PLSQL

```
DECLARE
        <declaration section>
BEGIN
        <exeatable camand>
EXCEPTION
        <exceph handly>
 END;
```

```
                 dbms_output.put_line(          );
      execube ← /
```

```
   DECLARE
        messge varchar(20) := '______';
        BEGIN
```

:= assignment operator

=> assosiation operatoy

NUMBE, DATE, BOOLEAN
INTEGER, FLOAT, REAL

→ View
→ INS
→ PLSQL

```
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 1..num LOOP
              factul := fact*i;
            END loop
```

Date:

WHEN NO_DATA_FOUND THEN

Trigger -> automatically execuled

Rowlevel	statemt level
1 fire each row	entire system

Cursor: A cursor hold multiple rows returned by a SQL statemt

Type
1) Implicit (oracle)
2) Explicit (users)

```
CURSOR C1 IS SELECT Statemt
OPEN C1
FETCH C1 INTO ..
CLOSE C1
```

declare <- Declare

Begin
- Open
- Fetch
- Close

Example.

```
    Declare
        C_id      customes.id %type;
        c-name    customer.name %type;
        Cursor  C1  IS
            SELECT  id, name, FROM custmrs;

  BEGIN
        OPEN C1;
        Loop
          Fetch C1 INTO  C-id, C-name;
           EXIT  WHEN  C1 %NOT FOUND.
           DBMS.OUTPUT.PUT.LINE(C-Id || '  ' ||C-n
        END Loop;
        close C1;
END;
```

```
  DELEAR
      Cursor  C1  IS
           SELECT  *  FROM Emp
  BEGIN
           OPEN  C1
```

```
DECLARE CURSOR EMCUR IS
  SELECT * FROM EMI;
BEGIN
  FOR EMP RC IN EMPCUR. LOOP
```

```
CREATE OR REPLACE PACKAGE EmpAS
  function fname (      ) RETUR: NUMBER
  END packge
```

```
CREATE OR REPLACE EMPLOUNT AS
  FUNCTION fnam(      )    RETUN ____
  END  Pkmn.
```