

CO' BẢN VỀ SQL

➤ *Giảng viên: Nguyễn Thị Thu Trang*

Nội dung chính

- Giới thiệu về SQL
- Truy vấn dữ liệu: Select
- Định nghĩa dữ liệu: DDL (Create, Alter, Drop)
- Thao tác dữ liệu: DML (Insert, Update, Delete)
- Tối ưu hóa câu lệnh SQL

SQL

- ❖ SQL: Structured Query Language
- ❖ Các loại:
 - DQL (Data Query Language)
 - DML (Data Manipulation Language)
 - DDL (Data Definition Language)
 - DCL (Data Control Language)

Kiểu dữ liệu

Character	Numeric	Datetime	LOB	ROWID	Binary
CHAR (<i>n</i>)	NUMBER(<i>m,n</i>)	DATE	CLOB	ROWID	RAW(<i>size</i>)
NCHAR(<i>n</i>)	FLOAT	TIMESTAMP WITH TIMEZONE	NCLOB	UROWID	LONG RAW
VARCHAR2(<i>n</i>)	BINARY_FLOAT	TIMESTAMP WITH LOCAL TIMEZONE	BLOB		
NVARCHAR2(<i>n</i>)	BINARY_DOUBLE	INTERVAL YEAR[(<i>n</i>)] TO MONTH	BFILE		
		INTERVAL DAY[(<i>m</i>)] TO SECOND[(<i>n</i>)]			

ABC

42



AAAM4yAABAAAO9KAAA

ORACLE

Kiểu dữ liệu

❖ Kiểu ký tự:

- Char(<size>): 1byte → 2000 byte
- Varchar2(<size>): 4000 byte
- Nvarchar2(<size>): 4000 byte

❖ Kiểu số:

- Number(p,s)
- **p** is the precision (1 → 38) and **s** (-84 → 127) is the scale

❖ Kiểu ngày tháng

- Date: Jan 1, 4712 BC → Dec 31, 9999 AD

Kiểu dữ liệu..

❖ Kiểu số: **Number (p , s)**

- NUMBER : có thể lưu trữ bất cứ giá trị nào (đến 38 chữ số)
- NUMBER(9): lưu trữ tối đa số có 9 chữ số, không có chữ số thập phân
- NUMBER(9, 2): Gồm 9 chữ số, trong đó 7 chữ số trước dấu phẩy và 2 chữ số sau dấu phẩy (-9999999.99 → 9999999.99)
- NUMBER (*, 2): Không giới hạn số các chữ số nhưng phần thập phân có hai chữ số.
- NUMBER (9, -2): Có 9 chữ số phần nguyên, làm tròn ở hai số cuối (987654321 thành 987654300)
- NUMBER(3, 8): Gồm 8 chữ số thập phân, trong đó 3 số cuối khác 0 (-0.00000999 → 0.00000999)

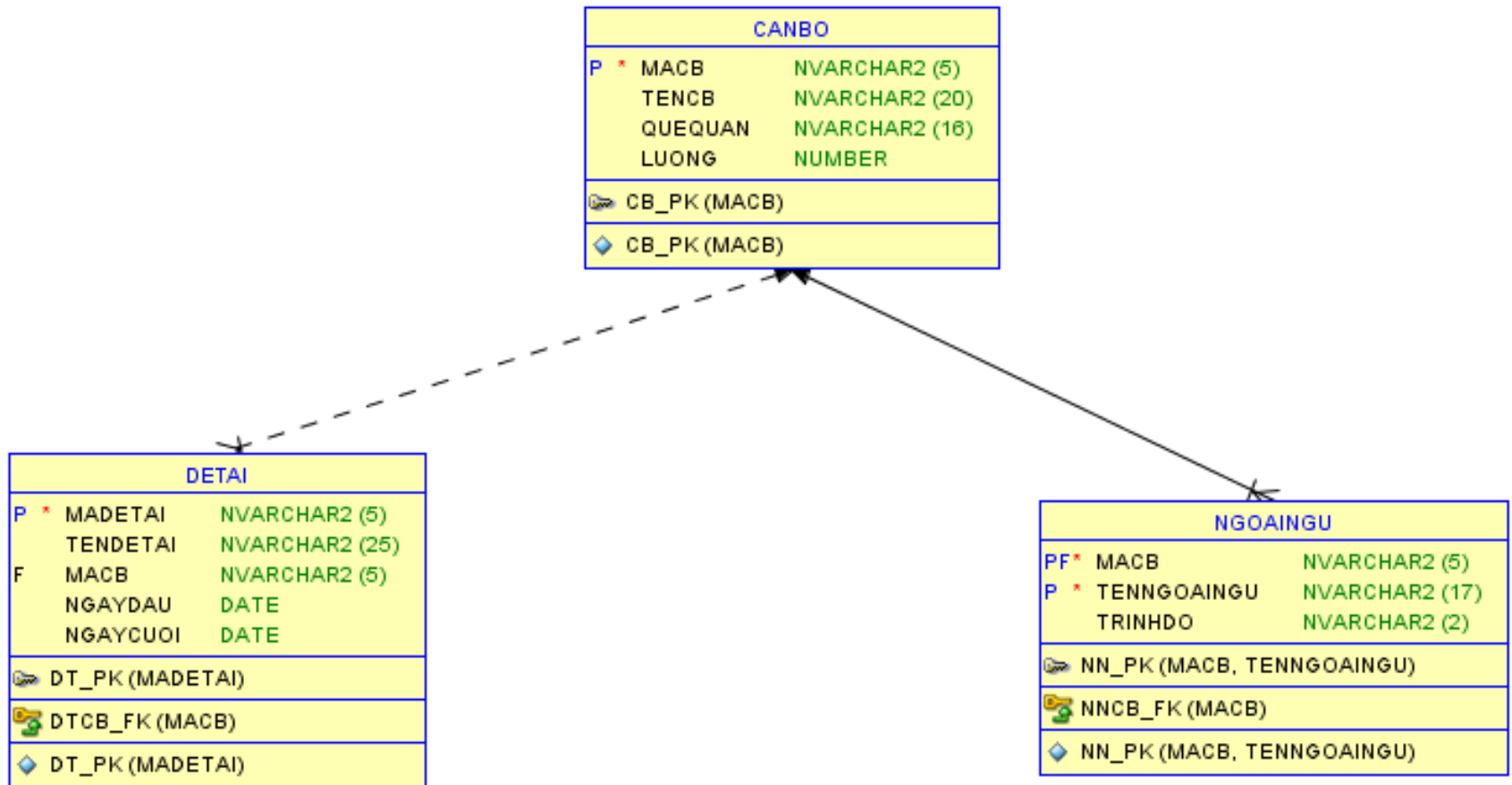
Lấy dữ liệu

```
SELECT [DISTINCT] danh_sách_cột  
FROM {table_name | view_name}  
[WHERE điều_kiện]  
[GROUP BY danh_sách_cột_1]  
[HAVING điều_kiện_lọc]  
[ORDER BY danh_sách_cột_2 [ASC | DESC]]
```

Phép toán

Loại phép toán	Phép toán
Toán học	+, -, *, /
So sánh	=, !=, <>, ^=, <, >, <=, >=
→ SOME/ANY, ALL	
Logic	NOT, AND, OR
Các phép toán chuỗi	, LIKE, NOT LIKE
Các phép toán khác	IN, NOT IN, BETWEEN, EXISTS, IS NULL, IS NOT NULL

VÍ DỤ



Câu lệnh SELECT đơn giản

```
SELECT [DISTINCT] {*, column_list [alias],...}  
FROM table-name
```

- ❖ Chọn tất cả các cột?
- ❖ Một cột cụ thể?
- ❖ Thay đổi tên hiển thị

Câu lệnh SELECT đơn giản

❖ `select tencb from canbo;`

❖ `select * from ngoaingu;`

Select (Limiting Results)

- ❖ Hiển thị N bản ghi đầu tiên của bảng

```
select      *  
from        sometable  
where       rownum <= N  
order by    name_col
```

select * from canbo where rownum<5

	MACB	TENCB	QUEQUAN	LUONG
1	cb1	Vu Thi Binh	Ha Noi	2000000
2	cb2	Luu Ngoc Duc	Ha Nam	3000000
3	cb3	Tran Thu Ha	Nam Dinh	2800000
4	cb4	Dang Thi Hang	Cao Bang	4000000

SQL Alias (Bí danh cột)

- ❖ `SELECT column_name AS alias_name
FROM table_name;`

Ví dụ

- ❖ `select macb as MaCanBo, tengoangu as "Ngoai Ngu"`
- ❖ `from ngoaingu;`

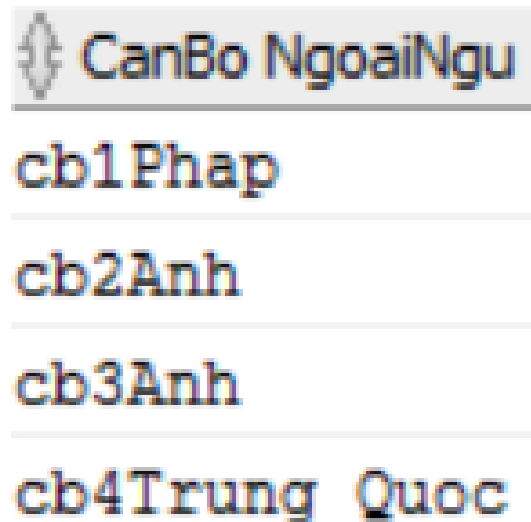
MACANBO	Ngoai Ngu
cb1	Phap
cb2	Anh
cb3	Anh
cb4	Trung Quoc

