

Nhóm: **18VP-CSDLNC-04**

Thành Viên:

- |             |                  |
|-------------|------------------|
| 1) 18126023 | Đặng Diễm Linh   |
| 2) 18126037 | Đinh Thành Việt  |
| 3) 18126038 | Dương Quang Vinh |

## **CƠ SỞ DỮ LIỆU NÂNG CAO**

### **BÁO CÁO ĐỒ ÁN 1**

## **ĐỀ TÀI: QUẢN LÝ BÁN HÀNG**

# MỤC LỤC

MỤC LỤC .....	2
CÀI ĐẶT TRIGGER.....	3
TRUY VẤN VÀ ĐỀ XUẤT CHỈ MỤC .....	6
LẬP TRÌNH GIAO DIỆN .....	16

# CÀI ĐẶT TRIGGER

## 1. Trigger trên ThanhTien của bảng CT\_HoaDon

**Yêu cầu:** ThanhTien = SoLuong\*(GiaBan – GiaGiam)

**Code SQL:**

```
IF OBJECT_ID(N'CTHD_ThanhTien') IS NOT NULL
    DROP TRIGGER CTHD_ThanhTien
GO

CREATE TRIGGER CTHD_ThanhTien ON CT_HoaDon FOR INSERT, UPDATE
AS
    IF (UPDATE(SoLuong) OR UPDATE(GiaBan) OR UPDATE(GiaGiam) OR UPDATE(ThanhTien))
    BEGIN
        UPDATE CT_HoaDon SET ThanhTien = SoLuong*(GiaBan - GiaGiam) WHERE EXISTS
        (SELECT * FROM inserted I WHERE i.MaHD = CT_HoaDon.MaHD AND i.MaSP = CT_HoaDon.MaSP)
    END
GO
```

## 2. Trigger trên TongTien của bảng HoaDon

**Yêu cầu:** TongTien = tổng thành tiền của mã hóa đơn tương ứng.

**Code SQL:**

```
IF OBJECT_ID(N'HoaDon_TongTien') IS NOT NULL
    DROP TRIGGER HoaDon_TongTien
GO

CREATE TRIGGER HoaDon_TongTien ON CT_HoaDon AFTER INSERT, UPDATE, DELETE
AS
    IF (UPDATE(SoLuong) OR UPDATE(GiaBan) OR UPDATE(GiaGiam) OR UPDATE(ThanhTien))
    BEGIN
        UPDATE HoaDon SET TongTien = (SELECT SUM(ThanhTien) FROM CT_HoaDon C WHERE
        C.MaHD = I.MaHD ) FROM HoaDon H JOIN inserted I ON H.MaHD = I.MaHD
    END
GO
```

### 3. Trigger trên NgayLap của bảng HoaDon

**Yêu cầu:** NgayLap của HoaDon phải sau NgaySinh của KhachHang

**Code SQL:**

```
IF OBJECT_ID(N'HoaDon_NgayLap') IS NOT NULL
    DROP TRIGGER HoaDon_NgayLap
GO

CREATE TRIGGER HoaDon_NgayLap ON HoaDon FOR INSERT, UPDATE
AS
    IF UPDATE(NgayLap)
    BEGIN
        IF EXISTS (SELECT * FROM inserted I JOIN KhachHang K ON I.MaKH = K.MaKH
WHERE I.NgayLap < K.NgSinh)
        BEGIN
            RAISERROR (N'Ngay lap hoa don phai sau ngay sinh cua khách hàng',
16, 1)
            ROLLBACK TRANSACTION
        END
    END
GO
```

### 4. Trigger trên SoLuong của bảng CT\_HoaDon

**Yêu cầu:** số lượng của sản phẩm khi lập hoặc cập nhật CT Hóa Đơn không được vượt quá Số lượng tồn hiện tại của sản phẩm đó trong bảng SanPham.

**Code SQL:**

```
IF OBJECT_ID(N'CT_HoaDon_SoLuong') IS NOT NULL
    DROP TRIGGER CT_HoaDon_SoLuong
GO

CREATE TRIGGER CT_HoaDon_SoLuong ON CT_HoaDon FOR INSERT, UPDATE
AS
    IF UPDATE(SoLuong)
    BEGIN
        DECLARE @SoLuongDat INT
        DECLARE @MaSP INT
```

```

        IF EXISTS (SELECT * FROM deleted D JOIN inserted I ON D.MaHD = I.MaHD WHERE
D.MaHD = I.MaHD)
        BEGIN
            SELECT @SoLuongDat = I.SoLuong - D.SoLuong, @MaSP = I.MaSP FROM
deleted D JOIN inserted I ON D.MaHD = I.MaHD
        END

        ELSE
        BEGIN
            SELECT @SoLuongDat = I.SoLuong, @MaSP = I.MaSP FROM inserted I
        END

        IF @SoLuongDat <= 0
        BEGIN
            UPDATE SanPham SET SoLuongTon = SoLuongTon - @SoLuongDat WHERE MaSP = @MaSP
            RETURN
        END

        ELSE
        IF @SoLuongDat > (SELECT SoLuongTon FROM SanPham WHERE MaSP = @MaSP)
        BEGIN
            RAISERROR (N'So luong dat vuot qua so luong hang ton hien tai cua san
pham', 16, 1)
            ROLLBACK TRANSACTION
        END

        ELSE
        BEGIN
            UPDATE SanPham SET SoLuongTon = SoLuongTon - @SoLuongDat WHERE MaSP = @MaSP
            RETURN
        END

    END

GO

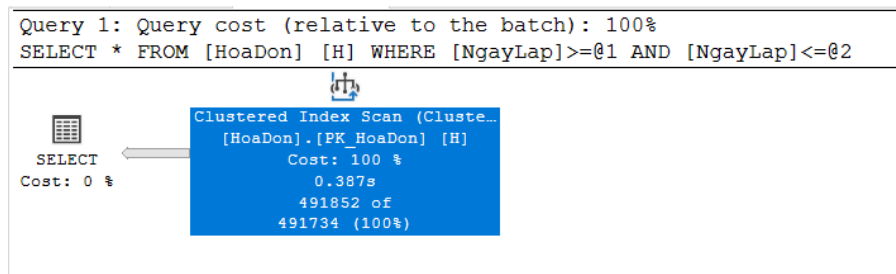
```

