Lab: Data Encryption

- This is worth 2 points.
- The due date is tomorrow midnight.
- Use the following naming convention: homework, underscore, last name, first initial, and extension (e.g., Lab_Encrypt_ImG.docx).

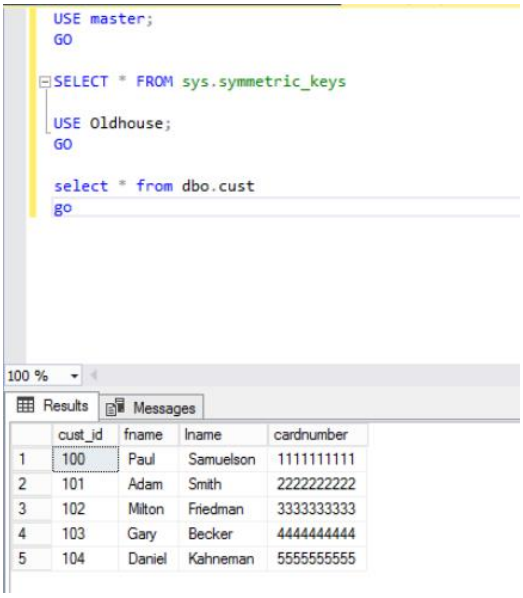
1. Preparation

First, if your SQL Server does not have Oldhouse database, create it using this script: **Oldhouse-Table-Create (Lab).sql**.

Next, perform the lab using this script: **Encryption-Cert (Lab).sql**.

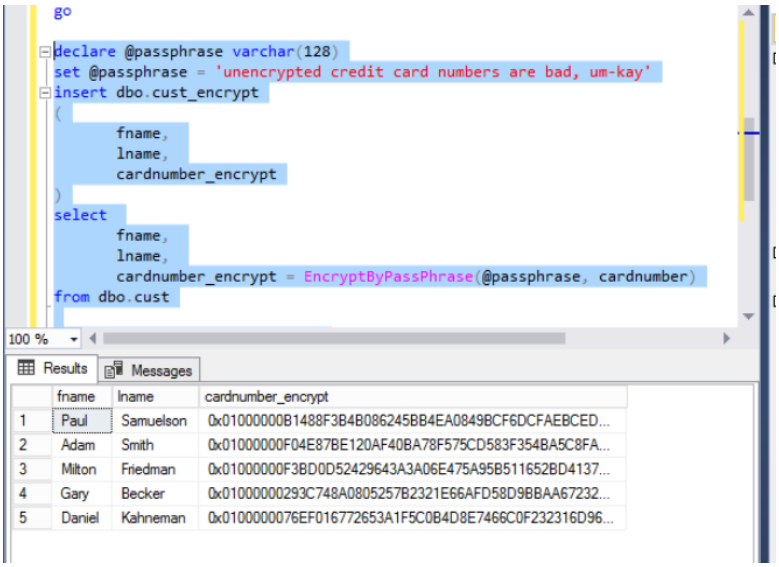
2. Deliverables

```
-- Display the original table
select * from dbo.cust
go
/* Task #1: Show the original table in a screen shot. */
```



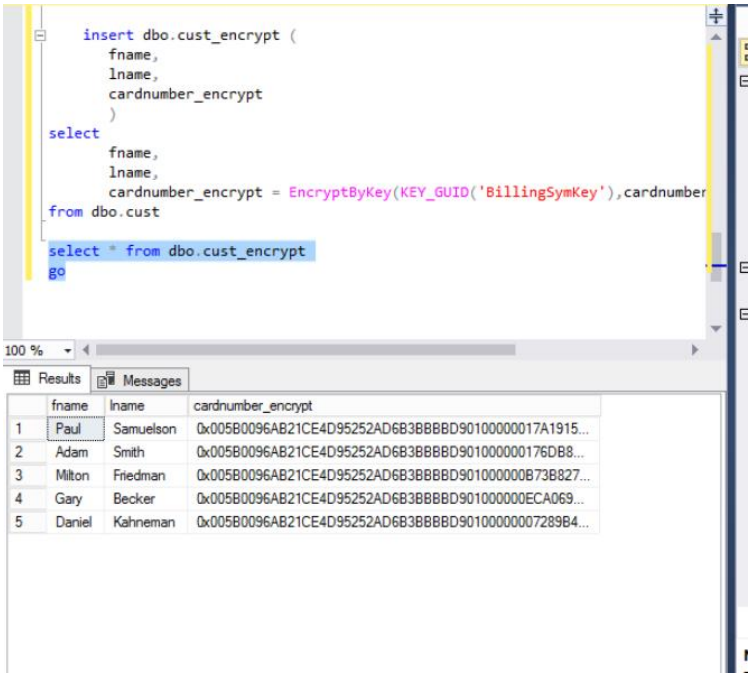
	cust_id	fname	lname	cardnumber
1	100	Paul	Samuelson	1111111111
2	101	Adam	Smith	2222222222
3	102	Milton	Friedman	3333333333
4	103	Gary	Becker	4444444444
5	104	Daniel	Kahneman	5555555555

```
-- Display the encrypted table
select * from dbo.cust_encrypt
go
/* Task #2: Show the encrypted table in a screen shot. Also, explain why we need to
change the data type for encryption. */
```