Toán tử nối

- ❖ Nối các cột hoặc chuỗi ký tự vào các cột
- ❖ Kí hiệu: ||
- ❖ Tạo ra các cột kết quả là một biểu thức ký tự

Ví dụ

```
Select mach || tengoaingu as "CanBo NgoaiNgu"  
From ngoaingu;
```



CanBo NgoaiNgu
cb1Phap
cb2Anh
cb3Anh
cb4Trung Quoc

SQL SELECT DISTINCT

❖ `SELECT DISTINCT column_name, column_name
FROM table_name;`

❖ Ví dụ 1: `Select quequan from canbo;`

❖ Ví dụ 2:

`Select distinct quequan from canbo;`

	QUEQUAN
1	Bac Ninh
2	Ha Noi
3	Ha Nam
4	Thanh Hoa
5	Nam Dinh
6	Cao Bang

	QUEQUAN
1	Ha Noi
2	Ha Nam
3	Nam Dinh
4	Cao Bang
5	Thanh Hoa
6	Bac Ninh
7	Ha Noi
8	Ha Noi

Biểu thức số học

❖ Phép toán số học: * / + -

Ví dụ: Thưởng cho mỗi cán bộ 1 tháng lương cộng thêm 500000đ

```
Select tencb as "Ho ten", luong as Luong, luong + 500000  
as Thuong  
from canbo;
```

Ho ten	LUONG	THUONG
Vu Thi Binh	2000000	2500000
Luu Ngoc Duc	3000000	3500000
Tran Thu Ha	2800000	3300000
Dang Thi Hang	4000000	4500000
Ha Thi Hien	5000000	5500000
Dinh Thuy Hien	3500000	4000000
Vu Thi Hoa	7000000	7500000
Hoang Thi Hue	8000000	8500000

SQL WHERE Syntax

```
SELECT column_name, column_name  
FROM   table_name  
WHERE  column_name comparison-operator value;
```

Comparison Operators	Description
=	equal to
<>, !=	is not equal to
<	less than
>	greater than
>=	greater than or equal to
<=	less than or equal to

SQL WHERE Syntax ...

❖ Ví dụ: Hiển thị tên và lương của các cán bộ có lương hơn 3 triệu:

```
Select tencb as "Ho Ten", luong as "Luong thang"  
From      canbo  
Where     luong > 3000000;
```

Ho Ten	Luong thang
Dang Thi Hang	4000000
Ha Thi Hien	5000000
Dinh Thuy Hien	3500000
Vu Thi Hoa	7000000
Hoang Thi Hue	8000000

ALL, ANY/SOME

- ❖ The ALL comparison condition is used to compare a value to a list or subquery.
- ❖ It must be preceded by =, !=, >, <, <=, >= and followed by a list or subquery.

ALL, ANY/SOME

```
Select      tencb, luong
from        canbo
where luong <= ALL (3000000, 4000000,
8000000);
```

	TENCB	LUONG
1	Vu Thi Binh	2000000
2	Luu Ngoc Duc	3000000
3	Tran Thu Ha	2800000

	LUONG
1	2000000
2	3000000
3	2800000
4	4000000
5	5000000
6	3500000
7	7000000
8	8000000

ALL, ANY/SOME

- ❖ The ANY comparison condition is used to compare a value to a list or subquery.
- ❖ It must be preceded by =, !=, >, <, <=, >= and followed by a list or subquery.
- ❖ The SOME and ANY comparison conditions do exactly the same thing and are completely interchangeable.

ALL, ANY/SOME

```
Select tenb as "Ho Ten", luong as "Luong thang"  
from canbo  
where luong > ANY (4000000, 7000000);
```

	⚡ Ho Ten	⚡ Luong thang
1	Ha Thi Hien	5000000
2	Vu Thi Hoa	7000000
3	Hoang Thi Hue	8000000

Một số phép toán khác

❖ Các phép toán so sánh khác

Comparison Operators	Description
LIKE	column value is similar to specified character(s).
IN	column value is equal to any one of a specified set of values.
BETWEEN...AND	column value is between two values, including the end values specified in the range.
IS NULL	column value does not exist.

Một số phép toán khác ...

❖ IN (NOT IN)

- Dùng để kiểm tra các giá trị thuộc (không thuộc) một danh sách cho trước

```
SELECT      column_name(s)
FROM        table_name
WHERE       column_name IN/NOT IN (value1,value2,...);
```


Một số phép toán khác ...

- ❖ Ví dụ: hiển thị lương các cán bộ thỏa mãn 1 trong các giá trị sau

```
Select tencb, lương  
from   canbo  
where  lương IN (2000000,3000000, 4000000,5000000, 6000000);
```

	TENCB	LUONG
1	Vu Thi Binh	2000000
2	Luu Ngoc Duc	3000000
3	Dang Thi Hang	4000000
4	Ha Thi Hien	5000000

Một số phép toán khác ...

```
Select    tencb, luong
from      canbo
where     luong NOT IN (2000000, 3000000,
4000000);
```


	TENCB	LUONG
1	Tran Thu Ha	2800000
2	Ha Thi Hien	5000000
3	Dinh Thuy Hien	3500000
4	Vu Thi Hoa	7000000
5	Hoang Thi Hue	8000000

Một số phép toán khác ...

❖ SQL LIKE Syntax

- The LIKE operator is used to search for a specified pattern in a column.

```
SELECT    column_name (s)  
FROM      table_name  
WHERE     column_name LIKE pattern;
```



Wildcard	Description
%	A substitute for zero or more characters
_	A substitute for a single character
[<i>charlist</i>]	Sets and ranges of characters to match
[<i>^charlist</i>] or [<i>!charlist</i>]	Matches only a character NOT specified within the brackets

Một số phép toán khác ...

❖ Ví dụ: hiển thị ra các cán bộ có quê quán bắt đầu bởi chữ H

```
Select tencb, quequan from canbo  
Where quequan LIKE 'H%';
```

	TENCB	QUEQUAN
1	Vu Thi Binh	Ha Noi
2	Luu Ngoc Duc	Ha Nam
3	Vu Thi Hoa	Ha Noi
4	Hoang Thi Hue	Ha Noi

Một số phép toán khác ...

❖ BETWEEN ... AND...

- Dùng để lựa chọn giá trị của trường trong một khoảng giá trị (numbers, text, dates) cho trước.

```
SELECT      column_name(s)
FROM        table_name
WHERE column_name BETWEEN value1 AND value2;
```

❖ Ví dụ 1:

```
select madetai, ngaycuoi
from detai
where ngaycuoi between '1-JUL-13' AND '1-AUG-13';
```

Một số phép toán khác ...

Ví dụ 2:

```
Select MADETAI   from detai  
where MADETAI BETWEEN 'dt2' and 'dt4';
```

	MADETAI
1	dt2
2	dt3
3	dt4

Một số phép toán khác ...

- ❖ IS NULL/ IS NOT NULL
- ❖ Giá trị NULL
 - Nếu một bản ghi thiếu giá trị tại một trường nào đó thì giá trị đó là NULL.
- ❖ Ví dụ: hiển thị các cán bộ không biết ngoại ngữ

```
select machb, tengoaingu  
from ngoaingu  
where tengoaingu IS NULL;
```

Một số phép toán khác ...

❖ EXISTS

```
Select MACB From CANBO
Where EXISTS
      (Select MACB From DETAI Where
        DETAI.MACB=CANBO.MACB)
```

	MACB
1	cb1
2	cb10
3	cb13
4	cb15
5	cb2
6	cb5
7	cb8
8	cb9

Phép toán logic

❖ AND, OR, NOT

Logical Operators	Description
OR	For the row to be selected at least one of the conditions must be true.
AND	For a row to be selected all the specified conditions must be true.
NOT	For a row to be selected the specified condition must be false.

Phép toán logic ...

❖ AND

- Ví dụ 1: hiển thị các cán bộ quê không ở Hà Nội và lương trên 4 triệu

```
select * from canbo  
where quequan NOT LIKE 'Ha Noi' AND luong >4000000;
```

❖ OR

- Ví dụ 2: hiển thị mã cán bộ của các cán bộ hoặc biết tiếng Pháp hoặc có trình độ B

```
select * from ngoaingu  
where tengoaingu LIKE 'Phap' OR trinhdo LIKE 'B';
```

Phép toán logic ...

