1. write a procedure named PRC_ADD_CUSTOMER to add a new customer to CUSTOMER table.

```
CREATE OR REPLACE PROCEDURE PRC_ADD_CUSTOMER(CUST_NUM IN
NUMBER,CUST_LNAME IN VARCHAR,CUST_FNAME IN VARCHAR,
                                    CUST_BALANCE IN NUMBER)
AS BEGIN
      INSERT INTO CUSTOMER VALUES (CUST_NUM, CUST_LNAME, CUST_FNAME,
CUST_BALANCE);
END;
```

```
SQL> CREATE OR REPLACE PROCEDURE PRC_ADD_CUSTOMER(CUST_NUM IN NUMBER,CUST_LNAME IN VARCHAR,CUST_FNAME IN VARCHAR,CUST_BALANCE IN NUMBER)
  2  AS BEGIN
  3  INSERT INTO CUSTOMER VALUES (CUST_NUM, CUST_LNAME, CUST_FNAME, CUST_BALANCE);
  4  END;
  5  /

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

SQL>
```

2. write a function name get_invoice_count to retrieve the number of invoices for a given customer.

```
CREATE OR REPLACE FUNCTION get_invoice_count(NUM IN NUMBER)
RETURN NUMBER IS
   invoicesCount NUMBER := 0;
BEGIN
 SELECT COUNT(1) INTO invoicesCount FROM INVOICE WHERE INVOICE.CUST_NUM =NUM;
RETURN invoicesCount;
END;
```

```
SQL> CREATE OR REPLACE FUNCTION get_invoice_count(NUM IN NUMBER)
RETURN NUMBER IS
    invoicesCount NUMBER := 0;
BEGIN
 SELECT COUNT(1) INTO invoicesCount FROM INVOICE WHERE INVOICE.CUST_NUM =NUM;
RETURN invoicesCount;
END;  2    3    4    5    6    7
  8  /

Function created.


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
```

3. write a trigger name TRG_UPDATE_CUST_BALANCE to update the CUST_BALANCE in the CUSTOMER table when a new invoice is added in the INVOICE table;

```
CREATE OR REPLACE TRIGGER TRG_UPDATE_CUST_BALANCE
   AFTER INSERT OR UPDATE OR DELETE ON INVOICE
   FOR EACH ROW
DECLARE
   TOTAL_BALANCE NUMBER := 0;
BEGIN
   IF INSERTING THEN
      SELECT NVL(CUST_BALANCE, 0) INTO TOTAL_BALANCE FROM CUSTOMER WHERE
CUSTOMER.CUST_NUM = :NEW.CUST_NUM;
      TOTAL_BALANCE := TOTAL_BALANCE + :NEW.INV_AMOUNT;
      UPDATE CUSTOMER SET CUST_BALANCE = TOTAL_BALANCE WHERE
CUSTOMER.CUST_NUM = :NEW.CUST_NUM;
   ELSIF UPDATING THEN
      SELECT CUST_BALANCE INTO TOTAL_BALANCE FROM CUSTOMER WHERE
CUSTOMER.CUST_NUM = :OLD.CUST_NUM;
      TOTAL_BALANCE := TOTAL_BALANCE - :OLD.INV_AMOUNT + :NEW.INV_AMOUNT;
      UPDATE CUSTOMER SET CUST_BALANCE = TOTAL_BALANCE WHERE
CUSTOMER.CUST_NUM = :OLD.CUST_NUM;
   ELSIF DELETING THEN
      SELECT CUST_BALANCE INTO TOTAL_BALANCE FROM CUSTOMER WHERE
```

```
CUSTOMER.CUST_NUM = :OLD.CUST_NUM;
      TOTAL_BALANCE := TOTAL_BALANCE – :OLD.INV_AMOUNT;
      UPDATE CUSTOMER SET CUST_BALANCE = TOTAL_BALANCE WHERE
CUSTOMER.CUST_NUM = :OLD.CUST_NUM;
   END IF;
END;
```

```
SQL> CREATE OR REPLACE TRIGGER TRG_UPDATE_CUST_BALANCE
    AFTER INSERT OR UPDATE OR DELETE ON INVOICE
    FOR EACH ROW
DECLARE
    TOTAL_BALANCE NUMBER := 0;
BEGIN
    IF INSERTING THEN
        SELECT NVL(CUST_BALANCE, 0) INTO TOTAL_BALANCE FROM CUSTOMER WHERE CUSTOMER.CUST_NUM = :NEW.CUST_NUM;
        TOTAL_BALANCE := TOTAL_BALANCE + :NEW.INV_AMOUNT;
        UPDATE CUSTOMER SET CUST_BALANCE = TOTAL_BALANCE WHERE CUSTOMER.CUST_NUM = :NEW.CUST_NUM;
    ELSIF UPDATING THEN
        SELECT CUST_BALANCE INTO TOTAL_BALANCE FROM CUSTOMER WHERE CUSTOMER.CUST_NUM = :OLD.CUST_NUM;
        TOTAL_BALANCE := TOTAL_BALANCE - :OLD.INV_AMOUNT + :NEW.INV_AMOUNT;
        UPDATE CUSTOMER SET CUST_BALANCE = TOTAL_BALANCE WHERE CUSTOMER.CUST_NUM = :OLD.CUST_NUM;
    ELSIF DELETING THEN
        SELECT CUST_BALANCE INTO TOTAL_BALANCE FROM CUSTOMER WHERE CUSTOMER.CUST_NUM = :OLD.CUST_NUM;
        TOTAL_BALANCE := TOTAL_BALANCE - :OLD.INV_AMOUNT;
        UPDATE CUSTOMER SET CUST_BALANCE = TOTAL_BALANCE WHERE CUSTOMER.CUST_NUM = :OLD.CU  2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18  ST_NUM;
    END IF;
END; 19   20
 21  /

Trigger created.

SQL> SELECT * FROM CUSTOMER;

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

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
---------- --------------- --------------- ------------
      1002 Rauthor         Peter                      0
      1000 Smith           Jeanne                1050.11
      1001 Ortega          Juan                  1066.32
```

4. write aprocedure named PRC_ADD_INVOICE to add a new invoice record to the INVOICE table.

```
CREATE OR REPLACE PROCEDURE PRC_ADD_INVOICE(INV_NUM IN NUMBER,CUST_NUM
IN NUMBER,INV_DATE IN DATE,INV_AMOUNT IN
   NUMBER)
AS
BEGIN
   INSERT INTO INVOICE VALUES (INV_NUM, CUST_NUM, INV_DATE, INV_AMOUNT);
END;
```

```
SQL> CREATE OR REPLACE PROCEDURE PRC_ADD_INVOICE(INV_NUM IN NUMBER,CUST_NUM IN NUMBER,INV_DATE IN DATE,INV_AMOUNT IN
    NUMBER)
AS
BEGIN
    INSERT INTO INVOICE VALUES (INV_NUM, CUST_NUM, INV_DATE, INV_AMOUNT);
END;  2    3    4    5    6
  7  /

Procedure created.
```

```
SQL> SELECT * FROM CUSTOMER;

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

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

SQL> EXEC PRC_ADD_INVOICE(8006,1002,TO_DATE('29-Apr-2016','DD-MON-YYYY'),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
      8006       1002 29-APR-16     175.85

6 rows selected.

SQL> SELECT * FROM CUSTOMER;

  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
---------- --------------- --------------- ------------
      1002 Rauthor         Peter                 175.85
      1000 Smith           Jeanne               1050.11
      1001 Ortega          Juan                  840.92
```