Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'Get-SqlSensitivityClassification'

PS:\>Get-HELP Get-SqlSensitivityClassification -Full

NAME

Get-SqlSensitivityClassification

SYNOPSIS

Get the sensitivity label and information type of columns in the database.

SYNTAX

Get-SqlSensitivityClassification [-ColumnName <String[]>] -ConnectionString <String> [-ProgressAction <ActionPreference>] [<CommonParameters>]

Get-SqlSensitivityClassification [-ColumnName <String[]>] [-Credential <PSCredential>] -DatabaseName <String> [-ProgressAction <ActionPreference>] -ServerInstance

<PSObject> [<CommonParameters>]

Get-SqlSensitivityClassification [-ColumnName <String[]>] -InputObject <Database> [-ProgressAction <ActionPreference>] [<CommonParameters>]

Get-SqlSensitivityClassification [-ColumnName <String[]>] -Path <String> [-ProgressAction <ActionPreference>] [<CommonParameters>]

Get-SqlSensitivityClassification [-ColumnName <String[]>] [-ProgressAction <ActionPreference>] [-SuppressProviderContextWarning] [<CommonParameters>]

DESCRIPTION

The Get-SqlSensitivityClassification cmdlet gets the sensitivity labels and information types of columns in the database.

The sensitivity labels and information types of columns can also be viewed using SQL Server Management Studio (SSMS) (/sql/ssms/sql-server-management-studio-ssms)
release 17.5 and above, or the [Extended Properties catalog
view](/sql/relational-databases/security/sql-data-discovery-and-classification?view=sql-server-2017#subheading-3).

The sensitivity labels and information types of columns can be set using SQL Server Management Studio (SSMS) (/sql/ssms/sql-server-management-studio-ssms)release 17.5
and above, or with the Set-SqlSensitivityClassification cmdlet.

> `Module requirements: version 21+ on PowerShell 5.1; version 22+ on PowerShell 7.x.`

PARAMETERS

-ColumnName <String[]>

Name(s) of columns for which information type and sensitivity label is fetched.

Required? false
Position? named
Default value None
Accept pipeline input? True (ByPropertyName)
Accept wildcard characters? false

-ConnectionString <String>

Specifies a connection string to connect to the database. If this parameter is present, other connection parameters will be ignored.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Credential <PSCredential>

Specifies a credential used to connect to the database.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-DatabaseName <String>

Specifies the name of a database. This cmdlet connects to this database in the instance that is specified in the ServerInstance parameter.

If the DatabaseName parameter is not specified, the database that is used depends on whether the current path specifies both the SQLSERVER:\SQL folder and a

database name. If the path specifies both the SQL folder and a database name, this cmdlet connects to the database that is specified in the path.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-InputObject <Database>

Specifies a SQL Server Management Object (SMO) that represent the database that this cmdlet uses.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Path <String>

Specifies the path to the instance of SQL Server on which this cmdlet runs the operation. If you do not specify a value for this parameter, the cmdlet uses the

current working location.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ProgressAction <ActionPreference>

Determines how PowerShell responds to progress updates generated by a script, cmdlet, or provider, such as the progress bars generated by the Write-Progress

cmdlet. The Write-Progress cmdlet creates progress bars that show a command's status.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ServerInstance <PSObject>

Specifies either the name of the server instance (a string) or SQL Server Management Objects (SMO) object that specifies the name of an instance of the Database

Engine. For default instances, only specify the computer name: MyComputer. For named instances, use the format

ComputerName\InstanceName.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-SuppressProviderContextWarning [<SwitchParameter>]

Indicates that this cmdlet suppresses the warning that this cmdlet has used in the database context from the current SQLSERVER:\SQL path setting to establish the

database context for the cmdlet.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,

ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see

about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

System.String[]

Microsoft.SqlServer.Management.Smo.Database

OUTPUTS

System.Object

NOTES

Example 1: Get all information types and sensitivity labels from a database using Windows authentication

```
PS C:\> Get-SqlSensitivityClassification -ServerInstance "MyComputer\MainInstance" -Database "myDatabase"

Column                    InformationType SensitivityLabel    SensitivityRank
------                    --------------- ----------------    ---------------
Sales.Customers.email       Contact Info    Confidential        Medium
Sales.Customers.first_name  Name            Confidential - GDPR Medium
```

This command gets the information type and sensitivity label of all columns in the database. The cmdlet returns only the columns that have an information type or a

sensitivity label (or both) defined.

Example 2: Get information type and sensitivity label of a single column from database by providing path

```
PS C:\> Get-SqlSensitivityClassification -Path "SQLSERVER:\SQL\MyComputer\MainInstance\Databases\MyDatabase"
-ColumnName "Sales.Customers.email"

Column                    InformationType SensitivityLabel SensitivityRank
------                    --------------- ---------------- ---------------
Sales.Customers.email     Contact Info    Confidential     Medium
```

This command gets the information type and sensitivity label of the `Sales.Customers.email` column in the database provided in the Path.

Example 3: Get sensitivity labels and information types of multiple columns using using current path context

```
PS C:\> $columns = @("Sales.Customers.ip_address" , "Sales.Customers.email")
PS C:\> Set-Location "SQLSERVER:\SQL\MyComputer\MainInstance\Databases\MyDatabase"
PS SQLSERVER:\SQL\MyComputer\MainInstance> Get-SqlSensitivityClassification -ColumnName $columns
 WARNING: Using provider context. Server = MyComputer, Database = MyDatabase.

Column                    InformationType SensitivityLabel SensitivityRank
------                    --------------- ---------------- ---------------
Sales.Customers.email       Contact Info    Confidential     Medium
Sales.Customers.ip_address
```

This command gets the information type and sensitivity label of multiple columns in the database using and array argument containing the column names and the current
path context to locate the database.

Columns that have no information type or sensitivity label, such as `Sales.Customers.ip_Addess` in the example, will return empty results.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/sqlserver/get-sqlsensitivityclassification

SQL Data Discovery and Classification