**PRACTICAL NO 3**

**CREATE CONDITIONAL STATEMENT USING PL/SQL**

1. USING IF STATEMENT

* The IF-THEN statement is mainly used to execute a particular section of codes only when the condition is satisfied.
* The condition should yield Boolean (True/False). It is a basic conditional statement which will allow the ORACLE to execute/skip a particular piece of code based on the pre-defined conditions.

**Syntax:-**

**IF <condition: returns Boolean>**

**THEN**

**-executed only if the condition returns TRUE**

**<action\_block>**

**END if;**

Example 1:- WAP to check if 2 strings are same or not.

Set serveroutput on;

DECLARE

str1 VARCHAR(12);

srt2 VARCHAR(12);

BEGIN

str1 := 'HELLO';

str2 := 'HELLO';

IF str1 LIKE str2 THEN

DBMS\_OUTPUT.PUT\_LINE(str1 || 'is same like ' || str2);

END IF;

END;

/

EXAMPLE 2: (Refer to customer table FROM PRACTICAL 1)

DECLARE

c\_id customers.id%type := 1;

c\_sal customers.salary%type;

BEGIN

SELECT salary

INTO c\_sal

FROM customers

WHERE id = c\_id;

IF (c\_sal <= 2000) THEN

UPDATE customers

SET salary = salary + 1000

WHERE id = c\_id;

dbms\_output.put\_line ('Salary updated');

END IF;

END;

/

EXAMPLE 3 :

Write a PL/SQL code block that will accept an account number from the user, check if the users balance is less than the minimum balance 2000rs only then deduct Rs. 100/- from the balance. The process is fired on acc\_mstr table.

SQL> create table acc\_mstr2(acc\_no number 2 (10),curr\_bal number(10,2));

SQL> select \* from acc\_mstr2;

declare mcur number(10,2);

mac number(10);

begin mac:=&mac;

select curr\_bal into mcur from acc\_mstr where acc\_no=mac;

if mcur<2000 then

update acc\_mstr set curr\_bal=curr\_bal-100 where acc\_no=mac;

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

end if;

end;

/

1. **USING IF-ELSE STATEMENT**

The IF-THEN-ELSE statement is mainly used to select between two alternatives based on the condition.

Syntax:-

IF <condition: returns Boolean>

THEN

-executed only if the condition returns TRUE

<action\_blockl>

ELSE

-execute if the condition failed (returns FALSE)

<action\_block2>

END if;

Example 1:- WAP to print message whether the given number is odd or even.

DECLARE

a NUMBER:= ‘&a’;

BEGIN

dbms\_output.put\_line (‘Program started');

IF( mod(a,2)=0) THEN

dbms\_output.put\_line('a is even number' );

ELSE

dbms\_output.put\_line('a is odd number1);

END IF;

dbms\_output.put\_line (‘Program completed.’);

END;

/

1. **USING ELSEIF LADDER**

* The IF-THEN-ELSIF statement is mainly used where one alternative should be chosen from a set of alternatives, where each alternative has its own conditions to be satisfied.
* The first conditions that return <TRUE> will be executed, and the remaining conditions will be skipped.
* The IF-THEN-ELSIF statement may contain ‘ELSE’ block in it. This ‘ELSE’ block will be executed if none of the conditions is satisfied.

Syntax:-

IF <conditionl: returns Boolean>

THEN

-executed only if the condition returns TRUE <

action\_blockl>

ELSIF <condition2 returns Boolean> <

action\_block2>

ELSIF <condition3:returns Boolean> <

action\_block3>

ELSE —optional

<action\_block\_else>

END if;

Example 1:-

Set serveroutput on;

DECLARE

percent NUMERIC;

BEGIN

percent := '83';

IF percent >= 75 THEN

DBMS\_OUTPUT.PUT\_LINE('DISINCTION');

ELSIF percent >= 60 AND percent <75 THEN

DBMS\_OUTPUT.PUT\_LINE('FIRST CLASS');

ELSIF percent >= 50 AND percent<60 THEN

DBMS\_OUTPUT.PUT\_LINE('SECOND CLASS ');

ELSIF percent >= '40' AND percent <50 THEN

DBMS\_OUTPUT.PUT\_LINE('PASS CLASS ');

ELSE

DBMS\_OUTPUT.PUT\_LINE('FAIL..... ');

END IF;

END;

/

Example 2:-

Create a table place(floor,room\_number,number\_of\_seats). Write a PL/SQL block to comment on type place as fairly small, little bigger, biggest depending on number\_of\_seats for a given room\_no.

