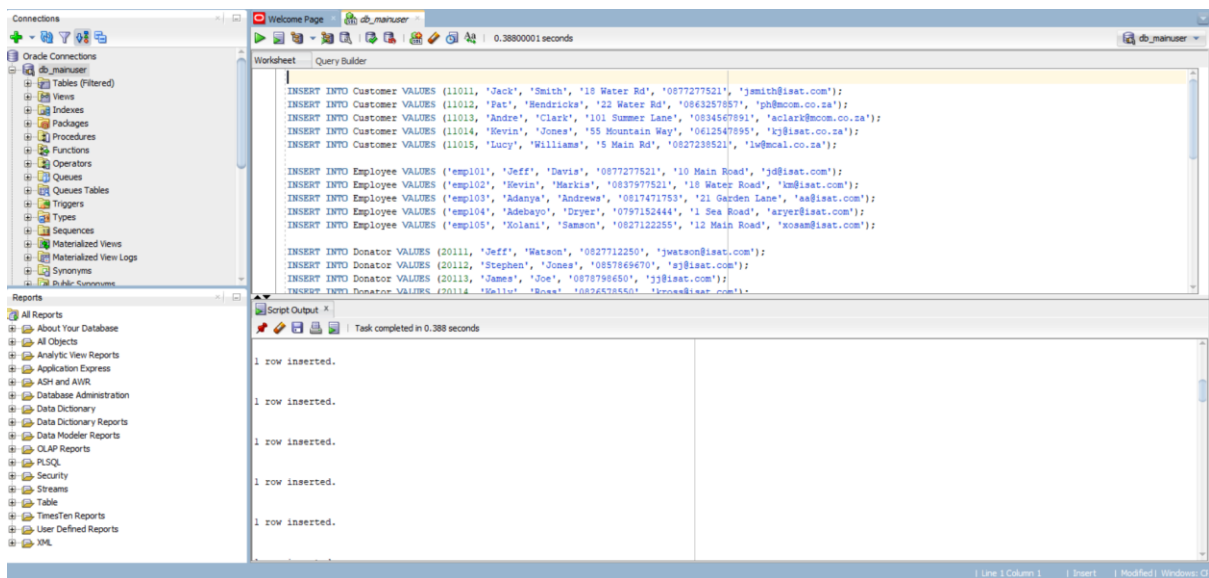
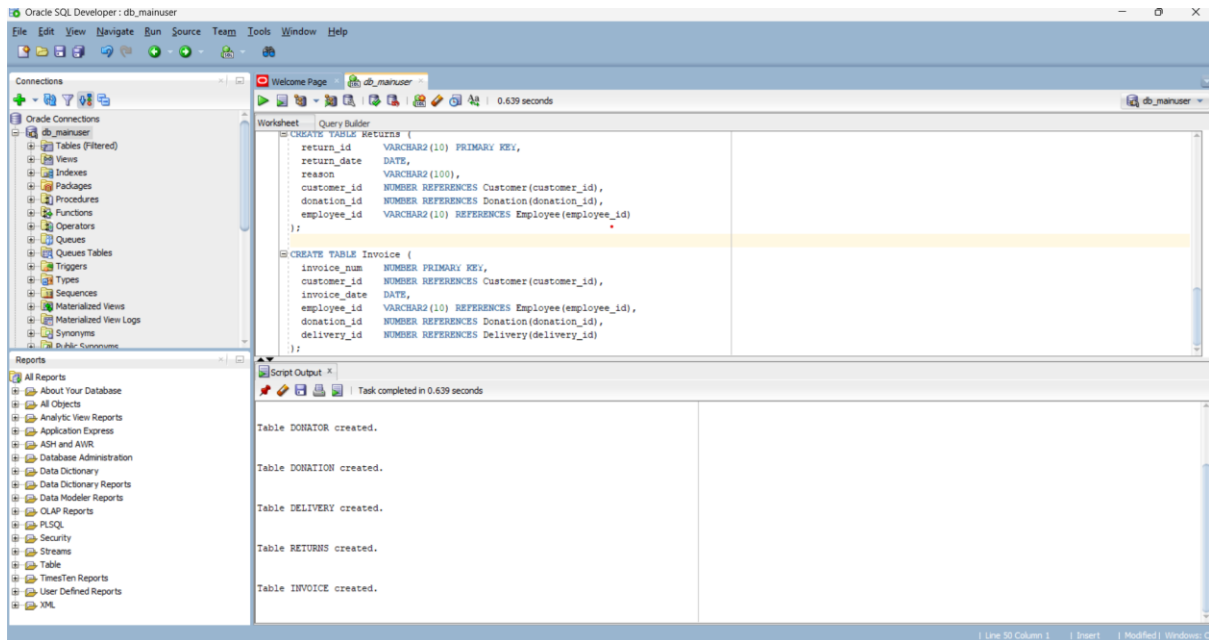


