**BỘ MÔN HỆ THỐNG THÔNG TIN – KHOA CÔNG NGHỆ THÔNG TIN**

**ĐẠI HỌC KHOA HỌC TỰ NHIÊN THÀNH PHỐ HỒ CHÍ MINH, ĐẠI HỌC QUỐC GIA TP HCM**

Sinh viên thực hiện: 19127471- Lương Kiếm Minh

19127380- Phan Tường Duy

19127608- Nguyễn Quang Trường

18127068- Bùi Nguyễn Minh Châu

18127074- Trần Hải Đăng

GV phụ trách: Nguyễn Trần Minh Thư

Đồ án #1 môn học  - CƠ SỞ DỮ LIỆU NÂNG CAO

HỌC KỲ I – NĂM HỌC 2020-2021

**môn cơ sở dữ liệu nâng cao**

# **BẢNG THÔNG TIN CHI TIẾT NHÓM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã nhóm:** | ADB2\_9 | | | |
| **GitHub** | [HCMUS-Eakan/Advanced-Database (github.com)](https://github.com/HCMUS-Eakan/Advanced-Database) | | | |
| **Google Drive** | [Group 9 - Advanced Database - Google Drive](https://drive.google.com/drive/folders/1hxMONzs39XJt7qqkeUz_DnvwShiCMJjF) | | | |
| **Số lượng:** | **5** | | | |
| **MSSV** | **Họ tên** | **Email** | **Điện thoại** | **Hình ảnh** |
| 19127380 | Phan Tường Duy | 19127380@student.hcmus.edu.vn | 0868059405 |  |
| 19127471 | Lương Kiếm Minh | 19127471@student.hcmus.edu.vn | 0332040243 |  |
| 19127608 | Nguyễn Quang Trường | 19127608@student.hcmus.edu.vn | 0854693783 |  |
| 18127068 | Bùi Nguyễn Minh Châu | 18127068@student.hcmus.edu.vn | 0941112831 |  |
| 18127074 | Trần Hải Đăng | 18127074@student.hcmus.edu.vn | 0366902023 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng phân công & đánh giá hoàn thành công việc** | | | | |
| **Công việc thực hiện** | **Người thực hiện** | **MSSV** | **Mức độ hoàn thành** | **Đánh giá của nhóm** |
| Thiết kế giao diện, index | Phan Tường Duy | 19127380 | 100% | Đạt |
| Viết truy vấn, index, làm báo cáo | Lương Kiếm Minh | 19127471 | 100% | Đạt |
| Thiết kế giao diện, tạo data, trigger, procedure. | Nguyễn Quang Trường | 19127608 | 100% | Đạt |
| Tạo data, trigger, làm báo cáo. | Bùi Nguyễn Minh Châu | 18127068 | 100% | Đạt |
| Viết truy vấn, index. | Trần Hải Đăng | 18127074 | 100% | Đạt |

**YÊU CẦU ĐỒ ÁN- BÀI TẬP**

|  |  |
| --- | --- |
| **Loại bài tập** | **□Lý thuyết ⌧ Thực hành ⌧ Đồ án □Bài tập** |
| **Ngày bắt đầu** |  |
| **Ngày kết thúc** | **14/11/2021** |

**MỤC LỤC**

[**Mô tả Đồ án #1**](#_Toc87728781)

[**Báo cáo**](#_Toc87728782)

[Câu a: Cho danh sách các hoá đơn lập trong năm 2020](#_Toc87728783)

[Câu b: Cho danh sách các khách hàng ở TPHCM](#_Toc87728784)

[Câu c: Cho danh sách các sản phẩm có giá trong một khoảng từ….đến](#_Toc87728785)

[Câu d: Cho danh sách các sản phẩm có số lượng tồn <100](#_Toc87728786)

[Câu e: Cho danh sách các sản phẩm bán chạy nhất (số lượng bán nhiều nhất).](#_Toc87728787)

[Câu f: Cho danh sách các sản phẩm có doanh thu cao nhất.](#_Toc87728788)

[Câu 5-a: Select \* from A join B join C on…. Và Select \* from A,B,C where A.x = B.x….](#_Toc87728789)

[Câu 5-b: Select \* from A jọin B (A có số dòng nhỏ, B rất lớn) và Select \* from B join A](#_Toc87728790)

