**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?

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

3. 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.

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

5. 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.

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

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

**Bài Làm**

**Câu 1 :**

**Có mấy loại khoá của bảng? Định nghĩa và ý nghĩa của từng loại?**

Khóa trong hệ quản trị cơ sở dữ liệu là một thuộc tính hoặc một tập hợp các thuộc tính giúp xác định một hàng hoặc một dữ liệu trong một quan hệ, hay còn gọi là bảng. Các khóa cho phép tìm kiếm mối quan hệ giữa hai bảng với nhau. Bên cạnh đó, khóa cũng được sử dụng để tìm kiếm các bản ghi hoặc các hàng cụ thể trong một bảng của hệ quản trị cơ sở dữ liệu.

Tại sao phải cần khóa?

* Các khóa giúp xác định bất kỳ hàng dữ liệu nào trong bảng. Trong một ứng dụng, một bảng có thể chứa hàng nghìn bản ghi và các tập dữ liệu. Hơn nữa, các bản ghi có thể được sao chép. Các khóa đảm bảo việc xác định duy nhất một bản ghi trong cơ sở dữ liệu.
* Cho phép thiết lập và xác định mối quan hệ giữa các bảng
* Giúp thực thi tính duy nhất và tính toàn vẹn trong các mối quan hệ

# Các loại khóa trong hệ quản trị cơ sở dữ liệu

1. Siêu khóa (Super key)
2. Khóa chính (Primary key)
3. Khóa ứng viên (Candidate key)
4. Khóa phụ (Alternate key)
5. Khóa ngoại (Foreign key)
6. Khóa tổng hợp (Composite key)

## Siêu khóa

Siêu khóa là một tập hợp của một hoặc nhiều khóa là duy nhất nhằm xác định các hàng trong bảng.

## Khóa chính

Khóa chính là một cột hoặc nhóm các cột trong bảng xác định duy nhất mỗi hàng trong bảng đó. Khóa chính không được phép trùng lặp và giá trị của khóa chính không được nhận giá trị NULL hay còn gọi là giá trị rỗng. Một bảng chỉ duy nhất một khóa chính.

## Khóa phụ

Khóa phụ là một cột hoặc tập hợp các cột trong bảng xác định duy nhất mỗi hàng trong bảng đó. Một bảng có thể có nhiều khóa được lựa chọn làm khóa chính, các khóa được lựa chọn này được gọi là khóa ứng viên và chỉ có một khóa được chọn có thể được đặt làm khóa chính. Tất cả các khóa không được đặt làm khóa chính được gọi là khóa phụ.

## Khóa ứng viên

Khóa ứng viên là một hoặc tập hợp các thuộc tính hoặc các cột xác định duy nhất các dữ liệu trong một bảng. Khóa chính là khóa được chọn từ các khóa ứng viên. Mỗi bảng phải có ít nhất một khóa ứng viên. Một bảng có thể có nhiều khóa ứng viên nhưng chỉ có một khóa chính duy nhất. Giá trị của khóa ứng viên có thể nhận giá trị NULL hoặc giá trị rỗng.

Các thuộc tính của khóa ứng viên:

* Có thể có nhiều khóa ứng viên trong một bảng
* Nó phải chứa các giá trị duy nhất.
* Khóa ứng viên có thể có nhiều thuộc tính.
* Có thể chứa giá trị rỗng.
* Nó phải chứa các trường tối thiểu để đảm bảo tính duy nhất cho mỗi dữ liệu trong bảng.
* Xác định duy nhất từng bản ghi trong bảng

## Khóa ngoại

Khóa ngoại là một cột tạo mối quan hệ giữa hai bảng. Mục đích của khóa ngoại là duy trì tính toàn vẹn của dữ liệu và cho phép tìm kiếm giữa nhiều dữ liệu khác nhau của một thực thể. Nó đóng vai trò là một tham chiếu giữa hai bảng vì nó tham chiếu đến khóa chính của một bảng khác.

