CLOUDERA
Cloudera JDBC Driver for Impala

Important Notice

Cloudera, Inc.
1001 Page Mill Road, Building 2
Palo Alto, CA 94304-1008
info@cloudera.com
US: 1-888-789-1488
Intl: 1-650-843-0595
www.cloudera.com

Release Information

Version: 2.6.23

Date: February 26, 2021

Table of Contents

About the Cloudera JDBC Driver for Impala

The Cloudera JDBC Driver for Impala is used for direct SQL and Impala SQL access to Apache Hadoop / Impala distributions, enabling Business Intelligence (BI), analytics, and reporting on Hadoop / Impala-based data. The driver efficiently transforms an application's SQL query into the equivalent form in Impala SQL, which is a subset of SQL-92. If an application is Impala-aware, then the driver is configurable to pass the query through to the database for processing. The driver interrogates Impala to obtain schema information to present to a SQL-based application. Queries, including joins, are translated from SQL to Impala SQL. For more information about the differences between Impala SQL and SQL, see "Features" on page 24.

The Cloudera JDBC Driver for Impala complies with the JDBC 4.1, and 4.2 data standards. JDBC is one of the most established and widely supported APIs for connecting to and working with databases. At the heart of the technology is the JDBC driver, which connects an application to the database. For more information about JDBC, see *Data Access Standards* on the Simba Technologies website: https://www.simba.com/resources/data-access-standards-glossary.

This guide is suitable for users who want to access data residing within Impala from their desktop environment. Application developers might also find the information helpful. Refer to your application for details on connecting via JDBC.

System Requirements

Each machine where you use the Cloudera JDBC Driver for Impala must have Java Runtime Environment (JRE) installed. The version of JRE that must be installed depends on the version of the JDBC API you are using with the driver. The following table lists the required version of JRE for each provided version of the JDBC API.

JDBC API Version	JRE Version
4.1	7.0 to 11.0
4.2	8.0 to 11.0

The driver is recommended for Impala versions 2.8 through 3.2, CDH versions 6.0 through 6.3, and CDP 7.0 and 7.1.

Cloudera JDBC Driver for Impala Files

The Cloudera JDBC Driver for Impala is delivered in the following ZIP archives, where *[Version]* is the version number of the driver:

- `ImpalaJDBC4_[Version].zip`
- `ImpalaJDBC41_[Version].zip`
- `ImpalaJDBC42_[Version].zip`

The archive contains the driver supporting the JDBC API version indicated in the archive name, as well as release notes and third-party license information. In addition, the required third-party libraries and dependencies are packaged and shared in the driver JAR file in the archive.

Installing and Using the Cloudera JDBC Driver for Impala

To install the Cloudera JDBC Driver for Impala on your machine, extract the files from the appropriate ZIP archive to the directory of your choice.

To access an Impala data store using the Cloudera JDBC Driver for Impala, you need to configure the following:

- The list of driver library files (see "Referencing the JDBC Driver Libraries" on page 8)
- The `Driver` or `DataSource` class (see "Registering the Driver Class" on page 9)
- The connection URL for the driver (see "Building the Connection URL" on page 10)

> **Important:**
>
> The Cloudera JDBC Driver for Impala is a forward-only, read-only driver with no transaction support. Because the driver does not support transactions, auto-commit is always set to **true**.

Referencing the JDBC Driver Libraries

Before you use the Cloudera JDBC Driver for Impala, the JDBC application or Java code that you are using to connect to your data must be able to access the driver JAR files. In the application or code, specify all the JAR files that you extracted from the ZIP archive.

Using the Driver in a JDBC Application

Most JDBC applications provide a set of configuration options for adding a list of driver library files. Use the provided options to include all the JAR files from the ZIP archive as part of the driver configuration in the application. For more information, see the documentation for your JDBC application.

Using the Driver in Java Code

You must include all the driver library files in the class path. This is the path that the Java Runtime Environment searches for classes and other resource files. For more information, see "Setting the Class Path" in the appropriate Java SE Documentation.

For Java SE 7:

- For Windows:
  http://docs.oracle.com/javase/7/docs/technotes/tools/windows/classpath.html
- For Linux and Solaris:
  http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/classpath.html

For Java SE 8:

- For Windows:
  http://docs.oracle.com/javase/8/docs/technotes/tools/windows/classpath.html
- For Linux and Solaris:
  http://docs.oracle.com/javase/8/docs/technotes/tools/unix/classpath.html

Registering the Driver Class

Before connecting to your data, you must register the appropriate class for your application.

The following classes are used to connect the Cloudera JDBC Driver for Impala to Impala data stores:

- The `Driver` classes extend `java.sql.Driver`.
- The `DataSource` classes extend `javax.sql.DataSource` and `javax.sql.ConnectionPoolDataSource`.

The driver supports the following fully-qualified class names (FQCNs) that are independent of the JDBC version:

- `com.cloudera.impala.jdbc.Driver`
- `com.cloudera.impala.jdbc.DataSource`

The following sample code shows how to use the `DriverManager` class to establish a connection for JDBC:

```
private static Connection connectViaDM() throws Exception
{
    Connection connection = null;
    Class.forName(DRIVER_CLASS);
    connection = DriverManager.getConnection(CONNECTION_URL);
    return connection;
}
```

The following sample code shows how to use the `DataSource` class to establish a connection:

```
private static Connection connectViaDS() throws Exception
{
    Connection connection = null;
    Class.forName(DRIVER_CLASS);
    DataSource ds = new com.cloudera.impala.jdbc.DataSource();
    ds.setURL(CONNECTION_URL);
    connection = ds.getConnection();
    return connection;
```

```
}
```

Building the Connection URL

Use the connection URL to supply connection information to the data store that you are accessing. The following is the format of the connection URL for the Cloudera JDBC Driver for Impala, where *[Host]* is the DNS or IP address of the Impala server and *[Port]* is the number of the TCP port that the server uses to listen for client requests:

```
jdbc:impala://[Host]:[Port]
```

> **Note:**
>
> By default, the driver uses port 28000 when `TransportMode` is set to `http`, and 21050 when `TransportMode` is not set or is set to `sasl` or `binary`.

By default, the driver uses the schema named **default**.

You can specify optional settings such as the schema to use or any of the connection properties supported by the driver. For a list of the properties available in the driver, see "Driver Configuration Options" on page 84.

> **Note:**
>
> If you specify a property that is not supported by the driver, then the driver attempts to apply the property as a Impala server-side property for the client session. For more information, see "Configuring Server-Side Properties" on page 21.

The following is the format of a connection URL that specifies some optional settings:

```
jdbc:impala://[Host]:[Port]/[Schema];[Property1]=[Value];
[Property2]=[Value];...
```

For example, to connect to port 18000 on an Impala server installed on the local machine, use a schema named default2, and authenticate the connection using a user name and password, you would use the following connection URL:

```
jdbc:impala://node1.example.com:18000/default2;AuthMech=3;
UID=cloudera;PWD=cloudera
```

> **Important:**
>
> - Properties are case-sensitive.
> - Do not duplicate properties in the connection URL.

Configuring Authentication

The Cloudera JDBC Driver for Impala supports the following authentication mechanisms:

- No Authentication
- Kerberos
- User Name
- User Name And Password
- Single Sign-On (SSO)

You configure the authentication mechanism that the driver uses to connect to Impala by specifying the relevant properties in the connection URL.

For information about configuring the authentication mechanism that Impala uses, see the Impala documentation: http://www.cloudera.com/content/cloudera/en/documentation.html.

For information about the properties you can use in the connection URL, see "Driver Configuration Options" on page 84.

> **Note:**
>
> In addition to authentication, you can configure the driver to connect over SSL. For more information, see "Configuring SSL" on page 19.

Using No Authentication

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see "Building the Connection URL" on page 10.

To configure a connection without authentication:

➢ Set the `AuthMech` property to `0`.

For example:

```
jdbc:impala://localhost:21050;AuthMech=0;
```

Using Kerberos

Kerberos must be installed and configured before you can use this authentication mechanism. For information about configuring and operating Kerberos on Windows, see "Configuring Kerberos Authentication for Windows" on page 14. For other operating systems, see the MIT Kerberos documentation: http://web.mit.edu/kerberos/krb5-latest/doc/.

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see "Building the Connection URL" on page 10.

> **Note:**
>
> The driver also supports Kerberos constrained delegation. For more details on this, see "Using Kerberos Constrained Delegation " on page 18.

To configure default Kerberos authentication:

1. Set the `AuthMech` property to `1`.

2. To use the default realm defined in your Kerberos setup, do not set the `KrbRealm` property.

   If your Kerberos setup does not define a default realm or if the realm of your Impala server is not the default, then set the `KrbRealm` property to the realm of the Impala server.

3. Set the `KrbHostFQDN` property to the fully qualified domain name of the Impala server host.

4. If you are using Kerberos Constrained Delegation, set the `userGSSCredential` property to your Kerberos GSS Credential.

5. Optionally, specify how the driver obtains the Kerberos Subject by setting the `KrbAuthType` property as follows:

   - To configure the driver to automatically detect which method to use for obtaining the Subject, set the `KrbAuthType` property to `0`. Alternatively, do not set the `KrbAuthType` property.

   - Or, to create a LoginContext from a JAAS configuration and then use the Subject associated with it, set the `KrbAuthType` property to `1`.

   - Or, to create a LoginContext from a Kerberos ticket cache and then use the Subject associated with it, set the `KrbAuthType` property to `2`.

   For more detailed information about how the driver obtains Kerberos Subjects based on these settings, see "KrbAuthType" on page 87.

For example, the following connection URL connects to a Impala server with Kerberos enabled, but without SSL enabled:

```
jdbc:impala://node1.example.com:21050;AuthMech=1;
KrbRealm=EXAMPLE.COM;KrbHostFQDN=node1.example.com;
KrbServiceName=impala
```

In this example, Kerberos is enabled for JDBC connections, the Kerberos service principal name is impala/node1.example.com@EXAMPLE.COM, the host name for the data source is node1.example.com, and the server is listening on port 21050 for JDBC connections.

Using User Name

This authentication mechanism requires a user name but does not require a password. The user name labels the session, facilitating database tracking.

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see "Building the Connection URL" on page 10.

To configure User Name authentication:

1. Set the `AuthMech` property to `2`.

2. Set the `UID` property to an appropriate user name for accessing the Impala server.

For example:
```
jdbc:impala://node1.example.com:21050;AuthMech=2;UID=impala
```

Using User Name And Password (LDAP)

This authentication mechanism requires a user name and a password. It is most commonly used with LDAP authentication.

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see "Building the Connection URL" on page 10.

To configure User Name And Password authentication:

1. Set the `AuthMech` property to `3`.

2. Set the `UID` property to an appropriate user name for accessing the Impala server.

3. Set the `PWD` property to the password corresponding to the user name you provided.

For example, the following connection URL connects to a Impala server with LDAP authentication enabled:
```
jdbc:impala://node1.example.com:21050;AuthMech=3;
UID=impala;PWD=cloudera;
```

In this example, user name and password (LDAP) authentication is enabled for JDBC connections, the LDAP user name is impala, the password is cloudera, and the server is listening on port 21050 for JDBC connections.

Using Single Sign-On

Single Sign-On (SSO) is a process that allows network users to access all authorized network resources without having to log in to each resource separately. For example, implementing SSO for users within an organization allows each user to authenticate to Impala without providing a separate set of Impala credentials.

You specify the properties in the connection URL. For more information about the syntax of the connection URL, see "Building the Connection URL" on page 10.

> **Important:**
>
> SSL is required for this authentication method. For more information, see "Configuring SSL" on page 19.

To configure Single Sign-On authentication:

1. Set the `AuthMech` property to `12`.

2. Set the `TransportMode` property to `http`.

3. Optionally, set the `SSOWebServerTimeout` property to the number of seconds that the driver waits before timing out while waiting for a browser response.

For example:

```
jdbc:impala://node1.example.com:28000;AuthMech=12;
SSL=1;TransportMode=http;httpPath=cliservice;SSOWebServerTimeout
=60;
```

Configuring Kerberos Authentication for Windows

You can configure your Kerberos setup so that you use the MIT Kerberos Ticket Manager to get the Ticket Granting Ticket (TGT), or configure the setup so that you can use the driver to get the ticket directly from the Key Distribution Center (KDC). Also, if a client application obtains a Subject with a TGT, it is possible to use that Subject to authenticate the connection.

