



BLACKDUCK | Hub

Report Database

Version 4.4.1



This edition of the *Report Database* refers to version 4.4.1 of the Black Duck Hub.

This document created or updated on Monday, January 15, 2018.

Please send your comments and suggestions to:

Black Duck Software, Incorporated
800 District Avenue, Suite 201
Burlington, MA 01803-5061 USA

Copyright © 2018 by **Black Duck Software, Inc.**

All rights reserved. All use of this documentation is subject to the license agreement between Black Duck Software, Inc. and the licensee. No part of the contents of this document may be reproduced or transmitted in any form or by any means without the prior written permission of Black Duck Software, Inc.

Black Duck, Know Your Code, and the Black Duck logo are registered trademarks of Black Duck Software, Inc. in the United States and other jurisdictions. Black Duck Code Center, Black Duck Code Sight, Black Duck Hub, Black Duck Protex, and Black Duck Suite are trademarks of Black Duck Software, Inc. All other trademarks or registered trademarks are the sole property of their respective owners.

| | |
|--|----------|
| Chapter 1: About the report database | 1 |
| Report database schema | 2 |
| Project Table | 2 |
| Project Version Table | 2 |
| Component Table | 3 |
| Component Usage Table | 3 |
| Component Match Type Table | 4 |
| Component License Table | 4 |
| Component Vulnerability Table | 4 |
| Chapter 2: Using Excel with the report database | 6 |
| Download and install the ODBC driver | 6 |
| Configure the ODBC driver | 6 |
| Configuring data source credentials | 7 |
| Extracting data from the report database | 8 |
| Extracting data to Excel | 8 |
| Using Microsoft Query | 9 |
| Manually querying the report database data using Microsoft Query | 12 |
| Query Example | 14 |

The Hub documentation

The documentation for the Hub consists of online help and these documents:

| Title | File | Description |
|-------------------------------------|---------------------------------|---|
| Release Notes | release_notes_bd_hub.pdf | Contains information about the new and improved features, resolved issues, and known issues in the current and previous releases. |
| Installing Hub using Docker Compose | hub_install_compose.pdf | Contains information about installing and upgrading the Hub using Docker Compose. |
| Installing Hub using Docker Swarm | hub_install_swarm.pdf | Contains information about installing and upgrading the Hub using Docker Swarm. |
| Installing Hub using Kubernetes | hub_install_kubernetes.pdf | Contains information about installing and upgrading the Hub using Kubernetes. |
| Installing Hub using OpenShift | hub_install_openshift.pdf | Contains information about installing and upgrading the Hub using OpenShift. |
| Getting Started | hub_getting_started.pdf | Provides first-time users with information on using the Hub. |
| Scanning Best Practices | hub_scanning_best_practices.pdf | Provides best practices for scanning. |
| Getting Started with the Hub SDK | getting_started_hub_sdk.pdf | Contains overview information and a sample use case. |
| Report Database | report_db_bd_hub.pdf | Contains information on using the report database. |

Hub integration documentation can be found on [Confluence](#).

Training

Black Duck Academy is a one-stop resource for all your Black Duck education needs. It provides you with 24x7 access to online training courses and how-to videos.

New videos and courses are added monthly.

At Black Duck Academy, you can:

- Learn at your own pace.
- Review courses as often as you wish.
- Take assessments to test your skills.
- Print certificates of completion to showcase your accomplishments.

