

Database Connections

How Are Databases Used in Ignition?

While connecting to a database is not required for basic status and control functionality, it can dramatically increase the possibilities that the system offers. There are a few places where databases are used in Ignition, such as historical data logging, reporting, storing alarm logs, and as your Tags storage.

Historical Data Logging

Logging data for historical analysis, either through [Tags Historian](#) or with the [SQL Bridge module](#), requires a database connection. Databases are great at handling historical data, and by using a standard relational database your data is stored in an open format that can be used in many ways.

Reports, Graphs, and Charts

The Vision module makes it easy to present data stored in databases in a variety of ways. You can [quickly create charts](#) that show performance over time, locate anomalies, detect trends, and more. Furthermore, it's important to remember that it is possible to pull data from any database that Ignition is connected to, even if the data wasn't placed there by Ignition. This means you can tie in data from other sources or areas of your company, such as pulling in inventory and staff information, as well.

Storing Alarm Logs

[Store alarm information historically](#) and examine it later for patterns or trouble spots.

Database-driven Tags

You can use a SQL database as your [Tags repository](#). Any other Ignition system with access to the database will be able to share and contribute Tags, allowing you to create highly integrated distributed systems. For example, multiple plant sites can use Tags to report current status over a secure network connection to a central corporate headquarters.

Getting Started with Databases

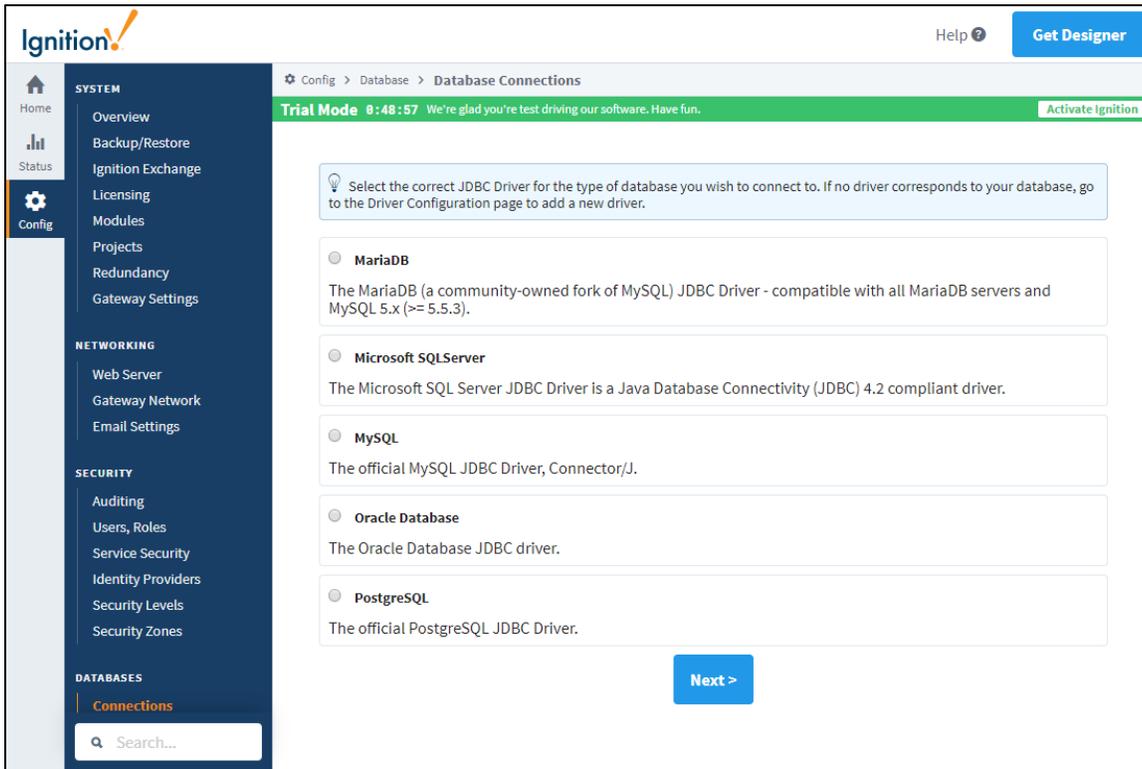
The first step in using a database with Ignition is to identify a database server. Many companies already have database servers maintained by their IT departments. If you do not, or wish to set up your own database server for Ignition, the [Supported Databases](#) section below offers some advice on choosing a database vendor.

Once you've identified a server, all you need to do is [create a connection](#) to that server to get up and running.

On this page

...