# **Mô tả Đồ án #1**

1. Chuẩn bị dữ liệu: Tạo csdl và phát sinh dữ liệu mẫu để test chỉ mục như sau:

* KhachHang (MakH, Ho, Ten, Ngsinh, SoNha, Duong, Phuong, Quan, Tpho, DienThoai)  100.000 rows, trong đó 2/3 số Khách hàng ở TPHCM, còn lại là các tỉnh thành khác.
* HoaDon (MaHD, MaKH, NgayLap, TongTien)  500.000 rows, trong đó ngày lập hoá đơn là từ 05-2020 đến 6/2021.
* CT\_HoaDon (MaHD, MaSP, SoLuong, GiaBan, GiaGiam, ThanhTien)  1 triệu rows.
* SanPham (MaSP, TenSP, SoLuongTon, Mota, Gia)  10.000 sp, trong đó 2/3 số dòng bỏ trống mô tả.
* Hướng dẫn sử dụng tool: <https://blog.devart.com/generate-test-data-with-sql-data-generator.html>
* Sinh viên tự xác định kiểu dữ liệu, độ dài dữ liệu cho lược đồ trên. Dữ liệu phát sinh phải đảm bảo liên hệ ý nghĩa:

1. Tổng tiền hoá đơn = sum (chi tiết hoá đơn) của hoá đơn đó.
2. HoaDon (MaKH) thuộc KhachHang(MaKH)
3. CT\_HoaDon (MaHD, MaSP) thuộc SanPham (MaSP) và HoaDon (MaHD)
4. Họ Tên khách hàng là chuỗi có ý nghĩa:

* Hồ Thị Hoàng Vy (DL có nghĩa);
* ABC DEF (DL không có nghĩa);

1. Cài trigger sau:
2. Thành tiền CTHD= (Số lượng \* (Giá bán-Giá giảm))
3. Tổng tiền (mahd) = sum (ThanhTien) cthd(mahd)
4. Viết các truy vấn sau:
5. Cho danh sách các hoá đơn lập trong năm 2020
6. Cho danh sách các khách hàng ở TPHCM
7. Cho danh sách các sản phẩm có giá trong một khoảng từ....đến
8. Cho danh sách các sản phẩm có số lượng tồn <100
9. Cho danh sách các sản phẩm bán chạy nhất (số lượng bán nhiều nhất)
10. Cho danh sách các sản phẩm có doanh thu cao nhất
11. Lập trình giao diện:
12. Thêm mới hoá đơn
13. Xem danh sách hoá đơn
14. Thống kê doanh thu theo tháng
15. Từ các kết quả truy vấn của câu 3, ghi nhận lại index recommendation từ execution plan (nếu có), Quan sát exection plan giải thích execution plan. Ngoài ra, nhận xét execution plan (thời gian thực thi) Cho một số trường hợp sau:
16. Select \* from A join B join C on.... Và Select \* from A,B,C where A.x = B.x....
17. Select \* from A jọin B (A có số dòng nhỏ, B rất lớn) và Select \* from B join A
18. Quy định nộp bài:

Thư mục nộp bài nén (.rar, .zip) đặt tên MaNhom\_DA1:

* Script tạo csdl trên (.sql),
* Script truy vấn, tạo index
* Báo cáo (.doc) giải thích kết quả chạy truy vấn và quan sát execution plan gồm có/không có index), kết quả điều chỉnh chỉ mục (nếu có) dựa vào gợi ý của sqlserver hoặc tự đề xuất.
* Source code lập trình giao diện post lên github
* Phân công công việc, % hoàn thành của nhóm và thành viên, xuất report từ Github
* Quay video có camera quá trình thảo luận, hoạt động nhóm (tạo drive folder đặt tên mã nhóm, nộp link vào báo cáo .doc ở trên, lưu ý drive folder này sử dụng cho các lần nộp sau)

