NAME: SHUBHAM TRIBEDI
ROLL NO. 18111000002037
PAPER: DBMS ASSIGNMENT NO. 9

1. Write a PL/SQL program which can update the cost of BOOKS_COPY table with 10 more cost where cost is less than 500 and show how many rows are affected (Use Implicit Cursor SQL%ROWCOUNT).

```
DECLARE
  rows_effected
NUMBER(4); BEGIN
  UPDATE BOOKS_COPY SET Cost=Cost+10 WHERE Cost<500 ;
  IF sql%FOUND THEN
    rows_effected:=sql%rowcount;

  dbms_output.Put_line('Total number of rows updated ' ||
  rows_effected); END IF;

END;
/
```

OUTPUT

Total number of rows updated 3

2. Write a PL/SQL program which can increment the value of MAX_BOOKS_ALLOWED of MEMBER_COPY table with 2 where MEMBER_ID = 5, and show a message if update is possible. (Use Implicit Cursor SQL%FOUND).

```
BEGIN
  UPDATE MEMBER_COPY SET Max_Books_Allowed=Max_Books_Allowed+2 WHERE
Member_Id=5;
  IF sql%FOUND THEN
     dbms_output.Put_line('Matching row found. Update is
  possible'); END IF;

END;
/
```

OUTPUT

Matching row found. Update is possible

3. Write a PL/SQL Program using Explicit Cursor and show the Member_ID, Member Name for every attribute of Member.

```
DECLARE
  CURSOR member_cursor IS SELECT Member_Id, Member_Name FROM MEMBER_COPY;
  Member_Id MEMBER_COPY.Member_Id%Type;
  Member_Name MEMBER_COPY.Member_Name%Type;
```

```
BEGIN
  OPEN member_cursor;


  dbms_output.Put_line('MEMBER ID MEMBER NAME' );
  LOOP
  FETCH member_cursor INTO Member_Id,
  Member_Name; EXIT WHEN
  member_cursor%NOTFOUND;
    dbms_output.Put_line(Member_Id || '       ' || Member_Name);
  END LOOP;

  CLOSE member_cursor;
END;
/
```

OUTPUT

MEMBER ID	MEMBER NAME
2	Abhirup Sarkar
3	Ritesh Bhuniya
4	Paresh Sen
6	Suparna Biswas
8	Arpita Roy
5	Sohini Haldar
7	Suranjana Basu
1	Sayantan Sinha

4. Write a PL/SQL program using Explicit Cursor which deducts the value of Max_Books_Allowed from MEMBER_COPY table. Deduct value means the value that how many times this member accesses the books. After deduction if value of Max_Books_Allowed is less than 0 the do not update it and show an error message.

```
DECLARE
  CURSOR member_cursor IS SELECT Member_Id,Max_Books_Allowed FROM
  MEMBER_COPY; Mem_Id MEMBER_COPY.Member_Id%TYPE;
  MaxBooks MEMBER_COPY.Max_Books_Allowed%TYPE;
  value NUMBER(4);
BEGIN
  OPEN member_cursor;

  LOOP
  FETCH member_cursor INTO
  Mem_Id,MaxBooks; EXIT WHEN
  member_cursor%NOTFOUND;
    SELECt COUNT(*) INTO value FROM ISSUE_COPY WHERE Member_Id=Mem_Id;
      IF(value<=MaxBooks)THEN
        UPDATE MEMBER_COPY SET Max_Books_Allowed=Max_Books_Allowed-value
WHERE Member_Id=Mem_Id;
        dbms_output.Put_line('Number of max books allowed is updated for Member with
ID ' || Mem_Id);
      ELSE
```

```
            dbms_output.Put_line('Value taken from ISSUE_COPY table is greater than the
current number of maximum books allowed. Deduction cannot be completed.');
          END
   IF;
   END
   LOOP;
   CLOSE member_cursor;
END;
/
```

