

# COLE.VN

Connecting knowledge

# **Tổng quan CSDL Truy vấn dữ liệu**

Trình bày: Nguyễn Văn Phúc

# NỘI DUNG CHÍNH



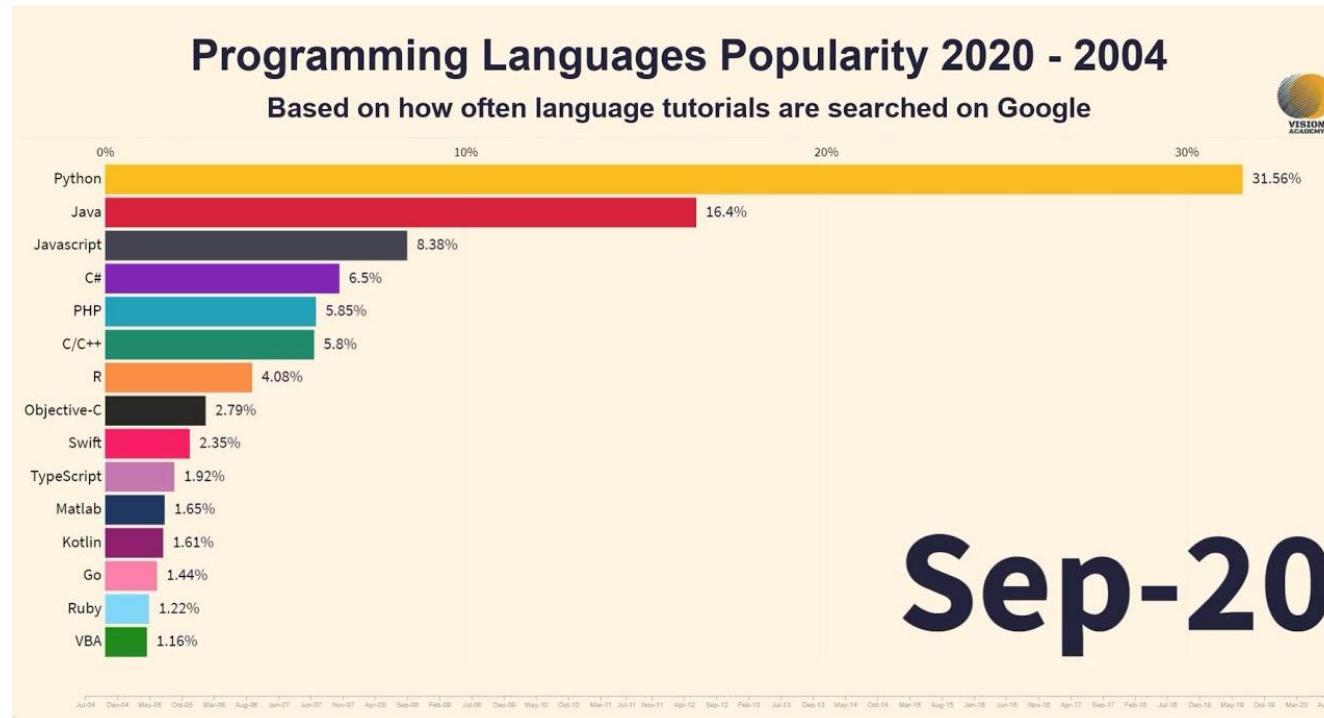
# Cơ sở dữ liệu

- ❖ Dữ liệu (Data)
  - ❖ Cơ sở dữ liệu (Database)
    - ❖ Hệ quản trị cơ sở dữ liệu (DBMS)
      - ❖ Hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) (Oracle Database, SQL Server, MySQL, ...)
        - Bộ xử lý truy vấn
        - Bộ xử lý dữ liệu

```
target_date> select target_time  
2016-12-26 | 02:29:30  
2016-12-26 | 02:32:29  
2016-12-26 | 02:32:29  
2016-12-26 | 02:35:29  
2016-12-26 | 02:35:29  
2016-12-26 | 02:38:29  
2016-12-26 | 02:38:29  
2016-12-26 | 02:41:30  
2016-12-26 | 02:41:30  
2016-12-26 | 02:44:29  
2016-12-26 | 02:44:29
```



