**1**

**NGÔN NGỮ TRUY VẤN DỮ LIỆU SQL**

CƠ SỞ DỮ LIỆU

Nội dung

**2**

****1 **Giới thiệu chung**2 **Lệnh định nghĩa – khởi tạo** 3 **Truy vấn cơ bản **4 **Sắp xếp – gom nhóm** 5 **Truy vấn lồng** 

Giới thiệu chung

**3**

❖ **RDBMS**(Relational database management system) là hệ thống quản lý các cơ sở dữ liệu quan hệ( relational database) như MS SQL, MySQL, Oracle. ❖ **Relational database**: là database cho phép liên kế dữ liệu để lưu trữ một lúc nhiều table, những liên kết thành lập mối quan hệ giữa những table cung cấp một cách thuận lợi nhất để lưu trữ dữ liệu, có thể nhập ở một nơi và tham chiếu đến nhiều table khác trong database



Giới thiệu chung

**4**

⮚ SQL : Structured Query Language ⮚ Tiền thân SEQUEL và SEQUEL-2 ⮚ Do IBM phát triển ( 1974-1976). ⮚ Các phiên bản

⮚ SQL - 86

⮚ SQL - 92

⮚ SQL - 99

⮚ Hệ quản trị cơ sở dữ liệu

⮚ Ngôn ngữ truy vấn dữ liệu

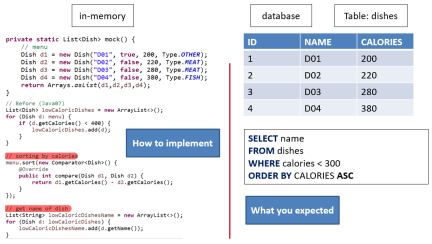
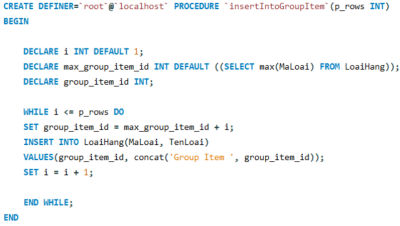
ANSI (American National Standards Institute) và ISO (International Standards Organization)

Giới thiệu chung

**5**

⮚ SQL is a language that all commercial RDBMS implementations understand.

⮚ SQL is a non-procedural language



Giới thiệu chung

**6**

⮚ Mỗi hệ quản trị CSDL đều phải có phải có ngôn ngữ giao tiếp giữa người sử dụng với cơ sở dữ liệu

⮚ Ngôn ngữ giao tiếp CSDL gồm các loại sau:

- Data Definition Language – **DDL**: Ngôn ngữ mô tả dữ liệu

- Data Manipulation Language – **DML**: Ngôn ngữ thao tác dữ liệu 

Giới thiệu chung **7**

Data Definition

Language

SQL

Data Manipulation Language

DDL DML Tạo cấu trúc CSDL Thao tác trên CSDL

Create Select Insert Drop

Truncate

Alter Update Delete

CẤU TRÚC CHUNG

**8**

****RDBMS 

Database

TABLE 

Rows Columns

DDL

**9**

Kiểu dữ liệu

❖ Kiểu số

- Interger, smallint, **int**

- Numberic, decimal, real, **float**

❖ Boolean

- bit

❖ Kiểu chuỗi ký tự - char (n), nchar (n) - varchar(n), nvarchar (n) - text

❖ Ngày giờ

- date: ngày, tháng, năm - time: giờ, phút, giây - datetime: date + time

❖ Bổ sung

- char: fixed length

- varchar: dynamic length - n: national language

DDL

**10**

Phân biệt char và varchar

The CHAR and VARCHAR types are similar, but differ in the way they are stored and retrieved. They also differ in **maximum length** and in **whether trailing spaces are retained**.

