

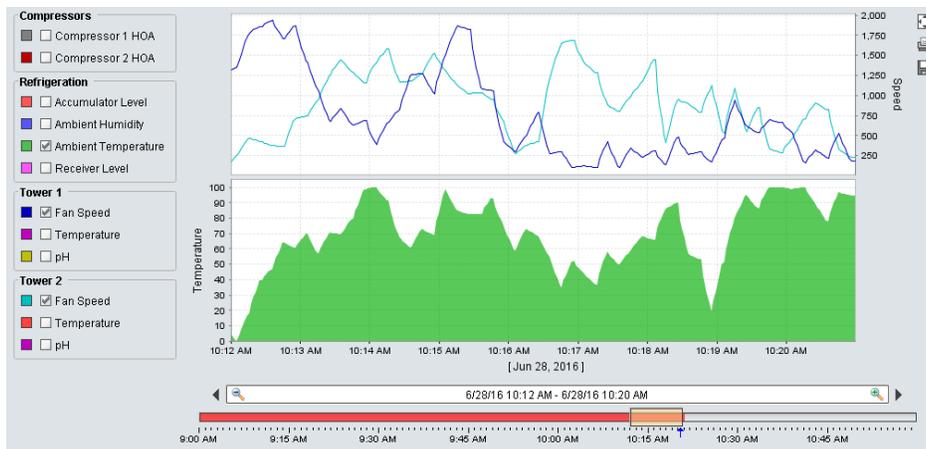
SQL in Ignition

Leveraging Databases

In addition to all the normal HMI functionality you would expect, Ignition has the ability to connect to databases, and this can greatly increase the functionality available to you! You can use databases to store history, create easy to search lists and configurations, and retrieve data from ERP or other systems. When you start using SQL, you can expand your project from a simple HMI to a project that brings your whole process together. The best part is that Ignition connects to as many databases as you want, all from one central location. Because the database lives outside of Ignition, we don't maintain any control or rules over what you can do with your data.

Displaying Data

You can easily display information from your databases on a window along with anything else in Ignition. You can show parts lists, step sequences, [realtime or historical charts](#), add the ability to search for inventory, or anything else you can think of.



On this page

...

- Leveraging Databases
 - Displaying Data
- What is SQL?
- SQL Queries
- Database Connections
- Using SQL in Ignition
 - Queries in Bindings
 - Queries in Scripting
 - Queries in Tags
 - Queries in Reports
 - Queries in Transaction Groups
 - Database Query Browser
- Auto Generated Queries
- Other Resources for Learning SQL

What is SQL?

SQL stands for "Structured Query Language" and is the backbone of most modern relational databases. It's often referred to as "S.Q.L." or "Sequel," but both are correct and widely recognized. This language (different than the [Python Scripting Language](#)) allows you to write out requests or "queries" against the existing data to [view](#), [add](#), [edit](#), or [remove](#) the information you want in a simple format.

Everything in a relational database is based around tables. Tables store the basic information for any system and can be combined together to make very efficient queries to retrieve your data.

	id	name	description	location	qty
▶	576	Mold 1	Mold for part # 1523	Warehouse A	23
	577	Mold 2	Mold for part # 8825	Warehouse A	12
	578	Bin 1	Finished Goods Bin 1	Warehouse B	9
	579	Bin 2	Scrap Bin 1	Warehouse B	14

```
SELECT *  
FROM inventory  
JOIN location on location.name = inventory.location;
```

SQL Queries

SQL queries are crucial to Ignition's database-centric model. Queries can show what is available or alter data in the databases, and some companies have positions just dedicated to running databases and creating queries. Anywhere Ignition is fetching data, you can choose to use your own custom queries to get exactly what you want out of the database. You can make your queries as simple or complex as you like. If your database is large, you might have a whole team dedicated to creating these queries for you and Ignition will happily execute them.

SQL - Select Statement - Select Data from a Table

```
SELECT * FROM mytable
```

SQL - Inner Join Statement - Selects Records that have Matching Values

```
SELECT users.id, users.firstname, users.lastname, roles.name as 'rolename'  
FROM users  
INNER JOIN mapping ON users.id = mapping.userid  
INNER JOIN roles ON mapping.roleid = roles.id  
WHERE roles.name = 'Administrator'
```

Database Connections

Any SQL query you use needs a [Database connection](#), but Ignition simplifies all that by creating database connections in the Gateway instead of in the clients. This means from one central location you can manage all your database connections, and you don't have to worry about planning around adding clients in the future. Any special rules or connection restrictions are taken care of in the Gateway.

