# **Bài tập lý thuyết về SQL:**

1. **Có mấy loại khoá của bảng?**

Có hai loại:

* Khóa chính (Primary key)
* Khóa ngoại (Foreign key)

**Định nghĩa và ý nghĩa của từng loại?**

* Khóa chính: chỉ ra dàng buộc duy nhất, tuy nhiên khóa là dạng khóa UNIQUE cấp cao nhất. Một table thì chỉ có một khóa chính (Primary key). Các giá trị trong khóa chính phải NOT NULL.
* Khóa ngoại: Chỉ ra mối liên hệ ràng buộc tham chiếu giữa table này với table khác, hoặc trong chính một table. Nó chỉ ra mối liên hệ cha-con và chỉ ra ràng buộc giữa FOREIGN KEY bảng này với PRIMARY KEY hoặc UNIQUE KEY của bảng khác.

**Một bảng có thể có nhiều loại khoá không?**

* Có thể có nhiều khóa ngoại, nhưng chỉ có duy nhất một khóa chính.

**Một bảng có thể có nhiều khoá cùng 1 loại hay không?**

* Có thể có nhiều khóa cùng 1 loại, tuy nhiên chỉ có thể là khóa ngoại.

**Các giá trị của khoá có thể là NULL hay không?**

* Khóa chính thì không thể NULL, còn khóa ngoại thì có thể NULL.

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

Có 4 loại Join chính:

* Inner Join: trả về các bản ghi mà trường được join hai bảng khớp nhau, các bản ghi chỉ xuất hiện một trong hai bảng sẽ bị loại bỏ.

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

* Left Outer Join: trả về kết tất cả các hàng ở bảng bên trái và các hàng tương đương bên phải. Chấp nhận cả dữ liệu Null ở bảng bên phải.

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

* Right Outer Join: trả về tất cả các hàng ở bảng bên phải và các hàng tương đương bên trái. Chấp nhận cả dữ liệu Null ở bảng bên trái.

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

* Full Outer Join: là sự kết hợp của Left Outer Join và Right Outer Join.

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

1. **Tìm hiểu về từ khoá UNION trong truy vấn dữ liệu?**

* UNION là một toán tử dùng để gộp kết quả của hai lệnh SELECT lại với nhau, nó sẽ loại bỏ những dòng trùng lặp và chỉ để lại những dòng khác nhau.
* Điều kiện để sử dụng UNION trong Oracle là các clounm ở hai lệnh SELECT phải giống nhau về cả tên colunm, thứ tự và số lượng.

**Có mấy loại UNION, cách sử dụng và ví dụ minh hoạ cho từng loại.**

Có hai loại UNINON:

* UNION: kết hợp nhiều câu SELECT lại với nhau và chỉ giữ lại những kết quả khác nhau, trùng nhau sẽ bị loại bỏ.

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

Result: CLERK

MANAGER

PRESIDENT

SALESMAN

* UNION ALL: kết hợp nhiều câu SELECT lại với nhau và các kết quả trùng nhau vẫn sẽ được lặp lại.

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

Result: MANAGER

PRESIDENT

CLERK

SALESMAN

SALESMAN

SALESMAN

MANAGER

SALESMAN

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

* Nested Query: tức là bạn sử dụng nhiều lệnh truy vấn con trong một câu lệnh cha.

**SELECT \*  
FROM DEPT  
WHERE DEPTNO NOT IN (SELECT DISTINCT DEPTNO  
 FROM EMP  
 WHERE DEPTNO IS NOT NULL);**

* Correlated Query: nó giống như Nested Query là sử dụng nhiều lệnh truy vấn con trong một câu lệnh cha, nhưng có quy trình riêng của nó. Truy vấn bên ngoài được thực hiện trước và sau đó sẽ thi hành truy vấn con bên trong cho mỗi dòng kết quả của truy vấn bên ngoài.

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

1. **Tìm hiểu về Index?**

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

**Phân biệt global index và local index?**

* Global index: là mối quan hệ một-nhiều, cho phép một index partition ánh xạ tới nhiêu phân vùng trong bảng. Global index có thể được phân vùng bằng phương thức phạm vi (the range) hoặc hash method và có nó có thể được xác định trên bất kỳ loại bảng nào được phân vùng hoặc không phân vùng.
* 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](http://www.remote-dba.net/t_11g_new_composite_partitioning.htm) .

**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

**Lấy ví dụ minh hoạ?**

* Tạo Index:

**CREATE INDEX idx ON emp (emp\_id);**

* Truy vấn bằng index:

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

1. **Tìm hiểu về partition của bảng, nên tạo partition cho bảng trong trường hợp nào?**

* 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.

**Lấy 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?**

* DELETE: Xóa một hay tất cả dòng trong một bảng theo một điều kiện nhất định, dữ liệu có thể phục hồi.
* TRUNCATE: Xóa toàn bộ các dòng trong bảng, giải phóng bộ nhớ và không thể phục hồi lại.

**Cách sử dụng?**

* DELETE:

Xóa một dòng trong bảng:

**DELETE FROM ten\_bang WHERE dieu\_kien;**

Xóa tất cả các dòng trong bảng:

**DELETE FROM ten\_bang;**

* TRUNCATE:

**TRUNCATE TABLE ten\_bang;**