Islamic University of Technology

RDBMS

<u>CSE 4508</u>

<u>Lab Report 9</u>

Name : Amina

Student ID : 200041155

Section : 1

Lab Group : 1B (Shifted from 1A)

Date of Performance : 15/11/23

Date of Submission : 15/11/23

Task 1

Differences between a PL/SQLprocedure and a function:

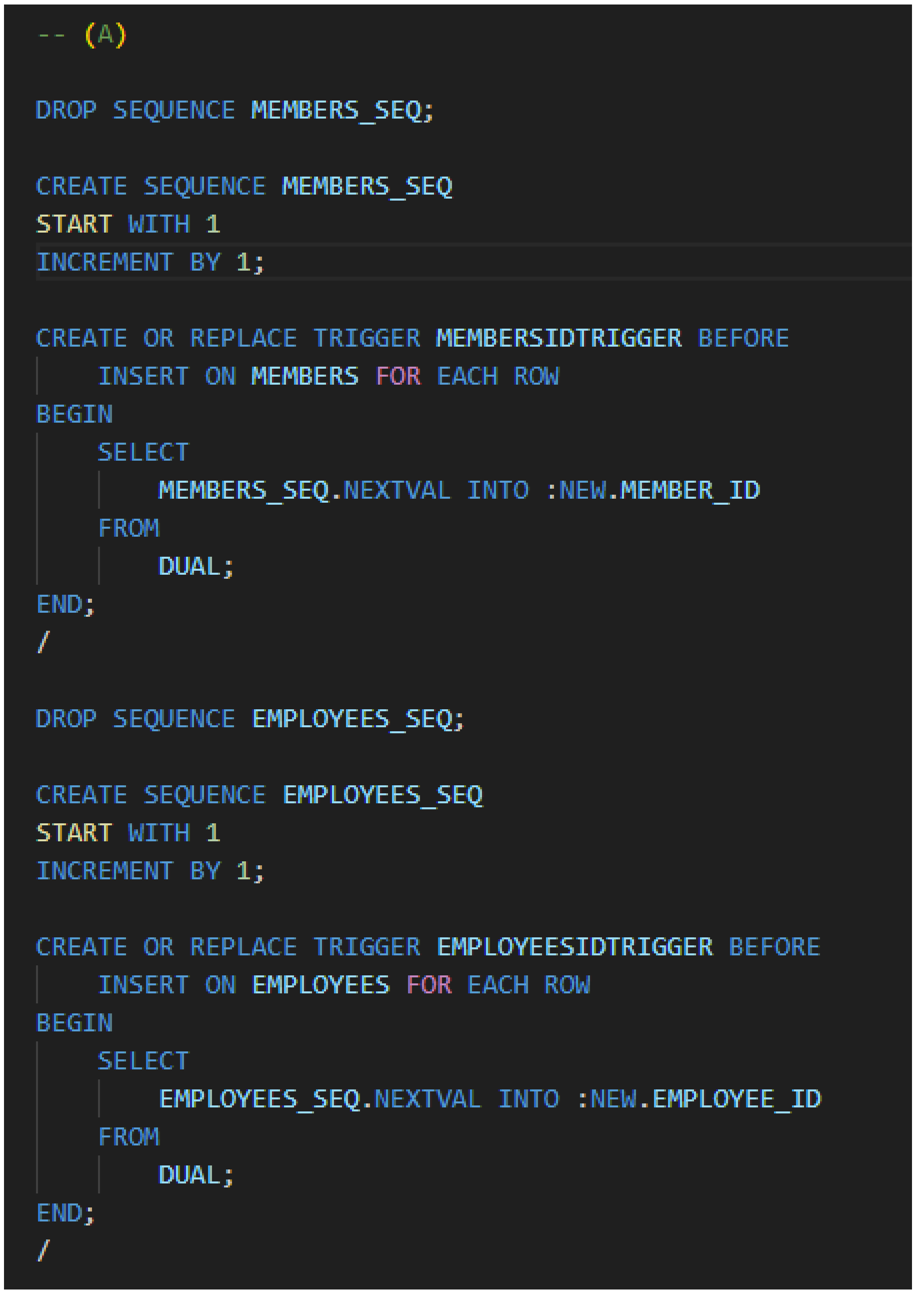
A procedure or function acts like a concise program, consisting of optional declaration, execution, and exception-handling parts.

- A procedure functions as a subprogram with a specific task. It requires specifications like its name, parameters, local variables, and a BEGIN-END block enclosing its code and exception-handling.

- In comparison, a function, structured like a procedure, is designed to calculate and provide a return value. The key difference is that functions give an output.

