CHƯƠNG 3:
XỬ LÝ THÔNG TIN TRÊN MÁY TÍNH:
TRUY VẤN DỮ LIỆU

Khoa Khoa học và kỹ thuật thông tin
Bộ môn Thiết bị di động và Công nghệ Web

NỘI DUNG

1. Truy vấn SQL.
2. Xpath/Xquerry.
3. Các dạng truy vấn select thường gặp.

SQL – truy vấn SQL

Giới thiệu

- SQL (Structured Query Language):
  + Là ngôn ngữ cấp cao.
  + Dùng để truy vấn dữ liệu trong CSDL quan hệ.
  + Được IBM phát triển (1970s).
  + Được gọi là SEQUEL.
  + Được ANSI công nhận và phát triển thành chuẩn SQL-86, SQL-92, SQL-99.
- Đây là ngôn ngữ chuẩn dùng để truy vấn trong các CSDL quan hệ. Các CSDL quan hệ dù khác nhau về nền tảng và hãng sản xuất nhưng luôn có điểm chung là dùng SQL làm ngôn ngữ truy vấn.

Các nhóm lệnh

- Nhóm định nghĩa dữ liệu (DDL):
  - + Gồm các lệnh tạo, thay đổi cấu trúc các bảng dữ liệu (*Create, Drop, Alter*)
- Nhóm thao tác dữ liệu (DML):
  - + Gồm các lệnh làm thay đổi dữ liệu lưu trong bảng (*Insert, Delete, Update, Select*)
- Nhóm điều khiển dữ liệu (DCL):
  - + Gồm các lệnh quản lý quyền truy cập vào dữ liệu và các bảng (*Grant, Revoke, Deny*)

Phép kết trên nhiều bảng

- Phép ϕ kết, kết tự nhiên
  - + Mệnh đề WHERE chỉ ra điều kiện kết giữa các thuộc tính của các bảng
  - + Hoặc dùng từ khóa **Inner Join** (hoặc Join) trong mệnh đề FROM.
- Phép kết trái, phải, ngoài
  - + Dùng Half Outer Join (**Left join**, **Right Join**), **Full Outer Join** trong mệnh đề FROM

Phép kết trên nhiều bảng

NHANVIEN (HONV, TENLOT, TENNV, MANV, NGSINH, DCHI, PHAI, LUONG, MA_NQL, PHG)

PHONGBAN (TENPHG, MAPHG, TRPHG, NG_NC)

DIADIEM_PHG (MAPHG, DIADIEM)

THANNHAN (MA_NVIEN, TENTN, PHAI, NGSINH, QUANHE)

DEAN (TENDA, MADA, DDIEM_DA, PHONG)

PHANCONG (MA_NVIEN, SODA, THOIGIAN)

Phép kết trên nhiều bảng

1. *In danh sách mã số, họ tên nhân viên và tên thân nhân của nhân viên đó.*
2. *In danh sách mã số, họ tên của tất cả các nhân viên và tên thân nhân của nhân viên đó (nếu có).*

TRUY VẤN LỒNG

- Các câu lệnh SELECT có thể lồng nhau ở nhiều mức.
- Các câu truy vấn con thường trả về một tập các giá trị.
- Các câu truy vấn con được kết hợp bằng phép nối logic với câu truy vấn cha.

In và Not In

+ Cú pháp:

<thuộc tính> (NOT) IN (<truy vấn con>)

+ Lưu ý:

- Thuộc tính ở mệnh đề SELECT của truy vấn con phải có cùng kiểu dữ liệu với thuộc tính ở mệnh đề WHERE của truy vấn cha

Ví dụ:

3. Tìm nhân viên chưa được phân công thực hiện đề án nào

```
SELECT manv, honv+' '+tenlot+' '+tennv as hoten
FROM Nhanvien
Where manv NOT IN
                (SELECT distinct ma_nvien FROM Phan cong)
```

Any/Some và All

- Cú pháp:

  <thuộc tính> <phép so sánh> Any/Some/All (<truy vấn con>)

- Lưu ý:
  - + Thuộc tính ở mệnh đề SELECT của truy vấn con phải có cùng kiểu dữ liệu với thuộc tính ở mệnh đề WHERE của truy vấn cha
  - + Any/Some: so sánh với bất kỳ giá trị nào đó trong tập hợp.
  - + All: so sánh với tất cả các giá trị trong tập hợp.

