**CƠ SỞ DỮ LIỆU**

Cơ sở dữ liệu (Database – DB)

Hệ cơ sở dữ liệu (Database System – DBS)

3 tín chỉ = 45 tiết = 30 LT + 15 TH => 16 tuần

Kiểm tra giữa kỳ (tuần 5) – thi tự luận 3đ 🡺 Thầy chỉ chỗ cho học bài

Thi cuối kỳ (tuần 16)

* Thi lý thuyết 3đ hoặc 4đ – trắc nghiệm chung
* Thi thực hành 4đ hoặc 3đ – đề riêng

NỘI DUNG

1. Tổng quan về CSDL
2. Đại số quan hệ
3. Phụ thuộc hàm và Khóa
4. Chuẩn hóa lược đồ quan hệ
5. Thực hành với ORACE

**TỔNG QUAN VỀ HỆ CƠ SỞ DỮ LIỆU**

Tổ chức dữ liệu/thông tin để khai thác có hiệu quả là điều cần thiết

Tổ chức như thế nào? (bằng CSDL)

**1. HỆ THỐNG QUẢN LÝ DỮ LIỆU BẰNG FILE**

**1.1. Đặc điểm**

* Dữ liệu được lưu trữ trên các file riêng biệt
* Chương trình, định nghĩa và quản lý dữ liệu của chính nó

**1.2. Những nhược điểm của hệ thống quản lý dữ liệu bằng file**

* Sự dư thừa dữ liệu
* Sự mâu thuẫn dữ liệu
* Sự cô lập dữ liệu
* Không có vấn đề bảo mật
* Không điều khiển truy xuất cạnh tranh

**2. HỆ THỐNG QUẢN LÝ DỮ LIỆU BẰNG CSDL**

**2.1. Cơ sở dữ liệu:**

Có nhiều cách nói về CSDL:

* Thông thường: là tập hợp các dữ liệu mang tính chất ***dùng chung***
* Kỹ thuật: là tập hợp dữ liệu có ***cấu trúc, lớn, được lưu trữ***
* Về tác nghiệp: là tập hợp dữ liệu ***tác nghiệp*** của tổ chức

🡺 **Cơ Sở Dữ Liệu: *là tập hợp dữ liệu tác nghiệp có cấu trúc, lớn, được lưu trữ và mang tính chất dùng chung***

**2.2. Hệ quản trị CSDL - DBMS**

Hệ quản trị CSDL là tập hợp các chương trình. Chúng cho phép người dùng tạo ra CSDL, duy trì hoạt động của CSDL và điều khiển tất cả các truy xuất đến CSDL

Hệ CSDL nghĩa là bao gồm:

* CSDL
* Hệ quản trị CSDL (Oracle)

**3. CÁC KIỂU MÔ HÌNH DỮ LIỆU**

**3.1. Kiến trúc CSDL**

Có 3 mức/tầng/lớp:

* ***Mức trong***: là mức vật lý 🡪 lưu trữ *(lưu trữ ntn?)*
* ***Mức khái niệm***: là mức logic 🡪 thiết kế *(có gì ở trong?)*
* ***Mức ngoại***: là mức người dùng 🡪 view *(lấy ra)*

**3.3. Các kiểu mô hình dữ liệu**

- Mô hình dữ liệu thường được sử dụng:

- Mô hình thực thể mối quan hệ ERM

- Mô hình (dl) quan hệ *(quan trọng I)*

* Table
* Mẫu tin: dòng/cột

- Mô hình dl phân cấp

- Mô hình dl mạng

**4. HỆ QUẢN TRỊ CSDL**

**4.3. Ngôn ngữ CSDL**

- Ngôn ngữ định nghĩa dữ liệu ĐL

- Ngôn ngữ thao tác dữ liệu:

* Insert
* Update
* Delete

- Ngôn ngữ truy vấn có cấu trúc SQL: dùng để truy vấn dl cần thiết

**ĐẠI SỐ QUAN HỆ**

**1. CSDL QUAN HỆ**

**1.1. Tập hợp thuộc tính và miền giá trị**

= A1, A2, A3, …, An 🡺 Tập hợp các thuộc tính

= D1, D2, D3, …, Dn 🡺 Tập hợp các miền giá trị

dom(Ai) = Di (i =1🡪n ): miền Di lấy giá trị tại thuộc tính Ai

**1.2. Quan hệ**

Quan hệ là 1 tập hợp con của tích Descartes

***R = {t = (d1, d2, …, dn) | d1 D1, d2 D2, .., dn Dn}***

**1.3. Bảng (Table)**

* Cột (thuộc tính)
* Dòng (bộ)