# **Báo cáo**

## Cho danh sách các hoá đơn lập trong năm 2020

SELECT \*

FROM HoaDon hd

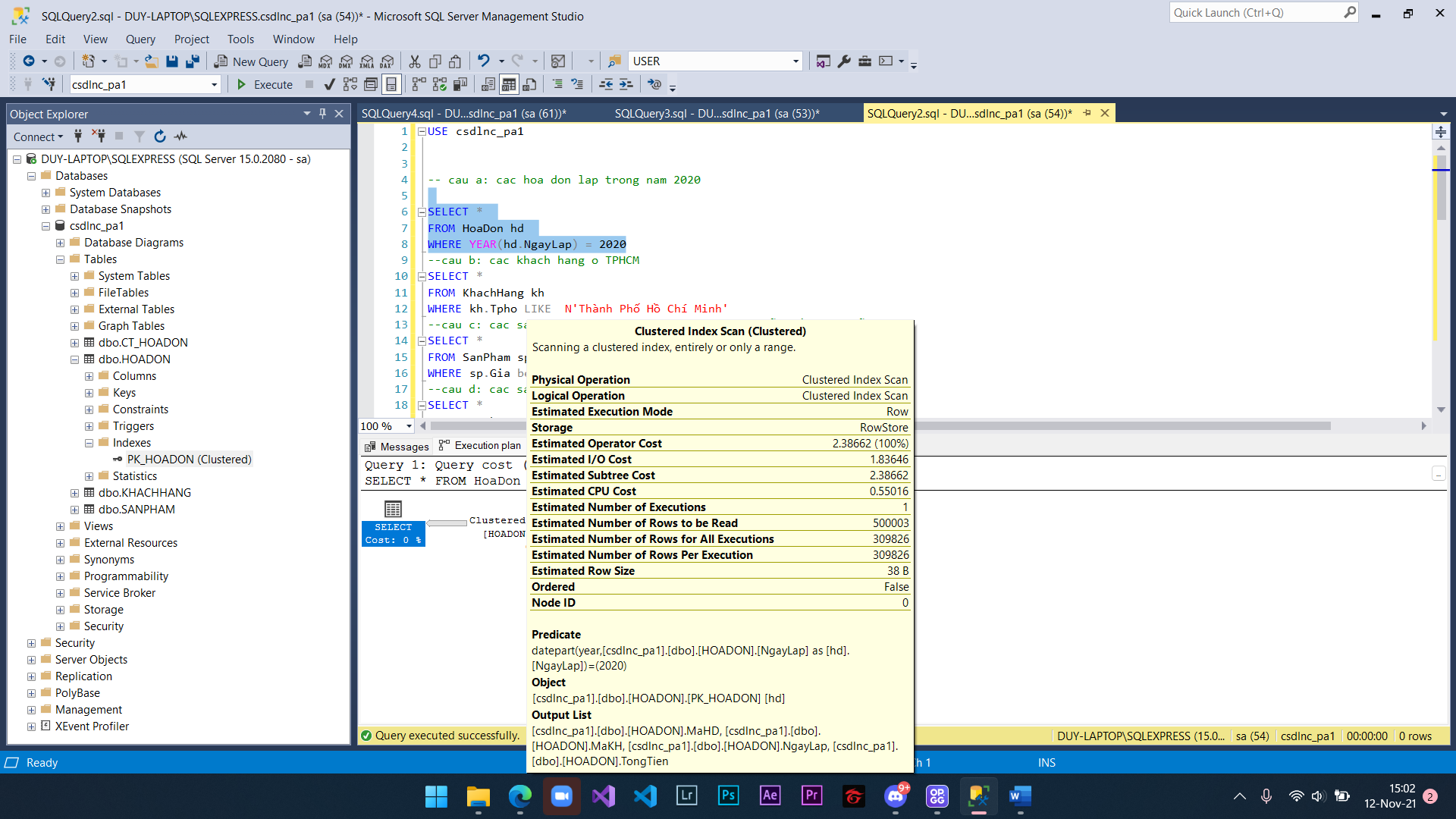