Exists và Not Exists

- Cú pháp:

  (NOT) EXISTS (<câu truy vấn con>)

- Lưu ý:
  - + Không cần có thuộc tính, hằng số hay biểu thức nào khác đứng trước
  - + Không nhất thiết liệt kê tên thuộc tính ở mệnh đề SELECT của truy vấn con
  - + Những câu truy vấn có điều kiện “= ANY” hay “IN” đều có thể chuyển thành câu truy vấn có EXISTS

Phân loại TRUY VẤN LỒNG

- Lồng phân cấp:
  - + Mệnh đề WHERE của truy vấn con **không tham chiếu** đến thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha
  - + Khi thực hiện, câu truy vấn con sẽ được thực hiện trước
- Lồng tương quan:
  - + Mệnh đề WHERE của truy vấn con **tham chiếu ít nhất một thuộc tính** của các quan hệ trong mệnh đề FROM ở truy vấn cha
  - + Khi thực hiện, câu truy vấn con sẽ được thực hiện nhiều lần, mỗi lần tương ứng với một bộ của truy vấn cha

Một số dạng truy vấn khác

- Câu truy vấn con không chỉ xuất hiện ở mệnh đề WHERE mà có thể xuất hiện ở những nơi khác (SELECT, FROM, HAVING,…)
- Kết quả trả về của câu truy vấn con
  + Là một bảng trung gian trong quá trình truy vấn
  + Bảng này không có lưu trữ thật sự

Các phép toán trên tập hợp

- SQL có cài đặt các phép toán
  - + Hội (UNION).
  - + Giao (INTERSECT).
  - + Trừ (EXCEPT).
- Kết quả trả về là tập hợp
  - + Loại bỏ các bộ trùng nhau.
  - + Để giữ lại các bộ trùng nhau:
    - UNION ALL
    - INTERSECT ALL
    - EXCEPT ALL

Các phép toán trên tập hợp

- Tuy nhiên, chúng ta có thể sử dụng **IN**, **NOT IN**, **EXISTS**, **NOT EXISTS**, … để thực hiện các phép toán hội, giao trừ trên tập hợp.
- Đối với phép chia: sử dụng **NOT EXISTS**.
- Ví dụ:
  1. Tìm họ tên nhân viên được phân công thực hiện tất cả các đề án.
  2. Tìm tên các đề án được phân công cho tất cả các nhân viên thuộc phòng số 5 thực hiện

Hàm kết hợp, gom nhóm

- Hàm kết hợp
- Top N
- Gom nhóm

Hàm kết hợp

- MIN, MAX, SUM, AVG:

  TÊN HÀM(<tên thuộc tính>)

- COUNT
  - ✓ COUNT(*) đếm số dòng
  - ✓ COUNT(<tên thuộc tính>) đếm số dòng thuộc tính có giá trị khác NULL
  - ✓ COUNT(DISTINCT <tên thuộc tính>) đếm số dòng thuộc tính có giá trị khác nhau và khác NULL

Ví dụ

20. Tính lương thấp nhất, cao nhất, trung bình và tổng lương của nhân viên
21. Có tất cả bao nhiêu nhân viên
22. Có bao nhiêu nhân viên được quản lý trực tiếp bởi người khác
23. Có tất cả bao nhiêu người quản lý
24. Có bao nhiêu nhân viên không có người quản lý trực tiếp

Top N

- Trả về N dòng kết quả đầu tiên của câu truy vấn
- Cú pháp: **TOP N**

  với N là số nguyên dương
- Nên sử dụng ORDER BY để sắp xếp kết quả

Vd 25

```
SELECT   TOP 1 luong as CN
FROM Nhanvien
ORDER BY luong DESC
```

Gom nhóm

– Cú pháp:

```
SELECT <các thuộc tính>
FROM <các bảng>
[WHERE <các điều kiện>]
GROUP BY <các thuộc tính gom nhóm>
[HAVING <các điều kiện>]
```

– Trong đó:

- Điều kiện ở WHERE thực hiện **trước** khi gom nhóm
- Điều kiện ở HAVING thực hiệu **sau** khi gom nhóm
- Các thuộc tính sau GROUP BY dùng để gom nhóm và phải có đầy đủ các thuộc tính sau SELECT (trừ những thuộc tính trong những hàm kết hợp)

