**Tài liệu nghiên cứu SQL Server 2019**

Contents

[Business Intelligence (BI) 1](#_Toc86356310)

[Thực trạng 1](#_Toc86356311)

[Giá trị BI đem lại 2](#_Toc86356312)

[Áp dụng BI 2](#_Toc86356313)

[SQL Server và BI 3](#_Toc86356314)

[Security 4](#_Toc86356315)

[Mã hóa dữ liệu 4](#_Toc86356316)

[Always Encrypted 5](#_Toc86356317)

[Always Encrypted with Secure Enclave 7](#_Toc86356318)

# Business Intelligence (BI)

## Thực trạng

Với các công ty lớn có lượng giao dịch hàng ngày đến hàng triệu giao dịch như

* công ty viễn thông Viettel, Vinaphone,…
* Ngân hàng, tổ chức tín dụng: Vietcombank, Agribank, MBS,…

Làm thế nào để khai thác được giá trị của dữ liệu này

* Để đánh giá tình hình kinh doanh của công ty
* Tăng người dùng sử dụng và cải thiện chất lượng dịch vụ
* Dự báo xu hướng sản phẩm/dịch vụ mà người dùng sẽ chọn lựa
* …..
* 1 trong những giải pháp là áp dụng BI

BI kết hợp việc phân tích/khai thác dữ liệu, trực quan hóa dữ liệu cùng nền tảng hạ tầng để giúp doanh nghiệp

## Giá trị BI đem lại

* Giúp doanh nghiệp nắm được tình hình kinh doanh
* Giảm thiểu chi phí, tối ưu hóa quy trình
* Phân tích thị trường, hành vi của khách hàng, từ đó đưa ra các quyết định nhanh chóng thích ứng với xu hướng của thị trường và người dùng

## Áp dụng BI

Để việc áp dụng BI hiệu quả hơn, bên cạnh hệ thống dữ liệu giao dịch (đã có), họ hướng đến xây dựng 1 hệ thống dữ liệu chuyên dụng cho BI (như Data warehouse/Data mart). 1 số khác biệt cơ bản giữa 2 hệ thống như sau

|  |  |
| --- | --- |
| Hệ thống dữ liệu giao dịch | Hệ thống dữ liệu phân tích |
| * Phục vụ các hoạt động kinh doanh của công ty * Thường là hệ thống CSDL quan hệ, lưu trữ dữ liệu có cấu trúc * Dữ liệu được cập nhật liên tục, thường không lưu dữ liệu quá khứ * Cần đảm bảo hiệu năng nhanh nhất để đáp ứng trải nghiệm của nhiều người dùng đồng thời | * Phục vụ việc phân tích, khai thác giá trị dữ liệu * Lưu cả dữ liệu cấu trúc và phi cấu trúc * Lưu trữ cả dữ liệu quá khứ, capture lại dữ liệu tại 1 số thời điểm để đánh giá * Chỉ có 1 số nhân viên/đơn vị trong công ty có quyền truy cập |

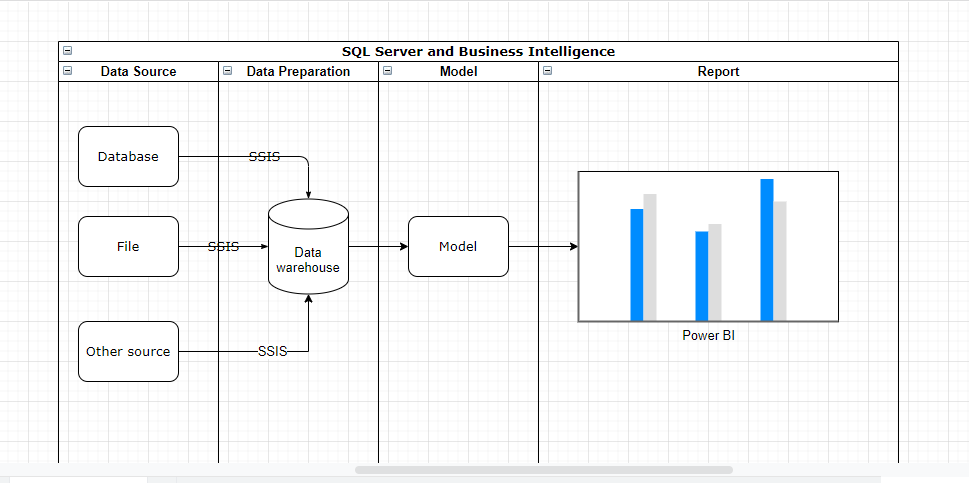
Hệ thống dữ liệu phân tích

1 số thuật ngữ

Data warehouse: hệ thống dữ liệu tập trung, lưu trữ lượng lớn dữ liệu và chuyên dùng cho nhu cầu phân tích dữ liệu