Task 2 - A

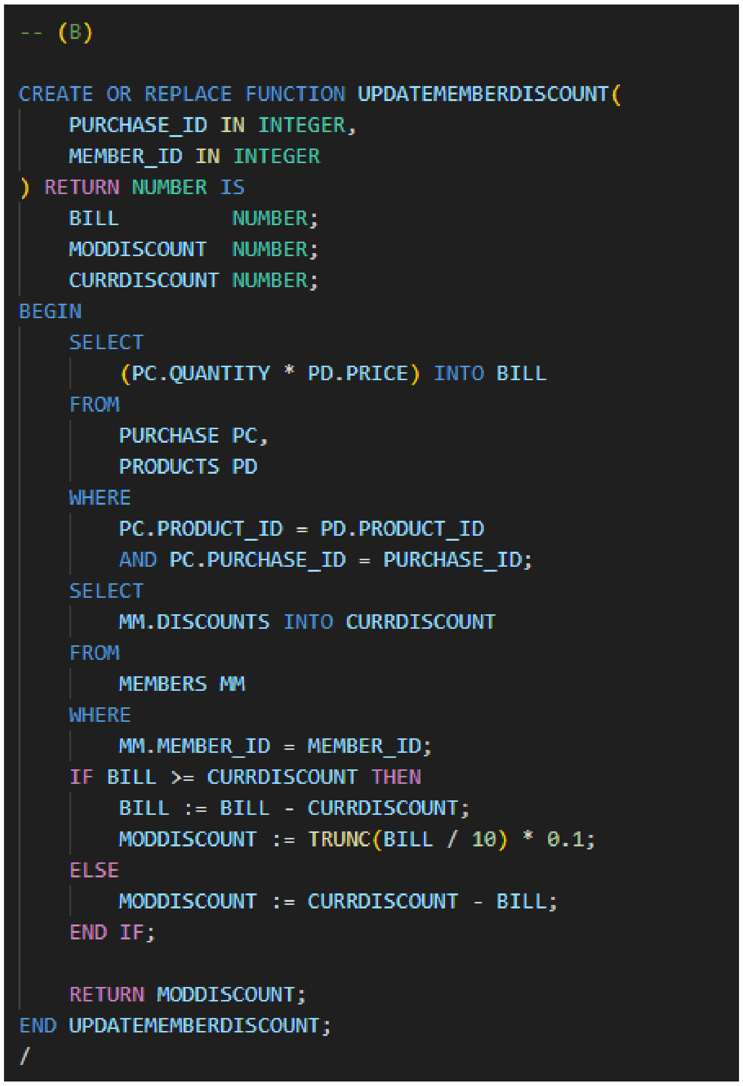
Here, I created two sequences, one for generating primary keys for Members table another for Employees table.



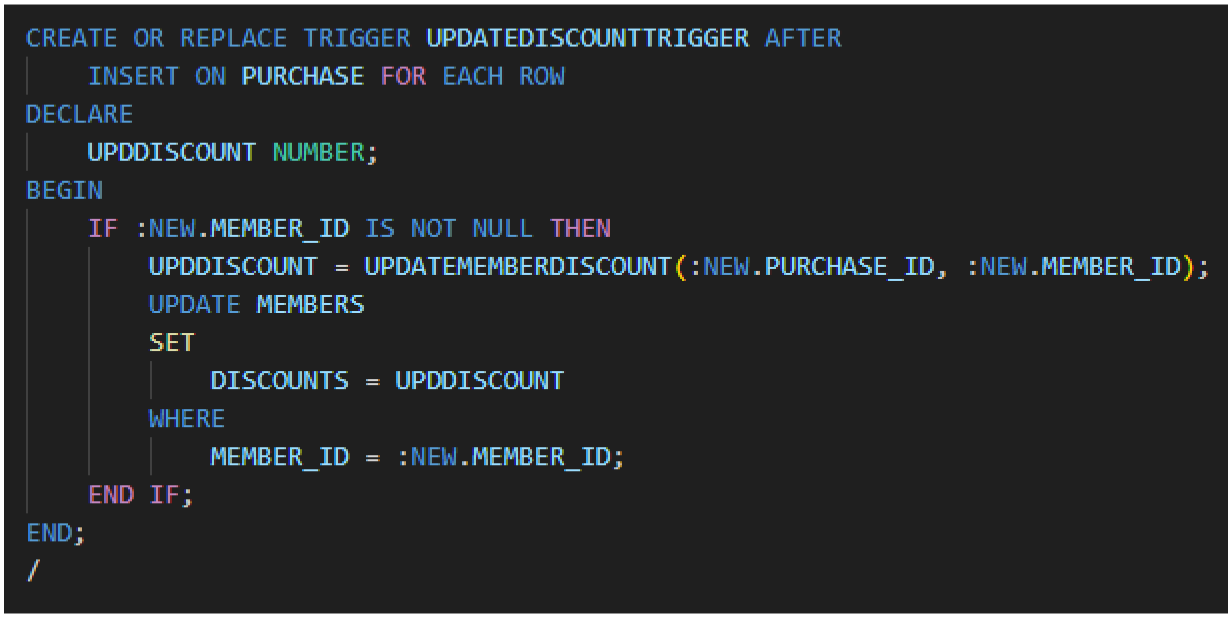
```
-- (A)

DROP SEQUENCE MEMBERS_SEQ;

CREATE SEQUENCE MEMBERS_SEQ
START WITH 1
INCREMENT BY 1;

CREATE OR REPLACE TRIGGER MEMBERSIDTRIGGER BEFORE
    INSERT ON MEMBERS FOR EACH ROW
BEGIN
    SELECT
        MEMBERS_SEQ.NEXTVAL INTO :NEW.MEMBER_ID
    FROM
        DUAL;
END;
/

DROP SEQUENCE EMPLOYEES_SEQ;

CREATE SEQUENCE EMPLOYEES_SEQ
START WITH 1
INCREMENT BY 1;

CREATE OR REPLACE TRIGGER EMPLOYEESIDTRIGGER BEFORE
    INSERT ON EMPLOYEES FOR EACH ROW
BEGIN
    SELECT
        EMPLOYEES_SEQ.NEXTVAL INTO :NEW.EMPLOYEE_ID
    FROM
        DUAL;
END;
/
```

Task 2 - B

Here, I made a trigger updateDiscountTrigger which takes purchase id and member id and calls a function updateMemberDiscount. The function calculates the bill and current discount. Then I calculated new discount modify the column.



```
-- (B)

CREATE OR REPLACE FUNCTION UPDATEMEMBERDISCOUNT(
    PURCHASE_ID IN INTEGER,
    MEMBER_ID IN INTEGER
) RETURN NUMBER IS
    BILL         NUMBER;
    MODDISCOUNT  NUMBER;
    CURRDISCOUNT NUMBER;
BEGIN
    SELECT
        (PC.QUANTITY * PD.PRICE) INTO BILL
    FROM
        PURCHASE PC,
        PRODUCTS PD
    WHERE
        PC.PRODUCT_ID = PD.PRODUCT_ID
        AND PC.PURCHASE_ID = PURCHASE_ID;
    SELECT
        MM.DISCOUNTS INTO CURRDISCOUNT
    FROM
        MEMBERS MM
    WHERE
        MM.MEMBER_ID = MEMBER_ID;
    IF BILL >= CURRDISCOUNT THEN
        BILL := BILL - CURRDISCOUNT;
        MODDISCOUNT := TRUNC(BILL / 10) * 0.1;
    ELSE
        MODDISCOUNT := CURRDISCOUNT - BILL;
    END IF;

    RETURN MODDISCOUNT;
END UPDATEMEMBERDISCOUNT;
/
```

