МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет

им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации

Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине

“Информационная безопасность”

Тема: “ ШИФРОВАНИЕ SQL SERVER”

Работу выполнил

Студент гр.4235

Белова А.А.

Принял Преподаватель

Кожевников К.Д.

Казань 2024

***Цель***: Изучить теоретическую часть. Рассмотреть основные методы шифрования. Зашифровать базу данных в среде Microsoft SQL Server.

**Задачи:**

* изучить теоретическую часть;
* выполнить практические указания;
* составить отчет по лабораторной работе.

ВЫПОЛНЕНИЕ РАБОТЫ

Рассмотрим пример шифрования с помощью асимметричного ключа:

1. Войдём в нашу базу данных (“hotel”) и создадим для неё новый запрос. Чтобы создать асимметричный ключ, напишем следующий код (Рис. 1):

CREATE ASYMMETRIC KEY SampleAsymKey WITH ALGORITHM = RSA\_2048 ENCRYPTION BY PASSWORD = '<Password>'

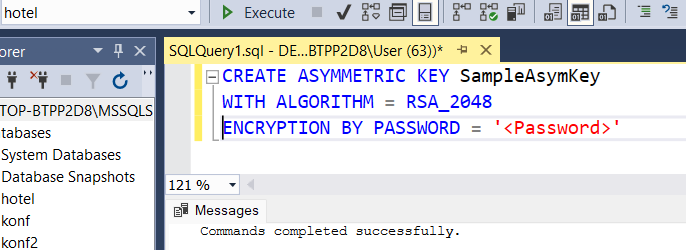


Рисунок 1 – Создание асимметричного ключа.

1. Следующий шаг – объявление переменных. Объявим две переменные (текстовая и бинарная). В текстовой записываем некий текст и выведем его на экран. Зашифруем его с помощью функции ENCRYPTBYASYMKEY и расшифруем с помощью функции DECRYPTBYASYMKEY (Рис. 2):

declare @plaintext nvarchar(60) declare @ciphertext varbinary(256)

--Инициализировать открытый текст set @plaintext='Это простой текст' print @plaintext

--Шифруем

set @ciphertext=ENCRYPTBYASYMKEY (AsymKey\_ID('SampleAsymKey'),@plaintext) print @ciphertext

--Расшифровка

set @ciphertext=DECRYPTBYASYMKEY(AsymKey\_ID('SampleAsymKey'),@ciphertext,N'<Password>') print cast (@plaintext as nvarchar(max))

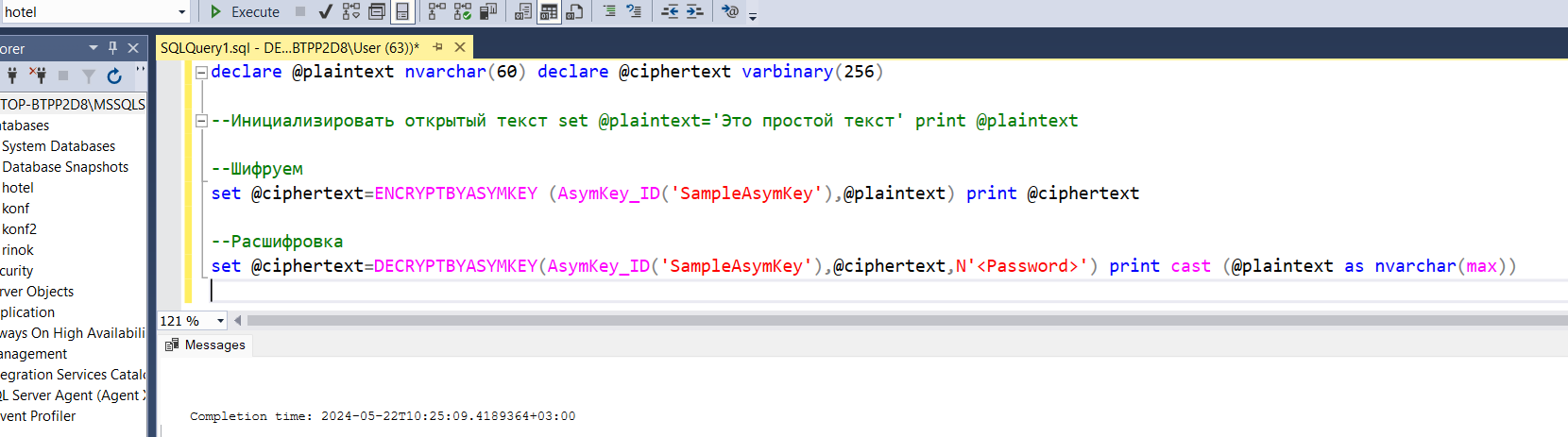


Рисунок 2 – Шифрование и дешифрование текста.

1. Чтобы удалить асимметричный ключ введите следующий запрос (Рис. 3):

DROP ASYMMETRIC KEY SampleAsymKey

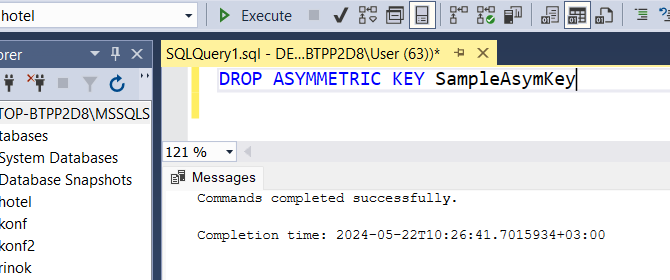


Рисунок 3 – Удаление асимметричного ключа.

