

Assignment 2

ADDB7311



October 7, 2024

ST10393280

Daniel Luke James

Table of Contents

[Question 1 9](#_Toc179234866)

[Question 2 14](#_Toc179234867)

[Question 3 15](#_Toc179234868)

[Question 4 16](#_Toc179234869)

[Question 5 17](#_Toc179234870)

[Question 6 18](#_Toc179234871)

[Question 7 19](#_Toc179234872)

[Question 8 22](#_Toc179234873)

[SQL Code 23](#_Toc179234874)

[ACADEMIC HONESTY DECLARATION 43](#_Toc179234875)

[43](#_Toc179234876)

[Reference List 44](#_Toc179234877)

# Question 1 A screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer screen Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generated

(IIE, 2024).

# Question 2

A screenshot of a computer

Description automatically generated

(IIE, 2024).

# Question 3

# A white background with black dots Description automatically generatedA screenshot of a computer Description automatically generatedA white screen with a black text Description automatically generated with medium confidence

(IIE, 2024).

# Question 4

A screenshot of a computer

Description automatically generated

(IIE, 2024).

# Question 5

A screenshot of a computer

Description automatically generated

(IIE, 2024).

# Question 6

A screenshot of a computer

Description automatically generated

(IIE, 2024).

# Question 7

A screenshot of a computer program

Description automatically generated A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated (IIE, 2024).

# Question 8

A screenshot of a computer

Description automatically generated

(IIE, 2024).

# SQL Code

-- ST10393280 Daniel Luke James

-- Question 1

-- New database schema

CREATE DATABASE CHARITY4AFRICA\_DBA;

USE CHARITY4AFRICA\_DBA;

-- Create Tables

CREATE TABLE CUSTOMER (

CUSTOMER\_ID VARCHAR2(10),

FIRST\_NAME VARCHAR2(50),

SURNAME VARCHAR2(50),

ADDRESS VARCHAR2(100),

CONTACT\_NUMBER VARCHAR2(20),

EMAIL VARCHAR2(50),

PRIMARY KEY (CUSTOMER\_ID)

);

CREATE TABLE EMPLOYEE (

EMPLOYEE\_ID VARCHAR2(10),

FIRST\_NAME VARCHAR2(50),

SURNAME VARCHAR2(50),

CONTACT\_NUMBER VARCHAR2(20),

ADDRESS VARCHAR2(100),

EMAIL VARCHAR2(50),

PRIMARY KEY (EMPLOYEE\_ID)

);

CREATE TABLE DONATOR (

DONATOR\_ID VARCHAR2(10),

FIRST\_NAME VARCHAR2(50),

SURNAME VARCHAR2(50),

CONTACT\_NUMBER VARCHAR2(20),

EMAIL VARCHAR2(50),

PRIMARY KEY (DONATOR\_ID)

);

CREATE TABLE DONATION (

DONATION\_ID VARCHAR2(10),

DONATOR\_ID VARCHAR2(10),

DONATION VARCHAR2(100),

PRICE NUMBER(10, 2),

DONATION\_DATE DATE,

PRIMARY KEY (DONATION\_ID),

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

);

CREATE TABLE DELIVERY (

DELIVERY\_ID VARCHAR2(10),

DELIVERY\_NOTES VARCHAR2(200),

DISPATCH\_DATE DATE,

DELIVERY\_DATE DATE,

PRIMARY KEY (DELIVERY\_ID)

);

CREATE TABLE RETURNS (

RETURN\_ID VARCHAR2(10),

RETURN\_DATE DATE,

REASON VARCHAR2(200),

CUSTOMER\_ID VARCHAR2(10),

DONATION\_ID VARCHAR2(10),

EMPLOYEE\_ID VARCHAR2(10),

PRIMARY KEY (RETURN\_ID),

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

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

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

);

CREATE TABLE INVOICE (

INVOICE\_NUM VARCHAR2(10),

CUSTOMER\_ID VARCHAR2(10),

INVOICE\_DATE DATE,

EMPLOYEE\_ID VARCHAR2(10),

DONATION\_ID VARCHAR2(10),

DELIVERY\_ID VARCHAR2(10),

PRIMARY KEY (INVOICE\_NUM),

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)