Assignment 2

ST10074970

Isaac Phiri

Question one



CREATE TABLE Customer (
customer_id NUMBER PRIMARY KEY,
first_name VARCHAR2(30),
surname VARCHAR2(30),

```
address    VARCHAR2(50),  
contact_number VARCHAR2(15),  
email      VARCHAR2(50)  
);
```

```
CREATE TABLE Employee (  
    employee_id  VARCHAR2(10) PRIMARY KEY,  
    first_name   VARCHAR2(30),  
    surname      VARCHAR2(30),  
    contact_number VARCHAR2(15),  
    address      VARCHAR2(50),  
    email        VARCHAR2(50)  
);
```

```
CREATE TABLE Donator (  
    donator_id   NUMBER PRIMARY KEY,  
    first_name   VARCHAR2(30),  
    surname      VARCHAR2(30),  
    contact_number VARCHAR2(15),  
    email        VARCHAR2(50)  
);
```

```
CREATE TABLE Donation (  
    donation_id  NUMBER PRIMARY KEY,  
    donator_id   NUMBER REFERENCES Donator(donator_id),  
    donation      VARCHAR2(50),  
    price         VARCHAR2(10),  
    donation_date DATE
```

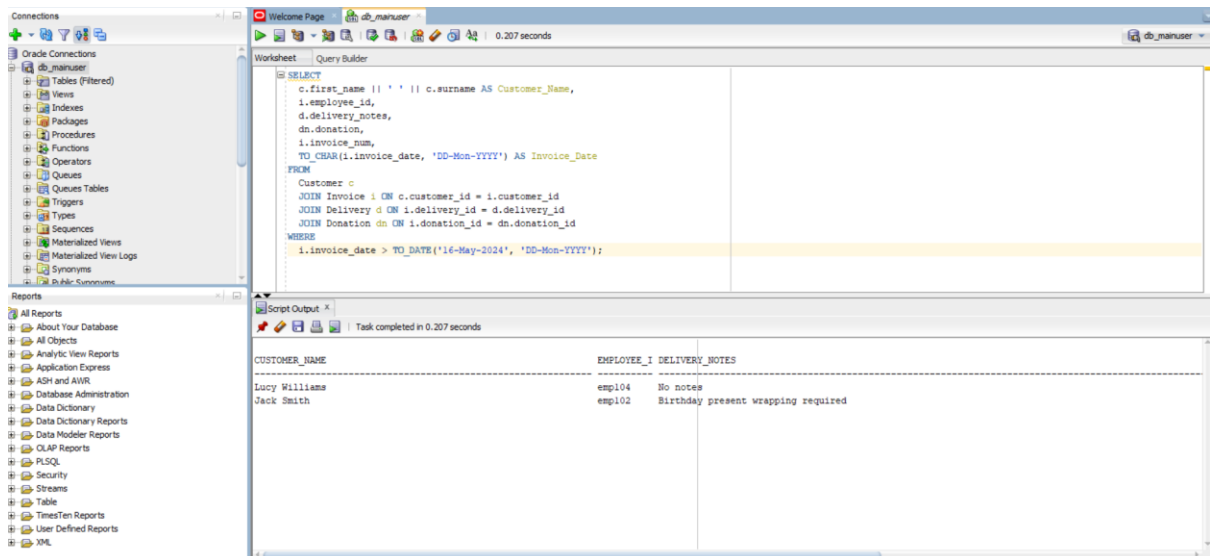
);

```
CREATE TABLE Delivery (  
    delivery_id NUMBER PRIMARY KEY,  
    delivery_notes VARCHAR2(100),  
    dispatch_date DATE,  
    delivery_date DATE  
);
```

```
CREATE TABLE Returns (  
    return_id VARCHAR2(10) PRIMARY KEY,  
    return_date DATE,  
    reason VARCHAR2(100),  
    customer_id NUMBER REFERENCES Customer(customer_id),  
    donation_id NUMBER REFERENCES Donation(donation_id),  
    employee_id VARCHAR2(10) REFERENCES Employee(employee_id)  
);
```

```
CREATE TABLE Invoice (  
    invoice_num NUMBER PRIMARY KEY,  
    customer_id NUMBER REFERENCES Customer(customer_id),  
    invoice_date DATE,  
    employee_id VARCHAR2(10) REFERENCES Employee(employee_id),  
    donation_id NUMBER REFERENCES Donation(donation_id),  
    delivery_id NUMBER REFERENCES Delivery(delivery_id)  
);
```