Lệnh định nghĩa - DDL

**11**

Lệnh CREATE được dùng để: ⮚ Tạo lược đồ (cơ sở dữ liệu)

CREATE DATABASE | SCHEMA

⮚ Tạo bảng

⮚ Tạo khung nhìn ⮚ Tạo ràng buộc ⮚ Xóa dữ liệu

CREATE TABLE DROP TABLE

ALTER TABLE

TRUNCATE TABLE

CREATE VIEW

DDL

**12**

Lệnh tạo bảng

Bảng: tên bảng

tập thuộc tính

CREATE TABLE TenBang

(

<TenCot> <kieu du lieu> [RBTV], <TenCot> <kieu du lieu> [RBTV], [RBTV]

)

Tên

Kiểu dữ liệu

Ràng buộc toàn vẹn (RBTV)

CREATE TABLE PhongBan

(

MaPB int primary key,

TenPB varchar(50) not null, NgayTao Date not null

)

DDL

**13**

Tên 

Kiểu dữ liệu

Ràng buộc toàn vẹn (RBTV)

CREATE TABLE PhongBan

(

MaPB int primary key,

TenPB varchar(50) not null,

NgayTao Date

)

DDL

**14**

Ràng buộc

Not null: Không cho phép thuộc tính chứa giá trị null

Null: Được chứa giá trị null

Khóa chính: Primary key

Khóa ngoại: Foreign key - references

Unique: Không trùng lắp

Default: Giá trị mặc định

Check: Kiểm tra điều kiện nào đó

Đặt tên ràng buộc: [CONSTRAINT <tên ràng buộc>] <RBTV>

DDL

**15**

****

DDL

**16**

ALTER TABLE : Thay đổi cấu trúc ràng buộc của bảng

❖ **Thêm cột**

ALTER TABLE <tên bảng> **ADD <tên côt>** <kiểu dữ liệu> [RBTV] *ALTER TABLE PhongBan ADD NgayTao Datetime null*

❖ **Xóa cột**

ALTER TABLE <tên bảng> **DROP COLUMN <tên cột>**

*ALTER TABLE PhongBan DROP NgayTao*

❖ **Thay đổi kiểu dữ liệu**

ALTER TABLE <tên bảng> ALTER COLUMN <tên côt> <tên mới mới> <kiểu dữ liệu mới> ALTER TABLE <tên bảng> **CHANGE** <tên cột> <tên mới mới> <KDL mới> <RBTV>

*ALTER TABLE PhongBan CHANGE NgayTao NgayTao Date*

DDL

**17**

Thay đổi cấu trúc ràng buộc của bảng ALTER TABLE

❖ **Thêm ràng buộc**

ALTER TABLE <tên bảng> ADD CONSTRAINT <tên ràng buộc> <RBTV> (referenced\_columns) [REFERENCES PARENT\_TABLE

❖ **Xóa ràng buộc**

ALTER TABLE <tên bảng> DROP **<kiểu ràng buộc>** <tên ràng buộc>

Lưu ý: Khi tạo khóa chính bên ngoài lệnh tạo bảng, thì các thuộc tính của khóa chính phải được khai báo là not null trong câu lệnh tạo.

(referenced\_columns)],

Lệnh thao tác dữ liệu - DML - Insert

**18**

Thêm 1 dòng dữ liệu



Các giá trị theo đúng thứ tự

các cột trong table

DML - Insert

**19**

Thêm nhiều dòng dữ liệu

INSERT INTO table\_name (column1, colname\_2, …. ) VALUES ( values\_1, values\_2,…,values\_n),

( values\_1, values\_2,…,values\_n)

*Note: Dòng cuối không có dấu phẩy “,”*

**INSERT INTO table\_name (column1, colname\_2, …. )

VALUES ( values\_1, values\_2,…,values\_n),



DML - Insert

**20**

**Lời khuyên khi tạo database và nhập liệu**

Nhập liệu

+ Nhập các bảng không có khóa ngoại trước

+ Nếu nhập bảng có khóa ngoại, cho giá trị bằng NULL

+ Nhập các bảng có khóa ngoại và dữ liệu đúng với dữ liệu ở referenced table **+ DIS/EN-ABLE FOREIGN\_KEY\_CHECKS**

DML - DELETE

**21**

1. Xóa toàn bộ dữ liệu

DELETE FROM TABLE



2. Xóa theo điều kiện

DELETE FROM table\_name

WHERE <condition>



Foreign Key Option



**There are three types of on delete associated with foreign key**

❖**On Delete Cascade**: when data is removed from a parent table, automatically data deleted from child table (foreign key table).

❖**On Delete set Null**: when data is removed from a parent table, the foreign key associated cell will be null in a child table.