Learn more at <https://www.blackducksoftware.com/services/training>

View the full catalog of courses and try some free courses at <https://academy.blackducksoftware.com>

When you are ready to learn, log in or sign up for an account: <https://academy.blackducksoftware.com>

Customer Success Community

The Black Duck Customer Success Community is our primary online resource for customer support, solutions and information. The Customer Success Community allows users to quickly and easily open support cases and monitor progress, learn important product information, search a knowledgebase, and gain insights from other Black Duck customers. The many features included in the Customer Success Community center around the following collaborative actions:

- Connect – Open support cases and monitor their progress, as well as, monitor issues that require Engineering or Product Management assistance
- Learn – Insights and best practices from other Black Duck product users to allow you to learn valuable lessons from a diverse group of industry leading companies. In addition, the Customer Hub puts all the latest product news and updates from Black Duck at your fingertips, helping you to better utilize our products and services to maximize the value of open source within your organization.
- Solve – Quickly and easily get the answers you're seeking with the access to rich content and product knowledge from Black Duck experts and our Knowledgebase.
- Share – Collaborate and connect with Black Duck staff and other customers to crowdsource solutions and share your thoughts on product direction.

[Access the Customer Success Community](#). If you do not have an account or have trouble accessing the system, please send an email to communityfeedback@blackducksoftware.com or call us at +1 781.891.5100 ext. 5.

To see all the ways you can interact with Black Duck Support, visit:

<https://www.blackducksoftware.com/support/contact-support>.

Chapter 1: About the report database

A PostgreSQL report database, `bds_hub_report`, provides access to the Hub data for reporting purposes. Use any reporting tool that supports JDBC connections, such as Jasper Reports, to access the data.

This database, separate from the Hub database, is created automatically when you install or upgrade the Hub and provides access to the following data:

- Projects
- Project versions
- Component versions
- Component usage
- Component match type
- Component licenses
- Component vulnerabilities

With the report database, you can:

- Create a report of the components in a project version.
- Create a report of the vulnerabilities in a component version.
- Query the database to obtain similar information across all of your projects, such as:
 - Selecting all projects with a particular license, phase, and/or distribution.
 - Selecting all components using a particular license.
 - Selecting all project/project versions having a particular component/component version.

Note the following:

- Database name: `bds_hub_report`
- Username: `blackduck_reporter`. This user has read-only access to the database.
- Exposed port: 55436
- Password for `blackduck_reporter`.
 - If using the database container that is automatically installed by the Hub, set the password before connecting to the database. For more information, see the installation guide for the orchestration tool you used to install the Hub.
 - If using an external PostgreSQL database, use your preferred PostgreSQL administration tool to configure the password.

Once the password is set you can connect to the report database. For example, using `psql`:

```
psql -U blackduck_reporter -p 55436 -h localhost -W bds_hub_report
```

Note: There will be a delay of up to 15 minutes for any changes made in the Hub to appear in the report database.

Report database schema

Project Table

| Column | Type | Description |
|--------------|------|-------------------------------------|
| project_id | UUID | Project ID. |
| project_name | text | Project name. |
| owner | UUID | User ID in the Hub. |
| tier | int | Project tier. A value between 1 -5. |
| description | text | Project description. |

Project Version Table

| Column | Type | Description |
|--------------|-----------------------------|--|
| project_id | UUID | Project ID. |
| version_id | UUID | Project version ID. |
| version_name | text | Project version name. |
| phase | text | Project phase: <ul style="list-style-type: none">• PLANNING• DEVELOPMENT• RELEASED• DEPRECATED• ARCHIVED |
| distribution | text | Project Distribution: <ul style="list-style-type: none">• EXTERNAL• SAAS• INTERNAL• OPENSOURCE |
| released_on | timestamp without time zone | Project release date. |

| Column | Type | Description |
|----------|------|--|
| notes | text | Notes about this version of the project. |
| nickname | text | Nickname for the project version. |

Component Table

| Column | Type | Description |
|------------------------|---------|---|
| id | bigint | ID. |
| project_version_id | UUID | Project version ID. |
| component_id | UUID | Component ID. |
| component_name | text | Component name. |
| component_version_id | UUID | Component version ID. |
| component_version_name | text | Component version name. |
| version_origin_id | UUID | Version origin ID. |
| origin_id | text | Origin ID. |
| origin_name | text | Name of the distribution (origin). |
| ignored | boolean | Indicates whether the component is ignored: <ul style="list-style-type: none">• "t" indicates that the component is ignored.• "f" indicates that the component is not ignored. |
| policy_approval_status | text | One of the following values: <ul style="list-style-type: none">• IN_VIOLATION• NOT_IN_VIOLATION• IN_VIOLATION_OVERRIDDEN |

