Axon DataStore v1 User Guide

Rev: 27 Jan 2026

Axon Enterprise, Inc.
17800 N 85th St
Scottsdale AZ 85255
USA

Contents

What's New?

This guide includes the following updates:

- Fine-grained access control using default and custom access profiles

Axon DataStore v1

The Axon DataStore provides SQL-based, read-only access to data entered in Axon Records and Axon Standards, enabling organizations to run reports, build dashboards, and integrate data with third-party systems. This section outlines how data is structured in the v1 DataStore, explains the use of base and derived views, and provides guidance on querying data, managing access, connecting with supported tools, and ensuring proper firewall configuration. It also includes recommendations for query performance, links to data dictionaries, and tips for using Axon Analytics with Power BI.

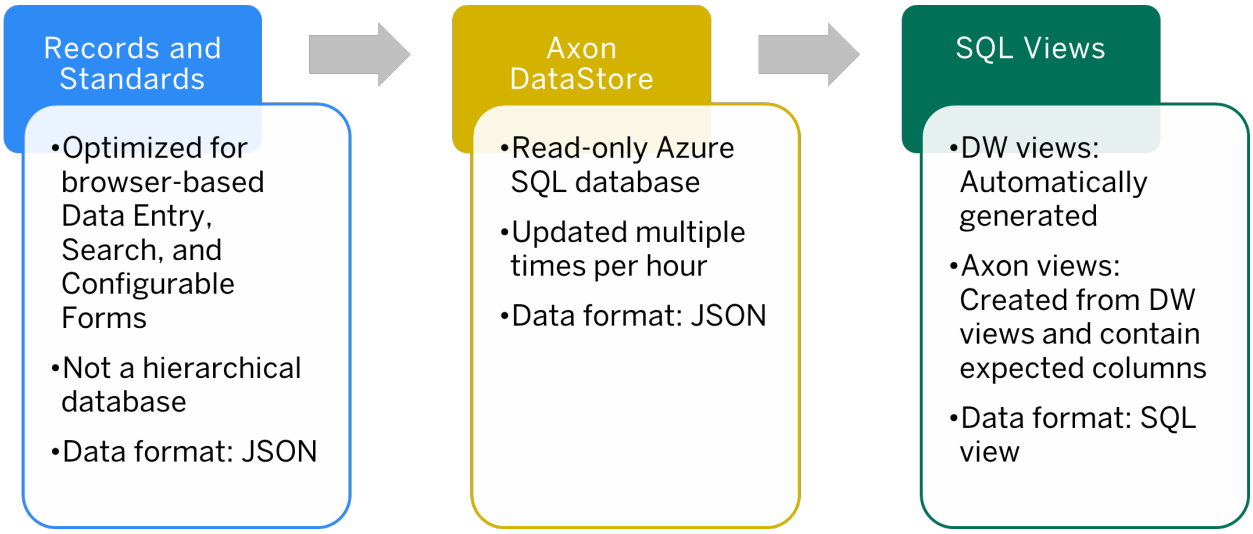
Introduction

The DataStore is a back-end access point to the data your organization's enters into Axon Records and Axon Standards. The DataStore lets users query directly against existing data without relying on the user interface of Axon Records or Axon Standards. Querying the DataStore requires the use of Structured Query Language (SQL) or an integration with an external software (e.g., Power BI, Crystal Reports, etc.). The goal of these queries is to search data, typically for the purposes of Reporting and Analytics.

> **Note**
>
> For server information, please submit a customer support ticket in the CSS portal.

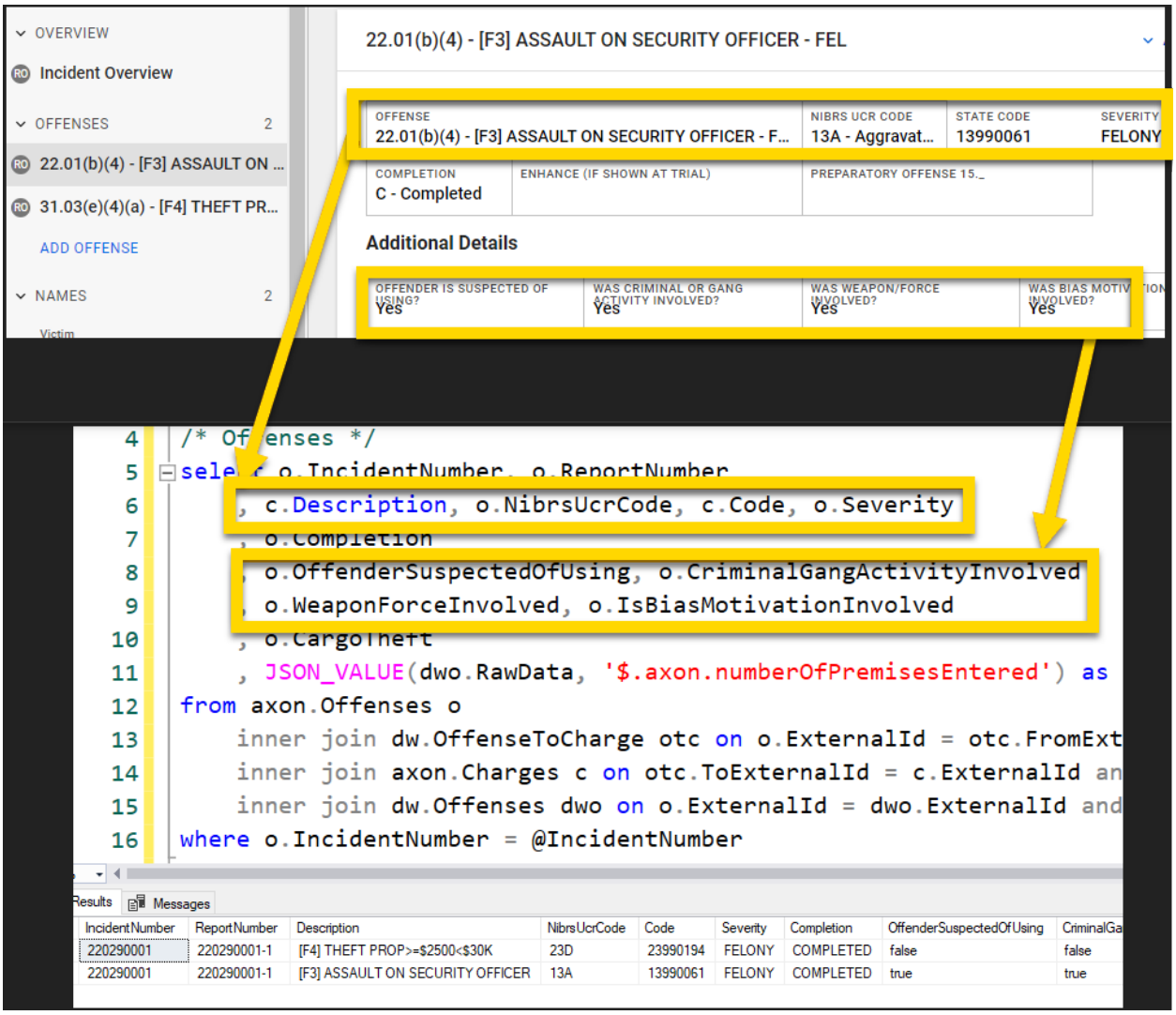
The DataStore is a read-only Azure SQL database you can access for reporting and analytical purposes.

Data is regularly pushed from Axon Records and Axon Standards and arrives at the Axon DataStore in a raw JSON format where it is displayed in a SQL View.



Columns in the Axon DataStore are pulled from the raw JSON using the SQL built-in JSON functions of Azure SQL Database and displayed in a SQL view. The columns in the Axon views match the field labels in the original report where the data was entered.

For example, data entered in the **Offense** section of a report appears in the Axon view as shown below:



For a full list of how the fields in the Axon Records Incident Report map to the columns in the Axon DataStore, see Axon Records DataStore field mappings.

Note

Custom views and form customizations will result in slightly different DataStore views compared to those shown in the link above.

What is in my DataStore?

All forms and events recorded in Axon Records and Axon Standards are broken down into Data Warehouse (**dw**) views. Under the latest "Medium-Rare" schema, the Data Warehouse views are further broken down into corresponding **axon** views. Most **axon** views also contain a **RawData** column where you can find all the data from the front-end, including data that is not parsed into individual columns.

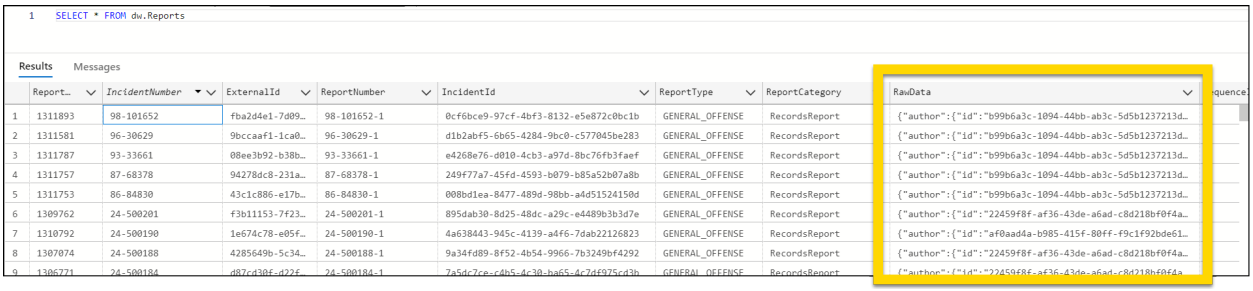
Custom form and field data is recorded in the DataStore but may not parsed into columns unless a custom view has been requested. You can use the **RawData** columns in the **dw** or **axon** views to retrieve this data.

Are there different Axon Records and Axon Standards views?

The Axon Records datastore pulls data from Axon Records, so the out of the box (OOTB, or "automatic") views correspond to Axon Records forms and reports. Some examples of Axon Records specific OOTB views include **axon.Arrests**, **axon.MissingPersons**, and **axon.RecordsLocations**.

Similarly for the Axon Standards datastore, there are OOTB views that correspond to the Axon Standards forms and reports. Examples of these OOTB Axon Standards views include **axon.EISPolicies**, **axon.OfficerInvolvements**, and **axon.UoF**.

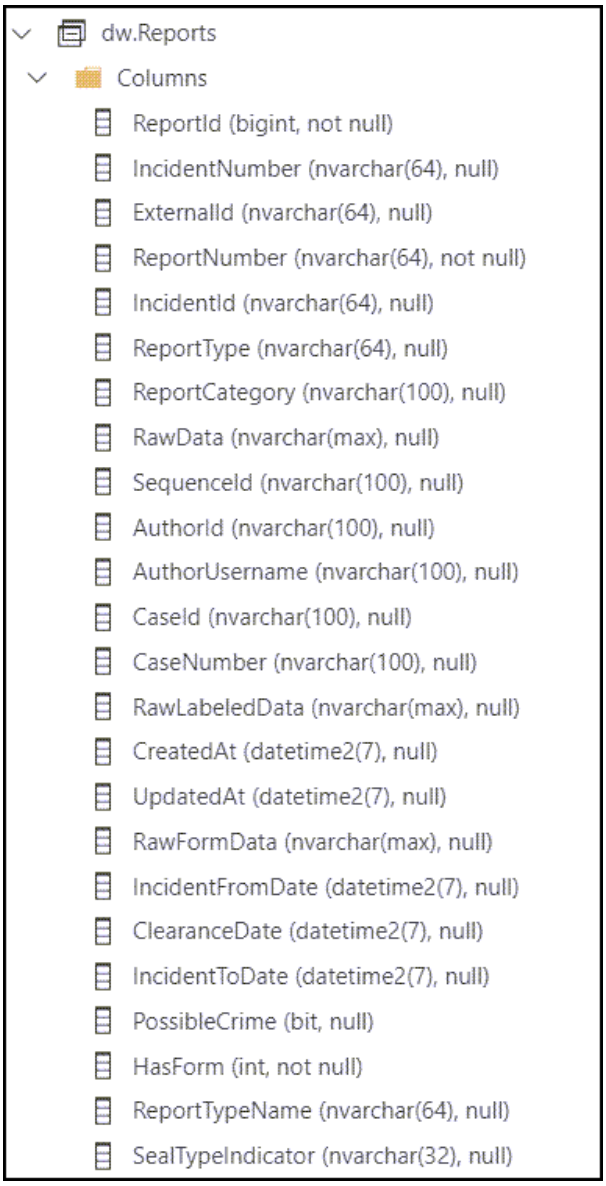
The images below show the **RawData** column, as well as all other columns that appear in the **dw.Reports** view:



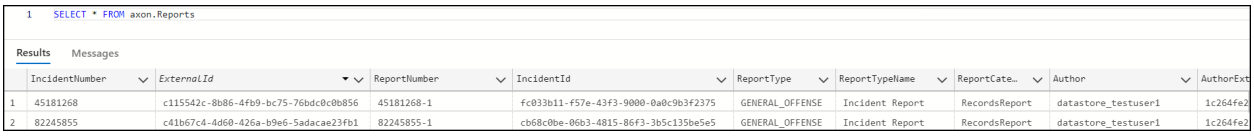
```
SELECT * FROM dw.Reports
```

Results | Messages

	Report…	IncidentNumber	ExternalId	ReportNumber	IncidentId	ReportType	ReportCategory	RawData
1	1311893	98-101652	fba2d4e1-7d09…	98-101652-1	0cf6bce9-97cf-4bf3-8132-e5e872c0bc1b	GENERAL_OFFENSE	RecordsReport	{"author":{"id":"b99b6a3c-1094-44bb-ab3c-5d5b1237213d…
2	1311581	96-30629	9bccaaf1-1ca0…	96-30629-1	d1b2abf5-6b65-4284-9bc0-c577045be283	GENERAL_OFFENSE	RecordsReport	{"author":{"id":"b99b6a3c-1094-44bb-ab3c-5d5b1237213d…
3	1311787	93-33661	08ee3b92-b38b…	93-33661-1	e4268e76-d010-4cb3-a97d-8bc76fb3faef	GENERAL_OFFENSE	RecordsReport	{"author":{"id":"b99b6a3c-1094-44bb-ab3c-5d5b1237213d…
4	1311757	87-68378	94278dc8-231a…	87-68378-1	249f77a7-45fd-4593-b079-b85a52b07a8b	GENERAL_OFFENSE	RecordsReport	{"author":{"id":"b99b6a3c-1094-44bb-ab3c-5d5b1237213d…
5	1311753	86-84830	43c1c886-e17b…	86-84830-1	008bd1ea-8477-489d-98bb-a4d51524150d	GENERAL_OFFENSE	RecordsReport	{"author":{"id":"b99b6a3c-1094-44bb-ab3c-5d5b1237213d…
6	1309762	24-500201	f3b11153-7f23…	24-500201-1	895dab30-8d25-48dc-a29c-e4489b3b3d7e	GENERAL_OFFENSE	RecordsReport	{"author":{"id":"22459f8f-af36-43de-a6ad-c8d218bf0f4a…
7	1310792	24-500190	1e674c78-e05f…	24-500190-1	4a638443-945c-4139-a4f6-7dab22126823	GENERAL_OFFENSE	RecordsReport	{"author":{"id":"af0aad4a-b985-415f-80ff-f9c1f92bde61…
8	1307074	24-500188	4285649b-5c34…	24-500188-1	9a34fd89-8f52-4b54-9966-7b3249bf4292	GENERAL_OFFENSE	RecordsReport	{"author":{"id":"22459f8f-af36-43de-a6ad-c8d218bf0f4a…
9	1306771	24-500184	d87cd30f-d22f…	24-500184-1	7a5dc7ce-c4b5-4c30-ba65-4c7df975cd3b	GENERAL_OFFENSE	RecordsReport	{"author":{"id":"22459f8f-af36-43de-a6ad-c8d218bf0f4a…



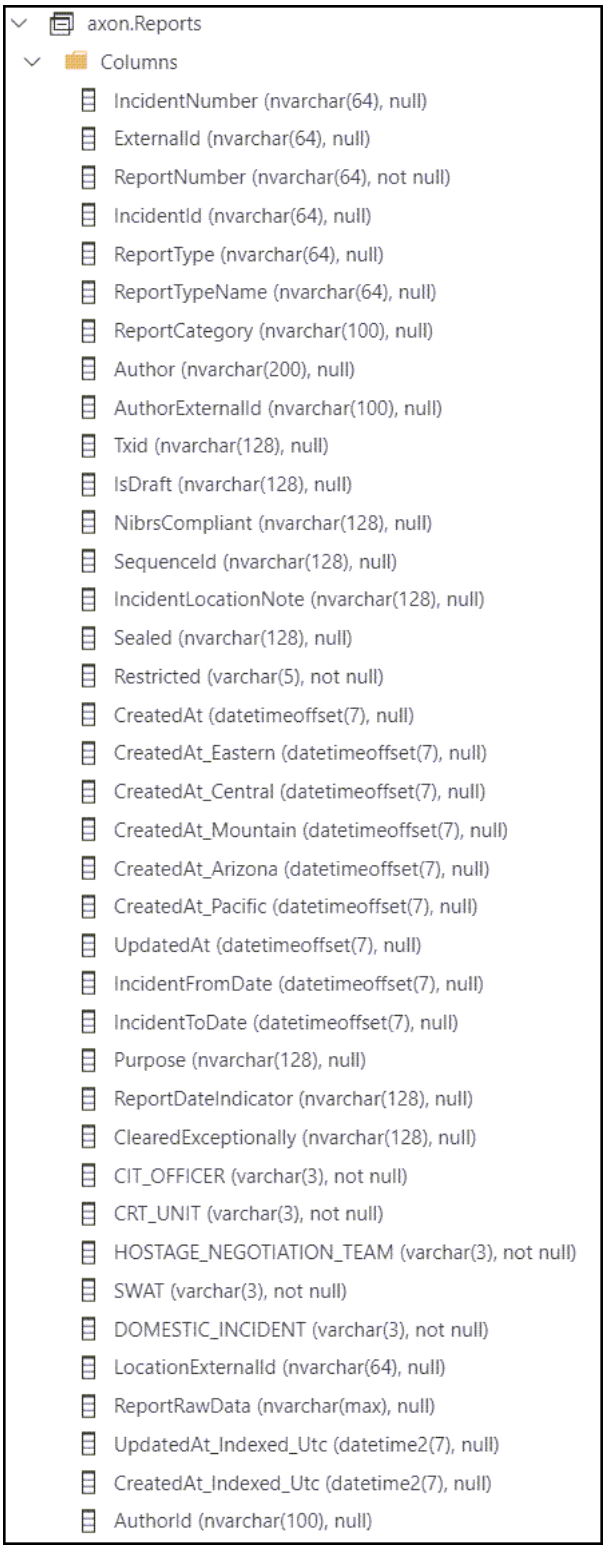
The images below show the columns that appear in the **`axon.Reports`** view:



```
1  SELECT * FROM axon.Reports
```

Results | Messages

	IncidentNumber	ExternalId	ReportNumber	IncidentId	ReportType	ReportTypeName	ReportCate…	Author	AuthorExt
1	45181268	c115542c-8b86-4fb9-bc75-76bdc0c0b856	45181268-1	fc033b11-f57e-43f3-9000-0a0c9b3f2375	GENERAL_OFFENSE	Incident Report	RecordsReport	datastore_testuser1	1c264fe2
2	82245855	c41b67c4-4d60-426a-b9e6-5adacae23fb1	82245855-1	cb68c0be-06b3-4815-86f3-3b5c135be5e5	GENERAL_OFFENSE	Incident Report	RecordsReport	datastore_testuser1	1c264fe2



Watch this video to see an introduction to the Axon DataStore.

Watch this video to see information about joining different tables in the DataStore.

Schemas and views

A schema defines how data is organized in the DataStore. Views in the Axon schema inherit rows from views in the **dw** schema. For this reason, the **dw** schema view is often referred to as the *Base* view, and the **axon** schema view is called the *Derived* view. While the **dw** schema is made up of minimal columns with the JSON object in the **RawData** column, the **axon** schema translates **RawData** into columns that can be selected as normal.