Ví dụ

26. Tìm số lượng nhân viên của từng phòng ban
27. Tìm số lượng nhân viên Nam (phai='Nam') của từng phòng ban
28. Tìm phòng ban có từ 2 nhân viên Nam (phai='Nam') trở lên
29. Tìm phòng ban có đông nhân viên nhất
30. Tìm đề án có ít nhân viên Nữ (phai='Nu') tham gia nhất
31. Tìm 3 nhân viên thuộc phòng số 4 (phg=4) có lương thấp nhất

Xpath/Xquery

Giới thiệu

- Xpath và Xquery:
  - + Là hai ngôn ngữ có rất nhiều mặt giống nhau, hỗ trợ tìm kiếm thông tin trong tài liệu XML.
  - + Có thể xem Xpath là tập hợp con của Xquery.
  - + Xquery sử dụng Xpath như là một ngôn ngữ chính để định hướng tìm kiếm thay vì dùng đệ qui để duyệt cây.

Xpath

- XML Path language (gọi tắt là Xpath) là một chuẩn để xử lý tài liệu XML (cũng như SQL là một chuẩn để làm việc với csdl).
- Dùng để xử lý nhiều kiểu truy vấn trong tài liệu XML và các biến thể của nó (như HTML).
- Là ngôn ngữ rất phổ biến.
- Tiết kiệm thời gian trích xuất dữ liệu.

Ví dụ 36: CSDL XML

```
1.  <users>
2.    <user>
3.      <name>
4.        <first>Lola</first>
5.        <last>Solis</last>
6.      </name>
7.      <age>2</age>
8.    </user>
9.    <user>
10.     <name>
11.       <first>Nina</first>
12.       <last>Serafina</last>
13.     </name>
14.     <age>4</age>
15. <visits>
16.       <first>2008-01-15</first>
17.       <last>2008-02-15</last>
18.     </visits>
19.   </user>
20.   <user>
21.     <name>
22.       <first>Tracy</first>
23.       <last>Keller</last>
24.     </name>
25.     <age>35</age>
26.   </user>
27. </users>
```

Ví dụ

- Đoạn Xpath sau: tìm tên của người dùng dưới 18 tuổi

```
1.  /user[age lt 18]/name/last/text()
2.
3.   (: Result
4.        Solis
5.        Serafina
6.   :)
```

- Nếu không dùng Xpath sẽ gặp chút khó khăn khi xử lý việc loại bỏ giá trị trong node visits.

Nhận xét về Xpath

- Biểu thức Xpath ngắn gọn, rõ ràng.
- Hiểu được các node phức tạp trong tài liệu XML và biết các mối quan hệ giữa chúng.
- Một hạn chế của Xpath là *không cung cấp cách chuyển đổi tập kết quả trả về*.
  + Ở ví dụ trên không thể sắp kết quả hiện thị tăng dần theo tên.

XQuery

- Phức tạp hơn so với Xpath.
- Sử dụng cú pháp pha trộn XML và Xpath.
- Khắc phục được nhược điểm của Xpath:
  - + Sắp xếp kết quả của câu truy vấn hoặc chuyển chúng thành HTML, CSV, SQL, XML …
  - + Cung cấp tính năng biểu thức FLWOR.
  - + Sử dụng hàm và đệ quy.
  - + Diễn tả các phép nối.

Biểu thức FLOWR

- Biểu thức FLWOR dùng để liên kết các tiêu chí rút trích dữ liệu và chuyển đổi tập kết quả trả về của câu truy vấn.
- FLWOR là viết tắt của các từ **f**or, **l**et, **w**here, **o**rder by và **r**eturn.
- Bắt đầu bằng một biểu thức *for* hoặc *let* và kết thúc bằng một biểu thức *return.*

Ví dụ 1

– Tìm tên của người dùng dưới 18 tuổi, có sắp xếp kết quả tăng dần.

```
1. let $xml:= _XML from Vidu36
2. for $user in $xml//user[age lt 18]
3. order by $user/name/last
4. return $user/name/last/text()

5. (: Result
6.          Serafina
7.          Solis
8. :)
```

Ví dụ 2

– Kết quả truy vấn trả về 1 đoạn HTML, danh sách có đánh số thứ tự.

```
1.  let $xml:= _XML from Vidu36
2.  return
3.          <ol>{
4.                  for $user in $xml//user[age lt 18]
5.                  order by $user/name/last
6.                  return <li>{$user/name/last/text()}</li>
7.          }</ol>
8.  (: Result
9.          <ol><li>Serafina</li><li>Solis</li></ol>
10.:)
```

Sử dụng hàm và đệ quy

- Xquery cung cấp các hàm, các phép toán được xây dựng sẵn và cho phép định nghĩa các hàm riêng.
- Xquery cũng hỗ trợ đệ quy: tiện lợi khi làm việc với XML (có thể chứa các node lồng nhau tùy ý).

