

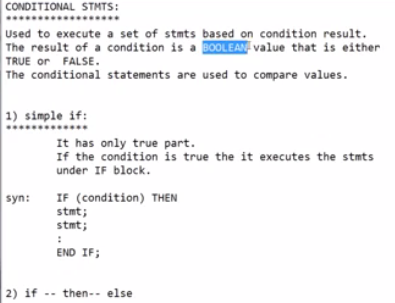
CONTROL STRUCUTRES

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CONDITIONAL STMTS

LOOPS

CASE



CONDITIONAL STMTS:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Used to execute a set of stmts based on condition result.

The result of a condition is a BOOLEAN value that is either

TRUE or FALSE.

The conditional statements are used to compare values.

1) simple if:

\*\*\*\*\*\*\*\*\*\*\*\*\*

It has only true part.

If the condition is true the it executes the stmts

under IF block.

syn: IF (condition) THEN

stmt;

stmt;

:

END IF;

2) if -- then-- else

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

It is having both true and false parts.

If the condition is true then it executes the true part and

if the condition was failed then it executes the false part.

syn: IF (condition) THEN

stmt;

stmt;

:

ELSE

stmt;

stmt;

:

END IF;

3) COMPUND IF / Nested IF:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

writing if condition with in other if condition.

syn: if (cond-1) then

stmts;

stmts;

if (cond-2) then

stmt;

:

else

stmt;

:

end if; --closing of cond-2

else

stmt;

:

end if; --closing of cond-1

4) ELSIF construct:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Writing if condition with in else part of other if condition.

If any condition is true, then it will execute set of stmts

from that if block.

IF all conditions are false , then it will execute set of stmts

from last ELSE part.

if (cond-1) then

--

elsif(cond-2) then

--

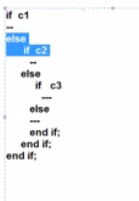
elsif(cond-3) then

--

else

--

end if;



SQL> Creating a procedure to display biggest of 2 integers with

SP2-0734: unknown command beginning "Creating a..." - rest of line ignored.

SQL> --simple if logic?

SQL> CREATE PROCEDURE BIG(X INT, Y INT)

2 AS

3 BEGIN

4 IF (X>Y) THEN

5 DBMS\_OUTPUT.PUT\_LINE(X||'IS GREATER THAN '||Y);

6 ELSIF (X<Y) THEN

7 DBMS\_OUTPUT.PUT\_LINE(Y||'IS GREATER THAN '||X);

8 ELSE

9 DBMS\_OUTPUT.PUT\_LINE(X||' IS EQUAL TO '||Y);

10 END IF;

11 END;

12 /

Procedure created.

SQL> --W A PROG. TO CHECK BIG NUMBER.

SQL> BEGIN

2 BIG(&X, &Y);

3 END;

4 /

Enter value for x: 12

Enter value for y: 13

old 2: BIG(&X, &Y);

new 2: BIG(12, 13);

13IS GREATER THAN 12

PL/SQL procedure successfully completed.

SQL> /

Enter value for x: 2000

Enter value for y: 1400

old 2: BIG(&X, &Y);

new 2: BIG(2000, 1400);

2000IS GREATER THAN 1400

PL/SQL procedure successfully completed.

SQL> /

Enter value for x: 100

Enter value for y: 100

old 2: BIG(&X, &Y);

new 2: BIG(100, 100);

100 IS EQUAL TO 100

PL/SQL procedure successfully completed.

SQL> /

Enter value for x: 100.34

Enter value for y: 134.34

old 2: BIG(&X, &Y);

new 2: BIG(100.34, 134.34);

134.34IS GREATER THAN 100.34

PL/SQL procedure successfully completed.

SQL> ED

Wrote file afiedt.buf

1 CREATE PROCEDURE HIGHER\_SALL\_EMPP(EMP1 VARCHAR2, EMP2 VARCHAR2)

2 AS

3 VSAL1 NUMBER(10);

4 VSAL2 NUMBER(10);

5 BEGIN

6 SELECT SAL INTO VSAL1 FROM EMP WHERE EMPNO=EMP1;

7 SELECT SAL INTO VSAL2 FROM EMP WHERE EMPNO=EMP2;

8 IF (VSAL1>VSAL2) THEN

9 DBMS\_OUTPUT.PUT\_LINE(EMP1||' SALARY IS GREATER THAN '||EMP2);

10 ELSIF (VSAL2>VSAL1) THEN

11 DBMS\_OUTPUT.PUT\_LINE(EMP2||' SALARY IS GREATER THAN '||EMP1);

12 ELSE

13 DBMS\_OUTPUT.PUT\_LINE(EMP1||' SALARY IS EQUAL TO '||EMP2);

14 END IF;

15\* END HIGHER\_SALL\_EMPP;

SQL> /

Procedure created.

