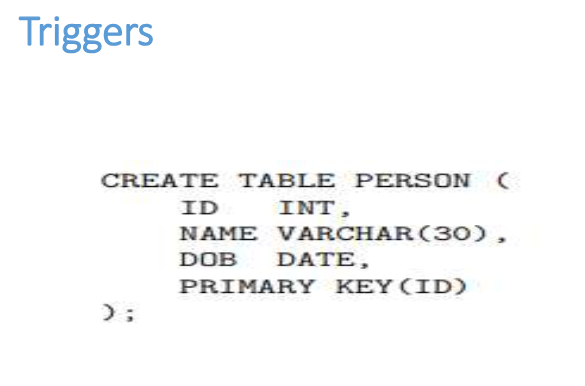
**2022503003**

**week-12**



CREATE TABLE person\_3003(

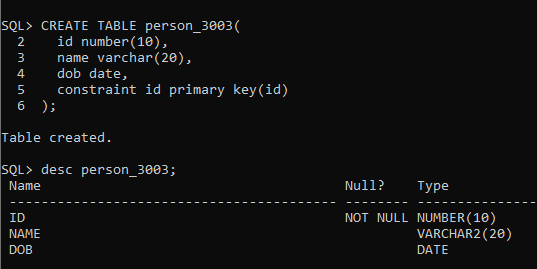
id number(10),

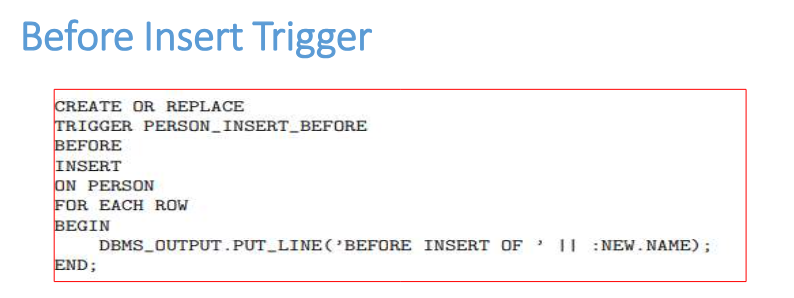
name varchar(20),

dob date,

constraint id primary key(id)

);





CREATE OR REPLACE TRIGGER BEFORE\_INSERT\_PERSON

BEFORE INSERT ON person\_3003

for each row

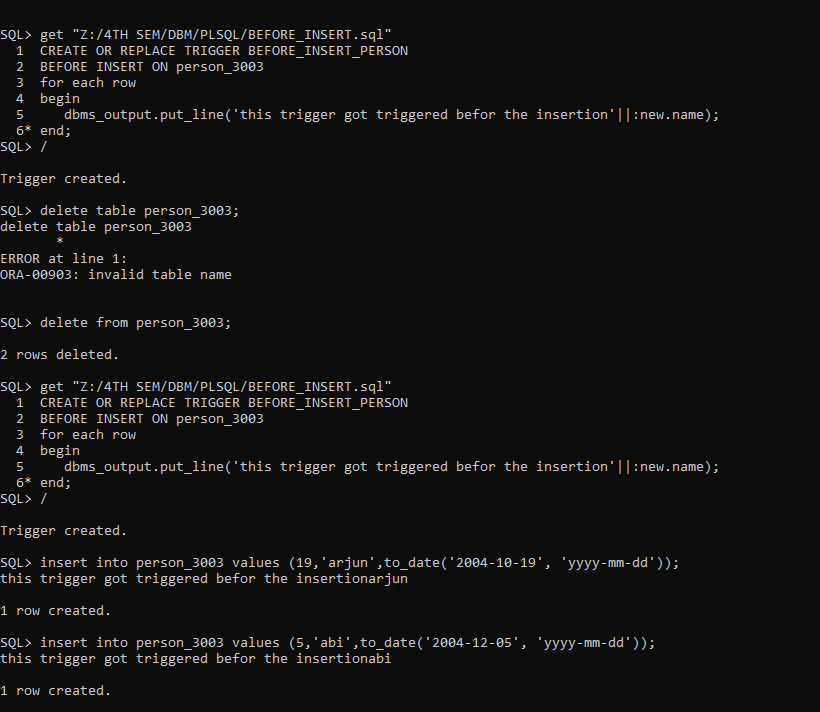
begin

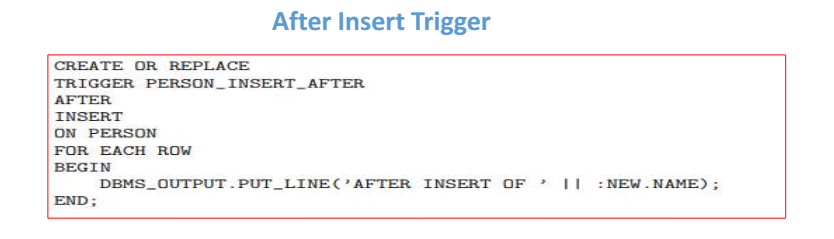
dbms\_output.put\_line('this trigger got triggered befor the insertion'||:new.name);

end;