# TRUY VẤN VÀ ĐỀ XUẤT CHỈ MỤC

## 1) Danh sách các hóa đơn trong năm 2020

```
SELECT * FROM HoaDon H WHERE NgayLap BETWEEN '2020-01-01' AND '2020-12-31'
```

### Execution Plan:



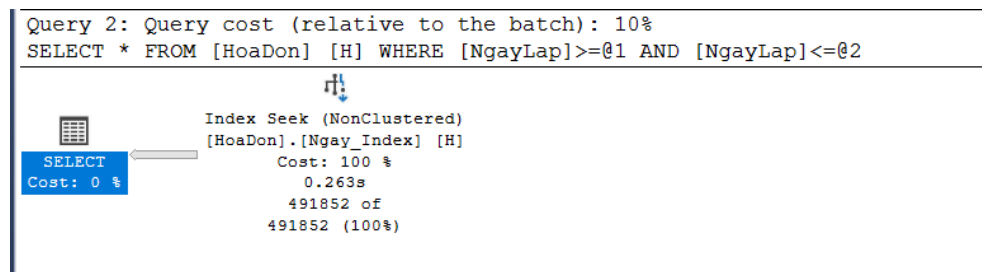
Hệ quản trị sử dụng Index Scan trên khóa chính MaHD của bảng HoaDon để chọn các dòng có NgayLap thỏa điều kiện có năm là 2020. Vì sử dụng Index Scan, hệ quản trị sẽ duyệt tất cả các dòng trong bảng, do đó, chi phí của truy vấn là 100% với thời gian là 0.387 giây.

### ⇒ ĐỀ XUẤT CHỈ MỤC CHO CÂU 2.1):

Tạo chỉ mục theo cột NgayLap kết hợp với MaKH và TongTien:

```
CREATE NONCLUSTERED INDEX Ngay_Index ON HoaDon(NgayLap) INCLUDE (MaKH, TongTien)
```

### Execution Plan:



Sau khi cài đặt chỉ mục tại NgayLap, hệ quản trị đối với cùng câu truy vấn đã có cải tiến về hiệu quả truy vấn. Thời gian truy vấn là 0.263 giây, nhanh hơn so với sử dụng index là khóa chính như trường hợp trên.

Nguyên nhân dẫn đến cải tiến này là do hệ quản trị sử dụng Index Seek dựa trên NgayLap, vì vậy giảm bớt số lượt đọc dữ liệu.

Đồng thời, vì số lượng ngày khác nhau quá nhiều, nhóm đề xuất kết hợp thêm hai thuộc tính MaKH và TongTien của HoaDon trong bảng Index để hạn chế đọc lại database sau khi truy vấn trên Index NgayLap.

### Kết quả:

#### Trước khi sử dụng chỉ mục

Number of Rows Read	500000
Actual Number of Rows for All Executions	491852
Actual Number of Batches	0
Estimated I/O Cost	1.28683
Estimated Operator Cost	1.83699 (100%)
Estimated CPU Cost	0.550157
Estimated Subtree Cost	1.83699
Number of Executions	1

#### Sau khi cài đặt chỉ mục Ngay\_Index

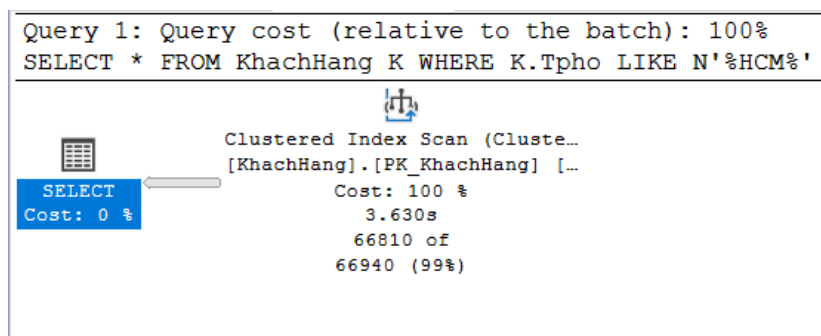
Number of Rows Read	491852
Actual Number of Rows for All Executions	491852
Actual Number of Batches	0
Estimated I/O Cost	1.13424
Estimated Operator Cost	1.67543 (100%)
Estimated CPU Cost	0.541194
Estimated Subtree Cost	1.67543
Estimated Number of Executions	1
Number of Executions	1

Nhìn chung, kết quả của cài đặt chỉ mục đã giúp truy vấn đã giảm các chi phí chung cũng như giảm số lượng dòng cần đọc để được kết quả trên cùng một số lượng thực thi.