Question two



SELECT

SELECT

c.first_name || ' ' || c.surname AS Customer_Name,

i.employee_id,

d.delivery_notes,

dn.donation,

i.invoice_num,

TO_CHAR(i.invoice_date, 'DD-Mon-YYYY') AS Invoice_Date

FROM

Customer c

JOIN Invoice i ON c.customer_id = i.customer_id

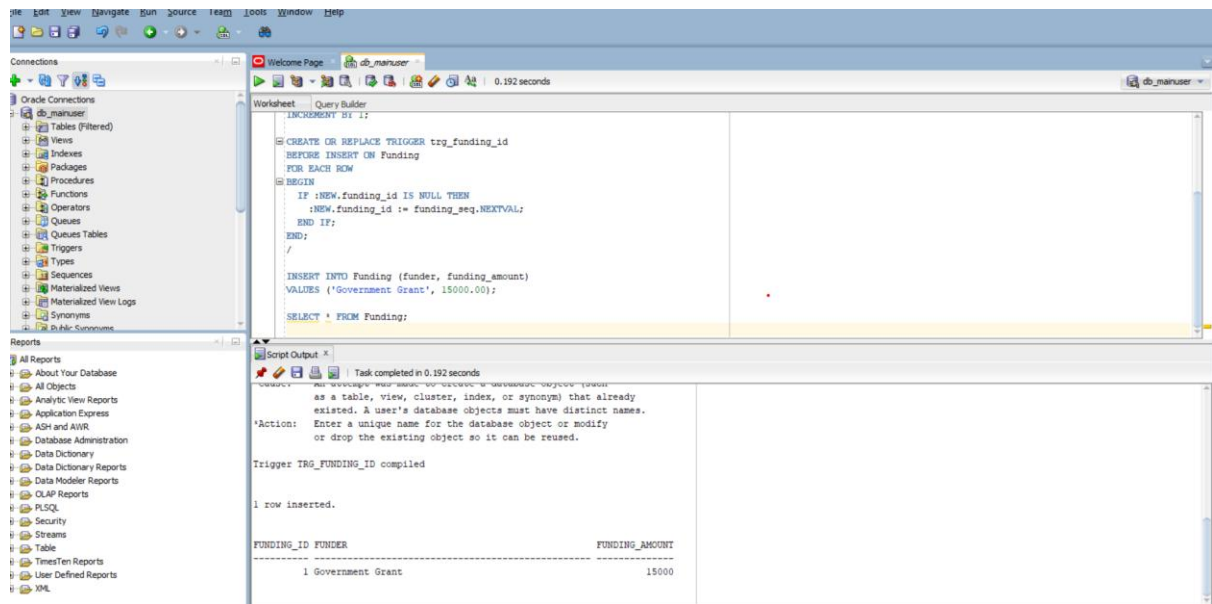
JOIN Delivery d ON i.delivery_id = d.delivery_id

