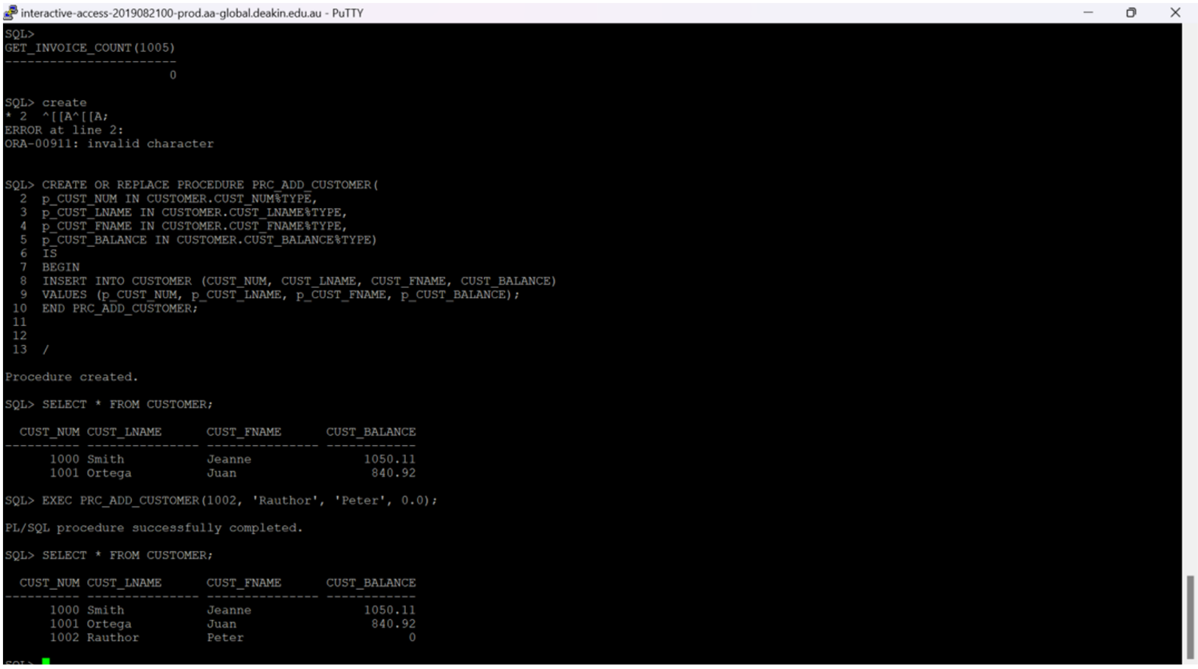
SIT 103 – DATABASE FUNDAMENTALS:

TASK 8.2D PL/PSQL - Trigger, Procedure and Function

SUBMITTED BY – JASVEENA-224001588

QUES 1: 1. Write a procedure named PRC_ADD_CUSTOMER to add a new customer to the CUSTOMER table. The procedure will take four IN arguments which are values of the four columns of the CUSTOMER table for the new customer to be added.

```
CODE: CREATE OR REPLACE PROCEDURE PRC_ADD_CUSTOMER (

 p_CUST_NUM IN CUSTOMER.CUST_NUM%TYPE,

 p_CUST_LNAME IN CUSTOMER.CUST_LNAME%TYPE,

p_CUST_FNAME IN CUSTOMER.CUST_FNAME%TYPE,

p_CUST_BALANCE IN CUSTOMER.CUST_BALANCE%TYPE)

IS

BEGIN INSERT INTO CUSTOMER (CUST_NUM, CUST_LNAME, CUST_FNAME, CUST_BALANCE)
VALUES (p_CUST_NUM, p_CUST_LNAME, p_CUST_FNAME, p_CUST_BALANCE);

 END PRC_ADD_CUSTOMER;

/
```



