**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 ngoại
* Khóa chính là khóa duy nhất để xác định 1 bản ghi trong bảng
* Khóa ngoại là 1 trường trong bảng và là khóa chính ở 1 bảng khác
* 1 bảng có thể có nhiều loại khóa
* 1 bảng chỉ có 1 khóa chính nhưng có thể có nhiều khóa ngoại
* Khóa chính không được phép 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, left join, right join, full outer join
* Inner join (join): trả về tất cả các hàng khi có ít nhất một giá trị ở cả 2 bảng
* Left join: trả lại tất cả các dòng từ bảng bên trái và tất cả các dòng đúng điều kiện từ bảng bên phải
* Right join: tương tự left join nhưng ngược lại bên
* Full outer join: trả về 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.

* Mệnh đề UNION được sử dụng để kết hợp các kết quả truy vấn select vào một truy vấn duy nhất từ tất cả truy vấn đã chọn
* Có 2 loại UNION: UNION và UNION ALL
* Cấu trúc : select <column> from <table>

UNION

select <column> from <table>

* Select ename from emp where deptno = 30  
  UNION  
  select ename from emp where deptno = 10

Kết quả trả về tên các nhân viên ở phòng 30 và phòng 10 không trùng nhau

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

* Nested Query: truy vấn lồng, một truy vấn đươc viết trong 1 truy vấn khác và kết quả của truy vấn trong được sử dụng để thực thi truy vấn ngoài
* Correlated Query: truy vấn con, truy vấn con là 1 truy vấn được viết trong 1 truy vấn khác và truy vấn con bên trong sử dụng giá trị của truy vấn vấn ngoài để thực thi câu lệnh
* Ví dụ Nested Query:  
  select ename from emp where job in (select job from emp where job like “salesman”);
* Ví dụ Correlated Query:  
  select ename,sal,deptno from emp outer  
   where sal > (select AVG(sal) from emp where deptno = outer.detpno);

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 dữ liệu, được sever sử dụng để tìm 1 row trong bảng một cách nhanh chóng nhất, index bao gồm một key value và ROWID
* global index: là mối quan hệ một nhiều cho phép index ánh xạ đến nhiều partition của bảng

vd: create index i\_sal on emp(sal)

global

(partition sal\_index1 values lessthan 15000),

(partition sal\_index1 values lessthan 10000),

(partition sal\_index1 values lessthan 5000);

* local index: là mối quan hệ một một, index ảnh xạ đến 1 partition của bảng giúp người dùng quản lý dễ hơn “chia để trị”

vd: vd: create index i\_sal on emp(sal);

* dùng index khi mà việc lấy dữ liệu <15% số row trong bảng
* không nên dùng index cho các bảng chỉ có vài row

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

* Parition: là việc phân chia một phần của table thành những phần nhỏ theo một logic nhất định được phân biệt bằng key
* Nên tạo partition trong trường hợp table có rất nhiều bản ghi
* Ví dụ minh họa: table nhân viên có trường tên và tuổi trước khi partition thì table mặc định lưu trữ thành 1 trunk trong file system và khi sử dụng partition lấy trường tuổi làm key thì khi table sẽ lưu trữ thành nhiều trunk và khi truy vấn sẽ giảm thời gian thực thi câu lệnh

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

* Đều dùng để xóa dữ liệu nhưng 2 lệnh này có những khác biệt cơ bản sau

|  |  |  |
| --- | --- | --- |
| * Tính năng | * DELETE | * TRUNCATE |
| * Lọc dữ liệu | * Delete cho phép lọc dữ liệu để xóa | * Truncate thực hiện xóa toán bộ dữ liệu bảng |
| * Roll back | * Có thể rollback data | * Không thể roll back data |
| * Không gian lưu trữ | * Không giải phóng không gian lưu trữ | * Xóa data giải phóng dung lượng |
| * Tốc độ | * Chậm | * nhanh |