❖ NOT

- Ví dụ: hiển thị mã cán bộ không biết tiếng Trung Quoc

```
select * from ngoaingu  
where NOT ( tengoaingu = 'Trung Quoc');
```

Mệnh đề ORDER BY

- ❖ Sắp xếp thứ tự hàng với mệnh đề ORDER BY

```
SELECT column_name1, column_name2,...  
FROM      table_name  
ORDER BY column_name1, column_name2,...  
        ASC | DESC ;
```

- ❖ Với ASC: thứ tự tăng (mặc định)

- DESC: thứ tự giảm

- ❖ Hiển thị thông tin của bảng cán bộ với cột lương giảm dần

```
Select * From CANBO ORDER BY Luong Desc;
```

Hàm đơn

Single – Row Functions

- ❖ Mô tả nhiều loại hàm khác nhau có sẵn trong SQL
- ❖ Các hàm này chỉ thao tác trên từng hàng và trả về kết quả theo từng hàng.
- ❖ Nhận nhiều đối số và trả về đơn trị
- ❖ Có thể hiệu chỉnh kiểu dữ liệu
- ❖ Có thể lồng nhau

Hàm đơn ...

❖ Các loại hàm đơn:

- Character
- Number
- Date
- Conversion

Một số hàm phục vụ cho truy vấn

Chuỗi	Số	Thời gian	Chuyển đổi	Rẽ nhánh	Gộp
LENGTH	FLOOR, CEIL, ROUND	ADD_MONTHS	CAST	CASE	MIN, MAX
LOWER, UPPER	MOD	SYSDATE	TO_CHAR	DECODE	COUNT
LPAD, RPAD	SQRT	EXTRACT	TO_DATE		AVG
LTRIM, RTRIM, TRIM		MONTHS_BETWEEN	TO_NUMBER		SUM
SUBSTR			SIGN		

Hàm Character

- ❖ Nhận dữ liệu character là input và có thể trả về giá trị character hoặc number

Tên hàm	Ý nghĩa
LOWER(string_value)	Chuyển chuỗi ký tự string_value sang chữ thường
UPPER (string_value)	Chuyển chuỗi ký tự string_value sang chữ hoa
INITCAP (string_value)	Chuyển ký tự đầu tiên sang chữ hoa
CONCAT(string1,string2)	Nối string1 với string2

- ❖ Ví dụ

```
select UPPER(tencb)
from canbo;
```

```
select initcap('xin chao')
from dual;
```



⚡ INITCAP('XINCHAO')
Xin Chao

Hàm Character ...

<i>Tên hàm</i>	<i>Ý nghĩa</i>
SUBSTR (string_value, m, n)	Returns ' <i>n</i> ' number of characters from ' <i>string_value</i> ' starting from the ' <i>m</i> ' position.
LENGTH (string_value) / LENGTHB(string_value)	Number of characters/bytes in ' <i>string_value</i> ' in returned.
TRIM (trim_text FROM string_value)	All occurrences of ' <i>trim_text</i> ' from the left and right of ' <i>string_value</i> ', ' <i>trim_text</i> ' can also be only one character long .
LTRIM(string_value, trim_text)	All occurrences of ' <i>trim_text</i> ' is removed from the left of ' <i>string_value</i> '.
RTRIM (string_value, trim_text)	All occurrences of ' <i>trim_text</i> ' is removed from the right of ' <i>string_value</i> '
LPAD (string_value, n, pad_value)	Returns ' <i>string_value</i> ' left-padded with ' <i>pad_value</i> '. The length of the whole string will be of ' <i>n</i> ' characters.
RPAD (string_value, n, pad_value)	Returns ' <i>string_value</i> ' right-padded with ' <i>pad_value</i> '. The length of the whole string will be of ' <i>n</i> ' characters.

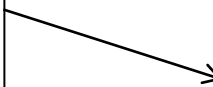
Tên hàm	Ví dụ	Giá trị trả về
LOWER(string_value)	LOWER('Good Morning')	good morning
UPPER(string_value)	UPPER('Good Morning')	GOOD MORNING
INITCAP(string_value)	INITCAP('GOOD MORNING')	Good Morning
LTRIM(string_value, trim_text)	LTRIM ('Good Morning', 'Good')	Morning
RTRIM (string_value, trim_text)	RTRIM ('Good Morning', 'Morning')	Good
TRIM (trim_text FROM string_value)	TRIM ('o' FROM 'Good Morning')	Gd Mrning
SUBSTR (string_value, m, n)	SUBSTR ('Good Morning', 6, 7)	Morning
LENGTH (string_value)	LENGTH ('Good Morning')	12
LPAD (string_value, n, pad_value)	LPAD ('Good', 6, '*')	**Good
RPAD (string_value, n, pad_value)	RPAD ('Good', 6, '*')	Good**

Hàm Character ...

TRIM([[LEADING | TRAILING | BOTH] trim_character FROM] string1)

TRIM(' tech ')	would return 'tech'
TRIM(' ' from ' tech ')	would return 'tech'
TRIM(leading '0' from '000123')	would return '123'
TRIM(trailing '1' from 'Tech1')	would return 'Tech'
TRIM(both '1' from '123Tech111')	would return '23Tech'

```
SELECT TRIM (' SINH VIEN ')  
FROM DUAL;
```



TRIM('SINHVIEN')
SINH VIEN

Hàm Character ...

❖ Hàm LTRIM

<code>LTRIM(' tech');</code>	would return 'tech'
<code>LTRIM(' tech', ' ');</code>	would return 'tech'
<code>LTRIM('000123', '0');</code>	would return '123'
<code>LTRIM('123123Tech', '123');</code>	would return 'Tech'
<code>LTRIM('123123Tech123', '123');</code>	would return 'Tech123'
<code>LTRIM('xyxzyyyTech', 'xyz');</code>	would return 'Tech'
<code>LTRIM('6372Tech', '0123456789');</code>	would return 'Tech'

Ví dụ các hàm ký tự ...

❖ Hàm RTRIM

<code>RTRIM('tech ');</code>	would return 'tech'
<code>RTRIM('tech ', ' ');</code>	would return 'tech'
<code>RTRIM('123000', '0');</code>	would return '123'
<code>RTRIM('Tech123123', '123');</code>	would return 'Tech'
<code>RTRIM('123Tech123', '123');</code>	would return '123Tech'
<code>RTRIM('Techxyxzyyy', 'xyz');</code>	would return 'Tech'
<code>RTRIM('Tech6372', '0123456789');</code>	would return 'Tech'

Hàm Number

Tên hàm	Mô tả
ABS (x)	Absolute value of the number 'x'
CEIL (x)	Integer value that is Greater than or equal to the number 'x'
FLOOR (x)	Integer value that is Less than or equal to the number 'x'
TRUNC (x, y)	Truncates value of number 'x' up to 'y' decimal places
ROUND (x, y)	Rounded off value of the number 'x' up to the number 'y' decimal places

Hàm Number ...

❖ Ví dụ

Function Name	Examples	Return Value
ABS (x)	ABS (1) ABS (-1)	1 -1
CEIL (x)	CEIL (2.83) CEIL (2.49) CEIL (-1.6)	3 3 -1
FLOOR (x)	FLOOR (2.83) FLOOR (2.49) FLOOR (-1.6)	2 2 -2
ROUND (x, y)	ROUND (125.456, 1) ROUND (125.456, 0) ROUND (124.456, -1)	125.5 125 120
TRUNC (x, y)	TRUNC (140.234, 2) TRUNC (-54, 1) TRUNC (5.7) TRUNC (142, -1)	140.23 -54 5 140

Hàm Number ...

- ❖ Hàm SIGN: trả về một con số quy định dấu của N

```
SELECT SIGN(N)  
FROM table_name;
```

- N is the number whose SIGN is to be determined.
- If $N > 0$ then 1 is returned
- If $N = 0$ then 0 is returned
- If $N < 0$ then -1 is returned

Hàm DATE

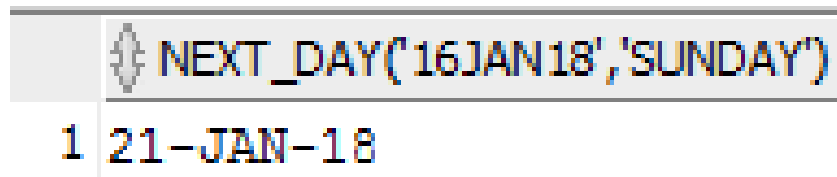
❖ Tất cả hàm Date trả về giá trị kiểu Date ngoại trừ hàm MONTHS_BETWEEN trả về numeric

Tên hàm	Mô tả
MONTHS_BETWEEN (x1, x2)	Returns the number of months between dates x1 and x2.
ADD_MONTHS (date, n)	Returns a date with a specified number of months added.
ROUND (x, date_format)	Returns the date 'x' rounded off to the nearest century, year, month, date, hour, minute, or second as specified by the 'date_format'.
TRUNC (x, date_format)	Returns the date 'x' lesser than or equal to the nearest century, year, month, date, hour, minute, or second as specified by the 'date_format'.

Hàm DATE ...