Axon recommends using the **axon** views because they provide:

- A general interface for viewing the data
- An efficient way to obtain, transform, and display the customized forms from Axon Records and Axon Standards.

All data entered in the form fields in Axon Records and Axon Standards reports is available in **axon** views.

Uptime and updates

Axon guarantees a minimum of 99.9% uptime for the Axon solutions 7 days per week on a 24-hour basis, apart from scheduled downtime, scheduled maintenance, and emergency maintenance.

Axon Records and Axon Standards are designed and operated as highly available cloud applications. Multiple redundant components are used throughout the system architecture to ensure high levels of reliability.

Data freshness

Data entered into Axon Records or Axon Standards reports does not appear in the DataStore in real-time. Instead, it appears in the DataStore after 15 to 30 minutes.

DataStore releases

DataStore code updates are performed on a 2-week cycle. Release notes for view changes are added to the Axon Records and Axon Standards release documentation and can be found on Axon Help.

Generally, changes will only add columns or make performance improvements and will not break an existing view or column. A breaking change is typically considered one where a column is removed or a column data type is changed.

Note

If a breaking change is introduced to the DataStore, that change will be first announced in release notes with a published date in the future for when the change will take place.

The release notes will also include the appropriate replacement for what should be used instead. The replacement will generally be available at the same time the breaking change is announced so you can start making changes right away.

Recommendations

In future releases of the Axon Records and Axon Standards DataStore, Axon may augment the definition of a view by adding columns to the end of the column list. We recommend that you do NOT use the syntax **`SELECT * FROM <view name>`** in production code. This syntax will pull more data than necessary and slow the performance of your query. Additionally, because the number of columns returned might change, this syntax could well break your application.

We also recommend that you do NOT use ordinal positioning in production code as there is no guarantee that column ordering will remain the same. To avoid any issues, use column names in your production code.

Disaster recovery

In the event of a major disaster that results in a full loss of a Microsoft Azure region, Axon has created the Axon Cloud Services Information System Contingency Plan (ISCP). The ISCP focuses on the recovery of Axon Records and Axon Standards to a secondary Microsoft Azure region.

Axon is confident that in the event of the complete destruction of a primary Microsoft Azure region, the Axon application services can be recovered and restored in the secondary Microsoft Azure region within, at most, a 24-hour window. However, Axon views the likelihood of such an occurrence as negligible, given the architecture of the underlying Microsoft Azure services.

Firewall requirements

Connectivity to the Axon DataStore requires that you adjust your organization's IP restrictions and network requirements.

IP restriction

All IP traffic to the DataStore is blocked by default. To access the DataStore, your organization's **public** IP address(es) must be added to the allow list. You can manage this allow list from the **Access Policy** tab in the DataStore Settings tool.

Note

The DataStore uses IPv4 (Internet Protocol Version 4) and does NOT currently support IPv6 (Internet Protocol Version 6).

Public vs private IP addresses

When submitting your request for Firewall Access, ensure the IP is a public IP and not a private IP. If your organization uses a private network, you can identify your public IP using any online service such as https://www.whatismyip.com/ or https://www.showmyip.com/.

What are private IP addresses?

Private IP addresses are a subset of IP addresses designated for use within private networks. These addresses are not routable on the public internet, meaning they cannot be used to communicate directly with devices outside the local network, such as the Axon DataStore.

How do I recognize a private IP address?

Private IP addresses are divided into three classes, as defined by the Internet Engineering Task Force (IETF) in RFC 1918:

Class	Range	Prefix	Usage
A	10.0.0.0 to 10.255.255.255	10	Large organizations and enterprises
B	172.16.0.0 to 172.31.255.255	172	Schools, universities, and businesses
C	192.168.0.0 to 192.168.255.255	192	Small office or home networks

Network requirements

Your organization's firewall must allow outbound traffic to the Azure Gov SQL IP Ranges on port 1433.

Refer to the following links for a weekly updated list of the region's SQL IP addresses that your organization should allow:

- Azure IP Ranges and Service Tags - US Government Cloud: US-based organizations
- Azure IP Ranges and Service Tags – Public Cloud: Non-US based organizations

Use the "Sql" Service Tag to make this easier to manage.

Tips

- You can detect updates from one publication to the next by noting increased changeNumber values in the JSON file. Each subsection (e.g., Storage.WestUS) has its own changeNumber that is incremented as changes occur. The top level of the file's changeNumber is incremented when any of the subsections is changed.
- For examples of how to parse the service tag information (e.g., get all address ranges for Storage in WestUS), see the Service Tag Discovery API PowerShell documentation.
- When new IP addresses are added to service tags, they will not be used in Azure for at least one week. This gives you time to update any systems that might need to track the IP addresses associated with service tags.
- You can also ensure connectivity by using Azure's Fully Qualified Domain Name (FQDN), using the server provided in the DataStore access control tool.
  - For information on how to set up access control, please submit a ticket in the Axon Support portal or reach out to softwaresupport@axon.com.

Connect to the DataStore

You can connect to the DataStore using Visual Studio Code or Microsoft SQL Server Management Studio.

Warning

Previously, you could connect to the DataStore using Azure Data Studio. However, this program is being deprecated in February 2026. To continue to access the DataStore, use either VS Code or Microsoft SQL Server Management Studio.

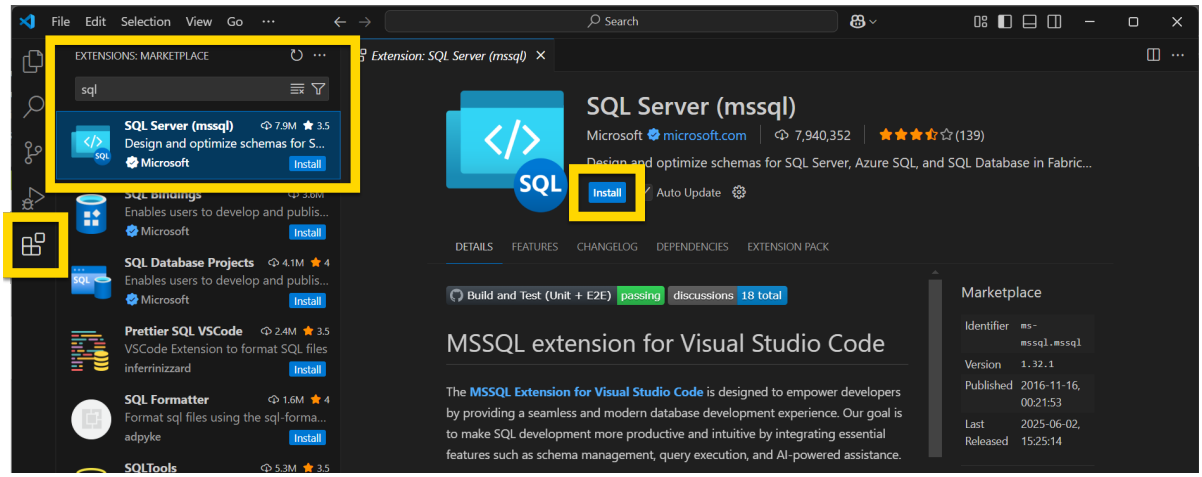
You can generate a DataStore account using the DataStore Secret Generation tool in the Administrator Console.

If you have questions about connecting to the DataStore, submit a ticket using the TSS portal or email softwaresupport@axon.com.

Connect with Visual Studio Code

To connect to the DataStore using VS Code, take these steps:

1. Open VS Code.
   - VS Code is a free software that can be downloaded here.
2. Select **Extensions** in the side menu.
3. Enter "sql" in the search box and select **SQL Server (mssql)**.
4. Select **Install**.

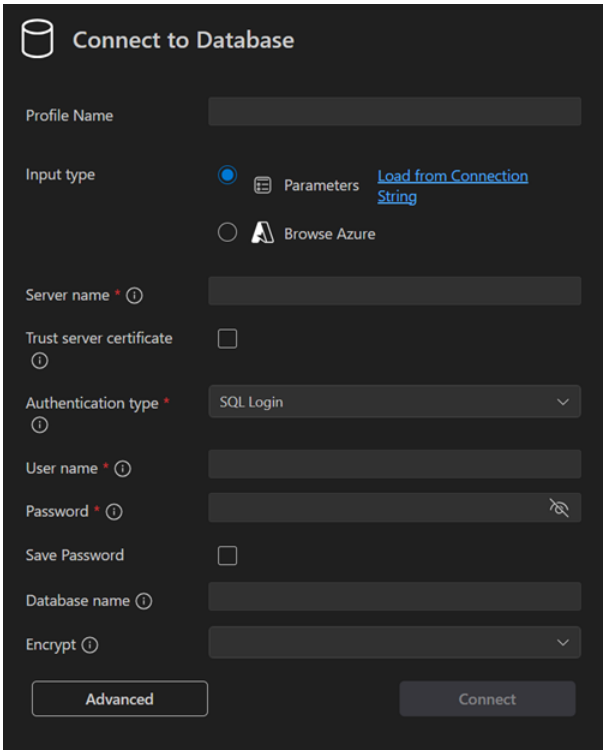


5. After a successful install, **SQL Server** will appear as an option in the side menu. Select this option.
6. Select **Add Connection**.

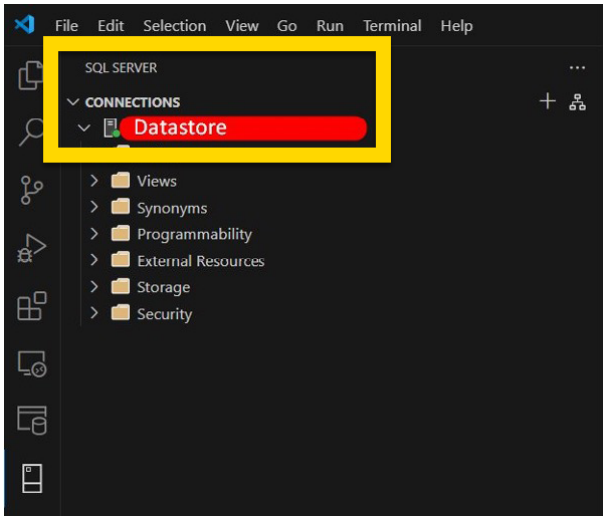
7. Complete the following fields in the Connect to Database window:
   - **Profile name:** This name will appear in your list of available connections.
     - Recommended format: “Axon [Records or Standards] [PROD or Training] [PW or Query] Datastore”
       - Example: Axon Records PROD PW Datastore
   - **Sever name:** The name of the server you are connecting to

- **Authentication type:** SQL Login
- **User name**
- **Password**
- **Save Password:** Check this box to avoid entering your password each time you use VS Code.
- **Database name:** The name of the database you are connecting to

8. Select **Connect**.



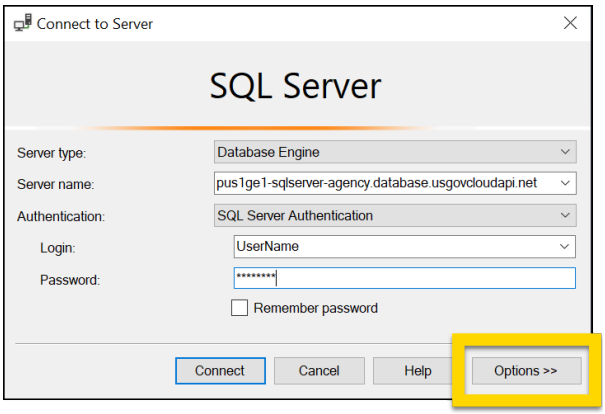
9. Following a successful connection, the Profile name will appear in your Connections list.



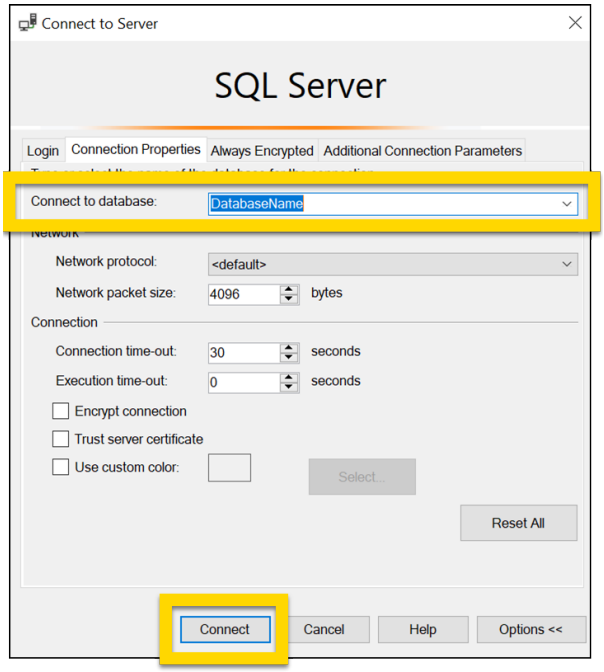
Connect with Microsoft SQL Server Management Studio

To perform queries using Microsoft SQL Server Management Studio, take these steps:

1. Open Microsoft SQL Server Management Studio.
2. A secondary Connect to Server will open.
3. Enter the Server Name, Username, and Password
4. Select **Options**.



5. Enter the Database name in the **Connect to database** field.
6. Select **Connect**.



Query the DataStore

The DataStore uses Structured Query Language (SQL) to display data. Users have Read Only permissions to the DataStore views. They cannot read or write to tables, or create or alter views.

When querying the DataStore, you can use the following SQL functions:

- **`SELECT`**
- **`JOIN`**
- **`WHERE`**
- **`GROUP BY`**
- **`ORDER BY`**

When creating more complex queries or using custom fields, you may also need to use **`CROSS APPLY`** and **`OPENJSON`** functions.

How do the DataStore views join together?

The DataStore views can typically be linked using External ID and Report Number.

* This image is an example and is not an actual Relationship Diagram.

This downloadable file contains sample SQL **`JOIN`** statements you can reference.

Watch this video to see information about joining different tables in the DataStore.

DataStore integrations

Connecting the DataStore to an external product is possible with any database compatible software. The DataStore can be connected via user or service accounts, using a SQL Server or ODBC connection. Axon Support can provide basic instructions on how to integrate the DataStore with your existing reporting software.

Note

If you integrate the DataStore with external products, it is your organization's responsibility to build and maintain any external reports or analytics.

Data dictionaries

The following Excel files contain the Data Dictionaries for the out of the box (OOTB) Axon Records and Axon Standards DataStores.

- Axon Records Data Dictionary
- Axon Standards Data Dictionary

Note

If your organization has custom views or columns these will not be included in the Data Dictionaries.

These Data Dictionaries were last updated on January 12, 2024.

Axon Analytics and Power BI

The Analytics module in Axon Records and Axon Standards uses Power BI to visualize data. The video below provides a comprehensive overview of setting up a dashboard in Power BI, from data import to final customization and filtering.

This video covers the following topics:

- Importing data into Power BI
- Preparing and transforming data
- Building and customizing visualizations
- Customizing dashboard layouts
- Using slicers for data filtering

Once a Power BI dashboard has been created, it can be uploaded into the Analytics module in Axon Axon Records and Axon Standards. The following video uses the Use of Force dashboard in Axon Standards to explain how users can interact with the Power BI dashboards and visualizations.

This video covers the following topics:

- Using tab and visualization filters
- Exploring map visualizations
- Viewing tool tips
- Interacting with tables and hyperlinks
- Dashboard privileges
- Activity log tabs
- Editing and exporting dashboards

Concepts and features

This section provides foundational guidance for working with the Axon DataStore v1, including how key fields like dates, identifiers, and employee data are structured and used. It also covers important concepts such as time zone handling, data types, and naming conventions, along with pre-built reporting views that streamline access to commonly requested records data.

Date and time fields

There are several different naming conventions to be aware of when viewing and using date and time fields from the Axon DataStore, including:

- Data types: datetime2 and datetimeoffset
- Time zone conversions
- Time zone conversion columns
- NoTz time zone columns
- Daylight savings time considerations

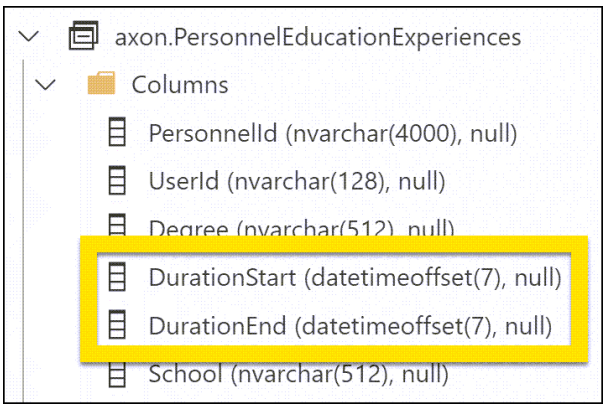
Data types: datetime2 and datetimeoffset

All date and time values in the DataStore are assumed to be in UTC (+00:00) unless a different time zone is explicitly specified.

There is a mixture of **`datetime2`** and **`datetimeoffset`** data types in the DataStore. The **`datetimeoffset`** data type is like **`datetime2`**, except it also has time zone awareness. The **`datetimeoffset`** data type has the time zone offset (as calculated from UTC) appended to the column value. For example:

datetime2:	**`2025-12-10 12:32:10`**
datetimeoffset:	**`2025-12-10 12:32:10 +00:00`**
datetimeoffset converted to EST:	**`2025-12-10 07:32:10 -05:00`**

In the Axon DataStore, you can recognize if the data type is a **`datetimeoffset`** if the time zone offset appears at the end in the format **[+|-] hh:mm**. A SQL Client like Azure Data Studio shows the column data types in a view:



Time zone conversions

For better performance when you are filtering by date, use the UTC version of the field in your **`WHERE`** clause. Then use two **`AT TIME ZONE`** functions on the value you are comparing against, first to your local time zone and second to UTC. Example:

```
WHERE r.[CreatedAt] >

= CONVERT(datetime2, '2024-01-01T00:00:00.000') AT TIME ZONE 'Central Standard
Time' AT TIME ZONE 'UTC'
```

This first converts the value to CST by adding the time zone offset:
`2024-01-01 00:00:00 -06:00`

Then, that CST value is converted to UTC:
`2024-01-01 06:00:00 +00:00`

Comparing the DataStore values against the UTC value (**`2024-01-01 06:00:00 +00:00`**) is the most efficient comparison and yields the fastest results since the values are stored and indexed in UTC.

The most common time zones are listed below, but the full list can be found here.

- Eastern Standard Time
- Central Standard Time
- Mountain Standard Time
- US Mountain Standard Time <-- use this one for Arizona
- Pacific Standard Time

Time zone conversion columns

For commonly selected fields in queries (e.g., **`OccurredTo`** and **`OccurredFrom`**), the values are converted to the five contiguous US time zones. This conversion is denoted by **`_Time Zone`** in the column name. For example:

- **`OccurredFromDate_Eastern`**
- **`OccurredFromDate_Central`**
- **`OccurredFromDate_Mountain`**
- **`OccurredFromDate_Arizona`**
- **`OccurredFromDate_Pacific`**

For better performance, use these converted columns only in select lists and not in any filters or **`WHERE`** clauses. For **`WHERE`** clauses, use the **UTC** column and convert your time zone to UTC as described above.

OccurredFromDate_Mountain	OccurredFromDate_Arizona	OccurredFromDate_Pacific
2018-12-18 18:47:00.0000000 -07:00	2018-12-18 18:47:00.0000000 -07:00	2018-12-18 17:47:00.0000000 -08:00
2021-07-09 10:40:00.0000000 -06:00	2021-07-09 09:40:00.0000000 -07:00	2021-07-09 09:40:00.0000000 -07:00

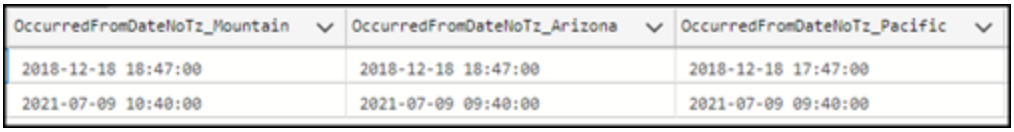
NoTz time zone columns

The time zone offset and millisecond precision can be challenging for tools like Excel or PowerBi to understand, so in some dashboard views the **`datetime2`** data type is converted and the milliseconds and offset (e.g., +5:00) do NOT appear at the end of the value.