SQL> EXEC HIGHER\_SALL\_EMPP(7788, 7654);

7788 SALARY IS GREATER THAN 7654

PL/SQL procedure successfully completed.

SQL> SELECT EMPNO, SAL FROM EMP;

EMPNO SAL

---------- ----------

7839 5000

7698 2850

7782 2450

7566 2975

7788 3000

7902 3000

7369 800

7499 1600

7521 1250

7654 1250

7844 1500

EMPNO SAL

---------- ----------

7876 1100

7900 950

7934 1300

14 rows selected.

SQL> EXEC HIGHER\_SALL\_EMPP(7788, 7839);

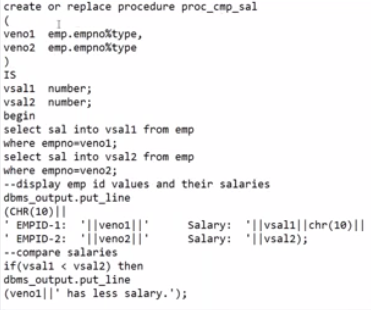
7839 SALARY IS GREATER THAN 7788

PL/SQL procedure successfully completed.

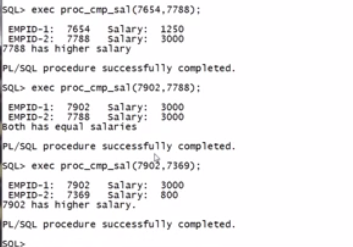
SQL> EXEC HIGHER\_SALL\_EMPP(7788, 7902);

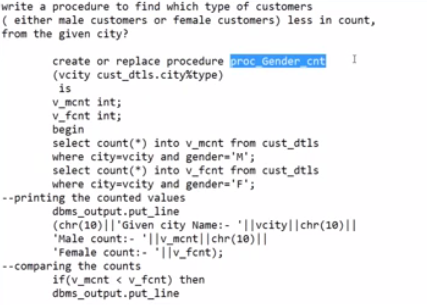
7788 SALARY IS EQUAL TO 7902

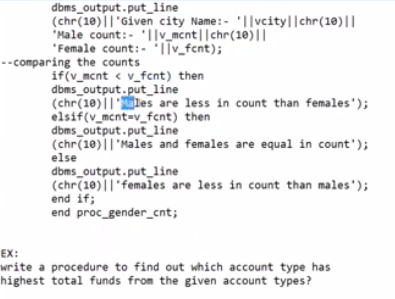
PL/SQL procedure successfully completed.

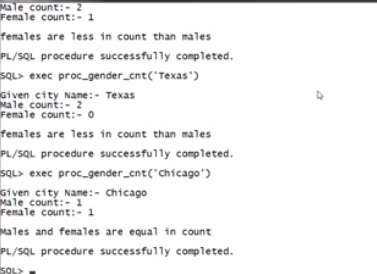


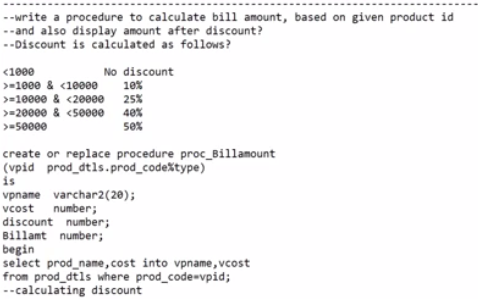


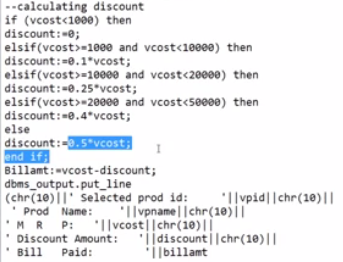


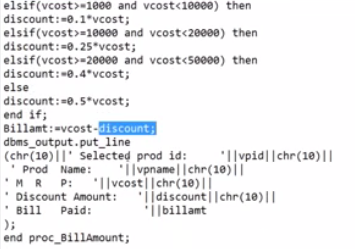


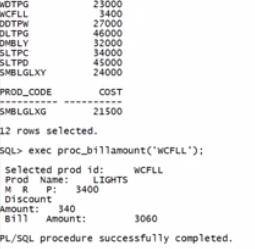












=========================================================================================

--Creating a procedure to display biggest of 2 integers with

--simple if logic?