# Cơ sở của truy vấn - SQL

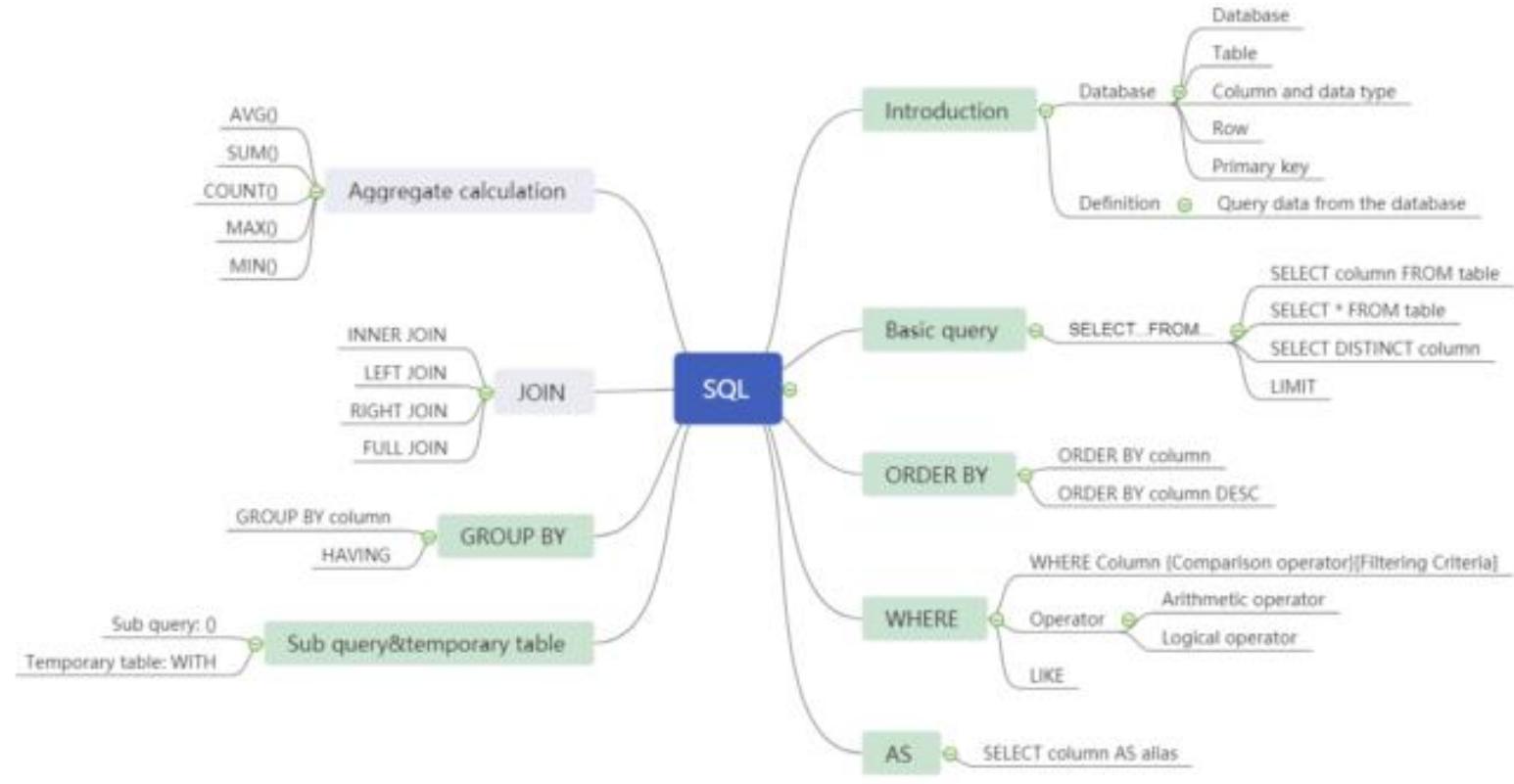