Downloading and Installing MIT Kerberos for Windows

To download and install MIT Kerberos for Windows 4.0.1:

1. Download the appropriate Kerberos installer:

   - For a 64-bit machine, use the following download link from the MIT Kerberos website: http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-amd64.msi.

   - For a 32-bit machine, use the following download link from the MIT Kerberos website: http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-i386.msi.

> **Note:**
>
> The 64-bit installer includes both 32-bit and 64-bit libraries. The 32-bit installer includes 32-bit libraries only.

2. To run the installer, double-click the `.msi` file that you downloaded.

3. Follow the instructions in the installer to complete the installation process.

4. When the installation completes, click **Finish**.

Using the MIT Kerberos Ticket Manager to Get Tickets

Setting the KRB5CCNAME Environment Variable

You must set the KRB5CCNAME environment variable to your credential cache file.

To set the KRB5CCNAME environment variable:

1. Click **Start** 🌏, then right-click **Computer**, and then click **Properties**.

2. Click **Advanced System Settings**.

3. In the System Properties dialog box, on the **Advanced** tab, click **Environment Variables**.

4. In the Environment Variables dialog box, under the System Variables list, click **New**.

5. In the **New System Variable** dialog box, in the Variable Name field, type **KRB5CCNAME**.

6. In the **Variable Value** field, type the path for your credential cache file. For example, type `C:\KerberosTickets.txt`.

7. Click **OK** to save the new variable.

8. Make sure that the variable appears in the System Variables list.

9. Click **OK** to close the Environment Variables dialog box, and then click **OK** to close the System Properties dialog box.

10. Restart your machine.

Getting a Kerberos Ticket

To get a Kerberos ticket:

1. Click **Start** 🔵, then click **All Programs**, and then click the **Kerberos for Windows (64-bit)** or **Kerberos for Windows (32-bit)** program group.

2. Click **MIT Kerberos Ticket Manager**.

3. In the MIT Kerberos Ticket Manager, click **Get Ticket**.

4. In the Get Ticket dialog box, type your principal name and password, and then click **OK**.

If the authentication succeeds, then your ticket information appears in the MIT Kerberos Ticket Manager.

Authenticating to the Impala Server

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see "Building the Connection URL" on page 10.

To authenticate to the Impala server:

➢ Use a connection URL that has the following properties defined:

- `AuthMech`
- `KrbHostFQDN`
- `KrbRealm`
- `KrbServiceName`

For detailed information about these properties, see "Driver Configuration Options" on page 84

Using the Driver to Get Tickets

Deleting the KRB5CCNAME Environment Variable

To enable the driver to get Ticket Granting Tickets (TGTs) directly, make sure that the KRB5CCNAME environment variable has not been set.

To delete the KRB5CCNAME environment variable:

1. Click the **Start** button 🌐, then right-click **Computer**, and then click **Properties**.

2. Click **Advanced System Settings**.

3. In the System Properties dialog box, click the **Advanced** tab and then click **Environment Variables**.

4. In the Environment Variables dialog box, check if the KRB5CCNAME variable appears in the System variables list. If the variable appears in the list, then select the variable and click **Delete**.

5. Click **OK** to close the Environment Variables dialog box, and then click **OK** to close the System Properties dialog box.

Setting Up the Kerberos Configuration File

To set up the Kerberos configuration file:

1. Create a standard `krb5.ini` file and place it in the `C:\Windows` directory.

2. Make sure that the KDC and Admin server specified in the `krb5.ini` file can be resolved from your terminal. If necessary, modify `C:\Windows\System32\drivers\etc\hosts`.

Setting Up the JAAS Login Configuration File

To set up the JAAS login configuration file:

1. Create a JAAS login configuration file that specifies a keytab file and `doNotPrompt=true`.

   For example:
   ```
   Client {
   com.sun.security.auth.module.Krb5LoginModule required
   useKeyTab=true
   keyTab="PathToTheKeyTab"
   principal="cloudera@CLOUDERA"
   doNotPrompt=true;
   };
   ```

2. Set the `java.security.auth.login.config` system property to the location of the JAAS file.

   For example: `C:\KerberosLoginConfig.ini`.

Authenticating to the Impala Server

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see "Building the Connection URL" on page 10.

To authenticate to the Impala server:

➢ Use a connection URL that has the following properties defined:

- `AuthMech`

- `KrbHostFQDN`

- `KrbRealm`

- `KrbServiceName`

For detailed information about these properties, see "Driver Configuration Options" on page 84.

Using an Existing Subject to Authenticate the Connection

If the client application obtains a Subject with a TGT, then that Subject can be used to authenticate the connection to the server.

To use an existing Subject to authenticate the connection:

1. Create a PrivilegedAction for establishing the connection to the database.

   For example:

```
// Contains logic to be executed as a privileged action
public class AuthenticateDriverAction
implements PrivilegedAction<Void>
{
// The connection, which is established as a
PrivilegedAction
Connection con;

// Define a string as the connection URL
static String ConnectionURL =
"jdbc:impala://192.168.1.1:21050";

/**
* Logic executed in this method will have access to the
* Subject that is used to "doAs". The driver will get
* the Subject and use it for establishing a connection
* with the server.
*/
@Override
public Void run()
{
try
{
// Establish a connection using the connection URL
con = DriverManager.getConnection(ConnectionURL);
}
catch (SQLException e)
{
// Handle errors that are encountered during
```

```
// interaction with the data store
e.printStackTrace();
}
catch (Exception e)
{
// Handle other errors
e.printStackTrace();
}
return null;
}
}
```

2. Run the PrivilegedAction using the existing Subject, and then use the connection.

   For example:

```
// Create the action
AuthenticateDriverAction authenticateAction = new
AuthenticateDriverAction();
// Establish the connection using the Subject for
// authentication.
Subject.doAs(loginConfig.getSubject(), authenticateAction);
// Use the established connection.
authenticateAction.con;
```

Using Kerberos Constrained Delegation

The driver can also be configured to use Kerberos Constrained Delegation. This feature allows a service to obtain service tickets to a restricted list of other services running on specific servers on the network after it has been presented with a service ticket. For more details on the process see: https://technet.microsoft.com/en-ca/library/cc995228.aspx.

The `userGSSCredential` connection property can be used in the connection URL to pass in a GSSCredential object. The following sample code shows how to use the property to pass the GSSCredential into the driver using JDBC 4.1.

```
GSSCredential userCredential = [GSSCredential]
Driver driver = (Driver) Class.forName
("com.cloudera.impala.jdbc.Driver").newInstance();
Properties properties = new Properties();
properties.put("userGSSCredential", userCredential);
Connection conn = driver.connect(

    "jdbc:
    impala
    ://node1.example.com:21050;AuthMech=1;KrbRealm=EXAMPLE.COM;
    KrbHostFQDN=node1.example.com;KrbServiceName=impala"
    ,properties);
```

Configuring SSL

> **Note:**
>
> In this documentation, "SSL" indicates both TLS (Transport Layer Security) and SSL (Secure Sockets Layer). The driver supports industry-standard versions of TLS/SSL.

If you are connecting to an Impala server that has Secure Sockets Layer (SSL) enabled, you can configure the driver to connect to an SSL-enabled socket. When connecting to a server over SSL, the driver uses one-way authentication to verify the identity of the server.

One-way authentication requires a signed, trusted SSL certificate for verifying the identity of the server. You can configure the driver to access a specific TrustStore or KeyStore that contains the appropriate certificate. If you do not specify a TrustStore or KeyStore, then the driver uses the default Java TrustStore named `jssecacerts`. If `jssecacerts` is not available, then the driver uses `cacerts` instead.

You provide this information to the driver in the connection URL. For more information about the syntax of the connection URL, see "Building the Connection URL" on page 10.

To configure SSL:

1. Set the `SSL` property to `1`.

2. If you are not using one of the default Java TrustStores, then do one of the following:

    - Create a TrustStore and configure the driver to use it:

        a. Create a TrustStore containing your signed, trusted server certificate.

        b. Set the `SSLTrustStore` property to the full path of the TrustStore.

        c. Set the `SSLTrustStorePwd` property to the password for accessing the TrustStore.

        > **Note:**
        >
        > If the TrustStore is not a JKS TrustStore, set the `SSLTrustStoreType` property to the correct type. To specify a Java Security API provider, set the `SSLTrustStoreProvider` property.

    - Or, create a KeyStore and configure the driver to use it:

        a. Create a KeyStore containing your signed, trusted server certificate.

        b. Set the `SSLKeyStore` property to the full path of the KeyStore.

        c. Set the `SSLKeyStorePwd` property to the password for accessing the KeyStore.

> **Note:**
>
> If the KeyStore is not a JKS KeyStore, set the `SSLKeyStoreType` property to the correct type. To specify a Java Security API provider, set the `SSLKeyStoreProvider` property.

3. Optionally, to allow the SSL certificate used by the server to be self-signed, set the `AllowSelfSignedCerts` property to `1`.

> **Important:**
>
> When the `AllowSelfSignedCerts` property is set to `1`, SSL verification is disabled. The driver does not verify the server certificate against the trust store, and does not verify if the server's host name matches the common name or subject alternative names in the server certificate.

4. Optionally, to allow the common name of a CA-issued certificate to not match the host name of the Impala server, set the `CAIssuedCertNamesMismatch` property to `1`.

For example, the following connection URL connects to a data source using username and password (LDAP) authentication, with SSL enabled:

```
jdbc:impala://localhost:21050;AuthMech=3;SSL=1;
SSLKeyStore=C:\\Users\\bsmith\\Desktop\\keystore.jks;SSLKeyStore
Pwd=clouderaSSL123;UID=impala;PWD=cloudera123
```

> **Note:**
>
> For more information about the connection properties used in SSL connections, see "Driver Configuration Options" on page 84.

Configuring Server-Side Properties

When connecting to a server that is running Impala 2.0 or later, you can use the driver to apply configuration properties to the server by setting the properties in the connection URL.

> **Important:**
>
> This feature is not supported for earlier versions of Impala, where the SET statement can only be executed from within the Impala shell.

For example, to set the `MEM_LIMIT` query option to 1 GB and the `REQUEST_POOL` query option to `myPool`, you would use a connection URL such as the following:

```
jdbc:impala://localhost:18000/default2;AuthMech=3;
UID=cloudera;PWD=cloudera;MEM_LIMIT=1000000000;REQUEST_
POOL=myPool
```

Configuring Logging

To help troubleshoot issues, you can enable logging in the driver.

> **Important:**
>
> Only enable logging long enough to capture an issue. Logging decreases performance and can consume a large quantity of disk space.
>
> The settings for logging apply to every connection that uses the Cloudera JDBC Driver for Impala, so make sure to disable the feature after you are done using it.

In the connection URL, set the `LogLevel` key to enable logging at the desired level of detail. The following table lists the logging levels provided by the Cloudera JDBC Driver for Impala, in order from least verbose to most verbose.

LogLevel Value	Description
0	Disable all logging.
1	Log severe error events that lead the driver to abort.
2	Log error events that might allow the driver to continue running.
3	Log events that might result in an error if action is not taken.
4	Log general information that describes the progress of the driver.
5	Log detailed information that is useful for debugging the driver.
6	Log all driver activity.

To enable logging:

1. Set the `LogLevel` property to the desired level of information to include in log files.

2. Set the `LogPath` property to the full path to the folder where you want to save log files. To make sure that the connection URL is compatible with all JDBC applications, escape the backslashes (\) in your file path by typing another backslash.

   For example, the following connection URL enables logging level 3 and saves the log files in the `C:\temp` folder:
   ```
   jdbc:impala://localhost:11000;LogLevel=3;LogPath=C:\\temp
   ```

3. To make sure that the new settings take effect, restart your JDBC application and reconnect to the server.

The Cloudera JDBC Driver for Impala produces the following log files in the location specified in the `LogPath` property:

- An `ImpalaJDBC_driver.log` file that logs driver activity that is not specific to a connection.

- An `Impala_connection_[Number].log` file for each connection made to the database, where *[Number]* is a number that identifies each log file. This file logs driver activity that is specific to the connection.

If the `LogPath` value is invalid, then the driver sends the logged information to the standard output stream (`System.out`).