WHERE YEAR(hd.NgayLap) = 2020

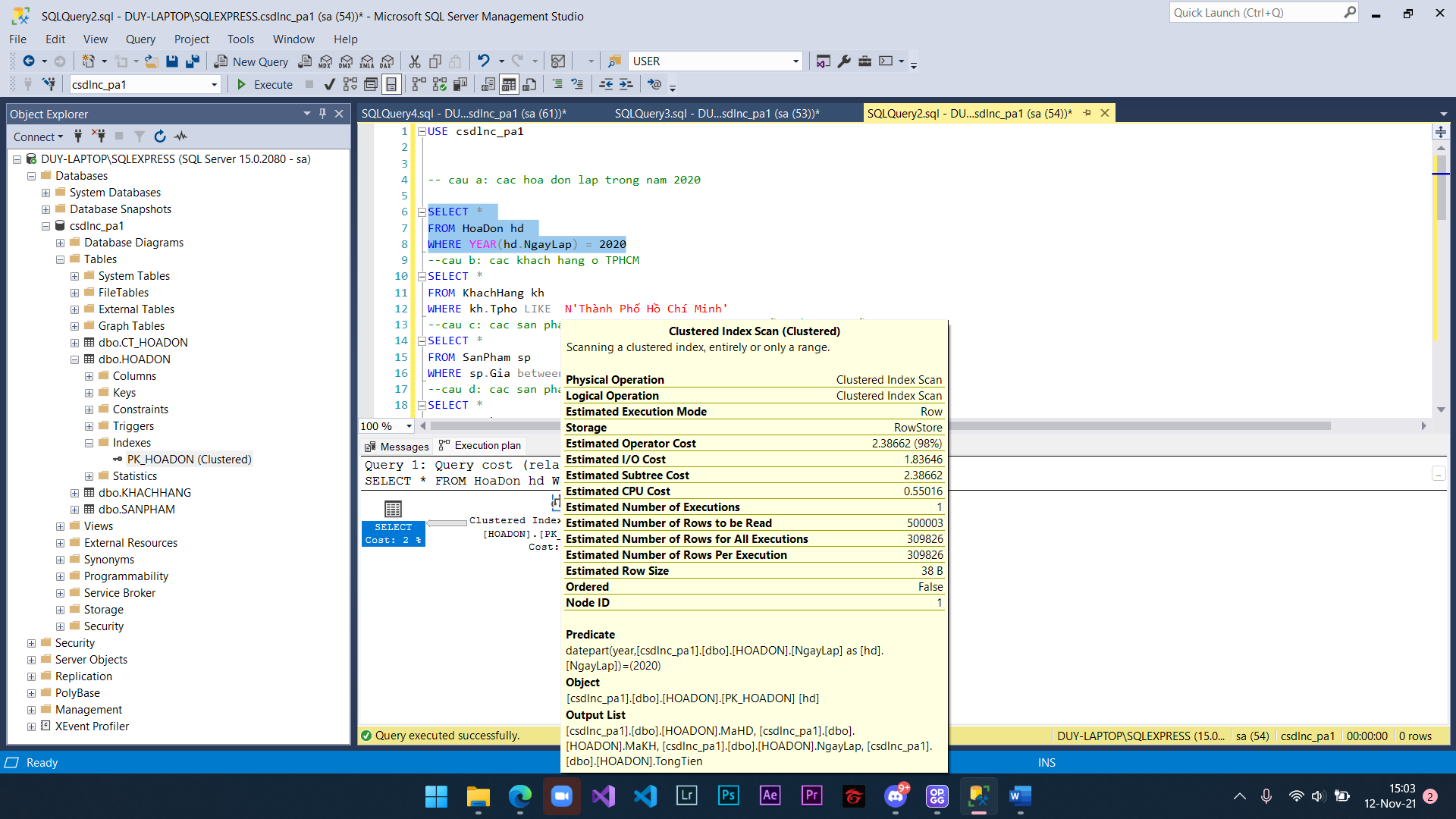
* Kết quả chạy truy vấn.