- [How Are Databases Used in Ignition?](#)
 - [Historical Data Logging](#)
 - [Reports, Graphs, and Charts](#)
 - [Storing Alarm Logs](#)
 - [Database-driven Tags](#)
- [Getting Started with Databases](#)
 - [Supported Databases in Ignition](#)
 - [Database](#)
 - [Version](#)
 - [Choosing other Databases](#)
 - [Installing and Connecting to a Database](#)
- [Database Drivers and Translators](#)
 - [What Is JDBC?](#)
 - [JDBC in Ignition](#)
 - [JDBC vs. ODBC](#)
 - [Can I Connect Using ODBC?](#)
- [Monitoring Connection Status](#)
- [Why Use Databases?](#)
 - [Central Management](#)



Supported Databases in Ignition

Ignition has been tested with the following databases, and can connect to them directly after installation. You can connect to other databases by installing additional JDBC drivers (the Java database connection specification), which are often provided by database vendors.

Database	Version
Full Support	
MySQL	5.0+ for full support. Ignition can connect to 4.x, but many features such as Tags are not tested.
Microsoft SQL Server	2005, 2008, 2012, 2014, 2016 (full and express editions). Ignition can connect to 2000, but has not been fully tested.
Oracle	10g, 11g, 12c (full and express). The letters stand for "grid" and "cloud"
PostgreSQL	8.0+
Firebird	All versions.
IBM DB2	9.5+
Limited support	
Microsoft Access	Access support is very limited, use it only to integrate existing data into the project, not for storing new data.
Other JDBC drivers	Due to variances in databases, some features may not work fully through other non-tested JDBC drivers. However, it is usually possible to get full functionality through the careful use of the database translator feature. For example, the JDBC driver for MariaDB could be downloaded and added to Ignition.

Choosing other Databases

If you are new to working with SQL databases and are trying to choose a vendor, you need to consider the following three factors:

1- Existing company usage

Many companies already use SQL databases for other purposes, and thus most IT departments already have a defined standard. Going along with your company's existing standard is usually recommended, as there will already be staff available who are knowledgeable about the system. Furthermore, you may be able to tie into your company's existing database system instead of maintaining your own.

2 - Price and Features

The fully supported databases shown above vary dramatically in price. Some systems can cost thousands of dollars, but may have a free "express" edition that will work perfectly well for your requirements. Others offer advanced features such as redundancy, which are either not offered or difficult to configure in the other systems. It is therefore important to clearly define the features and capabilities that you need.

3 - Most common among Inductive Automation users

Choosing a database that is commonly used by Inductive Automation users means that you are more likely to find examples and help in the [Forum](#), among other benefits. The supported database list above is sorted according to our current user install base.

Installing and Connecting to a Database

Once you've identified a server, all you need to do is create a connection to that server to get up and running. See the [Installing Databases](#) and [Connecting to Databases](#) sections for details about how to install and connect to different databases through Ignition.

If we don't already have a connector for your database type, you can [simply add it in yourself](#).

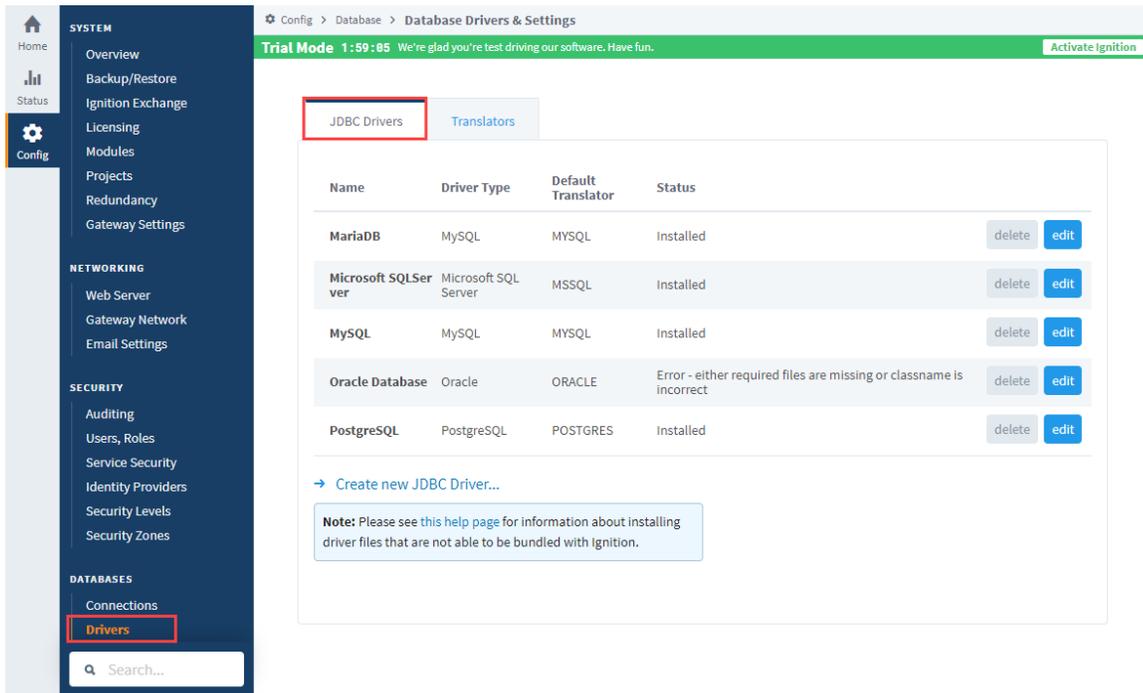
Database Drivers and Translators

What Is JDBC?

