**PL/SQL**

**Basic Structure**

-- DECLARE

--   message varchar2(100) : = 'Hello, World!';

-- BEGIN

--  dbms\_output.put\_line('WELCOME  '||message);

-- END;

-- DECLARE

-- NAME varchar2(20):='CHANDRA';

-- BEGIN

-- dbms\_output.put\_line('WELCOME '||NAME);

-- END;

**Employee Problem**

DECLARE

EID number(5):=101;

ENAME varchar2(20):='CHANDRA';

ESALARY number(20) default 5000;

BEGIN

dbms\_output.put\_line('EMPLOYEE ID: '||EID);

dbms\_output.put\_line('EMPLOYEE NAME: '||ENAME);

dbms\_output.put\_line('EMPLOYEE SALARY: '||ESALARY);

END;

**Input Taking From User In Pl/Sql**

DECLARE

N1 number(5);

N2 number(5);

BEGIN

dbms\_output.put\_line('ENTER THE VALUE FOR');

N1 := &N1;

dbms\_output.put\_line('ENTER THE VALUE FOR NUM2');

N2 := &N2;

dbms\_output.put\_line('ADDITION:  '||(N1+N2));

dbms\_output.put\_line('SUBSTRACTION:  '||(N1-N2));

dbms\_output.put\_line('MULTIPLICATION:  '||(N1\*N2));

dbms\_output.put\_line('DIVISION:  '||(N2/N1));

END;

**Fetching Data Type and Values From Another Table And Using**

create table tbl\_employee(eid number(10),ename varchar2(20),esalary number(10));

insert into tbl\_employee values(101,'Chandra',5000);

insert into tbl\_employee values(102,'Manu',12353);

insert into tbl\_employee values(103,'Raju',4000);

insert into tbl\_employee values(104,'Gowtham',7000);

select \*from tbl\_employee;

-- DECLARE

-- name tbl\_employee.ename%type;

-- salary tbl\_employee.esalary%type;

-- BEGIN

-- select ename,esalary into name,salary from tbl\_employee where eid=103;

-- dbms\_output.put\_line('employee name : '||name);

-- dbms\_output.put\_line('employee salary : '||salary);

-- end;

DECLARE

name tbl\_employee.ename%type;

salary tbl\_employee.esalary%type;

BEGIN

select ename,esalary into name,salary from tbl\_employee where eid=&eid;

dbms\_output.put\_line('employee name : '||name);

dbms\_output.put\_line('employee salary : '||salary);

end;

**--Nested Blocks**

DECLARE

v\_outer\_variable varchar2(20):='Global Variable';

BEGIN

DECLARE

v\_inner\_variable varchar2(20):='Local Variable';

BEGIN

dbms\_output.put\_line(v\_inner\_variable);

dbms\_output.put\_line(v\_outer\_variable);

END;

dbms\_output.put\_line(v\_outer\_variable);

END;

**--Conditional Statements**

**--if else**

DECLARE

num number(10);

num1 number(10);

BEGIN

num:=&num;

num1:=&num1;

if num>num1 then

dbms\_output.put\_line('Num is Greater');

else

dbms\_output.put\_line('Num1 is Greater');

end if;

END;

**--if elsif else**

DECLARE

num number(10);

num1 number(10);

num2 number(10);

BEGIN

num:=&num;

num1:=&num1;

num2:=&num2;

if num>num1 and num>num2 then

dbms\_output.put\_line('Num is Greater');

elsif num1>num and num1>num2 then

dbms\_output.put\_line('Num1 is Greater');

else

dbms\_output.put\_line('Num2 is Greater');

end if;

END;

**--Switch Case**

DECLARE

col char(1);

BEGIN

col:=&col;

case col

when 'r' then

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

when 'g' then

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

when 'b' then

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

else

dbms\_output.put\_line('UNKNOW COLOR');

end case;

END;