❖**On Delete Restrict**: when data is removed from a parent table, and there is a foreign key associated with child table it gives error, you can not delete the record.

Khởi tạo dữ liệu - UPDATE

**23**

Cập nhật dữ liệu

UPDATE table\_name 

SET table\_column = value

WHERE <condition>



DML - SELECT

**24**

Syntax

*Note*

Thứ tự hoạt động của câu lệnhTối thiểu có SELECT-FROM

Không thay đổi thứ tự các mệnh đề trong câu truy vấn

Không phân biệt hoa thường

DML - SELECT

**25**

Syntax Sau khi lọc bảng theo điều kiện

**7 **

[LIMIT offset, row\_count] restrict amount of data, pagination **6**

Example

Thống kê số lượng học sinh của mỗi lớp trong khoa CNTT Thống kê số lượng tin tức của mỗi loại danh mục tin Đếm số lượng sản phẩm của mỗi loại mặt hàng

where, lọc **các nhóm** trả về dữ liệu theo nhóm thỏa điều kiện

DML - SELECT

**26**

Truy vấn tất cả các dòng trong bảng

**SELECT { tất cả các cột } FROM <Tên bảng>**

**SELECT {danh sách cột cần lấy} FROM <Tên bảng>**

****

**SELECT** \* 

**FROM** <table\_name>

**SELECT** columnA, columnB **FROM** < table\_name >

DML - SELECT

**27**

**DISTINCT & TOP - LIMIT**

Loại trừ giá trị trùng nhau trong kết quả trả về

**SELECT DISTINCT** <Danh sách các thuộc tính> **FROM SELECT DISTINCT khoa FROM lop** 

**SELECT TOP 5 hodem,ten,ngaysinh**

**FROM sinhvien**

**SELECT TOP 10 PERCENT hodem,ten,ngaysinh FROM sinhvien**

**SELECT maHS, tenHS**

**FROM hocsinh**

**LIMIT offset, rowcount**

DML - SELECT

**28**

PHÉP TÍNH TRÊN THUỘC TÍNH

Lấy thông tin tất cả mặt hàng kèm theo giá bán mới sau khi đã giảm 10 %



DML - SELECT

**29**

Chọn ra một số dòng - WHERE

**SELECT** {danh sách tất cả các tên cột}

**FROM** <Tên bảng>

**WHERE** <điều kiện>

*Lấy thông tin chi tiết của mặt hàng có mã số “BL/12/001”*

Use CASE structure in SELECT

**30**

Syntax:

CASE biểu\_thức

WHEN biểu\_thức\_kiểm\_tra THEN kết\_quả [ ... ]

[ELSE kết\_quả\_của\_else]

END

SELECT masv,hodem,ten, 

CASE gioitinh

WHEN 1 THEN 'Nam'

ELSE 'Nữ'

END AS gioitinh

FROM sinhvien

Logical Conditions

| **Operator**  = | **Meaning**  Equal to |
| --- | --- |
| > | Greater than |
| >= | Greater than or equal to |
| < | Less than |
| <= | Less than or equal to |
| <> | Not equal to |
| BETWEEN …AND… | Between two values (inclusive) |
| IN (set) | Match any of a list of values |
| LIKE | Match a character pattern |
| IS NULL | Is a null value |
| AND | Returns TRUE if both component conditions are true |
| OR | Returns TRUE if either component condition is true |
| NOT | Returns TRUE if the following condition is false |

