

Cloudera JDBC Driver for Impala Version 2.5.16



Important Notice

© 2010-2015 Cloudera, Inc. All rights reserved.

Cloudera, the Cloudera logo, Cloudera Impala, Impala, and any other product or service names or slogans contained in this document, except as otherwise disclaimed, are trademarks of Cloudera and its suppliers or licensors, and may not be copied, imitated or used, in whole or in part, without the prior written permission of Cloudera or the applicable trademark holder.

Hadoop and the Hadoop elephant logo are trademarks of the Apache Software Foundation. All other trademarks, registered trademarks, product names and company names or logos mentioned in this document are the property of their respective owners. Reference to any products, services, processes or other information, by trade name, trademark, manufacturer, supplier or otherwise does not constitute or imply endorsement, sponsorship or recommendation thereof by us.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Cloudera.

Cloudera may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Cloudera, the furnishing of this document does not give you any license to these patents, trademarks copyrights, or other intellectual property.

The information in this document is subject to change without notice. Cloudera shall not be liable for any damages resulting from technical errors or omissions which may be present in this document, or from use of this document.

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: 2.5.16

Date: January 22, 2015

Table of Contents

INTRODUCTION	1
SYSTEM REQUIREMENTS	1
CLOUDERA JDBC DRIVER FOR IMPALA FILES	1
USING THE CLOUDERA JDBC DRIVER FOR IMPALA	3
SETTING THE CLASS PATH	3
INITIALIZING THE DRIVER CLASS.....	3
BUILDING THE CONNECTION URL.....	4
JAVA SAMPLE CODE	4
CONFIGURING AUTHENTICATION	7
USING NO AUTHENTICATION	8
USING KERBEROS	8
USING USER NAME	8
USING USER NAME AND PASSWORD.....	9
USING USER NAME AND PASSWORD WITH SECURE SOCKETS LAYER	9
USING NO AUTHENTICATION WITH SECURE SOCKETS LAYER	10
FEATURES	11
DATA TYPES	11
CATALOG AND SCHEMA SUPPORT.....	12
SQL TRANSLATION	12
INTERFACES AND SUPPORTED METHODS.....	12
<i>CallableStatement</i>	14
<i>Connection</i>	20
<i>DatabaseMetaData</i>	24
<i>DataSource</i>	34
<i>Driver</i>	35
<i>ParameterMetaData</i>	36
<i>PooledConnection</i>	37
<i>PreparedStatement</i>	38
<i>ResultSet</i>	41
<i>ResultSetMetaData</i>	53
<i>Statement</i>	54

CONTACT US	58
APPENDIX A: CONFIGURING KERBEROS AUTHENTICATION FOR WINDOWS	59
DOWNLOADING AND INSTALLING MIT KERBEROS FOR WINDOWS	59
USING THE MIT KERBEROS TICKET MANAGER TO GET TICKETS	59
<i>Setting the KRB5CCNAME Environment Variable</i>	<i>59</i>
<i>Getting a Kerberos Ticket.....</i>	<i>60</i>
<i>Authenticating to the Impala Server.....</i>	<i>60</i>
USING THE DRIVER TO GET TICKETS	60
<i>Deleting the KRB5CCNAME Environment Variable</i>	<i>60</i>
<i>Setting Up the Kerberos Configuration File.....</i>	<i>61</i>
<i>Setting Up the JAAS Login Configuration File.....</i>	<i>61</i>
<i>Authenticating to the Impala Server.....</i>	<i>61</i>
USING AN EXISTING SUBJECT TO AUTHENTICATE THE CONNECTION	62
APPENDIX B: DRIVER CONFIGURATION OPTIONS	64

Introduction

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 11.

The Cloudera JDBC Driver for Impala complies with the JDBC 3.0, 4.0, and 4.1 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.

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

System Requirements

Each computer 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. *Table 1* lists the required version of JRE for each version of the JDBC API.

JDBC API version	JRE version
3.0	1.4 or 5.0
4.0	6.0
4.1	7.0

Table 1 System Requirements

The driver is suitable for use with all versions of Cloudera Impala.

Cloudera JDBC Driver for Impala Files

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

- **Cloudera_ImpalaJDBC3_*version*.zip**
- **Cloudera_ImpalaJDBC4_*version*.zip**
- **Cloudera_ImpalaJDBC41_*version*.zip**

Cloudera JDBC Driver for Impala Files

Each archive contains the driver supporting the JDBC API version indicated in the archive name.

The **Cloudera_ImpalaJDBC3_***version*.zip archive contains the following files and folder structure:

- ImpalaJDBC3
 - hive_metastore.jar
 - hive_service.jar
 - ImpalaJDBC3.jar
 - libfb303-0.9.0.jar
 - libthrift-0.9.0.jar
 - log4j-1.2.14.jar
 - ql.jar
 - slf4j-api-1.5.11.jar
 - slf4j-log4j12-1.5.11.jar
 - TCLIServiceClient.jar

The **Cloudera_ImpalaJDBC4_***version*.zip archive contains the following files and folder structure:

- ImpalaJDBC4
 - hive_metastore.jar
 - hive_service.jar
 - ImpalaJDBC4.jar
 - libfb303-0.9.0.jar
 - libthrift-0.9.0.jar
 - log4j-1.2.14.jar
 - ql.jar
 - slf4j-api-1.5.11.jar
 - slf4j-log4j12-1.5.11.jar
 - TCLIServiceClient.jar

The **Cloudera_ImpalaJDBC41_***version*.zip archive contains the following files and folder structure:

- ImpalaJDBC41
 - hive_metastore.jar
 - hive_service.jar
 - ImpalaJDBC41.jar
 - libfb303-0.9.0.jar
 - libthrift-0.9.0.jar

- log4j-1.2.14.jar
- ql.jar
- slf4j-api-1.5.11.jar
- slf4j-log4j12-1.5.11.jar
- TCLIServiceClient.jar

Using the Cloudera JDBC Driver for Impala

To access an Impala data warehouse using the Cloudera JDBC Driver for Impala, you must set the following:

- Class path
- Driver class
- Connection URL

For sample code that demonstrates how to use the driver, see *Java Sample Code* on page 4.

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**

Setting the Class Path

To use the Cloudera JDBC Driver for Impala, you must set the class path to include all the JAR files from the ZIP archive containing the driver that you are using.

The class path is the path that the Java Runtime Environment searches for classes and other resource files. For information about setting the class path, see “Setting the class path” from the *Java SE Documentation* at <http://docs.oracle.com/javase/7/docs/technotes/tools/windows/classpath.html>

Initializing the Driver Class

Before connecting to the Impala server, initialize the appropriate class for the Impala server instance and your application.

The following is a list of the classes used to connect the Cloudera JDBC Driver for Impala to Impala instances. The Driver classes extend `java.sql.Driver`, and the `DataSource` classes extend `javax.sql.DataSource` and `javax.sql.ConnectionPoolDataSource`

To support JDBC 3.0, classes with the following fully-qualified class names (FQCNs) are available:

- `com.cloudera.impala.jdbc3.Driver`
- `com.cloudera.impala.jdbc3.DataSource`

Java Sample Code

To support JDBC 4.0, classes with the following FQCNs are available:

- `com.cloudera.impala.jdbc4.Driver`
- `com.cloudera.impala.jdbc4.DataSource`

To support JDBC 4.1, classes with the following FQCNs are available:

- `com.cloudera.impala.jdbc41.Driver`
- `com.cloudera.impala.jdbc41.DataSource`

Building the Connection URL

Using the connection URL, you supply connection information to the data source that you are accessing. The connection URL for the Cloudera JDBC Driver for Impala takes the following form:

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

The placeholders in the connection URL are defined as follows:

- *Subprotocol* is the value **impala** in order to connect to an Impala system.
- *Host* is the DNS or IP address of the server hosting the Impala data warehouse.
- *Port* is the port to connect to on *Host*
- *Schema* is the name of the schema/database you want to access. Specifying a schema is optional.
- *Property* is any one of the connection properties that you can. For information about the properties available in the driver, see *Appendix B: Driver Configuration Options* on page 64.

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

For example, to connect to an Impala instance installed on the local computer and authenticate the connection using a user name and password, you would use the following connection URL:

```
jdbc:impala://localhost:21050;AuthMech=3;UID=UserName;PWD=Password
```

UserName and *Password* specify credentials for an existing user on the host running Impala.