**THEN**

/





CREATE OR REPLACE TRIGGER AFTER\_INSERT\_PERSON

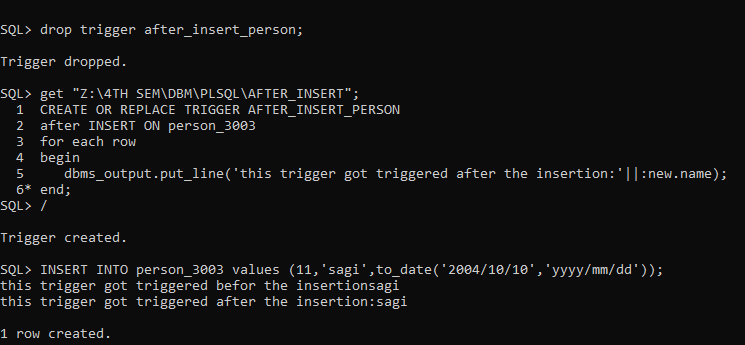
after INSERT ON person\_3003

for each row

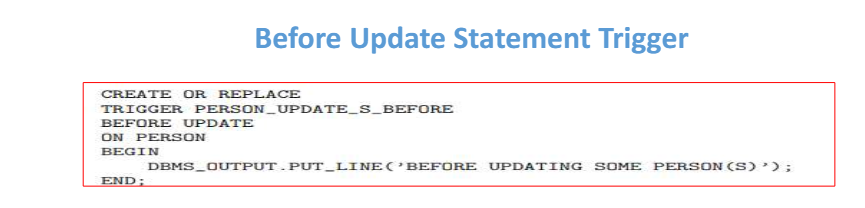
begin

dbms\_output.put\_line('this trigger got triggered after the insertion:'||:new.name);

end;



INSERT INTO person\_3003 values (11,'sagi',to\_date('2004/10/10','yyyy/mm/dd'));



CREATE OR REPLACE TRIGGER BEFORE\_UPDATE

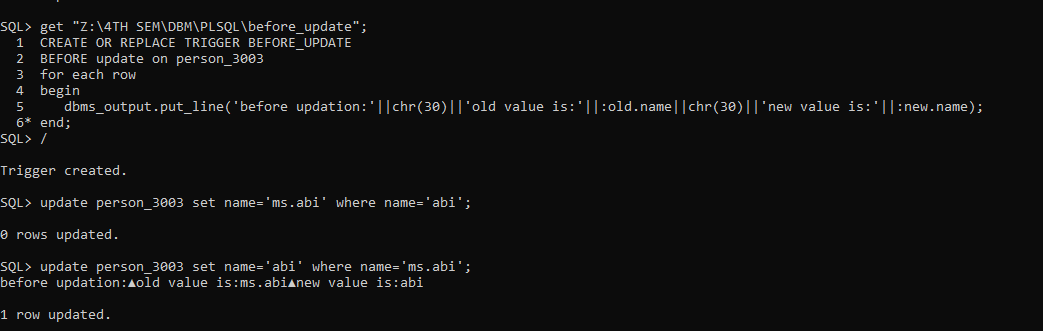
BEFORE update on person\_3003

for each row

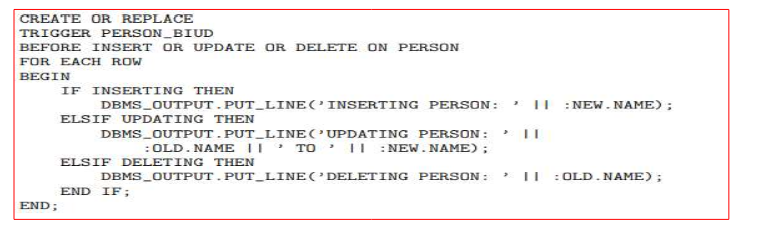
begin

dbms\_output.put\_line('before updation:'||chr(30)||'old value is:'||:old.name||chr(30)||'new value is:'||:new.name);

end;



update person\_3003 set name='abi' where name='ms.abi';



CREATE OR REPLACE TRIGGER INSERT\_UPDATE\_DEL

BEFORE INSERT OR UPDATE OR DELETE on PERSON\_3003

FOR EACH ROW

BEGIN

IF INSERTING THEN

dbms\_output.put\_line('this trigger got triggered before the insertion:'||:new.name);

ELSIF UPDATING THEN

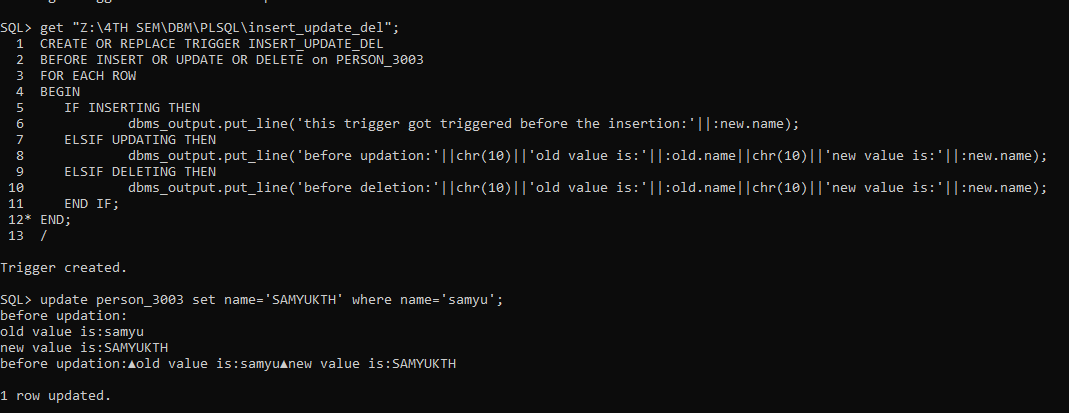
dbms\_output.put\_line('before updation:'||chr(10)||'old value is:'||:old.name||chr(10)||'new value is:'||:new.name);