## 2) Danh sách khách hàng sống tại TPHCM

```
SELECT * FROM KháchHang K WHERE K.Tpho LIKE N'%HCM%'
```

### Execution Plan:

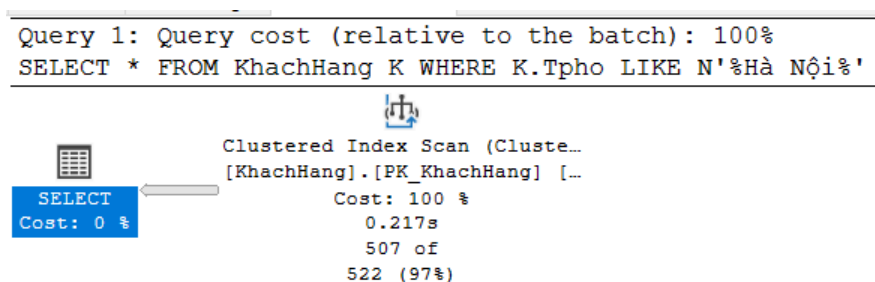


Hệ quản trị sử dụng Index Scan trên khóa chính MaKH của bảng KháchHang để chọn các dòng có Tpho thỏa điều kiện chứa chuỗi HCM. Vì sử dụng Index Scan, hệ quản trị sẽ duyệt tất cả các dòng trong bảng, do đó, chi phí của truy vấn là 100% với thời gian là 3.63 giây.

### 3) Danh sách khách hàng sống tại Hà Nội

```
SELECT * FROM KháchHang K WHERE K.Tpho LIKE N'%Hà Nội%'
```

#### Execution Plan:



Hệ quản trị sử dụng Index Scan trên khóa chính MaKH của bảng KháchHang để chọn các dòng có Tpho thỏa điều kiện chứa chuỗi 'Hà Nội'. Vì sử dụng Index Scan, hệ quản trị sẽ duyệt tất cả các dòng trong bảng, do đó, chi phí của truy vấn là 100% với thời gian là 0.217 giây.

#### ⇒ ĐỀ XUẤT CHỈ MỤC 2.2 VÀ 2.3:

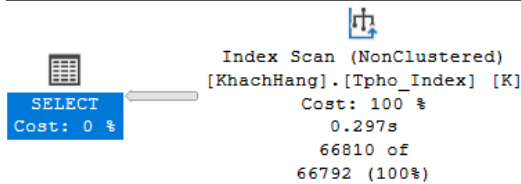
```
CREATE NONCLUSTERED INDEX Tpho_Index ON KháchHang(Tpho) INCLUDE (Ho, Ten, NgSinh, SoNha, Duong, Phuong, Quan, DienThoai)
```

Bảng chỉ mục lúc này sẽ sắp xếp chuỗi theo Tpho của KháchHang, đồng thời kết hợp thêm giá trị các cột khác để hạn chế việc đọc lại Database. Do đó, thời gian so sánh chuỗi và truy xuất trên chỉ mục sẽ được cải thiện đáng kể.

- Đối với truy vấn câu 2.2: thời gian truy vấn chỉ còn 0.297 giây, thấp hơn nhiều so với 3.63 giây khi chưa sử dụng chỉ mục.

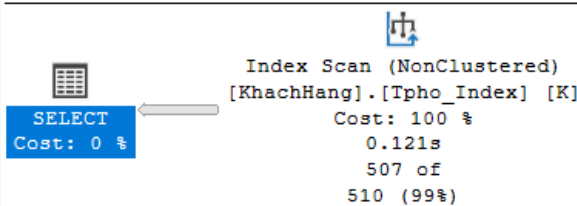


Query 1: Query cost (relative to the batch): 100%  
 SELECT \* FROM KhachHang K WHERE K.Tpho LIKE N'%HCM%'



- Đối với truy vấn câu 2.3: thời gian truy vấn chỉ còn 0.121 giây, thấp hơn so với 0.217 giây khi chưa sử dụng chỉ mục.

Query 1: Query cost (relative to the batch): 100%  
 SELECT \* FROM KhachHang K WHERE K.Tpho LIKE N'%Hà Nội%'

