**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**

**GV phụ trách**:

Cô **Nguyễn Trần Minh Thư**

Cô **Hồ Thị Hoàng Vy**

**CƠ SỞ DỮ LIỆU NÂNG CAO**

HỌC KỲ I – NĂM HỌC 2020 - 2021

Nhóm thực hiện: **Nhóm 6 – ADB1\_6**

**BẢNG THÔNG TIN CHI TIẾT NHÓM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã nhóm:** | ADB1\_6 | | | |
| **Tên nhóm:** | **Nhóm 6** | | | |
| **Số lượng:** | 4 | | | |
| **MSSV** | **Họ tên** | **Email** | **Điện thoại** | **Hình ảnh** |
| 19127095 | Ngô Huy Anh | nhanh19@clc.fitus.edu.vn | 0938447496 | A person standing in a room  Description automatically generated with low confidence |
| 19127449 | Phùng Anh Khoa | pakhoa19@clc.fitus.edu.vn | 0845500024 |  |
| 19127459 | Đoàn Nguyễn Hương Lan | dnhlan19@clc.fitus.edu.vn | 0335460332 |  |
| 19127569 | Nguyễn Huy Anh Thư | nhathu19@clc.fitus.edu.vn | 0938374163 | Ảnh có chứa người, lam, nam  Mô tả được tạo tự động |

**ĐỒ ÁN THỰC HÀNH #1**

|  |  |
| --- | --- |
| **Loại bài tập** | **🗹 Thực hành Lý thuyết Bài tập 🗹 Đồ án** |
| **Ngày bắt đầu** | **04/10/2021** |
| **Ngày kết thúc** | **14/11/2021** |

Mục lục

[A. Bảng phân công công việc 4](#_Toc87780627)

[B. Kết quả 5](#_Toc87780628)

[**I.** **Kết quả truy vấn** 5](#_Toc87780629)

[**Kết quả truy vấn 1:** 5](#_Toc87780630)

[**Kết quả truy vấn 2:** 6](#_Toc87780631)

[**Kết quả truy vấn 3:** 7](#_Toc87780632)

[**Kết quả truy vấn 4:** 8](#_Toc87780633)

[**Kết quả truy vấn 5:** 9](#_Toc87780634)

[**Kết quả truy vấn 6:** 10](#_Toc87780635)

[**II.** **Quan sát execution plan** 11](#_Toc87780636)

[**Kết quả truy vấn 1:** 11](#_Toc87780637)

[**Kết quả truy vấn 2:** 12](#_Toc87780638)

[**Kết quả truy vấn 3:** 13](#_Toc87780639)

[**Kết quả truy vấn 4:** 14](#_Toc87780640)

[**Kết quả truy vấn 5:** 15](#_Toc87780641)

[**Kết quả truy vấn 6:** 16](#_Toc87780642)

[**a. Select \* from A join B join C on.... Và Select \* from A,B,C where A.x = B.x....** 16](#_Toc87780643)

[**b. Select \* from A jọin B (A có số dòng nhỏ, B rất lớn) và Select \* from B JOIN A** 19](#_Toc87780644)

[**III.** **Link data & Github & Video** 22](#_Toc87780645)

[**IV.** **Xuất report từ github** 22](#_Toc87780646)

[**V.** **Tham khảo** 23](#_Toc87780647)

# Bảng phân công công việc

|  |  |  |  |
| --- | --- | --- | --- |
| **Bảng phân công & đánh giá hoàn thành công việc** | | | |
| **Công việc thực hiện** | **Người thực hiện** | **Mức độ hoàn thành** | **Đánh giá của nhóm** |
| Viết script tạo CSDL | Phùng Anh Khoa | 100% | 100% |
| Viết trigger | Phùng Anh Khoa | 100% | 100% |
| Chuẩn bị dữ liệu | Ngô Huy Anh  Phùng Anh Khoa | 100% | 100% |
| Viết các truy vấn | Đoàn Nguyễn Hương Lan | 100% | 100% |
| Lập trình giao diện | Ngô Huy Anh | 100% | 100% |
| Ghi nhận lại index recommendation từ execute plan | Nguyễn Huy Anh Thư | 100% | 100% |
| Quan sát exection plan giải thích execution plan. Nhận xét execution plan cho 2 trường hợp a) b) | Nguyễn Huy Anh Thư | 100% | 100% |
| Tổng hợp report | Nguyễn Huy Anh Thư | 100% | 100% |
| Kiểm tra toàn bộ | Ngô Huy Anh | 100% | 100% |