**31**

Truy vấn dữ liệu - WHERE

**32**

**Các điều kiện tìm kiếm**

❖ So sánh : =, >, <, >=, <=

❖ Miền giá trị : BETWEEN, NOT BETWEEN

❖ Tập hợp : IN, NOT IN ❖ Điều kiện tìm kiếm chuỗi

❖ Điều kiện phức : AND, OR, NOT

Truy vấn dữ liệu - WHERE

**33**

Phép toán so sánh : =, >, <, >=, <=

**SELECT** {danh sách tất cả các tên cột}

**FROM** <Tên bảng>

**WHERE** <Tên côt> <Phép toán so sánh> <giá trị>

Kiểu dữ liệu Kiểu dữ liệu **TƯƠNG**

**ĐƯƠNG**

****

Truy vấn dữ liệu - WHERE

**34**

Phép toán so sánh : kiểu ngày giờ MM-DD-YYY YYY-MM-DD

Truy vấn dữ liệu - WHERE

**35**

Miền giá trị : BETWEEN , NOT BETWEEN

**SELECT** {danh sách tất cả các tên cột}

**FROM** <Tên bảng>

**WHERE** <Tên côt> **BETWEEN** <giá trị 1> **AND** <giá trị 2>

**SELECT** {danh sách tất cả các tên cột}

**FROM** <Tên bảng>

**WHERE** <Tên côt> **>=** <giá trị 1> **AND** <Tên côt> <= <giá trị 2>

Truy vấn dữ liệu - WHERE

**36**

Tập hợp: IN, NOT IN

**SELECT** {danh sách tất cả các tên cột}

**FROM** <Tên bảng>

**WHERE** <Tên côt> **IN (** giá trị 1, giá trị 2,…, giá trị n **)**

****

Truy vấn dữ liệu - WHERE

**37**

Tập hợp: IN, NOT IN

**SELECT** {danh sách tất cả các tên cột}

**FROM** <Tên bảng>

**WHERE** <Tên côt> **NOT IN (** giá trị 1, giá trị 2,…, giá trị n **)**

****

Truy vấn dữ liệu - WHERE

**38**

Chuỗi ký tự

**SELECT** {danh sách tất cả các tên cột}

**FROM** <Tên bảng>

**WHERE** <Tên côt> **=** ‘*giatri*’

*Tìm loại hàng có tên là đầm dài*

**

Truy vấn dữ liệu - WHERE

**39**

Chuỗi ký tự - LIKE , NOT LIKE

**SELECT** {danh sách tất cả các tên cột}

**FROM** <Tên bảng>

**WHERE** <Tên côt> **LIKE** ‘*giatri*’

%: chuỗi ký tự có thể rỗng 

hoặc n ký tự

- : 1 ký tự bất kỳ



Truy vấn dữ liệu - WHERE

**40**

Chuỗi ký tự - LIKE , NOT LIKE

**SELECT** {danh sách tất cả các tên cột}

**FROM** <Tên bảng>

**WHERE** <Tên côt> **NOT LIKE** ‘*giatri*’



Truy vấn dữ liệu - WHERE

**41**

Kiểu dữ liệu null

**SELECT** {danh sách tất cả các tên cột} 

**FROM** <Tên bảng>

**WHERE** <Tên côt> **IS null**

**SELECT** {danh sách tất cả các tên cột} 

**FROM** <Tên bảng>

**WHERE** <Tên côt> **IS NOT null**

ORDER BY

**42**

**SELECT** {danh sách tất cả các tên cột} **FROM** <Tên bảng>

**ORDER BY** <tên cột> **ASC (DESC)**

Giảm dần

**SELECT** {danh sách tất cả các tên cột} **FROM** <Tên bảng>

Tăng dần. Mặc định

Trái 🡪 Phải

**ORDER BY** <cột1> <đk sắp xếp> <cột 2> <đk sắp xếp>…

Khi mệnh đề có GROUP BY. Dữ liệu sắp khi gom nhóm sẽ được sắp xếp.

ORDER BY

**43**

Ví dụ:

Sắp xếp mặt hàng theo thứ tự giá bán tăng dần và số lượng theo thứ tự từ nhiều đến ít

**SELECT \***

**FROM** MatHang

**ORDER BY** GiaBan ASC, SoLuong DESC

**SELECT \***

**FROM** MatHang

**ORDER BY** GiaBan, SoLuong DESC

GROUP BY

**44**

Tạo ra các nhóm dữ liệu có cùng giá trị của thuộc tính gom nhóm

Thường được dùng với các hàm min, max, avg, sum, count, group\_conca

**SELECT** {danh sách các tên cột}

**FROM** <Tên bảng>

**GROUP BY** <tên cột>

Thuộc tính gom nhóm

Ví dụ: Liệt kê các mặt hàng và phân theo nhóm loại hàng

**SELECT** MaMH, TenMH, MaLH

**FROM** MatHang

**GROUP BY** MaLH

GROUP BY

**45**

****

HAVING

**46**

Điều kiện trên nhóm

**SELECT** {danh sách tất cả các tên cột}

**FROM** <Tên bảng>

**GROUP BY** <tên thuộc tính gom nhóm>

**HAVING** <điều kiện>

*Ví dụ: Tìm tất cả các loại hàng có số lượng mặt hàng lớn hơn 1*

**SELECT** Loai, count(SoLuong)

**FROM** MatHang

**GROUP BY** Loai

**HAVING** count(soluong) > 1

HAVING

**47**

Điều kiện trên nhóm

Chỉ kiểm tra điều kiện trên nhóm, không là điều kiện lọc trên tất cả dòng dữ liệu của bảng