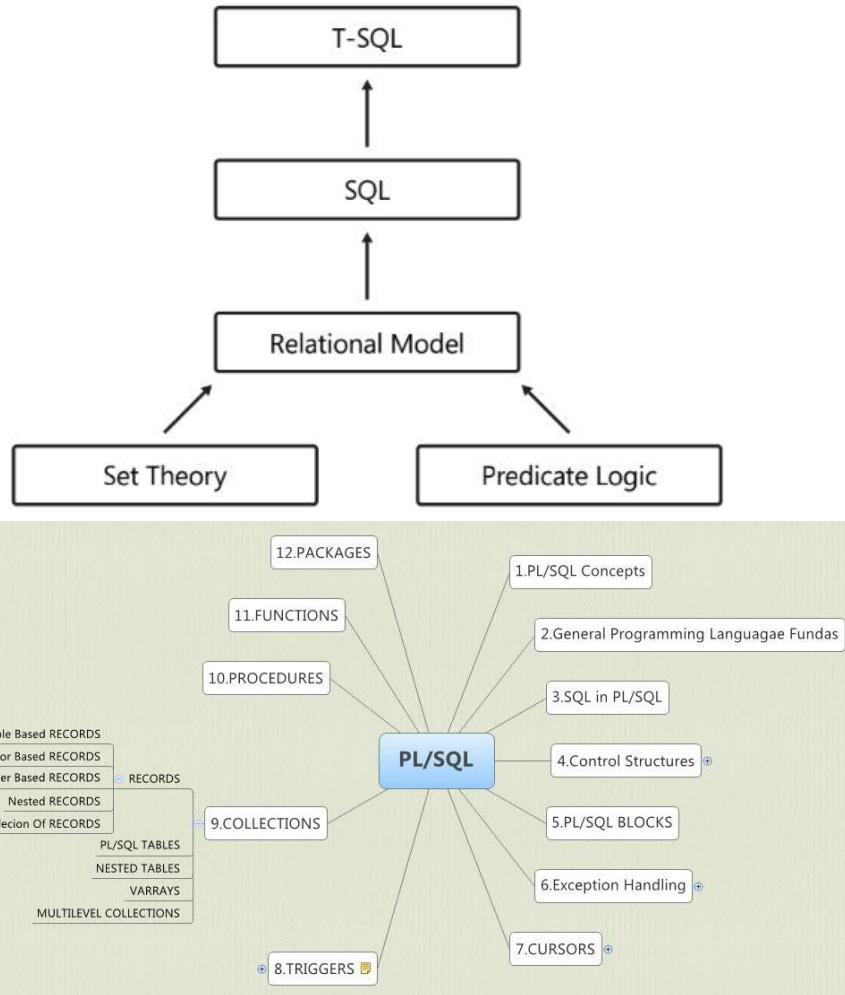