🡺CSDL quan hệ là CSDL mà trong đó dl được tổ chức dưới dạng bảng

* *Bảng là hình ảnh trực quan của quan hệ*
* *Quan hệ có thể biểu diễn bằng bảng*

**1.4. Bậc và lực lượng của 1 quan hệ**

Bậc: số cột

Lực lượng: số dòng

**1.5. Lược đồ quan hệ và thể hiện của quan hệ**

***\*Lược đồ qh: s*()**

* Tổng quát (cột/thuộc tính)
* Ổn định 🡪 Tương đối

🡺Lược đồ mô tả tổng quát các quan hệ bằng cách mô tả các cột

***\*Thể hiện:***

* Cụ thể (dòng/bộ)
* Thay đổi🡪tương đối

***\*Lược đồ CSDL***

* Là tập hợp tất cả các lược đồ quan hệ và các mối kết hợp trong CSDL đó

***\*Thể hiện của CSDL***

* Là nội dung thật sự của CSDL tại 1 thời điểm

**3. ĐẠI SỐ QUAN HỆ**

**4. PHÉP TOÁN QUAN HỆ**

**4.1. Phép chọn**

**4.2. Phép chiếu**

**4.3. Phép hợp**

**4.4 Phép giao**

**4.5. Phép trừ \**

**4.6. Phép chia /**

**4.7. Tích Descartes**

**4.8. Phép kết nối trong ►◄**

**4.9. Phép kết nối tự nhiên \***

**4.10. Phép kết nối mở rộng**

**4.11. Phép đặt lại tên**

**4.12. Phép gán 🡨**

**5. CÁC HÀM KẾT TẬP**

**5.1. Các hàm kết tập**

AVG: Tính giá trị trung bình

MIN: Tính giá trị nhỏ nhất

MAX: Tính giá trị lớn nhất

SUM: Tính tổng

COUNT: Đếm số giá trị

**5.2. Hàm kết tập trên nhóm**

**PHỤ THUỘC HÀM VÀ KHÓA**

**1. RÀNG BUỘC TOÀN VẸN**

**1.2. Tính hợp lý trên CSDL**

Ràng buộc vẹn toàn, gọi tắt là ràng buộc, để đảm bảo tính hợp lý của CSDL

Trong 1 CSDL, luôn tồn tại nhiều ràng buộc

* Giữa các thuộc tính
* Giữa các bộ
* Giữa các quan hệ

Các hệ quản trị CSDL có các câu lệnh để định nghĩa và quản lý các ràng buộc toàn vẹn:

* Thêm (Insert)
* Sửa (Update)
* Xóa (Delete)

**1.3. Biểu diễn ràng buộc toàn vẹn**

Nội dung của RBTV có thể được biểu diễn bằng

* Ngôn ngữ tự nhiên
* Ngôn ngữ giả
* Đại số quan hệ
* ***Phụ thuộc hàm***: 🡪Là công cụ biểu diễn RBTV

**1.4. Phạm vi của RBTV**

Mỗi RBTV có tác dụng trên 1 số các quan hệ, đgl phạm vi của nó

**1.5. Phân loại RBTV**

Phân theo ***ý nghĩa*** hoặc theo ***phạm vi***

+Theo ***ý nghĩa***:

* Toàn vẹn thực thể
* Toàn vẹn miền trị
* Toàn vẹn tham chiếu
* Toàn vẹn về luận lý

+Theo ***phạm v***i:

* RBTV trên 1 quan hệ
* RBTV trên nhiều quan hệ

**1.6. Bảng tầm ảnh hưởng tổng hợp**

- 3 ***thao tác cập nhật*** là: ***Xen, Sửa, Xóa***

RBTV *C* có phạm vi là các quan hệ R1, R2, …, Rn.