Tên hàm	Mô tả
NEXT_DAY (x, week_day)	Returns the next date of the 'week_day' on or after the date 'x' occurs.
LAST_DAY (x)	Returns the last day of the month based on a <i>date</i> value x.
SYSDATE	Returns the systems current date (host)
CURRENT_DATE	Returns the current date (client)
EXTRACT(c FROM date)	Returns and extracts a value c from a date or interval value.

VD: `Select next_day('16 jan 18','sunday') From dual;`



The screenshot shows a SQL query in a text editor: `NEXT_DAY('16JAN18','SUNDAY')`. Below the query, the result is displayed as `1 21-JAN-18`. The text is color-coded, with '1' in blue, '21-JAN-18' in red, and the rest in black.

Hàm DATE ...

Select Sysdate From Dual;	
Script Output x Query Result x	
SQL All Rows Fetched: 1 in 0 seconds	
	SYSDATE
1	16-JAN-18


Select trunc(Sysdate, 'Mon') From Dual;	
Script Output x Query Result x	
SQL All Rows Fetched: 1 in 0 seconds	
	TRUNC(SYSDATE,'MON')
1	01-JAN-18

Select round(Sysdate, 'Mon') From Dual;	
Script Output x Query Result x	
SQL All Rows Fetched: 1 in 0 seconds	
	ROUND(SYSDATE,'MON')
1	01-FEB-18

Hàm DATE ...


❖ Ví dụ

```
select add_months('1-JAN-2013', 5)  
from dual;
```



ADD_MONTHS('1-JAN-2013',5)
01-JUN-13

```
select months_between('1-JAN-2013', '1-AUG-2013') "So Thang"  
from dual;
```

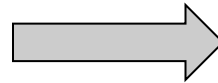


So Thang
-7

Hàm DATE ...

❖ Ví dụ

```
select sysdate  
from dual;
```



SYSDATE
02-SEP-13

```
select current_date  
from dual;
```



CURRENT_DATE
02-SEP-13

Hàm DATE ...

```
select * from detai
where extract(year from to_date(ngaycuoi, 'dd/mm/yy'))>2012;
```

Query Result x

SQL | All Rows Fetched: 8 in 0.006 seconds

	MADETAI	TENDETAI	MACB	NGAYDAU	NGAYCUOI
1	dt2	Quan Ly Tien Gui	cb2	01-JAN-13	16-JUL-13
2	dt5	Quan Ly Diem Thi	cb8	18-MAR-12	20-NOV-13
3	dt9	Tim hieu Core Banking	cb9	23-APR-13	23-DEC-13
4	dt10	Tim hieu DSS	cb10	17-JUL-13	23-JAN-14
5	dt3	Quan Ly Ky tuc xa	cb5	08-SEP-13	20-NOV-13
6	dt11	Quan ly rui ro	cb13	17-JUL-15	23-JAN-16
7	dt16	Datamining	cb10	01-APR-15	01-JAN-16
8	dt12	Mo hinh du doan no xau	cb15	20-MAY-13	20-MAY-14

Hàm DATE ...

❖ Ví dụ

```
select extract(year from date '2013-09-02')  
from dual;
```

→ 2013

```
-----  
select extract(month from date '2013-09-02')  
from dual;
```

→ 9

```
-----  
select extract(day from date '2013-09-02')  
from dual;
```

→ 2

```
-----  
select extract(year from to_date('2/9/2013', 'dd/mm/yy'))  
from dual;
```

→ 2013

```
-----  
select extract(month from to_date('2-9-2013', 'dd-mm-yy'))  
from dual;
```

→ 9

Hàm Conversion

- ❖ SQL cung cấp hàm chuyển đổi giữa các kiểu dữ liệu

<i>Tên hàm</i>	<i>Mô tả</i>
TO_CHAR (x [,y])	Converts Numeric and Date values to a character string value.
TO_DATE (x [, date_format])	Converts a valid Numeric and Character values to a Date value. Date is formatted to the format specified by 'date_format'.
TO_NUMBER(char)	Converts a string to a number
NVL(expr1, expr2)	Converts a NULL to a valid
CAST(x AS type)	Converts x to a compatible database type specified in type

Hàm Conversion ...

❖ Ví dụ 1: hiển thị thông tin về đề tài có ngày hết hạn là 20/11/2013

```
select * from detai  
where ngaycuoi = to_date('20/11/2013','dd/mm/yyyy');
```

	MADETAI	TENDETAI	MACB	NGAYDAU	NGAYCUOI
1	dt5	Quan Ly Diem Thi	cb8	18-MAR-12	20-NOV-13
2	dt3	Quan Ly Ky tuc xa	cb5	08-SEP-13	20-NOV-13

Hàm Conversion ...

- ❖ Ví dụ 2: hiển thị thông tin về đề tài với ngày nhận đề tài (16/1/2012) có dạng Sixteen of January 2012

```
select madetai, tendetai,
       initcap(to_char(ngaydau, 'ddspth "of" month yyyy')) "Ngày Nhan De Tai"
from detai;
```

	MADETAI	TENDETAI	ngày nhan de tai
1	dt1	Quan Ly sinh vien	Sixteenth Of January 2012
2	dt2	Quan Ly Tien Gui	First Of January 2013
3	dt4	Quan Ly Ban Hang	Eighteenth Of March 2011
4	dt5	Quan Ly Diem Thi	Eighteenth Of March 2012
5	dt9	Tim hieu Core Banking	Twenty-Third Of April 2013
6	dt10	Tim hieu DSS	Seventeenth Of July 2013
7	dt3	Quan Ly Ky tuc xa	Eighth Of September 2013
8	dt11	Quan ly rui ro	Seventeenth Of July 2015
9	dt16	Datamining	First Of April 2015
10	dt12	Mo hình du doan no xau	Twentieth Of May 2013
11	dt6	ERP cho doanh nghiep	(null)

Hàm Conversion ...

❖ Hàm CAST(<biểu thức> AS <kiểu dữ liệu>)

TO	FROM					
	char, varchar2	number	datetime / interval	raw	rowid, urowid	nchar, nvarchar2
char, varchar2	X	X	X	X	X	
number	X	X				
datetime / interval	X		X			
raw	X			X		
rowid, urowid	X				X	
nchar, nvarchar2		X	X	X	X	X

Hàm Conversion ...

```
select Madetai, tendetai, ngaycuoi-ngaydau "thoi gian"  
from detai;
```

Query Result x



MADETAI

- 1 dt1
- 2 dt2
- 3 dt4
- 4 dt5
- 5 dt9
- 6 dt10
- 7 dt3
- 8 dt11
- 9 dt16
- 10 dt12
- 11 dt6

```
select Madetai, tendetai, cast(ngaycuoi-ngaydau as VARCHAR2(30)) "thoi gian"  
from detai;
```

Query Result x



SQL | All Rows Fetched: 11 in 0 seconds

	MADETAI	TENDETAI	thoi gian
1	dt1	Quan Ly sinh vien	182
2	dt2	Quan Ly Tien Gui	196
3	dt4	Quan Ly Ban Hang	154
4	dt5	Quan Ly Diem Thi	612
5	dt9	Tim hieu Core Banking	244
6	dt10	Tim hieu DSS	190
7	dt3	Quan Ly Ky tuc xa	73
8	dt11	Quan ly rui ro	190
9	dt16	Datamining	275
10	dt12	Mo hình du doan no xau	365
11	dt6	ERP cho doanh nghiep	(null)

SQL GROUP Functions

❖ Hàm Group

- Cho phép thao tác trên một nhóm các bản ghi và cho kết quả ứng với từng nhóm đó
- **MAX, MIN, AVG, SUM, COUNT, DISTINCT ...**

```
SELECT [column names], group_function (column_name), ... ..  
FROM table  
[WHERE condition]  
[GROUP BY column names]  
[ORDER BY column names]
```

SQL GROUP Functions

❖ Một số chú ý

- Các hàm Group bỏ qua giá trị NULL của cột
- Hàm COUNT(*) đếm cả giá trị NULL, nếu dùng COUNT(expr) đếm giá trị khác NULL
- Hàm MIN, MAX dùng với mọi kiểu dữ liệu
- Dùng hàm NVL giúp hàm group tính giá trị NULL

SQL GROUP Functions ...

❖ Ví dụ minh họa

- Ví dụ 1: hiển thị lương cao nhất của các cán bộ

```
select max(luong) "Lương cao nhất"  
from canbo;
```

- Ví dụ 2: đếm số cán bộ biết tiếng anh

```
select count(*) "Số cán bộ biết tiếng Anh"  
from ngoaingu  
where tengoaingu= 'Anh';
```

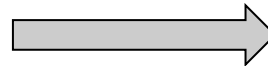
- Ví dụ 3: có bao nhiêu loại tiếng nước ngoài?

```
select count(distinct(tengoaingu)) "Số loại ngoại ngữ"  
from ngoaingu;
```

SQL GROUP Functions ...

