**Bài tập lý thuyết về SQL:**

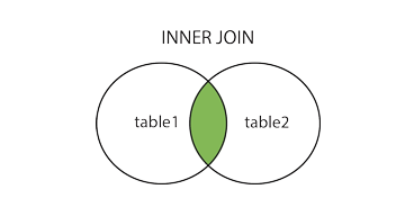
1. Có mấy loại khoá của bảng? Định nghĩa và ý nghĩa của từng loại? Một bảng có thể có nhiều loại khoá không? Một bảng có thể có nhiều khoá cùng 1 loại hay không? Các giá trị của khoá có thể là NULL hay không?

* Có 2 loại khóa của bảng là khóa chính và khóa phụ
* **Khóa chính**: là một trường hoặc nhiều trường được kết hợp là một bản ghi duy nhất. Không trường nào trong khóa chính được chứa giá trị NULL. Một bảng chỉ chứa duy nhất một khóa chính
* **Khóa phụ**: được dùng để tăng tính tham chiếu trong CSDL. Khóa ngoại nghĩa là giá trị trong bảng này phải xuất hiện trong bảng khác.
* Một bảng có thể có nhiều loại khóa xuất hiện
* Một bảng chỉ xuất hiện duy nhất một khóa chính, có thể xuất hiện nhiều khóa ngoại
* Giá trị của khóa chính không được chứa giá trị NULL, giá trị của khóa ngoại có thể chứa được giá trị NULL

1. Có mấy loại Join, ý nghĩa của từng loại, lấy ví dụ minh hoạ.

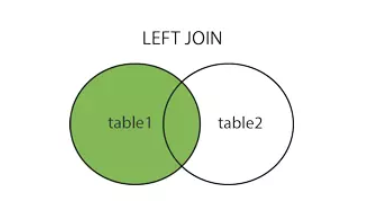
* Các loại JOIN:
* **INNER JOIN (Hoặc JOIN)**

Trả về tất cả các hàng khí có chứa ít nhất một giá trị ở cả hai bảng



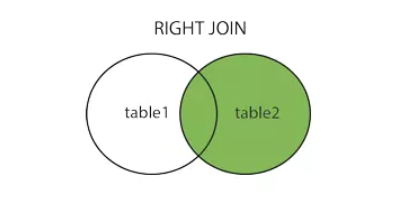
* **LEFT OUTER JOIN (Hoặc LEFT JOIN)**

Trả lại tất cả các dòng từ bảng bên trái, và các dòng đúng với điều kiện từ bảng bên phải



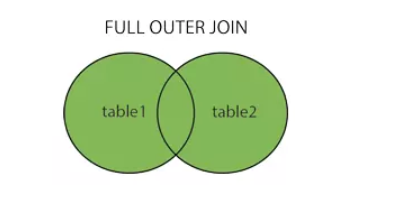
* **RIGHT OUTER JOIN (Hoặc RIGHT JOIN)**

Trả lại tất cả các hàng từ bảng bên phải, và các dòng thóa mãn điều kiện từ bàng bên trái



* **FULL OUTER JOIN (Hoặc OUTER JOIN)**

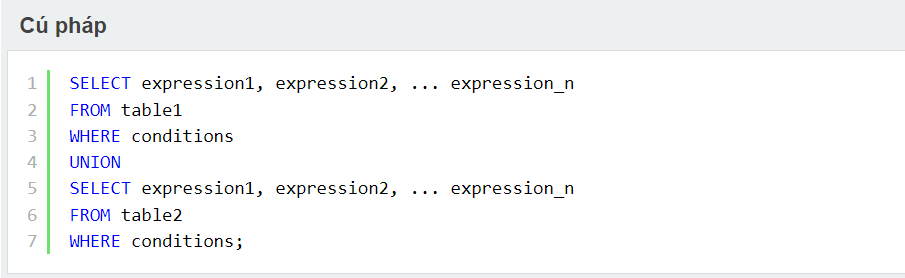
Trả lại tất cả các dòng đúng với 1 trong các bảng



1. Tìm hiểu về từ khoá UNION trong truy vấn dữ liệu? Có mấy loại UNION, cách sử dụng và ví dụ minh hoạ cho từng loại.

* Toán tử UNION được dùng để kết hợp 2 bộ kết quả từ 2 hoặc nhiều lệnh SELECT. Nó sẽ xóa các hàng trùng trong các lệnh SELECT này.

