

TRƯỜNG ĐẠI HỌC TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
TRUNG TÂM PHÁT TRIỂN CÔNG NGHỆ THÔNG TIN

BÀI TẬP THỰC HÀNH 5

Truy vấn thông tin (XPath, XQuery)



Mã Môn:	IE103
Tên Môn:	Quản Lý Thông Tin (Data Management)
Lớp:	LTK2025.2.CNTT
Giảng Viên:	Ths. Nguyễn Thành Luân
Thực Hiện:	Nhóm 02 (Omega)

Tuần 5

MỤC LỤC

Trang Bìa (Bài Tập Thực Hành 5)	i
Mục Lục	ii
Danh Sách Bảng	ii
Danh Sách Hình Ảnh	iii
Danh Sách Mã Nguồn	iii
Nhóm 02 (Omega)	iv
1. Bài 1. Chuẩn bị	1
1.1. Tạo Database và các bảng	1
1.2. Bảng KhoaHoc	1
1.3. Bảng SinhVien	1
1.4. Bảng MonHoc	2
1.5. Bảng KhoaHocMonHoc	2
1.6. Bảng Diem	3
1.7. Bảng QuanLySV (chứa dữ liệu XML)	3
1.8. Tổng kết phần chuẩn bị	5
2. Bài 2. Thực hành	6
2.1. Giới thiệu về XPath và XQuery	6
2.2. Câu 1: XPath lấy Sinh viên theo ID và vị trí	6
2.3. Câu 2: Lấy tất cả các nút con từ nút gốc	7
2.4. Câu 3: XQuery với điều kiện WHERE	7
2.5. Câu 4: XQuery với sắp xếp (ORDER BY)	8
2.6. Câu 5: Tạo XML từ dữ liệu quan hệ	8
2.7. Câu 6: Xóa thuộc tính bằng XQuery	8
2.8. Câu 7: XQuery với điều kiện OR	9
2.9. Câu 8: Thay thế giá trị thuộc tính	9
2.10. Câu 9: Kiểm tra sự tồn tại với exist()	10
2.11. Câu 10: Kiểm tra và INSERT dữ liệu XML	10
2.12. Câu 11: Phép nối (Join) với XQuery	11
2.13. Tổng kết	12

DANH SÁCH BẢNG

DANH SÁCH HÌNH ẢNH

DANH SÁCH MÃ NGUỒN

Mã nguồn 1	Bài 1. Tạo và sử dụng database	1
Mã nguồn 2	Bài 1. Tạo bảng KhoaHoc và chèn dữ liệu	1
Mã nguồn 3	Bài 1. Tạo bảng SinhVien và chèn dữ liệu	1
Mã nguồn 4	Bài 1. Tạo bảng MonHoc và chèn dữ liệu	2
Mã nguồn 5	Bài 1. Tạo bảng KhoaHocMonHoc và chèn dữ liệu	2
Mã nguồn 6	Bài 1. Tạo bảng Diem và chèn dữ liệu	3
Mã nguồn 7	Bài 1. Tạo bảng QuanLySV	3
Mã nguồn 8	Bài 1. Chèn dữ liệu XML cho trường DH CNTT	4
Mã nguồn 9	Bài 1. Chèn dữ liệu XML cho trường DH KHTN	5
Mã nguồn 10	Bài 2. XPath lấy sinh viên có ID=10	6
Mã nguồn 11	Bài 2. XPath lấy sinh viên ở vị trí cuối cùng	7
Mã nguồn 12	Bài 2. Lấy tất cả các nút con từ THONGTINSV	7
Mã nguồn 13	Bài 2. XQuery với điều kiện WHERE	7
Mã nguồn 14	Bài 2. XQuery với ORDER BY	8
Mã nguồn 15	Bài 2. Định dạng XML mẫu của Câu 5	8
Mã nguồn 16	Bài 2. Tạo XML từ dữ liệu quan hệ	8
Mã nguồn 17	Bài 2. Xóa thuộc tính bằng XQuery	9
Mã nguồn 18	Bài 2. XQuery với điều kiện OR	9
Mã nguồn 19	Bài 2. Thay thế giá trị thuộc tính	9
Mã nguồn 20	Bài 2. Kiểm tra sự tồn tại với exist()	10
Mã nguồn 21	Bài 2. Kiểm tra sinh viên tên ‘Lan’	10
Mã nguồn 22	Bài 2. Dữ liệu mẫu cần chèn	10
Mã nguồn 23	Bài 2. INSERT phần tử mới vào XML	11
Mã nguồn 24	Bài 2. Phép nối với XQuery	11
Mã nguồn 25	Bài 2. Kết quả mẫu của Câu 11	12

NHÓM 02 (OMEGA)

Thành viên của nhóm và bảng nhiệm vụ.

