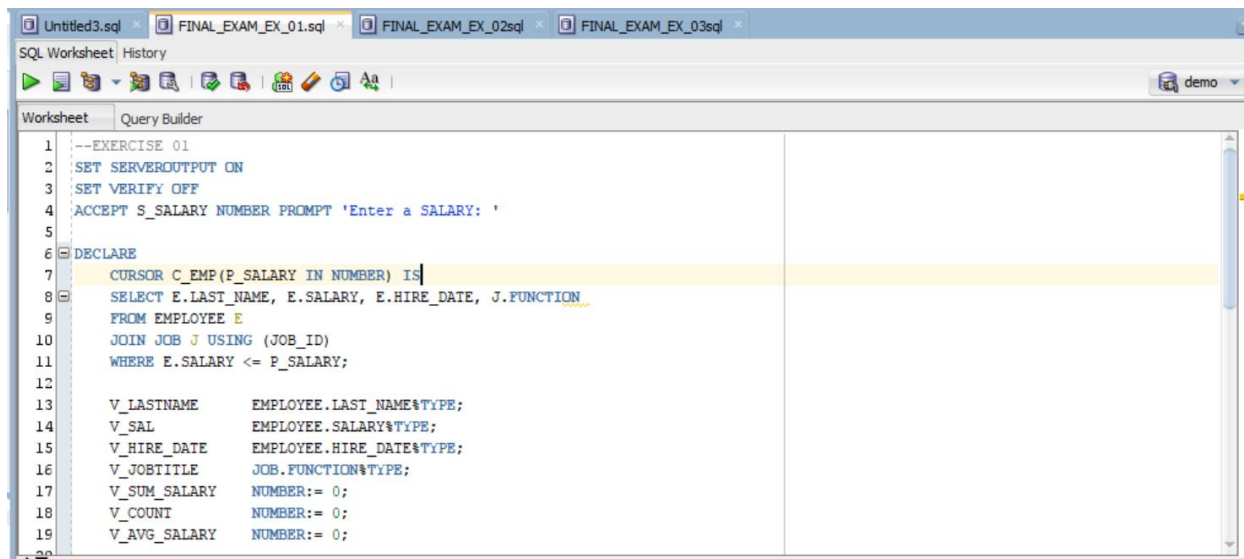


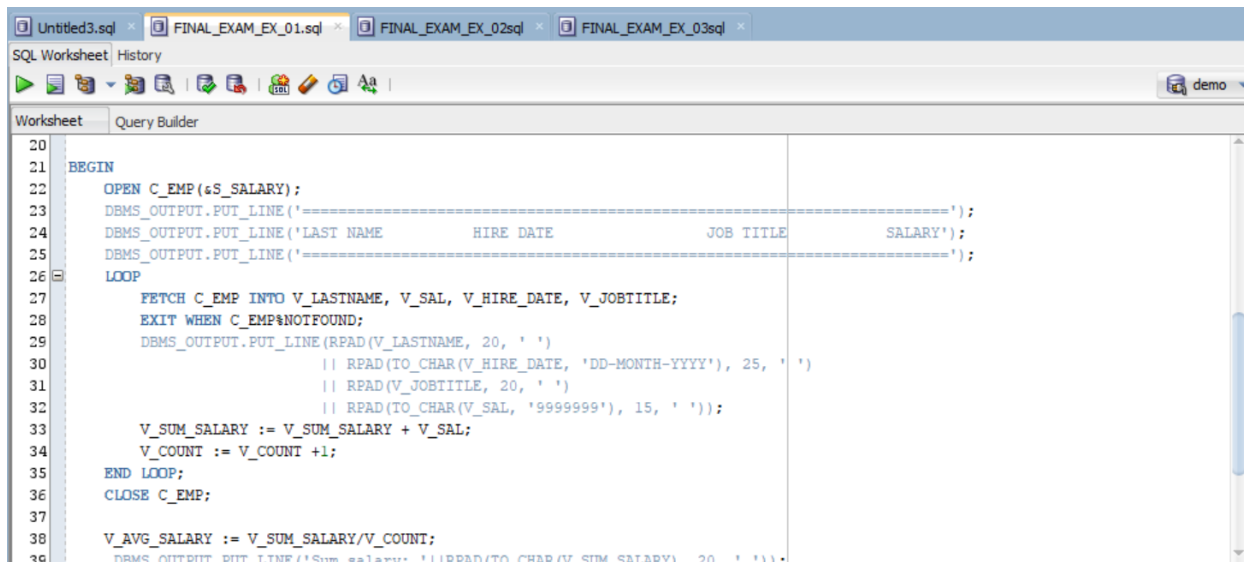
FINAL EXAM DB_2
THOMAS EDSON FREGONEZE

EXERCISE 01



Untitled3.sql * FINAL_EXAM_EX_01.sql * FINAL_EXAM_EX_02.sql * FINAL_EXAM_EX_03.sql *
SQL Worksheet History
Worksheet Query Builder
demo

```
1  --EXERCISE 01
2  SET SERVEROUTPUT ON
3  SET VERIFY OFF
4  ACCEPT S_SALARY NUMBER PROMPT 'Enter a SALARY: '
5
6  DECLARE
7      CURSOR C_EMP(P_SALARY IN NUMBER) IS
8          SELECT E.LAST_NAME, E.SALARY, E.HIRE_DATE, J.FUNCTION
9              FROM EMPLOYEE E
10             JOIN JOB J USING (JOB_ID)
11             WHERE E.SALARY <= P_SALARY;
12
13      V_LASTNAME      EMPLOYEE.LAST_NAME%TYPE;
14      V_SAL            EMPLOYEE.SALARY%TYPE;
15      V_HIRE_DATE      EMPLOYEE.HIRE_DATE%TYPE;
16      V_JOBTITLE       JOB.FUNCTION%TYPE;
17      V_SUM_SALARY     NUMBER:= 0;
18      V_COUNT          NUMBER:= 0;
19      V_AVG_SALARY     NUMBER:= 0;
```



Untitled3.sql * FINAL_EXAM_EX_01.sql * FINAL_EXAM_EX_02.sql * FINAL_EXAM_EX_03.sql *
SQL Worksheet History
Worksheet Query Builder
demo

```
20
21 BEGIN
22     OPEN C_EMP(&S_SALARY);
23     DBMS_OUTPUT.PUT_LINE('=====');
24     DBMS_OUTPUT.PUT_LINE('LAST NAME      HIRE DATE          JOB TITLE          SALARY');
25     DBMS_OUTPUT.PUT_LINE('=====');
26     LOOP
27         FETCH C_EMP INTO V_LASTNAME, V_SAL, V_HIRE_DATE, V_JOBTITLE;