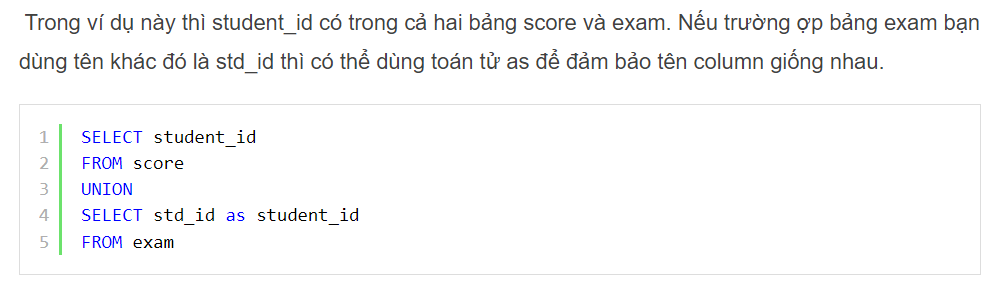
Mỗi SELECT trong toán tử UNION phải có cùng số cột trong bộ kết quả với kiểu dữ liệu tương ứng.



* Có 2 loại UNION: **UNION** và **UNION ALL**

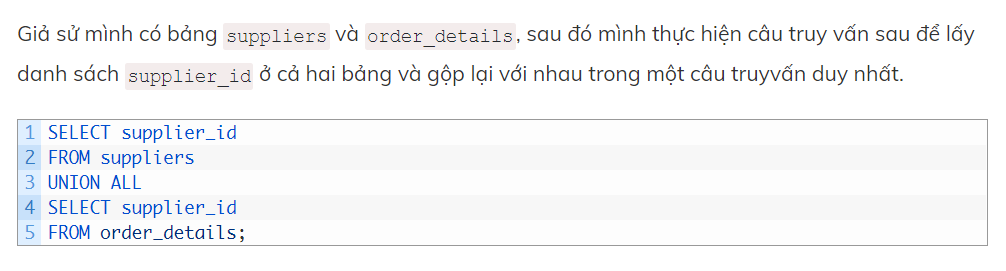
**UNION**: Kết hợp kết quả của nhiều câu hỏi với nhau, chỉ giữ lại một đại diện cho các mẫu tin trùng nhau.

Ví dụ:  Lấy danh sách student\_id từ bảng score và exam.

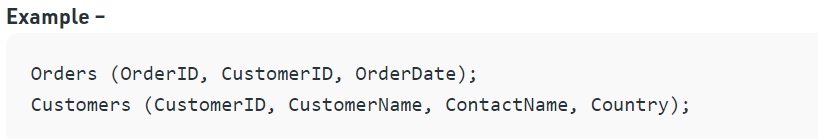


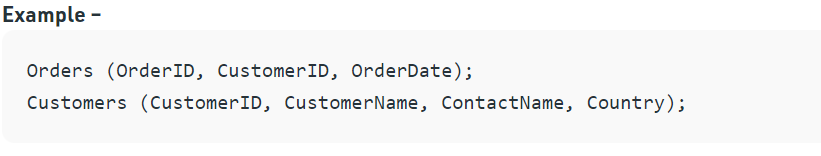
**UNION ALL**: Kết hợp kết quả của nhiều câu hỏi với nhau, các mẫu tin trùng nhau cũng được lặp lại

Ví dụ:



1. Tìm hiểu về Nested Query, Correlated Query. Phân biệt, lấy ví dụ minh hoạ.

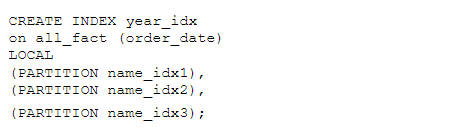
* **Nested Query**: Trong Truy vấn lồng nhau, Truy vấn bên trong chạy trước và chỉ một lần. Truy vấn bên ngoài được thực hiện với kết quả từ truy vấn Bên trong, do đó, truy vấn Bên trong được sử dụng để thực hiện truy vấn Bên ngoài.
* 
* **Correlated Query**: Trong Truy vấn Tương quan, Truy vấn Bên ngoài thực hiện đầu tiên và đối với mọi hàng Truy vấn Bên ngoài Truy vấn Bên trong được thực thi. Do đó, truy vấn Bên trong sử dụng các giá trị từ truy vấn Bên ngoài.



1. Tìm hiểu về Index? Phân biệt global index và local index ? Khi viết câu lệnh truy vấn, muốn các index được sử dụng phải chú ý gì? Lấy ví dụ minh hoạ.