Component Usage Table

| Column | Type | Description |
|--------------|--------|--|
| component_id | bigint | id field in the Component table. |
| usage | text | One of the following values: <ul style="list-style-type: none">• DYNAMICALLY_LINKED• STATICALLY_LINKED• SOURCE_CODE• DEV_TOOL_EXCLUDED• SEPARATE_WORK• IMPLEMENTATION_OF_STANDARD |

Component Match Type Table

| Column | Type | Description |
|--------------|--------|--|
| component_id | bigint | id field in the Component table. |
| match_type | text | One of the following values: <ul style="list-style-type: none">• FILE_FILES_ADDED_DELETED_AND_MODIFIED• FILE_DEPENDENCY• FILE_EXACT• FILE_EXACT_FILE_MATCH• FILE_SOME_FILES_MODIFIED• MANUAL_BOM_COMPONENT• MANUAL_BOM_FILE• PARTIAL_FILE |

Component License Table

| Column | Type | Description |
|--------------------|--------|--|
| id | bigint | ID. |
| component_table_id | bigint | id field in the Component table. |
| license_display | text | License name when it is a single license; license display when it is a complex license. For example, (License A OR license B). |

Component Vulnerability Table

| Column | Type | Description |
|--------------------|--------|--|
| component_table_id | bigint | ID field in the Component table. |
| vuln_id | text | Vulnerability ID, such as CVE-2017-1234 or 12345. |
| severity | text | One of the following values: <ul style="list-style-type: none">• HIGH• MEDIUM• LOW |

| Column | Type | Description |
|--------------------|--------------------------|---|
| remediation_status | text | Lists the remediation status. One of the following values: <ul style="list-style-type: none">• NEW• NEEDS_REVIEW• REMEDIATION_REQUIRED• REMEDIATION_COMPLETE• DUPLICATE• MITIGATED• PATCHED• IGNORED |
| target_date | timestamp with time zone | Target date to remediate the vulnerability |
| actual_date | timestamp with time zone | Actual date the vulnerability was remediated. |
| comment | text | Comments entered when remediating the vulnerability. |
| description | text | Description of the vulnerability. |
| base_score | numeric | Base score of the vulnerability. This score reflects the overall basic characteristics of a vulnerability that are constant over time and user environments. |
| exploit_score | numeric | Exploitability score of the vulnerability. This score measures how the vulnerability is accessed and if extra conditions are required to exploit it, taking into account access vector, complexity, and authentication. |
| impact_score | numeric | Impact score of the vulnerability. This score reflects the possible impact of successfully exploiting the vulnerability, considering the integrity, availability, and confidentiality impacts. |
| related_vuln_id | text | Empty except when BDSA has a related CVE vulnerability. If a BDSA vulnerability is mapped to a CVE, the related CVE is listed here; the BDSA vulnerability is listed in the vuln_id column. |

Chapter 2: Using Excel with the report database

Excel is a tool you can use for viewing data.

The following examples show how you can use Excel to retrieve and display data from the report database.

Note: The procedures you use to obtain and display data using Excel may be different than what is shown here as the process depends on your system and version of Excel. Refer to your Excel documentation for more information on using this application.

Since Excel requires an ODBC driver to access the report database, download, install, and configure the ODBC driver.

Download and install the ODBC driver

Excel requires an ODBC driver to access the report database.

1. Download and install the ODBC driver.

- Windows: <https://odbc.postgresql.org/>

Download the 32- or 64-bit version of the driver, depending on your system, and install it.

- Mac: <http://macappstore.org/psqlodbc/>

Configure the ODBC driver

1. Open the ODBC Data Source Administrator dialog box.

The method to open the dialog box depends on whether you are using a Mac or the version of Windows.

