Assignment 11

Name – Tanmay Garg

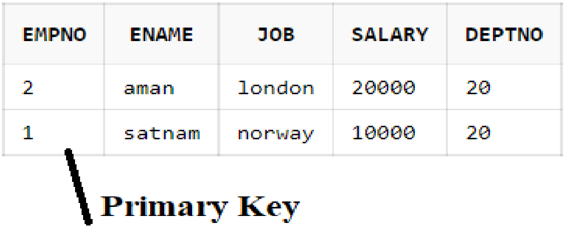
Batch – 3ENC1

Roll No. – 101915001

PL/SQL Lab Assignment-3

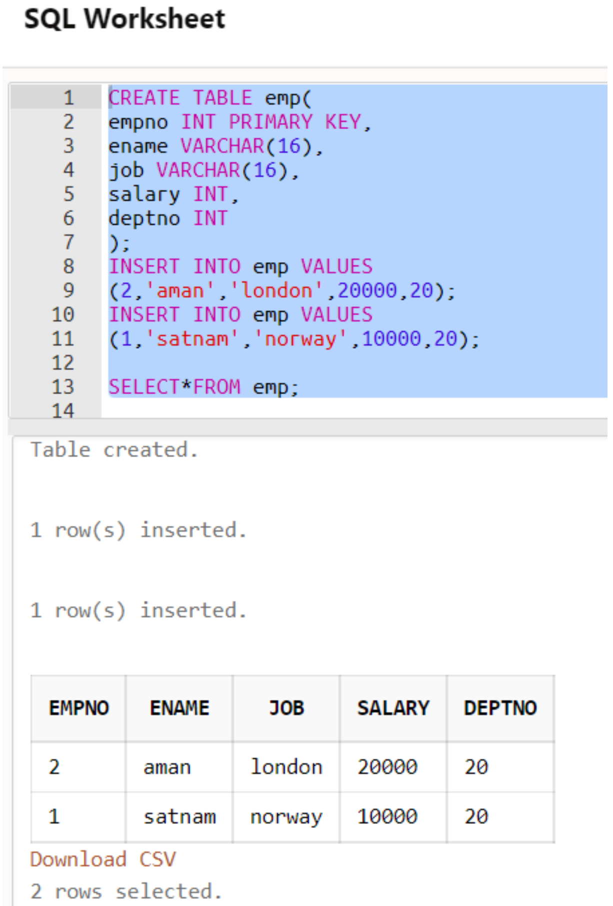
Consider the following table with the following values:

EMP (empno, ename, job, sal, deptno)



EMPNO	ENAME	JOB	SALARY	DEPTNO
2	aman	london	20000	20
1	satnam	norway	10000	20

Primary Key



SQL Worksheet

```
CREATE TABLE emp(
empno INT PRIMARY KEY,
ename VARCHAR(16),
job VARCHAR(16),
salary INT,
deptno INT
);
INSERT INTO emp VALUES
(2,'aman','london',20000,20);
INSERT INTO emp VALUES
(1,'satnam','norway',10000,20);

SELECT*FROM emp;
```

```
Table created.


1 row(s) inserted.


1 row(s) inserted.
```

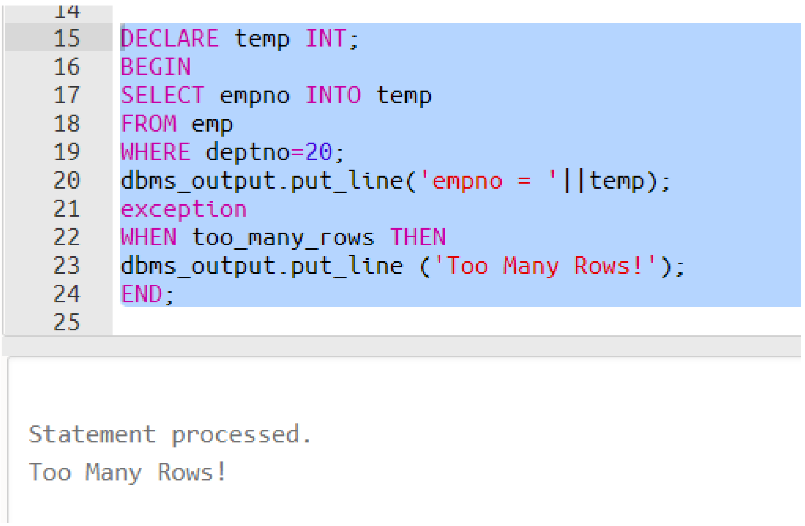
EMPNO	ENAME	JOB	SALARY	DEPTNO
2	aman	london	20000	20
1	satnam	norway	10000	20

Download CSV

2 rows selected.

