**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:

+Khóa chính(Primary key): Là một column hoặc một tập các column xác định tính duy nhất của các rows ở trong bảng và không thể NULL.

+Khóa ngoại(Foreign key): Chỉ ra mối liên hệ ràng buộc tham chiếu giữa table này với table khác, hoặc trong chính 1 table. Nó chỉ ra mối liên hệ cha-con và chỉ ràng buộc giữa FOREIGN KEY bảng này với PRIMARY KEY hoặc UNIQUE Key của bảng khác.

-Một bảng có thể có nhiều loại khóa.

-Một bảng chỉ có thể có 1 khóa chính, nhưng có thể có nhiều khóa ngoại.

-Khóa chính không thể NULL, khóa ngoại có thể NULL.

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

-Có 4 loại join:

+INNER JOIN: Trả về kết quả là những bản ghi có giá trị phù hợp ở cả 2 bảng.

*SELECT \*  
FROM* EMP E *INNER JOIN* DEPT D *ON* E.DEPTNO = D.DEPTNO

+LEFT (OUTER) JOIN:  Trả về tất cả bản ghi từ bảng bên trái và bản ghi trùng với bảng bên phải.

*SELECT \*  
FROM* EMP E *LEFT JOIN* DEPT D *ON* E.DEPTNO = D.DEPTNO

+RIGHT (OUTER) JOIN: Trả về tất cả bản ghi từ bảng bên phải và bản ghi trùng với bản bên trái.

*SELECT \*  
FROM* EMP E *RIGHT JOIN* DEPT D *ON* E.DEPTNO = D.DEPTNO

+FULL (OUTER) JOIN: Trả về tất cả bản ghi khi có một kết quả phù hợp trong bảng bên trái hoặc bên phải.

*SELECT \*  
FROM* EMP E *FULL JOIN* DEPT D *ON* E.DEPTNO = D.DEPTNO

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.**

-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.

-Có 2 loại UNION

+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.

*SELECT JOB FROM* EMP *WHERE* DEPTNO = 10  
*Union  
SELECT JOB FROM* EMP *WHERE* DEPTNO = 30;

+UNION ALL: dùng để kết hợp 2 bộ kết quả từ 2 hoặc nhiều lệnh SELECT. Các hàng trùng nhau cũng sẽ được lặp lại.

*SELECT JOB FROM* EMP *WHERE* DEPTNO = 10  
*UNION ALL  
SELECT JOB FROM* EMP *WHERE* DEPTNO = 30;

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

+Nested Query(Truy vấn lồng): Là một câu truy vấn được định nghĩa ở bên trong một câu truy vấn khác.

*SELECT* EMPNO, ENAME, SAL, DEPTNO  
*FROM* EMP  
*WHERE* SAL > *ANY* (  
 *SELECT AVG*(SAL)  
 *FROM* EMP  
 *GROUP BY* DEPTNO)

+Correlated Query(Truy vấn liên quan): là 1 truy vấn con (một truy vấn trong một truy vấn khác) sử dụng kết qủa của truy vấn cha. Truy vấn ngoài luôn được thực hiện trước  và sau đó thi hành truy vấn con bên trong cho mỗi dòng kết quả của truy vấn bên ngoài

*SELECT* ENAME,  
 (*SELECT* DNAME *FROM* DEPT D *WHERE* E.DEPTNO = D.DEPTNO) *AS* DEPT\_NAME  
*FROM* EMP E

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 một cấu trúc cơ sở dữ liệu, được sever sử dụng để tìm một row trong bảng một cách nhanh chóng. Index bao gồm một key value (một cột (column) trong hàng (row)) và ROWID.

**-**Phân biệt GLOBAL INDEX và LOCAL INDEX:

+ Local index: là ánh xạ 1-1 giữa phân vùng index và phân vùng table. Nói chung, các local indexcho phép cách tiếp cận “divide and conquer” rõ ràng hơn để tạo ra các kế hoạch thực thi SQL nhanh với việc cắt bớt phân vùng .

+Global index: Khi viết câu lệnh truy vấn, muốn các index được sử dụng phải chú ý gì? - Nên Index những cột được dùng trong WHERE, JOIN và ORDER BY - Dùng thuộc tính NOT NULL cho những cột được Index. - Không dùng Index cho các bảng thường xuyên có UPDATE, INSERT - Không dùng Index cho các cột mà giá trị thường xuyên bị thay đổi

-Ví dụ minh họa:

+Tạo index:

*CREATE INDEX* EMP\_IDX *ON* EMP (EMPNO);

+Truy vấn bằng index:

*SELECT \* FROM* emp *WHERE* EMPNO = 100;

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

- Khi làm việc với cơ sở dữ liệu của hệ thống lớn chúng ta sẽ cần đối mặt với một trong các vấn đề vô cùng hóc búa thiết kế cơ sở dữ liệu làm sao hiệu năng tốt và ổn định. Để tối ưu những việc đó thì chúng ta xem một thứ đó là PARTITION TABLE dưới ví dụ bên dưới.

-Ví dụ minh hoạ:

Giả sử có 2 phòng họp:

+ Trong phòng họp «CỔ ĐIỂN» có 1000 người. Những người trong phòng họp có thể tự do chọn vị trí, không có bất kỳ quy tắc nào cả.

+Tại phòng họp có tên là «TÂN TIẾN», chúng ta cũng có 1000 người, tuy nhiên trong phòng họp này lại được chia thành 63 ô nhỏ ứng với 63 tỉnh thành.

+ Những người có quê quán là Nam Định sẽ vào ô có đặt tên là Nam Định, những người có quê quán là Hà Nội sẽ vào vào ô có ghi biển Hà Nội Giả sử chúng ta cần tìm tất cả những người có quê ở Ninh Bình. Bạn sẽ thấy ngay sự khác biệt khi tìm kiếm ở 2 phòng trên:

+ Đối với phòng họp «CỔ ĐIỂN», chúng ta sẽ phải lọc trong 1000 người để tìm xem ai có quê NINH BÌNH

+ Đối với phòng họp «TÂN TIÊN», chúng ta chỉ cần bước vào phòng và đi tìm ô có tên «NINH BÌNH». Tất cả những người đứng trong ô này sẽ hiển nhiên thỏa mãn điều kiện ⎝ Trong ví dụ trên thiết kế TABLE chứa dữ liệu theo kiểu phòng họp TÂN TIẾN là cách thiết kế PARTITION TABLE.

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

-DELETE : Xóa một hay tất cả dòng trong một bảng theo một điều kiện nhất định, dữ liệu có thể phục hồi lại

*DELETE FROM* EMP *WHERE* EMPNO = '123'

-TRUNCATE : Xóa toàn bộ các dòng của bảng, giải phóng bộ nhớ và không thể phục hồi lại

*TRUNCATE TABLE* EMP