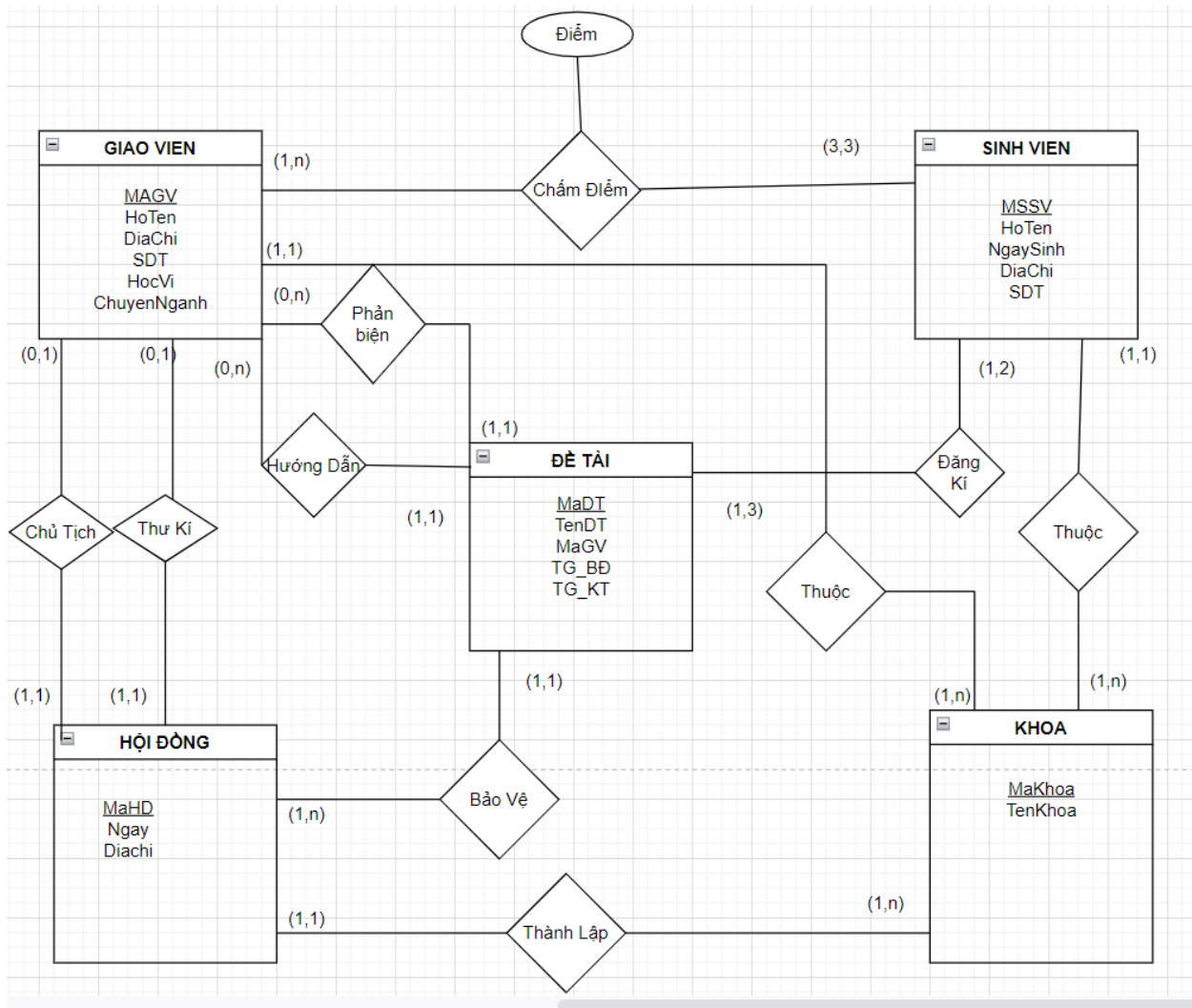


## 1. Chuyển đổi biểu diễn dữ liệu từ mức khái niệm (ERD, CD) ở BTVN Buổi 2 sang Biểu diễn theo mức logic.

Trả lời:

Mô hình ERD:



Mô hình Logic:

- SinhVien (MaSV, DiemTB, TenSV, MaKhoa, MaDeTaiThucHien, MaDeTaiBaoVe)
- Khoa (MaKhoa, TenKhoa)

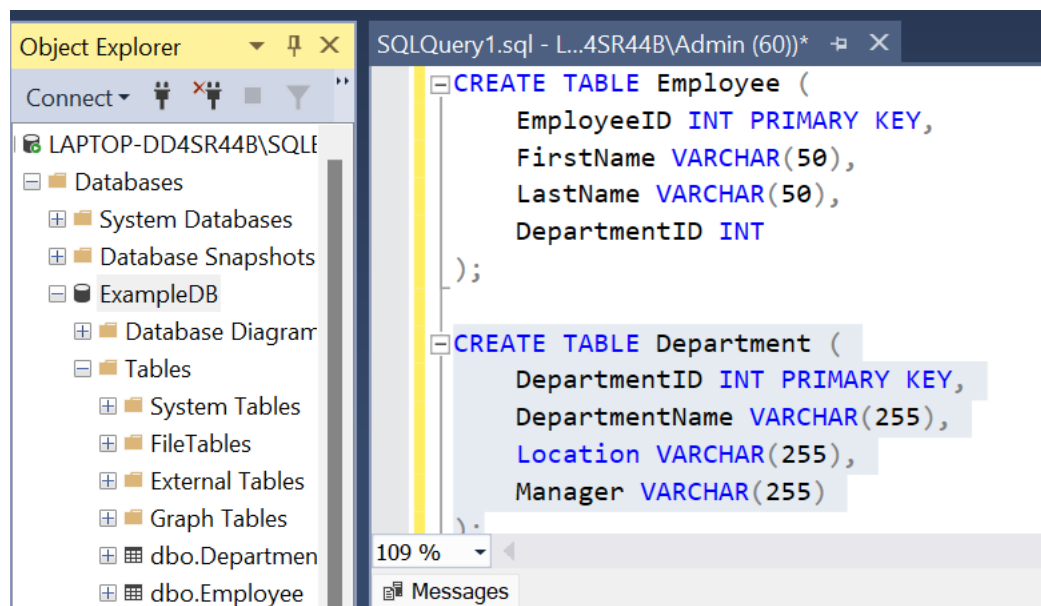
- HoiDong (MaHoiDong, ChuTich, NgayBaoVe, DiaChiBaoVe, ThuKy, MaKhoa)
- GiaoVien (MaGV, TenGV, SDT, HocVi, ChuyenNganh, DiaChi, MaHoiDong)
- DeTai (MaDeTai, TenDeTai, ThoiGianBatDau, ThoiGianKetThuc, MaKhoa, MaGVHD, MaGVPhanBien)

## 2. Thực hiện thử nghiệm các thao tác với Index trên một CSDL mẫu.

Trả lời:

- Tạo CSDL mẫu:

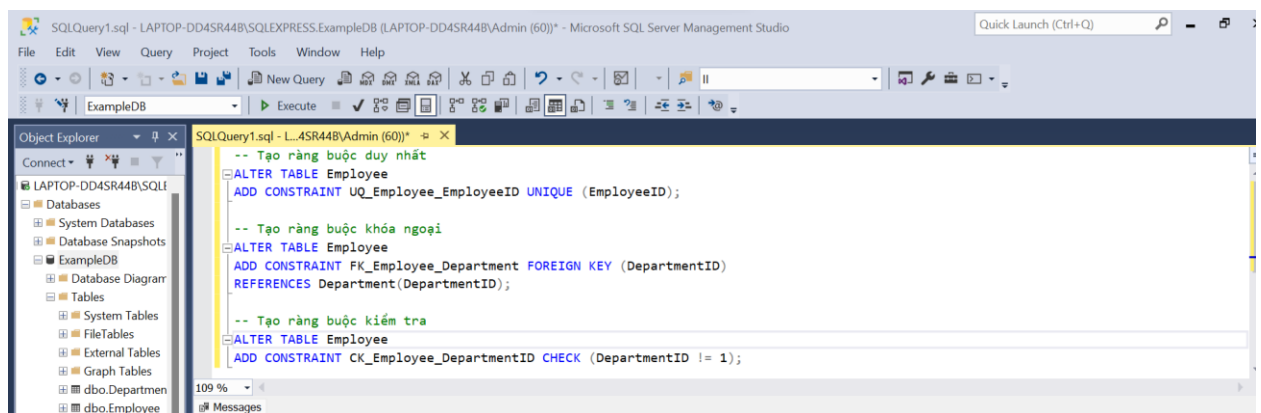
```
CREATE TABLE Employee (  
    EmployeeID INT PRIMARY KEY,  
    FirstName VARCHAR (50),  
    LastName VARCHAR (50),  
    DepartmentID INT);
```



- Phát hiện các ràng buộc:

- Ràng buộc duy nhất (Unique constraint): EmployeeID và DepartmentID là Unique
- Ràng buộc khóa chính (Primary key constraint): EmployeeID và DepartmentID là PrimaryKey
- Ràng buộc ngoại (Foreign key constraint): DepartmentID trong table Employee là khóa ngoại tham chiếu đến DepartmentID trong table Department
- Ràng buộc kiểm tra (Check constraint): DepartmentID phải khác 0

- Cài đặt các ràng buộc:

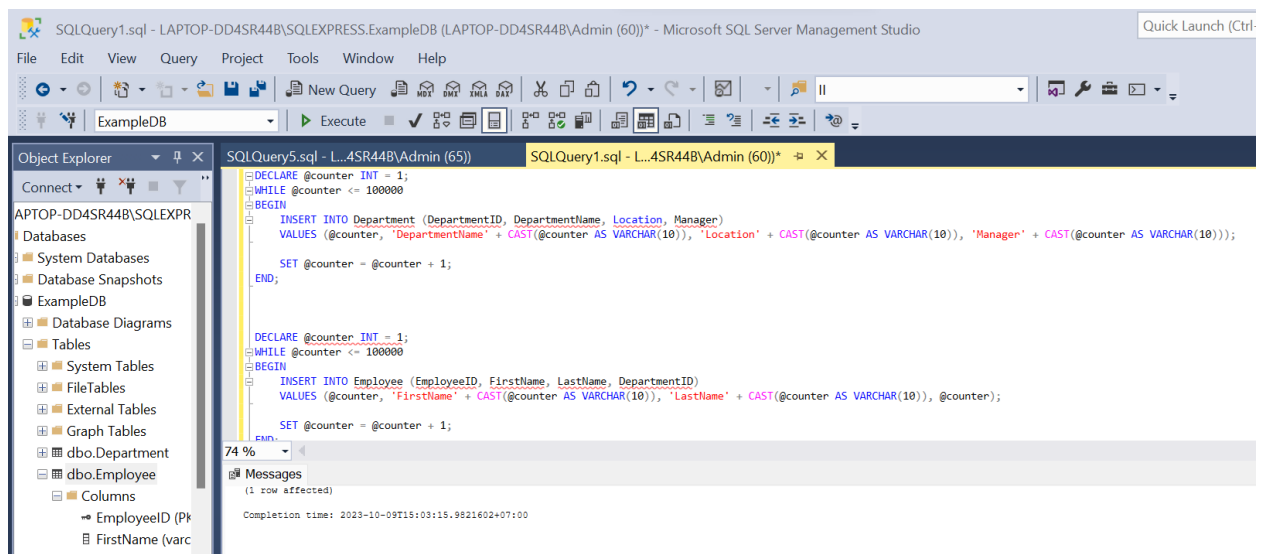


```
-- Tạo ràng buộc duy nhất
ALTER TABLE Employee
ADD CONSTRAINT UQ_Employee_EmployeeID UNIQUE (EmployeeID);

-- Tạo ràng buộc khóa ngoại
ALTER TABLE Employee
ADD CONSTRAINT FK_Employee_Department FOREIGN KEY (DepartmentID)
REFERENCES Department(DepartmentID);

-- Tạo ràng buộc kiểm tra
ALTER TABLE Employee
ADD CONSTRAINT CK_Employee_DepartmentID CHECK (DepartmentID != 1);
```