1. Write a PL/SQL program to demonstrate following exceptions:

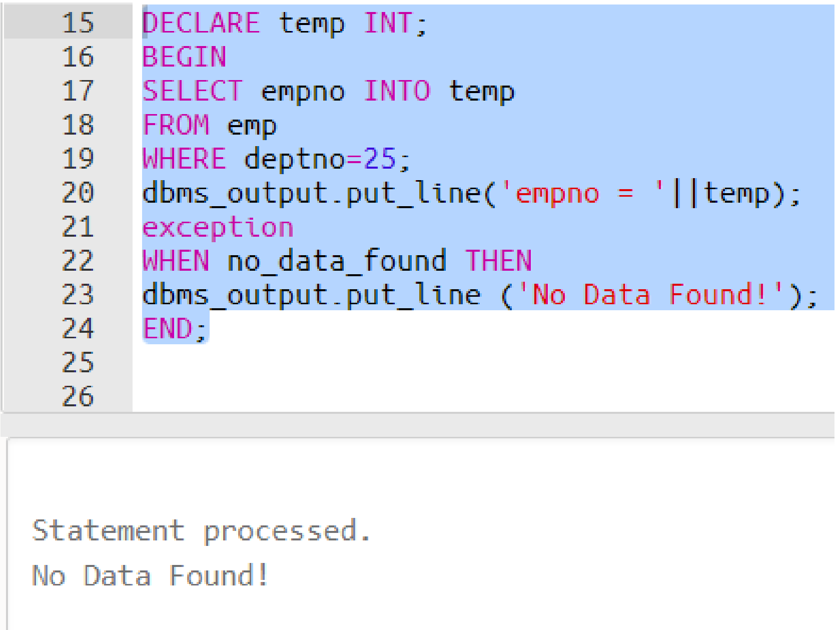
- Too Many Rows



```
DECLARE temp INT;
BEGIN
SELECT empno INTO temp
FROM emp
WHERE deptno=20;
dbms_output.put_line('empno = '||temp);
exception
WHEN too_many_rows THEN
dbms_output.put_line ('Too Many Rows!');
END;
```

```
Statement processed.
Too Many Rows!
```

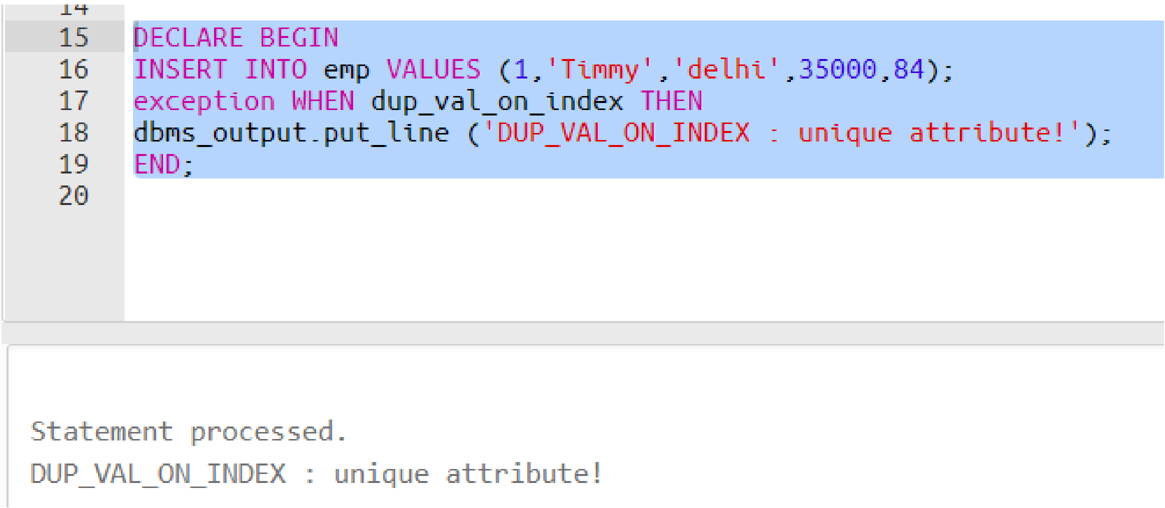
- No Data Found



```
DECLARE temp INT;
BEGIN
SELECT empno INTO temp
FROM emp
WHERE deptno=25;
dbms_output.put_line('empno = '||temp);
exception
WHEN no_data_found THEN
dbms_output.put_line ('No Data Found!');
END;
```

```
Statement processed.
No Data Found!
```

