

NGÔN NGỮ SQL

Biên soạn: Nguyễn Việt Hưng

Bộ môn: Khoa Học Máy Tính - Khoa Công Nghệ Thông Tin
Trường Đại Học Giao Thông Vận Tải

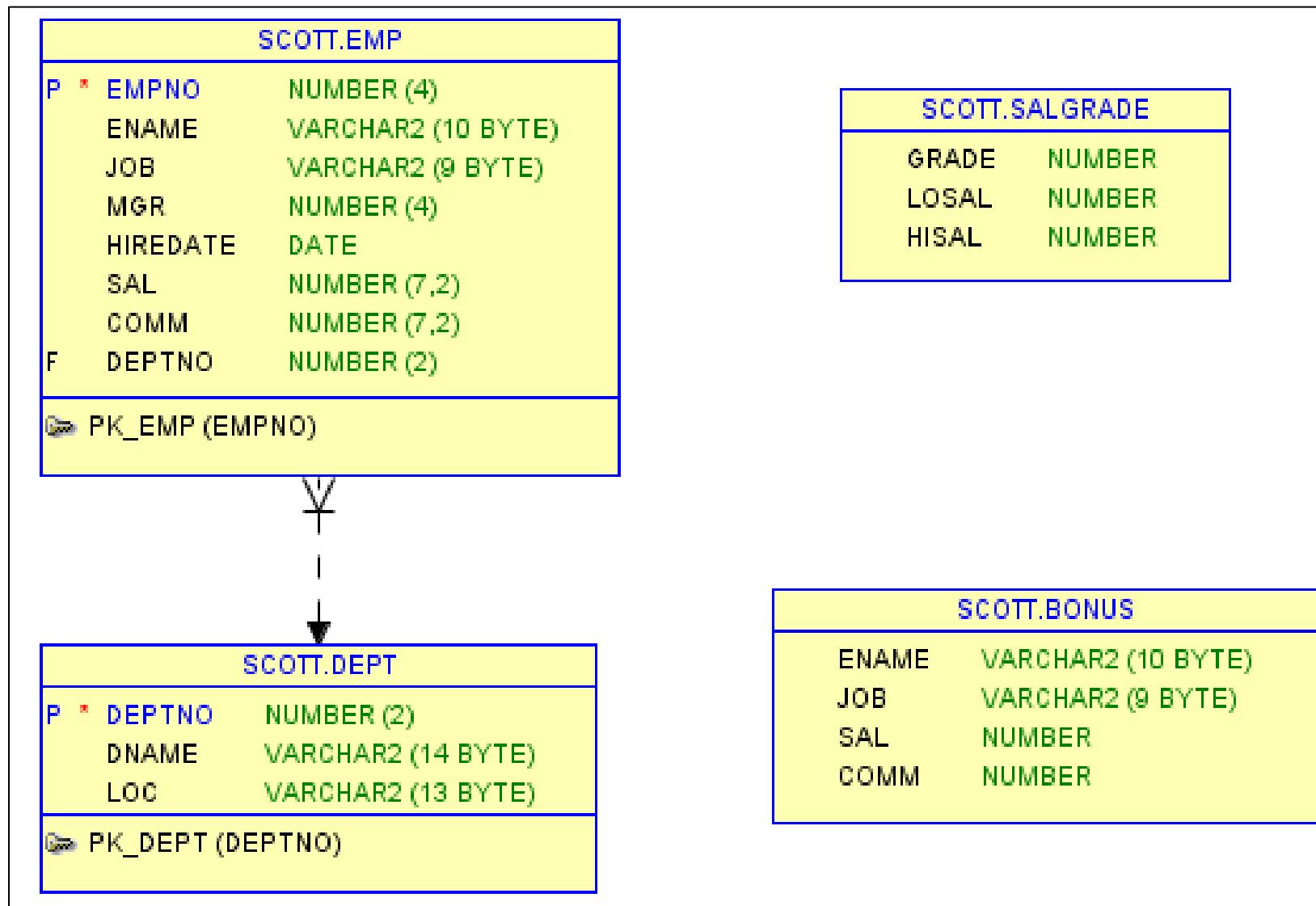
Website: <https://sites.google.com/site/viethung92gtvt/oracle>

Email: viethung92gtvt@gmail.com

- **Ngôn ngữ định nghĩa dữ liệu (DDL – Data Definition Language):** tạo table (bảng), view (khung nhìn), sửa cấu trúc table và thêm, xóa RBTV, xóa table, xóa view, đổi tên table.
- **Ngôn ngữ thao tác dữ liệu (DML – Data Manipulation Language):** thêm, xóa, sửa dữ liệu, và truy vấn dữ liệu (insert, delete, update, select).
- **Ngôn ngữ điều khiển dữ liệu (DCL – Data Control Language):** tạo quyền hạn, xóa quyền, làm cho hiệu lực/mất hiệu lực quyền, tạo người dùng, đổi mật khẩu, xóa người dùng, cấp quyền và thu hồi quyền sử dụng trên cơ sở dữ liệu.
- Ngoài ra còn có các lệnh điều khiển: lệnh COMMIT, lệnh ROLLBACK, lệnh SAVEPOINT, lệnh AUTOCOMMIT.
- Lệnh thao tác trên các thành phần CSDL khác: Synonym, Index và Sequence.

