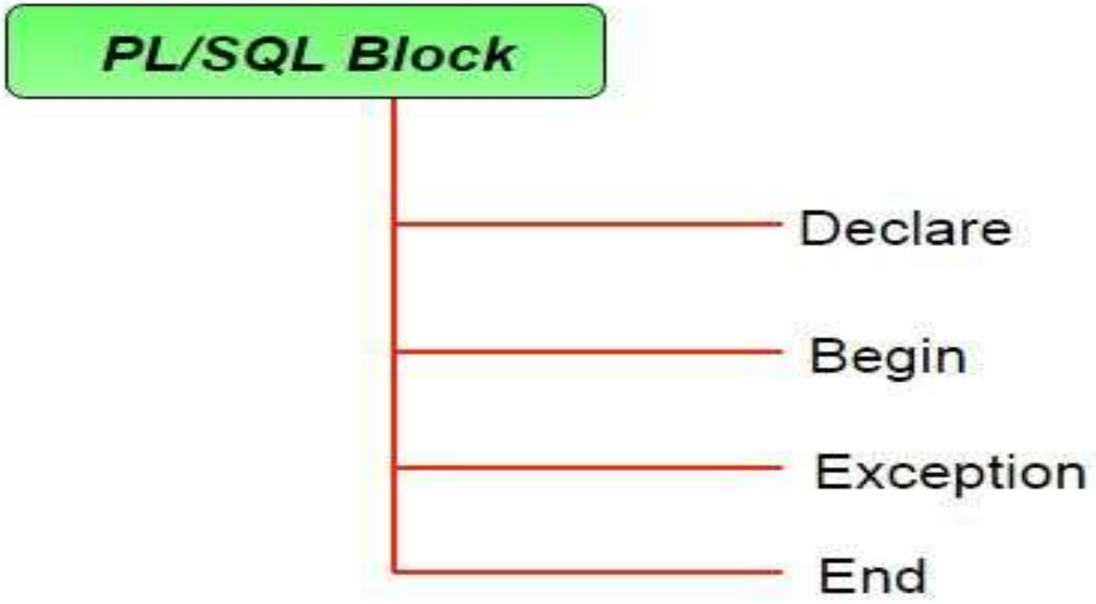
11.Demonstrate Basic variables, Anchored Declarations and Usage of Assignment Operation Using PL/SQL block

PL/SQL

- PL/SQL stands for Procedural Language extension to the Structured Query Language (SQL).
- PL/SQL is a combination of SQL along with the procedural features of programming languages.
- Oracle uses a PL/SQL engine to processes the PL/SQL statements.
- PL/SQL includes procedural language elements like conditions and loops.
- It allows declaration of constants and variables, procedures and functions, triggers,cursors .

Structure of PL/SQL Block

The basic unit in PL/SQL program is a block.



Structure of PL/SQL Block

- **DECLARE** --- Optional

  variables ,functions, constants, cursors, ….
- **BEGIN** --- Mandatory

  SQL statements(DML,TCL)

  Procedural statements(conditional ,loops etc)
- **EXCEPTIONS** --- optional

  exception handling statements ;
- **END;** ----Mandatory

Variables in PL/SQL

- A **variable** is a meaningful name that provides facility for programmer to store data temporary during execution of code. It helps to manipulate data in PL/SQL.
- **syntax:** variable_name datatype [(size)];

  variable_name **:=** value ;

Find addition of two numbers using PL/SQL block

```
DECLARE
   a integer := 10;
   b integer := 20;
   c integer;
BEGIN
  c := a + b;
  dbms_output.put_line(‘sum is ' || c);
END;
```

output :

```
Sum is 30
```

Find employee details using PL/SQL block

```
DECLARE
      v_ename varchar2(20);
      v_salary Number;
BEGIN
      SELECT  ename , sal
      INTO   v_ename, v_salary
      FROM emp WHERE eno = 101;
      DBMS_OUTPUT.PUT_LINE (‘Emp name is’ ||
       v_ename || ‘ And salary is’ || v_salary);
END;
```

Anchored Declarations

- When you declare a PL/SQL variable to hold the value of a table column, it must be declared as of column data type and precision, otherwise error will occur on execution.
- PL/SQL provides the facility to declare a variable without having to specify a particular data type using **%TYPE** attributes.

Anchored Declarations

- The **%TYPE** is used to declare variables according to the already declared PL/SQL variable or database column.
- The data type and precision of the variable declared using %TYPE attribute is the same as that of the column that is referred from a given table.
- This is particularly useful when declaring variables that will hold database values.