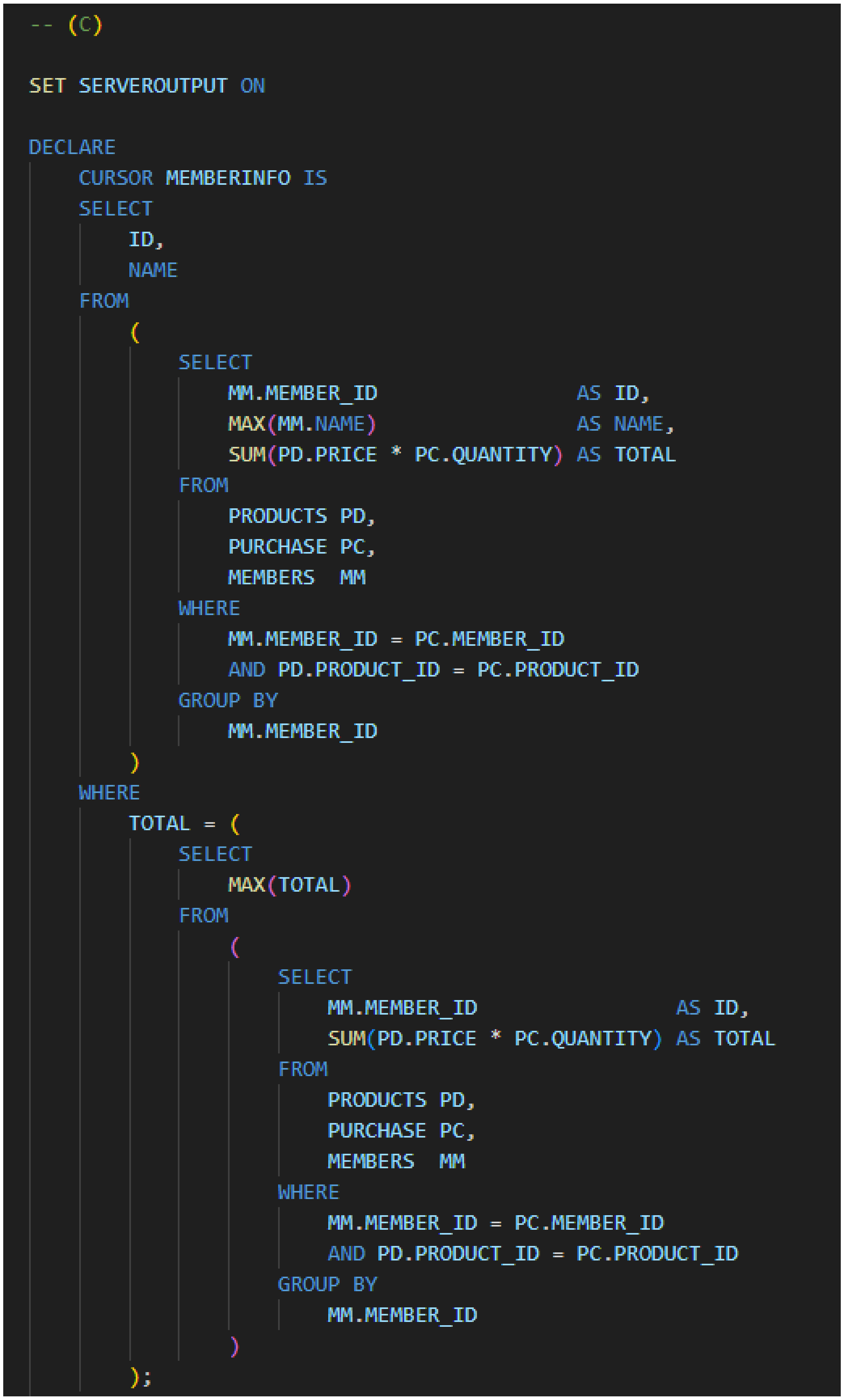


```sql
CREATE OR REPLACE TRIGGER UPDATEDISCOUNTTRIGGER AFTER
    INSERT ON PURCHASE FOR EACH ROW
DECLARE
    UPDDISCOUNT NUMBER;
BEGIN
    IF :NEW.MEMBER_ID IS NOT NULL THEN
        UPDDISCOUNT = UPDATEMEMBERDISCOUNT(:NEW.PURCHASE_ID, :NEW.MEMBER_ID);
        UPDATE MEMBERS
        SET
            DISCOUNTS = UPDDISCOUNT
        WHERE
            MEMBER_ID = :NEW.MEMBER_ID;
    END IF;
END;
/
```