JOIN Donation dn ON i.donation_id = dn.donation_id

WHERE

i.invoice_date > TO_DATE('16-May-2024', 'DD-Mon-YYYY');

Question 3



```

CREATE TABLE Funding (
    funding_id  NUMBER PRIMARY KEY,
    funder      VARCHAR2(50),
    funding_amount NUMBER(10,2)
);

```

```

CREATE SEQUENCE funding_seq
START WITH 1
INCREMENT BY 1;

```

```

CREATE OR REPLACE TRIGGER trg_funding_id
BEFORE INSERT ON Funding
FOR EACH ROW
BEGIN
    IF :NEW.funding_id IS NULL THEN
        :NEW.funding_id := funding_seq.NEXTVAL;
    END IF;
END;
/

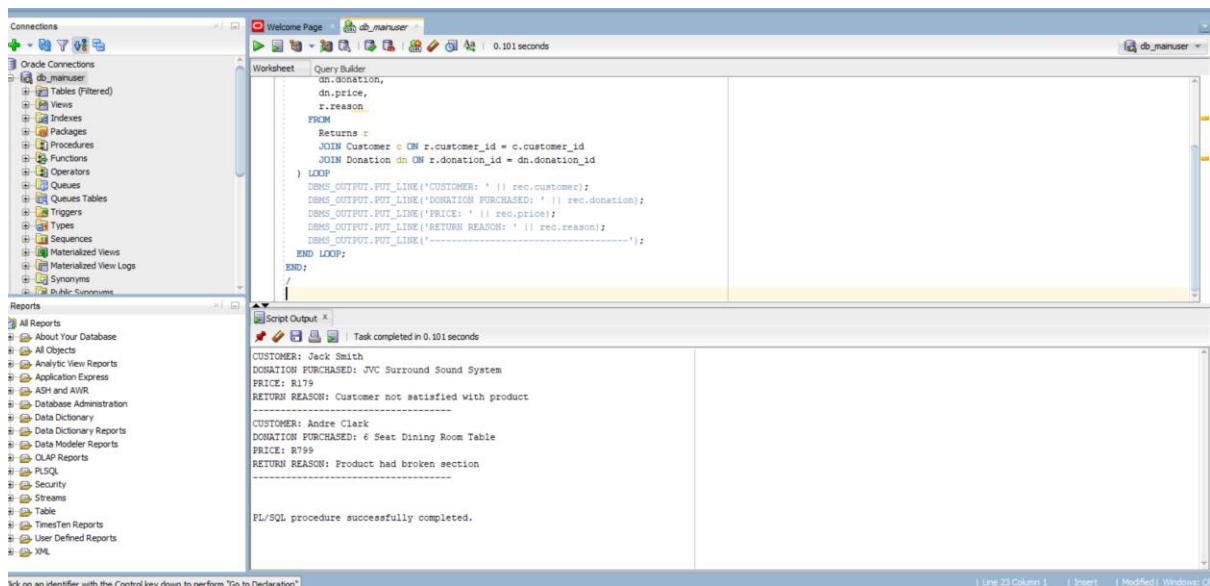
```

INSERT INTO Funding (funder, funding_amount)

VALUES ('Government Grant', 15000.00);

SELECT * FROM Funding;

Question four



SET SERVEROUTPUT ON;

BEGIN

FOR rec IN (

SELECT

COALESCE(c.first_name,'') || ' ' || COALESCE(c.surname,'') AS customer,

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
    ) LOOP

        DBMS_OUTPUT.PUT_LINE('CUSTOMER: ' || rec.customer);

        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('-----');