No	MSSV	Họ và Tên
01	25410291	Đinh Xuân Sâm
02	25410319	Đặng Hữu Toàn
03	25410321	Nguyễn Điền Triết
04	25410204	Trương Xuân Hậu
05	25410338	Lê Anh Vũ
06	25410176	Trần Sơn Bình
07	25410247	Lê Kim Long
08	25410337	La Anh Vũ
09	25410209	Lê Ngọc Hiệp
10	25410271	Nguyễn Thị Ngọc Nhung

Nhóm 02 (Omega)

BÀI 1. CHUẨN BỊ

Trước khi thực hành với XPath và XQuery, chúng ta cần chuẩn bị cơ sở dữ liệu và dữ liệu mẫu.

1.1. Tạo Database và các bảng

Thực hiện các lệnh sau để tạo database TEST_XML và các bảng cần thiết:

```
CREATE DATABASE TEST_XML
```

```
GO
```

```
USE TEST_XML
```

```
GO
```

Mã nguồn 1: Bài 1. Tạo và sử dụng database

1.2. Bảng KhoaHoc

```
-- KhoaHoc
```

```
CREATE TABLE KhoaHoc
```

```
(
```

```
    MaKhoaHoc INT IDENTITY(1,1) NOT NULL,
```

```
    TenKhoaHoc VARCHAR(200) NOT NULL,
```

```
    CONSTRAINT PK_KhoaHoc PRIMARY KEY(MaKhoaHoc)
```

```
)
```

```
INSERT INTO KhoaHoc (TenKhoaHoc) SELECT 'Mang May Tinh Truyen Thong'
```

```
INSERT INTO KhoaHoc (TenKhoaHoc) SELECT 'Khoa Hoc May Tinh'
```

```
INSERT INTO KhoaHoc (TenKhoaHoc) SELECT 'Ky Thuat May Tinh'
```

Mã nguồn 2: Bài 1. Tạo bảng KhoaHoc và chèn dữ liệu

1.3. Bảng SinhVien

```
-- SinhVien
```

```
CREATE TABLE SinhVien
```

```
(
```

```
    MSSV BIGINT IDENTITY(1,1)NOT NULL CONSTRAINT PK_SinhVien PRIMARY KEY(MSSV),
```

```
    TenSV VARCHAR(200) NOT NULL,
```

```
    MaKhoaHoc INT NOT NULL CONSTRAINT FK_SinhVien_MaKhoaHoc FOREIGN KEY
```

```
        REFERENCES KhoaHoc(MaKhoaHoc)
```

```
)
```

```
INSERT INTO SinhVien SELECT 'Anh',1
```

```
INSERT INTO SinhVien SELECT 'Son',2
```

```
INSERT INTO SinhVien SELECT 'Thuy',3
```

Mã nguồn 3: Bài 1. Tạo bảng SinhVien và chèn dữ liệu

1.4. Bảng MonHoc

```
-- MonHoc
CREATE TABLE MonHoc
(
    MaMonHoc INT IDENTITY NOT NULL CONSTRAINT PK_MonHoc PRIMARY
    KEY(MaMonHoc),
    TenMonHoc VARCHAR(200)
)
INSERT INTO MonHoc (TenMonHoc) SELECT ('Co So Du Lieu')
INSERT INTO MonHoc (TenMonHoc) SELECT ('Cau Truc Du Lieu')
INSERT INTO MonHoc (TenMonHoc) SELECT ('Lap Trinh Di Dong')
INSERT INTO MonHoc (TenMonHoc) SELECT ('Toan Giai Tich')
INSERT INTO MonHoc (TenMonHoc) SELECT ('Lap Trinh Java')
INSERT INTO MonHoc (TenMonHoc) SELECT ('He Quan Tri CSDL')
INSERT INTO MonHoc (TenMonHoc) SELECT ('Anh Van')
INSERT INTO MonHoc (TenMonHoc) SELECT ('Thiet Ke Web ')
INSERT INTO MonHoc (TenMonHoc) SELECT ('An Toan Thong Tin')
```

