**BÁO CÁO SƠ BỘ BÀI TẬP LỚN SỐ 2 MÔN HỆ CƠ SỞ DỮ LIỆU**

**HK231 - Lớp: L0x**

**Nhóm**: <tên nhóm>

**Danh sách thành viên**:

|  |  |  |
| --- | --- | --- |
| Stt | Họ tên | MSSV |
| 1 |  |  |
| 2 |  |  |

**Chủ đề**: Trang web bán khoá học online

**1. Thiết kế ERD cập nhật mới nhất** A diagram of a diagram

Description automatically generated

**2. Trigger – Thủ tục - Hàm**

**2.1 Thủ tục INSERT / UPDATE / DELETE dữ liệu vào 1 bảng dữ liệu**

Bảng dữ liệu: Courses

**a. Thủ tục INSERT**

**Mô tả:** Thủ tục này dùng để thêm một khóa học mới vào bảng **Courses**.  
**Input:**

* @CourseName NVARCHAR(100)
* @Description NVARCHAR(255)
* @Price DECIMAL(10, 2)

**Output:** Không có

**SQL Code:**

CREATE PROCEDURE InsertCourse

@CourseName NVARCHAR(100),

@Description NVARCHAR(255),

@Price DECIMAL(10, 2)

AS

BEGIN

INSERT INTO Courses (Name, Description, Price)

VALUES (@CourseName, @Description, @Price);

END;

**b. Thủ tục UPDATE**

**Mô tả:** Thủ tục này dùng để cập nhật thông tin của một khóa học trong bảng **Courses**.  
**Input:**

* @CourseID INT
* @CourseName NVARCHAR(100)
* @Description NVARCHAR(255)
* @Price DECIMAL(10, 2)

**Output:** Không có

**SQL Code:**

CREATE PROCEDURE UpdateCourse

@CourseID INT,

@CourseName NVARCHAR(100),

@Description NVARCHAR(255),

@Price DECIMAL(10, 2)

AS

BEGIN

UPDATE Courses

SET Name = @CourseName,

Description = @Description,

Price = @Price

WHERE CourseID = @CourseID;

END;

**c. Thủ tục DELETE**

**1. Xóa một khóa học khỏi bảng Courses**

**Mô tả:** Thủ tục này dùng để xóa một khóa học khỏi bảng **Courses**.  
**Input:**

* @CourseID INT

**Output:** Không có

**SQL Code:**

CREATE PROCEDURE DeleteCourse

@CourseID INT

AS

BEGIN

DELETE FROM Courses

WHERE CourseID = @CourseID;

END;

**2. Xóa sinh viên và các dữ liệu liên quan**

**Mô tả:** Thủ tục này xóa sinh viên dựa trên StudentID và các dữ liệu liên quan (giỏ hàng, đơn hàng, đánh giá).  
**Input:**

* @StudentID INT

**Output:** Không có

**SQL Code:**

CREATE PROCEDURE DeleteStudent

@StudentID INT

AS

BEGIN

DELETE FROM Reviews WHERE StudentID = @StudentID;

DELETE FROM CartItems WHERE CartID IN (SELECT CartID FROM Carts WHERE StudentID = @StudentID);

DELETE FROM Carts WHERE StudentID = @StudentID;

DELETE FROM Orders WHERE StudentID = @StudentID;

DELETE FROM Guardians WHERE StudentID = @StudentID;

DELETE FROM Students WHERE StudentID = @StudentID;

END;

**2.2 TRIGGER**

*Lưu ý: các trigger để kiểm tra các ràng buộc dữ liệu mà có thể định nghĩa được trong câu lệnh CREATE TABLE sẽ không được tính điểm.*

*Ví dụ: kiểm tra thuộc tính MSSV có duy nhất không (Unique), kiểm tra email có đúng định dạng không (dùng REGULAR EXPRESSION hoặc Like), số điện thoại sinh viên không được để trống (Not null)*

