[‏11/‏30/‏2018 2:55 PM]  Mandal, Devangana:    
1 IF ( a <= b ) THEN   
      dbms\_output.put\_line(a);   
   END IF;

   IF ( b >= a ) THEN   
      dbms\_output.put\_line(a);   
   END IF;   
     
   IF ( a <> b ) THEN   
      dbms\_output.put\_line(b);   
a = 18   
b= 8   
    
[‏11/‏30/‏2018 3:02 PM]  Mandal, Devangana:    
**2**1. Set serveroutput on   
2. declare   
3. cursor cur1 is select \* from emp;   
4. emprec emp%rowtype;   
5. begin   
6. open cur1;   
7. fetch cur1 into emprec;   
8. while(cur1%notfound)   
9. loop   
10. dbms\_output.put\_line(emprec.empno||' '||emprec.ename);   
11. fetch cur1 into emprec;   
12. end loop;   
13. close cur1;   
14. end;   
    
[‏11/‏30/‏2018 3:10 PM]  Mandal, Devangana:    
3IF NOT caller\_cur%ISOPEN   
THEN   
   OPEN caller\_cur;   
END IF;   
Open caller\_cur;   
FETCH caller\_cur INTO caller\_rec;   
    
4 [‏11/‏30/‏2018 3:12 PM]  Mandal, Devangana:    
CREATE OR REPLACE PROCEDURE many\_params (   
mesg1 IN     VARCHAR2,   
mesg2 OUT    VARCHAR2,   
mesg3 IN OUT VARCHAR2) IS   
BEGIN   
  mesg2 := mesg1 || 'Parameter As The OUT';   
  mesg3 := mesg3 || 'Returned';   
END many\_params;   
DECLARE   
 iparm VARCHAR2 (50) := 'This is the IN ';   
 oparm VARCHAR2 (50);   
 ioparm VARCHAR2 (50) := 'And This is the IN OUT ';   
BEGIN   
  many\_params (iparm,  oparm,  ioparm);   
  dbms\_output.put\_line (oparm || '.'|| ioparm);   
END;   
    
5 [‏11/‏30/‏2018 3:14 PM]  Gupta, Swati:    
It will print THIS IS THE IN PARAMETER AS THE OUT RETURNED   
    
[‏11/‏30/‏2018 3:16 PM]  Mandal, Devangana:    
Create or replace procedure getsal (dno IN number,  totsal OUT number) is   
begin   
 select sum(sal) into totsal from emp where deptno = dno;   
end;   
declare   
   vdno number(3) := 20;   
   vtsal number(8);   
begin   
   getsal  (totsal => vtsal, vdno);   
   dbms\_output.put\_line(vdno ||' '|| vtsal);   
end;   
    
    
  
8 [‏11/‏30/‏2018 4:39 PM]  Mandal, Devangana:    
DECLARE   
 myex EXCEPTION;   
 PRAGMA EXCEPTION\_INIT(myex,-20015);   
 n NUMBER := &n;   
BEGIN   
 FOR i IN 1..n LOOP   
  dbms\_output.put.line(i);   
  IF i=n THEN   
   RAISE myex;   
  END IF;   
 END LOOP;   
EXCEPTION   
 WHEN myex THEN   
  dbms\_output.put.line('loop finish');   
END;   
    
7 [‏11/‏30/‏2018 4:51 PM]  Mandal, Devangana:    
DECLARE   
  l\_simple\_integer SIMPLE\_INTEGER := 0;   
BEGIN   
  l\_simple\_integer := NULL;   
END;   
    
8 [‏11/‏30/‏2018 4:53 PM]  Mandal, Devangana:    
v\_n1= v\_n2 > ( 2 \* v\_n3)   
    
9 [‏11/‏30/‏2018 5:00 PM]  Mandal, Devangana:    
BEGIN   
INSERT INTO employees   
(employee\_id, first\_name, last\_name, email,   
hire\_date, job\_id, salary)   
VALUES   
(employees\_seq.NEXTVAL, 'Ruth', 'Cores', 'RCORES',   
sysdate, 'AD\_ASST', 4000);   
END;   
    
10 [‏11/‏30/‏2018 5:08 PM]  Mandal, Devangana:    
declare   
v\_ename emp.ename%type;   
begin   
select ename into v\_ename from emp where sal > 0;   
end;   
    
11 [‏11/‏30/‏2018 5:09 PM]  Mandal, Devangana:

create or replace function f2(p\_p1 number)   
return number PARALLEL\_ENABLE is   
begin   
 return p\_p1\*2;   
end f2;   
    
12[‏11/‏30/‏2018 5:13 PM]  Mandal, Devangana:    
CREATE OR REPLACE TRIGGER emp\_probation   
  BEFORE DELETE ON emp   
   FOR EACH ROW   
   BEGIN   
 IF (Sysdate - :old.hiredate) < 30 THEN   
RAISE\_APPLICATION\_ERROR(20103,'Emploee' || :old.ename || 'Probation');   
 END IF;   
END;   
    
13 [‏11/‏30/‏2018 5:16 PM]  Mandal, Devangana:    
wrapper utility DBMS\_DDL.CREATE\_WRAPPED DBMS\_DDL.OBFUSCATE      OBFUSCATE.EXE    
 

You were removed from the conversation.