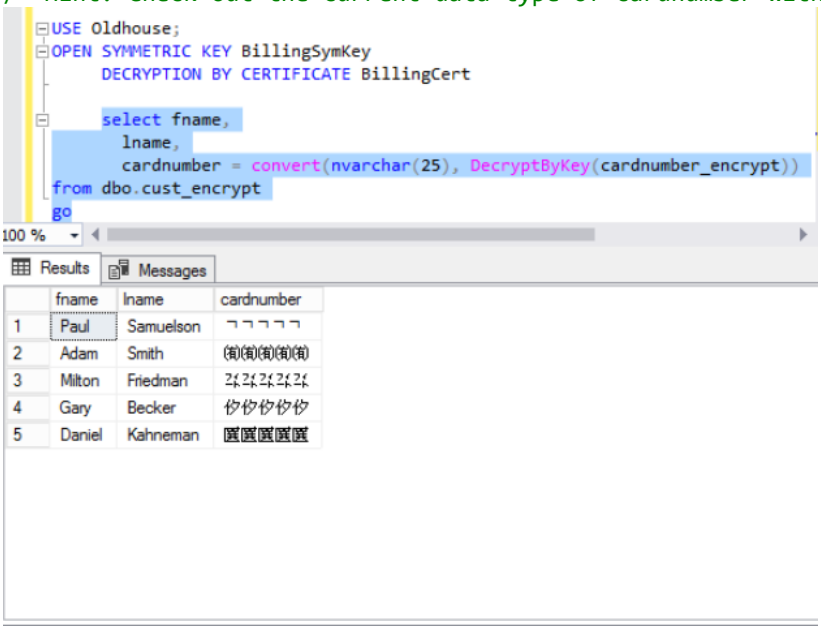
We change the data type for encryption because the data needs to be protected from those who have a chance to enter in the database to take sensitive information.

```
-- Display the encrypted table
select * from dbo.cust_encrypt
go
/* Task #3: Show the encrypted table in a screen shot. Also, explain the encryption
process after Task #2. */
```



The encryption process after task 2 is that we make a certificate using the database, and then creating a symmetric key for it. Then it becomes encrypted by using the certificate. Then the table is emptied by truncating it. Then the table key is decrypted using the certificate, and then the rows of the table are inserted by using the key encrypted by the Certificate.

```
-- Display the decrypted table
select fname,
          lname,
          cardnumber = convert(nvarchar(25), DecryptByKey(cardnumber_encrypt))
from dbo.cust_encrypt
go
/* Task #4: Show the encrypted table in a screen shot. Also, explain the decryption
process after Task #3.      */
/* Did you get the original data back? If not, what's wrong?                        */
/* Hint: Check out the current data type of cardnumber with the original one       */
```



The decryption process is started by selecting the Oldhouse database. Then the symmetric key is opened and used by the certificate for decryption. Once the certificate is used for decryption, the user can then display the table that has been decrypted. I did not get the original data back because the decrypted tables, data type is cardnumber_encrypt which still protects the data. In the original table, the data type for that is a varchar which can be displayed when selected. So, if the decrypted table does not have the correct data type, the data can't be shown in its original form.