OUTPUT

Number of max books allowed is updated for Member with ID 2 Number of max books allowed is updated for Member with ID 3 Number of max books allowed is updated for Member with ID 4 Number of max books allowed is updated for Member with ID 6 Number of max books allowed is updated for Member with ID 8 Number of max books allowed is updated for Member with ID 5 Number of max books allowed is updated for Member with ID 7 Number of max books allowed is updated for Member with ID 1

5. Create a table BOOK_UPDATE with attribute BOOK_NO, BOOK_NAME, INCREAMENT VALUE, UPDATE_DATE and write a PL/SQL program using Explicit Cursor which can update the cost value of BOOKS_COPY table with 10 and 20 where category is Science and database respectively, and if update is possible then insert BOOK_NO, BOOK_NAME, INCREAMENT VALUE, SYSDATE to the BOOK_UPDATE table.

```
CREATE TABLE BOOK_UPDATE(
BOOK_NO NUMBER(4),
BOOK_NAME VARCHAR(20),
INCREMENT_VALUE NUMBER(4),
UPDATE_DATE DATE
);

--PL SQL

Program

DECLARE
   CURSOR books_cursor IS SELECT Book_No, Category FROM
   BOOKS_COPY; BNo BOOKS_COPY.Book_No%TYPE;
   Bcategory
BOOKS_COPY.Category%TYPE; BEGIN

   OPEN books_cursor;

   LOOP
      FETCH books_cursor INTO BNo,BCategory;
      EXIT WHEN
```

```
books_cursor%NOTFOUND; IF

(BCategory = 'Science') THEN
```

```
        UPDATE BOOKS_COPY SET Cost=Cost+10 WHERE Category=BCategory;
        INSERT INTO BOOK_UPDATE SELECT BOOK_NO, BOOK_NAME, 10, SYSDATE
FROM BOOKS_COPY WHERE Book_No=BNo;
      ELSIF (BCategory = 'Database') THEN
        UPDATE BOOKS_COPY SET Cost=Cost+20 WHERE Category=BCategory;
        INSERT INTO BOOK_UPDATE SELECT BOOK_NO, BOOK_NAME, 20,
SYSDATE FROM BOOKS_COPY WHERE Book_No=BNo;
      END IF;
   END
   LOOP;
   CLOSE books_cursor;
END;
/
```

OUTPUT

BOOK_NO	BOOK_NAME	INCREMENT_VALUE	UPDATE_DATE
102	Oracle-Complete Ref	20	05/09/2020
105	PL SQL-Ref	20	05/09/2020
107	Optics	10	05/09/2020
104	Mastering SQL	20	05/09/2020

6. Write a PL/SQL program using Explicit Cursor which can display the all information of 5 books from BOOK_COPY table according to the higher cost.

```
DECLARE
   CURSOR books_cursor IS (SELECT * FROM (SELECT * FROM BOOKS_COPY ORDER BY
   Cost DESC));
   books_record books_cursor%ROWTYPE;
   countnum NUMBER(2);
BEGIN
   countnum:=0;

   OPEN books_cursor;
   dbms_output.Put_line('Book_No Book_Name        Author_Name      Cost
                                    Category'); dbms_output.Put_line('.....');

   LOOP
      FETCH books_cursor INTO books_record;
      EXIT WHEN countnum=5 OR books_cursor%NOTFOUND;
      countnum:=countnum+1;
      dbms_output.Put_line(Rpad(books_record.Book_No,11,' ') ||
Rpad(books_record.Book_Name, 22 , ' ') || Rpad(books_record.Author_name,21, ' ') ||
Rpad(books_record.Cost,9,' ') || books_record.Category);
   END LOOP;
```

```
  CLOSE books_cursor;
END;
```

OUTPUT

Book_No	Book_Name		Author_Name	Cost	Category
105	PL SQL-Ref		Scott Urman	1050	Database
102	Oracle-Complete	Ref	Loni	850	Database
106	UNIX		Sumitava Das	850	System
104	Mastering SQL		Loni	760	Database
103	Visual Basic 10		BPB	700	Others