

# Custom Query Guide

Version 24.2



## Legal Notices

Condrey Corporation makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Condrey Corporation reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Condrey Corporation makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Condrey Corporation reserves the right to make changes to any and all parts of the software at any time, without obligation to notify any person or entity of such revisions or changes. See the Software EULA for full license and warranty information with regard to the Software.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export, or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. Condrey Corporation assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 2024 Condrey Corporation. All Rights Reserved.

No part of this publication may be reproduced, photocopied, or transmitted in any fashion without the express written consent of the publisher.

Condrey Corporation  
122 North Laurens St.  
Greenville, SC 29601  
U.S.A.

<https://condreycorp.com/>



# Third-Party Systems

The software is designed to run in an environment containing third-party elements meeting certain prerequisites. These may include operating systems, directory services, databases, and other components or technologies. See the accompanying prerequisites list for details.

The software may require a minimum version of these elements to function. Further, these elements may require appropriate configuration and resources such as computing, memory, storage, or bandwidth for the software to be able to perform in a way that meets the customer requirements. The download, installation, performance, upgrade, backup, troubleshooting, and management of these elements is the responsibility of the customer using the third-party vendor's documentation and guidance.

Third-party systems emulating any of these elements must fully adhere to and support the appropriate APIs, standards, and protocols for the software to function. Support of the software in conjunction with such emulating third-party elements is determined on a case-by-case basis and may change at any time.



# Contents

---

<b>Custom Query Guide</b> .....	<b>1</b>
Version 24.2 .....	1
<b>Legal Notices</b> .....	<b>3</b>
<b>Third-Party Systems</b> .....	<b>5</b>
<b>Contents</b> .....	<b>7</b>
<b>About This Guide</b> .....	<b>1</b>
Audience .....	1
<b>1 - Updates and Breaking Changes</b> .....	<b>3</b>
1.1 - File Reporter 4.1 .....	3
1.1.1 - Additional Schema for Microsoft 365 .....	3
1.1.2 - Removed Tables .....	3
1.1.3 - Removed Columns .....	4
1.2 - File Reporter 4.0 .....	4
1.2.1 - Deprecated Views .....	4
<b>2 - Supported Constructs</b> .....	<b>5</b>
2.1 - Supported Schema Objects .....	5
2.2 - Schema Namespaces .....	5
2.3 - Supported Tables .....	6
2.4 - Supported Views .....	9
2.5 - Supported Functions .....	10
<b>3 - Navigating Scan Data</b> .....	<b>11</b>
3.1 - Windows File System .....	12
3.1.1 - Table Relationships .....	12
3.1.2 - Scoping and Filtering .....	13
3.1.3 - File System Target Paths .....	22
3.2 - Active Directory Identities .....	26
<b>4 - Example Scenarios</b> .....	<b>29</b>
4.1 - Content Hash Duplicate File Reports .....	29

---

4.1.1 - Determining Prerequisites .....	29
4.1.2 - Designing the Report .....	29
4.2 - Microsoft 365 Reports .....	33
4.2.1 - Determining Prerequisites .....	33
4.2.2 - Designing the Report .....	33
4.3 - Active Directory Identity Enrichment .....	37
4.3.1 - Determining Prerequisites .....	37
4.3.2 - Designing the Report .....	37
<b>5 - Schema Reference .....</b>	<b>43</b>
5.1 - Tables .....	43
ad.domains .....	43
ad.ds_objects .....	44
srs.analysis.file_scan_entries .....	49
ms365.drive_item_types .....	50
ms365.drive_items .....	51
ms365.drive_scans .....	53
ms365.drive_scans_history .....	54
ms365.drives .....	55
ms365.group_drives .....	56
ms365.group_member_types .....	57
ms365.group_members .....	58
ms365.group_owners .....	59
ms365.group_sites .....	60
ms365.groups .....	61
ms365.identity_types .....	62
ms365.jobs .....	63
ms365.jobs_history .....	64
ms365.permissions .....	65
ms365.sharing_link_members .....	68
ms365.sites .....	70

---

ms365.sp_base_permissions .....	71
ms365.sp_group_members .....	72
ms365.sp_groups .....	73
ms365.sp_permission_levels .....	74
ms365.sp_permissions .....	76
ms365.sp_site_permissions .....	77
ms365.sp_users .....	78
ms365.team_channels .....	80
ms365.teams .....	81
ms365.tenants .....	82
ms365.user_drives .....	83
ms365.users .....	84
srs.ad_memberships .....	86
srs.ad_objects .....	87
srs.identity_systems .....	88
srs.ntfs_aces .....	89
srs.scan_data .....	91
srs.scan_directory_data .....	94
srs.scan_history .....	95
srs.scan_targets .....	98
srs.scans .....	99
srs.security_descriptors .....	102
srs.tend_volume_freespace .....	104
5.2 - Temp Tables .....	105
tmp_cq_fs_paths .....	105
5.3 - Views .....	108
ad.ds_objects_view .....	108
srs.baseline_fs_scandata .....	113
srs.baseline_fs_scans .....	116
srs.baseline_ntfs_aces .....	118

srs.baseline_permissions_scans .....	122
srs.current_fs_scandata .....	124
srs.current_fs_scans .....	127
srs.current_ntfs_aces .....	129
srs.current_permissions_scans .....	133
srs.previous_fs_scandata .....	135
srs.previous_fs_scans .....	138
srs.previous_ntfs_aces .....	140
srs.previous_permissions_scans .....	144
5.4 - Functions .....	146
srs.access_mask_basic_string .....	146
srs.access_mask_string .....	148
srs.ace_flags_string .....	150
srs.ace_type_string .....	151
srs.ad_account_name .....	153
srs.attribute_string .....	154
srs.byte_string .....	156
srs.byte_unit_string .....	157
srs.bytes_to_hex_string .....	158
srs.guid_bytes .....	159
srs.guid_text .....	160
srs.hex_string_to_bytes .....	161
srs.path_hash .....	162
srs.path_hash_sha256 .....	163
srs.sid_bytes .....	164
srs.sid_text .....	165

## About This Guide

The Custom Query guide is written to provide guidance for the development of SQL queries for use with Custom Query reports in File Reporter 24.2.

### Audience

This guide is intended for network administrators and report developers responsible for developing SQL queries for use with Custom Query reports in File Reporter 24.2.



# 1 - Updates and Breaking Changes

## 1.1 - File Reporter 4.1

### 1.1.1 - Additional Schema for Microsoft 365

Supported schema for extended Microsoft 365 SharePoint Online data has been added with this release.

A new set of SharePoint-specific tables have been added for improved analysis of permissions in OneDrive for Business and SharePoint Online document libraries.

The new set of tables includes:

- ms365.sp\_base\_permissions
- ms365.sp\_group\_members
- ms365.sp\_groups
- ms365.sp\_permission\_levels
- ms365.sp\_permissions
- ms365.sp\_site\_permissions
- ms365.sp\_users

In addition, supported references for SharePoint identifiers have been added to the *ms365.permissions* table:

- grantedto\_sp\_user\_id
- grantedto\_sp\_group\_id
- grantedto\_sp\_login\_name
- site\_collection\_id

### 1.1.2 - Removed Tables

The *ms365.site\_drives* table has been removed as of File Reporter 4.1.

The *ms365.drives* table now include a *site\_id* reference column that replaces the need for this bridge table.

Upgrading from File Reporter 4.0 to 4.1 automatically extends this table and populates the corresponding new reference column using the legacy *ms365.site\_drives* table before dropping it.



**IMPORTANT:** Any Custom Queries that reference the legacy *ms365.site\_drives* table will need to be updated to make use of the new *ms365.drives.site\_id* column instead.

Any queries that continue to reference the legacy table will no longer work after upgrading to File Reporter 4.1 or later until this change has been made.

### 1.1.3 - Removed Columns

The *grantedto\_id\_type* string-typed column in the *ms365.permissions* table has been removed as of File Reporter 4.1.

A replacement column *grantedto\_type* has been added which is an integer type representing a discrete enumeration.

## 1.2 - File Reporter 4.0

### 1.2.1 - Deprecated Views

The following views were deprecated as of File Reporter 4.0 in favor of their corresponding generic view names:

- *srs.current\_fs\_scandata\_ad*
- *srs.previous\_fs\_scandata\_ad*
- *srs.baseline\_fs\_scandata\_ad*

Please use the following views instead, as the \*\_ad views are subject to removal in a later release:

- *srs.current\_fs\_scandata*
- *srs.previous\_fs\_scandata*
- *srs.baseline\_fs\_scandata*

## 2 - Supported Constructs

### 2.1 - Supported Schema Objects

The supported database schema objects include entries in the following categories:

- Identity Systems - system name, users, groups, other security principals
- Windows File System - file system metadata, permissions
- File Content Analysis Data - data related to discovery of search expressions over file content
- Microsoft 365 Data – data related to drives, drive items and supporting meta data and permissions as well as basic teams and sites info in Microsoft 365

Although any tables, views, stored procedures, and functions in the database may be accessed via custom queries, only the tables, views and functions listed here are supported for use with Custom Query development.



**NOTE:** Users who are new to SQL may find the supported views easier to start with as each view provides a simple presentation of several key tables. In addition, the **current\_\*** views are pre-filtered for only the most recent scan data. More experienced users may find performance benefits from making direct inline queries against the tables themselves, especially for complex queries.

### 2.2 - Schema Namespaces

All supported database objects and functions reside in specific schema namespaces. For example, the distinguished name for the table *scan\_data* would be referenced as *srs.scan\_data* when using the namespace prefix.

Although use of the namespace prefix is not required in all cases, there are some cases where it is required, such as when referencing a user defined function in Microsoft SQL Server, or when another database object of the same name exists in the schema search path. For these reasons you should always reference each supported database object and function with its documented namespace prefix.

The following table lists the namespaces containing database objects supported for use with custom SQL queries.

Schema Name	Notes
ad	Contains the Active Directory identity data structures

## 2 - Supported Constructs

Schema Name	Notes
analysis	Contains file content analysis data structures
ms365	Contains Microsoft 365 data structures and functions
srs	Primary namespace containing all file system data structures as well as general functions

## 2.3 - Supported Tables

Category	Table Name	Notes	
Windows File System	srs.identity_systems	List of all identity systems	
	srs.ad_objects	List of all scanned Active Directory security principals	
	srs.ad_memberships	Active Directory group memberships	
	srs.scan_targets	List of all configured scan targets (volumes, shares, etc.)	
	srs.scans	List of all available scans	
	srs.scan_history	Historical scan summary records	
	srs.scan_data	All scan data – includes all path and file-specific metadata info	
	srs.scan_directory_data	All directory-specific scan data	
	srs.trend_volume_freespace	List of all volume free space records	
	srs.ntfs_aces	Scanned NTFS ACEs	
	srs.security_descriptors	Scanned NTFS security descriptors	
	Active	ad.domain	List of scanned Active Directory domains in the

Category	Table Name	Notes
<b>Directory</b>		forest
	ad.ds_objects	List of scanned security principals in the Active Directory forest
<b>File Content Analysis</b>	analysis.file_scan_entries	Summary classification data for file content analysis entries
<b>Microsoft 365</b>	ms365.drive_items	Files and folders in drives, document libraries
	ms365.drive_item_types	Enumeration table of drive item types
	ms365.drive_scans	List of scans against MS365 drives
	ms365.drive_scans_history	Historical summary of drive scans
	ms365.drives	List of MS365 drives (document libraries, OneDrive for Business drives)
	ms365.group_drives	Mapping of MS365 groups (teams) to associated drives
	ms365.group_member_types	Enumeration table of group member types
	ms365.group_members	MS365 group membership associations
	ms365.group_owners	MS365 group owner associations
	ms365.group_sites	Mapping of MS365 groups (teams) to associated sites
	ms365.groups	List of discovered MS365 groups
	ms365.identity_types	Enumeration table of identity types
	ms365.jobs	List of jobs to enumerate MS365 tenant objects (teams, sites, groups, users, drives, etc.)

## 2 - Supported Constructs

Category	Table Name	Notes
	ms365.jobs_history	Historical summary of tenant scans
	ms365.permissions	Sharing links and direct access permissions for drive items
	ms365.sharing_link_members	List of security principals associated with a specific sharing link
	ms365.sites	List of discovered MS365 SharePoint sites
	ms365.sp_base_permissions	Lookup table for SharePoint permission levels / roles.
	ms365.sp_group_members	SharePoint group member associations
	ms365.sp_groups	List of SharePoint groups
	ms365.sp_permission_levels	List of SharePoint permission levels / roles
	ms365.sp_permissions	List of SharePoint permissions (assigned permission levels)
	ms365.sp_site_permissions	List of SharePoint site permissions
	ms365.sp_users	List of SharePoint users
	ms365.team_channels	List of discovered Teams Channels
	ms365.teams	List of discovered MS365 Teams
	ms365.tenants	Configured MS365 tenants for scan
	ms365.user_drives	Mapping of MS365 users to drives (OneDrive for Business drives)
	ms365.users	List of discovered MS365 users
<b>Session Specific</b>	tmp_cq_fs_paths	Temporary table injected into custom query sessions for report-defined target paths

## 2.4 - Supported Views

Category	View Name	Notes
<b>Windows File System</b>	srs.current_fs_scans	List of Current file system scans
	srs.current_permissions_scans	List of Current permissions scans
	srs.previous_fs_scans	List of Previous file system scans
	srs.previous_permissions_scans	List of Preview permissions scans
	srs.baseline_fs_scans	List of Baseline file system scans
	srs.baseline_permissions_scans	List of Baseline permissions scans
	srs.current_fs_scandata	List of all Current file system scan data
	srs.previous_fs_scandata	List of all Previous file system scan data
	srs.baseline_fs_scandata	List of all Baseline file system scan data
	srs.current_ntfs_aces	All Current permissions scan data for NTFS-compatible file systems
	srs.previous_ntfs_aces	All Previous permissions scan data for NTFS-compatible file systems
	srs.baseline_ntfs_aces	All Baseline permissions scan data for NTFS-compatible file systems
<b>Active Directory</b>	ad.ds_objects_view	All primary properties from ad.ds_objects and ad.domains with binary GUIDs and SIDs converted to equivalent text variants.

## 2.5 - Supported Functions

Category	View Name	Description
<b>General</b>	srs.byte_string	Converts raw number to byte string such as 10 MB or 3.25 KB
	srs.byte_unit_string	Converts raw number to byte string with specified unit and precision.
	srs.attribute_string	Converts attributes to string representation
	srs.guid_bytes	Converts GUID from string to binary
	srs.guid_text	Converts GUID from binary to string
	srs.path_hash	Calculates SHA-1 hash of lowercase input (typically a path)
	srs.path_hash_sha256	Calculates SHA256 hash of lowercase input (typically a path)
	srs.bytes_to_hex_string	Converts byte array to equivalent hex string
	srs.hex_string_to_bytes	Converts hex string to equivalent byte array
<b>Identity Systems</b>	srs.sid_bytes	Converts SID from string to binary
	srs.sid_text	Converts SID from binary to string
	srs.ad_account_name	Combines AD account name elements into a single display name
<b>Permissions</b>	srs.access_mask_basic_string	Converts access mask value to basic permissions string
	srs.access_mask_string	Converts access mask value to string representation
	srs.ace_flags_string	Translates ACE flag to string values
	srs.ace_type_string	Translates ACE type to string value

## 3 - Navigating Scan Data

Writing queries that are both useful and accurate require a proper understanding of how to navigate collected scan data.

Due to the nature of how File Reporter collects and curates scan data, this section is broken up by resource type. In addition, it also provides guidance on how to report across these resource types in a single report query when applicable.

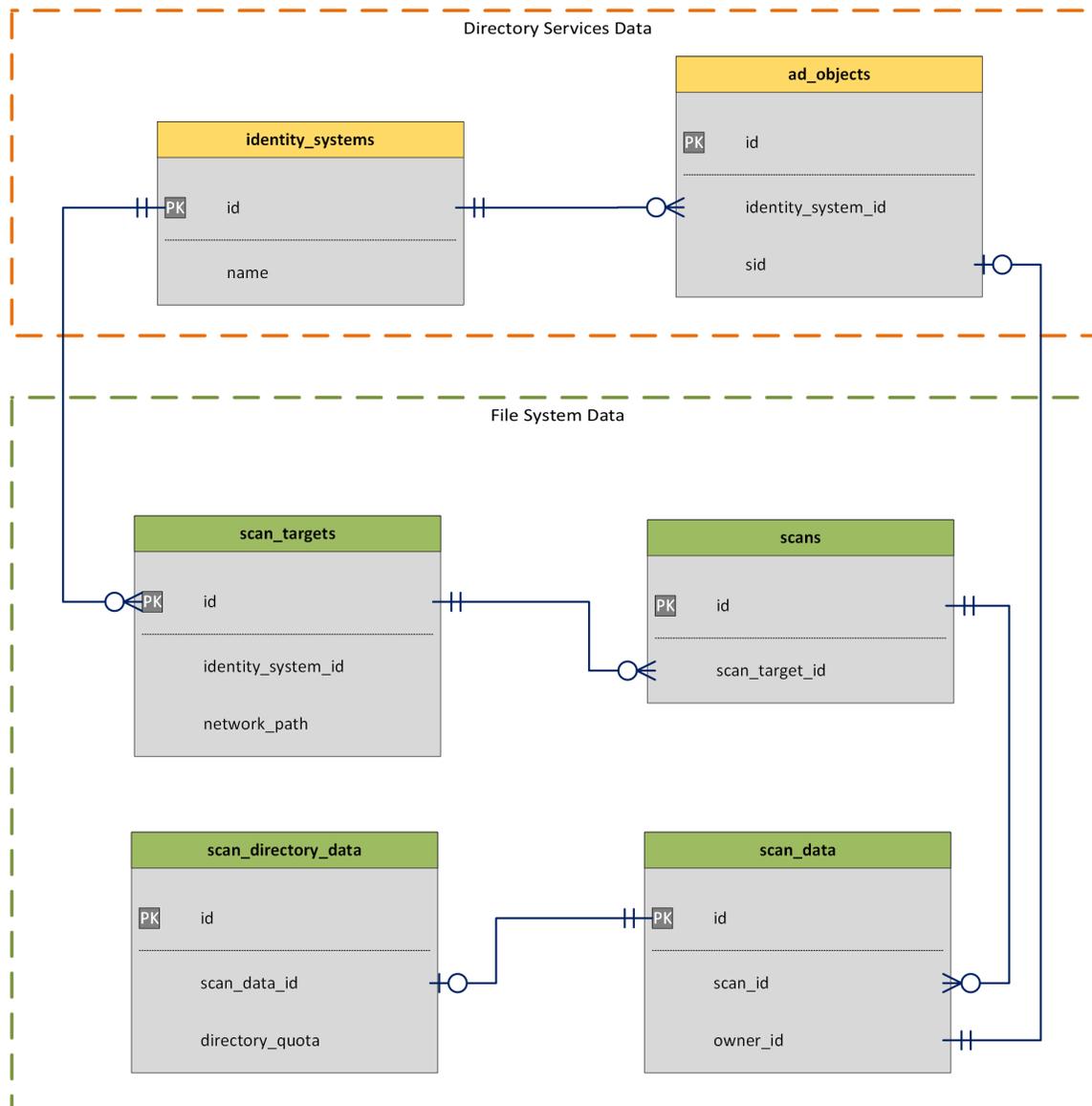
## 3.1 - Windows File System

### 3.1.1 - Table Relationships

#### Windows File System Metadata

The collected scan data is generally broken down into three major areas: Identity System info, File System data, and Permissions data.

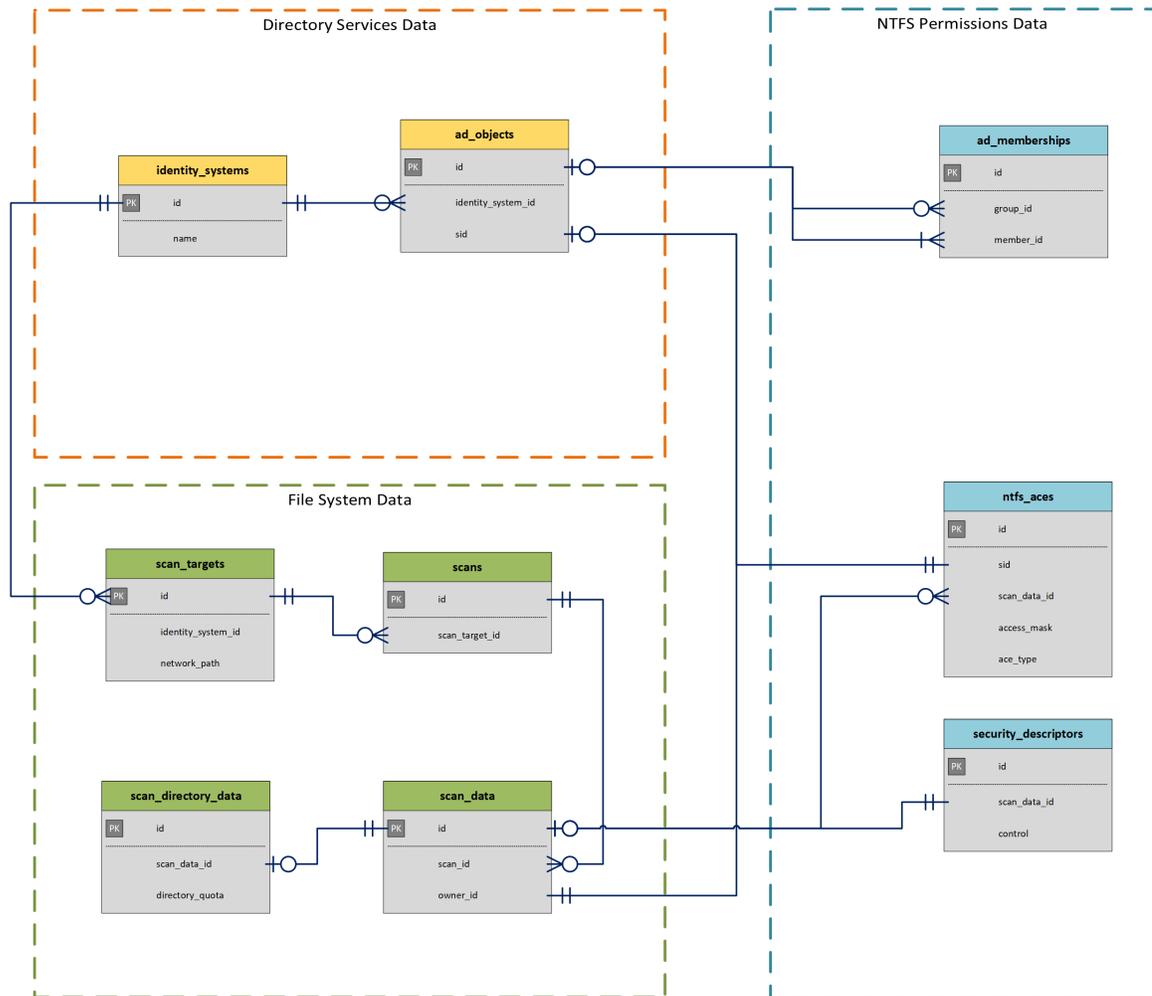
For general file system metadata collection, only file system data is collected, along with minimal identity system data pertaining to file and folder owners.