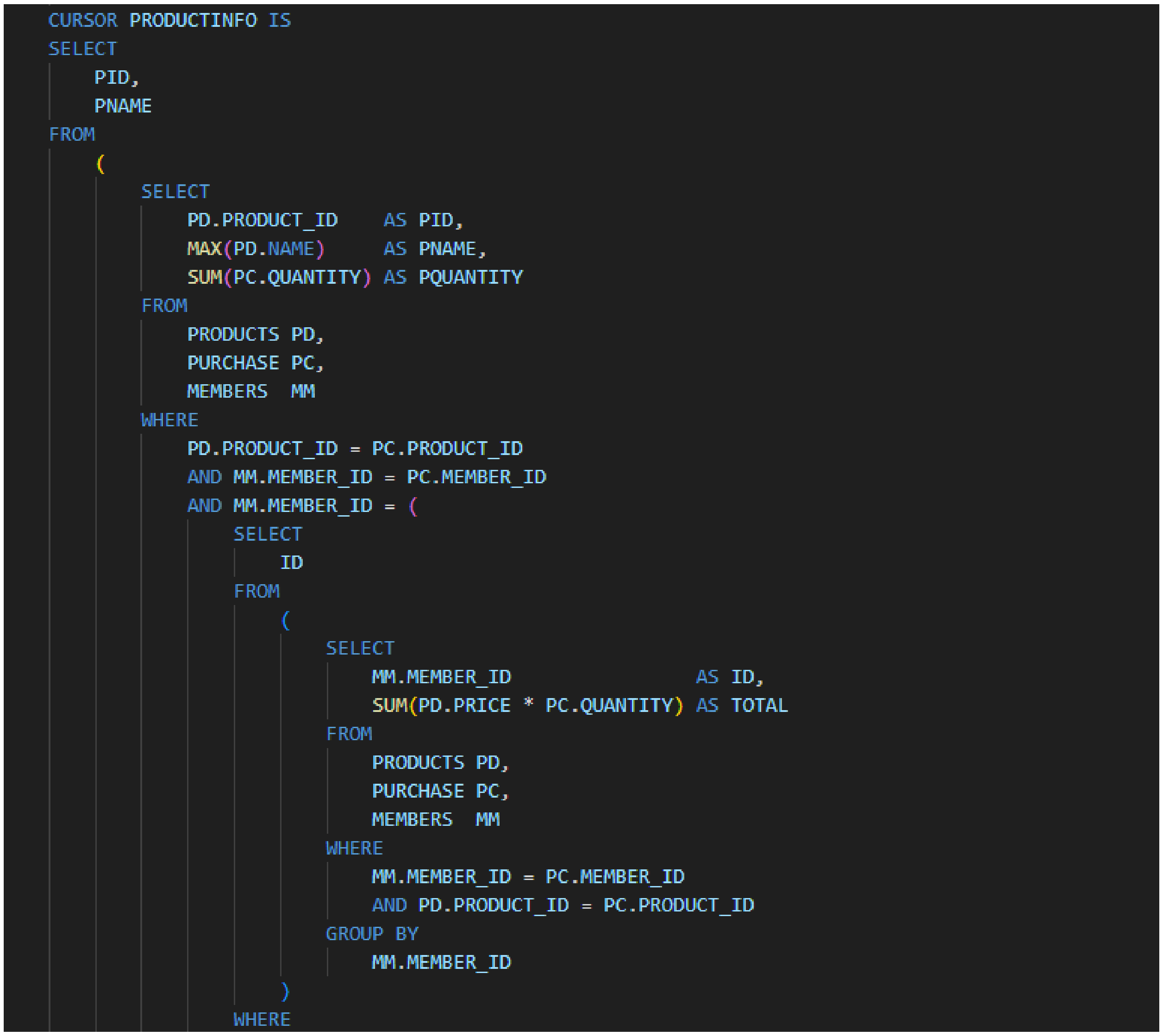
Task 2 - C

Initially, a joint table combining member, product, and purchase data was created and grouped by member ID. Each member ID had multiple rows with different products. The total cost for each member was calculated by multiplying the product price and quantity, summing them for each member.

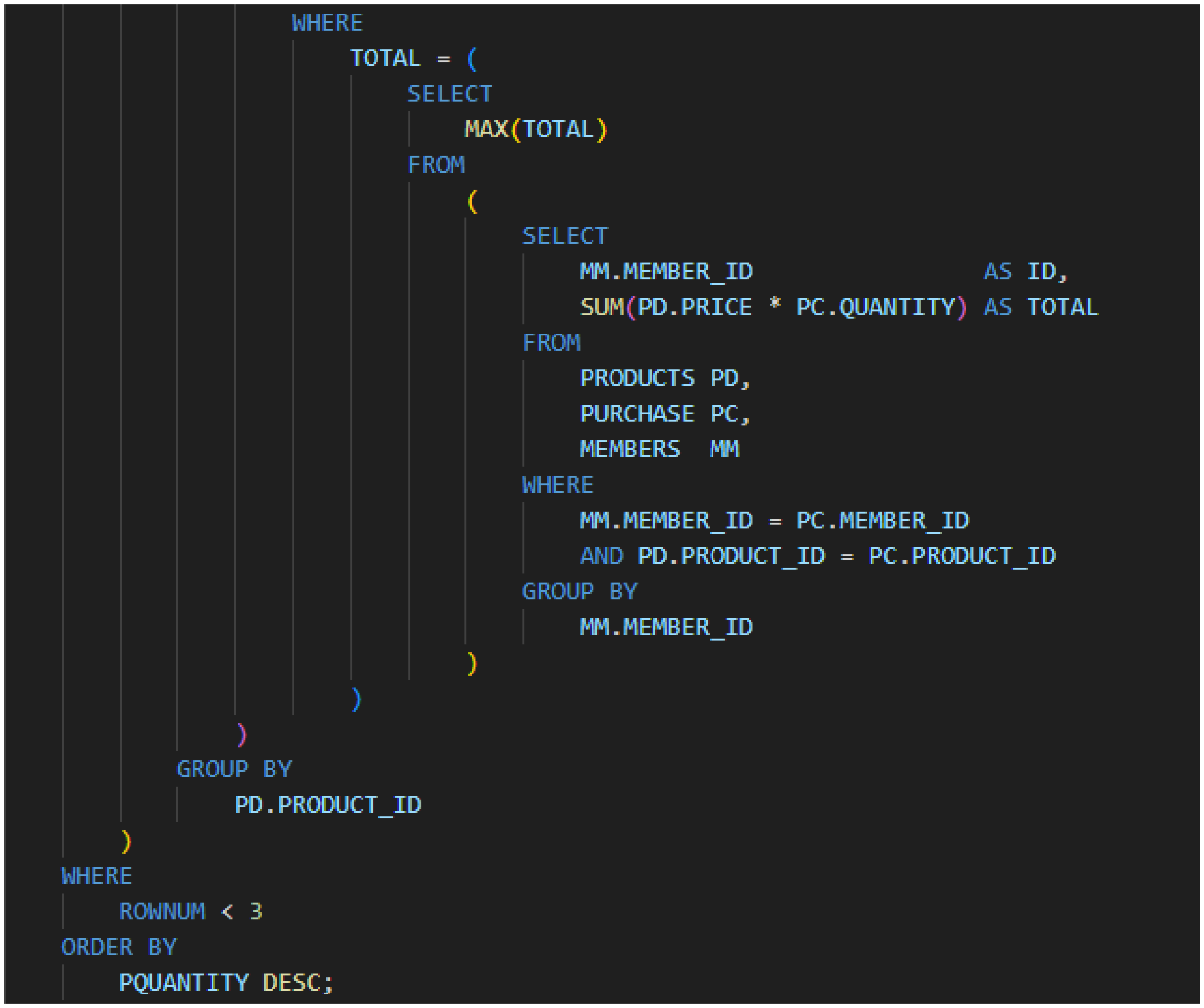
Next, the member who spent the maximum was identified, and their ID and name were extracted. From the records of this top-spending member, the data was grouped by product ID. The total quantity of each particular product was determined. Finally, the top two products with the highest quantities were identified.