To disable logging:

1. Set the `LogLevel` property to `0`.

2. To make sure that the new setting takes effect, restart your JDBC application and reconnect to the server.

Features

More information is provided on the following features of the Cloudera JDBC Driver for Impala:

- "SQL Translation" on page 24
- "Data Types" on page 24
- "Catalog and Schema Support" on page 25
- "Write-back" on page 25
- "Security and Authentication" on page 26
- "Interfaces and Supported Methods" on page 27
- "Multithreading Support " on page 26

SQL Translation

The Cloudera JDBC Driver for Impala is able to parse queries locally prior to sending them to the Impala server. This feature allows the driver to calculate query metadata without executing the query, support query parameters, and support extra SQL features such as JDBC escape sequences and additional scalar functions that are not available in the Impala-shell tool.

> **Note:**
>
> The driver does not support translation for queries that reference a field contained in a nested column (an ARRAY, MAP, or STRUCT column). To retrieve data from a nested column, make sure that the query is written in valid Impala SQL syntax.

Data Types

The Cloudera JDBC Driver for Impala supports many common data formats, converting between Impala, SQL, and Java data types.

The following table lists the supported data type mappings.

Impala Type	SQL Type	Java Type
ARRAY	VARCHAR	String
BIGINT	BIGINT	java.math.BigInteger
BINARY	VARBINARY	byte[]
BOOLEAN	BOOLEAN	Boolean
CHAR (Available only in CDH 5.2 or later)	CHAR	String

Impala Type	SQL Type	Java Type
DATE	DATE	java.sql.Date
DECIMAL (Available only in CDH 5.1 or later)	DECIMAL	java.math.BigDecimal
DOUBLE (REAL is an alias for DOUBLE)	DOUBLE	Double
FLOAT	REAL	Float
INT	INTEGER	Long
MAP	VARCHAR	String
SMALLINT	SMALLINT	Integer
STRUCT	VARCHAR	String
TIMESTAMP	TIMESTAMP	java.sql.Timestamp
TINYINT	TINYINT	Short
VARCHAR (Available only in CDH 5.2 or later)	VARCHAR	String

Catalog and Schema Support

The Cloudera JDBC Driver for Impala supports both catalogs and schemas to make it easy for the driver to work with various JDBC applications. Since Impala only organizes tables into schemas/databases, the driver provides a synthetic catalog named IMPALA under which all of the schemas/databases are organized. The driver also maps the JDBC schema to the Impala schema/database.

> **Note:**
>
> Setting the `CatalogSchemaSwitch` connection property to `1` will cause Impala catalogs to be treated as schemas in the driver as a restriction for filtering.

Write-back

The Cloudera JDBC Driver for Impala supports translation for the following syntax:

- INSERT
- CREATE
- DROP

The driver also supports translation for UPDATE and DELETE syntax, but only when querying Kudu tables while connected to an Impala server that is running Impala 2.7 or later.

If the statement contains non-standard SQL-92 syntax, then the driver is unable to translate the statement to SQL and instead falls back to using Impala SQL.

Security and Authentication

To protect data from unauthorized access, some Impala data stores require connections to be authenticated with user credentials or the SSL protocol. The Cloudera JDBC Driver for Impala provides full support for these authentication protocols.

> **Note:**
>
> In this documentation, "SSL" indicates both TLS (Transport Layer Security) and SSL (Secure Sockets Layer). The driver supports industry-standard versions of TLS/SSL.

The driver provides mechanisms that allow you to authenticate your connection using the Kerberos protocol, your Impala user name only, or your Impala user name and password. You must use the authentication mechanism that matches the security requirements of the Impala server. For detailed driver configuration instructions, see "Configuring Authentication" on page 11.

Additionally, the driver supports SSL connections with one-way authentication. If the server has an SSL-enabled socket, then you can configure the driver to connect to it.

It is recommended that you enable SSL whenever you connect to a server that is configured to support it. SSL encryption protects data and credentials when they are transferred over the network, and provides stronger security than authentication alone. For detailed configuration instructions, see "Configuring SSL" on page 19.

The SSL version that the driver supports depends on the JVM version that you are using. For information about the SSL versions that are supported by each version of Java, see "Diagnosing TLS, SSL, and HTTPS" on the Java Platform Group Product Management Blog: https://blogs.oracle.com/java-platform-group/entry/diagnosing_tls_ssl_and_https.

> **Note:**
>
> The SSL version used for the connection is the highest version that is supported by both the driver and the server, which is determined at connection time.

Multithreading Support

The Cloudera JDBC Driver for Impala supports multithreaded processing.

Within a given process, the driver enables multiple threads to access different connections concurrently. For each connection, the driver enables multiple threads to access different statements from the connection concurrently.

Interfaces and Supported Methods

The Cloudera JDBC Driver for Impala implements the following JDBC interfaces:

However, the driver does not support every method from these interfaces. For information about whether a specific method is supported by the driver and which version of the JDBC API is the earliest version that supports the method, refer to the following sections.

The driver does not support the following JDBC features:

- Array
- Blob
- Clob
- Ref
- Savepoint
- SQLData
- SQLInput
- SQLOutput
- Struct

CallableStatement

The `CallableStatement` interface extends the `PreparedStatement` interface.

The following table lists the methods that belong to the `CallableStatement` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Impala and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `CallableStatement` interface, see the Java API documentation:
http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/CallableStatement.html.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`Array getArray(int i)`	3.0	No	
`Array getArray(String parameterName)`	3.0	No	
`BigDecimal getBigDecimal(int parameterIndex)`	3.0	Yes	
`BigDecimal getBigDecimal(int parameterIndex, int scale)`	3.0	Yes	Deprecated.
`BigDecimal getBigDecimal(String parameterName)`	3.0	Yes	
`Blob getBlob(int i)`	3.0	No	
`Blob getBlob(String parameterName)`	3.0	No	
`boolean getBoolean(int parameterIndex)`	3.0	Yes	
`boolean getBoolean (String parameterName)`	3.0	Yes	
`byte getByte(int parameterIndex)`	3.0	Yes	
`byte getByte(String parameterName)`	3.0	Yes	
`byte[] getBytes(int parameterIndex)`	3.0	Yes	
`byte[] getBytes(String parameterName)`	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
Clob getClob(int i)	3.0	No	
Clob getClob(String parameterName)	3.0	No	
Date getDate(int parameterIndex)	3.0	Yes	
Date getDate(int parameterIndex, Calendar cal)	3.0	Yes	
Date getDate(String parameterName)	3.0	Yes	
Date getDate(String parameterName, Calendar cal)	3.0	Yes	
double getDouble(int parameterIndex)	3.0	Yes	
double getDouble (String parameterName)	3.0	Yes	
float getFloat(int parameterIndex)	3.0	Yes	
float getFloat(String parameterName)	3.0	Yes	
int getInt(int parameterIndex)	3.0	Yes	
int getInt(String parameterName)	3.0	Yes	
long getLong(int parameterIndex)	3.0	Yes	
long getLong(String parameterName)	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
Reader getNCharacterStream (int parameterIndex)	4.0	No	
Reader getNCharacterStream (String parameterName)	4.0	No	
NClob getNClob(int parameterIndex)	4.0	No	
NClob getNClob(String parameterName)	4.0	No	
String getNString(int parameterIndex)	4.0	No	
String getNString (String parameterName)	4.0	No	
Object getObject(int parameterIndex)	3.0	Yes	
<T> T getObject(int parameterIndex, Class<T> type)	4.1	No	
Object getObject(int i, Map<String,Class<?>> map)	3.0	No	
Object getObject (String parameterName)	3.0	Yes	
<T> T getObject(String parameterName, Class<T> type)	4.1	No	
Object getObject	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
(String parameterName, Map<String,Class<?>> map)			
Ref getRef(int i)	3.0	No	
Ref getRef(String parameterName)	3.0	No	
RowId getRowId(int parameterIndex)	4.0	No	
RowId getRowId(String parameterName)	4.0	No	
short getShort(int parameterIndex)	3.0	Yes	
short getShort(String parameterName)	3.0	Yes	
SQLXML getSQLXML(int parameterIndex)	4.0	No	
SQLXML getSQLXML (String parameterName)	4.0	No	
String getString(int parameterIndex)	3.0	Yes	
String getString (String parameterName)	3.0	Yes	
Time getTime(int parameterIndex)	3.0	Yes	
Time getTime(int parameterIndex, Calendar cal)	3.0	Yes	
Time getTime(String	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
parameterName)			
Time getTime(String parameterName, Calendar cal)	3.0	Yes	
Timestamp getTimestamp (int parameterIndex)	3.0	Yes	
Timestamp getTimestamp (int parameterIndex, Calendar cal)	3.0	Yes	
Timestamp getTimestamp (String parameterName)	3.0	Yes	
Timestamp getTimestamp (String parameterName, Calendar cal)	3.0	Yes	
URL getURL(int parameterIndex)	3.0	No	
URL getURL(String parameterName)	3.0	No	
void registerOutParameter (int parameterIndex, int sqlType)	3.0	Yes	
void registerOutParameter (int parameterIndex, int sqlType, int scale)	3.0	Yes	
void registerOutParameter (int paramIndex, int sqlType, String typeName)	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`void registerOutParameter (String parameterName, int sqlType)`	3.0	Yes	
`void registerOutParameter (String parameterName, int sqlType, int scale)`	3.0	Yes	
`void registerOutParameter (String parameterName, int sqlType, String typeName)`	3.0	Yes	
`void setAsciiStream (String parameterName, InputStream x)`	4.0	Yes	
`void setAsciiStream (String parameterName, InputStream x, int length)`	3.0	Yes	
`void setAsciiStream (String parameterName, InputStream x, long length)`	4.0	Yes	
`void setBigDecimal (String parameterName, BigDecimal x)`	3.0	Yes	
`void setBinaryStream (String parameterName, InputStream x)`	4.0	Yes	
`setBinaryStream(String parameterName, InputStream x, int length)`	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
void setBinaryStream (String parameterName, InputStream x, long length)	4.0	Yes	
void setBlob(String parameterName, Blob x)	4.0	Yes	
void setBlob(String parameterName, InputStream inputStream)	4.0	Yes	
void setBlob(String parameterName, InputStream inputStream, long length)	4.0	Yes	
void setBoolean(String parameterName, boolean x)	3.0	Yes	
void setByte(String parameterName, byte x)	3.0	Yes	
void setBytes(String parameterName, byte[] x)	3.0	Yes	
void setCharacterStream (String parameterName, Reader reader)	4.0	Yes	
void setCharacterStream (String parameterName, Reader reader, int length)	3.0	Yes	
void	4.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
setCharacterStream (String parameterName, Reader reader, long length)			
void setClob(String parameterName, Clob x)	4.0	Yes	
void setClob(String parameterName, Reader reader)	4.0	Yes	
void setClob(String parameterName, Reader reader, long length)	4.0	Yes	
void setDate(String parameterName, Date x)	3.0	Yes	
void setDate(String parameterName, Date x, Calendar cal)	3.0	Yes	
void setDouble(String parameterName, double x)	3.0	Yes	
void setFloat(String parameterName, float x)	3.0	Yes	
void setInt(String parameterName, int x)	3.0	Yes	
void setLong(String parameterName, long x)	3.0	Yes	
void setNCharacterStream (String parameterName, Reader value)	4.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
void setNCharacterStream (String parameterName, Reader value, long length)	4.0	Yes	
void setNClob(String parameterName, NClob value)	4.0	Yes	
void setNClob(String parameterName, Reader reader)	4.0	Yes	
void setNClob(String parameterName, Reader reader, long length)	4.0	Yes	
void setNString(String parameterName, String value)	4.0	Yes	
void setNull(String parameterName, int sqlType)	3.0	Yes	
void setNull(String parameterName, int sqlType, String typeName)	3.0	Yes	
void setObject(String parameterName, Object x)	3.0	Yes	
void setObject(String parameterName, Object x, int targetSqlType)	3.0	Yes	
void setObject(String parameterName, Object x, int targetSqlType, int scale)	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`void setRowId(String parameterName, RowId x)`	4.0	Yes	
`void setShort(String parameterName, short x)`	3.0	Yes	
`void setSQLXML(String parameterName, SQLXML xmlObject)`	4.0	Yes	
`void setString(String parameterName, String x)`	3.0	Yes	
`void setTime(String parameterName, Time x)`	3.0	Yes	
`void setTime(String parameterName, Time x, Calendar cal)`	3.0	Yes	
`void setTimestamp (String parameterName, Timestamp x)`	3.0	Yes	
`void setTimestamp (String parameterName, Timestamp x, Calendar cal)`	3.0	Yes	
`void setURL(String parameterName, URL val)`	3.0	Yes	
`boolean wasNull()`	3.0	Yes	
`boolean isWrapperFor (Class<?> iface)`	4.0	Yes	
`<T> T unwrap(Class<T> iface)`	4.0	Yes	

