






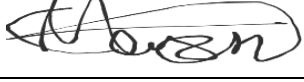
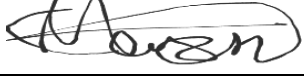
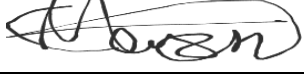



ASSIGNMENT 2**ASSESSMENT DECLARATION**

	SIGN
I have read the assessment rules provided in this declaration.	
This assessment is my own work.	
I have not copied any other student's work in this assessment.	
I have not uploaded the assessment question to any website or App offering assessment assistance.	
I have not downloaded my assessment response from a website.	
I have not used any AI tool without reviewing, re-writing, and re-working this information, and referencing any AI tools in my work.	
I have not shared this assessment with any other student.	
I have not presented the work of published sources as my own work.	
I have correctly cited all my sources of information.	
My referencing is technically correct, consistent, and congruent.	
I have acted in an academically honest way in this assessment.	

--Question 1

```
CREATE TABLE CUSTOMER (  
    CUSTOMER_ID INT PRIMARY KEY,  
    FIRST_NAME VARCHAR(50),  
    SURNAME VARCHAR(50),  
    ADDRESS VARCHAR(100),  
    CONTACT_NUMBER VARCHAR(15),  
    EMAIL VARCHAR(100)  
);
```

```
CREATE TABLE EMPLOYEE (  
    EMPLOYEE_ID VARCHAR(10) PRIMARY KEY,  
    FIRST_NAME VARCHAR(50),  
    SURNAME VARCHAR(50),  
    CONTACT_NUMBER VARCHAR(15),  
    ADDRESS VARCHAR(100),  
    EMAIL VARCHAR(100)  
);
```

```
CREATE TABLE DONATOR (  
    DONATOR_ID INT PRIMARY KEY,  
    FIRST_NAME VARCHAR(50),  
    SURNAME VARCHAR(50),  
    CONTACT_NUMBER VARCHAR(15),  
    EMAIL VARCHAR(100)  
);
```

```
CREATE TABLE DONATION (  
    DONATION_ID INT PRIMARY KEY,  
    DONATOR_ID INT,  
    DONATION VARCHAR(100),  
    PRICE DECIMAL(10, 2),
```

```
DONATION_DATE DATE,  
FOREIGN KEY (DONATOR_ID) REFERENCES DONATOR(DONATOR_ID)  
);
```

```
CREATE TABLE DELIVERY (  
    DELIVERY_ID INT PRIMARY KEY,  
    DELIVERY_NOTES VARCHAR(200),  
    DISPATCH_DATE DATE,  
    DELIVERY_DATE DATE  
);
```

```
CREATE TABLE RETURNS (  
    RETURN_ID VARCHAR(10) PRIMARY KEY,  
    RETURN_DATE DATE,  
    REASON VARCHAR(200),  
    CUSTOMER_ID INT,  
    DONATION_ID INT,  
    EMPLOYEE_ID VARCHAR(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)  
);
```

```
CREATE TABLE INVOICE (  
    INVOICE_NUM INT PRIMARY KEY,  
    CUSTOMER_ID INT,  
    INVOICE_DATE DATE,  
    EMPLOYEE_ID VARCHAR(10),  
    DONATION_ID INT,  
    DELIVERY_ID INT,  
    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 statements for CUSTOMER table

INSERT ALL

INTO CUSTOMER (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, CONTACT_NUMBER,
EMAIL)

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

INTO CUSTOMER (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, CONTACT_NUMBER,
EMAIL)

VALUES (11012, 'Pat', 'Andre', '101 Summer Lane', '0863257857',
'ph@mcom.co.za')

INTO CUSTOMER (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, CONTACT_NUMBER,
EMAIL)

VALUES (11013, 'Hendricks', 'Clark', '22 Water Rd', '0834567891',
'aclark@mcom.co.za')

INTO CUSTOMER (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, CONTACT_NUMBER,
EMAIL)

VALUES (11014, 'Kevin', 'Jones', '5 Main rd', '0612547895',
'kj@isat.co.za')

INTO CUSTOMER (CUSTOMER_ID, FIRST_NAME, SURNAME, ADDRESS, CONTACT_NUMBER,
EMAIL)

VALUES (11015, 'Lucy', 'Williams', '55 Mountain way', '0827238521',
'lw@mcal.co.za')

SELECT * FROM DUAL;

-- Insert statements for EMPLOYEE table

INSERT ALL

INTO EMPLOYEE (EMPLOYEE_ID, FIRST_NAME, SURNAME, CONTACT_NUMBER, ADDRESS,
EMAIL)

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

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', 'Adebayo', '0817117528', '21 circle lane',
'aa@isat.com')

INSERT INTO EMPLOYEE (EMPLOYEE_ID, FIRST_NAME, SURNAME, CONTACT_NUMBER, ADDRESS,
EMAIL)

VALUES ('emp104', 'Xolani', 'Andrews', '0797215244', '1 sea road',
'aryer@isat.com')

INSERT INTO EMPLOYEE (EMPLOYEE_ID, FIRST_NAME, SURNAME, CONTACT_NUMBER, ADDRESS,
EMAIL)

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

SELECT * FROM DUAL;
```