For more information about the properties that you can use in the connection URL, see *Appendix B: Driver Configuration Options* on page 64.

Java Sample Code

The following Java code provides an example demonstrating how to use the JDBC API to do the following:

- Register the Cloudera JDBC Driver for Impala
- Establish a connection to an Impala database

- Query the database
- Parse a result set
- Handle exceptions
- Clean up to avoid memory leakage

Important: To use the Cloudera JDBC Driver for Impala in an application, you must include all the JAR files from the ZIP archive in the class path for your Java project.

```
// java.sql packages are required
import java.sql.*;
// java.math packages are required
import java.math.*;

class ClouderaJDBCImpalaExample {

    // Define a string as the fully qualified class name (FQCN) of
    // the desired JDBC driver
    static String JDBCdriver = "com.cloudera.impala.jdbc3.Driver";
    // Define a string as the connection URL
    static String ConnectionURL = "jdbc:impala://192.168.1.1:21050";

    public static void main(String[] args) {

        Connection con = null;
        Statement stmt = null;
        ResultSet rs = null;

        // Define a plain query
        String query = "SELECT store_sales, store_cost FROM
            default.test LIMIT 20";

        // Define a parametrized query
        String prepQuery = "SELECT first_name, last_name, emp_id
            FROM default.emp where store_id = ?";

        try {

            // Register the driver using the class name
            Class.forName(JDBCdriver);

            // Establish a connection using the connection URL
            con = DriverManager.getConnection(ConnectionURL);

            // Create a Statement object for sending SQL
            // statements to the database
            stmt = con.createStatement();

            // Execute the SQL statement
```

```

rs = stmt.executeQuery(query);

// Display a header line for output appearing in
// the Console View
System.out.printf("%20s%20s\r\n", "STORE SALES",
    "STORE COST");

// Step through each row in the result set returned
// from the database
while(rs.next()) {
    // Retrieve values from the row where the
    // cursor is currently positioned using column
    // names
    BigDecimal StoreSales = rs.getBigDecimal
        ("store_sales");
    BigDecimal StoreCost = rs.getBigDecimal
        ("store_cost");

    // Display values in columns 20 characters
    // wide in the Console View using the
    // Formatter
    System.out.printf("%20s%20s\r\n",
        StoreSales.toString(), StoreCost.toString());
}

// Create a prepared statement
PreparedStatement prep =
    m_conn.prepareStatement(preparedStatement);

// Bind the query parameter with a value
prep.setInt(1, 204);

// Execute the query
rs = prep.execute();

// Step through each row in the result set returned
// from the database
while(rs.next()) {
    // Retrieve values from the row where the
    // cursor is currently positioned using
    // column names
    String FirstName = rs.getString("first_name");
    String LastName = rs.getString("last_name");
    String EmployeeID = rs.getString("emp_id");

    // Display values in columns 20 characters
    // wide in the Console View using the
    // Formatter
    System.out.printf("%20s%20s%20s\r\n",
        FirstName, LastName, EmployeeID);
}

```

```

    } catch (SQLException se) {
        // Handle errors encountered during interaction
        // with the data source
        se.printStackTrace();
    } catch (Exception e) {
        // Handle other errors
        e.printStackTrace();
    } finally {
        // Perform clean up
        try {
            if (rs != null) {
                rs.close();
            }
        } catch (SQLException se1) {
            // Log this
        }

        try {
            if (stmt!=null) {
                stmt.close();
            }
        } catch (SQLException se2) {
            // Log this
        }

        try {
            if (con!=null) {
                con.close();
            }
        } catch (SQLException se3) {
            // Log this
            se3.printStackTrace();
        } // End try

    } // End try

} // End main
} // End ClouderaJDBCImpalaExample

```

Configuring Authentication

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

- No Authentication
- Kerberos
- User Name
- User Name and Password

Configuring Authentication

- User Name and Password with Secure Sockets Layer
- No Authentication with Secure Sockets Layer

For information about configuring the authentication mechanism that Impala uses, refer to the [Impala documentation on the Cloudera web site](#).

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

For information about the properties you can use in the connection URL, see *Appendix B: Driver Configuration Options* on page 64.

Using No Authentication

To use no 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 *Appendix A: Configuring Kerberos Authentication for Windows* on page 59. For other operating systems, refer to the MIT Kerberos documentation.

To configure Kerberos authentication:

1. Set the **AuthMech** property to 1.
2. 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 host.
3. Set the **KrbHostFQDN** property to the fully qualified domain name of the Impala host.
4. Set the **KrbServiceName** property to the service name of the Impala server.

For example:

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

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.

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://localhost:21050;AuthMech=2;UID=impala
```

Using User Name and Password

This authentication mechanism requires a user name and a password.

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 in step 2.

For example:

```
jdbc:impala://localhost:21050;AuthMech=3;UID=impala;PWD=*****
```

Using User Name and Password with Secure Sockets Layer

Secure Sockets Layer (SSL) connections require a TrustStore. You can create a TrustStore and configure the driver to use it, or allow the driver to use one of the default TrustStores. If you do not configure the driver to use a specific TrustStore, then the driver uses the Java TrustStore named `jssecacerts`. If `jssecacerts` is not available, then the driver uses a TrustStore named `cacerts` instead.

If your Impala server is configured to use client authentication, then you must create a KeyStore and configure the driver to use it. Otherwise, you do not need to create the KeyStore.

To configure User Name and Password authentication using SSL:

1. To create a TrustStore and configure the driver to use it, do the following:
 - a. Create a TrustStore containing your signed, trusted SSL certificate.
 - b. Set the **SSLTrustStore** property to the full path of the TrustStore, including the file name.
 - c. Set the **SSLTrustStorePwd** property to the password for the TrustStore.
2. Optionally, to create a KeyStore and configure the driver to use it, do the following:
 - a. Create a KeyStore containing your signed, trusted SSL certificate.
 - b. Set the **SSLKeyStore** property to the full path of the KeyStore, including the file name.
 - c. Set the **SSLKeyStorePwd** property to the password for the KeyStore.

Configuring Authentication

3. Set the **AuthMech** property to 4
4. Set the **UID** property to an appropriate user name for accessing the Impala server.
5. Set the **PWD** property to the password corresponding to the user name you provided in step 4.
6. Optionally, to allow the SSL certificate used by the server to be self-signed, set the **AllowSelfSignedCerts** property to 1
7. 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

Note: For self-signed certificates, the driver always allows the common name of the certificate to not match the host name.

For example:

```
jdbc:impala://localhost:21050;AuthMech=4;SSLTrustStore=C:\\Users\\  
\\bsmith\\Desktop\\keystore.jks;SSLTrustStorePwd=*****;UID=impala;  
PWD=*****
```

Note: For more information about the connection properties used in SSL connections, see *Appendix B: Driver Configuration Options* on page 64.

Using No Authentication with Secure Sockets Layer

Secure Sockets Layer (SSL) connections require a TrustStore. You can create a TrustStore and configure the driver to use it, or allow the driver to use one of the default TrustStores. If you do not configure the driver to use a specific TrustStore, then the driver uses the Java TrustStore `jssecacerts`. If `jssecacerts` is not available, then the driver uses `cacerts` instead.

If your Impala server is configured to use client authentication, then you must create a KeyStore and configure the driver to use it. Otherwise, you do not need to create the KeyStore.

To configure no authentication using SSL:

1. To create a TrustStore and configure the driver to use it, do the following:
 - a. Create a TrustStore containing your signed, trusted SSL certificate.
 - b. Set the **SSLTrustStore** property to the full path of the TrustStore, including the file name.
 - c. Set the **SSLTrustStorePwd** property to the password for the TrustStore.
2. Optionally, to create a KeyStore and configure the driver to use it, do the following:
 - a. Create a KeyStore containing your signed, trusted SSL certificate.
 - b. Set the **SSLKeyStore** property to the full path of the KeyStore, including the file name.
 - c. Set the **SSLKeyStorePwd** property to the password for the KeyStore.
3. Set the **AuthMech** property to 5

4. Optionally, to allow the SSL certificate used by the server to be self-signed, set the **AllowSelfSignedCerts** property to 1
5. 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

Note: For self-signed certificates, the driver always allows the common name of the certificate to not match the host name.

For example:

```
jdbc:impala://localhost:10000;AuthMech=5;SSLTrustStore=C:\\Users\\
\bsmith\\Desktop\\keystore.jks;SSLTrustStorePwd=*****
```

Note: For more information about the connection properties used in SSL connections, see *Appendix B: Driver Configuration Options* on page 64.

Features

Data Types

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

Table 2 lists the supported data type mappings.

Impala Type	SQL Type	Java Type
BIGINT	BIGINT	java.math.BigInteger
BINARY	VARBINARY	byte[]
BOOLEAN	BOOLEAN	Boolean
CHAR Note: Only available in CDH 5.2 or later.	CHAR	String
DATE	DATE	java.sql.Date
DECIMAL Note: Only available in CDH 5.1 or later.	DECIMAL	java.math.BigDecimal
DOUBLE Note: REAL is an alias for DOUBLE.	DOUBLE	Double
INT	INTEGER	Long
FLOAT	REAL	Float
SMALLINT	SMALLINT	Integer
TINYINT	TINYINT	Short
TIMESTAMP	TIMESTAMP	java.sql.Timestamp

Features

Impala Type	SQL Type	Java Type
VARCHAR Note: Only available in CDH 5.2 or later.	VARCHAR	String

Table 2 Supported Data Types

Catalog and Schema Support

The Cloudera JDBC Driver for Impala supports both catalogs and schemas in order 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.

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.

Interfaces and Supported Methods

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

- CallableStatement
- Connection
- DatabaseMetaData
- DataSource
- Driver
- ParameterMetaData
- PooledConnection
- PreparedStatement
- ResultSet
- ResultSetMetaData
- Statement

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 tables in the following sections.

Important: The driver *does not* support the following JDBC features:

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

Features

CallableStatement

The CallableStatement interface extends the PreparedStatement interface.

Table 3 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 available at <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 <code>getArray(int i)</code>	3.0	No	
Array <code>getArray(String parameterName)</code>	3.0	No	
BigDecimal <code>getBigDecimal(int parameterIndex)</code>	3.0	Yes	
BigDecimal <code>getBigDecimal(int parameterIndex, int scale)</code>	3.0	Yes	Deprecated
BigDecimal <code>getBigDecimal(String parameterName)</code>	3.0	Yes	
Blob <code>getBlob(int i)</code>	3.0	No	
Blob <code>getBlob(String parameterName)</code>	3.0	No	
boolean <code>getBoolean(int parameterIndex)</code>	3.0	Yes	
boolean <code>getBoolean(String parameterName)</code>	3.0	Yes	
byte <code>getByte(int parameterIndex)</code>	3.0	Yes	
byte <code>getByte(String parameterName)</code>	3.0	Yes	
byte[] <code>getBytes(int parameterIndex)</code>	3.0	Yes	
byte[] <code>getBytes(String parameterName)</code>	3.0	Yes	
Clob <code>getClob(int i)</code>	3.0	No	

Method	Supported since JDBC version	Supported by the driver	Notes
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	
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	

Features

Method	Supported since JDBC version	Supported by the driver	Notes
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(String parameterName, Map<String,Class<?>> map)	3.0	Yes	
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 parameterName)	3.0	Yes	
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	

Method	Supported since JDBC version	Supported by the driver	Notes
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	
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	
void setBinaryStream(String parameterName, InputStream x, long length)	4.0	Yes	
void setBlob(String parameterName, Blob x)	4.0	Yes	

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void setBlob(String parameterName, InputStream inputStream)</code>	4.0	Yes	
<code>void setBlob(String parameterName, InputStream inputStream, long length)</code>	4.0	Yes	
<code>void setBoolean(String parameterName, boolean x)</code>	3.0	Yes	
<code>void setByte(String parameterName, byte x)</code>	3.0	Yes	
<code>void setBytes(String parameterName, byte[] x)</code>	3.0	Yes	
<code>void setCharacterStream(String parameterName, Reader reader)</code>	4.0	Yes	
<code>void setCharacterStream(String parameterName, Reader reader, int length)</code>	3.0	Yes	
<code>void setCharacterStream(String parameterName, Reader reader, long length)</code>	4.0	Yes	
<code>void setClob(String parameterName, Clob x)</code>	4.0	Yes	
<code>void setClob(String parameterName, Reader reader)</code>	4.0	Yes	
<code>void setClob(String parameterName, Reader reader, long length)</code>	4.0	Yes	
<code>void setDate(String parameterName, Date x)</code>	3.0	Yes	
<code>void setDate(String parameterName, Date x, Calendar cal)</code>	3.0	Yes	
<code>void setDouble(String parameterName, double x)</code>	3.0	Yes	
<code>void setFloat(String parameterName, float x)</code>	3.0	Yes	
<code>void setInt(String parameterName, int x)</code>	3.0	Yes	
<code>void setLong(String parameterName, long x)</code>	3.0	Yes	
<code>void setNCharacterStream(String parameterName, Reader value)</code>	4.0	Yes	

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void setNCharacterStream(String parameterName, Reader value, long length)</code>	4.0	Yes	
<code>void setNClob(String parameterName, NClob value)</code>	4.0	Yes	
<code>void setNClob(String parameterName, Reader reader)</code>	4.0	Yes	
<code>void setNClob(String parameterName, Reader reader, long length)</code>	4.0	Yes	
<code>void setNString(String parameterName, String value)</code>	4.0	Yes	
<code>void setNull(String parameterName, int sqlType)</code>	3.0	Yes	
<code>void setNull(String parameterName, int sqlType, String typeName)</code>	3.0	Yes	
<code>void setObject(String parameterName, Object x)</code>	3.0	Yes	
<code>void setObject(String parameterName, Object x, int targetSqlType)</code>	3.0	Yes	
<code>void setObject(String parameterName, Object x, int targetSqlType, int scale)</code>	3.0	Yes	
<code>void setRowId(String parameterName, RowId x)</code>	4.0	Yes	
<code>void setShort(String parameterName, short x)</code>	3.0	Yes	
<code>void setSQLXML(String parameterName, SQLXML xmlObject)</code>	4.0	Yes	
<code>void setString(String parameterName, String x)</code>	3.0	Yes	
<code>void setTime(String parameterName, Time x)</code>	3.0	Yes	
<code>void setTime(String parameterName, Time x, Calendar cal)</code>	3.0	Yes	
<code>void setTimestamp(String parameterName, Timestamp x)</code>	3.0	Yes	
<code>void setTimestamp(String parameterName, Timestamp x, Calendar cal)</code>	3.0	Yes	

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void setURL(String parameterName, URL val)</code>	3.0	Yes	
<code>boolean wasNull()</code>	3.0	Yes	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

Table 3 Methods in the CallableStatement Interface

Connection

Table 4 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 available at <http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/Connection.html>

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void clearWarnings()</code>	3.0	Yes	
<code>void close()</code>	3.0	Yes	
<code>void commit()</code>	3.0	Yes	Auto-commit cannot be set to false because it is hard-coded as true
<code>Array createArrayOf(String typeName, Object[] elements)</code>	4.0	No	
<code>Blob createBlob()</code>	4.0	No	
<code>Clob createClob()</code>	4.0	No	
<code>NClob createNClob()</code>	4.0	No	
<code>SQLXML createSQLXML()</code>	4.0	No	

Method	Supported since JDBC version	Supported by the driver	Notes
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 typeName, Object[] attributes)	4.0	No	
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	

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>SQLWarning getWarnings()</code>	3.0	Yes	
<code>boolean isClosed()</code>	3.0	Yes	
<code>boolean isReadOnly()</code>	3.0	Yes	Returns <code>true</code>
<code>boolean isValid(int timeout)</code>	4.0	Yes	
<code>String nativeSQL(String sql)</code>	3.0	Yes	
<code>CallableStatement prepareCall(String sql)</code>	3.0	No	
<code>CallableStatement prepareCall(String sql, int resultSetType, int resultSetConcurrency)</code>	3.0	No	
<code>CallableStatement prepareCall(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability)</code>	3.0	No	
<code>PreparedStatement prepareStatement(String sql)</code>	3.0	Yes	
<code>PreparedStatement prepareStatement(String sql, int autoGeneratedKeys)</code>	3.0	No	
<code>PreparedStatement prepareStatement(String sql, int[] columnIndexes)</code>	3.0	No	
<code>PreparedStatement prepareStatement(String sql, int resultSetType, int resultSetConcurrency)</code>	3.0	No	
<code>PreparedStatement prepareStatement(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability)</code>	3.0	No	
<code>PreparedStatement prepareStatement(String sql, String[] columnNames)</code>	3.0	No	
<code>void releaseSavepoint(Savepoint savepoint)</code>	3.0	No	Savepoints are not available because transactions are not supported.

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void rollback()</code>	3.0	No	Savepoints are not available because transactions are not supported.
<code>void rollback(Savepoint savepoint)</code>	3.0	No	Savepoints are not available because transactions are not supported.
<code>void setAutoCommit(boolean autoCommit)</code>	3.0	Yes	Ignored because auto-commit is hard-coded to <code>true</code>
<code>void setCatalog(String catalog)</code>	3.0	Yes	
<code>void setClientInfo(Properties properties)</code>	4.0	Yes	
<code>void setClientInfo(String name, String value)</code>	4.0	Yes	