To indicate that these columns are different from other views, **`NoTz`** appears in the column name. For example:

- **`OccurredFromDateNoTz_Eastern`**
- **`OccurredFromDateNoTz_Central`**
- **`OccurredFromDateNoTz_Mountain`**
- **`OccurredFromDateNoTz_Arizona`**
- **`OccurredFromDateNpTz_Pacific`**

Note that while the values themselves have been converted to the indicated time zone, the time zone offset is not included at the end of the value. This makes export and analysis to external tools (e.g., Tableau, Excel) easier.



OccurredFromDateNoTz_Mountain	OccurredFromDateNoTz_Arizona	OccurredFromDateNoTz_Pacific
2018-12-18 18:47:00	2018-12-18 18:47:00	2018-12-18 17:47:00
2021-07-09 10:40:00	2021-07-09 09:40:00	2021-07-09 09:40:00

Daylight savings time considerations

When you are performing an analysis where precision and seconds count (e.g., how long did it take to arrive on-scene after being dispatched), Axon recommends using the **`UTC`** column.

Using the **`UTC`** column bypasses complications that arise from inaccurate daylight savings time conversions (which could incorrectly indicate that the officer arrived 60 minutes earlier or later than actual).

Identifiers

The table primary key ID columns are named **`Id`** prefixed by the singular version of the topic contained in the table.

These primary keys are auto-increment bigint values unique to the DataStore; they do not come from the source data and are typically regenerated when the data updates.

Fields ending in **`Number`** (e.g., **`IncidentNumber`**, **`ReportNumber`**, **`CallForServiceNumber`**) generally refer to the friendly ID of the entity in question.

Fields ending in **`ExternalId`** (e.g., **`CallForServiceExternalId`**, **`IncidentExternalId`**) refer to the unique ID in the native system of the entity in question.

- For Axon Records and Axon Standards, these are generally UUIDs.
- For Axon Dispatch, in many cases these are UUIDs, but for things like history records they can be bigint.

Employee data

Employees appear in the DataStore in three ways:

- Users
- Officers
- Authors

Users

In the DataStore, employees listed in `Users` means they were added to a report because they were involved somehow (for example, as a driver, passenger, arresting officer, witness, etc.). As a result, you will see **`ReportNumber`** as a column.

An employee might be listed in **Users** many times since there is a row for each report the employee has been added to.

Officers

`Officers` includes all people employed at the agency. These are not linked to reports, and each employee should only be listed once.

To retrieve employee information for a user, link **`Users`** to **`Officers`** using **`ExternalId`**:

```
SELECT o.FirstName,
    o.LastName,
    o.Username,
    o.BadgeNumber
FROM axon.Users u
INNER JOIN axon.Officers o ON o.OfficerExternalId = u.ExternalId
AND u.ReportNumber = '<insert report number here>'
```

Author

If the **`Author`** of the report is also a **`User`** (they were directly involved in the incident and appear in the report as an involved party), the author is listed in **`axon.Users`** for that report.

If the author was not involved in the incident and is only authoring the report, they are only listed under **`Author`**.

To retrieve the employee information for an author, link **`axon.Reports.AuthorExternalId`** to **`axon.Officers`**:

```
select o.FirstName,
    o.LastName,
    o.Username,
    o.BadgeNumber
FROM dw.Reports r
INNER JOIN axon.Officers o on o.OfficerExternalId = r.AuthorExternalId
AND r.ReportNumber = '<insert report number here>'
```

Reporting views for common Records requests

Reporting views provide all information in one view, so you don't need to join on additional objects. These views can be integrated to create dashboards or external reports an external tool (e.g., PowerBi, Tableau, Excel).

Contact your Axon representative if there are other dashboards type views you would like to see to answer your common reporting needs.

Note

An Incident Report is what Axon uses to refer to a Crime Report or General Offense Report (GO).

Information from Incident Reports are sent to NIBRS and contain data about crimes that occurred.

Offense views

`axon.MostRecentGeneralOffense`

This view contains the most up to date information for an incident report and shows the latest data combined with the most recent contribution or supplement.

This view can give you information, such as:

- List incidents that occurred after Dec 1, 2023

```
SELECT *
FROM axon.MostRecentGeneralOffense
WHERE OccurredFromDate >

CONVERT(datetime2, '2023-12-01') AT TIME ZONE 'Eastern Standard Time' AT TIME ZONE 'UTC'
```

- List incidents that were created in the last 24 hours

```
SELECT *
FROM axon.MostRecentGeneralOffense
WHERE CreatedAt > DATEADD(hour, -24, GETDATE())
```

`axon.OffensesToLatestReport`

This view contains all offenses in the latest information for an incident report. This view contains the following report location columns, so the data can be exported to an external mapping tool:

- Latitude
- Longitude
- District
- Sector
- Zone

This view can give you information, such as:

- List offenses that have individual exceptional clearance dates/types
- How many <insert offense here> happened in this district?

```
SELECT *
FROM axon.OffensesToLatestReport
WHERE NibrsUcrCode = '13A'
   AND District = 'NORTHWEST'
```

- How many <insert offense here> happened during this period?

```
SELECT *
FROM axon.OffensesToLatestReport
WHERE OccurredFromDateNoTz_Eastern > CONVERT(datetime2, '2023-10-01')
AT TIME ZONE 'Eastern Standard Time' AT TIME ZONE 'UTC'
   AND OccurredFromDateNoTz_Eastern < CONVERT(datetime2, '2024-01-01')
AT TIME ZONE 'Eastern Standard Time' AT TIME ZONE 'UTC'
   AND NibrsUcrCode = '220'
```

Vehicle views

`axon.VehicleTheftsAndRecovery`

This view contains all vehicle information where the involvement starts with "STOLEN" or "RECOVER." If there is a recovered date or recovered value, that information is also shown.

This view can give you information, such as:

- How many x make and y model were stolen or recovered in the last 6 months?

```
SELECT *
FROM axon.VehicleTheftsAndRecovery
WHERE Make = 'HONDA'
   AND Model = 'CIVIC'
   AND OccurredFromDate > DATEADD(Month, -6, GETDATE())
```

Property views

Note

The Property views only include data for property items in the Axon Property Management module. These views do not contain data for property recorded in third-party property management systems.

axon.PropertyInCustody

This view allows for easy viewing of property item details (storage/home location, who collected). If the piece of property also has a related incident in Axon Records, some incident information is also included (Latitude, Longitude, Recovered Details).

This view can give you information, such as:

- How many x are on-hand?
- What is the property in location x?
- Who collected property x?

axon.PropertyStolenAndRecovery

This view lists property from incident reports where the involvement contains "STOLEN" or "RECOVER."

This view can give you information, such as:

- How many catalytic converters were stolen in the last 7 days?
- How many black bicycles were stolen this year?

Victim and suspect views

axon.VictimOffensesDashboard

This view allows you to easily see victim information (e.g., Age, Race, Ethnicity) and the related offenses (NIBRS Code, Description, Occurred From/To). Specifically, this view pulls in data for people from the latest incident report where the person's involvement or role is Victim.

This view can give you information, such as:

- How many people were victimized for x offense during last month?
- What were the injuries or weapons related to x offense?

axon.SuspectOffensesDashboard

This view allows you to easily see suspect information (e.g., Age, Race, Ethnicity) and the related offenses (NIBRS Code, Description, Occurred From/To). Specifically, this view pulls in data for people from the latest incident report where the person's involvement or role is Arrestee or Offender.

This view can give you information, such as:

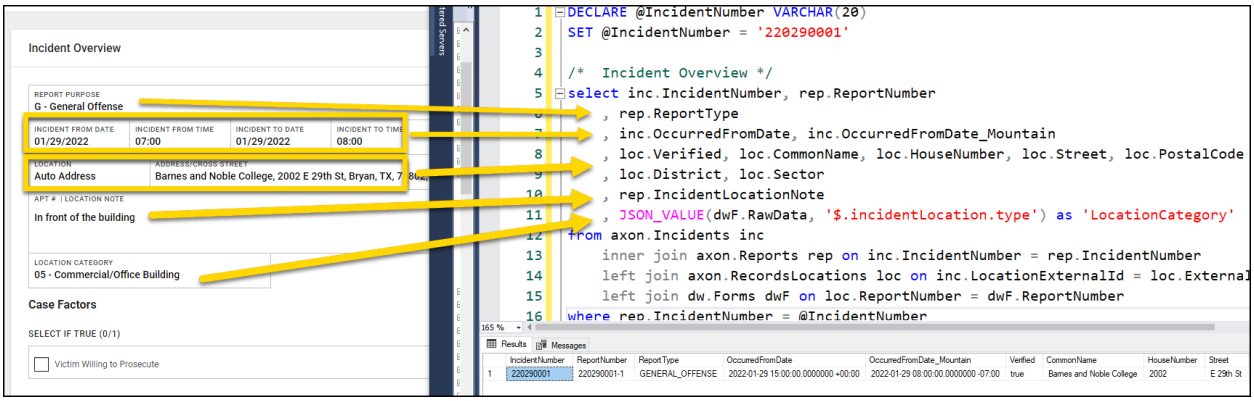
- Demographic information (Height, Sex, Race, etc.) for x offense in the last 7 days
- What weapons were involved with x offense?

Field mappings

This guide shows how the fields in the Axon Records incident report map to the columns in the Axon DataStore. Each section also includes examples of SQL Select statements.

Incident

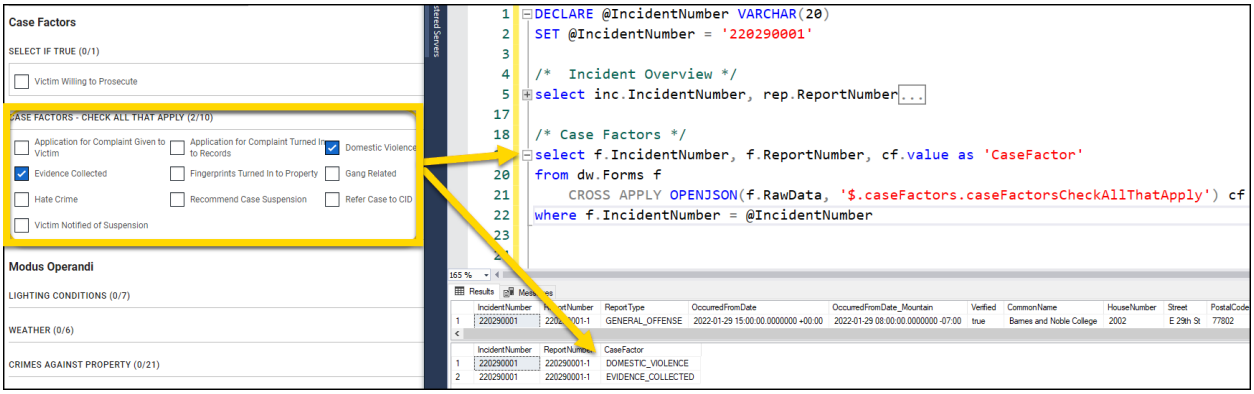
Overview



SQL Select Statement

```
/* Incident Overview */
SELECT inc.IncidentNumber,
    rep.ReportNumber,
    rep.ReportType,
    inc.OccurredFromDate,
    inc.OccurredFromDate_Mountain,
    loc.Verified,
    loc.CommonName,
    loc.HouseNumber,
    loc.Street,
    loc.PostalCode,
    loc.District,
    loc.Sector,
    rep.IncidentLocationNote,
    JSON_VALUE(dwF.RawData, '$.incidentLocation.type') AS 'LocationCategory'
FROM axon.Incidents inc
INNER JOIN axon.Reports rep ON inc.IncidentNumber = rep.IncidentNumber
LEFT JOIN axon.RecordsLocations loc ON inc.LocationExternalId = loc.ExternalId
AND inc.IncidentNumber = loc.IncidentNumber
LEFT JOIN dw.Forms dwF ON loc.ReportNumber = dwF.ReportNumber
WHERE rep.IncidentNumber = @IncidentNumber
```

Case factors

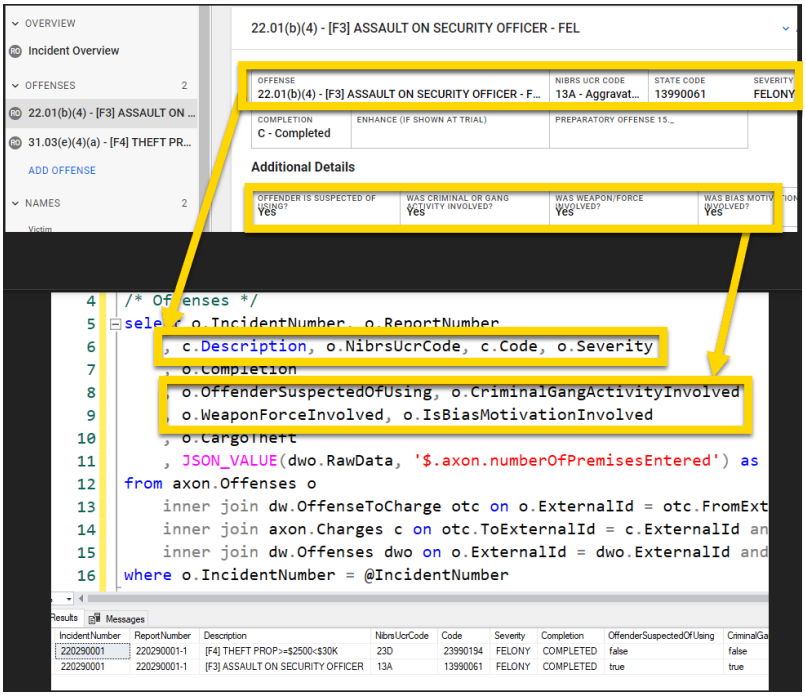


SQL Select Statement

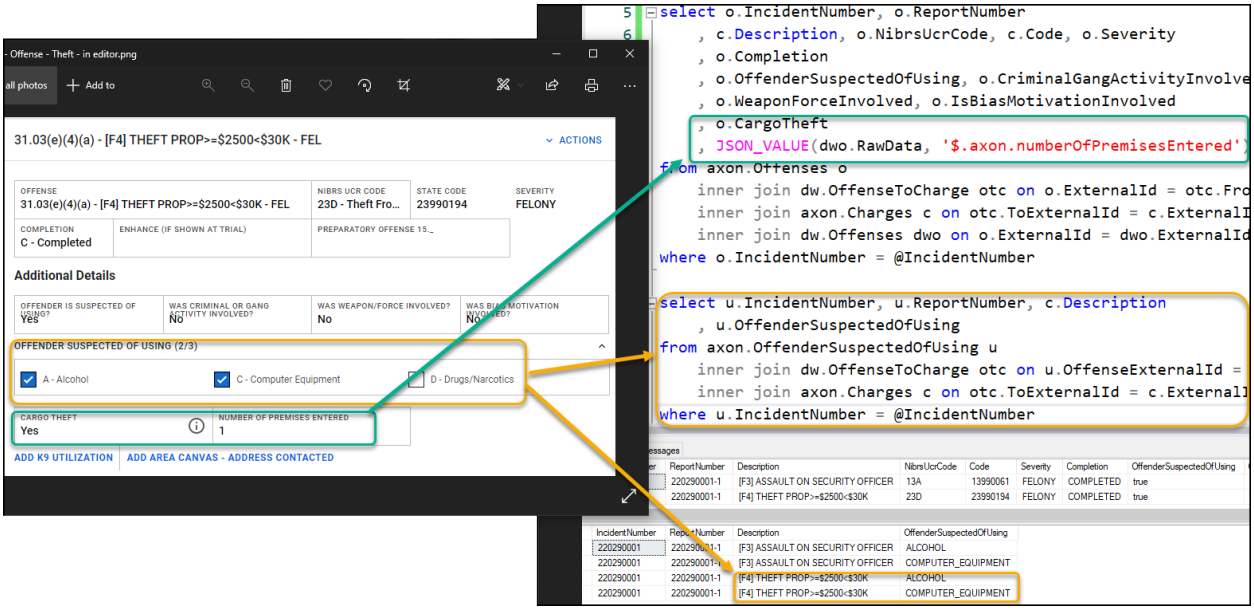
```
/* Case Factors */
SELECT f.IncidentNumber,
    f.ReportNumber,
    cf.value AS 'CaseFactor'
FROM dw.Forms f CROSS APPLY OPENJSON(f.RawData, '$.caseFactors.caseFactors') cf
WHERE f.IncidentNumber = @IncidentNumber
```

Offense

Assault



Theft

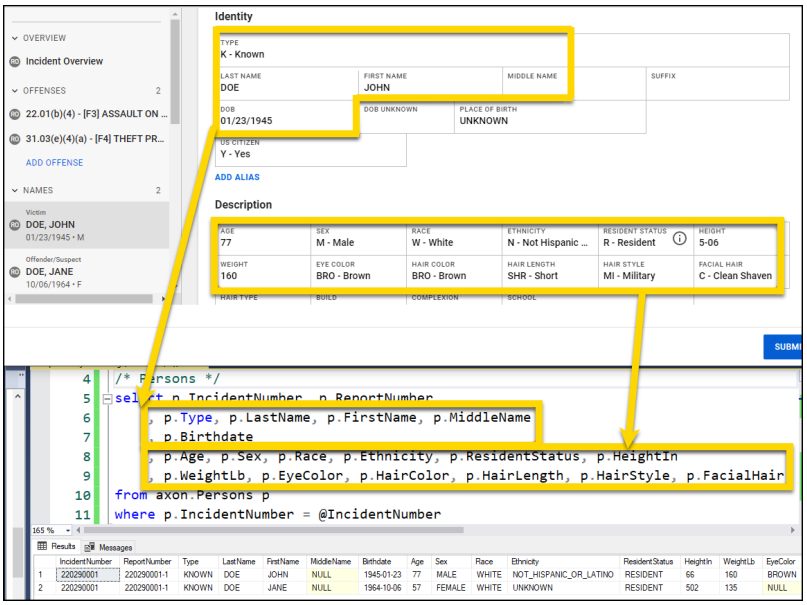


SQL Select Statement

```
/* Offenses */
SELECT o.IncidentNumber,
    o.ReportNumber,
    c.Description,
    o.NibrsUcrCode,
    c.Code,
    o.Severity,
    o.NibrsUcrCode,
    c.Ucr,
    o.Completion,
    o.OffenderSuspectedOfUsing,
    o.CriminalGangActivityInvolved,
    o.WeaponForceInvolved,
    o.IsBiasMotivationInvolved,
    o.CargoTheft /* Suspected of Using */
SELECT u.IncidentNumber,
    u.ReportNumber,
        c.Description,
    u.OffenderSuspectedOfUsing
FROM axon.OffenderSuspectedOfUsing u
INNER JOIN dw.OffenseToCharge otc ON u.OffenseExternalId = otc.FromExternalId
AND otc.ReportNumber = u.ReportNumber
INNER JOIN axon.Charges c ON otc.ToExternalId = c.ExternalId
AND otc.ReportNumber = c.ReportNumber
WHERE u.IncidentNumber = @IncidentNumber
```