interactive-access-2019082100-prod.aa-global.deakin.edu.au - PuTTY

```
SQL>
GET_INVOICE_COUNT(1005)
-----------------------
                      0

SQL> create
* 2  ^[[A^[[A;
ERROR at line 2:
ORA-00911: invalid character


SQL> CREATE OR REPLACE PROCEDURE PRC_ADD_CUSTOMER(
     p_CUST_NUM IN CUSTOMER.CUST_NUM%TYPE,
     p_CUST_LNAME IN CUSTOMER.CUST_LNAME%TYPE,
     p_CUST_FNAME IN CUSTOMER.CUST_FNAME%TYPE,
     p_CUST_BALANCE IN CUSTOMER.CUST_BALANCE%TYPE)
     IS
     BEGIN
     INSERT INTO CUSTOMER (CUST_NUM, CUST_LNAME, CUST_FNAME, CUST_BALANCE)
     VALUES (p_CUST_NUM, p_CUST_LNAME, p_CUST_FNAME, p_CUST_BALANCE);
     END PRC_ADD_CUSTOMER;


     /

Procedure created.

SQL> SELECT * FROM CUSTOMER;

  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne               1050.11
      1001 Ortega          Juan                  840.92

SQL> EXEC PRC_ADD_CUSTOMER(1002, 'Rauthor', 'Peter', 0.0);

PL/SQL procedure successfully completed.

SQL> SELECT * FROM CUSTOMER;

  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne               1050.11
      1001 Ortega          Juan                  840.92
      1002 Rauthor         Peter                      0

SQL>
```

QUES 2: Write a function named get_invoice_count to retrieve the number of invoices for a given customer. Your function will take CUST_NUM as argument and returns the number of invoices for the customer.

CREATE OR REPLACE FUNCTION get_invoice_count(

p_CUST_NUM IN INVOICE.CUST_NUM%TYPE)

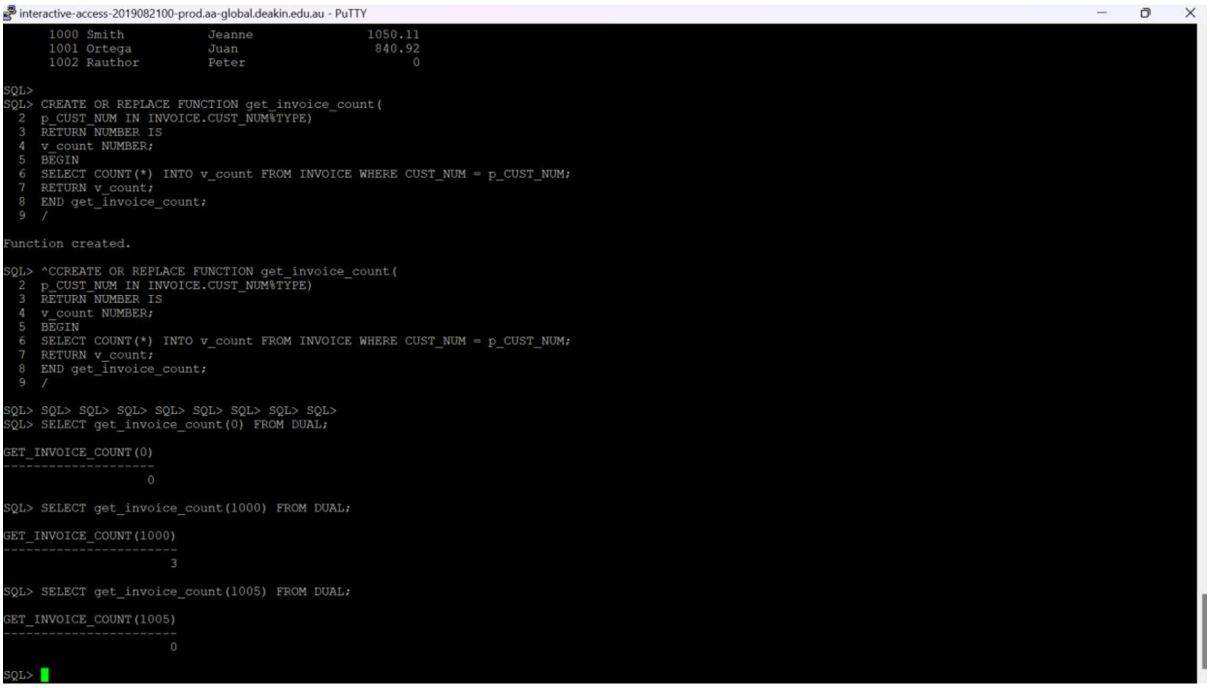
RETURN NUMBER IS

v_count NUMBER;