Ví dụ 4

- Định nghĩa hàm transform-names dùng để thay đổi tên các node trong bất kỳ tài liệu XML nào.

Ví dụ 4 – xquery (1)

```
1. (: Part 1 :)
2. define function transform-names($node as node()) as node() {
3.   element{replace(name($node), "_", "-")} {
4.     $node/text(), for $subnode in $node/* return transform-names($subnode)
5.   }
6. }
```

Ví dụ 4 – xquery (2)

```
(: Part 2 :)
let $xml:=
<item>
  <item_type>book</item_type>
  <contributors>
    <author>
      <first_name>Charles</first_name>
      <last_name>Edward</last_name>
      <home_address>
        <home_street>206 S. Solomon St.</home_street>
        <home_city>New Orleans</home_city>
        <home_state>LA</home_state>
        <home_zip>70119</home_zip>
      </home_address>
    </author>
    <artist>
      <last_name>Salinas</last_name>
    </artist>
  </contributors>
</item>
return transform-names($xml)
```

Ví dụ 4 – xquery (3)

```
(: Result
    <item>
      <item-type>book</item-type>
      <contributors>

        <author>
          <first-name>Charles</first-name>
          <last-name>Edward</last-name>
          <home-address>
            <home-street>206 S. Solomon St.</home-street>
            <home-city>New Orleans</home-city>
            <home-state>LA</home-state>
            <home-zip>70119</home-zip>
          </home-address>
        </author>
        <artist>
          <last-name>Salinas</last-name>
        </artist>
      </contributors>
    </item>
:)
```

MỘT SỐ DẠNG TRUY VẤN SELECT THƯỜNG GẶP

CSDL QUẢN LÝ BÁN HÀNG

KHACHHANG (MAKH, HOTEN, DCHI, SODT, NGSINH, DOANHSO, NGDK)

NHANVIEN (MANV,HOTEN, NGVL, SODT)

SANPHAM (MASP,TENSP, DVT, NUOCSX, GIA)

HOADON (SOHD, NGHD, MAKH, MANV, TRIGIA)

CTHD (SOHD,MASP,SL)

TRUY VẤN SELECT

– Cú pháp câu truy vấn *SELECT*:

```
SELECT <cột 1>, <cột 2>, ....
FROM <tên bảng>
WHERE <điều kiện>
ORDER BY <tên cột> ASC | DESC
GROUP BY <tên cột 1>, <tên cột 2>, ....
HAVING <điều kiện>
```

– Lưu ý:

+ Mệnh đề HAVING sử dụng cho các **hàm gom nhóm**.

+ ASC – sắp xếp tăng dần; DESC – sắp xếp giảm dần.

TRUY VẤN SELECT – DẠNG 1

– Dạng 1: Truy vấn lấy dữ liệu tất cả.

```
SELECT * FROM <tên bảng>
hoặc
SELECT <danh sách cột> FROM <tên bảng>
```

VD:

```
SELECT * FROM KHACHHANG
SELECT MAKH, HOTEN, DCHI FROM KHACHHANG
```

TRUY VẤN SELECT – DẠNG 2

– Dạng 2: Truy vấn dữ liệu có điều kiện:

```
SELECT <danh sách cột> FROM <tên bảng>
WHERE <điều kiện>
```

VD: Lấy thông tin MAKH, HOTEN DCHI của khách hàng có doanh số trên 1000

```
SELECT MAKH, HOTEN, DCHI FROM KHACHHANG
WHERE DOANHSO > 10000
```

TRUY VẤN SELECT – DẠNG 3

– Dạng 3: Truy vấn dữ liệu có kết bảng.

```
SELECT <danh sách cột> FROM <tên bảng 1>
INNER JOIN <tên bảng 2> ON <tên bảng 1>.<mã khoá
ngoại> = <tên bảng 2>.<mã khoá chính>
[WHERE <điều kiện>]
```

– Các phép kết:

+ *INNER JOIN*: kết bằng.

+ *LEFT OUTER JOIN*: kết mở rộng về bên trái.

+ *RIGHT OUTER JOIN*: kết mở rộng về bên phải.