Using SQL in Ignition

There are many types of queries, and many ways to use them in Ignition. Some provide an easy to use builder to automatically store or fetch data, and some allow you to completely customize your queries.

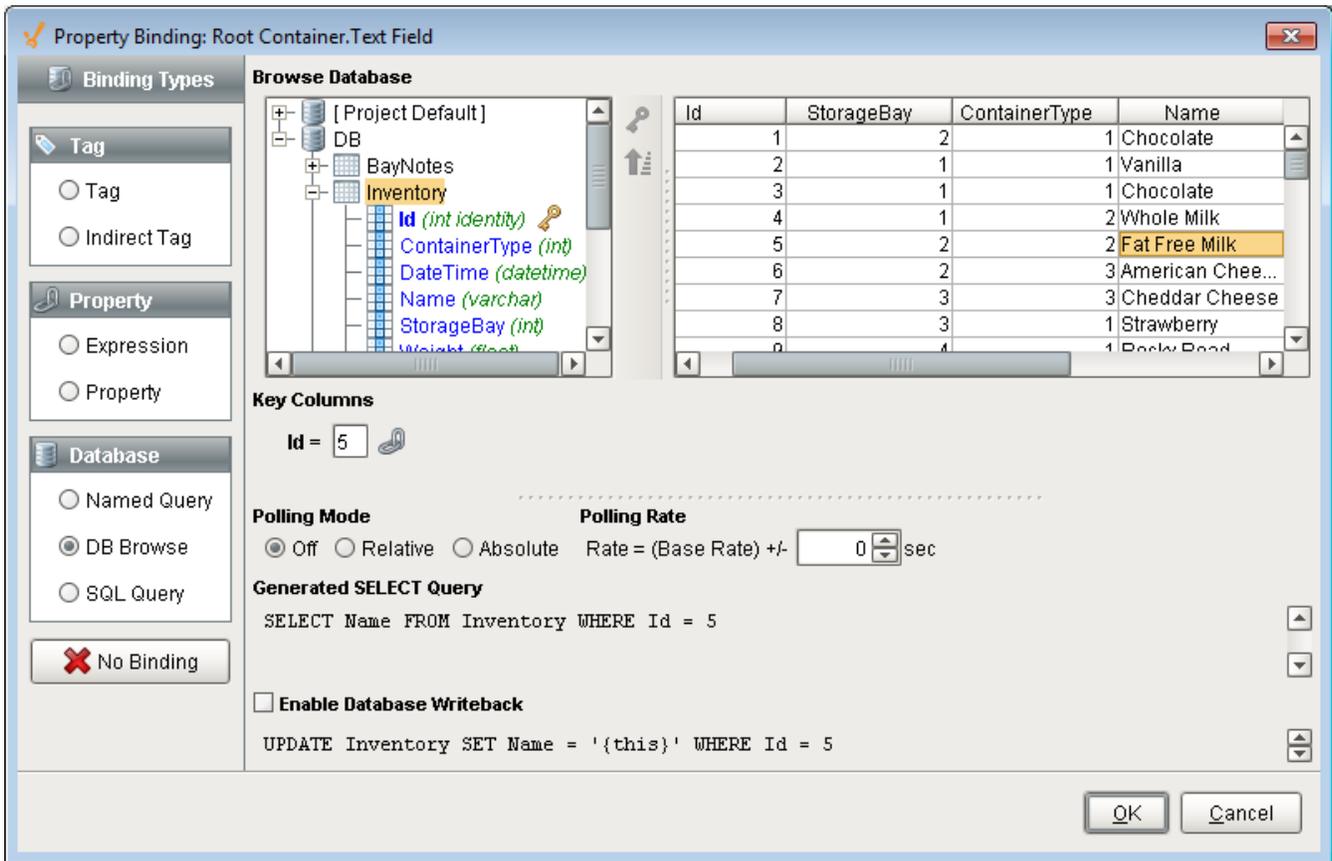
Queries in Bindings

Ignition's binding system offers a lot of flexibility in how database data can be used in a binding. The **Named Query** binding allows you to select one of the Named Queries that were previously built for that project, offering a very secure method of pulling data from the database. The **DB Browse** builder provides an interface that will build the query based on the data in the table that was selected. This allows even users with little SQL knowledge to pull data from the database. Finally, the **SQL Query** option will accept a straight query, so that a query specific to that binding can be written directly in the binding. When binding to a basic (non-dataset) data type, you can use the Writeback feature directly to send any changes back to the database.