Рассмотрим пример использования функций для управления сертификатами:

Вначале создадим папку (Cerf), где будут храниться сертификаты (Рис. 4)

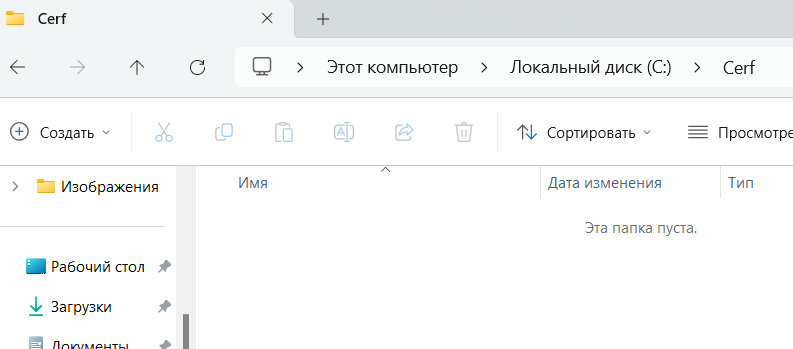


Рисунок 4 – Папка для хранения сертификатов.

Создаём сертификат в нашей базе данных, который шифруется с помощью пароля, указываем тему сертификата и время действия сертификата. Экспортируем его в файловую систему и указываем приватный ключ. Задаём пароль для шифрации и дешифрации (Рис.5):

USE hotel go

--Создаём сертификат

CREATE CERTIFICATE SampleCert ENCRYPTION BY PASSWORD = 'Password'

WITH SUBJECT = 'Sample certificate', EXPIRY\_DATE = '2026-10-10';

--Сохраняем сертификат в виде файла

BACKUP CERTIFICATE SampleCert

TO FILE='C:\Cert\BackupSampleCert.cer'WITH PRIVATE KEY ( FILE='C:\Cert\BackupSampleCert.pvk', ENCRYPTION BY PASSWORD='p@ssword',

DECRYPTION BY PASSWORD='Password')

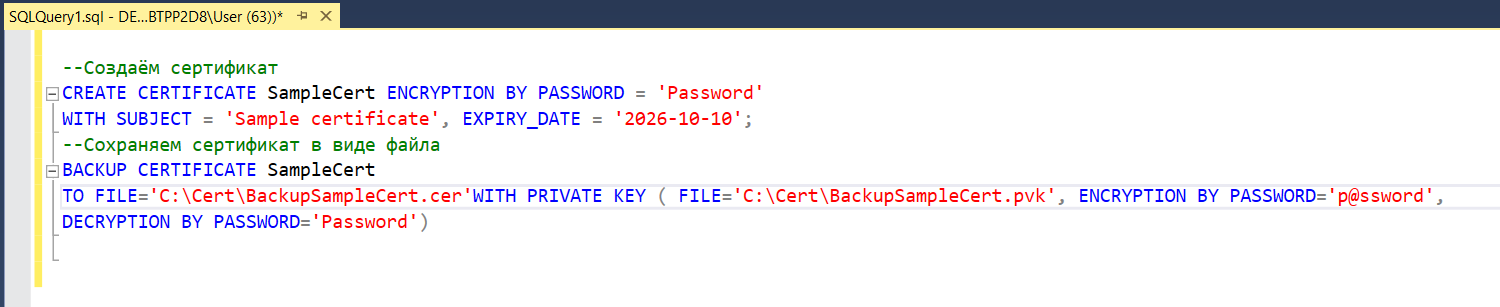


Рисунок 5 – Создание сертификата.

Чтобы удалить сертификат из sql server введите следующий запрос (Рис. 6):

DROP CERTIFICATE SampleCert

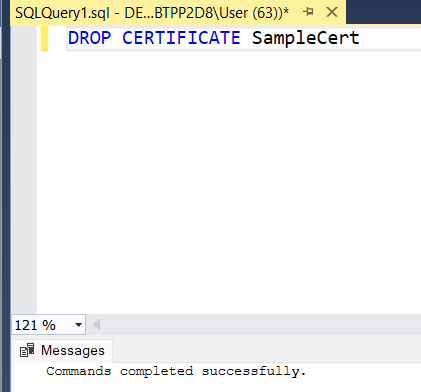


Рисунок 6 – Удаление сертификата.

