**Task Details:**

**You are part of a team developing a new datalake. The Data Engineers implemented a process to retrieve data from several sources. Your task is to create a method to identify PII data coming from these sources and restrict access to the data only to the auditors. How would you implement this process? Describe the process using the cloud provider of your choice.**

Proposed Solution:

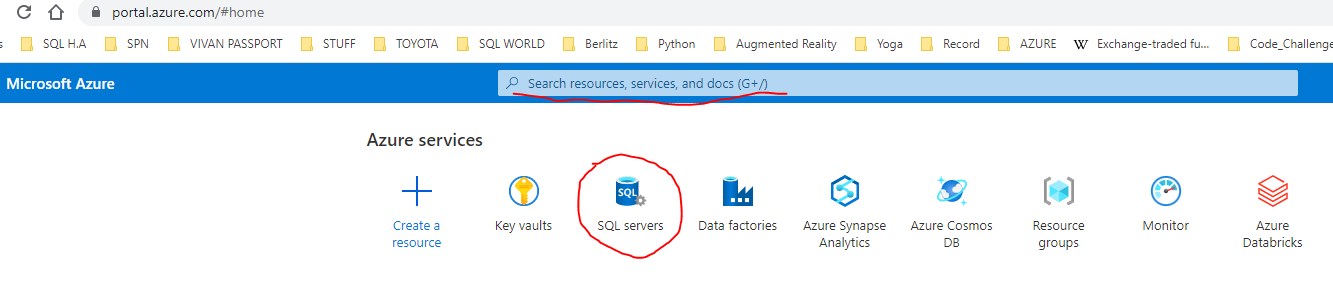
Personally Identifiable information (PII) is any data that can be used to identify an individual such as names, driver’s license number, SSNs, bank account numbers, passport numbers, email addresses and more. Many regulations from GDPR to HIPPA require strict protection of user privacy.

The Azure **PII Detection** skill (Currently in Preview) extracts personally identifiable information from an input text and gives you the option to mask it from that text in various ways. This skill uses the machine learning models provided by [Text Analytics](https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/overview/?WT.mc.id=aiapril-medium-abornst) in Cognitive Services.

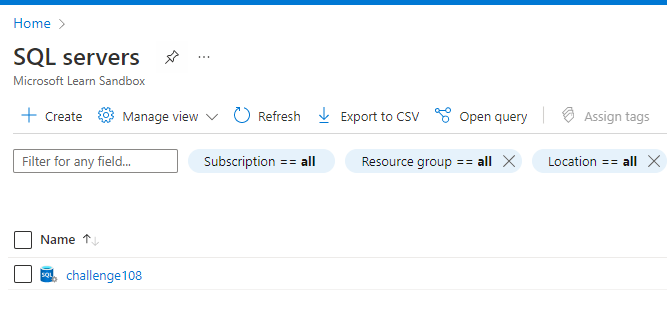
**PII data encryption can be achieved by implementing in SQL server. Below are the steps to encrypt data and to access the same.**