# Kết quả

1. **Kết quả truy vấn**

### **Kết quả truy vấn 1:**

Cho danh sách các hoá đơn lập trong năm 2020

Graphical user interface, application, table

Description automatically generated with medium confidence

### **Kết quả truy vấn 2:**

Cho danh sách các khách hàng ở TPHCM

Table

Description automatically generated

### **Kết quả truy vấn 3:**

Cho danh sách các sản phẩm có giá trong một khoảng từ 10000 đến 100000

Table

Description automatically generated

### **Kết quả truy vấn 4:**

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

Table

Description automatically generated with medium confidence

### **Kết quả truy vấn 5:**

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

Graphical user interface, application

Description automatically generated

### **Kết quả truy vấn 6:**

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

A picture containing graphical user interface

Description automatically generated

1. **Quan sát execution plan**

### **Kết quả truy vấn 1:**

Graphical user interface, text, application, email

Description automatically generated

Ở excution plan ta có thể thấy câu lệnh sử dụng một Clustered Index scan từ HoaDon vì trong hoá đơn có khoá chính nên nó xài luôn khoá chính để sắp xếp kết quả trả về.

Table

Description automatically generated

Ở câu lệnh này chỉ sử dụng 1 bảng và select toàn bộ giá trị trong bảng đó, nên thời gian chạy của cậu lênh đều nằm hết ở quá trình kiểm tra 500000 dòng với lượng data là 10MB Table

Description automatically generated with medium confidence

### **Kết quả truy vấn 2:**

Graphical user interface, text, application, email

Description automatically generated

Trong Khách Hàng cũng có khoá chính và câu lệnh chỉ sử dụng 1 bảng để select nên câu lệnh sử dụng luôn khoá chính của khách hàng để trả về kết quả.

Table

Description automatically generated with medium confidence

Ở câu lệnh này chỉ sử dụng 1 bảng và select toàn bộ giá trị trong bảng đó, nên thời gian chạy của cậu lênh đều nằm hết ở quá trình kiểm tra 100000 dòng với lượng data là 20MB

Table

Description automatically generated with medium confidence

### **Kết quả truy vấn 3:**

Graphical user interface, text, application

Description automatically generated

Trong Sản phẩm cũng có khoá chính và câu lệnh chỉ sử dụng 1 bảng để select nên câu lệnh sử dụng luôn khoá chính của Sản phẩm để trả về kết quả.

Text

Description automatically generated with low confidence

Ở câu lệnh này chỉ sử dụng 1 bảng và select toàn bộ giá trị trong bảng đó, nên thời gian chạy của cậu lênh đều nằm hết ở quá trình kiểm tra 10000 dòng với lượng data là 95KB

Graphical user interface

Description automatically generated with medium confidence

### **Kết quả truy vấn 4:**

Graphical user interface, text, application

Description automatically generated

trong Sản phẩm cũng có khoá chính và câu lệnh chỉ sử dụng 1 bảng để select nên câu lệnh sử dụng luôn khoá chính của Sản phẩm để trả về kết quả.

Graphical user interface, application

Description automatically generated with medium confidence

Ở câu lệnh này chỉ sử dụng 1 bảng và select toàn bộ giá trị trong bảng đó, nên thời gian chạy của cậu lênh đều nằm hết ở quá trình kiểm tra 10000 dòng với lượng data là 105B vì không có kết quả trả về Table

Description automatically generated

### **Kết quả truy vấn 5:**

Graphical user interface, application, Word

Description automatically generated

Ở đây sử dụng 2 bảng để kết xuất dữ liệu là bảng sản phẩm và bảng hoá đơn.