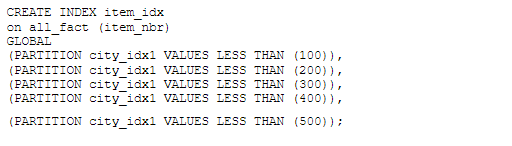
* Phương thức chỉ mục được phân vùng đầu tiên được gọi là phân vùng LOCAL. Chỉ mục được phân vùng cục bộ tạo kết quả khớp một đối một giữa các chỉ mục và các phân vùng trong bảng. Tất nhiên, giá trị khóa cho phân vùng bảng và giá trị cho chỉ mục cục bộ phải giống hệt nhau. Phương thức thứ hai được gọi là GLOBAL và cho phép chỉ mục có bất kỳ số lượng phân vùng nào.
* Việc phân vùng các chỉ mục là minh bạch đối với tất cả các truy vấn SQL. Lợi ích tuyệt vời là công cụ truy vấn Oracle sẽ chỉ quét phân vùng chỉ mục được yêu cầu để phục vụ truy vấn, do đó tăng tốc truy vấn đáng kể. Ngoài ra, công cụ truy vấn song song của Oracle sẽ cảm nhận rằng chỉ mục được phân vùng và sẽ kích hoạt các truy vấn đồng thời để quét các chỉ mục.

**LOCAL INDEX**: cho phép DBA lấy các phân vùng riêng lẻ của bảng và các chỉ mục ngoại tuyến để bảo trì (hoặc tổ chức lại) mà không ảnh hưởng đến các phân vùng và chỉ mục khác trong bảng.  
  
Trong chỉ mục được phân vùng cục bộ, các giá trị khóa và số lượng phân vùng chỉ mục sẽ khớp với số lượng phân vùng trong bảng cơ sở.



**GLOBAL INDEX**: được sử dụng cho tất cả các chỉ mục khác ngoại trừ chỉ mục được sử dụng làm khóa phân vùng bảng. Các ứng dụng OLTP (xử lý giao dịch trực tuyến) phân vùng chỉ mục toàn cầu trong đó yêu cầu ít đầu dò chỉ mục hơn so với các chỉ mục được phân vùng cục bộ. Trong lược đồ phân vùng chỉ mục chung, chỉ mục khó duy trì hơn vì chỉ mục có thể kéo dài các phân vùng trong bảng cơ sở.

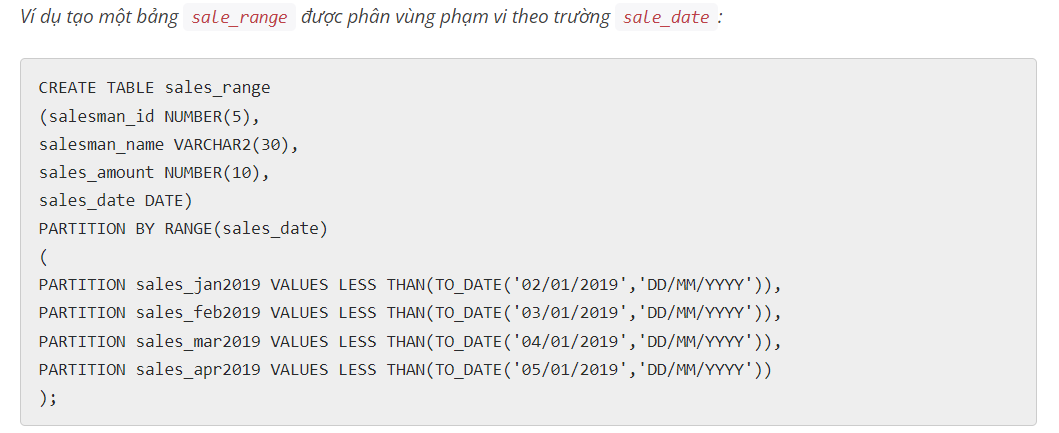
**Ví dụ:** khi phân vùng bảng bị loại bỏ như một phần của việc tổ chức lại, toàn bộ chỉ mục chung sẽ bị ảnh hưởng. Khi xác định một chỉ mục được phân vùng toàn cục, DBA hoàn toàn có quyền tự do chỉ định bao nhiêu phân vùng cho chỉ mục như mong muốn.



