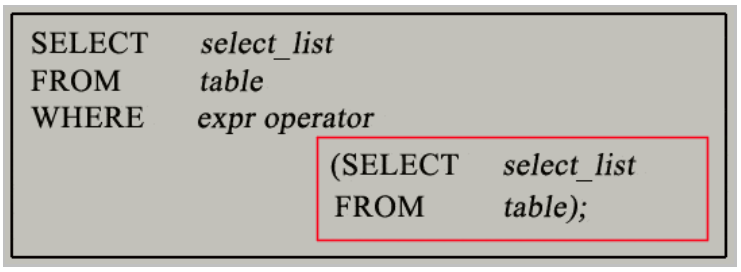
1. **Subquery**

Khi dùng Subquery để lấy ra 1 table, 1 record hoặc 1 value, sau đó result của Subquery sẽ được sử dụng tiếp trong outer SELECT.

Subquery có thể sử dụng trong WHERE với các keyword

* IN
* EXISTS
* ALL, ANY, SOME

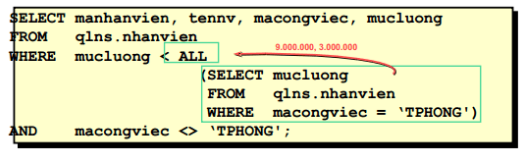


*Cú pháp*

VD: SELECT officeCode, city, phone

FROM offices

WHERE country IN ('USA','France');



Chú ý:

SOME tương tự ANY

* SOME(1,2,3) 🡺 a > 1 thì đều thỏa mãn
* = SOME, = ANY 🡺 tương đương với IN
* ! SOME, ! ANY 🡺 tương đương với NOT IN

Cú pháp:

SELECT  column\_name  
FROM  table\_name  
WHERE  EXISTS

(SELECT  column\_name

FROM  table\_name

WHERE  condition

….);

1. **VIEW**

* Một view có thể coi như logic của table, và là một "bảng ảo" có nội dụng được xác định từ một câu truy vấn
* View không lưu trực tiếp data trong Database 🡺 Khi thay đổi dữ liệu của các bảng có liên quan đến View, thì khi truy xuất vào View, dữ liệu sẽ thay đổi
* View không tự thay đổi nếu cấu trúc của bảng có thay đổi
* View chỉ nên Select không nên Update, Insert, Delete Record

Thực chất việc Update, Insert, Delete Record là gọi tới các table gốc và update trên đó

* Khi **Insert, Update, Delete** Record vào View thì câu lệnh phải đảm bảo
  + Không chứa từ khóa DISTINCT
  + Không chứa Aggregate functions (COUNT, SUM, MIN, MAX, AVG)
  + Không chứa Order By, Group By, Having
  + Where không chứa SubQuery
  + Cú pháp

UPDATE *name\_view*

SET *colume\_1* = *value\_1*

WHERE *condition*

* Ưu điểm
  + Bảo mật dữ liệu
  + Đơn giản hóa các thao tác truy vấn dữ liệu
  + Tập trung và đơn giản hóa dữ liệu
  + Độc lập với structure của table
  + Thường dùng làm report
* Nhược điểm
  + Query lâu hơn bình thường (do phải chuyển đổi từ select 🡪 View)
* Cú pháp:
  + CREATE VIEW

CREATE VIEW view\_name AS

SELECT *column\_name\_1, column\_name\_2*

FROM *table\_name*

…;

* + SELECT data

SELECT *column\_name\_1, column\_name\_2*

FROM *view\_name*

* + DROP VIEW

DROP VIEW IF EXISTS view\_name