-- Insert statements for DONATOR table

```
INSERT ALL

INSERT INTO DONATOR (DONATOR_ID, FIRST_NAME, SURNAME, CONTACT_NUMBER, EMAIL)
VALUES (20111, 'Jeff', 'Watson', '0827172250', 'iwatson@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', 'j@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 DUAL;
```

-- Insert statements for DONATION table

```
INSERT ALL

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'))
```

```
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'))
```

```
INTO DONATION (DONATION_ID, DONATOR_ID, DONATION, PRICE, DONATION_DATE)
```

```
VALUES (7113, 20113, 'Sharp Microwave', 1599, TO_DATE('2024-05-03', 'YYYY-MM-DD'))
```

```
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'))
```

```
INTO DONATION (DONATION_ID, DONATOR_ID, DONATION, PRICE, DONATION_DATE)
```

```
VALUES (7115, 20114, 'Lazyboy Sofa', 1199, TO_DATE('2024-05-07', 'YYYY-MM-DD'))
```

```
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 DUAL;
```

```
-- Insert statements for DELIVERY table
```

```
INSERT ALL
```

```
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'))
```

```
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'))
```

```
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'))
```

```
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'))
```

```
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'))
```

```
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 DUAL;
```

```
-- Insert statements for RETURNS table
```

```
INSERT ALL
```

```
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')
```

```
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 DUAL;
```

```
-- Insert statements for INVOICE table
```

```
INSERT ALL
```

```
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)
```

```
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)
```

```
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)
```

```
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)
```

```
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)

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 DUAL;

SELECT * FROM CUSTOMER;

100% All Rows Filtered: 0.110.000 seconds

	CUSTOMER_ID	FIRST_NAME	SURNAME	ADDRESS	CONTACT_NUMBER	EMAIL
1	11011	Jack	Smith	18 Water Rd	0877277521	jsmith@isat.com
2	11012	Pat	Andre	101 Summer Lane	0863257857	ph@mcom.co.za
3	11013	Hendricks	Clark	22 Water Rd	0834567891	aclark@mcom.co.za
4	11014	Kevin	Jones	5 Main rd	0612547895	kj@isat.co.za
5	11015	Lucy	Williams	55 Mountain way	0827238521	lw@mcal.co.za

SELECT * FROM EMPLOYEE;

	EMPLOYEE_ID	FIRST_NAME	SURNAME	CONTACT_NUMBER	ADDRESS	EMAIL
1	emp101	Jeff	Davis	0877277521	10 main road	jand@isat.com
2	emp102	Kevin	Marks	0837377522	18 water road	km@isat.com
3	emp103	Adanya	Adebayo	0817117528	21 circle lane	aa@isat.com
4	emp104	Xolani	Andrews	0797215244	1 sea road	aryer@isat.com
5	emp105	Dryer	Samson	0827122255	12 main road	xosam@isat.com

SELECT * FROM DONATOR;

	DONATOR_ID	FIRST_NAME	SURNAME	CONTACT_NUMBER	EMAIL
1	20111	Jeff	Watson	0827172250	iwatson@ymail.com
2	20112	Stephen	Jones	0837865670	jones@ymail.com
3	20113	James	Joe	0878978650	j@isat.com