28         EXIT WHEN C_EMP%NOTFOUND;
29         DBMS_OUTPUT.PUT_LINE(RPAD(V_LASTNAME, 20, ' ')
30                               || RPAD(TO_CHAR(V_HIRE_DATE, 'DD-MONTH-YYYY'), 25, ' ')
31                               || RPAD(V_JOBTITLE, 20, ' ')
32                               || RPAD(TO_CHAR(V_SAL, '9999999'), 15, ' '));
33         V_SUM_SALARY := V_SUM_SALARY + V_SAL;
34         V_COUNT := V_COUNT + 1;
35     END LOOP;
36     CLOSE C_EMP;
37
38     V_AVG_SALARY := V_SUM_SALARY/V_COUNT;
39     DBMS_OUTPUT.PUT_LINE('Sum salary: ' || RPAD(TO_CHAR(V_SUM_SALARY), 20, ' '));
```

```
29      DBMS_OUTPUT.PUT_LINE(RPAD(V_LASTNAME, 20, ' '))
30      || RPAD(TO_CHAR(V_HIRE_DATE, 'DD-MONTH-YYYY'), 25, ' ')
31      || RPAD(V_JOBTITLE, 20, ' ')
32      || RPAD(TO_CHAR(V_SAL, '9999999'), 15, ' ');
33      V_SUM_SALARY := V_SUM_SALARY + V_SAL;
34      V_COUNT := V_COUNT + 1;
35  END LOOP;
36  CLOSE C_EMP;
37
38  V_AVG_SALARY := V_SUM_SALARY/V_COUNT;
39  DBMS_OUTPUT.PUT_LINE('Sum salary: '||RPAD(TO_CHAR(V_SUM_SALARY), 20, ' '));
40  DBMS_OUTPUT.PUT_LINE('Avg salary: '||RPAD(TO_CHAR(V_AVG_SALARY), 25, ' '));
41
42  EXCEPTION
43      WHEN NO_DATA_FOUND THEN
44          DBMS_OUTPUT.PUT_LINE('No data found: ' || &S_SALARY||' doesn't exist');
45      WHEN OTHERS THEN
46          DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
47  END;
```