Mã nguồn 4: Bài 1. Tạo bảng MonHoc và chèn dữ liệu

1.5. Bảng KhoaHocMonHoc

```
-- KhoaHocMonHoc
CREATE TABLE KhoaHocMonHoc
(
    MaKhoaHoc INT CONSTRAINT FK_KhoaHocMonHoc_MaKhoaHoc FOREIGN KEY
    REFERENCES
        KhoaHoc(MaKhoaHoc),
    MaMonHoc INT CONSTRAINT FK_KhoaHocMonHoc_MaMonHoc FOREIGN KEY
    REFERENCES
        MonHoc(MaMonHoc)
)
INSERT INTO KhoaHocMonHoc (MaKhoaHoc,MaMonHoc) SELECT 1,1
INSERT INTO KhoaHocMonHoc (MaKhoaHoc,MaMonHoc) SELECT 1,2
INSERT INTO KhoaHocMonHoc (MaKhoaHoc,MaMonHoc) SELECT 1,3
INSERT INTO KhoaHocMonHoc (MaKhoaHoc,MaMonHoc) SELECT 2,4
INSERT INTO KhoaHocMonHoc (MaKhoaHoc,MaMonHoc) SELECT 2,5
INSERT INTO KhoaHocMonHoc (MaKhoaHoc,MaMonHoc) SELECT 2,6
INSERT INTO KhoaHocMonHoc (MaKhoaHoc,MaMonHoc) SELECT 3,7
INSERT INTO KhoaHocMonHoc (MaKhoaHoc,MaMonHoc) SELECT 3,8
INSERT INTO KhoaHocMonHoc (MaKhoaHoc,MaMonHoc) SELECT 3,9
```

Mã nguồn 5: Bài 1. Tạo bảng KhoaHocMonHoc và chèn dữ liệu

1.6. Bảng Diem

```
-- Diem
CREATE TABLE Diem
(
    MSSV BIGINT CONSTRAINT FK_Diem_MSSV FOREIGN KEY REFERENCES SinhVien(MSSV),
    MaMonHoc INT CONSTRAINT FK_Diem_MaMonHoc FOREIGN KEY REFERENCES
    MonHoc(MaMonHoc),
    Diem INT
)
INSERT INTO Diem (MSSV,MaMonHoc,Diem) SELECT 1,1,75
INSERT INTO Diem (MSSV,MaMonHoc,Diem) SELECT 1,2,80
INSERT INTO Diem (MSSV,MaMonHoc,Diem) SELECT 1,3,70
INSERT INTO Diem (MSSV,MaMonHoc,Diem) SELECT 2,4,80
INSERT INTO Diem (MSSV,MaMonHoc,Diem) SELECT 2,5,80
INSERT INTO Diem (MSSV,MaMonHoc,Diem) SELECT 2,6,90
INSERT INTO Diem (MSSV,MaMonHoc,Diem) SELECT 3,7,80
INSERT INTO Diem (MSSV,MaMonHoc,Diem) SELECT 3,8,80
INSERT INTO Diem (MSSV,MaMonHoc,Diem) SELECT 3,9,90
```

Mã nguồn 6: Bài 1. Tạo bảng Diem và chèn dữ liệu

1.7. Bảng QuanLySV (chứa dữ liệu XML)

Bảng quan trọng nhất cho bài tập này là `QuanLySV`, chứa cột `ChiTietSV` kiểu XML.

```
-- QuanLySV
CREATE TABLE QuanLySV
(
    MSDH INT NOT NULL,
    TenDH VARCHAR(20),
    ChiTietSV XML
)
```

Mã nguồn 7: Bài 1. Tạo bảng QuanLySV

1.7.1. Chèn dữ liệu XML cho trường DH CNTT

```
INSERT INTO QuanLySV VALUES (
    1,
    'DH CNTT',
    '<THONGTINSV>
        <sinhvien ID="10" Ten="Nam">
            <monhoc ID="1" Ten="Co So Du Lieu" />
            <monhoc ID="2" Ten="Cau Truc Du Lieu" />
            <monhoc ID="3" Ten="Lap Trinh Mobile" />
        </sinhvien>
        <sinhvien ID="11" Ten="An">
            <monhoc ID="4" Ten="Toan Giai Tich" />
            <monhoc ID="5" Ten="Lap Trinh Java" />
            <monhoc ID="6" Ten="He Quan Tri CSDL" />
        </sinhvien>
        <sinhvien ID="12" Ten="Thanh">
            <monhoc ID="7" Ten="Anh Van" />
            <monhoc ID="8" Ten="Thiet Ke Web" />
            <monhoc ID="9" Ten="An Toan Thong Tin" />
        </sinhvien>
    </THONGTINSV>'
)
```