JDBC stands for the Java DataBase Connectivity API. It is a standardized way for Java-based applications to interact with a wide range of databases and data sources. A JDBC Driver enables Ignition to connect to, and use data from, a particular database system.

JDBC in Ignition

Ignition, being a Java-based application, leverages JDBC in order to connect to a variety of data sources. This enables Ignition to offer a standardized set of functionality on a wide range of different systems and databases. This includes not only commonly-used databases such as MySQL, Microsoft SQL Server, and Oracle, but additionally other lesser-known systems as well, provided the manufacturer offers a JDBC driver for the system.



JDBC vs. ODBC

JDBC differs from ODBC (Microsoft's OpenDataBase Connectivity standard) primarily in the fact that JDBC is written in Java, and thus can be used without modification in cross-platform environments. Additionally, whereas ODBC is a complex standard that is becoming technically out-dated, JDBC is a modern, clean specification for cross-vendor database access.

Can I Connect Using ODBC?

While it is indeed possible to connect to an ODBC data source using the JDBC-ODBC bridge, this is generally not advised. The bridge is designed to offer a minimal amount of functionality, and is considered a "transitional solution", meaning that it should only be used when JDBC is not available. In other words, if a JDBC option is available, ODBC should not be used.

Since most commercial databases offer JDBC drivers, transition is usually as simple as recreating your database connections inside of Ignition. The lack of a JDBC connection inside of Ignition does not necessarily indicate that JDBC isn't available for your particular database. Licensing restrictions sometime prevent the inclusion of drivers with 3rd-party software. Therefore, before using ODBC, due diligence should be taken to verify that no JDBC solution is available.

Monitoring Connection Status

The state or status of a database can be monitored from the **Status** section of the Gateway Webpage, under **Connections > Databases**. The status panels show the current state and a fault message, if applicable, or throughput statistics if the connection is active.

When a connection is not available, it is re-tested every 10 seconds, and the status is updated.

Config > Database > Database Connections

Name	Description	JDBC Driver	Translator	Status		
DB		MariaDB	MYSQL	Faulted	delete	edit
MSSQL		MySQL	MYSQL	Valid	delete	edit
SQLServer		Microsoft SQLServer	MSSQL	Reconnecting	delete	edit

[→ Create new Database Connection...](#)

Note: For details about a connection's status, see the [Database Connection Status](#) page.

Why Use Databases?

Providing access to relational databases is at the heart of the Ignition platform. Ignition can connect to any SQL database that has a JDBC driver, though depending on the database's capabilities, some features may not be available. By connecting Ignition to one or more SQL databases, you can query existing data, update data, store historical information, and more.

Ignition can perform many tasks without the use of a database. For instance, the Vision and OPC-UA modules let you create powerful HMI status and control screens, or Tags can be used to generate alarms that can be sent over email. However, tightly integrated database access is a key feature that makes Ignition stand out from its competitors.

Modern relational databases offer amazing storage and querying capabilities with great performance at a price that is incomparable to older legacy historians. While it is true that historians still have a place in the industry, for most applications relational SQL databases not only suffice, but offer much more than what was previously available. Using SQL, you can store and track production information with ease. However, you can also correlate that data to who was on shift, previous runs, downtime, inventory levels and more, naturally and easily. Make the data available to more people using the Vision module's web-launch clients, or integrate the data directly into your company's internal or external website. SQL databases are at the heart of the web and modern corporate IT systems, and now thanks to Ignition, the plant floor as well.

simple_histo...	accumulator...	ambientHum	ambientTemp	receiverLevel	valveDischar...	valveKing	t_stamp	quality_code
773	76	53.56	93.29	19	1		0 Nov 30, 2015 9:...	192
774	82	54.4	94.55	10	1		0 Nov 30, 2015 9:...	192
775	80	54.97	100	14	1		0 Nov 30, 2015 9:...	192
776	80	54.97	100	14	1		0 Nov 30, 2015 9:...	192
777	84	47.2	99.54	7	1		0 Nov 30, 2015 9:...	192
778	82	44.84	97.14	5	1		0 Nov 30, 2015 9:...	192
779	92	44.15	100	13	1		0 Nov 30, 2015 9:...	192
780	48	34.48	86.49	57	0		1 Nov 30, 2015 1:...	192
781	55	30.07	76.21	50	0		1 Nov 30, 2015 1:...	192

Central Management

Database connections have a lot of rules for connecting and querying them, 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. One massive problem with databases can be setting up a connection because everything is based on IP Addresses and computer names. Since Ignition connects through the Gateway, all clients have access to your databases and you don't have to worry about adding individual clients to the database rules. Just set up the Gateway and launch clients wherever you'd like.

Related Topics ...

- [SQL in Ignition](#)
- [JDBC Drivers and Translators](#)

In This Section ...