CREATE OR REPLACE PROCEDURE PROC\_BIGINT\_1( X INT, Y INT)

IS

BEGIN

DBMS\_OUTPUT.PUT\_LINE('X='||X||' '||'Y='||Y);

IF (X>Y) THEN

DBMS\_OUTPUT.PUT\_LINE('FIRST VALUE IS BIG');

END IF;

DBMS\_OUTPUT.PUT\_LINE('end of execution');

END PROC\_BIGINT\_1;

EXEC PROC\_BIGINT\_1(10,20);

EXEC PROC\_BIGINT\_1(100,20);

--Creating a procedure to display biggest of 2 integers

--with if-then-else logic?

CREATE OR REPLACE PROCEDURE PROC\_BIGINT\_2( X INT, Y INT)

IS

BEGIN

DBMS\_OUTPUT.PUT\_LINE('X='||X||' '||'Y='||Y);

IF (X>Y) THEN

DBMS\_OUTPUT.PUT\_LINE('FIRST VALUE IS BIG');

else

DBMS\_OUTPUT.PUT\_LINE('Second VALUE IS BIG');

END IF;

DBMS\_OUTPUT.PUT\_LINE('end of execution');

END PROC\_BIGINT\_2;

exec proc\_bigint\_2(20,10);

exec proc\_bigint\_2(100,100);

--Creating a procedure to display biggest of 2 integers with

Elsif logic?

CREATE OR REPLACE PROCEDURE PROC\_BIGINT\_3( X INT, Y INT)

IS

BEGIN

DBMS\_OUTPUT.PUT\_LINE('X='||X||' '||'Y='||Y);

IF (X>Y) THEN

DBMS\_OUTPUT.PUT\_LINE('FIRST VALUE IS BIG');

elsif(x=y) then

dbms\_output.put\_line(' x and y are equal');

else

DBMS\_OUTPUT.PUT\_LINE('Second VALUE IS BIG');

END IF;

DBMS\_OUTPUT.PUT\_LINE('end of execution');

END PROC\_BIGINT\_3;

exec proc\_bigint\_3(4,5);

exec proc\_bigint\_3(5,5);

exec proc\_bigint\_3(14,5);

Ex:

Write a program to display biggest of integers if any

one given value is null?

CREATE OR REPLACE PROCEDURE PROC\_BIGINT\_4( X INT, Y INT)

IS

a int;

b int;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('X='||X||' '||'Y='||Y);

if x is null then

dbms\_output.put\_line(' Enter a valid value in the First position');

else

a:=x;

end if;

if y is null then

dbms\_output.put\_line(' Enter a valid value in the second position');

else

b:=y;

end if;

IF (a>b) THEN

DBMS\_OUTPUT.PUT\_LINE('FIRST VALUE IS BIG');

elsif(a=b) then

dbms\_output.put\_line(' Values are equal');

else

DBMS\_OUTPUT.PUT\_LINE('Second VALUE IS BIG');

END IF;

DBMS\_OUTPUT.PUT\_LINE('end of execution');

END PROC\_BIGINT\_4;

Ex:

write a procedure to find which emp getting less salary from

given 2 emp numbers?

create or replace procedure proc\_cmp\_sal

(

veno1 emp.empno%type,

veno2 emp.empno%type

)

IS

vsal1 number;

vsal2 number;

begin

select sal into vsal1 from emp

where empno=veno1;

select sal into vsal2 from emp

where empno=veno2;

--display emp id values and their salaries

dbms\_output.put\_line

(CHR(10)||

' EMPID-1: '||veno1||' Salary: '||vsal1||chr(10)||

' EMPID-2: '||veno2||' Salary: '||vsal2);

--compare salaries

if(vsal1 < vsal2) then

dbms\_output.put\_line

(veno1||' has less salary.');

elsif(vsal1=vsal2) then

dbms\_output.put\_line

('Both has equal salaries');

else

dbms\_output.put\_line

(veno2||' has less salary');

end if;

end proc\_cmp\_sal;

Ex:

write a procedure to find which type of customers

( either male customers or female customers) less in count,

from the given city?

create or replace procedure proc\_Gender\_cnt

(vcity cust\_dtls.city%type)

is

v\_mcnt int;

v\_fcnt int;

begin

select count(\*) into v\_mcnt from cust\_dtls

where city=vcity and gender='M';

select count(\*) into v\_fcnt from cust\_dtls

where city=vcity and gender='F';

--printing the counted values

dbms\_output.put\_line

(chr(10)||'Given city Name:- '||vcity||chr(10)||

'Male count:- '||v\_mcnt||chr(10)||

'Female count:- '||v\_fcnt);

