**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 hai loại khóa của bảng: primary key và foreign key
* Primary key: Nó là một cột (hoặc các cột) có giá trị duy nhất và không được chứa giá trị Null. Một bảng chỉ có thể có một khóa chính, khóa này có thể bao gồm một hoặc nhiều cột
* Foreign key: Nó là một cột(hoặc các cột) tham chiếu đến một cột( thường là khóa chính) của một bảng khác, có thể chứa giá trị trùng lặp và giá trị Null. Một bảng có thể có nhiều khóa ngoại

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

Có 5 loại join:

* Outer join: Lấy đủ cả 2 table, row nào ở 2 table không có dữ liệu tương ứng thì đặt null

Vd: Hiển thị tên nhân viên, tên phòng làm việc

select ename**,** dname from emp e join dept d on **(**e**.**deptno **=** d**.**deptno**)**

* Left join: Muốn lấy toàn bộ Left table, nếu right table không có dữ liệu tương ứng thì đặt Null

Vd: Hiển thị tên nhân viên, phòng ban cả nhân viên chưa làm việc trong phòng ban nào

select ename**,** dname from emp e left join dept d on **(**e**.**deptno **=** d**.**deptno**)**

* Right join: Muốn lấy toàn bộ Right table, nếu left table không có dữ liệu tương ứng thì đặt Null

Vd: Hiển thị tên nhân viên, tên phòng ban làm việc cả phòng ban chưa có nhân viên làm việc

select ename**,** dname from emp e right join dept d on **(**e**.**deptno **=** d**.**deptno**)**

* Cross join: Từng row của left table nối với từng row của right table.

Vd:

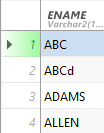
select ename**,** loc**,** dname from emp e cross join dept d

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.**

* Có hai loại UNION: UNION và UNION ALL
* UNION thực hiện DISTINCT trên tập kết quả, loại bỏ mọi hàng trùng lặp.
* UNION ALL không loại bỏ các bản trùng lặp
* Ví dụ
* xóa bỏ các tên trùng lặp trong kết quả truy vấn

select ename from emp\_tmp where ename like 'A%'

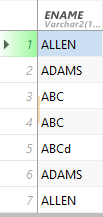
Union select ename from emp where ename like 'A%'



* hiển thị tất cả các bản ghi không xóa bản ghi trùng lặp giữa hai bảng

select ename from emp\_tmp where ename like 'A%'

Union all select ename from emp where ename like 'A%'



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

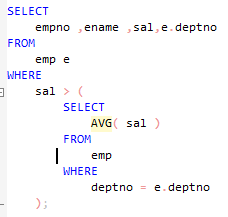
* Nested Query: Là một câu lệnh Select được lồng trong một câu lệnh select khác và trả về kết quả trung gian. Truy vấn con được tách biệt độc lập và không phụ thuộc vào giá trị truy vấn từ bên ngoài

Vd: Hiển thị thông tin của nhân viên có lương lớn nhất

select ename**,** job**,** deptno from emp where sal in**(**select max**(**sal**)** from emp**)**

* Correlated Query: Là một câu lệnh Select được lồng trong một câu lệnh select khác. Truy vấn con phụ thuộc vào giá trị truy vấn từ bên ngoài

Vd: Hiển thị tên nhân viên, mã nhân viên, lương, mã phòng với lương của nhân viên lớn hơn lương trung bình tại phòng ban mà nhân viên đó đang làm việc



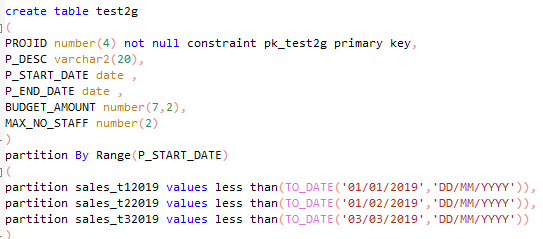
1. **Tìm hiểu về Index? Phân biệt global index và local index? Khi viết câu lệnh truy vấn, muốn các index được sử dụng phải chú ý gì? Lấy ví dụ minh hoạ.**

* Index là cấu trúc hình cây cho phép truy xuất trực tiếp một row trong table.
* Global index: Là mối quan hệ 1-nhiều, cho phép một phân vùng chỉ mục ánh xạ tới nhiều phân vùng bảng
* Local index: Là ánh xạ 1-1 giữa phân vùng chỉ mục và phân vùng bảng
* Khi viết câu lệnh truy vấn, muốn các index được sử dụng phải chú ý
* Không nên sử dụng trong các bảng nhỏ, ít bản ghi.
* Chọn những column được dùng nhiều trong mệnh đề where
* Chọn những column được dùng nhiều để join các bảng
* 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.
* Vd: Tạo index có tên projid cho column projid trong bảng ass\_giangnp

create index PROJID\_index on ass\_giangnp**(**PROJID**)**

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

* **Partitioned Table là 1 bảng đã được chia nhỏ thành các phần (hay các partition) theo 1 cột nào đó trong bảng.**
* Nên tạo partition cho bảng trong trường hợp
  + Table có dung lượng từ 2GB trở lên nên cân nhắc được partition.
  + Table có nội dung có thể được chia thành các khu vực có mức độ ưu tiên khác nhau (VD: khu vực dành cho dữ liệu thường xuyên được truy vấn,hoặc ít được truy vấn, …)
* Vd: Tạo phân vùng theo tháng với khóa phân vùng là trường P\_Start\_date



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

* DELETE xóa một hoặc tất cả các hàng từ một bảng dựa trên điều kiện và có thể được phục hồi lại.
* TRUNCATE xóa tất cả các hàng từ một bảng bằng cách phân bổ các trang bộ nhớ và không thể phục hồi lại