4	20114	Kelly	Ross	0826575650	kross@gsat.com
5	20115	Abraham	Clark	0797656430	aclark@ymail.com

SELECT * FROM DONATION;

	DONATION_ID	DONATOR_ID	DONATION	PRICE	DONATION_DATE
1	7111	20111	KIC Fridge	599	01-MAY-24
2	7112	20112	Samsung 42inch LCD	1299	03-MAY-24
3	7113	20113	Sharp Microwave	1599	03-MAY-24
4	7114	20115	6 Seat Dining room table	799	05-MAY-24
5	7115	20114	Lazyboy Sofa	1199	07-MAY-24
6	7116	20113	JVC Surround Sound System	179	09-MAY-24

SELECT * FROM DELIVERY;

	DELIVERY_ID	DELIVERY_NOTES	DISPATCH_DATE	DELIVERY_DATE
1	511	Double packaging requested	10-MAY-24	15-MAY-24
2	512	Delivery to work address	12-MAY-24	15-MAY-24
3	513	Signature required	12-MAY-24	17-MAY-24
4	514	No notes	12-MAY-24	15-MAY-24
5	515	Birthday present wrapping required	18-MAY-24	19-MAY-24
6	516	Delivery to work address	20-MAY-24	25-MAY-24

SELECT * FROM RETURNS;

	RETURN_ID	RETURN_DATE	REASON	CUSTOMER_ID	DONATION_ID	EMPLOYEE_ID
1	ret001	25-MAY-24	Customer not satisfied with product	11011	7116	emp101
2	ret002	25-MAY-24	Product had broken section	11013	7114	emp103

SELECT * FROM INVOICE;

	INVOICE_NUM	CUSTOMER_ID	INVOICE_DATE	EMPLOYEE_ID	DONATION_ID	DELIVERY_ID
1	8111	11011	15-MAY-24	emp103	7111	511
2	8112	11013	15-MAY-24	emp101	7114	512
3	8113	11012	17-MAY-24	emp101	7112	513
4	8114	11015	17-MAY-24	emp102	7113	514
5	8115	11011	17-MAY-24	emp102	7115	515
6	8116	11015	18-MAY-24	emp103	7116	516

--Question 2

SELECT

c.FIRST_NAME || ', ' || c.SURNAME AS CUSTOMER_NAME,

i.EMPLOYEE_ID,

d.DELIVERY_NOTES,

dn.DONATION,

i.INVOICE_NUM,

i.INVOICE_DATE

FROM

CUSTOMER c

JOIN

INVOICE i ON c.CUSTOMER_ID = i.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');

 All Rows Fetched: 5 in 0.011 seconds

CUSTOMER_NAME	EMPLOYEE_ID	DELIVERY_NOTES	DONATION	INVOICE_NUM	INVOICE_DATE
1 Jack, Smith	emp103	Double packaging requested	KIC Fridge	8111	15-MAY-24
2 Pat, Andre	emp101	Signature required	Samsung 42inch LCD	8113	17-MAY-24
3 Lucy, Williams	emp102	No notes	Sharp Microwave	8114	17-MAY-24
4 Hendricks, Clark	emp101	Delivery to work address	6 Seat Dining room table	8112	15-MAY-24
5 Jack, Smith	emp102	Birthday present wrapping required	Lazyboy Sofa	8115	17-MAY-24

--Question 3

-- Create a sequence to generate unique IDs for the Funding table

CREATE SEQUENCE funding_seq

START WITH 1

INCREMENT BY 1

NOMAXVALUE;

-- Create the Funding table

CREATE TABLE Funding (

funding_id INT PRIMARY KEY,

funder VARCHAR(100),

funding_amount DECIMAL(10, 2)

);

-- Create a trigger to automatically populate the funding_id column with the next value from the sequence

CREATE OR REPLACE TRIGGER funding_id_trigger

BEFORE INSERT ON Funding

FOR EACH ROW

BEGIN

SELECT funding_seq.NEXTVAL

INTO :new.funding_id

FROM dual;

```
END;
```

```
/
```

```
Sequence FUNDING_SEQ created.
```

```
Table FUNDING created.
```


```
Trigger FUNDING_ID_TRIGGER compiled
```

```
1 row inserted.
```

```
-- Example of an insert statement
```

```
INSERT INTO Funding (funder, funding_amount) VALUES ('Fenty Beauty',
5000.00);
```

```
SELECT * FROM FUNDING;
```

 All Rows Fetched: 1 in 0.007 seconds

	FUNDING_ID	FUNDER	FUNDING_AMOUNT
1	1	Fenty Beauty	5000

