* Link Github : <https://github.com/nguyenducthau1234/TK-AND-QT-CSDL>

1. **Cài đặt và restore**

* ***Cài đặt SQL Server :***

Graphical user interface

Description automatically generated

* ***Restore DB :***

Graphical user interface, text

Description automatically generated

1. **Thực hiện câu lệnh và so sánh**
2. **Sử dụng Index và không Index trên một bảng**

* ***NoIndex :***

Graphical user interface, text, application, email

Description automatically generated

* ***Index :***

A picture containing text

Description automatically generated

1. **Các bước thực hiện truy vấn**

* ***NoIndex :*** Hệ thống sẽ thực thi câu lệnh bằng cách duyệt qua cả bảng (table scan) và tìm ra các bản ghi thỏa mãn yêu cầu tìm kiếm. Thao tác duyệt bảng có nghĩa là hệ thống cần phải đọc tuần tự từng bản ghi từ đầu đến cuối để tìm ra kết quả. Trong trường hợp này, nó phải đọc toàn bộ 606 bản ghi và tìm ra bản ghi có ProductKey = 600. Đây là một thao tác rất chậm vì nó phải xử lý tất cả các bản ghi trong bảng. Nên nhớ hệ thống sẽ không dừng lại khi nó tìm được bản ghi đầu tiên có ProductKey = 600, vì nó không biết liệu còn bản ghi nào khác có giá trị ProductKey tương tự hay không, cho nên để chắc chắn trả lại kết quả đầy đủ hệ thống vẫn phải tiếp tục đọc các bản ghi còn lại.
* ***Index :*** Lần này không thấy thao tác table scan nữa, mà thay vào đó là index seek và RID lookup. Index seek là khi hệ thống có thể nhảy đến được node trên cây index chứa khóa thỏa mãn yêu cầu tìm kiếm. Index là một cấu trúc dữ liệu có dạng B-tree, nên nó rất thích hợp với các thao tác tìm kiếm theo kiểu key=value, chỉ cần vài phép so sánh là hệ thống định vị được node chứa khóa cần tìm. Node này chứa khóa (trường được index, ở đây là giá trị của ProductKey) và RID là ID của bản ghi tương ứng trong bảng (đây là giá trị nội bộ chỉ dùng bên trong hệ thống, ta không truy cập được giá trị này). Vì thế bước tiếp theo là dùng RID này để nhảy đến bản ghi tương ứng trong bảng (RID lookup) để lấy các trường dữ liệu cần thiết.

1. **Tiêu chí xem hiệu năng của câu truy vấn**

* ***Query cost***
* Đối với NoIndex thì hiệu năng của câu truy vấn này là 97% :

A picture containing waterfall chart

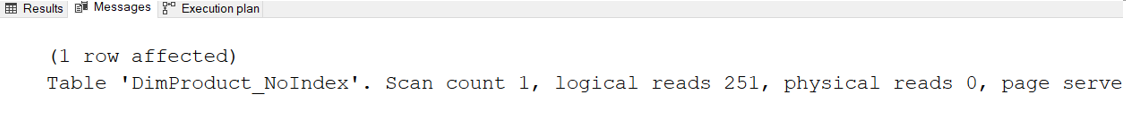
Description automatically generated

* Đối với Index thì chi phí của câu truy vấn này là 3% :

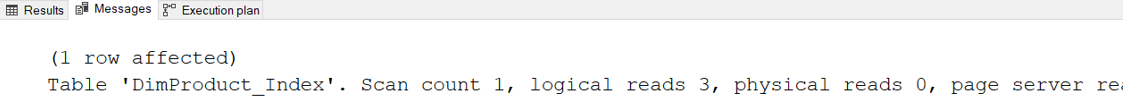
Diagram

Description automatically generated

* Hiệu năng thực hiện truy vấn của NoIndex tốn hiệu năng hơn Index.
* ***Page Reads***
* Đối với NoIndex kết quả trả về :

******

* Theo như kết quả trả về thì số trang đọc được ở dữ liệu cache là 251.
* Đối với Index kết quả trả về :



* Theo như kết quả trả về thì số trang đọc được ở dữ liệu cache là 3.
* Qua đó kết luận tốc độ đọc trang dữ liệu của Index nhanh hơn NoIndex.
* ***Query Execution Time***
* Đối với NoIndex kết quả trả về :Text, letter

  Description automatically generated
* Theo như kết quả trả về thì thời gian biên dịch và phân tích cú pháp SQL Server là 1 ms và thời gian thực thi SQL Server là 1 ms.
* Đối với Index kết quả trả về :

Text, letter

Description automatically generated

* Theo như kết quả trả về thì thời gian biên dịch và phân tích cú pháp SQL Server là 1 ms và thời gian thực thi SQL Server là 35 ms.

1. **Sử dụng Clustered index và no-Clustered index**

Graphical user interface, text, application, email

Description automatically generated

1. **Đánh giá tiêu chí xem xét hiệu năng câu truy vấn**

* ***Query cost***
* Đối với Clustered thì chi phí của câu truy vấn này là 50% :

***Graphical user interface, text, application

Description automatically generated***

* Đối với Non-Clustered thì chi phí của câu truy vấn này là 50% :

**Graphical user interface, text, application

Description automatically generated**

* Hiệu năng thực hiện truy vấn của Clustered bằng với hiệu năng thực hiện truy vấn của Non – Clustered