* Khi viết câu lệnh sử dụng INDEX thì lưu ý:
  + Không nên sử dụng trong các bảng nhỏ, ít bản ghi.
  + Không nên sử dụng Index trong bảng mà các hoạt động UPDATE, INSERT xảy ra thường xuyên với tần suất lớn.
  + Không nên sử dụng cho các cột mà chứa một số lượng lớn giá trị NULL.
  + Không nên dùng Index cho các cột mà thường xuyên bị sửa đổi.
  + INDEX sẽ mất công dụng khi bị gán giá trị với hàm

1. Tìm hiểu về partition của bảng, nên tạo partition cho bảng trong trường hợp nào? Lấy ví dụ minh hoạ?

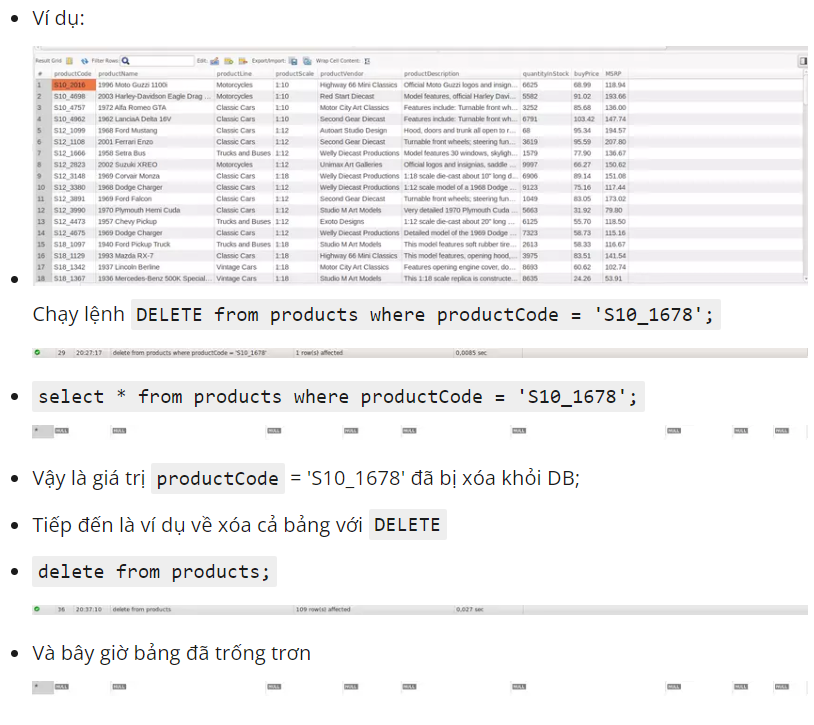
* Trường hợp tạo **PARTITION** cho bảng:
* Bảng lớn hơn 2GB luôn được coi là ứng viên cho phân vùng
* Bảng chứa dữ liệu lịch sử, những dữ liệu mới sẽ được thêm vào phân vùng mới
* Khi nội dung của bảng cần được phân phối trên nhiều loại thiết bị storage khác nhau



1. Phân biệt câu lệnh delete và truncate? Cách sử dụng?

* **DELETE**:

Khi chạy lệnh **DELETE** thì SQL sẽ log lại từng dòng đã xóa vào transacion log, vì thế khi bạn tạo record mới, giá trị id sẽ không bao giờ bắt đầu từ 1 mà sẽ bắt đầu với giá trị là n+1 với n là giá trị của record cuối cùng được tạo

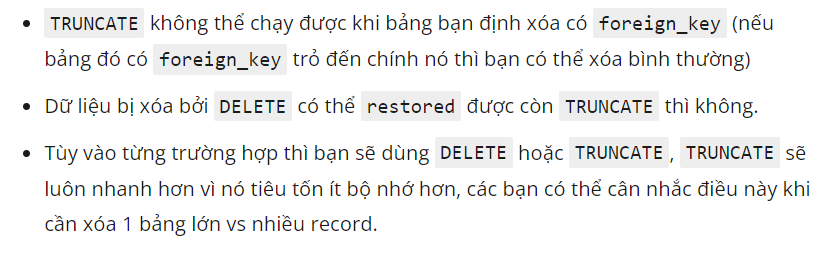


* **TRUNCATE:**

Đối với TRUNCATE chúng ta chỉ có thể dùng cho bảng chứ không thể dùng cho từng record

Khi chạy TRUNCATE thì SQL sẽ xóa hết dữ liệu quả bảng và reset transacion log, vì thế khi tạo record mới thì id sẽ bắt đầu với giá trị từ 1

**Lưu ý:**

****