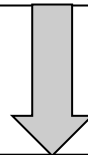
❖ Ví dụ 4: Ngày nhận đề tài sớm nhất?

```
select min(ngaydau)  
from detai;
```



```
MIN(NGAYDAU)  
01-FEB-10
```

```
select min(ngaydau), madetai, tendetai  
from detai;
```



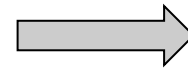
```
ORA-00937: not a single-group group function  
00937. 00000 - "not a single-group group function"
```


SQL GROUP Functions ...

❖ Ví dụ 5: Tính lương trung bình của các cán bộ

- Trường hợp 1:

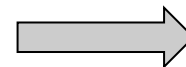
```
Select round(avg(luong), 3)  
"Luong trung binh" from canbo;
```



	Luong trung binh
1	4746153.846

Trường hợp 2:

```
Select  
round(avg(nvl(luong, 0)), 3)  
"Luong trung binh" from canbo;
```



	Luong trung binh
1	3629411.765

SQL GROUP BY Clause

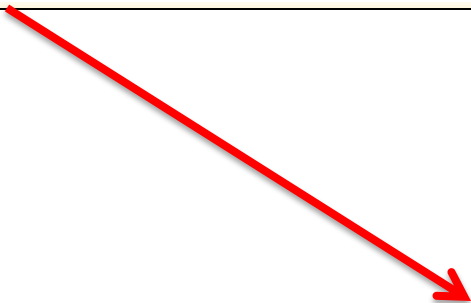
❖ Mệnh đề GROUP BY

- Dùng để chia các bản ghi thành từng nhóm. Sau đó dùng hàm Group để tính thông tin tổng hợp của từng nhóm
- Không được dùng bí danh cột trong mệnh đề Group by
- Cột GROUP BY không nhất thiết phải có trong SELECT
- Tất cả các trường trong SELECT mà không có trong HÀM GROUP thì **PHẢI** có trong GROUP BY
- Mệnh đề WHERE có thể loại bỏ trước các bản ghi trước khi chia chúng thành nhóm

SQL GROUP BY Clause ...

- ❖ Ví dụ: hiển thị tên ngoại ngữ và số người có trình độ tương ứng

```
select tengoaingu, count(*) "So can bo"  
from ngoaingu  
group by tengoaingu;
```



TENGOAINGU	So can bo
(null)	2
Anh	3
Nga	1
Phap	1
Trung Quoc	1

SQL GROUP BY Clause ...

- ❖ GROUP BY trên nhiều cột: **GROUP BY c1, c2**
 - Đầu tiên các bản ghi được nhóm theo c1
 - Trong từng nhóm theo c1, được phân nhóm theo c2

SQL HAVING Clause

- ❖
SELECT column_name, group_function(column_name)
FROM table_name
WHERE column_name operator value
GROUP BY column_name
HAVING group_function(column_name) operator value;
- ❖ Dùng mệnh đề HAVING để giới hạn các nhóm
- ❖ Ví dụ: hiển thị lương trung bình lớn hơn 3 triệu theo vùng miền

```
select  quequan, avg(luong)
from    canbo
group by quequan
having  avg(luong) > 3000000;
```

Hàm rẽ nhánh

❖ Hàm DECODE

```
DECODE( expression , search , result [, search , result]... [, default] )
```

- *expression* is the value to compare.
- *search* is the value that is compared against *expression*.
- *result* is the value returned, if *expression* is equal to *search*.
- *default* is optional. If no matches are found, the **DECODE function** will return *default*. If *default* is omitted, then the **DECODE function** will return null (if no matches are found).

Hàm rẽ nhánh ...

❖ Hàm DECODE

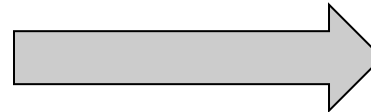
- Ví dụ

```
select decode( 2*3-1, 5, 20,9)  
from dual;
```



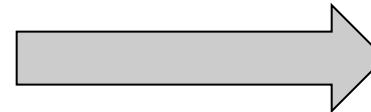
20

```
select decode( 2*3-1, 6, 20,9)  
from dual;
```



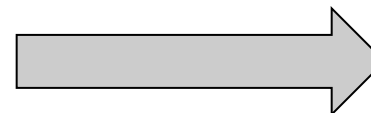
9

```
select decode( 2*3-1, 5, 20)  
from dual;
```



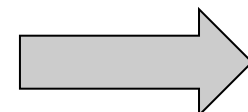
20

```
select decode( 2*3-1, 6, 20)  
from dual;
```



null

```
select decode( 2*3-1, 6, 20, 7, 40, 5, 50, 0)  
from dual;
```

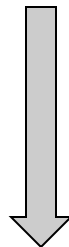


50

Hàm rẽ nhánh

❖ Ví dụ: so sánh giá trị 20 và 10

```
select decode(sign(20-10),1, '20 is greater than 10',  
               0, '20 is equal 10',  
               -1, '20 is less than 10') "Compare..."  
from dual;
```



```
⌵ Compare...  
20 is greater than 10
```


Hàm rẽ nhánh ...

❖ Hàm CASE

```
CASE [ expression ]  
  
    WHEN condition_1 THEN result_1  
    WHEN condition_2 THEN result_2  
    ...  
    WHEN condition_n THEN result_n  
  
    ELSE result  
  
END
```

❖ *Chú ý: Nếu không có ELSE thì hàm CASE sẽ trả về giá trị NULL*

Hàm rẽ nhánh ...

❖ Ví dụ

```
select Macb, tencb, case
    when luong<=3000000 then 'Luong thap'
    when luong<=5000000 then 'Luong TB'
    else 'Luong cao'
end as "nhan xet"
From canbo;
```

Query Result x | Script Output x | Query Result 1 x | Query Result 2 x | Qu

SQL | All Rows Fetched: 16 in 0.003 seconds

	MACB	TENCB	nhan xet
1	cb1	Vu Thi Binh	Luong thap
2	cb2	Luu Ngoc Duc	Luong thap
3	cb3	Tran Thu Ha	Luong thap
4	cb4	Dang Thi Hang	Luong TB
5	cb5	Ha Thi Hien	Luong TB
6	cb6	Dinh Thuy Hien	Luong TB
7	cb7	Vu Thi Hoa	Luong cao
8	cb8	Hoang Thi Hue	Luong cao
9	cb9	Le Viet Hung	Luong TB

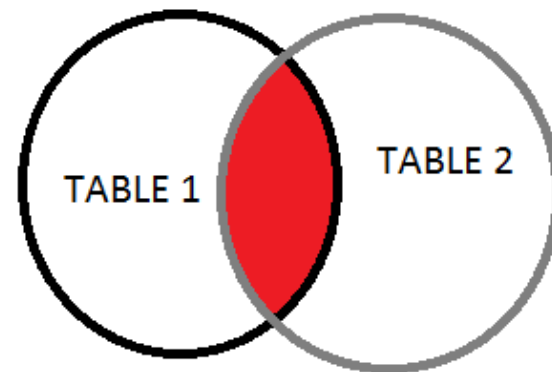
Lấy dữ liệu từ nhiều bảng

- ❖ Sử dụng phép nối
 - INNER JOIN
 - NATURAL JOIN
 - OUTER JOIN
 - LEFT OUTER JOIN
 - RIGHT OUTER JOIN
 - FULL OUTER JOIN

Lấy dữ liệu từ nhiều bảng ...

❖ INNER JOIN

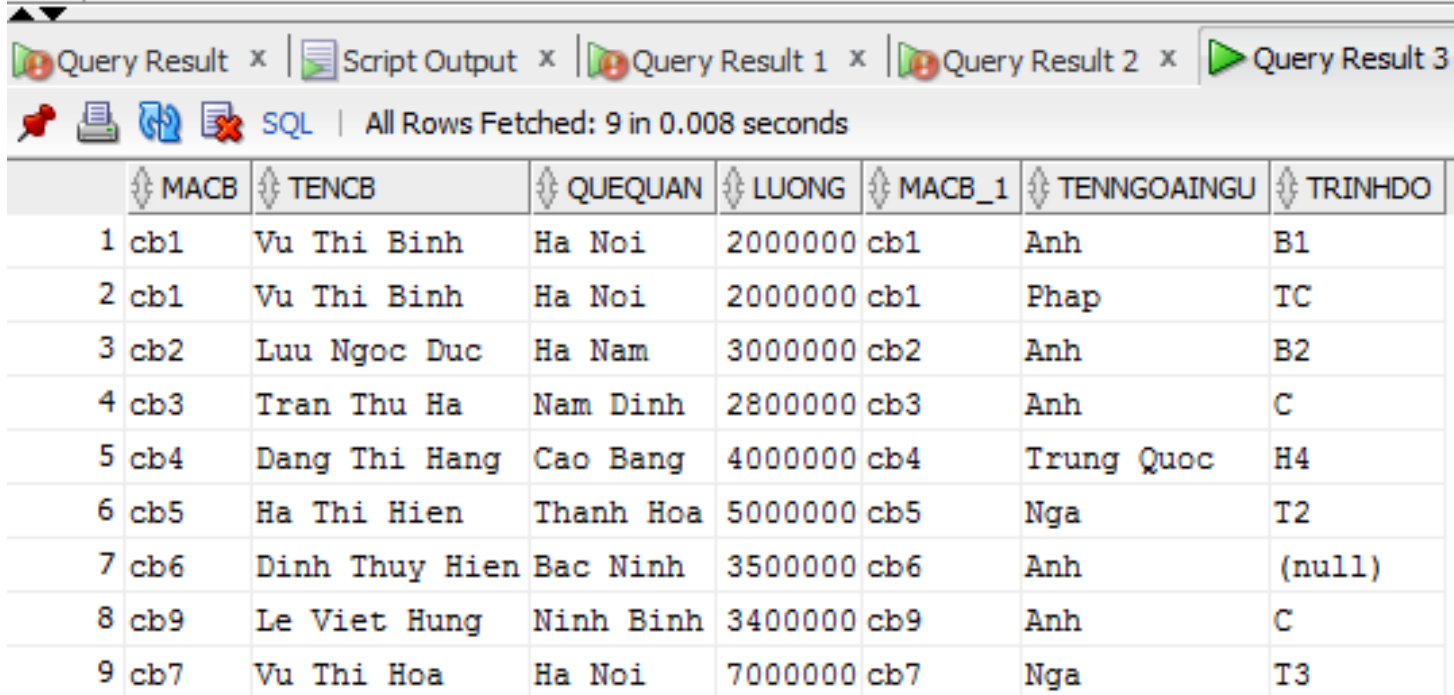
```
SELECT columns  
FROM table1  
INNER JOIN table2  
ON table1.column = table2.column;
```