## Khóa tổng hợp

Khóa tổng hợp là khóa ứng viên mà có nhiều thuộc tính cho phép nhận dạng duy nhất một hàng của một bảng trong hệ quản trị cơ sở dữ liệu.

**Một bảng có thể có nhiều loại khoá không? Một bảng có thể có nhiều khoá cùng 1 loại hay không? Một bảng có thể có nhiều khoá cùng 1 loại hay không?** **Các giá trị của khoá có thể là NULL hay không?**

**Khóa chính**    Một bảng có thể chỉ có một khóa chính. Một khóa chính chứa một hoặc nhiều trường giúp nhận dạng riêng từng bản ghi mà bạn lưu trữ trong bảng

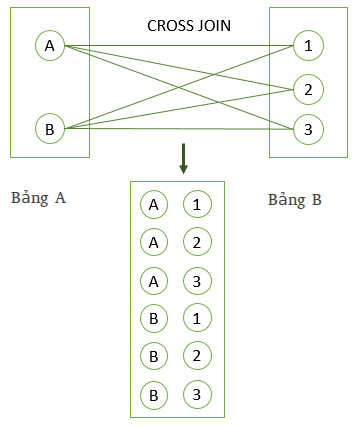
**Khóa ngoại**    Một bảng cũng có thể có một hoặc nhiều khóa ngoại. Một khóa ngoại sẽ chứa các giá trị tương ứng với các giá trị trong khóa chính của một bảng khác

Một bảng có thể có nhiều khoá nhưng chỉ có một khóa chính. Việc xác định khoá phụ thuộc vào quan hệ logic của các dữ liệu chứ không phụ thuộc vào giá trị các dữ liệu và nên chọn khoá chính là khoá có ít thuộc tính nhất.

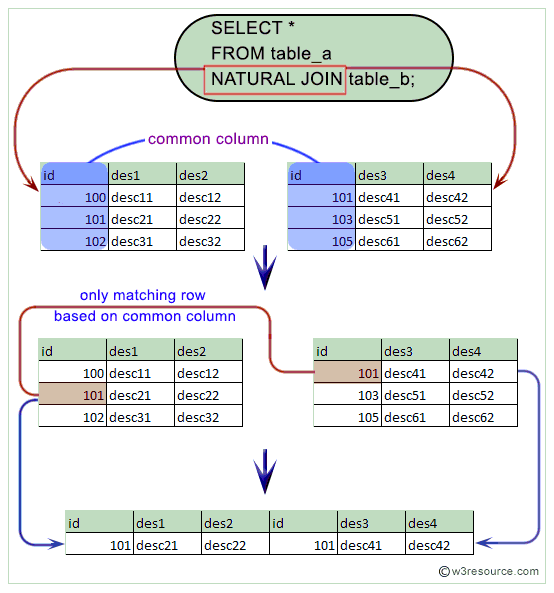
Các giá trị của khóa không thể nhận giá trị NULL hay các giá trị không xác định.

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

CROSS JOIN - Nối chéo dữ liệu các bảng với nhau. - Cú pháp: SELECT table1.column, table2.column FROM table1 CROSS JOIN table2;

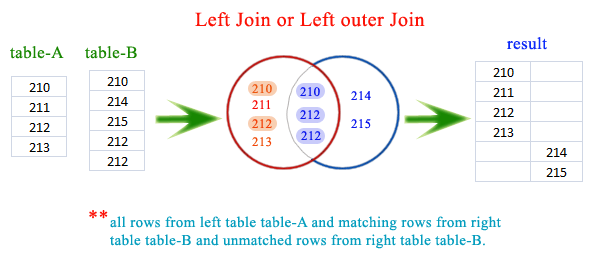


NATURAL JOIN - Phép nối tạo một mệnh đề tham gia ngầm dựa trên các cột chung trong hai bảng. Mặc định tham gia như inner - Cú pháp: SELECT table1.column, table2.column FROM table1 NATURAL JOIN table2;