How to Install Scott Schema in Oracle 19C: <https://youtu.be/TU0P7nQpdqg>

HIỂN THỊ TÊN NHÂN VIÊN, TÊN NGƯỜI QUẢN LÝ CỦA NHÂN VIÊN ĐÓ?



❖ Câu lệnh truy vấn cơ bản

SELECT [DISTINCT] {*, column [alias],} FROM table;

- DISTINCT Phân biệt nội dung giữa các dòng dữ liệu trả về
 - * Lấy tất cả các cột trong bảng
 - column Tên cột dữ liệu cần trả về
 - alias phần tiêu đề của cột dữ liệu trả về, ko được có dấu cách.
 - Nếu muốn có dấu cách, cần phải đặt trong cặp dấu ngoặc kép ""
 - table Tên bảng cần truy vấn dữ liệu

❖ Câu lệnh truy vấn cơ bản

Các thành phần khác có thể đưa vào mệnh đề SELECT trong câu lệnh truy vấn

❑ Biểu thức toán học:

Các toán tử được dùng là (+), (-), (*), (/). Độ ưu tiên của các toán tử giống trong phần số học.

Ví dụ: Lương và thưởng của các nhân viên trong 1 năm.

```
SELECT ename, sal *12, comm FROM emp;
```

❑ Tiêu đề của cột (column alias), ghép tiếp các cột dữ liệu và ghép tiếp các chuỗi ký tự

Toán tử ghép tiếp chuỗi (||) cho phép ghép tiếp dữ liệu trong các cột khác nhau của cùng một dòng dữ liệu với nhau thành một chuỗi hoặc ghép với các chuỗi ký tự.

VD: select empno || ename || ‘work in department’ || deptno || ‘employee detail’ from emp;

❖ Mệnh đề WHERE

Cú pháp:

SELECT [DISTINCT] {*, column [alias],...} FROM table [WHERE condition (s)];

Mệnh đề WHERE dùng để đặt điều kiện cho toàn bộ câu lệnh truy vấn. Trong mệnh đề WHERE có thể có các thành phần:

- Tên column
- Toán tử so sánh
- Tên column, hằng số hoặc danh sách các giá trị.

Ví dụ:

select deptno, job, ename, sal from emp where sal between 1000 and 2000 ;

❖ Mệnh đề ORDER BY

SELECT [DISTINCT] {*, column [alias],...} FROM table [WHERE condition] [**ORDER BY expr/position [DESC/ASC]**];

Mệnh đề ORDER BY dùng để sắp xếp số liệu được hiển thị và phải đặt ở vị trí sau cùng của câu lệnh truy vấn.

Ví dụ:

*select ename, job, sal*12, deptno from emp order by ename;*

❖ Mệnh đề GROUP BY [HAVING]

Cú pháp:

SELECT [DISTINCT] {*, column [alias],...} FROM table [WHERE condition] [GROUP BY expr] [HAVING condition] [ORDER BY expr/position [DESC/ASC]];;

Mệnh đề GROUP BY sẽ nhóm các dòng dữ liệu có cùng giá trị của expr. Ví dụ: GROUP BY JOB nghĩa là sẽ nhóm các nghề giống nhau.

Mệnh đề HAVING là đặt điều kiện của nhóm dữ liệu. Có thể đặt ngay trước hoặc ngay sau mệnh đề GROUP BY. Mệnh đề này khác mệnh đề WHERE ở chỗ mệnh đề WHERE đặt điều kiện cho toàn bộ câu lệnh SELECT.

VD: *select job, max(sal) from emp group by job having max(sal)>=3000;*

a. Kết hợp dữ liệu từ nhiều bảng

Mối liên kết tương đương (kết nối tự nhiên)

Mối liên kết tương đương được thể hiện trong mệnh đề WHERE hoặc thông qua INNER JOIN trong mệnh đề FROM.

VD: Liệt kê các nhân viên có dname=RESEARCH

Cách 1: `select emp.* , dname from scott.emp, scott.dept where emp.deptno = dept.deptno and dname='RESEARCH' ;`

Cách 2: `select emp.* , dname from scott.emp inner join scott.dept on emp.deptno=dept.deptno where dname='RESEARCH' ;`--Có thể bỏ từ khóa inner

a. Kết hợp dữ liệu từ nhiều bảng

Liên kết của bảng với chính nó (tự thân)

Có thể liên kết bảng với chính nó bằng cách đặt alias.