Person

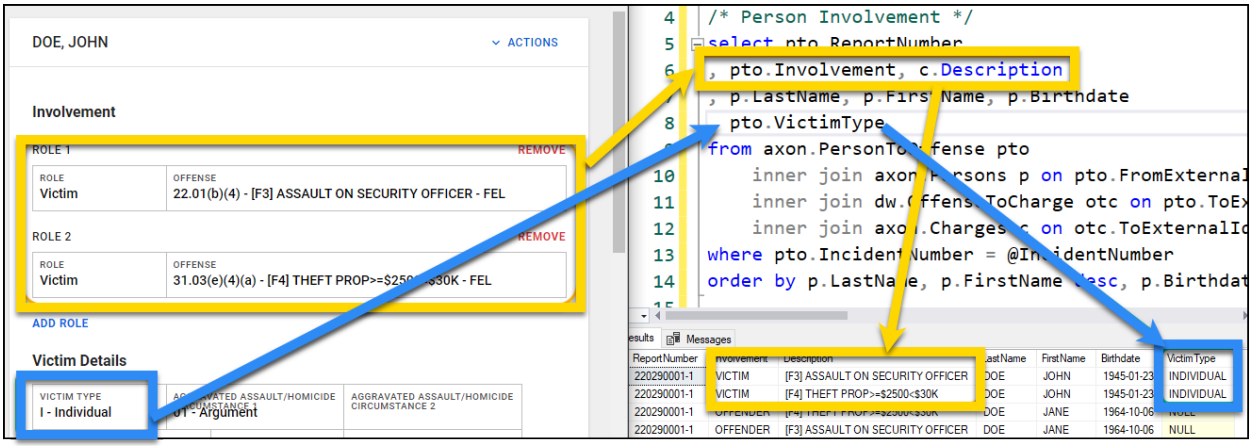
General description



SQL Select Statement

```
/* Persons */
SELECT p.IncidentNumber,
    p.ReportNumber,
    p.Type,
    p.LastName,
    p.FirstName,
    p.MiddleName,
    p.Birthdate,
    p.Age,
    p.Sex,
    p.Race,
    p.Ethnicity,
    p.ResidentStatus,
    p.HeightIn,
    p.WeightLb,
    p.EyeColor,
    p.HairColor,
    p.HairLength,
    p.HairStyle,
    p.FacialHair
FROM axon.Persons p
WHERE p.IncidentNumber = @IncidentNumber
```

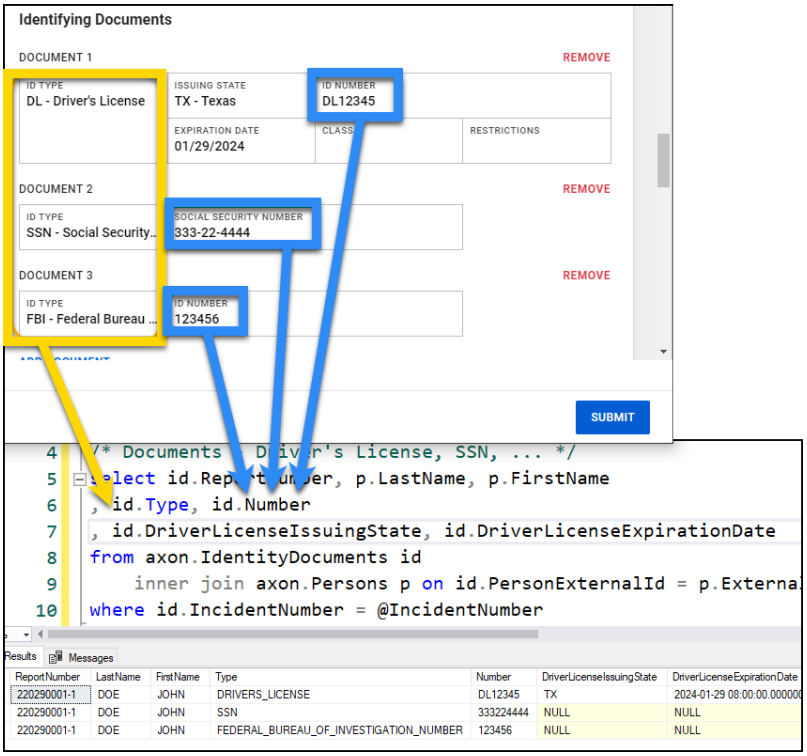
Involvement



SQL Select Statement

```
/* Person Involvement */
SELECT pto.ReportNumber,
    pto.Involvement,
    c.Description,
    p.LastName,
    p.FirstName,
    p.Birthdate,
    pto.VictimType
FROM axon.PersonToOffense pto
INNER JOIN axon.Persons p ON pto.FromExternalId = p.ExternalId
AND pto.ReportNumber = p.Report Number
INNER JOIN dw.OffenseToCharge otc ON pto.ToExternalId = otc.FromExternalId
AND pto.ReportNumber = otc.ReportNumber
INNER JOIN axon.Charges c ON otc.ToExternalId = c.ExternalId
AND otc.ReportNumber = c.ReportNumber
WHERE pto.IncidentNumber = @IncidentNumber
ORDER BY p.LastName,
    p.FirstName DESC,
    p.Birthdate
```

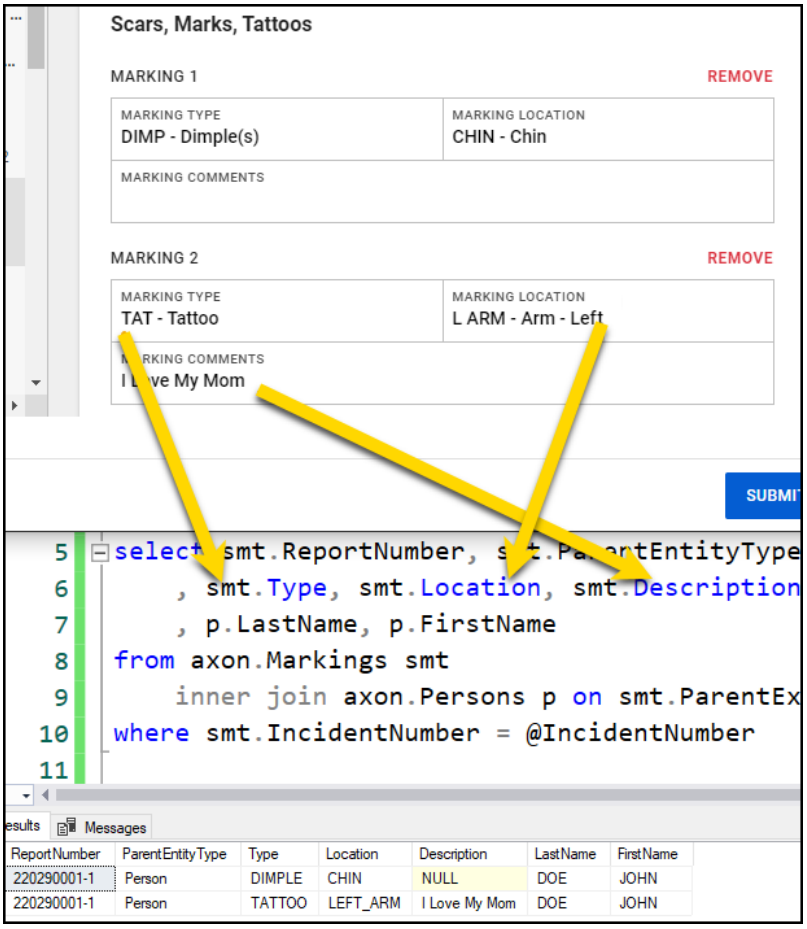
Identifying documents



SQL Select Statement

```
/* Documents - Driver's License, SSN, ... */
SELECT id.ReportNumber,
    p.LastName,
    p.FirstName,
    id.Type,
    id.Number,
    id.DriverLicenseIssuingState,
    id.DriverLicenseExpirationDate
FROM axon.IdentityDocuments id
INNER JOIN axon.Persons p ON id.PersonExternalId = p.ExternalId
AND id.ReportNumber = p.Report Number
WHERE id.IncidentNumber = @IncidentNumber
```

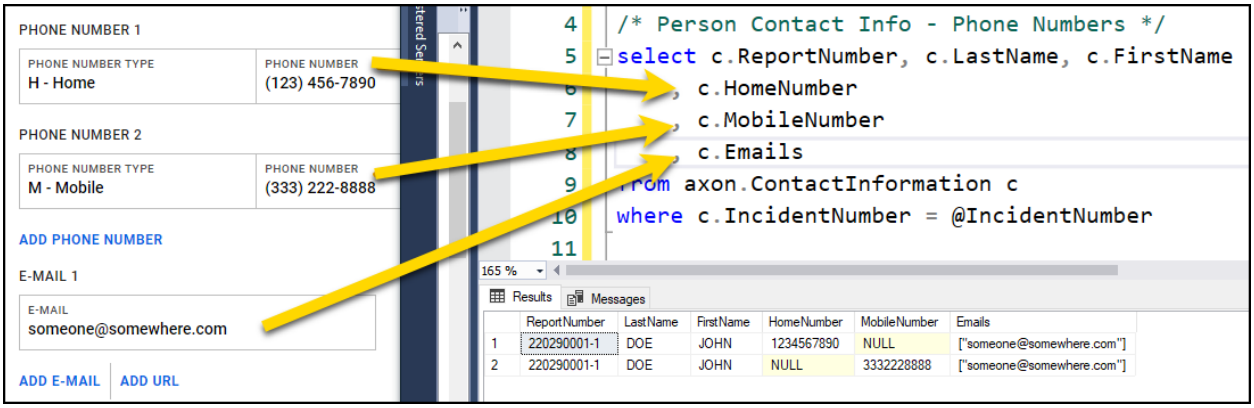
Markings



SQL Select Statement

```
/* Scars/Marks/Tattoos */
SELECT smt.ReportNumber,
    smt.ParentEntityType,
    smt.Type,
    smt.Location,
    smt.Description,
    p.LastName,
    p.FirstName
FROM axon.Markings smt
INNER JOIN axon.Persons p ON smt.ParentExternalId = p.ExternalId
AND p.ReportNumber = smt.ReportNumber
WHERE smt.IncidentNumber = @IncidentNumber
```

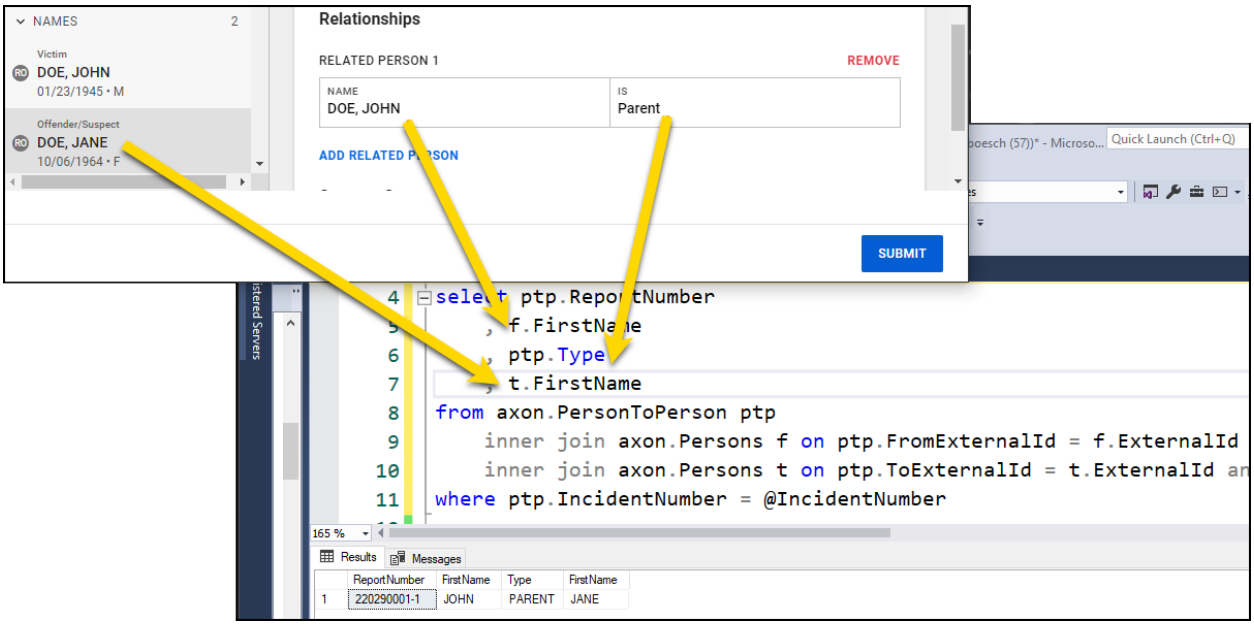
Phone numbers



SQL Select Statement

```sql
/* Person Contact Info - Phone Numbers */
SELECT c.ReportNumber,
    c.LastName,
    c.FirstName,
    c.HomeNumber,
    c.MobileNumber,
    c.Emails
FROM axon.ContactInformation c
WHERE c.IncidentNumber = @IncidentNumber
```

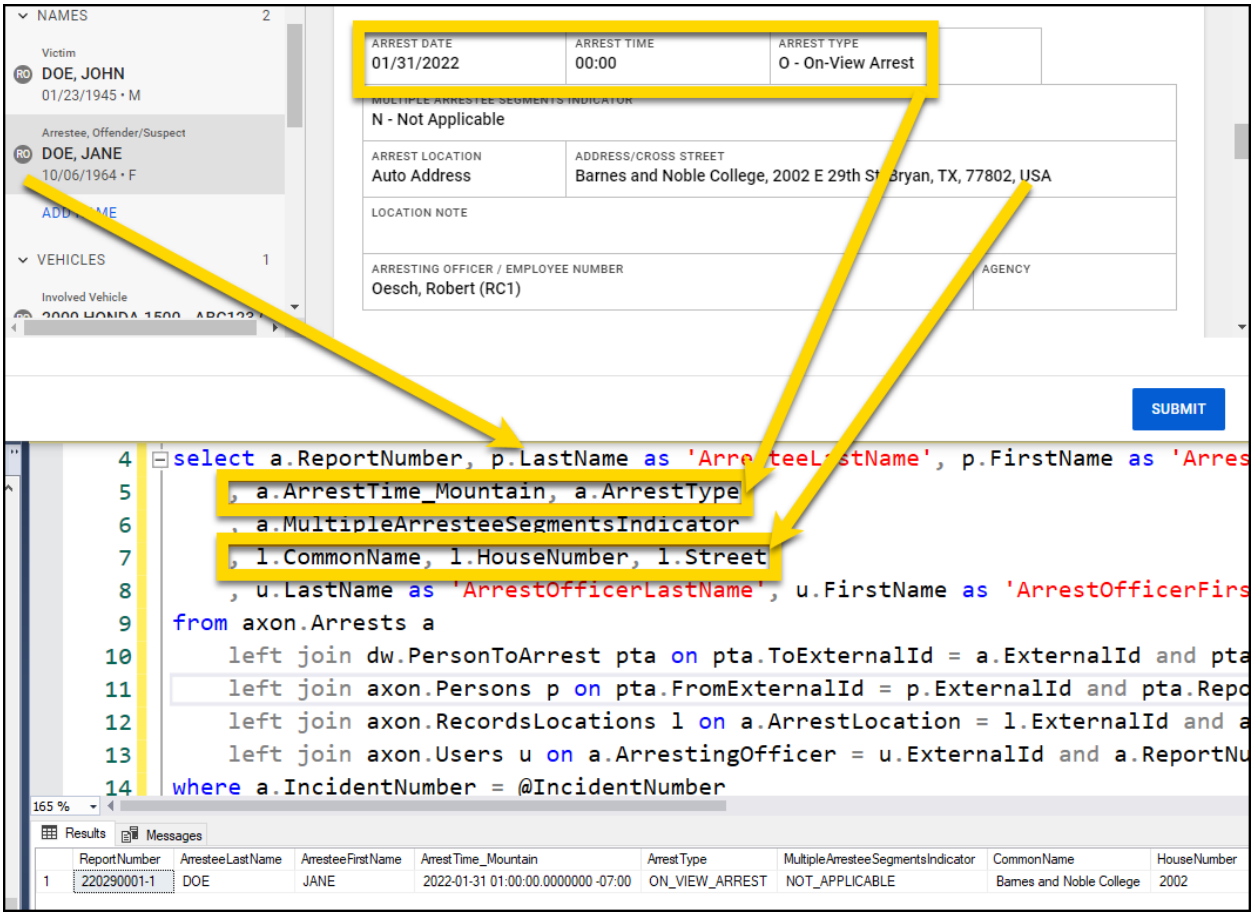
Person-to-person relationships



SQL Select Statement

```
/* Person to Person Relationships */
SELECT ptp.ReportNumber,
    f.FirstName,
    ptp.Type,
    t.FirstName
FROM axon.PersonToPerson ptp
INNER JOIN axon.Persons f ON ptp.FromExternalId = f.ExternalId
AND f.ReportNumber = ptp.Report Number
INNER JOIN axon.Persons t ON ptp.ToExternalId = t.ExternalId
AND t.ReportNumber = ptp.ReportNumber
WHERE ptp.IncidentNumber = @IncidentNumber
```

Arrest

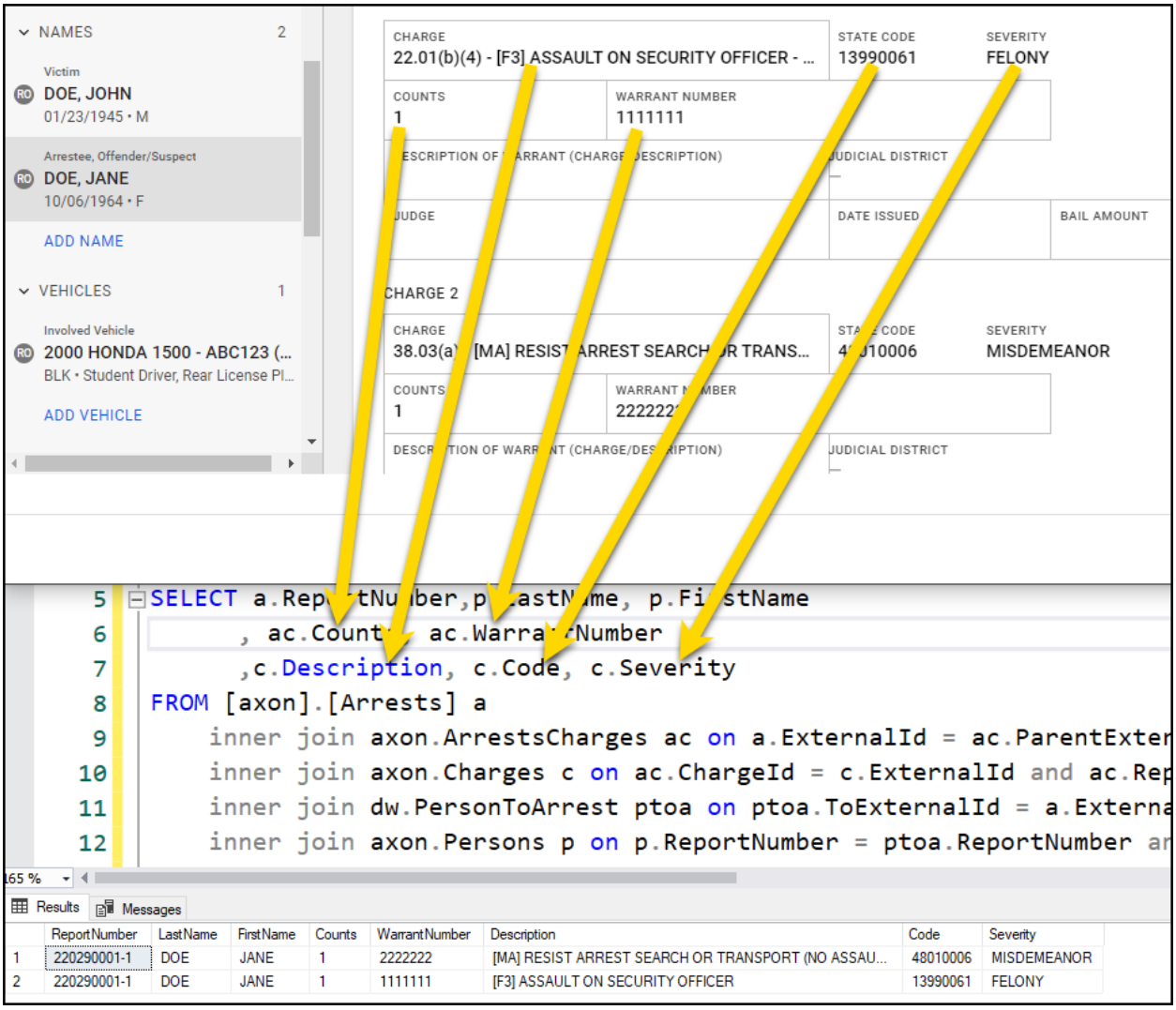


SQL Select Statement

```
/* Person - Arrest */
SELECT a.ReportNumber,
    p.LastName AS 'ArresteeLastName',
    p.FirstName AS 'ArresteeFirstName',
```

```
    a.ArrestTime_Mountain,
    a.ArrestType,
    a.MultipleArresteeSegmentsIndicator,
    l.CommonName,
    l.HouseNumber,
    l.Street,
    u.LastName AS 'ArrestOfficerLastName',
    u.FirstName AS 'ArrestOfficerFirstName',
    u.BadgeNumber AS 'ArrestOfficerBadge'
FROM axon.Arrests a
LEFT JOIN dw.PersonToArrest pta ON pta.ToExternalId = a.ExternalId
AND pta.ReportNumber = a.ReportNumber
LEFT JOIN axon.Persons p ON pta.FromExternalId = p.ExternalId
AND pta.ReportNumber = p.ReportNumber
LEFT JOIN axon.RecordsLocations l ON a.ArrestLocation = l.ExternalId
AND a.ReportNumber = l.ReportNumber
LEFT JOIN axon.Users u ON a.ArrestingOfficer = u.ExternalId
AND a.ReportNumber = u.ReportNumber
WHERE a.IncidentNumber = @IncidentNumber
```