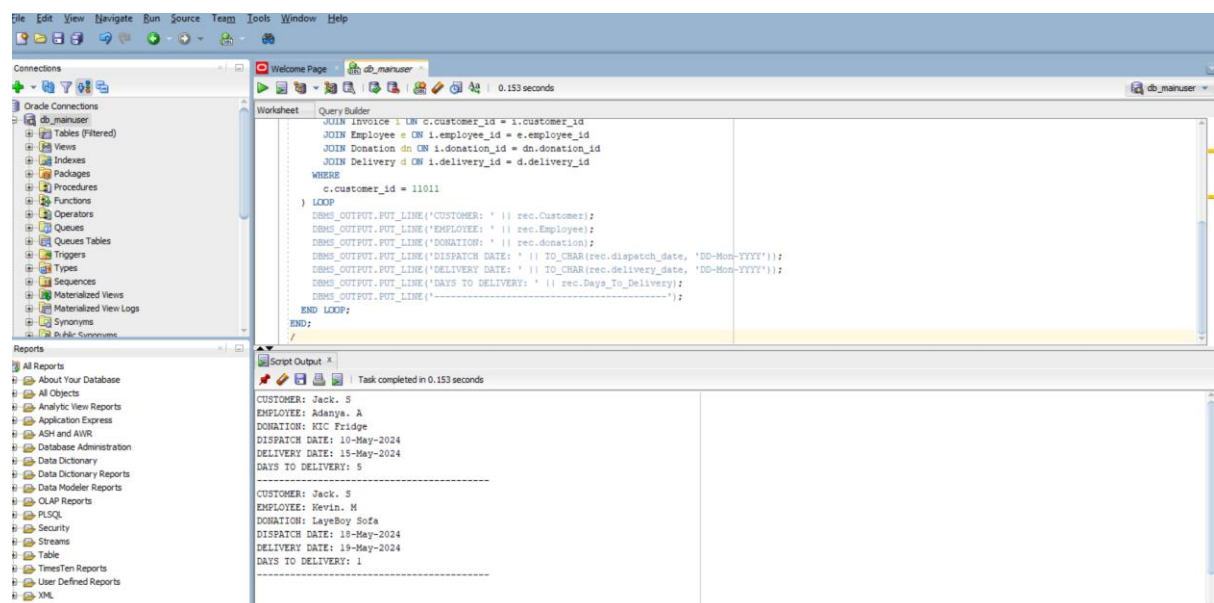
    END LOOP;

END;

/

```

Question 5



The screenshot shows the Oracle SQL Developer interface. The 'Query Builder' window displays a PL/SQL script that joins the INVOICE, EMPLOYEE, DONATION, and DELIVERY tables for a specific customer (c.customer_id = 11011). The script uses a loop to iterate through the results and prints out details for each record, including customer, employee, donation, dispatch date, delivery date, and days to delivery.

The 'Script Output' window shows the results of the script execution, which completed in 0.153 seconds. It displays two records of data:

Customer	Employee	Donation	Dispatch Date	Delivery Date	Days to Delivery
Jack. S	Adanya. A	KIC Fridge	10-May-2024	15-May-2024	5
Jack. S	Hevin. M	LayeBoy Sofa	15-May-2024	19-May-2024	4

SET SERVEROUTPUT ON;

BEGIN

FOR rec IN (

SELECT

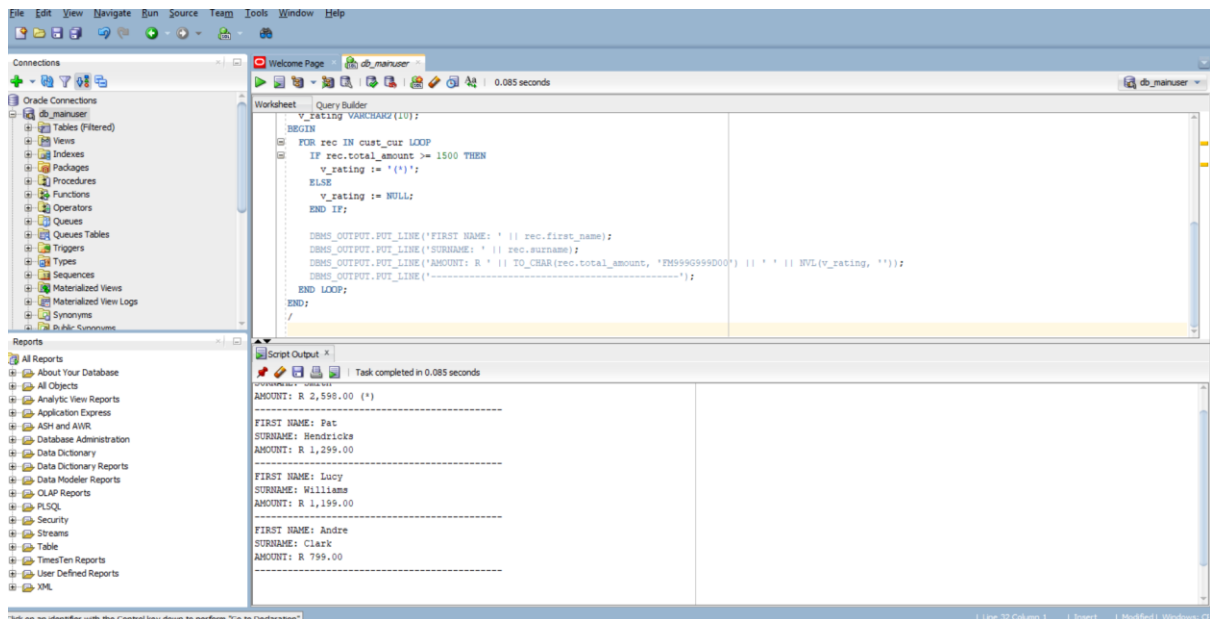
c.first_name || ' ' || SUBSTR(c.surname, 1, 1) AS Customer,

