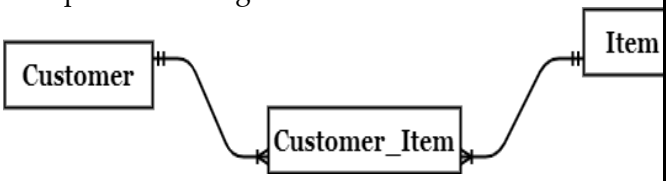
CORRECTION OF CGCE JUNE 2025

SECTION A: DATABASE

Task I

(i) **Customer**(*CustomerID*, *CustomerName*)
Item(*ItemCode*, *ItemName*, *UnitPrice*)
Customer_Item(*CustomerID*, *ItemCode*, *Quantity*, *TotalPrice*, *TDate*)

(ii) Completed ER Diagram



Task II

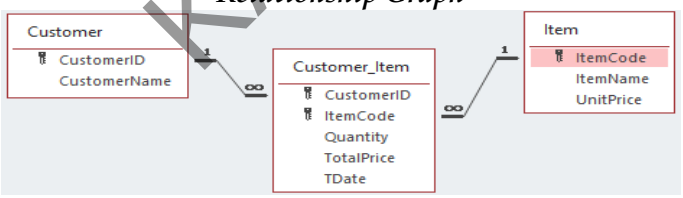
(i)
```
CREATE DATABASE NinashopDB;
```

(ii)
```
CREATE TABLE Customer(
    CustomerID VARCHAR(5) NOT NULL,
    CustomerName VARCHAR(20),
    PRIMARY KEY(CustomerID)
);
```

SQL Code for the creation of other two tables

```
CREATE TABLE Item(
  ItemCode VARCHAR(5) NOT NULL,
  ItemName VARCHAR(10),
  UnitPrice INT,
  PRIMARY KEY(ItemCode)
);
CREATE TABLE Customer_Item(
  CustomerID VARCHAR(5),
  ItemCode VARCHAR(5),
  Quantity INT,
  TotalPrice INT,
  TDate DATE,
  PRIMARY KEY(CustomerID, ItemCode),
  FOREIGN KEY(CustomerID) REFERENCES
  Customer(CustomerID),
  FOREIGN KEY(ItemCode) REFERENCES
  Item(ItemCode)
);
```

Relationship Graph



(iii)
```
ALTER TABLE Customerr_Item
  RENAME COLUMN Quantity  TO Qty;
```

Task III

(i)
```
INSERT INTO CUSTOMER
VALUES(("C001", "Ebai Paul"), ("C002", "Eyere
Esther"));
```

(ii)
```
UPDATE ITEM SET UnitPrice = 200
WHERE ItemCode = "IT003";
```

(iii) **Populated tables**

Customer

CustomerID	CustomerName
C001	Ebai Paul
C002	Eyere Esther

Item

ItemCode	ItemName	UnitPrice
IT001	Rice	200
IT002	Beans	250
IT003	Corn	175

Customer_Item

CustomerID	ItemCode	Qty	TotalPrice	TDate
C001	IT001	3	600	09/09/2021
C001	IT002	2	500	09/09/2021
C001	IT003	2	350	10/10/2021
C002	IT003	5	875	11/12/2021

(iv) **SQL Code:** *Approach 1*

```
SELECT Customer.CustomerID,
Customer.CustomerName, Customer_Item.Qty,
Customer_Item.TotalPrice
FROM Item INNER JOIN (Customer INNER
JOIN Customer_Item ON Customer.CustomerID =
Customer_Item.CustomerID) ON Item.ItemCode =
Customer_Item.ItemCode
WHERE Item.ItemCode ="IT002";
```

SQL Code: *Approach 2*

```
SELECT Customer.CustomerID,
Customer.CustomerName, Customer_Item.Qty,
Customer_Item.TotalPrice
FROM Item, Customer, Customer_Item
WHERE (Customer.CustomerID =
Customer_Item.CustomerID) AND
(Item.ItemCode=Customer_Item.ItemCode) AND
(Item.ItemCode="IT002");
```

Output

Query App1

CustomerID	CustomerName	Qty	TotalPrice
C001	Ebai Paul	2	500

SECTION B: PROGRAMMING

Task IV

(i) (a) As stated in the description: 1year.

(b) Initially: 27; **Year 1**: $27 + 9 - 6 = \mathbf{30} < \mathbf{37}$
Year 2: $30 + 10 - 7 = \mathbf{43} \geq \mathbf{37}$ Hence, ***2 years***

(ii) Initially: 4; **Year 1**: $4 - 1 = \mathbf{3}$; **Year 2**: $3 - 0 = \mathbf{3}$;
Year 3: $3 - 0 = \mathbf{3}$; So, ***never will it drop to 0.***

Task V

(i)
```
FUNCTION InputStartSize()
    INTEGER population;
    PRINT("Enter the starting population size: ");
    READ(population)
    WHILE(population < 9) DO
        PRINT("The population size MUST 9 or above");
        PRINT("Enter the starting population size: ");
        READ(population);
    ENDWHILE
    RETURN population;
ENDFUNCTION
```

(ii)
```
FUNCTION CalculateYears(INTEGER start_size,
    INTEGER end_size)
        INTEGER year = 0, size = start_size;
        WHILE(size < end_size) DO
            size = size + (size/4) – (size/3);
            year = year + 1;
        ENDWHILE
        RETURN year;
    ENDFUNCTION
```

TASK VI: Full Code:

```
#include<stdio.h>
int InputStartSize(){
        int population;
        printf("Enter the population size: ");
        scanf("%d", &population);
        while(population<9){
                printf("The population must be greater than or equals to 9.\n");
                printf("Enter the population size: ");
                scanf("%d", &population);
        }
        return population;
}
int CalculateYears(int start_size, int end_size){
        int year = 0, size = start_size;
        while(size<end_size){
                size = size+(size/3)-(size/4);
                year++;
        }
        return year;
}
int InputEndSize(int current_size){
        int target;
        printf("Enter the target population size: ");
        scanf("%d", &target);
        while(target<current_size){
                printf("The population must be greater than or equals to %d.\n", current_size);
                printf("Enter the target population size: ");
                scanf("%d", &target);
        }
        return target;
}
int main(){
        int start_size, end_size, noOfYears;
        start_size = InputStartSize();
        end_size = InputEndSize(start_size);
        noOfYears = CalculateYears(start_size, end_size);
        printf("Years: %d",noOfYears);
        return 0;
}
```

TASK VII: Output

```
Enter the population size: 5
The population must be greater than or equals to 9.
Enter the population size: 7
The population must be greater than or equals to 9.
Enter the population size: 18
Enter the target population size: 10
The population must be greater than or equals to 18.
Enter the target population size: 17
The population must be greater than or equals to 18.
Enter the target population size: 25
Years: 4
```