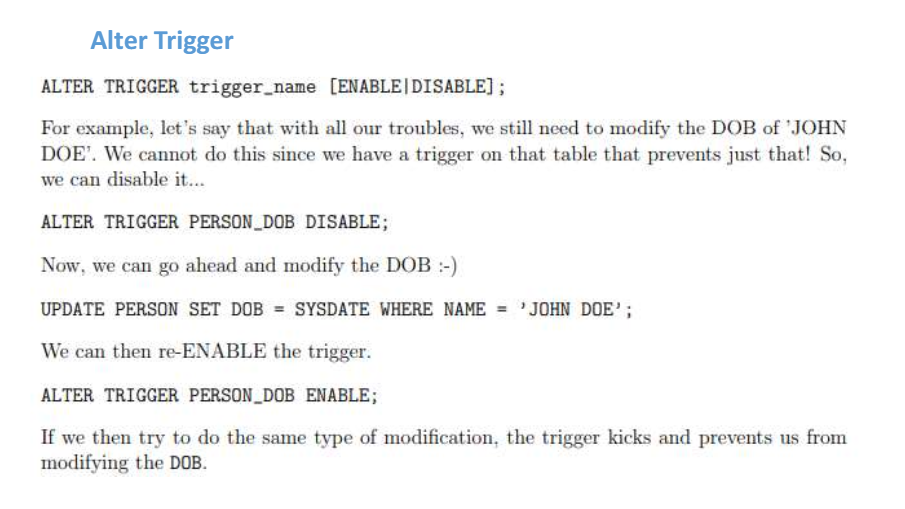
ELSIF DELETING THEN

dbms\_output.put\_line('before deletion:'||chr(10)||'old value is:'||:old.name||chr(10)||'new value is:'||:new.name);

END IF;

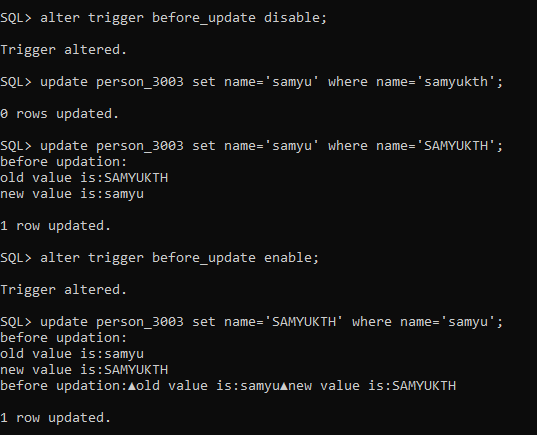
END;

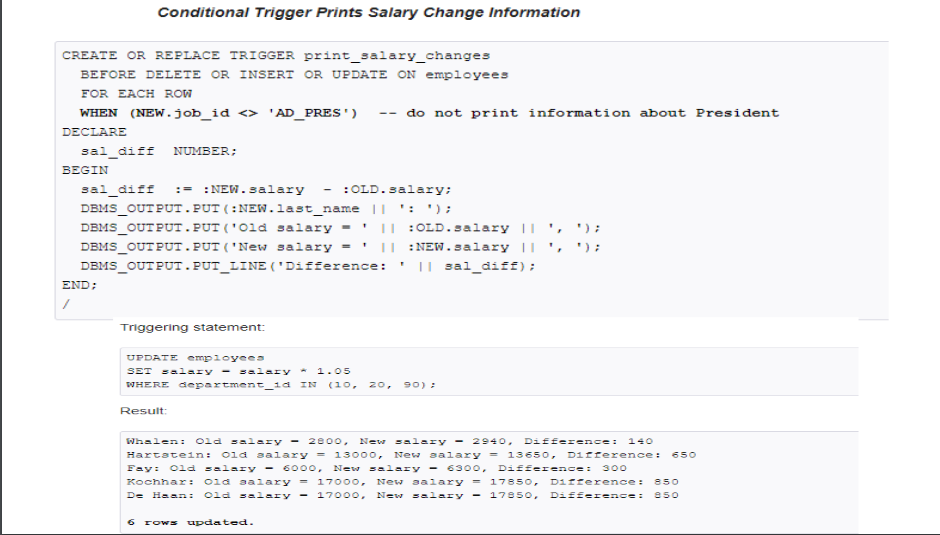




alter trigger before\_update disable;

