

# PLSQL

Date: \_\_\_\_\_

DECLARE

&lt; declaration section &gt;

BEGIN

&lt; executable command &gt;

EXCEPTION

&lt; exception handling &gt;

END;

dbms\_output.put\_line ( \_\_\_\_\_ ),  
 executable c /

DECLARE

message varchar(20) := \_\_\_\_\_;

BEGIN

= assignment operator  
 => association operator

NUMBER, DATE, BOOLEAN  
 INTEGER, FLOAT, REAL

DECLARE

num1 INTEGER  
num2

V\_Emp-Job Emp.JOB%.TYPE

DECLARE

N NUMBER := 1432

R NUMBER;

BEGIN

R: MOD(N, 2)

IF R = 0 THEN

dbms\_output.put\_line('Even');

ELSE

(dbms\_output.put\_line('Odd'))

END IF

END;



DECLARE  
num1 NUMBER := 14

~~BE~~  
BEGIN

IF num1 < 0 THEN

~~ELSE IF~~ ELSIF THEN

ELSE  
END IF  
END;

BEGIN

CASE OPERATOR

WHEN '+' THEN

WHEN '-' THEN

loop

FOR i IN

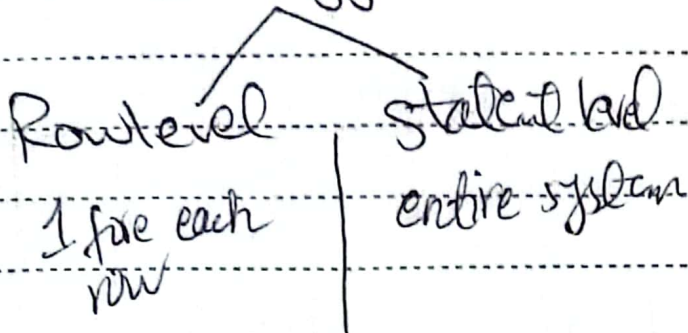
fact := fact \* i,

END loop

num loop

WHEN NO DATA FOUND THEN

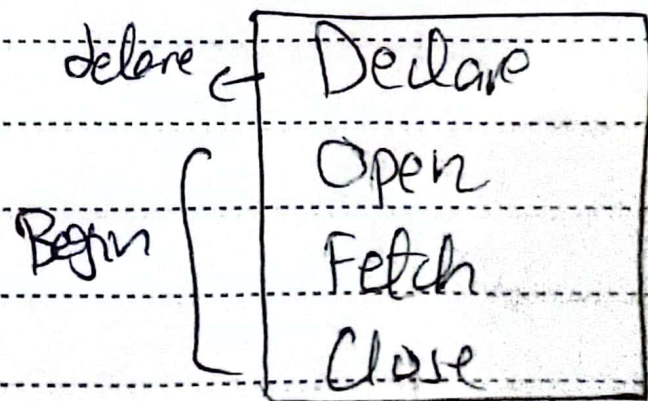
Trigger → automatically executed



Cursor : A cursor hold multiple rows returned by a SQL statement.

Type { 1) Implicit (Oracle)  
2) Explicit (users)

CURSOR CI IS SELECT statement  
OPEN CI  
FETCH CI INTO ...  
CLOSE CI





Example

Declare

```
c_id      customerid %type;
c_name    customername %type;
Cursor CI IS
    SELECT id, name, FROM customers;
```

BEGIN

OPEN CI;

Loop:

Fetch CI INTO c\_id, c\_name;

EXIT WHEN CI %NOT FOUND;

DBMS\_OUTPUT.PUT\_LINE(c\_id || ' ' || c\_name);

END Loop;

Close CI;

END;

~~DECLARE~~

~~Cursor CI IS~~

~~SELECT \* FROM Emp~~

~~BEGIN~~

~~open CI~~

Date: \_\_\_\_\_

DECLARE CURSOR EMP\_CUR IS  
SELECT \* FROM EMP1;

BEGIN  
FOR EMP\_A IN EMP\_CUR LOOP

CREATE OR REPLACE PACKAGE Emp\_A  
function f\_name ( ) RETURN NUMBER  
END Package;

CREATE OR REPLACE EMPLOYEES AS  
FUNCTION f\_name ( ) RETURN \_\_\_\_\_  
END Pkname;