Connection

The following table lists the methods that belong to the `Connection` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Impala and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `Connection` interface, see the Java API documentation: http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/Connection.html.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`void clearWarnings()`	3.0	Yes	
`void close()`	3.0	Yes	
`void commit()`	3.0	Yes	Auto-commit cannot be set to `false` because it is hard-coded to `true`.
`Array createArrayOf(String typeName, Object[] elements)`	4.0	No	
`Blob createBlob()`	4.0	No	
`Clob createClob()`	4.0	No	
`NClob createNClob()`	4.0	No	
`SQLXML createSQLXML()`	4.0	No	
`Statement createStatement ()`	3.0	Yes	
`Statement createStatement (int resultSetType, int resultSetConcurrency)`	3.0	No	
`Statement createStatement (int resultSetType, int resultSetConcurrency, int resultSetHoldability)`	3.0	No	
`Struct createStruct(String`	4.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`typeName, Object[] attributes)`			
`boolean getAutoCommit()`	3.0	Yes	Hard-coded to `true`.
`String getCatalog()`	3.0	Yes	
`Properties getClientInfo()`	4.0	Yes	
`String getClientInfo (String name)`	4.0	Yes	
`int getHoldability()`	3.0	Yes	Hard-coded to `CLOSE_CURSORS_AT_COMMIT`.
`DatabaseMetaData getMetaData()`	3.0	Yes	
`int getNetworkTimeout()`	4.1	No	
`String getSchema()`	4.1	Yes	The returned schema name does not always match the one used by statements. Statements use the schema name defined in the connection URL.
`int getTransactionIsolation()`	3.0	Yes	Hard-coded to `TRANSACTION_READ_UNCOMMITTED`.
`Map<String,Class<?>> getTypeMap()`	3.0	No	
`SQLWarning getWarnings()`	3.0	Yes	
`boolean isClosed()`	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`boolean isReadOnly()`	3.0	Yes	Returns `true`.
`boolean isValid(int timeout)`	4.0	Yes	
`String nativeSQL(String sql)`	3.0	Yes	
`CallableStatement prepareCall(String sql)`	3.0	No	
`CallableStatement prepareCall(String sql, int resultSetType, int resultSetConcurrency)`	3.0	No	
`CallableStatement prepareCall(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability)`	3.0	No	
`PreparedStatement prepareStatement(String sql)`	3.0	Yes	
`PreparedStatement prepareStatement(String sql, int autoGeneratedKeys)`	3.0	No	
`PreparedStatement prepareStatement(String sql, int[] columnIndexes)`	3.0	No	
`PreparedStatement prepareStatement(String sql, int resultSetType, int resultSetConcurrency)`	3.0	No	
`PreparedStatement prepareStatement(String sql, int resultSetType, int resultSetConcurrency,`	3.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`int resultSetHoldability)`			
`PreparedStatement prepareStatement(String sql, String[] columnNames)`	3.0	No	
`void releaseSavepoint (Savepoint savepoint)`	3.0	No	Savepoints are not available because transactions are not supported.
`void rollback()`	3.0	No	Savepoints are not available because transactions are not supported.
`void rollback(Savepoint savepoint)`	3.0	No	Savepoints are not available because transactions are not supported.
`void setAutoCommit(boolean autoCommit)`	3.0	Yes	Ignored because auto-commit is hard-coded to `true`.
`void setCatalog(String catalog)`	3.0	Yes	
`void setClientInfo (Properties properties)`	4.0	Yes	
`void setClientInfo(String name, String value)`	4.0	Yes	
`void setHoldability(int holdability)`	3.0	Yes	
`void setNetworkTimeout (Executor executor, int milliseconds)`	4.1	No	
`void setReadOnly(boolean`	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
readOnly)			
Savepoint setSavepoint()	3.0	No	Savepoints are not available because transactions are not supported.
Savepoint setSavepoint (String name)	3.0	No	Savepoints are not available because transactions are not supported.
void setSchema(String schema)	4.1	Yes	Does not actually change the schema name used by newly created statements; only changes the value returned by getSchema().
void setTransactionIsolation (int level)	3.0	Yes	
void setTypeMap (Map<String,Class<?>> map)	3.0	No	
boolean isWrapperFor (Class<?> iface)	4.0	Yes	
<T> T unwrap(Class<T> iface)	4.0	Yes	

DatabaseMetaData

The following table lists the methods that belong to the DatabaseMetaData interface, and describes whether each method is supported by the Cloudera JDBC Driver for Impala and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the DatabaseMetaData interface, see the Java API documentation:http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/DatabaseMetaData.html.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`boolean allProceduresAreCallable()`	3.0	Yes	Returns `true`.
`boolean allTablesAreSelectable()`	3.0	Yes	Returns `true`.
`boolean autoCommitFailureClosesAllResultSets()`	4.0	Yes	Returns `true`.
`boolean dataDefinitionCausesTransactionCommit()`	3.0	Yes	Returns `false`.
`boolean dataDefinitionIgnoredInTransactions()`	3.0	Yes	Returns `false`.
`boolean deletesAreDetected(int type)`	3.0	Yes	Returns `true`.
`boolean doesMaxRowSizeIncludeBlobs()`	3.0	Yes	Returns `false`.
`boolean generatedKeyAlwaysReturned()`	4.1	Yes	
`ResultSet getAttributes(String catalog, String schemaPattern, String typeNamePattern, String attributeNamePattern)`	3.0	Yes	
`ResultSet getBestRowIdentifier(String catalog, String schema, String table, int scope, boolean nullable)`	3.0	Yes	
`ResultSet getCatalogs()`	3.0	Yes	
`String getCatalogSeparator()`	3.0	Yes	
`String getCatalogTerm()`	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`ResultSet getClientInfoProperties()`	4.0	Yes	
`ResultSet getColumnPrivileges (String catalog, String schema, String table, String columnNamePattern)`	3.0	Yes	
`ResultSet getColumns(String catalog, String schemaPattern, String tableNamePattern, String columnNamePattern)`	3.0	Yes	
`Connection getConnection()`	3.0	Yes	
`ResultSet getCrossReference(String primaryCatalog, String primarySchema, String primaryTable, String foreignCatalog, String foreignSchema, String foreignTable)`	3.0	Yes	
`int getDatabaseMajorVersion()`	3.0	Yes	
`int getDatabaseMinorVersion()`	3.0	Yes	
`String getDatabaseProductName()`	3.0	Yes	Hard-coded to `Impala.`
`String getDatabaseProductVersion()`	3.0	Yes	
`int getDefaultTransactionIsolation ()`	3.0	Yes	Hard-coded to `TRANSACTION_ READ_ UNCOMMITTED.`
`int getDriverMajorVersion()`	3.0	Yes	
`int getDriverMinorVersion()`	3.0	Yes	
`String getDriverName()`	3.0	Yes	Hard-coded to

Method	Supported Since JDBC Version	Supported by the Driver	Notes
			`ImpalaJDBC.`
`String getDriverVersion()`	3.0	Yes	
`ResultSet getExportedKeys(String catalog, String schema, String table)`	3.0	Yes	
`String getExtraNameCharacters()`	3.0	Yes	Returns an empty String.
`ResultSet getFunctionColumns(String catalog, String schemaPattern, String functionNamePattern, String columnNamePattern)`	4.0	Yes	
`ResultSet getFunctions(String catalog, String schemaPattern, String functionNamePattern)`	4.0	Yes	
`String getIdentifierQuoteString()`	3.0	Yes	Returns a backquote (`` ` ``)
`ResultSet getImportedKeys(String catalog, String schema, String table)`	3.0	Yes	
`ResultSet getIndexInfo(String catalog, String schema, String table, boolean unique, boolean approximate)`	3.0	Yes	
`int getJDBCMajorVersion()`	3.0	Yes	
`int getJDBCMinorVersion()`	3.0	Yes	
`int getMaxBinaryLiteralLength()`	3.0	Yes	Returns `0`.
`int getMaxCatalogNameLength()`	3.0	Yes	Returns `128`.

Method	Suppor ted Since JDBC Versio n	Suppor ted by the Driver	Notes
`int getMaxCharLiteralLength()`	3.0	Yes	Returns 0.
`int getMaxColumnNameLength()`	3.0	Yes	Returns 128.
`int getMaxColumnsInGroupBy()`	3.0	Yes	Returns 0.
`int getMaxColumnsInIndex()`	3.0	Yes	Returns 0.
`int getMaxColumnsInOrderBy()`	3.0	Yes	Returns 0.
`int getMaxColumnsInSelect()`	3.0	Yes	Returns 0.
`int getMaxColumnsInTable()`	3.0	Yes	Returns 0.
`int getMaxConnections()`	3.0	Yes	Returns 0.
`int getMaxCursorNameLength()`	3.0	Yes	Returns 0.
`int getMaxIndexLength()`	3.0	Yes	Returns 0.
`int getMaxProcedureNameLength()`	3.0	Yes	Returns 0.
`int getMaxRowSize()`	3.0	Yes	Returns 0.
`int getMaxSchemaNameLength()`	3.0	Yes	Returns 128.
`int getMaxStatementLength()`	3.0	Yes	Returns 0.
`int getMaxStatements()`	3.0	Yes	Returns 0.
`int getMaxTableNameLength()`	3.0	Yes	Returns 128.
`int getMaxTablesInSelect()`	3.0	Yes	Returns 0.
`int getMaxUserNameLength()`	3.0	Yes	Returns 0.
`String getNumericFunctions()`	3.0	Yes	Returns the Numeric Functions list from the

Method	Suppor ted Since JDBC Versio n	Suppor ted by the Driver	Notes
			specification related to the JDBC version of the driver.
`ResultSet getPrimaryKeys(String catalog, String schema, String table)`	3.0	Yes	
`ResultSet getProcedureColumns (String catalog, String schemaPattern, String procedureNamePattern, String columnNamePattern)`	3.0	Yes	
`ResultSet getProcedures(String catalog, String schemaPattern, String procedureNamePattern)`	3.0	Yes	
`String getProcedureTerm()`	3.0	Yes	Returns `procedure`.
`ResultSet getPseudoColumns(String catalog, String schemaPattern, String tableNamePattern, String columnNamePattern)`	4.1	Yes	
`int getResultSetHoldability()`	3.0	Yes	Returns `CLOSE_ CURSORS_AT_ COMMIT`.
`RowIdLifetime getRowIdLifetime()`	4.0	Yes	Returns `ROWID_ UNSUPPORTED`.
`ResultSet getSchemas()`	3.0	Yes	
`ResultSet getSchemas(String catalog, String schemaPattern)`	4.0	Yes	
`String getSchemaTerm()`	3.0	Yes	Returns schema.
`String getSearchStringEscape()`	3.0	Yes	Returns a backslash

Method	Supported Since JDBC Version	Supported by the Driver	Notes
			(\).
String getSQLKeywords()	3.0	Yes	Returns an empty String.
int getSQLStateType()	3.0	Yes	Returns sqlStateSQL99.
String getStringFunctions()	3.0	Yes	Returns the String Functions list from the specification related to the JDBC version of the driver.
ResultSet getSuperTables(String catalog, String schemaPattern, String tableNamePattern)	3.0	Yes	
ResultSet getSuperTypes(String catalog, String schemaPattern, String typeNamePattern)	3.0	Yes	
String getSystemFunctions()	3.0	Yes	Returns DATABASE,IFNULL,USER.
ResultSet getTablePrivileges(String catalog, String schemaPattern, String tableNamePattern)	3.0	Yes	
ResultSet getTables(String catalog, String schemaPattern, String tableNamePattern, String[] types)	3.0	Yes	
ResultSet getTableTypes()	3.0	Yes	
String getTimeDateFunctions()	3.0	Yes	Returns the Time and Date Functions list from the