Чтобы воссоздать из файла введите следующий вопрос (Рис. 7):

CREATE CERTIFICATE SampleCert

FROM FILE='C:\Cert\BackupSampleCert.cer' WITH PRIVATE KEY(

FILE='C:\Cert\BackupSampleCert.pvk', ENCRYPTION BY PASSWORD='p@ssword', DECRYPTION BY PASSWORD='Password')

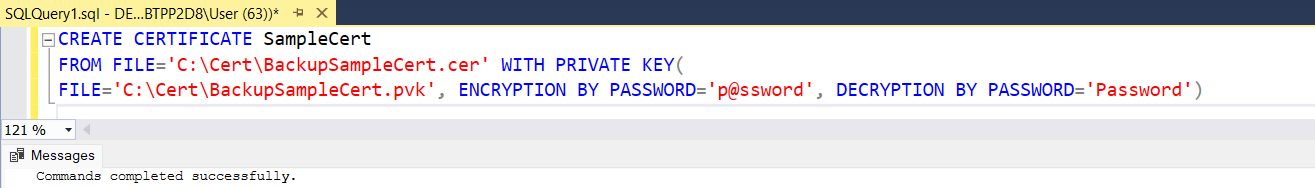


Рисунок 7 – Воссоздание сертификата из созданного файла.

Рассмотрим пример шифрования с помощью симметричного ключа. Будем создавать два ключа первый будет шифровать данные, а второй будет шифровать первый ключ.

1. Сначала создаём второй ключ, а потом создаём ключ, который шифрует данные (Рис. 8):

USE hotel

go

--Создаём симметричный ключ для шифрования другого симметричного ключа

CREATE SYMMETRIC KEY SymKey WITH ALGORITHM =Triple\_DES ENCRYPTION BY PASSWORD='password'

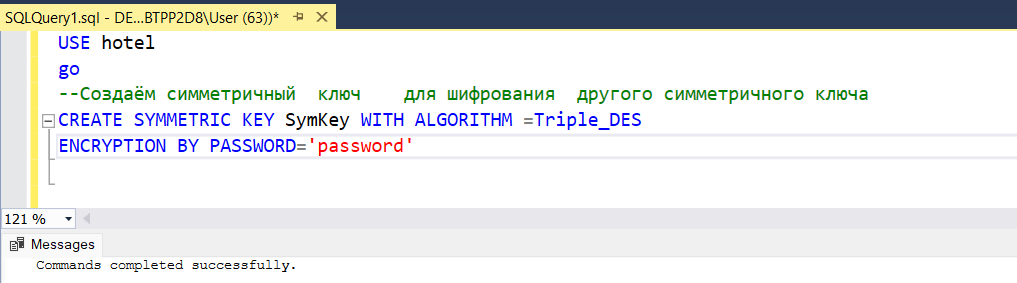


Рисунок 8 –Создание ключа для шифрования другого ключа.

1. Открываем этот ключ:

OPEN SYMMETRIC KEY SymKey DECRYPTION BY PASSWORD='password'

1. Создаём и открываем ключ, который шифрует данные

(Рис.9):

--Создаём симметричный ключ для шифрования данных

CREATE SYMMETRIC KEY SymData

WITH ALGORITHM=Triple\_DES ENCRYPTION BY SYMMETRIC KEY SymKey

--Открываем ключ для шифрования данных OPEN SYMMETRIC KEY SymData DECRYPTION BY SYMMETRIC KEY SymKey

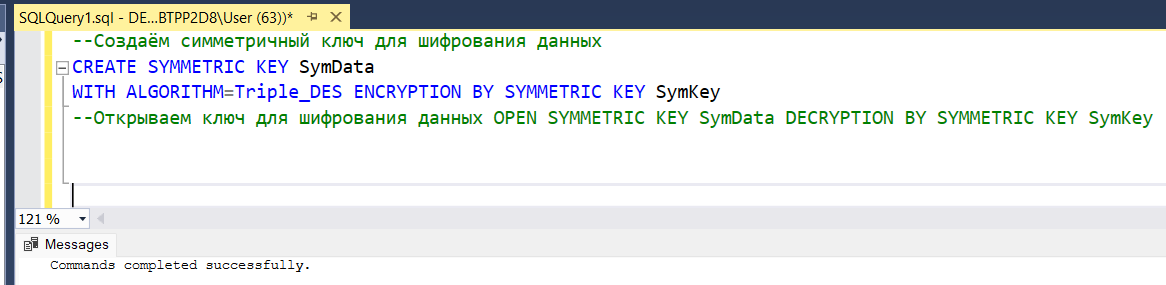


Рисунок 9 –Создание ключа для шифрования данных.

1. Инициализируем текст, для шифрования, шифруем его и сразу же расшифровываем (Рис. 10):

--Инициализируем открытый текст DECLARE @plaintext nvarchar(512) SET @plaintext='Добрый день'

print @plaintext

--Шифруем данные

DECLARE @ciphertext varbinary(1024) SET @ciphertext=ENCRYPTBYKEY (KEY\_GUID('SymData'),@plaintext) print @ciphertext