Data mart: cấp độ nhỏ hơn data warehouse, chỉ chứa dữ liệu phục vụ cho 1 đơn vị/phòng ban cụ thể để thuận tiện/nhanh chóng xây dựng và quản trị hơn

## SQL Server và BI

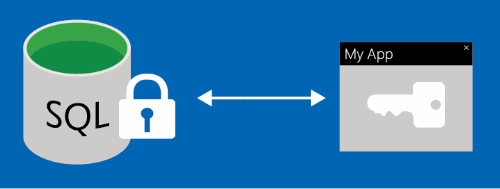


Sơ đồ thiết kế SQL Server và Business Intelligence gồm các phần chính

* Data Source: hệ thống Database/file dùng cho hệ thống giao dịch/quản lý
* Data Preparation: Data warehouse/Data mart dùng để lưu trữ dữ liệu phục vụ cho việc phân tích, thống kê
* Model: 1 mẫu dữ liệu để đáp ứng được yêu cầu cụ thể
* Report: Trực quan hóa dữ liệu dưới dạng báo cáo, dashboard, biểu đồ
* Xây dựng thử nghiệm 1 flow như trên để demo đồ án

# Security

## Mã hóa dữ liệu



Mã hóa dữ liệu giúp bảo vệ dữ liệu nhạy cảm ( như số thẻ tín dụng, số tài khoản ngân hàng, ….) khỏi những rủi ro

* Hacker lấy cắp dc dữ liệu
* Leak dữ liệu bởi những cá nhân trong nội bộ có ý đồ xấu
* Người dùng muốn thuê DB Admin để quản trị DB, nhưng ko muốn họ có thể xem được những dữ liệu nhạy cảm

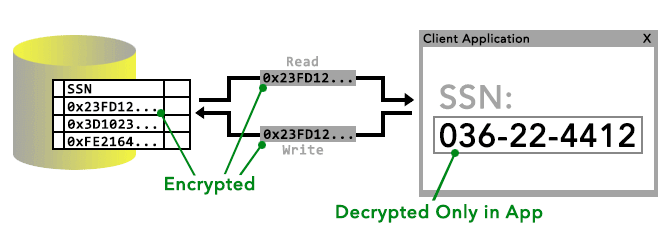
Một số phương thức mã hóa dữ liệu trong SQL Server

* TDE: mã hóa toàn bộ Database
* Always Encrypted: mã hóa 1 số cột xác định trên bảng dữ liệu

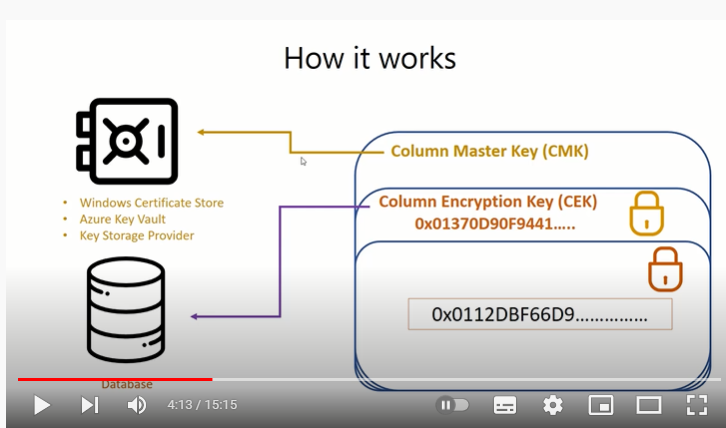
### Always Encrypted

Tính năng always encrypted trong SQL Server (đc giới thiệu ở SQL Server 2016)

* Dữ liệu luôn luôn được mã hóa, cụ thể:
  + dữ liệu trên bảng ở Server đã được mã hóa
  + Việc mã hóa/giải mã đều được thực hiện ở phía client, kể cả DBA cũng ko thể giải mã dc dữ liệu
  + Truy vấn Client gửi lên Server được mã hóa tham số
  + dữ liệu Server trả về client được mã hóa