Arrest charges



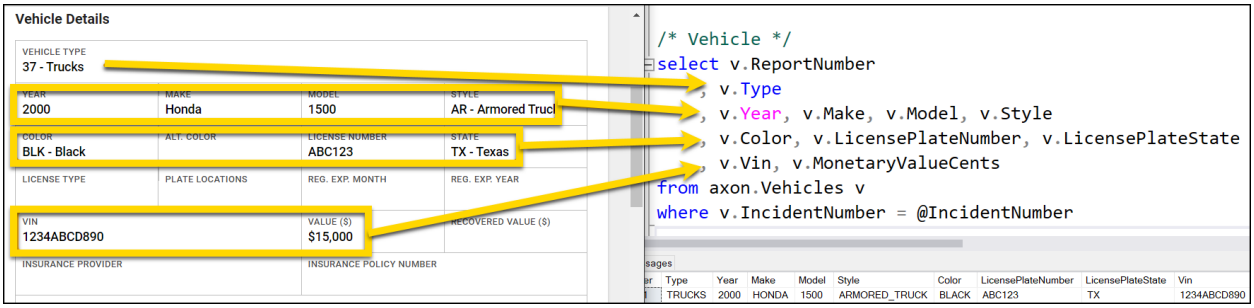
SQL Select Statement

```
/* Person - Arrest Charge */
SELECT a.ReportNumber,
    p.LastName,
    p.FirstName,
    ac.Counts,
    ac.WarrantNumber,
    c.Description,
    c.Code,
    c.Severity
FROM [axon].[Arrests] a
INNER JOIN axon.ArrestsCharges ac ON a.ExternalId = ac.ParentExternalId
AND a.ReportNumber = a c.ReportNumber
INNER JOIN axon.Charges c ON ac.ChargeId = c.ExternalId
AND ac.ReportNumber = c.ReportNumber
INNER JOIN dw.PersonToArrest ptoa ON ptoa.ToExternalId = a.ExternalId
AND ptoa.ReportNumber = a.ReportNumber
```

```
INNER JOIN axon.Persons p ON p.ReportNumber = ptoa.ReportNumber
AND ptoa.FromExternalId = p.ExternalId
WHERE a.IncidentNumber = @IncidentNumber
```

Vehicle

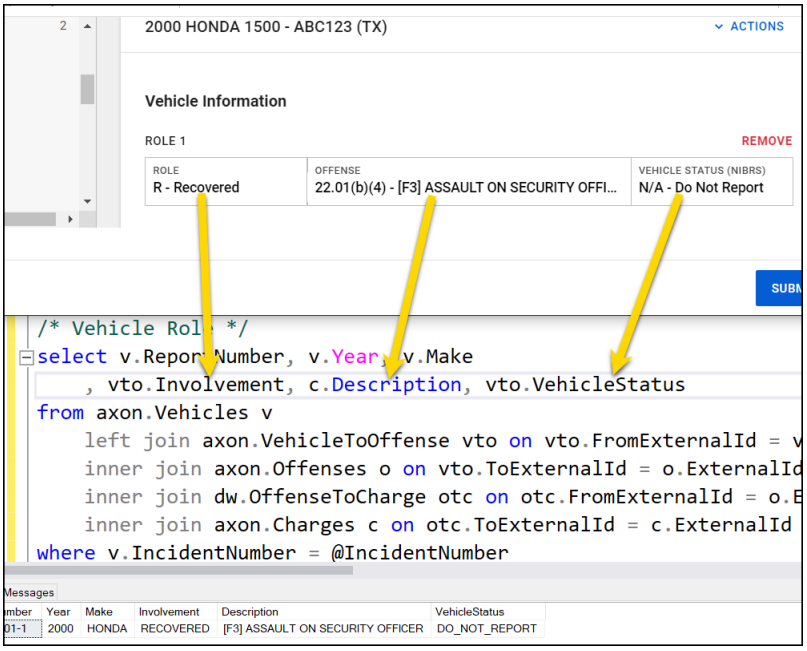
Details



SQL Select Statement

```
/* Vehicle */
SELECT v.ReportNumber,
    v.Type,
    v.Year,
    v.Make,
    v.Model,
    v.Style,
    v.Color,
    v.LicensePlateNumber,
    v.LicensePlateState,
    v.Vin,
    v.MonetaryValueCents
FROM axon.Vehicles v
WHERE v.IncidentNumber = @IncidentNumber
```

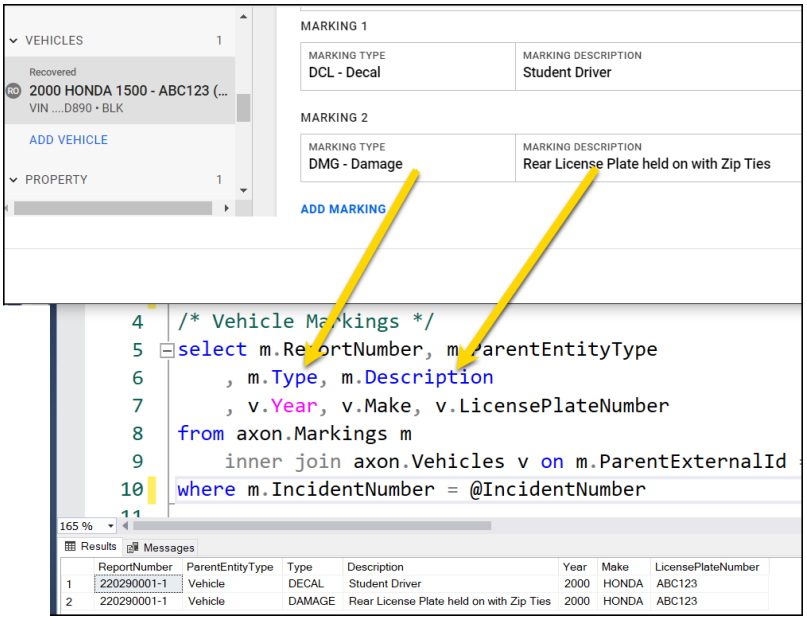
Role



SQL Select Statement

```sql
/* Vehicle Role */
SELECT v.ReportNumber,
    v.Year,
    v.Make,
    vto.Involvement,
    c.Description,
    vto.VehicleStatus
FROM axon.Vehicles v
LEFT JOIN axon.VehicleToOffense vto ON vto.FromExternalId = v.ExternalId
AND vto.ReportNumber = v.ReportNumber
INNER JOIN axon.Offenses o ON vto.ToExternalId = o.ExternalId
AND vto.ReportNumber = o.ReportNumber
INNER JOIN dw.OffenseToCharge otc ON otc.FromExternalId = o.ExternalId
AND otc.ReportNumber = o.ReportNumber
INNER JOIN axon.Charges c ON otc.ToExternalId = c.ExternalId
AND otc.ReportNumber = c.ReportNumber
WHERE v.IncidentNumber = @IncidentNumber
```

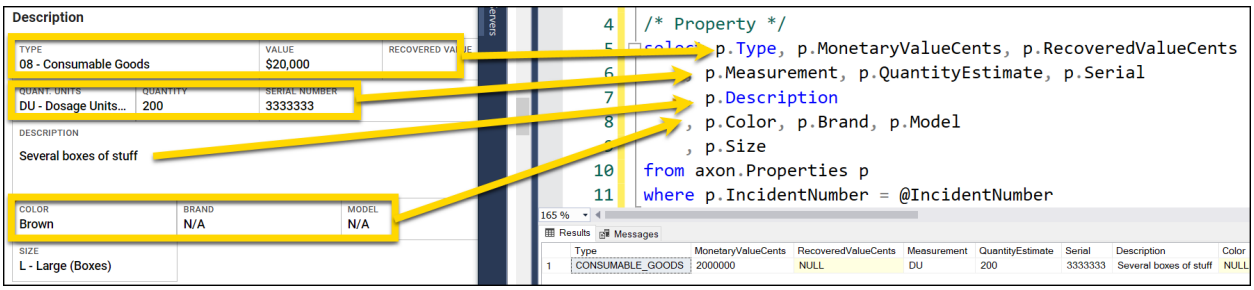
Markings



SQL Select Statementc

```sql
/* Vehicle Markings */
SELECT m.ReportNumber,
    m.ParentEntityType,
    m.Type,
    m.Description,
    v.Year,
    v.Make,
    v.LicensePlateNumber
FROM axon.Markings m
INNER JOIN axon.Vehicles v ON m.ParentExternalId = v.ExternalId
AND v.ReportNumber = m.ReportNumber
WHERE m.IncidentNumber = @IncidentNumber
```

Property



SQL Select Statement

```
/* Property */
SELECT p.Type,
    p.MonetaryValueCents,
    p.RecoveredValueCents
```

SQL select statements: Full list

```
DECLARE @IncidentNumber VARCHAR(20)
SET @IncidentNumber = '22033001' /* Incident Overview */
SELECT inc.IncidentNumber,
    rep.ReportNumber,
    rep.ReportType,
    inc.OccurredFromDate,
    inc.OccurredFromDate_Mountain,
    loc.Verified,
    loc.CommonName,
    loc.HouseNumber,
    loc.Street,
    loc.PostalCode,
    loc.District,
    loc.Sector,
    rep.IncidentLocationNote,
    JSON_VALUE(dwF.RawData, '$.incidentLocation.type') AS 'LocationCategory'
FROM axon.Incidents inc
INNER JOIN axon.Reports rep ON inc.IncidentNumber = rep.IncidentNumber
LEFT JOIN axon.RecordsLocations loc ON inc.LocationExternalId = loc.ExternalId
AND inc.Inciden tNumber = loc.IncidentNumber
LEFT JOIN dw.Forms dwF ON loc.ReportNumber = dwF.ReportNumber
WHERE rep.IncidentNumber = @IncidentNumber

/* Case Factors */
SELECT f.IncidentNumber,
    f.ReportNumber,
    cf.value AS 'CaseFactor'
FROM dw.Forms f CROSS APPLY OPENJSON(f.RawData, '$.caseFactors.caseFactors') cf
WHERE f.IncidentNumber = @IncidentNumber

/* Offenses */
SELECT o.IncidentNumber,
    o.ReportNumber,
    c.Description,
    o.NibrsUcrCode,
    c.Code,
    o.Severity,
    o.NibrsUcrCode,
    c.Ucr,
    o.Completion,
    o.OffenderSuspectedOfUsing,
    o.CriminalGangActivityInvolved,
    o.WeaponForceInvolved,
    o.IsBiasMotivationInvolved,
    o.CargoTheft,
JSON_VALUE(dwo.RawData, '$.axon.numberOfPremisesEntered') AS 'NumberOfPremisesEntered'
FROM axon.Offenses o
```

```
INNER JOIN dw.OffenseToCharge otc ON o.ExternalId = otc.FromExternalId
AND otc.ReportNumber = o.ReportNumber
INNER JOIN axon.Charges c ON otc.ToExternalId = c.ExternalId
AND otc.ReportNumber = c.ReportNu mber
INNER JOIN dw.Offenses dwo ON o.ExternalId = dwo.ExternalId
AND o.ReportNumber = dwo.ReportNum ber WHERE o.IncidentNumber = @IncidentNumber

/* Suspected of Using */
SELECT u.IncidentNumber,
    u.ReportNumber,
    c.Description,
    u.OffenderSuspectedOfUsing
FROM axon.OffenderSuspectedOfUsing u
INNER JOIN dw.OffenseToCharge otc ON u.OffenseExternalId = otc.FromExternalId
AND otc.ReportNu mber = u.ReportNumber
INNER JOIN axon.Charges c ON otc.ToExternalId = c.ExternalId
AND otc.ReportNumber = c.ReportNu mber WHERE u.IncidentNumber = @IncidentNumber
/* Persons */
SELECT p.IncidentNumber,
    p.ReportNumber,
    p.Type,
    p.LastName,
    p.FirstName,
    p.MiddleName,
    p.Birthdate,
    p.Age,
    p.Sex,
    p.Race,
    p.Ethnicity,
    p.ResidentStatus,
    p.HeightIn,
    p.WeightLb,
    p.EyeColor,
    p.HairColor,
    p.HairLength,
    p.HairStyle,
    p.FacialHair
FROM axon.Persons p WHERE p.IncidentNumber = @IncidentNumber

/* Person Involvement */
SELECT pto.ReportNumber,
    pto.Involvement,
    c.Description,
    p.LastName,
    p.FirstName,
    p.Birthdate,
    pto.VictimType
FROM axon.PersonToOffense pto
INNER JOIN axon.Persons p ON pto.FromExternalId = p.ExternalId
AND pto.ReportNumber = p.Report Number
INNER JOIN dw.OffenseToCharge otc ON pto.ToExternalId = otc.FromExternalId
AND pto.ReportNumbe r = otc.ReportNumber
INNER JOIN axon.Charges c ON otc.ToExternalId = c.ExternalId
AND otc.ReportNumber = c.ReportNumber WHERE pto.IncidentNumber = @IncidentNumber
ORDER BY p.LastName,
    p.FirstName DESC,
    p.Birthdate /* Documents - Driver's License, SSN, ... */
SELECT id.ReportNumber,
    p.LastName,
    p.FirstName,
    id.Type,
```

```
    id.Number,
    id.DriverLicenseIssuingState,
    id.DriverLicenseExpirationDate
FROM axon.IdentityDocuments id
INNER JOIN axon.Persons p ON id.PersonExternalId = p.ExternalId
AND id.ReportNumber = p.Report Number
WHERE id.IncidentNumber = @IncidentNumber

/* Scars/Marks/Tattoos */
SELECT smt.ReportNumber,
    smt.ParentEntityType,
    smt.Type,
    smt.Location,
    smt.Description,
    p.LastName,
    p.FirstName
FROM axon.Markings smt
INNER JOIN axon.Persons p ON smt.ParentExternalId = p.ExternalId
AND p.ReportNumber = smt.ReportNumber WHERE smt.IncidentNumber = @IncidentNumber

/* Person Contact Info - Phone Numbers */
SELECT c.ReportNumber,
    c.LastName,
    c.FirstName,
    c.HomeNumber,
    c.MobileNumber,
    c.Emails
FROM axon.ContactInformation c WHERE c.IncidentNumber = @IncidentNumber

/* Person to Person Relationships */
SELECT ptp.ReportNumber,
    f.FirstName,
    ptp.Type,
    t.FirstName
FROM axon.PersonToPerson ptp
INNER JOIN axon.Persons f ON ptp.FromExternalId = f.ExternalId
AND f.ReportNumber = ptp.Report Number
INNER JOIN axon.Persons t ON ptp.ToExternalId = t.ExternalId
AND t.ReportNumber = ptp.ReportNumber WHERE ptp.IncidentNumber = @IncidentNumber

/* Person - Arrest */
SELECT a.ReportNumber,
    p.LastName AS 'ArresteeLastName',
    p.FirstName AS 'ArresteeFirstName',
    a.ArrestTime_Mountain,
    a.ArrestType,
    a.MultipleArresteeSegmentsIndicator,
    l.CommonName,
    l.HouseNumber,
    l.Street,
    u.LastName AS 'ArrestOfficerLastName',
    u.FirstName AS 'ArrestOfficerFirstName',
    u.BadgeNumber AS 'ArrestOfficerBadge'
FROM axon.Arrests a
LEFT JOIN dw.PersonToArrest pta ON pta.ToExternalId = a.ExternalId
AND pta.ReportNumber = a.ReportNumber
LEFT JOIN axon.Persons p ON pta.FromExternalId = p.ExternalId
AND pta.ReportNumber = p.ReportNumber
LEFT JOIN axon.RecordsLocations l ON a.ArrestLocation = l.ExternalId
AND a.ReportNumber = l.ReportNumber
LEFT JOIN axon.Users u ON a.ArrestingOfficer = u.ExternalId
```

```
AND a.ReportNumber = u.ReportNumber WHERE a.IncidentNumber = @IncidentNumber

/* Person - Arrest Charge */
SELECT a.ReportNumber,
    p.LastName,
    p.FirstName,
    ac.Counts,
    ac.WarrantNumber,
    c.Description,
    c.Code,
    c.Severity
FROM [axon].[Arrests] a
INNER JOIN axon.ArrestsCharges ac ON a.ExternalId = ac.ParentExternalId
AND a.ReportNumber = a c.ReportNumber
INNER JOIN axon.Charges c ON ac.ChargeId = c.ExternalId
AND ac.ReportNumber = c.ReportNumber
INNER JOIN dw.PersonToArrest ptoa ON ptoa.ToExternalId = a.ExternalId
AND ptoa.ReportNumber = a.ReportNumber
INNER JOIN axon.Persons p ON p.ReportNumber = ptoa.ReportNumber
AND ptoa.FromExternalId = p.ExternalId WHERE a.IncidentNumber = @IncidentNumber

/* Vehicle */
    SELECT v.ReportNumber,
    v.Type,
    v.Year,
    v.Make,
    v.Model,
    v.Style,
    v.Color,
    v.LicensePlateNumber,
    v.LicensePlateState,
    v.Vin,
    v.MonetaryValueCents
FROM axon.Vehicles v WHERE v.IncidentNumber = @IncidentNumber

/* Vehicle Role */
SELECT v.ReportNumber,
    v.Year,
    v.Make,
    vto.Involvement,
    c.Description,
    vto.VehicleStatus
FROM axon.Vehicles v
LEFT JOIN axon.VehicleToOffense vto ON vto.FromExternalId = v.ExternalId
AND vto.ReportNumber = v.ReportNumber
INNER JOIN axon.Offenses o ON vto.ToExternalId = o.ExternalId
AND vto.ReportNumber = o.ReportNumber
INNER JOIN dw.OffenseToCharge otc ON otc.FromExternalId = o.ExternalId
AND otc.ReportNumber = o.ReportNumber
INNER JOIN axon.Charges c ON otc.ToExternalId = c.ExternalId
AND otc.ReportNumber = c.ReportNumber WHERE v.IncidentNumber = @IncidentNumber

/* Vehicle Markings */
SELECT m.ReportNumber,
    m.ParentEntityType,
    m.Type,
    m.Description,
    v.Year,
    v.Make,
    v.LicensePlateNumber
FROM axon.Markings m
```