Step 1: Login to Azure port with this link <https://portal.azure.com/#home>

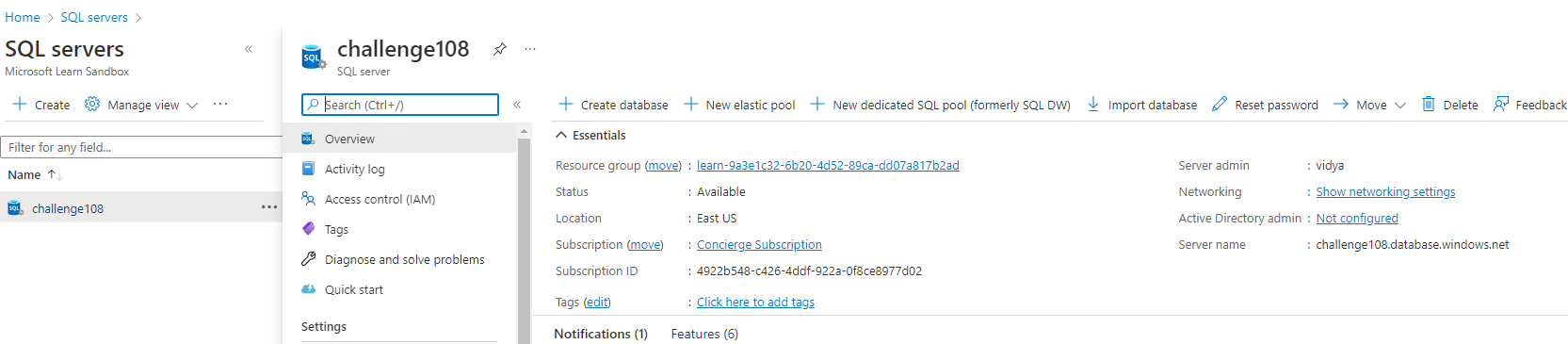
Step 2: Search ‘SQL servers’ search box



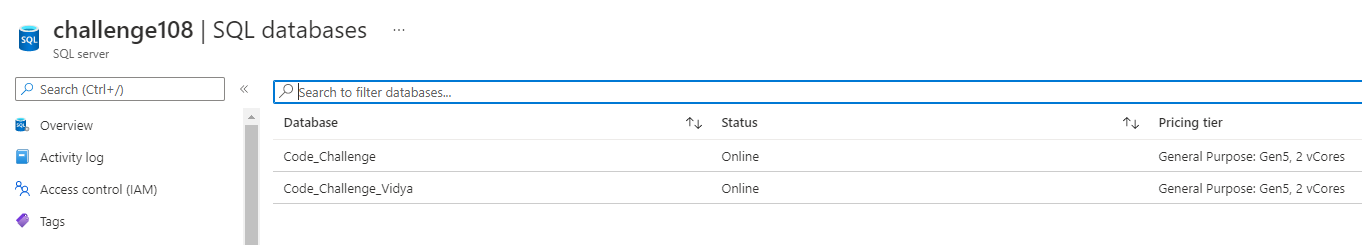
Step 3: Click on ‘SQL servers’ and click on ‘Create’ option



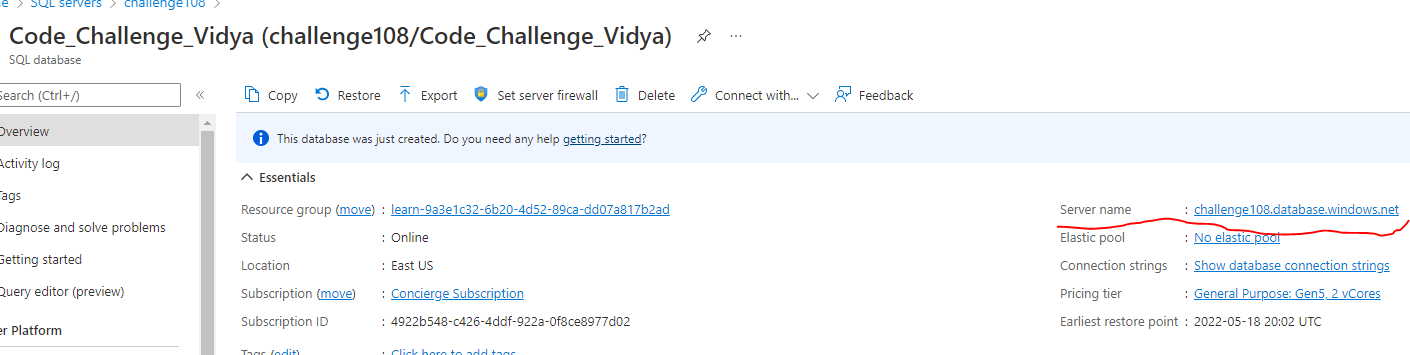
Step 4: Create server by providing all relevant details (server name, username, and password).



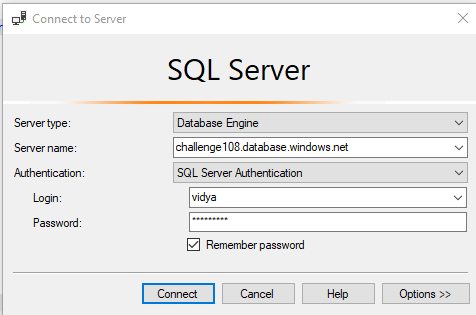
Step 5: Click on ‘Create database’ option and provide all relevant details.