LEFT OUTER JOIN - Nối hai bảng A(trái) và B(phải). Trả về tất cả giá trị bảng A và giá trị bảng B tương ứng. Nếu bảng B không có giá trị tương ứng sẽ trả về giá trị NULL. - Cú pháp:

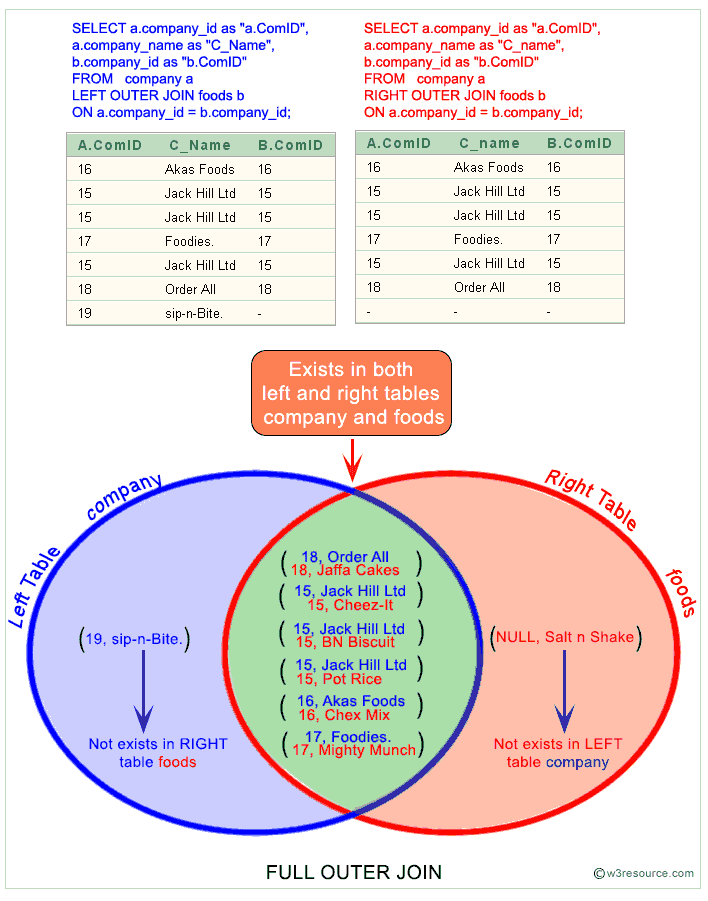
SELECT table1.column, table2.column FROM table1 LEFT OUTER JOIN table2 ON (table1.column = table2.column);



RIGHT OUTER JOIN - Nối hai bảng A(trái) và B(phải). Trả về tất cả giá trị bảng B và giá trị bảng A tương ứng. Nếu bảng A không có giá trị tương ứng sẽ trả về giá trị NULL. - Cú pháp: SELECT columns FROM table1 RIGHT [OUTER] JOIN table2 ON table1.column = table2.column



FULL OUTER JOIN - Nối dữ liệu 2 bảng A và B. Trả về tất cả dữ liệu từ hai bảng, trả về giá trị NULL ở vị trí điều kiện nối không đáp ứng. - Cú pháp: SELECT columns FROM table1 FULL [OUTER] JOIN table2 ON table1.column = table2.column;



**3. 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.

· UNOIN ALL: Kết hợp kết quả của nhiều câu hỏi với nhau, các mẫu tin trùng nhau cũng được lặp lại

UNION :

SELECT tên cột FROM bảng1 UNION SELECT tên cột FROM bảng2

Select job from emp where deptno = 10 Union Select job from emp where deptno = 30;

UNION ALL SELECT :

tên cột FROM bảng1 UNION ALL SELECT tên cột FROM bảng2

Select job from emp where deptno = 10 Union ALL Select job from emp where deptno = 30;

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

**Nested Query :**

Một truy vấn lồng nhau bao gồm hai hoặc nhiều truy vấn thông thường được lồng theo cách mà kết quả của mỗi truy vấn bên trong ( subselect ) được sử dụng trong kiểm tra so sánh cho mệnh đề lựa chọn của truy vấn bên ngoài tiếp theo (hoặc một câu lệnh khác).