<code>void setHoldability(int holdability)</code>	3.0	Yes	
<code>void setNetworkTimeout(Executor executor, int milliseconds)</code>	4.1	No	
<code>void setReadOnly(boolean readOnly)</code>	3.0	Yes	
<code>Savepoint setSavepoint()</code>	3.0	No	Savepoints are not available because transactions are not supported.
<code>Savepoint setSavepoint(String name)</code>	3.0	No	Savepoints are not available because transactions are not supported.

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void setSchema(String schema)</code>	4.1	Yes	Does not actually change the schema name used by newly created statements; only changes the value returned by <code>getSchema()</code>
<code>void setTransactionIsolation(int level)</code>	3.0	Yes	
<code>void setTypeMap(Map<String,Class<?>> map)</code>	3.0	No	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

Table 4 Methods in the Connection Class

DatabaseMetaData

Table 5 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 available at <http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/DatabaseMetaData.html>

Method	Supported since JDBC version	Supported by the driver	Notes
<code>boolean allProceduresAreCallable()</code>	3.0	Yes	Returns true
<code>boolean allTablesAreSelectable()</code>	3.0	Yes	Returns true
<code>boolean autoCommitFailureClosesAllResultSets()</code>	4.0	Yes	Returns true
<code>boolean dataDefinitionCausesTransactionCommit()</code>	3.0	Yes	Returns false

Method	Supported since JDBC version	Supported by the driver	Notes
<code>boolean dataDefinitionIgnoredInTransactions()</code>	3.0	Yes	Returns false
<code>boolean deletesAreDetected(int type)</code>	3.0	Yes	Returns true
<code>boolean doesMaxRowSizeIncludeBlobs()</code>	3.0	Yes	Returns false
<code>boolean generatedKeyAlwaysReturned()</code>	4.1	Yes	
<code>ResultSet getAttributes(String catalog, String schemaPattern, String typeNamePattern, String attributeNamePattern)</code>	3.0	Yes	
<code>ResultSet getBestRowIdentifier(String catalog, String schema, String table, int scope, boolean nullable)</code>	3.0	Yes	
<code>ResultSet getCatalogs()</code>	3.0	Yes	
<code>String getCatalogSeparator()</code>	3.0	Yes	
<code>String getCatalogTerm()</code>	3.0	Yes	
<code>ResultSet getClientInfoProperties()</code>	4.0	Yes	
<code>ResultSet getColumnPrivileges(String catalog, String schema, String table, String columnNamePattern)</code>	3.0	Yes	
<code>ResultSet getColumns(String catalog, String schemaPattern, String tableNamePattern, String columnNamePattern)</code>	3.0	Yes	
<code>Connection getConnection()</code>	3.0	Yes	
<code>ResultSet getCrossReference(String primaryCatalog, String primarySchema, String primaryTable, String foreignCatalog, String foreignSchema, String foreignTable)</code>	3.0	Yes	
<code>int getDatabaseMajorVersion()</code>	3.0	Yes	
<code>int getDatabaseMinorVersion()</code>	3.0	Yes	
<code>String getDatabaseProductName()</code>	3.0	Yes	Hard-coded to Cloudera Impala

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>String getDatabaseProductVersion()</code>	3.0	Yes	
<code>int getDefaultTransactionIsolation()</code>	3.0	Yes	Hard-coded to TRANSACTION_READ_UNCOMMITTED
<code>int getDriverMajorVersion()</code>	3.0	Yes	
<code>int getDriverMinorVersion()</code>	3.0	Yes	
<code>String getDriverName()</code>	3.0	Yes	Hard-coded to ImpalaJDBC
<code>String getDriverVersion()</code>	3.0	Yes	
<code>ResultSet getExportedKeys(String catalog, String schema, String table)</code>	3.0	Yes	
<code>String getExtraNameCharacters()</code>	3.0	Yes	Returns an empty String.
<code>ResultSet getFunctionColumns(String catalog, String schemaPattern, String functionNamePattern, String columnNamePattern)</code>	4.0	Yes	
<code>ResultSet getFunctions(String catalog, String schemaPattern, String functionNamePattern)</code>	4.0	Yes	
<code>String getIdentifierQuoteString()</code>	3.0	Yes	Returns a backquote (`)
<code>ResultSet getImportedKeys(String catalog, String schema, String table)</code>	3.0	Yes	
<code>ResultSet getIndexInfo(String catalog, String schema, String table, boolean unique, boolean approximate)</code>	3.0	Yes	
<code>int getJDBCMajorVersion()</code>	3.0	Yes	
<code>int getJDBCMinorVersion()</code>	3.0	Yes	
<code>int getMaxBinaryLiteralLength()</code>	3.0	Yes	Returns 0

Method	Supported since JDBC version	Supported by the driver	Notes
<code>int getMaxCatalogNameLength()</code>	3.0	Yes	Returns 128
<code>int getMaxCharLiteralLength()</code>	3.0	Yes	Returns 0
<code>int getMaxColumnNameLength()</code>	3.0	Yes	Returns 128
<code>int getMaxColumnsInGroupBy()</code>	3.0	Yes	Returns 0
<code>int getMaxColumnsInIndex()</code>	3.0	Yes	Returns 0
<code>int getMaxColumnsInOrderBy()</code>	3.0	Yes	Returns 0
<code>int getMaxColumnsInSelect()</code>	3.0	Yes	Returns 0
<code>int getMaxColumnsInTable()</code>	3.0	Yes	Returns 0
<code>int getMaxConnections()</code>	3.0	Yes	Returns 0
<code>int getMaxCursorNameLength()</code>	3.0	Yes	Returns 0
<code>int getMaxIndexLength()</code>	3.0	Yes	Returns 0
<code>int getMaxProcedureNameLength()</code>	3.0	Yes	Returns 0
<code>int getMaxRowSize()</code>	3.0	Yes	Returns 0
<code>int getMaxSchemaNameLength()</code>	3.0	Yes	Returns 128
<code>int getMaxStatementLength()</code>	3.0	Yes	Returns 0
<code>int getMaxStatements()</code>	3.0	Yes	Returns 0
<code>int getMaxTableNameLength()</code>	3.0	Yes	Returns 128
<code>int getMaxTablesInSelect()</code>	3.0	Yes	Returns 0
<code>int getMaxUserNameLength()</code>	3.0	Yes	Returns 0

Features

Method	Supported since JDBC version	Supported by the driver	Notes
String getNumericFunctions()	3.0	Yes	Returns the Numeric Functions list from the 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 (\)
String getSQLKeywords()	3.0	Yes	Returns an empty String.

Method	Supported since JDBC version	Supported by the driver	Notes
<code>int getSQLStateType()</code>	3.0	Yes	Returns <code>sqlStateSQL99</code>
<code>String getStringFunctions()</code>	3.0	Yes	Returns the String Functions list from the specification related to the JDBC version of the driver.
<code>ResultSet getSuperTables(String catalog, String schemaPattern, String tableNamePattern)</code>	3.0	Yes	
<code>ResultSet getSuperTypes(String catalog, String schemaPattern, String typeNamePattern)</code>	3.0	Yes	
<code>String getSystemFunctions()</code>	3.0	Yes	Returns <code>DATABASE, IFNULL, USER</code>
<code>ResultSet getTablePrivileges(String catalog, String schemaPattern, String tableNamePattern)</code>	3.0	Yes	
<code>ResultSet getTables(String catalog, String schemaPattern, String tableNamePattern, String[] types)</code>	3.0	Yes	
<code>ResultSet getTableTypes()</code>	3.0	Yes	
<code>String getTimeDateFunctions()</code>	3.0	Yes	Returns the Time and Date Functions list from the specification related to the JDBC version of the driver.
<code>ResultSet getTypeInfo()</code>	3.0	Yes	
<code>ResultSet getUDTs(String catalog, String schemaPattern, String typeNamePattern, int[] types)</code>	3.0	Yes	
<code>String getURL()</code>	3.0	Yes	

Features

Method	Supported since JDBC version	Supported by the driver	Notes
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	
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

Method	Supported since JDBC version	Supported by the driver	Notes
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
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 supportsCatalogsInPrivilegeDefinitions()	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 supportsDataDefinitionAndDataManipulationTransactions()	3.0	Yes	Returns false
boolean supportsDataManipulationTransactionsOnly()	3.0	Yes	Returns false
boolean supportsDifferentTableCorrelationNames()	3.0	Yes	Returns false

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>boolean supportsExpressionsInOrderBy()</code>	3.0	Yes	Returns true
<code>boolean supportsExtendedSQLGrammar()</code>	3.0	Yes	Returns false
<code>boolean supportsFullOuterJoins()</code>	3.0	Yes	Returns true
<code>boolean supportsGetGeneratedKeys()</code>	3.0	Yes	Returns false
<code>boolean supportsGroupBy()</code>	3.0	Yes	Returns true
<code>boolean supportsGroupByBeyondSelect()</code>	3.0	Yes	Returns true
<code>boolean supportsGroupByUnrelated()</code>	3.0	Yes	Returns false
<code>boolean supportsIntegrityEnhancementFacility()</code>	3.0	Yes	Returns false
<code>boolean supportsLikeEscapeClause()</code>	3.0	Yes	Returns true
<code>boolean supportsLimitedOuterJoins()</code>	3.0	Yes	Returns false
<code>boolean supportsMinimumSQLGrammar()</code>	3.0	Yes	Returns true
<code>boolean supportsMixedCaseIdentifiers()</code>	3.0	Yes	Returns false
<code>boolean supportsMixedCaseQuotedIdentifiers()</code>	3.0	Yes	Returns true
<code>boolean supportsMultipleOpenResults()</code>	3.0	Yes	Returns false
<code>boolean supportsMultipleResultSets()</code>	3.0	Yes	Returns false
<code>boolean supportsMultipleTransactions()</code>	3.0	Yes	Returns true
<code>boolean supportsNamedParameters()</code>	3.0	Yes	Returns false
<code>boolean supportsNonNullableColumns()</code>	3.0	Yes	Returns false
<code>boolean supportsOpenCursorsAcrossCommit()</code>	3.0	Yes	Returns false
<code>boolean supportsOpenCursorsAcrossRollback()</code>	3.0	Yes	Returns false
<code>boolean supportsOpenStatementsAcrossCommit()</code>	3.0	Yes	Returns true

Method	Supported since JDBC version	Supported by the driver	Notes
<code>boolean supportsOpenStatementsAcrossRollback()</code>	3.0	Yes	Returns true
<code>boolean supportsOrderByUnrelated()</code>	3.0	Yes	Returns false
<code>boolean supportsOuterJoins()</code>	3.0	Yes	Returns false
<code>boolean supportsPositionedDelete()</code>	3.0	Yes	Returns false
<code>boolean supportsPositionedUpdate()</code>	3.0	Yes	Returns false
<code>boolean supportsResultSetConcurrency(int type, int concurrency)</code>	3.0	Yes	
<code>boolean supportsResultSetHoldability(int holdability)</code>	3.0	Yes	
<code>boolean supportsResultSetType(int type)</code>	3.0	Yes	
<code>boolean supportsSavepoints()</code>	3.0	Yes	Returns false
<code>boolean supportsSchemasInDataManipulation()</code>	3.0	Yes	Returns true
<code>boolean supportsSchemasInIndexDefinitions()</code>	3.0	Yes	Returns true
<code>boolean supportsSchemasInPrivilegeDefinitions()</code>	3.0	Yes	Returns true
<code>boolean supportsSchemasInProcedureCalls()</code>	3.0	Yes	Returns false
<code>boolean supportsSchemasInTableDefinitions()</code>	3.0	Yes	Returns true
<code>boolean supportsSelectForUpdate()</code>	3.0	Yes	Returns false
<code>boolean supportsStatementPooling()</code>	3.0	Yes	Returns false
<code>boolean supportsStoredFunctionsUsingCallSyntax()</code>	4.0	Yes	Returns false
<code>boolean supportsStoredProcedures()</code>	3.0	Yes	Returns true
<code>boolean supportsSubqueriesInComparisons()</code>	3.0	Yes	Returns true
<code>boolean supportsSubqueriesInExists()</code>	3.0	Yes	Returns true
<code>boolean supportsSubqueriesInIns()</code>	3.0	Yes	Returns true

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>boolean supportsSubqueriesInQuantifieds()</code>	3.0	Yes	Returns true
<code>boolean supportsTableCorrelationNames()</code>	3.0	Yes	Returns true
<code>boolean supportsTransactionIsolationLevel(int level)</code>	3.0	Yes	
<code>boolean supportsTransactions()</code>	3.0	Yes	Returns false
<code>boolean supportsUnion()</code>	3.0	Yes	Returns true
<code>boolean supportsUnionAll()</code>	3.0	Yes	Returns true
<code>boolean updatesAreDetected(int type)</code>	3.0	Yes	Returns true
<code>boolean usesLocalFilePerTable()</code>	3.0	Yes	Returns false
<code>boolean usesLocalFiles()</code>	3.0	Yes	Returns false
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

Table 5 Methods in the DatabaseMetadata Class

DataSource

Table 6 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 available at <http://docs.oracle.com/javase/1.5.0/docs/api/javax/sql/DataSource.html>

Method	Supported since JDBC version	Supported by the driver	Notes
<code>Connection getConnection()</code>	3.0	Yes	
<code>Connection getConnection(String username, String password)</code>	3.0	Yes	

Method	Supported since JDBC version	Supported by the driver	Notes
<code>int getLoginTimeout()</code>	3.0	Yes	
<code>PrintWriter getLogWriter()</code>	3.0	Yes	
<code>Logger getParentLogger()</code>	4.1	No	The driver does not use <code>java.util.logging</code>
<code>void setLoginTimeout(int seconds)</code>	3.0	Yes	
<code>void setLogWriter(PrintWriter out)</code>	3.0	Yes	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

Table 6 Methods in the DataSource Class

Driver

Table 7 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 available at <http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/Driver.html>

Method	Supported since JDBC version	Supported by the driver	Notes
<code>boolean acceptsURL(String url)</code>	3.0	Yes	
<code>Connection connect(String url, Properties info)</code>	3.0	Yes	
<code>int getMajorVersion()</code>	3.0	Yes	
<code>int getMinorVersion()</code>	3.0	Yes	
<code>Logger getParentLogger()</code>	4.1	No	

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>DriverPropertyInfo[] getPropertyInfo(String url, Properties info)</code>	3.0	Yes	
<code>boolean jdbcCompliant()</code>	3.0	Yes	

Table 7 Methods in the Driver Class

ParameterMetaData

Table 8 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 available at <http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/ParameterMetaData.html>

Method	Supported since JDBC version	Supported by the driver	Notes
<code>String getParameterClassName(int param)</code>	3.0	Yes	
<code>int getParameterCount()</code>	3.0	Yes	
<code>int getParameterMode(int param)</code>	3.0	Yes	
<code>int getParameterType(int param)</code>	3.0	Yes	
<code>String getParameterTypeName(int param)</code>	3.0	Yes	
<code>int getPrecision(int param)</code>	3.0	Yes	
<code>int getScale(int param)</code>	3.0	Yes	
<code>int isNullable(int param)</code>	3.0	Yes	
<code>boolean isSigned(int param)</code>	3.0	Yes	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	

Method	Supported since JDBC version	Supported by the driver	Notes
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

Table 8 Methods in the ParameterMetaData Class

PooledConnection

Table 9 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 available at <http://docs.oracle.com/javase/1.5.0/docs/api/javax/sql/PooledConnection.html>

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void addConnectionEventListener(ConnectionEventListener listener)</code>	3.0	Yes	
<code>void addStatementEventListener(StatementEventListener listener)</code>	4.0	Yes	
<code>void close()</code>	3.0	Yes	
<code>Connection getConnection()</code>	3.0	Yes	
<code>void removeConnectionEventListener(ConnectionEventListener listener)</code>	3.0	Yes	

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void removeStatementEventListener(StatementEventListener listener)</code>	4.0	Yes	Removes the specified <code>StatementEventListener</code> from the list of components that will be notified when the driver detects that a <code>PreparedStatement</code> has been closed or is invalid.

Table 9 Methods in the PooledConnection Class

PreparedStatement

The `PreparedStatement` interface extends the `Statement` interface.

Table 10 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 available at <http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/PreparedStatement.html>

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void addBatch()</code>	3.0	Yes	
<code>void clearParameters()</code>	3.0	Yes	
<code>boolean execute()</code>	3.0	Yes	
<code>ResultSet executeQuery()</code>	3.0	Yes	
<code>int executeUpdate()</code>	3.0	Yes	
<code>ResultSetMetaData getMetaData()</code>	3.0	Yes	

Method	Supported since JDBC version	Supported by the driver	Notes
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 parameterIndex, InputStream x)	4.0	Yes	
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 length)	3.0	Yes	

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void setCharacterStream(int parameterIndex, Reader reader, long length)</code>	4.0	Yes	
<code>void setClob(int parameterIndex, Clob x)</code>	3.0	No	
<code>void setClob(int parameterIndex, Reader reader)</code>	4.0	No	
<code>void setClob(int parameterIndex, Reader reader, long length)</code>	4.0	No	
<code>void setDate(int parameterIndex, Date x)</code>	3.0	Yes	
<code>void setDate(int parameterIndex, Date x, Calendar cal)</code>	3.0	Yes	
<code>void setDouble(int parameterIndex, double x)</code>	3.0	Yes	
<code>void setFloat(int parameterIndex, float x)</code>	3.0	Yes	
<code>void setInt(int parameterIndex, int x)</code>	3.0	Yes	
<code>void setLong(int parameterIndex, long x)</code>	3.0	Yes	
<code>void setNCharacterStream(int parameterIndex, Reader value)</code>	4.0	No	
<code>void setNCharacterStream(int parameterIndex, Reader value, long length)</code>	4.0	No	
<code>void setNClob(int parameterIndex, NClob value)</code>	4.0	No	
<code>void setNClob(int parameterIndex, Reader reader)</code>	4.0	No	
<code>void setNClob(int parameterIndex, Reader reader, long length)</code>	4.0	No	
<code>void setNString(int parameterIndex, String value)</code>	4.0	No	
<code>void setNull(int paramIndex, int sqlType, String typeName)</code>	3.0	Yes	
<code>void setObject(int parameterIndex, Object x)</code>	3.0	Yes	

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void setObject(int parameterIndex, Object x, int targetSqlType)</code>	3.0	Yes	
<code>void setObject(int parameterIndex, Object x, int targetSqlType, int scale)</code>	3.0	Yes	
<code>void setRef(int parameterIndex, Ref x)</code>	3.0	No	
<code>void setRowId(int parameterIndex, RowId x)</code>	4.0	No	
<code>void setShort(int parameterIndex, short x)</code>	3.0	No	
<code>void setSQLXML(int parameterIndex, SQLXML xmlObject)</code>	4.0	Yes	
<code>void setString(int parameterIndex, String x)</code>	3.0	Yes	
<code>void setTime(int parameterIndex, Time x)</code>	3.0	Yes	
<code>void setTime(int parameterIndex, Time x, Calendar cal)</code>	3.0	Yes	
<code>void setTimestamp(int parameterIndex, Timestamp x)</code>	3.0	Yes	
<code>void setTimestamp(int parameterIndex, Timestamp x, Calendar cal)</code>	3.0	Yes	
<code>void setUnicodeStream(int parameterIndex, InputStream x, int length)</code>	3.0	Yes	Deprecated
<code>void setURL(int parameterIndex, URL x)</code>	3.0	No	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

Table 10 Methods in the PreparedStatement Class

ResultSet

Table 11 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.

Features

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

Method	Supported since JDBC version	Supported by the driver	Notes
<code>boolean absolute(int row)</code>	3.0	No	
<code>void afterLast()</code>	3.0	No	
<code>void beforeFirst()</code>	3.0	No	
<code>void cancelRowUpdates()</code>	3.0	No	Not valid because the driver is read-only.
<code>void clearWarnings()</code>	3.0	Yes	
<code>void close()</code>	3.0	Yes	