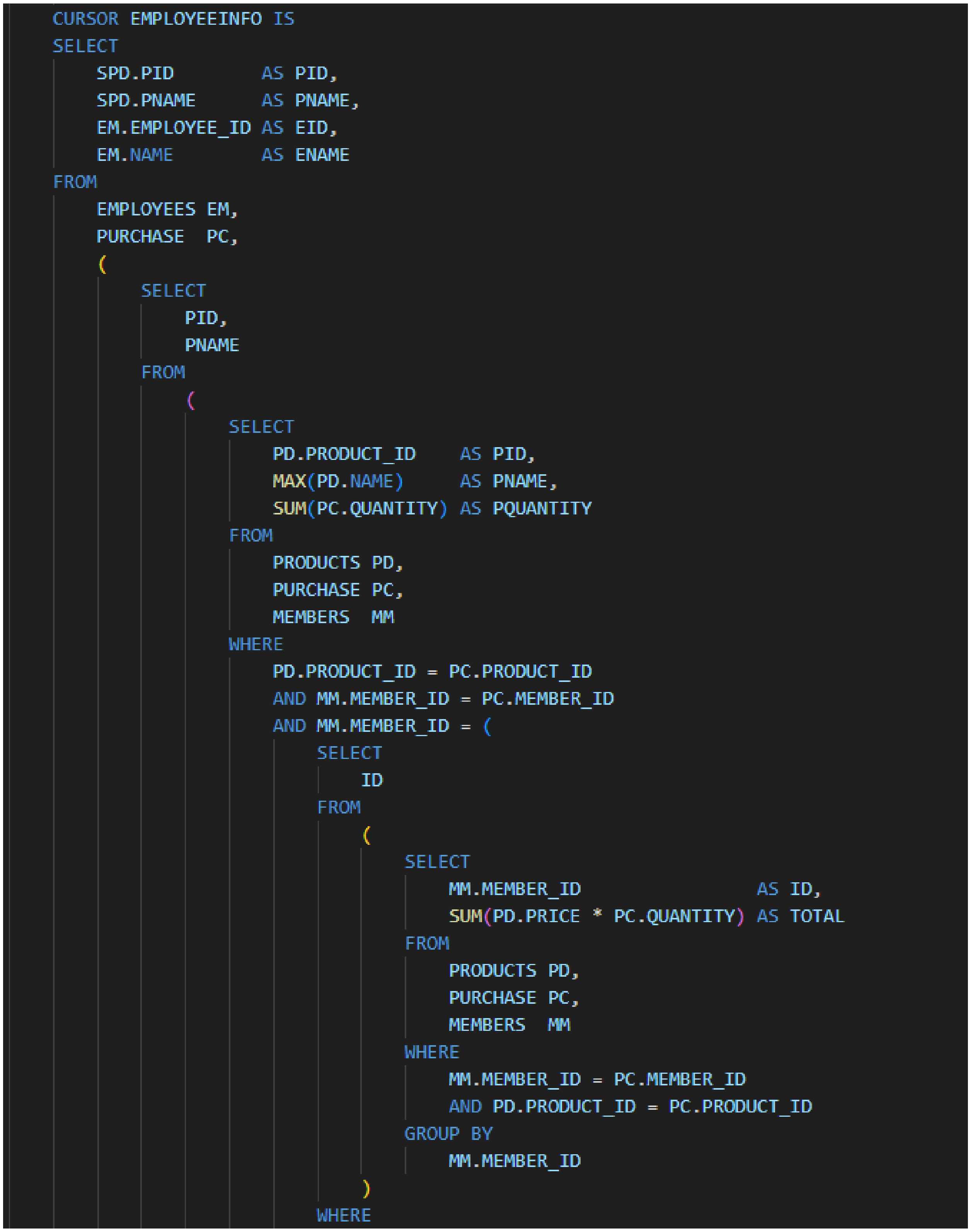
```
-- (C)

SET SERVEROUTPUT ON

DECLARE
    CURSOR MEMBERINFO IS
    SELECT
        ID,
        NAME
    FROM
        (
            SELECT
                MM.MEMBER_ID                AS ID,
                MAX(MM.NAME)                AS NAME,
                SUM(PD.PRICE * PC.QUANTITY) AS TOTAL
            FROM
                PRODUCTS PD,
                PURCHASE PC,
                MEMBERS  MM
            WHERE
                MM.MEMBER_ID = PC.MEMBER_ID
                AND PD.PRODUCT_ID = PC.PRODUCT_ID
            GROUP BY
                MM.MEMBER_ID
        )
    WHERE
        TOTAL = (
            SELECT
                MAX(TOTAL)
            FROM
                (
                    SELECT
                        MM.MEMBER_ID                AS ID,
                        SUM(PD.PRICE * PC.QUANTITY) AS TOTAL
                    FROM
                        PRODUCTS PD,
                        PURCHASE PC,
                        MEMBERS  MM
                    WHERE
                        MM.MEMBER_ID = PC.MEMBER_ID
                        AND PD.PRODUCT_ID = PC.PRODUCT_ID
                    GROUP BY
                        MM.MEMBER_ID
                )
        );
```



