IF k\_total\_spent >= 1500 THEN

k\_customer\_rating := '\*';

ELSE

k\_customer\_rating := '';

END IF;

-- Display

DBMS\_OUTPUT.PUT\_LINE('Name: ' || v\_customer\_name);

DBMS\_OUTPUT.PUT\_LINE('Surname: ' || v\_customer\_surname);

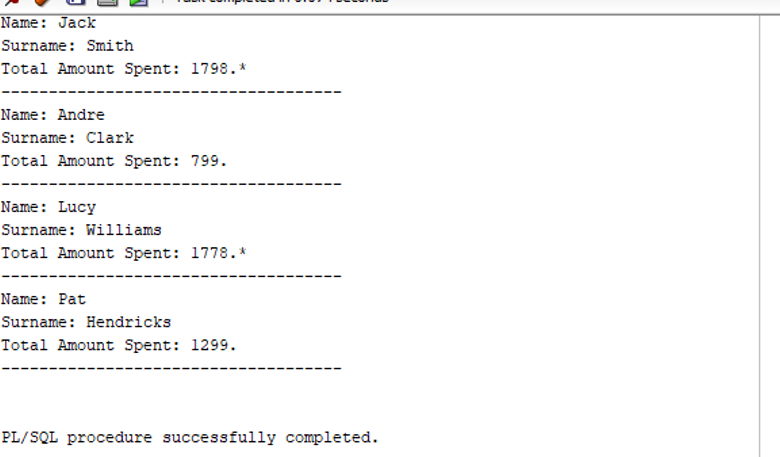
DBMS\_OUTPUT.PUT\_LINE('Total Amount Spent: ' || TO\_CHAR(v\_total\_spent, 'FM9999.99') || v\_customer\_rating);

DBMS\_OUTPUT.PUT\_LINE('------------------------------------');

END LOOP;

END;

Output:



QUESTION 7

Q7.1

SET SERVEROUTPUT ON;

DECLARE

m\_customer\_name CUSTOMER.FIRST\_NAME%TYPE;

m\_employee\_name EMPLOYEE.FIRST\_NAME%TYPE;

m\_donation DONATION.DONATION%TYPE;

m\_dispatch\_date DELIVERY.DISPATCH\_DATE%TYPE;

m\_delivery\_date DELIVERY.DELIVERY\_DATE%TYPE;

m\_days\_between NUMBER;

CURSOR INFO IS

SELECT c.FIRST\_NAME, c.SURNAME, i.EMPLOYEE\_ID, d.DELIVERY\_NOTES, dn.DONATION, d.DISPATCH\_DATE, d.DELIVERY\_DATE

FROM INVOICE i

JOIN CUSTOMER c ON i.CUSTOMER\_ID = c.CUSTOMER\_ID

JOIN DELIVERY d ON i.DELIVERY\_ID = d.DELIVERY\_ID

JOIN DONATION dn ON i.DONATION\_ID = dn.DONATION\_ID;

BEGIN

-- Loop through the cursor

FOR rec IN INFO

LOOP

-- Assign values to variables

m\_customer\_name := cat.FIRST\_NAME || ' ' || cat.SURNAME;

m\_employee\_name := cat.EMPLOYEE\_ID; -- Assuming EMPLOYEE\_ID holds the employee's first name

m\_donation := cat.DONATION;

m\_dispatch\_date := cat.DISPATCH\_DATE;

m\_delivery\_date := cat.DELIVERY\_DATE;

-- Calculate days between dispatch and delivery

m\_days\_between := cat.DELIVERY\_DATE - cat.DISPATCH\_DATE;

-- Display the information

DBMS\_OUTPUT.PUT\_LINE('Customer: ' || m\_customer\_name);

DBMS\_OUTPUT.PUT\_LINE('Employee ID: ' || m\_employee\_name);

DBMS\_OUTPUT.PUT\_LINE('Donation: ' || m\_donation);

DBMS\_OUTPUT.PUT\_LINE('Dispatch Date: ' || TO\_CHAR(m\_dispatch\_date, 'YYYY-MM-DD'));

DBMS\_OUTPUT.PUT\_LINE('Delivery Date: ' || TO\_CHAR(m\_delivery\_date, 'YYYY-MM-DD'));

DBMS\_OUTPUT.PUT\_LINE('Days Between: ' || m\_days\_between);

DBMS\_OUTPUT.PUT\_LINE('------------------------------------');

END LOOP;

END;

Sample output:

A screenshot of a computer

Description automatically generated

Q7.2

SET SERVEROUTPUT ON;

DECLARE

-- Declare a variable of type

m\_donation\_rec DONATION%ROWTYPE;

-- Cursor fetches data

CURSOR donation\_cursor IS

SELECT \* FROM DONATION;

BEGIN

-- Open the cursor

OPEN donation\_cursor;

-- Fetch data and loop

LOOP

-- Fetching data into the variable

FETCH donation\_cursor INTO m\_donation\_rec;

-- Exit loop if there is no more rows

EXIT WHEN donation\_cursor%NOTFOUND;

-- Display values

DBMS\_OUTPUT.PUT\_LINE('Donation ID: ' || m\_donation\_rec.DONATION\_ID);

DBMS\_OUTPUT.PUT\_LINE('Donator ID: ' || m\_donation\_rec.DONATOR\_ID);

DBMS\_OUTPUT.PUT\_LINE('Donation: ' || m\_donation\_rec.DONATION);

DBMS\_OUTPUT.PUT\_LINE('Price: ' || m\_donation\_rec.PRICE);

DBMS\_OUTPUT.PUT\_LINE('Donation Date: ' || TO\_CHAR(m\_donation\_rec.DONATION\_DATE, 'YYYY-MM-DD'));

DBMS\_OUTPUT.PUT\_LINE('------------------------------------');

END LOOP;

-- Close cursor

CLOSE donation\_cursor;

END;

Output:



Q7.3

SET SERVEROUTPUT ON;

DECLARE

p\_customer\_id CUSTOMER.CUSTOMER\_ID%TYPE := 99999;

-- The customer ID 99999 does not exist on the table

p\_customer\_name CUSTOMER.FIRST\_NAME%TYPE;

BEGIN

-- Attempt to select customer

SELECT FIRST\_NAME INTO p\_customer\_name FROM CUSTOMER WHERE CUSTOMER\_ID = p\_customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Customer Name: ' || p\_customer\_name);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Error: No customer found with ID ' || p\_customer\_id);

END;

Output:

A white background with black text

Description automatically generated

Q7.4

Source Code:

-- PL/SQL Block to create and handle a user-defined exception

SET SERVEROUTPUT ON;

DECLARE

m\_customer\_id CUSTOMER.CUSTOMER\_ID%TYPE := 11011;

m\_donation\_amount DONATION.PRICE%TYPE := -100; -- Invalid donation amount

b\_invalid\_donation\_amount EXCEPTION; -- Declare user-defined exception

BEGIN

-- Check for invalid donation amount

IF m\_donation\_amount < 0 THEN

RAISE m\_invalid\_donation\_amount; -- Raise the exception

END IF;

-- If no exception, proceed with the logic

DBMS\_OUTPUT.PUT\_LINE('Valid donation amount: ' || m\_donation\_amount);

EXCEPTION

WHEN b\_invalid\_donation\_amount THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Invalid donation amount ' || m\_donation\_amount);

END;

Sample Output:

