Manage sensitive data with Dynamic Data Masking

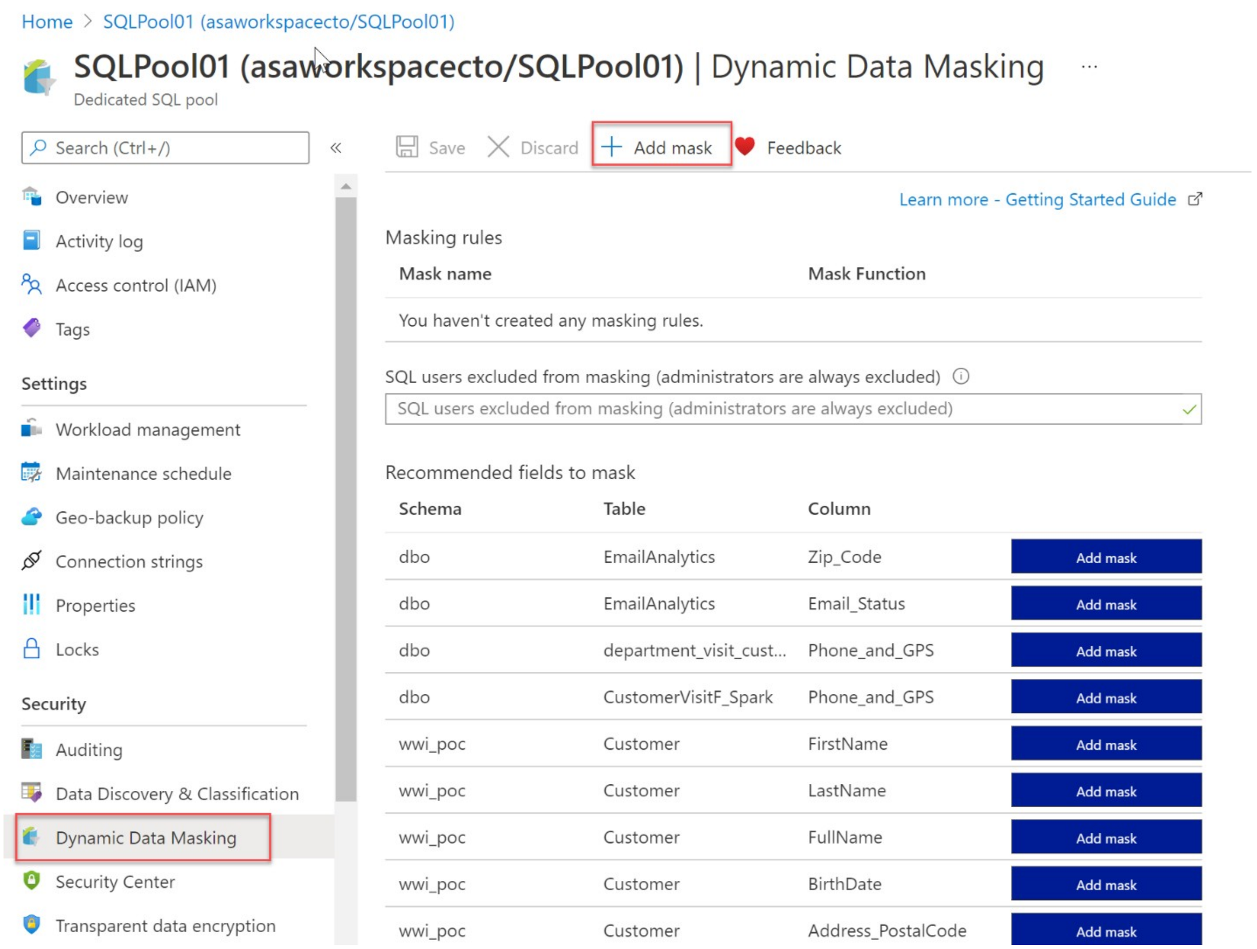
Note:

You are not required to complete the processes, tasks, activities, or steps presented in this example. The various samples provided are for illustrative purposes only and it’s likely that if you try this out you will encounter issues in your system.

Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics support Dynamic Data Masking. Dynamic Data Masking ensures limited data exposure to non-privileged users, such that they cannot see the data that is being masked. It also helps you in preventing unauthorized access to sensitive information that has minimal impact on the application layer. Dynamic Data Masking is a policy-based security feature. It will hide the sensitive data in a result set of a query that runs over designated database fields.

Let's give you an example how it works. Let's say you work at a bank as a service representative in a call center. Due to compliance, any caller must identify themselves by providing several digits of their credit card number. In this scenario, the full credit card number should not be fully exposed to the service representative in the call center. You can define a masking rule, that masks all but the last four digits of a credit card number, so that you would get a query that only gives as a result the last four digits of the credit card number. This is just one example that could be equally applied to a variety of personal data such that compliance is not violated. For Azure Synapse Analytics, the way to set up a Dynamic Data Masking policy is using PowerShell or the REST API. The configuration of the Dynamic Data Masking policy can be done by the Azure SQL Database admin, server admin, or SQL Security Manager roles.

In Azure Synapse Analytics, you can find Dynamic Data Masking here;



Masking rules

Mask name	Mask Function
You haven't created any masking rules.	

SQL users excluded from masking (administrators are always excluded)

SQL users excluded from masking (administrators are always excluded)

Recommended fields to mask

Schema	Table	Column	
dbo	EmailAnalytics	Zip_Code	Add mask
dbo	EmailAnalytics	Email_Status	Add mask
dbo	department_visit_cust...	Phone_and_GPS	Add mask
dbo	CustomerVisitF_Spark	Phone_and_GPS	Add mask
wwi_poc	Customer	FirstName	Add mask
wwi_poc	Customer	LastName	Add mask
wwi_poc	Customer	FullName	Add mask
wwi_poc	Customer	BirthDate	Add mask
wwi_poc	Customer	Address_PostalCode	Add mask

Dynamic Data Masking Azure Synapse Analytics

Looking into Dynamic Data Masking Policies:

- **SQL users excluded from Dynamic Data Masking Policies** The following SQL users or Azure AD identities can get unmasked data in the SQL query results. Users with administrator privileges are always excluded from masking, and will see the original data without any mask.
- **Masking rules** - Masking rules are a set of rules that define the designated fields to be masked including the masking function that is used. The designated fields can be defined using a database schema name, table name, and column name.
- **Masking functions** - Masking functions are a set of methods that control the exposure of data for different scenarios.

Set up Dynamic Data Masking for your database in Azure Synapse Analytics using PowerShell cmdlets

In this part, we are going to look into Dynamic Data Masking for a database in Azure Synapse Analytics using PowerShell cmdlets.

- Data masking policies
- Get-AzSqlDatabaseDataMaskingPolicy

The Get-AzSqlDatabaseDataMaskingPolicy gets the data masking policy for a database.

The syntax for the Get-AzSqlDatabaseDataMaskingPolicy in PowerShell is as follows:

```
Get-AzSqlDatabaseDataMaskingPolicy [-ServerName] <String> [-DatabaseName] <String
>
 [-ResourceGroupName] <String> [-DefaultProfile <IAzureContextContainer>] [-
WhatIf] [-Confirm]
 [<CommonParameters>]
```

What the **Get-AzSqlDatabaseDataMaskingPolicy** cmdlet does, is getting the data masking policy of an Azure SQL database.

To use this cmdlet in PowerShell, you'd have to specify the following parameters to identify the database:

- *ResourceGroupName*: name of the resource group you deployed the database in
- *ServerName*: sql server name
- *DatabaseName* : name of the database

This cmdlet is also supported by the SQL Server Stretch Database service on Azure.

- Set-AzSqlDatabaseDataMaskingPolicy

The Set-AzSqlDatabaseDataMaskingPolicy sets data masking for a database.

The syntax for the Set-AzSqlDatabaseDataMaskingPolicy in PowerShell is as follows:

```
Set-AzSqlDatabaseDataMaskingPolicy [-PassThru] [-PrivilegedUsers <String>] [-
DataMaskingState <String>]
 [-ServerName] <String> [-DatabaseName] <String> [-ResourceGroupName] <String>
 [-DefaultProfile <IAzureContextContainer>] [-WhatIf] [-Confirm] [<CommonParamete
rs>]
```

What the **Set-AzSqlDatabaseDataMaskingPolicy** cmdlet does is setting the data masking policy for an Azure SQL database.

To use this cmdlet in PowerShell, you'd have to specify the following parameters to identify the database:

- *ResourceGroupName*: name of the resource group that you deployed the database in
- *ServerName* : sql server name
- *DatabaseName* : name of the database

In addition, you will need to set the *DataMaskingState* parameter to specify whether data masking operations are enabled or disabled.

If the cmdlet succeeds and the *PassThru* parameter is used, it will return an object describing the current data masking policy in addition to the database identifiers.

Database identifiers can include, **ResourceGroupName**, **ServerName**, and **DatabaseName**.

This cmdlet is also supported by the SQL Server Stretch Database service on Azure.

- Data masking rules
- Get-AzSqlDatabaseDataMaskingRule

The Get-AzSqlDatabaseDataMaskingRule Gets the data masking rules from a database.

The syntax for the Get-AzSqlDatabaseDataMaskingRule in PowerShell is as follows:

```
Get-AzSqlDatabaseDataMaskingRule [-SchemaName <String>] [-TableName <String>] [-ColumnName <String>]
 [-ServerName] <String> [-DatabaseName] <String> [-ResourceGroupName] <String>
 [-DefaultProfile <IAzureContextContainer>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

What the **Get-AzSqlDatabaseDataMaskingRule** cmdlet does it getting either a specific data masking rule or all of the data masking rules for an Azure SQL database.

To use the cmdlet in PowerShell, you'd have to specify the following parameters to identify the database:

To use this cmdlet in PowerShell, you'd have to specify the following parameters to identify the database:

- *ResourceGroupName*: name of the resource group that you deployed the database in
- *ServerName* : sql server name
- *DatabaseName* : name of the database

You'd also have to specify the *RuleId* parameter to specify which rule this cmdlet returns.

If you do not provide *RuleId*, all the data masking rules for that Azure SQL database are returned.

This cmdlet is also supported by the SQL Server Stretch Database service on Azure.

- New-AzSqlDatabaseDataMaskingRule

The New-AzSqlDatabaseDataMaskingRule creates a data masking rule for a database.

The syntax for the New-AzSqlDatabaseDataMaskingRule in PowerShell is as follows:

```
New-AzSqlDatabaseDataMaskingRule -MaskingFunction <String> [-PrefixSize <UInt32>]
[-ReplacementString <String>]
 [-SuffixSize <UInt32>] [-NumberFrom <Double>] [-NumberTo <Double>] [-PassThru]
-SchemaName <String>
 -TableName <String> -ColumnName <String> [-ServerName] <String> [-DatabaseName]
<String>
 [-ResourceGroupName] <String> [-DefaultProfile <IAzureContextContainer>] [-
WhatIf] [-Confirm]
 [<CommonParameters>]
```

What the **New-AzSqlDatabaseDataMaskingRule** cmdlet does is creating a data masking rule for an Azure SQL database.

To use this cmdlet in PowerShell, you'd have to specify the following parameters to identify the rule:

- *ResourceGroupName*: name of the resource group that you deployed the database in
- *ServerName* : sql server name
- *DatabaseName* : name of the database

Providing the *TableName* and *ColumnName* is necessary in order to specify the target of the rule.

The *MaskingFunction* parameter is necessary to define how the data is masked.

If *MaskingFunction* has a value of Number or Text, you can specify the *NumberFrom* and *NumberTo* parameters, for number masking, or the *PrefixSize*, *ReplacementString*, and *SuffixSize* for text masking.

If the command succeeds and the *PassThru* parameter is used, the cmdlet returns an object describing the data masking rule properties in addition to the rule identifiers.

Rule identifiers can be, for example, *ResourceGroupName*, *ServerName*, *DatabaseName*, and *RuleID*.

This cmdlet is also supported by the SQL Server Stretch Database service on Azure.

- Remove-AzSqlDatabaseDataMaskingRule

The Remove-AzSqlDatabaseDataMaskingRule removes a data masking rule from a database.

The syntax for the Remove-AzSqlDatabaseDataMaskingRule in PowerShell is as follows:

```
Remove-AzSqlDatabaseDataMaskingRule [-PassThru] [-Force] -SchemaName <String> -
TableName <String>
 -ColumnName <String> [-ServerName] <String> [-DatabaseName] <String> [-
ResourceGroupName] <String>
 [-DefaultProfile <IAzureContextContainer>] [-WhatIf] [-Confirm] [<CommonParamete
rs>]
```

What the **Remove-AzSqlDatabaseDataMaskingRule** cmdlet does, is it removes a specific data masking rule from an Azure SQL database.

To use this cmdlet in PowerShell, you'd have to specify the following parameters to identify the rule that needs to be removed:

- *ResourceGroupName*: name of the resource group that you deployed the database in
- *ServerName* : sql server name
- *DatabaseName* : name of the database
- *RuleId* : identifier of the rule

This cmdlet is also supported by the SQL Server Stretch Database service on Azure.

- Set-AzSqlDatabaseDataMaskingRule

The Set-AzSqlDatabaseDataMaskingRule Sets the properties of a data masking rule for a database.

The syntax for the Set-AzSqlDatabaseDataMaskingRule in PowerShell is as follows:

```
Set-AzSqlDatabaseDataMaskingRule [-MaskingFunction <String>] [-PrefixSize <UInt32>]
 [-ReplacementString <String>] [-SuffixSize <UInt32>] [-NumberFrom <Double>] [-NumberTo <Double>] [-PassThru]
 -SchemaName <String> -TableName <String> -ColumnName <String> [-ServerName] <String> [-DatabaseName] <String>
 [-ResourceGroupName] <String> [-DefaultProfile <IAzureContextContainer>] [-WhatIf] [-Confirm]
 [<CommonParameters>]
```

What the **Set-AzSqlDatabaseDataMaskingRule** cmdlet does is setting a data masking rule for an Azure SQL database.

To use this cmdlet in PowerShell, you'd have to specify the following parameters to identify the rule:

- *ResourceGroupName*: name of the resource group that you deployed the database in
- *ServerName* : sql server name
- *DatabaseName* : name of the database
- *RuleId* : identifier of the rule

You can provide any of the parameters of *SchemaName*, *TableName*, and *ColumnName* to retarget the rule.

Specify the *MaskingFunction* parameter to modify how the data is masked.

If you specify a value of Number or Text for *MaskingFunction*, you can specify the *NumberFrom* and *NumberTo* parameters for number masking or the *PrefixSize*, *ReplacementString*, and *SuffixSize* parameters for text masking.

If the command succeeds, and if you specify the *PassThru* parameter, the cmdlet returns an object that describes the data masking rule properties and the rule identifiers.

Rule identifiers can be, **ResourceGroupName**, **ServerName**, **DatabaseName**, and **RuleId**.

This cmdlet is also supported by the SQL Server Stretch Database service on Azure.

Set up Dynamic Data Masking for your database in Azure Synapse Analytics using the REST API

For setting up Dynamic Data Masking in Azure Synapse Analytics, you can also make use of the REST API. It will enable you to programmatically manage data masking policy and rules.

The REST API will support the following operations:

- Data masking policies
- Create Or Update

The Create Or Update masking policy using the REST API will create or update a database data masking policy.

In HTTP the following request can be made:

```
PUT https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Sql/servers/{serverName}/databases/{databaseName}/dataMaskingPolicies/Default?api-version=2014-04-01
```

The following parameters need to be passed through:

- *SubscriptionID*: the ID of the subscription
- *ResourceGroupName*: name of the resource group that you deployed the database in
- *ServerName* : sql server name
- *DatabaseName* : name of the database
- *dataMaskingPolicyName*: the name of the data masking policy
- *api version*: version of the api that is used.
- Get

The Get policy, gets a database data masking policy.

In HTTP the following request can be made:

```
GET https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Sql/servers/{serverName}/databases/{databaseName}/dataMaskingPolicies/Default?api-version=2014-04-01
```

The following parameters need to be passed through:

- *SubscriptionID*: the ID of the subscription
- *ResourceGroupName*: name of the resource group that you deployed the database in

- *ServerName* : sql server name
- *DatabaseName* : name of the database
- *dataMaskingPolicyName*: the name of the data masking policy
- *api version*: version of the api that is used.
- Data masking rules
- Create Or Update

The Create or Update masking rule creates or updates a database data masking rule.

In HTTP the following request can be made:

```
PUT https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Sql/servers/{serverName}/databases/{databaseName}/dataMaskingPolicies/Default/rules/{dataMaskingRuleName}?api-version=2014-04-01
```

The following parameters need to be passed through:

- *SubscriptionID*: the ID of the subscription
- *ResourceGroupName*: name of the resource group that you deployed the database in
- *ServerName* : sql server name
- *DatabaseName* : name of the database
- *dataMaskingPolicyName*: the name of the data masking policy
- *dataMaskingRuleName*: the name of the rule for data masking
- *api version*: version of the api that is used.
- List By Database

The List By Database request gets a list of database data masking rules.

In HTTP the following request can be made:

```
GET https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Sql/servers/{serverName}/databases/{databaseName}/dataMaskingPolicies/Default/rules?api-version=2014-04-01
```

The following parameters need to be passed through:

- *SubscriptionID*: the ID of the subscription
- *ResourceGroupName*: name of the resource group that you deployed the database in
- *ServerName* : sql server name
- *DatabaseName* : name of the database
- *dataMaskingPolicyName*: the name of the data masking policy
- *api version*: version of the api that is used.