VD

VD:

```
SELECT MAKH, HOTEN
FROM HOADON INNER JOIN KHACHHANG ON KHACHHANG.MAKH
= HOADON.MAKH
WHERE NGHD = '16/7/2019'
```

TRUY VẤN SELECT – DẠNG 4

– Dạng 4: Truy vấn dữ liệu có sắp xếp

```
SELECT <danh sách tên cột> FROM <tên bảng>
[WHERE <điều kiện>]
ORDER BY <danh sách cột cần sắp xếp> ASC hoặc DESC
```

VD: Sắp xếp khách hàng theo ngày sinh giảm dần

```
SELECT MAKH, HOTEN, NGSINH FROM KHACHHANG
ORDER BY NGSINH DESC
```

TRUY VẤN SELECT – DẠNG 5

– Dạng 5: Truy vấn sử dụng các hàm gom nhóm

```
SELECT <các hàm gom nhóm> FROM <tên bảng>
[WHERE <điều kiện>]
GROUP BY <tên cột 1>, <tên cột 2>, ...
```

– Các hàm gom nhóm: *COUNT(), AVG(), MAX(), MIN(), SUM().*

– Lưu ý: Các thuộc tính trong mệnh đề SELECT (trừ các hàm kết hợp), phải xuất hiện trong mệnh đề GROUP BY.

VD

VD: Tính giá trị trung bình doanh số theo từng MAKH đối với khách hàng có doanh số trên 10000

```
SELECT MAKH, AVG(DOANHSO) FROM KHACHHANG
WHERE DOANHSO > 10000
GROUP BY MAKH
```

TRUY VẤN SELECT – DẠNG 6

– Dạng 6: Truy vấn sử dụng hội - giao - trừ

```
SELECT <danh sách cột 1> FROM <tên bảng>
[WHERE <điều kiện 1>]
UNION (hội) | INTERSECT (giao) | EXCEPT (trừ)
SELECT <danh sách cột 2> FROM <tên bảng>
[WHERE <điều kiện 2>]
```

Lưu ý: Để sử dụng các phép hội giao trừ thì 2 quan hệ phải khả hợp, tức *<danh sách cột 1> = <danh sách cột 2>*

VD

VD1: Tìm khách hàng mua hoá đơn HD01 hoặc HD02

```
SELECT MAKH, HOTEN FROM KHACHHANG INNER JOIN
HOADON ON KHACHHANG.MAKH = HOADON.MAKH
WHERE MAHD = 'HD01'
UNION
SELECT MAKH, HOTEN FROM KHACHHANG INNER JOIN
HOADON ON KHACHHANG.MAKH = HOADON.MAKH
WHERE MAHD = 'HD02'
```

VD

VD2: Tìm khách hàng mua cùng lúc hoá đơn HD01 và HD02

```
SELECT MAKH, HOTEN FROM KHACHHANG INNER JOIN
HOADON ON KHACHHANG.MAKH = HOADON.MAKH
WHERE MAHD = 'HD01'
INTERSECT
SELECT MAKH, HOTEN FROM KHACHHANG INNER JOIN
HOADON ON KHACHHANG.MAKH = HOADON.MAKH
WHERE MAHD = 'HD02'
```

VD

VD3: Tìm khách hàng không mua hoá đơn nào

```
SELECT MAKH, HOTEN FROM KHACHHANG
EXCEPT
SELECT MAKH, HOTEN FROM KHACHHANG INNER JOIN
HOADON ON KHACHHANG.MAKH = HOADON.MAKH
```

TRUY VẤN SELECT – DẠNG 7

– Dạng 7: Truy vấn lồng:

```
SELECT <danh sách cột> FROM <tên bảng>
WHERE <so sánh tập hợp> (
      SELECT <danh sách cột> FROM <tên bảng>
      WHERE <điều kiện>
)
```

<so sánh tập hợp>: *ALL, IN, NOT IN, ALL, ANY, EXISTS, NOT EXISTS.*

VD

VD: Tìm thông tin mã hoá đơn có trị giá cao nhất

```
SELECT SOHD FROM HOADON
WHERE TRIGIA = (
    SELECT MAX(TRIGIA) FROM HOADON
    GROUP BY SOHD
)
```

TRUY VẤN SELECT – DẠNG 8

– Dạng 8: Truy vấn lồng tương quan

```
SELECT <danh sách cột> FROM <tên bảng> AS OB1
WHERE <so sánh tập hợp> (
      SELECT <danh sách cột> FROM <tên bảng> AS OB2
      WHERE OB1.<tên cột> = OB2.<tên cột>
)
```