BEGIN

SELECT COUNT(*) INTO v_count FROM INVOICE WHERE CUST_NUM = p_CUST_NUM; RETURN v_count; END get_invoice_count;

/


interactive-access-2019082100-prod.aa-global.deakin.edu.au - PuTTY

```
      1000 Smith           Jeanne               1050.11
      1001 Ortega          Juan                  840.92
      1002 Rauthor         Peter                      0

SQL>
SQL> CREATE OR REPLACE FUNCTION get_invoice_count(
     p_CUST_NUM IN INVOICE.CUST_NUM%TYPE)
     RETURN NUMBER IS
     v_count NUMBER;
     BEGIN
     SELECT COUNT(*) INTO v_count FROM INVOICE WHERE CUST_NUM = p_CUST_NUM;
     RETURN v_count;
     END get_invoice_count;
     /

Function created.

SQL> ^CCREATE OR REPLACE FUNCTION get_invoice_count(
     p_CUST_NUM IN INVOICE.CUST_NUM%TYPE)
     RETURN NUMBER IS
     v_count NUMBER;
     BEGIN
     SELECT COUNT(*) INTO v_count FROM INVOICE WHERE CUST_NUM = p_CUST_NUM;
     RETURN v_count;
     END get_invoice_count;
     /

SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL>
SQL> SELECT get_invoice_count(0) FROM DUAL;

GET_INVOICE_COUNT(0)
--------------------
                   0

SQL> SELECT get_invoice_count(1000) FROM DUAL;

GET_INVOICE_COUNT(1000)
-----------------------
                      3

SQL> SELECT get_invoice_count(1005) FROM DUAL;

GET_INVOICE_COUNT(1005)
-----------------------
                      0

SQL>
```