VD: Hiển thị thông tin bao gồm tên nhân viên, lương nhân viên, tên người quản lý của nhân viên đó, lương người quản lý đó với điều kiện lương của nhân viên lớn hơn lương người quản lý nhân viên đó.

```
Select e.ename ten_nhan_vien, e.sal luong,
m.ename ten_quan_ly, m.sal luong_quan_ly from
scott.emp e, scott.emp m where e.mgr = m.empno
and e.sal > m.sal;
```

b. Lệnh truy vấn lồng nhau

- *Câu lệnh SELECT lồng nhau*

+ Trong mệnh đề WHERE

VD: Tìm những nhân viên làm cùng nghề với BLAKE

```
Select ename, job from scott.emp where job = (select job  
from scott.emp where ename = 'BLAKE');
```

+ Trong mệnh đề HAVING

VD: Tìm những phòng có mức lương trung bình lớn hơn phòng 30.

```
Select dept.deptno, dname, avg(sal) from scott.emp,  
scott.dept where emp.deptno = dept.deptno group by  
dept.deptno, dname having avg(sal) > (select avg(sal) from  
scott.emp where deptno = 30);
```

b. Lệnh truy vấn lồng nhau

- *Toán tử SOME/ANY/ALL/NOT IN/EXISTS*

TÊN TOÁN TỬ	DIỄN GIẢI
NOT IN / IN	Không thuộc / Thuộc
ANY và SOME	So sánh một giá trị với mỗi giá trị trong kết quả trả về của câu truy vấn con. Trả về đúng khi phép so sánh với bất kỳ giá trị nào là đúng.
ALL	So sánh một giá trị với mọi giá trị trong danh sách hay trong kết quả trả về của câu hỏi con. Trả về đúng khi phép so sánh với tất cả các giá trị đều đúng.
EXISTS	Trả về TRUE nếu có tồn tại.

b. Lệnh truy vấn lồng nhau

- *Toán tử SOME/ANY/ALL/NOT IN/EXITS*

VD: Hiển thị các nhân viên có lương lớn hơn các nhân viên ở phòng 30.

```
Select * from scott.emp where sal >= all  
(select distinct sal from scott.emp where  
deptno = 30) order by sal desc;
```

- **Các hàm xử lý chuỗi, ký tự**

LENGTH(<chuỗi>): trả về độ dài chuỗi, nếu chuỗi là trống hoặc là NULL, thì hàm sẽ trả về NULL.

VD: `select length('ORACLE') from dual;` => 6

+ LOWER(*string1*): chuyển đổi tất cả các ký tự trong chuỗi *string1* thành ký tự thường.

VD: `select LOWER('ORACLE') from dual;` =>
oracle

+ UPPER(*string1*): ngược lại của LOWER

- *Các hàm số học:*

- + ABS(n): Trị tuyệt đối của n

VD: SELECT ABS (-15) "Absolute" FROM
DUAL;

- + MOD(a,b): Lấy phần dư của a chia cho b

VD: SELECT MOD (11, 4) "Modulus" FROM
DUAL;

- + ROUND(n,i): Làm tròn số n tới i chữ số thập phân.

- + POWER(n,i): lũy thừa i của n

- **Các hàm chuyển đổi kiểu:**

- + TO_NUMBER (<chuỗi số>): Chuyển ký tự có nội dung số sang số
- + TO_CHAR(<value>[, format_mask]): chuyển đổi một giá trị số hoặc ngày tháng sang chuỗi. Trong đó, *value* là giá trị cần chuyển, *format_mask* là định dạng sẽ sử dụng để chuyển đổi.

VD: `select TO_CHAR(1210.73, '9999.9') from dual;`

`=> 1210.7`

`select TO_CHAR(1210.73, '$9,999.99') from dual;`

`=> $1,210.73`

`select TO_CHAR(sysdate, 'yyyy/mm/dd') from dual;`

- + TO_DATE(<value> [, format_mask]): chuyển đổi một chuỗi sang định dạng ngày tháng.

- **Các hàm xử lý ngày tháng**

+ EXTRACT(YEAR | MONTH | DAY FROM <ngày>): Trả về thành phần, ngày, tháng, hoặc năm của một dữ liệu kiểu date.

VD: `SELECT EXTRACT (YEAR FROM TO_DATE ('3/3/2016', 'dd/mm/yyyy')) FROM DUAL;`
`(=>2016)`

+ ADD_MONTHS(<ngày x>,<số tháng n>): Trả về ngày mới sau khi cộng n tháng vào ngày x.

VD: `SELECT add_months(sysdate, 3) FROM DUAL;`

+ MONTHS_BETWEEN(<ngày 1>,<ngày 2>): Số tháng giữa 2 ngày.

+ SYSDATE: Trả về ngày tháng hiện tại.