Lấy dữ liệu từ nhiều bảng ...

❖ Ví dụ:

```
select *  
From canbo inner join ngoaingu on canbo.macb=ngoaingu.macb;
```

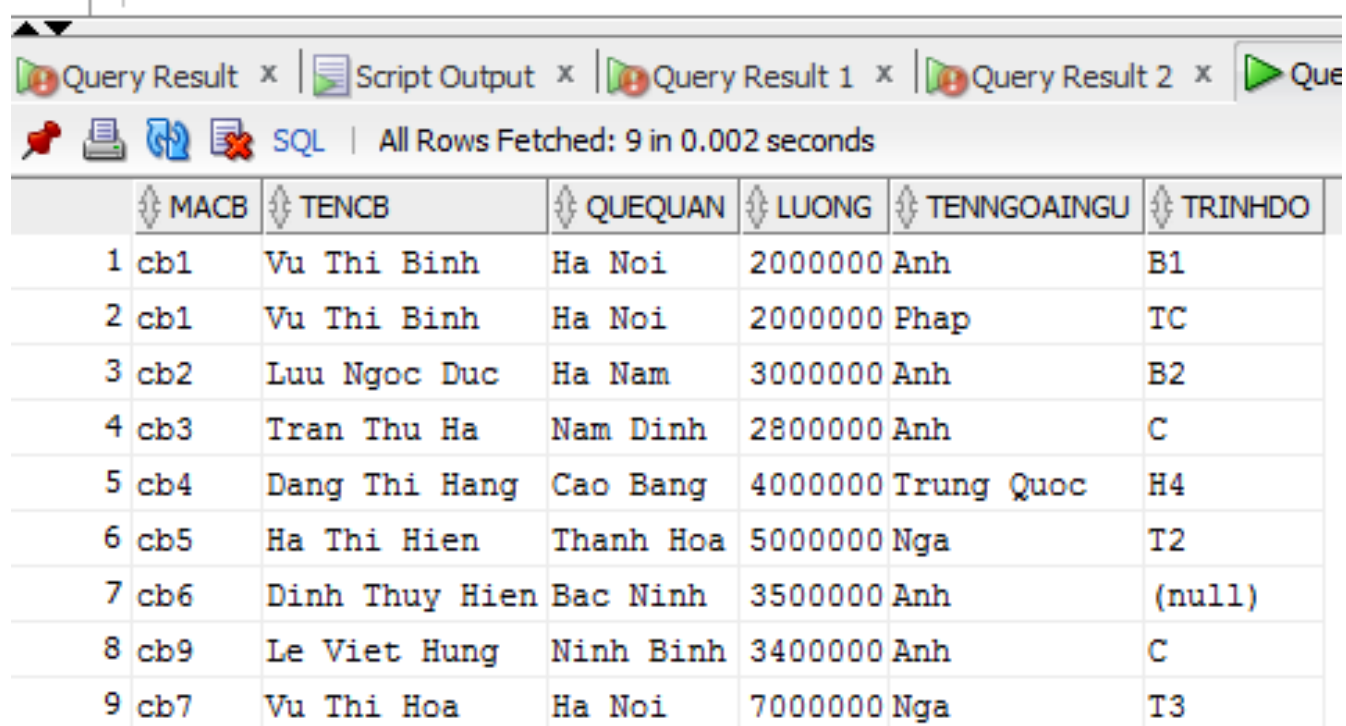


	MACB	TENCB	QUEQUAN	LUONG	MACB_1	TENNGOAINGU	TRINHDO
1	cb1	Vu Thi Binh	Ha Noi	2000000	cb1	Anh	B1
2	cb1	Vu Thi Binh	Ha Noi	2000000	cb1	Phap	TC
3	cb2	Luu Ngoc Duc	Ha Nam	3000000	cb2	Anh	B2
4	cb3	Tran Thu Ha	Nam Dinh	2800000	cb3	Anh	C
5	cb4	Dang Thi Hang	Cao Bang	4000000	cb4	Trung Quoc	H4
6	cb5	Ha Thi Hien	Thanh Hoa	5000000	cb5	Nga	T2
7	cb6	Dinh Thuy Hien	Bac Ninh	3500000	cb6	Anh	(null)
8	cb9	Le Viet Hung	Ninh Binh	3400000	cb9	Anh	C
9	cb7	Vu Thi Hoa	Ha Noi	7000000	cb7	Nga	T3

Lấy dữ liệu từ nhiều bảng ...

- ❖ NATURAL JOIN: kết nối dựa vào thuộc tính cùng tên & cùng kiểu dữ liệu

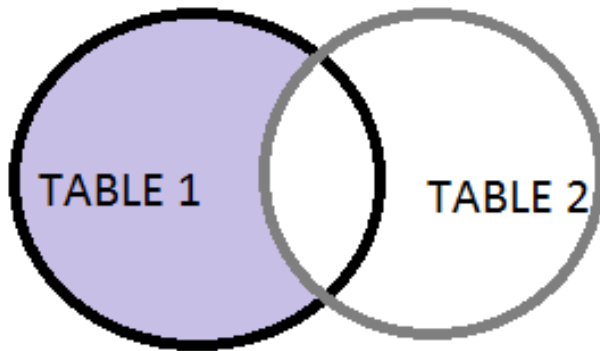
```
select *  
From canbo NATURAL JOIN ngoaingiu;
```



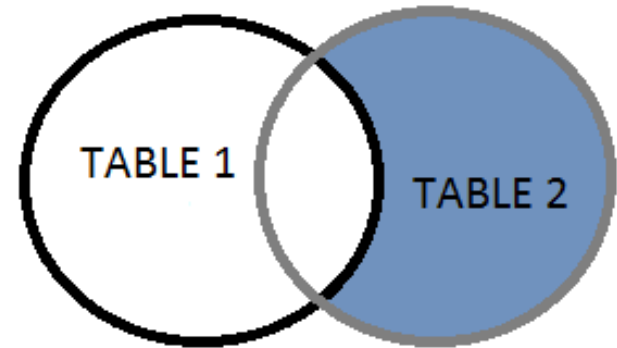
	MACB	TENCB	QUEQUAN	LUONG	TENNGOAINGU	TRINHDO
1	cb1	Vu Thi Binh	Ha Noi	2000000	Anh	B1
2	cb1	Vu Thi Binh	Ha Noi	2000000	Phap	TC
3	cb2	Luu Ngoc Duc	Ha Nam	3000000	Anh	B2
4	cb3	Tran Thu Ha	Nam Dinh	2800000	Anh	C
5	cb4	Dang Thi Hang	Cao Bang	4000000	Trung Quoc	H4
6	cb5	Ha Thi Hien	Thanh Hoa	5000000	Nga	T2
7	cb6	Dinh Thuy Hien	Bac Ninh	3500000	Anh	(null)
8	cb9	Le Viet Hung	Ninh Binh	3400000	Anh	C
9	cb7	Vu Thi Hoa	Ha Noi	7000000	Nga	T3

Lấy dữ liệu từ nhiều bảng ...