QUES 3: CREATE OR REPLACE TRIGGER TRG_UPDATE_CUST_BALANCE

AFTER INSERT ON INVOICE

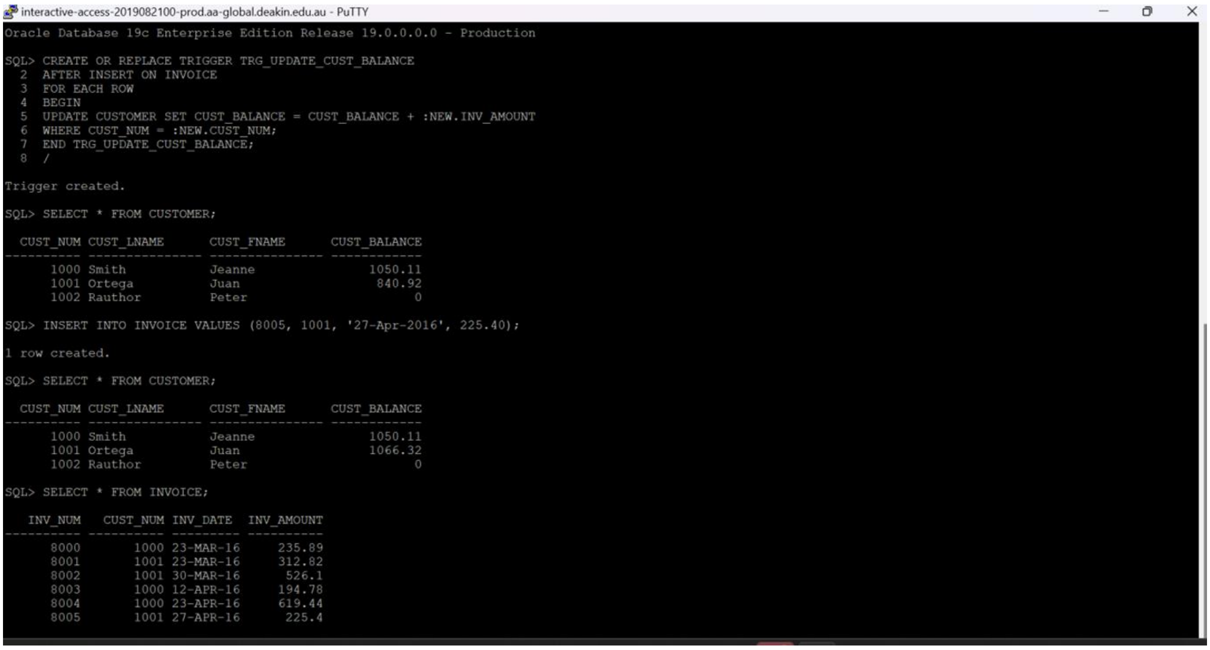
FOR EACH ROW

BEGIN

UPDATE CUSTOMER SET CUST_BALANCE = CUST_BALANCE + :NEW.INV_AMOUNT

WHERE CUST_NUM = :NEW.CUST_NUM; END TRG_UPDATE_CUST_BALANCE;