```
--Question 4
```

```
SET SERVEROUTPUT ON;
```

```
DECLARE
```

```
    CURSOR return_cursor IS
```

```
        SELECT
```

```
            c.FIRST_NAME || ', ' || c.SURNAME AS CUSTOMER_NAME,
```

```
            dn.DONATION,
```

```
            dn.PRICE,
```

```
            r.REASON
```

```
        FROM
```

```
            RETURNS r
```

```

        JOIN
            CUSTOMER c ON r.CUSTOMER_ID = c.CUSTOMER_ID
        JOIN
            DONATION dn ON r.DONATION_ID = dn.DONATION_ID;

BEGIN
    FOR return_rec IN return_cursor LOOP
        DBMS_OUTPUT.PUT_LINE('CUSTOMER: ' || return_rec.CUSTOMER_NAME);
        DBMS_OUTPUT.PUT_LINE('DONATION PURCHASED: ' ||
return_rec.DONATION);
        DBMS_OUTPUT.PUT_LINE('PRICE: ' || return_rec.PRICE);
        DBMS_OUTPUT.PUT_LINE('RETURN REASON: ' || return_rec.REASON);
        DBMS_OUTPUT.PUT_LINE('-----
');
    END LOOP;
END;

```

```

CUSTOMER: Jack, Smith
DONATION PURCHASED: JVC Surround Sound System
PRICE: 179
RETURN REASON: Customer not satisfied with product
-----
CUSTOMER: Hendricks, Clark
DONATION PURCHASED: 6 Seat Dining room table
PRICE: 799
RETURN REASON: Product had broken section
-----

PL/SQL procedure successfully completed.

```

--Question 5

```
SET SERVEROUTPUT ON;
```

```
DECLARE
```

```

    v_customer_name VARCHAR2(100);
    v_employee_name VARCHAR2(100);
    v_donation VARCHAR2(100);
    v_dispatch_date DATE;

```

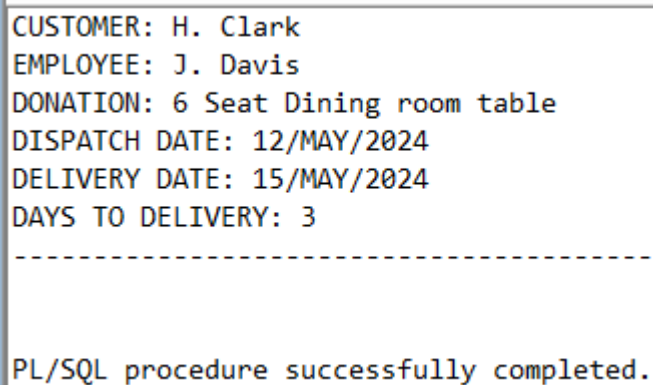
```
v_delivery_date DATE;
v_days_to_delivery NUMBER;
BEGIN
SELECT
    SUBSTR(c.FIRST_NAME, 1, 1) || '. ' || c.SURNAME,
    SUBSTR(e.FIRST_NAME, 1, 1) || '. ' || e.SURNAME,
    dn.DONATION,
    d.DISPATCH_DATE,
    d.DELIVERY_DATE,
    d.DELIVERY_DATE - d.DISPATCH_DATE AS DAYS_TO_DELIVERY
INTO
    v_customer_name,
    v_employee_name,
    v_donation,
    v_dispatch_date,
    v_delivery_date,
    v_days_to_delivery
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 dn ON i.DONATION_ID = dn.DONATION_ID
JOIN
    DELIVERY d ON i.DELIVERY_ID = d.DELIVERY_ID
WHERE
    c.CUSTOMER_ID = 11013;

DBMS_OUTPUT.PUT_LINE('CUSTOMER: ' || v_customer_name);
DBMS_OUTPUT.PUT_LINE('EMPLOYEE: ' || v_employee_name);
```

```

DBMS_OUTPUT.PUT_LINE('DONATION: ' || v_donation);
DBMS_OUTPUT.PUT_LINE('DISPATCH DATE: ' || TO_CHAR(v_dispatch_date,
'DD/MON/YYYY'));
DBMS_OUTPUT.PUT_LINE('DELIVERY DATE: ' || TO_CHAR(v_delivery_date,
'DD/MON/YYYY'));
DBMS_OUTPUT.PUT_LINE('DAYS TO DELIVERY: ' || v_days_to_delivery);
DBMS_OUTPUT.PUT_LINE('-----');
END;

```