Chỉ thực hiện khi có GROUP BY

Sử dụng các hàm kết hợp trong mệnh đề SELECT để gom nhóm

CÁC HÀM TÍNH TOÁN **48**

Count()

Tìm giá trị lớn nhất

MAX

Đếm số giá trị khác null

Tìm giá trị nhỏ nhất

MIN

Hàm tính tổng các giá trị

SUM AVG

Tính giá trị trung bình

CÁC HÀM TÍNH TOÁN **49**

Đặc điểm

1. Nhận tên một cột

Một giá trị Trả về

2. Hàm sum, avg: chỉ áp dụng cho trường kiểu số

3. Hàm count, min, max có thể áp dụng cho trường kiểu số và kiểu dữ liệu khác

4. Chỉ có hàm count(\*) thực hiện được trên giá trị null, đếm số dòng 5. Mệnh đề SELECT chứa hàm tính toán nếu

Có GROUP BY 🡪 Chỉ được liệt kê thuộc tính đơn trong group by

CÁC HÀM TÍNH TOÁN

**50**

Count star vs Count column vs Count 1

SELECT COUNT(\*) FROM TABLE1

COUNT(\*): Đếm số lượng dòng của table = COUNT(1)

COUNT(column): Đếm số lượng giá trị not-null của column trong table Nếu run count(column) với column là non-null thì RDBMS(depends) sẽ convert nó thành count(\*). Bởi vì performance cao hơn. Nó không phải kiểm tra giá trị của mỗi dòng, chỉ cần đếm số dòng.

COUNT(distinct column): đếm số giá trị UNIQUE-NOT NULL trong column.

WHERE – OR - AND

**51**

OR = hoặc : hoặc thỏa điều kiện này hoặc thỏa điều kiện khác AND = và : Thỏa cùng lúc điều kiện này và điều kiện khác

*Ví dụ: Tìm những mặc hàng có số lượng lớn hơn 20 hoặc có giá bán là 150.000*

**SELECT\***

**FROM** MatHang

**WHERE** SoLuong >= 20 OR GiaBan = 150000



WHERE – OR - AND

**52**

*Ví dụ: Tìm những chiếc đầm có số bán là dưới 250.000*

**SELECT \***

**FROM** MatHang

**WHERE** TenMH like N’%đầm%’ AND GiaBan < 250000



UNION - INTERSECT **53**

UNION : phép hội

**SELECT** <danh sách tên cột> **FROM** <Tên bảng 1>

**WHERE** <điều kiện1>

UNION

**SELECT** <danh sách tên cột> **FROM** <Tên bảng 2>

**WHERE** <điều kiện2>

**SELECT** MaKH

**FROM** KhachHang **WHERE** LoaiKH = 2 UNION

**SELECT** KhachHang **FROM** DonHang

Số column được chọn từ những table phải bằng nhau và nên cùng là một column

*Tìm khách hàng có mã loại = 2 hoặc khách hàng đã mua hàng*

UNION - INTERSECT **54**

INTERSECT: phép giao

**SELECT** <danh sách tên cột>

**FROM** <Tên bảng1>

**WHERE** <điều kiện1>

INTERSECT

**SELECT** <danh sách tên cột>

**FROM** <Tên bảng 2>

**WHERE** <điều kiện2>

**SELECT** MaKH

**FROM** KhachHang **WHERE** LoaiKH = 2 INTERSECT

**SELECT** KhachHang **FROM** DonHang

*Tìm khách hàng có mã loại = 2 và đã mua hàng*

UNION - INTERSECT

**55**

Notes:

Trả về kết quả không trùng nhau

Dể lấy tất cả các kết quả, thêm từ khóa ALL vào - **UNION** ALL

- INTERSECT ALL

- **ALREADY DEFAULT**

PHÉP KẾT

**56**

****PHÉP KẾT

Làm thế nào để lấy thông tin chi tiết (tên mặt hàng, số lượng, giá bán) các mặt hàng trong đơn hàng?