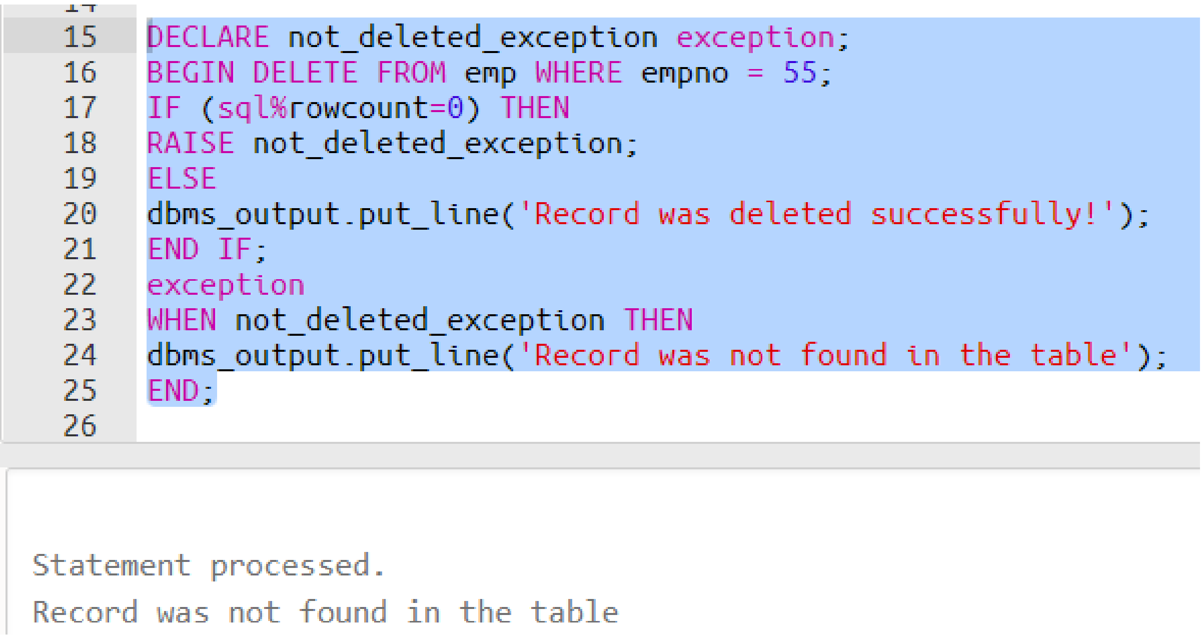
2. Try inserting duplicates in Empno and catch DUP_VAL_ON_INDEX



```
DECLARE BEGIN
INSERT INTO emp VALUES (1,'Timmy','delhi',35000,84);
exception WHEN dup_val_on_index THEN
dbms_output.put_line ('DUP_VAL_ON_INDEX : unique attribute!');
END;
```

```
Statement processed.
DUP_VAL_ON_INDEX : unique attribute!
```

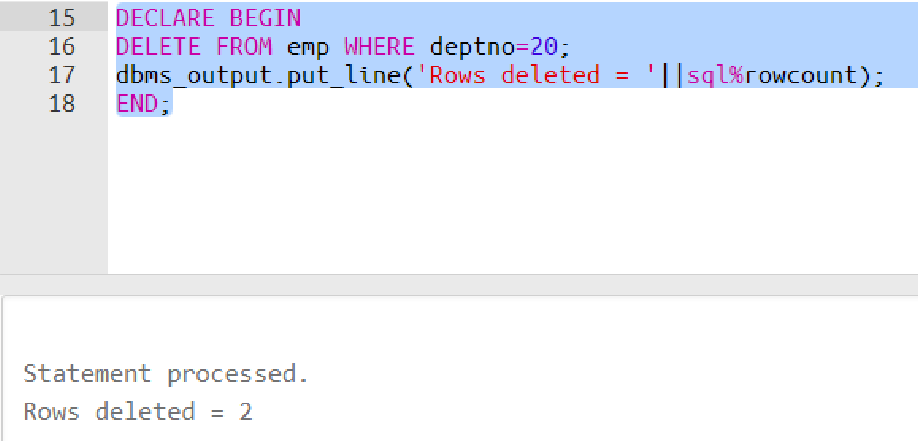
3. Write a PL/SQL code to display a message to check whether the record is deleted ornot. Also, raise a valid user defined exception if the record does not exist; otherwise display record deleted successfully upon checking the deletion of the desired row.



```
DECLARE not_deleted_exception exception;
BEGIN DELETE FROM emp WHERE empno = 55;
IF (sql%rowcount=0) THEN
RAISE not_deleted_exception;
ELSE
dbms_output.put_line('Record was deleted successfully!');
END IF;
exception
WHEN not_deleted_exception THEN
dbms_output.put_line('Record was not found in the table');
END;
```

```
Statement processed.
Record was not found in the table
```

