Lesson 03 – Handling variables in PL/SQL

SLIDE 15 - Base Scalar Data Types - Example

DECLARE

v\_job VARCHAR2(9) := 'TRAINER';

v\_count BINARY\_INTEGER := 0;

v\_total\_sal NUMBER(9,2) := 0;

v\_orderdate DATE := SYSDATE + 7;

c\_tax\_rate CONSTANT NUMBER(3,2) := 8.25;

v\_valid BOOLEAN NOT NULL := TRUE;

BEGIN

dbms\_output.put\_line('The job is :' || v\_job);

dbms\_output.put\_line('The count is :' || v\_count);

dbms\_output.put\_line('The total salary is :' || v\_total\_sal);

dbms\_output.put\_line('The order date is :' || v\_orderdate);

dbms\_output.put\_line('The tax rate is :' || c\_tax\_rate);

IF(v\_valid = TRUE) THEN

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

end if;

END;

/

SLIDE 18 - Declaring Datatype by using %ROWTYPE

DECLARE

nRecord staff\_masters%rowtype;

BEGIN

SELECT \* into nrecord

FROM staff\_masters

WHERE staff\_code = 100001;

UPDATE staff\_masters

SET staff\_sal = staff\_sal + 101

WHERE staff\_code = 100001;

END;

/

SLIDE 19 - Inserting and Updating using records

DECLARE

dept\_info department\_masters%ROWTYPE;

BEGIN

-- dept\_code, dept\_name are the table columns.

-- The record picks up these names from the %ROWTYPE.

dept\_info.dept\_code := 70;

dept\_info.dept\_name := 'PERSONNEL';

/\*Using the %ROWTYPE means we can leave out the column list (deptno, dname) from the INSERT statement. \*/

INSERT into department\_masters VALUES dept\_info;

END;

/

SLIDE 26 - Here is an example for declaring and using Record datatype:

DECLARE

TYPE recname is RECORD

(customer\_id number,

customer\_name varchar2(20));

var\_rec recname;

BEGIN

var\_rec.customer\_id:=20;

var\_rec.customer\_name:='Smith';

dbms\_output.put\_line(var\_rec.customer\_id||' '||var\_rec.customer\_name);

END;

/

SLIDE 31 - Here is an example of referencing PL/SQL tables

DECLARE

TYPE staff\_table is table of staff\_masters.staff\_name%type

INDEX BY BINARY\_INTEGER;

staff\_tab staff\_table;

BEGIN

staff\_tab(1) := 'Smith'; --update Smith's salary

UPDATE staff\_masters

SET staff\_sal = 1.1 \* staff\_sal

WHERE staff\_name = staff\_tab(1);

END;

/

SLIDE 36 - Scope and Visibility of Variables

<<OUTER>>

DECLARE

V\_Flag BOOLEAN ;

V\_Var1 CHAR(9) := 'outer';

BEGIN

<<INNER>>

DECLARE

V\_Var1 NUMBER(9) := '100';

V\_Date DATE;

BEGIN

dbms\_output.put\_line('Inner number variable - v\_var1 :' || v\_var1);

END;

dbms\_output.put\_line('Outer char variable - v\_var1 :' || v\_var1);

END;

/

SLIDE 37 - Types of Statements- Insert

DECLARE

v\_dname varchar2(15) := 'Accounts';

BEGIN

INSERT into department\_masters

VALUES (60, v\_dname);

END;

/

SELECT \* FROM DEPARTMENT\_MASTERS

/

SLIDE 38 - Types of Statements – DELETE

DECLARE

v\_sal\_cutoff number := 2000;

BEGIN

DELETE FROM staff\_masters

WHERE staff\_sal < v\_sal\_cutoff;

END;

/

SLIDE 39 - Types of Statements – UPDATE

DECLARE

v\_sal\_incr number(5) := 1000;

BEGIN

UPDATE staff\_masters

SET staff\_sal = staff\_sal + v\_sal\_incr

WHERE staff\_name='Smith';

END;

/

SLIDE 42 - Types of Statements – SELECT

<<Block1>> --to be included in the program. This is a flag which is referred in the program

DECLARE

deptno number(10) := 70;

dname varchar2(15) ;

BEGIN

SELECT dept\_name INTO dname FROM department\_masters WHERE dept\_code = Block1.deptno;

DELETE FROM department\_masters

WHERE dept\_code = Block1. deptno ;

END;

/

SLIDE 46 - IF Construct – Example

DECLARE

v\_staffno NUMBER := &staffno;

BEGIN

IF v\_staffno = 100003

THEN

UPDATE staff\_masters

SET staff\_sal = staff\_sal + 100

WHERE staff\_code = 100003 ;

END IF;

END;

/

SLIDE 51 and SLIDE 54- Simple Loop with EXIT

DECLARE

v\_counter number := 70 ;

BEGIN

LOOP

INSERT INTO department\_masters

VALUES(v\_counter,'new dept');

v\_counter := v\_counter + 10;

EXIT WHEN v\_counter >= 80;

END LOOP;

COMMIT ;

END ;

/

SLIDE 56 - For Loop – Example

ALTER TABLE department\_masters

disable constraint SYS\_C0010213

/ -- the above code needs to be executed to disable unique constraint applied on dept\_name column.

DECLARE

v\_counter number := 60 ;

BEGIN

FOR Loop\_Counter IN 2..5

LOOP

INSERT INTO dept(deptno,dname)

VALUES(v\_counter ,'NEW DEPT') ;

v\_counter := v\_counter + 10 ;

END LOOP;

COMMIT ;

END ;

/