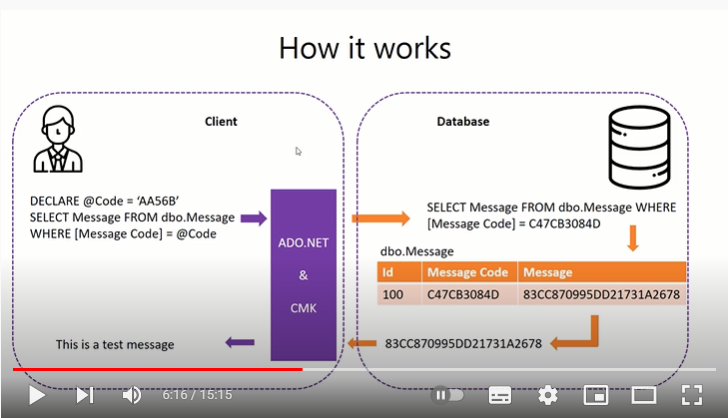


* mô tả dữ liệu luôn luôn đc mã hóa, chỉ đến khi dc giải mã ở app client



**Cách thực hiện mã hóa dữ liệu**

* Xác định các cột nào trên bảng cần dc mã hóa
* Tạo column encryption key (CEK): key để mã hóa các cột dữ liệu (lưu trên DB SQL Server)
* Tạo column master key CMK: key để bảo vệ CEK ( CMK lưu ở 1 vùng khác tách biệt với DB, ví dụ như Window Certificate Store, Azure,...)
* Chỉ client nào có CMK mới có thể giải mã được dữ liệu đã được mã hóa bằng CEK



* Dữ liệu trên DB đã được mã hóa ( cột message code và message)
* Client thực hiện gửi request lên DB để truy vấn dữ liệu (lọc code = AA56B)
* DB Server sẽ nhận được câu truy vấn đã được mã hóa
* DB tìm kiếm dữ liệu theo code dc mã hóa đó, trả về cho client dữ liệu đã dc mã hóa
* Client có CMK sẽ giải mã đc dữ liệu dạng text ban đầu
* Từ client gửi request đến server, toàn bộ thông tin truy vấn và dữ liệu nhận được đều dc mã hóa, ko ai có thể đọc đc kể cả DB admin
* Chỉ client có CMK mới có thể giải mã dc dữ liệu

**Nhược điểm của phương pháp mã hóa Always Encrypted**

Do phương pháp này dữ liệu được mã hóa trên DB, chỉ client có CMK mới có thể giải mã

* client phải truy vấn theo điều kiện chính xác tuyệt đối mới có thể lấy được dữ liệu theo mong muốn (ko thể thực hiện các truy vấn tìm kiếm theo mẫu pattern matching như like,. ..)
* Phải truy vấn toàn bộ dữ liệu => ảnh hưởng đến hiệu năng khi truy vấn đến các bảng dữ liệu lớn

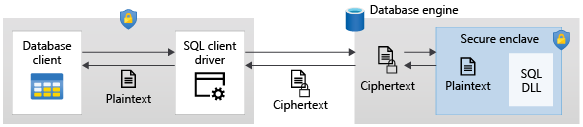
Always Encrypted suffered from severe limitations and did not achieve wide acceptance and deployment. The types of SQL queries and operations you could perform were minimal. You could not do basic SQL query operations that most businesses rely on. So Always Encrypted has not been deployed much.

SQL Server 2019 khắc phục vấn đề trên với phương thức bổ sung Always Encrypted with Secure Enclave

### Always Encrypted with Secure Enclave

Tính năng này được giới thiệu trong SQL Server 2019

Secure Enclave: 1 vùng bảo mật trên hệ thống, tại đây sẽ xử lý việc truy vấn dữ liệu và mã hóa. Vùng này giống như 1 hộp đen bảo mật (secure black box)



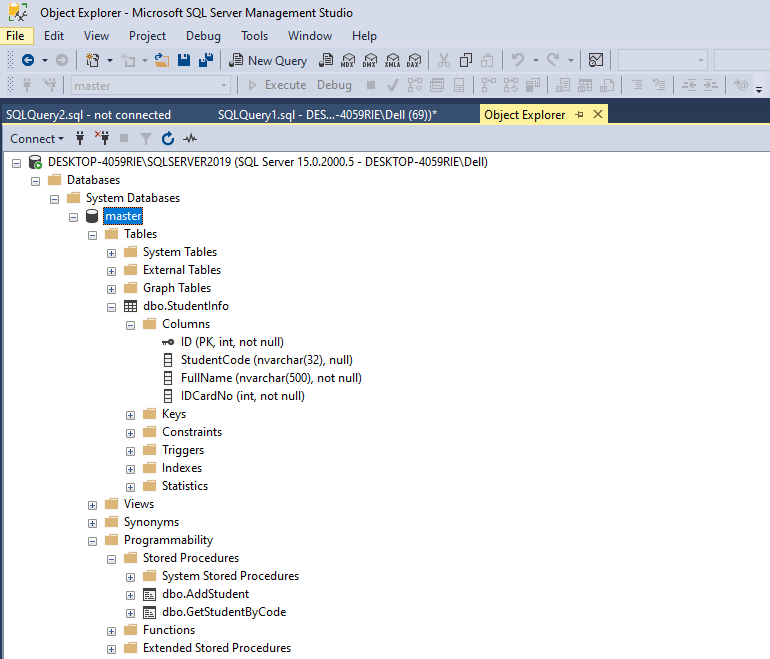
* Client gửi truy vấn đến Server (thông qua kênh bảo mật)
* Truy vấn được mã hóa khi gửi lên Server
* Server xử lý câu truy vấn bên trong Secure enclave
* Secure enclave tính toán dữ liệu và mã hóa kết quả trả về
* Server gửi kết quả đc mã hóa về client
* Client giải mã kết quả nhận đc