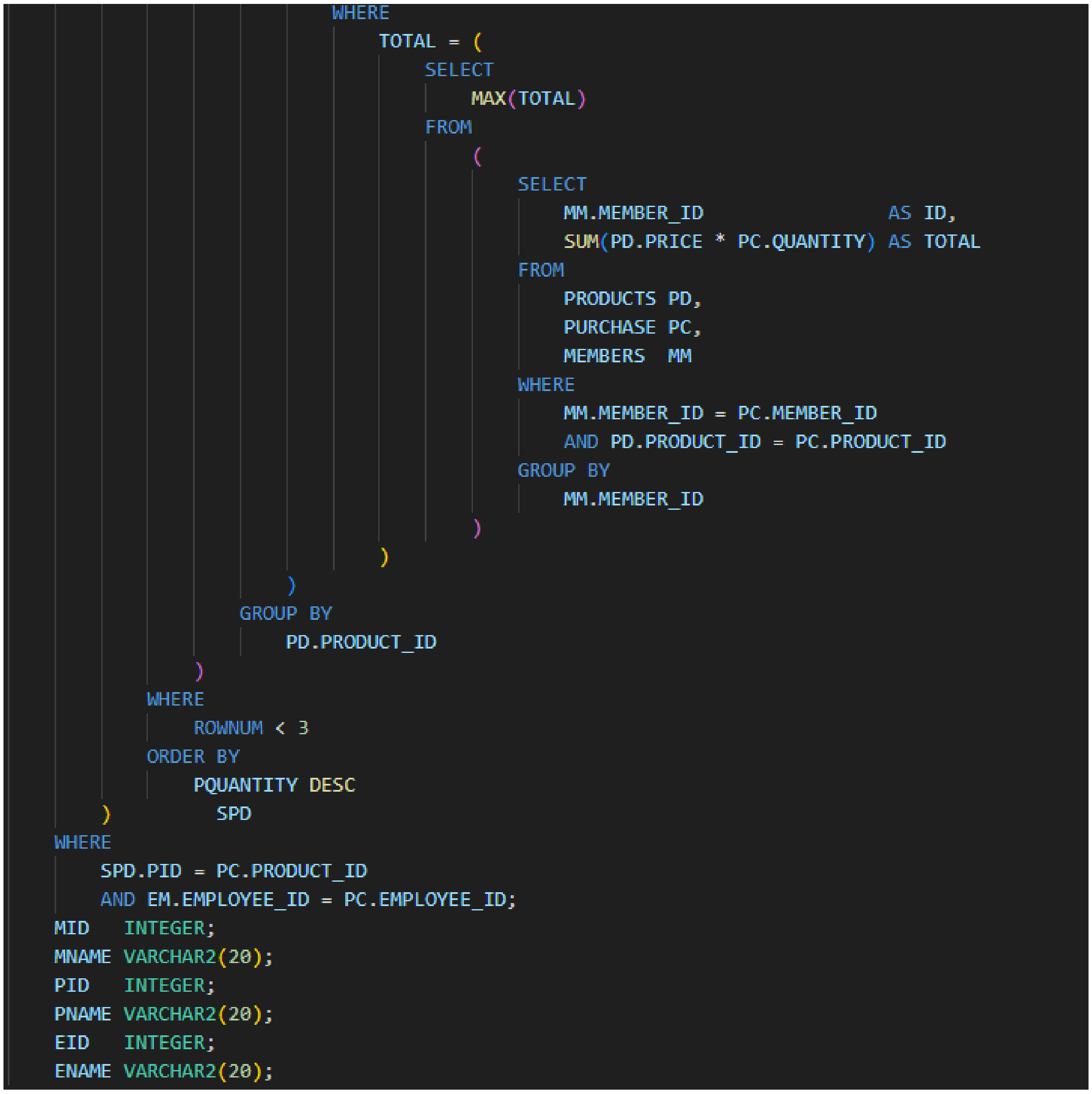
```
CURSOR PRODUCTINFO IS
SELECT
    PID,
    PNAME
FROM
    (
        SELECT
            PD.PRODUCT_ID    AS PID,
            MAX(PD.NAME)     AS PNAME,
            SUM(PC.QUANTITY) AS PQUANTITY
        FROM
            PRODUCTS PD,
            PURCHASE PC,
            MEMBERS  MM
        WHERE
            PD.PRODUCT_ID = PC.PRODUCT_ID
            AND MM.MEMBER_ID = PC.MEMBER_ID
            AND MM.MEMBER_ID = (
                SELECT
                    ID
                FROM
                    (
                        SELECT
                            MM.MEMBER_ID                AS ID,
                            SUM(PD.PRICE * PC.QUANTITY) AS TOTAL
                        FROM
                            PRODUCTS PD,
                            PURCHASE PC,
                            MEMBERS  MM
                        WHERE
                            MM.MEMBER_ID = PC.MEMBER_ID
                            AND PD.PRODUCT_ID = PC.PRODUCT_ID
                        GROUP BY
                            MM.MEMBER_ID
                    )
                WHERE
```



