

CREAZIONE DEL DATABASE

```
CREATE DATABASE IF NOT EXISTS CustomerOrders;
```

```
USE CustomerOrders;
```

Creazione della tabella clienti

```
CREATE TABLE Customers (  
    CustomerID INT PRIMARY KEY AUTO_INCREMENT,  
    CustomerName VARCHAR(100),  
    Email VARCHAR(100),  
    JoinDate DATE  
);
```

Creazione della tabella ordini

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY AUTO_INCREMENT,  
    CustomerID INT,  
    OrderDate DATE,  
    Amount DECIMAL(10, 2),  
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);
```

Creazione della tabella dei log

```
CREATE TABLE OrderLogs (  
    LogID INT PRIMARY KEY AUTO_INCREMENT,  
    OrderID INT,  
    LogMessage VARCHAR(255),  
    LogDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

INSERIMENTO DATI:

Inserimento dei dati nella tabella Customers

```
INSERT INTO Customers (CustomerName, Email, JoinDate) VALUES  
( 'Alice Rossi', 'alice@example.com', '2022-01-15'),  
( 'Bob Bianchi', 'bob@example.com', '2023-03-10'),  
( 'Carla Verdi', 'carla@example.com', '2021-06-21');
```

Inserimento dei dati nella tabella Orders

```
INSERT INTO Orders (CustomerID, OrderDate, Amount) VALUES  
(1, '2023-01-10', 100.50),  
(1, '2023-02-15', 200.00),  
(2, '2023-03-20', 150.75),  
(3, '2023-04-05', 300.00),  
(2, '2023-05-12', 450.25);
```

RICHIESTE:

ESERCIZIO 1: Stored Function per calcolare il totale degli ordini di un cliente

Questa funzione accetta un CustomerID come parametro e restituisce il totale degli ordini per quel cliente.

ESERCIZIO 2: Stored Procedure per elencare tutti gli ordini di un cliente

Questa procedura accetta un CustomerID come parametro e mostra tutti gli ordini di quel cliente.

ESERCIZIO 3 : Stored Procedure per aggiungere un nuovo ordine e calcolare il totale aggiornato

Questa procedura accetta CustomerID, OrderDate e Amount come parametri, inserisce un nuovo ordine e restituisce il totale aggiornato degli ordini per quel cliente.

ESERCIZIO 4: Funzione per ottenere il totale medio degli ordini di un cliente

Questa funzione accetta un CustomerID e restituisce il valore medio degli ordini per quel cliente.

ESERCIZIO 5: Procedura per aggiungere un nuovo cliente

Questa procedura accetta il nome del cliente, l'email e la data di iscrizione e aggiunge un nuovo record alla tabella Customers.

ESERCIZIO 6: Procedura per aggiornare l'email di un cliente

Questa procedura accetta un CustomerID e un nuovo indirizzo email per aggiornare l'indirizzo email del cliente.

ESERCIZIO 7: Procedura per eliminare tutti gli ordini di un cliente e calcolare il totale degli ordini eliminati

Questa procedura elimina tutti gli ordini di un cliente specificato e restituisce il totale dell'importo degli ordini eliminati.

ESERCIZIO 8: Costruire un trigger verifica che il valore Amount in Orders sia positivo prima di inserire un nuovo ordine.

ESERCIZIO 9: trigger che inserisce un messaggio di log in una tabella di log degli ordini ogni volta che viene creato un nuovo ordine.

ESERCIZIO 10: trigger verifica se esistono ordini per un cliente prima di eliminare il cliente dalla tabella Customers. Se esistono ordini, impedisce l'eliminazione.

ESERCIZIO 11: trigger registra un messaggio di log ogni volta che un ordine viene eliminato.

ESERCIZIO 12: trigger controlla se un aggiornamento di un ordine tenta di impostare un valore negativo per Amount. Se accade, l'aggiornamento viene bloccato.