- Thêm vào các bảng dữ liệu mẫu bằng vòng lặp for (khoảng 100000 dòng)

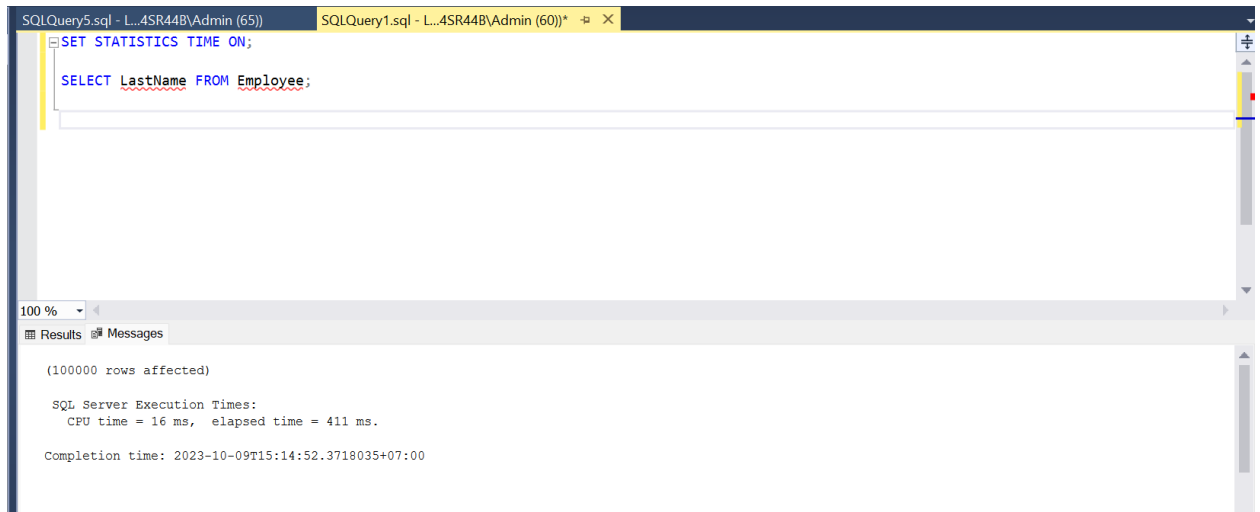


```
DECLARE @counter INT = 1;
WHILE @counter <= 100000
BEGIN
    INSERT INTO Department (DepartmentID, DepartmentName, Location, Manager)
    VALUES (@counter, 'DepartmentName' + CAST(@counter AS VARCHAR(10)), 'Location' + CAST(@counter AS VARCHAR(10)), 'Manager' + CAST(@counter AS VARCHAR(10)));
    SET @counter = @counter + 1;
END;

DECLARE @counter INT = 1;
WHILE @counter <= 100000
BEGIN
    INSERT INTO Employee (EmployeeID, FirstName, LastName, DepartmentID)
    VALUES (@counter, 'FirstName' + CAST(@counter AS VARCHAR(10)), 'LastName' + CAST(@counter AS VARCHAR(10)), @counter);
    SET @counter = @counter + 1;
END;
```

Messages  
(1 row affected)  
Completion time: 2023-10-09T15:03:15.9821602+07:00

- Viết câu select đơn giản trên 1 cột (c1) chưa index và đề ý đến thời gian thực hiện T1



```
SQLQuery5.sql - L...4SR44B\Admin (65))  SQLQuery1.sql - L...4SR44B\Admin (60)) * X
```

```
SET STATISTICS TIME ON;  
  
SELECT LastName FROM Employee;
```

100 %

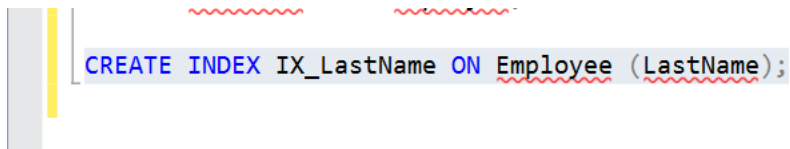
Results Messages

(100000 rows affected)

SQL Server Execution Times:  
CPU time = 16 ms, elapsed time = 411 ms.