Mã nguồn 8: Bài 1. Chèn dữ liệu XML cho trường DH CNTT

1.7.2. Chèn dữ liệu XML cho trường DH KHTN

```
INSERT INTO QuanLySV VALUES (
    2,
    'DH KHTN',
    '<THONGTINSV>
        <sinhvien ID="10" Ten="Khang">
            <monhoc ID="1" Ten="Co So Du Lieu" />
            <monhoc ID="2" Ten="Cau Truc Du Lieu" />
            <monhoc ID="3" Ten="Lap Trinh Mobile" />
        </sinhvien>
        <sinhvien ID="11" Ten="Vinh">
            <monhoc ID="4" Ten="Toan Giai Tich" />
            <monhoc ID="5" Ten="Lap Trinh Java" />
            <monhoc ID="6" Ten="He Quan Tri CSDL" />
        </sinhvien>
        <sinhvien ID="12" Ten="Hoa">
            <monhoc ID="7" Ten="Anh Van" />
            <monhoc ID="8" Ten="Thiet Ke Web" />
            <monhoc ID="9" Ten="An Toan Thong Tin" />
        </sinhvien>
    </THONGTINSV>
)
```

Mã nguồn 9: Bài 1. Chèn dữ liệu XML cho trường DH KHTN

1.8. Tổng kết phần chuẩn bị

Sau khi hoàn thành các bước trên, chúng ta đã có:

- Database TEST_XML với các bảng quan hệ: KhoaHoc, SinhVien, MonHoc, KhoaHocMonHoc, Diem.
- Bảng QuanLySV chứa dữ liệu XML trong cột ChiTietSV, bao gồm thông tin về sinh viên và môn học của hai trường: DH CNTT và DH KHTN.

Dữ liệu XML trong bảng QuanLySV sẽ được sử dụng để thực hành các câu lệnh XPath và XQuery trong phần tiếp theo.

BÀI 2. THỰC HÀNH

Hãy sử dụng XPath, XQuery để thực hiện các yêu cầu sau:

1. Câu 1: Viết lệnh XPath lấy Sinh viên có ID=10 và lệnh XPath lấy sinh viên ở vị trí cuối cùng ở trường CNTT.
2. Câu 2: Viết lệnh trả về tất cả các nút từ nút gốc là THONGTINSV.
3. Câu 3: Viết lệnh XQuery trả về danh sách sinh viên có ID < 12 với MSDH = 1.
4. Câu 4: Viết lệnh XQuery trả về danh sách sinh viên sắp xếp theo tên với MSDH=2.
5. Câu 5: Viết lệnh XQuery trả về MSDH và TenDH theo định dạng XML.
6. Câu 6: Viết lệnh XQuery xóa tên các sinh viên trường DH KHTN.
7. Câu 7: Viết lệnh XQuery trả về thông tin các sinh viên có tên là ‘Nam’ hoặc ‘Thanh’.
8. Câu 8: Viết lệnh XQuery thay đổi tên sinh viên thứ 2 thành tên ‘Binh’ trong trường CNTT.
9. Câu 9: Viết lệnh XQuery kiểm tra xem có tồn tại sinh viên có ID là 12 trong trường KHTN không? (Nếu có trả về 1, nếu không thì trả về 0).
10. Câu 10: Viết lệnh XQuery kiểm tra xem có tồn tại sinh viên tên ‘Lan’ trong trường CNTT không? (Nếu có trả về 1, nếu không thì trả về 0). Và INSERT thêm vào THONGTINSV.
11. Câu 11: Thực hiện phép nối bằng các lệnh XQuery như một câu truy vấn trong SQL để trả về thông tin là sinh viên nào học khóa học tên là gì, sinh viên nào học môn học gì?

2.1. Giới thiệu về XPath và XQuery

XPath và XQuery là các ngôn ngữ truy vấn XML được sử dụng để truy xuất và thao tác dữ liệu XML trong SQL Server.

- **XPath:** Ngôn ngữ để định vị và chọn các phần tử trong tài liệu XML.
- **XQuery:** Ngôn ngữ truy vấn XML mạnh mẽ, cho phép truy vấn, chuyển đổi và xây dựng tài liệu XML.

