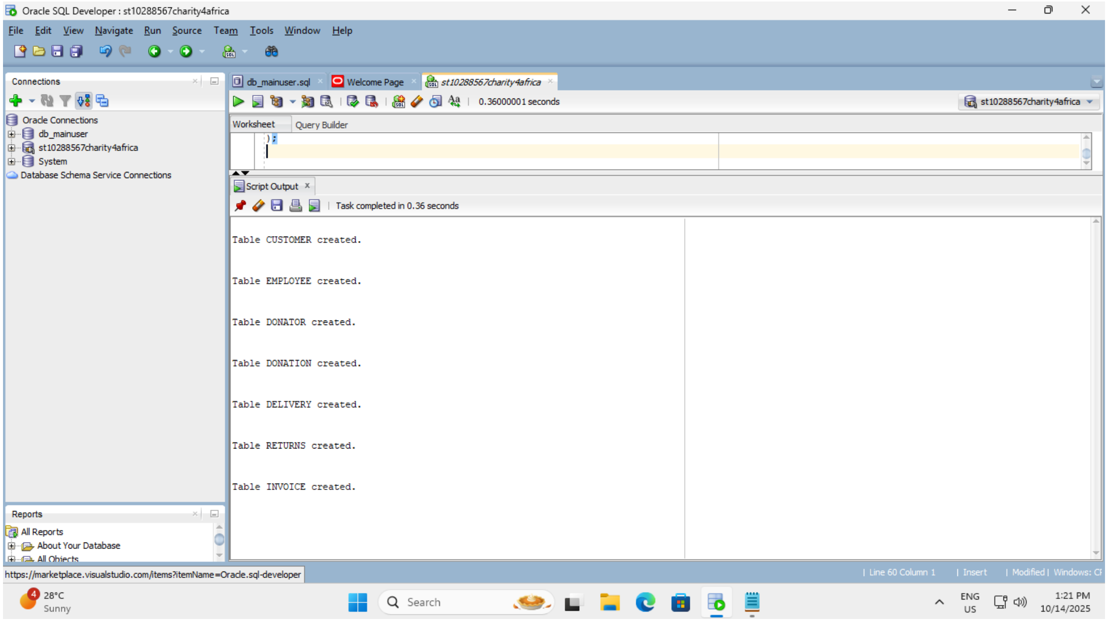
INSY7213-PRACTICAL ASSIGMENT 2

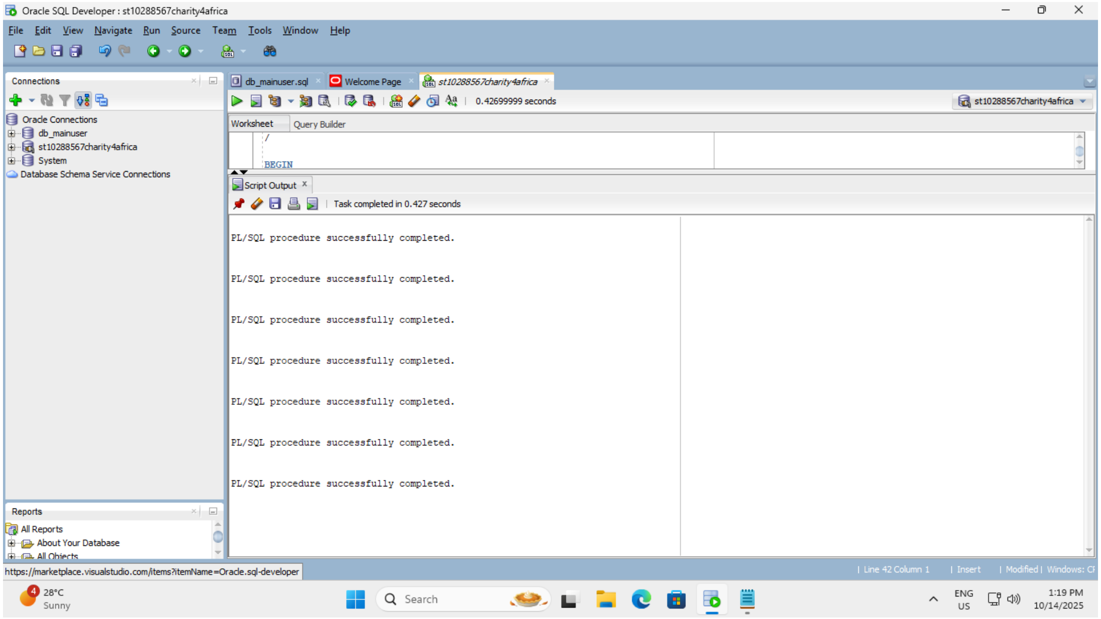
ST10288567

Darsh Somayi

Question 1 image:

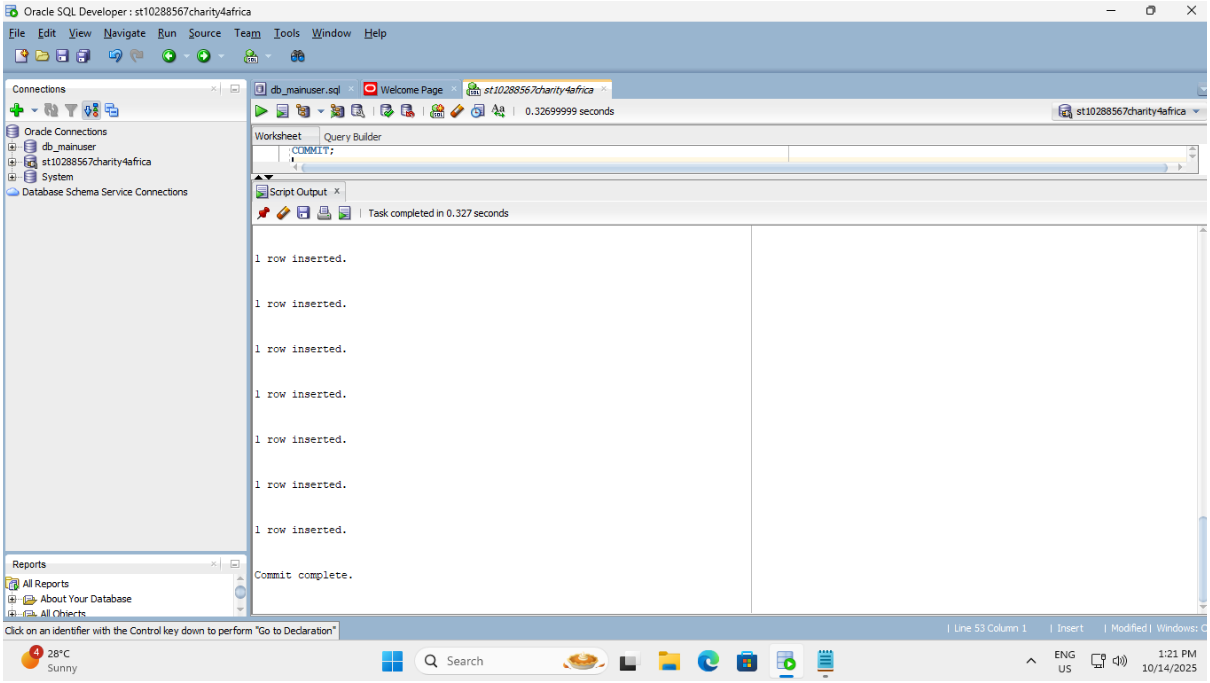


```
Table CUSTOMER created.


Table EMPLOYEE created.


Table DONATOR created.


Table DONATION created.


Table DELIVERY created.


Table RETURNS created.


Table INVOICE created.
```



```
/

BEGIN
```

```
PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
```



```
COMMIT;
```

Task completed in 0.327 seconds

```
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

Commit complete.
```

(Oracle Corporation, 2023).

QUESTION 1 — code for Creating and Populate Tables

Step 1:objects dropped code

```
BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE Invoice CASCADE CONSTRAINTS';
EXCEPTION WHEN OTHERS THEN NULL;
END;
/

BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE Returns CASCADE CONSTRAINTS';
EXCEPTION WHEN OTHERS THEN NULL;
END;/

BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE Delivery CASCADE CONSTRAINTS';
```

```
EXCEPTION WHEN OTHERS THEN NULL;
END;
/

BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE Donation CASCADE CONSTRAINTS';
EXCEPTION WHEN OTHERS THEN NULL;
END;
/

BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE Donator CASCADE CONSTRAINTS';
EXCEPTION WHEN OTHERS THEN NULL;
END;
/

BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE Employee CASCADE CONSTRAINTS';
EXCEPTION WHEN OTHERS THEN NULL;
END;
/

BEGIN
  EXECUTE IMMEDIATE 'DROP TABLE Customer CASCADE CONSTRAINTS';
EXCEPTION WHEN OTHERS THEN NULL;
END;/
```

Step 2 — Created tables code

```
CREATE TABLE Customer (
  Customer_ID NUMBER(5) PRIMARY KEY,
  First_Name VARCHAR2(20),
  Surname VARCHAR2(20),
  Address VARCHAR2(50),
  Contact_Number VARCHAR2(15),
  Email VARCHAR2(40)
);

CREATE TABLE Employee (
  Employee_ID VARCHAR2(10) PRIMARY KEY,
  First_Name VARCHAR2(20),
  Surname VARCHAR2(20),
  Contact_Number VARCHAR2(15),
  Address VARCHAR2(50),
  Email VARCHAR2(40)
);

CREATE TABLE Donator (
  Donator_ID NUMBER(6) PRIMARY KEY,
  First_Name VARCHAR2(20),
  Surname VARCHAR2(20),
  Contact_Number VARCHAR2(15),
  Email VARCHAR2(40)
);

CREATE TABLE Donation (
  Donation_ID NUMBER(6) PRIMARY KEY,
  Donator_ID NUMBER(6),
  Donation VARCHAR2(50),
  Price NUMBER(6),
  Donation_Date DATE,
  FOREIGN KEY (Donator_ID) REFERENCES Donator(Donator_ID)
```

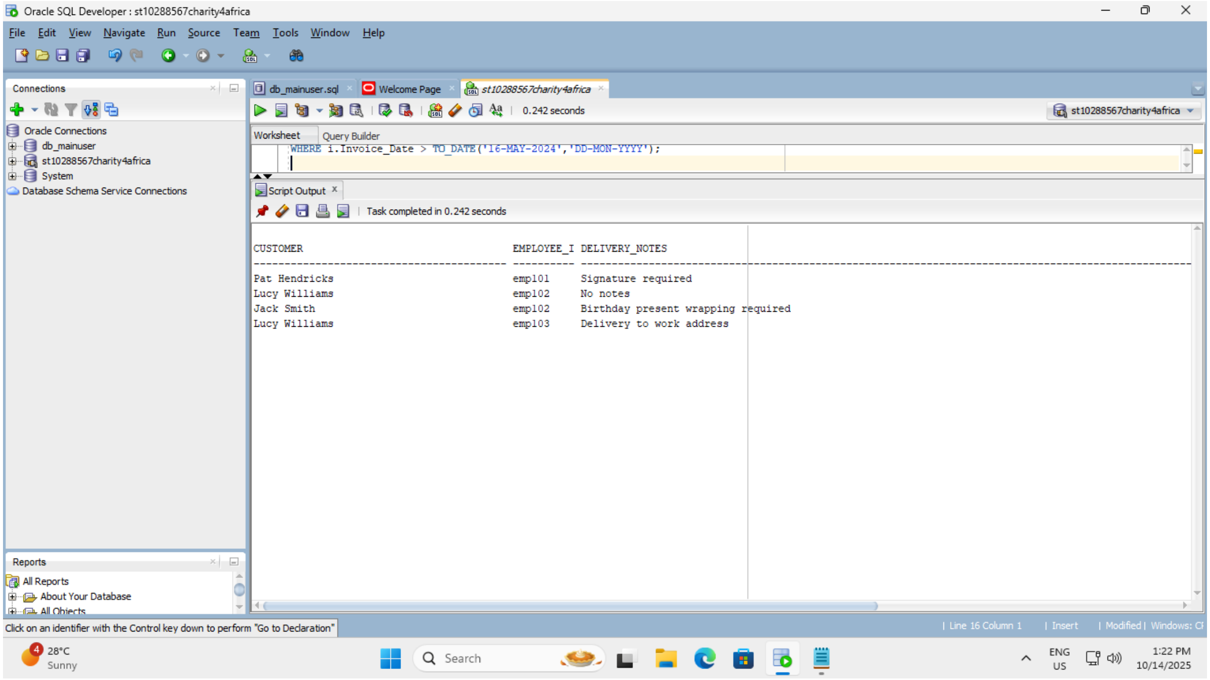
```
);

CREATE TABLE Delivery (
  Delivery_ID NUMBER(5) 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(5),
  Donation_ID NUMBER(6),
  Employee_ID VARCHAR2(10)
);

CREATE TABLE Invoice (
  Invoice_Num NUMBER(6) PRIMARY KEY,
  Customer_ID NUMBER(5),
  Invoice_Date DATE,
  Employee_ID VARCHAR2(10),
  Donation_ID NUMBER(6),
  Delivery_ID NUMBER(5)
);
```

