AF-SM-04\_05.2020.G

Supplier ცხრილი სვეტებით SupplierID (მთელი რიცხვი, ძირითადი გასაღები), SupplierName (varchar, არა null) და ContactEmail (varchar). დაამატეთ შემოწმების შეზღუდვა, რათა დარწმუნდეთ, რომ ContactEmail შეიცავს '@' სიმბოლოს.

ALTER TABLE Supplier

ADD CONSTRAINT chk\_ContactEmail\_AtSymbol

CHECK (ContactEmail LIKE '%@%');

SupplyOrder ცხრილი სვეტებით OrderID (მთელი რიცხვი, ძირითადი გასაღები), SupplierID (მთელი რიცხვი), OrderDate (თარიღი), TotalAmount (ათწილადი). დაამატეთ foreign გასაღების შეზღუდვა SupplierID-დან SupplyOrder-ში SupplierID-ს Supplier-ში და შეამოწმეთ შეზღუდვა, რათა უზრუნველყოთ TotalAmount 0-ზე მეტობა.

ALTER TABLE SupplyOrder

ADD CONSTRAINT fk\_SupplyOrder\_Supplier

FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID);

ALTER TABLE SupplyOrder

ADD CONSTRAINT chk\_TotalAmount\_Positive

CHECK (TotalAmount > 0);

ჩააინსერტეთ სამი მომწოდებელი („ABC Supplies“ ID 1-ით და ელფოსტა „abc@supplies.com“, „XYZ Distributors“ ID 2-ით და ელფოსტა „xyz@distributors.com“ და „LMN Providers“ ID 3-ით და ელფოსტა „lmn@providers.com“) მიმწოდებლის ცხრილში.

INSERT INTO Supplier (SupplierID, SupplierName, ContactEmail) VALUES

(1, 'ABC Supplies', 'abc@supplies.com'),

(2, 'XYZ Distributors', 'xyz@distributors.com'),

(3, 'LMN Providers', 'lmn@providers.com');

ჩააინსერტეთ სამი supply orders (Order ID 101 for Supplier ID 1 on '2024-01-01' with total amount 5000, Order ID 102 for Supplier ID 2 on '2024-02-01' with total amount 6000, and Order ID 103 for Supplier ID 3 on '2024-03-01' with total amount 7000) SupplyOrder თებილში.

INSERT INTO SupplyOrder (OrderID, SupplierID, OrderDate, TotalAmount) VALUES

(101, 1, '2024-01-01', 5000),

(102, 2, '2024-02-01', 6000),

(103, 3, '2024-03-01', 7000);

დააფდეითეთ შეკვეთების ჯამური რაოდენობა( TotalAmount), გაიზარდეთ 10%-ით, სადაც OrderDate არის „2024-02-15“-მდე.

UPDATE SupplyOrder

SET TotalAmount = TotalAmount \* 1.10

WHERE OrderDate < '2024-02-15';

წაშალეთ მომწოდებლები, რომლებსაც არ გაუკეთებიათ მიწოდების შეკვეთები.

3) შექმენით ფუნქცია სახელად GetSupplierOrderCount, რომელიც იღებს SupplierID-ს (მთლიანი რიცხვი) და აბრუნებს შეკვეთების რაოდენობას ამ მომწოდებლისთვის.

CREATE FUNCTION GetSupplierOrderCount (@SupplierID INT)

RETURNS INT

AS

BEGIN

DECLARE @OrderCount INT;

SELECT @OrderCount = COUNT(\*)

FROM SupplyOrder

WHERE SupplierID = @SupplierID;

RETURN @OrderCount;

END;

4) დაწერეთ სკრიპტი IF...ELSE და WHILE გამოყენებით 1-დან 10-მდე რიცხვების ყოველი მომდევნო რიცხვის ნამრლავით (1\*2, 2\*3) გამოიტანეთ შედეგი. თუ რიცხვი ბოლოვდება 0-ით ლუპმა არ უნდა გამოიტანოს რიცხვი.

5) შექმენით ტრიგერი სახელად trgAfterUpdateSupplyOrder SupplyOrder ცხრილზე, რომელიც ჩაწერს ჩანაწერს OrderHistory(შექმენით ცხრილი) ცხრილში სვეტებით HistoryID, OrderID, OldTotalAmount, NewTotalAmount და ChangeDate, როდესაც განახლდება შეკვეთის TotalAmount - SupplyOrder ცხრილში.