1. Trigger 1 (hoặc Nhóm các Trigger 1)

* ***Mô tả trigger:****Trigger này dùng để tự động cập nhật tổng giá trị đơn hàng trong bảng Orders khi có sự thay đổi về sản phẩm trong bảng OrderItems. Được kích hoạt sau khi có thao tác INSERT, UPDATE, DELETE trên bảng OrderItems. Trigger này bao gồm việc tính lại tổng giá trị đơn hàng và cập nhật vào bảng Orders.*

CREATE TRIGGER trg\_UpdateOrderTotal

ON OrderItems

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

DECLARE @OrderID INT;

SELECT @OrderID = COALESCE(INSERTED.OrderID, DELETED.OrderID)

FROM INSERTED

FULL OUTER JOIN DELETED ON INSERTED.OrderID = DELETED.OrderID;

UPDATE Orders

SET TotalAmount = (

SELECT SUM(Courses.Price)

FROM OrderItems

JOIN Courses ON OrderItems.CourseID = Courses.CourseID

WHERE OrderID = @OrderID

)

WHERE OrderID = @OrderID;

END;

Trigger 2

* ***Mô tả trigger:****Trigger này dùng để ghi lại lịch sử chỉnh sửa khóa học vào bảng Edits khi có cập nhật trên bảng Courses. Được kích hoạt sau khi UPDATE trên bảng Courses. Trigger bao gồm việc lưu thông tin khóa học đã được thay đổi, ai thay đổi và thời gian thay đổi.*

CREATE TRIGGER trg\_LogCourseEdits

ON Courses

AFTER UPDATE

AS

BEGIN

INSERT INTO Edits (CourseID, AdminID, EditTime)

SELECT

INSERTED.CourseID,

SYSTEM\_USER,

GETDATE()

FROM INSERTED;

END;

Trigger 3

* ***Mô tả trigger:***Cập nhật trạng thái đơn hàng tự động. Trigger này tự động chuyển trạng thái đơn hàng thành "Đã thanh toán" khi có bản ghi thanh toán mới.

CREATE TRIGGER trg\_UpdateOrderStatusOnPayment

ON Payments

AFTER INSERT

AS

BEGIN

UPDATE Orders

SET Status = 'Paid'

WHERE OrderID IN (SELECT OrderID FROM inserted);

END;

Trigger 4

* ***Mô tả trigger:***Tự động cập nhật trạng thái của khóa học. Trigger tự động cập nhật trạng thái của khóa học trong bảng Courses thành "Finished" khi EndDate của khóa học đã qua ngày hiện tại.

CREATE TRIGGER trg\_UpdateCourseStatus

ON Courses

AFTER UPDATE

AS

BEGIN

UPDATE Courses

SET Status = 'Finished'

WHERE EndDate < GETDATE() AND Status <> 'Finished';

END;

**2.3 THỦ TỤC**

1. Thủ tục 1

**Mô tả thủ tục:** Thủ tục này dùng để lấy danh sách các khóa học trong giỏ hàng của một người dùng. Sử dụng khi cần hiển thị các khóa học mà người dùng đã thêm vào giỏ hàng. Thủ tục này bao gồm việc truy vấn bảng CartItems và Courses để lấy thông tin khóa học.

**Input:**

* @UserID INT

**Output:**

* Danh sách các khóa học trong giỏ hàng của người dùng
* GO
* CREATE PROCEDURE GetCoursesInCart
* @UserID INT
* AS
* BEGIN
* SELECT Courses.\*
* FROM CartItems
* JOIN Carts ON CartItems.CartID = Carts.CartID
* JOIN Courses ON CartItems.CourseID = Courses.CourseID
* WHERE Carts.StudentID = @UserID;
* END;
* GO
* EXEC GetCoursesInCart @UserID = 1;

1. Thủ tục 2

**Mô tả thủ tục:** Thủ tục này dùng để lấy danh sách các khóa học trong một đơn đặt hàng. Sử dụng khi cần hiển thị các khóa học mà người dùng đã mua trong một đơn đặt hàng cụ thể. Thủ tục này bao gồm việc truy vấn bảng OrderItems và Courses để lấy thông tin khóa học.