CREATE TABLE place (Floor Number(4), Room\_no Number(4) CONSTRAINT p\_1 PRIMARY KEY, No\_of\_seats Number(6) );

INSERT INTO place VALUES(1,101,65);

INSERT INTO place VALUES (2,201,135);

INSERT INTO place VALUES (3,301,40);

DECLARE rmid number(4);

seats number(6);

BEGIN

rmid:=&rmid;

SELECT No\_of\_seats INTO seats FROM place WHERE rmid=Room\_no;

DBMS\_OUTPUT.PUT\_LINE('No. of seats '||seats);

IF (seats<=60) THEN

DBMS\_OUTPUT.PUT\_LINE ('Fairly small');

ELSIF (seats>60 and seats<=100) THEN

DBMS\_OUTPUT.PUT\_LINE ('Little Bigger');

ELSE DBMS\_OUTPUT.PUT\_LINE ('Biggest');

END IF;

END;

/

1. **USING CASE-WHEN LOOP**

A CASE statement is similar to IF-THEN-ELSIF statement that selects one alternative based on the condition from the available options:-

* CASE statement uses “selector” rather than a Boolean expression to choose the sequence.
* The value of the expression in the CASE statement will be treated as a selector.
* Each alternative is assigned with a certain pre-defined value (selector), and the alternative with selector value that matches the conditional expression value will get executed.
* ELSE block in CASE statement holds the sequence that needs to be executed when none of the alternatives got selected.

Syntax :-

CASE (expression)

WHEN <valuel> THEN action\_blockl;

WHEN <value2> THEN action\_block2;

WHEN <value3> THEN action\_block3;

ELSE action\_block\_default;

END CASE;

**Example 1:-**

set serveroutput on;

DECLARE

grade CHAR(10):= '&grade';

BEGIN

--grade := 'A';

CASE grade

WHEN 'O' THEN DBMS\_OUTPUT.PUT\_LINE('Outstanding');

WHEN 'A' THEN DBMS\_OUTPUT.PUT\_LINE('Excellent');

WHEN 'B' THEN DBMS\_OUTPUT.PUT\_LINE('Good');

WHEN 'C' THEN DBMS\_OUTPUT.PUT\_LINE('Satisfactory');

WHEN 'F' THEN DBMS\_OUTPUT.PUT\_LINE('Fail');

ELSE DBMS\_OUTPUT.PUT\_LINE('Invalid grade');

END CASE;

END;

/

EXAMPLE 2:-

Create a table lecturer (id,name,major\_subject,doj). Write a PL/SQL block with case when statement which print course name depending on major subject for specified lecture id.

CREATE TABLE lecturer ( Lecturer\_id Varchar2(4) CONSTRAINT z PRIMARY KEY, Name Varchar2(15), Major\_sub Varchar2(10), DOJ Date );

INSERT INTO lecturer VALUES('L001','Mayekar','Accounts','25-Jun-1999');

INSERT INTO lecturer VALUES('L002','Kulkarni','Sanskrit','25-May-1999');

INSERT INTO lecturer VALUES('L003','Sharma','Physics','15-May-1999');

INSERT INTO lecturer VALUES('L004','Varma','Biology','13-May-1999');

DECLARE

lect\_id Varchar2(4);

sub Varchar2(10);

BEGIN

lect\_id :=’&lect\_id’

SELECT Major\_sub INTO sub FROM lecturer WHERE lect\_id=Lecturer\_id;

CASE upper(sub)

WHEN ‘ACCOUNTS’ THEN

DBMS\_output.put\_line('Commerce');

WHEN 'SANSKRIT' THEN

DBMS\_output.put\_line('Arts');

WHEN 'PHYSICS' THEN

DBMS\_output.put\_line(‘Science’);

ELSE

DBMS\_output.put\_line(‘Invalid subject’);

END CASE;

END;

/