**Querying Data from
Database**

[Watch the Video](#)



Queries in Scripting

Ignition offers a number of built-in scripting functions for you to use to query your databases. This makes it very simple to view data, create dynamic scripts that use real data, and more. You can pull individual pieces of information, return whole tables of data, or update your database directly. Depending on the type of query and the sort of results you want, you will use different functions. The following functions are the ones you will use most, and all of them can use a special placeholder (?) to allow for dynamic query building.

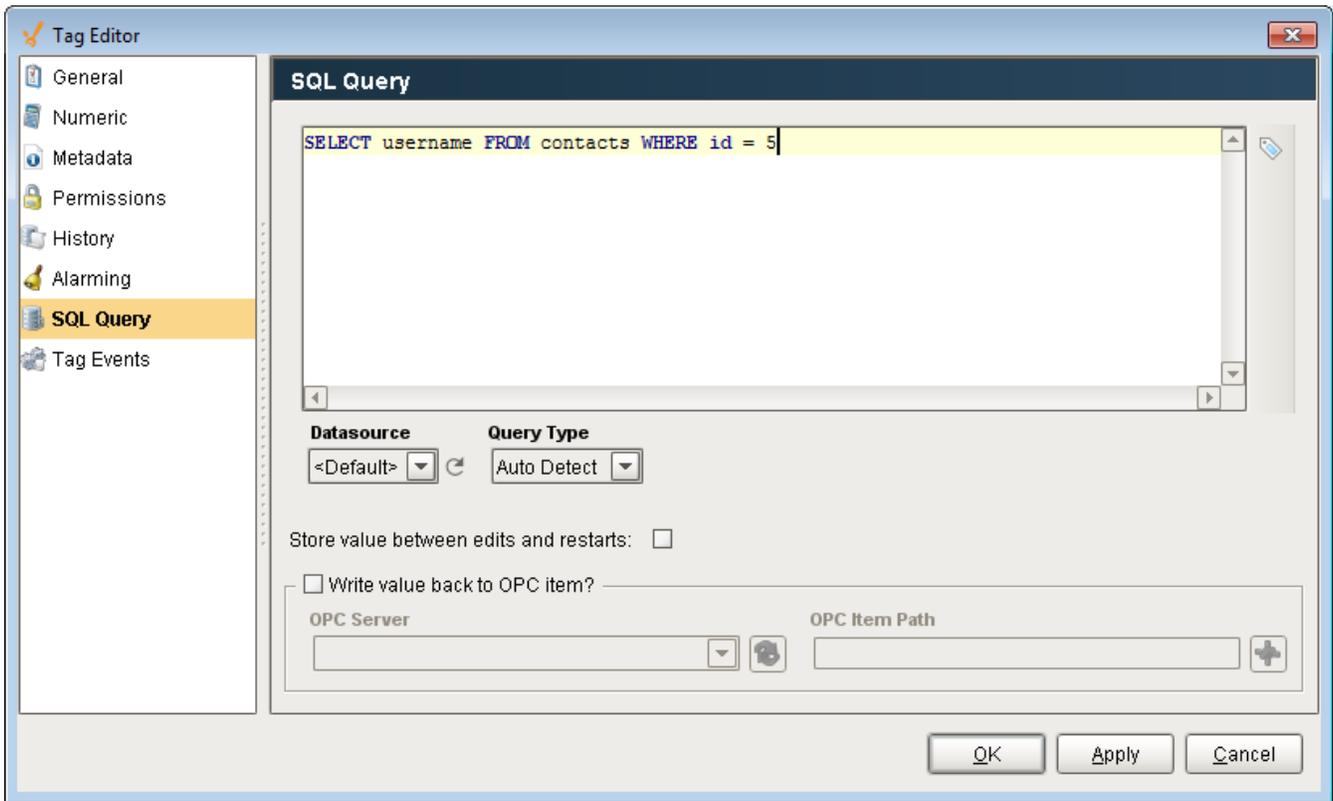
Scripting Function	Description
<code>system.db.runNamedQuery()</code>	Used to run a previously setup Named Query from within a script.
<code>system.db.runPrepQuery()</code>	Used to run basic SELECT queries to fetch whole datasets. This can be used to populate tables, or to sift through the data to do your own calculations.
<code>system.db.runPrepUpdate()</code>	Used to run queries that change the data in the database. Usually used on input form windows to update your database records.
<code>system.db.runScalarPrepQuery()</code>	Used when you want only one value from your results. Perfect for fetching a single value like the highest ID, or first timestamp of a result set.