Vd :

select ename from emp where job in(select job from emp where job like 'SALEMAN'

**Correlated Query :**

Trong truy vấn cơ sở dữ liệu SQL, truy vấn con tương quan là truy vấn con sử dụng các giá trị từ truy vấn bên ngoài. Bởi vì truy vấn con có thể được đánh giá một lần cho mỗi hàng được xử lý bởi truy vấn bên ngoài, nên nó có thể bị chậm

Ví dụ

select ename from emp where exists(select job from emp where job like 'SALEMAN')

**5. 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.

Cú pháp:

CREATE [UNIQUE]] INDEX index\_name

ON TABLE ( column [,column...]);

Global index - Chỉ mục toàn cục là mối quan hệ một-nhiều, cho phép một phân vùng chỉ mục ánh xạ tới nhiều phân vùng bảng. - Có thể được xác định trên bất kỳ loại phân vùng hoặc không phân vùng, bảng

Local index - Chỉ mục cục bộ 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:

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME); Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

Khi mà việc lấy dữ liệu <15% số row trong bảng.

- Index column nào dùng để nối giữa các bảng lẫn nhau.

- Không nên dung Index cho các bảng nào chỉ có vài row.

- Primaryvà unique key (khóa chính và khóa duy nhất) tự động có index, nhưng nên có index cho foreign key (khóa ngoại).

- Quá nhiều index sẽ gây ảnh hưởng đến số liệu khi mà dữ liệu trong table bị thay đổi thứ tự theo index

Ví dụ:

CREATE INDEX i-ENAME ON EMP (ENAME);

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

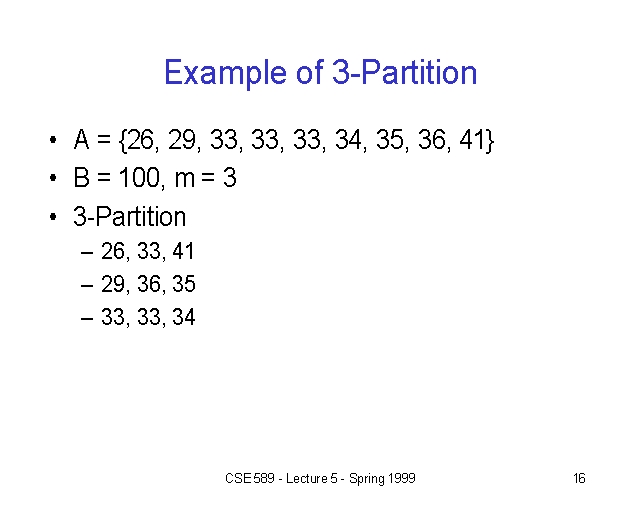
Khái niệm: là việc phân chia một table thành những phần nhỏ theo một logic nhất định, được phân biệt bằng key, key này thường là tên column trong table.

· Trường hợp tạo partition cho bảng:

- Bảng lớn hơn 2 GB luôn được coi là ứng viên cho phân vùng.

- Bảng chứa dữ liệu lịch sử, những dữ liệu mới sẽ được thêm vào phân vùng mới.

- Khi nội dung của bảng cần được phân phối trên nhiều loại thiết bị storage khác nhau.



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

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.

DELETE FROM table\_name WHERE condition;

TRUNCATE Xóa tất cả các hàng từ một bảng bằng cách phân bổ các trang bộ nhớ và không thể phục hồi lại.

TRUNCATE TABLE table\_name