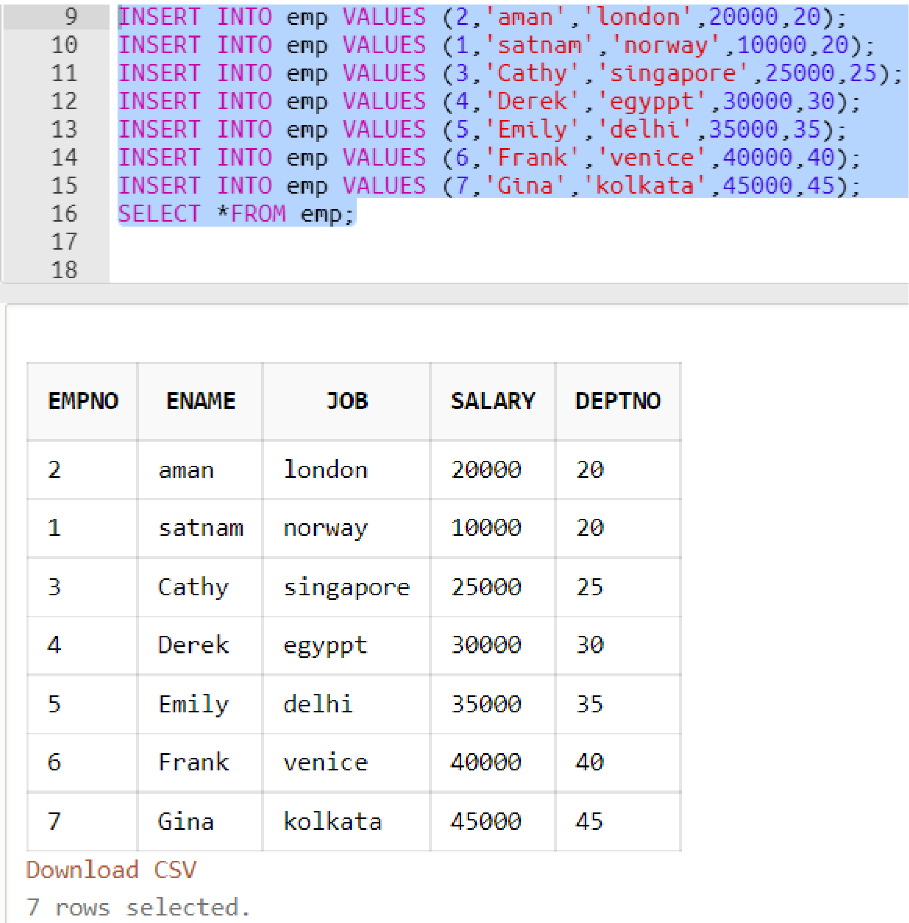
4. Write a PL/SQL code to display a message to provide the information about thenumber of records deleted by the delete statement issued in a PL/SQL block.



```
DECLARE BEGIN
DELETE FROM emp WHERE deptno=20;
dbms_output.put_line('Rows deleted = '||sql%rowcount);
END;
```

```
Statement processed.
Rows deleted = 2
```

5. Write a PL/SQL code to display the empno, ename, job of employees of departmentnumber 10 for EMP table.



```
INSERT INTO emp VALUES (2,'aman','london',20000,20);
INSERT INTO emp VALUES (1,'satnam','norway',10000,20);
INSERT INTO emp VALUES (3,'Cathy','singapore',25000,25);
INSERT INTO emp VALUES (4,'Derek','egyppt',30000,30);
INSERT INTO emp VALUES (5,'Emily','delhi',35000,35);
INSERT INTO emp VALUES (6,'Frank','venice',40000,40);
INSERT INTO emp VALUES (7,'Gina','kolkata',45000,45);
SELECT *FROM emp;


```

EMPNO	ENAME	JOB	SALARY	DEPTNO
2	aman	london	20000	20
1	satnam	norway	10000	20
3	Cathy	singapore	25000	25
4	Derek	egyppt	30000	30
5	Emily	delhi	35000	35
6	Frank	venice	40000	40
7	Gina	kolkata	45000	45

Download CSV

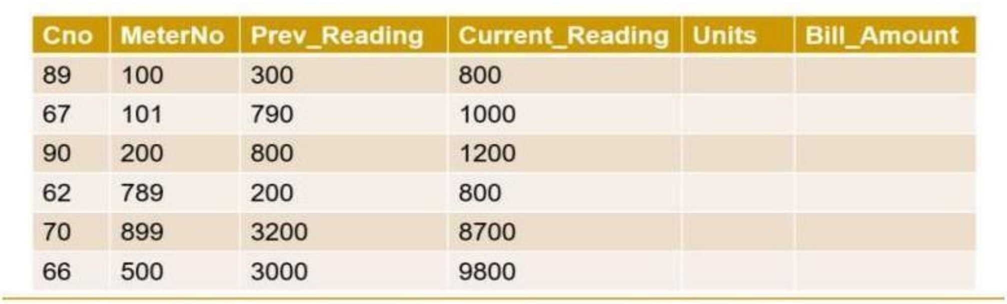
7 rows selected.