Method	Supported Since JDBC Version	Supported by the Driver	Notes
			specification related to the JDBC version of the driver.
`ResultSet getTypeInfo()`	3.0	Yes	
`ResultSet getUDTs(String catalog, String schemaPattern, String typeNamePattern, int[] types)`	3.0	Yes	
`String getURL()`	3.0	Yes	
`String getUserName()`	3.0	Yes	
`ResultSet getVersionColumns(String catalog, String schema, String table)`	3.0	Yes	
`boolean insertsAreDetected(int type)`	3.0	Yes	
`boolean isCatalogAtStart()`	3.0	Yes	
`boolean isReadOnly()`	3.0	Yes	Returns `true`.
`boolean locatorsUpdateCopy()`	3.0	Yes	Returns `false`.
`boolean nullPlusNonNullIsNull()`	3.0	Yes	Returns `true`.
`boolean nullsAreSortedAtEnd()`	3.0	Yes	Returns `false`.
`boolean nullsAreSortedAtStart()`	3.0	Yes	Returns `false`.
`boolean nullsAreSortedHigh()`	3.0	Yes	Returns `false`.
`boolean nullsAreSortedLow()`	3.0	Yes	Returns `true`.
`boolean othersDeletesAreVisible(int type)`	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`boolean othersInsertsAreVisible(int type)`	3.0	Yes	
`boolean othersUpdatesAreVisible(int type)`	3.0	Yes	
`boolean ownDeletesAreVisible(int type)`	3.0	Yes	
`boolean ownInsertsAreVisible(int type)`	3.0	Yes	
`boolean ownUpdatesAreVisible(int type)`	3.0	Yes	
`boolean storesLowerCaseIdentifiers ()`	3.0	Yes	Returns `false`.
`boolean storesLowerCaseQuotedIdentifiers()`	3.0	Yes	Returns `false`.
`boolean storesMixedCaseIdentifiers ()`	3.0	Yes	Returns `true`.
`boolean storesMixedCaseQuotedIdentifiers()`	3.0	Yes	Returns `true`.
`boolean storesUpperCaseIdentifiers ()`	3.0	Yes	Returns `false`.
`boolean storesUpperCaseQuotedIdentifiers()`	3.0	Yes	Returns `false`.
`boolean supportsAlterTableWithAddColumn()`	3.0	Yes	Returns `false`.
`boolean supportsAlterTableWithDropColumn()`	3.0	Yes	Returns `false`.
`boolean supportsANSI92EntryLevelSQL ()`	3.0	Yes	Returns `true`.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`boolean supportsANSI92FullSQL()`	3.0	Yes	Returns `false`.
`boolean supportsANSI92IntermediateSQL()`	3.0	Yes	Returns `false`.
`boolean supportsBatchUpdates()`	3.0	Yes	Returns `false`.
`boolean supportsCatalogsInDataManipulation ()`	3.0	Yes	Returns `true`.
`boolean supportsCatalogsInIndexDefinitions ()`	3.0	Yes	Returns `true`.
`boolean supportsCatalogsInPrivilegeDefiniti ons()`	3.0	Yes	Returns `true`.
`boolean supportsCatalogsInProcedureCalls()`	3.0	Yes	Returns `true`.
`boolean supportsCatalogsInTableDefinitions ()`	3.0	Yes	Returns `true`.
`boolean supportsColumnAliasing()`	3.0	Yes	Returns `true`.
`boolean supportsConvert()`	3.0	Yes	Returns `true`.
`boolean supportsConvert(int fromType, int toType)`	3.0	Yes	
`boolean supportsCoreSQLGrammar()`	3.0	Yes	Returns `true`.
`boolean supportsCorrelatedSubqueries()`	3.0	Yes	Returns `true`.
`boolean supportsDataDefinitionAndDataManipu lationTransactions()`	3.0	Yes	Returns `false`.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`boolean supportsDataManipulationTransactionsOnly()`	3.0	Yes	Returns `false`.
`boolean supportsDifferentTableCorrelationNames()`	3.0	Yes	Returns `false`.
`boolean supportsExpressionsInOrderBy()`	3.0	Yes	Returns `true`.
`boolean supportsExtendedSQLGrammar ()`	3.0	Yes	Returns `false`.
`boolean supportsFullOuterJoins()`	3.0	Yes	Returns `true`.
`boolean supportsGetGeneratedKeys()`	3.0	Yes	Returns `false`.
`boolean supportsGroupBy()`	3.0	Yes	Returns `true`.
`boolean supportsGroupByBeyondSelect ()`	3.0	Yes	Returns `true`.
`boolean supportsGroupByUnrelated()`	3.0	Yes	Returns `false`.
`boolean supportsIntegrityEnhancementFacility()`	3.0	Yes	Returns `false`.
`boolean supportsLikeEscapeClause()`	3.0	Yes	Returns `true`.
`boolean supportsLimitedOuterJoins()`	3.0	Yes	Returns `false`.
`boolean supportsMinimumSQLGrammar()`	3.0	Yes	Returns `true`.
`boolean supportsMixedCaseIdentifiers()`	3.0	Yes	Returns `false`.
`boolean`	3.0	Yes	Returns `true`.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
supportsMixedCaseQuotedIdentifiers ()			
boolean supportsMultipleOpenResults ()	3.0	Yes	Returns false.
boolean supportsMultipleResultSets ()	3.0	Yes	Returns false.
boolean supportsMultipleTransactions()	3.0	Yes	Returns true.
boolean supportsNamedParameters()	3.0	Yes	Returns false.
boolean supportsNonNullableColumns ()	3.0	Yes	Returns false.
boolean supportsOpenCursorsAcrossCommit()	3.0	Yes	Returns false.
boolean supportsOpenCursorsAcrossRollback()	3.0	Yes	Returns false.
boolean supportsOpenStatementsAcrossCommit ()	3.0	Yes	Returns true.
boolean supportsOpenStatementsAcrossRollbac k()	3.0	Yes	Returns true.
boolean supportsOrderByUnrelated()	3.0	Yes	Returns false.
boolean supportsOuterJoins()	3.0	Yes	Returns false.
boolean supportsPositionedDelete()	3.0	Yes	Returns false.
boolean supportsPositionedUpdate()	3.0	Yes	Returns false.
boolean	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`supportsResultSetConcurrency(int type, int concurrency)`			
`boolean supportsResultSetHoldability(int holdability)`	3.0	Yes	
`boolean supportsResultSetType(int type)`	3.0	Yes	
`boolean supportsSavepoints()`	3.0	Yes	Returns `false`.
`boolean supportsSchemasInDataManipulation()`	3.0	Yes	Returns `true`.
`boolean supportsSchemasInIndexDefinitions()`	3.0	Yes	Returns `true`.
`boolean supportsSchemasInPrivilegeDefinitions()`	3.0	Yes	Returns `true`.
`boolean supportsSchemasInProcedureCalls()`	3.0	Yes	Returns `false`.
`boolean supportsSchemasInTableDefinitions()`	3.0	Yes	Returns `true`.
`boolean supportsSelectForUpdate()`	3.0	Yes	Returns `false`.
`boolean supportsStatementPooling()`	3.0	Yes	Returns `false`.
`boolean supportsStoredFunctionsUsingCallSyntax()`	4.0	Yes	Returns `false`.
`boolean supportsStoredProcedures()`	3.0	Yes	Returns `true`.

Method	Suppor ted Since JDBC Versio n	Suppor ted by the Driver	Notes
`boolean supportsSubqueriesInComparisons()`	3.0	Yes	Returns `true`.
`boolean supportsSubqueriesInExists ()`	3.0	Yes	Returns `true`.
`boolean supportsSubqueriesInIns()`	3.0	Yes	Returns `true`.
`boolean supportsSubqueriesInQuantifieds()`	3.0	Yes	Returns `true`.
`boolean supportsTableCorrelationNames()`	3.0	Yes	Returns `true`.
`boolean supportsTransactionIsolationLevel (int level)`	3.0	Yes	
`boolean supportsTransactions()`	3.0	Yes	Returns `false`.
`boolean supportsUnion()`	3.0	Yes	Returns `true`.
`boolean supportsUnionAll()`	3.0	Yes	Returns `true`.
`boolean updatesAreDetected(int type)`	3.0	Yes	Returns `true`.
`boolean usesLocalFilePerTable()`	3.0	Yes	Returns `false`.
`boolean usesLocalFiles()`	3.0	Yes	Returns `false`.
`boolean isWrapperFor(Class<?> iface)`	4.0	Yes	
`<T> T unwrap(Class<T> iface)`	4.0	Yes	

DataSource

The following table lists the methods that belong to the `DataSource` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Impala and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `DataSource` interface, see the Java API documentation: http://docs.oracle.com/javase/1.5.0/docs/api/javax/sql/DataSource.html.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`Connection getConnection()`	3.0	Yes	
`Connection getConnection(String username, String password)`	3.0	Yes	
`int getLoginTimeout()`	3.0	Yes	
`PrintWriter getLogWriter()`	3.0	Yes	
`Logger getParentLogger ()`	4.1	No	The driver does not use `java.util.logging`.
`void setLoginTimeout (int seconds)`	3.0	Yes	
`void setLogWriter (PrintWriter out)`	3.0	Yes	
`boolean isWrapperFor (Class<?> iface)`	4.0	Yes	
`<T> T unwrap(Class<T> iface)`	4.0	Yes	

Driver

The following table lists the methods that belong to the `Driver` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Impala and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `Driver` interface, see the Java API documentation: http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/Driver.html.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`boolean acceptsURL (String url)`	3.0	Yes	
`Connection connect (String url, Properties info)`	3.0	Yes	
`int getMajorVersion()`	3.0	Yes	
`int getMinorVersion()`	3.0	Yes	
`Logger getParentLogger ()`	4.1	No	
`DriverPropertyInfo[] getPropertyInfo(String url, Properties info)`	3.0	Yes	
`boolean jdbcCompliant ()`	3.0	Yes	

ParameterMetaData

The following table lists the methods that belong to the `ParameterMetaData` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Impala and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `ParameterMetaData` interface, see the Java API documentation:
http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/ParameterMetaData.html.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`String getParameterClassName(int param)`	3.0	Yes	
`int getParameterCount()`	3.0	Yes	
`int getParameterMode(int`	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
param)			
int getParameterType(int param)	3.0	Yes	
String getParameterTypeName(int param)	3.0	Yes	
int getPrecision(int param)	3.0	Yes	
int getScale(int param)	3.0	Yes	
int isNullable(int param)	3.0	Yes	
boolean isSigned(int param)	3.0	Yes	
boolean isWrapperFor (Class<?> iface)	4.0	Yes	
<T> T unwrap(Class<T> iface)	4.0	Yes	

PooledConnection

The following table lists the methods that belong to the `PooledConnection` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Impala and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `PooledConnection` interface, see the Java API documentation:
http://docs.oracle.com/javase/1.5.0/docs/api/javax/sql/PooledConnection.html.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
void addConnectionEventListener	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
(ConnectionEventListener listener)			
void addStatementEventListener (StatementEventListener listener)	4.0	Yes	
void close()	3.0	Yes	
Connection getConnection()	3.0	Yes	
void removeConnectionEventList ener (ConnectionEventListener listener)	3.0	Yes	
void removeStatementEventListe ner(StatementEventListener listener)	4.0	Yes	Removes the specified StatementEventListen er from the list of components that will be notified when the driver detects that a PreparedStatement has been closed or is invalid.

PreparedStatement

The PreparedStatement interface extends the Statement interface.