RESULT EX1

Untitled3.sql x FINAL_EXAM_EX_01.sql x FINAL_EXAM_EX_02sql x FINAL_EXAM_EX_03sql x

SQL Worksheet History

Worksheet Query Builder

Script Output x

Task completed in 1.495 seconds

LAST NAME	HIRE DATE	JOB TITLE	SALARY
JENSEN	15-JANUARY -1987	CLERK	750
DOUGLAS	04-JANUARY -1987	CLERK	800
JAMES	03-DECEMBER -1985	CLERK	950
ADAMS	12-JANUARY -1987	CLERK	1100
SMITH	17-DECEMBER -1984	CLERK	800
MURRAY	16-JANUARY -1987	CLERK	750
MARTIN	28-SEPTEMBER-1985	SALESPERSON	1250
LANGE	01-JUNE -1985	SALESPERSON	1250
DUNCAN	31-MAY -1985	SALESPERSON	1250
SHAW	02-APRIL -1985	SALESPERSON	1250
PETERS	31-MARCH -1985	SALESPERSON	1250
WARD	22-FEBRUARY -1985	SALESPERSON	1250
PORTER	15-APRIL -1985	SALESPERSON	1250
Sum salary: 13900			
Avg salary: 1069.23076923076923076923			

EXERCISE 2 Q1

Untitled3.sql x FINAL_EXAM_EX_01.sql x FINAL_EXAM_EX_02sql x FINAL_EXAM_EX_03sql x

Worksheet Query Builder

```

1  --EXERCISE 02 Q1
2  CREATE OR REPLACE FUNCTION empSalesPerson(P_CUSTID IN NUMBER) RETURN VARCHAR2
3  IS
4
5  V_LASTNAME EMPLOYEE.LAST_NAME%TYPE;
6
7  BEGIN
8      SELECT E.LAST_NAME INTO V_LASTNAME
9      FROM EMPLOYEE E
10     JOIN CUSTOMER C ON E.EMPLOYEE_ID = C.SALESPERSON_ID
11     WHERE C.CUSTOMER_ID = P_CUSTID;
12
13     RETURN V_LASTNAME;
14
15 EXCEPTION
16     WHEN NO_DATA_FOUND THEN
17         RETURN NULL;
18 END;