PHÉP KẾT

**57**

- Dùng khi truy xuất dữ liệu từ nhiều bảng - Liên kết 2 hay nhiều bảng với nhau dựa trên 1 thuộc tính(khóa ngoại)

Các loại phép kết

Phép kết tự nhiên **JOIN**

Phép kết trái

**LEFT JOIN**

Phép kết trong **INNER JOIN**

Phép kết phải

**RIGHT JOIN**

Phép kết ngoài **OUTER JOIN**

PHÉP KẾT

**58**

**SELECT** <danh sách thuộc tính>

**FROM** Table1 tbl1 **JOIN** Table2 tbl2 **ON** tbl1.ThuocTinh = tb2.ThuocTinh **OUTER JOIN INNER JOIN LEFT JOIN RIGHT JOIN**

| **MaMH TenMH Loai** |
| --- |
| **001** Áo sơ mi A |
| **002** Đầm B |
| **003** Jean C |
| **004** Áo thun A |

| MaMH | SoHD | SoLuong |
| --- | --- | --- |
| 001 | 001 | 3 |
| 003 | 002 | 2 |

JOIN

**59**

**SELECT** \*

**INNER JOIN**

**FROM** Mathang a **JOIN** ChiTietDonHang b **ON** a.maMH = b.MaMH

| **MaMH TenMH Loai** |
| --- |
| **001** Áo sơ mi A |
| **002** Đầm B |
| **003** Jean C |
| **004** Áo thun A |

| MaMH | SoHD | SoLuong |
| --- | --- | --- |
| 001 | 001 | 3 |
| 003 | 002 | 2 |

| **MaMH TenMH Loai MaMH SoHD SoLuong** |
| --- |
| **001** Áo sơ mi A 001 001 3 |
| **003** Jean C 003 002 2 |

LEFT JOIN

**60**

**SELECT** \*

**FROM** Mathang a **LEFT JOIN** ChiTietDonHang b **ON** a.maMH = b.MaMH

| **MaMH TenMH Loai** |
| --- |
| **001** Áo sơ mi A |
| **002** Đầm B |
| **003** Jean C |
| **004** Áo thun A |

| MaMH | SoHD | SoLuong |
| --- | --- | --- |
| 001 | 001 | 3 |
| 003 | 002 | 2 |

| **MaMH TenMH Loai MaMH SoHD SoLuong** |
| --- |
| **001** Áo sơ mi A 001 001 3 |
| **002** Đầm B Null Null Null |
| **003** Jean C 003 002 2 |
| **004** Áo thun A Null Null null |

RIGHT JOIN

**61**

**SELECT** \*

**FROM** ChiTietDonHang a **RIGHT JOIN** Mathang b **ON** a.maMH = b.MaMH

| MaMH | SoHD | SoLuong |
| --- | --- | --- |
| 001 | 001 | 3 |
| 003 | 002 | 2 |

| **MaMH TenMH Loai** |
| --- |
| **001** Áo sơ mi A |
| **002** Đầm B |
| **003** Jean C |
| **004** Áo thun A |

| **MaMH SoHD SoLuong MaMH TenMH Loai** |
| --- |
| **001** 001 3 001 Áo sơ mi A |
| **Null** Null Null 002 Đầm B |
| **003** 002 2 003 Jean C |
| **Null** Null null 004 Áo thun A |

OUTER JOIN

**62**

**SELECT** \*

**FROM** ChiTietDonHang a **OUTER JOIN** Mathang b **ON** a.maMH = b.MaMH

| MaMH | SoHD | SoLuong |
| --- | --- | --- |
| 001 | 001 | 3 |
| 003 | 002 | 2 |
| 005 | 004 | 1 |

| **MaMH TenMH Loai** |
| --- |
| **001** Áo sơ mi A |
| **002** Đầm B |
| **003** Jean C |
| **004** Áo thun A |

| **MaMH SoHD SoLuong MaMH TenMH Loai** |
| --- |
| **001** 001 3 001 Áo sơ mi A |
| **Null** Null Null 002 Đầm B |
| **003** 002 2 003 Jean C |
| **Null** Null null 004 Áo thun A |
| **005** 004 1 Null Null Null |