## Windows File System Permissions

NTFS Permissions data is limited to folder structure as well as assigned and inherited NTFS access control entries (ACEs).

It should be noted that permissions scans do not include metadata specific information such as directory quota, nor do they include any file-entry data that is not a folder. Only permissions for folder, share, and DFS entries are currently collected.



### 3.1.2 - Scoping and Filtering

Scoping is the process by which selected data is limited to areas of interest. Areas of interest may include all file system data related to a specific identity system, or only data within one or more subdirectories. Additionally, data could be scoped as it relates to a given owner or trustee.

#### Scope by Identity System

Scoping by identity system is as simple as limiting a query to a specific `srs.identity_system.id` value, or using one of the supported `srs.current_*` views, a specific identity system name.

### 3 - Navigating Scan Data

The following example selects file system data from a given identity system, limited to 100 entries.

#### Example (SQL Server)

```
1 | SELECT TOP(100) *
2 | FROM srs.current_fs_scandata
3 | WHERE identity_system = 'ad.test.lab';
```

#### Example (PostgreSQL)

```
1 | SELECT *
2 | FROM srs.current_fs_scandata
3 | WHERE identity_system = 'ad.test.lab'
4 | LIMIT 100;
```

#### Scope by Server

Scoping by server is as simple as filtering by the server column in the *srs.scan\_targets* table or in one of the supported *srs.current\_\** views.

Also note that the server name may be case sensitive depending on the database collation.

The following example selects all file system data from a specific server, limited to 100 entries.

#### Example (SQL Server)

```
1 | SELECT TOP(100) *
2 | FROM srs.current_fs_scandata
3 | WHERE server = 'server1.ad.test.lab';
```

#### Example (PostgreSQL)

```
1 | SELECT *
2 | FROM srs.current_fs_scandata
3 | WHERE server = 'server1.ad.test.lab'
4 | LIMIT 100;
```

### Scope by Scan Target

Scoping by scan target is useful where a specific CIFS share name or DFS target is known.

Note that the scan target name may be case sensitive depending on the database collation.

Example: select file system data from a particular scan target (share or volume) limited to 100 entries

#### Example (SQL Server)

```

1 | SELECT TOP(100)
2 |     *
3 | FROM srs.current_fs_scandata
4 | WHERE scan_target = '\\server1.ad.test.lab\Data';

```

#### Example (PostgreSQL)

```

1 | SELECT
2 |     *
3 | FROM srs.current_fs_scandata
4 | WHERE scan_target = '\\server1.ad.test.lab\Data'
5 | LIMIT 100;

```

### Scope by Directory

Scoping by a particular directory or folder requires the use of the hierarchical markers in the *srs.scan\_data* table.

These markers assist with determining parent and child folders as well as all subordinate file system entries for a given directory or set of directories.

Field	Description	Notes
<b>idx</b>	Entry index	Unique per scan
<b>parent_idx</b>	Index of parent directory, share, or DFS name space entry	All sibling file system entries will have the same parent index.
<b>path_depth</b>	Current path depth relative to the root path	The root path is always depth zero (0). Other paths such as shares may have the same

### 3 - Navigating Scan Data

Field	Description	Notes
		depth as the root path, but can be distinguished by <i>path_type</i> .  Entries occurring above the root path (such as DFS name spaces) will have a negative value.
<b>ns_left</b> <b>ns_right</b>	Nested set indexes for current entry	Nested set markers provide a quick way to determine all subordinates for a given directory.  See examples below for details.

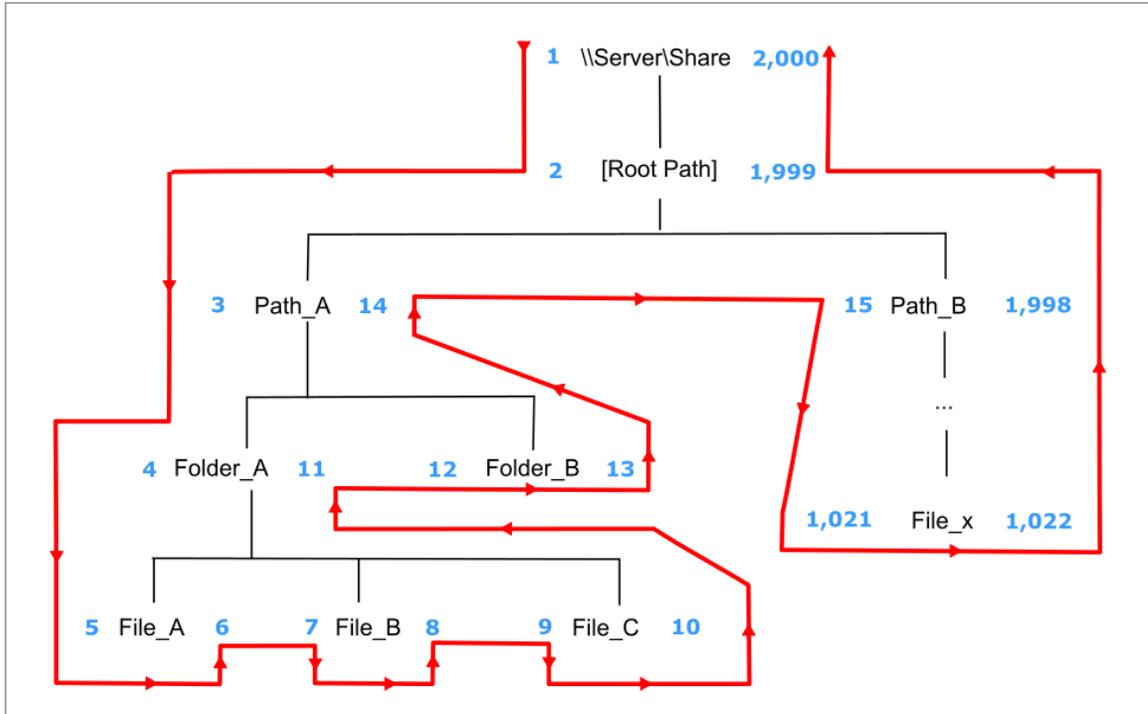
The following example selects all NTFS file system entries subordinate to and including the specified target path.

#### Example - Scope by Directory

```
1 WITH root_path AS (  
2     SELECT  
3         sd.ns_left,  
4         sd.ns_right,  
5         sd.scan_id  
6     FROM srs.current_fs_scandata_ad AS sd  
7     WHERE sd.fullpath_hash = srs.path_hash  
8           ('\\server1.ad.test.lab\Share\path\subpath')  
9           AND sd.path_type = 2  
10 )  
11 SELECT  
12     sd.*  
13 FROM srs.current_fs_scandata_ad AS sd  
14 JOIN root_path AS rp ON rp.scan_id = sd.scan_id  
15     AND rp.ns_left <= sd.ns_left  
16     AND rp.ns_right >= sd.ns_right;
```

In this example, we are using two SELECT statements: one to get the information for the desired root path, and one to pull all subordinate entries along with the root path. Notice how the JOIN filter in the second SELECT statement uses not only the *scan\_id* to limit the particular scan(s) of interest, but also uses the *ns\_left* and *ns\_right* fields to keep the data set limited to file entries in the folder hierarchy.

In the following diagram, an example of the nested set model calculations are shown with an example structure under \\Server\Share. In this example, exactly 1,000 file system entries exist, including files, folders, and the share itself.



For each node in the scanned file structure, a left (*ns\_left*) and right (*ns\_right*) value are assigned. The values are assigned by traversing the imaginary path from the root down the left side of the structure, incrementing the *ns\_left* values by one. Once a leaf node is encountered, the incrementing value continues, but is now assigned to *ns\_right*.

This process continues until the entire graph of the file structure has been traversed, and the root path is finally assigned the last number for its *ns\_right* value.

The nested set model has the following characteristics, some of which are vital to hierarchical processing, such as determining subordinate objects:

- The root path will always have a *ns\_left* value of 1 and an *ns\_right* value of  $2n$ , where  $n$  = the total number of entries
- For any given container object (folder, share, etc.), all subordinate entries can be found by searching for all objects in the scan having an *ns\_left* value greater than the container path's *ns\_left* value, and an *ns\_right* value less than the container path's *ns\_right* value.
- Nested set is generally the fastest method available in relational data models for retrieving all subordinate objects when representing hierarchical data.

For more information on the nested set model, see [https://en.wikipedia.org/wiki/Nested\\_set\\_model](https://en.wikipedia.org/wiki/Nested_set_model).

### Scope by Directory with Path Depth

In addition to scoping by directory, it may be useful to start with a given path, but then only include subordinate paths within a given range below the selected path.

### 3 - Navigating Scan Data

In this case, we make use of the same nested set model calculations seen in the previous section, but include the use of the `path_depth` parameter as well.

The following example selects all paths starting two levels below a given path:

#### Example - Start with Path Depth 2

```
1 | WITH root_path AS (  
2 |     SELECT  
3 |         sd.ns_left,  
4 |         sd.ns_right,  
5 |         sd.scan_id,  
6 |         sd.path_depth  
7 |     FROM srs.current_fs_scandata_ad AS sd  
8 |     WHERE sd.fullpath_hash = srs.path_hash  
9 |     ('\\server1.ad.test.lab\Share\Groups')  
10 |     AND sd.path_type = 2  
11 | )  
12 | SELECT sd.*  
13 | FROM srs.current_fs_scandata_ad AS sd  
14 | JOIN root_path AS rp ON rp.scan_id = sd.scan_id  
15 |     AND rp.ns_left <= sd.ns_left  
16 |     AND rp.ns_right >= sd.ns_right  
17 |     AND sd.path_depth > rp.path_depth + 2;    -- Upper bound
```

This example is common when folder structures have managed content, such as collaborative or group folders, organized below division or department folders one or more layers deep.

In order to pull all the content from just the group folders themselves, and not include the structural folders, we can make use of path depth, but assign the selected path to the root structural folder.

For a share organized as:

```
\\Server\Share\Groups\Departments\GroupA
```

the selected path could be `\\Server\Share\Groups` and the `path_depth` could be assigned to the `root_path + 2` or greater, as in the `SELECT` statement above.

We could just as easily limit the depth of paths searched by adding another comparison of `path_depth` as a lower bounds:

**Example - Upper and Lower Path Depth**

```

1 | WITH root_path AS (
2 |     SELECT
3 |         sd.ns_left,
4 |         sd.ns_right,
5 |         sd.scan_id,
6 |         sd.path_depth
7 |     FROM srs.current_fs_scandata_ad AS sd
8 |     WHERE sd.fullpath_hash = srs.path_hash
9 |           ('\\dbdev.db.dtest.lab\home')
10 |    AND sd.path_type = 2
11 | )
12 | SELECT sd.*
13 | FROM srs.current_fs_scandata_ad AS sd
14 | JOIN root_path AS rp ON rp.scan_id = sd.scan_id
15 | AND rp.ns_left <= sd.ns_left
16 | AND rp.ns_right >= sd.ns_right
17 | AND sd.path_depth > rp.path_depth + 2 -- Upper bound
18 | AND sd.path_depth < rp.path_depth + 3; -- Note that we have
19 | a lower bound as well

```

**Scope by Security Principal**

Scoping by security principal is useful when querying for scan data specific to a given set of owners or trustees.

This example selects all files for a given server owned by a specific AD user, limited to 100 entries.

**Example (SQL Server)**

```

1 | SELECT TOP(100) *
2 | FROM srs.current_fs_scandata_ad
3 | WHERE owner_domain = 'AD'
4 |    AND owner_name = 'user1';

```

**Example (PostgreSQL)**

```

1 | SELECT *
2 | FROM srs.current_fs_scandata_ad

```

### 3 - Navigating Scan Data

```
3 | WHERE owner_domain = 'DB'  
4 |     AND owner_name = 'test1'  
5 | LIMIT 100;
```

This next example selects all folders where a user is a direct trustee (not inherited) for NTFS folders, limited to 100 entries.

#### Example (SQL Server)

```
1 | SELECT TOP(100) *  
2 | FROM srs.current_ntfs_aces  
3 | WHERE trustee_domain = 'DB'  
4 |     AND trustee_name = 'test1'  
5 |     AND ace_flags & 16 <> 16;
```

#### Example (PostgreSQL)

```
1 | SELECT *  
2 | FROM srs.current_ntfs_aces  
3 | WHERE trustee_domain = 'DB'  
4 |     AND trustee_name = 'test1'  
5 |     AND ace_flags & 16 <> 16  
6 | LIMIT 100;
```

#### Basic Filtering

In addition to using filters to scope the range of scan data, basic filtering can also be used to limit the results to only records of interest.

The following is a list of basic filtering examples that may be used as starting templates for queries.

##### Filter by Path Type

In cases where aggregation or calculations against a discrete set of files is desired, it may be necessary to filter out any directories or shares first, since those entries contain size and name data that may skew the desired results.

```
SELECT *  
FROM srs.current_fs_scandata_ad
```

```
WHERE path_type = 1          -- Note: 1 = file entry
      AND server='Server1';
```

#### Filter by File Extension

This example filters the set of file entries within a given directory structure to just those defined as media types.

```
SELECT *
FROM srs.current_fs_scandata_ad
WHERE path_type = 1
      AND filename_extension IN ('mp3', 'mp4', 'avi', 'ogg', 'png',
      'jpg', 'jpeg');
```

Note that for *filename\_extension*, all values should be lower case.

#### Filter by Date Range

This example selects all files on the specific server from November 1, 2013 midnight, through November 2, 2013 11:59 PM.

```
SELECT *
FROM srs.current_fs_scandata_ad
WHERE modify_time BETWEEN '2013-11-01 00:00:00' AND '2013-11-02
23:59:59'
      AND server='dbdev.db.dtest.lab'
      AND path_type = 1  -- Files only
```

We can also use the familiar  $\geq$  and  $\leq$  comparison operators to accomplish the same:

```
SELECT *
FROM srs.current_fs_scandata_ad
WHERE modify_time >= '2013-11-01 00:00:00'
      AND modify_time <= '2013-11-02 23:59:59'
      AND server='dbdev.db.dtest.lab'
      AND path_type = 1  -- Files only
```

Note that the behavior of the BETWEEN operator is inclusive, not exclusive, to the parameters given.

### 3 - Navigating Scan Data

It is important to note with date-time ranges, that a simple date such as '2013-11-02' actually represents '2013-11-02 00:00:00', so be careful to include 23:59:59 to the ending date as appropriate.

Finally, it is important to remember that all timestamps stored in the database are stored as UTC values, so consideration for time zone offsets may be needed.

#### Filter by File Name

This example shows how to filter by a given file name.

```
SELECT *
FROM srs.current_fs_scandata
WHERE LOWER(name) = 'document1.txt';
```

Note the use of the LOWER operator to force a case-insensitive search. Depending on the collation of the database instance and the database itself, this operator may be required.

For wildcard matches, the standard SQL flags \_ and % can be used to represent single or multiple characters.

```
SELECT *
FROM srs.current_fs_scandata
WHERE LOWER(name) LIKE 'document1.%';
```

See the following links for database specific info regarding wildcards and other search patterns:

- SQL Server: <https://msdn.microsoft.com/en-us/library/ms190301>
- PostgreSQL: <https://www.postgresql.org/docs/current/static/functions-matching.html>

#### 3.1.3 - File System Target Paths

You can define and manage a Custom Query report's selected target paths via the report definition itself separate from any associated SQL queries.

This process is accomplished via a temporary table that is injected into the SQL query session at runtime when using any of the File Reporter tools such as Report Designer or the SQL query editor in the File Reporter web application for Custom Query reports.

Newer report templates available on the File Query Cookbook site (<https://filequerycookbook.com>) make use of this construct which provides a more hands-off approach for users not as comfortable with modifying SQL queries directly but who need the flexibility to define and change a report's file system target paths.

### Example Query

To understand this process, the following example illustrates a custom query that reports on NTFS file system permissions for one or more target paths selected with the *File System Target Paths* dialog in Report Designer.



**IMPORTANT:** SQL Server requires a hash '#' prefix when referencing temporary tables.

When using SQL Server as the backend database, be sure that any references to `tmp_cq_fs_paths` in your SQL queries are changed to `#tmp_cq_fs_paths` instead.

Conversely, PostgreSQL cannot use hash marks '#' as part of the table name, so be sure that this prefix does not exist in your SQL queries when using PostgreSQL as the backend database.

1. Launch the File Reporter Report Designer application and create a new empty report.  
See *Creating a Report* in the *File Reporter Client Tools Guide* for details.
2. Depending on the database in use, enter one of the following SQL queries into the SQL query editor dialog.

#### Example (SQL Server)

```

1 | SELECT
2 |   *
3 | FROM srs.current_ntfs_aces AS ace
4 | JOIN #tmp_cq_fs_paths AS cq
5 |   ON cq.target_path_hash = ace.fullpath_hash
6 |   AND cq.is_current = 'true'
7 |   AND cq.is_permission_scan = 'true';

```

#### Example (PostgreSQL)

```

1 | SELECT
2 |   *
3 | FROM srs.current_ntfs_aces AS ace
4 | JOIN tmp_cq_fs_paths AS cq
5 |   ON cq.target_path_hash = ace.fullpath_hash
6 |   AND cq.is_current = 'true'
7 |   AND cq.is_permission_scan = 'true';

```

### 3 - Navigating Scan Data

3. Click *Save* to save the SQL query.
4. Click *File System Paths* to open the File System Target Paths dialog.
5. Select one or more paths to report on then save the selection.

Be sure to select paths that are marked as having Permissions scan data available as seen in the *File System Target Paths* dialog.

6. Click *Execute Query* to run the SQL query and see the results.

#### Using Alternate SQL Query Editors

When developing a SQL query for a Custom Query report, you may wish to develop the query itself in a SQL query editor of your choice, such as SQL Server Management Studio (SSMS) or PgAdmin for PostgreSQL.

In these development environments, the injected temporary table is not available by default. To stage the temporary table, use the following approach.



**IMPORTANT:** Although any existing report definition may be used as a reference, we strongly advise creating a new Report Definition and using its associated ID.

This process allows flexibility for changing the selected target paths during the query design phase without impacting other report definitions.

1. Create a new Custom Query Report.  
See *Creating a Report* in the *File Reporter 24.2 Client Tools Guide*.
2. Assign one or more File System target paths to the report definition.  
See *File System Paths Selector* in the *File Reporter 24.2 Client Tools Guide*.
3. Find the report ID for the newly created report.

Report Name	Report Type	Paths	Report Owner	Last Modified	Id
ntfs_aces in path	Custom Query		sp\administrator	3/1/2022 1:13:...	68
Extension report by category - summary	Custom Query		sp\administrator	2/23/2022 2:17:...	45
long path and filenames	Custom Query		sp\administrator	2/23/2022 1:39:...	37
Copy Of long path and filenames	Custom Query		sp\administrator	2/23/2022 1:38:...	67
Group Memberships ms365	Custom Query		sp\administrator	2/23/2022 1:35:...	2
Files created in the future, or modified before created	Custom Query		sp\administrator	2/23/2022 1:02:...	56
FR 4.1 - Security2 - Disabled Inheritance on Department Share	Custom Query		sp\administrator	2/21/2022 2:54:...	65
FR 4.1 - Direct Folder Permissions with AD Attributes V1	Custom Query		sp\m1-localadmin	2/17/2022 2:08:...	57
duplicate file - hash	Custom Query		sp\administrator	2/17/2022 1:36:...	3
Extension report by category - detailed	Custom Query		sp\administrator	2/15/2022 7:08:...	46
Duplicate Files across Tenants	Custom Query		sp\administrator	2/15/2022 6:53:...	14
query builder	Custom Query		sp\m1-localadmin	2/11/2022 5:05:...	62
test00	Custom Query		sp\m1-localadmin	2/11/2022 4:56:...	61
Security - Users with Direct Access to Folders	Custom Query		sp\administrator	2/11/2022 4:04:...	60

- In the Main form of the Report Designer, find the name of the newly created report definition.
  - The column at the far right of the grid indicates the ID for each report. Make note of the new report definition's ID number.
4. Insert the following SQL code at the start of the query.

#### Example (SQL Server)

```

1 | IF OBJECT_ID('#tmp_cq_fs_paths', 'U') IS NULL
2 |     SELECT * INTO #tmp_cq_fs_paths
3 |     FROM srs.cq_fs_paths_by_report_id(17);

```

#### Example (PostgreSQL)

```

1 | CREATE TEMP TABLE IF NOT EXISTS tmp_cq_fs_paths AS
2 | SELECT * FROM srs.cq_fs_paths_by_report_id(17);

```

- Be sure to change the example's report ID of "17" to the report ID identified from the previous step.
- Add SQL statements as needed to complete the query.

### 3 - Navigating Scan Data

7. When the SQL query development is complete, copy all of the SQL statements into the Custom Query report definition except for the initial lines used to stage the temporary table.

Using the example query from earlier, a complete query using a staged temporary table with an alternate SQL query editor looks as follows:

#### Example (SQL Server)

```
1 | IF OBJECT_ID('#tmp_cq_fs_paths', 'U') IS NULL
2 |     SELECT * INTO #tmp_cq_fs_paths
3 |     FROM srs.cq_fs_paths_by_report_id(17);
4 |
5 | SELECT
6 |     *
7 | FROM srs.current_ntfs_aces AS ace
8 | JOIN #tmp_cq_fs_paths AS cq
9 |     ON cq.target_path_hash = ace.fullpath_hash
10 |    AND cq.is_current = 'true'
11 |    AND cq.is_permission_scan = 'true';
```

#### Example (PostgreSQL)

```
1 | CREATE TEMP TABLE IF NOT EXISTS tmp_cq_fs_paths AS
2 | SELECT * FROM srs.cq_fs_paths_by_report_id(17);
3 |
4 | SELECT
5 |     *
6 | FROM srs.current_ntfs_aces AS ace
7 | JOIN tmp_cq_fs_paths AS cq
8 |     ON cq.target_path_hash = ace.fullpath_hash
9 |    AND cq.is_current = 'true'
10 |    AND cq.is_permission_scan = 'true';
```

## 3.2 - Active Directory Identities

The extended data for Active Directory identities is stored in the *ad.domains* and *ad.ds\_objects* tables.