--comparing the counts

if(v\_mcnt < v\_fcnt) then

dbms\_output.put\_line

(chr(10)||'Males are less in count than females');

elsif(v\_mcnt=v\_fcnt) then

dbms\_output.put\_line

(chr(10)||'Males and females are equal in count');

else

dbms\_output.put\_line

(chr(10)||'females are less in count than males');

end if;

end proc\_gender\_cnt;

EX:

write a procedure to find out which account type has

highest total funds from the given account types?

create or replace procedure proc\_high\_funds

(

vact1 cust\_act\_dtls.act\_type%type,

vact2 cust\_act\_dtls.act\_type%type

)

IS

TOT1 NUMBER;

TOT2 NUMBER;

BEGIN

SELECT SUM(ACT\_BAL) INTO TOT1

FROM CUST\_ACT\_DTLS

WHERE ACT\_TYPE=VACT1;

SELECT SUM(ACT\_BAL) INTO TOT2

FROM CUST\_ACT\_DTLS

WHERE ACT\_TYPE=VACT2;

--display total values from each account type

DBMS\_OUTPUT.PUT\_LINE

(' FUNDS FROM '||VACT1||' IS '||TOT1);

DBMS\_OUTPUT.PUT\_LINE

(' FUNDS FROM '||VACT2||' IS '||TOT2);

--comparing totals

IF (TOT1 > TOT2) THEN

DBMS\_OUTPUT.PUT\_LINE

(chr(10)||

VACT1||' HAS HIGHEST FUNDS');

ELSIF(TOT1 = TOT2) THEN

DBMS\_OUTPUT.PUT\_LINE(chr(10)||

' BOTH HAVE EQUAL FUNDS ');

ELSE

DBMS\_OUTPUT.PUT\_LINE(chr(10)||

VACT2||' HAS HIGHEST FUNDS');

END IF;

END PROC\_high\_FUNDS;

Ex:

Find which account has highest bal from given 2 account

numbers?

create or replace procedure proc\_high\_bal

(

actno1 cust\_act\_dtls.actno%type,

actno2 cust\_act\_dtls.actno%type

)

is

bal1 cust\_act\_dtls.act\_bal%type;

bal2 cust\_act\_dtls.act\_bal%type;

begin

select act\_bal into bal1 from cust\_act\_dtls where actno=actno1;

select act\_bal into bal2 from cust\_act\_dtls where actno=actno2;

dbms\_output.put\_line('First Actno: '||actno1||' ActBal: '||bal1);

dbms\_output.put\_line('Second Actno: '||actno2||' ActBal: '||bal2);

if bal1>bal2 then

dbms\_output.put\_line('First account has highest balance');

elsif(bal1=bal2) then

dbms\_output.put\_line('Both accounts having equal balance');

else

dbms\_output.put\_line('Second account has highest balance');

end if;

end proc\_high\_bal;

Ex:

create or replace procedure proc\_high\_sal

(

eid1 emp.empno%type,

eid2 emp.empno%type

)

is

sal1 number;

sal2 number;

begin

select sal into sal1 from emp

where empno=eid1;

select sal into sal2 from emp

where empno=eid2;

dbms\_output.put\_line

(

'EID-1='||eid1 ||' sal= '||sal1||chr(10)||

'EID-2='||eid2 ||' sal= '||sal2

);

if (sal1 > sal2 )

then

dbms\_output.put\_line

(eid1 ||' getting highest salary');

elsif(sal1=sal2)

then

dbms\_output.put\_line

('both are getting equal salary');

else

dbms\_output.put\_line

(eid2||' getting highest salary');

end if;

end proc\_high\_sal;

Ex:

write a procedure to find out from which city we have more

number of customers from given cities?

create or replace procedure proc\_high\_cust\_cnt

(

vcity1 cust\_dtls.city%type,

vcity2 cust\_dtls.city%type

)

is

cnt1 int;

cnt2 int;

begin

select count(\*) into cnt1 from cust\_dtls

where city=vcity1;

select count(\*) into cnt2 from cust\_dtls

where city=vcity2;

dbms\_output.put\_line

(

' city-1 : '||vcity1||' cust count: '||cnt1||chr(10)||

' city-2 : '||vcity2||' cust count: '||cnt2

);

if (cnt1>cnt2) then

dbms\_output.put\_line

(vcity1||'has higher number of customers');

elsif(cnt1=cnt2) then

dbms\_output.put\_line

('Equal number of cusotmers from the cities');

else

dbms\_output.put\_line

(vcity2||' has higher number of customers');

end if;

end proc\_high\_cust\_cnt;