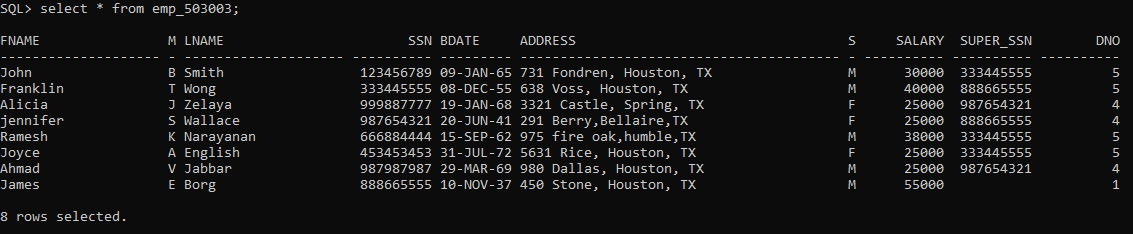
alter trigger before\_update enable;





‘ EMP\_503003’

select \* from emp\_503003;



create or replace trigger salary

after insert or update or delete on emp\_503003

for each row

when (old.salary>26000)

declare

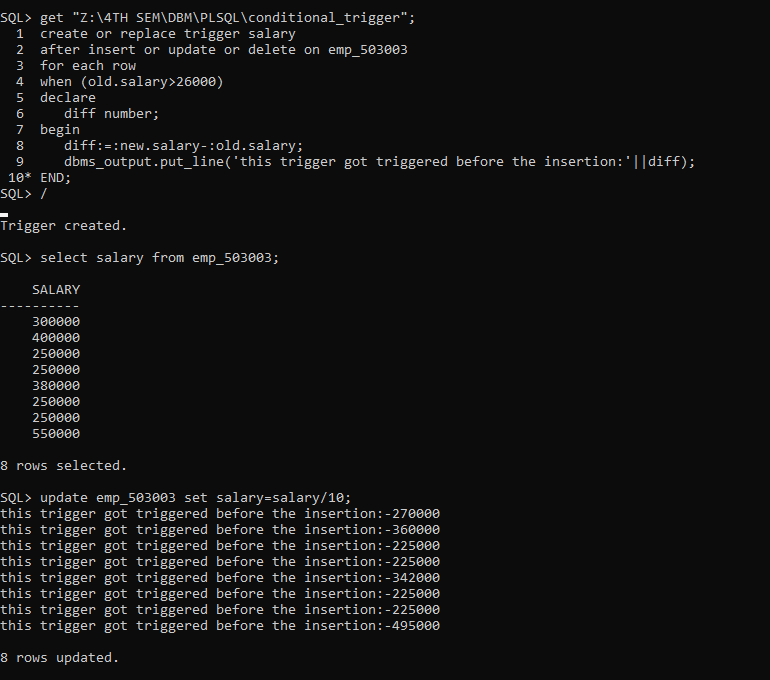
diff number;

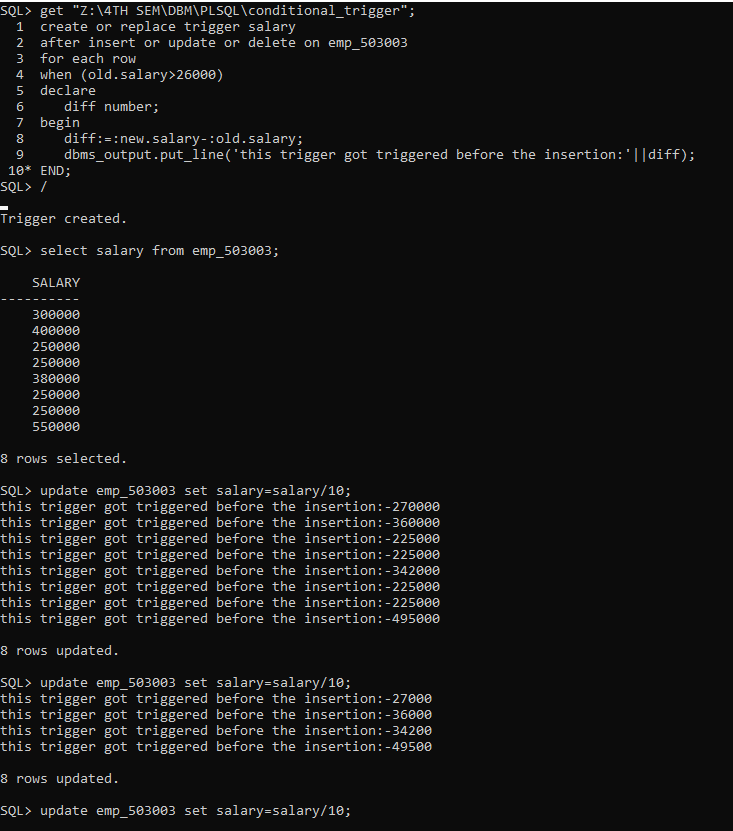
begin

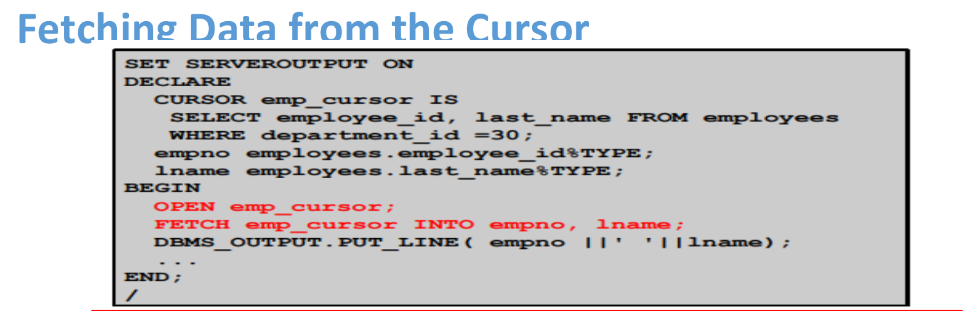
diff:=:new.salary-:old.salary;

dbms\_output.put\_line('this trigger got triggered before the insertion:'||diff);