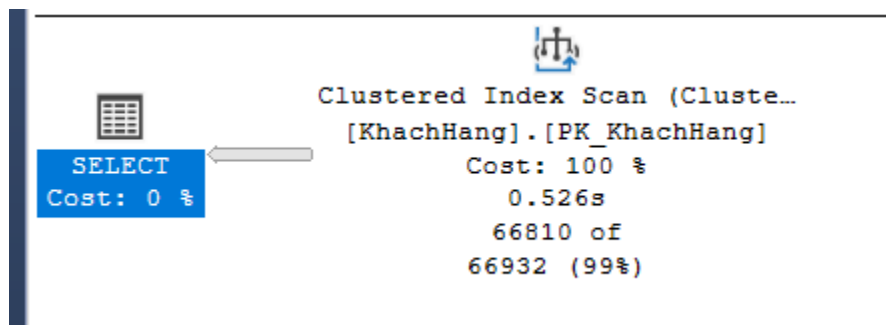


## ⇒ ĐỀ XUẤT THÊM

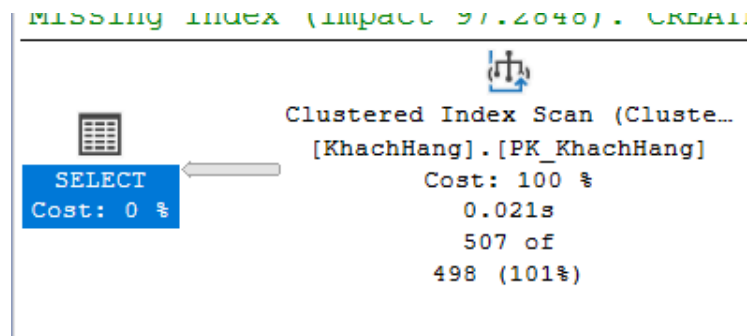
Vì Tpho của bảng KhachHang có dữ liệu là chuỗi, vì vậy việc so sánh và sắp xếp chuỗi sẽ tốn nhiều thời gian hơn so với trên dữ liệu số. Do đó, nhóm đề xuất sẽ chuẩn hóa Tpho thành dữ liệu số, tương ứng với mã của 63 tỉnh thành tại Việt Nam.

*Kết quả sau khi chuẩn hóa cột Tpho thành Ma\_Tpho:*

- ⇒ Danh sách khách hàng sống tại TP.HCM:



- ⇒ Danh sách khách hàng sống tại Hà Nội:

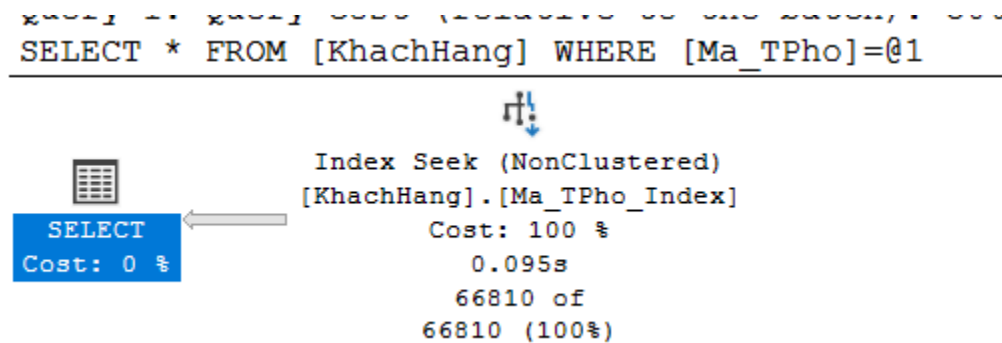


Có thể thấy, thời gian truy vấn bình thường khi không có chỉ mục cũng thấp hơn so với cách cài đặt cũ. Tuy nhiên, số lượng dòng cần đọc của cả hai truy vấn vẫn là toàn bộ bảng vì hệ quản trị sử dụng Cluster Index Scan, tương ứng với khóa chính **MaKH** của **KhachHang**.

### ⇒ ĐỀ XUẤT CHỈ MỤC:

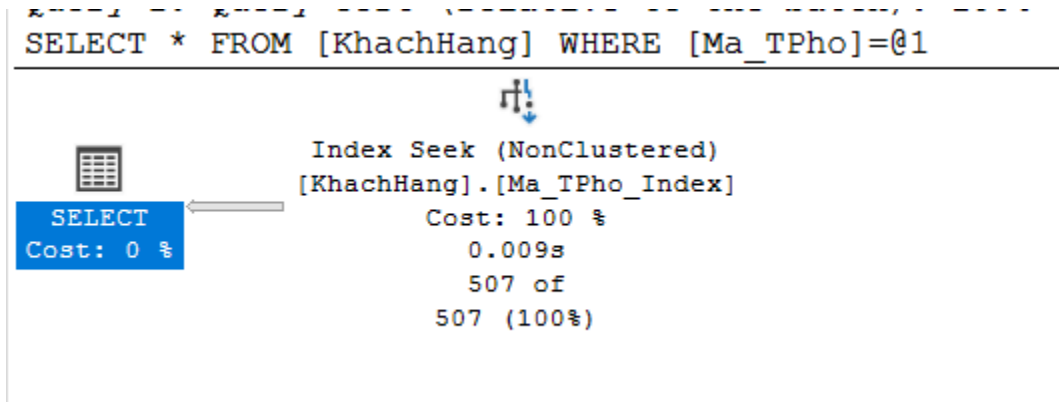
`CREATE NONCLUSTERED INDEX Ma_TPho_Index ON KhachHang(Ma_TPho) INCLUDE (Ho, Ten, NgSinh, SoNha, Duong, Phuong, Quan, DienThoai)`

### ⇒ Kết quả truy vấn khách hàng sống tại TP.HCM:



Vì sử dụng Index Seek trên Index là MaTP, hệ quản trị rất nhanh chóng tìm được mã TP tương ứng với TP.HCM, do đó thời gian truy vấn giảm đi nhiều so với chưa cài đặt chỉ mục, đồng thời, số lượng dòng cần phải đọc của hệ quản trị cũng chỉ còn 66810 dòng, tương ứng với đúng số lượng dòng trong bảng kết quả.