The tables used to map basic identity information for owners and permission trustees may be joined to these tables for extended information.





## 4 - Example Scenarios

### 4.1 - Content Hash Duplicate File Reports

A Content Hash Duplicate File report provides more advanced duplicate file detection over the Duplicate File built-in report which compares only file names and metadata.



**NOTE:** For information on collecting content hashes, see Creating A Scan Policy in the *File Reporter 24.2 Administration Guide*.

Through <https://filequerycookbook.com> you can copy and paste the Content Hash Duplicate File Report custom query into the Query Editor and import a report layout into the Report Designer. This custom query and associated report identifies duplicate files based on hash comparisons and the parameters you set.

#### 4.1.1 - Determining Prerequisites

- Create a file system scan policy for each of the target paths on which you want to report.
- With the *Generate content file hashes* option selected in the Scan Policy Editor of each scan policy, conduct a file system scan on each target path.
- Install the Client Tools.

The Client Tools include the Query Editor and the Report Designer that will be used in these procedures.

- Decide how you want the report to be generated and follow the applicable procedures.
  - To generate a delimited text file that you can take into other tools for customized searching and presentations you can copy or create an SQL query with the query editor covered in Creating a Report in the *File Reporter 24.2 Client Tools Guide*
  - To generate the report using the Report Designer and produce a formatted report layout, proceed with Using Report Designer in the *File Reporter 24.2 Client Tools Guide*.

#### 4.1.2 - Designing the Report

This option lets you utilize both the custom query and the associated report layout design for the “Content Hash Duplicate File Report” from <https://filequerycookbook.com>.

## 4 - Example Scenarios



**NOTE:** A detailed discussion of the Report Designer along with procedures for familiarizing yourself with the interface are available in Using Report Designer in the *File Reporter 24.2 Client Tools Guide*.

1. On the File Query Cookbook site <https://filequerycookbook.com> locate and download the “Content Hash Duplicate File Report.”

The file is saved as a zip archive.

2. Unzip the downloaded file and open the .sql file in a text editor.

You will eventually paste this custom query into the Query Editor.

3. From the *Start* menu, launch the File Reporter 24.2 Report Designer.

Web Service URI:

User Name:

Password:

4. Enter the login credentials and click *Login*.

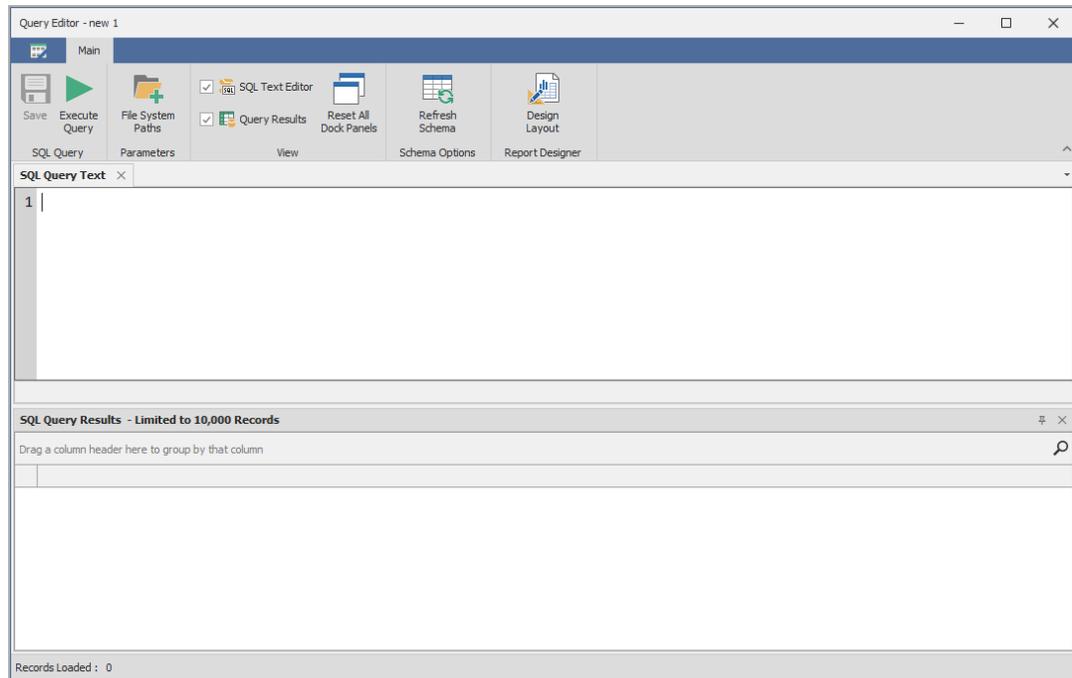
Report Name	Report Type	Paths	Report Owner	Last Modified	Id
No i...	Custom Query	=	#D:	=	=
ATL SFO duplicate hash	Custom Query		2 sp\administrator	2/24/2022 6:4...	3
Owner report using cq paths and ad.ds_objects	Custom Query		1 sp\administrator	2/11/2022 4:4...	1

Report Type = Custom Query

All of your saved Custom Query reports are listed.

- Click *New Custom Query*, give it a name, then click *Create*.

The Report Designer Query Editor is launched.



- From the text editor you used in Step 2, copy the custom query and paste it into the Query Editor.
- In the line beginning with `WHERE`, edit the UNC paths so that they are specific to the content file hashed shares on which you want to report.

The custom query only includes two paths so if you want more, extend the line to include more paths by adding `srs.path_hash(' \\server\share\path')` to the comma delimited `sd.fullpath_hash IN` portion of the where clause for each desired path.

- (Conditional) At the bottom of the custom query, modify the `q.item_count` and `q.size` settings to the minimum number of duplicates and file sizes (in bytes), respectively, to include in the report.
- Click *Execute* to see a preview of the report data.

## 4 - Example Scenarios

The screenshot shows the SQL Query Editor interface. The top toolbar includes buttons for Save, Execute Query, File System Paths, SQL Text Editor, Query Results, Reset All Dock Panels, Refresh Schema, and Design Layout. The main area displays the following SQL query:

```

1 WITH
2 q(fullpath, size, create_time, modify_time, access_time, name, item_count, total_hash_size, content_hash) AS (SELECT sd.fullp
3     sd.size,
4     sd.create_time,
5     sd.modify_time,
6     sd.access_time,
7     sd.name,
8     COUNT(*) OVER (PARTITION BY sd.content_hash) AS item_count,
9     Sum(sd.size) OVER (PARTITION BY sd.content_hash) AS total_hash_size,
10    srs.bytes_to_hex_string(sd.content_hash) as content_hash
11 FROM srs.srs_data AS sd

```

Below the query, the 'SQL Query Results - Limited to 10,000 Records' pane shows a table with the following columns: fullpath, item\_count, size\_string, total\_size\_string, wasted\_space, wasted\_space\_string, total\_hash\_size, size, and content\_hash. The table contains 8 rows of data. At the bottom, it indicates 'Records Loaded: 184'.

10. Click **Save**.

11. Click **Design Layout**.

The screenshot shows the Report Designer interface. The top toolbar includes buttons for Open..., Save, Edit Query, Download All Data, File System Paths, Refresh Data Bindings, Cut, Copy, Paste, Undo, Redo, Font settings (Times New Roman, 9.75), Alignment, Layout, Zoom Out, Zoom In, Windows, and Scripts. The main area displays a report layout with a grid and a 'Detail' section. The 'Group and Sort' pane is visible at the bottom, and the 'Properties' pane on the right shows settings for 'TopMargin' and 'Background Color'.

12. Click **Open**.

13. Locate the **.repx** file that you saved and unzipped in Step 2 and click **Open**.

The layout template appears in the Report Designer.

14. Click *Download All Data*.
15. In the subsequent dialog box, click *Yes*.

This runs the query in the database and loads data into the report template.

16. Click *Print Preview* to review the report findings.

Note how the hashes are listed with a total number for each and the location of each, meaning the total number of duplicate files and their locations.

17. Save the report by doing one of the following:
  - From the *Export To* drop-down menu, select the file type you want to save the report layout to.
  - Click *Save Report* to save the report as a .prnx file that you can open in the Report Viewer and if you want later, export the report to the desired file type.

## 4.2 - Microsoft 365 Reports

Once Agent365 has scanned the data and associated permissions for Microsoft 365 file repositories, including OneDrive for Business, SharePoint Online document libraries, and Teams document libraries, you can use the prebuilt custom queries and associated report layouts in <https://filequerycookbook.com> to generate reports.

### 4.2.1 - Determining Prerequisites

- Install and configure Agent365. See Agent365 in the *File Reporter 24.2 Installation Guide*.
- Scan the tenant. See Tenants in the *File Reporter 24.2 Administration Guide*.
- Install the Client Tools. See Installing the Client Tools in the *File Reporter 24.2 Client Tools Guide*.

### 4.2.2 - Designing the Report

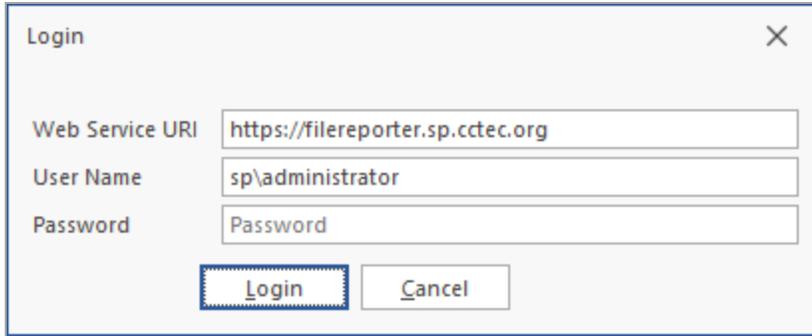
The Client Tools include the Report Designer application that will be used in these procedures.

1. Using File Query Cookbook located at <https://filequerycookbook.com>, locate and download one of the custom queries and associated reports for Microsoft 365.

The file is saved as a zip archive.

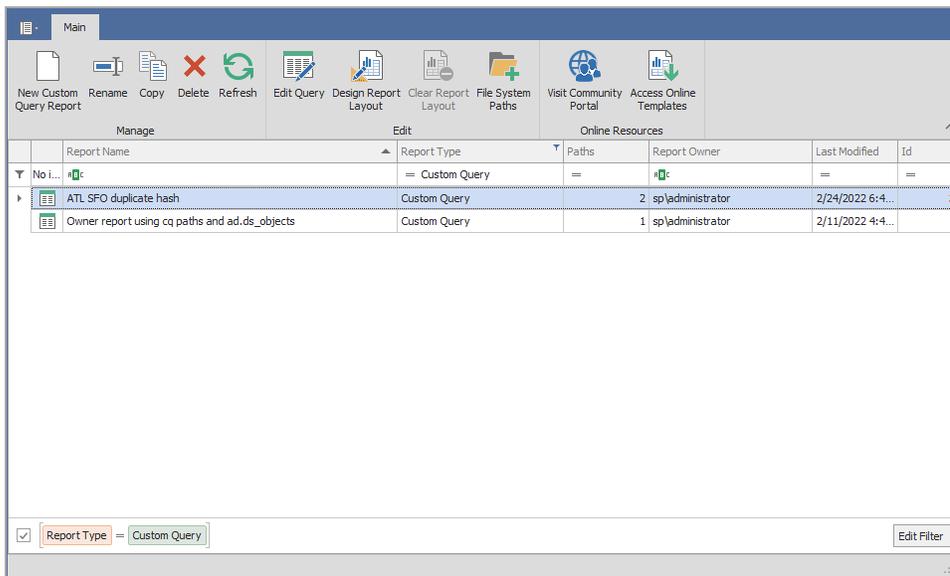
2. Unzip the downloaded file and open the .sql file in a text editor.  
You will eventually paste this custom query into the Query Editor.
3. From the *Start* menu, launch the File Reporter 24.2 Report Designer.

## 4 - Example Scenarios



A login dialog box titled "Login" with a close button (X) in the top right corner. It contains three text input fields: "Web Service URI" with the value "https://filereporter.sp.ctec.org", "User Name" with the value "sp\administrator", and "Password" with the value "Password". Below the fields are two buttons: "Login" and "Cancel".

4. Enter the login credentials and click *Login*.



A screenshot of the Report Designer interface. The top menu bar includes "Main" and various icons for actions like "New Custom Query Report", "Rename", "Copy", "Delete", "Refresh", "Edit Query", "Design Report Layout", "Clear Report Layout", "File System Paths", "Visit Community Portal", and "Access Online Templates". Below the menu is a table listing saved Custom Query reports.

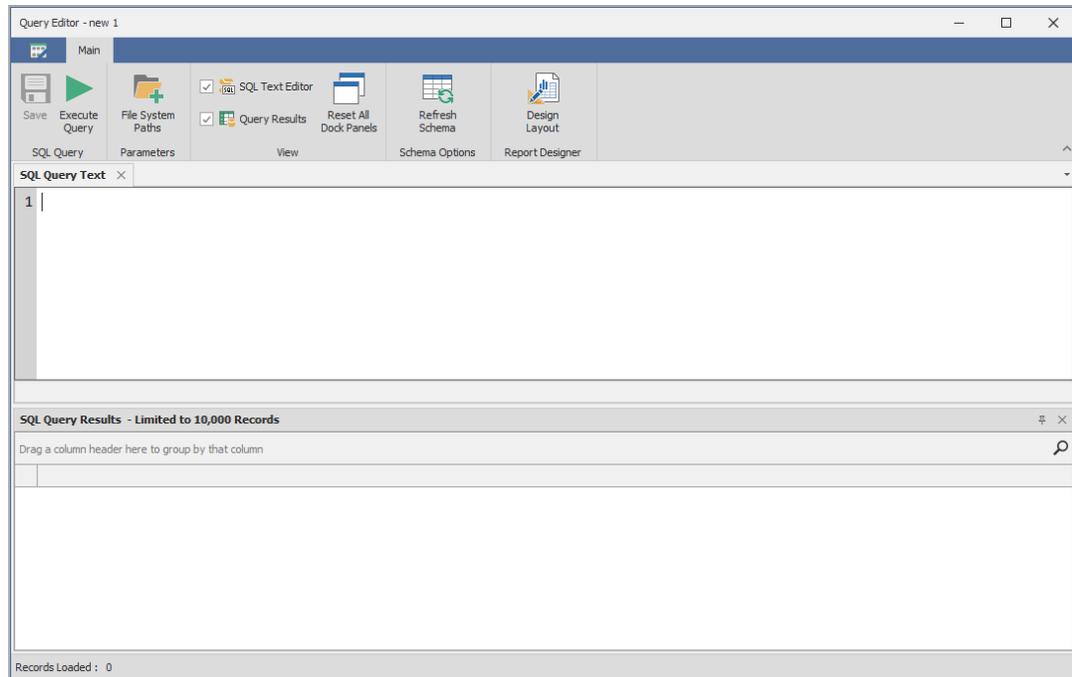
Manage		Edit		Online Resources	
Report Name	Report Type	Paths	Report Owner	Last Modified	Id
No l...	Custom Query	=	#D	=	=
ATL SFO duplicate hash	Custom Query	2	sp\administrator	2/24/2022 6:4...	3
Owner report using cq paths and ad.ds_objects	Custom Query	1	sp\administrator	2/11/2022 4:4...	1

At the bottom, there is a filter bar with a checked checkbox, a dropdown menu showing "Report Type = Custom Query", and an "Edit Filter" button.

All of your saved Custom Query reports are listed.

5. Click *New Custom Query*, give it a name, then click *Create*.

The Report Designer Query Editor is launched.



6. From the text editor you used in [Unzip the downloaded file and open the .sql file in a text editor. \(page 33\)](#) copy the custom query and paste it into the Query Editor.
7. (Conditional) If there are target paths or other modifications that need to be made for your environment, follow the procedures for the recipe.
8. Click *Execute* to get a preview of the report data in the bottom portion of the editor.

## 4 - Example Scenarios

The screenshot shows the SQL Query Editor interface. The top toolbar includes buttons for Save, Execute Query, File System Paths, SQL Text Editor, Query Results, Reset All Dock Panels, Refresh Schema, and Design Layout. The main area contains the following SQL query:

```
6      di.modified_by,  
7      COUNT(*) OVER (PARTITION BY di.file_hash) AS total_hash_count,  
8      di.item_type,  
9      RIGHT(pp.web_url, length(pp.web_url) - length(d.web_url)) AS parent_path,  
10     d.web_url AS drive_path,  
11     srs.bytes_to_hex_string(di.file_hash) as file_hash,  
12     CASE  
13     WHEN udm.id IS NOT NULL THEN 'OneDrive'  
14     WHEN gdm.id IS NOT NULL THEN 'Teams'  
15     ELSE 'SharePoint'  
16     END AS drive_category
```

Below the query, the 'SQL Query Results - Limited to 10,000 Records' pane displays a table with the following data:

drive_path	parent_path	filename	item_size	total_hash
https://condreycorprpl-my.sharepoint.com/personal/gnance_sp_ctec_org/Documents	/Microsoft%20Teams%20Chat%20Files	Meeting Notes 20201020.txt	67	
https://condreycorprpl-my.sharepoint.com/personal/flagger_condreycorprpl_omicrosoft_com/Documents	/Microsoft%20Teams%20Chat%20Files	Meeting Notes 20201020.txt	67	
https://condreycorprpl.sharepoint.com/sites/ProjectsTeam/Shared%20Documents	/	Meeting Notes 20201020.txt	67	
https://condreycorprpl.sharepoint.com/sites/condreycorprpl/Shared%20Documents	/General	Meeting Notes 20201020.txt	67	
https://condreycorprpl-my.sharepoint.com/personal/acoax_sp_ctec_org/Documents	/Microsoft%20Teams%20Chat%20Files	Meeting Notes 20201020.txt	67	
https://condreycorprpl-my.sharepoint.com/personal/amartin_sp_ctec_org/Documents	/Microsoft%20Teams%20Chat%20Files	Meeting Notes 20201020.txt	67	
https://condreycorprpl-my.sharepoint.com/personal/ajames_sp_ctec_org/Documents	/Microsoft%20Teams%20Chat%20Files	Meeting Notes 20201020.txt	67	

Records Loaded : 60

9. Click **Save**.

10. Click *Design Layout*.

The screenshot shows the Report Designer interface. The top toolbar includes buttons for Open..., Save, Edit Query, Download All Data, File System Paths, Refresh Data Bindings, Cut, Copy, Paste, Undo, Redo, Times New Roman font, 9.75 font size, Bold, Italic, Underline, Alignment, Layout, Zoom Out, Zoom In, Windows, and Scripts. The main area displays a report layout with a grid and a 'Detail' section. The 'Group and Sort' pane is visible at the bottom, and the 'Field List' and 'Properties' panes are on the right.

11. Click **Open**.

12. Locate the `.rep` file that you saved and unzipped in [Unzip the downloaded file and open the .sql file in a text editor. \(page 33\)](#) and click *Open*.

The layout template appears in the Report Designer.

13. Click *Download All Data*.
14. In the subsequent dialog box, click *Yes*.

This runs the query in the database and loads data into the report template.

15. Click *Print Preview* to review the report findings.
16. Save the report by doing one of the following:
  - From the *Export To* drop-down menu, select the file type you want to save the report as.
  - Click *Save Report* to save the report as a `.prnx` file that you can open in the Report Viewer and if you want later, export the report to the desired file type.

## 4.3 - Active Directory Identity Enrichment

You can provide extended data for identities in Custom Query reports or create identity reports for security principals in Active Directory.

### 4.3.1 - Determining Prerequisites

- File Reporter collects Active Directory identity data once per day by default.  
For instructions on running a collection manually, see *Active Directory Identity Scans in the File Reporter 24.2 Administration Guide*.
- Decide whether you wish to extend an existing Custom Query file system metadata or permissions report or if you wish to report just on Active Directory identities themselves.
  - If extending an existing Custom Query report determine whether that report data already includes the owner or permissions trustee Security Identifiers (SIDs) or GUIDs.
  - If reporting solely on Active Directory identities, determine which of the extended attributes to include in the report.

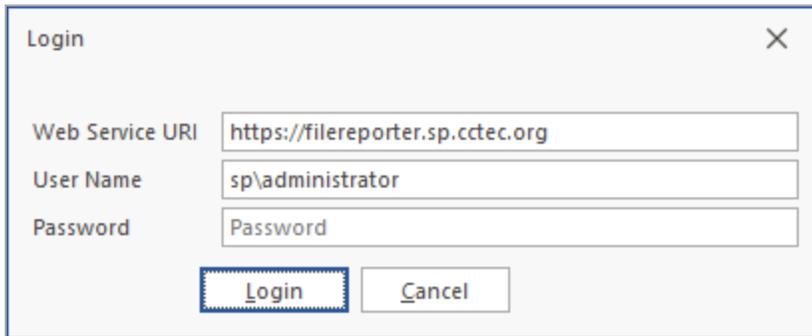
See the table and view definitions for [ad.domains \(page 43\)](#), [ad.ds\\_objects \(page 44\)](#), and [ad.ds\\_objects\\_view \(page 108\)](#) for details on available attributes.

### 4.3.2 - Designing the Report

This example extends a "Direct User Assignment" Custom Query report which identifies user accounts that have been assigned permissions directly to folders (as opposed to using group membership) and shows a summary of the count of direct permissions per user by share path.