```

```
19
20 SELECT STATUS FROM USER_OBJECTS
21 WHERE OBJECT_NAME = UPPER('empSalesPerson')
22
23 SET SERVEROUTPUT ON
24 SET VERIFY OFF
25 ACCEPT S_CUSTID PROMPT 'ENTER THE CUSTOMER ID: '
26 DECLARE
27 V_CUSTOMERID CUSTOMER.CUSTOMER_ID%TYPE;
28 V_LASTNAME EMPLOYEE.LAST_NAME%TYPE;
29
30 BEGIN
31     V_CUSTOMERID := '&S_CUSTID';
32     V_LASTNAME:= empSalesPerson(V_CUSTOMERID);
33     IF V_LASTNAME IS NOT NULL THEN
34         DBMS_OUTPUT.PUT_LINE('THE LAST NAME THAT CORRESPONDS TO THE CUSTOMER ID: '||V_CUSTOMERID||' IS: '||V_LASTNAME);
35     ELSE
36         DBMS_OUTPUT.PUT_LINE('THE INFORMATION THAT CORRESPOND TO THE ID: '||V_CUSTOMERID||' DOES NOT EXIST');
37     END IF;
```

```
24 SET VERIFY OFF
25 ACCEPT S_CUSTID PROMPT 'ENTER THE CUSTOMER ID: '
26 DECLARE
27 V_CUSTOMERID CUSTOMER.CUSTOMER_ID%TYPE;
28 V_LASTNAME EMPLOYEE.LAST_NAME%TYPE;
29
30 BEGIN
31     V_CUSTOMERID := '&S_CUSTID';
32     V_LASTNAME:= empSalesPerson(V_CUSTOMERID);
33     IF V_LASTNAME IS NOT NULL THEN
34         DBMS_OUTPUT.PUT_LINE('THE LAST NAME THAT CORRESPONDS TO THE CUSTOMER ID: '||V_CUSTOMERID||' IS: '||V_LASTNAME);
35     ELSE
36         DBMS_OUTPUT.PUT_LINE('THE INFORMATION THAT CORRESPOND TO THE ID: '||V_CUSTOMERID||' DOES NOT EXIST');
37     END IF;
38 EXCEPTION
39     WHEN OTHERS THEN
40         DBMS_OUTPUT.PUT_LINE('ERROR : WRONG DATA ENTRY NUMBER');
41 END;
```

RESULT EX 02 Q1

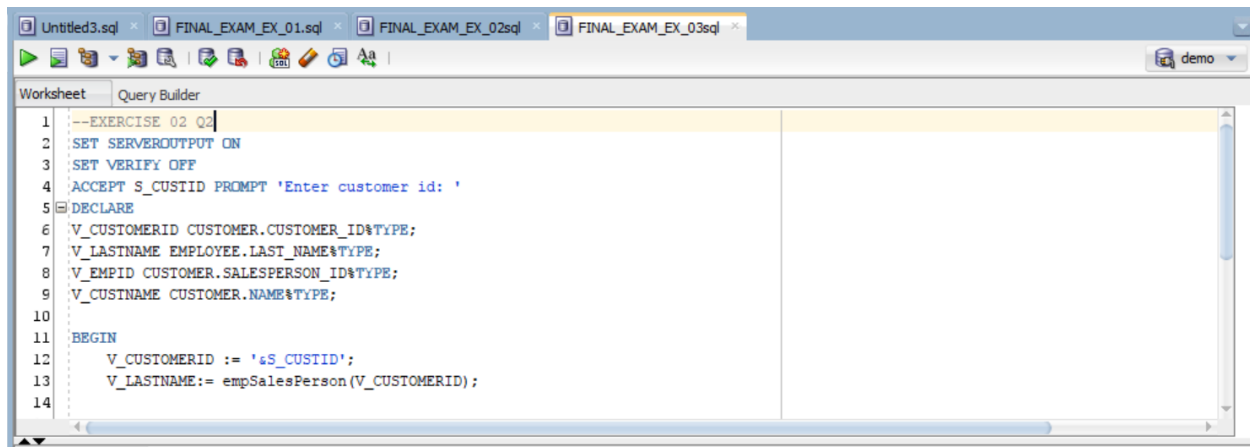
```
Script Output x Query Result x
Task completed in 4.149 seconds

Function EMPSALESPERSON compiled

THE LAST NAME THAT CORRESPONDS TO THE CUSTOMER ID: 100 IS:TURNER

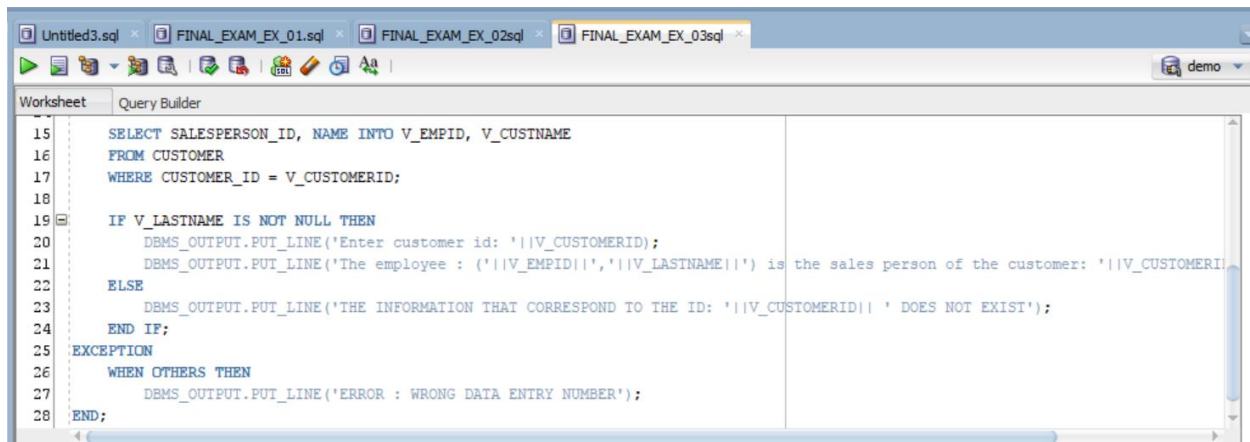
PL/SQL procedure successfully completed.
```