For example, one way to open the ODBC Data Source Administrator dialog box in Windows 8 is by selecting **Administrative Tools** from the Control Panel in Windows and double-clicking **ODBC Data Source (64-bit)**.

2. Select the **User DSN** tab and click **Add**. The Create New Data Source dialog box appears.
3. Select **PostgreSQL Unicode** or **PostgreSQL ANSI** and click **Finish**. The PostgreSQL Unicode/ANSI ODBC Driver Setup dialog box appears.
4. Complete the fields in this dialog box as follows:

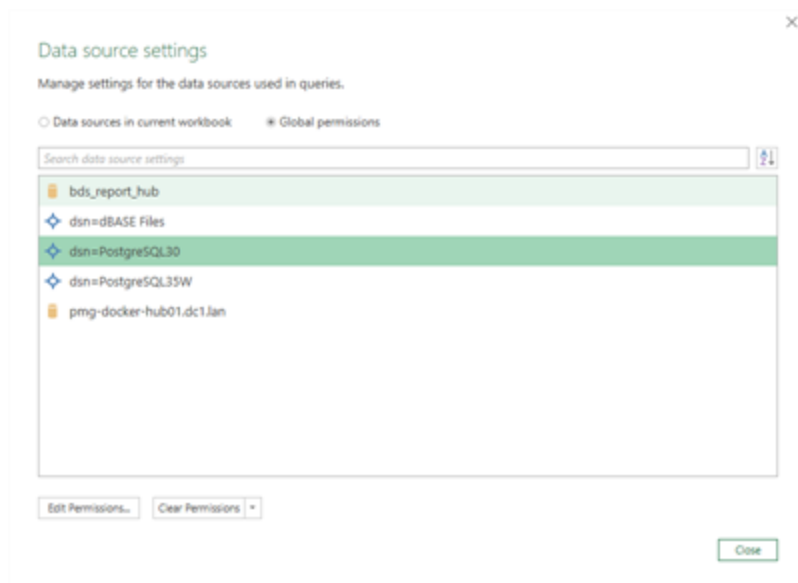
- **Data Source:** Enter the data source such as **PostgreSQL30** or **PostgreSQL35W**. Those values may already appear here.
 - **Database:** Enter **bds_hub_report**.
 - **Server:** Enter the IP address (x.x.x.x) or hostname.
 - **User Name:** Enter **blackduck_reporter**.
 - **SSL Mode:** Select **allow** from the list.
 - **Port:** Enter **55436** unless a different port was selected during the install or upgrade process.
 - **Password:** Enter the password for the report_database.
5. Exit all dialog boxes: click **Save** in the PostgreSQL Unicode ODBC Driver Setup dialog box **Finish** to exit the Create New Data Source dialog box and then **OK** in the ODBC Data Source Administrator dialog box.

Configuring data source credentials

After configuring the ODBC driver, use Excel to confirm the data source credentials.

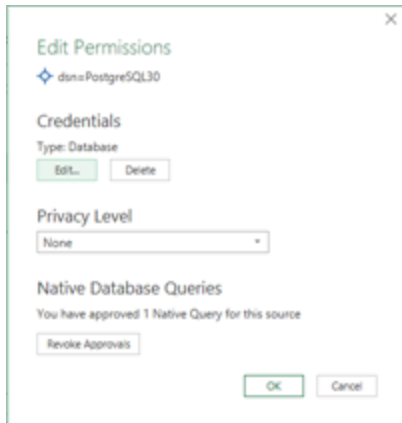
1. Open Excel and select to open a new blank workbook.
2. Select **Data > Get Data > Data Source Settings**.

The Data source settings dialog box appears.



3. Select the data source, as defined when configuring the ODBC driver, to connect to and click **Edit Permissions**.

The Edit Permissions dialog box appears.