2.2. Câu 1: XPath lấy Sinh viên theo ID và vị trí

2.2.1. XPath lấy Sinh viên có ID=10

Sử dụng biểu thức XPath với điều kiện lọc theo thuộc tính `@ID`.

```
-- Xpath lấy Sinh viên có ID=10
SELECT ChiTietSV.query('/THONGTINSV/sinhvien[@ID="10"]') AS KetQua
FROM QuanLySV WHERE TenDH = 'DH CNTT'
GO
```

Mã nguồn 10: Bài 2. XPath lấy sinh viên có ID=10

Giải thích:

- `/THONGTINSV/sinhvien[@ID="10"]` : Đường dẫn XPath tìm phần tử `sinhvien` có thuộc tính `ID` bằng “10”.
- `query()` : Phương thức để thực thi biểu thức XPath/XQuery trên cột XML.

2.2.2. XPath lấy sinh viên ở vị trí cuối cùng

Sử dụng hàm `last()` để lấy phần tử cuối cùng trong tập hợp.

```
-- Xpath lấy sinh viên ở vị trí cuối cùng ở trường CNTT  
SELECT ChiTietSV.query('/THONGTINSV/sinhvien[last()]') AS KetQua  
FROM QuanLySV WHERE TenDH = 'DH CNTT'  
GO
```

Mã nguồn 11: Bài 2. XPath lấy sinh viên ở vị trí cuối cùng

Giải thích:

- `[last()]` : Hàm XPath trả về phần tử cuối cùng trong tập hợp các phần tử `sinhvien`.

2.3. Câu 2: Lấy tất cả các nút con từ nút gốc

Sử dụng ký tự đại diện `*` để lấy tất cả các phần tử con trực tiếp của nút gốc `THONGTINSV`.

```
SELECT ChiTietSV.query('/THONGTINSV/*') as KetQua  
FROM QuanLySV  
GO
```

Mã nguồn 12: Bài 2. Lấy tất cả các nút con từ THONGTINSV

Giải thích:

- `/THONGTINSV/*` : Ký tự `*` đại diện cho tất cả các phần tử con trực tiếp của `THONGTINSV`.
- Kết quả trả về tất cả các phần tử `sinhvien` trong XML.

2.4. Câu 3: XQuery với điều kiện WHERE

Sử dụng cú pháp FLWOR (For-Let-Where-Order by-Return) của XQuery để lọc dữ liệu.

```
SELECT ChiTietSV.query(  
    for $sv in /THONGTINSV/sinhvien  
    where $sv/@ID < 12  
    return $sv  
) AS KetQua  
FROM QuanLySV WHERE MSDH = 1  
GO
```

Mã nguồn 13: Bài 2. XQuery với điều kiện WHERE

Giải thích:

- `for $sv in /THONGTINSV/sinhvien` : Duyệt qua tất cả các phần tử `sinhvien`.
- `where $sv/@ID < 12` : Lọc các sinh viên có ID nhỏ hơn 12.
- `return $sv` : Trả về phần tử sinh viên thỏa mãn điều kiện.

2.5. Câu 4: XQuery với sắp xếp (ORDER BY)

Sử dụng mệnh đề `order by` trong XQuery để sắp xếp kết quả.

```
SELECT ChiTietSV.query('
for $sv in /THONGTINSV/sinhvien
order by $sv/@Ten
return $sv
') as KetQua
FROM QuanLySV WHERE MSDH = 2
GO
```

Mã nguồn 14: Bài 2. XQuery với ORDER BY

Giải thích:

- `order by $sv/@Ten` : Sắp xếp các sinh viên theo thuộc tính `Ten` (tên).
- Kết quả được sắp xếp theo thứ tự bảng chữ cái.

2.6. Câu 5: Tạo XML từ dữ liệu quan hệ

Viết lệnh XQuery trả về MSDH và TenDH theo định dạng sau:

```
<!-- Dữ liệu mẫu -->
<QuanLySV>
  <ChiTietSV>1 DH CNTT</ChiTietSV>
</QuanLySV>
```

Mã nguồn 15: Bài 2. Định dạng XML mẫu của Câu 5

Sử dụng `FOR XML PATH` để tạo cấu trúc XML từ dữ liệu quan hệ.

```
SELECT
  CAST(MSDH AS VARCHAR) + ' ' + TenDH AS ChiTietSV
FROM QuanLySV
FOR XML PATH('QuanLySV')
GO
```

Mã nguồn 16: Bài 2. Tạo XML từ dữ liệu quan hệ

Giải thích:

- `FOR XML PATH('QuanLySV')` : Tạo cấu trúc XML với phần tử gốc là `QuanLySV`.
- `CAST(MSDH AS VARCHAR) + ' ' + TenDH` : Nối MSDH và TenDH thành một chuỗi.

