

Name: Ngọc Phương Trang Duong
ID: 223990468

Task 8.2D

Question 1:

```
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;
/
```

2 3 4 5 6 7 8 9
Trigger created.

```
SQL> SELECT * FROM CUSTOMER;
```

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

```
SQL> INSERT INTO INVOICE (INV_NUM, CUST_NUM, INV_DATE, INV_AMOUNT)
VALUES (8005, 1001, '27-Apr-2022', 225.40); 2
```

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

Question 2:

```
SQL> CREATE OR REPLACE PROCEDURE PRC_ADD_CUSTOMER(  
    p_cust_num IN NUMBER,  
    p_cust_lname IN VARCHAR2,  
    p_cust_fname IN VARCHAR2,  
    p_cust_balance IN NUMBER  
) AS  
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;  
/  
2      3      4      5      6      7      8      9      10     11  
Procedure created.
```

```
[SQL> SELECT * FROM CUSTOMER;
```

CUST_NUM	CUST_LNAME	CUST_FNAME	CUST_BALANCE
1000	Smith	Jeanne	1050.11
1001	Ortega	Juan	1066.32

```
[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	1066.32
1002	Rauthor	Peter	0

Question 3:

```
SQL> CREATE OR REPLACE PROCEDURE PRC_ADD_INVOICE(  
    p_inv_num IN NUMBER,  
    p_cust_num IN NUMBER,  
    p_inv_date IN DATE,  
    p_inv_amount IN NUMBER  
) AS  
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;  
/  
2      3      4      5      6      7      8      9      10     11  
Procedure created.
```

```
[SQL> SELECT * FROM INVOICE;
```

INV_NUM	CUST_NUM	INV_DATE	INV_AMOUNT
8000	1000	23-MAR-22	235.89
8001	1001	23-MAR-22	312.82
8002	1001	30-MAR-22	526.1
8003	1000	12-APR-22	194.78
8004	1000	23-APR-22	619.44
8005	1001	27-APR-22	225.4

6 rows selected.

```
[SQL> EXEC PRC_ADD_INVOICE(8006, 1002, '29-Apr-2022', 175.85);
```

PL/SQL procedure successfully completed.

```
[SQL> SELECT * FROM INVOICE;
```

INV_NUM	CUST_NUM	INV_DATE	INV_AMOUNT
8000	1000	23-MAR-22	235.89
8001	1001	23-MAR-22	312.82
8002	1001	30-MAR-22	526.1
8003	1000	12-APR-22	194.78
8004	1000	23-APR-22	619.44
8005	1001	27-APR-22	225.4
8006	1002	29-APR-22	175.85

7 rows selected.

Question 4:

```
SQL> CREATE OR REPLACE TRIGGER TRG_UPDATE_CUST_BALANCE2
AFTER DELETE ON INVOICE
FOR EACH ROW
BEGIN
    UPDATE CUSTOMER
    SET CUST_BALANCE = CUST_BALANCE - :OLD.INV_AMOUNT
    WHERE CUST_NUM = :OLD.CUST_NUM;
END;
/
```

2 3 4 5 6 7 8 9
Trigger created.

```
SQL> SELECT * FROM INVOICE;
SELECT * FROM CUSTOMER;
```

INV_NUM	CUST_NUM	INV_DATE	INV_AMOUNT
8000	1000	23-MAR-22	235.89
8001	1001	23-MAR-22	312.82
8002	1001	30-MAR-22	526.1
8003	1000	12-APR-22	194.78
8004	1000	23-APR-22	619.44
8005	1001	27-APR-22	225.4
8006	1002	29-APR-22	175.85

7 rows selected.

```
SQL> DELETE FROM INVOICE WHERE INV_NUM = 8000;
```

```
SELECT * FROM CUSTOMER;
```

```
DELETE FROM INVOICE WHERE INV_NUM = 8000;
```

```
1 row deleted.
```

```
SQL> SQL>
```

CUST_NUM	CUST_LNAME	CUST_FNAME	CUST_BALANCE
1000	Smith	Jeanne	814.22
1001	Ortega	Juan	1066.32
1002	Rauthor	Peter	175.85

```
SQL> SQL>
```

```
0 rows deleted.
```

```
SQL> SELECT * FROM INVOICE;
```

```
SELECT * FROM CUSTOMER;
```

INV_NUM	CUST_NUM	INV_DATE	INV_AMOUNT
8001	1001	23-MAR-22	312.82
8002	1001	30-MAR-22	526.1
8003	1000	12-APR-22	194.78
8004	1000	23-APR-22	619.44
8005	1001	27-APR-22	225.4
8006	1002	29-APR-22	175.85

```
6 rows selected.
```

Question 5:

```
SQL> CREATE OR REPLACE FUNCTION GET_INVOICE_COUNT(p_cust_num IN NUMBER)
RETURN NUMBER IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM INVOICE
    WHERE CUST_NUM = p_cust_num;
```

```
    RETURN v_count;
END;
```

```
/
```

```
   2       3       4       5       6       7       8       9      10      11      12
Function created.
```

```
SQL> SELECT GET_INVOICE_COUNT(1000) FROM DUAL;
SELECT GET_INVOICE_COUNT(1001) FROM DUAL;
SELECT GET_INVOICE_COUNT(1002) FROM DUAL;
GET_INVOICE_COUNT(1000)
```

```
-----
                                2
```

```
SQL>
GET_INVOICE_COUNT(1001)
```

```
-----
                                3
```

```
[SQL> SELECT GET_INVOICE_COUNT(1000) FROM DUAL;
```

```
GET_INVOICE_COUNT(1000)
-----
                                2
```

```
[SQL> SELECT GET_INVOICE_COUNT(1001) FROM DUAL;
```

```
GET_INVOICE_COUNT(1001)
-----
                                3
```

```
[SQL> SELECT GET_INVOICE_COUNT(1002) FROM DUAL;
```

```
GET_INVOICE_COUNT(1002)
-----
                                1
```