The following table lists the methods that belong to the PreparedStatement interface, and describes whether each method is supported by the Cloudera JDBC Driver for Impala and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the PooledConnection interface, see the Java API documentation:
http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/PreparedStatement.html.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`void addBatch()`	3.0	Yes	
`void clearParameters()`	3.0	Yes	
`boolean execute()`	3.0	Yes	
`ResultSet executeQuery()`	3.0	Yes	
`int executeUpdate()`	3.0	Yes	If an updated row count is not available from the server, the driver returns a row count of -1.
`ResultSetMetaData getMetaData()`	3.0	Yes	
`ParameterMetaData getParameterMetaData()`	3.0	Yes	
`void setArray(int parameterIndex, Array x)`	3.0	No	
`void setAsciiStream(int parameterIndex, InputStream x)`	4.0	Yes	
`void setAsciiStream(int parameterIndex, InputStream x, int length)`	3.0	Yes	
`void setAsciiStream(int parameterIndex, InputStream x, long length)`	4.0	Yes	
`void setBigDecimal(int parameterIndex, BigDecimal x)`	3.0	Yes	
`void setBinaryStream(int`	4.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
parameterIndex, InputStream x)			
void setBinaryStream(int parameterIndex, InputStream x, int length)	3.0	Yes	
void setBinaryStream(int parameterIndex, InputStream x, long length)	4.0	Yes	
void setBlob(int parameterIndex, Blob x)	3.0	No	
void setBlob(int parameterIndex, InputStream inputStream)	4.0	No	
void setBlob(int parameterIndex, InputStream inputStream, long length)	4.0	No	
void setBoolean(int parameterIndex, boolean x)	3.0	Yes	
void setByte(int parameterIndex, byte x)	3.0	Yes	
void setBytes(int parameterIndex, byte[] x)	3.0	Yes	
void setCharacterStream (int parameterIndex, Reader reader)	4.0	Yes	
void setCharacterStream (int parameterIndex, Reader reader, int	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
length)			
void setCharacterStream (int parameterIndex, Reader reader, long length)	4.0	Yes	
void setClob(int parameterIndex, Clob x)	3.0	No	
void setClob(int parameterIndex, Reader reader)	4.0	No	
void setClob(int parameterIndex, Reader reader, long length)	4.0	No	
void setDate(int parameterIndex, Date x)	3.0	Yes	
void setDate(int parameterIndex, Date x, Calendar cal)	3.0	Yes	
void setDouble(int parameterIndex, double x)	3.0	Yes	
void setFloat(int parameterIndex, float x)	3.0	Yes	
void setInt(int parameterIndex, int x)	3.0	Yes	
void setLong(int parameterIndex, long x)	3.0	Yes	
void setNCharacterStream (int parameterIndex, Reader value)	4.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`void setNCharacterStream (int parameterIndex, Reader value, long length)`	4.0	No	
`void setNClob(int parameterIndex, NClob value)`	4.0	No	
`void setNClob(int parameterIndex, Reader reader)`	4.0	No	
`void setNClob(int parameterIndex, Reader reader, long length)`	4.0	No	
`void setNString(int parameterIndex, String value)`	4.0	No	
`void setNull(int paramIndex, int sqlType, String typeName)`	3.0	Yes	
`void setObject(int parameterIndex, Object x)`	3.0	Yes	
`void setObject(int parameterIndex, Object x, int targetSqlType)`	3.0	Yes	
`void setObject(int parameterIndex, Object x, int targetSqlType, int scale)`	3.0	Yes	
`void setRef(int parameterIndex, Ref x)`	3.0	No	
`void setRowId(int parameterIndex, RowId x)`	4.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`void setShort(int parameterIndex, short x)`	3.0	No	
`void setSQLXML(int parameterIndex, SQLXML xmlObject)`	4.0	Yes	
`void setString(int parameterIndex, String x)`	3.0	Yes	
`void setTime(int parameterIndex, Time x)`	3.0	Yes	
`void setTime(int parameterIndex, Time x, Calendar cal)`	3.0	Yes	
`void setTimestamp(int parameterIndex, Timestamp x)`	3.0	Yes	
`void setTimestamp(int parameterIndex, Timestamp x, Calendar cal)`	3.0	Yes	
`void setUnicodeStream (int parameterIndex, InputStream x, int length)`	3.0	Yes	Deprecated.
`void setURL(int parameterIndex, URL x)`	3.0	No	
`boolean isWrapperFor (Class<?> iface)`	4.0	Yes	
`<T> T unwrap(Class<T> iface)`	4.0	Yes	

ResultSet

The following table lists the methods that belong to the `ResultSet` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Impala and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `ResultSet` interface, see the Java API documentation: http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/ResultSet.html.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`boolean absolute(int row)`	3.0	No	
`void afterLast()`	3.0	No	
`void beforeFirst()`	3.0	No	
`void cancelRowUpdates()`	3.0	No	Not valid because the driver is read-only.
`void clearWarnings()`	3.0	Yes	
`void close()`	3.0	Yes	
`void deleteRow()`	3.0	No	Not valid because the driver is read-only.
`int findColumn(String columnName)`	3.0	Yes	
`boolean first()`	3.0	No	
`Array getArray(int i)`	3.0	No	
`Array getArray(String colName)`	3.0	No	
`InputStream getAsciiStream (int columnIndex)`	3.0	Yes	
`InputStream getAsciiStream (String columnName)`	3.0	Yes	
`BigDecimal getBigDecimal (int columnIndex)`	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
BigDecimal getBigDecimal (int columnIndex, int scale)	3.0	Yes	Deprecated.
BigDecimal getBigDecimal (String columnName)	3.0	Yes	
BigDecimal getBigDecimal (String columnName, int scale)	3.0	Yes	Deprecated.
InputStream getBinaryStream(int columnIndex)	3.0	Yes	
InputStream getBinaryStream(String columnName)	3.0	Yes	
Blob getBlob(int i)	3.0	No	
Blob getBlob(String colName)	3.0	No	
boolean getBoolean(int columnIndex)	3.0	Yes	
boolean getBoolean(String columnName)	3.0	Yes	
getByte(int columnIndex)	3.0	Yes	
byte getByte(String columnName)	3.0	Yes	
byte[] getBytes(int columnIndex)	3.0	Yes	
byte[] getBytes(String columnName)	3.0	Yes	
Reader getCharacterStream (int columnIndex)	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
Reader getCharacterStream (String columnName)	3.0	Yes	
Clob getClob(int i)	3.0	No	
Clob getClob(String colName)	3.0	No	
int getConcurrency()	3.0	Yes	
String getCursorName()	3.0	Yes	
Date getDate(int columnIndex)	3.0	Yes	
Date getDate(int columnIndex, Calendar cal)	3.0	Yes	
Date getDate(String columnName)	3.0	Yes	
Date getDate(String columnName, Calendar cal)	3.0	Yes	
double getDouble(int columnIndex)	3.0	Yes	
double getDouble(String columnName)	3.0	Yes	
int getFetchDirection()	3.0	Yes	
int getFetchSize()	3.0	Yes	
float getFloat(int columnIndex)	3.0	Yes	
float getFloat(String columnName)	3.0	Yes	
int getHoldability()	4.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`int getInt(int columnIndex)`	3.0	Yes	
`int getInt(String columnName)`	3.0	Yes	
`long getLong(int columnIndex)`	3.0	Yes	
`long getLong(String columnName)`	3.0	Yes	
`ResultSetMetaData getMetaData()`	3.0	Yes	
`Reader getNCharacterStream (int columnIndex)`	4.0	No	
`Reader getNCharacterStream (String columnLabel`	4.0	No	
`NClob getNClob(int columnIndex)`	4.0	No	
`NClob getNClob(String columnLabel)`	4.0	No	
`String getNString(int columnIndex)`	4.0	No	
`String getNString(String columnLabel)`	4.0	No	
`Object getObject(int columnIndex)`	3.0	Yes	
`<T> T getObject(int columnIndex, Class<T> type)`	4.1	No	
`Object getObject(int i, Map<String,Class<?>> map)`	3.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
Object getObject(String columnName)	3.0	No	
<T> T getObject(String columnName, Class<T> type)	4.1	No	
Object getObject(String colName, Map<String,Class<?>> map)	3.0	Yes	
Ref getRef(int i)	3.0	No	
Ref getRef(String colName)	3.0	No	
int getRow()	3.0	Yes	
RowId getRowId(int columnIndex)	4.0	No	
RowId getRowId(String columnLabel)	4.0	No	
short getShort(int columnIndex)	3.0	Yes	
short getShort(String columnName)	3.0	Yes	
SQLXML getSQLXML(int columnIndex)	4.0	No	
SQLXML getSQLXML(String columnLabel)	4.0	No	
Statement getStatement()	3.0	Yes	
String getString(int columnIndex)	3.0	Yes	
String getString(String columnName)	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`Time getTime(int columnIndex)`	3.0	Yes	
`Time getTime(int columnIndex, Calendar cal)`	3.0	Yes	
`Time getTime(String columnName)`	3.0	Yes	
`Time getTime(String columnName, Calendar cal)`	3.0	Yes	
`Timestamp getTimestamp(int columnIndex)`	3.0	Yes	
`Timestamp getTimestamp(int columnIndex, Calendar cal)`	3.0	Yes	
`Timestamp getTimestamp (String columnName)`	3.0	Yes	
`Timestamp getTimestamp (String columnName, Calendar cal)`	3.0	Yes	
`int getType()`	3.0	Yes	
`InputStream getUnicodeStream(int columnIndex)`	3.0	Yes	Deprecated.
`InputStream getUnicodeStream(String columnName)`	3.0	Yes	Deprecated.
`URL getURL(int columnIndex)`	3.0	No	
`URL getURL(String columnName)`	3.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`SQLWarning getWarnings()`	3.0	Yes	
`void insertRow()`	3.0	No	Not valid because the driver is read-only.
`boolean isAfterLast()`	3.0	Yes	
`boolean isBeforeFirst()`	3.0	Yes	
`boolean isClosed()`	4.0	Yes	
`boolean isFirst()`	3.0	Yes	
`boolean isLast()`	3.0	No	
`boolean last()`	3.0	No	
`void moveToCurrentRow()`	3.0	No	Not valid because the driver is read-only.
`void moveToInsertRow()`	3.0	No	Not valid because the driver is read-only.
`boolean next()`	3.0	Yes	
`boolean previous()`	3.0	No	
`void refreshRow()`	3.0	No	
`boolean relative(int rows)`	3.0	No	
`boolean rowDeleted()`	3.0	Yes	Hard-coded to `false`.
`boolean rowInserted()`	3.0	Yes	Hard-coded to `false`.
`boolean rowUpdated()`	3.0	Yes	Hard-coded to `false`.
`void setFetchDirection(int direction)`	3.0	No	Not valid because the driver is forward-only.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
void setFetchSize(int rows)	3.0	Yes	
void updateArray(int columnIndex, Array x)	3.0	No	
void updateArray(String columnName, Array x)	3.0	No	
void updateAsciiStream(int columnIndex, InputStream x)	4.0	No	Not valid because the driver is read-only.
void updateAsciiStream(int columnIndex, InputStream x, int length)	3.0	No	Not valid because the driver is read-only.
void updateAsciiStream(int columnIndex, InputStream x, long length)	4.0	No	Not valid because the driver is read-only.
void updateAsciiStream (String columnName, InputStream x)	4.0	No	Not valid because the driver is read-only.
void updateAsciiStream (String columnName, InputStream x, int length)	3.0	No	Not valid because the driver is read-only.
void updateAsciiStream (String columnName, InputStream x, long length)	4.0	No	Not valid because the driver is read-only.
void updateBigDecimal(int columnIndex, BigDecimal x)	3.0	No	Not valid because the driver is read-only.
void updateBigDecimal (String columnName, BigDecimal x)	3.0	No	Not valid because the driver is read-only.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`void updateBinaryStream` `(int columnIndex,` `InputStream x)`	4.0	No	Not valid because the driver is read-only.
`void updateBinaryStream` `(int columnIndex,` `InputStream x, int length)`	3.0	No	Not valid because the driver is read-only.
`void updateBinaryStream` `(int columnIndex,` `InputStream x, long` `length)`	4.0	No	Not valid because the driver is read-only.
`void updateBinaryStream` `(String columnName,` `InputStream x)`	4.0	No	Not valid because the driver is read-only.
`void updateBinaryStream` `(String columnName,` `InputStream x, int length)`	3.0	No	Not valid because the driver is read-only.
`void updateBinaryStream` `(String columnName,` `InputStream x, long` `length)`	4.0	No	Not valid because the driver is read-only.
`void updateBlob(int` `columnIndex, InputStream` `inputStream)`	4.0	No	
`void updateBlob(int` `columnIndex, Blob x)`	3.0	No	
`void updateBlob(int` `columnIndex, InputStream` `inputStream, long length)`	4.0	No	
`void updateBlob(String` `columnName, InputStream` `inputStream)`	4.0	No	
`void updateBlob(String`	3.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
columnName, Blob x)			
void updateBlob(String columnLabel, InputStream inputStream, long length)	4.0	No	
void updateBoolean(int columnIndex, boolean x)	3.0	No	Not valid because the driver is read-only.
void updateBoolean(String columnName, boolean x)	3.0	No	Not valid because the driver is read-only.
void updateByte(int columnIndex, byte x)	3.0	No	Not valid because the driver is read-only.
void updateByte(String columnName, byte x)	3.0	No	Not valid because the driver is read-only.
void updateBytes(int columnIndex, byte[] x)	3.0	No	Not valid because the driver is read-only.
void updateBytes(String columnName, byte[] x)	3.0	No	Not valid because the driver is read-only.
void updateCharacterStream (int columnIndex, Reader x, int length)	3.0	No	Not valid because the driver is read-only.
void updateCharacterStream (String columnName, Reader reader, int length)	3.0	No	Not valid because the driver is read-only.
void updateBlob(int columnIndex, InputStream inputStream)	4.0	No	
void updateClob(int columnIndex, Clob x)	3.0	No	
void updateBlob(int columnIndex, InputStream inputStream, long length)	4.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`void updateBlob(String columnName, InputStream inputStream)`	4.0	No	
`void updateClob(String columnName, Clob x)`	3.0	No	
`void updateBlob(String columnName, InputStream inputStream, long length)`	4.0	No	
`void updateDate(int columnIndex, Date x)`	3.0	No	Not valid because the driver is read-only.
`void updateDate(String columnName, Date x)`	3.0	No	Not valid because the driver is read-only.
`void updateDouble(int columnIndex, double x)`	3.0	No	Not valid because the driver is read-only.
`void updateDouble(String columnName, double x)`	3.0	No	Not valid because the driver is read-only.
`void updateFloat(int columnIndex, float x)`	3.0	No	Not valid because the driver is read-only.
`void updateFloat(String columnName, float x)`	3.0	No	Not valid because the driver is read-only.
`void updateInt(int columnIndex, int x)`	3.0	No	Not valid because the driver is read-only.
`void updateInt(String columnName, int x)`	3.0	No	Not valid because the driver is read-only.
`void updateLong(int columnIndex, long x)`	3.0	No	Not valid because the driver is read-only.
`void updateLong(String columnName, long x)`	3.0	No	Not valid because the driver is read-only.
`void `	4.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`updateNCharacterStream(int columnIndex, Reader x)`			
`void updateNCharacterStream(int columnIndex, Reader x, long length)`	4.0	No	
`void updateNCharacterStream (String columnName, Reader reader)`	4.0	No	
`void updateNCharacterStream (String columnName, Reader reader, long length)`	4.0	No	
`void updateNClob(int columnIndex, NClob nClob)`	4.0	No	
`void updateNClob(int columnIndex, Reader reader)`	4.0	No	
`void updateNClob(int columnIndex, Reader reader, long length)`	4.0	No	
`void updateNClob(String columnName, NClob nClob)`	4.0	No	
`void updateNClob(String columnName, Reader reader)`	4.0	No	
`void updateNClob(String columnName, Reader reader, long length)`	4.0	No	
`void updateNString(int columnIndex, String nString)`	4.0	No	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
void updateNString(String columnName, String nString)	4.0	No	
void updateNull(int columnIndex)	3.0	No	Not valid because the driver is read-only.
void updateNull(String columnName)	3.0	No	Not valid because the driver is read-only.
void updateObject(int columnIndex, Object x)	3.0	No	Not valid because the driver is read-only.
void updateObject(int columnIndex, Object x, int scale)	3.0	No	Not valid because the driver is read-only.
void updateObject(String columnName, Object x)	3.0	No	Not valid because the driver is read-only.
void updateObject(String columnName, Object x, int scale)	3.0	No	Not valid because the driver is read-only.
void updateRef(int columnIndex, Ref x)	3.0	No	Not valid because the driver is read-only.
void updateRef(String columnName, Ref x)	3.0	No	Not valid because the driver is read-only.
void updateRow()	3.0	No	Not valid because the driver is read-only.
void updateRowId(int columnIndex, RowId x)	4.0	No	
void updateRowId(String columnName, RowId x)	4.0	No	
void updateShort(int columnIndex, short x)	3.0	No	Not valid because the driver is read-only.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
void updateShort(String columnName, short x)	3.0	No	Not valid because the driver is read-only.
void updateSQLXML(int columnIndex, SQLXML xmlObject)	4.0	No	
void updateSQLXML(String columnName, SQLXML xmlObject)	4.0	No	
void updateString(int columnIndex, String x)	3.0	No	Not valid because the driver is read-only.
void updateString(String columnName, String x)	3.0	No	Not valid because the driver is read-only.
void updateTime(int columnIndex, Time x)	3.0	No	Not valid because the driver is read-only.
void updateTime(String columnName, Time x)	3.0	No	Not valid because the driver is read-only.
void updateTimestamp(int columnIndex, Timestamp x)	3.0	No	Not valid because the driver is read-only.
void updateTimestamp (String columnName, Timestamp x)	3.0	No	Not valid because the driver is read-only.
boolean wasNull()	3.0	Yes	
boolean isWrapperFor (Class<?> iface)	4.0	Yes	
<T> T unwrap(Class<T> iface)	4.0	Yes	