## 4 - Example Scenarios

1. From the *Start* menu, launch the File Reporter 24.2 Report Designer.

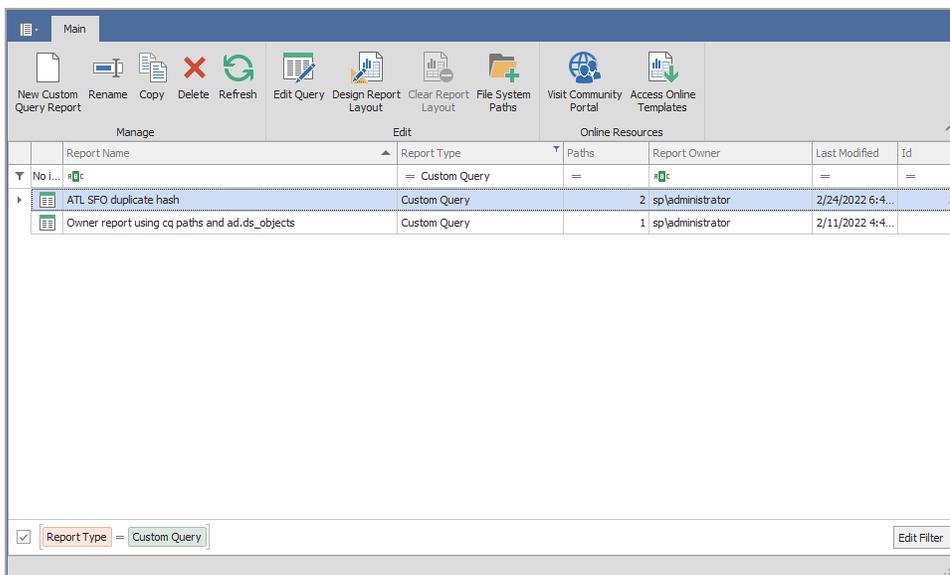


Web Service URI:

User Name:

Password:

2. Enter the login credentials and click *Login*.



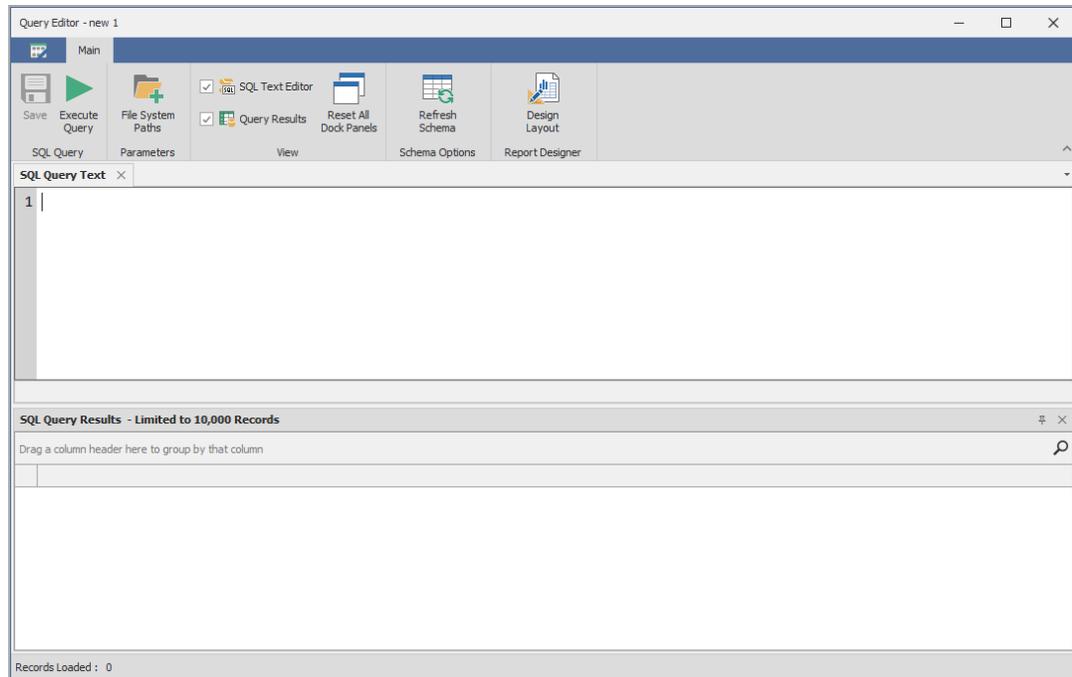
	Report Name	Report Type	Paths	Report Owner	Last Modified	Id
▼	No i...	= Custom Query	=	#	=	=
▶	ATL SFO duplicate hash	Custom Query	2	sp\administrator	2/24/2022 6:4...	3
	Owner report using cq paths and ad.ds_objects	Custom Query		1 sp\administrator	2/11/2022 4:4...	1

Report Type = Custom Query

All of your saved Custom Query reports are listed.

3. Click *New Custom Query*, give it a name, then click *Create*.

The Report Designer Query Editor is launched.



4. Enter the following SQL statements into the Query Editor:

### Basic Query - User Direct Permissions Summary

```

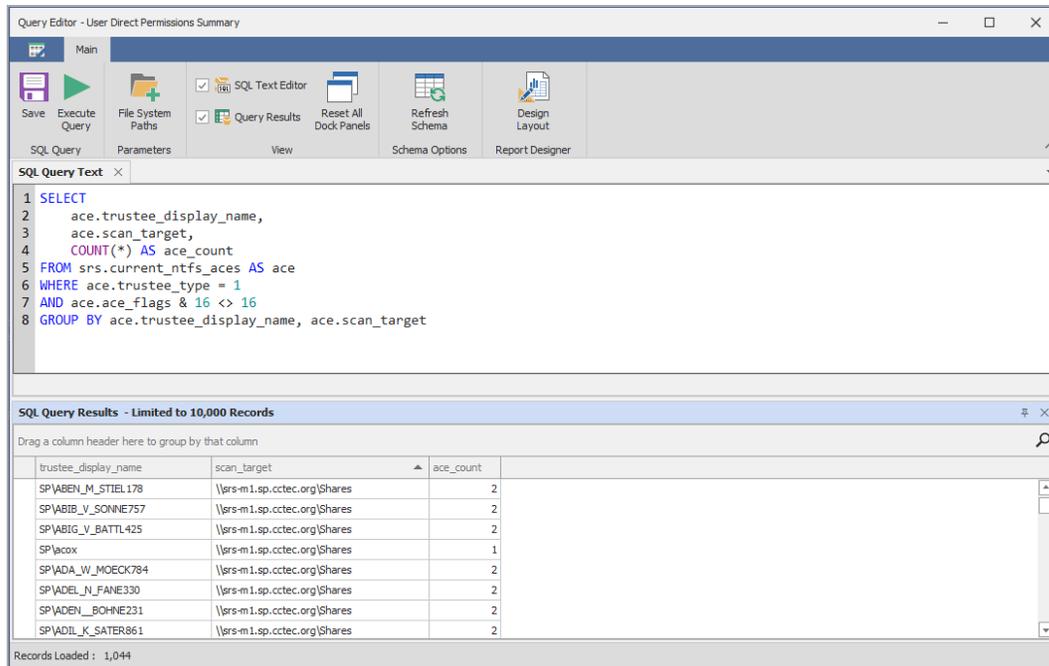
1 | SELECT
2 |     ace.trustee_display_name,
3 |     ace.scan_target,
4 |     COUNT(*) AS ace_count
5 | FROM srs.current_ntfs_aces AS ace
6 | WHERE ace.trustee_type = 1
7 |     AND ace.ace_flags & 16 <> 16
8 | GROUP BY
9 |     ace.trustee_display_name,
10 |    ace.scan_target

```

5. Click *Execute* to see a preview of the report data.

This query will produce a basic result similar to the following:

## 4 - Example Scenarios



The screenshot shows a SQL Query Editor window titled "Query Editor - User Direct Permissions Summary". The main toolbar includes buttons for Save, Execute Query, File System Paths, SQL Text Editor, Query Results, Reset All Dock Panels, Refresh Schema, and Design Layout. The SQL Query Text area contains the following query:

```
1 SELECT
2     ace.trustee_display_name,
3     ace.scan_target,
4     COUNT(*) AS ace_count
5 FROM srs.current_ntfs_aces AS ace
6 WHERE ace.trustee_type = 1
7 AND ace.ace_flags & 16 <> 16
8 GROUP BY ace.trustee_display_name, ace.scan_target
```

The SQL Query Results section shows a table with 10,000 records limited. The table has columns: trustee\_display\_name, scan\_target, and ace\_count. The data is as follows:

trustee_display_name	scan_target	ace_count
SPIABEN_M_STIEL178	\\srs-m1.sp.cctec.org\Shares	2
SPIABIB_V_SONNE757	\\srs-m1.sp.cctec.org\Shares	2
SPIABIG_V_BATTL425	\\srs-m1.sp.cctec.org\Shares	2
SP\acox	\\srs-m1.sp.cctec.org\Shares	1
SPIADA_W_MOECK784	\\srs-m1.sp.cctec.org\Shares	2
SPIADEL_N_FANE330	\\srs-m1.sp.cctec.org\Shares	2
SPIADEN__BOHNE231	\\srs-m1.sp.cctec.org\Shares	2
SPIADIL_K_SATER861	\\srs-m1.sp.cctec.org\Shares	2

Records Loaded: 1,044

6. Click Save to save the SQL entered so far.
7. Augment the data by joining with the *ad.ds\_objects* table to include the Active Directory user *display\_name* and *title* fields.

### Enhanced Query - User Direct Permissions Summary

```
1 SELECT
2     dso.display_name,
3     dso.title,
4     ace.trustee_display_name,
5     ace.scan_target,
6     COUNT(*) AS ace_count
7 FROM srs.current_ntfs_aces AS ace
8 JOIN ad.ds_objects AS dso
9     ON dso.object_sid = ace.sid
10 WHERE ace.trustee_type = 1
11     AND ace.ace_flags & 16 <> 16
12 GROUP BY
13     ace.trustee_display_name,
14     ace.scan_target,
15     dso.display_name,
16     dso.title
```

8. Click Execute and see the updated results that include the *title* and *display\_name* fields.

Query Editor - User Direct Permissions Summary

Main

Save Execute Query File System Paths SQL Text Editor Query Results Reset All Dock Panels Refresh Schema Design Layout

SQL Query Parameters View Schema Options Report Designer

SQL Query Text X

```

1 SELECT dso.display_name,
2        dso.title,
3        ace.trustee_display_name,
4        ace.scan_target,
5        COUNT(*) AS ace_count
6 FROM srs.current_ntfs_aces AS ace
7 JOIN ad.ds_objects AS dso ON dso.object_sid = ace.sid
8 WHERE ace.trustee_type = 1
9 AND ace.ace_flags & 16 <> 16
10 GROUP BY ace.trustee_display_name, ace.scan_target, dso.display_name, dso.title

```

SQL Query Results - Limited to 10,000 Records

Drag a column header here to group by that column

display_name	title	trustee_display_name	scan_target	ace_count
Abeni_Stely	Employee	SP\ABEN_M_STIEL178	\\srs-m1.sp.cctec.org\Shares	2
Abiba_Sonnek	Employee	SP\ABIB_V_SONNE757	\\srs-m1.sp.cctec.org\Shares	2
Abigale_Battle	Employee	SP\ABIG_V_BATTL425	\\srs-m1.sp.cctec.org\efs-share	1
Abigale_Battle	Employee	SP\ABIG_V_BATTL425	\\srs-m1.sp.cctec.org\Shares	2
Amanda Cox	HQ Employee	SP\acox	\\srs-m1.sp.cctec.org\Shares	1
Amanda Cox	HQ Employee	SP\acox	\\srs-m1.sp.cctec.org\Shares2	1
Ada_Moock	Employee	SP\ADA_W_MOECK784	\\srs-m1.sp.cctec.org\Shares	2
Adela_Fane	Employee	SP\ADEL_N_FANE330	\\srs-m1.sp.cctec.org\Shares	2

Records Loaded : 1,044



## 5 - Schema Reference

### 5.1 - Tables

ad.domains

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint		Primary key
<b>db_last_update</b>	datetime2(3)	timestamp without time zone	Last update time for this entry in the database
<b>domain_netbios</b>	nvarchar(15)	varchar(15)	Domain NetBIOS name
<b>domain_dns</b>	nvarchar(256)	varchar(256)	Domain DNS name
<b>domain_sid</b>	varbinary(68)	bytea	Domain security identifier
<b>forest_dns</b>	nvarchar(2560)	varchar(256)	Forest DNS name

## ad.ds\_objects

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>db_domain_sid</b>	varbinary(68)	bytea	SID of the domain itself
<b>db_last_update</b>	datetime2(3)	timestamp	Last update time for this entry in the database
<b>object_guid</b>	binary(16)	bytea	Object's GUID
<b>object_category</b>	nvarchar(256)	varchar(256)	Using LDAP display name, not FDN.
<b>object_class</b>	nvarchar(256)	varchar(256)	Only includes structural class value from this multi-value attribute.
<b>object_sid</b>	varbinary(68)	bytea	Object's Security Identifier
<b>dn</b>	nvarchar(max)	text	Distinguished name
<b>upn</b>	nvarchar(1024)	varchar(1024)	User principal name
<b>sam_account_name</b>	nvarchar(256)	varchar(256)	SAM account name
<b>sam_account_type</b>	integer	integer	<p>See <a href="https://docs.microsoft.com/en-us/windows/win32/adschema/a-samaccounttype">https://docs.microsoft.com/en-us/windows/win32/adschema/a-samaccounttype</a> for details.</p> <p>Enum values:</p> <ul style="list-style-type: none"> <li>0x00000000 - Domain</li> <li>0x10000000 - Group</li> <li>0x10000001 - Non-security Group object</li> <li>0x20000000 - Alias object</li> <li>0x20000001 - Non-security Alias object</li> <li>0x30000000 - Normal User account</li> <li>0x30000001 - Machine (computer) account</li> </ul>

Column Name	SQL Server	PostgreSQL	Notes
			<p>0x30000002 - Trust account</p> <p>0x40000000 - APP_BASIC Group</p> <p>0x40000001 - APP_QUERY Group</p>
<b>sam_principal_name</b>	nvarchar(256)	varchar(256)	<p>NetBIOS\SamAccountName. From msDS-PrincipalName.</p> <p>Note that the NetBIOS name here may be different from the associated domain NetBIOS name where this account was scanned.</p> <p>This is especially true for domain Builtin\* accounts and foreign security principals.</p>
<b>display_name</b>	nvarchar(256)	varchar(256)	
<b>uac_flags</b>	integer	integer	<p>Combines both userAccessControl and msDs-User-Account-Control-Computed attribute values into a single flag.</p> <p>See the following for details:</p> <ul style="list-style-type: none"> <li>• <a href="https://docs.microsoft.com/en-us/windows/win32/adschema/a-useraccountcontrol">https://docs.microsoft.com/en-us/windows/win32/adschema/a-useraccountcontrol</a></li> <li>• <a href="https://docs.microsoft.com/en-us/windows/win32/adschema/a-msds-user-account-control-computed">https://docs.microsoft.com/en-us/windows/win32/adschema/a-msds-user-account-control-computed</a></li> </ul> <p>Flags values:</p> <p>0x00000001 - Logon script is executed</p> <p>0x00000002 - User Account disabled</p> <p>0x00000008 - Home directory</p>

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			required
			0x00000010 - Account currently locked out
			0x00000020 - No password required
			0x00000040 - User cannot change password
			0x00000080 - User can send encrypted password
			0x00000100 - Temporary duplicate account
			0x00000200 - Normal account - typical user
			0x00000800 - Inter-domain trust account
			0x00001000 - Computer (Workstation / Member Server) account
			0x00002000 - Domain controller computer account
			0x00010000 - Password does not expire
			0x00020000 - Majority Node Set (MNS) logon account
			0x00040000 - Smart card required for logon
			0x00080000 - Service account trusted for Kerberos delegation
			0x00100000 - Account not allowed trust for delegation
			0x00200000 - Account can only use DES keys
			0x00400000 - Account does not require Kerberos pre-authentication for logon

Column Name	SQL Server	PostgreSQL	Notes
			<p>0x00800000 - User password has expired</p> <p>0x01000000 - Account enabled for delegation</p> <p>0x04000000 - Partial secrets account</p> <p>0x08000000 - Account can only use Use AES keys</p>
<b>account_expires</b>	datetime2(0)	timestamp	
<b>create_timestamp</b>	datetime2(0)	timestamp	
<b>description</b>	nvarchar(1024)	varchar(1024)	Only uses first value of this multi-value attribute
<b>mail</b>	nvarchar(256)	varchar(256)	
<b>given_name</b>	nvarchar(64)	varchar(64)	
<b>surname</b>	nvarchar(64)	varchar(64)	
<b>last_logon_timestamp</b>	datetime2(0)	timestamp	<p>NOTE: This attribute only has 14-day granularity.</p> <p>See: <a href="https://docs.microsoft.com/en-us/windows/win32/adschema/a-lastlogontimestamp">https://docs.microsoft.com/en-us/windows/win32/adschema/a-lastlogontimestamp</a></p>
<b>department</b>	nvarchar(64)	varchar(64)	
<b>title</b>	nvarchar(128)	varchar(128)	
<b>primary_group_sid</b>	varbinary(68)	bytea	SID of referenced object
<b>managed_by_guid</b>	binary(16)	bytea	GUID of referenced DS object
<b>manager_guid</b>	binary(16)	bytea	GUID of referenced DS object
<b>group_type</b>	integer	integer	See <a href="https://docs.microsoft.com/en-us/">https://docs.microsoft.com/en-us/</a>

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			<p><a href="https://msdn.microsoft.com/en-us/windows/win32/adschema/a-grouptype">us/windows/win32/adschema/a-grouptype</a> for details.</p> <p>Flags:</p> <ul style="list-style-type: none"> <li>0x01 - System created group</li> <li>0x02 - Global group</li> <li>0x04 - Domain Local group</li> <li>0x08 - Universal group</li> <li>0x10 - APP_BASIC group for Windows Server Authorization Manager</li> <li>0x20 - APP_QUERY group for Windows Server Authorization Manager</li> <li>0x80000000 - Security Group. If not set, then a Distribution Group</li> </ul>
<b>dns_host_name</b>	nvarchar(2048)	varchar(2048)	Applies to Computer objects

## srs.analysis.file\_scan\_entries

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_time</b>	datetime2(3)	timestamp	Time when file content was scanned
<b>fullpath</b>	nvarchar(max)	text	Full UNC path to the file
<b>fullpath_hash</b>	binary(20)	bytea	SHA-1 hash of lowercase fullpath
<b>content_hash</b>	binary(32)	bytea	SHA-2 hash of file content
<b>size</b>	bigint	bigint	File size
<b>modify_time</b>	datetime2(2)	timestamp	Last write time of file
<b>classification</b>	nvarchar(64)	varchar(64)	Classification name
<b>category</b>	nvarchar(64)	varchar(64)	Category name
<b>search_pattern_name</b>	nvarchar(64)	varchar(64)	Search pattern name
<b>search_pattern_string</b>	nvarchar(1024)	varchar(1024)	Search pattern string
<b>match_count</b>	int	int	Number of matches for Search Pattern on this path
<b>match_confidence</b>	int	int	1 = Low 2 = Medium 3 = High
<b>job_id</b>	int	int	File content scan job ID
<b>job_definition</b>	nvarchar(64)	varchar(64)	Job definition name
<b>status_code</b>	int	int	Processing status code for this file entry

## ms365.drive\_item\_types

Column Name	SQL Server	PostgreSQL	Notes
<b>item_type</b>	int	int	0 = unknown 1 = file 2 = folder 3 = remote_item
<b>item_type_name</b>	nvarchar(32)	varchar(32)	Item type description

## ms365.drive\_items

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_id</b>	bigint	bigint	Reference to primary key in ms365.drive_scans
<b>drive_id</b>	bigint	bigint	Reference to associated drive in ms365.drives
<b>ms365_id</b>	nvarchar(256)	varchar(256)	Unique ID provided by MS GraphAPI
<b>ms365_drive_id</b>	nvarchar(256)	varchar(256)	Unique ID provided by MS GraphAPI for the associated drive
<b>ms365_parent_id</b>	nvarchar(256)	varchar(256)	Unique ID provided by MS GraphAPI for parent path
<b>created_by</b>	nvarchar(256)	varchar(256)	Unique ID provided by MS GraphAPI for the associated identity
<b>created_by_name</b>	nvarchar(256)	varchar(256)	Display name of the "created_by" account
<b>create_time</b>	datetime2(3)	timestamp	Create time for entry
<b>item_type</b>	integer	integer	Note: Only one of these values is set as a "primary" value for this entry as opposed to the item_facets column  0 = unknown 1 = file 2 = folder 4 = package 8 = remote item
<b>item_facets</b>	integer	integer	Note: All applicable flags are set for this value, as opposed to the item_type column  0 = none 1 = file

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			2 = folder 4 = package 8 = remote item
<b>file_hash</b>	varbinary(64)	varchar(64)	Files only - QuickXorHash of entry See <a href="https://docs.microsoft.com/en-us/graph/api/resources/hashe?view=graph-rest-1.0">https://docs.microsoft.com/en-us/graph/api/resources/hashe?view=graph-rest-1.0</a>
<b>child_count</b>	bigint	bigint	Folders only - number of child entries in the folder Only includes immediate children, not recursive.
<b>modified_by</b>	nvarchar(256)	varchar(256)	Unique ID provided by MS GraphAPI for the associated identity
<b>modified_by_name</b>	nvarchar(256)	varchar(256)	Display name of the "modified_by" account
<b>modify_time</b>	datetime2(3)	timestamp	Last modified time
<b>name</b>	nvarchar(256)	varchar(256)	Name of entry
<b>file_extension</b>	nvarchar(32)	varchar(32)	File name extension
<b>size</b>	bigint	bigint	Size in bytes
<b>web_url</b>	nvarchar(max)	text	Full path to item
<b>web_url_hash</b>	varbinary(32)	bytea	SHA-256 hash of web_url

## ms365.drive\_scans

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	integer	integer	Reference to primary key in ms365.jobs
<b>drive_id</b>	bigint	bigint	Reference to primary key in ms365.drives
<b>scan_status</b>	integer	integer	0 = Queued 1 = In progress 2 = Completed 3 = Failed 99 = Canceled
<b>scan_state</b>	integer	integer	0 = Pending 1 = Current 99 = Marked for cleanup
<b>delegated_time</b>	datetime2(3)	timestamp	Time at which scan was requested
<b>start_time</b>	datetime2(3)	timestamp	Time when scan started
<b>stop_time</b>	datetime2(3)	timestamp	Time when scan stopped
<b>scan_progress_data</b>	nvarchar(max)	text	JSON data with scan progress details
<b>agent_name</b>	nvarchar(256)	varchar(256)	Name of Agent365 server performing the scan

