

# Data & Information Management

## Assignment 8.2D

```
1. 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;
```

### Test

```
INSERT INTO INVOICE VALUES (8005, 1001, '27-Apr-2016', 225.40)
```

```
2. CREATE OR REPLACE PROCEDURE PRC_ADD_CUSTOMER
(
P_CUST_NUM NUMBER,
P_CUST_FNAME VARCHAR2,
P_CUST_LNAME VARCHAR2,
P_CUST_BALANCE NUMBER
)
IS
BEGIN
INSERT INTO CUSTOMER VALUES (P_CUST_NUM, P_CUST_FNAME, P_CUST_LNAME,
P_CUST_BALANCE) VALUES (P_CUST_NUM, P_CUST_FNAME, P_CUST_LNAME,
P_CUST_BALANCE);
END;
```

### Test

```
EXEC PRC_ADD_CUSTOMER(1002, 'Rauthor', 'Peter', 0)
```

3. CREATE OR REPLACE PROCEDURE PRC\_ADD\_INVOICE

```
(  
  P_INV_NUM NUMBER,  
  P_CUST_NUM NUMBER,  
  P_INV_DATE DATE,  
  P_INV_AMOUNT NUMBER  
)  
IS  
BEGIN  
  INSERT INTO INVOICE VALUES (P_INV_NUM, P_CUST_NUM, P_INV_DATE, P_INV_AMOUNT);  
END;
```

**Test**

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

4. CREATE OR REPLACE FUNCTION get\_invoice\_count(p\_cust\_num 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;
```

**Test**

```
SELECT get_invoice_count(0) FROM DUAL;  
SELECT get_invoice_count(1000) FROM DUAL;  
SELECT get_invoice_count(1005) FROM DUAL;
```