Step 3 — Insert data code

```
-- CUSTOMER

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

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

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

INSERT INTO Customer VALUES (11014,'Kevin','Jones','55 Mountain
Way','0612547895','kj@isat.co.za');

INSERT INTO Customer VALUES (11015,'Lucy','Williams','5 Main
Rd','0827238521','lw@mcal.co.za');


-- EMPLOYEE

INSERT INTO Employee VALUES ('emp101','Jeff','Davis','0877277521','10 Main
Rd','jand@isat.com');

INSERT INTO Employee VALUES ('emp102','Kevin','Marks','0837773521','18 Water
Rd','km@isat.com');

INSERT INTO Employee VALUES ('emp103','Adanya','Andrews','0811717523','21 Circle
Lane','aa@isat.com');

INSERT INTO Employee VALUES ('emp104','Adebayo','Dryer','0797151244','1 Sea
Rd','aryer@isat.com');

INSERT INTO Employee VALUES ('emp105','Xolani','Samson','0827122255','12 Main
Rd','xosam@isat.com');


-- DONATOR

INSERT INTO Donator VALUES (20111,'Jeff','Watson','0827172250','jwatson@gmail.com');

INSERT INTO Donator VALUES (20112,'Stephen','Jones','0837865670','jones@gmail.com');

INSERT INTO Donator VALUES (20113,'James','Joe','0878798650','jj@isat.com');

INSERT INTO Donator VALUES (20114,'Kelly','Ross','0825675650','kross@gsat.com');

INSERT INTO Donator VALUES (20115,'Abraham','Clark','0797656430','aclark@gmail.com');


-- DONATION

INSERT INTO Donation VALUES (7111,20111,'KIC Fridge',599,TO_DATE('01-MAY-
2024','DD-MON-YYYY'));

INSERT INTO Donation VALUES (7112,20112,'Samsung 42inch LCD',1299,TO_DATE('03-
MAY-2024','DD-MON-YYYY'));
```

```sql
INSERT INTO Donation VALUES (7113,20113,'Sharp Microwave',1599,TO_DATE('03-MAY-2024','DD-MON-YYYY'));

INSERT INTO Donation VALUES (7114,20115,'6 Seat Dining room table',799,TO_DATE('05-MAY-2024','DD-MON-YYYY'));

INSERT INTO Donation VALUES (7115,20114,'Lazyboy Sofa',1199,TO_DATE('07-MAY-2024','DD-MON-YYYY'));

INSERT INTO Donation VALUES (7116,20113,'JVC Surround Sound System',179,TO_DATE('09-MAY-2024','DD-MON-YYYY'));


-- DELIVERY

INSERT INTO Delivery VALUES (511,'Double packaging requested',TO_DATE('10-MAY-2024','DD-MON-YYYY'),TO_DATE('15-MAY-2024','DD-MON-YYYY'));

INSERT INTO Delivery 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 VALUES (513,'Signature required',TO_DATE('12-MAY-2024','DD-MON-YYYY'),TO_DATE('17-MAY-2024','DD-MON-YYYY'));

INSERT INTO Delivery VALUES (514,'No notes',TO_DATE('12-MAY-2024','DD-MON-YYYY'),TO_DATE('15-MAY-2024','DD-MON-YYYY'));

INSERT INTO Delivery 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 VALUES (516,'Delivery to work address',TO_DATE('20-MAY-2024','DD-MON-YYYY'),TO_DATE('25-MAY-2024','DD-MON-YYYY'));


-- RETURNS

INSERT INTO Returns VALUES ('ret001',TO_DATE('25-MAY-2024','DD-MON-YYYY'),'Customer not satisfied with product',11011,7116,'emp101');

INSERT INTO Returns VALUES ('ret002',TO_DATE('25-MAY-2024','DD-MON-YYYY'),'Product had broken section',11013,7114,'emp103');


-- INVOICE

INSERT INTO Invoice VALUES (8111,11011,TO_DATE('15-MAY-2024','DD-MON-YYYY'),'emp103',7111,511);

INSERT INTO Invoice VALUES (8112,11013,TO_DATE('15-MAY-2024','DD-MON-YYYY'),'emp101',7114,512);

INSERT INTO Invoice VALUES (8113,11012,TO_DATE('17-MAY-2024','DD-MON-YYYY'),'emp101',7112,513);

INSERT INTO Invoice VALUES (8114,11015,TO_DATE('17-MAY-2024','DD-MON-YYYY'),'emp102',7113,514);
INSERT INTO Invoice VALUES (8115,11011,TO_DATE('17-MAY-2024','DD-MON-YYYY'),'emp102',7115,515);

INSERT INTO Invoice VALUES (8116,11015,TO_DATE('18-MAY-2024','DD-MON-YYYY'),'emp103',7116,516);


COMMIT;
```

Question 2 image:



```
WHERE i.Invoice_Date > TO_DATE('16-MAY-2024','DD-MON-YYYY');
```