## Demo tính năng Always Encryted trong SQL Server 2019

Để demo tính năng, ta xây dựng như sau:

* Database Server:
  + B1: tạo khóa bảo mật (CEK và CMK)
  + B2: tạo bảng chứa cột cần mã hóa dữ liệu (theo tên khóa bảo mật ở trên)
  + B3: Tạo các procedure để truy vấn và thêm mới dữ liệu
* Ứng dụng Win form: là 1 app ở phía Client để tra cứu dữ liệu đã được mã hóa
  + B4: Thiết kế form giao diện
  + B5: Xử lý các chức năng tương ứng khi tương tác với form (sẽ gọi đến các procedure trên Server)
  + B6: Hiển thị dữ liệu đã được giải mã từ DB trả về

### Thiết kế Database



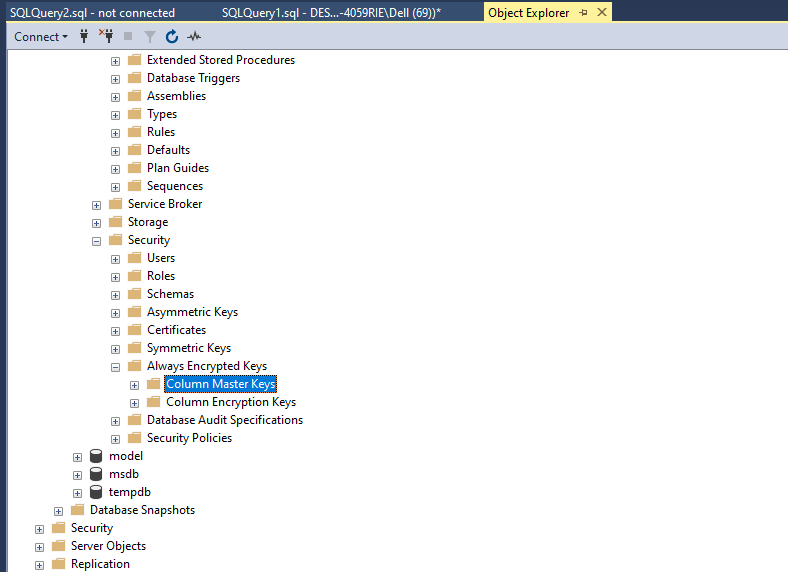
Gồm có

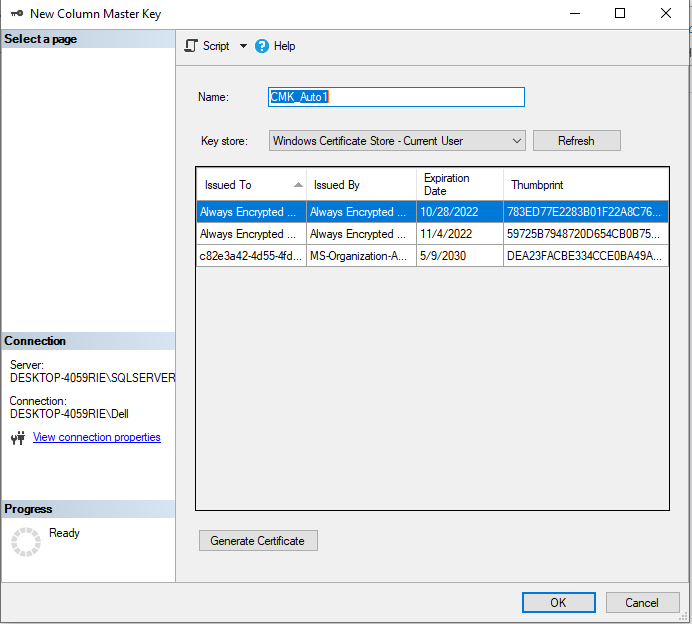
* 1 bảng StudentInfo: lưu thông tin cá nhân của sinh viên
  + Mã số SV
  + Họ tên
  + Số CMT
* 2 PROCEDURE
  + AddStudent: để thêm sinh viên mới
  + GetStudentByCode: lấy thông tin sv dựa theo tham số mã số SV