```

e.first_name || ' ' || SUBSTR(e.surname, 1, 1) AS Employee,
dn.donation,
d.dispatch_date,
d.delivery_date,
TRUNC(d.delivery_date - d.dispatch_date) AS 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 = 11011
) LOOP
DBMS_OUTPUT.PUT_LINE('CUSTOMER: ' || rec.Customer);
DBMS_OUTPUT.PUT_LINE('EMPLOYEE: ' || rec.Employee);
DBMS_OUTPUT.PUT_LINE('DONATION: ' || rec.donation);
DBMS_OUTPUT.PUT_LINE('DISPATCH DATE: ' || TO_CHAR(rec.dispatch_date, 'DD-
Mon-YYYY'));
DBMS_OUTPUT.PUT_LINE('DELIVERY DATE: ' || TO_CHAR(rec.delivery_date, 'DD-Mon-
YYYY'));
DBMS_OUTPUT.PUT_LINE('DAYS TO DELIVERY: ' || rec.Days_To_Delivery);
DBMS_OUTPUT.PUT_LINE('-----');
END LOOP;
END;
/

```

Question Six



SET SERVEROUTPUT ON;

DECLARE

CURSOR cust_cur IS

SELECT

c.first_name,

c.surname,

SUM(TO_NUMBER(REPLACE(REPLACE(d.price, 'R', ''), ', ', ''))) 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_rating VARCHAR2(10);

BEGIN

FOR rec IN cust_cur LOOP

```
IF rec.total_amount >= 1500 THEN
```

```
    v_rating := '('*);
```

```
ELSE
```

```
    v_rating := NULL;
```

```
END IF;
```

```
DBMS_OUTPUT.PUT_LINE('FIRST NAME: ' || rec.first_name);
```

```
DBMS_OUTPUT.PUT_LINE('SURNAME: ' || rec.surname);
```

```
DBMS_OUTPUT.PUT_LINE('AMOUNT: R ' || TO_CHAR(rec.total_amount,  
'FM999G999D00') || ' ' || NVL(v_rating, ''));
```