Ảnh có chứa bàn

Mô tả được tạo tự động

* Execution plan không có index, hệ thống không đề xuất index. Clustered index scan toàn table, chi phí truy xuất 100% , tốc độ truy xuất 2,3s.



* Exection plan do sinh viên tự đề xuất. Clustered index scan toàn table, chi phí truy xuất 98%, giảm 2% so với ban đầu, tốc độ truy xuất giữ nguyên.
* 

## Cho danh sách các khách hàng ở TPHCM

SELECT \*

FROM KhachHang kh

WHERE kh.Tpho LIKE N'Thành Phố Hồ Chí Minh'

* Kết quả chạy truy vấn.

Graphical user interface, application, table

Description automatically generated

* Execution plan không có index. Clustered index scan toàn bộ table, chi phí truy xuất 100%, tốc độ truy xuất 1,75s.

Graphical user interface

Description automatically generated

* Hệ thống có đề xuất index để giảm chi phí 52.5564%.

Text, application

Description automatically generated

* Execution plan sau khi đã thêm index mà hệ thống đề xuất. Index seek 66685/100000 dòng trong table, thời gian truy xuất giảm xuống còn 1,15s, ordered từ False chuyển sang True, có nghĩa table đã được sắp xếp.

A screenshot of a computer

Description automatically generated

## Cho danh sách các sản phẩm có giá trong một khoảng từ….đến

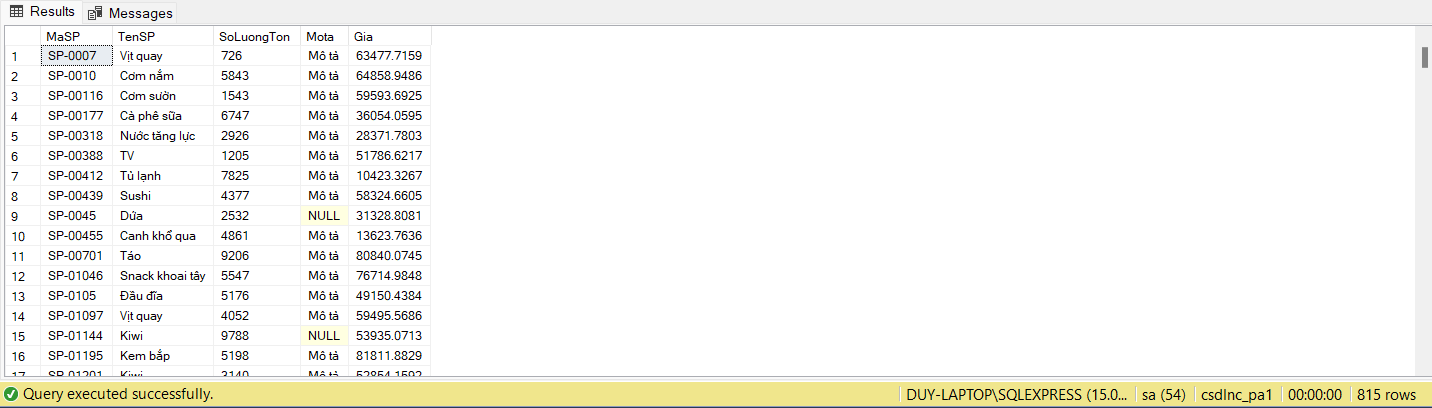
Cho ví dụ từ 10000 đến 90000

SELECT \*

FROM SanPham sp

WHERE sp.Gia between 0 and 100000

* Kết quả chạy truy vấn



* Execution plan không có index hệ thống scan toàn bộ table (10000 dòng dữ liệu), chi phí truy xuất 100%, truy xuất thành công trong 0.064s.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Execution plan có index do sinh viên tự đề xuất. Không nhận thấy có sự thay đổi trong hiệu quả truy xuất.

Ảnh có chứa văn bản

Mô tả được tạo tự động

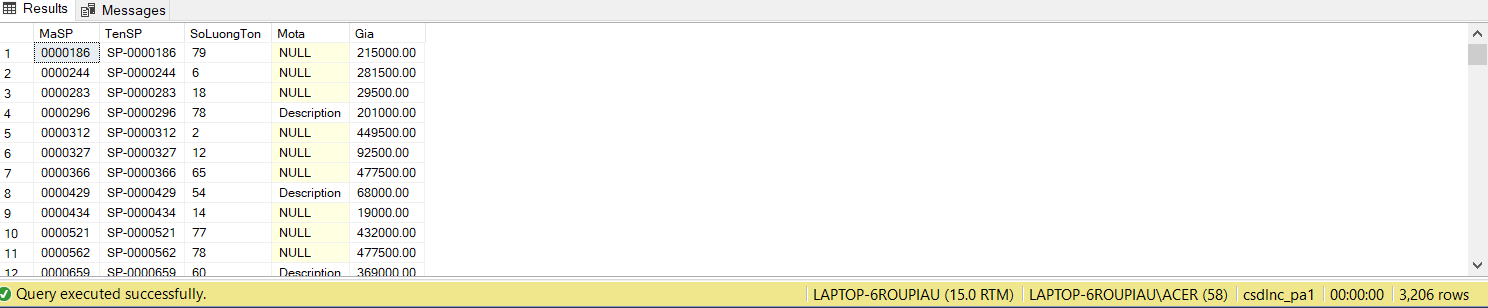
## Cho danh sách các sản phẩm có số lượng tồn <100

SELECT \*

FROM SanPham sp

WHERE sp.SoLuongTon < 100

* Kết quả chạy truy vấn



* Execution plan không có index. Hệ thống scan trên toàn bộ bảng sản phẩm, chi phí truy xuất 100%, tốc độ 0.097s

Ảnh có chứa văn bản

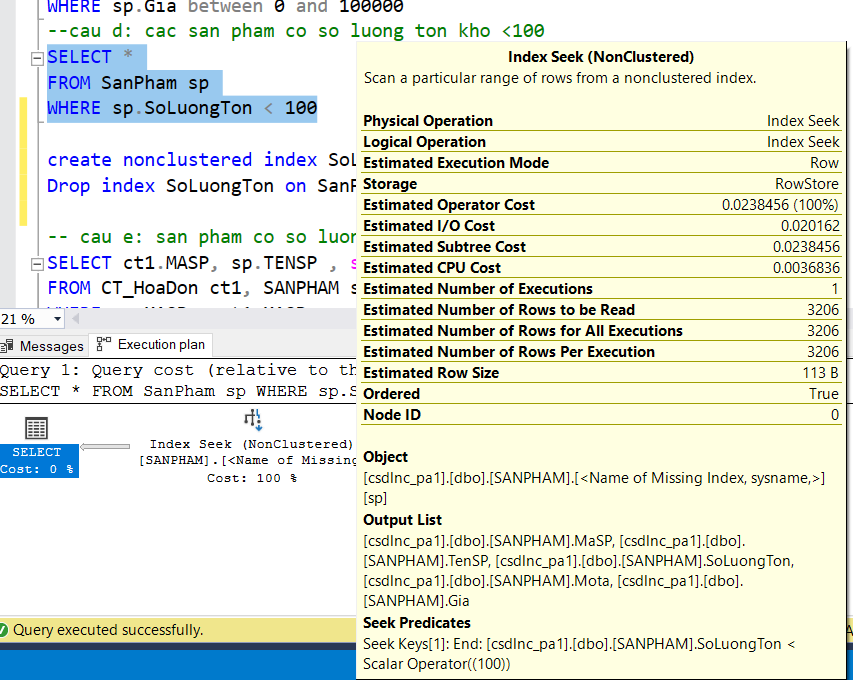
Mô tả được tạo tự động

* Hệ thống có đề xuất index để giảm chi phí 56.6024%

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Execution plan có index được gợi ý từ hệ thống, tốc độ truy xuất giảm còn 0.023s, ordered từ False chuyển sang True, có nghĩa table đã được sắp xếp.