Task completed in 0.242 seconds

```
CUSTOMER                                 EMPLOYEE_I DELIVERY_NOTES
---------------------------------------- ---------- --------------------------------------------------------------------------------
Pat Hendricks                            emp101     Signature required
Lucy Williams                            emp102     No notes
Jack Smith                               emp102     Birthday present wrapping required
Lucy Williams                            emp103     Delivery to work address
```

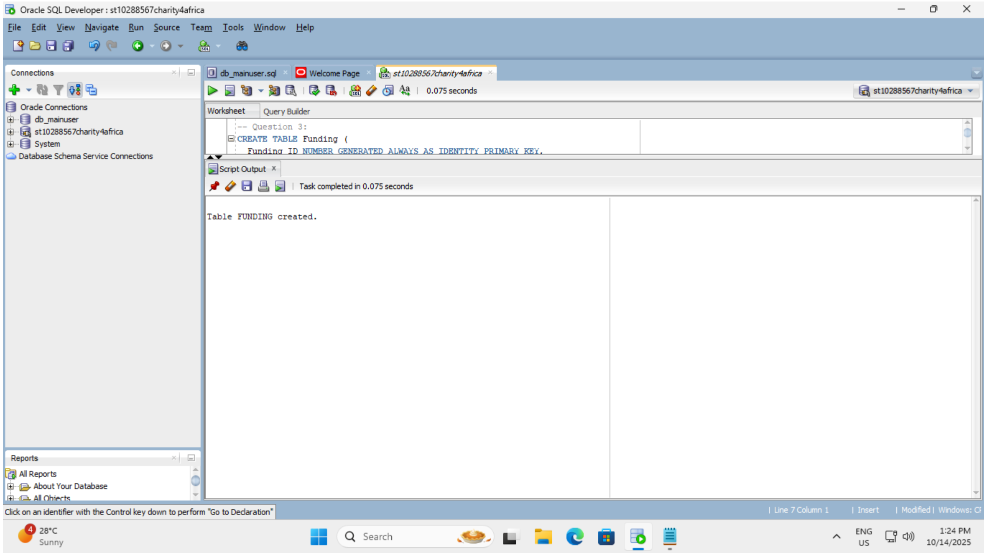
Click on an identifier with the Control key down to perform "Go to Declaration"

(Oracle Corporation, 2023).

QUESTION 2 — Report after 16 May 2024 code

```
SELECT
  c.First_Name || ' ' || c.Surname AS Customer,
  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('16-MAY-2024','DD-MON-YYYY');
```

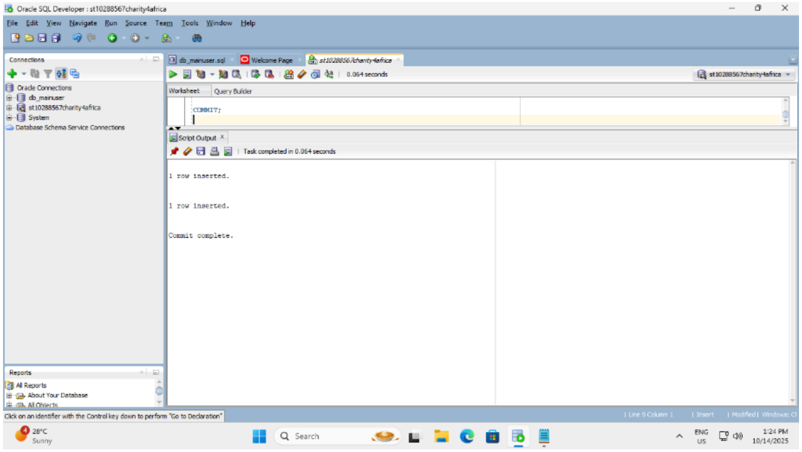
QUESTION 3



```
-- Question 3:
CREATE TABLE Funding (
  Funding_ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
```

Task completed in 0.075 seconds

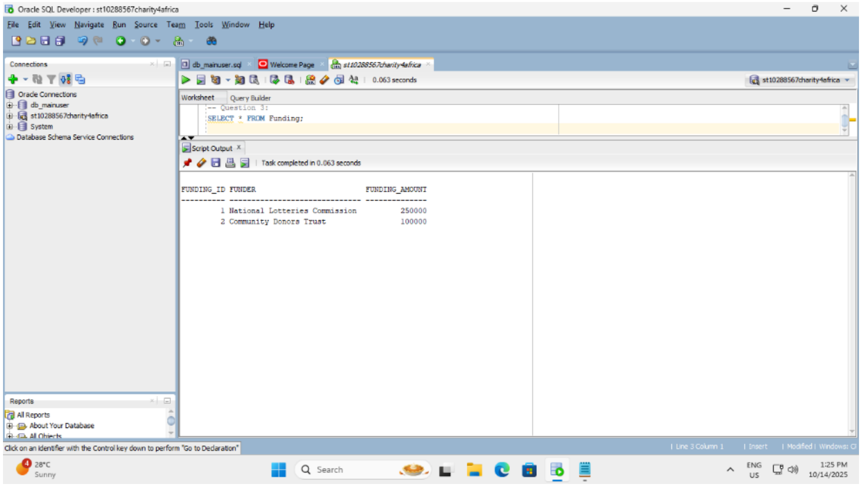
```
Table FUNDING created.
```



```
COMMIT;
```

Task completed in 0.064 seconds

```
1 row inserted.


1 row inserted.


Commit complete.
```



```
-- Question 3:
SELECT * FROM Funding;
```

Task completed in 0.063 seconds

```
FUNDING_ID FUNDER                         FUNDING_AMOUNT
---------- ------------------------------ --------------
         1 National Lotteries Commission          250000
         2 Community Donors Trust                 100000
```

(Oracle Corporation, 2023).