END;







DECLARE

CURSOR emp\_cursor

IS SELECT ssn,fname,lname FROM emp\_503003

WHERE dno=5;

emp\_ssn emp\_503003.ssn%TYPE;

emp\_fname emp\_503003.fname%type;

emp\_lname emp\_503003.lname%type;

BEGIN

OPEN emp\_cursor;

LOOP

FETCH emp\_cursor INTO emp\_ssn,emp\_fname,emp\_lname;

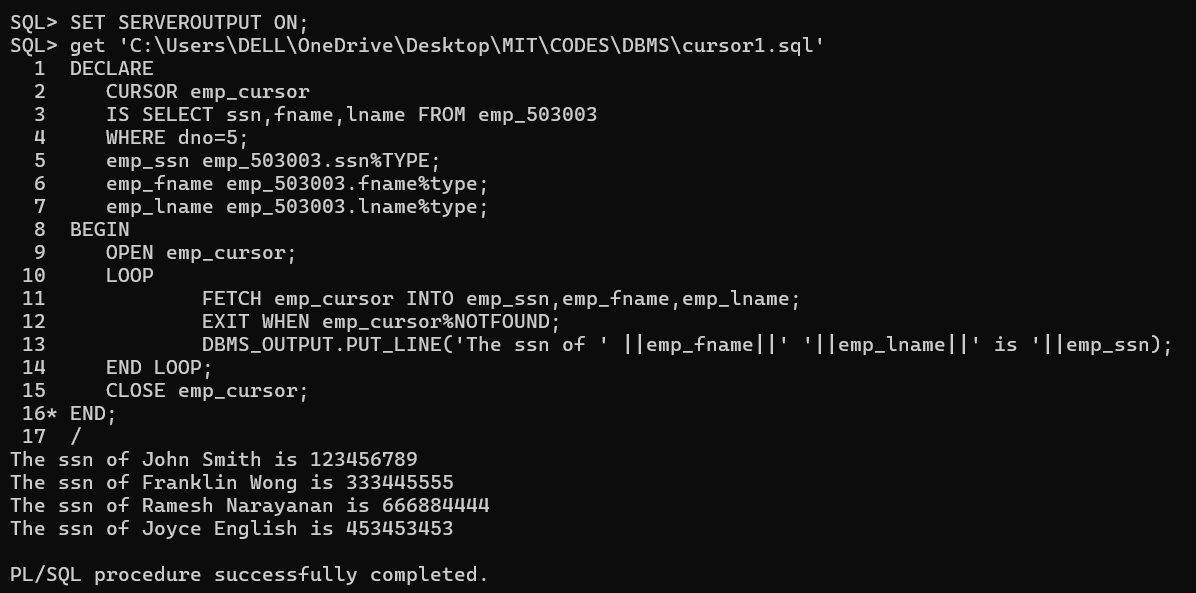
EXIT WHEN emp\_cursor%NOTFOUND;

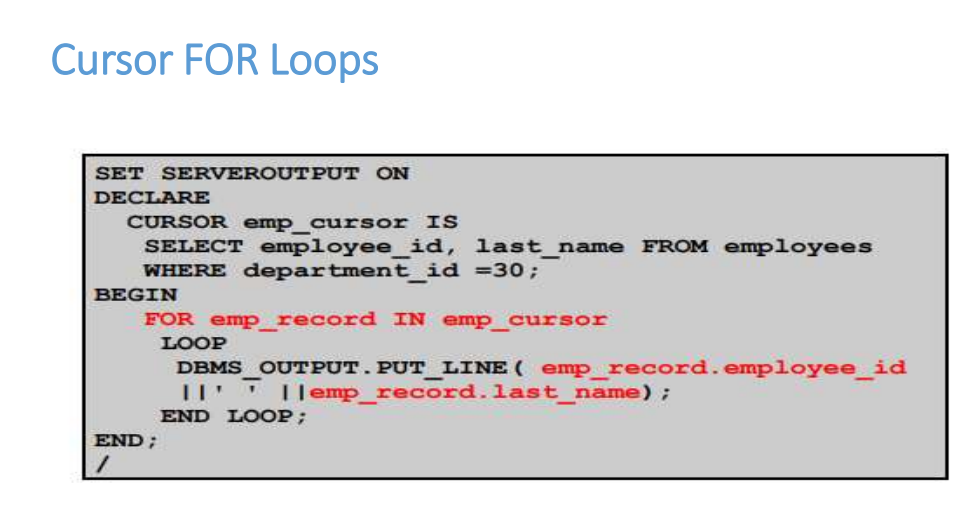
DBMS\_OUTPUT.PUT\_LINE('The ssn of ' ||emp\_fname||' '||emp\_lname||' is '||emp\_ssn);