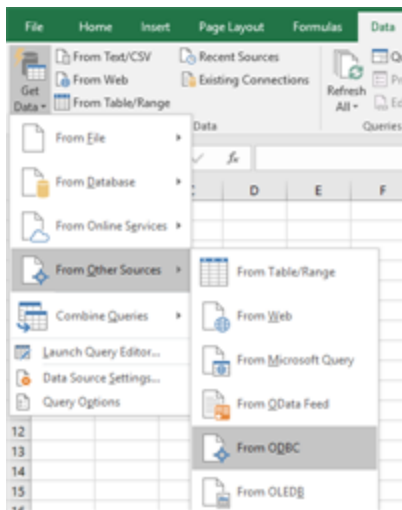
4. Select **Edit** under **Credentials** to open the ODBC driver dialog box.
5. Enter the username and password. Ensure that the values you entered match what was entered when configuring the ODBC driver and click **Connect**.
6. Once the connection is confirmed, click **OK** in the Edit Permissions dialog box and click **Close** in the Data source settings dialog box.

Extracting data from the report database

There are several methods to extract data from the report database. You can extract the data directly to Excel or use Microsoft Query to retrieve the data.

Extracting data to Excel

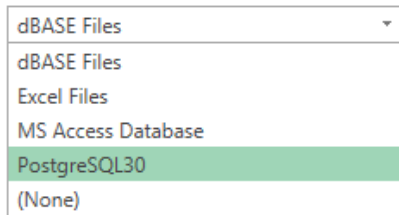
1. In Excel, select **Data > Get Data > From Other Source > From ODBC**.



The From ODBC dialog box appears.

From ODBC

Data source name (DSN)



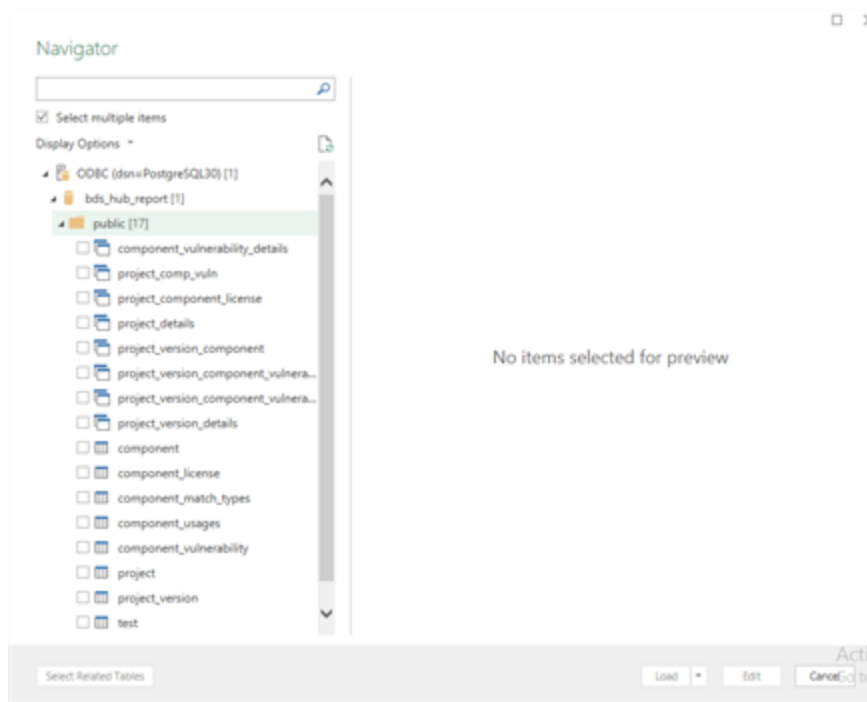
OK

Cancel

2. Select the database and click **OK**.

The Navigator window appears.