--Расшифровываем данные

SET @plaintext=CAST

(DECRYPTBYKEY(@ciphertext) as nvarchar(512)) print @plaintext

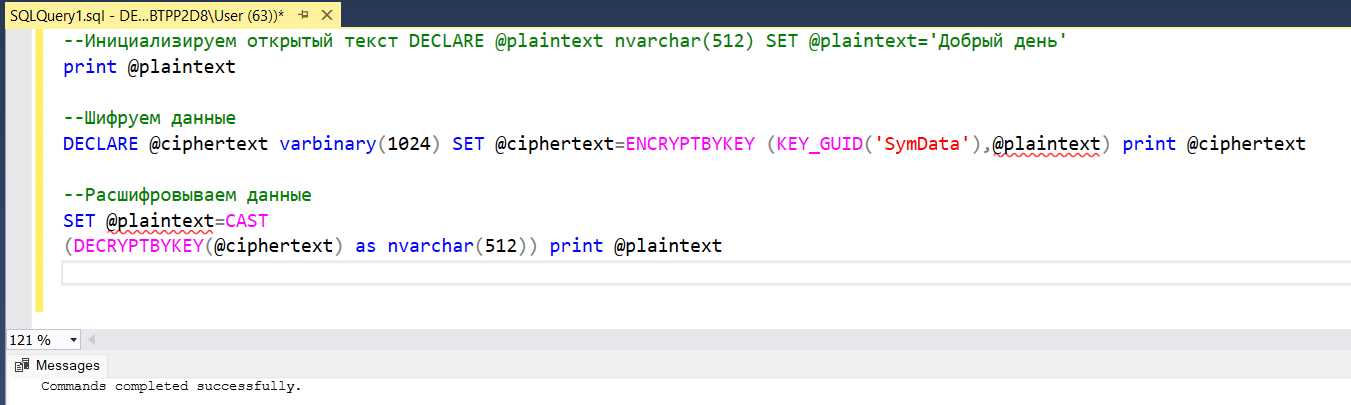


Рисунок 10 – Шифрование и дешифрование текста.

1. Закрываем ключ шифрования данных и ключ шифрования ключа (Рис.11):

--Закрываем ключ шифрования данных

CLOSE SYMMETRIC KEY SymData

--Закрываем ключ шифрования ключа

CLOSE SYMMETRIC KEY SymKey

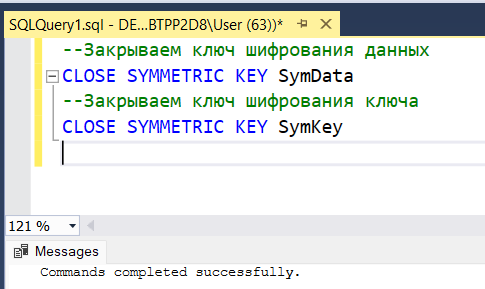


Рисунок 11 –Закрытие симметричных ключей.

Ключевая фраза - это строка или двоичное значение от которого sql server может наследовать симметричный ключ для шифрования данных.

Функция ENCRYPTBYPASSPHRASE - шифрование данных с помощью парольной фразы с использованием алгоритма TRIPLE DES и 128-битного ключа.

DECRYPTBYPASSPHRASE - расшифровывает данные, зашифрованные с помощью парольной фразы.

Пример (Рис. 12)

Declare @plaintext nvarchar(1000), @enc\_text varbinary(2000) SET @plaintext='Я помню чудное мгновенье'

SET @enc\_text=ENCRYPTBYPASSPHRASE('LOM',@plaintext)

SELECT 'Оригинальный текст: ',@plaintext SELECT 'Зашифрованный текст: ',@enc\_text

SELECT 'Расшифровка:', CAST (DECRYPTBYPASSPHRASE

('LOM',@enc\_text) as nvarchar(1000))

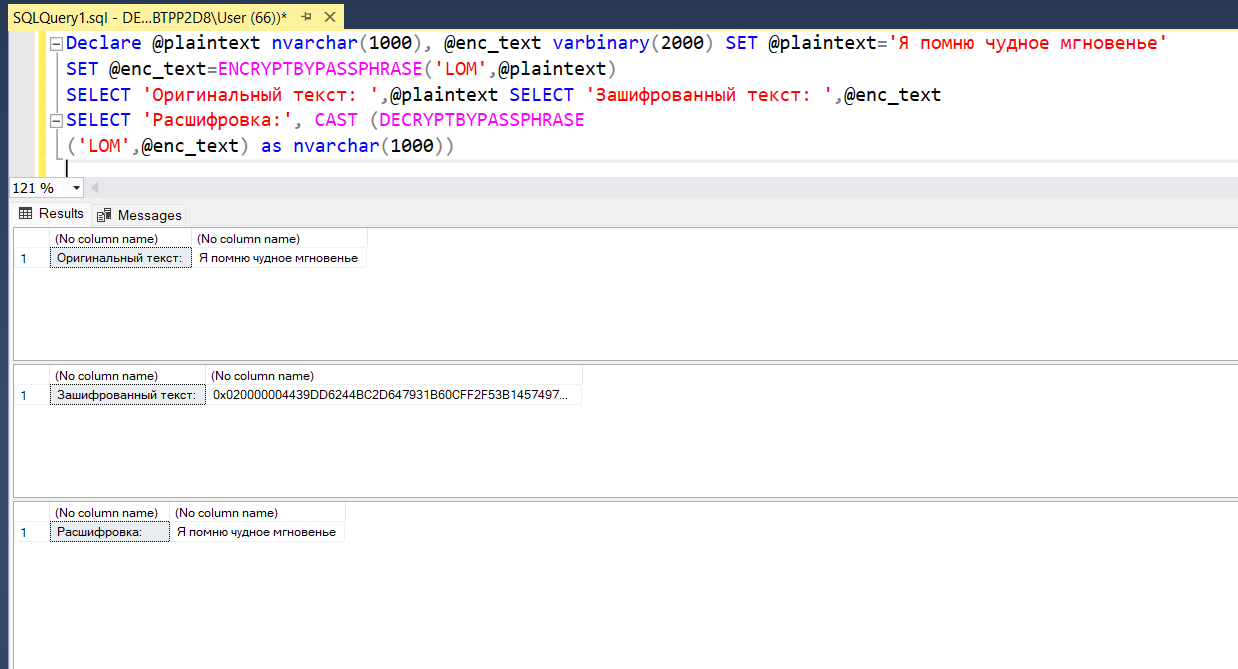


Рисунок 12 – Шифрование с помощью ключевой фразы.