Sep-20



SQL

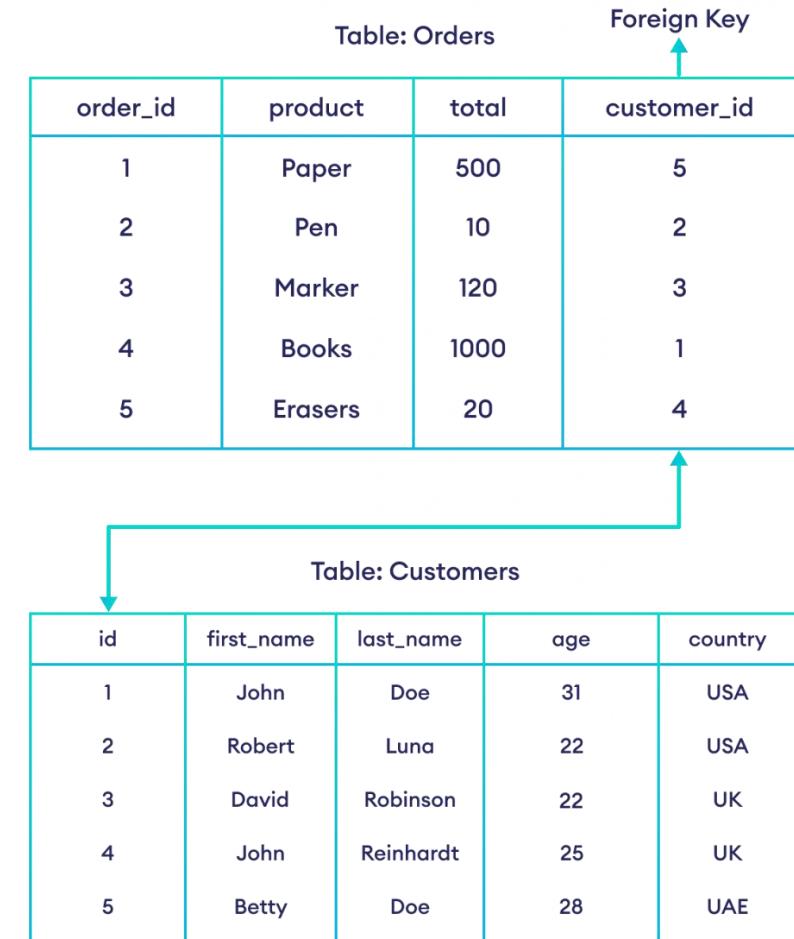
# Cơ sở của truy vấn - SQL



# Các khái niệm cơ bản trong CSDL

Các khái niệm:

- Bảng, cột, hàng
- Views
- Ngôn ngữ truy vấn SQL (select, insert, delete, update, ...)
- Kiểu dữ liệu
  - Ký tự: char, nchar, varchar, nvarchar, text
  - Số: int, numeric, float
  - Ngày giờ: date, datetime
- Primary key, foreign key, constraint
- Index (cluster index, non-cluster index)
- Data dictionary (db metadata)
- Function, store, package
- User, phân quyền
- ...



# Các nhóm lệnh cơ bản trong SQL

Tên lệnh	Diễn giải
SELECT	Là lệnh thông dụng nhất, dùng để lấy, xem dữ liệu trong CSDL.
INSERT UPDATE DELETE	Là 3 lệnh dùng để nhập thêm những row mới, thay đổi nội dung dữ liệu trên các row hay xoá các row trong table. Những lệnh này được gọi là các lệnh thao tác dữ liệu DML (Data Manipulation Language)
CREATE ALTER DROP RENAME	Là các lệnh dùng để thiết lập, thay đổi hay xoá bỏ cấu trúc dữ liệu như là table, view, index. Những lệnh này được gọi là các lệnh định nghĩa dữ liệu DDL (Data Definition Language)
COMMIT ROLLBACK SAVE POINT	Quản lý việc thay đổi dữ liệu bằng các lệnh DML. Việc thay đổi dữ liệu có thể được nhóm lại thành các transaction.
GRANT REVOKE	2 lệnh này dùng để gán hoặc huỷ các quyền truy nhập vào CSDL Oracle và các cấu trúc bên trong nó. Những lệnh này được gọi là các lệnh điều khiển dữ liệu DCL (Data Control Language)



# Các loại dữ liệu trong SQL

Loại dữ liệu	Kích thước	Diễn giải
CHAR(n)	Tối đa 8.000 ký tự	<p>n là kích thước hay số ký tự có thể lưu trữ.</p> <p>Độ dài của chuỗi ký tự là cố định.</p> <p>Không thể dùng được cho các ký tự Unicode.</p>
VARCHAR(n) VARCHAR(max)	Tối đa 8000 ký tự hoặc theo số tối đa.	<p>n là số ký tự lưu trữ.</p> <p>Độ dài tùy biến.</p> <p>Nếu chỉ định là <b>max</b> thì tối đa là 2GB.</p> <p>Không chứa ký tự Unicode.</p>
TEXT	Tối đa 2GB	<p>Độ dài tùy biến.</p> <p>Không chứa ký tự Unicode.</p>
NCHAR(n)	Tối đa 4000 ký tự.	<p>Độ dài cố định.</p> <p>Ký tự Unicode.</p>
NVARCHAR(n) NVARCHAR(max)	Tối đa 4000 ký tự hoặc theo số tối đa.	<p>n là số ký tự lưu trữ.</p> <p>Độ dài tùy biến.</p> <p>Nếu số <b>max</b> được chỉ định thì số ký tự tối đa là 2GB.</p> <p>Ký tự Unicode.</p>