## ms365.drive\_scans\_history

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>scan_id</b>	bigint	bigint	Reference to primary key in ms365.drive_scans
<b>start_time</b>	datetime2(3)	timestamp	Drive scan start time
<b>stop_time</b>	datetime2(3)	timestamp	Drive scan stop time
<b>drive_id</b>	bigint	bigint	Reference to primary key in ms365.drives
<b>drive_name</b>	nvarchar(256)	varchar(256)	Drive name
<b>web_url</b>	nvarchar(max)	text	Full path to drive
<b>ms365_drive_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
<b>scan_progress_status</b>	nvarchar(max)	text	JSON data with scan progress details
<b>agent_name</b>	nvarchar(256)	varchar(256)	Name of Agent365 server that performed the scan
<b>scan_status</b>	int	int	0 = Queued 1 = In progress 2 = Completed 3 = Failed 99 = Canceled
<b>scan_state</b>	int	int	0 = Pending 1 = Current 99 = Marked for cleanup
<b>result_string</b>	nvarchar(max)	text	Success or error message

## ms365.drives

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants table
<b>site_id</b>	bigint	bigint	Reference to primary key in ms365.sites table
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
<b>name</b>	nvarchar(256)	varchar(256)	Drive name
<b>ms365_owner_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
<b>quota</b>	nvarchar(256)	varchar(256)	JSON data including quota details
<b>web_url</b>	nvarchar(max)	text	Full web path to drive
<b>drive_type</b>	nvarchar(64)	varchar(64)	<p>Known values in MS GraphAPI include</p> <ul style="list-style-type: none"> <li>• business</li> <li>• documentLibrary</li> </ul> <p>See: <a href="https://docs.microsoft.com/en-us/graph/api/resources/drive?view=graph-rest-1.0">https://docs.microsoft.com/en-us/graph/api/resources/drive?view=graph-rest-1.0</a></p>

## ms365.group\_drives

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>ms365_group_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated group
<b>ms365_drive_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated drive

## ms365.group\_member\_types

Column Name	SQL Server	PostgreSQL	Notes
<b>member_type</b>	int	int	0 = direct 1 = transitive
<b>member_type_name</b>	nvarchar(32)	varchar(32)	Member type description

## ms365.group\_members

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>ms365_group_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated group
<b>ms365_member_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated member
<b>member_type</b>	int	int	0 = direct 1 = transitive

## ms365.group\_owners

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>ms365_group_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated group
<b>ms365_owner_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated owner

## ms365.group\_sites

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>ms365_group_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated group
<b>ms365_site_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated SharePoint site

## ms365.groups

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
<b>display_name</b>	nvarchar(256)	varchar(256)	Friendly name of group
<b>email</b>	nvarchar(256)	varchar(256)	Email address
<b>group_types</b>	nvarchar(64)	varchar(64)	<p>One or more of the following from MS GraphAPI:</p> <ul style="list-style-type: none"> <li>• Unified</li> <li>• DynamicMembership</li> <li>• [empty string]</li> </ul> <p>See: <a href="https://docs.microsoft.com/en-us/graph/api/resources/group?view=graph-rest-1.0">https://docs.microsoft.com/en-us/graph/api/resources/group?view=graph-rest-1.0</a></p>
<b>onprem_sid</b>	varbinary(68)	bytea	On-premises Security Identifier (SID)
<b>onprem_dnsdomain</b>	nvarchar(256)	varchar(256)	On-premises DNS domain
<b>onprem_netbios</b>	nvarchar(256)	varchar(256)	On-premises NetBIOS domain
<b>onprem_samaccount</b>	nvarchar(256)	varchar(256)	On-premises SAM Account Name

## ms365.identity\_types

Column Name	SQL Server	PostgreSQL	Notes
<b>identity_type</b>	int	int	0 = unknown 1 = user 2 = group 3 = device 4 = application
<b>identity_type_name</b>	nvarchar(32)	varchar(32)	Identity type description

## ms365.jobs

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	int	int	Primary key
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants
<b>start_time</b>	datetime2(3)	timestamp	Time job started
<b>stop_time</b>	datetime2(3)	timestamp	Time job stopped
<b>job_status</b>	int	int	0 = Queued 1 = In progress 2 = Completed 3 = Failed 99 = Canceled
<b>job_progress_data</b>	nvarchar(max)	text	JSON data with job progress details
<b>agent_name</b>	nvarchar(256)	varchar(256)	Agent365 server performing the scan

## ms365.jobs\_history

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	int	int	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants
<b>tenant_name</b>	nvarchar(256)	varchar(256)	Associated *.onmicrosoft.com tenant name
<b>start_time</b>	datetime2(3)	timestamp	Time when job started
<b>stop_time</b>	datetime2(3)	timestamp	Time when job stopped
<b>job_status</b>	int	int	0 = Queued 1 = In progress 2 = Completed 3 = Failed 99 = Canceled
<b>result_string</b>	nvarchar(1024)	varchar(1024)	Success or failure message
<b>job_progress_data</b>	nvarchar(max)	text	JSON data with job progress details
<b>agent_name</b>	nvarchar(256)	varchar(256)	Agent365 server performing the scan

## ms365.permissions

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_id</b>	bigint	bigint	Reference to primary key in ms365.drive_scans
<b>site_collection_id</b>	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
<b>drive_item_id</b>	bigint	bigint	Reference to primary key in ms365.drive_items
<b>ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
<b>expire_time</b>	datetime2(3)	timestamp	Timestamp when link expires
<b>is_inherited</b>	bit	boolean	true = inherited false = not inherited
<b>has_password</b>	bit	boolean	This currently applies only to Anonymous sharing links
<b>grantedto_ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated trustee
<b>grantedto_type</b>	integer	integer	0 = unknown 1 = user 2 = group 3 = device 4 = application
<b>grantedto_sp_user_id</b>	integer	integer	Reference to an associated SharePoint site collection's user account
<b>grantedto_sp_group_id</b>	integer	integer	Reference to an associated SharePoint site collection's group account
<b>grantedto_sp_login_name</b>	nvarchar(256)	varchar(256)	SharePoint-specific login name for the trustee
<b>grantedto_</b>	nvarchar(256)	varchar(256)	Friendly name of trustee

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
<b>display_name</b>			
<b>grantedto_email</b>	nvarchar(256)	varchar(256)	Email address of trustee
<b>invite_email</b>	nvarchar(256)	varchar(256)	Email address of recipient (trustee)
<b>invite_sentby_ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated sender
<b>invite_sentby_display_name</b>	nvarchar(256)	varchar(256)	Friendly name of sender
<b>invite_signin_required</b>	bit	boolean	true = sign-in required false = sign-in not required
<b>link_app_display_name</b>	nvarchar(256)	varchar(256)	Friendly name of application
<b>link_app_ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated application
<b>link_type</b>	nvarchar(32)	varchar(32)	One of: <ul style="list-style-type: none"> <li>view</li> <li>edit</li> </ul> See: <a href="https://docs.microsoft.com/en-us/graph/api/resources/sharinglink?view=graph-rest-1.0">https://docs.microsoft.com/en-us/graph/api/resources/sharinglink?view=graph-rest-1.0</a>
<b>link_scope</b>	nvarchar(32)	varchar(32)	One of the following from MS GraphAPI: <ul style="list-style-type: none"> <li>anonymous</li> <li>organization</li> </ul> See : <a href="https://docs.microsoft.com/en-us/graph/api/resources/sharinglink?view=graph-rest-1.0">https://docs.microsoft.com/en-us/graph/api/resources/sharinglink?view=graph-rest-1.0</a>
<b>link_prevents_download</b>	bit	boolean	true = view only (download not allowed)
<b>roles</b>	nvarchar(128)	varchar(128)	One of the following from MS GraphAPI:

Column Name	SQL Server	PostgreSQL	Notes
			<ul style="list-style-type: none"><li>• read</li><li>• write</li><li>• owner</li></ul> <p>See: <a href="https://docs.microsoft.com/en-us/graph/api/resources/permission?view=graph-rest-1.0">https://docs.microsoft.com/en-us/graph/api/resources/permission?view=graph-rest-1.0</a></p>

## ms365.sharing\_link\_members

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>permission_id</b>	bigint	bigint	Reference to primary key in ms365.permissions
<b>scan_id</b>	bigint	bigint	Reference to primary key in ms365.drive_scans
<b>site_collection_id</b>	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
<b>ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated member
<b>member_type</b>	integer	integer	0 - Direct membership 1 - Transitive (nested membership)
<b>display_name</b>	nvarchar(256)	varchar(256)	Friendly name of member
<b>email</b>	nvarchar(256)	varchar(256)	Email address of member
<b>sp_group_id</b>	integer	integer	Reference to an associated SharePoint site collection's group account
<b>sp_user_id</b>	integer	integer	Reference to an associated SharePoint site collection's user account
<b>sp_login_name</b>	nvarchar(256)	varchar(256)	SharePoint-specific

Column Name	SQL Server	PostgreSQL	Notes
			login name for the member
<b>sp_display_name</b>	nvarchar(256)	varchar(256)	Friendly name of member's associated SharePoint account

## ms365.sites

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants
<b>site_collection_id</b>	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
<b>ms365_parent_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated parent site
<b>display_name</b>	nvarchar(256)	varchar(256)	Friendly name of SharePoint site
<b>name</b>	nvarchar(256)	varchar(256)	Site name
<b>is_root</b>	bit	boolean	true = root site (no parent sites) false = child site
<b>web_url</b>	nvarchar(max)	text	Full path to SharePoint site

## ms365.sp\_base\_permissions

Column Name	SQL Server	PostgreSQL	Notes
<b>flag</b>	bigint	bigint	Base permissions flag value
<b>name</b>	nvarchar(64)	varchar(64)	Flag entry name
<b>description</b>	nvarchar(1024)	varchar(1024)	Flag entry description

This is a pre-populated lookup table.

Values are derived from SharePoint client and server .NET APIs.

See [https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-server/ee536458\(v=office.15\)](https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-server/ee536458(v=office.15)) and [https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-server/ms412690\(v=office.15\)](https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-server/ms412690(v=office.15)).

## ms365.sp\_group\_members

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	integer	integer	Reference to primary key in ms365.jobs
<b>tenant_id</b>	integer	integer	Reference to primary key in ms365.tenants
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>site_collection_id</b>	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
<b>sp_group_id</b>	integer	integer	Reference to an associated SharePoint site collection's group account
<b>sp_member_id</b>	integer	integer	Reference to an associated SharePoint site collection's user account
<b>ms365_member_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated member

## ms365.sp\_groups

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	integer	integer	Reference to primary key in ms365.jobs
<b>tenant_id</b>	integer	integer	Reference to primary key in ms365.tenants
<b>site_collection_id</b>	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
<b>sp_id</b>	integer	integer	SharePoint ID for this entry, unique per site collection
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>login_name</b>	nvarchar(256)	varchar(256)	SharePoint account name for this group
<b>title</b>	nvarchar(256)	varchar(256)	Group's title
<b>description</b>	nvarchar(1024)	varchar(1024)	Group's description
<b>is_hidden</b>	bit	boolean	Flag indicating whether this is a hidden group

## ms365.sp\_permission\_levels

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	integer	integer	Reference to primary key in ms365.jobs
<b>tenant_id</b>	integer	integer	Reference to primary key in ms365.tenants
<b>site_collection_id</b>	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
<b>sp_id</b>	integer	integer	SharePoint ID for this entry, unique per site collection
<b>name</b>	nvarchar (256)	varchar (256)	Name of Permission Level (role)
<b>description</b>	nvarchar (1024)	varchar (1024)	Description for this Permission Level
<b>base_permissions</b>	bigint	bigint	<p>Flags value indicating the underlying permissions this Permission Level (Role) defines</p> <p>Query or join with the descriptions table ms365.sp_base_permissions.</p> <p>See</p> <ul style="list-style-type: none"> <li>• <a href="https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-server/ee536458(v=office.15)">https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-server/ee536458(v=office.15)</a></li> <li>• <a href="https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-server/ms412690(v=office.15)">https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-server/ms412690(v=office.15)</a></li> </ul>
<b>role_type</b>	integer	integer	<p>0 - None</p> <p>1 - Guest</p> <p>2 - Reader</p> <p>3 - Contributor</p> <p>4 - Web Designer</p> <p>5 - Administrator</p> <p>6 - Editor</p>

Column Name	SQL Server	PostgreSQL	Notes
			7 - Reviewer 8 - Restricted Reader 9 - Restricted Guest 255 - System  See <a href="https://docs.microsoft.com/en-us/dotnet/api/microsoft.sharepoint.client.roletype?view=sharepoint-csom">https://docs.microsoft.com/en-us/dotnet/api/microsoft.sharepoint.client.roletype?view=sharepoint-csom</a>
<b>is_hidden</b>	bit	boolean	Indicates whether this is a hidden role

## ms365.sp\_permissions

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_id</b>	bigint	bigint	Reference to primary key in ms365.drive_scans
<b>site_collection_id</b>	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
<b>drive_item_id</b>	bigint	bigint	Reference to primary key in ms365.drive_items
<b>sp_user_id</b>	integer	integer	Reference to an associated SharePoint site collection's user account
<b>sp_group_id</b>	integer	integer	Reference to an associated SharePoint site collection's group account
<b>sp_login_name</b>	nvarchar(256)	varchar(256)	SharePoint account name for the trustee
<b>sp_display_name</b>	nvarchar(256)	varchar(256)	Display name for the trustee
<b>sp_permission_level_id</b>	integer	integer	Reference to primary key in ms365.sp_permission_levels
<b>is_inherited</b>	bit	boolean	Flag indicating whether this is an inherited permission

## ms365.sp\_site\_permissions

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	integer	integer	Reference to primary key in ms365.jobs
<b>site_id</b>	bigint	bigint	Reference to primary key in ms365.sites for the associated site
<b>site_collection_id</b>	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
<b>drive_item_id</b>	bigint	bigint	Reference to primary key in ms365.drive_items
<b>sp_user_id</b>	integer	integer	Reference to an associated SharePoint site collection's user account
<b>sp_group_id</b>	integer	integer	Reference to an associated SharePoint site collection's group account
<b>sp_login_name</b>	nvarchar(256)	varchar(256)	SharePoint account name for the trustee
<b>sp_display_name</b>	nvarchar(256)	varchar(256)	Display name for the trustee
<b>sp_permission_level_id</b>	integer	integer	Reference to primary key in ms365.sp_permission_levels
<b>is_inherited</b>	bit	boolean	Flag indicating whether this is an inherited permission

## ms365.sp\_users

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants table
<b>site_collection_id</b>	bigint	bigint	Reference to primary key in ms365.sites for the site collection root site
<b>ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
<b>sp_id</b>	integer	integer	SharePoint ID for this entry, unique per site collection
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>login_name</b>	nvarchar(256)	varchar(256)	SharePoint account name for this user
<b>upn</b>	nvarchar(256)	varchar(256)	User principal name
<b>email</b>	nvarchar(256)	varchar(256)	User's email address
<b>title</b>	nvarchar(256)	varchar(256)	User's title
<b>principal_type</b>	smallint	smallint	One of the following values as defined by the CSOM 'PrincipalType' enumeration: <ul style="list-style-type: none"> <li>• 0 : None</li> <li>• 1 : User</li> <li>• 2 : Distribution List</li> <li>• 4 : Security Group</li> <li>• 8 : SharePoint Group</li> </ul>
<b>is_site_admin</b>	bit	boolean	Flag indicating whether this

Column Name	SQL Server	PostgreSQL	Notes
			user is assigned as a SharePoint admin for the associated site.
<b>is_hidden</b>	bit	boolean	Flag indicating a hidden account
<b>is_guest</b>	bit	boolean	Flag indicating a guest account
<b>is_email_authenticated</b>	bit	boolean	Only applies to "external" users with sharing

## ms365.team\_channels

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
<b>team_id</b>	bigint	bigint	Reference to primary key in ms365.teams
<b>display_name</b>	nvarchar(256)	varchar(256)	Friendly name of channel
<b>web_url</b>	nvarchar(256)	varchar(256)	Full path to channel
<b>ms365_files_folder_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated path
<b>ms365_files_folder_drive_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated path's drive

## ms365.teams

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
<b>display_name</b>	nvarchar(256)	varchar(256)	Friendly name of team
<b>visibility</b>	int	int	0 = private 1 = public
<b>web_url</b>	nvarchar(max)	text	Full path to team

## ms365.tenants

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	int	int	Primary key
<b>tenant_name</b>	nvarchar(256)	varchar(256)	Official registered tenant name ending with '.onmicrosoft.com'
<b>ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
<b>display_name</b>	nvarchar(256)	varchar(256)	Tenant display name
<b>default_name</b>	nvarchar(256)	varchar(256)	Optionally registered DNS name set as the "default" e.g. corp.example.com

## ms365.user\_drives

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>ms365_user_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated user
<b>ms365_drive_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI for the associated drive

## ms365.users

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>job_id</b>	int	int	Reference to primary key in ms365.jobs
<b>tenant_id</b>	int	int	Reference to primary key in ms365.tenants
<b>last_update</b>	datetime2(3)	timestamp	Last update time for database entry
<b>ms365_id</b>	nvarchar(256)	varchar(256)	Unique id provided by MS GraphAPI
<b>display_name</b>	nvarchar(256)	varchar(256)	Display name – typically First Last name
<b>upn</b>	nvarchar(1024)	varchar(1024)	User Principal Name
<b>given_name</b>	nvarchar(64)	varchar(64)	First name
<b>surname</b>	nvarchar(64)	varchar(64)	Last name
<b>onprem_sid</b>	varbinary(68)	bytea	On-premises Security Identifier (SID)
<b>onprem_dn</b>	nvarchar(max)	text	On-premises distinguished name
<b>onprem_upn</b>	nvarchar(1024)	varchar(1024)	On-premises User Principal Name
<b>onprem_dnsdomain</b>	nvarchar(256)	varchar(256)	On-premises DNS domain name
<b>onprem_samaccount</b>	nvarchar(256)	varchar(256)	On-premises SAM Account Name
<b>onprem_immutable_id</b>	nvarchar(256)	varchar(256)	Unique id mapping synced on-prem user to associated MS365 user
<b>account_enabled</b>	bit	boolean	Account is enabled
<b>user_type</b>	nvarchar(64)	varchar(64)	Known values from MS GraphAPI include: <ul style="list-style-type: none"> <li>Member</li> </ul>

Column Name	SQL Server	PostgreSQL	Notes
			<ul style="list-style-type: none"><li>• Guest</li></ul> See: <a href="https://docs.microsoft.com/en-us/graph/api/resources/user?view=graph-rest-1.0">https://docs.microsoft.com/en-us/graph/api/resources/user?view=graph-rest-1.0</a>
<b>creation_type</b>	nvarchar(64)	varchar(64)	Known values from MS GraphAPI include: <ul style="list-style-type: none"><li>• [null]</li><li>• Invitation</li><li>• LocalAccount</li><li>• EmailVerified</li></ul> See : <a href="https://docs.microsoft.com/en-us/graph/api/resources/user?view=graph-rest-1.0">https://docs.microsoft.com/en-us/graph/api/resources/user?view=graph-rest-1.0</a>

## srs.ad\_memberships

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>group_id</b>	integer	integer	Reference to primary key in srs.ad_objects
<b>member_id</b>	integer	integer	Reference to primary key in srs.ad_objects

## srs.ad\_objects

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	integer	integer	Primary key
<b>name</b>	nvarchar(256)	varchar(256)	SAM Account Name
<b>fdn</b>	nvarchar(512)	varchar(512)	Full distinguished object name
<b>domain</b>	nvarchar(256)	varchar(256)	Domain name
<b>guid</b>	binary(16)	bytea	Globally Unique Identifier
<b>sid</b>	varbinary(68)	bytea	Security Identifier
<b>object_type</b>	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer
<b>identity_system_id</b>	integer	integer	Reference to primary key of identity_systems table

## srs.identity\_systems

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	integer	integer	Primary key
<b>type</b>	integer	integer	0 = Unknown 1 = Active Directory 3 = Windows Local
<b>name</b>	nvarchar(256)	varchar(256)	One of: AD Forest Root DNS name Member server NetBIOS name Built-in Account Prefix
<b>domain</b>	nvarchar(256)	varchar(256)	AD Forest Root NetBIOS name
<b>proxy_account</b>	nvarchar(256)	varchar(256)	
<b>is_primary</b>	bit	boolean	0 = Not the primary identity system 1 = Primary identity system for authentication
<b>is_managed</b>	bit	boolean	0 = Not managed (member server, built-in domain, etc.) 1 = Managed, configured system
<b>last_modified</b>	datetime2(0)	timestamp	

## srs.ntfs\_aces

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_data_id</b>	bigint	bigint	Reference to scan_data table
<b>flags</b>	smallint	smallint	0x1 = Object Inherit 0x2 = Container Inherit 0x4 = No Propagate 0x8 = Inherit Only 0x10 = Inherited 0x40 = Successful Access 0x80 = Failed Access
<b>ace_type</b>	smallint	smallint	0 = Access Allowed 1 = Access Denied 2 = System Audit 9 = Allowed Callback 10 = Denied Callback 13 = System Audit Callback 17 = System Mandatory Label
<b>access_mask</b>	integer	integer	0x1 = Read Data / List Directory 0x2 = Write Data / Create File 0x4 = Append Data / Create Subdirectory 0x8 = Read Extended Attributes 0x10 = Write Extended Attributes 0x20 = File Execute / Traverse 0x40 = Delete Child 0x80 = Read Attributes

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			0x100 = Write Attributes 0x10000 = Delete 0x20000 = Read Permissions 0x40000 = Change Permissions 0x80000 = Change Owner 0x100000 = Synchronize 0x1000000 = Access System Security 0x10000000 = Generic All 0x20000000 = Generic Execute 0x40000000 = Generic Write 0x80000000 = Generic Read
<b>sid</b>	varbinary(68)	bytea	Trustee Security Identifier (SID)
<b>index_on_disk</b>	smallint	smallint	Discovered order of this ACE for the associated entry as read from the file system

## srs.scan\_data

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_id</b>	integer	integer	Reference to primary key in srs.scans
<b>path_type</b>	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub 13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
<b>is_link</b>	bit	boolean	Flag indicating entry is a link (symlink, hardlink, etc.)
<b>name</b>	nvarchar(256)	varchar(256)	File or directory name
<b>fullpath</b>	nvarchar(max)	text	Full UNC path to the file system entry
<b>fullpath_hash</b>	binary(20)	bytea	SHA-1 hash of lowercase fullpath