Ta thấy 2 clustered index của Sản phẩm và hoá đơn để câu lệnh lấy dữ liệu từ 2 bảng này

Graphical user interface, text, application, letter

Description automatically generated

Câu lệnh sort bảng *CT\_HoaDon* theo mã sản phẩm để có thể Merger với Sản phẩm :

**db.CT\_Hoadon.masp = SanPham.MaSP**

Graphical user interface, text

Description automatically generated

Group theo *SanPham.MaSP* và tính toán giá trị mới là :

**soluong = SUM(CT\_HoaDon.Soluong)**

Text

Description automatically generated

Sort kết quả theo order by soluong (giảm dần) rồi trả về kết quả

Graphical user interface, text

Description automatically generated

### **Kết quả truy vấn 6:**

Graphical user interface, application

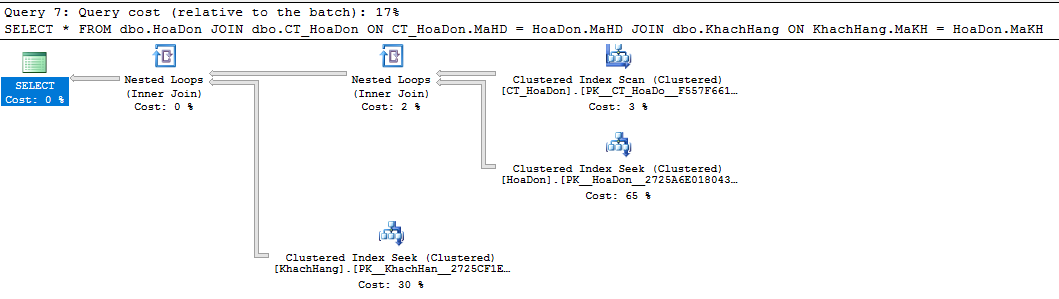
Description automatically generated

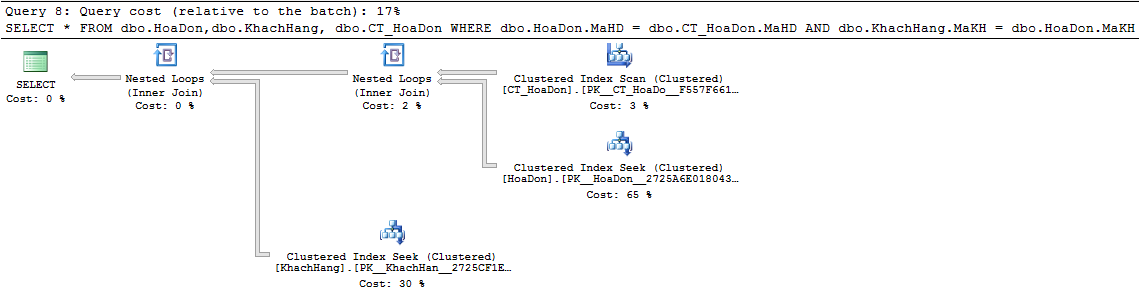
Tương tự như truy vấn 5 nhưng thay vì tính toán giá trị mới là số lượng thì ta tính toán giá trị mới là **DoanhThu = SUM(CT\_HoaDon.ThanhTien)**

**Giải thích truy vấn 5 và 6:**

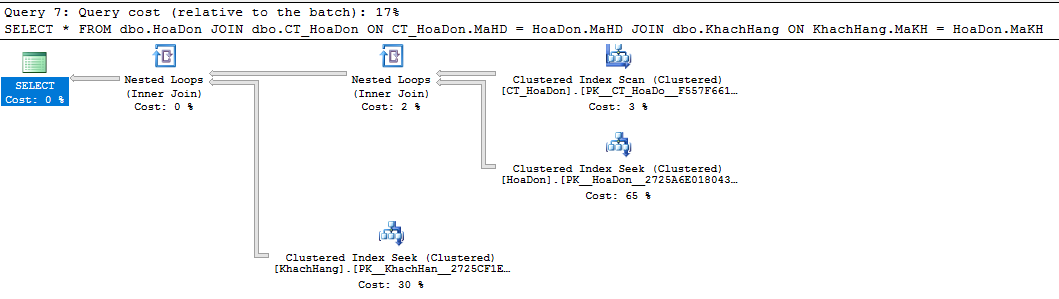
* Đầu tiên lấy giá trị từ bảng *CT\_HoaDon*
* Sort lại theo *MaSP* nằm trong *CT\_HoaDon* rồi merge với bảng *SanPham* với điều kiệu **CT\_HoaDon.MaSP = SanPham.HoaDong**
* Kế tiếp Group các giá trị theo *MaSP*
* tính toán giá trị mới *DoanhThu(soluong)* từ các thông tin có sau khi Group by
* Sort theo danh sách giảm dần theo *DoanhThu(soluong)* rồi xuất kết quả