```
DBMS_OUTPUT.PUT_LINE('-----');
```

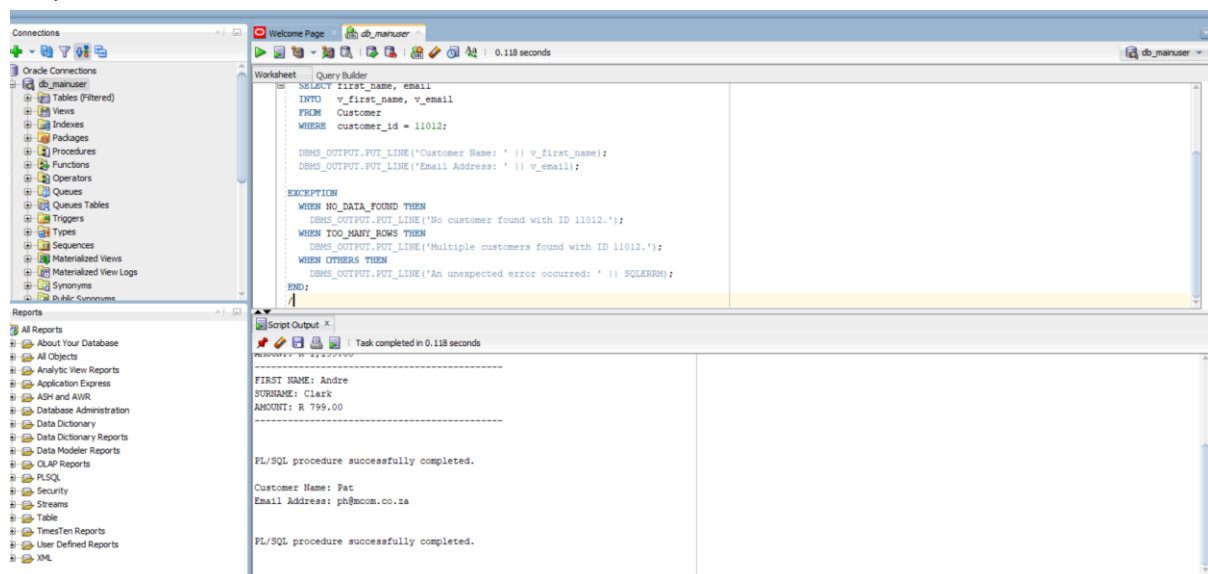
```
END LOOP;
```