/

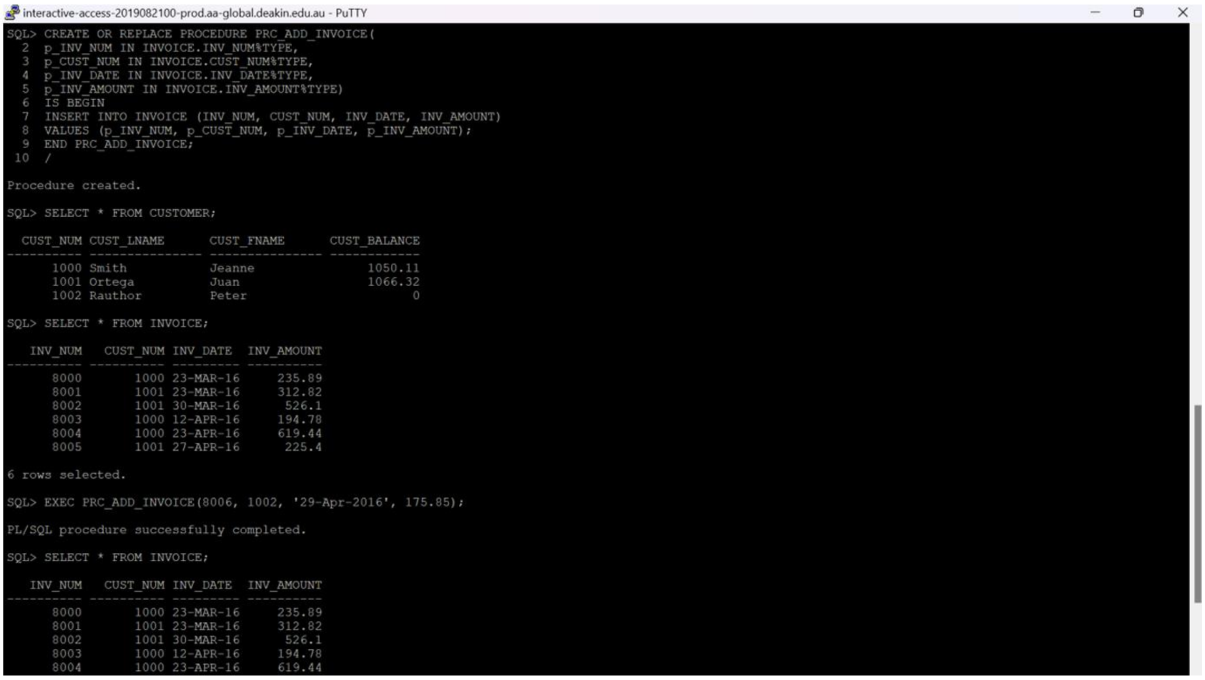


interactive-access-2019082100-prod.aa-global.deakin.edu.au - PuTTY

```
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

SQL> CREATE OR REPLACE TRIGGER TRG_UPDATE_CUST_BALANCE
     AFTER INSERT ON INVOICE
     FOR EACH ROW
     BEGIN
     UPDATE CUSTOMER SET CUST_BALANCE = CUST_BALANCE + :NEW.INV_AMOUNT
     WHERE CUST_NUM = :NEW.CUST_NUM;
     END TRG_UPDATE_CUST_BALANCE;
     /

Trigger created.

SQL> SELECT * FROM CUSTOMER;

  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne               1050.11
      1001 Ortega          Juan                  840.92
      1002 Rauthor         Peter                      0

SQL> INSERT INTO INVOICE VALUES (8005, 1001, '27-Apr-2016', 225.40);

1 row created.

SQL> SELECT * FROM CUSTOMER;

  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne               1050.11
      1001 Ortega          Juan                 1066.32
      1002 Rauthor         Peter                      0

SQL> SELECT * FROM INVOICE;

   INV_NUM   CUST_NUM INV_DATE  INV_AMOUNT
---------- ---------- --------- ----------
      8000       1000 23-MAR-16     235.89
      8001       1001 23-MAR-16     312.82
      8002       1001 30-MAR-16      526.1
      8003       1000 12-APR-16     194.78
      8004       1000 23-APR-16     619.44
      8005       1001 27-APR-16      225.4
```

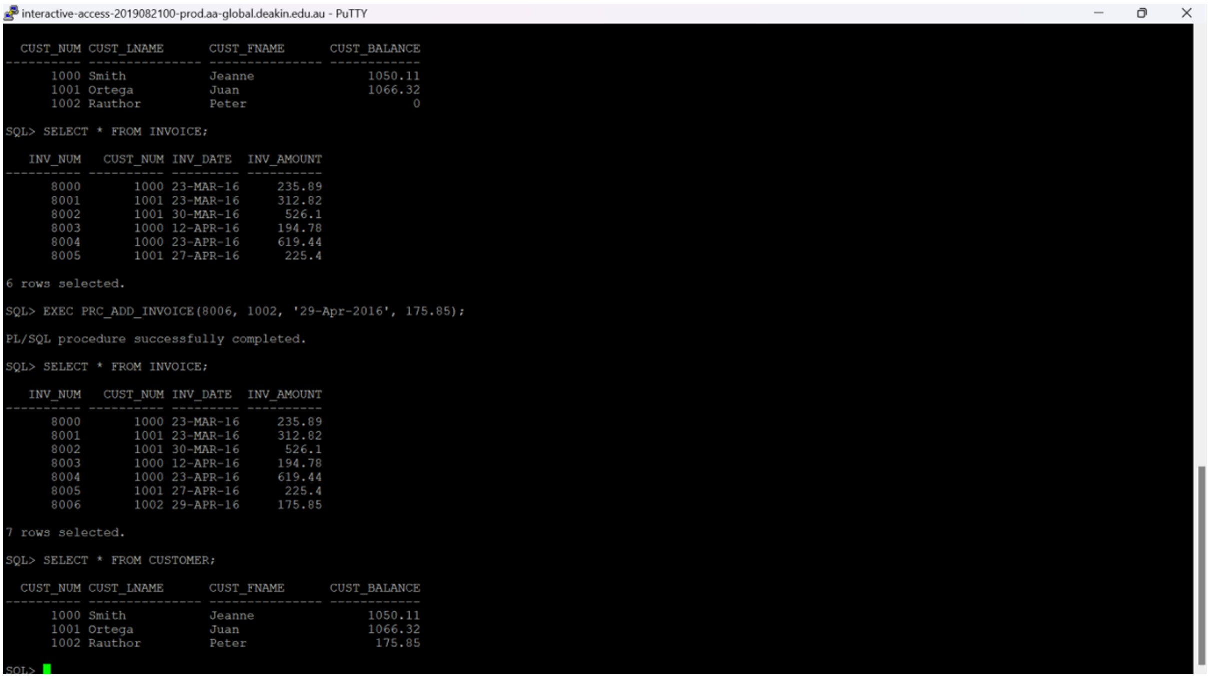
QUES 4: Write a procedure named PRC_ADD_INVOICE to add a new invoice record to the INVOICE table. The procedure will take four IN arguments which are values of the four columns of the INVOICE table for the new invoice to be added.

```
CREATE OR REPLACE PROCEDURE PRC_ADD_INVOICE (
 p_INV_NUM IN INVOICE.INV_NUM%TYPE,
p_CUST_NUM IN INVOICE.CUST_NUM%TYPE,
p_INV_DATE IN INVOICE.INV_DATE%TYPE,
 p_INV_AMOUNT IN INVOICE.INV_AMOUNT%TYPE)
 IS
 BEGIN
 INSERT INTO INVOICE (INV_NUM, CUST_NUM, INV_DATE, INV_AMOUNT)
 VALUES (p_INV_NUM, p_CUST_NUM, p_INV_DATE, p_INV_AMOUNT);
 END PRC_ADD_INVOICE;
 /
```