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





Cả hai câu lệnh đều có execution plan như nhau: đều đi qua 2 lần nested loops để tìm matching giữa 2 bảng để tạo ra bảng join cuối cùng



Tại **Nested Loops** đầu tiên có outer table là bảng CT\_HoaDon (sau khi được scan tại **Index Scan** để lấy hết dữ liệu của bảng) và inner table là bảng HoaDon (sau khi thực hiện **Index Seek**, ở bước này lấy dữ liệu từ bảng HoaDon và sort nó theo thứ tự từ bé đến lớn bằng B-Tree nên nó tiêu tốn nhiều thời gian nhất – 65% tổng thời gian).

Ảnh có chứa văn bản

Mô tả được tạo tự động

(pseodo-code được copy từ [Craig Freedman’s Blog](http://blogs.msdn.com/b/craigfr/archive/2006/07/26/679319.aspx))

Theo nguyên lý hoạt động của **Nested Loops** (hình trên) tiến hành lấy từng cột của bảng CT\_HoaDon so với toàn bảng HoaDon, nếu tìm thấy matching thì return => Kết quả có được bảng join được sort từ nhỏ đến lớn từ 2 bảng CT\_HoaDon và HoaDon.

* Kích thước và số hàng của bảng CT\_HoaDon:

Ảnh có chứa bàn

Mô tả được tạo tự động

* Kích thước và số hàng của bảng HoaDon:

Ảnh có chứa bàn

Mô tả được tạo tự động

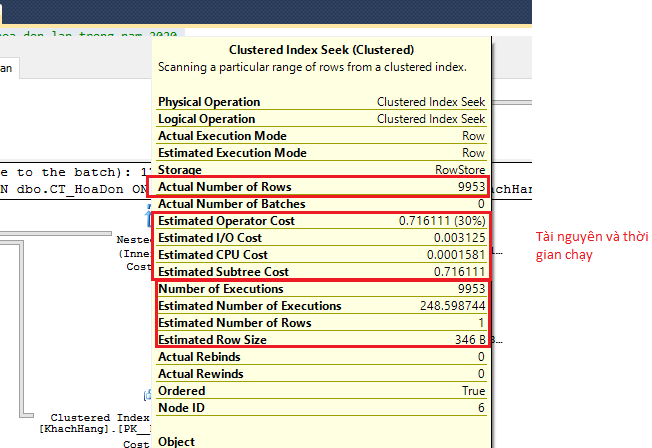
* Kết quả nested loops lần 1:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Tương tự với **Index Seek** bảng KhachHang (chiếm 30% tổng thời gian) sau khi được seek và sort sẽ cùng bảng kết quả nested loops trên tiếp tục nested loops (outer table: bảng kết quả 1, inner table: KhachHang) để tìm ra matching => Ra bảng kết quả cuối cùng

* Kích thước và số hàng của bảng KhachHang:



* Kích thước và số hàng của bảng kết quả cuối cùng;