);

-- Insert Data

-- 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 Summer 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');

-- 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');

-- 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');

-- DONATION Table

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

VALUES ('7111', '20111', 'KIC Fridge', 599, TO\_DATE('1-MAY-2024', 'DD-MON-YYYY'));

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

VALUES ('7112', '20112', 'Samsung 42inch LCD', 1299, TO\_DATE('3-MAY-2024', 'DD-MON-YYYY'));

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

VALUES ('7113', '20113', 'Sharp Microwave', 1599, TO\_DATE('3-MAY-2024', 'DD-MON-YYYY'));

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

VALUES ('7114', '20115', '6 Seat Dinning room table', 799, TO\_DATE('5-MAY-2024', 'DD-MON-YYYY'));

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

VALUES ('7115', '20114', 'Lazyboy Sofa', 1199, TO\_DATE('7-MAY-2024', 'DD-MON-YYYY'));

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

VALUES ('7116', '20113', 'JVC Surround Sound Systme', 179, TO\_DATE('9-MAY-2024', 'DD-MON-YYYY'));

-- DELIVERY Table

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

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

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

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

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

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

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

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

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

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

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

VALUES ('516', 'Delivary to work address', TO\_DATE('20-MAY-2024', 'DD-MON-YYYY'), TO\_DATE('25-MAY-2024', 'DD-MON-YYYY'));

-- RETURNS Table

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

VALUES ('ret001', TO\_DATE('25-MAY-2024', 'DD-MON-YYYY'), '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('25-MAY-2024', 'DD-MON-YYYY'), 'Product had broken section', '11013', '7114', 'emp103');

-- INVOICE Table

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

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

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

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

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

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

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

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

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

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

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

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

-- View Table Data (Confirmation)

SELECT \* FROM CUSTOMER;

SELECT \* FROM EMPLOYEE;

SELECT \* FROM DONATOR;

SELECT \* FROM DONATION;

SELECT \* FROM DELIVERY;

SELECT \* FROM RETURNS;

SELECT \* FROM INVOICE;

-- Question 2

SELECT

CUSTOMER.FIRST\_NAME || ' ' || CUSTOMER.SURNAME AS CUSTOMER\_NAME,

INVOICE.EMPLOYEE\_ID,

DELIVERY.DELIVERY\_NOTES,

DONATION.DONATION,

INVOICE.INVOICE\_NUM,

TO\_CHAR(INVOICE.INVOICE\_DATE, 'DD/MON/YY') AS INVOICE\_DATE

FROM

CUSTOMER

JOIN INVOICE ON CUSTOMER.CUSTOMER\_ID = INVOICE.CUSTOMER\_ID

JOIN DELIVERY ON INVOICE.DELIVERY\_ID = DELIVERY.DELIVERY\_ID

JOIN DONATION ON INVOICE.DONATION\_ID = DONATION.DONATION\_ID

WHERE INVOICE.INVOICE\_DATE > TO\_DATE('16-MAY-2024', 'DD-MON-YYYY');

-- Question 3

CREATE TABLE FUNDING (

FUNDING\_ID NUMBER GENERATED ALWAYS AS IDENTITY,

FUNDER VARCHAR2(100),

FUNDING\_AMOUNT NUMBER(10, 2),

PRIMARY KEY (FUNDING\_ID)

);

-- FUNDING Table

INSERT INTO FUNDING (FUNDER, FUNDING\_AMOUNT)

VALUES ('Daniel James', 8000);

INSERT INTO FUNDING (FUNDER, FUNDING\_AMOUNT)

VALUES ('Sai Surujpal', 5000);

SELECT \* FROM FUNDING;

-- JUSTIFICATION

-- Without having to explicitly indicate it during inserts, each FUNDING\_ID is guaranteed to be unique when using the GENERATED ALWAYS AS IDENTITY.

-- This eliminates the possibility of ID duplication issues.