**Bước 1: Thiết lập khóa bảo mật**

* Tạo khóa CMK

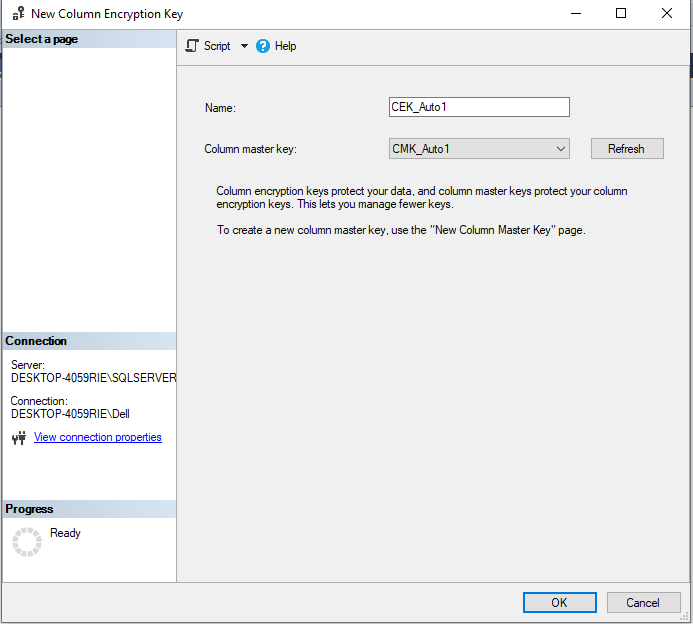
Trong mục Security, chuột phải vào mục Column Master Keys, chọn New Column Master Key





Nhập tên (ví dụ CMK\_Auto1) rồi ấn OK

* Tạo khóa CEK: Trong mục Security, chuột phải vào mục Column Encrytion Keys, chọn New Column Encryption Key
* Nhập tên (ví dụ CEK\_Auto1), chọn khóa master key đã tạo ở trên (CMK\_Auto1) rồi ấn OK



**Bước 2: Tạo bảng**

Câu lệnh tạo bảng

|  |
| --- |
| CREATE TABLE StudentInfo  (  ID INT IDENTITY(1,1) PRIMARY KEY,  StudentCode NVARCHAR(32),  FullName NVARCHAR(500) COLLATE Latin1\_General\_BIN2  ENCRYPTED WITH  (  ENCRYPTION\_TYPE = DETERMINISTIC,  ALGORITHM = 'AEAD\_AES\_256\_CBC\_HMAC\_SHA\_256',  COLUMN\_ENCRYPTION\_KEY = CEK\_Auto1  ) NOT NULL,    IDCardNo INT  ENCRYPTED WITH  (  ENCRYPTION\_TYPE = RANDOMIZED,  ALGORITHM = 'AEAD\_AES\_256\_CBC\_HMAC\_SHA\_256',  COLUMN\_ENCRYPTION\_KEY = CEK\_Auto1  ) NOT NULL  );  GO |

**Bước 3: Tạo procedure**

|  |
| --- |
| CREATE PROCEDURE dbo.AddStudent  @StudentCode NVARCHAR(32),  @FullName NVARCHAR(500),  @IDCardNo INT  AS  BEGIN  INSERT dbo.StudentInfo(StudentCode, FullName, IDCardNo) SELECT @StudentCode, @FullName, @IDCardNo;  END  GO  CREATE PROCEDURE dbo.GetStudentByCode  @StudentCode NVARCHAR(32)  AS  BEGIN  SELECT ID, StudentCode, FullName, IDCardNo  FROM dbo.StudentInfo  WHERE StudentCode = @StudentCode;  END  GO |

