Họ và tên: Nguyễn Minh Đức

MSSV: 20187162

Lớp: IT-GINP 17

Github: https://github.com/NguyenMinhDucIT/Tk-QT-CSDL

1. Chụp màn hình cài đặt thành công SQL Server, restore DB

Graphical user interface, table

Description automatically generated

1. Chạy so sánh kết quả thực hiện với các trường hợp:
   1. **Không sử dụng Index và sử dụng Index trên 1 bảng**

* Trường hợp không index:

Graphical user interface, text, application, email

Description automatically generated

* Trường hợp có index:

A picture containing timeline

Description automatically generated

1. **Giải thích các bước xử lý câu truy vấn**

* **No\_index:** 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. **Các tiêu chí xem xét hiệu năng của câu truy vấn**

* Query cost.
* Đối với NoIndex thì chi phí của câu truy vấn này là 97%.

Graphical user interface, text, application

Description automatically generated

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

Diagram

Description automatically generated

* Chí phí thực hiện truy vấn của NoIndex tốn hơn chi phí thực hiện truy vấn của Index
* Page Reads
* Đối với NoIndex thì kết quả trả về

Graphical user interface, text, application, email

Description automatically generated

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

Graphical user interface, text, application, email

Description automatically generated

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

Text

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 thì kết quả trả về:

Graphical user interface, text

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%

***Text

Description automatically generated***

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

Graphical user interface, text, application, email

Description automatically generated

* Chí phí thực hiện truy vấn của Clustered bằng với chi phí thực hiện truy vấn của Non – Clustered