### ⇒ Kết quả truy vấn khách hàng sống tại Hà Nội:

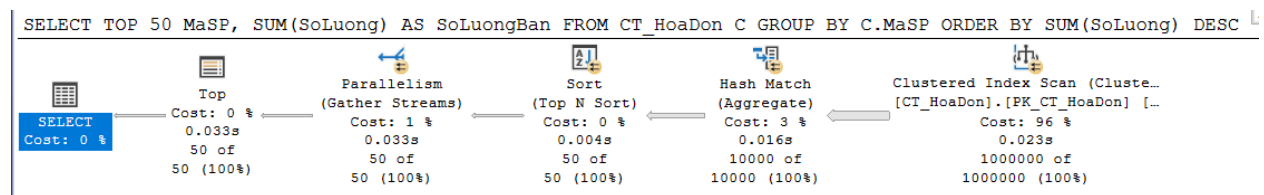


Vì sử dụng Index Seek trên Index là MaTP, hệ quản trị rất nhanh chóng tìm được mã TP tương ứng với Hà Nội, do đó thời gian truy vấn giảm đi nhiều so với chưa cài đặt chỉ mục, đồng thời, số lượng dòng cần phải đọc của hệ quản trị cũng chỉ còn 507 dòng, tương ứng với đúng số lượng dòng trong bảng kết quả.

#### 4) Danh sách sản phẩm bán chạy nhất

SELECT TOP 50 MaSP, SUM(SoLuong) AS SoLuongBan FROM CT\_HoaDon C GROUP BY C.MaSP ORDER BY SUM(SoLuong) DESC

##### Execution Plan:



Đi từ phải sang trái, hệ quản trị lần lượt thực thi lệnh truy vấn như sau:

- 1) Sử dụng Index Scan trên khóa chính của bảng CT\_HoaDon và duyệt tất cả các dòng, sau đó trả ra Output là danh sách gồm hai cột MaSP và SoLuong của CT\_HoaDon.

Node ID	Object	Output List
4	[QuanLyBanHang].[dbo].[CT_HoaDon].[PK_CT_HoaDon] [C]	[QuanLyBanHang].[dbo].[CT_HoaDon].MaSP, [QuanLyBanHang].[dbo].[CT_HoaDon].SoLuong

- 2) Tại bước Hash Match, các dòng sẽ được gom nhóm theo MaSP và tính toán tổng SoLuong của từng MaSP thông qua **Expr1001 = SUM(CT\_HoaDon.SoLuong as C.SoLuong)**

Output List	
[QuanLyBanHang].[dbo].[CT_HoaDon].MaSP, Expr1001	

- 3) SORT: Hệ quản trị chọn sắp xếp theo TOP N Sort, nghĩa là chỉ sắp xếp 50 dòng có giá trị SUM của SoLuong là lớn nhất (*vì chọn thứ tự Desc*). Do đó, hệ quản trị chỉ làm việc trên 50 dòng.

Estimated Execution Mode	Batch
Actual Number of Rows for All Executions	50
Actual Number of Batches	1
Estimated Operator Cost	0.01604 (0%)
Estimated I/O Cost	0.0009384
Estimated CPU Cost	0.015108

Quá trình này cho ra OUTPUT là bảng có sắp xếp giảm dần tại 50 dòng đầu tiên với tổng SoLuong bán là cao nhất.

Node ID	
Output List	
[QuanLyBanHang].[dbo].[CT_HoaDon].MaSP, Expr1001	
Order By	
Expr1001 Descending	

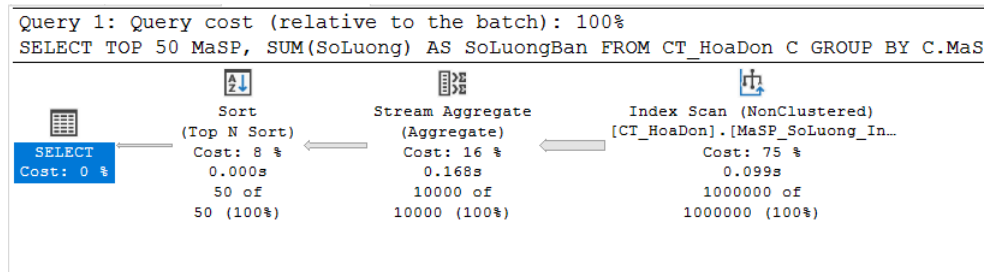
- 4) Cuối cùng, tại chuỗi Gather Streams => TOP => Select, top 50 dòng đầu tiên của bảng CT\_HoaDon sau sắp xếp, ứng với 50 mã sản phẩm có số lượng bán được cao nhất, và xuất ra kết quả cuối cùng.