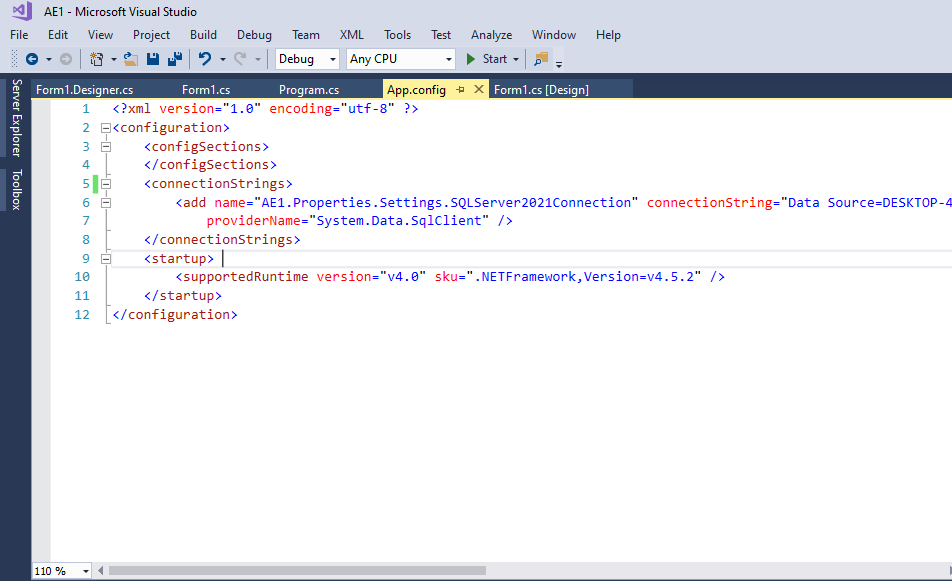
**Bước 4: Thiết kế 1 ứng dụng Win form**