END LOOP;

CLOSE emp\_cursor;

END;





DECLARE

CURSOR emp\_cursor\_1 IS SELECT

DNO,LNAME,FNAME,SSN FROM emp\_503003

WHERE dno=4;

BEGIN

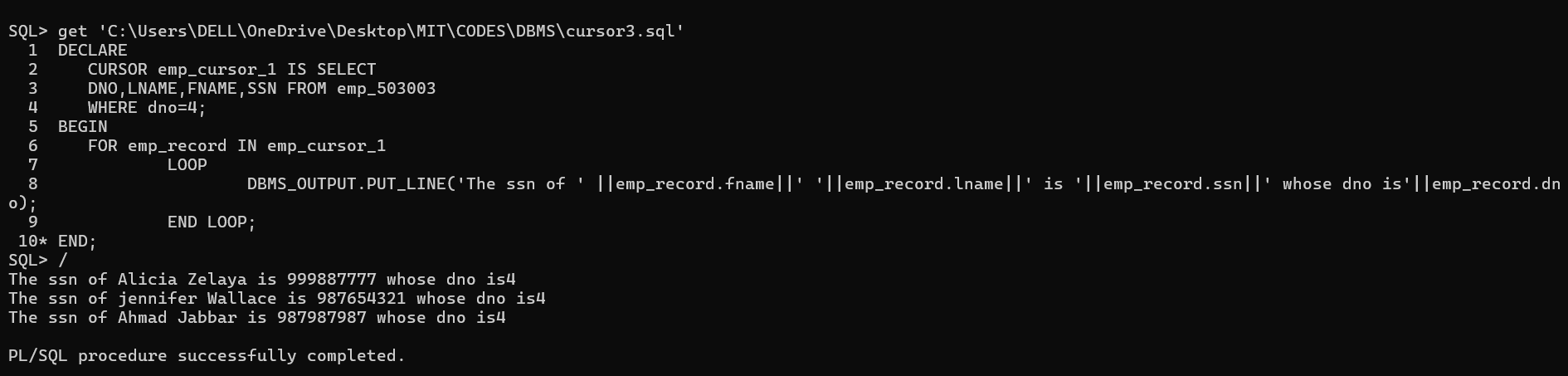
FOR emp\_record IN emp\_cursor\_1

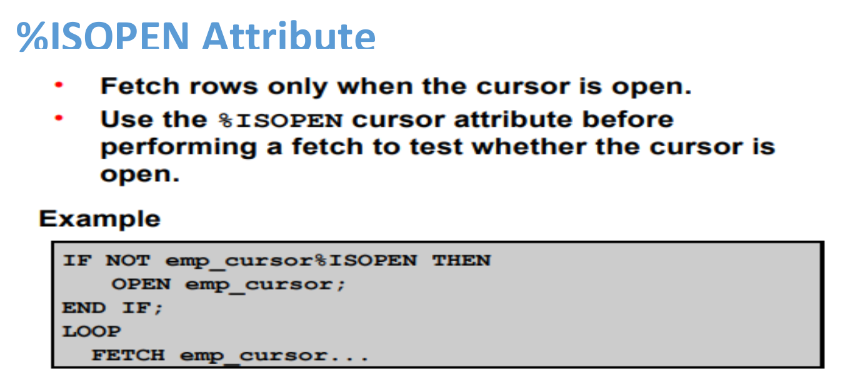
LOOP

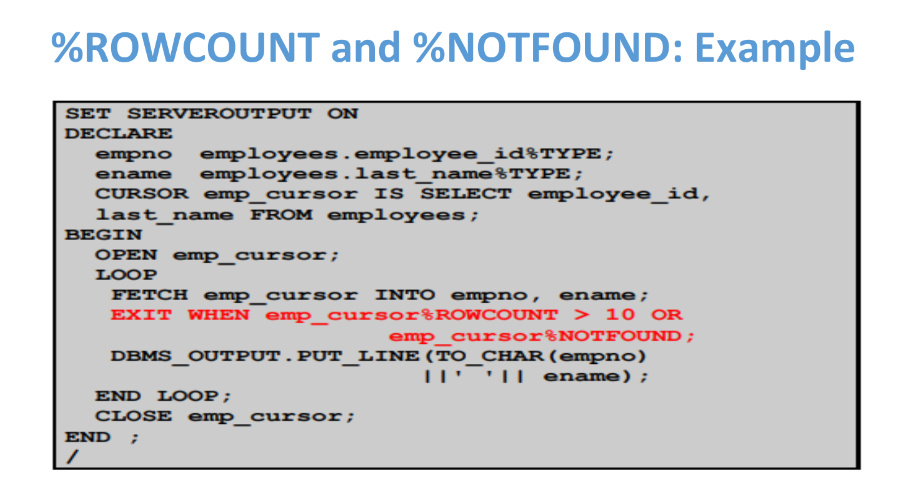
DBMS\_OUTPUT.PUT\_LINE('The ssn of ' ||emp\_record.fname||' '||emp\_record.lname||' is '||emp\_record.ssn||' whose dno is'||emp\_record.dno);