Step 6: Copy server name as shown in below screen



Step 7: Open sql server management studio and connect with server name and login details.



Step 8: Create ‘Employee’ table with below mentioned script.

CREATE TABLE [dbo].[employees](

[employee\_id] [int] IDENTITY(1,1) NOT NULL,

[first\_name] [varchar](20) NULL,

[last\_name] [varchar](25) NOT NULL,

[email] [varchar](100) COLLATE Latin1\_General\_BIN2 ENCRYPTED WITH (COLUMN\_ENCRYPTION\_KEY = [CEK\_Auto1], ENCRYPTION\_TYPE = Deterministic, ALGORITHM = 'AEAD\_AES\_256\_CBC\_HMAC\_SHA\_256') NOT NULL,

[phone\_number] [varchar](20) NULL,

[hire\_date] [date] NOT NULL,

[job\_id] [int] NOT NULL,

[salary] [decimal](8, 2) NOT NULL,

[manager\_id] [int] NULL,

[department\_id] [int] NULL,

PRIMARY KEY CLUSTERED

(

[employee\_id] ASC

)WITH (STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[employees] ADD DEFAULT (NULL) FOR [first\_name]

GO

ALTER TABLE [dbo].[employees] ADD DEFAULT (NULL) FOR [phone\_number]

GO

ALTER TABLE [dbo].[employees] ADD DEFAULT (NULL) FOR [manager\_id]

GO

ALTER TABLE [dbo].[employees] ADD DEFAULT (NULL) FOR [department\_id]

GO

Step 9: Insert data into ‘Employee’ table.

/\*Data for the table employees \*/

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,manager\_id,department\_id) VALUES (100,'Steven','King','steven.king@sqltutorial.org','515.123.4567','1987-06-17',4,24000.00,NULL,9);

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,manager\_id,department\_id) VALUES (101,'Neena','Kochhar','neena.kochhar@sqltutorial.org','515.123.4568','1989-09-21',5,17000.00,100,9);

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,manager\_id,department\_id) VALUES (102,'Lex','De Haan','lex.de haan@sqltutorial.org','515.123.4569','1993-01-13',5,17000.00,100,9);

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,manager\_id,department\_id) VALUES (103,'Alexander','Hunold','alexander.hunold@sqltutorial.org','590.423.4567','1990-01-03',9,9000.00,102,6);

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,manager\_id,department\_id) VALUES (104,'Bruce','Ernst','bruce.ernst@sqltutorial.org','590.423.4568','1991-05-21',9,6000.00,103,6);

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,manager\_id,department\_id) VALUES (105,'David','Austin','david.austin@sqltutorial.org','590.423.4569','1997-06-25',9,4800.00,103,6);

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,manager\_id,department\_id) VALUES (106,'Valli','Pataballa','valli.pataballa@sqltutorial.org','590.423.4560','1998-02-05',9,4800.00,103,6);

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,manager\_id,department\_id) VALUES (107,'Diana','Lorentz','diana.lorentz@sqltutorial.org','590.423.5567','1999-02-07',9,4200.00,103,6);

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,manager\_id,department\_id) VALUES (108,'Nancy','Greenberg','nancy.greenberg@sqltutorial.org','515.124.4569','1994-08-17',7,12000.00,101,10);

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,manager\_id,department\_id) VALUES (109,'Daniel','Faviet','daniel.faviet@sqltutorial.org','515.124.4169','1994-08-16',6,9000.00,108,10);

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,manager\_id,department\_id) VALUES (110,'John','Chen','john.chen@sqltutorial.org','515.124.4269','1997-09-28',6,8200.00,108,10);

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,manager\_id,department\_id) VALUES (111,'Ismael','Sciarra','ismael.sciarra@sqltutorial.org','515.124.4369','1997-09-30',6,7700.00,108,10);

Step 10: Run below mentioned script to find PII data / column from table.