# Các loại dữ liệu trong SQL

Loại dữ liệu	Kích thước	Diễn giải
BIT	0 hoặc 1	
TINYINT	Từ <b>0</b> đến <b>2<sup>8</sup>-1</b> (hay 255).	Số nguyên (Z)
SMALLINT	Từ <b>-2<sup>15</sup></b> (hay -32.768) Đến <b>2<sup>15</sup>-1</b> (hay 32.767)	Số nguyên (Z)
INT	Từ <b>-2<sup>31</sup></b> (-2.147.483.648) Đến <b>2<sup>31</sup>-1</b> (2.147.483.647).	Số nguyên (Z)
BIGINT	Từ <b>-2<sup>63</sup></b> (-9.223.372.036.854.775.808) Đến <b>2<sup>63</sup>-1</b> (9.223.372.036.854.775.807)	Số nguyên (Z)
DECIMAL(m,d)	<b>m</b> sẽ được mặc định là 18 nếu có sự chỉ định cụ thể từ phía người dùng. <b>d</b> được mặc định là 0 nếu có sự chỉ định cụ thể từ phía người dùng.	<b>m</b> tổng số lượng các chữ số. <b>d</b> là số lượng các chữ số nằm bên phải của dấu phẩy.
NUMERIC(m,d)	<b>m</b> sẽ được mặc định là 18 nếu có sự chỉ định cụ thể từ phía người dùng. <b>d</b> được mặc định là 0 nếu có sự chỉ định cụ thể từ phía người dùng.	Được hiểu tương đương với kiểu dữ liệu DECIMAL. <b>m</b> tổng số lượng các chữ số. <b>d</b> là số lượng các chữ số nằm bên phải của dấu phẩy.
FLOAT(n)	Số dấu phẩy động <b>n</b> sẽ được mặc định là 53 nếu có sự chỉ định cụ thể từ phía người dùng.	<b>n</b> là số lượng các bit được lưu trữ thuộc trong một kí hiệu hóa học.



# Các loại dữ liệu trong SQL

Loại dữ liệu	Kích thước	Diễn giải
DATE	Dữ liệu ngày được lấy từ từ '0001-01-01' đến '9999-12-31'.	Dữ liệu sau khi nhập đủ sẽ được hiển thị dưới dạng 'YYYY-MM-DD'
DATETIME	Từ '1753-01-01 00:00:00' Đến '9999-12-31 23:59:59'. Dữ liệu giờ được lấy từ '00:00:00' cho đến '23:59:59.997'	Dữ liệu sau khi nhập đủ sẽ được hiển thị dưới dạng 'YYYY-MM-DD hh:mm:ss[.mmm]'
DATETIME2	Dữ liệu ngày được lấy từ từ '0001-01-01' đến '9999-12-31'. Dữ liệu giờ lấy từ '00:00:00' đến '23:59:59.999999'.	Dữ liệu sau khi nhập đủ sẽ được hiển thị dưới dạng 'YYYY-MM-DD [hh:mm:ss]' (phần thập phân).
SMALLDATETIME	Dữ liệu ngày từ '1900-01-01' đến '2079-06-06'. Dữ liệu giờ từ '00:00:00' đến '23:59:59'.	Dữ liệu sau khi nhập đủ sẽ được hiển thị dưới dạng 'YYYY-MM-DD hh:mm:ss'.
TIME	Dữ liệu ngày từ '0001-01-01' đến '9999-12-31'. Dữ liệu giờ từ '00:00:00.0000000' đến '23:59:59.9999999'.	Dữ liệu sau khi nhập đủ sẽ được hiển thị dưới dạng 'YYYY-MM-DD hh:mm:ss[.nnnnnnnn]'
DATETIMEOFFSET	Dữ liệu ngày từ '0001-01-01' đến '9999-12-31'. Dữ liệu giờ giá trị từ '00:00:00' đến '23:59:59.999999'. Dữ liệu múi giờ lấy từ -14:00 đến +14:00.	Dữ liệu chính xác tới số thập phân của giây. Dữ liệu sau khi nhập đủ sẽ được hiển thị dưới dạng YYYY-MM-DD hh:mm:ss[.nnnnnnnn]'{[+ -]}hh:mm