Создадим базу данных, данные в которой шифруются:

1. с использованием симметричного ключа, который будет зашифрован с помощью асимметричного ключа.
2. с использованием симметричного ключа, но на этот раз симметричный ключ шифруется сертификатом.

Создание базы данных:

USE [master] GO

CREATE DATABASE [DB] GO

Запустим код T-SQL для создания таблицы с именем TelephonNumber в базе данных DB:

USE [DB] GO

CREATE TABLE [dbo].[ TelephonNumber] ([PersonID] [int] PRIMARY KEY,

[TelephonNumber] [varbinary](max))GO

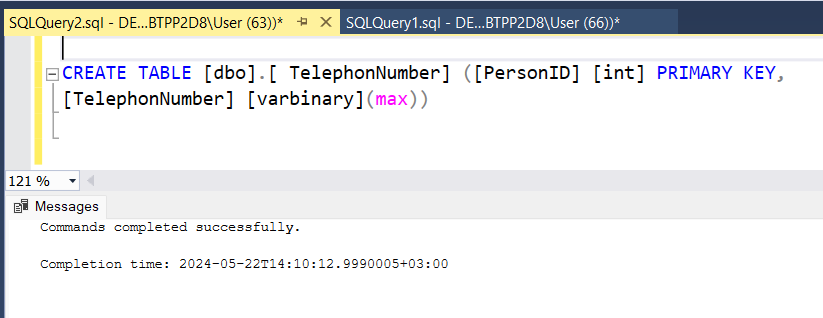


Рисунок 13 – Создание таблицы с названием TelephonNumber.

Используем следующий программный код для создания главного ключа DMK базы данных DB, шифруемого с помощью парольной фразы $tr0nGPa$$w0rd:

USE [DB] GO

CREATE MASTER KEY ENCRYPTION BY PASSWORD ='$tr0nGPa$$w0rd' GO

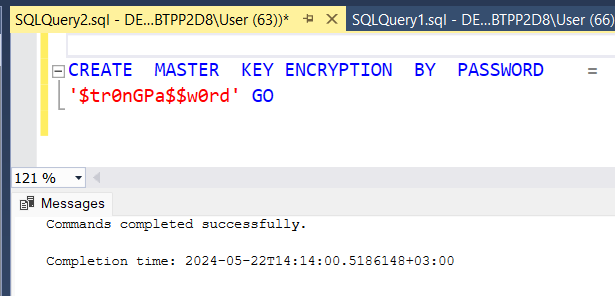


Рисунок 14 – Создание главного ключа DMK.

Необходимо создать асимметричный ключ, зашифровать его парольной фразой $tr0nGPa$$w0rd, создать симметричный ключ и зашифровать симметричный ключ с помощью только что созданного асимметричного ключа. Выполнить эти задачи можно, запустив программный код:

USE [DB] GO

--Создание асимметричного ключа, зашифрованного парольной фразой StrongPa$$w0rd!

CREATE ASYMMETRIC KEY MyAsymmetricKey WITH ALGORITHM = RSA\_2048

ENCRYPTION BY PASSWORD = 'StrongPa$$w0rd!' GO

--Создание симметричного ключа, зашифрованного асимметричным ключом

CREATE SYMMETRIC KEY MySymmetricKey WITH ALGORITHM = AES\_256

ENCRYPTION BY ASYMMETRIC KEY MyAsymmetricKey GO

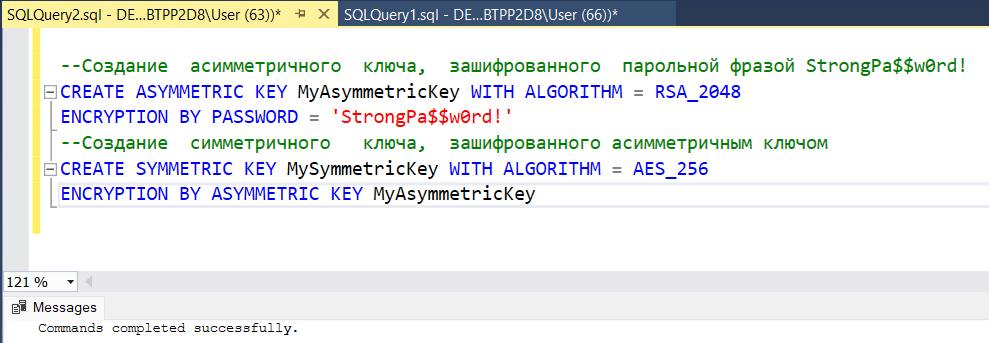


Рисунок 15 – Процесс выполнения.