TRUY VẤN CON

**63**

Là câu **truy vấn** xuất hiện trong một câu truy vấn khác

Có thể xuất hiện trong các mệnh đề WHERE, HAVING,

INSERT, UPDATE, DELETE

*Tìm khách hàng có mã loại = 2 đã mua hàng (Xuất hiện trong mệnh đề WHERE)*

**SELECT** MaKH

**FROM** KhachHang

**WHERE** LoaiKH = 2 AND MaKH IN (**SELECT**

KhachHang

**FROM** DonHang )

TRUY VẤN CON

**64**

*Cập nhật tất cả các sản phẩm thuộc loại hàng "T-Shirt" với giá bán là 99 nghìn (Xuất hiện trong mệnh đề WHERE của câu lệnh UPDATE)*

**UPDATE mathang**

**SET** GiaBan **= 99**

**WHERE** MaLoai **= (SELECT** MaLoai

**FROM loaihang**

**WHERE** TenLoai **= "T-Shirt");**

TRUY VẤN CON

**65**

*Tạo mới table LoaiHangGoc và sao chép dữ liệu từ table LoaiHang sang LoaiHangGoc (Xuất hiện trong mệnh đề INSERT)*

**INSERT INTO LoaiHangGoc(**MaLHGoc, TenLHGoc**)**

**SELECT MaLoai,TenLoai**

**FROM LoaiHang**

TRUY VẤN LỒNG

**66**

Là câu truy vấn có chứa “truy vấn con” ở mệnh đề WHERE

**SELECT** <danh sách tên cột>

**FROM** <Tên bảng>

**WHERE** <biểu thức so sánh> (**SELECT** <danh sách thuộc tính> **FROM** Tên bảng

**WHERE** <điều kiện>)

Biểu thức so sánh thường có: IN, NOT IN, ALL, ANY EXISTS, NOT EXISTS

TRUY VẤN LỒNG

**67**

Có 2 loại truy vấn lồng **Lồng phân cấp = Truy vấn con Lồng tương quan**

❖ Lồng phân cấp

**SELECT** MaKH

**FROM** KhachHang

**WHERE** LoaiKH = 2 AND MaKH IN (**SELECT** KhachHang

**FROM** DonHang

**WHERE** SoLuong > 1)

Sử dụng toán tử IN, NOT IN, =

Mệnh đề WHERE của câu truy vấn con **không tham chiếu** đến giá trị thuộc tính của các bảng ở mệnh đề FROM của truy vấn cha

Thực thi câu truy vấn con trước

TRUY VẤN LỒNG

**68**

❖ Lồng tương quan

**SELECT** MaKH

**FROM** KhachHang

**WHERE** LoaiKH = 2 AND EXISTS (**SELECT** MaKH || 1 || \*

**FROM** DonHang

**WHERE** MaKH = MaKH)

Sử dụng toán tử EXIST, NOT EXIST

Mệnh đề WHERE của câu truy vấn con tham chiếu đến ít nhất 1 thuộc tính của các bảng ở mệnh đề FROM của truy vấn cha

Thực hiện song song theo cơ chế

“**at least one found**”

TRUY VẤN LỒNG **69**

IN và EXISTS

IN

EXISTS

<tên côt> IN <câu truy vấn con> EXISTS <câu truy vấn con>

Thuộc tính trong SELECT cuả truy vấn con CÙNG kiểu dữ liệu thuộc tính trong WHERE của truy vấn cha

Không nhất thiết liệt kê thuộc tính ở mệnh đề SELECT của câu truy vấn con

TRUY VẤN LỒNG

**70**

****

INDEX in SQL

**71**

****

INDEX in SQL

**72**

****

INDEX in SQL

**73**

****

INDEX in SQL

**74**

****

INDEX in SQL

**75**

Verifying Index Usage



INDEX in SQL 

**76**

****

PROCEDURE

**77**

****

PROCEDURE

**78**

****

FUNCTION

**79**

****

FUNCTION

**80**

****

****