# Cơ sở của truy vấn - SQL

## Câu hỏi:

- Tiêu chuẩn SQL dựa trên cái gì? (What is Standard SQL based on?)
- Cơ sở toán học của SQL?
- Khác nhau giữa SQL, T-SQL và PL/SQL?

## Trả lời:

- Tiêu chuẩn SQL dựa trên mô hình dữ liệu quan hệ, là một mô hình toán học để quản lý và thao tác dữ liệu.
- Set theory (Lý thuyết tập hợp) và predicate logic (logic tuyến tính)
- SQL là chuẩn; T-SQL và PL/SQL là mở rộng của SQL; T-SQL do Microsoft phát triển và áp dụng cho SQL Server; PL/SQL do Oracle phát triển và áp dụng cho Oracle Database



# Data dictionary

- Data dictionary lưu trữ thông tin về định nghĩa của cơ sở dữ liệu. Bao gồm các thông tin như tables, indexes, columns, datatypes, and views.
- Có thể khai thác các dữ liệu trong Data dictionary để có cái nhìn tổng quan khi mới tiếp cận một database server hoặc một database.
  - Database server có bao nhiêu database?
  - Chi tiết các database có trong server?
  - Database server có bao nhiêu user?
  - Database server đang có bao nhiêu connection?
  - Database có bao nhiêu table, mỗi table có bao nhiêu row?
  - Chi tiết column trong các table?
  - Toàn bộ các foreign key trong database?

# Truy vấn dữ liệu - SELECT

➤ Thứ tự viết câu Select:

1. SELECT
2. FROM
3. WHERE
4. GROUP BY
5. HAVING
6. ORDER BY

➤ Thứ tự thực hiện câu Select:

1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT
6. ORDER BY

```
SELECT
    state
    , city
    , substring(zip_code, 1, 3) as short_zip_code
    , COUNT(*) AS num_customers
FROM
    sales.customers
WHERE
    phone is not null
GROUP BY
    state, city, substring(zip_code, 1, 3)
HAVING
    COUNT(*) > 5
ORDER BY
    state, city
```



# Truy vấn dữ liệu – SELECT – From

➤ Thứ tự thực hiện câu Select:

1. **FROM**: chuẩn bị dữ liệu
2. **WHERE**: lọc dữ liệu
3. **GROUP BY**: nhóm dữ liệu
4. **HAVING**: lọc dữ liệu (theo nhóm trả về từ group by)
5. **SELECT**: nhặt dữ liệu
6. **ORDER BY**: sắp xếp dữ liệu

```
SELECT
    state
    , city
    , substring(zip_code, 1, 3) as short_zip_code
    , COUNT(*) AS num_customers
FROM
    sales.customers
WHERE
    phone is not null
GROUP BY
    state, city, substring(zip_code, 1, 3)
HAVING
    COUNT(*) > 5
ORDER BY
    state, city
```



# Truy vấn dữ liệu - SELECT

➤ Thứ tự viết câu Select:

1. SELECT
2. FROM
3. WHERE
4. GROUP BY
5. HAVING
6. ORDER BY

➤ Thứ tự thực hiện câu Select:

1. FROM
2. ON
3. JOIN
4. WHERE
5. GROUP BY
6. GROUPING SETS or WITH CUBE or WITH ROLLUP
7. HAVING
8. SELECT
9. DISTINCT
10. ORDER BY
11. TOP