3. Expand the tables shown by selecting **ODBC > bds_hub_report > public**.



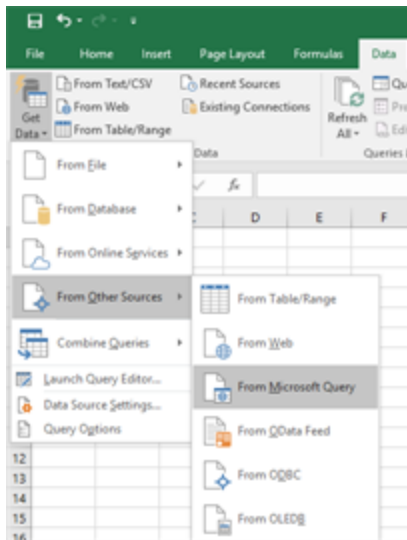
4. Select the table(s) to view and click **Load**.

The tables are loaded into Excel.

Using Microsoft Query

You can use Microsoft Query to retrieve your data from the report database.

1. In Excel, select **Data > Get Data > From Other Sources > From Microsoft Query**.

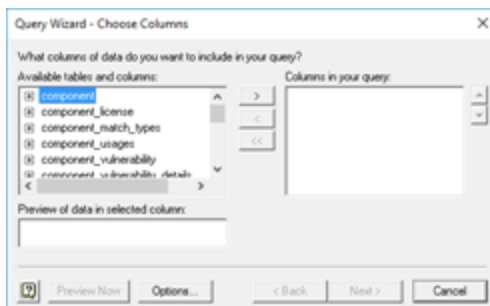


The Choose Data Source dialog box appears.



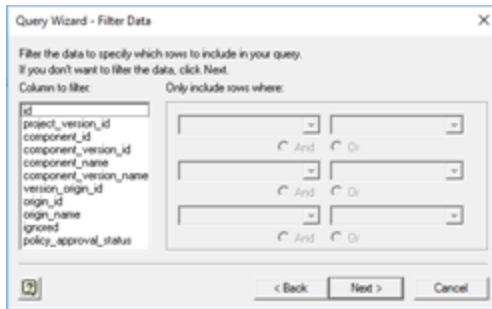
2. Select the data source you entered when configuring the ODBC driver and click **OK**.

The Query Wizard - Choose Columns dialog box appears.



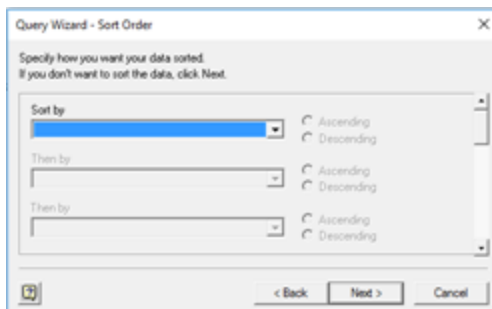
3. Select the tables and columns to be viewed by selecting them and clicking > to move them to the **Columns in your query** section. Click **Next**.

The Query Wizard - Filter Data dialog box appears.



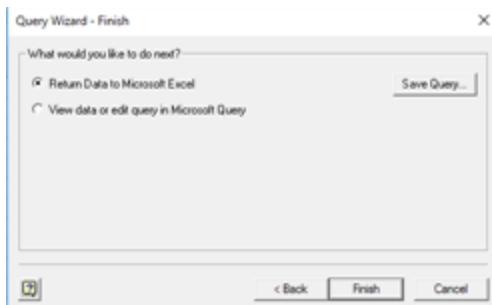
4. Select the filters for your query and click **Next**.

The Query Wizard - Sort Order dialog box appears.



5. Select to sort the data in ascending or descending order and click **Next**.

The Query Wizard - Finish dialog box appears.

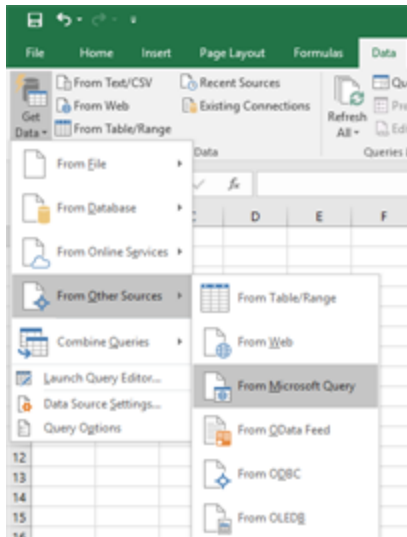


6. Select **Return Data to Microsoft Excel** and click **Finish**.

The data displays in Microsoft Excel.

