1. **Index**
2. **Index là gì?**

Index (hay chỉ mục) là một cấu trúc dữ liệu để tăng hiệu suất truy vấn của cơ sở dữ liệu. Index cho phép cơ sở dữ liệu thực hiện một số câu truy vấn có điều kiện nhanh hơn so với thông thường. Nhưng index cũng được lưu trên bộ nhớ và tiêu tốn không gian bộ nhớ và thời gian để tạo, cập nhật index nên khi sử dụng index cần phải suy xét kĩ.

1. **Khi nào sử dụng index ?**

Index giúp tăng tốc độ truy vấn của một số câu lệnh select có điều kiện vì vậy nó đặc biệt hữu dụng khi câu lệnh truy vấn được sử dụng thường xuyên (hoặc cột được tạo index thường được truy vấn) và số lượng bản ghi lớn.

Khó để xác định khi nào sử dụng index, nó phụ thuộc nhiều vào các bài toán thực tế tuy nhiên có một số quy luật thường thấy khi chọn một cột (hoặc tập các cột) để tạo index:

+ Khóa và các cột có giá trị độc nhất (unique): Database thường sẽ tự động tạo index cho các cột này nên để tranh việc trùng lặp và tiêu tốn bộ nhớ ta không nên tạo thêm index cho chúng.

+ Tần suất được sử dụng: Khi tần suất sử dụng câu truy vấn càng lớn thì việc tạo index sẽ giúp làm giảm càng nhiều thời gian truy vấn (tính tổng).

+ Số lượng bản ghi của bảng: Số lượng bản ghi của bảng càng nhiều thì tốc độ truy vấn sẽ càng giảm lợi thế của việc sử dụng index trên các bảng này lại càng rõ ràng so với những bảng có số lượng bản ghi ít. Đặc biệt đối với trường hợp một bảng có ít bản ghi (100 - vài nghìn) ta không nên tạo chỉ mục cho chúng.

+ Dữ liệu của bảng tăng trưởng nhanh: Index sẽ tự động cập nhật khi có một bản ghi được thêm vào cơ sở dữ liệu, vì vậy khi đánh chỉ mục cho 1 bảng nó sẽ làm chậm lại các hành động thêm sửa xóa bản ghi. Vậy nên một bảng thường xuyên được cập nhật nên có ít index hơn một bảng hiếm khi cập nhật.

+ Không gian bộ nhớ: Khi tạo index sẽ sử dụng chính không gian bộ nhớ của cơ sở dữ liệu nên khi cơ sở dữ liệu có kích thước lớn ta cần lựa chọn cẩn thận trường nào sẽ sử dụng làm index.

+ Dữ liệu có đa dạng giá trị: Index được tạo dựa trên các giá trị trong cột mà nó trỏ tới ví dụ như cột index được tạo chỉ có 3 giá trị A, B, C thì index được tạo sẽ có giá trị nhỏ hơn nhiều so với cột có dải giá trị trải dài cả bảng chữ cái. Index trên cột có ít giá trị ví dụ cột sex sẽ không làm tăng nhiều tốc độ truy vấn tuy nhiên đối với những cột có nhiều giá trị riêng biệt như cột name sẽ làm tăng tốc độ truy vấn đáng kể.

1. **Ưu điểm của việc sử dụng index**

+ Thường sẽ làm tăng hiệu năng truy vấn khi điều kiện rơi vào các cột được đánh chỉ mục.

Giúp ta có thể truy vấn dữ liệu nhanh hơn.

1. **Nhược điểm của sử dụng index**

+ Làm giảm hiệu năng các câu lệnh insert, update ,delete.

+ Chiếm dụng bộ nhớ.

1. **Composite index**

+ Index cho 2 hay nhiều trường trở lên.

CREATE INDEX index\_name

ON table\_name(c1,c2,c3);

Query trong các trường hợp sau sẽ cho index:

(c1)

(c1,c2)

(c1,c2,c3).

+ Thực tế test thử: (c1,c3) vẫn cho index, ngay cả khi đổi thứ tự (c3, c1). (MySQL version 8)

+ Dùng OR không cho index. Query LIKE, ….

Các loại index? B-Tree vs Hash?

<https://viblo.asia/p/su-dung-index-trong-database-nhu-the-nao-cho-hieu-qua-4P856q69lY3>

1. CASE EXCLUSIVE

+ SELECT \* FROM parent WHERE NAME = 'Jones' FOR SHARE; (transaction A)

Lock các bản ghi có name = 'Jones', các transaction khác chỉ có thể đọc, không thể chỉnh sửa các bản ghi này cho đến khi transaction A kết thúc.

+ SELECT counter FROM child\_codes FOR UPDATE; (transaction B)

Đọc giá trị counter mới nhất và set exclusive lock trên mỗi row nó đọc. 🡪 Đảm bảo tại 1 thời điểm chỉ có 1 transaction có thể đọc được giá trị của bản ghi. Khác với SELECT … FOR SHARE, nhiều transaction có thể đọc được bản ghi tại 1 thời điểm.