## Cho danh sách các sản phẩm bán chạy nhất (số lượng bán nhiều nhất).

SELECT ct1.MASP, sp.TENSP , sum(ct1.SoLuong) as SoSanPhamBanRa

FROM CT\_HoaDon ct1, SANPHAM sp

WHERE sp.MASP = ct1.MASP

GROUP BY ct1.MASP, sp.TENSP

HAVING sum(ct1.SoLuong) >= ALL (SELECT sum(ct.SoLuong)

FROM CT\_HoaDon ct, SANPHAM sp

WHERE ct.MASP = sp.MASP

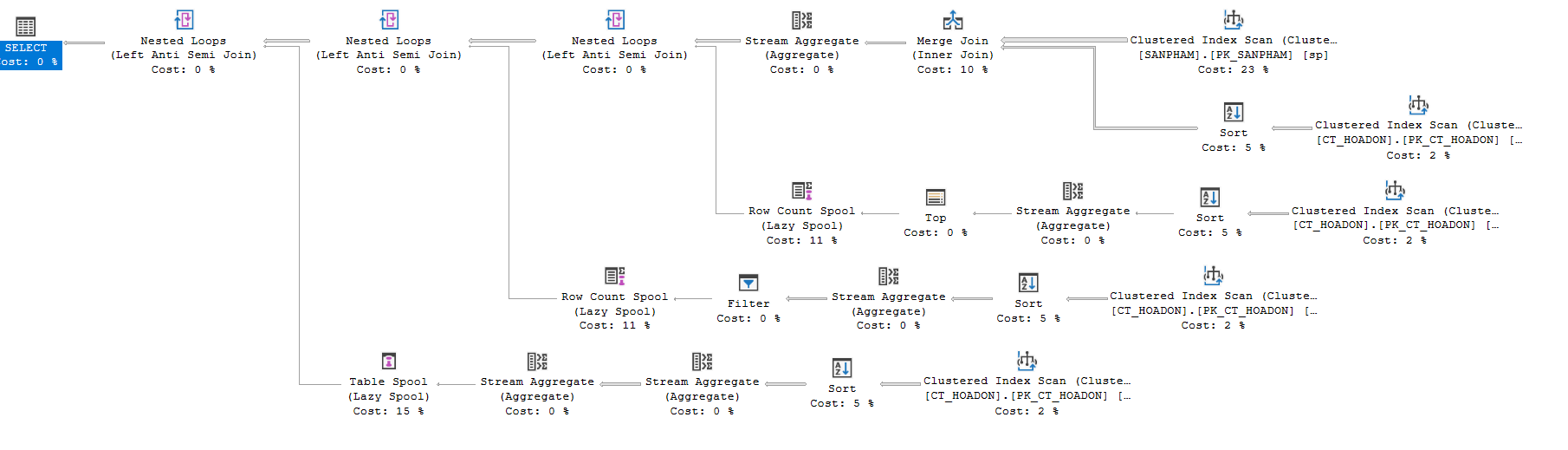
GROUP BY ct.MASP)