6. Write a PL/SQL code to display the employee number and name of top 5 highest paidemployees.

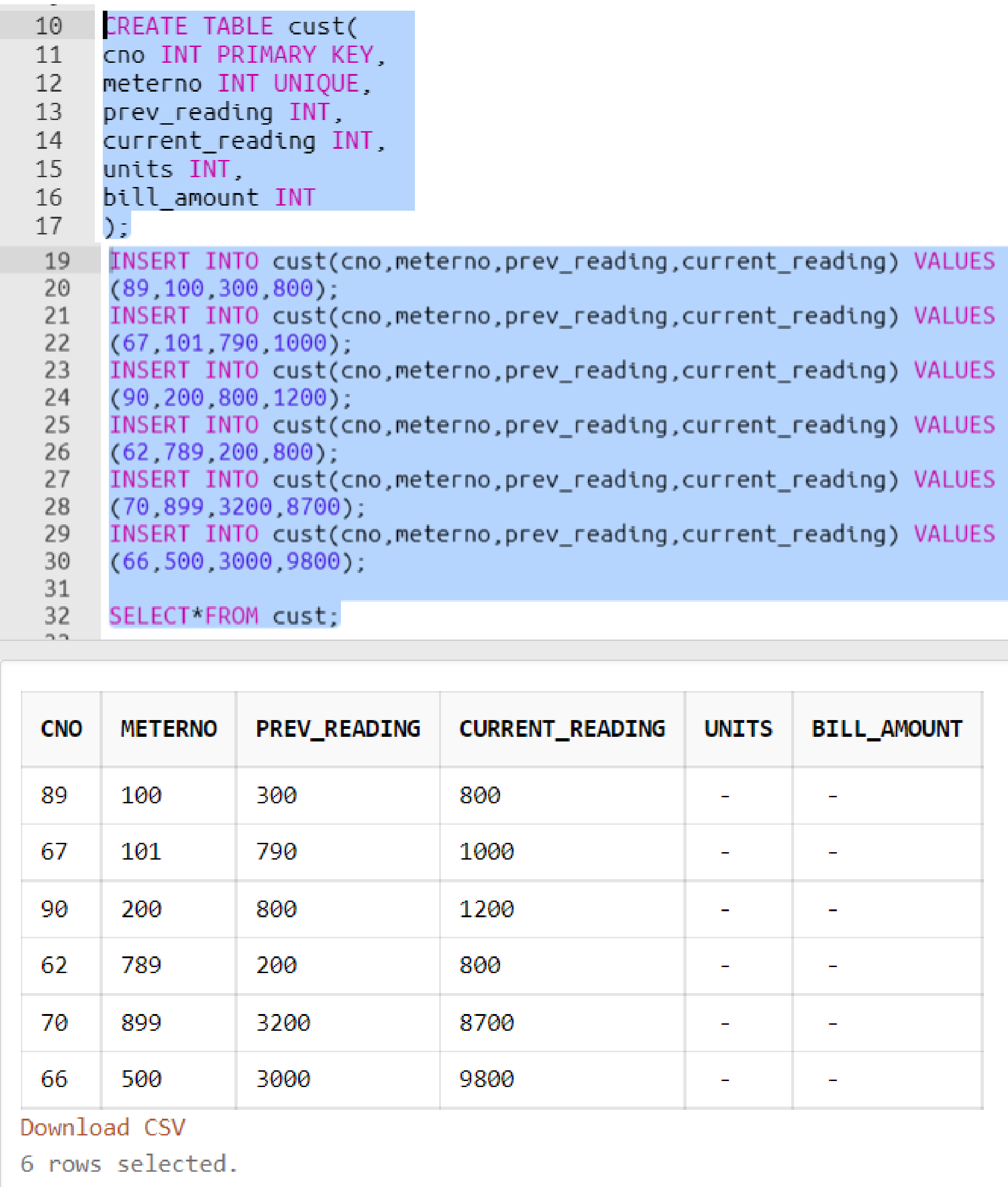
Let us consider CUST table with columns Cno PK, MeterNo Unique, Prev_Reading,Current_Reading, Units, Bill_Amount Units=Current_Reading-Prev_Reading

• Bill will be charged according to following rate:

• First 100 Units 0.5 Rs per unit and after 0.75 Rs for every unit consume



Cno	MeterNo	Prev_Reading	Current_Reading	Units	Bill_Amount
89	100	300	800		
67	101	790	1000		
90	200	800	1200		
62	789	200	800		
70	899	3200	8700		
66	500	3000	9800		