Node ID	
Output List	
[QuanLyBanHang].[dbo].[CT_HoaDon].MaSP, Expr1001	
Top Expression	
(50)	

## ⇒ ĐỀ XUẤT CHỈ MỤC CHO CÂU 2.4:

```
CREATE NONCLUSTERED INDEX MaSP_SoLuong_Index ON CT_HoaDon(MaSP) INCLUDE (SoLuong)
```

### Execution Plan:



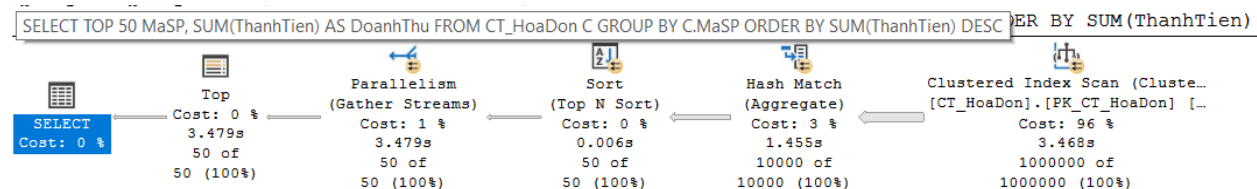
Có thể thấy, sau khi sử dụng chỉ mục, số lượng thao tác trong truy vấn đã giảm xuống. Điều này đến từ việc hệ quản trị đã thay thế Hash Match bằng Stream Aggregate. Do đó, lệnh truy vấn này *không cần thao tác gom nhóm* mà chỉ cần tính tổng SoLuong như yêu cầu.

Cùng với đó, vì cuối cùng, danh sách được sắp xếp có sẵn trong bảng chỉ mục, nên không cần phải chia luồng khi sắp xếp và vì vậy, kết quả sau sắp xếp sẽ được truy vấn thẳng mà không cần tập hợp lại, do đó, thao tác TOP cũng không cần được sử dụng.

## 5) Danh sách sản phẩm có doanh thu cao nhất

```
SELECT TOP 50 MaSP, SUM(SoLuong) AS SoLuongBan FROM CT_HoaDon C GROUP BY C.MaSP ORDER BY SUM(ThanhTien) DESC
```

### Execution Plan:



Đi từ phải sang trái, hệ quản trị lần lượt thực thi lệnh truy vấn như sau:

- 1) Sử dụng Index Scan trên khóa chính của bảng CT\_HoaDon và duyệt tất cả các dòng, sau đó trả ra Output là danh sách gồm hai cột MaSP và ThanhTien của CT\_HoaDon.

<b>Node ID</b>	4
<b>Object</b>	[QuanLyBanHang].[dbo].[CT_HoaDon].[PK_CT_HoaDon] [C]
<b>Output List</b>	[QuanLyBanHang].[dbo].[CT_HoaDon].MaSP, [QuanLyBanHang].[dbo].[CT_HoaDon].ThanhTien

- 2) Tại bước Hash Match, các dòng sẽ được gom nhóm theo MaSP và tính toán tổng ThanhTien của từng MaSP thông qua **Expr1001 = SUM(CT\_HoaDon.ThanhTien as C.ThanhTien)**

<b>Actual Rowids</b>	0
<b>Node ID</b>	3
<b>Output List</b>	[QuanLyBanHang].[dbo].[CT_HoaDon].MaSP, Expr1001

- 3) SORT: Hệ quản trị chọn sắp xếp theo TOP N Sort, nghĩa là chỉ sắp xếp 50 dòng có giá trị SUM của ThanhTien lớn nhất (*vì chọn thứ tự Desc*). Do đó, hệ quản trị chỉ làm việc trên 50 dòng.

<b>Actual Number of Rows for All Executions</b>	50
<b>Actual Number of Batches</b>	1
<b>Estimated Operator Cost</b>	0.01604 (0%)
<b>Estimated I/O Cost</b>	0.0009384

Quá trình này cho ra OUTPUT là bảng có sắp xếp giảm dần tại 50 dòng đầu tiên với tổng ThanhTien bán là cao nhất.

<b>Output List</b>
[QuanLyBanHang].[dbo].[CT_HoaDon].MaSP, Expr1001
<b>Order By</b>
Expr1001 Descending

- 4) Cuối cùng, tại chuỗi Gather Streams => TOP => Select, top 50 dòng đầu tiên của bảng CT\_HoaDon sau sắp xếp, ứng với 50 mã sản phẩm có doanh thu cao nhất, và xuất ra kết quả cuối cùng.

### Output List

[QuanLyBanHang].[dbo].[CT\_HoaDon].MaSP, Expr1001

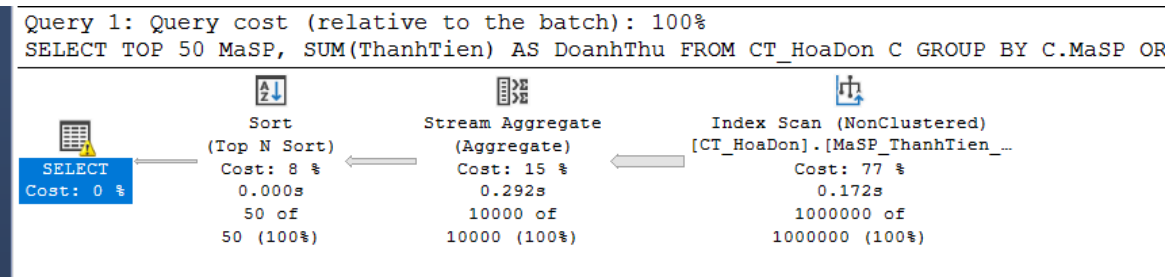
### Top Expression

(50)

⇒ ĐỀ XUẤT CHỈ MỤC CHO CÂU 2.5:

```
CREATE NONCLUSTERED INDEX MaSP_ThanhTien_Index ON CT_HoaDon(MaSP) INCLUDE (ThanhTien)
```

### Execution Plan:



Có thể thấy, sau khi sử dụng chỉ mục, số lượng thao tác trong truy vấn đã giảm xuống. Điều này đến từ việc hệ quản trị đã thay thế Hash Match bằng Stream Aggregate. Do đó, lệnh truy vấn này *không cần thao tác gom nhóm* mà chỉ cần tính tổng ThanhTien như yêu cầu.