# Truy vấn dữ liệu - SELECT

## Câu hỏi:

- Tại sao không được phép tham chiếu đến cột alias định nghĩa ở mệnh đề SELECT trong mệnh đề WHERE?
- Tại sao không được phép tham chiếu đến cột alias định nghĩa ở mệnh đề SELECT trong mệnh đề SELECT ?
- Khác nhau giữa WHERE và HAVING

```
SELECT
    state
    , city
    , substring(zip_code, 1, 3) as short_zip_code
FROM
    sales.customers
WHERE
    short_zip_code = '958'
```

```
SELECT
    state
    , city
    , substring(zip_code, 1, 3) as short_zip_code
    , short_zip_code + '000'
FROM
    sales.customers
WHERE
    phone is not null
```



# Select distinct, Select top

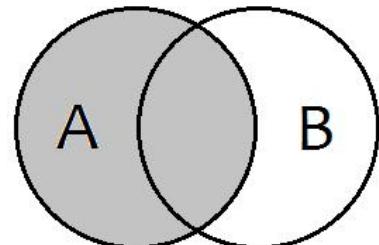
1. Lấy danh sách không trùng lặp thông tin gồm state và city của sales.customers

```
select distinct
    state
    ,city
from
    sales.customers
```

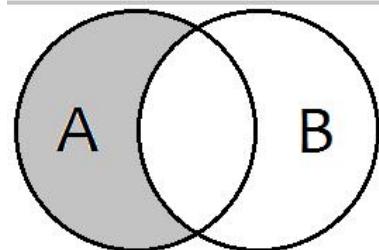
2. Lấy danh sách 100 khách hàng đầu tiên sắp xếp theo tên

```
select top 100
*
from
    sales.customers
order by
    first_name
```

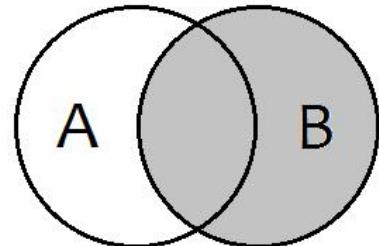
# SELECT JOIN



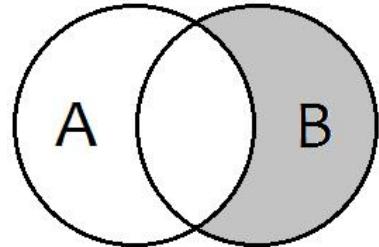
```
SELECT *  
FROM TableA a  
LEFT JOIN TableB b  
ON a.Key = b.Key
```



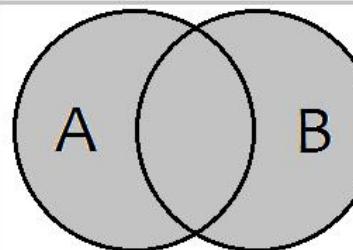
```
SELECT *  
FROM TableA a  
LEFT JOIN TableB b  
ON a.Key = b.Key  
WHERE b.Key IS NULL
```



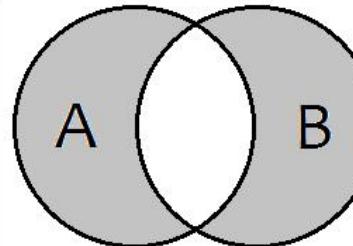
```
SELECT *  
FROM TableA a  
RIGHT JOIN TableB b  
ON a.Key = b.Key
```



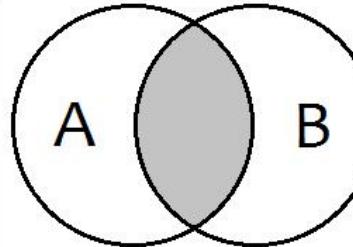
```
SELECT *  
FROM TableA a  
RIGHT JOIN TableB b  
ON a.Key = b.Key  
WHERE a.Key IS NULL
```



```
SELECT *  
FROM TableA a  
FULL OUTER JOIN TableB b  
ON a.Key = b.Key
```



```
SELECT *  
FROM TableA a  
FULL OUTER JOIN TableB b  
ON a.Key = b.Key  
WHERE a.Key IS NULL  
OR b.Key IS NULL
```



```
SELECT *  
FROM TableA a  
INNER JOIN TableB b  
ON a.Key = b.Key
```

## SQL JOINS



# SELECT JOIN

## Combining Data Tables – SQL Joins Explained

A JOIN clause in SQL is used to combine rows from two or more tables, based on a related column between them.