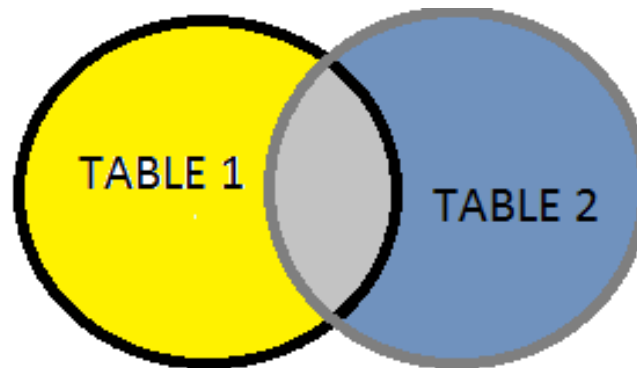
❖ OUTER JOIN: LEFT JOIN



❖ OUTER JOIN: RIGHT JOIN



❖ OUTER JOIN: FULL JOIN



Lấy dữ liệu từ nhiều bảng ...

❖ Ví dụ

```
select *  
From canbo Left outer join ngoaingu on canbo.macb=ngoaingu.macb;
```

Query Result x | Script Output x | Query Result 1 x | Query Result 2 x | Query Result 3 x

SQL | All Rows Fetched: 17 in 0.003 seconds

	MACB	TENCB	QUEQUAN	LUONG	MACB_1	TENNGOAINGU	TRINHDO
1	cb1	Vu Thi Binh	Ha Noi	2000000	cb1	Anh	B1
2	cb1	Vu Thi Binh	Ha Noi	2000000	cb1	Phap	TC
3	cb10	Tran Viet Hung	Lang Son	(null)	(null)	(null)	(null)
4	cb11	Vu Viet Hung	Hai Duong	(null)	(null)	(null)	(null)
5	cb12	Pham Quang Huy	Hai Duong	(null)	(null)	(null)	(null)
6	cb13	Bui Thi Thuong Huyen	(null)	(null)	(null)	(null)	(null)
7	cb14	Do Thi Huyen	Lao Cai	2000000	(null)	(null)	(null)
8	cb15	Vu Thi Ngoc Huyen	Hai Phong	3000000	(null)	(null)	(null)
9	cb16	Dinh Van Hung	(null)	7000000	(null)	(null)	(null)
10	cb2	Luu Ngoc Duc	Ha Nam	3000000	cb2	Anh	B2
11	cb3	Tran Thu Ha	Nam Dinh	2800000	cb3	Anh	C
12	cb4	Dang Thi Hang	Cao Bang	4000000	cb4	Trung Quoc	H4
13	cb5	Ha Thi Hien	Thanh Hoa	5000000	cb5	Nga	T2
14	cb6	Dinh Thuy Hien	Bac Ninh	3500000	cb6	Anh	(null)
15	cb7	Vu Thi Hoa	Ha Noi	7000000	cb7	Nga	T3
16	cb8	Hoang Thi Hue	Ha Noi	8000000	(null)	(null)	(null)
17	cb9	Le Viet Hung	Ninh Binh	3400000	cb9	Anh	C

Lấy dữ liệu từ nhiều bảng ...

❖ Ví dụ

```
select *  
From canbo right join ngoaingu on canbo.macb=ngoaingu.macb;
```

Query Result x | Script Output x | Query Result 1 x | Query Result 2 x | Query Result 3 x

SQL | All Rows Fetched: 9 in 0.003 seconds

	MACB	TENCB	QUEQUAN	LUONG	MACB_1	TENNGOAINGU	TRINHDO
1	cb1	Vu Thi Binh	Ha Noi	2000000	cb1	Anh	B1
2	cb1	Vu Thi Binh	Ha Noi	2000000	cb1	Phap	TC
3	cb2	Luu Ngoc Duc	Ha Nam	3000000	cb2	Anh	B2
4	cb3	Tran Thu Ha	Nam Dinh	2800000	cb3	Anh	C
5	cb4	Dang Thi Hang	Cao Bang	4000000	cb4	Trung Quoc	H4
6	cb5	Ha Thi Hien	Thanh Hoa	5000000	cb5	Nga	T2
7	cb6	Dinh Thuy Hien	Bac Ninh	3500000	cb6	Anh	(null)
8	cb9	Le Viet Hung	Ninh Binh	3400000	cb9	Anh	C
9	cb7	Vu Thi Hoa	Ha Noi	7000000	cb7	Nga	T3

Lấy dữ liệu từ nhiều bảng ...

❖ Ví dụ

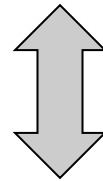
```
select *  
From canbo Full join ngoaingu on canbo.macb=ngoaingu.macb;
```

	MACB	TENCB	QUEQUAN	LUONG	MACB_1	TENNGOAINGU	TRINHDO
1	cb1	Vu Thi Binh	Ha Noi	2000000	cb1	Anh	B1
2	cb1	Vu Thi Binh	Ha Noi	2000000	cb1	Phap	TC
3	cb2	Luu Ngoc Duc	Ha Nam	3000000	cb2	Anh	B2
4	cb3	Tran Thu Ha	Nam Dinh	2800000	cb3	Anh	C
5	cb4	Dang Thi Hang	Cao Bang	4000000	cb4	Trung Quoc	H4
6	cb5	Ha Thi Hien	Thanh Hoa	5000000	cb5	Nga	T2
7	cb6	Dinh Thuy Hien	Bac Ninh	3500000	cb6	Anh	(null)
8	cb9	Le Viet Hung	Ninh Binh	3400000	cb9	Anh	C
9	cb7	Vu Thi Hoa	Ha Noi	7000000	cb7	Nga	T3
10	cb14	Do Thi Huyen	Lao Cai	2000000	(null)	(null)	(null)
11	cb15	Vu Thi Ngoc Huyen	Hai Phong	3000000	(null)	(null)	(null)
12	cb8	Hoang Thi Hue	Ha Noi	8000000	(null)	(null)	(null)
13	cb13	Bui Thi Thuong Huyen	(null)	(null)	(null)	(null)	(null)
14	cb11	Vu Viet Hung	Hai Duong	(null)	(null)	(null)	(null)
15	cb16	Dinh Van Hung	(null)	7000000	(null)	(null)	(null)
16	cb12	Pham Quang Huy	Hai Duong	(null)	(null)	(null)	(null)
17	cb10	Tran Viet Hung	Lang Son	(null)	(null)	(null)	(null)

Lấy dữ liệu từ nhiều bảng ...

❖ JOIN... USING.... (INNER JOIN)

```
select *  
From canbo join ngoaingu  
on canbo.macb=ngoingu.macb;
```



```
Select *  
From canbo join ngoaingu using(macb);
```

Truy vấn con - Subqueries

- ❖ Khái niệm
- ❖ Ví dụ
- ❖ Phân loại

Truy vấn con - Subqueries ...

❖ Ví dụ 1: Ai chưa có ngoại ngữ

	MACB	TENCB
1	cb10	Tran Viet Hung
2	cb11	Vu Viet Hung
3	cb12	Pham Quang Huy
4	cb13	Bui Thi Thuong Huyen
5	cb14	Do Thi Huyen
6	cb15	Vu Thi Ngoc Huyen
7	cb16	Dinh Van Hung
8	cb8	Hoang Thi Hue

Truy vấn con - Subqueries ...

- ❖ A subquery is a query within a query
- ❖ Syntax

```
SELECT    select_list
FROM      table
WHERE     expr operator
          (SELECT    select_list
           FROM      table);
```

Truy vấn con - Subqueries ...

- ❖ Các truy vấn con có thể được lồng vào:
 - A SELECT
 - A INSERT
 - A UPDATE
 - A DELETE
 - A another subquery

Truy vấn con - Subqueries ...

- ❖ Chú ý khi dùng truy vấn con:
 - Câu truy vấn con đặt trong ngoặc đơn
 - Không bổ sung mệnh đề ORDER BY cho câu truy vấn con
 - Dùng toán tử đơn hàng với câu truy vấn đơn hàng
 - Toán tử: =, >, >=, <=, <, <>
 - Dùng toán tử đa hàng với câu truy vấn đa hàng
 - Toán tử: IN, ALL, ANY/SOME

Truy vấn con - Subqueries ...

❖ Chú ý

Operation	Meaning
<ANY	Less than the maximum
<=ANY	Less than or equal to the maximum
>ANY	More than the minimum
=ANY	Equivalent to the IN operator
<ALL	Less than the minimum
>ALL	More than the maximum
<>ALL	Equivalent to the NOT IN operator

Truy vấn con - Subqueries ...

❖ Ví dụ 2: Hiển thị thông tin của cán bộ, với điều kiện quê ở Hà nội và có lương nhỏ nhất trong những người quê ở Hà nội

Truy vấn con - Subqueries ...

❖ Ví dụ 3: Hiển thị mã đề tài, tên đề tài, ngày phải nộp đề tài với các đề tài đã được phân công cho cán bộ

Truy vấn con - Subqueries ...