Cùng với đó, vì cuối cùng, danh sách được sắp xếp có sẵn trong bảng chỉ mục, nên không cần phải chia luồng khi sắp xếp và vì vậy, kết quả sau sắp xếp sẽ được truy vấn thẳng mà không cần tập hợp lại, do đó, thao tác TOP cũng không cần được sử dụng.

# LẬP TRÌNH GIAO DIỆN

## 1. Giao diện Xem Danh sách Hóa Đơn

Khu vực chỉnh sửa hóa đơn

### Danh sách hóa đơn

Mã hóa đơn

Thời gian

Mã khách hàng

Tổng tiền

Thêm hóa đơn

Chỉnh sửa

Xóa

Hủy bỏ

Lưu

Tìm kiếm theo khoảng thời gian

Mã khách hàng

Từ ngày

Tới ngày

Tìm kiếm

Tìm kiếm theo mã khách hàng

Danh sách hóa đơn

Thống kê doanh thu

MaHD	MaKH	NgayLap	TongTien
1	99957	2020-01-05	75466936
369	59818	2020-01-05	1613246957
737	2921	2020-01-05	338830809
1105	42161	2020-01-05	-1671574099
1473	31244	2020-01-05	-1255731198
1841	95073	2020-01-05	437509917
2209	7968	2020-01-05	-1393676244
2577	96446	2020-01-05	-222666385
2945	96059	2020-01-05	492524196
3313	58263	2020-01-05	-8270213
3681	520	2020-01-05	-1381813941
4049	91057	2020-01-05	1888032945
4417	91693	2020-01-05	-1873314010
4785	87456	2020-01-05	1410189608
5153	7014	2020-01-05	-1477975323
5521	46518	2020-01-05	1152788474
5889	98879	2020-01-05	-453394807

<

1

Tiến

>

## 2. Giao diện Thêm Hóa Đơn





Các thông tin về sản phẩm sẽ được cập nhật tự động khi nhấp chọn Sản phẩm:

Số lượng sản phẩm  Tên sản phẩm  Giá tiền  Mã

Tên sản phẩm  Giá thấp nhất  Giá cao nhất

MaSP	TenSP	SoLuongTon	MoTa	Gia
1	Supmutra	6		311
2	Armbandry	64		736
3	Amptopra	7		13851605
4	Chartara	9		762
5	Armteller	90		17445399
6	Monoholdollry	64		4693995
7	Miccordonar	44	Rem possimus alias tempo	157

Giao diện Thêm Hóa Đơn sau khi thêm sản phẩm:

Thông tin hóa đơn

Mã khách hàng:

Thời gian:

MaSP	SoLuong	GiaBan	GiaGiam	ThanhTien
4	1	762	152	610
6	1	4693995	938799	3755196

### 3. Giao diện Thống kê doanh thu theo tháng

Thông tin các  
tháng cần xem  
doanh thu

## THỐNG KÊ DOANH THU THEO THÁNG

TỪ THÁNG: 1

NĂM: 2021

Xem

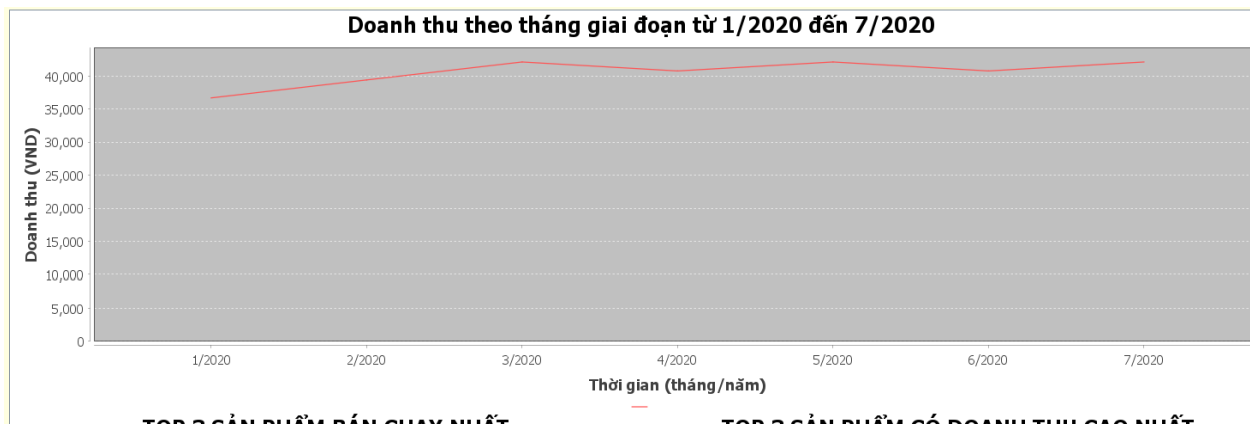
Quay lại

TỪ THÁNG: 1

NĂM: 2021

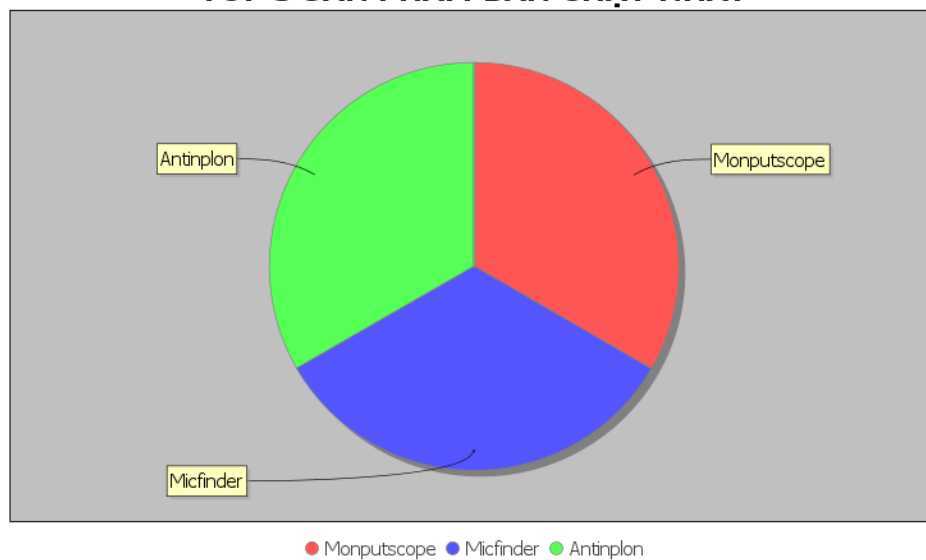
Sau khi chọn tháng và năm bắt đầu => kết thúc, giao diện thống kê sẽ hiển thị 3 thông tin:

- Biểu đồ đường cho doanh thu trong giai đoạn đã chọn.



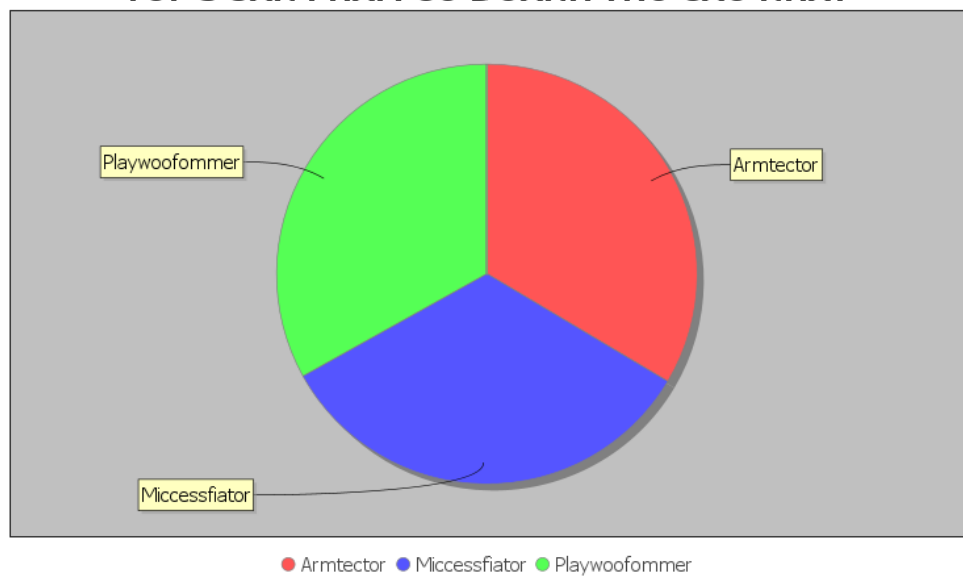
- Biểu đồ tròn cho top 3 sản phẩm bán chạy nhất trong giai đoạn tương ứng:

### TOP 3 SẢN PHẨM BÁN CHẠY NHẤT



- Biểu đồ tròn cho top 3 sản phẩm có doanh thu cao nhất trong giai đoạn tương ứng:

### TOP 3 SẢN PHẨM CÓ DOANH THU CAO NHẤT



## PHÂN CÔNG THÀNH VIÊN

Nhiệm vụ	Người thực hiện	Mức độ hoàn thành
Tạo Database và trigger theo yêu cầu	Quang Vinh	100%
Generate Data	Quang Vinh	100%
Trigger đề xuất	Diễm Linh	100%
Execution Plan	Diễm Linh	100%
Đề xuất chỉ mục	Diễm Linh	100%
Thiết kế giao diện	Thành Việt Quang Vinh	100%
Lập trình giao diện _ Form Thêm Hóa Đơn	Thành Việt	100%
Lập trình giao diện _ Form Danh Sách Hóa Đơn	Thành Việt	100%
Lập trình giao diện _ Form Thống Kê Doanh Thu	Diễm Linh	100%

### Mức độ đóng góp đồ án của thành viên

Diễm Linh	35%
Thành Việt	35%
Quang Vinh	30%