foreign key - It establishes a relationship between two tables based on the values of the foreign key column and the primary key in the referenced table.

-- Create the department table

CREATE TABLE department (

dept\_id INT PRIMARY KEY,

dept\_name VARCHAR(50)

);

-- Create the employee table with a foreign key constraint

CREATE TABLE employee (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR(50),

dept\_id INT,

FOREIGN KEY (dept\_id) REFERENCES department(dept\_id) ON DELETE CASCADE

);

ASSIGNMENT – 1 ShoppingCartDb

-- Create database

CREATE DATABASE ShoppingCartDb;

-- Use the ShoppingCartDb database

USE ShoppingCartDb;

1. Create the following tables :

a. Create Products table :

CREATE TABLE Products (

ProductId INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(30) NOT NULL,

QuantityInStock INT NOT NULL,

UnitPrice FLOAT NOT NULL CHECK (UnitPrice > 0),

Category VARCHAR(100) NOT NULL

);

b. Create Users table :

CREATE TABLE Users (

UserId INT PRIMARY KEY AUTO\_INCREMENT,

UserName VARCHAR(50) NOT NULL,

Password VARCHAR(50) NOT NULL,

ContactNumber VARCHAR(20) NOT NULL UNIQUE,

Address VARCHAR(100) NOT NULL

);

c. Create Cart table :

CREATE TABLE Carts (

CartId INT PRIMARY KEY AUTO\_INCREMENT,

ProductId INT,

Quantity INT NOT NULL CHECK (Quantity > 0),

UserId INT NOT NULL,

FOREIGN KEY (ProductId) REFERENCES Products(ProductId) ON DELETE CASCADE,,

FOREIGN KEY (UserId) REFERENCES Users(UserId) ON DELETE CASCADE,

);

d. Create Orders table :

CREATE TABLE Orders (

OrderId INT PRIMARY KEY AUTO\_INCREMENT,

CartId INT NOT NULL,

OrderDate DATE,

UserId INT,

FOREIGN KEY (CartId) REFERENCES Cart(CartId) ON DELETE CASCADE,,

FOREIGN KEY (UserId) REFERENCES Users(UserId) ON DELETE CASCADE,

);

2. Insert 5 records in each table :

a. Insert records in the above tables :

------ Insert records into Products table ---------

INSERT INTO Products (Name, QuantityInStock, UnitPrice, Category) VALUES

('iphone 14 pro', 10, 150000.00, 'phone'),

('samsung galaxy 23', 5, 100000.00, 'phone'),

('pen', 0, 20.00, 'stationary'),

('kurti', 20, 750.00, 'cloth'),

('ring', 15, 9000.00, 'accessory');

------ Insert records into Users table -----------

INSERT INTO Users (UserName, Password, ContactNumber, Address) VALUES

('nisha', 'password1', '1234567890', 'mumbai'),

('neha', 'password2', '0987654321', 'pune'),

('abhishek', 'password3', '1112223333', 'delhi'),

('suraj', 'password4', '4445556666', 'mumbai'),

('vikram', 'password5', '7778889999', 'uttarakahnd');

------- Insert records into Cart table --------

INSERT INTO Carts (ProductId, Quantity, UserId) VALUES

(1, 2, 1),

(2, 3, 2),

(3, 1, 3),

(4, 4, 4),

(5, 2, 5);

-------- Insert records into Orders table ---------

INSERT INTO Orders (CartId, OrderDate, UserId) VALUES

(1, '2024-04-01', 1),

(2, '2024-04-02', 2),

(3, '2024-04-03', 3),

(4, '2024-04-04', 4),

(5, '2024-04-05', 5);

b. Try to verify with delete cascade and without delete cascade :

------- Attempt to delete a user without delete cascade ----------

DELETE FROM Users WHERE UserId = 1;

Without on delete cascade, this deletion operation will fail if there are related records in the Cart and Orders tables referencing the user with UserId = 1, In this case first we have to remove all record of UserId = 1 from Carts and Orders table then we can delete record from User table.

------- Attempt to delete a user delete cascade -------

DELETE FROM Users WHERE UserId = 1;

With on delete cascade, this deletion operation will automatically delete related records in the Carts and Orders tables that reference the user with UserId = 1.

3. Write the select queries on the above tables for the following requirements :

i. On Product Table :

a. Display all Products :

SELECT \* FROM Products;

b. Display Products belongs to particular Category :

SELECT \* FROM Products WHERE Category = 'phone';

c. Display out of stock products (means quantity is zero) :

SELECT \* FROM Products WHERE QuantityInStock = 0;

d. Display the products which price between 1000 to 10000 :

SELECT \* FROM Products WHERE UnitPrice BETWEEN 1000 AND 10000;

e. Display the product details based on the ProductId :

SELECT \* FROM Products WHERE ProductId = 3;

ii. On Cart Table :

a. Display data based on the given CartId :

SELECT \* FROM Cart WHERE CartId = 2;

b. Check is there any products added in Cart based on the ProductId :

SELECT \* FROM Cart WHERE ProductId = 4;

iii. On Orders table :

a. Display OrderDetails based on OrderId :

SELECT \* FROM Orders WHERE OrderId = 3;

b. Display OrderDetails based on UserId :

SELECT \* FROM Orders WHERE UserId = 1;

c. Display OrderDetails based on OrderDate :

SELECT \* FROM Orders WHERE OrderDate = '2024-04-03';