- ❖ Hiển thị thông tin 3 cán bộ có lương cao nhất (thấp nhất)
 - B1: sắp xếp bảng canbo có lương giảm dần
 - B2: Lấy ra 3 bản ghi đầu tiên

```
select *  
from  
    (  
        select *  
        from canbo  
        where luong IS NOT NULL  
        order by luong desc  
    )  
where rownum <= 3
```

Null Values in a Subquery

- ❖ Nếu có giá trị NULL trong tập trả về của Subquery
- ❖ Câu lệnh sau trả về bao nhiêu bản ghi

```
Select Tencb  
From canbo cb  
Where not exists (select Macb from detai dt  
                  Where cb.macb=dt.macb);
```

SQL TUNING

❖ Tối ưu hóa truy vấn

- Chỉ SELECT những cột và những bản ghi cần thiết
- Sử dụng JOIN thay vì subquery
- Tránh truy vấn trên view
- Gọi tên cột tường minh
- Dùng CASE thay vì sử dụng nhiều truy vấn
- Dùng INDEX
- Dùng WHERE tốt hơn HAVING
- Dùng EXISTS/NOT EXISTS tốt hơn IN/NOT IN
- Hạn chế sử dụng các phép tính toán trong mệnh đề WHERE

DML

Data Manipulation Language

- ❖ Một câu lệnh DML được thực hiện khi:
 - Thêm một hàng vào bảng
 - Thay đổi một hàng đã có trong bảng
 - Xóa hàng đang tồn tại trong bảng

DML – Sửa dữ liệu

- ❖ Sửa dữ liệu: thay đổi hàng đã tồn tại trong bảng

```
UPDATE Tên_bảng  
SET cột1 = giá_trị1, ..., cộtn = giá_trị_n  
[WHERE điều_kiện];
```

- ❖ Tăng lương của mỗi cán bộ thêm 500000 đồng

DML – Xóa dữ liệu

❖ Xóa dữ liệu

- Xóa từng bản ghi

```
DELETE Tên_bảng WHERE [điều_kiện];
```

- Xóa toàn bộ dữ liệu trong bảng

```
TRUNCATE TABLE Tên_bảng;
```

Thêm mới dữ liệu

- ❖ Nhập giá trị cho mọi cột trong bảng

```
INSERT INTO Tên_bảng VALUES (gt1, gt2, ...)
```

- ❖ Nhập giá trị cho một số cột trong bảng

```
INSERT INTO Tên_bảng (cột1, cột2, ...)  
VALUES (gt1, gt2, ...)
```

- ❖ Lấy giá trị từ bảng khác

```
INSERT INTO Tên_bảng (cột1, cột2, ...)  
SELECT gt1, gt2, ...  
FROM...WHERE...
```

DDL

- ❖ Tạo bảng
- ❖ Hiệu chỉnh cấu trúc bảng
- ❖ Xóa bảng
- ❖ Đổi tên bảng

Bảng

- ❖ Tên bảng & tên cột tuân theo quy tắc:
 - 1-30 kí tự
 - Bắt đầu bằng chữ cái
 - Bao gồm **chữ cái, số, _, #, \$** (hạn chế dùng #, \$)
 - Không dùng những từ có sẵn trong Oracle (NUMBER, INDEX...)
 - Tên cột phải duy nhất trong bảng
 - Tên bảng phải duy nhất trong **namespace**

Các thao tác liên quan tới bảng

> Tạo bảng
Sửa bảng
Xóa bảng

❖ Ví dụ:

```
CREATE TABLE change_log  
(log_id      NUMBER  
 ,who        VARCHAR2(64)  
 ,when       TIMESTAMP  
 ,what       VARCHAR2(200)  
 );
```

```
CREATE TABLE change_log  
(log_id      NUMBER  
 ,who        VARCHAR2(64)  
 ,when       TIMESTAMP  
 ,what       VARCHAR2(200)  
 ) TABLESPACE users;
```

SQL> describe change_log

Name	Null?	Type
LOG_ID		NUMBER
WHO		VARCHAR2(64)
WHEN		TIMESTAMP(6)
WHAT		VARCHAR2(200)

Các thao tác liên quan tới bảng

Tạo bảng
> Sửa bảng
Xóa bảng

❖ Sửa giá trị mặc định của cột

```
ALTER TABLE change_log MODIFY  
who          VARCHAR2(64) DEFAULT USER;
```

❖ Đổi tên bảng

```
RENAME change_log TO demo_change_log;
```

```
ALTER TABLE change_log RENAME TO demo_change_log;
```

❖ Xóa cột

```
ALTER TABLE change_log DROP (how,why);
```

```
ALTER TABLE change_log DROP COLUMN how;
```

Các thao tác liên quan tới bảng

Tạo bảng
> Sửa bảng
Xóa bảng

❖ Sửa cột

```
ALTER TABLE change_log MODIFY what VARCHAR2(250);
```

```
ALTER TABLE change_log MODIFY  
(what    VARCHAR2(250)  
,who    VARCHAR2(50)  DEFAULT user  
);
```

❖ Thêm cột

```
ALTER TABLE change_log ADD how VARCHAR2(45);
```

```
ALTER TABLE change_log ADD  
(how    VARCHAR2(45)  
,why    VARCHAR2(60)  
);
```

Các thao tác liên quan tới bảng

Tạo bảng
Sửa bảng
> Xóa bảng

❖ Khi xóa bảng, Oracle tiến hành:

- Xóa dữ liệu
- Xóa cấu trúc dữ liệu lưu trữ bảng
- Xóa các trigger liên quan tới bảng
- Xóa các quyền liên quan tới bảng

```
DROP TABLE hr.employees PURGE;
```

❖ Một số tùy chọn cho câu lệnh xóa bảng

- PURGE: không cho phép flashback
- CASCADE CONSTRAINTS: xóa mọi ràng buộc dữ liệu có liên quan

Constraint

- ❖ Đảm bảo tính toàn vẹn của dữ liệu
- ❖ Có thể tạo constraint lúc tạo bảng hoặc sau khi tạo bảng



Các loại constraint

❖ NOT NULL

❖ UNIQUE

- Không cho phép nhập giá trị giống nhau
- Oracle tự động tạo unique index cho cột có ràng buộc UNIQUE

❖ PRIMARY KEY

- Có thể tạo khóa chính cho 1 hoặc nhiều cột
- Oracle tự động tạo unique index cho cột làm khóa chính

❖ FOREIGN KEY

- Thiết lập mối quan hệ của 1 bảng với bảng khác

❖ CHECK

- Kiểm tra giá trị của một cột thỏa mãn điều kiện cho trước

Khai báo constraint - 1

❖ NOT NULL constraint

```
CREATE TABLE employees
(employee_id    NUMBER    CONSTRAINT nn_emp_id NOT NULL
,hire_date      DATE                                NOT NULL
,first_name     VARCHAR2(42)
,last_name      VARCHAR2(42)
);
```

❖ UNIQUE constraint

```
CREATE TABLE employees
(employee_id    NUMBER    NOT NULL
      CONSTRAINT uniq_payroll_id UNIQUE
,hire_date      DATE    NOT NULL
,first_name     VARCHAR2(42)
,last_name      VARCHAR2(42)
,payroll_id     VARCHAR2(10)
);
```

Khai báo constraint - 2

❖ PRIMARY KEY constraint

```
CREATE TABLE employees
(employee_id    NUMBER          NOT NULL
,hire_date     DATE            NOT NULL
,first_name    VARCHAR2(42)
,last_name     VARCHAR2(42)
,payroll_id    VARCHAR2(10)
,CONSTRAINT employees_pk PRIMARY KEY (employee_id)
        USING INDEX TABLESPACE indx
);
```

Khai báo constraint - 3

❖ FOREIGN KEY constraint

- **Chú ý**: Không được phép tạo khóa ngoại cho những cột có kiểu dữ liệu: CLOB, NCLOB, BLOB, LONG, LONG RAW, TIMESTAMP WITH TIMEZONE

❖ Ví dụ:

```
CREATE TABLE employees  
ALTER TABLE employees  
  ADD CONSTRAINT emp_dept_fk FOREIGN KEY (dept_nbr)  
    REFERENCES departments(dept_nbr) ON DELETE CASCADE;  
  
ALTER TABLE departments ADD CONSTRAINT  
  dept_mgr_fk FOREIGN KEY (manager_id) REFERENCES  
    employees(employee_id) ON DELETE SET NULL;  
  REFERENCES departments(dept_nbr)  
);
```

Khai báo constraint - 4

❖ CHECK constraint

- Kiểm tra giá trị của một cột có thỏa mãn điều kiện cho trước hay không

❖ Ví dụ:

```
ALTER TABLE employees ADD CONSTRAINT  
    validate_hire_date CHECK  
    (hire_date > TO_DATE('15-Apr-1999', 'DD-Mon-YYYY'));
```

Sửa constraint

❖ Xóa

```
ALTER TABLE employees DROP CONSTRAINT validate_hire_date;
```

```
ALTER TABLE employees DROP PRIMARY KEY CASCADE;
```

❖ Đổi tên

```
ALTER TABLE employees  
    RENAME CONSTRAINT validate_hire_date TO hire_date_check;
```

❖ Vô hiệu hóa (disable)

```
ALTER TABLE employees DISABLE CONSTRAINT mgr_emp_fk;  
-- bulk load the table  
ALTER TABLE employees ENABLE CONSTRAINT mgr_emp_fk;
```