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
<b>filename_extension</b>	nvarchar(32)	varchar(32)	Extensions having more than 32 characters are treated as if they have none
<b>owner_id</b>	varbinary(68)	bytea	Security Identifier (SID)
<b>attributes</b>	integer	integer	0x0 = None 0x1 = Read Only 0x2 = Archive 0x4 = System 0x8 = Hidden 0x10 = Directory 0x20 = Compressed 0x40 = Offline 0x80 = NTFS device 0x100 = NTFS Normal 0x200 = NTFS Temporary 0x400 = NTFS Sparse File 0x800 = NTFS Reparse Point 0x1000 = NTFS Not content indexed 0x2000 = NTFS Encrypted 0x4000 = NTFS Virtual
<b>create_time</b>	datetime2(0)	timestamp	
<b>modify_time</b>	datetime2(0)	timestamp	
<b>access_time</b>	datetime2(0)	timestamp	
<b>size</b>	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
<b>size_on_disk</b>	bigint	bigint	Assumes typical allocation unit

Column Name	SQL Server	PostgreSQL	Notes
			size of 4K
<b>size_compressed</b>	bigint	bigint	Only accurate for NTFS file systems
<b>idx</b>	integer	integer	Scan index; unique per scan
<b>parent_idx</b>	integer	integer	Parent index. Used for hierarchical relation processing
<b>path_depth</b>	integer	integer	Entry depth with respect to the scan target's root path.
<b>ns_left</b>	integer	integer	Nested-set Left index – used for hierarchical relation processing
<b>ns_right</b>	integer	integer	Nested-set Right index – used for hierarchical relation processing
<b>status_code</b>	integer	integer	

## srs.scan\_directory\_data

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_data_id</b>	bigint	bigint	Reference to scan_data table
<b>file_count</b>	integer	integer	Count of all files subordinate to this directory
<b>directory_count</b>	integer	integer	Count of all subdirectories
<b>directory_quota</b>	bigint	bigint	Directory quota for this directory
<b>directory_quota_flags</b>	integer	integer	0 = Unknown 1 = Enforced 2 = Disabled 4 = Incomplete 8 = Rebuilding
<b>child_file_count</b>	integer	integer	Count of all immediately subordinate files
<b>child_link_count</b>	integer	integer	Count of all immediately subordinate links
<b>child_directory_count</b>	integer	integer	Count of all immediately subordinate directories
<b>child_size</b>	bigint	bigint	Size of all immediately subordinate files
<b>child_size_on_disk</b>	bigint	bigint	Size on disk of all immediately subordinate files (assumes 4K allocation size)
<b>child_size_compressed</b>	bigint	bigint	Size on disk of all immediately subordinate compressed files (only accurate with NTFS)
<b>child_link_size</b>	bigint	bigint	Size of all immediately subordinate links

## srs.scan\_history

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	integer	integer	Primary key
<b>identity_system</b>	nvarchar(256)	text	Identity system associated with this scan target
<b>scan_target</b>	nvarchar(1024)	text	UNC path of scan target
<b>file_size</b>	bigint	bigint	Total aggregate size of all files
<b>file_count</b>	integer	integer	Total count of all files
<b>directory_count</b>	integer	integer	Total count of all directories
<b>scan_policy_name</b>	nvarchar(64)	varchar(64)	Scan policy associated with this scan
<b>agent_name</b>	nvarchar(256)	text	
<b>scan_id</b>	integer	integer	Scan ID
<b>scan_type</b>	integer	integer	0 = None 1 = File System Data 2 = Permissions 4 = Volume Free Space
<b>triggered_start_time</b>	datetime2(3)	timestamp	Initial time scan delegation starts
<b>scan_start_time</b>	datetime2(3)	timestamp	Start time when agent begins physical scan
<b>scan_stop_time</b>	datetime2(3)	timestamp	Stop time when agent completes physical scan
<b>enum_start_time</b>	datetime2(3)	timestamp	Agent metrics related to file system object enumeration
<b>enum_stop_time</b>	datetime2(3)	timestamp	Agent metrics related to file system object enumeration
<b>enum_file_count</b>	integer	integer	Agent metrics related to file system object enumeration

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
<b>enum_directory_count</b>	integer	integer	Agent metrics related to file system object enumeration
<b>enum_link_count</b>	integer	integer	Agent metrics related to file system object enumeration
<b>caching_start_time</b>	datetime2(3)	timestamp	Metrics related to agent caching
<b>caching_stop_time</b>	datetime2(3)	timestamp	Metrics related to agent caching
<b>cached_file_count</b>	integer	integer	Metrics related to agent caching
<b>cached_directory_count</b>	integer	integer	Metrics related to agent caching
<b>cached_link_count</b>	integer	integer	Metrics related to agent caching
<b>cache_size</b>	integer	integer	Metrics related to agent caching
<b>cache_size_max</b>	integer	integer	Metrics related to agent caching
<b>metadata_start_time</b>	datetime2(3)	timestamp	Agent metrics related to filesystem metadata collection
<b>metadata_stop_time</b>	datetime2(3)	timestamp	Agent metrics related to filesystem metadata collection
<b>metadata_file_count</b>	integer	integer	Agent metrics related to filesystem metadata collection
<b>metadata_directory_count</b>	integer	integer	Agent metrics related to filesystem metadata collection
<b>metadata_link_count</b>	integer	integer	Agent metrics related to filesystem metadata collection
<b>accounts_start_time</b>	datetime2(3)	timestamp	Agent metrics related to security principal collection

Column Name	SQL Server	PostgreSQL	Notes
<b>accounts_stop_time</b>	datetime2(3)	timestamp	Agent metrics related to security principal collection
<b>accounts_object_count</b>	integer	integer	Agent metrics related to security principal collection
<b>transfer_start_time</b>	datetime2(3)	timestamp	Related to transfer of scan file from the Agent to the Engine
<b>transfer_stop_time</b>	datetime2(3)	timestamp	Related to transfer of scan file from the Agent to the Engine
<b>db_start_time</b>	datetime2(3)	timestamp	Database insert start time
<b>db_stop_time</b>	datetime2(3)	timestamp	Database insert stop time
<b>status_code</b>	integer	integer	Internal status code
<b>error_string</b>	nvarchar(1024)	varchar(1024)	

## srs.scan\_targets

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>network_path</b>	nvarchar(256)	varchar(256)	Root path for scan target
<b>network_path_lower</b>	nvarchar(256)	[ Not applicable ]	Computed column
<b>server</b>	nvarchar(256)	varchar(256)	
<b>identity_system_id</b>	integer	integer	Reference to identity_systems table
<b>platform</b>	smallint	smallint	0 = Unknown 1 = Windows
<b>filesystem</b>	smallint	smallint	0 = Unknown 1 = NTFS
<b>cost_per_unit</b>	money	money	Not currently used

## srs.scans

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_policy_id</b>	integer	integer	Reference to scan_policies table
<b>triggered_start_time</b>	datetime2(3)	timestamp	Initial time scan delegation starts
<b>scan_start_time</b>	datetime2(3)	timestamp	Start time when agent begins physical scan
<b>scan_stop_time</b>	datetime2(3)	timestamp	Stop time when agent completes physical scan
<b>enum_start_time</b>	datetime2(3)	timestamp	Agent metrics related to file system object enumeration
<b>enum_stop_time</b>	datetime2(3)	timestamp	Agent metrics related to file system object enumeration
<b>enum_file_count</b>	integer	integer	Agent metrics related to file system object enumeration
<b>enum_directory_count</b>	integer	integer	Agent metrics related to file system object enumeration
<b>enum_link_count</b>	integer	integer	Agent metrics related to file system object enumeration
<b>caching_start_time</b>	datetime2(3)	timestamp	Metrics related to agent caching
<b>caching_stop_time</b>	datetime2(3)	timestamp	Metrics related to agent caching
<b>cached_file_count</b>	integer	integer	Metrics related to agent caching
<b>cached_directory_count</b>	integer	integer	Metrics related to agent caching
<b>cached_link_count</b>	integer	integer	Metrics related to agent caching

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
<b>cache_size</b>	integer	integer	Metrics related to agent caching
<b>cache_size_max</b>	integer	integer	Metrics related to agent caching
<b>metadata_start_time</b>	datetime2(3)	timestamp	Agent metrics related to filesystem metadata collection
<b>metadata_stop_time</b>	datetime2(3)	timestamp	Agent metrics related to filesystem metadata collection
<b>metadata_file_count</b>	integer	integer	Agent metrics related to filesystem metadata collection
<b>metadata_directory_count</b>	integer	integer	Agent metrics related to filesystem metadata collection
<b>metadata_link_count</b>	integer	integer	Agent metrics related to filesystem metadata collection
<b>accounts_start_time</b>	datetime2(3)	timestamp	Agent metrics related to security principal collection
<b>accounts_stop_time</b>	datetime2(3)	timestamp	Agent metrics related to security principal collection
<b>accounts_object_count</b>	integer	integer	Agent metrics related to security principal collection
<b>transfer_start_time</b>	datetime2(3)	timestamp	Related to transfer of scan file from the Agent to the Engine
<b>transfer_stop_time</b>	datetime2(3)	timestamp	Related to transfer of scan file from the Agent to the Engine
<b>db_start_time</b>	datetime2(3)	timestamp	Database insert start time*
<b>db_stop_time</b>	datetime2(3)		Database insert stop time*
<b>scan_type</b>	integer	integer	0 = None 1 = File System Data 2 = Permissions

Column Name	SQL Server	PostgreSQL	Notes
			4 = Volume Free Space
<b>scan_target_id</b>	integer	integer	Reference to scan_targets table
<b>local_identity_system_id</b>	integer	integer	
<b>retry_count</b>	integer	integer	Current number of scan attempts
<b>status_code</b>	integer	integer	Internal status code
<b>error_string</b>	nvarchar(1024)	varchar(1024)	
<b>progress_status</b>	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 6 = Previous 7 = Retained
<b>next_retry_time</b>	datetime2(0)	timestamp	Next scheduled time to retry a failed scan
<b>ntfs_abe_enabled</b>	bit	boolean	Flag indicating that the Windows share has ABE enabled
<b>is_valid</b>	bit	boolean	[Deprecated]
<b>agent_name</b>	nvarchar(256)	varchar(256)	

## srs.security\_descriptors

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_data_id</b>	bigint	bigint	Reference to scan_data table
<b>control</b>	integer	integer	<p>Security descriptor control flags</p> <p>See <a href="https://docs.microsoft.com/en-us/windows/win32/secauthz/security-descriptor-control">https://docs.microsoft.com/en-us/windows/win32/secauthz/security-descriptor-control</a></p> <p>Possible flags:</p> <ul style="list-style-type: none"> <li>0x0001 - Owner defaulted</li> <li>0x0002 - Group defaulted</li> <li>0x0004 - DACL present</li> <li>0x0008 - DACL defaulted</li> <li>0x0010 - SACL present</li> <li>0x0020 - SACL defaulted</li> <li>0x0100 - DACL auto inherit required</li> <li>0x0200 - SACL auto inherit required</li> <li>0x0400 - DACL auto Inherited</li> <li>0x0800 - SACL auto inherited</li> <li>0x1000 - DACL Protected (inheritance disabled)</li> <li>0x2000 - SACL protected (inheritance disabled)</li> <li>0x4000 - Resource Manager control is valid</li> <li>0x8000 - Security Descriptor is self relative</li> </ul>
<b>dacl_present</b>	bit	boolean	Indicates presence of DACL entries

Column Name	SQL Server	PostgreSQL	Notes
			for this security descriptor
<b>sacl_present</b>	bit	boolean	Indicates presence of SAcl entries for this security descriptor

## srs.tend\_volume\_freespace

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	integer	integer	Primary key
<b>scan_id</b>	integer	integer	Scan ID
<b>identity_system</b>	nvarchar(256)	text	
<b>network_path</b>	nvarchar(max)	text	Scan target path
<b>server</b>	nvarchar(256)	text	
<b>filesystem</b>	integer	integer	0 = Unknown 1 = NTFS
<b>volume_guid</b>	uniqueidentifier	uuid	
<b>volume_label</b>	nvarchar(256)	text	
<b>volume_bytes_total</b>	bigint	bigint	
<b>volume_bytes_free</b>	bigint	bigint	
<b>volume_bytes_used</b>	bigint	bigint	
<b>allocation_unit_size</b>	integer	integer	
<b>allocation_units_total</b>	bigint	bigint	
<b>allocation_units_free</b>	bigint	bigint	
<b>allocation_units_used</b>	bigint	bigint	
<b>status</b>	integer	integer	
<b>scan_time</b>	datetime2(0)	timestamp	

## 5.2 - Temp Tables

### tmp\_cq\_fs\_paths

Column Name	SQL Server	PostgreSQL	Notes
<b>report_id</b>	integer	integer	Reference to primary key of associated srs.report_definitions entry
<b>scan_id</b>	integer	integer	Reference to primary key of associated srs.scans entry
<b>scan_type</b>	integer	integer	0 = None 1 = File System Data 2 = Permissions 4 = Volume Free Space
<b>progress_status</b>	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 6 = Previous 7 = Retained
<b>scan_start_time</b>	datetime3(2)	timestamp	Start time when agent begins physical scan
<b>scan_target_id</b>	integer	integer	Reference to primary key of associated srs.scan_targets

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			entry
<b>target_path</b>	nvarchar(max)	text	Selected path for this report
<b>target_path_hash</b>	binary(20)	bytea	SHA-1 hash of normalized target path
<b>path_index</b>	integer	integer	Used for hierarchical relation processing
<b>ns_left</b>	integer	integer	Nested-set Left index – used for hierarchical relation processing
<b>ns_right</b>	integer	integer	Nested-set Right index – used for hierarchical relation processing
<b>path_depth</b>	integer	integer	Entry depth with respect to the scan target's root path
<b>path_type</b>	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub 13 = Reparse Point Unknown

Column Name	SQL Server	PostgreSQL	Notes
			17 = Single Instance Storage Stub 18 = Named Stream
<b>is_permission_scan</b>	bit	boolean	Flag indicating whether this entry is for a file system permissions scan Can be used in place of scan_type
<b>is_filesystem_scan</b>	bit	boolean	Flag indicating whether this entry is for a file system metadata scan Can be used in place of scan_type
<b>is_current</b>	bit	boolean	Flag indicating whether this entry is for the current scan Can be used in place of progress_status
<b>is_previous</b>	bit	boolean	Flag indicating whether this entry is for a previous scan Can be used in place of progress_status
<b>is_baseline</b>	bit	boolean	Flag indicating whether this entry is for a baseline scan Can be used in place of progress_status

## 5.3 - Views

## ad.ds\_objects\_view

Column Name	SQL Server	PostgreSQL	Notes
<b>forest_dns</b>	nvarchar(256)	varchar(256)	Forest DNS name
<b>domain_dns</b>	nvarchar(256)	varchar(256)	Domain DNS name
<b>domain_netbios</b>	nvarchar(15)	varchar(15)	Domain NetBIOS name
<b>id</b>	bigint	bigint	Primary key
<b>dn</b>	nvarchar(max)	text	Distinguished name
<b>db_domain_sid</b>	nvarchar(256)	varchar(256)	SID of the domain itself
<b>db_last_update</b>	datetime2(3)	timestamp	Last update time for this entry in the database
<b>account_expires</b>	datetime2(0)	timestamp	
<b>create_timestamp</b>	datetime2(0)	timestamp	
<b>department</b>	nvarchar(64)	varchar(64)	
<b>description</b>	nvarchar(1024)	varchar(1024)	Only uses first value of this multi-value attribute
<b>display_name</b>	nvarchar(256)	varchar(256)	
<b>dns_host_name</b>	nvarchar(2048)	varchar(2048)	Applies to Computer objects
<b>given_name</b>	nvarchar(64)	varchar(64)	
<b>group_type</b>	integer	integer	See <a href="https://docs.microsoft.com/en-us/windows/win32/adschema/a-grouptype">https://docs.microsoft.com/en-us/windows/win32/adschema/a-grouptype</a> for details.  Flags: 0x01 - System created group 0x02 - Global group

Column Name	SQL Server	PostgreSQL	Notes
			<p>0x04 - Domain Local group</p> <p>0x08 - Universal group</p> <p>0x10 - APP_BASIC group for Windows Server Authorization Manager</p> <p>0x20 - APP_QUERY group for Windows Server Authorization Manager</p> <p>0x80000000 - Security Group. If not set, then a Distribution Group</p>
<b>last_logon_timestamp</b>	datetime2(0)	timestamp	<p>NOTE: This attribute only has 14-day granularity.</p> <p>See: <a href="https://docs.microsoft.com/en-us/windows/win32/adschema/a-lastlogontimestamp">https://docs.microsoft.com/en-us/windows/win32/adschema/a-lastlogontimestamp</a></p>
<b>mail</b>	nvarchar(256)	varchar(256)	
<b>managed_by_guid</b>	nvarchar(36)	varchar(36)	GUID of referenced DS object
<b>manager_guid</b>	nvarchar(36)	varchar(36)	GUID of referenced DS object
<b>object_category</b>	nvarchar(256)	varchar(256)	Using LDAP display name, not FDN.
<b>object_class</b>	nvarchar(256)	varchar(256)	Only includes structural class value from this multi-value attribute.
<b>object_guid</b>	nvarchar(36)	varchar(36)	Object's GUID
<b>object_sid</b>	nvarchar(256)	varchar(256)	Object's Security Identifier
<b>primary_group_sid</b>	varbinary(68)	varchar(256)	SID of referenced object
<b>sam_account_name</b>	nvarchar(256)	varchar(256)	SAM account name
<b>sam_account_type</b>	integer	integer	See <a href="https://docs.microsoft.com/en-us/windows/win32/adschema/a-samaccounttype">https://docs.microsoft.com/en-us/windows/win32/adschema/a-samaccounttype</a> for details.

5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			<p>Enum values:</p> <p>0x00000000 - Domain</p> <p>0x10000000 - Group</p> <p>0x10000001 - Non-security Group object</p> <p>0x20000000 - Alias object</p> <p>0x20000001 - Non-security Alias object</p> <p>0x30000000 - Normal User account</p> <p>0x30000001 - Machine (computer) account</p> <p>0x30000002 - Trust account</p> <p>0x40000000 - APP_BASIC Group</p> <p>0x40000001 - APP_QUERY Group</p>
<b>sam_principal_name</b>	nvarchar(256)	varchar(256)	<p>NetBIOS\SamAccountName. From msDS-PrincipalName.</p> <p>Note that the NetBIOS name here may be different from the associated domain NetBIOS name where this account was scanned.</p> <p>This is especially true for domain Builtin\* accounts and foreign security principals.</p>
<b>surname</b>	nvarchar(64)	varchar(64)	
<b>title</b>	nvarchar(128)	varchar(128)	
<b>uac_flags</b>	integer	integer	<p>Combines both userAccessControl and msDs-User-Account-Control-Computed attribute values into a single flag.</p> <p>See the following for details:</p>

Column Name	SQL Server	PostgreSQL	Notes
			<ul style="list-style-type: none"> <li>• <a href="https://docs.microsoft.com/en-us/windows/win32/adschema/a-useraccountcontrol">https://docs.microsoft.com/en-us/windows/win32/adschema/a-useraccountcontrol</a></li> <li>• <a href="https://docs.microsoft.com/en-us/windows/win32/adschema/a-msds-user-account-control-computed">https://docs.microsoft.com/en-us/windows/win32/adschema/a-msds-user-account-control-computed</a></li> </ul> <p>Flags values:</p> <p>0x00000001 - Logon script is executed</p> <p>0x00000002 - User Account disabled</p> <p>0x00000008 - Home directory required</p> <p>0x00000010 - Account currently locked out</p> <p>0x00000020 - No password required</p> <p>0x00000040 - User cannot change password</p> <p>0x00000080 - User can send encrypted password</p> <p>0x00000100 - Temporary duplicate account</p> <p>0x00000200 - Normal account - typical user</p> <p>0x00000800 - Inter-domain trust account</p> <p>0x00001000 - Computer (Workstation / Member Server) account</p> <p>0x00002000 - Domain controller computer account</p>

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			<p>0x00010000 - Password does not expire</p> <p>0x00020000 - Majority Node Set (MNS) logon account</p> <p>0x00040000 - Smart card required for logon</p> <p>0x00080000 - Service account trusted for Kerberos delegation</p> <p>0x00100000 - Account not allowed trust for delegation</p> <p>0x00200000 - Account can only use DES keys</p> <p>0x00400000 - Account does not require Kerberos pre-authentication for logon</p> <p>0x00800000 - User password has expired</p> <p>0x01000000 - Account enabled for delegation</p> <p>0x04000000 - Partial secrets account</p> <p>0x08000000 - Account can only use Use AES keys</p>
<b>upn</b>	nvarchar(1024)	varchar(1024)	User principal name

## srs.baseline\_fs\_scandata

Column Name	SQL Server	PostgreSQL	Notes
<b>identity_system</b>	nvarchar(256)	varchar(256)	Identity system name
<b>domain</b>	nvarchar(256)	varchar(256)	Active Directory domain
<b>server</b>	nvarchar(256)	varchar(256)	Server name
<b>scan_target</b>	nvarchar(256)	varchar(256)	UNC root path for scan target
<b>fullpath</b>	nvarchar(max)	text	Full UNC path to the file system entry
<b>name</b>	nvarchar(256)	varchar(256)	File or directory name
<b>filename_extension</b>	nvarchar(32)	varchar(32)	File name extension
<b>create_time</b>	datetime2(0)	timestamp	Stored as UTC time
<b>modify_time</b>	datetime2(0)	timestamp	Stored as UTC time
<b>access_time</b>	datetime2(0)	timestamp	Stored as UTC time
<b>size</b>	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
<b>size_on_disk</b>	bigint	bigint	Assumes typical allocation unit size of 4K
<b>size_compressed</b>	bigint	bigint	Only accurate for NTFS file systems
<b>owner_identity_system</b>	nvarchar(256)	varchar(256)	Owner's Identity System name
<b>owner_domain</b>	nvarchar(256)	varchar(256)	Owner's Active Directory domain
<b>owner_name</b>	nvarchar(256)	varchar(256)	SAM Account name
<b>owner_fdn</b>	nvarchar(512)	varchar(512)	Full distinguished object name
<b>owner_display_name</b>	nvarchar(max)	text	Domain\SamAccountName

5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
<b>owner_id</b>	varbinary(68)	bytea	Security Identifier (SID)
<b>attributes</b>	integer	integer	0x0 = None 0x1 = Read Only 0x2 = Archive 0x4 = System 0x8 = Hidden 0x10 = Directory 0x20 = Compressed 0x40 = Offline 0x80 = NTFS device 0x100 = NTFS Normal 0x200 = NTFS Temporary 0x400 = NTFS Sparse File 0x800 = NTFS Reparse Point 0x1000 = NTFS Not content indexed 0x2000 = NTFS Encrypted 0x4000 = NTFS Virtual
<b>attribute_string</b>	nvarchar(256)	varchar(256)	See srs.attribute_string function
<b>fullpath_hash</b>	binary(20)	bytea	SHA-1 hash of lowercase fullpath
<b>idx</b>	integer	integer	Scan index; unique per scan
<b>parent_idx</b>	integer	integer	Parent index. Used for hierarchical relation processing
<b>path_depth</b>	integer	integer	Entry depth with respect to the scan target's root path.