-- Question 4

-- Enable server output

SET SERVEROUTPUT ON;

-- PL/SQL block

BEGIN

FOR rec IN (

SELECT

CUSTOMER.FIRST\_NAME || ', ' || CUSTOMER.SURNAME AS CUSTOMER\_NAME,

DONATION.DONATION,

DONATION.PRICE,

RETURNS.REASON

FROM

CUSTOMER

JOIN RETURNS ON CUSTOMER.CUSTOMER\_ID = RETURNS.CUSTOMER\_ID

JOIN DONATION ON RETURNS.DONATION\_ID = DONATION.DONATION\_ID

)

LOOP

DBMS\_OUTPUT.PUT\_LINE('CUSTOMER: ' || rec.CUSTOMER\_NAME);

DBMS\_OUTPUT.PUT\_LINE('DONATION PURCHASED: ' || rec.DONATION);

DBMS\_OUTPUT.PUT\_LINE('PRICE: ' || rec.PRICE);

DBMS\_OUTPUT.PUT\_LINE('RETURN REASON: ' || rec.REASON);

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

END LOOP;

END;

/

-- Question 5

-- Enable server output

SET SERVEROUTPUT ON;

-- PL/SQL block

BEGIN

FOR rec IN (

SELECT

SUBSTR(C.FIRST\_NAME, 1, 1) || '.' || C.SURNAME AS CUSTOMER\_NAME,

SUBSTR(E.FIRST\_NAME, 1, 1) || '.' || E.SURNAME AS EMPLOYEE\_NAME,

D.DONATION,

DL.DISPATCH\_DATE,

DL.DELIVERY\_DATE,

(DL.DELIVERY\_DATE - DL.DISPATCH\_DATE) AS DAYS\_BETWEEN

FROM

CUSTOMER C

JOIN INVOICE I ON C.CUSTOMER\_ID = I.CUSTOMER\_ID

JOIN EMPLOYEE E ON I.EMPLOYEE\_ID = E.EMPLOYEE\_ID

JOIN DONATION D ON I.DONATION\_ID = D.DONATION\_ID

JOIN DELIVERY DL ON I.DELIVERY\_ID = DL.DELIVERY\_ID

WHERE

C.CUSTOMER\_ID = '11011'

)

LOOP

DBMS\_OUTPUT.PUT\_LINE(RPAD('CUSTOMER: ', 20) || rec.CUSTOMER\_NAME);

DBMS\_OUTPUT.PUT\_LINE(RPAD('EMPLOYEE: ', 20) || rec.EMPLOYEE\_NAME);

DBMS\_OUTPUT.PUT\_LINE(RPAD('DONATION: ', 20) || rec.DONATION);

DBMS\_OUTPUT.PUT\_LINE(RPAD('DISPATCH DATE: ', 20) || TO\_CHAR(rec.DISPATCH\_DATE, 'DD/MON/YY'));

DBMS\_OUTPUT.PUT\_LINE(RPAD('DELIVERY DATE: ', 20) || TO\_CHAR(rec.DELIVERY\_DATE, 'DD/MON/YY'));

DBMS\_OUTPUT.PUT\_LINE(RPAD('DAYS TO Delivery: ', 20) || rec.DAYS\_BETWEEN);

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

END LOOP;

END;

/

-- Question 6

-- Enable server output

SET SERVEROUTPUT ON;

-- PL/SQL block

BEGIN

FOR rec IN (

SELECT

c.FIRST\_NAME AS FIRST\_NAME,

c.SURNAME AS SURNAME,

SUM(d.PRICE) AS TOTAL\_AMOUNT

FROM

CUSTOMER c

JOIN

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

JOIN

DONATION d ON i.DONATION\_ID = d.DONATION\_ID

GROUP BY

c.FIRST\_NAME, c.SURNAME

ORDER BY

c.FIRST\_NAME, c.SURNAME

) LOOP

DBMS\_OUTPUT.PUT\_LINE('First Name: ' || rec.FIRST\_NAME);

DBMS\_OUTPUT.PUT\_LINE('Surname: ' || rec.SURNAME);