Each of the different functions takes in different arguments (values) and provides slightly different options and functionality. For example, the `runPrepUpdate()` can return the auto-generated key from insert queries. This can be extremely helpful and eliminate the need to hit the database multiple times if you are using linked tables.

You can find examples of each of these and all the other database functions in the [system.db](#) section of the appendix.

Queries in Tags

Ignition offers [Query Tags](#), which can run queries and return the result as a Tag value, giving all of the projects in the Gateway access to the same Database values.



Queries in Reports

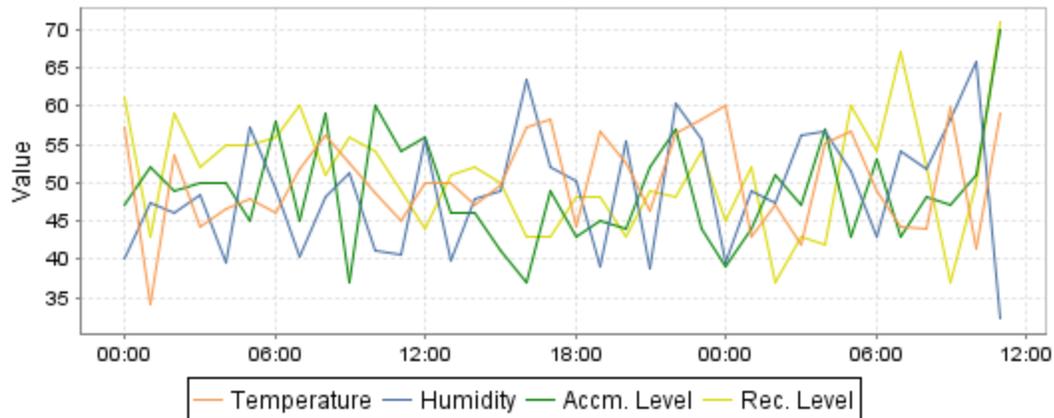
You can leverage queries to access data from all database connections to [create rich reports](#), from simple data logging to complex reports with grouped charts and datasets.