Manually querying the report database data using Microsoft Query

1. In Excel, select **Data > Get Data > From Other Sources > From Microsoft Query**.

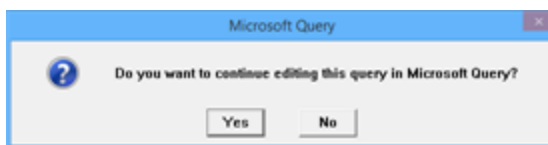


The Choose Data Source dialog box appears. Select the database and click **OK**.



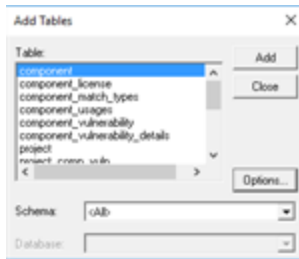
2. Click **Cancel**.


The following message appears.

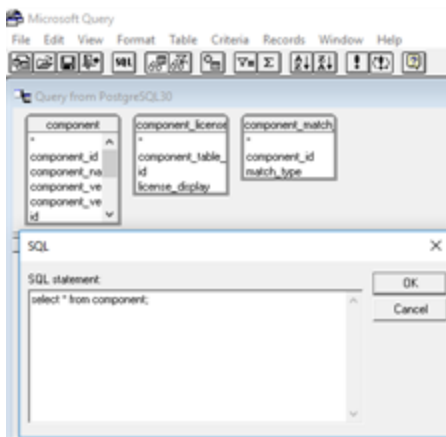


3. Click **Yes**.

The Add Tables dialog box appears.



4. Select the tables from the list for your query and click **Add**. Once the tables have been selected, click **Close**.
5. Microsoft Query opens. If it does not automatically open, click .

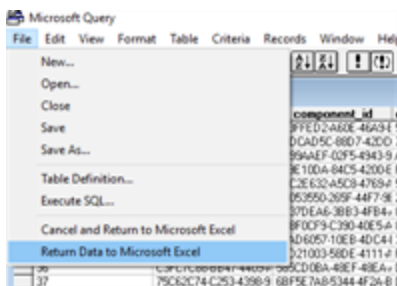


6. Type the SQL query in the **SQL statement** field and click **OK** to view the results.

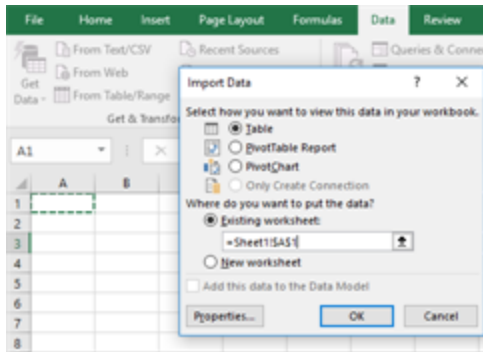
The results appear in Microsoft Query.

Click **OK** if a message appears stating that the data can't be represented graphically.

7. To view these results in Excel, in Microsoft Query, select **File > Return Data to Microsoft Excel**.



8. Excel opens and the Import Data dialog box appears.



9. Complete the information in the dialog box, select a cell in a workbook where the data should load, and click **OK**.

The data is loaded into Excel.

Query Example

The following is a sample query that lists the projects and components which have a policy violation:

```
select project_name, version_name, phase, component_name, component_version_name, policy_approval_status from project_version
inner join project on project.project_id= project_version.project_id
inner join component on project_version.version_id=component.project_version_id
where policy_approval_status= 'IN_VIOLATION' and phase != 'ARCHIVED' order by project_name asc;
```