ResultSetMetaData

The following table lists the methods that belong to the ResultSetMetaData interface, and describes whether each method is supported by the Cloudera JDBC Driver for Impala and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `ResultSetMetaData` interface, see the Java API documentation:
http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/ResultSetMetaData.html.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`String getCatalogName (int column)`	3.0	Yes	
`String getColumnClassName(int column)`	3.0	Yes	
`int getColumnCount()`	3.0	Yes	
`int getColumnDisplaySize (int column)`	3.0	Yes	
`String getColumnLabel (int column)`	3.0	Yes	
`String getColumnName (int column)`	3.0	Yes	
`int getColumnType(int column)`	3.0	Yes	
`String getColumnTypeName(int column)`	3.0	Yes	
`int getPrecision(int column)`	3.0	Yes	
`int getScale(int column)`	3.0	Yes	
`String getSchemaName (int column)`	3.0	Yes	
`String getTableName (int column)`	3.0	Yes	
`boolean isAutoIncrement(int`	3.0	Yes	

Method	Supported Since JDBC Version	Supported by the Driver	Notes
column)			
boolean isCaseSensitive(int column)	3.0	Yes	
boolean isCurrency(int column)	3.0	Yes	
boolean isDefinitelyWritable (int column)	3.0	Yes	
int isNullable(int column)	3.0	Yes	
boolean isReadOnly(int column)	3.0	Yes	
boolean isSearchable (int column)	3.0	Yes	
boolean isSigned(int column)	3.0	Yes	
boolean isWritable(int column)	3.0	Yes	
boolean isWrapperFor (Class<?> iface)	4.0	Yes	
<T> T unwrap(Class<T> iface)	4.0	Yes	

Statement

The following table lists the methods that belong to the `Statement` interface, and describes whether each method is supported by the Cloudera JDBC Driver for Impala and which version of the JDBC API is the earliest version that supports the method.

For detailed information about each method in the `Statement` interface, see the Java API documentation: http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/Statement.html.

Method	Supported Since JDBC Version	Supported by the Driver	Notes
`void addBatch(String sql)`	3.0	Yes	
`void cancel()`	3.0	Yes	
`void clearBatch()`	3.0	Yes	
`void clearWarnings()`	3.0	Yes	
`void close()`	3.0	Yes	
`void closeOnCompletion()`	4.1	Yes	
`boolean execute(String sql)`	3.0	Yes	
`boolean execute(String sql, int autoGeneratedKeys)`	3.0	No	
`boolean execute(String sql, int[] columnIndexes)`	3.0	No	
`boolean execute(String sql, String[] columnNames)`	3.0	No	
`int[]executeBatch()`	3.0	No	
`ResultSet executeQuery (String sql)`	3.0	Yes	
`int executeUpdate(String sql)`	3.0	Yes	If an updated row count is not available from the server, the driver returns a row count of -1.
`int executeUpdate(String sql, int autoGeneratedKeys)`	3.0	No	If an updated row count is not available from the server, the driver returns a row count of -1.
`int executeUpdate(String`	3.0	No	If an updated row

Method	Supported Since JDBC Version	Supported by the Driver	Notes
sql, int[] columnIndexes)			count is not available from the server, the driver returns a row count of -1.
int executeUpdate(String sql, String[] columnNames)	3.0	No	If an updated row count is not available from the server, the driver returns a row count of -1.
Connection getConnection()	3.0	Yes	
int getFetchDirection()	3.0	Yes	
int getFetchSize()	3.0	Yes	
ResultSet getGeneratedKeys ()	3.0	Yes	
int getMaxFieldSize()	3.0	Yes	
int getMaxRows()	3.0	Yes	
boolean getMoreResults()	3.0	Yes	
boolean getMoreResults(int current)	3.0	No	
int getQueryTimeout()	3.0	Yes	
ResultSet getResultSet()	3.0	Yes	
int getResultSetConcurrency ()	3.0	Yes	Hard-coded to CONCUR_READ_ ONLY.
int getResultSetHoldability ()	3.0	Yes	Hard-coded to CLOSE_CURSORS_ AT_COMMIT.
int getResultSetType()	3.0	Yes	Hard-coded to TYPE_

Method	Supported Since JDBC Version	Supported by the Driver	Notes
			FORWARD_ONLY.
`int getUpdateCount()`	3.0	Yes	
`SQLWarning getWarnings()`	3.0	Yes	
`boolean isClosed()`	4.0	Yes	
`boolean isCloseOnCompletion ()`	4.1	Yes	
`boolean isPoolable()`	4.0	Yes	
`void setCursorName(String name)`	3.0	No	
`void setEscapeProcessing (boolean enable)`	3.0	Yes	
`void setFetchDirection(int direction)`	3.0	No	
`void setFetchSize(int rows)`	3.0	Yes	
`void setMaxFieldSize(int max)`	3.0	Yes	
`void setMaxRows(int max)`	3.0	Yes	
`void setPoolable(boolean poolable)`	4.0	Yes	
`void setQueryTimeout(int seconds)`	3.0	Yes	
`boolean isWrapperFor (Class<?> iface)`	4.0	Yes	
`<T> T unwrap(Class<T> iface)`	4.0	Yes	

Driver Configuration Options

Driver Configuration Options lists and describes the properties that you can use to configure the behavior of the Cloudera JDBC Driver for Impala.

You can set configuration properties using the connection URL. For more information, see "Building the Connection URL" on page 10.

> **Note:**
>
> Property names and values are case-sensitive.

AllowSelfSignedCerts

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies whether the driver allows the server to use self-signed SSL certificates.

- `1`: The driver allows self-signed certificates.

> **Important:**
>
> When this property is set to `1`, SSL verification is disabled. The driver does not verify the server certificate against the trust store, and does not verify if the server's host name matches the common name in the server certificate.

- `0`: The driver does not allow self-signed certificates.

> **Note:**
>
> This property is applicable only when SSL connections are enabled.

AsyncExecPollInterval

Default Value	Data Type	Required
10	Integer	No

Description

The time in milliseconds between each poll for the asynchronous query execution status.

"Asynchronous" refers to the fact that the RPC call used to execute a query against Impala is asynchronous. It does not mean that JDBC asynchronous operations are supported.

AuthMech

Default Value	Data Type	Required
Depends on the `transportMode` setting. For more information, see "TransportMode" on page 95.	Integer	No

Description

The authentication mechanism to use. Set the property to one of the following values:

- `0` for No Authentication.
- `1` for Kerberos.
- `2` for User Name.
- `3` for User Name And Password.
- `12` for Single Sign-On

CAIssuedCertsMismatch

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies whether the driver requires the name of the CA-issued SSL certificate to match the host name of the Impala server.

- `0`: The driver requires the names to match.
- `1`: The driver allows the names to mismatch.

> **Note:**
>
> This property is applicable only when SSL connections are enabled.

CatalogSchemaSwitch

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies whether the driver treats Impala catalogs as schemas or as catalogs.

- `1`: The driver treats Impala catalogs as schemas as a restriction for filtering.
- `0`: Impala catalogs are treated as catalogs, and Impala schemas are treated as schemas.

DefaultStringColumnLength

Default Value	Data Type	Required
255	Integer	No

Description

The maximum number of characters that can be contained in STRING columns. The range of `DefaultStringColumnLength` is `0` to `32767`.