Tag History Report

A report that queries tag history and displays a graph and table

Report Start: 6/27/2016 12:00 AM

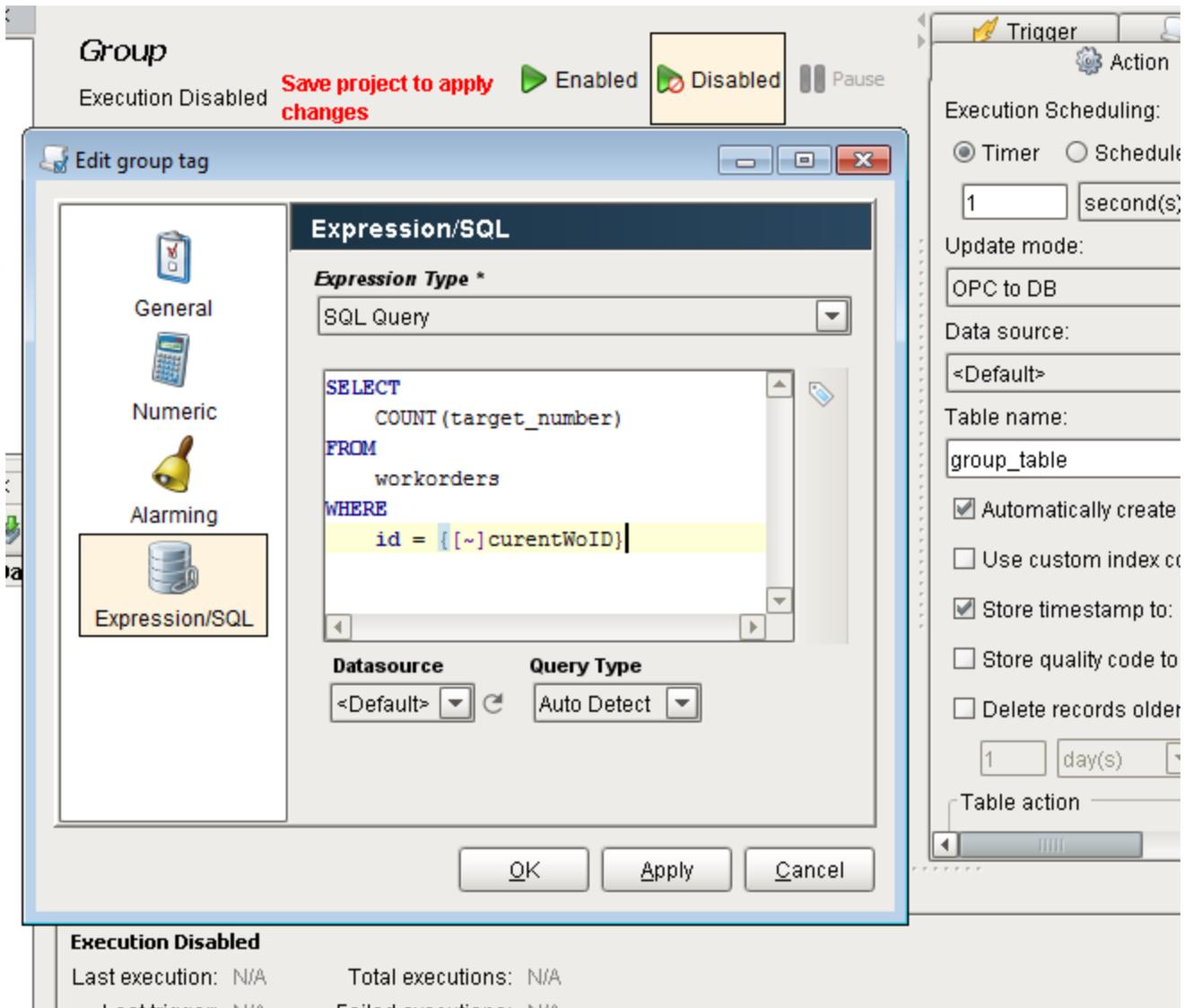
Report End: 6/28/2016 11:00 AM



Date / Time	Temperature	Humidity	Accm. Level	Rec. Level
6/27/2016 12:00:00 AM	57.34	40.14	47	61
6/27/2016 1:00:00 AM	33.98	47.33	52	43
6/27/2016 2:00:00 AM	53.61	46.15	49	59
6/27/2016 3:00:00 AM	44.3	48.5	50	52
6/27/2016 4:00:00 AM	46.57	39.59	50	55
6/27/2016 5:00:00 AM	47.9	57.16	45	55
6/27/2016 6:00:00 AM	46.04	49.16	58	56
6/27/2016 7:00:00 AM	51.87	40.42	45	60
6/27/2016 8:00:00 AM	56.24	48.07	59	51
6/27/2016 9:00:00 AM	52.42	51.17	37	56
6/27/2016 10:00:00 AM	48.74	41.01	60	54
6/27/2016 11:00:00 AM	44.89	40.67	54	49
6/27/2016 12:00:00 PM	49.85	55.73	56	44
6/27/2016 1:00:00 PM	49.82	39.84	46	54

