

```
SQL> CREATE TABLE EMPLOY_7(ENUMBER VARCHAR(5) PRIMARY KEY, SALARY  
NUMBER(6), UPDATE_DATE DATE, NET_SALARY NUMBER(6));
```

Table created.

```
SQL> DESC EMPLOY_7;
```

Name	Null?	Type
ENUMBER	NOT NULL	VARCHAR2(5)
SALARY		NUMBER(6)
UPDATE_DATE		DATE
NET_SALARY		NUMBER(6)

```
SQL> INSERT INTO EMPLOY_7 (ENUMBER, SALARY) VALUES ('E101', 30000);
```

1 row created.

```
SQL> INSERT INTO EMPLOY_7 (ENUMBER, SALARY) VALUES ('E102',40000);
```

1 row created.

```
SQL> INSERT INTO EMPLOY_7 (ENUMBER, SALARY) VALUES ('E103',50000);
```

1 row created.

```
SQL> INSERT INTO EMPLOY_7 (ENUMBER, SALARY) VALUES ('E104',35000);
```

1 row created.

```
SQL> INSERT INTO EMPLOY_7 (ENUMBER, SALARY) VALUES ('E105',84000);
```

1 row created.

```
SQL> INSERT INTO EMPLOY_7 (ENUMBER, SALARY) VALUES ('E106',63000);
```

1 row created.

SQL> SELECT * FROM EMPLOY_7;

ENUMB	SALARY	UPDATE_DA	NET_SALARY

E101	30000		
E102	40000		
E103	50000		
E104	35000		
E105	84000		
E106	63000		

6 rows selected.

incfn.sql

```
CREATE OR REPLACE FUNCTION INC (EID IN VARCHAR, INC_AMT IN NUMBER)  
RETURN NUMBER  
IS  
NEW_SAL NUMBER;  
ENO EMPLOY_7.ENUMBER%TYPE;  
SAL EMPLOY_7.SALARY%TYPE;  
  
BEGIN  
SELECT ENUMBER, SALARY INTO ENO,SAL FROM EMPLOY_7 WHERE ENUMBER=EID;  
NEW_SAL := SAL + INC_AMT;  
  
UPDATE EMPLOY_7 SET NET_SALARY = NEW_SAL, UPDATE_DATE = SYSDATE WHERE  
ENUMBER=EID;  
RETURN NEW_SAL;  
  
EXCEPTION
```

```
WHEN NO_DATA_FOUND
THEN return null;
```

```
END;
```

inccall.sql

```
DECLARE
```

```
  n varchar(5);
```

```
  amt number(6);
```

```
  new_amt number(6);
```

```
BEGIN
```

```
  n:='&emp_no';
```

```
  amt:=&increment_amount;
```

```
  new_amt:=INC(n,amt);
```

```
  if(new_amt is not null) then
```

```
    dbms_output.put_line('New incremented salary :'||new_amt);
```

```
  else
```

```
    dbms_output.put_line('Employee not found:');
```

```
  end if;
```

```
END;
```

OUTPUT

```
SQL> @incfn
```

```
23 /
```

Function created.

```
SQL> @inccall
```

```
16 /
```

```
Enter value for emp_no: E101
```

```
old 6: n:='&emp_no';
```

```
new 6:      n:='E101';
```

```
Enter value for increment_amount: 20000
```

```
old 7: amt:=&increment_amount;
new 7:      amt:=20000;
New incremented salary :50000
```

PL/SQL procedure successfully completed.

Commit complete.

SQL> @inccall

16 /

Enter value for emp_no: E102

```
old 6: n:='&emp_no';
new 6:      n:='E102';
Enter value for increment_amount: 15000
old 7: amt:=&increment_amount;
new 7:      amt:=15000;
New incremented salary :55000
```

PL/SQL procedure successfully completed.

Commit complete.

SQL> /

Enter value for emp_no: E103

```
old 6: n:='&emp_no';
new 6:      n:='E103';
Enter value for increment_amount: 4000
old 7: amt:=&increment_amount;
new 7:      amt:=4000;
New incremented salary :54000
```

PL/SQL procedure successfully completed.

Commit complete.

SQL> /

Enter value for emp_no: E104

```
old 6: n:='&emp_no';
new 6:      n:='E104';
Enter value for increment_amount: 18000
old 7: amt:=&increment_amount;
new 7:      amt:=18000;
```

New incremented salary :53000

PL/SQL procedure successfully completed.

Commit complete.

SQL> /

Enter value for emp_no: E105

old 6: n:='&emp_no';

new 6: n:='E105';

Enter value for increment_amount: 35000

old 7: amt:='&increment_amount';

new 7: amt:=35000;

New incremented salary :119000

PL/SQL procedure successfully completed.

Commit complete.

SQL> /

Enter value for emp_no: E106

old 6: n:='&emp_no';

new 6: n:='E106';

Enter value for increment_amount: 12000

old 7: amt:='&increment_amount';

new 7: amt:=12000;

New incremented salary :75000

PL/SQL procedure successfully completed.

Commit complete.

SQL> /

Enter value for emp_no: E110

old 6: n:='&emp_no';

new 6: n:='E110';

Enter value for increment_amount: 2000

old 7: amt:='&increment_amount';

new 7: amt:=2000;

Employee not found:

PL/SQL procedure successfully completed.

Commit complete.

SQL> SELECT * FROM EMPLOY_7;

ENUMB	SALARY	UPDATE_DA	NET_SALARY
E101	30000	31-OCT-23	50000
E102	40000	31-OCT-23	55000
E103	50000	31-OCT-23	54000
E104	35000	31-OCT-23	53000
E105	84000	31-OCT-23	119000
E106	63000	31-OCT-23	75000

6 rows selected.