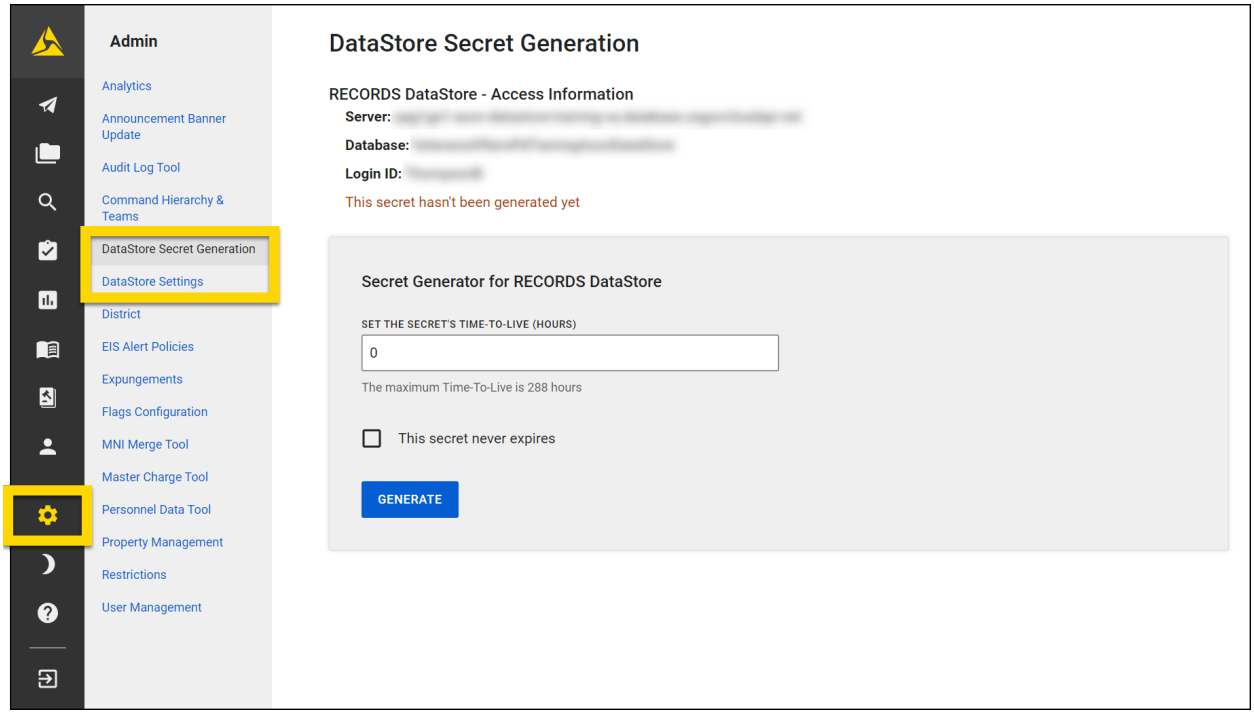
```
INNER JOIN axon.Vehicles v ON m.ParentExternalId = v.ExternalId
AND v.ReportNumber = m.ReportNumber WHERE m.IncidentNumber = @IncidentNumber

/* Property */
SELECT p.Type,
    p.MonetaryValueCents,
    p.RecoveredValueCents
```

Access control

The DataStore tools in the Administrator Console let administrators with the appropriate privileges manage which users can access the DataStore. Using these tools, administrators can:

- DataStore Settings: Lets users manage DataStore configurations and user credentials
- DataStore Secret Generation: Lets users create secrets that allow direct access to the DataStore



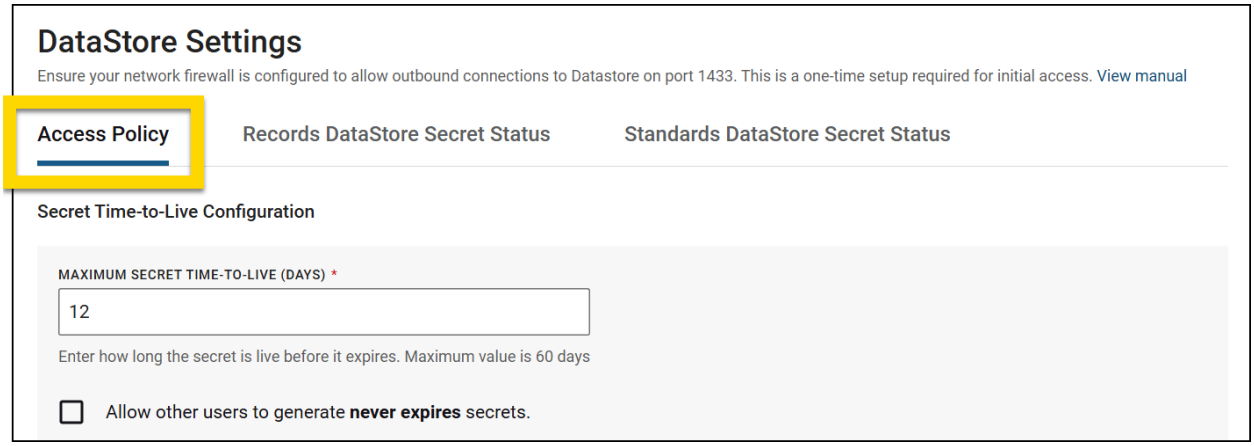
DataStore settings

Users who belong to Groups or teams with the DataStore Management privilege can view the DataStore Settings tool. Using this tool, users can manage the organization's access policy and secret statuses for Axon Records and Axon Standards users.

Access policy

From the **Access Policy** tab in the DataStore Settings tool, you can adjust the following settings:

- Secret time-to-live configuration
- Allowed IP addresses for DataStore access



Secret time-to-live configuration

Enter a number in the **Maximum Secret Time-To-Live (Days)** field to indicate the maximum number of days a secret can be available after it is generated.

Note that during secret generation, you can enter a TTL in hours, which allows for more granular control than the days setting in the DataStore Settings tool. However, the number of hours you enter during secret generation must be equal to or fewer than the day duration specified in this field in the DataStore Settings tool.

Select the **Allow other users to generate never expires secrets** checkbox to allow users to generate secrets that never expire.

After making changes, select **Save Settings**.

Allowed IP addresses for DataStore access

The IP addresses listed in this table are the only addresses from which the DataStore can be accessed.

To add a new IP address:

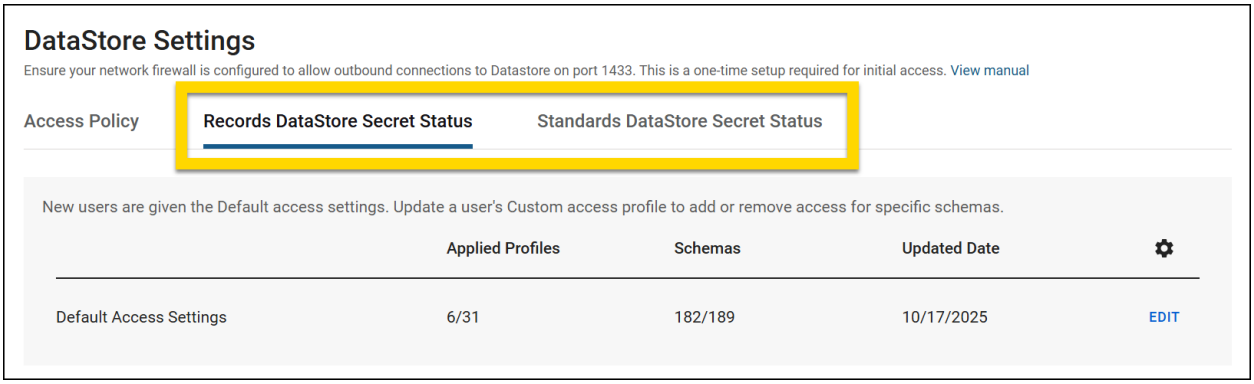
1. Select **Add IP**.
2. Select the type of address:
   - Single IP address: Lets you add one address
   - Range of IP address: Lets you add an IP address range. You must provide a starting IP address and an Ending IP address.
3. Enter either the single IP address or the range.
4. Select **Add**.
5. Select **Save Settings**.

To remove an IP address (either a single IP address or a range):

1. Select **Remove** in the row corresponding to the IP address you want to remove.
2. Select **Remove** again in the confirmation window that appears.
3. Select **Save Settings**.

Secret status

The **Secret Status** tabs in the DataStore Settings tool display a list of all users who have been given access to the DataStore. If your organization is configured for both Axon Records and Axon Standards, you will see two **Secret Status** tabs, one for Axon Records and one for Axon Standards.



These tabs are split into two sections:

- Default access settings
- Access profile list

Default access settings

The top section on the **Secret Status** tabs displays the settings for your organization's default access profile. The default access profile is applied whenever you give a new user access to the DataStore and gives users access to a predefined set of accessible views and schemas.

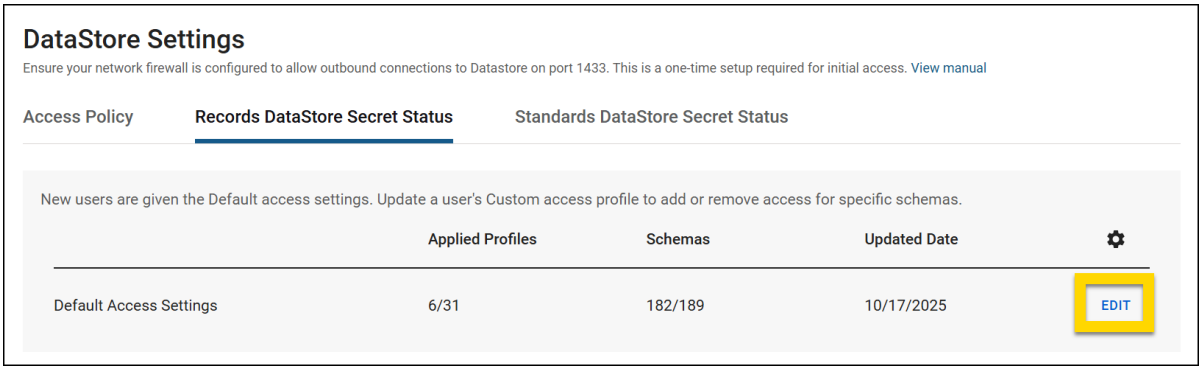
The Default Access Settings section displays the following information:

- **Applied Profiles:** The number of users who have been given default access
- **Schemas:** The number of tables/views out of the total number of tables/views in the entire DataStore that are included in the default access profile.
  - For example, if this section displays `182/189`, there are 189 total tables/views in the DataStore but the default access profile only includes 182.
- **Updated Date:** The last date when the default access settings were updated.

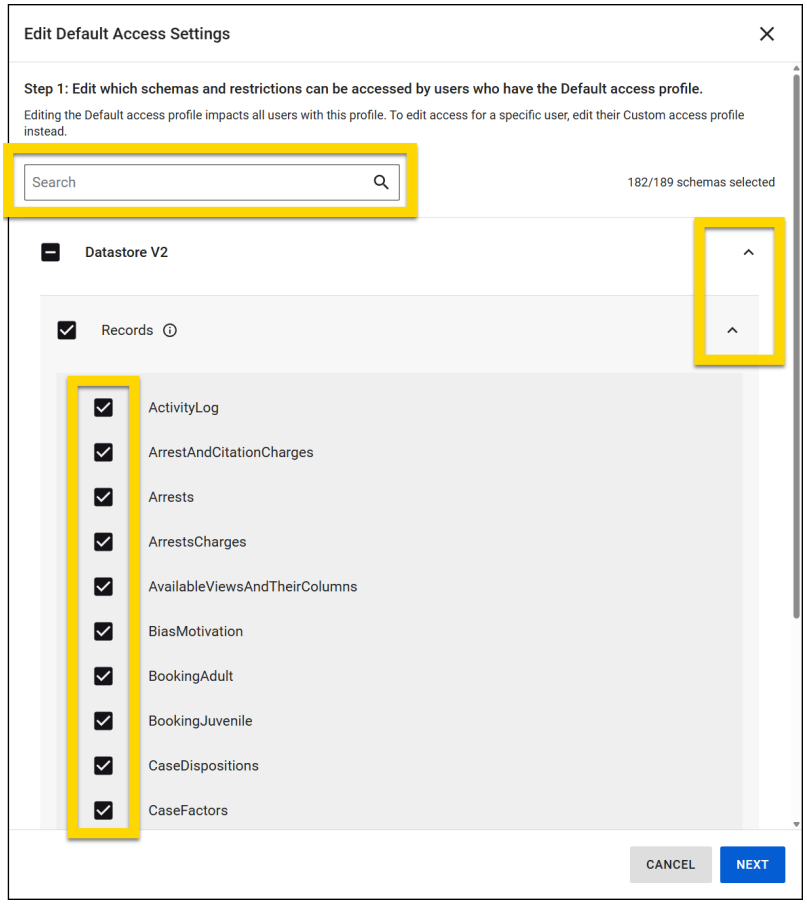
When you edit your DataStore's default access settings, all users who have been given default access will receive the access updates you make. If you only want to adjust access settings for a single user, edit their custom access profile instead.

To edit the default access settings:

1. Select **Edit** in the Default Access Settings section at the top of the **Secret Status** tab.



2. Use the checkboxes to set which schemas are included in the default access profile.
   - Use the search bar to find specific tables or views.
   - Hover over the information icon to view more details about that schema. See Concepts and features to learn more about schemas.
   - Select the down arrows to reveal the nested schema levels.



3. After adjusting the schemas, select **Next**.
4. Use the checkboxes to set which users have the default access profile.
   - All users who have previously been given the default access profile are pre-selected.
   - Use the search box to find specific users.
5. After adjusting the user list, select **Save**.
   - It may take several minutes for your updates to save. **Do not close the tab during this process.**

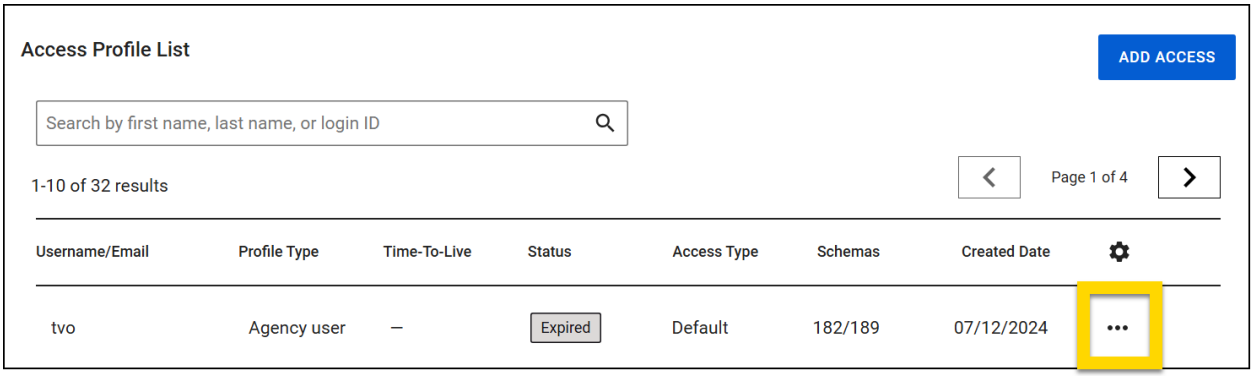
Access profile list

The table in the Access Profile List section provides detailed information about all user profiles that have access to DataStore, along with their current access status. The following information is included about each profile:

- **Username/Email:** The username or email associated with the profile
- **Profile Type:**
  - Agency user: Users who have an Axon Evidence account with the agency and for whom a DataStore secret has been generated.
  - Third party: Users who belong to third-party organizations but can still access the organization's DataStore.
  - Axon: Axon representatives who have been granted access to the organization's DataStore.
- **Time-To-Live:** Shows the TTL that was specified when the profile's secret was generated
- **Status:**
  - Setup Pending: A secret has NOT yet been generated.
  - Active: A secret has been generated and is currently active.
  - Expired: A secret was generated, but the TTL has passed and the secret has expired.
- **Access Type:** Whether the profile has default or custom access
- **Schemas:** The number of tables/views out of the total number of tables/views in the entire DataStore that are included in the default access profile.
  - For example, if this section displays `182/189`, there are 189 total tables/views in the DataStore but the default access profile only includes 182.
  - If a profile has default access, this schema count is the same as the count in the Default Access Settings section.
- **Created Date:** The date the access profile was created

Depending on the profile type and status, various options appear in the **More Actions [...]** menu:

- Edit access control: Appears for all profiles and statuses
- Regenerate secret: Appears for third-party profiles that are in Setup Pending or Active status
- Revoke secret: Appears for all profiles that are in Active status
- Remove access profile: Appears for all Axon profiles in any status

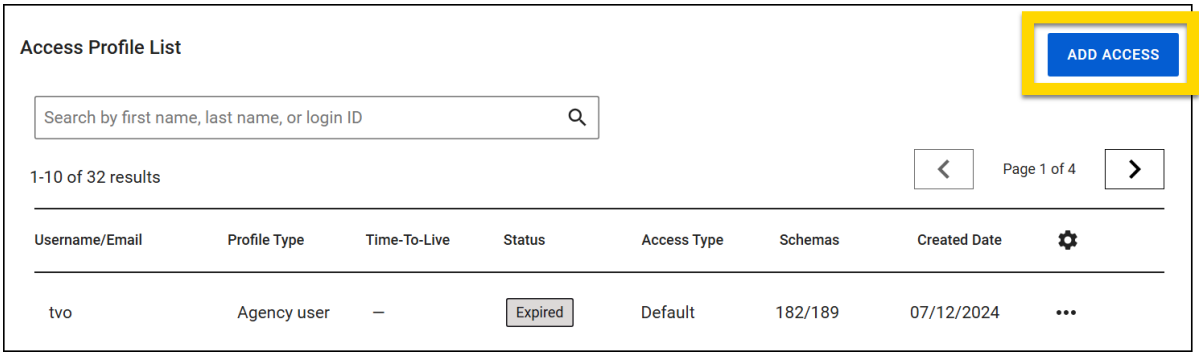


Create access profile

To grant DataStore access to an agency user, first generate a secret using the DataStore Secret Generation tool. Once the secret has generated, an access profile appears in the Access Profile List where you can edit the access control if you want the profile to have more or less access than is granted by the default access profile.

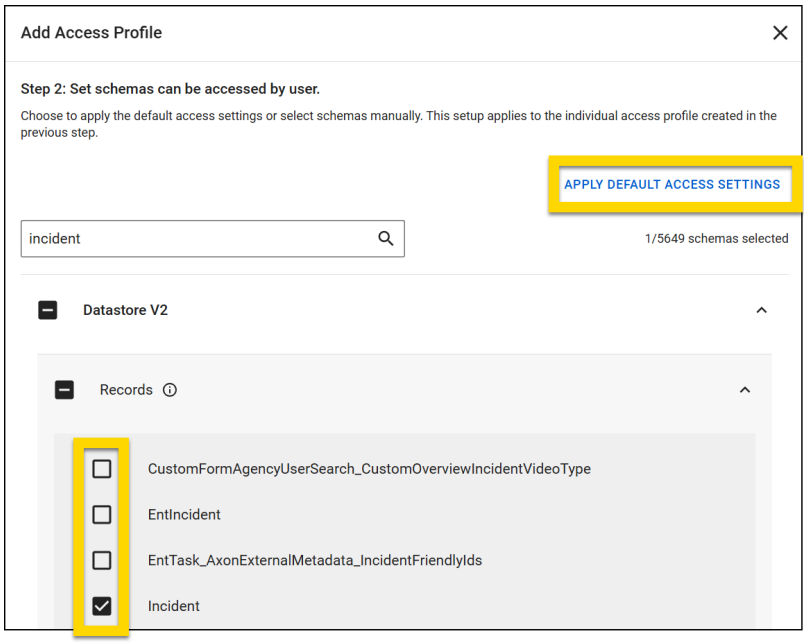
To create a new access profile for an Axon representative or a third-party user:

1. Select **Add Access**.



2. Specify if the user is an Axon representative or a third-party user.
   - If the user is an Axon representative, enter their email address.
   - If the user is a third party, enter a username.

3. Select the checkbox to acknowledge that you are a designated administrator responsible for granting DataStore access to users outside your organization.
4. Select **Next**.
5. Set which schemas the profile can access.
   - To grant default access, select **Apply Default Access Settings**.
   - To grant custom access, use the checkboxes to select specific schemas.
     - Tip: To use the default profile as a starting point, select **Apply Default Access Settings** then adjust the checkboxes to refine the schemas included in the profile's access.