IF rec.TOTAL\_AMOUNT >= 1500 THEN

DBMS\_OUTPUT.PUT\_LINE('Amount: R ' || rec.TOTAL\_AMOUNT || ' (\*\*\*)');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Amount: R ' || rec.TOTAL\_AMOUNT);

END IF;

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

END LOOP;

END;

/

-- Question 7

-- Q.7.1

DECLARE

-- Declaring a Variable

v\_price DONATION.PRICE%TYPE;

v\_donation\_id DONATION.DONATION\_ID%TYPE;

BEGIN

-- Recieving Donation Price Example

SELECT PRICE INTO v\_price

FROM DONATION

WHERE DONATION\_ID = '7111';

-- Output Donation Price

DBMS\_OUTPUT.PUT\_LINE('Price of donation (ID: 7111): ' || v\_donation\_id || ': ' || v\_price);

END;

-- Q.7.2

DECLARE

-- Declaring a Variable

v\_donation\_record DONATION%ROWTYPE;

BEGIN

-- Recieving Donation Price Example

SELECT \*

INTO v\_donation\_record

FROM DONATION

WHERE DONATION\_ID = '7111';

-- Output Donation Details

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

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

DBMS\_OUTPUT.PUT\_LINE('Donation Description: ' || v\_donation\_record.DONATION);

DBMS\_OUTPUT.PUT\_LINE('Price: ' || v\_donation\_record.PRICE);

DBMS\_OUTPUT.PUT\_LINE('Donation Date: ' || TO\_CHAR(v\_donation\_record.DONATION\_DATE, 'DD-MON-YYYY'));

END;

-- Q.7.3

DECLARE

-- Declaring a User-Defined Exception

-- Example of a Donation ID that does not exit

donation\_not\_found EXCEPTION;

v\_donation\_id DONATION.DONATION\_ID%TYPE := '9999';

v\_customer\_id CUSTOMER.CUSTOMER\_ID%TYPE := '11012';

v\_employee\_id EMPLOYEE.EMPLOYEE\_ID%TYPE := 'emp101';

v\_invoice\_num INVOICE.INVOICE\_NUM%TYPE;

BEGIN

-- Check if the donation exists

SELECT DONATION\_ID INTO v\_donation\_id

FROM DONATION

WHERE DONATION\_ID = v\_donation\_id;

-- If the donation exists, then create the invoice

v\_invoice\_num := 'INV' || TO\_CHAR(SYSDATE, 'YYYYMMDDHH24MISS');

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

VALUES (v\_invoice\_num, v\_customer\_id, SYSDATE, v\_employee\_id, v\_donation\_id, NULL);

-- Output Invoice Number

DBMS\_OUTPUT.PUT\_LINE('Invoice created successfully with Invoice Number: ' || v\_invoice\_num);

EXCEPTION

-- Exception Handling

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Donation ID ' || v\_donation\_id || ' does not exist. Cannot create invoice.');

WHEN donation\_not\_found THEN

DBMS\_OUTPUT.PUT\_LINE('A user-defined error occurred: Donation ID does not exist.');

END;

-- Question 8

SELECT

c.FIRST\_NAME AS FIRST\_NAME,

c.SURNAME AS SURNAME,

SUM(d.PRICE) AS AMOUNT,

CASE

WHEN SUM(d.PRICE) >= 1500 THEN '\*\*\*'

WHEN SUM(d.PRICE) BETWEEN 1000 AND 1400 THEN '\*\*'

ELSE '\*'

END AS CUSTOMER\_RATING

FROM

CUSTOMER c

JOIN

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

JOIN

DONATION d ON i.DONATION\_ID = d.DONATION\_ID

GROUP BY

c.FIRST\_NAME, c.SURNAME

ORDER BY

c.FIRST\_NAME, c.SURNAME;

# ACADEMIC HONESTY DECLARATION

# 

(IIE, 2024).

# Reference List

* IIE. (2024). Advanced Databases Module Manual (First ed.) IIE.

\*\*Note: I had to put the in-text referencing at the end of each question, as the images were moving around in a weird format. Thank you \*\*