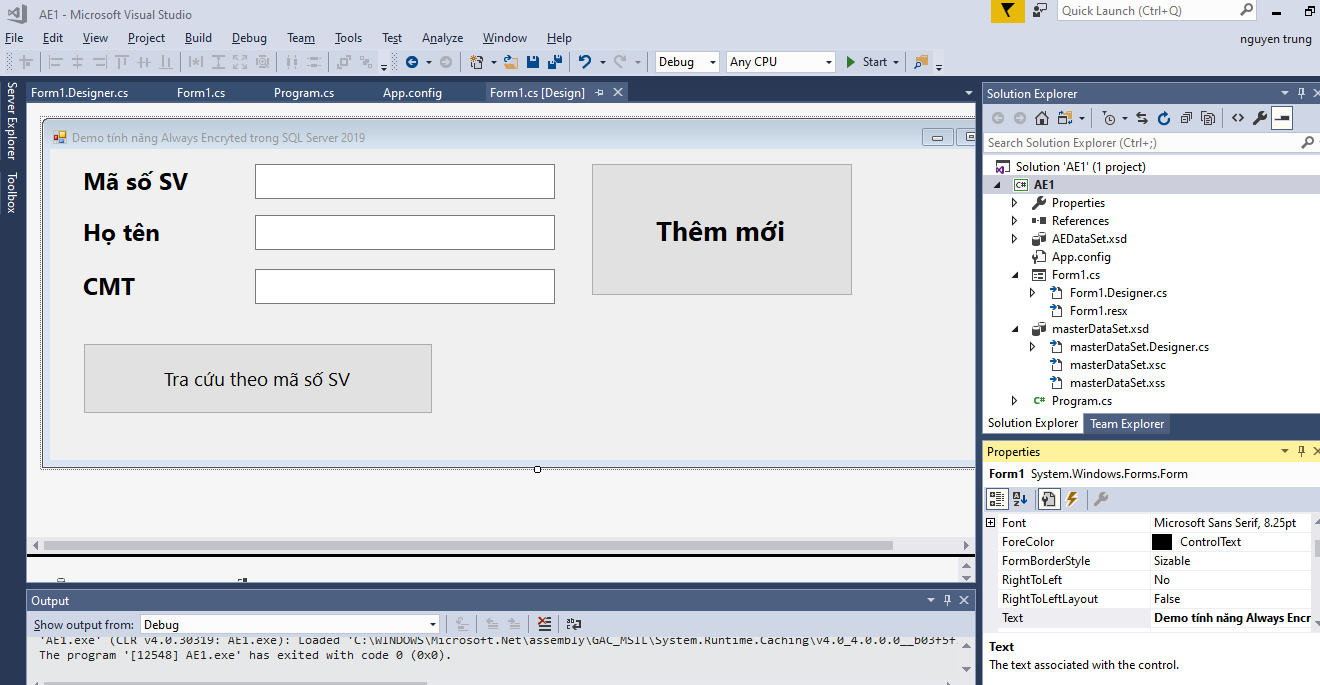
**Note:**

**Chỉ cần open project đã code xong, chỉnh sửa thông tin kết nối đến Database ở file App.config**

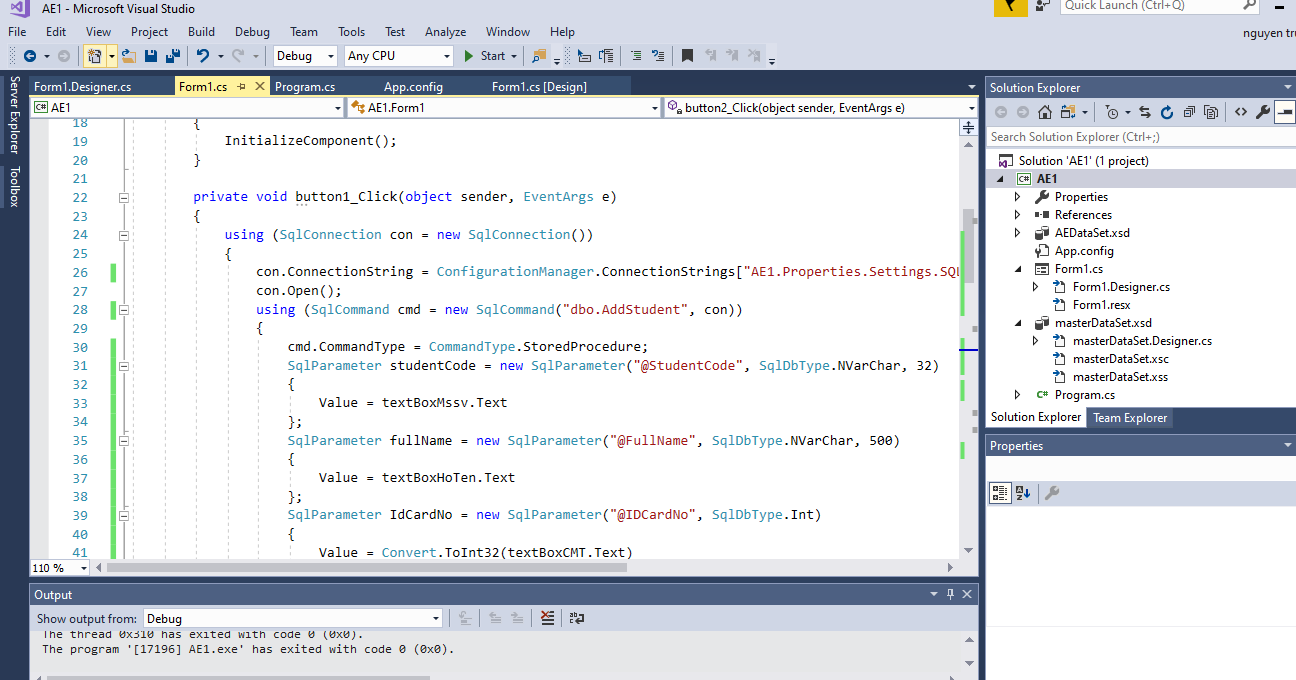
**Sau đó ấn demo để chạy thử nghiệm**

**File App.config (chỉnh trong thẻ connectionString)**

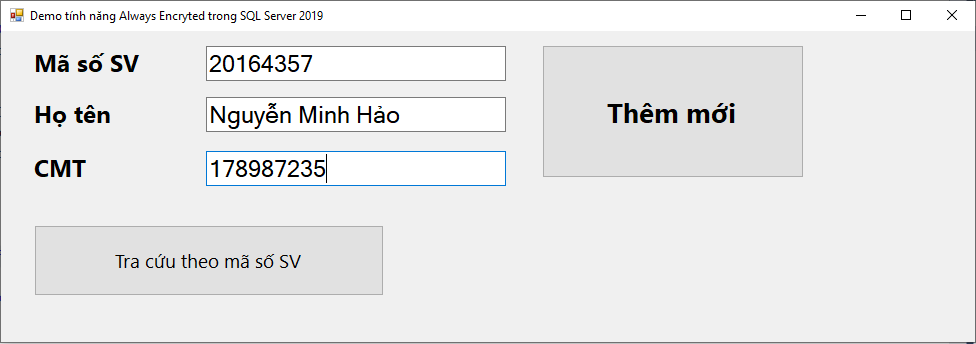


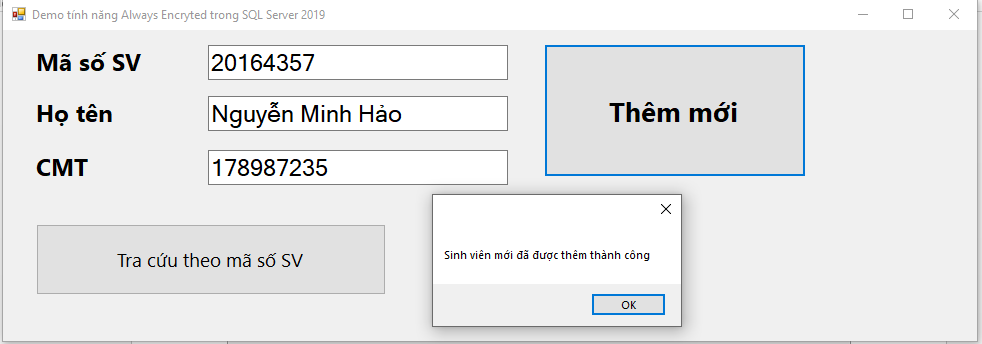


**Bước 5: Xử lý sự kiện khi tương tác form (ấn nút thêm mới/tra cứu)**



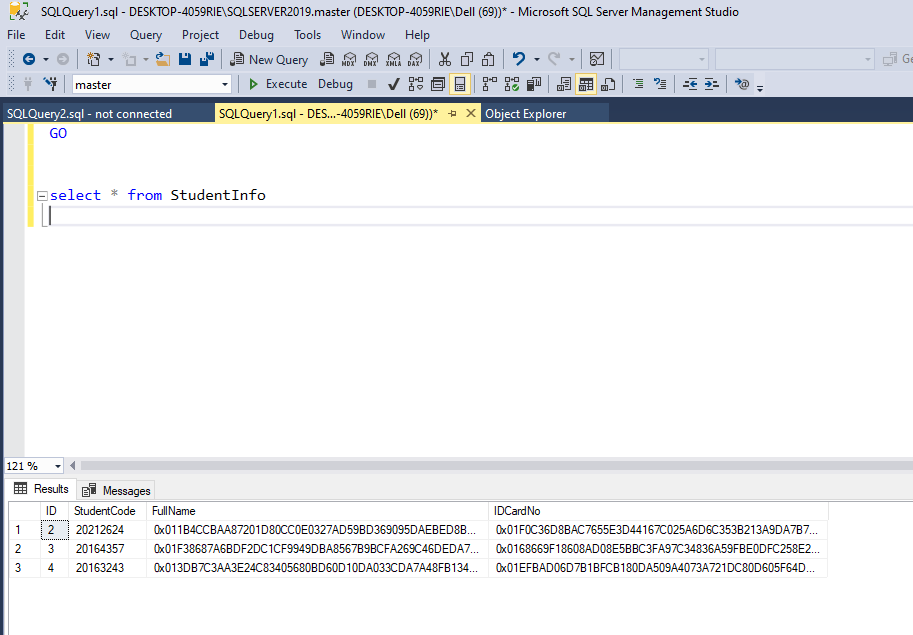