* Kết quả chạy truy vấn

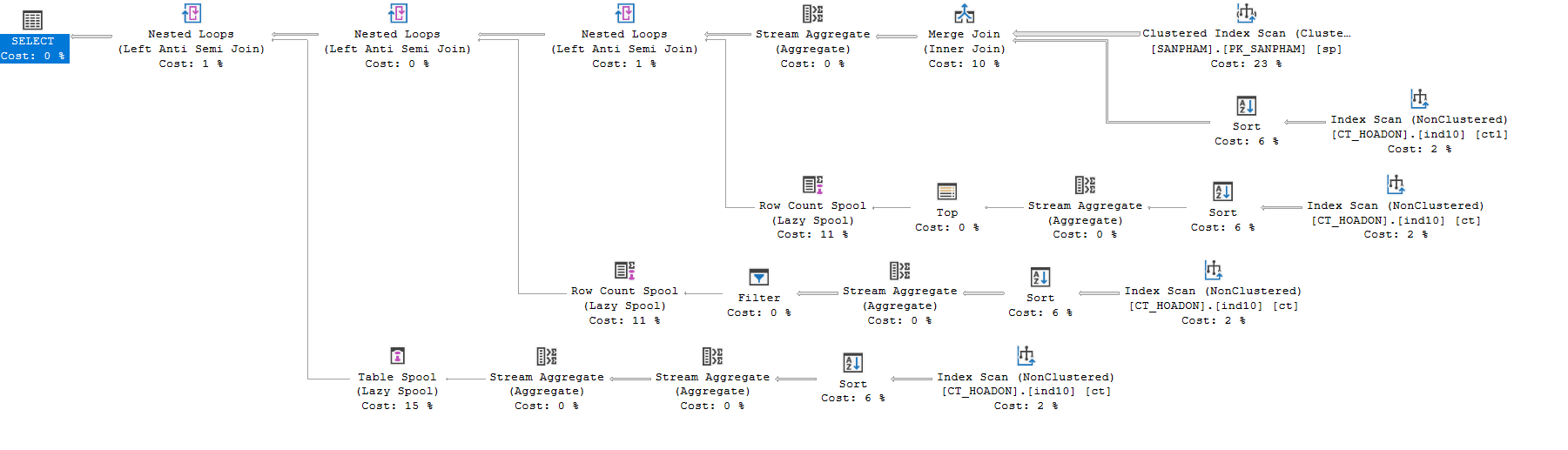
Ảnh có chứa bàn

Mô tả được tạo tự động

* Execution plan không có index.



* Execution plan có index



## Cho danh sách các sản phẩm có doanh thu cao nhất.

SELECT ct1.MASP, sp.TENSP , sum(ct1.ThanhTien) as SoTien

FROM CT\_HoaDon ct1, SANPHAM sp

WHERE sp.MASP = ct1.MASP

GROUP BY ct1.MASP, sp.TENSP

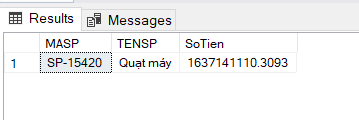
HAVING sum(ct1.ThanhTien) >= ALL ( SELECT sum(ct.ThanhTien)

FROM CT\_HoaDon ct, SANPHAM sp

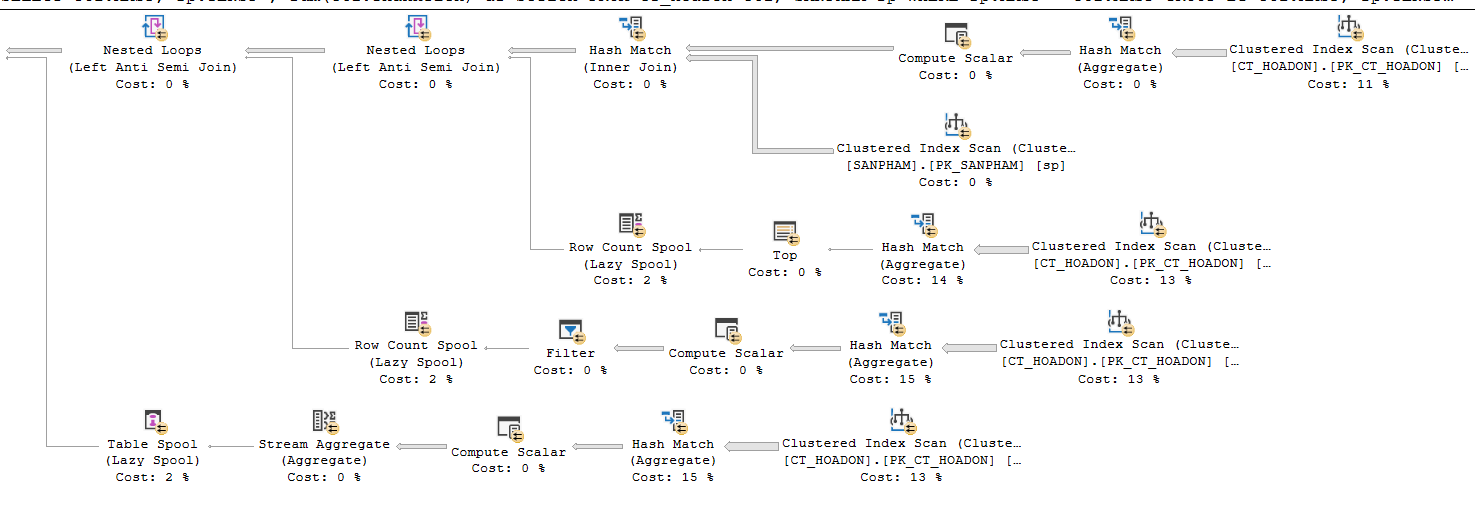
WHERE ct.MASP = sp.MASP

GROUP BY ct.MASP)