Table 1

1		
2		

Table 2

1	
3	
4	

Outer Join

1			
2			
3			
4			

Union

1		
2		
1		
3		

Inner Join

1			

Left Join

1			
2			

Cross Join

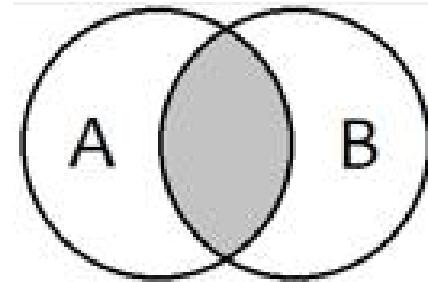
1			1
1			3
1			4
2			1
2			3
2			4



# SELECT JOIN – Inner Join

- Lấy danh sách Tên sản phẩm, Tên nhãn hàng, Tên loại sản phẩm

```
select
    pr.product_name
    ,br.brand_name
    ,ca.category_name
from
    production.products pr
    inner join production.brands br on br.brand_id=pr.brand_id
    inner join production.categories ca on ca.category_id=pr.category_id
order by
    ca.category_name
```



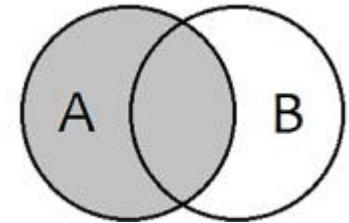
- Lấy danh sách Tên nhân viên và tên quản lý của nhân viên đó

```
select
    st.staff_id
    ,st.first_name+ ' '+st.last_name name
    ,ma.staff_id manager_id
    ,ma.first_name+ ' '+ma.last_name manager_name
from sales.staffs st
    join sales.staffs ma on ma.staff_id=st.manager_id
order by
    st.staff_id
```

# SELECT JOIN – Left Join, Right join

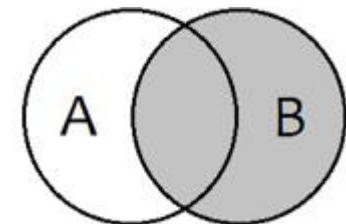
- Left join: Lấy danh sách Tên nhân viên và tên quản lý của nhân viên đó bao gồm cả nhân viên không có quản lý

```
select
    st.staff_id
    ,st.first_name+' '+st.last_name staff_name
    ,ma.staff_id manager_id
    ,ma.first_name+' '+ma.last_name manager_name
from sales.staffs st
    left join sales.staffs ma on ma.staff_id=st.manager_id
order by
    st.staff_id
```



- Right join:

```
select
    st.staff_id
    ,st.first_name+' '+st.last_name name
    ,ma.staff_id manager_id
    ,ma.first_name+' '+ma.last_name manager_name
from sales.staffs st
    right join sales.staffs ma on ma.staff_id=st.manager_id
order by
    manager_id
```



# Bài tập thực hành

- Viết câu truy vấn:
  - Lấy thông tin đầy đủ của một order
  - Lấy thông tin chi tiết của từng dòng bán hàng order\_items
  - Lấy danh sách customer có số điện thoại
  - Thống kê Doanh số bán hàng theo từng sản phẩm
  - Thống kê số lượng đơn hàng của từng cửa hàng
  - Thống kê doanh số bán hàng của từng cửa hàng
  - Thống kê doanh số bán hàng theo từng nhân viên
  - Thống kê số lượng tồn kho theo từng sản phẩm
  - Thống kê doanh số bán hàng theo từng thương hiệu sản phẩm



# THANK YOU !

**COLE.VN**  
Connecting knowledge



[www.cole.vn](http://www.cole.vn)