* Dấu + cho biết cần phải kiểm tra C *(kiểm tra rồi mới làm)*
* Dấu – cho biết ko cần kiểm tra C *(cứ làm)*
* Dấu \* cho biết ko đc thao tác đối với giá trị khóa *(cứ làm nhưng trừ khóa)*

**2. PHỤ THUỘC HÀM**

**2.1. Phụ thuộc hàm**

**THỰC HÀNH VỚI ORACLE**

**2. NGÔN NGỮ TRUY VẤN CÓ CẤU TRÚC SQL**

**2.1. Cú pháp câu lệnh SQL theo BNF (Backus – Naur Form)**

Câu lệnh SQL thường gồm có 2 thành phần: ***từ khóa*** và  ***từ do người dùng*** định nghĩa. Cú pháp của câu lệnh SQL đc trình bày theo dạng mở rộng của BNF có những qui ước như sau:

1. ***Chữ hoa*** đại diện cho từ khóa
2. ***Chữ thường*** đại diện cho các từ do người dùng định nghĩa
3. Ký hiệu ***A | B*** chỉ sự lựa chọn hoặc là A hoặc là B
4. Ký hiệu ***{…}*** chỉ thành phần bắt buộc phải có
5. Ký hiệu ***[…]*** chỉ thành phần tùy chọn (ko bắt buộc)
6. Ký hiệu ***…..….*** chỉ thành phần có thể lặp lại

Tuy câu lệnh SQL có hình thức tự do nhưng ***dễ đọc*** ng ta khuyên nên:

1. Viết mỗi mệnh đề của lệnh trên 1 dòng riêng
2. Viết bắt đầu 1 mệnh đề thẳng hàng với các mệnh đề khác
3. Nếu 1 mệnh đề có nhiều vế thì nên viết mỗi vế trên 1 dòng và thụt vào trong mệnh đề đó

**2.5. CSDL dùng cho các ví dụ**

**3. USER VÀ SCHEMA TRONG ORACLE**

- User là 1 tài khoản trong CSDL Oracle

- Schema là 1 tập hợp các object trong CSDL Oracle, đc quản lý bởi 1 user nào đó.

- Schema đc tạo ra tự động cùng lúc với User khi thực thi lệnh CREATE USER

- Mối qhe giữa user và schema là 1-1

- Để tạo ra 1 schema khả dụng trong Oracle chúng ta cần thực hiện các bước sau:

1. Tạo ra 1 user mới trong Oracle
2. Gán đặc quyền hệ thống cho user mới
3. Tạo ra các object trong schema
4. Cấp cho object các đặc quyền
5. Tạo ra từ đồng nghĩa cho các object

**3.1. Tạo và xóa user**

-***Tạo*** ra 1 user:

**CREATE USER** username

IDENTIFIED BY password

[DEFAULT TABLESPACE tablespace]

[QUOTA {size | UNLIMITED} ON tablespace]

[PROFILE profile]

[PASSWORD EXPIRE]

[ACCOUNT {LOCK | UNLOCK}];

-***Xóa*** 1 user

**DROP USER** username [CASCADE]

**3.2. Gán và thu hồi đặc quyền hệ thống cho user mới**

-***Cấp*** đặc quyền cho user:

**GRANT** {system\_privileges | object\_privileges} TO username;

WITH ADMIN OPTION]

+*system\_privileges* có thể là các giá trị sau đây

* CREATE SESSION
* CREATE TABLE
* CREATE VIEW
* CREATE PROCEDURE
* SYSDBA
* SYSOPER
* ALL PRIVILEGES

+*object\_privileges* có thể là các giá trị sau đây

* INSERT
* UPDATE
* DELETE
* INDEX
* EXECUTE

-***Thu hồi*** đặc quyền đã cấp cho user

**REVOKE** {system\_privileges | object\_privileges} FROM user;

***VD:*** Tạo ra 1 user *qlnv* có password là B2105569 và cấp mọi đặc quyền cho user đó:

CREATE USER qlnv INDENTIFIED BY ‘B2105569’ ;

GRANT ALL PRIVILEGES TO qlnv

**3.3. Tạo và xóa bảng (table)**

-***Tạo table***

**CREATE TABLE** schema\_name.table\_name **(**

column\_1 data\_type column\_constrain**,**

column\_2 data\_type column\_constrain**,**

...

table\_constrain

**);**

+Ràng buộc đối với cột (column\_constrain) thường có các tùy chọn sau:

* NOT NULL: cột ko đc nhận dữ liệu NULL
* DEFAULT: gán giá trị mặc định cho cột
* UNIQUE: đảm bảo giá trị của cột ko trùng lặp
* PRIMARY KEY: thiết lập khóa chính
* REFERENCES: thiết lập khóa ngoại
* CHECK: đảm bảo giá trị của cột thỏa 1 điều kiện
* ......................