EXERCISE 2 Q2



The screenshot shows the SQL Developer interface with a worksheet titled 'Worksheet'. The code is as follows:

```
1  --EXERCISE 02 Q2
2  SET SERVEROUTPUT ON
3  SET VERIFY OFF
4  ACCEPT S_CUSTID PROMPT 'Enter customer id: '
5  DECLARE
6  V_CUSTOMERID CUSTOMER.CUSTOMER_ID%TYPE;
7  V_LASTNAME EMPLOYEE.LAST_NAME%TYPE;
8  V_EMPID CUSTOMER.SALESPERSON_ID%TYPE;
9  V_CUSTNAME CUSTOMER.NAME%TYPE;
10
11 BEGIN
12     V_CUSTOMERID := '&S_CUSTID';
13     V_LASTNAME := empSalesPerson(V_CUSTOMERID);
14
```



The screenshot shows the SQL Developer interface with a worksheet titled 'Worksheet'. The code continues from the previous part:

```
15     SELECT SALESPERSON_ID, NAME INTO V_EMPID, V_CUSTNAME
16     FROM CUSTOMER
17     WHERE CUSTOMER_ID = V_CUSTOMERID;
18
19     IF V_LASTNAME IS NOT NULL THEN
20         DBMS_OUTPUT.PUT_LINE('Enter customer id: '||V_CUSTOMERID);
21         DBMS_OUTPUT.PUT_LINE('The employee : ('||V_EMPID||','||V_LASTNAME||') is the sales person of the customer: '||V_CUSTOMERID);
22     ELSE
23         DBMS_OUTPUT.PUT_LINE('THE INFORMATION THAT CORRESPOND TO THE ID: '||V_CUSTOMERID|| ' DOES NOT EXIST');
24     END IF;
25 EXCEPTION
26     WHEN OTHERS THEN
27         DBMS_OUTPUT.PUT_LINE('ERROR : WRONG DATA ENTRY NUMBER');
28 END;
```

EXERCISE 02 Q2 RESULT

Worksheet Query Builder

```
16 FROM CUSTOMER
17 WHERE CUSTOMER_ID = V_CUSTOMERID;
18
19 IF V_LASTNAME IS NOT NULL THEN
20     DBMS_OUTPUT.PUT_LINE('Enter customer id: '||V_CUSTOMERID);
21     DBMS_OUTPUT.PUT_LINE('The employee : ('||V_EMPID||','||V_LASTNAME||') is the sales person of the customer: '||V_CUSTOMERID);
22 ELSE
23     DBMS_OUTPUT.PUT_LINE('THE INFORMATION THAT CORRESPOND TO THE ID: '||V_CUSTOMERID|| ' DOES NOT EXIST');
```

Script Output x

Task completed in 2.022 seconds

Enter customer id: 100
The employee : (7844,TURNER) is the sales person of the customer: 100,JOCKSPORTIS

PL/SQL procedure successfully completed.