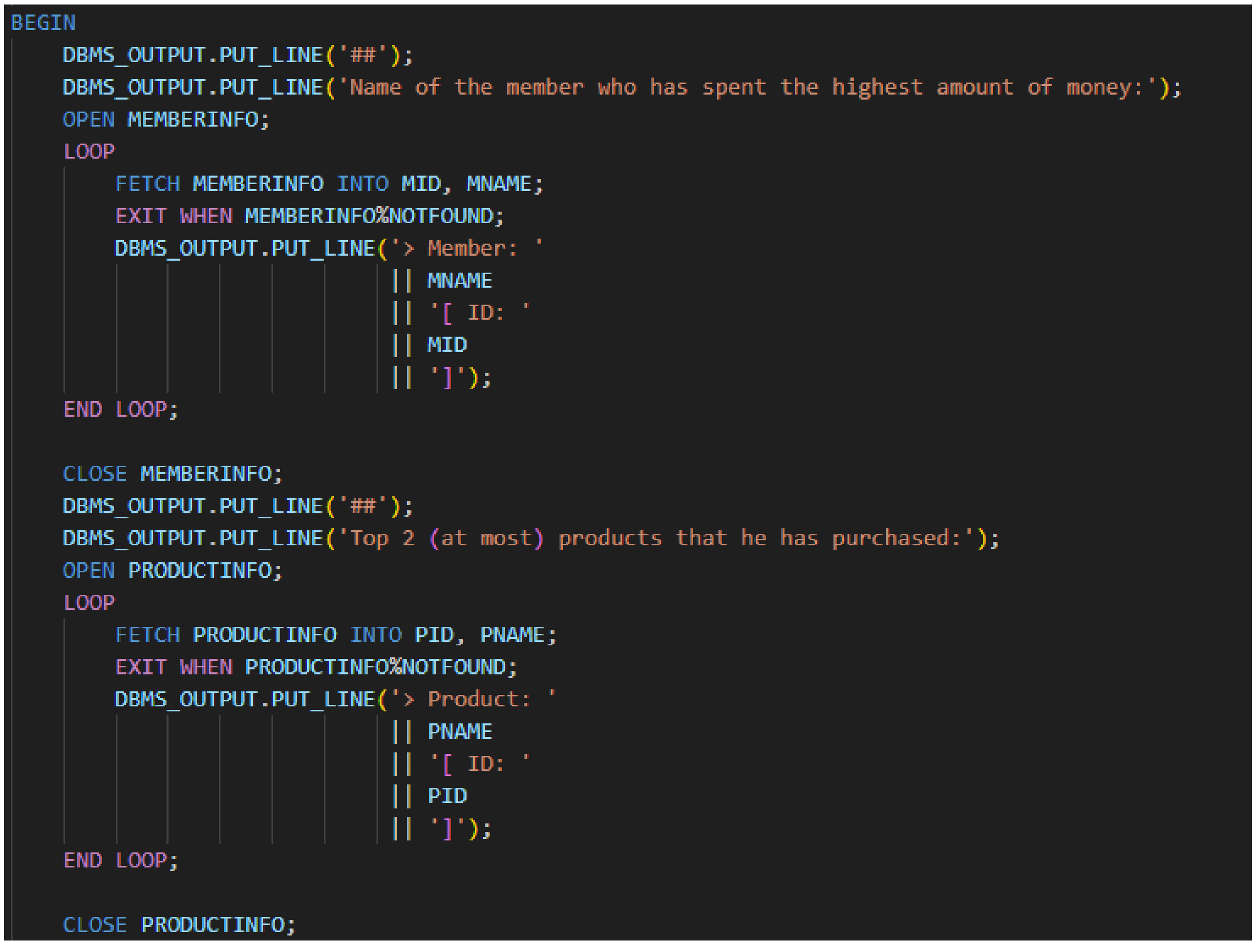
```sql
                WHERE
                    TOTAL = (
                        SELECT
                            MAX(TOTAL)
                        FROM
                            (
                                SELECT
                                    MM.MEMBER_ID                AS ID,
                                    SUM(PD.PRICE * PC.QUANTITY) AS TOTAL
                                FROM
                                    PRODUCTS PD,
                                    PURCHASE PC,
                                    MEMBERS  MM
                                WHERE
                                    MM.MEMBER_ID = PC.MEMBER_ID
                                    AND PD.PRODUCT_ID = PC.PRODUCT_ID
                                GROUP BY
                                    MM.MEMBER_ID
                            )
                    )
            )
        GROUP BY
            PD.PRODUCT_ID
    )
WHERE
    ROWNUM < 3
ORDER BY
    PQUANTITY DESC;
```