2.7. Câu 6: Xóa thuộc tính bằng XQuery

Sử dụng phương thức `modify()` với lệnh `delete` để xóa thuộc tính.

```

UPDATE QuanLySV
SET ChiTietSV.modify('delete /THONGTINSV/sinhvien/@Ten')
WHERE TenDH = 'DH KHTN'
GO

```

Mã nguồn 17: Bài 2. Xóa thuộc tính bằng XQuery

Giải thích:

- `modify('delete ...')` : Phương thức để sửa đổi XML.
- `/THONGTINSV/sinhvien/@Ten` : Đường dẫn đến thuộc tính `Ten` của tất cả các phần tử `sinhvien`.
- Sau khi thực thi, tất cả các thuộc tính `Ten` của sinh viên trong trường DH KHTN sẽ bị xóa.

2.8. Câu 7: XQuery với điều kiện OR

Sử dụng toán tử logic `or` trong mệnh đề `where` để lọc dữ liệu theo nhiều điều kiện.

```

SELECT ChiTietSV.query(
  for $s in /THONGTINSV/sinhvien
  where $s/@Ten = "Nam" or $s/@Ten = "Thanh"
  return $s
) as KetQua
FROM QuanLySV
GO

```

Mã nguồn 18: Bài 2. XQuery với điều kiện OR

Giải thích:

- `where $s/@Ten = "Nam" or $s/@Ten = "Thanh"` : Lọc các sinh viên có tên là “Nam” hoặc “Thanh”.
- Kết quả trả về tất cả các phần tử `sinhvien` thỏa mãn một trong hai điều kiện.

2.9. Câu 8: Thay thế giá trị thuộc tính

Sử dụng `replace value of` để thay đổi giá trị của thuộc tính tại vị trí cụ thể.

```

UPDATE QuanLySV
SET ChiTietSV.modify('replace value of [/THONGTINSV/sinhvien](2)/@Ten with "Binh")'
WHERE TenDH = 'DH CNTT'
GO

```

Mã nguồn 19: Bài 2. Thay thế giá trị thuộc tính

Giải thích:

- `[/THONGTINSV/sinhvien](2)` : Chọn phần tử `sinhvien` thứ 2 (chỉ số bắt đầu từ 1).
- `replace value of ... with "Binh"` : Thay thế giá trị của thuộc tính `Ten` bằng “Binh”.

2.10. Câu 9: Kiểm tra sự tồn tại với exist()

Phương thức `exist()` trả về giá trị boolean (1 nếu tồn tại, 0 nếu không tồn tại).

```
SELECT ChiTietSV.exist('/THONGTINSV/sinhvien[@ID="12"]') as KetQua  
FROM QuanLySV WHERE TenDH = 'DH KHTN'  
GO
```

Mã nguồn 20: Bài 2. Kiểm tra sự tồn tại với exist()

Giải thích:

- `exist()` : Phương thức kiểm tra sự tồn tại của phần tử hoặc điều kiện trong XML.
- Trả về 1 nếu tìm thấy sinh viên có ID=12, 0 nếu không tìm thấy.

2.11. Câu 10: Kiểm tra và INSERT dữ liệu XML

Viết lệnh XQuery kiểm tra xem có tồn tại sinh viên tên ‘Lan’ trong trường CNTT không? (Nếu có trả về 1, nếu không thì trả về 0). Và INSERT thêm vào THONGTINSV:

2.11.1. Phần 1: Kiểm tra sự tồn tại

```
-- Xquery kiểm tra xem có tồn tại sinh viên tên 'Lan' trong trường CNTT  
SELECT ChiTietSV.exist('/THONGTINSV/sinhvien[@Ten="Lan"]') as KetQua  
FROM QuanLySV WHERE TenDH = 'DH CNTT'
```

Mã nguồn 21: Bài 2. Kiểm tra sinh viên tên ‘Lan’

Giải thích:

- `exist()` : Phương thức kiểm tra sự tồn tại của phần tử hoặc điều kiện trong XML.
- Trả về 1 nếu tìm thấy sinh viên tên ‘Lan’, 0 nếu không tìm thấy.