**Loops**

**--Simple Loop**

DECLARE

num pls\_integer;

rem pls\_integer default 0;

-- rev number(3) default 0;

BEGIN

num:=&num;

loop

rem:=rem\*10+mod(num,10);

-- rev:=rev\*10+rem;

num:=trunc((num/10),0);

exit when num = 0;

end loop;

dbms\_output.put\_line(rem);

end;

**--While Loop**

DECLARE

num pls\_integer;

rem pls\_integer default 0;

rev pls\_integer default 0;

BEGIN

num:=&num;

while num!=0

loop

rem:=mod(num,10);

rev:=rev\*10+rem;

num:=trunc((num/10),0) ;

end loop;

dbms\_output.put\_line(rev);

end;

**--For Loop in Forword Direction**

DECLARE

sum1 number(20) default 0;

x number(10) default 0;

BEGIN

for x in 1..11

loop

sum1:=sum1+x;

end loop;

dbms\_output.put\_line(sum1);

end;

**--For Loop in Reverse Direction**

DECLARE

sum1 number(20) default 0;

x number(10) default 0;

BEGIN

for x in reverse 1..11

loop

sum1:=sum1+x;

end loop;

dbms\_output.put\_line(sum1);

end;

**--Records**

create table tbl\_employee(eid number(10),ename varchar2(20),esalary number(10));

insert into tbl\_employee values(101,'Chandra',5000);

insert into tbl\_employee values(102,'Manu',12353);

insert into tbl\_employee values(103,'Raju',4000);

insert into tbl\_employee values(104,'Gowtham',7000);

select \*from tbl\_employee;

DECLARE

my\_rec tbl\_employee%rowtype;

BEGIN

select \* into my\_rec from tbl\_employee where eid=&eid;

dbms\_output.put\_line('Employee Id: '||my\_rec.eid);

dbms\_output.put\_line('Employee Name: '||my\_rec.ename);

dbms\_output.put\_line('Employee Salary: '||my\_rec.esalary);

end;

**--Procedure**

create table tbl\_employee(eid number(10),ename varchar2(20),esalary number(10));

insert into tbl\_employee values(101,'Chandra',5000);

insert into tbl\_employee values(102,'Manu',12353);

insert into tbl\_employee values(103,'Raju',4000);

insert into tbl\_employee values(104,'Gowtham',7000);

select \*from tbl\_employee;

create or replace procedure MyProc is

v\_name varchar2(30);

begin

select ename into v\_name from tbl\_employee where eid=103;

dbms\_output.put\_line('Employee Name: '||v\_name);

end;

/

--Procedure call

begin

MyProc;

end;

--Division

create or replace procedure division(a number,b number) is

div number(5);

begin

div:=a/b;

dbms\_output.put\_line('Division: '||div);

end;

/

begin

division(7,3);

end;

**--Function**

create or replace function division(a number,b number) return number is

div number(5);

begin

div := a / b;

return div;

end;

/

declare

res number;

begin

res := division(7,3);

dbms\_output.put\_line('Division: '||res);

end;

**--Parameters**

**--In Parameter**

create or replace procedure inpar(a IN number) is

begin

a := a +1;

dbms\_output.put\_line('InPar: '||a);

end;

/

begin

inpar(10);

end;

**--Out Parameter**

create or replace procedure outpar(a out number) is

begin

a := 10;

dbms\_output.put\_line('OutPar: '||a);

end;

/

declare

no number(3):=20;

begin

outpar(no);

end;

**--In Out Parameter**

create or replace procedure inoutpar(a in out number) is

begin

dbms\_output.put\_line('OutPar: '||a);

end;

/

declare

no number(3):=20;

begin

inoutpar(no);

end;

**--Cursor**

**--Explicit Cursor**

create table tbl\_employee(eid number(10),ename varchar2(20),esalary number(10));

insert into tbl\_employee values(101,'Chandra',5000);

insert into tbl\_employee values(102,'Manu',12353);

insert into tbl\_employee values(103,'Raju',4000);

insert into tbl\_employee values(104,'Gowtham',7000);

select \*from tbl\_employee;