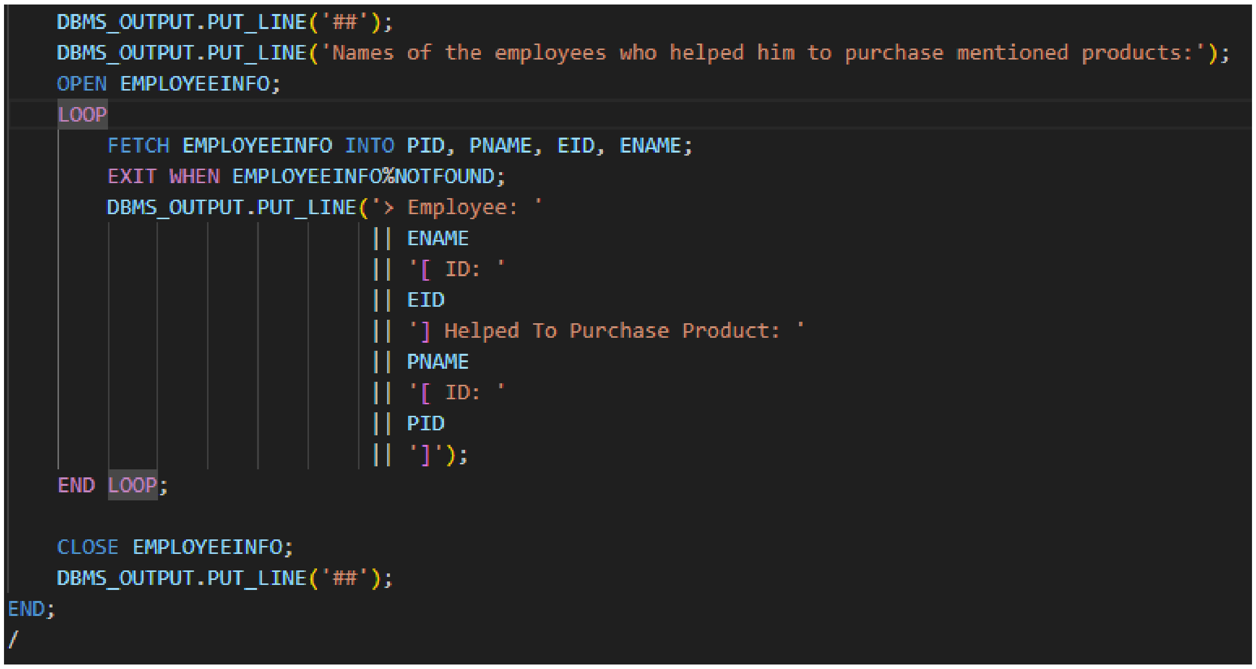
```
CURSOR EMPLOYEEINFO IS
SELECT
    SPD.PID         AS PID,
    SPD.PNAME       AS PNAME,
    EM.EMPLOYEE_ID AS EID,
    EM.NAME         AS ENAME
FROM
    EMPLOYEES EM,
    PURCHASE  PC,
    (
        SELECT
            PID,
            PNAME
        FROM
            (
                SELECT
                    PD.PRODUCT_ID    AS PID,
                    MAX(PD.NAME)     AS PNAME,
                    SUM(PC.QUANTITY) AS PQUANTITY
                FROM
                    PRODUCTS PD,
                    PURCHASE PC,
                    MEMBERS  MM
                WHERE
                    PD.PRODUCT_ID = PC.PRODUCT_ID
                    AND MM.MEMBER_ID = PC.MEMBER_ID
                    AND MM.MEMBER_ID = (
                        SELECT
                            ID
                        FROM
                            (
                                SELECT
                                    MM.MEMBER_ID                AS ID,
                                    SUM(PD.PRICE * PC.QUANTITY) AS TOTAL
                                FROM
                                    PRODUCTS PD,
                                    PURCHASE PC,
                                    MEMBERS  MM
                                WHERE
                                    MM.MEMBER_ID = PC.MEMBER_ID
                                    AND PD.PRODUCT_ID = PC.PRODUCT_ID
                                GROUP BY
                                    MM.MEMBER_ID
                            )
                        WHERE
```



```
                        WHERE
                            TOTAL = (
                                SELECT
                                    MAX(TOTAL)
                                FROM
                                    (
                                        SELECT
                                            MM.MEMBER_ID                AS ID,
                                            SUM(PD.PRICE * PC.QUANTITY) AS TOTAL
                                        FROM
                                            PRODUCTS PD,
                                            PURCHASE PC,
                                            MEMBERS  MM
                                        WHERE
                                            MM.MEMBER_ID = PC.MEMBER_ID
                                            AND PD.PRODUCT_ID = PC.PRODUCT_ID
                                        GROUP BY
                                            MM.MEMBER_ID
                                    )
                            )
                    )
                GROUP BY
                    PD.PRODUCT_ID
            )
        WHERE
            ROWNUM < 3
        ORDER BY
            PQUANTITY DESC
    )         SPD
WHERE
    SPD.PID = PC.PRODUCT_ID
    AND EM.EMPLOYEE_ID = PC.EMPLOYEE_ID;
MID   INTEGER;
MNAME VARCHAR2(20);
PID   INTEGER;
PNAME VARCHAR2(20);
EID   INTEGER;
ENAME VARCHAR2(20);
```



```
BEGIN
    DBMS_OUTPUT.PUT_LINE('##');
    DBMS_OUTPUT.PUT_LINE('Name of the member who has spent the highest amount of money:');
    OPEN MEMBERINFO;
    LOOP
        FETCH MEMBERINFO INTO MID, MNAME;
        EXIT WHEN MEMBERINFO%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('> Member: '
                             || MNAME
                             || '[ ID: '
                             || MID
                             || ']');
    END LOOP;

    CLOSE MEMBERINFO;
    DBMS_OUTPUT.PUT_LINE('##');
    DBMS_OUTPUT.PUT_LINE('Top 2 (at most) products that he has purchased:');
    OPEN PRODUCTINFO;
    LOOP
        FETCH PRODUCTINFO INTO PID, PNAME;
        EXIT WHEN PRODUCTINFO%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('> Product: '
                             || PNAME
                             || '[ ID: '
                             || PID
                             || ']');
    END LOOP;

    CLOSE PRODUCTINFO;
```



```
    DBMS_OUTPUT.PUT_LINE('##');
    DBMS_OUTPUT.PUT_LINE('Names of the employees who helped him to purchase mentioned products:');
    OPEN EMPLOYEEINFO;
    LOOP
        FETCH EMPLOYEEINFO INTO PID, PNAME, EID, ENAME;
        EXIT WHEN EMPLOYEEINFO%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('> Employee: '
                             || ENAME
                             || '[ ID: '
                             || EID
                             || '] Helped To Purchase Product: '
                             || PNAME
                             || '[ ID: '
                             || PID
                             || ']');
    END LOOP;

    CLOSE EMPLOYEEINFO;
    DBMS_OUTPUT.PUT_LINE('##');
END;
/
```