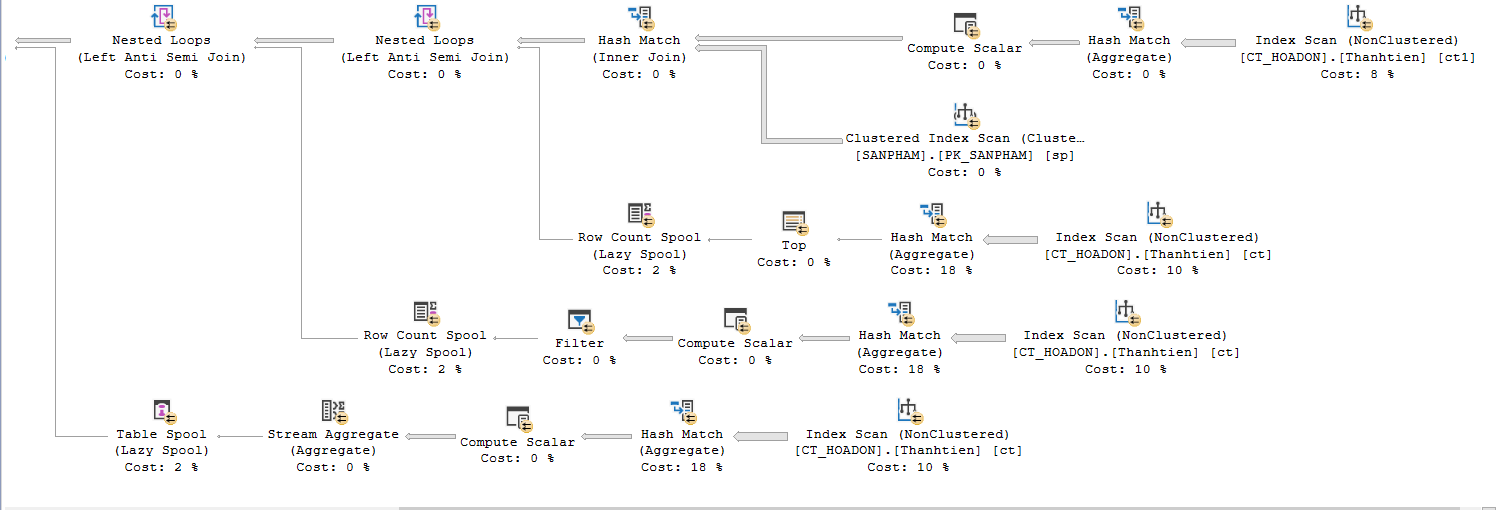
* Kết quả truy vấn



* Execution plan không có index



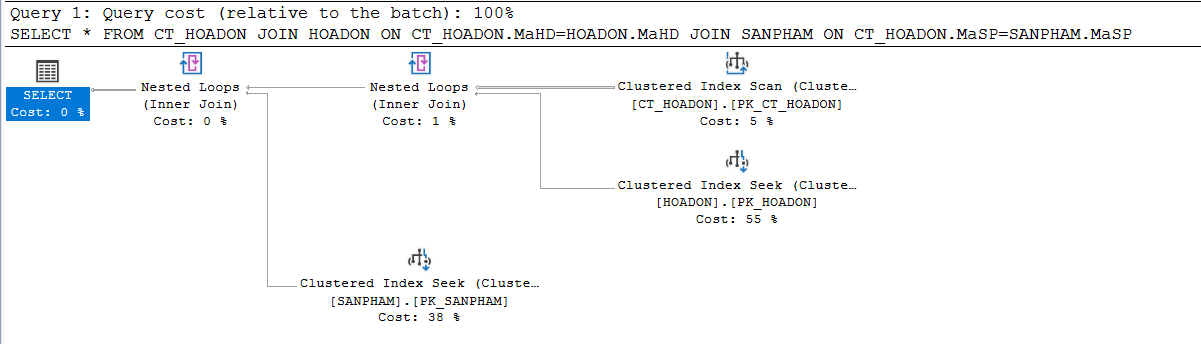
* Execution plan có index



## Select \* from A join B join C on…. Và Select \* from A,B,C where A.x = B.x….

SELECT \* FROM CT\_HOADON JOIN HOADON ON CT\_HOADON.MaHD=HOADON.MaHD

JOIN SANPHAM ON CT\_HOADON.MaSP=SANPHAM.MaSP



SELECT \*FROM CT\_HOADON,HOADON, SANPHAM

WHERE HOADON.MaHD=CT\_HOADON.MaHD AND CT\_HOADON.MaSP=SANPHAM.MaSP

Ảnh có chứa văn bản

Mô tả được tạo tự động

## Select \* from A jọin B (A có số dòng nhỏ, B rất lớn) và Select \* from B join A

Select \* from SANPHAM join CT\_HOADON ON SANPHAM.MaSP=CT\_HOADON.MaSP

Ảnh có chứa văn bản

Mô tả được tạo tự động

Select \* from CT\_HOADON join SANPHAM ON SANPHAM.MaSP=CT\_HOADON.MaSP

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Cách chạy thứ 2 tối ưu hơn vì cùng là inner join nhưng cách 2 chỉ tốn 27% chi phí truy vấn thấp hơn 5% so với cách truy vấn thứ 1.

/\*Note: Link Google Drive và GitHub chúng em để ở [Bảng thông tin chi tiết nhóm](#_BẢNG_THÔNG_TIN) ạ.\*/