-***Xóa bảng***

**DROP TABLE** schema\_name.table\_name**;**

***\*Chú ý:*** Thứ tự tạo bảng là quan trọng

***VD:*** Tạo bảng tên PHONG

CREATE TABLE phong (

maphong Number NOT NULL PRIMARY KEY,

tenphong VarChar2(20),

truongphong VarChar2(3),

);

***VD:*** Tạo bảng tên NHANVIEN

CREATE TABLE nhanvien(

manv VarChar2(3) NOT NULL PRIMARY KEY,

hoten VarChar2(25),

phai VarChar2(3),

ngaysinh Date,

congviec VarChar2(15) NOT NULL,

luong Number(8,2),

phucap Number(8,2),

maphong Number NOT NULL REFERENCES phong(maphong)

);

**3.4. Thêm 1 dòng vào bảng**

**INSERT INTO** table\_name (column\_list) VALUES (value\_list)**;**

**INSERT INTO** table\_name VALUES (value\_list);

***VD:***

+ Thêm vào bảng PHONG thông tin của phòng mới có: mã phòng 50, tên phòng là Phòng Thiết Bị và nhân viên có mã số 063 là trưởng phòng

INSERT INTO phong VALUES (50, ‘Phong Thiet Bi’, ‘063’);

+ Thêm vào bảng NHANVIEN 1 nhân viên có mã số 123, tên Nguyễn Văn A, phái nam, mã phòng 50 và làm công việc quản lý kho

INSERT INTO nhanvien (manv, hoten, phai, maphong, congviec)

VALUES (‘123’, ‘Nguyen Van A’, ‘nam’, 50, ‘Quan Ly Kho’);

**3.5. Thay đổi cập nhật/ dữ liệu đã có trong bảng**

**UPDATE** table\_name

**SET**

column1 = value1,

column2 = value2,

column3 = value3,

...

**[WHERE** condition**];**

***VD:***

+Chuyển nhân viên Lê Quỳnh Như sang phòng 20 với công việc là Thư ký

UPDATE nhanvien

SET

maphong = 20,

congviec = ‘Thu ky’

WHERE hoten = ‘Le Quynh Nhu’;

+Chuyển tất cả nhân viên sang phòng 40

UPDATE nhanvien

SET maphong = 40;

**3.6. Xóa dòng trong bảng**

Xóa 1 hoặc nhiều dòng trong bảng:

**DELETE [FROM]** table\_name

**[WHERE condition];**

***VD:*** Nhân viên Lê Quỳnh Như đã nghỉ việc. Hủy bỏ dòng thông tin tương ứng

DELETE nhanvien

WHERE hoten = ‘Le Quynh Nhu’;

***VD:*** Hủy bỏ tất cả thông tin (dòng) trong bảng TDONN

DELETE tdonn;

\****Chú ý:*** Ko còn dòng nhưng vẫn còn bảng 🡺 Bảng rỗng

**3.7. Sửa đổi cấu trúc (các cột) của bảng**

**ALTER TABLE** table\_name **action;**

*+action*: chỉ dẫn hành động mà muốn thực hiện lên bảng

* Add one or more columns
* Modify column definition
* Drop one or more columns
* Rename columns
* Rename table

***VD:***

+Thêm vào bảng NHANVIEN cột tên MOBILE kiểu Varchar2(8) sau đó đổi thành kiểu Varchar2(11), đổi tên cột thành DTDD, cuối cùng xóa nó đi

ALTER TABLE NHANVIEN ADD MOBILE Varchar2(8);

ALTER TABLE NHANVIEN MODIFY MOBILE Varchar2(11);

ALTER TABLE NHANVIEN RENAME COLUMN MOBILE TO DTDT;