<so sánh tập hợp>: *ALL, IN, NOT IN, ALL, ANY, EXISTS, NOT EXISTS.*

VD

VD: Tìm sản phẩm có giá cao nhất theo từng nước sản xuất.

```
SELECT NUOCSX FROM SANPHAM AS SP1
WHERE GIA = (
    SELECT MAX(GIA) FROM SANPHAM AS SP2
    WHERE SP1.MASP = SP2.MASP
)
```

TRUY VẤN SELECT – DẠNG 9

– Dạng 9: Truy vấn dùng bảng "con" (inner aggregate).

```
SELECT <danh sách cột 1> FROM (
      SELECT <danh sách cột 2> FROM <tên bảng>
      WHERE <điều kiện>
)AS <tên bảng con>
```

Lưu ý: *<danh sách cột 1>* phụ thuộc vào *<danh sách cột 2>* trả về từ câu truy vấn con (subquery).

VD

VD: Tìm khách hàng có số lần mua hàng nhiều nhất.

```
SELECT MAKH FROM HOADON
GROUP BY MAKH
HAVING COUNT(SOHD) = (
     SELECT MAX(SL_HD) FROM (
          SELECT MAKH, COUNT(SOHD) AS SL_HD
          FROM HOADON
          GROUP BY MAKH)AS T
)
```

Lưu ý: Không thể sử dụng dạng *COUNT(MAX(SOHD))* trong SQL Server

TRUY VẤN SELECT – DẠNG 10

– Dạng 10: Phép chia

Tìm *<đối tượng 1>* đã ... tất cả *<đối tượng 2>*

Cần xác định:

Đối tượng 1 (MaDT1, ...).

Đối tượng 2 (MaDT2, ...).

Quan hệ Đối tượng 1 và đối tượng 2 (*MaDT1*, *MaDT2*, ...).

VD

```
SELECT <danh sách cột> FROM <tên bảng đối tượng 1>
AS OB1 WHERE NOT EXISTS (
    SELECT <danh sách cột> FROM <tên bảng đối
tượng 2> AS OB2 WHERE AND NOT EXISTS (
        SELECT * FROM <tên bảng quan hệ đối tượng
1 và 2> as OB3 WHERE OB2.<khoá chính> = OB3.<khoá
ngoại> and OB3.<khoá ngoại> = OB1.<khoá chính>
    )
)
```

VD

– VD: Tìm hoá đơn đã mua tất cả sản phẩm xuất xứ Thái Lan:

Đối tượng 1: HOADON(SOHD, NGHD, ...)

Đối tượng 2: SANPHAM(MASP, TENSP, XUATXU)

Quan hệ giữa 2 đối tượng: CTHD(*MASP*, *TENSP*).

```
SELECT SOHD FROM HOADON AS T1 WHERE NOT EXISTS (
    SELECT MASP FROM SANPHAM AS T2 WHERE XUATXU =
    "Thái Lan" AND NOT EXISTS (
        SELECT MASP, TENSP FROM CTHD AS T3 WHERE
    T2.MASP = T3.MASP AND T1.SOHD = T3.SOHD)
)
```

Tổng kết

1. Để truy vấn dữ liệu trong CSDL quan hệ, ta dùng ngôn ngữ SQL.
2. SQL có 3 nhóm lệnh chính: DDL, DML và DCL.
3. Các truy vấn nâng cao trên SQL: kết (nhiều bảng), gom nhóm, truy vấn lồng.
4. Khung nhìn View là một bảng ảo được tạo ra dựa trên bảng vật lý, dùng để đảm bảo an ninh dữ liệu.
5. Xpath/Xquery là ngôn ngữ dùng để truy vấn CSDL XML.
6. Xquery là phiên bản mở rộng của Xpath, hỗ trợ các hàm, đệ quy, ... mà bản thân Xpath không làm được.

TÀI LIỆU THAM KHẢO

1. Nguyễn Gia Tuấn Anh, Trương Châu Long, *Bài tập và bài giải SQL Server*, NXB Thanh niên (2005).
2. Đỗ Phúc, Nguyễn Đăng Tỵ, *Cơ sở dữ liệu*, NXB Đại học quốc gia TPHCM (2010).
3. *Nguyễn Gia Tuấn Anh, Mai Văn Cường, Bùi Danh Hường*, *Cơ sở dữ liệu nâng cao*, NXB Đại học quốc gia TPHCM (2019).
4. Itzik Ben-Gan, *Microsoft SQL Server 2012- TSQL Fundamentals*.


Q
A