QUESTION 3 — Created Funding Table code

Step 1 code:

```
CREATE TABLE Funding (
  Funding_ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
  Funder VARCHAR2(30),
  Funding_Amount NUMBER(8,2)
);
```

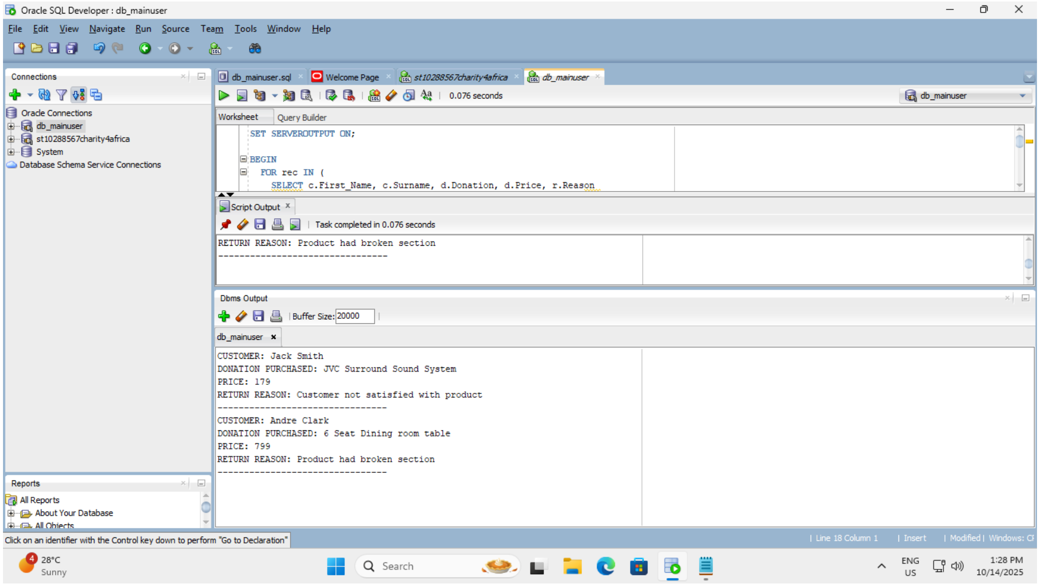
Step 2 code:

```
INSERT INTO Funding (Funder, Funding_Amount)
VALUES ('National Lotteries Commission', 250000);

INSERT INTO Funding (Funder, Funding_Amount)
VALUES ('Community Donors Trust', 100000);

COMMIT;
```

Brief comment:

```
-- The IDENTITY column automatically generates a new unique Funding_ID
-- each time data is inserted, ensuring no manual duplication.
```

QUESTION 4



```
SET SERVEROUTPUT ON;

BEGIN
  FOR rec IN (
    SELECT c.First_Name, c.Surname, d.Donation, d.Price, r.Reason
```

Task completed in 0.076 seconds

```
RETURN REASON: Product had broken section
--------------------------------
```

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

Click on an identifier with the Control key down to perform "Go to Declaration"

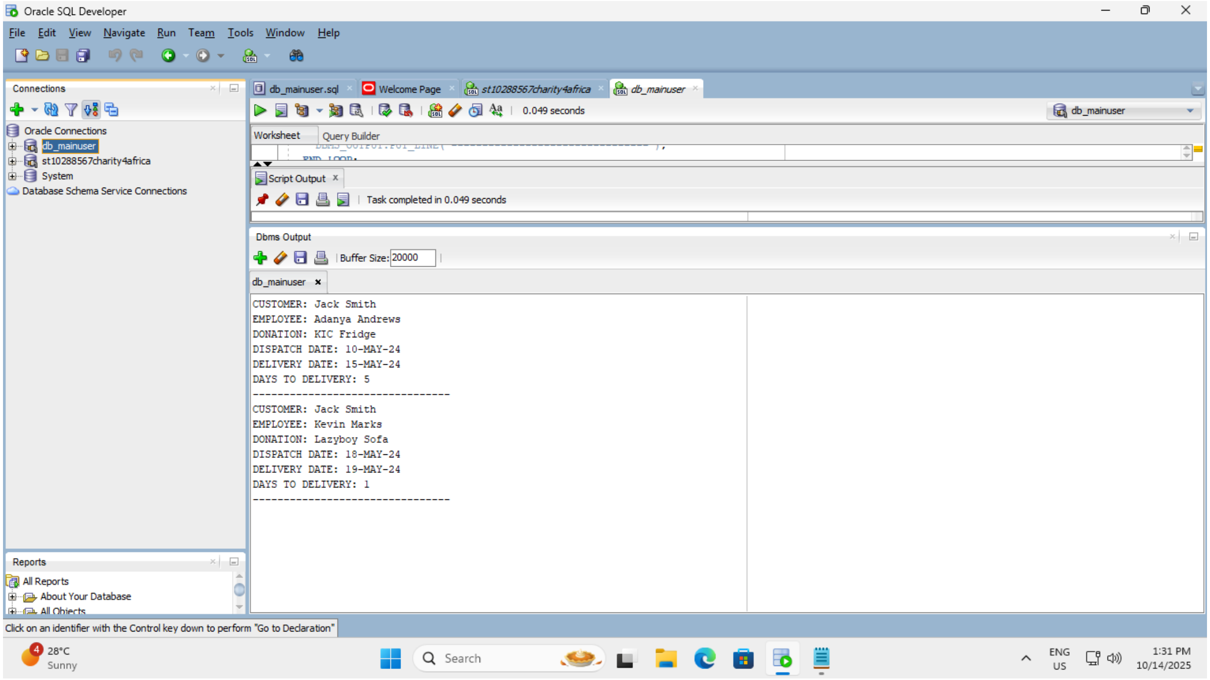
(Oracle Corporation, 2023).

