

*Lập trình toàn năng*

*Training Assignments*

|  |  |
| --- | --- |
| **Program Code** | **DTN2503** |
| **Issue/Revision** |  |
| **Effective date** |  |
| **Author** | **Huu Nguyen Van** |

**Bài tập cuối học phần 1**

Database để quản lý điểm của học sinh như dưới (trường có dấu gạch chân là PrimaryKey)

Student(ID,Name,Age,Gender)

Subject(ID, Name)

StudentSubject(StudentID,SubjectID,Mark,Date)

**Question 1:** Tạo table với các ràng buộc và kiểu dữ liệu, thêm ít nhất 3 bản ghi vào table

Trả lời:

CREATE TABLE Student (

ID INT PRIMARY KEY,

Name VARCHAR(100),

Age INT,

Gender TINYINT -- 0 = Male, 1 = Female, NULL = Unknown

);

CREATE TABLE Subject (

ID INT PRIMARY KEY,

Name VARCHAR(100)

);

CREATE TABLE StudentSubject (

StudentID INT,

SubjectID INT,

Mark DECIMAL(5,2),

Date DATE,

PRIMARY KEY (StudentID, SubjectID, Date),

FOREIGN KEY (StudentID) REFERENCES Student(ID),

FOREIGN KEY (SubjectID) REFERENCES Subject(ID)

);

INSERT INTO Student (ID, Name, Age, Gender) VALUES

(1, 'Alice', 18, 1),

(2, 'Bob', 19, 0),

(3, 'Charlie', 20, NULL);

INSERT INTO Subject (ID, Name) VALUES

(1, 'Math'),

(2, 'Physics'),

(3, 'Chemistry');

INSERT INTO StudentSubject (StudentID, SubjectID, Mark, Date) VALUES

(1, 1, 9.0, '2025-04-01'),

(2, 1, 8.5, '2025-03-01'),

(1, 2, 7.5, '2025-02-01');

**Question 2:** Viết lệnh để

a) Lấy tất cả các môn học không có bất kì điểm nào.

Trả lời:

SELECT s.ID, s.Name

FROM Subject s

LEFT JOIN StudentSubject ss ON s.ID = ss.SubjectID

WHERE ss.SubjectID IS NULL;

b) Lấy danh sách các môn học có ít nhất 2 điểm.

Trả lời:

SELECT s.ID, s.Name

FROM Subject s

JOIN StudentSubject ss ON s.ID = ss.SubjectID

GROUP BY s.ID, s.Name

HAVING COUNT(1) >= 2;

**Question 3:** Tạo view có tên là "StudentInfo" lấy các thông tin về học sinh bao gồm:

Student ID,Subject ID, Student Name, Student Age, Student Gender, Subject Name, Mark, Date

(Với cột Gender show 'Male' để thay thế cho 0, 'Female' thay thế cho 1 và

'Unknow' thay thế cho null)

Trả lời:

CREATE VIEW StudentInfo AS

SELECT

s.ID AS StudentID, sub.ID AS SubjectID, s.Name AS StudentName, s.Age AS StudentAge,

CASE

WHEN s.Gender = 0 THEN 'Male' WHEN s.Gender = 1 THEN 'Female' ELSE 'Unknown'

END AS Gender,

sub.Name AS SubjectName, ss.Mark, ss.Date

FROM Student s

JOIN StudentSubject ss ON s.ID = ss.StudentID

JOIN Subject sub ON sub.ID = ss.SubjectID;

**Question 4:**. Không sử dụng On Update Cascade & On Delete Cascade

a) Tạo trigger cho table Subject có tên là SubjectUpdateID:

Khi thay đổi data của cột ID của table Subject, thì giá trị tương ứng với cột SubjectID của table StudentSubject cũng thay đổi theo

**Trả lời:**

DELIMITER $$

CREATE TRIGGER SubjectUpdateID

BEFORE UPDATE ON Subject

FOR EACH ROW

BEGIN

UPDATE StudentSubject

SET SubjectID = NEW.ID

WHERE SubjectID = OLD.ID;

END$$

DELIMITER ;

b) Tạo trigger cho table StudentSubject có tên là StudentDeleteID:

Khi xóa data của cột ID của table Student, thì giá trị tương ứng với cột StudentID của table StudentSubject cũng bị xóa theo.

**Trả lời:**

DELIMITER $$

CREATE TRIGGER StudentDeleteID

BEFORE DELETE ON Student

FOR EACH ROW

BEGIN

DELETE FROM StudentSubject

WHERE StudentID = OLD.ID;

END$$

DELIMITER ;

**Question 5**: Viết 1 store procedure (có 2 parameters: student name) sẽ xóa tất cả các thông tin liên quan tới học sinh có cùng tên như parameter.

Trong trường hợp nhập vào student name = "\*" thì procedure sẽ xóa tất cả các học sinh

**Trả lời:**

DELIMITER $$

CREATE PROCEDURE DeleteStudentByName(IN input\_name VARCHAR(100))

BEGIN

IF input\_name = '\*' THEN

DELETE FROM StudentSubject;

DELETE FROM Student;

ELSE

DELETE FROM StudentSubject

WHERE StudentID IN (

SELECT ID FROM Student WHERE Name = input\_name

);

DELETE FROM Student

WHERE Name = input\_name;

END IF;

END$$

DELIMITER ;