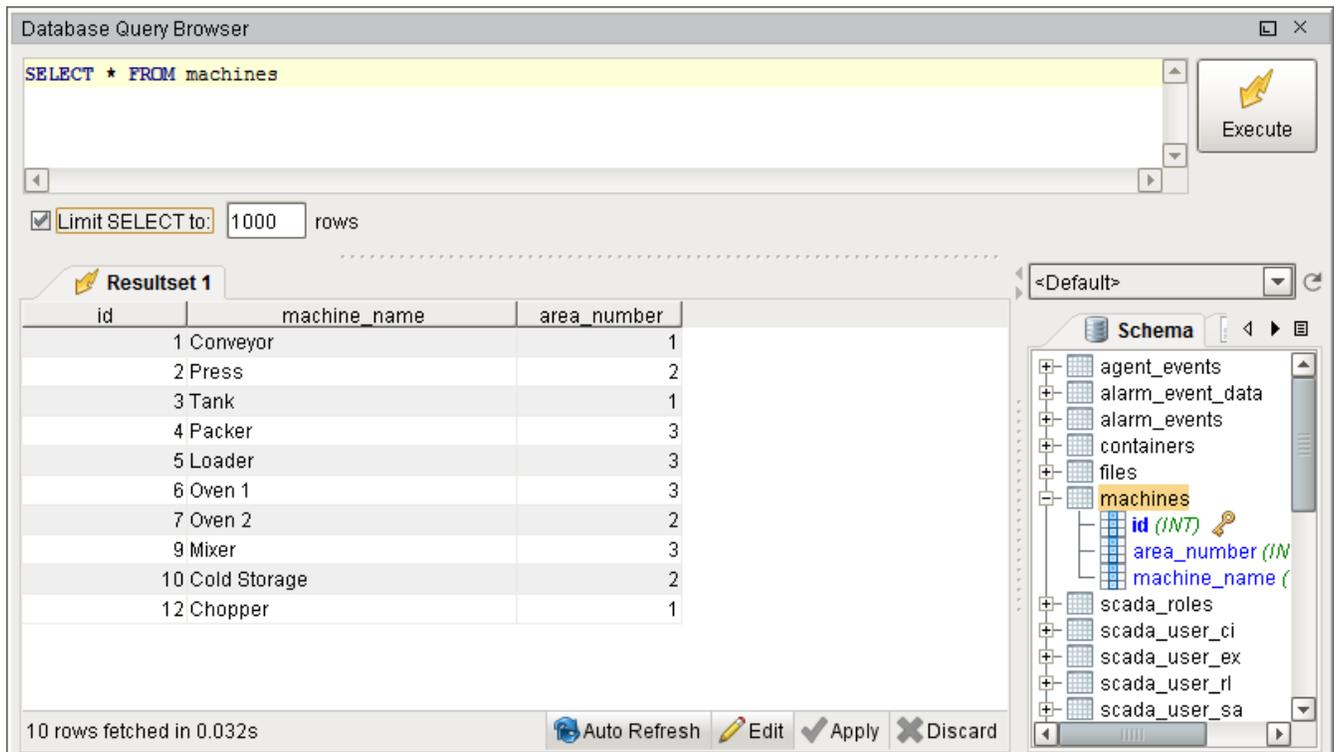
Queries in Transaction Groups

While [Transaction Groups](#) are great at storing Tag data to a Database automatically, the built-in [Expression Items](#) can execute a SQL Query within the Transaction Group.



Database Query Browser

The [Database Query Browser](#) offers an easy to use environment to run queries in for testing. Here, queries can be tested to figure out what values get returned, or data can be updated through raw queries, or the Database Query Browsers easy to use GUI editor.



Auto Generated Queries

Many systems within Ignition utilize a database connection, but the queries that are executed are constructed automatically by the system and do not require you to build the queries manually. These systems such as the [Tag Historian](#), the [Alarm Journal](#), or the [Database User Source](#) are very easy to setup and use since each system will automatically generate the necessary tables in the database, insert the relevant data, and even has prebuilt tools to extract the data. However, it is important to note that while these systems can automatically generate queries for you using the various components, these systems are simply storing data in a database which you can manually query out by building your own SQL queries.

Other Resources for Learning SQL

The following resources are unrelated to Inductive Automation and in no way affiliated, but provide good tutorials and basic SQL querying and syntax.

- [W3Schools SQL Tutorial](#)
- [SQL commands and syntax](#)
- [SQLZOO Tutorial](#)

Related Topics ...

- [Database Connections](#)
- [Connecting to Databases](#)
- [Scripting](#)

In This Section ...