<code>void deleteRow()</code>	3.0	No	Not valid because the driver is read-only.
<code>int findColumn(String columnName)</code>	3.0	Yes	
<code>boolean first()</code>	3.0	No	
<code>Array getArray(int i)</code>	3.0	No	
<code>Array getArray(String colName)</code>	3.0	No	
<code>InputStream getAsciiStream(int columnIndex)</code>	3.0	Yes	
<code>InputStream getAsciiStream(String columnName)</code>	3.0	Yes	
<code>BigDecimal getBigDecimal(int columnIndex)</code>	3.0	Yes	
<code>BigDecimal getBigDecimal(int columnIndex, int scale)</code>	3.0	Yes	Deprecated
<code>BigDecimal getBigDecimal(String columnName)</code>	3.0	Yes	
<code>BigDecimal getBigDecimal(String columnName, int scale)</code>	3.0	Yes	Deprecated
<code>InputStream getBinaryStream(int columnIndex)</code>	3.0	Yes	

Method	Supported since JDBC version	Supported by the driver	Notes
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	
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	

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>int getFetchDirection()</code>	3.0	Yes	
<code>int getFetchSize()</code>	3.0	Yes	
<code>float getFloat(int columnIndex)</code>	3.0	Yes	
<code>float getFloat(String columnName)</code>	3.0	Yes	
<code>int getHoldability()</code>	4.0	Yes	
<code>int getInt(int columnIndex)</code>	3.0	Yes	
<code>int getInt(String columnName)</code>	3.0	Yes	
<code>long getLong(int columnIndex)</code>	3.0	Yes	
<code>long getLong(String columnName)</code>	3.0	Yes	
<code>ResultSetMetaData getMetaData()</code>	3.0	Yes	
<code>Reader getNCharacterStream(int columnIndex)</code>	4.0	No	
<code>Reader getNCharacterStream(String columnLabel)</code>	4.0	No	
<code>NClob getNClob(int columnIndex)</code>	4.0	No	
<code>NClob getNClob(String columnLabel)</code>	4.0	No	
<code>String getNString(int columnIndex)</code>	4.0	No	
<code>String getNString(String columnLabel)</code>	4.0	No	
<code>Object getObject(int columnIndex)</code>	3.0	Yes	
<code><T> T getObject(int columnIndex, Class<T> type)</code>	4.1	No	
<code>Object getObject(int i, Map<String,Class<?>> map)</code>	3.0	No	
<code>Object getObject(String columnName)</code>	3.0	No	
<code><T> T getObject(String columnName, Class<T> type)</code>	4.1	No	

Method	Supported since JDBC version	Supported by the driver	Notes
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 columnName)	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 columnName)	4.0	No	
Statement getStatement()	3.0	Yes	
String getString(int columnIndex)	3.0	Yes	
String getString(String columnName)	3.0	Yes	
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	

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>int getType()</code>	3.0	Yes	
<code>InputStream getUnicodeStream(int columnIndex)</code>	3.0	Yes	Deprecated
<code>InputStream getUnicodeStream(String columnName)</code>	3.0	Yes	Deprecated
<code>URL getURL(int columnIndex)</code>	3.0	No	
<code>URL getURL(String columnName)</code>	3.0	No	
<code>SQLWarning getWarnings()</code>	3.0	Yes	
<code>void insertRow()</code>	3.0	No	Not valid because the driver is read-only.
<code>boolean isAfterLast()</code>	3.0	Yes	
<code>boolean isBeforeFirst()</code>	3.0	Yes	
<code>boolean isClosed()</code>	4.0	Yes	
<code>boolean isFirst()</code>	3.0	Yes	
<code>boolean isLast()</code>	3.0	No	
<code>boolean last()</code>	3.0	No	
<code>void moveToCurrentRow()</code>	3.0	No	Not valid because the driver is read-only.
<code>void moveToInsertRow()</code>	3.0	No	Not valid because the driver is read-only.
<code>boolean next()</code>	3.0	Yes	
<code>boolean previous()</code>	3.0	No	
<code>void refreshRow()</code>	3.0	No	
<code>boolean relative(int rows)</code>	3.0	No	

Method	Supported since JDBC version	Supported by the driver	Notes
<code>boolean rowDeleted()</code>	3.0	Yes	Hard-coded to <code>false</code>
<code>boolean rowInserted()</code>	3.0	Yes	Hard-coded to <code>false</code>
<code>boolean rowUpdated()</code>	3.0	Yes	Hard-coded to <code>false</code>
<code>void setFetchDirection(int direction)</code>	3.0	No	Not valid because the driver is forward-only.
<code>void setFetchSize(int rows)</code>	3.0	Yes	
<code>void updateArray(int columnIndex, Array x)</code>	3.0	No	
<code>void updateArray(String columnName, Array x)</code>	3.0	No	
<code>void updateAsciiStream(int columnIndex, InputStream x)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateAsciiStream(int columnIndex, InputStream x, int length)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateAsciiStream(int columnIndex, InputStream x, long length)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateAsciiStream(String columnName, InputStream x)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateAsciiStream(String columnName, InputStream x, int length)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateAsciiStream(String columnName, InputStream x, long length)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateBigDecimal(int columnIndex, BigDecimal x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBigDecimal(String columnName, BigDecimal x)</code>	3.0	No	Not valid because the driver is read-only.

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void updateBinaryStream(int columnIndex, InputStream x)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateBinaryStream(int columnIndex, InputStream x, int length)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBinaryStream(int columnIndex, InputStream x, long length)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateBinaryStream(String columnName, InputStream x)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateBinaryStream(String columnName, InputStream x, int length)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBinaryStream(String columnName, InputStream x, long length)</code>	4.0	No	Not valid because the driver is read-only.
<code>void updateBlob(int columnIndex, InputStream inputStream)</code>	4.0	No	
<code>void updateBlob(int columnIndex, Blob x)</code>	3.0	No	
<code>void updateBlob(int columnIndex, InputStream inputStream, long length)</code>	4.0	No	
<code>void updateBlob(String columnName, InputStream inputStream)</code>	4.0	No	
<code>void updateBlob(String columnName, Blob x)</code>	3.0	No	
<code>void updateBlob(String columnLabel, InputStream inputStream, long length)</code>	4.0	No	
<code>void updateBoolean(int columnIndex, boolean x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBoolean(String columnName, boolean x)</code>	3.0	No	Not valid because the driver is read-only.

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void updateByte(int columnIndex, byte x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateByte(String columnName, byte x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBytes(int columnIndex, byte[] x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBytes(String columnName, byte[] x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateCharacterStream(int columnIndex, Reader x, int length)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateCharacterStream(String columnName, Reader reader, int length)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateBlob(int columnIndex, InputStream inputStream)</code>	4.0	No	
<code>void updateClob(int columnIndex, Clob x)</code>	3.0	No	
<code>void updateBlob(int columnIndex, InputStream inputStream, long length)</code>	4.0	No	
<code>void updateBlob(String columnName, InputStream inputStream)</code>	4.0	No	
<code>void updateClob(String columnName, Clob x)</code>	3.0	No	
<code>void updateBlob(String columnName, InputStream inputStream, long length)</code>	4.0	No	
<code>void updateDate(int columnIndex, Date x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateDate(String columnName, Date x)</code>	3.0	No	Not valid because the driver is read-only.

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void updateDouble(int columnIndex, double x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateDouble(String columnName, double x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateFloat(int columnIndex, float x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateFloat(String columnName, float x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateInt(int columnIndex, int x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateInt(String columnName, int x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateLong(int columnIndex, long x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateLong(String columnName, long x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateNCharacterStream(int columnIndex, Reader x)</code>	4.0	No	
<code>void updateNCharacterStream(int columnIndex, Reader x, long length)</code>	4.0	No	
<code>void updateNCharacterStream(String columnName, Reader reader)</code>	4.0	No	
<code>void updateNCharacterStream(String columnName, Reader reader, long length)</code>	4.0	No	
<code>void updateNClob(int columnIndex, NClob nClob)</code>	4.0	No	
<code>void updateNClob(int columnIndex, Reader reader)</code>	4.0	No	

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void updateNClob(int columnIndex, Reader reader, long length)</code>	4.0	No	
<code>void updateNClob(String columnName, NClob nClob)</code>	4.0	No	
<code>void updateNClob(String columnName, Reader reader)</code>	4.0	No	
<code>void updateNClob(String columnName, Reader reader, long length)</code>	4.0	No	
<code>void updateNString(int columnIndex, String nString)</code>	4.0	No	