```
END;
```

```
/
```

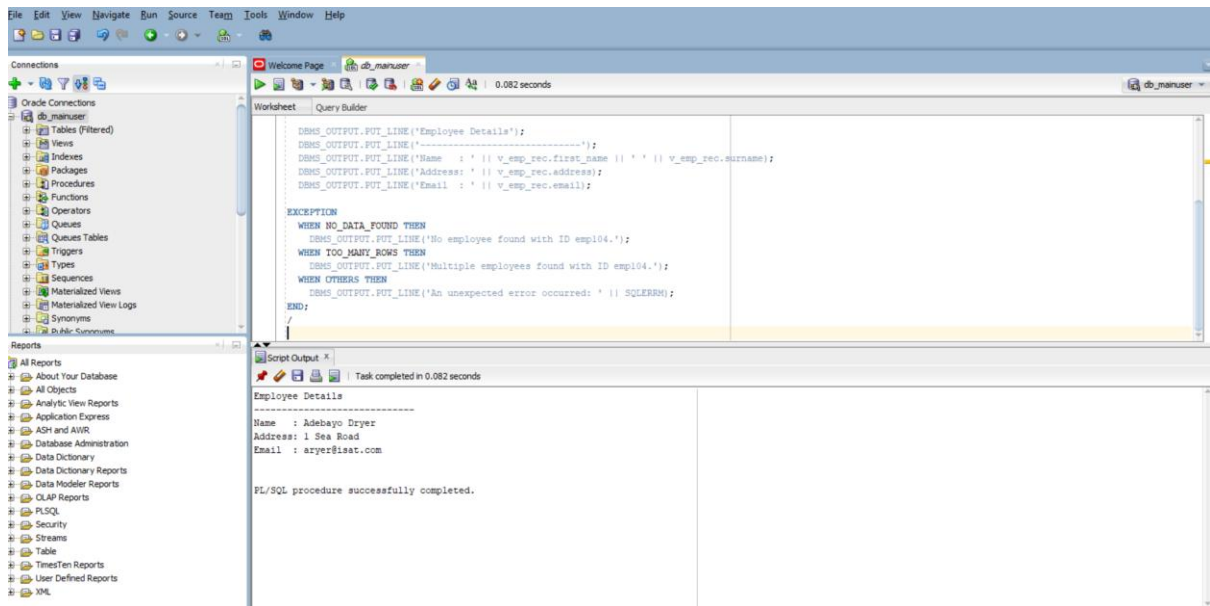
Question Seven

7.1)



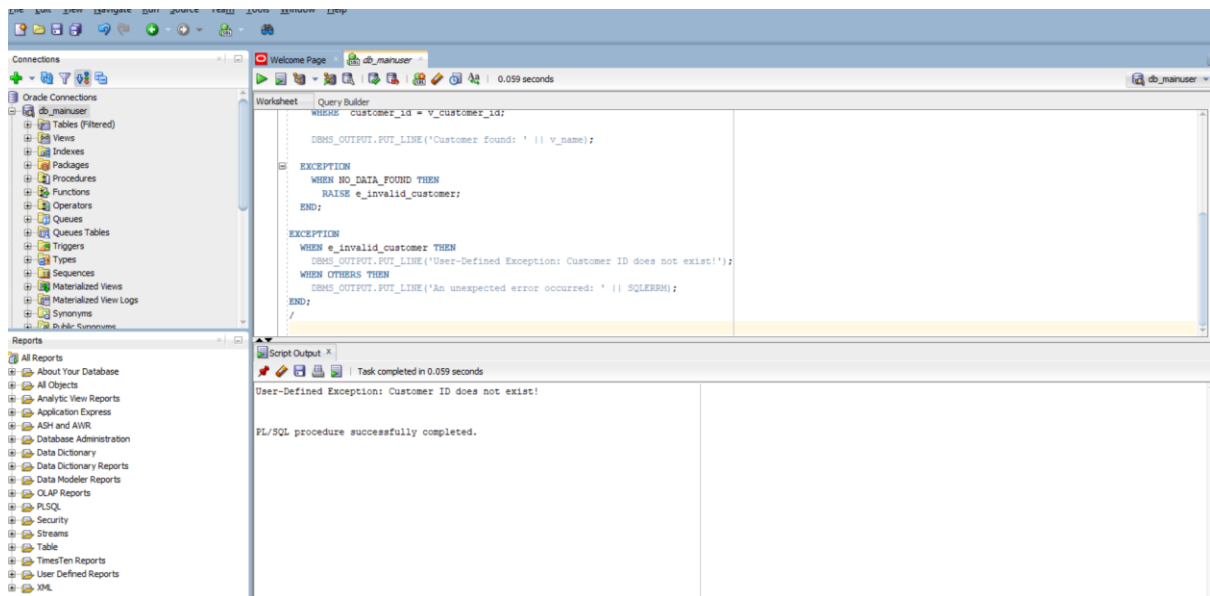
%TYPE allows the variable to match the datatype of the columns linked

7.2)



%ROWTYPE pulls a entire table row into one record variable

7.3)



Question Eight

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page db_manuser 0.034 seconds

Oracle Connections
db_manuser
Tables (Filtered)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Materialized Views
Materialized View Logs
Synonyms
DB Public Synonyms

Worksheet Query Builder
0.034 seconds

```
WHEN AMOUNT BETWEEN 1000 AND 1499 THEN '*'
ELSE '*'
END AS customer_rating
FROM (
SELECT
c.first_name,
c.surname,
SUM(TO_NUMBER(REPLACE(REPLACE(d.price, 'R', ''), ','))) AS 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 amount DESC;
```

Reports
All Reports
About Your Database
All Objects
Analytic View Reports
Application Express
ASH and AWR
Database Administration
Data Dictionary
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
PL/SQL
Security
Streams
Table
TimesTen Reports
User Defined Reports
XML

Script Output x
Task completed in 0.034 seconds
User-Defined Exception: Customer ID does not exist!
PL/SQL procedure successfully completed.

FIRST_NAME	SURNAME	AMOUNT	C
Jack	Smith	2598	*
Pat	Hendricks	1299	
Lucy	Williams	1199	
Andre	Clark	799	*