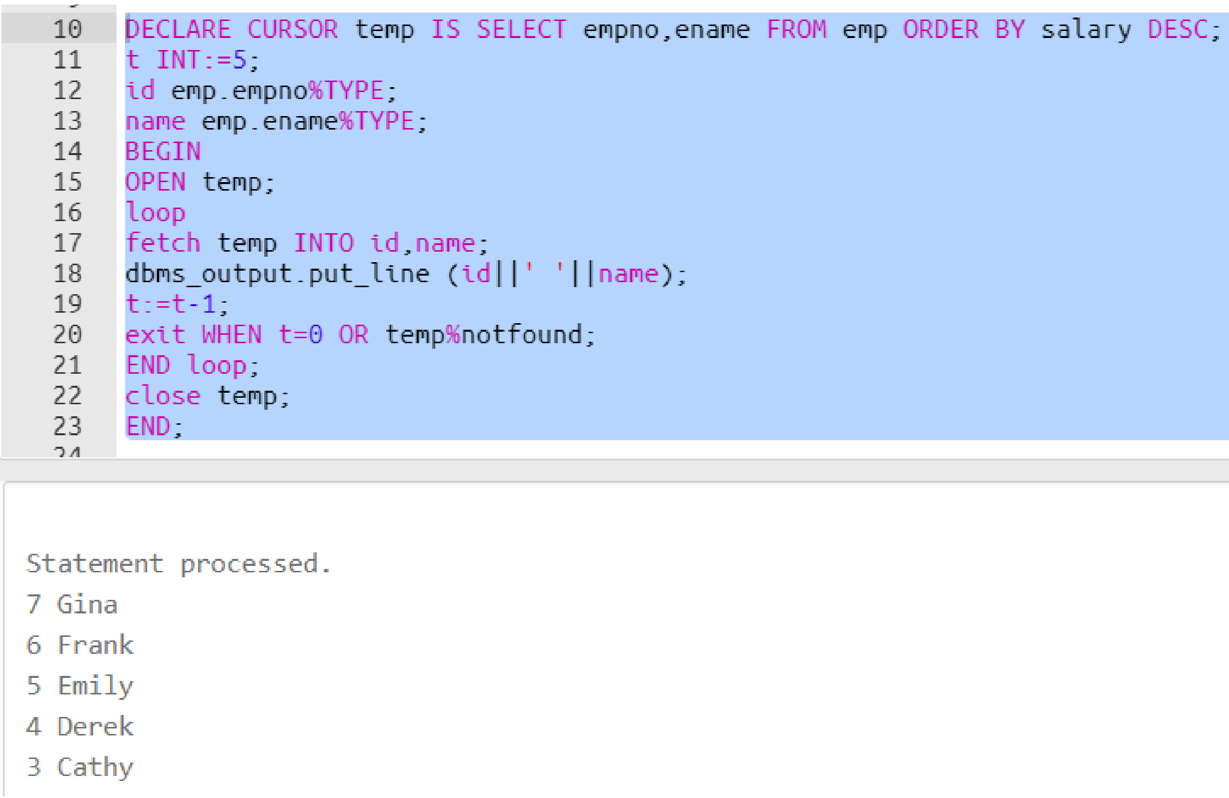


```
CREATE TABLE cust(
cno INT PRIMARY KEY,
meterno INT UNIQUE,
prev_reading INT,
current_reading INT,
units INT,
bill_amount INT
);
INSERT INTO cust(cno,meterno,prev_reading,current_reading) VALUES
(89,100,300,800);
INSERT INTO cust(cno,meterno,prev_reading,current_reading) VALUES
(67,101,790,1000);
INSERT INTO cust(cno,meterno,prev_reading,current_reading) VALUES
(90,200,800,1200);
INSERT INTO cust(cno,meterno,prev_reading,current_reading) VALUES
(62,789,200,800);
INSERT INTO cust(cno,meterno,prev_reading,current_reading) VALUES
(70,899,3200,8700);
INSERT INTO cust(cno,meterno,prev_reading,current_reading) VALUES
(66,500,3000,9800);

SELECT*FROM cust;
```

CNO	METERNO	PREV_READING	CURRENT_READING	UNITS	BILL_AMOUNT
89	100	300	800	-	-
67	101	790	1000	-	-
90	200	800	1200	-	-
62	789	200	800	-	-
70	899	3200	8700	-	-
66	500	3000	9800	-	-