Ex:

write a procedure to find out from which city we have

more number of customers with given gender, from given cities?

create or replace procedure proc\_high\_cust\_cnt

(

vcity1 cust\_dtls.city%type,

vcity2 cust\_dtls.city%type,

vgen char

)

is

cnt1 int;

cnt2 int;

begin

select count(\*) into cnt1

from cust\_dtls

where city=vcity1 and gender=vgen;

select count(\*) into cnt2

from cust\_dtls

where city=vcity2 and gender=vgen;

dbms\_output.put\_line

(chr(10)||' Gender: '||vgen||chr(10)||

' city-1 : '||vcity1||' cust count: '||cnt1||chr(10)||

' city-2 : '||vcity2||' cust count: '||cnt2

);

if (cnt1>cnt2) then

dbms\_output.put\_line

(vcity1||' has higher number of Male customers');

elsif(cnt1=cnt2) then

dbms\_output.put\_line

(' Equal number of cusotmers from the cities');

else

dbms\_output.put\_line

(vcity2||' has higher number of customers');

end if;

end proc\_high\_cust\_cnt;

Ex:

display a message based on balance like

INSUFFICIENT FUNDS

OR

TAKE YOUR MONEY?

create or replace procedure proc\_verify\_withdrawl

(

vactno cust\_act\_dtls.actno%type,

vamt number

)

is

vbal number;

begin

select act\_bal into vbal

from cust\_act\_dtls

where actno=vacctno;

if(vbal<=vamt) then

dbms\_output.put\_line

(' Insufficient balance');

else

update cust\_act\_dtls

set act\_bal=act\_bal-vamt

where actno=vactno;

dbms\_output.put\_line

(' take your money');

end if;

commit;

end proc\_verify\_withdrawl;

------------------------------------------------------------------------------------------

Ex: finding less investment for a dept from given 2 deptnos?

create or replace procedure proc\_less\_fund\_dept

(

vdno1 emp.deptno%type,

vdno2 emp.deptno%type

)

as

tot1 number;

tot2 number;

begin

select sum(sal) into tot1 from emp

where deptno=vdno1;

select sum(sal) into tot2 from emp

where deptno=vdno2;

dbms\_output.put\_line

('deptno: '||vdno1||' Investment: '|| tot1||chr(10)||

' Deptno: '||vdno2||' Investment: '||tot2);

if (tot1 < tot2) then

dbms\_output.put\_line

(' less investment for deptno: '||vdno1);

elsif(tot1=tot2) then

dbms\_output.put\_line

(' Investment for depts is equal: '||tot1);

else

dbms\_output.put\_line

(' less investment for deptno: '||vdno2);

end if;

end;

------------------------------------------------------------------------------------------

--write a procedure to calculate bill amount, based on given product id?

--Bill amount is calculated by deducting discount from original cost?

--Discount is calculated as follows?

<1000 No discount

>=1000 & <10000 10%

>=10000 & <20000 25%

>=20000 & <50000 40%

>=50000 50%

create or replace procedure proc\_Billamount

(vpid prod\_dtls.prod\_code%type)

is

vpname varchar2(20);

vcost number;

discount number;

Billamt number;

begin

select prod\_name,cost into vpname,vcost

from prod\_dtls where prod\_code=vpid;

--calculating discount

if (vcost<1000) then

discount:=0;

elsif(vcost>=1000 and vcost<10000) then

discount:=0.1\*vcost;

elsif(vcost>=10000 and vcost<20000) then

discount:=0.25\*vcost;

elsif(vcost>=20000 and vcost<50000) then

discount:=0.4\*vcost;

else

discount:=0.5\*vcost;

end if;

Billamt:=vcost-discount;

dbms\_output.put\_line

(chr(10)||' Selected prod id: '||vpid||chr(10)||

' Prod Name: '||vpname||chr(10)||

' M R P: '||vcost||chr(10)||

' Discount Amount: '||discount||chr(10)||

' Bill Amount: '||billamt

);

end proc\_BillAmount;

Ex:

write a procedure to find the higher number of customers

from the given 2 cities?

Ex:

write a procedure to find the total balance amount in the

given account type?

Ex:

Write a procedure to display the number of products from the

given 2 company names and find out from which company we have less number of products?

Ex:

Write a procedure to display the number of mobile products

from the given 2 company names?

Ex:

Write a procedure to display the total number of customers

opted for given account type?

Ex:

write a procedure to display the higher total funds from the

given 3 types of accounts?

------------------------------------------------------------------------------------------