--Script examines all columns in all DBs for PII sounding names

SELECT db\_name() as [DB],

object\_name(o.object\_id) AS [Object],

c.name AS [Column]

FROM sys.columns c join sys.objects o ON c.object\_id = o.object\_id

WHERE o.type IN ('U','V') /\* tables and views \*/

AND (c.name like '%email%'

OR c.name like '%first\_name%'

OR c.name like '%last%name%'

OR c.name like '%phone\_number%'

OR c.name like '%salary%'

OR c.name like '%address%'

OR c.name like '%phone%'

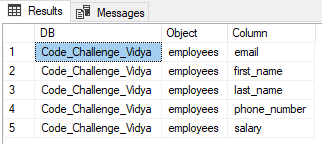
OR c.name like '%social%'

OR c.name like '%ssn%'

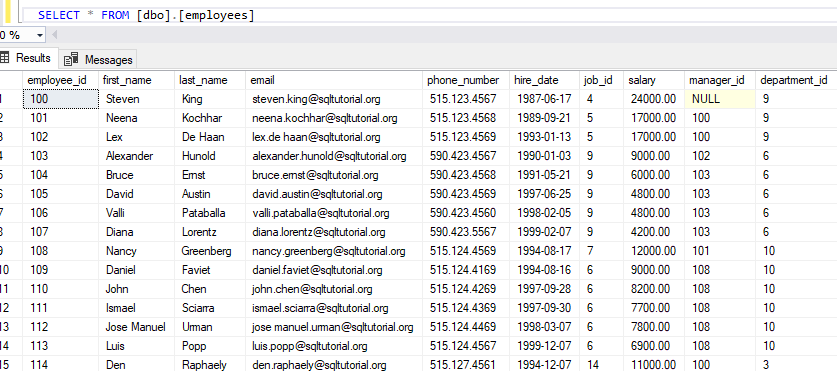
OR c.name like '%gender%')

AND db\_name() NOT IN ('msdb','tempdb','master') and schema\_id <> 4

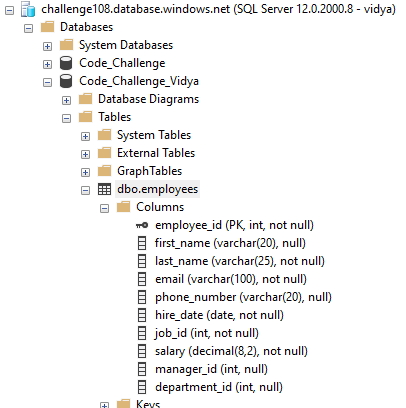
Below is the output:



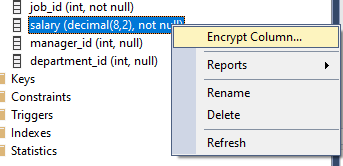
Step 11: Select data from ‘Employee’ table before encryption.



Step 11: Expand table and column details from Object Explorer

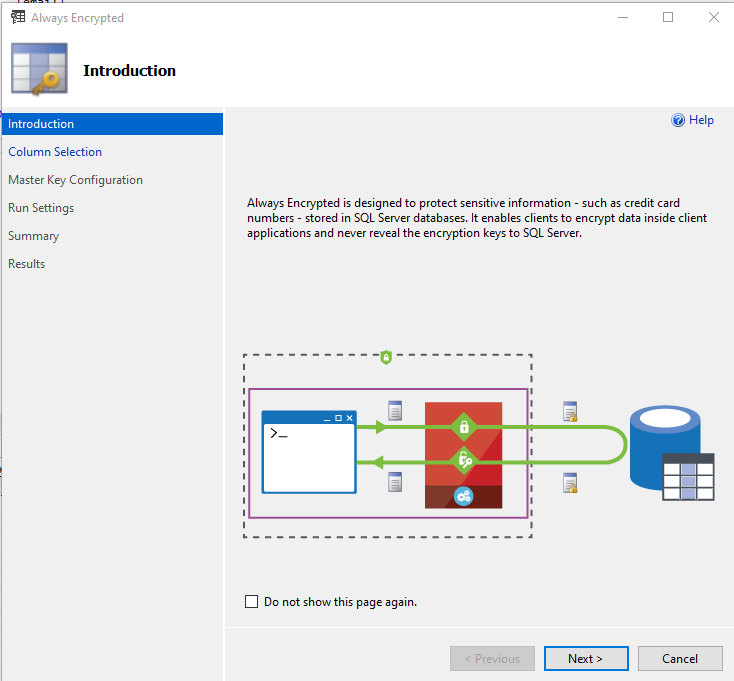


Step 12: Select PII column to encrypt data and right click, click on Encrypt Column option