2.11.2. Phần 2: INSERT phần tử mới vào XML

Dữ liệu mẫu cần chèn:

```
<!-- Dữ liệu mẫu -->  
<sinhvien ID="15" Ten="Lan">  
  <monhoc ID="10" Ten="Toan Roi Rac" />  
  <monhoc ID="11" Ten="Lap Trinh C#" />  
  <monhoc ID="12" Ten="CSDL Nang Cao" />  
</sinhvien>
```

Mã nguồn 22: Bài 2. Dữ liệu mẫu cần chèn

Sử dụng lệnh `insert` với các tùy chọn vị trí: `as first` , `as last` , `before` , `after` .

```
-- INSERT dữ liệu mẫu vào THONGTINSV
UPDATE QuanLySV
SET ChiTietSV.modify(
    insert <sinhvien ID="15" Ten="Lan">
        <monhoc ID="10" Ten="Toan Roi Rac" />
        <monhoc ID="11" Ten="Lap Trinh C#" />
        <monhoc ID="12" Ten="CSDL Nang Cao" />
    </sinhvien>
    as last into [/THONGTINSV](1)
)
WHERE TenDH = 'DH CNTT'
GO
```

Mã nguồn 23: Bài 2. INSERT phần tử mới vào XML

Giải thích:

- `insert ... as last into [/THONGTINSV](1)` : Chèn phần tử mới vào cuối cùng của phần tử `THONGTINSV` đầu tiên.
- `(1)` : Chỉ định phần tử đầu tiên (vì có thể có nhiều phần tử `THONGTINSV`).

2.12. Câu 11: Phép nối (Join) với XQuery

Thực hiện phép nối bằng các lệnh XQuery như một câu truy vấn trong SQL để trả về thông tin là sinh viên nào học khóa học tên là gì, sinh viên nào học môn học gì?

```
SELECT ChiTietSV.query(
    for $sv in /THONGTINSV/sinhvien
    for $mh in $sv/monhoc
    return
        <row>
            <TenKhoaHoc>{sql:column("TenDH")}</TenKhoaHoc>
            <TenSinhVien>{data($sv/@Ten)}</TenSinhVien>
            <TenMonHoc>{data($mh/@Ten)}</TenMonHoc>
        </row>
) AS KetQua
FROM QuanLySV
```

Mã nguồn 24: Bài 2. Phép nối với XQuery

Giải thích:

- `for $sv in /THONGTINSV/sinhvien` : Vòng lặp ngoài duyệt qua tất cả sinh viên.
- `for $mh in $sv/monhoc` : Vòng lặp trong duyệt qua tất cả môn học của mỗi sinh viên.
- `sql:column("TenDH")` : Lấy giá trị từ cột quan hệ `TenDH` của bảng.
- `data($sv/@Ten)` : Lấy giá trị của thuộc tính `Ten` từ biến `$sv`.
- Kết quả: Mỗi dòng sẽ là một cặp (Sinh viên, Môn học) với tên khóa học tương ứng.

Kết quả mẫu:

```
<row>
  <TenKhoaHoc>DH CNTT</TenKhoaHoc>
  <TenSinhVien>Nam</TenSinhVien>
  <TenMonHoc>Co So Du Lieu</TenMonHoc>
</row>
<row>
  <TenKhoaHoc>DH CNTT</TenKhoaHoc>
  <TenSinhVien>Nam</TenSinhVien>
  <TenMonHoc>Cau Truc Du Lieu</TenMonHoc>
</row>
...

```

Mã nguồn 25: Bài 2. Kết quả mẫu của Câu 11

2.13. Tổng kết

Trong bài này, chúng ta đã học cách:

- Sử dụng XPath để định vị và chọn phần tử XML.
- Sử dụng XQuery với cú pháp FLWOR để truy vấn dữ liệu XML.
- Sử dụng phương thức `modify()` để sửa đổi dữ liệu XML:
 - ▶ `insert` : Chèn phần tử mới.
 - ▶ `delete` : Xóa phần tử hoặc thuộc tính.
 - ▶ `replace value of` : Thay thế giá trị.
- Sử dụng phương thức `exist()` để kiểm tra sự tồn tại.
- Thực hiện phép nối (join) giữa các phần tử XML và dữ liệu quan hệ.