<code>void updateNString(String columnName, String nString)</code>	4.0	No	
<code>void updateNull(int columnIndex)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateNull(String columnName)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateObject(int columnIndex, Object x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateObject(int columnIndex, Object x, int scale)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateObject(String columnName, Object x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateObject(String columnName, Object x, int scale)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateRef(int columnIndex, Ref x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateRef(String columnName, Ref x)</code>	3.0	No	Not valid because the driver is read-only.

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void updateRow()</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateRowId(int columnIndex, RowId x)</code>	4.0	No	
<code>void updateRowId(String columnName, RowId x)</code>	4.0	No	
<code>void updateShort(int columnIndex, short x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateShort(String columnName, short x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateSQLXML(int columnIndex, SQLXML xmlObject)</code>	4.0	No	
<code>void updateSQLXML(String columnName, SQLXML xmlObject)</code>	4.0	No	
<code>void updateString(int columnIndex, String x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateString(String columnName, String x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateTime(int columnIndex, Time x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateTime(String columnName, Time x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateTimestamp(int columnIndex, Timestamp x)</code>	3.0	No	Not valid because the driver is read-only.
<code>void updateTimestamp(String columnName, Timestamp x)</code>	3.0	No	Not valid because the driver is read-only.
<code>boolean wasNull()</code>	3.0	Yes	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	

Method	Supported since JDBC version	Supported by the driver	Notes
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

Table 11 Methods in the ResultSet Class

ResultSetMetaData

Table 12 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 available at <http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/ResultSetMetaData.html>

Method	Supported since JDBC version	Supported by the driver	Notes
<code>String getCatalogName(int column)</code>	3.0	Yes	
<code>String getColumnClassName(int column)</code>	3.0	Yes	
<code>int getColumnCount()</code>	3.0	Yes	
<code>int getColumnDisplaySize(int column)</code>	3.0	Yes	
<code>String getColumnLabel(int column)</code>	3.0	Yes	
<code>String getColumnName(int column)</code>	3.0	Yes	
<code>int getColumnType(int column)</code>	3.0	Yes	
<code>String getColumnName(int column)</code>	3.0	Yes	
<code>int getPrecision(int column)</code>	3.0	Yes	
<code>int getScale(int column)</code>	3.0	Yes	
<code>String getSchemaName(int column)</code>	3.0	Yes	
<code>String getTableName(int column)</code>	3.0	Yes	

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>boolean isAutoIncrement(int column)</code>	3.0	Yes	
<code>boolean isCaseSensitive(int column)</code>	3.0	Yes	
<code>boolean isCurrency(int column)</code>	3.0	Yes	
<code>boolean isDefinitelyWritable(int column)</code>	3.0	Yes	
<code>int isNullable(int column)</code>	3.0	Yes	
<code>boolean isReadOnly(int column)</code>	3.0	Yes	
<code>boolean isSearchable(int column)</code>	3.0	Yes	
<code>boolean isSigned(int column)</code>	3.0	Yes	
<code>boolean isWritable(int column)</code>	3.0	Yes	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

Table 12 Methods in the ResultSetMetaData Class

Statement

Table 13 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 available at <http://docs.oracle.com/javase/1.5.0/docs/api/java/sql/Statement.html>

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void addBatch(String sql)</code>	3.0	Yes	
<code>void cancel()</code>	3.0	Yes	
<code>void clearBatch()</code>	3.0	Yes	
<code>void clearWarnings()</code>	3.0	Yes	
<code>void close()</code>	3.0	Yes	
<code>void closeOnCompletion()</code>	4.1	Yes	
<code>boolean execute(String sql)</code>	3.0	Yes	
<code>boolean execute(String sql, int autoGeneratedKeys)</code>	3.0	No	
<code>boolean execute(String sql, int[] columnIndexes)</code>	3.0	No	
<code>boolean execute(String sql, String[] columnNames)</code>	3.0	No	
<code>int[] executeBatch()</code>	3.0	No	
<code>ResultSet executeQuery(String sql)</code>	3.0	Yes	
<code>int executeUpdate(String sql)</code>	3.0	Yes	
<code>int executeUpdate(String sql, int autoGeneratedKeys)</code>	3.0	No	
<code>int executeUpdate(String sql, int[] columnIndexes)</code>	3.0	No	
<code>int executeUpdate(String sql, String[] columnNames)</code>	3.0	No	
<code>Connection getConnection()</code>	3.0	Yes	
<code>int getFetchDirection()</code>	3.0	Yes	
<code>int getFetchSize()</code>	3.0	Yes	
<code>ResultSet getGeneratedKeys()</code>	3.0	Yes	
<code>int getMaxFieldSize()</code>	3.0	Yes	

Features

Method	Supported since JDBC version	Supported by the driver	Notes
<code>int getMaxRows()</code>	3.0	Yes	
<code>boolean getMoreResults()</code>	3.0	Yes	
<code>boolean getMoreResults(int current)</code>	3.0	No	
<code>int getQueryTimeout()</code>	3.0	Yes	
<code>ResultSet getResultSet()</code>	3.0	Yes	
<code>int getResultSetConcurrency()</code>	3.0	Yes	Hard-coded to CONCUR_READ_ONLY
<code>int getResultSetHoldability()</code>	3.0	Yes	Hard-coded to CLOSE_CURSORS_AT_COMMIT
<code>int getResultSetType()</code>	3.0	Yes	Hard-coded to TYPE_FORWARD_ONLY
<code>int getUpdateCount()</code>	3.0	Yes	
<code>SQLWarning getWarnings()</code>	3.0	Yes	
<code>boolean isClosed()</code>	4.0	Yes	
<code>boolean isCloseOnCompletion()</code>	4.1	Yes	
<code>boolean isPoolable()</code>	4.0	Yes	
<code>void setCursorName(String name)</code>	3.0	No	
<code>void setEscapeProcessing(boolean enable)</code>	3.0	Yes	
<code>void setFetchDirection(int direction)</code>	3.0	No	
<code>void setFetchSize(int rows)</code>	3.0	Yes	
<code>void setMaxFieldSize(int max)</code>	3.0	Yes	

Method	Supported since JDBC version	Supported by the driver	Notes
<code>void setMaxRows(int max)</code>	3.0	Yes	
<code>void setPoolable(boolean poolable)</code>	4.0	Yes	
<code>void setQueryTimeout(int seconds)</code>	3.0	Yes	
<code>boolean isWrapperFor(Class<?> iface)</code>	4.0	Yes	
<code><T> T unwrap(Class<T> iface)</code>	4.0	Yes	

Table 13 Methods in the Statement Class

Contact Us

If you have difficulty using the driver, you can contact Cloudera Technical Support. We welcome your questions, comments and feature requests.

Important:

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

For details on contacting Technical Support, see
<http://www.cloudera.com/content/cloudera/en/products/cloudera-support.html>

Appendix A: 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:

1. To download the Kerberos installer for 64-bit computers, 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>

The 64-bit installer includes both 32-bit and 64-bit libraries.

OR

To download the Kerberos installer for 32-bit computers, 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>

The 32-bit installer includes 32-bit libraries only.


2. To run the installer, double-click the .msi file that you downloaded in step 1.
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 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, 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: **C:\KerberosTickets.txt**
7. Click **OK** to save the new variable.
8. Ensure that the variable appears in the **System variables** list.

Appendix A: Configuring Kerberos Authentication for Windows

9. Click **OK** to close the Environment Variables dialog box, and then click **OK** to close the System Properties dialog box.
10. To ensure that Kerberos uses the new settings, restart your computer.

Getting a Kerberos Ticket

To get a Kerberos ticket:

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

To authenticate to the Impala Server:

- Use a connection string that has the following properties defined:
 - AuthMech
 - KrbHostFQDN
 - KrbRealm
 - KrbServiceName


For detailed information about these properties, see *Appendix B: Driver Configuration Options* on page 64.

Using the Driver to Get Tickets

Deleting the KRB5CCNAME Environment Variable

To enable the driver to get Ticket Granting Tickets (TGTs) directly, you must ensure 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**

Appendix A: Configuring Kerberos Authentication for Windows

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. Ensure 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** environment variable to the location of the JAAS file.

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

Authenticating to the Impala Server

To authenticate to the Impala Server:

- Use a connection string that has the following properties defined:
 - AuthMech
 - KrbHostFQDN
 - KrbRealm
 - KrbServiceName

For detailed information about these properties, see *Appendix B: Driver Configuration Options* on page 64.

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 source
            e.printStackTrace();
        }
        catch (Exception e)
        {
            // Handle other errors
            e.printStackTrace();
        }

        return null;
    }
}
```