**Input:**

* @OrderID INT

**Output:**

* Danh sách các khóa học trong đơn đặt hàng

1. Thủ tục 3

**Mô tả thủ tục:**  **Thủ tục trả về danh sách các khóa học thuộc một danh mục nhất định.  
Input:**

* **@CategoryID INT**

**Output: Danh sách các khóa học trong danh mục.**

**SQL Code:**

CREATE PROCEDURE GetCoursesByCategory

@CategoryID INT

AS

BEGIN

SELECT CourseID, Name, Description, Price, Status

FROM Courses

WHERE CategoryID = @CategoryID;

END;

**2.4 HÀM**

1. Hàm 1

Mô tả hàm: Hàm này dùng để tính tổng số lượng khóa học trong một đơn đặt hàng. Sử dụng khi cần biết tổng số lượng khóa học mà người dùng đã mua trong một đơn đặt hàng cụ thể. Hàm này bao gồm việc truy vấn bảng OrderItems để tính tổng số lượng khóa học.

Input:

@OrderID INT

Output:

Tổng số lượng khóa học trong đơn đặt hàng.

CREATE FUNCTION GetTotalCoursesInOrder

(

    @OrderID INT

)

RETURNS INT

AS

BEGIN

    DECLARE @TotalCourses INT;

    SELECT @TotalCourses = COUNT(\*)

    FROM OrderItems

    WHERE OrderID = @OrderID;

    RETURN @TotalCourses;

END;

GO

SELECT dbo.GetTotalCoursesInOrder(1);

1. Hàm 2

***Mô tả hàm***: Hàm này dùng để lấy tổng giá trị của một đơn đặt hàng. Sử dụng khi cần biết tổng giá trị mà người dùng phải trả cho một đơn đặt hàng cụ thể. Hàm này bao gồm việc truy vấn bảng OrderItems và Courses để tính tổng giá trị đơn đặt hàng.

*Input:*

*@OrderID INT*

*Output:*

*Tổng giá trị của đơn đặt hàng*

CREATE FUNCTION GetTotalOrderValue

(

    @OrderID INT

)

RETURNS DECIMAL(18, 2)

AS

BEGIN

    DECLARE @TotalValue DECIMAL(18, 2);

    SELECT @TotalValue = SUM(Courses.Price)

    FROM OrderItems

    JOIN Courses ON OrderItems.CourseID = Courses.CourseID

    WHERE OrderID = @OrderID;

    RETURN @TotalValue;

END;

**Hàm 3**

***Mô tả hàm:*** Lấy danh sách sản phẩm theo giá. Hàm trả về danh sách sản phẩm có giá nằm trong khoảng được chỉ định.

**Input:**

* @MinPrice DECIMAL(10,2)
* @MaxPrice DECIMAL(10, 2)

**Output:**

* Danh sách các khóa học trong mức giá

CREATE FUNCTION GetProductsByPriceRange(

@MinPrice DECIMAL(10, 2),

@MaxPrice DECIMAL(10, 2)

)

RETURNS TABLE

AS

RETURN (

SELECT CourseID, Name, Price

FROM Courses

WHERE Price BETWEEN @MinPrice AND @MaxPrice

);

SELECT \* FROM dbo.GetProductsByPriceRange(10, 40);

**Hàm 4**

***Mô tả hàm:*** Tính số lượng khoá học trong một danh mục. Hàm này trả về số lượng khoá học thuộc một danh mục cụ thể.

**Input:**

* @ CategoryID INT

**Output:**

* Số lượng khóa học thuộc cùng danh mục

CREATE FUNCTION CountCoursesInCategory(@CategoryID INT)

RETURNS INT

AS

BEGIN

RETURN (SELECT COUNT(\*) FROM Courses WHERE CategoryID = @CategoryID);

END;

SELECT dbo.CountCoursesInCategory(1) AS TotalCourses;