```

CUSTOMER: H. Clark
EMPLOYEE: J. Davis
DONATION: 6 Seat Dining room table
DISPATCH DATE: 12/MAY/2024
DELIVERY DATE: 15/MAY/2024
DAYS TO DELIVERY: 3
-----
PL/SQL procedure successfully completed.

```

--Question 6

```

SET SERVEROUTPUT ON;

DECLARE
    CURSOR customer_cursor IS
        SELECT
            c.FIRST_NAME,
            c.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;

v_first_name VARCHAR2(50);
v_surname VARCHAR2(50);
v_total_amount NUMBER;
BEGIN
    FOR customer_rec IN customer_cursor LOOP
        v_first_name := customer_rec.FIRST_NAME;
        v_surname := customer_rec.SURNAME;
        v_total_amount := customer_rec.TOTAL_AMOUNT;

        IF v_total_amount >= 1500 THEN
            DBMS_OUTPUT.PUT_LINE('FIRST NAME: ' || v_first_name);
            DBMS_OUTPUT.PUT_LINE('SURNAME: ' || v_surname);
            DBMS_OUTPUT.PUT_LINE('AMOUNT: R' || v_total_amount || ' (*)');
        ELSE
            DBMS_OUTPUT.PUT_LINE('FIRST NAME: ' || v_first_name);
            DBMS_OUTPUT.PUT_LINE('SURNAME: ' || v_surname);
            DBMS_OUTPUT.PUT_LINE('AMOUNT: R' || v_total_amount);
        END IF;
        DBMS_OUTPUT.PUT_LINE('-----');
    END LOOP;
END;
```

```
FIRST NAME: Jack  
SURNAME: Smith  
AMOUNT: R1798 (*)  
-----
```

```
FIRST NAME: Pat  
SURNAME: Andre  
AMOUNT: R1299  
-----
```

```
FIRST NAME: Lucy  
SURNAME: Williams  
AMOUNT: R1778 (*)  
-----
```

```
FIRST NAME: Hendricks  
SURNAME: Clark  
AMOUNT: R799  
-----
```

--Question 7

--7.1

DECLARE

 v_customer_name CUSTOMER.FIRST_NAME%TYPE; -- Declaring a variable
with the same data type as the FIRST_NAME column in the CUSTOMER table

BEGIN

 -- Assigning a value to the variable

 v_customer_name := 'Jack';

 -- Displaying the value

 DBMS_OUTPUT.PUT_LINE('Customer Name: ' || v_customer_name);

END;

```
Customer Name: Jack
```

```
PL/SQL procedure successfully completed.
```

--7.2

DECLARE

 v_customer_record CUSTOMER%ROWTYPE; -- Declaring a record variable
with the same structure as the CUSTOMER table

BEGIN

 -- Fetching a row from the CUSTOMER table into the record variable


```
SELECT * INTO v_customer_record FROM CUSTOMER WHERE CUSTOMER_ID =  
11011;
```

```
-- Displaying the values of the fetched row
```

```
DBMS_OUTPUT.PUT_LINE('Customer ID: ' ||  
v_customer_record.CUSTOMER_ID);
```

```
DBMS_OUTPUT.PUT_LINE('First Name: ' || v_customer_record.FIRST_NAME);
```

```
DBMS_OUTPUT.PUT_LINE('Surname: ' || v_customer_record.SURNAME);
```

```
-- Other columns can be similarly accessed
```

```
END;
```

```
Customer ID: 11011
```

```
First Name: Jack
```

```
Surname: Smith
```

```
PL/SQL procedure successfully completed.
```

```
--7.3
```

```
DECLARE
```

```
-- Declaring a user-defined exception
```

```
ex_custom_exception EXCEPTION;
```

```
BEGIN
```

```
-- Checking a condition and raising the user-defined exception if it's  
true
```

```
IF 1 = 1 THEN
```

```
    RAISE ex_custom_exception;
```

```
END IF;
```

```
EXCEPTION
```

```
    WHEN ex_custom_exception THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Custom exception raised.');
```

```
END;
```

```
Custom exception raised.
```

```
PL/SQL procedure successfully completed.
```