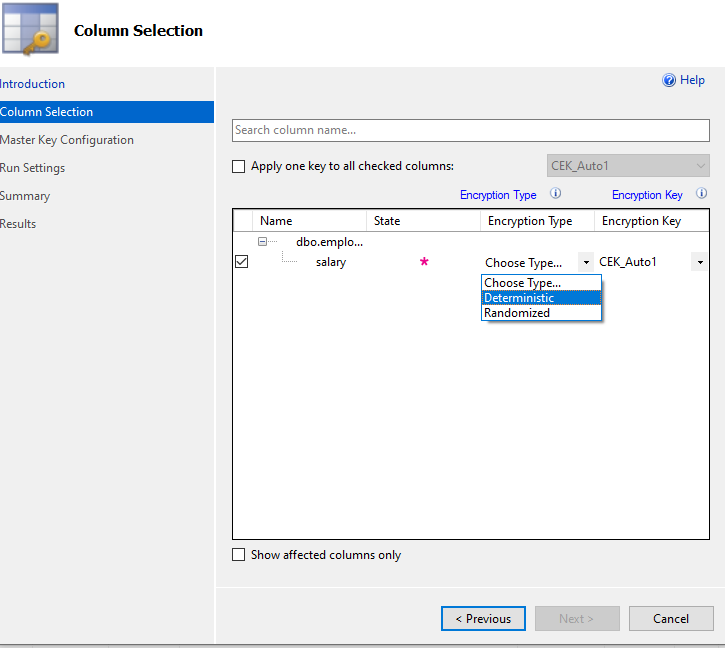


Step 13: Perform below mentioned steps to encrypt data.

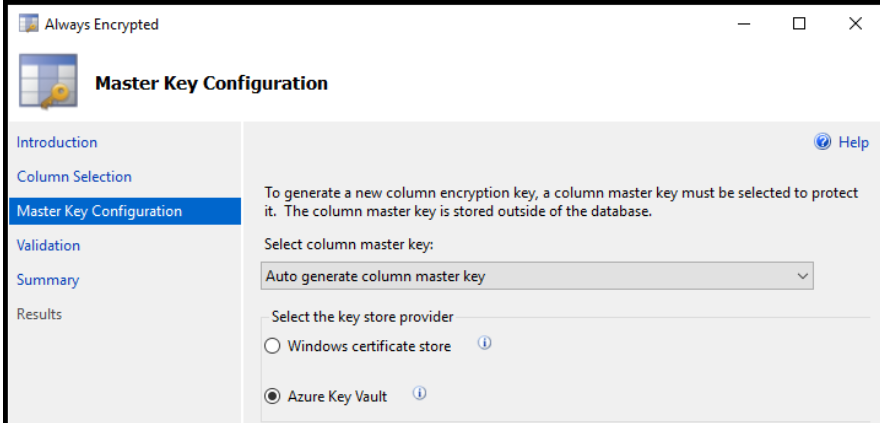
1. Click ‘Next’



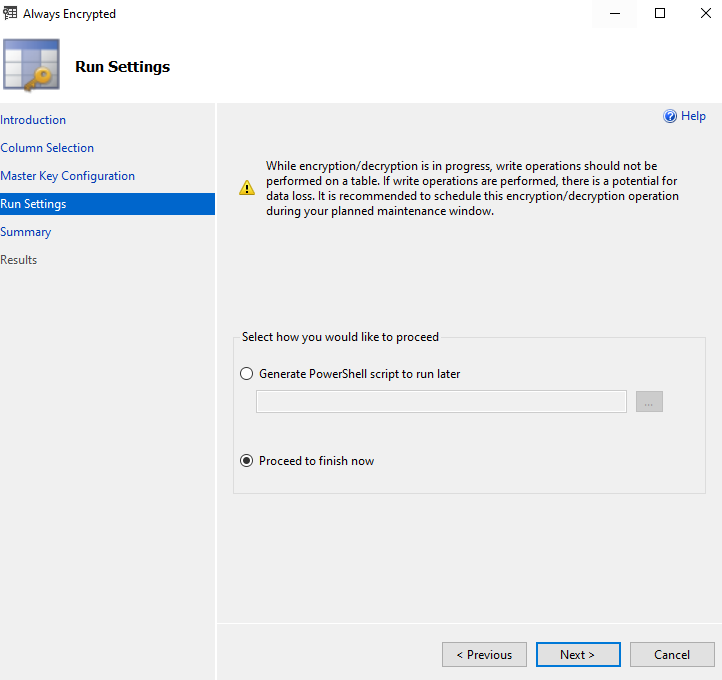
1. Click check box and select ‘Deterministic’ option from drop down and click ‘Next’ button



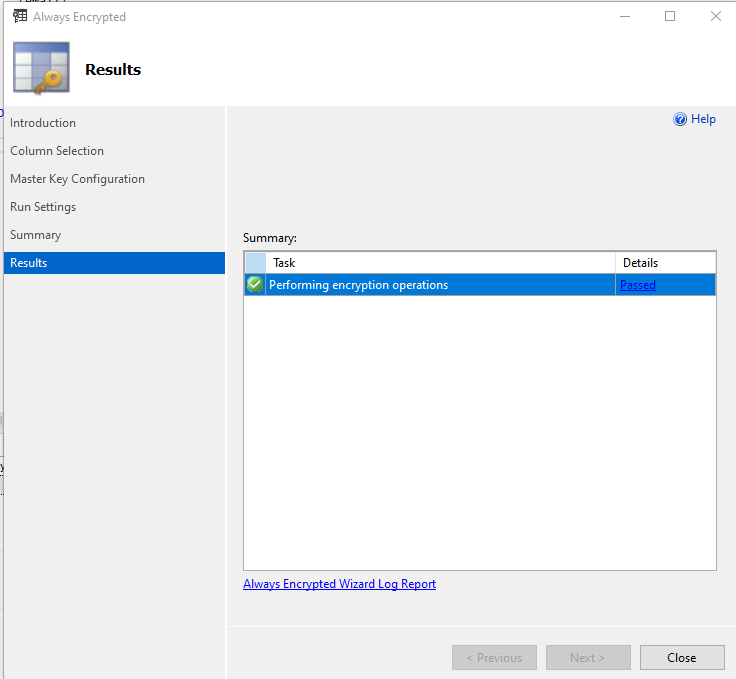
1. Select ‘Windows certificate store’ radio button and click ‘Next’ button. You can also create an entry in Azure Key Vault and can select here.



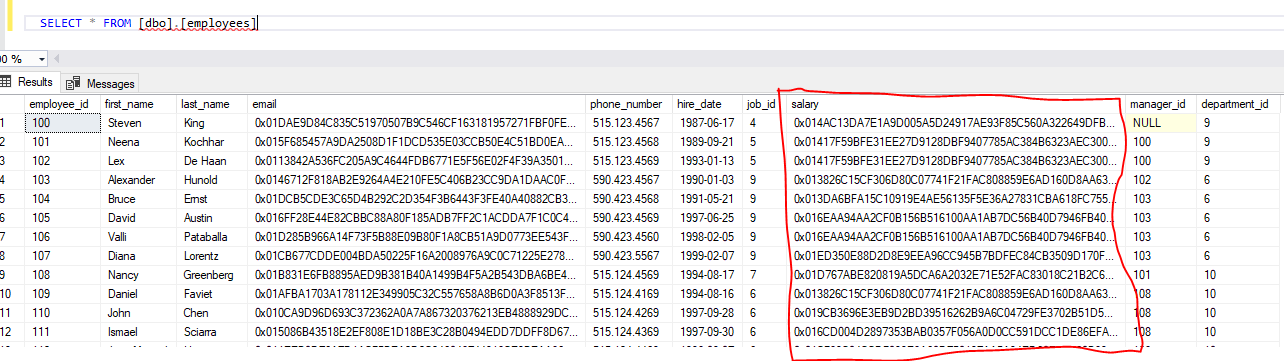
1. Click 2 times ‘Next’ button.