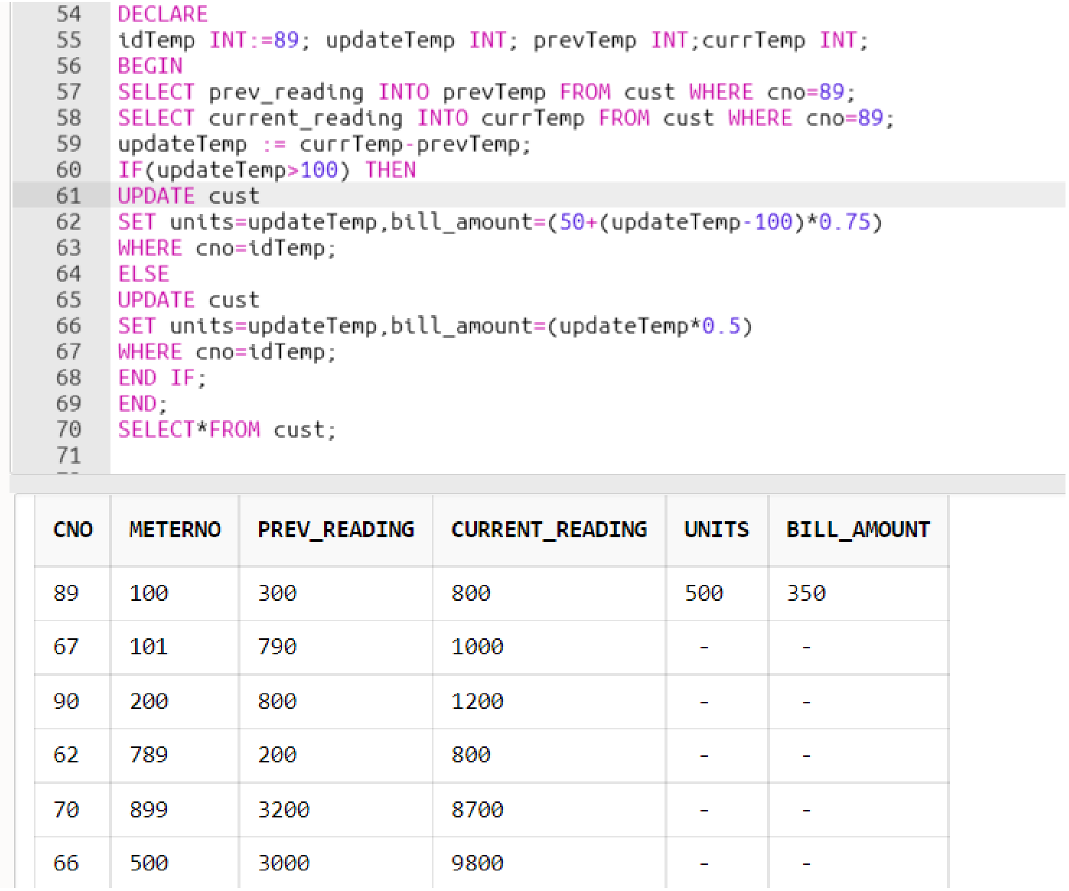
```
6 rows selected.
```



```
DECLARE CURSOR temp IS SELECT empno,ename FROM emp ORDER BY salary DESC;
t INT:=5;
id emp.empno%TYPE;
name emp.ename%TYPE;
BEGIN
OPEN temp;
loop
fetch temp INTO id,name;
dbms_output.put_line (id||' '||name);
t:=t-1;
exit WHEN t=0 OR temp%notfound;
END loop;
close temp;
END;
```

```
Statement processed.
7 Gina
6 Frank
5 Emily
4 Derek
3 Cathy
```