ALTER TABLE NHANVIEN DROP COLUMN DTDT;

+Thêm ràng buộc NOT NULL cho cột trinhdo trong bảng TDNN

ALTER TABLE TDONN MODIFY (trinhdo Varchar2(1) NOT NULL);

+Thêm ràng buộc khóa chính PRIMARY KEY gồm 2 cột manv và mann cho bảng TDONN

ALTER TABLE TDONN CONSTRAINT pk\_nv\_nn PRIMARY KEY (manv, mann);

**3.8. Hủy bỏ bảng khỏi CSDL (schema)**

**DROP TABLE** schema\_name.table\_name

[CASCADE CONSTRAINTS | PURGE];

***VD:*** Hủy bỏ bảng tên NHANVIEN

DROP TABLE nhanvien;

**4. TRUY VẪN DỮ LIỆU**

Truy vấn dl là 1 trong các thao tác cơ bản, đc thực hiện bởi câu lệnh **SELECT**. Câu lệnh này gồm các mệnh đề chính như sau:

**SELECT** {Select\_list}

**FROM** {Table\_source\_list}

[**WHERE** {Search\_condition}]

[**GROUP BY** {Group\_by\_expression}]

[**HAVING** {Group\_condition}]

[**ORDER BY** {Ordered\_expression} [ASC | DESC] ]

**4.1. Truy vấn dữ liệu từ 1 bảng, định thứ tự xuất hiện các cột**

-TVDL từ 1 hoặc nhiều cột của 1 bảng:

**SELECT** [DISTINCT]

column\_1,

column\_2,

...

**FROM** table\_name;

+Muốn truy vấn tất cả các cột của bảng thì nên dùng dấu **\***

+Mệnh đề **DISTINCT**: lọc những dòng trùng lặp trong bảng kq *(loại bỏ những dòng giống nhau, chỉ giữ lại 1)*

***VD:***

+Cho biết toàn bộ thông tin (tất cả các cột) của các nhân viên

SELECT \*

FROM nhanvien;

+Cho biết danh sách gồm họ tên và phái của nhân viên

SELECT hoten,

phai

FROM nhanvien;

+Cho biết danh sách gồm các mức lương có thể có của nhân viên (*Cần loại bỏ các mức lương giống*)

SELECT **DISTINCT** luong

FROM nhanvien;

**4.2. Sắp xếp kết quả truy xuất (dòng) theo 1 hay nhiều cột**

- Để sắp xếp kết quả truy vấn ta sử dụng mệnh đề **ORDER BY** ở cuối lệnh SELECT như sau:

**SELECT** [DISTINCT]

column\_1,

column\_2,

...

**FROM** table\_name

**ORDER BY**

column\_1 [ASC | DESC] [NULLS FIRST | NULLS LAST],

column\_1 [ASC | DESC] [NULLS FIRST | NULLS LAST],

...

;

* ASC: chỉ định việc sắp thứ tự tăng dần. Đây là giá trị mặc định
* DESC: chỉ định việc sắp thứ tự giảm dần
* NULL FIRST: chỉ định việc sắp giá trị NULL trước giá trị NON-NULL
* NULL LAST: chỉ định việc sắp giá trị NULL sau giá trị NON-NULL

***VD:***

+Cho biết toàn bộ thông tin của nhân viên theo mã phòng tăng dần , và lương giảm dần

SELECT \*

FROM nhanvien

**ORDER BY** maphong,

luong DESC;

+Cho biết họ tên, lương, phụ cấp và thu nhập trong năm của tất cả nhân viên, sắp xếp thu nhập trong năm giảm dần (đặt tên cho cột)

SELECT hoten,

luong,

phucap,

(NVL(luong,0)+NVL(phucap,0))\*12 thunhap

FROM nhanvien

ORDER BY 4 DESC

**4.3. Truy vấn thỏa điều kiện**

- Sử dụng mệnh đề **WHERE** sau mệnh đề FROM

SELECT select\_list

FROM table\_name

**WHERE** **search\_condition**

+**search\_condition**: thường có các dạng sau:

1. So sánh gtr thuộc tính với 1 gtr cụ thể:

<Cột> so sánh <gtr>

1. So sánh gtr của 2 thuộc tính với nhau:

<Cột 1> so sánh <Cột 2>

1. So sánh gtr thuộc tính với hằng có kiểu tương thích:

<Cột> so sánh <Hằng>

1. So sánh cột với 1 tập hợp gồm nhiều gtr:

<Cột> **IN** (<Danh sách các gtr>)

<Cột> NOT IN (<Danh sách các gtr>)

1. Kiểm tra điều kiện theo phạm vi:

<Cột> **BETWEEN** <Gtr đầu> AND <Gtr cuối>

<Cột> NOT BETWEEN <Gtr đầu> AND <Gtr cuối>

1. Kiểm tra điều kiện theo mẫu dạng chuỗi:

<Cột> **LIKE** <Chuỗi>

<Cột> NOT LIKE <Chuỗi>

Trong <Chuỗi> có thể có ký tự đại diện:

**\_** đại diện cho ký tự bất kỳ

**%** đại diện cho 1 chuỗi ký tự bất kỳ

1. Kiểm tra điều kiện theo gtr NULL

<Cột> **IS NULL**

<Cột> IS NOT NULL

***VD:***

+Cho biết họ tên và công việc của các nhân viên nam đang làm việc trong các phòng có mã phòng 30 hoặc 40

SELECT hoten,

congviec

FROM nhanvien

WHERE phai = ‘Nam’

AND (maphong=30 OR maphong=40);

+Cho biết họ tên, lương và phụ cấp của những nhân viên có lương bằng hai lần phụ cấp

SELECT hoten,

luong,

phucap

FROM nhanvien

WHERE luong = 2\*phucap;

+Cho biết mã và họ tên các nhân viên nam làm công việc là THU KY (Thu ky)

SELECT ma,

hoten

FROM nhanvien

WHERE phai = ‘Nam’

AND UPPER(congviec) = ‘THU KY’;

+Cho biết họ tên các nhân viên nữ không làm việc ở một trong các phòng 10, 30 hoặc 50

SELECT hoten

FROM nhanvien

WHERE phai = ‘Nu’

AND (maphong NOT IN(10, 30, 50));

+Cho biết họ tên và mức lương của các nhân viên có mức lương từ 3500 đến 4500

+Cho biết mã, họ tên và tuổi của các nhân viên có tên lót là “Van”, và sắp xếp theo tuổi giảm dần

+Cho biết toàn bộ thông tin của các nhân viên nữ mà họ tên của họ có vần cuối là ‘ao’ như Bao, Thao, Hao…

+Cho biết toàn bộ thông tin của các nhân viên ko có phụ cấp

+Hiển thị họ tên và lương của các nhân viên nữ 25 tuổi

**4.4. Các hàm kết tập (Aggregate Functions)**

1. Tính giá trị trung bình của các biểu thức

**AVG**

1. Đếm số biểu thức

**COUNT**

1. Tính giá trị lớn nhất

**MAX**

1. Tính giá trị nhỏ nhất

**MIN**

1. Tính tổng

**SUM**

\****Chú ý:*** Trong mđ SELECT nếu có dùng hàm kết tập thì các tên cột phải được đặt trong hàm kết tập

**4.5. Nhóm dữ liệu, thực hiện hàm kết tập trên từng nhóm**

- Sử dụng mđ **GROUP BY** sau mđ FROM hoặc WHERE

- Mđ SELECT và GROUP BY có liên quan chặt chẽ với nhau.

- Cột nào nằm trong GROUP BY thì mới có thể nằm trong SELECT

SELECT [column\_list,] {function\_list}

FROM Table\_name

WHERE condition

GROUP BY c1, c2, c3, ...;

**4.6. Lọc các nhóm thỏa điều kiện**

- Mđ **HAVING** theo sau mđ GROUP BY

SELECT [column\_list,] {function\_list}

FROM Table\_name

WHERE condition

GROUP BY Column\_name

HAVING group\_condition;

**5. TRUY VẤN TRÊN NHIỀU BẢNG**

- Có nhiều bảng khác nhau đc liệt kê sau mđ FROM

- Mđ WHERE để tạo ra các kết nối giữa nhiều bảng

- Thường có đặc tả tên bảng cho các cột để tránh sự lẫn lộn giữa các cột trùng tên theo cú pháp:

**TableName**.ColumnName