Ảnh có chứa văn bản

Mô tả được tạo tự động

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

Ảnh có chứa văn bản

Mô tả được tạo tự động

Cả hai câu lệnh đều có execution plan như nhau: Đều xuất hiện **Hash Match** để join 2 bảng *CT\_HoaDon* và *SanPham.* Đầu tiên hash bảng nhỏ hơn theo hàng từ nhỏ đến lớn rồi đưa chúng vào 1 hash table. Sau đó lấy từng hàng 1 của bảng lớn hơn so với hash table vừa lập, tìm kết quả phù hợp mà tại đó các hàng cần được join.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Tại bảng này **Hash match** 2 bảng *CT\_HoaDon* và *SanPham*. Đầu tiên tạo hash table từ bảng *CT\_HoaDon* theo thứ tự từ nhỏ đến lớn. Sau đó ở từng hàng của bảng *SanPham* đem so với từng hàng của hash table vừa lập để tìm ra matching (lấy MaSP: **CT\_HoaDon.MaSP = dbo.SanPham.MaSP** )

* Bảng *CT\_HoaDon* ban đầu có 9953 hàng:

Ảnh có chứa văn bản

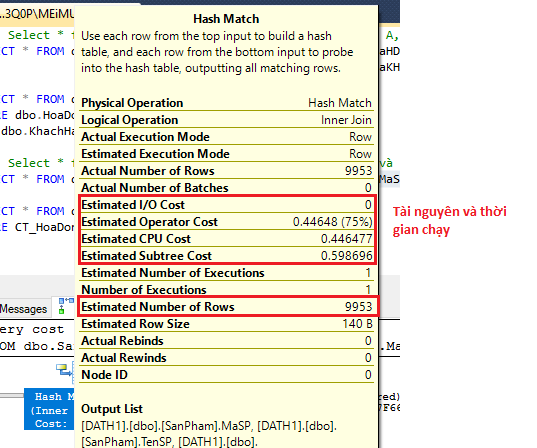
Mô tả được tạo tự động

* Bảng *SanPham* ban đầu có 10000 hàng:

Ảnh có chứa văn bản, bàn

Mô tả được tạo tự động

* Sau khi thực hiện **Hash match** ta thu được bảng có số hàng bằng với số hàng của bảng *CT\_HoaDon*:



Kết quả ta có bảng join theo thứ tự từ bé đến lớn của *MaSP*. Lúc này bảng kết quả sẽ là bảng join giữa 2 bảng *CT\_HoaDon* và *SanPham* nên kích thước sẽ lớn hơn (bé hoặc bằng 2 bảng cộng lại do là join nên sẽ có phần lấy phần không)

* Kích thước hàng của bảng *CT\_HoaDon* (43B)

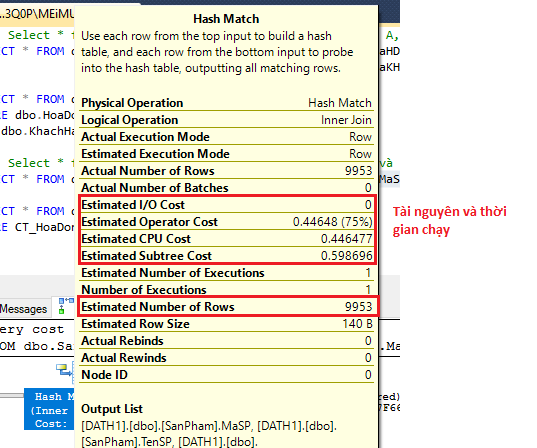
Ảnh có chứa bàn

Mô tả được tạo tự động

* Kích thước hàng của bảng *SanPham* (100B):

Ảnh có chứa bàn

Mô tả được tạo tự động

* Kích thước bảng kết quả (140B):
* 

1. **Link data & Github & Video**

Link data : [Data Generator](https://drive.google.com/drive/folders/1hMvJTTezJwE8Hd5pdr2w-aryFKz-6qGx?usp=sharing)

Link github: [Github](https://github.com/tristanhuyanhngo/Advanced-Database_Project01)

Link video: [Video](https://drive.google.com/drive/folders/1mokppJBf1GGgmanTPZHk5SJjvQJuCDAG?usp=sharing)

1. **Xuất report từ github**
2. **Tham khảo**
3. **Video seminar Đồ án#1** – Cô **Hồ Thị Hoàng Vy**
4. **Bộ slides nội dung thực hành –** Cô **Hồ Thị Hoàng Vy**