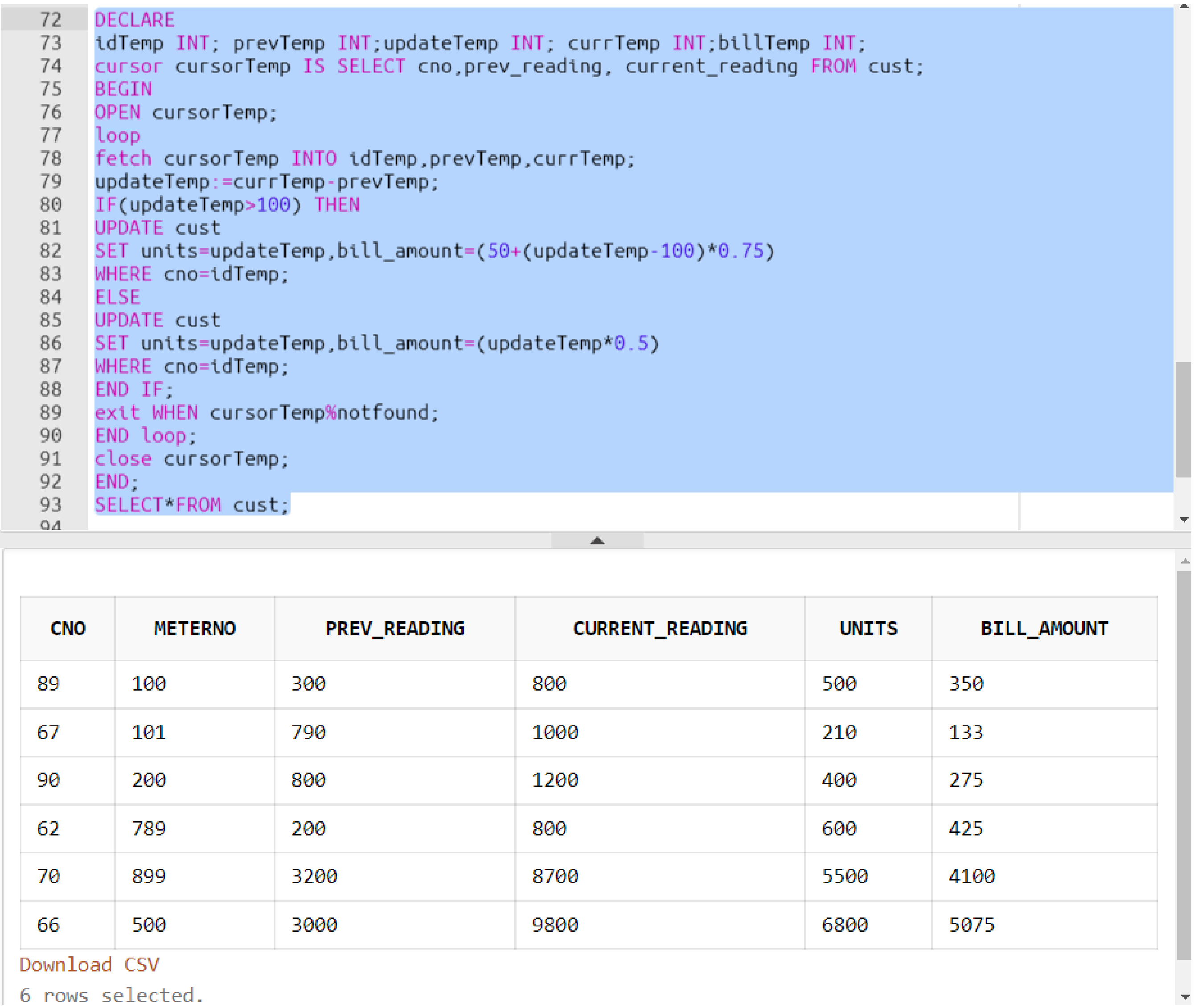
7. Write a PL/SQL code to update units and bill amount of customer number 89 inCUST table



```
DECLARE
idTemp INT:=89; updateTemp INT; prevTemp INT;currTemp INT;
BEGIN
SELECT prev_reading INTO prevTemp FROM cust WHERE cno=89;
SELECT current_reading INTO currTemp FROM cust WHERE cno=89;
updateTemp := currTemp-prevTemp;
IF(updateTemp>100) THEN
UPDATE cust
SET units=updateTemp,bill_amount=(50+(updateTemp-100)*0.75)
WHERE cno=idTemp;
ELSE
UPDATE cust
SET units=updateTemp,bill_amount=(updateTemp*0.5)
WHERE cno=idTemp;
END IF;
END;
SELECT*FROM cust;
```

CNO	METERNO	PREV_READING	CURRENT_READING	UNITS	BILL_AMOUNT
89	100	300	800	500	350
67	101	790	1000	-	-
90	200	800	1200	-	-
62	789	200	800	-	-
70	899	3200	8700	-	-
66	500	3000	9800	-	-

8. Write a PL/SQL code to update units and bill amount of all the customers in CUSTtable



```
DECLARE
idTemp INT; prevTemp INT;updateTemp INT; currTemp INT;billTemp INT;
cursor cursorTemp IS SELECT cno,prev_reading, current_reading FROM cust;
BEGIN
OPEN cursorTemp;
loop
fetch cursorTemp INTO idTemp,prevTemp,currTemp;
updateTemp:=currTemp-prevTemp;
IF(updateTemp>100) THEN
UPDATE cust
SET units=updateTemp,bill_amount=(50+(updateTemp-100)*0.75)
WHERE cno=idTemp;
ELSE
UPDATE cust
SET units=updateTemp,bill_amount=(updateTemp*0.5)
WHERE cno=idTemp;
END IF;
exit WHEN cursorTemp%notfound;
END loop;
close cursorTemp;
END;
SELECT*FROM cust;
```

CNO	METERNO	PREV_READING	CURRENT_READING	UNITS	BILL_AMOUNT
89	100	300	800	500	350
67	101	790	1000	210	133
90	200	800	1200	400	275
62	789	200	800	600	425
70	899	3200	8700	5500	4100
66	500	3000	9800	6800	5075

Download CSV

6 rows selected.