Anchored Declarations

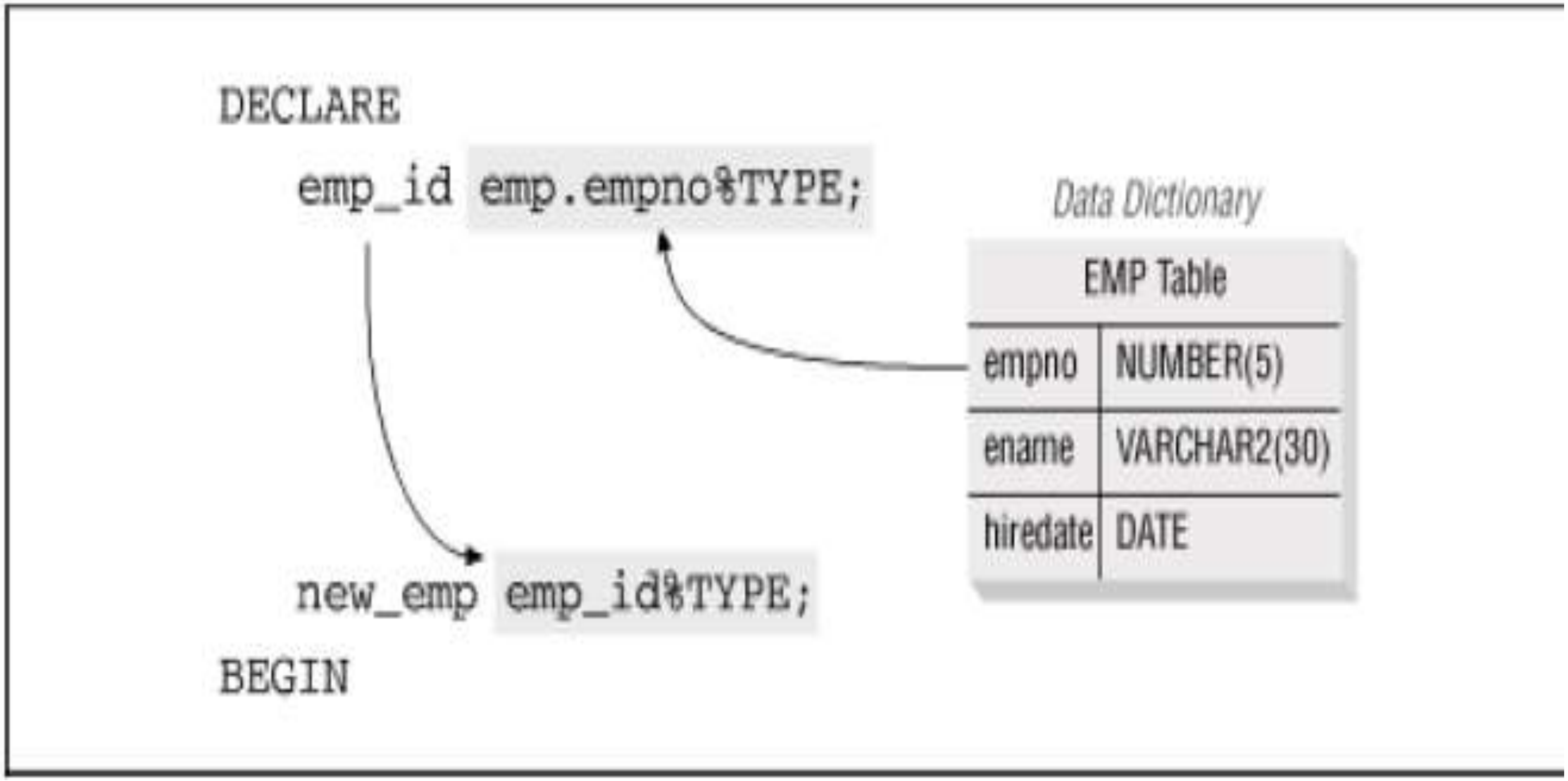
syntax for declaring a variable with %TYPE

var_name <tab_name>.<column_name> **%TYPE**;

Example: SALARY EMP.SAL % TYPE;

- This declaration will declare a variable **SALARY** that has the same data type as column **SAL** of the **EMP** table.

Anchored Declarations



```
DECLARE
   emp_id emp.empno%TYPE;


   new_emp emp_id%TYPE;
BEGIN
```

Data Dictionary

EMP Table	
empno	NUMBER(5)
ename	VARCHAR2(30)
hiredate	DATE

Find employee salary using PL/SQL BLOCK

```
DECLARE
SALARY  EMP.SAL % TYPE;
ECODE    EMP.eno % TYPE;
BEGIN
Ecode :=  &Ecode;
SELECT SAL INTO SALARY FROM EMP WHERE ENO =
    Ecode;
DBMS_OUTPUT.PUT_LINE('Salary of ' || ECODE ||
   ‘ is ’ || salary);
END;
```

Find employee salary using PL/SQL BLOCK

OUTPUT:

```
Enter value for ecode: 102
Salary of 102 is = 1500
PL/SQL procedure successfully completed.
```