**Bước 6: Demo ứng dụng**



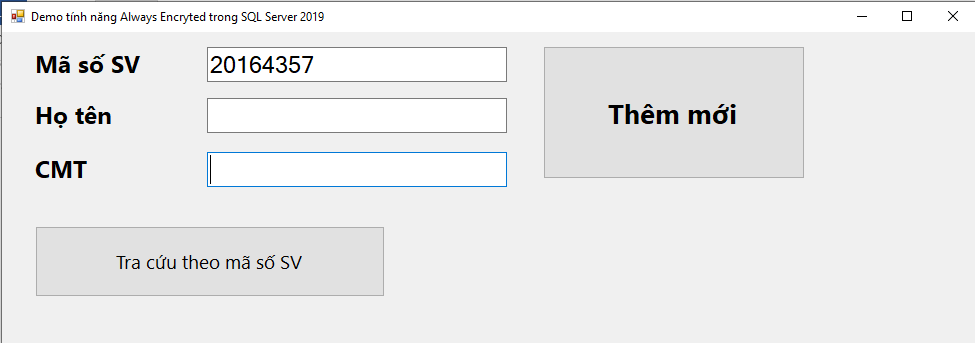


Dữ liệu được insert vào bảng StudentInfo trên Database thành công (bản ghi có studentCode = 20164357)

* 2 cột FullName và IDCardNo đều đc mã hóa để bảo mật dữ liệu (ngay cả user DB Admin cũng ko thể xem dc)



Tra cứu thông tin SV theo mã số: Nhập mã số SV và ấn nút tra cứu theo mã số SV



Kết quả