* 1. Теперь мы можем приступить к шифрованию данных. Для этого необходимо сначала открыть симметричный ключ, только что созданный с помощью команды OPEN SYMMETRIC KEY, за которой следует имя симметричного ключа. Затем указать, что нужно расшифровать его с использованием заданного асимметричного ключа. Программный код выглядит следующим образом:

OPEN SYMMETRIC KEY MySymmetricKey

DECRYPTION BY ASYMMETRIC KEY MyAsymmetricKey WITH PASSWORD = 'StrongPa$$w0rd!'

После выполнения этого кода направляем запрос в представление sys.openkeys, чтобы убедиться, что ключ открыт (Рис.16):

SELECT \* FROM [sys].[openkeys]

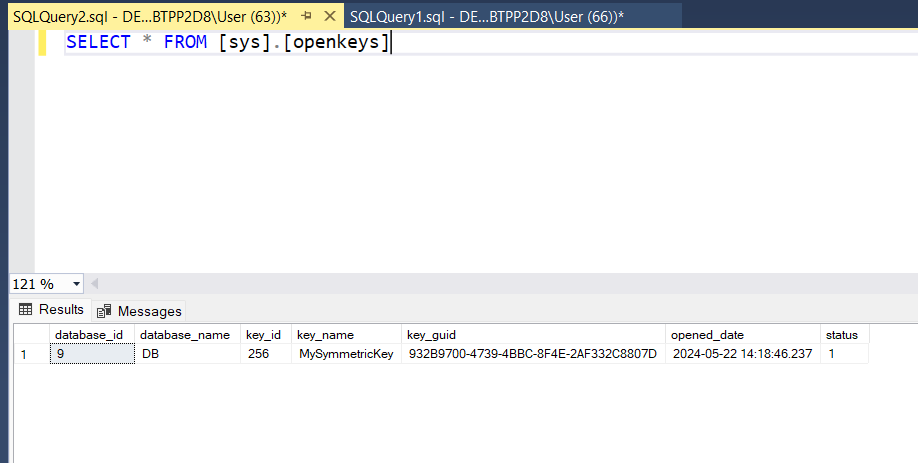


Рисунок 16 – Открытие симметричного ключа.

Введем несколько телефонных номеров в таблицу TelephoneNumber, запустив код:

DECLARE @SymmetricKeyGUID AS [uniqueidentifier]

SET @SymmetricKeyGUID = KEY\_GUID('MySymmetricKey') IF (@SymmetricKeyGUID IS NOT NULL)

BEGIN

INSERT INTO [dbo].[ TelephonNumber]

VALUES (01, ENCRYPTBYKEY(@SymmetricKeyGUID, N'8-761-123-87-63'))

INSERT INTO [dbo].[ TelephonNumber]

VALUES (02, ENCRYPTBYKEY(@SymmetricKeyGUID, N'8-768-765-87-65'))

INSERT INTO [dbo].[ TelephonNumber]

VALUES (03, ENCRYPTBYKEY(@SymmetricKeyGUID, N'8-761-234-11-11'))

END

TRUNCATE TABLE [dbo].[TelephonNumber]

Выведем получившуюся таблицу на экран (Рис. 17):

SELECT \* FROM [dbo].[ TelephonNumber]

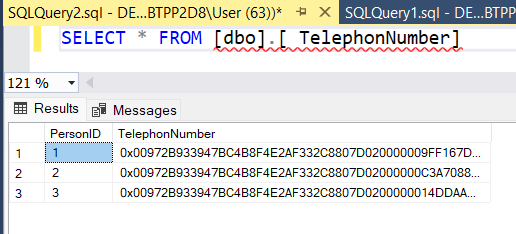


Рисунок 17 – Зашифрованные данные из таблицы TelephoneNumber.

Все данные в столбце TelephoneNumber представлены в двоичном формате. С помощью функции DECRYPTBYKEY можно просмотреть зашифрованные данные (Рис. 18):

SELECT [PersonID],

CONVERT([nvarchar](32), DECRYPTBYKEY(TelephonNumber)) AS [TelephonNumber]

FROM [dbo].[TelephonNumber] GO

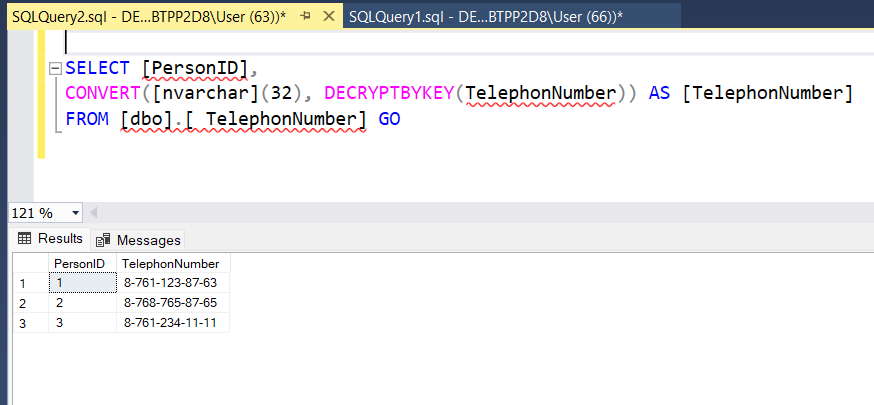


Рисунок 18 – Просмотр зашифрованных данных.

