

EXERCISE 7 AND 8

PL/SQL Programming and Sequential Constructs

LIVESQL LINKS:

Exercise 7: <https://livesql.oracle.com/apex/livesql/s/pfufuo62nrfcuyqqobbraxj3u>

Exercise 8: <https://livesql.oracle.com/apex/livesql/s/pfuf7sean8kekl12trre05nwi>

Exercise 7

Aim: To understand the concept of PL/SQL Programming

1. Write a PL/SQL block to accept a value from primary key and display any corresponding value.

The screenshot shows the Live SQL interface with a PL/SQL block executed. The block declares variables PNUM, PNAME, PLOC, and PDEPT, then selects project details from the PROJECT table where PNUM is 1945. The output displays the project name, location, and department.

```
DECLARE
PNUM NUMBER(5,0) := 1945;
PNAME VARCHAR(15);
PLOC VARCHAR(15);
PDEPT NUMBER(5,0);
BEGIN
SELECT NAME, LOCATION, DEPTNUM
INTO PNAME, PLOC, PDEPT
FROM PROJECT
WHERE NUM = PNUM;
DBMS_OUTPUT.PUT_LINE('PROJECT NAME: ' || PNAME);
DBMS_OUTPUT.PUT_LINE('PROJECT LOCATION: ' || PLOC);
DBMS_OUTPUT.PUT_LINE('PROJECT DEPARTMENT: ' || PDEPT);
END;
```

Statement processed.
PROJECT NAME: PROJECTB
PROJECT LOCATION: SALT LAKE CITY
PROJECT DEPARTMENT: 3

2. Write a PL/SQL program to delete one record from the table

The screenshot shows the Live SQL interface with two statements. Statement 10 is a PL/SQL block that deletes a record from the PROJECT table where NUM is 1945. Statement 11 is a query that selects all records from the PROJECT table. The result is displayed as a table with 4 rows.

```
Statement 10
DECLARE
BEGIN
DELETE FROM PROJECT WHERE NUM = 1945;
END;
```

Statement processed.

```
Statement 11
SELECT * FROM PROJECT
```

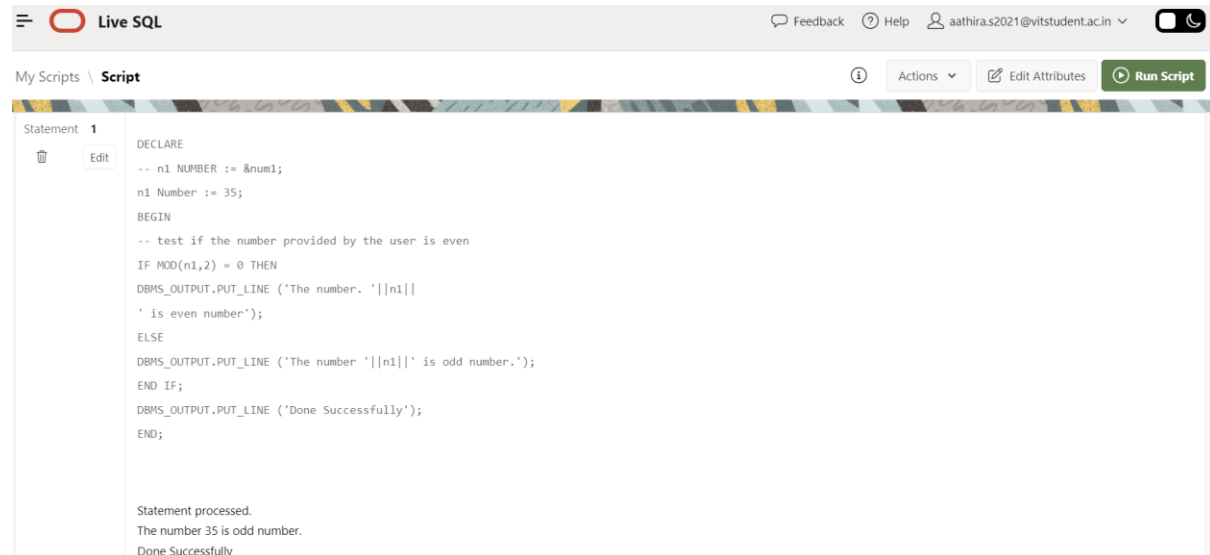
| NAME | NUM | LOCATION | DEPTNUM |
|----------|------|-----------|---------|
| PROJECTA | 3388 | HOUSTON | 1 |
| PROJECTC | 6688 | HOUSTON | 5 |
| PROJECTD | 2423 | BELALIRE | 4 |
| PROJECTE | 7745 | SUGARLAND | 5 |

Download CSV
4 rows selected.