END LOOP;

END;







DECLARE

CURSOR emp\_cursor

IS SELECT ssn,fname,lname FROM emp\_503003

WHERE dno=5;

emp\_ssn emp\_503003.ssn%TYPE;

emp\_fname emp\_503003.fname%type;

emp\_lname emp\_503003.lname%type;

BEGIN

IF NOT emp\_cursor%ISOPEN THEN

OPEN emp\_cursor;

END IF;

LOOP

FETCH emp\_cursor INTO emp\_ssn,emp\_fname,emp\_lname;

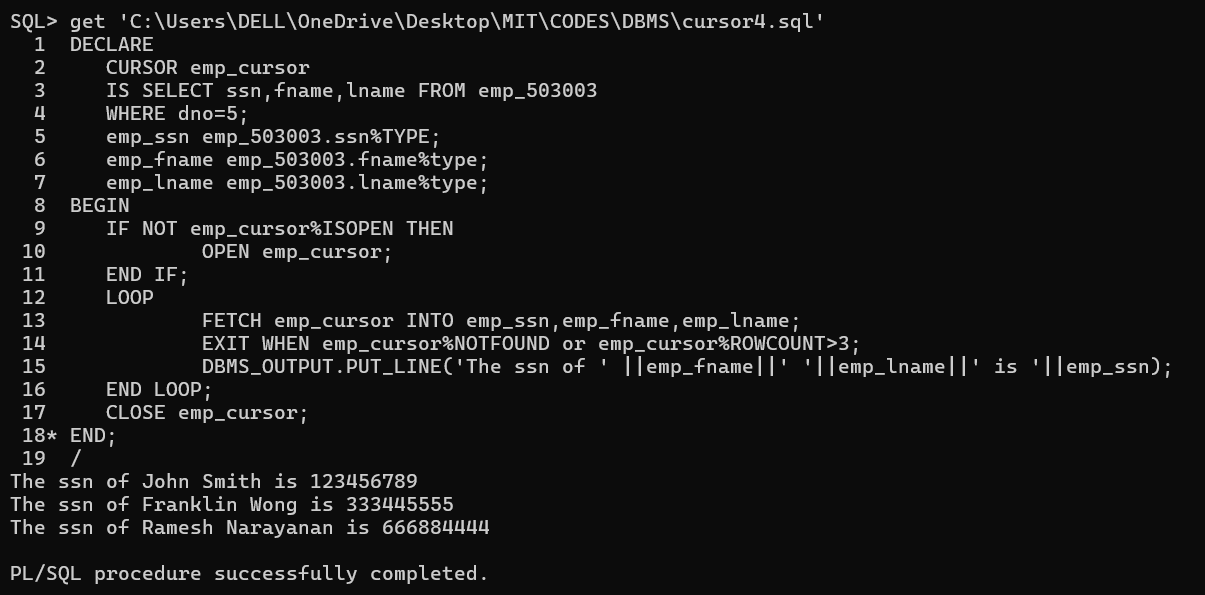
EXIT WHEN emp\_cursor%NOTFOUND or emp\_cursor%ROWCOUNT>3;

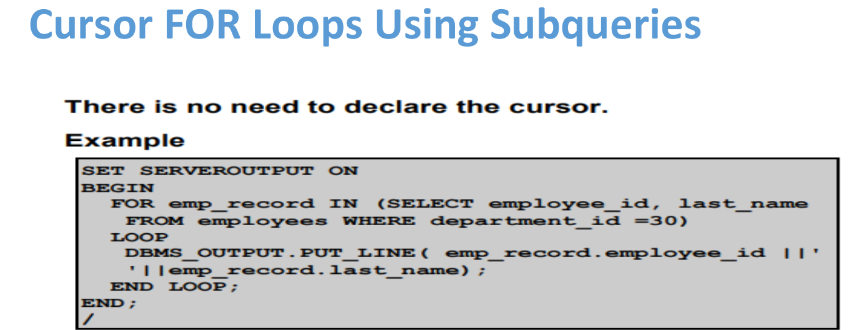
DBMS\_OUTPUT.PUT\_LINE('The ssn of ' ||emp\_fname||' '||emp\_lname||' is '||emp\_ssn);

END LOOP;

CLOSE emp\_cursor;

END;





DECLARE

CURSOR emp\_cursor\_1 IS SELECT

DNO,LNAME,FNAME,SSN FROM emp\_503003

WHERE dno=4;