interactive-access-2019082100-prod.aa-global.deakin.edu.au - PuTTY

```
SQL> CREATE OR REPLACE PROCEDURE PRC_ADD_INVOICE(
  2  p_INV_NUM IN INVOICE.INV_NUM%TYPE,
  3  p_CUST_NUM IN INVOICE.CUST_NUM%TYPE,
  4  p_INV_DATE IN INVOICE.INV_DATE%TYPE,
  5  p_INV_AMOUNT IN INVOICE.INV_AMOUNT%TYPE)
  6  IS BEGIN
  7  INSERT INTO INVOICE (INV_NUM, CUST_NUM, INV_DATE, INV_AMOUNT)
  8  VALUES (p_INV_NUM, p_CUST_NUM, p_INV_DATE, p_INV_AMOUNT);
  9  END PRC_ADD_INVOICE;
 10  /

Procedure created.

SQL> SELECT * FROM CUSTOMER;

  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne               1050.11
      1001 Ortega          Juan                 1066.32
      1002 Rauthor         Peter                      0

SQL> SELECT * FROM INVOICE;

   INV_NUM   CUST_NUM INV_DATE  INV_AMOUNT
---------- ---------- --------- ----------
      8000       1000 23-MAR-16     235.89
      8001       1001 23-MAR-16     312.82
      8002       1001 30-MAR-16      526.1
      8003       1000 12-APR-16     194.78
      8004       1000 23-APR-16     619.44
      8005       1001 27-APR-16      225.4

6 rows selected.

SQL> EXEC PRC_ADD_INVOICE(8006, 1002, '29-Apr-2016', 175.85);

PL/SQL procedure successfully completed.

SQL> SELECT * FROM INVOICE;

   INV_NUM   CUST_NUM INV_DATE  INV_AMOUNT
---------- ---------- --------- ----------
      8000       1000 23-MAR-16     235.89
      8001       1001 23-MAR-16     312.82
      8002       1001 30-MAR-16      526.1
      8003       1000 12-APR-16     194.78
      8004       1000 23-APR-16     619.44
```



interactive-access-2019082100-prod.aa-global.deakin.edu.au - PuTTY

```
  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne               1050.11
      1001 Ortega          Juan                 1066.32
      1002 Rauthor         Peter                      0

SQL> SELECT * FROM INVOICE;

   INV_NUM   CUST_NUM INV_DATE  INV_AMOUNT
---------- ---------- --------- ----------
      8000       1000 23-MAR-16     235.89
      8001       1001 23-MAR-16     312.82
      8002       1001 30-MAR-16      526.1
      8003       1000 12-APR-16     194.78
      8004       1000 23-APR-16     619.44
      8005       1001 27-APR-16      225.4

6 rows selected.

SQL> EXEC PRC_ADD_INVOICE(8006, 1002, '29-Apr-2016', 175.85);

PL/SQL procedure successfully completed.

SQL> SELECT * FROM INVOICE;

   INV_NUM   CUST_NUM INV_DATE  INV_AMOUNT
---------- ---------- --------- ----------
      8000       1000 23-MAR-16     235.89
      8001       1001 23-MAR-16     312.82
      8002       1001 30-MAR-16      526.1
      8003       1000 12-APR-16     194.78
      8004       1000 23-APR-16     619.44
      8005       1001 27-APR-16      225.4
      8006       1002 29-APR-16     175.85

7 rows selected.

SQL> SELECT * FROM CUSTOMER;

  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1000 Smith           Jeanne               1050.11
      1001 Ortega          Juan                 1066.32
      1002 Rauthor         Peter                 175.85

SQL>
```