**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 khoá bảng:

* Khoá chính(Primary key): là một cột hoặc một tập hợp cột được xác định mang tính duy nhất.
* Khoá ngoại(Foreign Key): Là một cột hoặc một tập các cột có tham chiếu tới chính bảng đó hoặc một bảng khác.

Một bảng có thể có nhiều loại khoá.

Một bảng chỉ có thể có 1 khoá chính, và có thể có 1 hoặc nhiều khoá ngoại, hoặc không có khoá ngoại.

Các giá trị của khoá chính không thể null, các giá trị của khoá 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: lấy dữ liệu chung của cả 2 bảng
  + SELECT \*  
    FROM PROJECT\_HIENNV  
     INNER JOIN ASSIGNMENTS\_HIENNV ON PROJECT\_HIENNV.PROJID = ASSIGNMENTS\_HIENNV.PROJID;
* Left join: Lấy toàn bộ dữ liệu của bảng bên trái và dữ liệu chung của bảng bên phải.
  + SELECT \*  
    FROM PROJECT\_HIENNV  
     LEFT JOIN ASSIGNMENTS\_HIENNV ON PROJECT\_HIENNV.PROJID = ASSIGNMENTS\_HIENNV.PROJID;
* Right join: Lấy toàn bộ dự liệu của bảng bên phải và dữ liệu chung của bảng bên trái.
  + SELECT \*  
    FROM PROJECT\_HIENNV  
     RIGHT JOIN ASSIGNMENTS\_HIENNV ON PROJECT\_HIENNV.PROJID = ASSIGNMENTS\_HIENNV.PROJID;
* Full join: Lấy toàn bộ dữ liệu của cả 2 bảng.
  + SELECT \*  
    FROM PROJECT\_HIENNV  
     FULL JOIN ASSIGNMENTS\_HIENNV ON PROJECT\_HIENNV.PROJID = ASSIGNMENTS\_HIENNV.PROJID;

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

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.
  + SELECT ASSIGN\_TYPE  
    FROM ASSIGNMENTS\_HIENNV  
    WHERE PROJID = 3  
     UNION  
    SELECT ASSIGN\_TYPE  
    FROM ASSIGNMENTS\_HIENNV  
    WHERE PROJID = 4;
* UNION ALL: Kết hợp kết quả của nhiều câu hỏi với nhau, các mẫu tin trùng nhau cũng được lặp lại
  + SELECT ASSIGN\_TYPE  
    FROM ASSIGNMENTS\_HIENNV  
    WHERE PROJID = 3  
     UNION ALL  
    SELECT ASSIGN\_TYPE  
    FROM ASSIGNMENTS\_HIENNV  
    WHERE PROJID = 4;

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 đặt bên trong một câu truy vấn khác được gọi là câu truy vấn con.

* SELECT \*  
  FROM ASSIGNMENTS\_HIENNV  
  WHERE ASSIGNMENTS\_HIENNV.PROJID IN (SELECT PROJID FROM PROJECT\_HIENNV WHERE PROJID = 3);

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 DNAME  
  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:

* 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 .

Khi viết câu lệnh truy vấn, muốn các index được sử dụng phải chú ý:

- 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

Tạo index:

* CREATE INDEX INDEX\_PROJID\_HIENNV ON ASSIGNMENTS\_HIENNV (PROJID);

Truy vấn index:

* SELECT \* FROM ASSIGNMENTS\_HIENNV WHERE PROJID = 3;

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.

CREATE TABLE PARTITION  
(  
 ID INT,  
 NAME VARCHAR2(100),  
 AGE INT,  
 ADDRESS VARCHAR2(50)  
) PARTITION BY LIST (ADDRESS)  
(  
 PARTITION ADDRESS\_A VALUES ('Bắc Ninh','Hà Nội'),  
 PARTITION ADDRESS\_B VALUES ('Nghệ An','Hà Tĩnh'),  
 PARTITION ADDRESS\_C VALUES ('TP. Hồ Chí Minh','Bình Dương')  
);

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 ASSIGNMENTS\_HIENNV  
  WHERE PROJID = 3;

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

* TRUNCATE TABLE ASSIGNMENTS\_HIENNV