Column Name	SQL Server	PostgreSQL	Notes
<b>ns_left</b>	integer	integer	Nested-set Left index – used for hierarchical relation processing
<b>ns_right</b>	integer	integer	Nested-set Right index – used for hierarchical relation processing
<b>scan_id</b>	integer	integer	Reference to scans table
<b>scan_data_id</b>	bigint	bigint	Reference to scan_data table
<b>path_type</b>	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub 13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
<b>status_code</b>	integer	integer	

## srs.baseline\_fs\_scans

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_id</b>	integer	integer	Reference to scans table
<b>identity_system</b>	nvarchar(256)	varchar(256)	Identity system name
<b>domain</b>	nvarchar(256)	varchar(256)	Active Directory domain
<b>server</b>	nvarchar(256)	varchar(256)	Server name
<b>scan_target</b>	nvarchar(256)	varchar(256)	UNC root path for scan target
<b>platform</b>	integer	integer	0 = Unknown 1 = Windows
<b>filesystem</b>	integer	integer	0 = Unknown 1 = NTFS
<b>scan_type</b>	integer	integer	Should always be 1
<b>progress_status</b>	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 6 = Previous 7 = Retained
<b>identity_system_id</b>	integer	integer	

Column Name	SQL Server	PostgreSQL	Notes
<b>scan_target_id</b>	integer	integer	
<b>status_code</b>	integer	integer	
<b>ntfs_abe_enabled</b>	bit	boolean	Flag indicating that the Windows share has ABE enabled
<b>agent</b>	nvarchar(256)	varchar(256)	Name of agent that performed the scan
<b>file_count</b>	integer	integer	Number of files in the scan
<b>directory_count</b>	integer	integer	Number of directories in the scan
<b>link_count</b>	integer	integer	Number of links (junctions, symbolic links, reparse points) in the scan

## srs.baseline\_ntfs\_aces

Column Name	SQL Server	PostgreSQL	Notes
<b>identity_system</b>	nvarchar(256)	varchar(256)	Identity system name
<b>domain</b>	nvarchar(256)	varchar(256)	Active Directory domain
<b>server</b>	nvarchar(256)	varchar(256)	Server name
<b>scan_target</b>	nvarchar(256)	varchar(256)	UNC root path for scan target
<b>fullpath</b>	nvarchar(max)	text	Full UNC path to the file system entry
<b>trustee_identity_system</b>	nvarchar(256)	varchar(256)	Trustee's Identity System name
<b>trustee_domain</b>	nvarchar(256)	varchar(256)	Trustee's Active Directory domain
<b>trustee_name</b>	nvarchar(256)	varchar(256)	SAMAccount name
<b>trustee_fdn</b>	nvarchar(512)	varchar(512)	Full distinguished name
<b>trustee_display_name</b>	nvarchar(max)	text	DOMAIN\SAMAccount
<b>trustee_type</b>	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer
<b>sid</b>	varbinary(68)	bytea	
<b>access_mask</b>	integer	integer	0x1 = Read Data / List Directory 0x2 = Write Data / Create File 0x4 = Append Data / Create Subdirectory 0x8 = Read Extended Attributes 0x10 = Write Extended Attributes 0x20 = File Execute / Traverse

Column Name	SQL Server	PostgreSQL	Notes
			0x40 = Delete Child 0x80 = Read Attributes 0x100 = Write Attributes 0x10000 = Delete 0x20000 = Read Permissions 0x40000 = Change Permissions 0x80000 = Change Owner 0x100000 = Synchronize 0x1000000 = Access System Security 0x10000000 = Generic All 0x20000000 = Generic Execute 0x40000000 = Generic Write 0x80000000 = Generic Read
<b>access_mask_string</b>	nvarchar(128)	varchar(128)	See srs.access_mask_string
<b>basic_permissions</b>	nvarchar(128)	varchar(128)	See srs.access_mask_basic_string
<b>ace_type</b>	smallint	smallint	0 = Access Allowed 1 = Access Denied 2 = System Audit 9 = Allowed Callback 10 = Denied Callback 13 = System Audit Callback 17 = System Mandatory Label
<b>ace_type_string</b>	nvarchar(128)	varchar(128)	See srs.ace_type_string
<b>ace_flags</b>	smallint	smallint	0x1 = Object Inherit

5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			0x2 = Container Inherit 0x4 = No Propagate 0x8 = Inherit Only 0x10 = Inherited 0x40 = Successful Access 0x80 = Failed Access
<b>ace_flags_string</b>	nvarchar(128)	varchar(128)	See srs.ace_flags_string
<b>idx</b>	integer	integer	Scan index; unique per scan
<b>parent_idx</b>	integer	integer	Parent index. Used for hierarchical relation processing
<b>path_depth</b>	integer	integer	Entry depth with respect to the scan target's root path.
<b>ns_left</b>	integer	integer	Nested-set Left index – used for hierarchical relation processing
<b>ns_right</b>	integer	integer	Nested-set Right index – used for hierarchical relation processing
<b>scan_id</b>	integer	integer	Reference to scans table
<b>scan_data_id</b>	bigint	bigint	Reference to scan_data table
<b>path_type</b>	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share

Column Name	SQL Server	PostgreSQL	Notes
			8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub 13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
<b>status_code</b>	integer	integer	
<b>identity_system_id</b>	integer	integer	Reference to identity_systems table
<b>scan_target_id</b>	integer	integer	Reference to scan_targets table
<b>ad_object_id</b>	integer	integer	Reference to ad_objects table

## srs.baseline\_permissions\_scans

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_id</b>	integer	integer	Reference to scans table
<b>identity_system</b>	nvarchar(256)	varchar(256)	Identity system name
<b>domain</b>	nvarchar(256)	varchar(256)	Active Directory domain
<b>server</b>	nvarchar(256)	varchar(256)	Server name
<b>scan_target</b>	nvarchar(256)	varchar(256)	UNC root path for scan target
<b>platform</b>	smallint	smallint	0 = Unknown 1 = Windows
<b>filesystem</b>	smallint	smallint	0 = Unknown 1 = NTFS
<b>scan_type</b>	integer	integer	Should always be 2
<b>progress_status</b>	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 6 = Previous 7 = Retained
<b>identity_system_id</b>	integer	integer	Reference to identity_systems

Column Name	SQL Server	PostgreSQL	Notes
			table
<b>scan_target_id</b>	integer	integer	Reference to scan_targets table
<b>status_code</b>	integer	integer	
<b>ntfs_abe_enabled</b>	bit	boolean	Flag indicating that the Windows share has ABE enabled
<b>agent</b>	nvarchar(256)	varchar(256)	Name of agent that performed the scan
<b>directory_count</b>	integer	integer	Number of directories in the scan

## srs.current\_fs\_scandata

Column Name	SQL Server	PostgreSQL	Notes
<b>identity_system</b>	nvarchar(256)	varchar(256)	Identity system name
<b>domain</b>	nvarchar(256)	varchar(256)	Active Directory domain
<b>server</b>	nvarchar(256)	varchar(256)	Server name
<b>scan_target</b>	nvarchar(256)	varchar(256)	UNC root path for scan target
<b>fullpath</b>	nvarchar(max)	text	Full UNC path to the file system entry
<b>name</b>	nvarchar(256)	varchar(256)	File or directory name
<b>filename_extension</b>	nvarchar(32)	varchar(32)	File name extension
<b>create_time</b>	datetime2(0)	timestamp	Stored as UTC time
<b>modify_time</b>	datetime2(0)	timestamp	Stored as UTC time
<b>access_time</b>	datetime2(0)	timestamp	Stored as UTC time
<b>size</b>	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
<b>size_on_disk</b>	bigint	bigint	Assumes typical allocation unit size of 4K
<b>size_compressed</b>	bigint	bigint	Only accurate for NTFS file systems
<b>owner_identity_system</b>	nvarchar(256)	varchar(256)	Owner's Identity System name
<b>owner_domain</b>	nvarchar(256)	varchar(256)	Owner's Active Directory domain
<b>owner_name</b>	nvarchar(256)	varchar(256)	SAM Account name
<b>owner_fdn</b>	nvarchar(512)	varchar(512)	Full distinguished object name
<b>owner_display_name</b>	nvarchar(max)	text	Domain\SamAccountName

Column Name	SQL Server	PostgreSQL	Notes
<b>owner_id</b>	varbinary(68)	bytea	Security Identifier (SID)
<b>attributes</b>	integer	integer	0x0 = None 0x1 = Read Only 0x2 = Archive 0x4 = System 0x8 = Hidden 0x10 = Directory 0x20 = Compressed 0x40 = Offline 0x80 = NTFS device 0x100 = NTFS Normal 0x200 = NTFS Temporary 0x400 = NTFS Sparse File 0x800 = NTFS Reparse Point 0x1000 = NTFS Not content indexed 0x2000 = NTFS Encrypted 0x4000 = NTFS Virtual
<b>attribute_string</b>	nvarchar(256)	varchar(256)	See srs.attribute_string function
<b>fullpath_hash</b>	binary(20)	bytea	SHA-1 hash of lowercase fullpath
<b>idx</b>	integer	integer	Scan index; unique per scan
<b>parent_idx</b>	integer	integer	Parent index. Used for hierarchical relation processing
<b>path_depth</b>	integer	integer	Entry depth with respect to the scan target's root path.
<b>ns_left</b>	integer	integer	Nested-set Left index – used for hierarchical relation

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			processing
<b>ns_right</b>	integer	integer	Nested-set Right index – used for hierarchical relation processing
<b>scan_id</b>	integer	integer	Reference to scans table
<b>scan_data_id</b>	bigint	bigint	Reference to scan_data table
<b>path_type</b>	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub 13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
<b>status_code</b>	integer	integer	

## srs.current\_fs\_scans

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_id</b>	integer	integer	Reference to scans table
<b>identity_system</b>	nvarchar(256)	varchar(256)	Identity system name
<b>domain</b>	nvarchar(256)	varchar(256)	Active Directory domain
<b>server</b>	nvarchar(256)	varchar(256)	Server name
<b>scan_target</b>	nvarchar(256)	varchar(256)	UNC root path for scan target
<b>platform</b>	integer	integer	0 = Unknown 1 = Windows
<b>filesystem</b>	integer	integer	0 = Unknown 1 = NTFS
<b>scan_type</b>	integer	integer	Should always be 1
<b>progress_status</b>	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 6 = Previous 7 = Retained
<b>identity_system_id</b>	integer	integer	Reference to identity_systems

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			table
<b>scan_target_id</b>	integer	integer	Reference to scan_targets table
<b>status_code</b>	integer	integer	
<b>ntfs_abe_enabled</b>	bit	boolean	Flag indicating that the Windows share has ABE enabled
<b>is_valid</b>	bit	boolean	[Deprecated]
<b>agent</b>	nvarchar(256)	varchar(256)	Name of agent that performed the scan
<b>file_count</b>	integer	integer	Number of files in the scan
<b>directory_count</b>	integer	integer	Number of directories in the scan
<b>link_count</b>	integer	integer	Number of links (junctions, symbolic links, reparse points) in the scan

## srs.current\_ntfs\_aces

Column Name	SQL Server	PostgreSQL	Notes
<b>identity_system</b>	nvarchar(256)	varchar(256)	Identity system name
<b>domain</b>	nvarchar(256)	varchar(256)	Active Directory domain
<b>server</b>	nvarchar(256)	varchar(256)	Server name
<b>scan_target</b>	nvarchar(256)	varchar(256)	UNC root path for scan target
<b>fullpath</b>	nvarchar(max)	text	Full UNC path to the file system entry
<b>trustee_identity_system</b>	nvarchar(256)	varchar(256)	Trustee's Identity System name
<b>trustee_domain</b>	nvarchar(256)	varchar(256)	Trustee's Active Directory domain
<b>trustee_name</b>	nvarchar(256)	varchar(256)	SAMAccount name
<b>trustee_fdn</b>	nvarchar(512)	varchar(512)	Full distinguished name
<b>trustee_display_name</b>	nvarchar(max)	text	DOMAIN\SAMAccount
<b>trustee_type</b>	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer
<b>sid</b>	varbinary(68)	bytea	
<b>access_mask</b>	integer	integer	0x1 = Read Data / List Directory 0x2 = Write Data / Create File 0x4 = Append Data / Create Subdirectory 0x8 = Read Extended Attributes 0x10 = Write Extended Attributes 0x20 = File Execute / Traverse

5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			0x40 = Delete Child 0x80 = Read Attributes 0x100 = Write Attributes 0x10000 = Delete 0x20000 = Read Permissions 0x40000 = Change Permissions 0x80000 = Change Owner 0x100000 = Synchronize 0x1000000 = Access System Security 0x10000000 = Generic All 0x20000000 = Generic Execute 0x40000000 = Generic Write 0x80000000 = Generic Read
<b>access_mask_string</b>	nvarchar(128)	varchar(128)	See srs.access_mask_string
<b>basic_permissions</b>	nvarchar(128)	varchar(128)	See srs.access_mask_basic_string
<b>ace_type</b>	smallint	smallint	0 = Access Allowed 1 = Access Denied 2 = System Audit 9 = Allowed Callback 10 = Denied Callback 13 = System Audit Callback 17 = System Mandatory Label
<b>ace_type_string</b>	nvarchar(128)	varchar(128)	See srs.ace_type_string
<b>ace_flags</b>	smallint	smallint	0x1 = Object Inherit

Column Name	SQL Server	PostgreSQL	Notes
			0x2 = Container Inherit 0x4 = No Propagate 0x8 = Inherit Only 0x10 = Inherited 0x40 = Successful Access 0x80 = Failed Access
<b>ace_flags_string</b>	nvarchar(128)	varchar(128)	See srs.ace_flags_string
<b>idx</b>	integer	integer	Scan index; unique per scan
<b>parent_idx</b>	integer	integer	Parent index. Used for hierarchical relation processing
<b>path_depth</b>	integer	integer	Entry depth with respect to the scan target's root path.
<b>ns_left</b>	integer	integer	Nested-set Left index – used for hierarchical relation processing
<b>ns_right</b>	integer	integer	Nested-set Right index – used for hierarchical relation processing
<b>scan_id</b>	integer	integer	Reference to scans table
<b>scan_data_id</b>	bigint	bigint	Reference to scan_data table
<b>path_type</b>	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub 13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
<b>status_code</b>	integer	integer	
<b>identity_system_id</b>	integer	integer	Reference to identity_systems table
<b>scan_target_id</b>	integer	integer	Reference to scan_targets table
<b>ad_object_id</b>	integer	integer	Reference to ad_objects table

## srs.current\_permissions\_scans

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_id</b>	integer	integer	Reference to scans table
<b>identity_system</b>	nvarchar(256)	varchar(256)	Identity system name
<b>domain</b>	nvarchar(256)	varchar(256)	Active Directory domain
<b>server</b>	nvarchar(256)	varchar(256)	Server name
<b>scan_target</b>	nvarchar(256)	varchar(256)	UNC root path for scan target
<b>platform</b>	smallint	smallint	0 = Unknown 1 = Windows
<b>filesystem</b>	smallint	smallint	0 = Unknown 1 = NTFS
<b>scan_type</b>	integer	integer	Should always be 2
<b>progress_status</b>	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 6 = Previous 7 = Retained
<b>identity_system_id</b>	integer	integer	Reference to identity_systems

## 5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			table
<b>scan_target_id</b>	integer	integer	Reference to scan_targets table
<b>status_code</b>	integer	integer	
<b>ntfs_abe_enabled</b>	bit	boolean	Flag indicating that the Windows share has ABE enabled
<b>is_valid</b>	bit	boolean	[Deprecated]
<b>agent</b>	nvarchar(256)	varchar(256)	Name of agent that performed the scan
<b>directory_count</b>	integer	integer	Number of directories in the scan

## srs.previous\_fs\_scandata

Column Name	SQL Server	PostgreSQL	Notes
<b>identity_system</b>	nvarchar(256)	varchar(256)	Identity system name
<b>domain</b>	nvarchar(256)	varchar(256)	Active Directory domain
<b>server</b>	nvarchar(256)	varchar(256)	Server name
<b>scan_target</b>	nvarchar(256)	varchar(256)	UNC root path for scan target
<b>fullpath</b>	nvarchar(max)	text	Full UNC path to the file system entry
<b>name</b>	nvarchar(256)	varchar(256)	File or directory name
<b>filename_extension</b>	nvarchar(32)	varchar(32)	File name extension
<b>create_time</b>	datetime2(0)	timestamp	Stored as UTC time
<b>modify_time</b>	datetime2(0)	timestamp	Stored as UTC time
<b>access_time</b>	datetime2(0)	timestamp	Stored as UTC time
<b>size</b>	bigint	bigint	For files, actual size; for directories, accumulative size of all subordinate files
<b>size_on_disk</b>	bigint	bigint	Assumes typical allocation unit size of 4K
<b>size_compressed</b>	bigint	bigint	Only accurate for NTFS file systems
<b>owner_identity_system</b>	nvarchar(256)	varchar(256)	Owner's Identity System name
<b>owner_domain</b>	nvarchar(256)	varchar(256)	Owner's Active Directory domain
<b>owner_name</b>	nvarchar(256)	varchar(256)	SAM Account name
<b>owner_fdn</b>	nvarchar(512)	varchar(512)	Full distinguished object name
<b>owner_display_name</b>	nvarchar(max)	text	Domain\SamAccountName

5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
<b>owner_id</b>	varbinary(68)	bytea	Security Identifier (SID)
<b>attributes</b>	integer	integer	0x0 = None 0x1 = Read Only 0x2 = Archive 0x4 = System 0x8 = Hidden 0x10 = Directory 0x20 = Compressed 0x40 = Offline 0x80 = NTFS device 0x100 = NTFS Normal 0x200 = NTFS Temporary 0x400 = NTFS Sparse File 0x800 = NTFS Reparse Point 0x1000 = NTFS Not content indexed 0x2000 = NTFS Encrypted 0x4000 = NTFS Virtual
<b>attribute_string</b>	nvarchar(256)	varchar(256)	See srs.attribute_string function
<b>fullpath_hash</b>	binary(20)	bytea	SHA-1 hash of lowercase fullpath
<b>idx</b>	integer	integer	Scan index; unique per scan
<b>parent_idx</b>	integer	integer	Parent index. Used for hierarchical relation processing
<b>path_depth</b>	integer	integer	Entry depth with respect to the scan target's root path.
<b>ns_left</b>	integer	integer	Nested-set Left index – used for hierarchical relation

Column Name	SQL Server	PostgreSQL	Notes
			processing
<b>ns_right</b>	integer	integer	Nested-set Right index – used for hierarchical relation processing
<b>scan_id</b>	integer	integer	Reference to scans table
<b>scan_data_id</b>	bigint	bigint	Reference to scan_data table
<b>path_type</b>	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share 8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub 13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
<b>status_code</b>	integer	integer	

## srs.previous\_fs\_scans

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_id</b>	integer	integer	Reference to scans table
<b>identity_system</b>	nvarchar(256)	varchar(256)	Identity system name
<b>domain</b>	nvarchar(256)	varchar(256)	Active Directory domain
<b>server</b>	nvarchar(256)	varchar(256)	Server name
<b>scan_target</b>	nvarchar(256)	varchar(256)	UNC root path for scan target
<b>platform</b>	integer	integer	0 = Unknown 1 = Windows
<b>filesystem</b>	integer	integer	0 = Unknown 1 = NTFS
<b>scan_type</b>	integer	integer	Should always be 1
<b>progress_status</b>	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 6 = Previous 7 = Retained
<b>identity_system_id</b>	integer	integer	

Column Name	SQL Server	PostgreSQL	Notes
<b>scan_target_id</b>	integer	integer	
<b>status_code</b>	integer	integer	
<b>ntfs_abe_enabled</b>	bit	boolean	Flag indicating that the Windows share has ABE enabled
<b>agent</b>	nvarchar(256)	varchar(256)	Name of agent that performed the scan
<b>file_count</b>	integer	integer	Number of files in the scan
<b>directory_count</b>	integer	integer	Number of directories in the scan
<b>link_count</b>	integer	integer	Number of links (junctions, symbolic links, reparse points) in the scan

## srs.previous\_ntfs\_aces

Column Name	SQL Server	PostgreSQL	Notes
<b>identity_system</b>	nvarchar(256)	varchar(256)	Identity system name
<b>domain</b>	nvarchar(256)	varchar(256)	Active Directory domain
<b>server</b>	nvarchar(256)	varchar(256)	Server name
<b>scan_target</b>	nvarchar(256)	varchar(256)	UNC root path for scan target
<b>fullpath</b>	nvarchar(max)	text	Full UNC path to the file system entry
<b>trustee_identity_system</b>	nvarchar(256)	varchar(256)	Trustee's Identity System name
<b>trustee_domain</b>	nvarchar(256)	varchar(256)	Trustee's Active Directory domain
<b>trustee_name</b>	nvarchar(256)	varchar(256)	SAMAccount name
<b>trustee_fdn</b>	nvarchar(512)	varchar(512)	Full distinguished name
<b>trustee_display_name</b>	nvarchar(max)	text	DOMAIN\SAMAccount
<b>trustee_type</b>	integer	integer	0 = Unknown / Other 1 = User 2 = Group 3 = Computer
<b>sid</b>	varbinary(68)	bytea	
<b>access_mask</b>	integer	integer	0x1 = Read Data / List Directory 0x2 = Write Data / Create File 0x4 = Append Data / Create Subdirectory 0x8 = Read Extended Attributes 0x10 = Write Extended Attributes 0x20 = File Execute / Traverse