BEGIN

FOR emp\_record IN (SELECT

DNO,LNAME,FNAME,SSN FROM emp\_503003

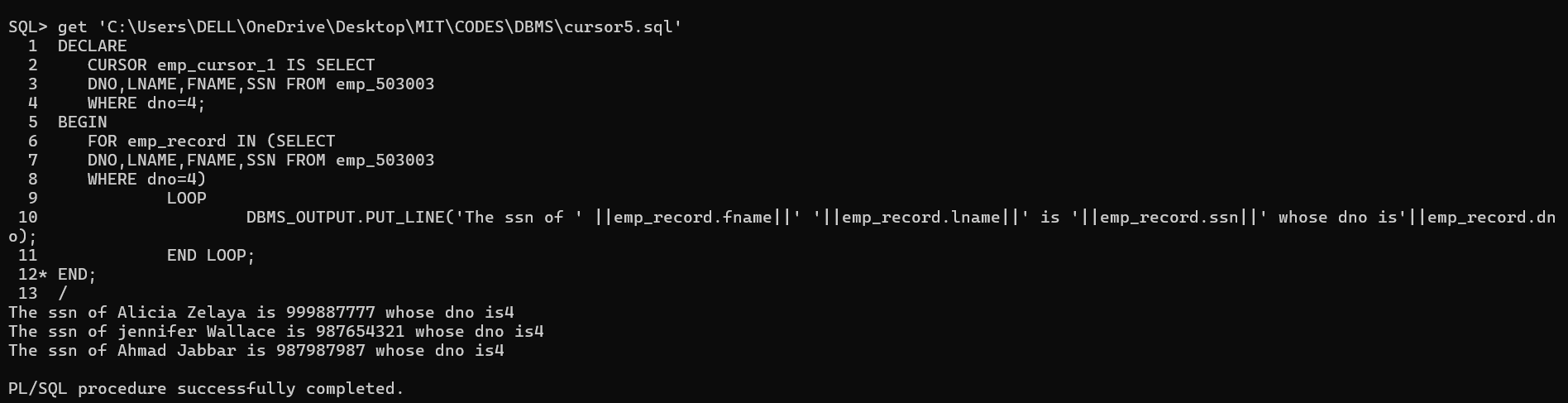
WHERE dno=4)

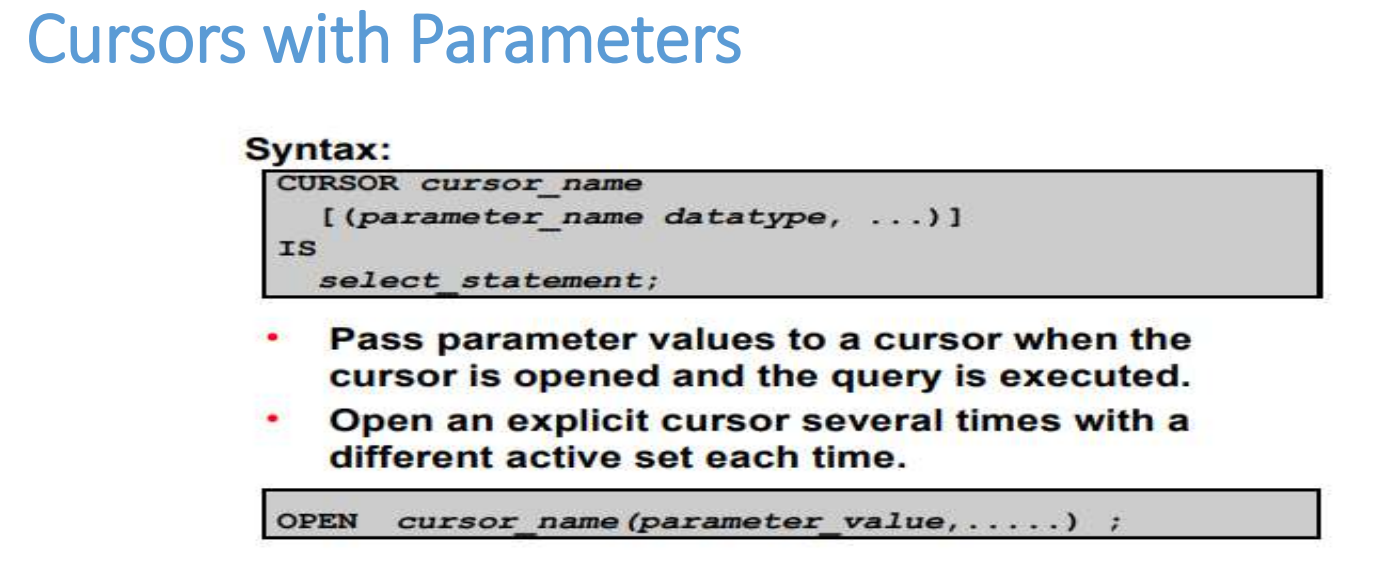
LOOP

DBMS\_OUTPUT.PUT\_LINE('The ssn of ' ||emp\_record.fname||' '||emp\_record.lname||' is '||emp\_record.ssn||' whose dno is'||emp\_record.dno);

END LOOP;

END;





DECLARE

CURSOR emp\_cursor (dept\_id NUMBER) IS

SELECT ssn, fname, lname

FROM emp\_503003

WHERE dno = dept\_id;

emp\_ssn emp\_503003.ssn%TYPE;

emp\_fname emp\_503003.fname%TYPE;

emp\_lname emp\_503003.lname%TYPE;

emp\_dept\_id NUMBER;

BEGIN

emp\_dept\_id:= &emp\_dept\_id;

OPEN emp\_cursor(emp\_dept\_id);

LOOP

FETCH emp\_cursor INTO emp\_ssn, emp\_fname, emp\_lname;

EXIT WHEN emp\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('The SSN of ' || emp\_fname || ' ' || emp\_lname || ' is ' || emp\_ssn);

END LOOP;

CLOSE emp\_cursor;

END;

