# **Clustered Index và Non-clustered Index**

1. **Clustered Index**

* Là chỉ mục sắp xếp lại dữ liệu thực tế trong bảng theo thứ tự của cột được chỉ mục. Vì dữ liệu trong bảng thực sự được sắp xếp theo chỉ mục này, một bảng chỉ có thể có 1 clustered index.
* Nếu có một clustered index trên cột ID, dữ liệu trong bảng sẽ được lưu trữ theo thứ tự tăng dần của ID.
* Hiệu năng:
* Tìm kiếm nhanh hơn cho các truy vấn cần dữ liệu theo thứ tự của chỉ mục vì dữ liệu thực sự được sắp xếp.
* Insert, Update, Delete chậm hơn nếu phải thêm hoặc thay đổi dữ liệu vì bảng cần phải tái sắp xếp lại để duy trì thứ tự chỉ mục.
* Lý do: Clustered index sắp xếp dữ liệu thực tế theo chỉ mục, giúp truy vấn tìm kiếm nhanh hơn vì dữ liệu đã sẵn sàng theo đúng thứ tự yêu cầu. Tuy nhiên, vì phải sắp xếp lại dữ liệu thực tế trong bảng khi có thay đổi, nên các thao tác ghi sẽ bị ảnh hưởng.

1. **Non-clustered Index**

* Là một chỉ mục chứa bản sao của các giá trị của cột được chỉ mục cùng với một con trỏ (pointer) trỏ tới vị trí thực tế của hàng trong bảng. Nó không sắp xếp lại dữ liệu trong bảng mà chỉ tạo ra một cấu trúc bổ sung để giúp tăng tốc truy vấn.
* Một bảng có thể có nhiều non-clustered index.
* Hiệu năng:
* Tìm kiếm không nhanh bằng clustered index do phải sử dụng con trỏ để trỏ tới dữ liệu thực.
* Insert, Update ít bị ảnh hưởng vì không cần tái sắp xếp dữ liệu, nhưng việc thêm nhiều chỉ mục có thể làm giảm hiệu suất của các thao tác ghi (insert, update, delete).
* Lý do: Non-clustered index chỉ lưu trữ các con trỏ tới dữ liệu, giúp tăng tốc độ truy vấn nhưng không thay đổi cấu trúc của bảng.

# Giới hạn việc đánh bao nhiêu index trong một bảng

* Không có giới hạn cứng về số lượng index có thể tạo trên một bảng, nhưng SQL Server giới hạn một bảng có thể có tối đa 999 non-clustered index và một clustered index.
* Lý do: Vì việc có quá nhiều index sẽ làm giảm hiệu suất của các thao tác ghi dữ liệu (insert, update, delete), do mỗi khi dữ liệu thay đổi, tất cả các index liên quan cũng phải được cập nhật.

# Việc đánh index dựa trên cơ sở nào

* **Tần suất truy vấn**: Những cột thường xuyên xuất hiện trong các điều kiện tìm kiếm (WHERE, JOIN, GROUP BY) hoặc sắp xếp (ORDER BY) nên được đánh index.
* **Kích thước bảng**: Với những bảng lớn, index giúp cải thiện đáng kể hiệu suất truy vấn. Tuy nhiên, đối với bảng nhỏ, việc sử dụng index có thể không cần thiết vì chi phí xử lý index có thể lớn hơn lợi ích thu được.
* **Độ phân tán của dữ liệu**: Cột có độ phân tán cao (nhiều giá trị khác nhau) là ứng cử viên tốt cho việc đánh index, vì nó giúp giảm đáng kể số lượng hàng cần kiểm tra.

# Nhận biết một câu query đã sử dụng index hay chưa

* Sử dụng EXPLAIN (trong MySQL) hoặc EXPLAIN QUERY PLAN (trong SQLite) để phân tích cách mà hệ quản trị cơ sở dữ liệu xử lý truy vấn.
* Trong SQL Server, có thể sử dụng Execution Plan. Kế hoạch thực thi sẽ hiển thị chi tiết từng bước trong quá trình thực hiện câu truy vấn và chỉ ra những index nào đã được sử dụng.