Exercise 8

Aim: To know the usage of different sequential control structures in PL/SQL effective programming

1. Write a PL/SQL block to find whether a given number is odd or even.



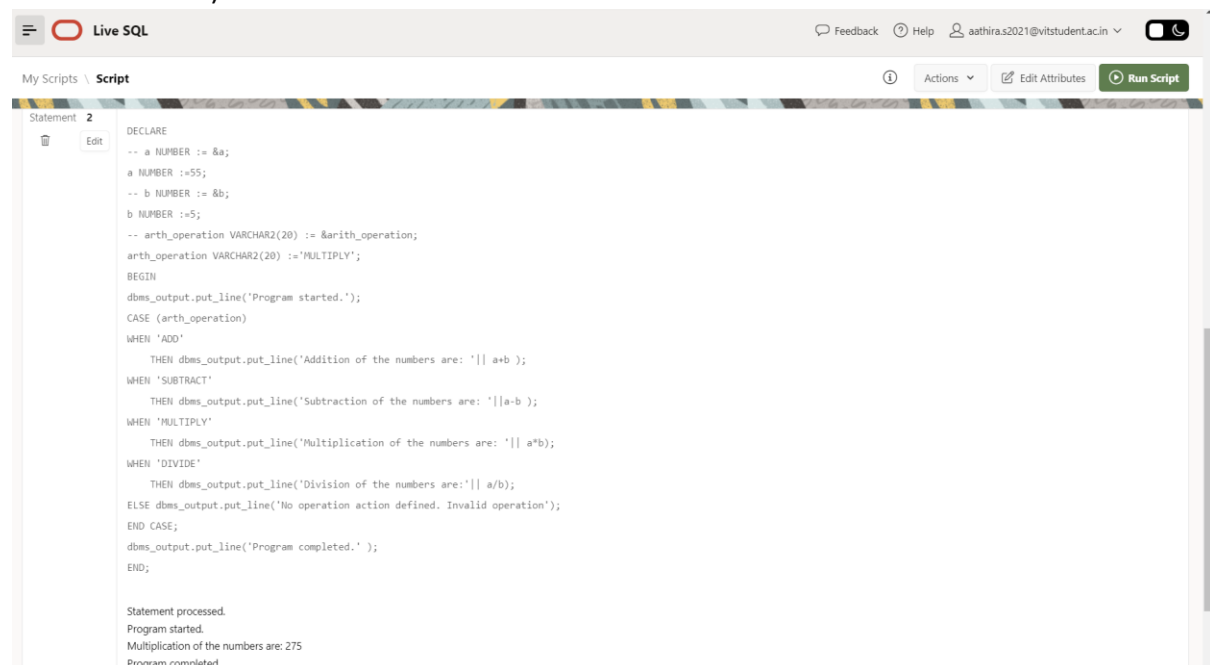
The screenshot shows the Live SQL interface with a script titled 'Script'. The script is as follows:

```
DECLARE
-- n1 NUMBER := &num1;
n1 Number := 35;
BEGIN
-- test if the number provided by the user is even
IF MOD(n1,2) = 0 THEN
DBMS_OUTPUT.PUT_LINE ('The number. '||n1||
' is even number');
ELSE
DBMS_OUTPUT.PUT_LINE ('The number '||n1||' is odd number. ');
END IF;
DBMS_OUTPUT.PUT_LINE ('Done Successfully');
END;
```

The output of the script is:

```
Statement processed.
The number 35 is odd number.
Done Successfully
```

2. Write a PL/SQL code as menu driven to perform arithmetic operations. (hint: use case selector....)



The screenshot shows the Live SQL interface with a script titled 'Script'. The script is as follows:

```
DECLARE
-- a NUMBER := &a;
a NUMBER :=55;
-- b NUMBER := &b;
b NUMBER :=5;
-- arth_operation VARCHAR2(20) := &arith_operation;
arth_operation VARCHAR2(20) := 'MULTIPLY';
BEGIN
dbms_output.put_line('Program started. ');
CASE (arth_operation)
WHEN 'ADD'
THEN dbms_output.put_line('Addition of the numbers are: '|| a+b );
WHEN 'SUBTRACT'
THEN dbms_output.put_line('Subtraction of the numbers are: '||a-b );
WHEN 'MULTIPLY'
THEN dbms_output.put_line('Multiplication of the numbers are: '|| a*b);
WHEN 'DIVIDE'
THEN dbms_output.put_line('Division of the numbers are: '|| a/b);
ELSE dbms_output.put_line('No operation action defined. Invalid operation');
END CASE;
dbms_output.put_line('Program completed. ');
END;
```

The output of the script is:

```
Statement processed.
Program started.
Multiplication of the numbers are: 275
Program completed.
```