DECLARE

CURSOR c\_employee IS

SELECT eid, ename, esalary FROM tbl\_employee;

id tbl\_employee.eid%TYPE;

name tbl\_employee.ename%TYPE;

salary tbl\_employee.esalary%TYPE;

BEGIN

OPEN c\_employee;

LOOP

FETCH c\_employee INTO id, name, salary;

EXIT WHEN c\_employee%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(id || ' ' || name || ' ' || salary);

END LOOP;

CLOSE c\_employee;

END;

**--Exception**

DECLARE

V\_NAME tbl\_employee.ename%TYPE;

BEGIN

select ename into V\_NAME from tbl\_employee where eid=105;

DBMS\_OUTPUT.PUT\_LINE('Employee Name '|| V\_NAME);

exception

WHEN NO\_DATA\_FOUND then

DBMS\_OUTPUT.PUT\_LINE('Invalid Employee Id');

END;

**--User defined Exception**

create or replace procedure MyProc(v\_age number) is

Age\_error exception;

BEGIN

if v\_age<=0 then

raise Age\_error;

else

DBMS\_OUTPUT.PUT\_LINE('You are '||v\_age);

end if;

exception

WHEN Age\_error then

DBMS\_OUTPUT.PUT\_LINE('Age can not be 0 or less than 0');

end;

/

BEGIN

MyProc(0);

END;

**--Package**

CREATE OR REPLACE PACKAGE arithmetic\_pkg AS

FUNCTION add(a IN NUMBER, b IN NUMBER) RETURN NUMBER;

FUNCTION sub(p\_num1 IN NUMBER, p\_num2 IN NUMBER) RETURN NUMBER;

END arithmetic\_pkg;

/

CREATE OR REPLACE PACKAGE BODY arithmetic\_pkg AS

FUNCTION add(a IN NUMBER, b IN NUMBER) RETURN NUMBER IS

addition number(5);

BEGIN

addition:=a+b;

RETURN addition;

END add;

FUNCTION sub(p\_num1 IN NUMBER, p\_num2 IN NUMBER) RETURN NUMBER IS

BEGIN

RETURN p\_num1 - p\_num2;

END sub;

END arithmetic\_pkg;

/

DECLARE

v\_sum NUMBER;

v\_diff NUMBER;

BEGIN

v\_sum := arithmetic\_pkg.add(10, 5);

v\_diff := arithmetic\_pkg.sub(10, 5);

DBMS\_OUTPUT.PUT\_LINE('Sum: ' || v\_sum); -- Output: Sum: 15

DBMS\_OUTPUT.PUT\_LINE('Difference: ' || v\_diff); -- Output: Difference: 5

END;

**--Trigger**

CREATE TABLE product (

product\_id NUMBER(5),

product\_name VARCHAR2(32),

supplier\_name VARCHAR2(32),

unit\_price NUMBER(7,2)

);

INSERT INTO product (product\_id, product\_name, supplier\_name, unit\_price) VALUES (1312, 'Wooden\_Door', 'Galaxy', 950.00);

INSERT INTO product (product\_id, product\_name, supplier\_name, unit\_price) VALUES (1313, 'Plastic\_Door', 'Tanmay', 1950.00);

INSERT INTO product (product\_id, product\_name, supplier\_name, unit\_price) VALUES (1314, 'Metal\_Door', 'Sun', 11450.00);

CREATE OR REPLACE TRIGGER my\_product

BEFORE INSERT or delete or UPDATE ON product

FOR EACH ROW

BEGIN

dbms\_output.put\_line('Product Table Modified');

END;

/

delete FROM product WHERE product\_id = 1312;

delete FROM product WHERE product\_id = 1313;

SELECT \* FROM product;