6. Select **Add Access Profile** to create the profile.
7. When you create a profile for a third-party user, you can immediately generate a secret for the profile.
   a. Select a TTL option.
      - Options include days, hours, or never expires.
      - The maximum number is controlled by the maximum TTL setting on the **Access Policy** tab.
   b. Select **Generate**.
   c. The secret will generate and appear below the **Generate** button. **This secret must be copied immediately, as it will not be displayed again.**
   d. Select **Copy Secret and Close**.
   e. Follow your organization's security practices to safely share the secret with the user.

8. When you create a profile for an Axon representative, they receive an email directing them to log into the internal Axon secured administration portal to generate a secret.
    - Axon representatives can never generate “never expire” secrets. Any secrets they generate will adhere to the maximum TTL setting on the **Access Policy** tab.

Edit access control

You can edit a profile’s access settings at any time. Changing access does not affect the user’s existing secret; it only modifies their ability to SELECT tables or views within a schema. When new access is granted, the user will be able to see the additional tables or views shortly after the update. Conversely, if a user’s access to specific tables or views is revoked, their access to those resources will be removed immediately. All access changes—both granting and revoking—take effect within approximately 30 seconds.

To edit access:

1. Use the search bar to find a profile.
2. Select **More Actions [...] > Edit Access Control**.
3. Use the checkboxes to set which schemas are included in the profile.
    - If a profile previously had the default access profile and you add or remove access to any schemas, their access type changes to Custom.
    - To return a user to the default access profile, select **Apply Default Access Settings**.
4. After making all necessary adjustments, select **Save**.

Regenerate secret

Regenerating a secret lets you refresh secrets for third-party users outside your organization. Agency users and Axon representatives who have access to the DataStore can generate secrets for themselves as necessary.

To regenerate a secret for a third-party user:

1. Use the search bar to find a profile.
2. Select **More Actions [...] > Regenerate Secret**.
3. Select the check box to acknowledge that you are a designated administrator responsible for granting DataStore access to users outside your organization.
4. Select a TTL option.
    - Options include days, hours, or never expires.
    - The maximum number is controlled by the settings on the **Access Policy** tab.
5. Select **Generate**.

6. The secret will generate and appear below the **Generate** button. **This secret must be copied immediately, as it will not be displayed again.**
7. Select **Copy Secret and Close**.
8. Follow your organization's security practices to safely share the secret with the user.

Revoke secret

Once a secret is revoked, it immediately changes to an Expired status and can no longer be used to access the DataStore. If the user is currently accessing the DataStore when this occurs, they will no longer be able to perform any queries and will be logged out when their session times out.

Secrets for Axon representatives can't be revoked. Instead, the access profile must be removed.

To revoke a secret:

1. Use the search bar to find a profile.
2. Select **More Actions [...] > Revoke**.
3. Select **Revoke** in the confirmation window that appears.

Remove access profile

Access profiles for Axon representatives can be removed at any time. Once an access profile is removed, it cannot be restored.

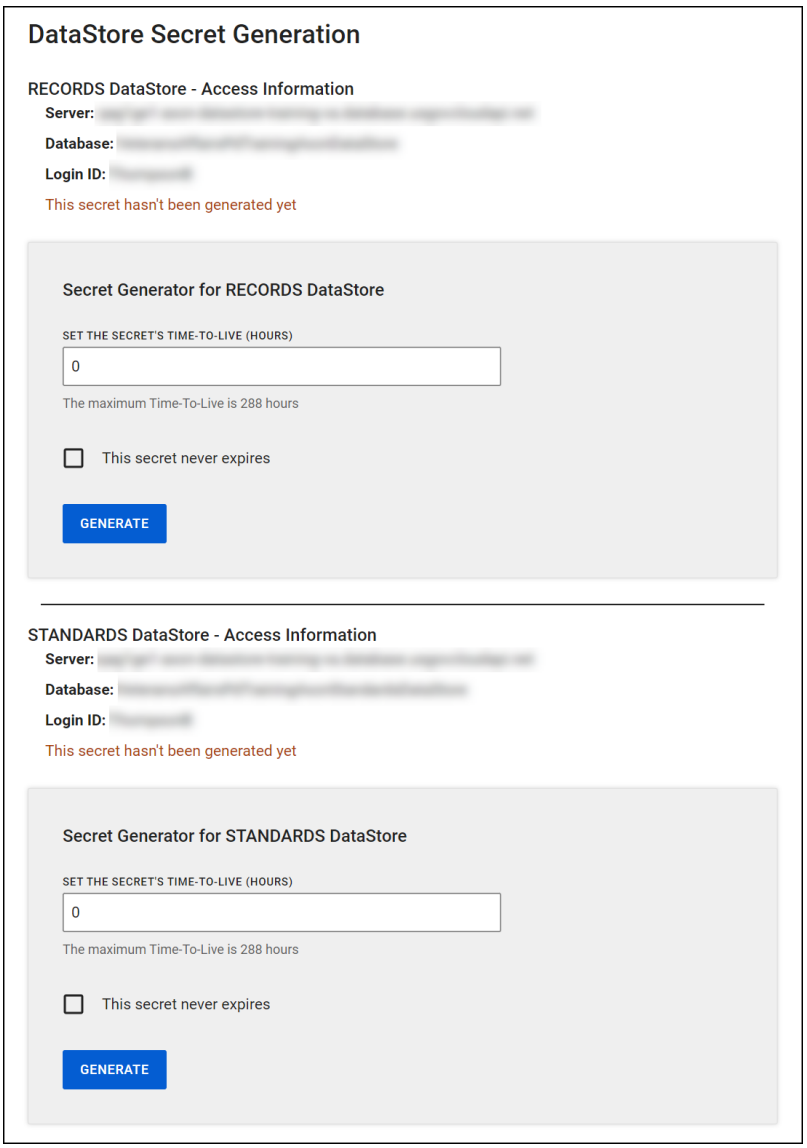
To remove an access profile:

1. Use the search bar to find a profile.
2. Select **More Actions [...] > Remove access profile**.
3. Select **Remove** in the confirmation window that appears.

Secret generation

Users who belong to Groups or teams with the DataStore Access privilege can view the DataStore Secret Generation tool. Using this tool, users can generate secrets that allow access to the DataStore. If your organization is configured for both Axon Records and

Axon Standards, you will see sections on this page: one for generating Axon Records secrets and one for Axon Standards.



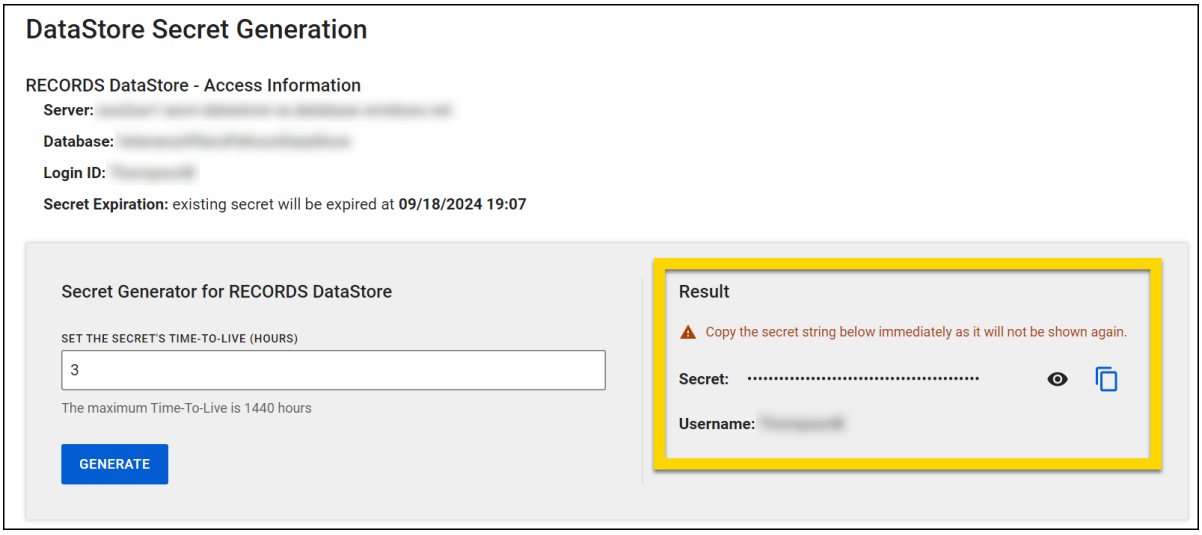
Each section provides the following information:

- Server name
- Database name
- Login ID: The username of the user who is viewing and using the tool

To generate a new secret:

1. Go to either the Axon Records or Axon Standards section and enter the time-to-live (TTL) of the secret in hours.
   - You can't enter a longer TTL than the maximum set by administrators.
   - To generate a secret that never expires, select the **This secret never expires** checkbox.

2. Select **Generate**.
3. The secret will generate and appear on the right side of the gray box. **This secret must be copied immediately, as it will not be displayed again.**
4. The secret will be added to the DataStore Settings tool where users with the appropriate privileges can revoke it if necessary.



Privileges

The privileges related to the Axon Records DataStore appear in the **DataStore - Records** category, and the privileges related to the Axon Standards DataStore appear in the **DataStore - Standards** category, as shown below:

Name	Description
DataStore - Records	
Manage the Records DataStore using the DataStore Settings tool	Lets users access the DataStore Settings tool in the Administrator Console and manage configurations for the Axon Records DataStore.
Use the Records DataStore Secret Generation tool to create DataStore secrets	Lets users access the DataStore Secret Generation tool in the Administrator Console and generate secrets for the Axon Records DataStore.
DataStore - Standards	
Manage the Standards DataStore using the DataStore Settings tool	Lets users access the DataStore Settings tool in the Administrator Console and manage configurations for the Axon Standards DataStore.

Name	Description
Use the Standards DataStore Secret Generation tool to create DataStore secrets	Lets users access the DataStore Secret Generation tool in the Administrator Console and generate secrets for the Axon Standards DataStore.

ODBC server

To connect to the Axon DataStore, you can create a Microsoft Open Database Connectivity (ODBC) server. ODBC is a C programming language interface that lets applications access data from a variety of database management systems (DBMSs). ODBC is a low-level, high-performance interface designed specifically for relational datastores.

After creating an ODBC server, you can link it to Microsoft Excel or Microsoft Access to view the data.

Before creating an ODBC server, contact your Axon representative to receive the following information:

- Server name
- Database name
- Login information (username and password)

Once you have the above information, take these steps:

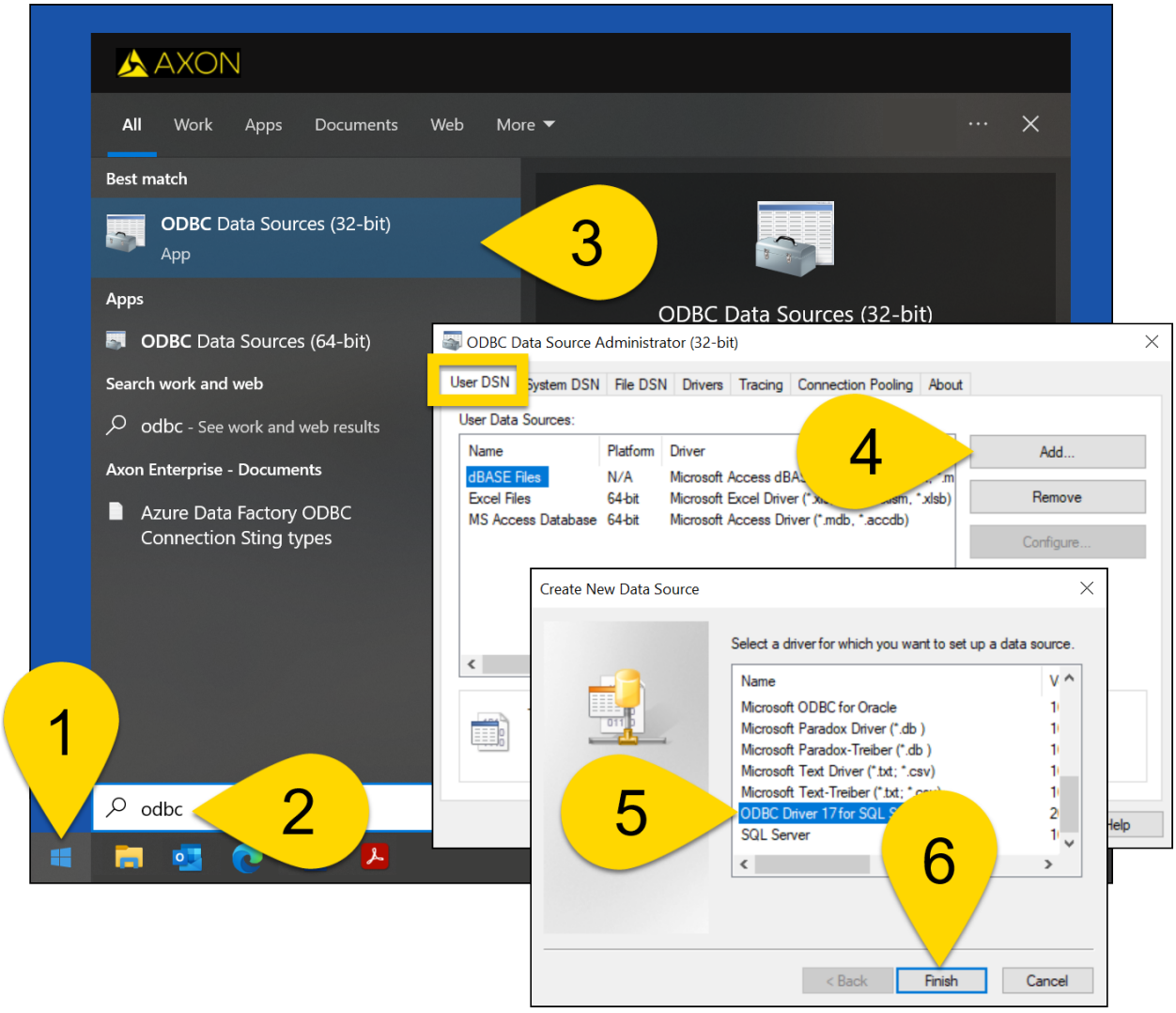
1. Create a data source
2. Connect to SQL server
3. Test the data source
4. After creating and connecting to the server, you can link it to Microsoft Excel or Microsoft Access.

Create a data source

The first step in creating an ODBC server is to create a data source. To do this from a Windows machine, take these steps:

1. Select the Windows icon in your start/task bar.
2. Type "ODBC."
   - After selecting the Windows icon, you will not see the search bar, but it will appear once you begin typing.
3. Select **ODBC Data source (32 bit)**.
   - This program is part of the Windows operating system and does not need to be downloaded.
4. On the **User DSN** tab, select **Add**.
5. Select **ODBC Driver 17 for SQL Server**.
6. Select **Finish**.
   - You must use version 17 or greater. If you do not see this option, download and

install the driver here.



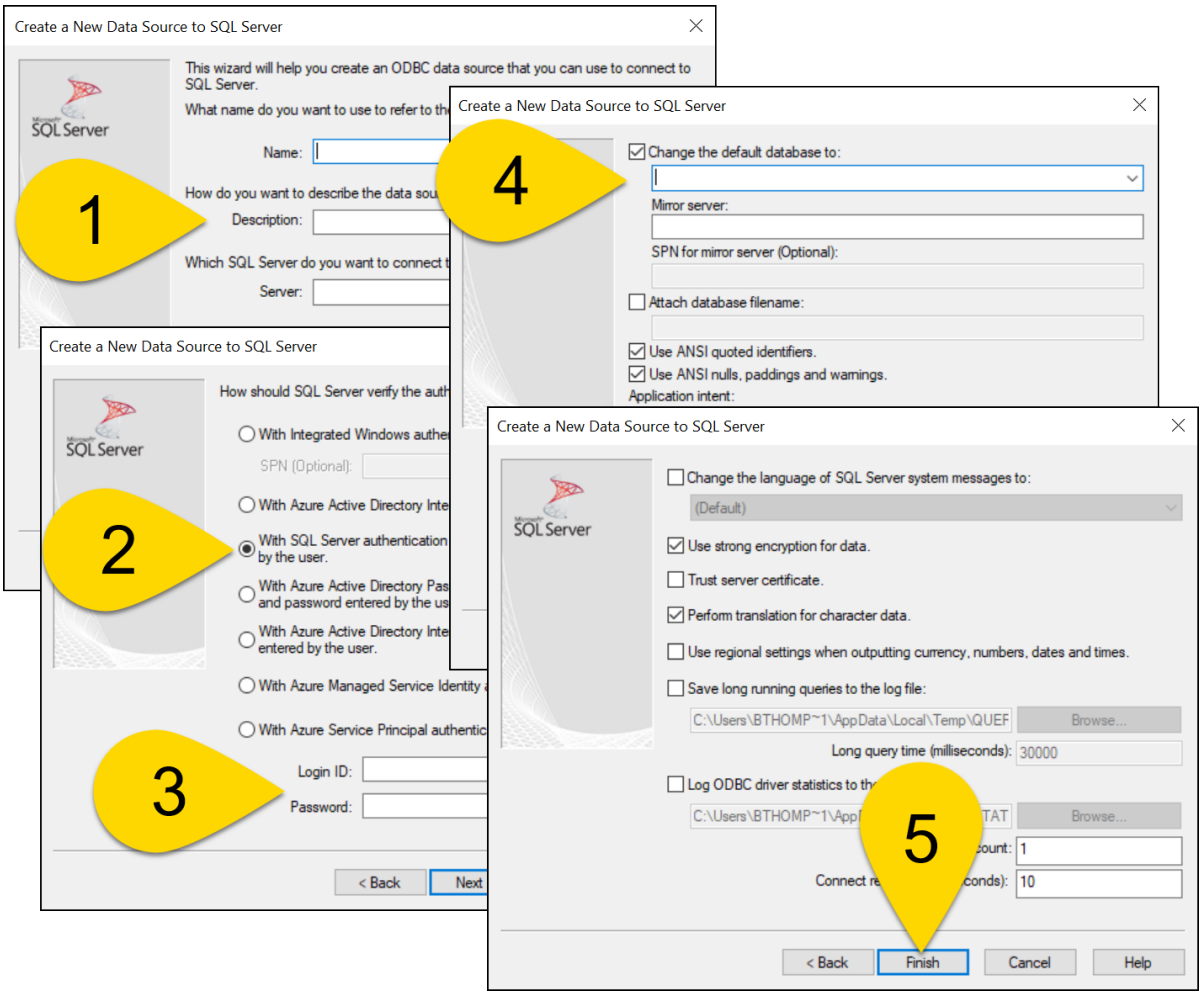
Connect to the SQL server

Once you create a data source as explained above, you need to connect the data source to the SQL server.

1. When you select **Finish** after creating the data source, a new window will appear. Enter a name for the data source (free-form text), description (free-form text), and the server name. Select **Next**.
   - Contact your Axon representative to receive the server name for your agency.
2. Select the **With SQL Server authentication using a login ID and password entered by the user**
3. Enter your username and password.
   - Contact your Axon representative to receive this login information.

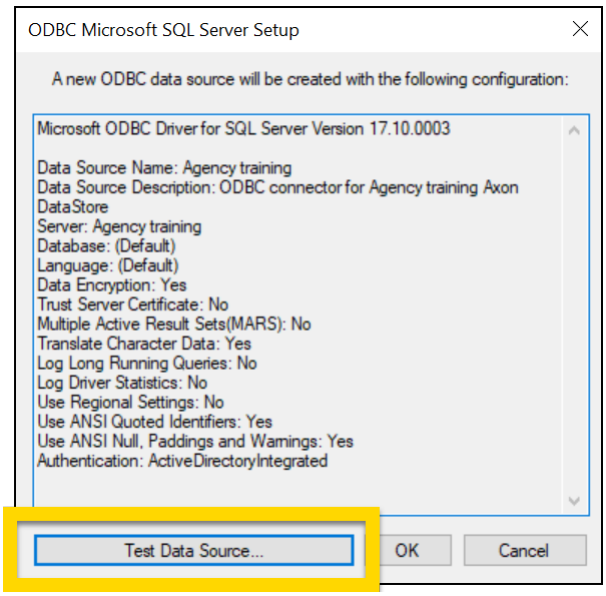
4. Select the **Change the default database to option,** enter your database name, and select **Next**. The rest of the entries in this section do not need to be adjusted.
   - Contact your Axon representative to receive the database name for your agency.
5. Nothing on this screen should be adjusted. Select **Finish**.