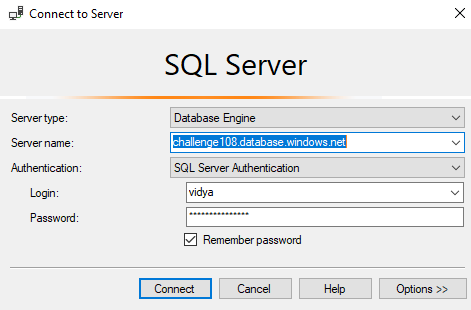
1. Once encryption is completed, you should see below screen and click ‘Close’ button.



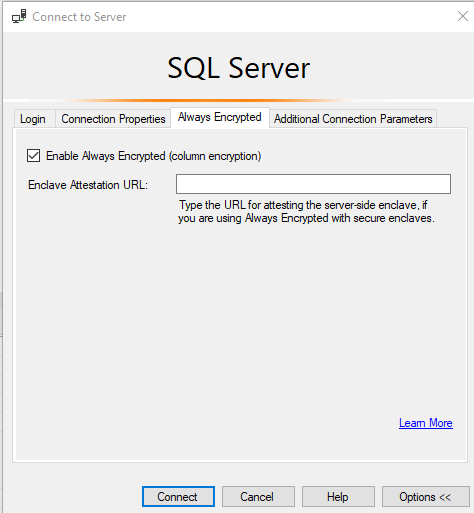
1. Open ‘New Query’ and Select employee table to check salary column data encryption.



1. To see original salary data, disconnect SQL server and connect with below mentioned option.

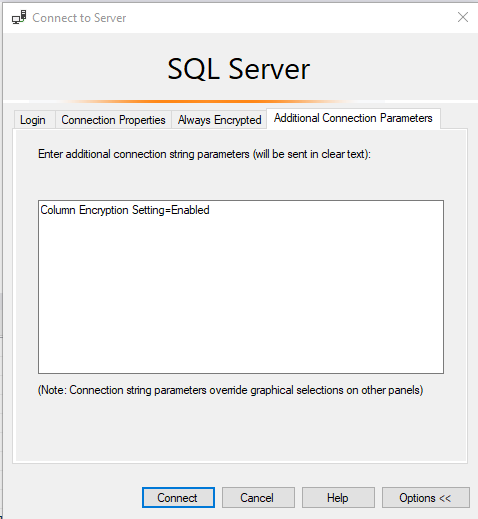


1. Click on ‘Options>>’ button and select ‘Always Encrypted’ tab and click check box before Enable Always Encrypted (column encryption).

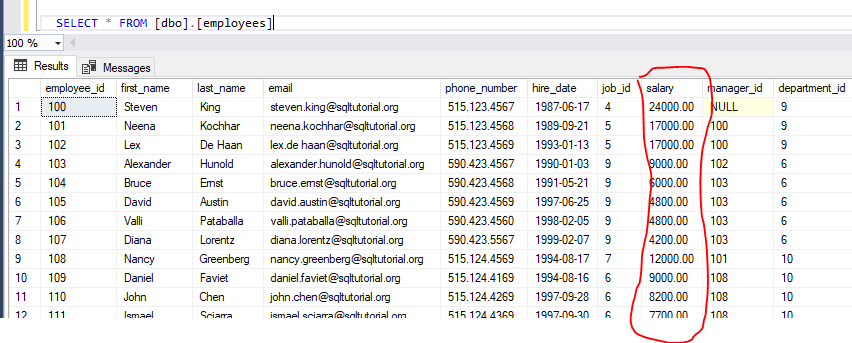


1. Click on ‘Additional Connection Parameter’ tab and type below mentioned code

Column Encryption Setting=Enabled



1. Click on ‘Connect’ button to connect to server and open a ‘New Query’ and select data from ‘Employee’ table



1. Below mentioned code can be used in application configuration file to access actual data

// Instantiate a SqlConnectionStringBuilder.

SqlConnectionStringBuilder connStringBuilder = new SqlConnectionStringBuilder("replace with your connection string");

// Enable Always Encrypted.

connStringBuilder.ColumnEncryptionSetting = SqlConnectionColumnEncryptionSetting.Enabled;

Reference Link:

<https://docs.microsoft.com/en-us/azure/azure-sql/database/always-encrypted-azure-key-vault-configure?view=azuresql&tabs=azure-powershell>