Создадим сертификат с помощью инструкции CREATE CERTIFICATE. Затем создается симметричный ключ, шифруемый сертификатом. Наконец, открыв симметричный ключ, код вставляет три строки в таблицу TelephonNumber:

--Создание сертификата

CREATE CERTIFICATE

[CertToEncryptSymmetricKey]

WITH SUBJECT ='Самозаверяющий сертификат для шифрования симметричного ключа.'

--Создание симметричного ключа, зашифрованного сертификатом

CREATE SYMMETRIC KEY

[SymmetricKeyEncryptedWithCert] WITH ALGORITHM = AES\_256 ENCRYPTION BY CERTIFICATE

[CertToEncryptSymmetricKey]

--Открытие симметричного ключа

OPEN SYMMETRIC KEY

[SymmetricKeyEncryptedWithCert] DECRYPTION BY CERTIFICATE

[CertToEncryptSymmetricKey]

--Усечение таблицы TelephonNumber TRUNCATE TABLE

[dbo].[ TelephonNumber]

--Вставка данных в таблицу DECLARE @SymmetricKeyGUID AS [uniqueidentifier]

SET @SymmetricKeyGUID = KEY\_GUID

('SymmetricKeyEncryptedWithCert')

IF (@SymmetricKeyGUID IS NOT NULL) BEGIN

INSERT INTO [dbo].[ TelephonNumber] VALUES (01, ENCRYPTBYKEY

(@SymmetricKeyGUID, N'8-861-123-87-63'))

INSERT INTO [dbo].[ TelephonNumber] VALUES (02, ENCRYPTBYKEY

(@SymmetricKeyGUID, N'8-868-765-87-65'))

INSERT INTO [dbo].[ TelephonNumber]

VALUES (03, ENCRYPTBYKEY

(@SymmetricKeyGUID, N'8-861-234-11-11')) END

Просмотр зашифрованных данных (Рис. 18):

SELECT \* FROM[dbo].[ TelephonNumber]

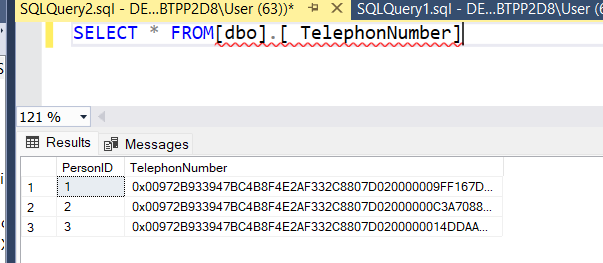


Рисунок 19 – Вывод данных таблицы “TelephonNumber”.

Расшифровка (Рис. 20):

SELECT [PersonID],

CONVERT([nvarchar](32), DECRYPTBYKEY(TelephonNumber)) AS [TelephonNumber]

FROM [dbo].[ TelephonNumber]

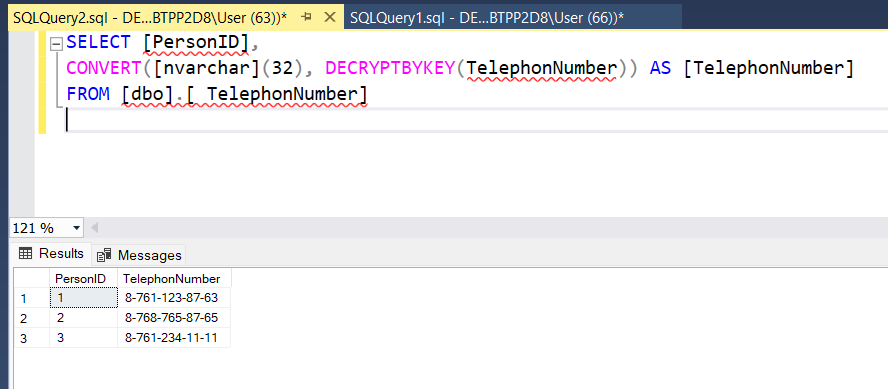


Рисунок 20 – Дешифровка данных таблицы “TelephonNumber”.

Рассмотрим пример прозрачного шифрования данных (Рис.20)

Создаем главный ключ шифрования

CREATE MASTER KEY ENCRYPTION

BY PASSWORD = 'StrongPassword#1';

После того как создали главный ключ, необходимо сделать его резервную копию и поместить резервную копию в надежное место:

BACKUP MASTER KEY TO FILE = 'C:\Cert\MasterBackup.bak' ENCRYPTION BY PASSWORD = 'Password1'

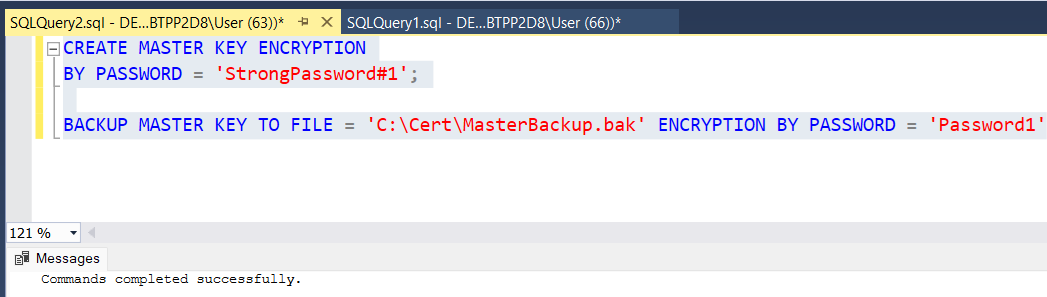


Рисунок 21 – Создание главного ключа и сертификата.