> **Warning**
>
> **Always** use strong encryption.
>
> **Do NOT** blindly trust the server certificate (this forces the client to verify the identity of the TLS certificate received from the server)



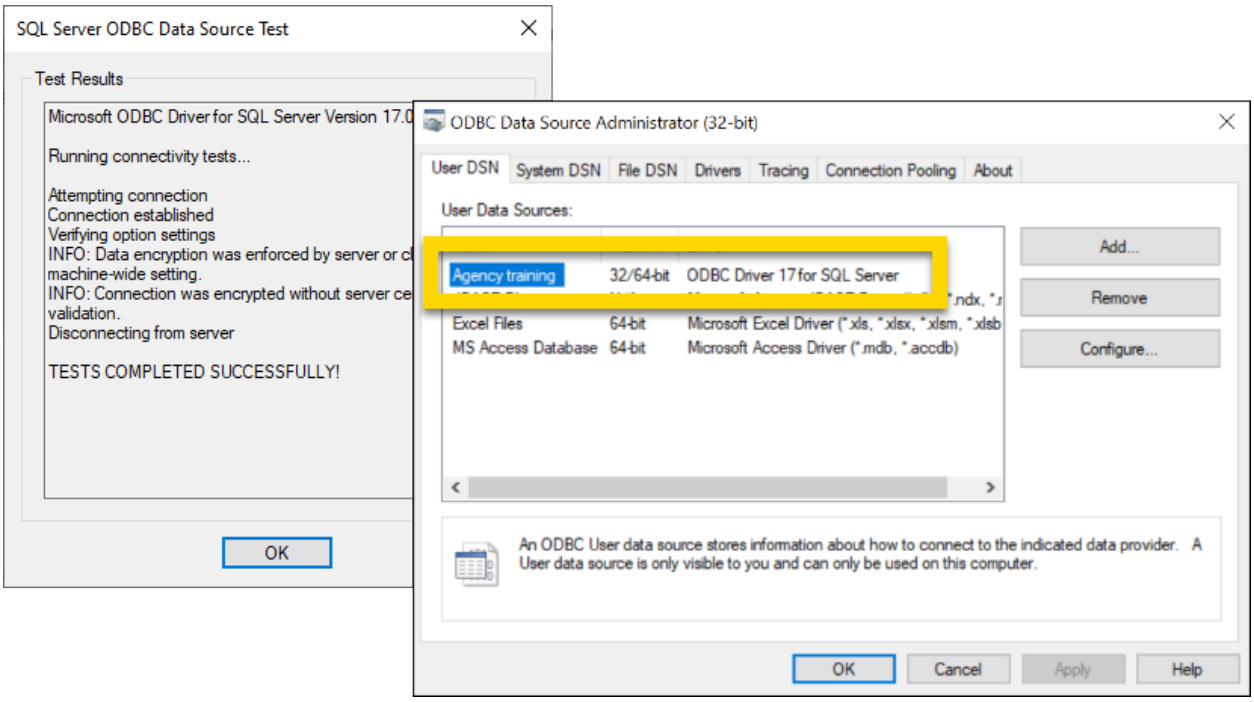
Test the data source

When you select **Finish** after setting up the SQL server connection, a new window will appear and display a configuration summary. Select **Test Data Source**.



Test completed successfully

If your test passes, select **OK**. The data source testing window will close, and you will again see the ODBC Data Administrator application. Your new data source will now display on the **User DSN** tab.

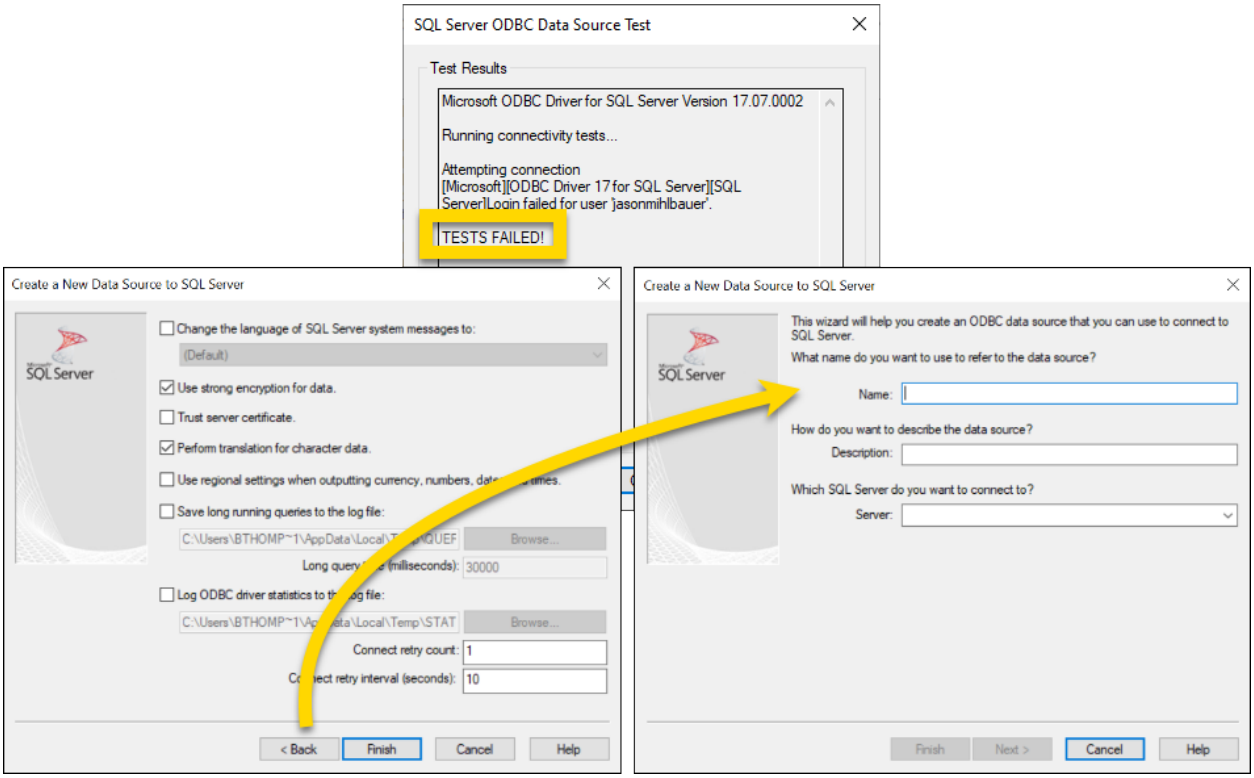


Test failed

If your test fails, select **OK**.

You will return to the final window in the SQL server connection workflow (step 5 in the previous section). Select **Back** three times until you reach the first window in this workflow (step 1 in the previous section).

Repeat steps 1-5 from the previous section and again test the data source until the test passes and your new data source appears in the ODBC Data Administrator application.

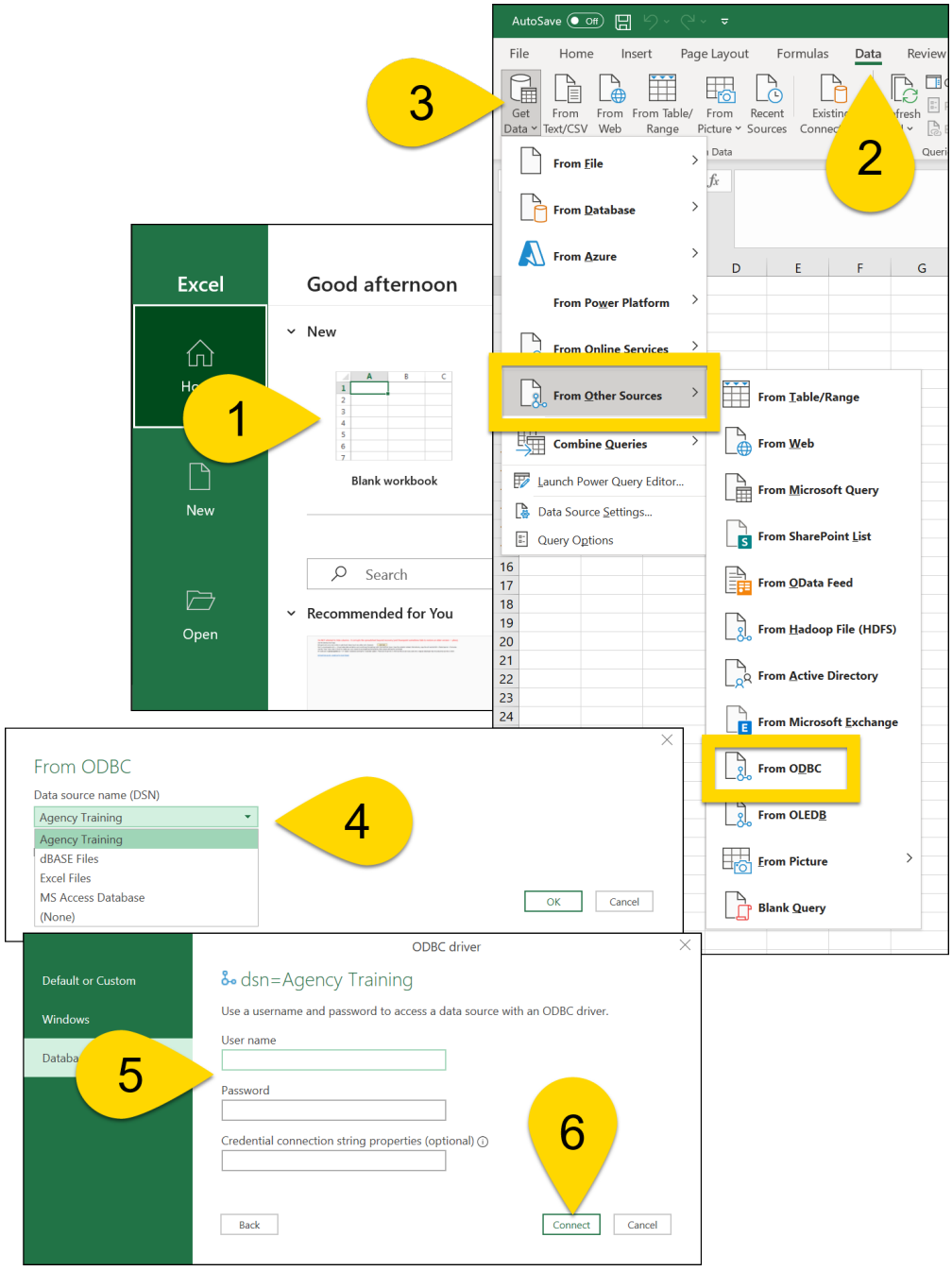


Link ODBC server to Microsoft Excel

Once you have created an ODBC server, you can link it to Microsoft Excel and access the Axon DataStore. To create this link, follow these steps:

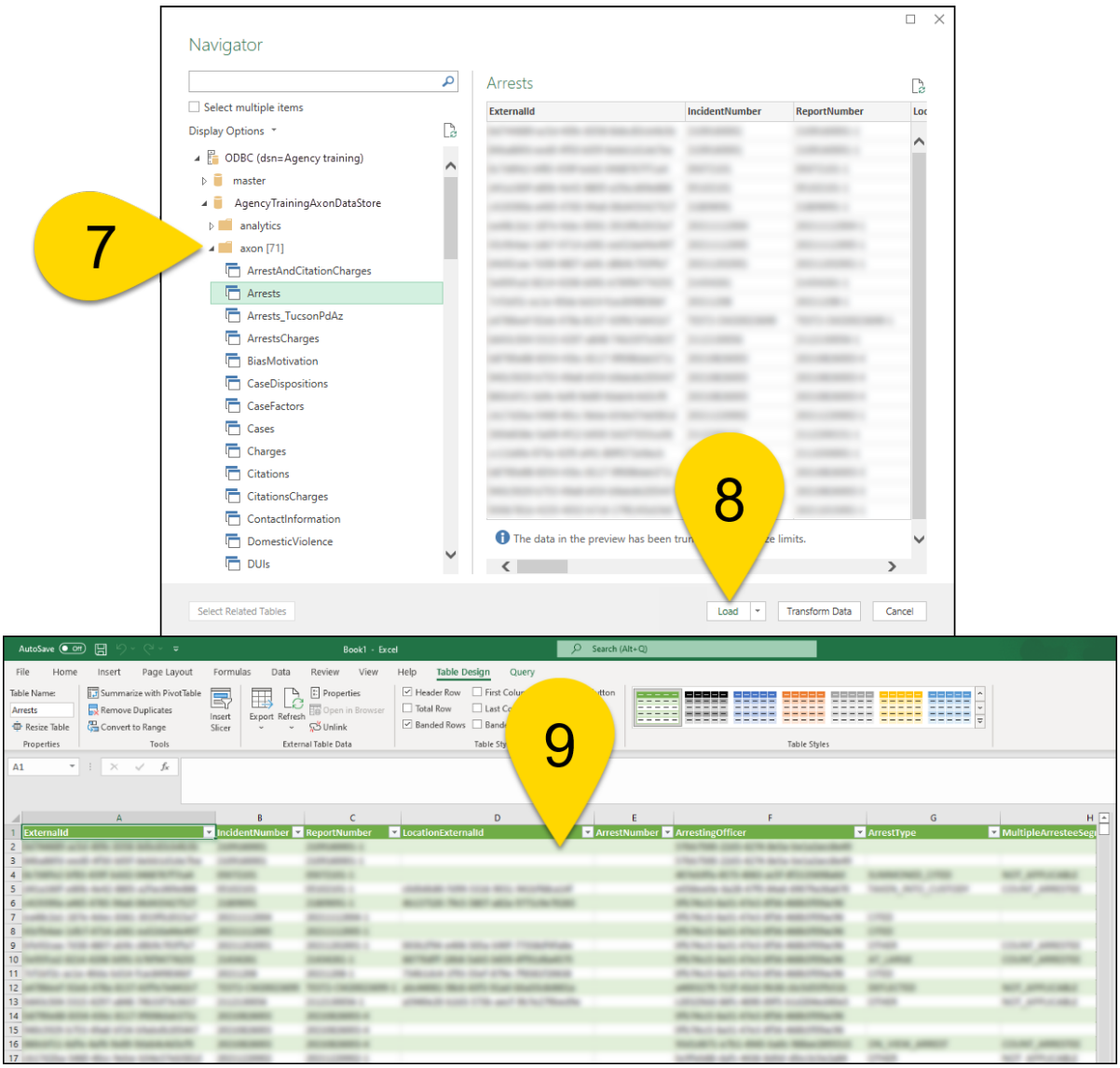
1. Open a blank workbook in Microsoft Excel.
2. Go to the **Data** tab.
3. Select **Get Data > From Other Sources > From ODBC**.
4. Reveal the list of data sources, choose your ODBC server, and select **OK**.
5. Enter your username and password associated with that Axon DataStore. You can leave the **Credential connection string properties (optional)** field blank.

6. Select **Connect**.



7. The navigator will open where you can select the arrow beside *axon*, *dw* or *raw*, depending on which data you want to populate in Excel.
8. After a data preview appears, select **Load**.

9. The data will populate in Excel.

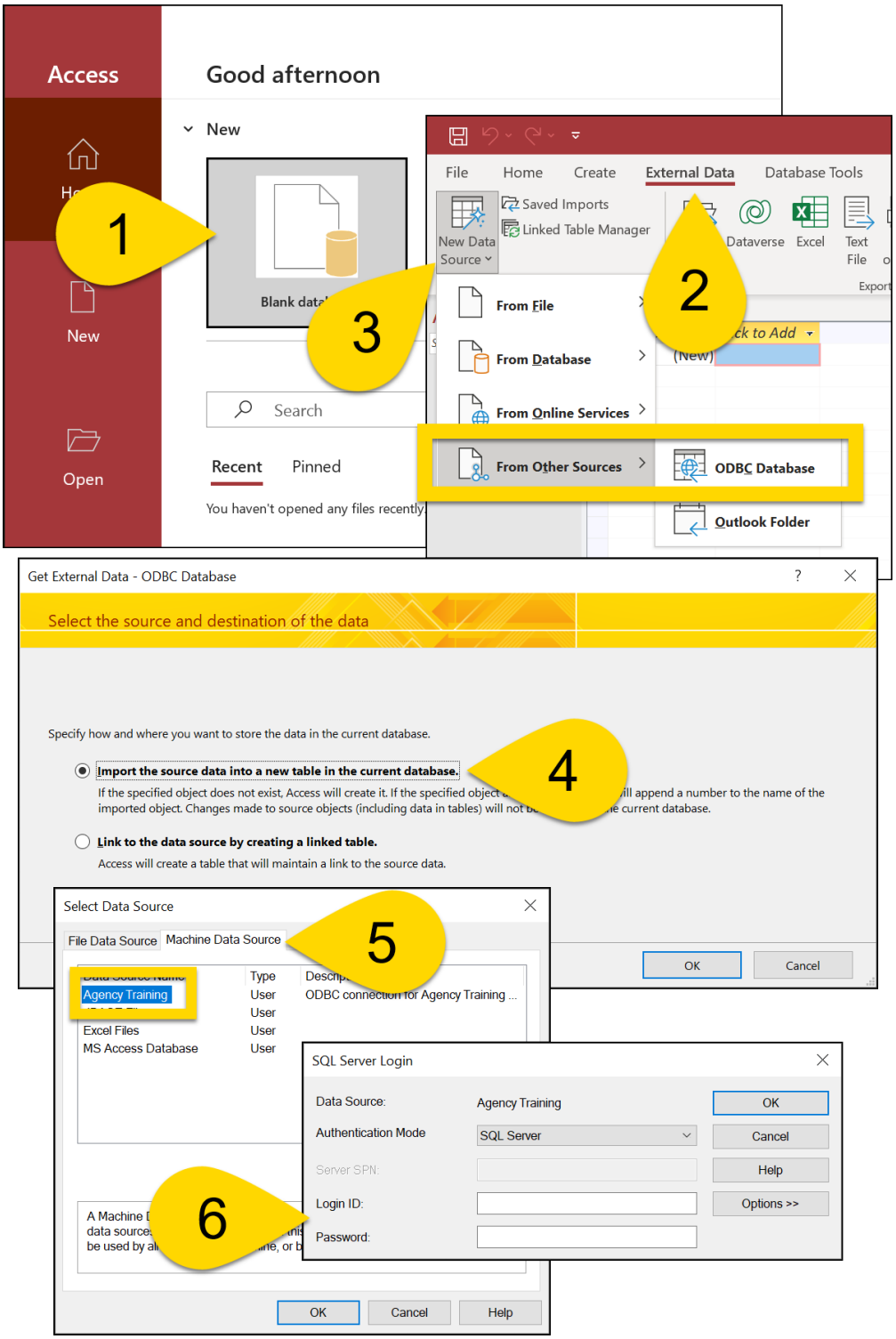


Link ODBC server to Microsoft Access

Once you have created an ODBC server, you can link it to Microsoft Access and access the Axon DataStore. To create this link, follow these steps:

1. Open a blank database in Microsoft Access.
2. Go to the **External Data** tab.
3. Select **New Data Source > From Other Sources > From ODBC**.
4. Choose **Import the source data into a new table in the current database** and select **OK**.

5. Select the **Machine Data Source** tab in the explorer window that appears and choose the ODBC server connection.
6. Enter your username and password for with that Axon DataStore and select **OK**.



7. Select which views to import into the Microsoft Access database.

- You can also save your password for future ease.
- Once you have selected all views you want to import, select **OK**.

8. It may take some time for the views to import. Once the import is complete, you can save the import steps and close out of the import process.
9. Double-click a view in the left panel to display the data for that view.