QUESTION 4 — PL/SQL Query: Returns Report

server output code:

```
SET SERVEROUTPUT ON;


BEGIN
  FOR rec IN (
    SELECT c.First_Name, c.Surname, d.Donation, d.Price, r.Reason
    FROM Returns r
    JOIN Customer c ON r.Customer_ID = c.Customer_ID
    JOIN Donation d ON r.Donation_ID = d.Donation_ID
  ) LOOP
    DBMS_OUTPUT.PUT_LINE('CUSTOMER: ' || rec.First_Name || ' ' || rec.Surname);
    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



0.049 seconds

Task completed in 0.049 seconds

Buffer Size: 20000

```
CUSTOMER: Jack Smith
EMPLOYEE: Adanya Andrews
DONATION: KIC Fridge
DISPATCH DATE: 10-MAY-24
DELIVERY DATE: 15-MAY-24
DAYS TO DELIVERY: 5
--------------------------------
CUSTOMER: Jack Smith
EMPLOYEE: Kevin Marks
DONATION: Lazyboy Sofa
DISPATCH DATE: 18-MAY-24
DELIVERY DATE: 19-MAY-24
DAYS TO DELIVERY: 1
--------------------------------
```

Click on an identifier with the Control key down to perform "Go to Declaration"

(Oracle Corporation, 2023).

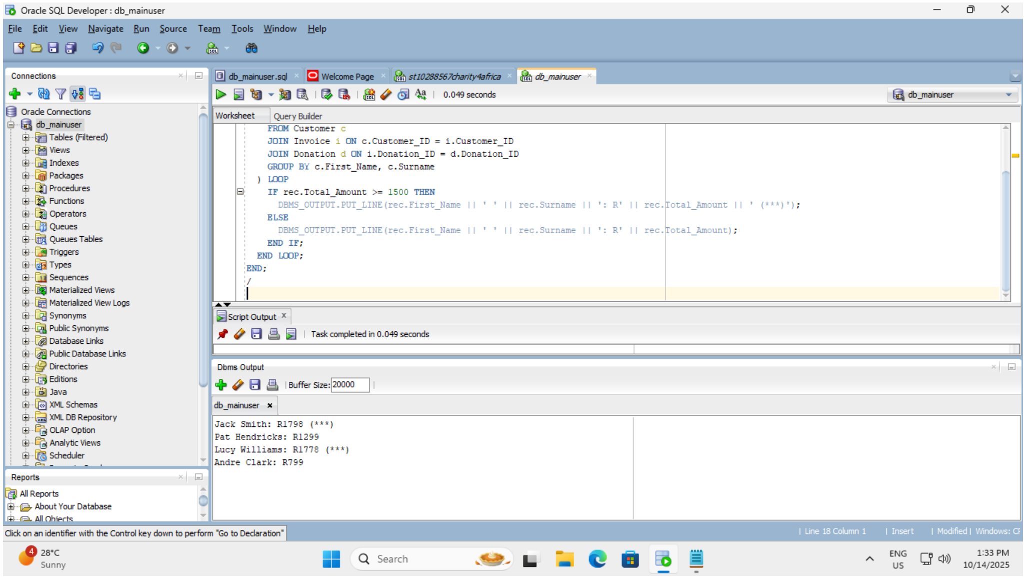
QUESTION 5

Delivery Days Difference code:

```
SET SERVEROUTPUT ON;

BEGIN
  FOR rec IN (
    SELECT c.First_Name, c.Surname, e.First_Name AS EmpFirst,
           e.Surname AS EmpLast, dly.Dispatch_Date, dly.Delivery_Date,
           dn.Donation,
           (dly.Delivery_Date - dly.Dispatch_Date) AS Days_to_Deliver
    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 dly ON i.Delivery_ID = dly.Delivery_ID
    WHERE c.Customer_ID = 11011
  ) LOOP
```

```
    DBMS_OUTPUT.PUT_LINE('CUSTOMER: ' || rec.First_Name || ' ' || rec.Surname);
    DBMS_OUTPUT.PUT_LINE('EMPLOYEE: ' || rec.EmpFirst || ' ' || rec.EmpLast);
    DBMS_OUTPUT.PUT_LINE('DONATION: ' || rec.Donation);
    DBMS_OUTPUT.PUT_LINE('DISPATCH DATE: ' || rec.Dispatch_Date);
    DBMS_OUTPUT.PUT_LINE('DELIVERY DATE: ' || rec.Delivery_Date);
    DBMS_OUTPUT.PUT_LINE('DAYS TO DELIVERY: ' || rec.Days_to_Deliver);
    DBMS_OUTPUT.PUT_LINE('------------------------------');
  END LOOP;
END;/
```

QUESTION 6



```
    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
  ) LOOP
    IF rec.Total_Amount >= 1500 THEN
      DBMS_OUTPUT.PUT_LINE(rec.First_Name || ' ' || rec.Surname || ': R' || rec.Total_Amount || ' (***)');
    ELSE
      DBMS_OUTPUT.PUT_LINE(rec.First_Name || ' ' || rec.Surname || ': R' || rec.Total_Amount);
    END IF;
  END LOOP;
END;
/
```

Task completed in 0.049 seconds

Buffer Size: 20000

```
Jack Smith: R1798 (***)
Pat Hendricks: R1299
Lucy Williams: R1778 (***)
Andre Clark: R799
```

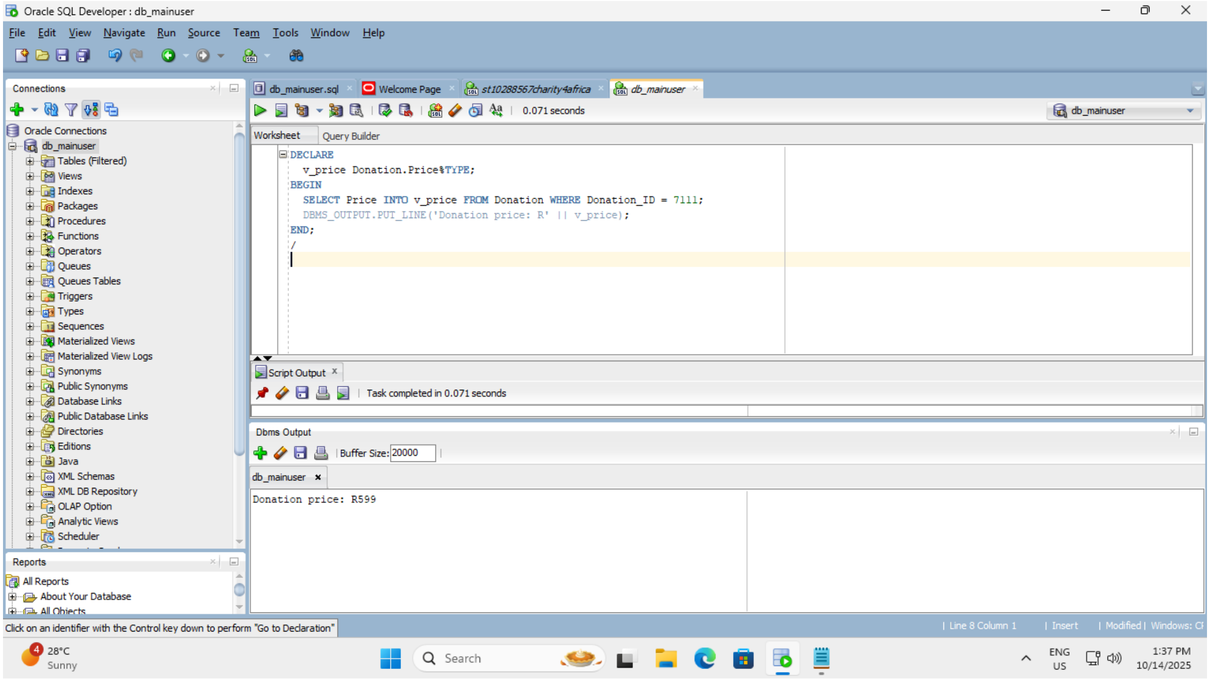
(Oracle Corporation, 2023).

QUESTION 6

Total Spend and Star Rating code:

```
SET SERVEROUTPUT ON;

BEGIN

  FOR rec IN (

    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

  ) LOOP

    IF rec.Total_Amount >= 1500 THEN

      DBMS_OUTPUT.PUT_LINE(rec.First_Name || ' ' || rec.Surname || ': R' ||
rec.Total_Amount || ' (***)');

    ELSE

      DBMS_OUTPUT.PUT_LINE(rec.First_Name || ' ' || rec.Surname || ': R' ||
rec.Total_Amount);

    END IF;

  END LOOP;

END;/
```

QUESTION 7



```
DECLARE
  v_price Donation.Price%TYPE;
BEGIN
  SELECT Price INTO v_price FROM Donation WHERE Donation_ID = 7111;
  DBMS_OUTPUT.PUT_LINE('Donation price: R' || v_price);
END;
/
```

Task completed in 0.071 seconds

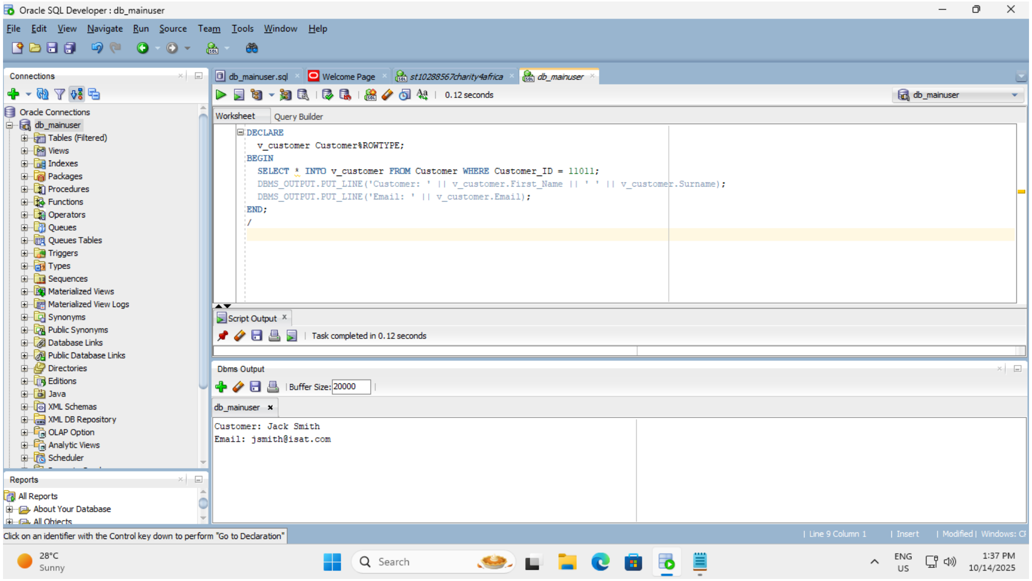
```
Donation price: R599
```

(Oracle Corporation, 2023).

7.1 %TYPE code:

```
DECLARE
  v_price Donation.Price%TYPE;
BEGIN
  SELECT Price INTO v_price FROM Donation WHERE Donation_ID = 7111;
  DBMS_OUTPUT.PUT_LINE('Donation price: R' || v_price);
END;
/
```

7.2.



```
DECLARE
  v_customer Customer%ROWTYPE;
BEGIN
  SELECT * INTO v_customer FROM Customer WHERE Customer_ID = 11011;
  DBMS_OUTPUT.PUT_LINE('Customer: ' || v_customer.First_Name || ' ' || v_customer.Surname);
  DBMS_OUTPUT.PUT_LINE('Email: ' || v_customer.Email);
END;
/
```

Task completed in 0.12 seconds

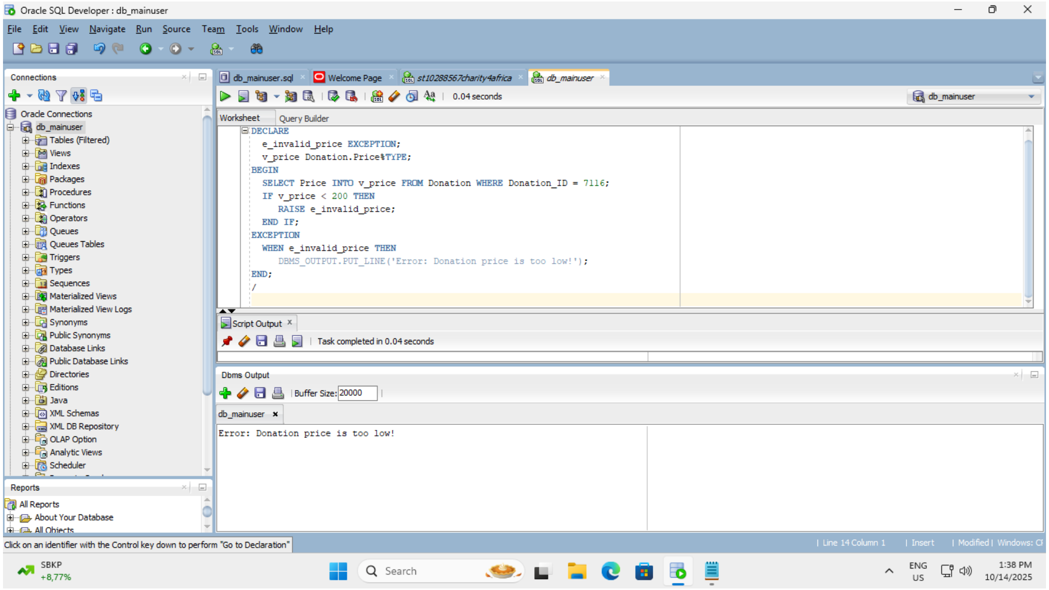
```
Customer: Jack Smith
Email: jsmith@isat.com
```

(Oracle Corporation, 2023).

7.2 %ROWTYPE code:

```
DECLARE
  v_customer Customer%ROWTYPE;
BEGIN
  SELECT * INTO v_customer FROM Customer WHERE Customer_ID = 11011;
  DBMS_OUTPUT.PUT_LINE('Customer: ' || v_customer.First_Name || ' ' ||
v_customer.Surname);
  DBMS_OUTPUT.PUT_LINE('Email: ' || v_customer.Email);
END;
/
```

7.3.



```
DECLARE
  e_invalid_price EXCEPTION;
  v_price Donation.Price%TYPE;
BEGIN
  SELECT Price INTO v_price FROM Donation WHERE Donation_ID = 7116;
  IF v_price < 200 THEN
     RAISE e_invalid_price;
  END IF;
EXCEPTION
  WHEN e_invalid_price THEN
     DBMS_OUTPUT.PUT_LINE('Error: Donation price is too low!');
END;
/
```

```
Task completed in 0.04 seconds
```

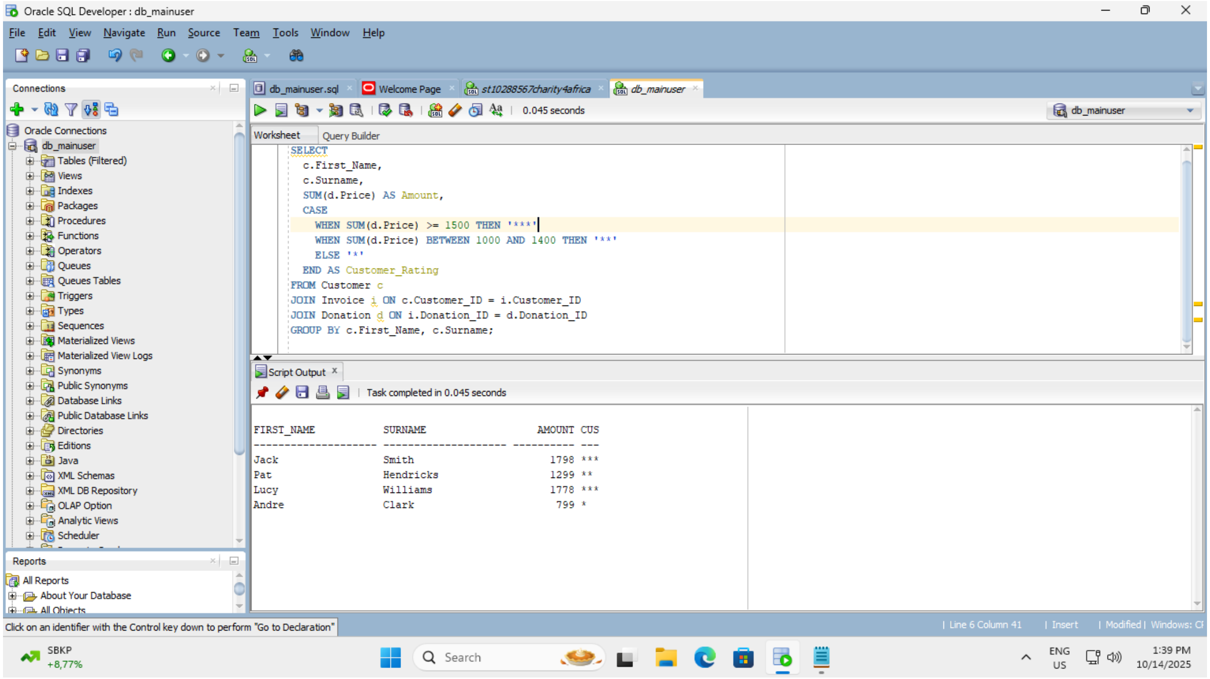
```
Error: Donation price is too low!
```

(Oracle Corporation, 2023).

7.3 User-defined exception code:

```
DECLARE
  e_invalid_price EXCEPTION;
  v_price Donation.Price%TYPE;
BEGIN
  SELECT Price INTO v_price FROM Donation WHERE Donation_ID = 7116;
  IF v_price < 200 THEN
    RAISE e_invalid_price;
  END IF;
EXCEPTION
  WHEN e_invalid_price THEN
    DBMS_OUTPUT.PUT_LINE('Error: Donation price is too low!');
END;
/
```

QUESTION 8



```
SELECT
  c.First_Name,
  c.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;
```

```
FIRST_NAME           SURNAME                  AMOUNT CUS
-------------------- -------------------- ---------- ---
Jack                 Smith                      1798 ***
Pat                  Hendricks                  1299 **
Lucy                 Williams                   1778 ***
Andre                Clark                       799 *
```

(Oracle Corporation, 2023).

QUESTION 8 — SQL with CASE Statement code:

```
SELECT
  c.First_Name,
  c.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;
```

(Oracle Corporation, 2023).

Reference List

Oracle Corporation. (2023) Oracle Database Net Services Administrator’s Guide, 21c. Redwood City, CA: Oracle. Available at: https://docs.oracle.com/en/database/oracle/oracle-database/21/netag/index.html

(Accessed: 10 October 2025).