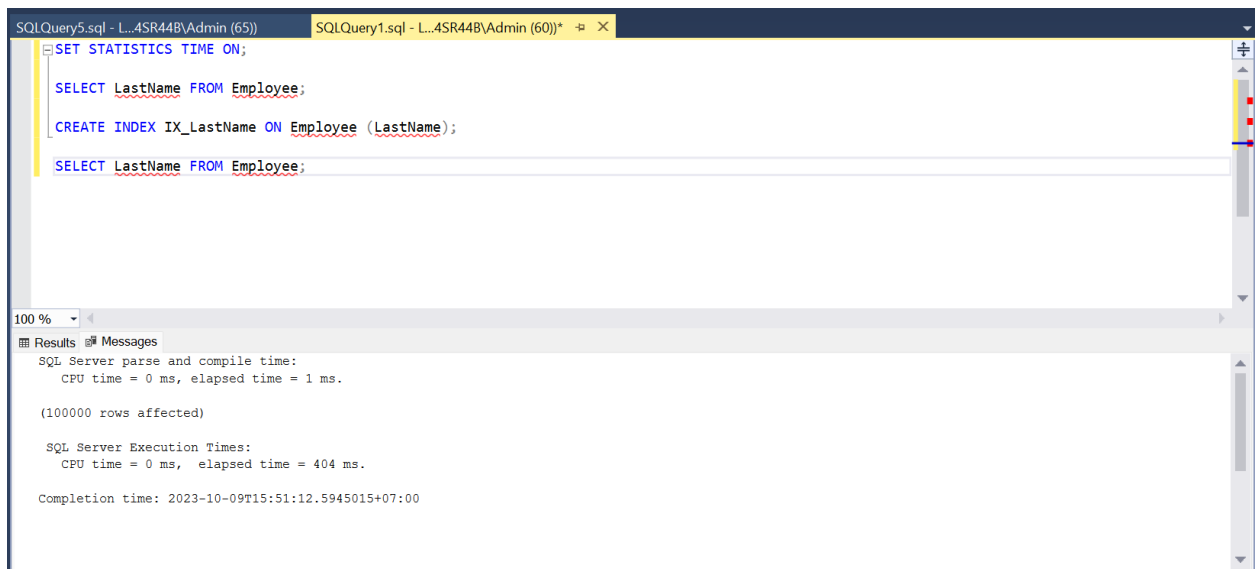
Completion time: 2023-10-09T15:14:52.3718035+07:00

- Cài đặt Index cho cột LastName



```
CREATE INDEX IX_LastName ON Employee (LastName);
```

- Viết câu select đơn giản trên cột LastName đã index và so sánh thời gian thực hiện T2 với T1



```
SQLQuery5.sql - L...4SR44B\Admin (65))  SQLQuery1.sql - L...4SR44B\Admin (60)) * X
```

```
SET STATISTICS TIME ON;  
  
SELECT LastName FROM Employee;  
  
CREATE INDEX IX_LastName ON Employee (LastName);  
  
SELECT LastName FROM Employee;
```

100 %

Results Messages

SQL Server parse and compile time:  
CPU time = 0 ms, elapsed time = 1 ms.

(100000 rows affected)

SQL Server Execution Times:  
CPU time = 0 ms, elapsed time = 404 ms.

Completion time: 2023-10-09T15:51:12.5945015+07:00

**Nhận xét:** Thời gian thực hiện T2 nhanh hơn so với T1