Создание сертификата осуществляется следующим запросом:

CREATE CERTIFICATE Сертификат

WITH SUBJECT ='Сертификат для библиотеки'

Проверка наличия созданного сертификата: select \* from sys.certificates where name='Сертификат'

Создание резервной копии сертификата с закрытым ключом (Рис. 22):

BACKUP CERTIFICATE Сертификат

TO FILE = 'C:\Cert\Certif' WITH PRIVATE KEY

(FILE = 'C:\Cert\PrivateCertif',

ENCRYPTION BY PASSWORD = 'Password#3');

Выполнив всё вышеизложенное приступим к созданию ключа шифрования в нашей базе данных с использование нашего сертификата:

CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES\_128

ENCRYPTION BY SERVER CERTIFICATE Сертификат;

Включаем шифрование для нашей базы данных: ALTER DATABASE Библиотека SET ENCRYPTION ON;

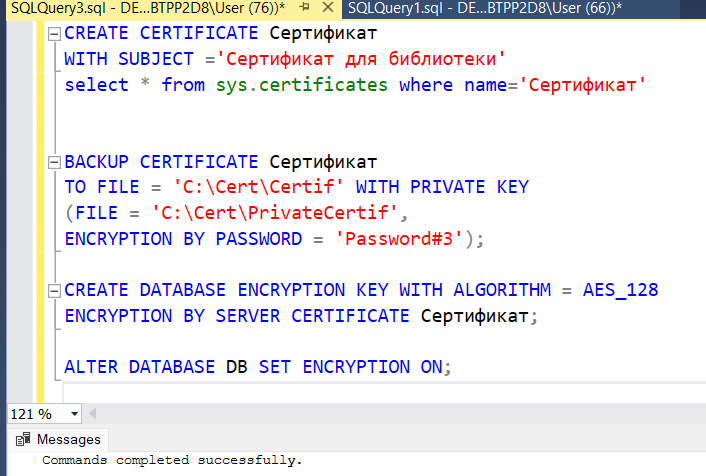


Рисунок 22 – Резервные копии сертификата и главного ключа.

Создание ключа шифрования в нашей базе данных.

Чтобы посмотреть зашифрованные базы данных, нужно ввести следующий запрос (Рис. 23):

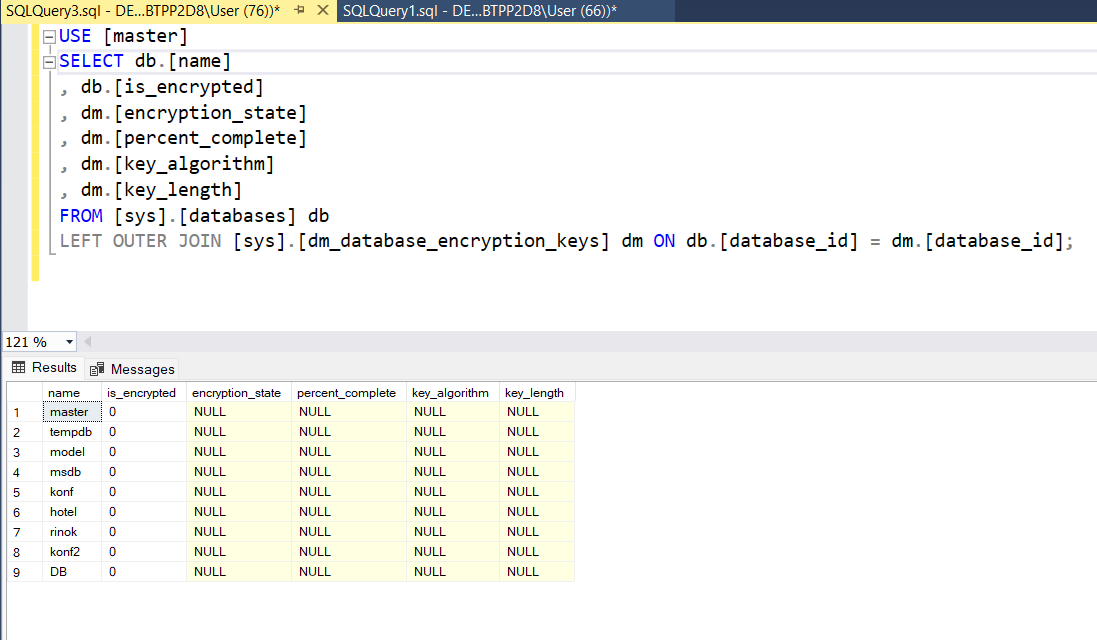


Рисунок 23 – Вывод списка баз данных.

Вывод: мы приобрели навыки использования основных методов шифрования.