Column Name	SQL Server	PostgreSQL	Notes
			0x40 = Delete Child 0x80 = Read Attributes 0x100 = Write Attributes 0x10000 = Delete 0x20000 = Read Permissions 0x40000 = Change Permissions 0x80000 = Change Owner 0x100000 = Synchronize 0x1000000 = Access System Security 0x10000000 = Generic All 0x20000000 = Generic Execute 0x40000000 = Generic Write 0x80000000 = Generic Read
<b>access_mask_string</b>	nvarchar(128)	varchar(128)	See srs.access_mask_string
<b>basic_permissions</b>	nvarchar(128)	varchar(128)	See srs.access_mask_basic_string
<b>ace_type</b>	smallint	smallint	0 = Access Allowed 1 = Access Denied 2 = System Audit 9 = Allowed Callback 10 = Denied Callback 13 = System Audit Callback 17 = System Mandatory Label
<b>ace_type_string</b>	nvarchar(128)	varchar(128)	See srs.ace_type_string
<b>ace_flags</b>	smallint	smallint	0x1 = Object Inherit

5 - Schema Reference

Column Name	SQL Server	PostgreSQL	Notes
			0x2 = Container Inherit 0x4 = No Propagate 0x8 = Inherit Only 0x10 = Inherited 0x40 = Successful Access 0x80 = Failed Access
<b>ace_flags_string</b>	nvarchar(128)	varchar(128)	See srs.ace_flags_string
<b>idx</b>	integer	integer	Scan index; unique per scan
<b>parent_idx</b>	integer	integer	Parent index. Used for hierarchical relation processing
<b>path_depth</b>	integer	integer	Entry depth with respect to the scan target's root path.
<b>ns_left</b>	integer	integer	Nested-set Left index – used for hierarchical relation processing
<b>ns_right</b>	integer	integer	Nested-set Right index – used for hierarchical relation processing
<b>scan_id</b>	integer	integer	Reference to scans table
<b>scan_data_id</b>	bigint	bigint	Reference to scan_data table
<b>path_type</b>	integer	integer	0 = Unknown 1 = File 2 = Directory 3 = File Symbolic Link 4 = Directory Symbolic Link 5 = Junction 6 = Mount Point 7 = Share

Column Name	SQL Server	PostgreSQL	Notes
			8 = Volume 9 = DFS Link 10 = DFS Folder 11 = DFS Root 12 = HSM Stub 13 = Reparse Point Unknown 17 = Single Instance Storage Stub 18 = Named Stream
<b>status_code</b>	integer	integer	
<b>identity_system_id</b>	integer	integer	Reference to identity_systems table
<b>scan_target_id</b>	integer	integer	Reference to scan_targets table
<b>ad_object_id</b>	integer	integer	Reference to ad_objects table

## srs.previous\_permissions\_scans

Column Name	SQL Server	PostgreSQL	Notes
<b>id</b>	bigint	bigint	Primary key
<b>scan_id</b>	integer	integer	Reference to scans table
<b>identity_system</b>	nvarchar(256)	varchar(256)	Identity system name
<b>domain</b>	nvarchar(256)	varchar(256)	Active Directory domain
<b>server</b>	nvarchar(256)	varchar(256)	Server name
<b>scan_target</b>	nvarchar(256)	varchar(256)	UNC root path for scan target
<b>platform</b>	smallint	smallint	0 = Unknown 1 = Windows
<b>filesystem</b>	smallint	smallint	0 = Unknown 1 = NTFS
<b>scan_type</b>	integer	integer	Should always be 2
<b>progress_status</b>	integer	integer	-2 = Waiting for retry -1 = Ready for cleanup 0 = Waiting for delegation 1 = Delegated / scan in progress 2 = Scan file transfer in progress 3 = Database update in progress 4 = Current - scan process complete 5 = Database update pending 6 = Previous 7 = Retained
<b>identity_system_id</b>	integer	integer	Reference to identity_systems

Column Name	SQL Server	PostgreSQL	Notes
			table
<b>scan_target_id</b>	integer	integer	Reference to scan_targets table
<b>status_code</b>	integer	integer	
<b>ntfs_abe_enabled</b>	bit	boolean	Flag indicating that the Windows share has ABE enabled
<b>agent</b>	nvarchar(256)	varchar(256)	Name of agent that performed the scan
<b>directory_count</b>	integer	integer	Number of directories in the scan

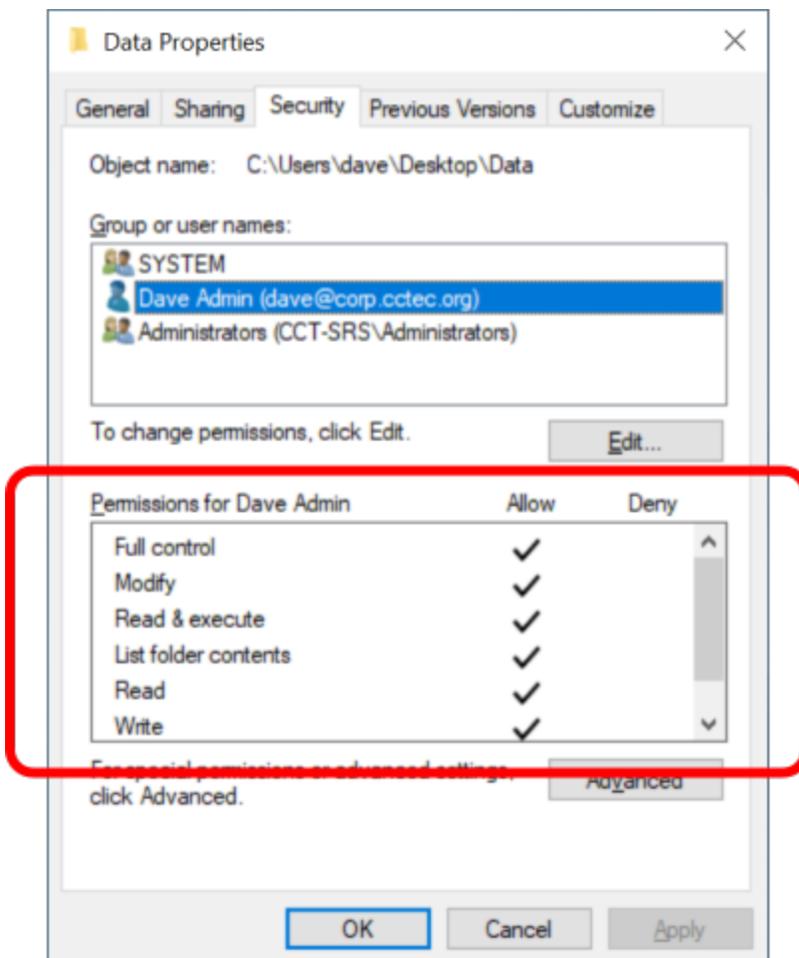
## 5.4 - Functions

srs.access\_mask\_basic\_string

Parameters	SQL Server	PostgreSQL
@mask	integer	integer
@path_type	integer	integer
Return Value	nvarchar(128)	varchar(128)

**Description:** Converts an NTFS access mask value to its basic permissions string equivalent.

Note that the values displayed here are functionally equivalent to what is seen in the primary window of the security tab for an NTFS file system entry:



- Entries having permissions that do not fit the basic permissions (such as Special permissions) include an asterisk \*.
- The path\_type is required since the same flags represent different semantic values for folders, files and shares. Path type must be one of 1 (file), 2 (folder) or 7 (share)
- Permissions flags are mapped to one or more of the following values:
  - Full Control
  - Modify
  - Read & Execute
  - List Folder Contents (Folders only)
  - Read
  - Write
  - Special Permissions

### Example (SQL Server)

```

1 | SELECT TOP(100)
2 |     sd.fullpath,
3 |     srs.access_mask_basic_string(ntfs.access_mask, 2) AS basic_
   | permissions
4 | FROM srs.ntfs_aces AS ntfs
5 | JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id
6 | WHERE sd.path_type = 2;

```

### Example (PostgreSQL)

```

1 | SELECT
2 |     sd.fullpath,
3 |     srs.access_mask_basic_string(ntfs.access_mask, 2) AS basic_
   | permissions
4 | FROM srs.ntfs_aces AS ntfs
5 | JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id
6 | WHERE sd.path_type = 2
7 | LIMIT 100;

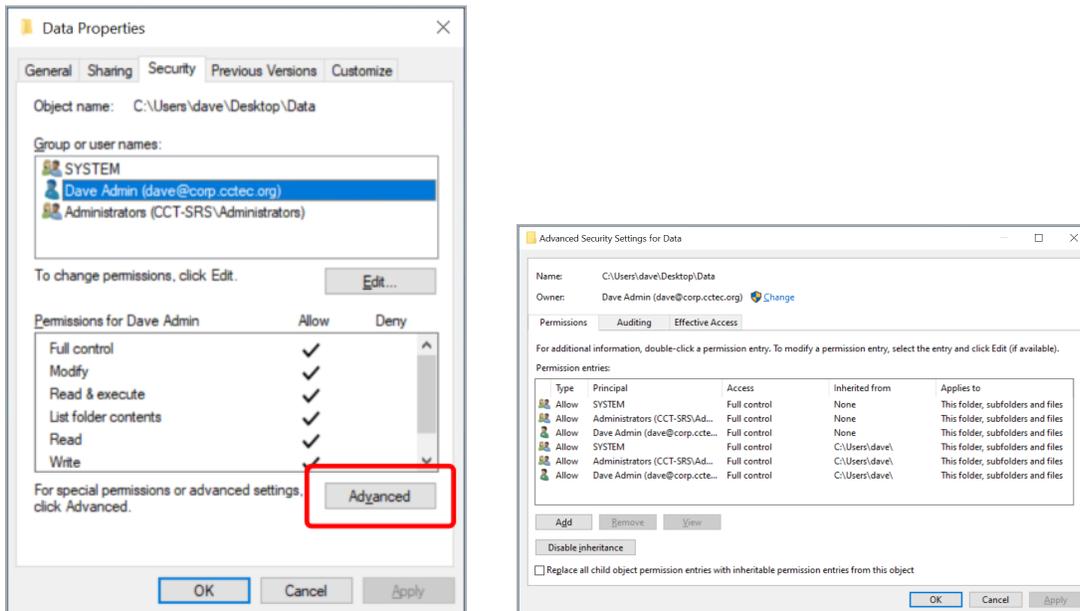
```

srs.access\_mask\_string

Parameters	SQL Server	PostgreSQL
@mask	integer	integer
@path_type	integer	integer
Return Value	nvarchar(128)	varchar(128)

**Description:** Converts an NTFS access mask value to its advanced permissions string equivalent.

Note that the values displayed here are functionally equivalent to what is seen in the advanced section of the security tab for an NTFS file system entry:



- The path\_type is required since the same flags represent different semantic values for folders, files and shares. Path type must be one of 1 (file), 2 (folder) or 7 (share)

- Flags correspond to the following values:

0x00000001	Rd / Lf	Read data / List folder
0x00000002	Wd / Cf	Write data / Create file
0x00000004	Ad / Cs	Append data / Create subdirectory
0x00000008	Rx	Read extended attributes
0x00000010	Wx	Write extended attributes

0x00000020	Xf / Tf	File execute / Traverse
0x00000040	Ds	Delete child (subdirectory)
0x00000080	Ra	Read attributes
0x00000100	Wa	Write attributes
0x00010000	De	Delete
0x00020000	Rp	Read permissions
0x00040000	Cp	Change permissions
0x00080000	To	Change owner (take ownership)
0x00100000	Sy	Synchronize
0x01000000	Ss	Access system security
0x10000000	Ga	Generic All
0x20000000	Ge	Generic Execute
0x40000000	Gw	Generic Write
0x80000000	Gr	Generic Read

### Example (SQL Server)

```

1 | SELECT TOP(100)
2 |     sd.fullpath,
3 |     srs.access_mask_string(ntfs.access_mask, sd.path_type) AS
   | access_mask
4 | FROM srs.ntfs_aces AS ntfs
5 | JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id;

```

### Example (PostgreSQL)

```

1 | SELECT
2 |     sd.fullpath,
3 |     srs.access_mask_string(ntfs.access_mask, sd.path_type) AS
   | access_mask
4 | FROM srs.ntfs_aces AS ntfs
5 | JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id
6 | LIMIT 100;

```

## srs.ace\_flags\_string

Parameters	SQL Server	PostgreSQL
@flags	integer	integer
Return Value	nvarchar(128)	varchar(128)

**Description:** Converts the access mask flags to a string representation. Flags are converted as follows:

0x001	(OI)	Object inherit
0x002	(CI)	Container inherit
0x004	(NP)	No propagate
0x008	(IO)	Inherit only
0x010	(ID)	Inherited
0x040	(SA)	Successful access
0x080	(FA)	Failed access

**Example (SQL Server)**

```

1 | SELECT TOP(100)
2 |     sd.fullpath,
3 |     srs.access_mask_string(ntfs.access_mask, sd.path_type) AS
   | access_mask,
4 |     srs.ace_flags_string(ntfs.flags) AS ace_flags
5 | FROM srs.ntfs_aces AS ntfs
6 | JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id;

```

**Example (PostgreSQL)**

```

1 | SELECT
2 |     sd.fullpath,
3 |     srs.access_mask_string(ntfs.access_mask, sd.path_type) AS
   | access_mask,
4 |     srs.ace_flags_string(ntfs.flags) AS ace_flags
5 | FROM srs.ntfs_aces AS ntfs
6 | JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id
7 | LIMIT 100;

```

## srs.ace\_type\_string

Parameters	SQL Server	PostgreSQL
@ace_type	integer	integer
Return Value	nvarchar(128)	varchar(128)

**Description:** Converts the access mask type value to a corresponding text value.

Flags correspond as follows:

0	Access Allowed
1	Access Denied
2	System Audit
3	System Alarm
4	Allowed Compound
5	Allowed Object
6	Denied Object
7	System Audit Object
8	System Alarm Object
9	Allowed Callback
10	Denied Callback
11	Allowed Callback Object
12	Denied Callback Object
13	System Audit Callback
14	System Alarm Callback
15	System Audit Callback Object
16	System Alarm Callback Object
17	System Mandatory Label

For NTFS file systems the primary values of concern are Allowed (0), Denied (1), Audit (2), and System Mandatory Label (17).

**Example (SQL Server)**

```

1 | SELECT TOP(100)
2 |     sd.fullpath,
3 |     srs.access_mask_string(ntfs.access_mask, sd.path_type) AS
   | access_mask,
4 |     srs.ace_flags_string(ntfs.flags) AS ace_flags,

```

```
5 |     srs.ace_type_string(ntfs.ace_type) AS ace_type
6 | FROM srs.ntfs_aces AS ntfs
7 | JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id;
```

### Example (PostgreSQL)

```
1 | SELECT sd.fullpath,
2 |     srs.access_mask_string(ntfs.access_mask, sd.path_type) AS
   | access_mask,
3 |     srs.ace_flags_string(ntfs.flags) AS ace_flags,
4 |     srs.ace_type_string(ntfs.ace_type) AS ace_type
5 | FROM srs.ntfs_aces AS ntfs
6 | JOIN srs.scan_data AS sd ON sd.id = ntfs.scan_data_id
7 | LIMIT 100;
```

## srs.ad\_account\_name

Parameters	SQL Server	PostgreSQL
@domain	nvarchar(1024)	varchar(1024)
@name	nvarchar(1024)	varchar(1024)
@sid	binary(68)	bytea
Return Value	nvarchar(max)	text

**Description:** Converts primary naming values for an Windows security principal to a display name.

- If domain is null or empty, the leading backslash is not included in the result
- If the name is null or empty, the result value is the SDDL sid representation
- If the sid is needed but is invalid, the return value is [Invalid SID]

**Example - Domain and Name**

```
1 | SELECT srs.ad_account_name('BUILTIN', 'Administrators', null);
```

**Example - SID**

```
1 |
2 | SELECT srs.ad_account_name('', '',
   | 0x01020000000000005200000002002000);
```

## srs.attribute\_string

Parameters	SQL Server	PostgreSQL
@flags	integer	integer
Return Value	nvarchar(256)	varchar(256)

**Description:** Converts an attributes value to its equivalent string representation. Flags correspond to the following values:

0x00000000		None
0x00000001	Ro	Read Only
0x00000002	Ar	Archive
0x00000004	Sy	System
0x00000008	Hi	Hidden
0x00000010	Dr	Directory
0x00000020	Co	Compressed
0x00000040	OI	Offline
0x00000080	De	NTFS device
0x00000100	No	NTFS Normal
0x00000200	Te	NTFS Temporary
0x00000400	Sp	NTFS Sparse File
0x00000800	Rp	NTFS Reparse Point
0x00001000	Nc	NTFS Not content indexed
0x00002000	En	NTFS Encrypted
0x00004000	Vi	NTFS Virtual

**Example (SQL Server)**

```

1 | SELECT TOP(100)
2 |     fullpath,
3 |     srs.attribute_string(attributes)
4 | FROM srs.scan_data;
```

**Example (PostgreSQL)**

```

1 | SELECT
```

```
2 |     fullpath,  
3 |     srs.attribute_string(attributes)  
4 | FROM srs.scan_data  
5 | LIMIT 100;
```

## srs.byte\_string

Parameters	SQL Server	PostgreSQL
@size	bigint	bigint
Return Value	nvarchar(64)	text

**Description:** Converts a size value to a string representation of the closest unit.

- The return value has a maximum precision of two decimal places.
- Units include kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB), petabyte (PB), and exabyte (EB).

**Example**

```
1 | SELECT srs.byte_string(1287168);
```

## srs.byte\_unit\_string

Parameters	SQL Server	PostgreSQL
@size	bigint	bigint
@unit	nvarchar(10)	varchar(10)
@precision	integer	integer
Return Value	nvarchar(64)	text

**Description:** Converts a number to a string representation of the specified unit with the specified precision.

- The specified precision is limited to a value from 0 to 3. Values outside this range will be adjusted to 0 or 3 accordingly.
- Unit specifiers are case insensitive and include:
  - byte
  - KB (kilobyte)
  - MB (megabyte)
  - GB (gigabyte)
  - TB (terabyte)
  - PB (petabyte)
  - EB (exabyte)

**Example**

```
1 | SELECT srs.byte_unit_string(1287168, 'KB', 3)
```

## srs.bytes\_to\_hex\_string

Parameters	SQL Server	PostgreSQL
@byte_sequence	varbinary(max)	bytea
Return Value	nvarchar(max)	text

**Description:** Converts a byte sequence to its equivalent hex string representation.

- Returned hex string is lower case with no separators and no prefix.

**Example**

```
1 | SELECT
2 |     srs.bytes_to_hex_string(ad.sid)
3 | FROM srs.ad_objects AS ad;
```

## srs.guid\_bytes

Parameters	SQL Server	PostgreSQL
@guid_text	nvarchar(38)	varchar(38)
Return Value	varbinary(16)	bytea

**Description:** Converts a compatible GUID text string to its equivalent binary representation.

Recommended input format: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

- Surrounding curly braces are optional
- Hex values A-F may be in upper or lower case
- Hyphen separators must be present at the specified 4 locations or not at all.

**Example**

```
1 | SELECT srs.guid_bytes('12345678-1234-5678-9abc-123456789abc');
```

## srs.guid\_text

Parameters	SQL Server	PostgreSQL
@guid_binary	varbinary(16)	bytea
Return Value	nvarchar(36)	varchar(36)

**Description:** Converts a binary GUID value to its equivalent string representation.

Note that returned strings are in the canonical lower-case GUID format xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx.

**Example**

```
1 | SELECT fdn, srs.guid_text(guid) FROM srs.ad_objects WHERE  
   | id=1;
```

## srs.hex\_string\_to\_bytes

Parameters	SQL Server	PostgreSQL
@byte_sequence	varbinary(max)	bytea
Return Value	nvarchar(max)	text

**Description:** Converts a hex string to its equivalent bytes.

- Hex values A-F may be in upper or lower case
- Hex string must be a proper string with an even number of characters —leading zeros are required for each hex value having a single digit.
- Do not include separators such as hyphens between hex values

### Example

```
1 | SELECT srs.hex_string_to_bytes('01ab3d4407');
```

## srs.path\_hash

Parameters	SQL Server	PostgreSQL
@path	nvarchar(max)	text
Return Value	binary(20)	bytea

**Description:** Returns the binary SHA-1 hash for a given path.

- The input path is first converted to lower case
- The input path is then converted to byte representation using the default text encoding of the database for string values (typically UTF-8 on PostgreSQL, and Unicode UCS-2 on SQL Server)
- Useful for finding a path in the srs.scan\_data table using the fullpath\_hash indexed column

**Example**

```
1 | SELECT * FROM srs.scan_data
2 | WHERE fullpath_hash = srs.path_hash('\\server-
   | 1.ad.cctec.org\Users\user1');
```

## srs.path\_hash\_sha256

Parameters	SQL Server	PostgreSQL
@path	nvarchar(max)	text
Return Value	binary(32)	bytea

**Description:** Returns the binary SHA256 hash for a given path.

- The input path is first converted to lower case
- The input path is then converted to byte representation using the default text encoding of the database for string values (typically UTF-8 on PostgreSQL, and Unicode UCS-2 on SQL Server)
- Useful for finding a path (web URL) in the ms365.drive\_items table using the web\_url\_hash indexed column

**Example**

```

1 | SELECT * FROM ms365.drive_items
2 | WHERE web_url_hash = srs.path_hash_sha256
   | ('https://mysite.sharepoint.com/sites/Shared%20Documents');

```

## srs.sid\_bytes

Parameters	SQL Server	PostgreSQL
@sid	nvarchar(256)	varchar(256)
Return Value	varbinary(68)	bytea

**Description:** Converts an SDDL representation of a Security Identifier value to its binary form.

- Input SID values must be in proper SDDL form

**Example**

```
1 | SELECT * FROM srs.ad_objects WHERE srs.sid_bytes('S-1-5-32-544') = sid;
```

## srs.sid\_text

Parameters	SQL Server	PostgreSQL
@sid_bytes	varbinary(68)	bytea
Return Value	nvarchar(256)	varchar(256)

**Description:** Converts a binary Security Identifier to its SDDL string representation.

**Example**

```
1 | SELECT domain, name, srs.sid_text(sid) FROM srs.ad_objects;
```