By default, the columns metadata for Impala does not specify a maximum data length for STRING columns.

DelegationUID

Default Value	Data Type	Required
None	String	No

Description

Use this option to delegate all operations against Impala to a user that is different than the authenticated user for the connection.

httpPath

Default Value	Data Type	Required
None	String	Yes, if `transportMode=http`.

Description

The partial URL corresponding to the Impala server.

The driver forms the HTTP address to connect to by appending the `httpPath` value to the host and port specified in the connection URL. For example, to connect to the HTTP address `http://localhost:10002/cliservice`, you would use the following connection URL:

```
jdbc:impala://localhost:10002;AuthMech=3;transportMode=http;
httpPath=cliservice;UID=jsmith;PWD=cloudera123;
```

> **Note:**
>
> By default, Impala servers use `cliservice` as the partial URL.

KrbAuthType

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies how the driver obtains the Subject for Kerberos authentication.

- `0`: The driver automatically detects which method to use for obtaining the Subject:

    1. First, the driver tries to obtain the Subject from the current thread's inherited AccessControlContext. If the AccessControlContext contains multiple Subjects, the driver uses the most recent Subject.

    2. If the first method does not work, then the driver checks the `java.security.auth.login.config` system property for a JAAS configuration. If a JAAS configuration is specified, the driver uses that information to create a LoginContext and then uses the Subject associated with it.

    3. If the second method does not work, then the driver checks the KRB5_CONFIG and KRB5CCNAME system environment variables for a Kerberos ticket cache. The driver uses the information from the cache to create a LoginContext and then uses the Subject associated with it.

- `1`: The driver checks the `java.security.auth.login.config` system property for a JAAS configuration. If a JAAS configuration is specified, the driver uses that information to create a LoginContext and then uses the Subject associated with it.

- `2`: The driver checks the KRB5_CONFIG and KRB5CCNAME system environment variables for a Kerberos ticket cache. The driver uses the information from the cache to create a LoginContext and then uses the Subject associated with it.

KrbHostFQDN

Default Value	Data Type	Required
None	String	Yes, if `AuthMech=1`.

Description

The fully qualified domain name of the Impala host.

KrbRealm

Default Value	Data Type	Required
Depends on your Kerberos configuration	String	No

Description

The realm of the Impala host.

If your Kerberos configuration already defines the realm of the Impala host as the default realm, then you do not need to configure this property.

KrbServiceName

Default Value	Data Type	Required
None	String	Yes, if `AuthMech=1`.

Description

The Kerberos service principal name of the Impala server.

LogLevel

Default Value	Data Type	Required
0	Integer	No

Description

Use this property to enable or disable logging in the driver and to specify the amount of detail included in log files.

> **Important:**
>
> Only enable logging long enough to capture an issue. Logging decreases performance and can consume a large quantity of disk space.
>
> The settings for logging apply to every connection that uses the Cloudera JDBC Driver for Impala, so make sure to disable the feature after you are done using it.

Set the property to one of the following numbers:

- `0`: Disable all logging.
- `1`: Enable logging on the FATAL level, which logs very severe error events that will lead the driver to abort.

- `2`: Enable logging on the ERROR level, which logs error events that might still allow the driver to continue running.

- `3`: Enable logging on the WARNING level, which logs events that might result in an error if action is not taken.

- `4`: Enable logging on the INFO level, which logs general information that describes the progress of the driver.

- `5`: Enable logging on the DEBUG level, which logs detailed information that is useful for debugging the driver.

- `6`: Enable logging on the TRACE level, which logs all driver activity.

When logging is enabled, the driver produces the following log files in the location specified in the `LogPath` property:

- An `ImpalaJDBC_driver.log` file that logs driver activity that is not specific to a connection.

- An `Impala_connection_[Number].log` file for each connection made to the database, where *[Number]* is a number that identifies each log file. This file logs driver activity that is specific to the connection.

If the `LogPath` value is invalid, then the driver sends the logged information to the standard output stream (`System.out`).

LogPath

Default Value	Data Type	Required
The current working directory	String	No

Description

The full path to the folder where the driver saves log files when logging is enabled.

> **Note:**
>
> To make sure that the connection URL is compatible with all JDBC applications, it is recommended that you escape the backslashes (\) in your file path by typing another backslash.

LowerCaseResultSetColumnName

Default Value	Data Type	Required
1	Integer	No

Description

This property specifies the letter case that the driver uses when returning the column name aliases in the `ResultSetMetadata`.

- `1`: The column name aliases in the `ResultSetMetadata` are returned in lower-case characters, matching the server-side behavior.

- `0`: The column name aliases are returned in the same letter case as specified in the query.

OptimizedInsert

Default Value	Data Type	Required
1	Integer	No

Description

This property specifies whether the driver tries to optimize INSERT statements by bypassing translation.

Each time the driver translates an INSERT statement, it executes the DESCRIBE command to identify the data types of the columns that it is inserting data into. These additional commands consume resources and might reduce driver performance.

- `1`: The driver tries to optimize INSERT statements by bypassing translation and using other methods to identify column types.

- `0`: The driver does not attempt the optimization, and translates INSERT statements normally.

> **Note:**
>
> If the optimization fails, the driver falls back to translating INSERT statements normally. This additional overhead might further reduce driver performance.

PreparedMetaLimitZero

Default Value	Data Type	Required
1	Integer	No

Description

This property specifies whether the `PreparedStatement.getMetadata()` call will request metadata from the server with `LIMIT 0`, increasing performance.

- `1`: The `PreparedStatement.getMetadata()` call uses `LIMIT 0`.

- `0`: The `PreparedStatement.getMetadata()` call does not use `LIMIT 0`.

PWD

Default Value	Data Type	Required
None	String	Yes, if `AuthMech=3`.

Description

The password corresponding to the user name that you provided using the property "UID" on page 96.

RowsFetchedPerBlock

Default Value	Data Type	Required
`10000`	Integer	No

Description

The maximum number of rows that a query returns at a time.

Any positive 32-bit integer is a valid value, but testing has shown that performance gains are marginal beyond the default value of 10000 rows.

SocketTimeout

Default Value	Data Type	Required
0	Integer	No

Description

The number of seconds that the TCP socket waits for a response from the server before raising an error on the request.

When this property is set to `0`, the connection does not time out.

SSL

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies whether the driver communicates with the Impala server through an SSL-enabled socket.

- `1`: The driver connects to SSL-enabled sockets.
- `2`: The driver connects to SSL-enabled sockets using two-way authentication.
- `0`: The driver does not connect to SSL-enabled sockets.

> **Note:**
>
> SSL is configured independently of authentication. When authentication and SSL are both enabled, the driver performs the specified authentication method over an SSL connection.

SSLKeyStore

Default Value	Data Type	Required
None	String	No

Description

The full path of the Java KeyStore containing the server certificate for one-way SSL authentication.

See also the property "SSLKeyStorePwd" on page 92.

> **Note:**
>
> The Cloudera JDBC Driver for Impala accepts TrustStores and KeyStores for one-way SSL authentication. See also the property "SSLTrustStore" on page 93.

SSLKeyStoreProvider

Default Value	Data Type	Required
None	String	No

Description

The provider of the Java Security API for the KeyStore that is being used for one-way SSL authentication.

SSLKeyStorePwd

Default Value	Data Type	Required
None	Integer	Yes, if you are using a KeyStore for connecting over SSL.

Description

The password for accessing the Java KeyStore that you specified using the property "SSLKeyStore" on page 92.

SSLKeyStoreType

Default Value	Data Type	Required
JKS	String	No

Description

The type of Java KeyStore that is being used for one-way SSL authentication.

SSLTrustStoreProvider

Default Value	Data Type	Required
None	String	No

Description

The provider of the Java Security API for the TrustStore that is being used for one-way SSL authentication.

SSLTrustStore

Default Value	Data Type	Required
`jssecacerts`, if it exists. If `jssecacerts` does not exist, then `cacerts` is used. The default location of `cacerts` is `jre\lib\security\`.	String	No

Description

The full path of the Java TrustStore containing the server certificate for one-way SSL authentication.

If the trust store requires a password, provide it using the property `SSLTrustStorePwd`. See "SSLTrustStorePwd" on page 94.

> **Note:**
>
> The Cloudera JDBC Driver for Impala accepts TrustStores and KeyStores for one-way SSL authentication. See also the property "SSLKeyStore" on page 92.

SSLTrustStorePwd

Default Value	Data Type	Required
None	String	Yes, if using a TrustStore.

Description

The password for accessing the Java TrustStore that you specified using the property "SSLTrustStore" on page 93.

SSLTrustStoreType

Default Value	Data Type	Required
JKS	String	No

Description

The type of Java TrustStore that is being used for one-way SSL authentication.

SSOWebServerTimeout

Default Value	Data Type	Required
120	Integer	No

Description

This property specifies the number of seconds that the driver waits before timing out while waiting for a browser response when authenticating using Single Sign-On (SSO).

If this property is set to `0`, the driver will wait for an indefinite amount of time

StripCatalogName

Default Value	Data Type	Required
1	Integer	No

Description

This property specifies whether the driver removes catalog names from query statements if translation fails or if the `UseNativeQuery` property is set to `1`.

- `1`: If query translation fails or if the `UseNativeQuery` property is set to `1`, then the driver removes catalog names from the query statement.
- `0`: The driver does not remove catalog names from query statements.

SupportTimeOnlyTimestamp

Default Value	Data Type	Required
1	Integer	No

Description

This property specifies whether the driver supports TIMESTAMP data that only contains a time value.

- `1`: The driver supports TIMESTAMP data that only contains a time value.
- `0`: The driver returns an error when working with TIMESTAMP data that only contains a time value.

TransportMode

Default Value	Data Type	Required
sasl	String	No

Description

The transport protocol to use in the Thrift layer.

- `binary`: The driver uses the Binary transport protocol.

  If you use this setting and do not specify the `AuthMech` property, then the driver uses `AuthMech=0` by default. This setting is valid only when the `AuthMech` property is set to `0` or `3`.

- `sasl`: The driver uses the SASL transport protocol.

  If you use this setting but do not specify the `AuthMech` property, then the driver uses `AuthMech=2` by default. This setting is valid only when the `AuthMech` property is set to `1`, `2`, or `3`.

- `http`: The driver uses the HTTP transport protocol.

  If you use this setting but do not specify the `AuthMech` property, then the driver uses `AuthMech=0` by default. This setting is valid only when the `AuthMech` property is set to `0`, `3`, or `12`.

> **Note:**
>
> This option replaces and supersedes the deprecated `UseSasl` option.

UID

Default Value	Data Type	Required
anonymous	String	Yes, if `AuthMech=3`.

Description

The user name that you use to access the Impala server.

UpperCaseResultSetColName

Default Value	Data Type	Required
false	Boolean	No

Description

This property specifies whether the driver converts the result set column name to upper case if translation fails or if the `UseNativeQuery` property is set to 1.

- `true`: If query translation fails or if the `UseNativeQuery` property is set to 1, the driver converts the result set column names to upper-case characters.
- `false`: The driver does not convert the result set column names to upper-case characters.

UseNativeQuery

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies whether the driver transforms the queries emitted by applications.

- `1`: The driver does not transform the queries emitted by applications, so the native query is used.
- `0`: The driver transforms the queries emitted by applications and converts them into an equivalent form in Impala SQL.

> **Note:**
>
> If the application is Impala-aware and already emits Impala SQL, then enable this option to avoid the extra overhead of query transformation.

UseSasl (deprecated)

Default Value	Data Type	Required
1	Integer	No

Description

This option is deprecated. Use `TransportMode` instead (see "TransportMode" on page 95).

This property indicates if SASL is used in conjunction with the User Name and Password Authentication Mechanism (`AuthMech=3`).

- `0`: No SASL authentication is used. User credentials are still passed to the server for services such as Sentry.
- `1`: SASL authentication is used.

Contact Us

If you are having difficulties using the driver, our [Community Forum](#) may have your solution. In addition to providing user to user support, our forums are a great place to share your questions, comments, and feature requests with us.

If you are a Subscription customer you may also use the [Cloudera Support Portal](#) to search the Knowledge Base or file a Case.

> **Important:**
>
> To help us assist you, prior to contacting Cloudera Support please prepare a detailed summary of the client and server environment including operating system version, patch level, and configuration.