Appendix A: Configuring Kerberos Authentication for Windows

```
}
```

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;
```

Appendix B: Driver Configuration Options

Table 14 lists and describes the properties available to configure the behavior of the Cloudera JDBC Driver for Impala.

Note: You can set configuration properties using the connection URL. For details on the connection URL, see *Building the Connection URL* on page 4.

Key	Default Value	Description
AllowSelfSignedCerts	0	<p>When set to the default value of 0, the SSL certificate used by the server cannot be self-signed.</p> <p>When set to 1, the SSL certificate used by the server can be self-signed.</p> <p>Note: The setting is only applicable to the SSL authentication mechanisms.</p> <p>(Optional)</p>
AuthMech	0	<p>The authentication mechanism to use. Set the value to one of the following:</p> <ul style="list-style-type: none"> • 0 for No Authentication • 1 for Kerberos • 2 for User Name • 3 for User Name and Password • 4 for User Name and Password with Secure Sockets Layer • 5 for No Authentication with Secure Sockets Layer <p>(Optional)</p>
CAIssuedCertNamesMismatch	0	<p>When this property is set to 0, the name of the CA-issued SSL certificate must match the host name of the Impala server.</p> <p>When this property is set to 1, the names of the certificate and the host name of the server are allowed to mismatch.</p> <p>Note: This property is applicable only to the SSL authentication mechanisms.</p> <p>(Optional)</p>

Appendix B: Driver Configuration Options

Key	Default Value	Description
CatalogSchemaSwitch	0	When this property is set to 1, the driver treats Impala catalogs as schemas as a restriction for filtering. When this property is set to 0, Impala catalogs are treated as catalogs, and Impala schemas are treated as schemas. (Optional)
DefaultStringColumnLength	255	The maximum data length for string columns. Note: Impala does not provide the maximum data length for String columns in the columns metadata. This option allows you to tune the maximum data length for String columns. (Optional)
DelegationUID	None	Used to delegate all operations against Impala to a user that is different than the authenticated user for the connection. (Optional)
KrbHostFQDN	None	The fully qualified domain name of the Impala host. (Required if AuthMech is Kerberos)
KrbRealm	Depends on Kerberos configuration	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 option. (Optional)
KrbServiceName	None	The Kerberos service principal name of the Impala server. (Required if AuthMech is Kerberos)

Appendix B: Driver Configuration Options

Key	Default Value	Description
PreparedMetalimitZero	1	When this property is set to 1, the <code>PreparedStatement.getMetadata()</code> call will request metadata from the server with "LIMIT 0", increasing performance. (Optional)
PWD	None	The password corresponding to the user name that you provided in the UID property. (Required if AuthMech is set to User Name and Password or User Name and Password with SSL)
RowsFetchedPerBlock	10000	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. (Optional)
SocketTimeout	0	The number of seconds after which Impala closes the connection with the client application if the connection is idle. The default value of 0 indicates that an idle connection is not closed. (Optional)
SSLKeyStore	None	The full path and file name of the Java KeyStore containing an SSL certificate to use during authentication. See also the <code>SSLKeyStorePwd</code> property. (Required if AuthMech is User Name and Password with SSL or No Authentication with SSL)
SSLKeyStorePwd	None	The password required to access the Java KeyStore specified in the <code>SSLKeyStore</code> property. (Required if AuthMech is User Name and Password with SSL or No Authentication with SSL)

Appendix B: Driver Configuration Options

Key	Default Value	Description
SSLTrustStore	jssecacerts , if it exists. If jssecacerts does not exist, then cacerts is used. The default location of cacerts is <code>jre\lib\security</code>	The full path and file name of the Java TrustStore containing an SSL certificate to use during authentication. See also the SSLTrustStorePwd property. (Optional)
SSLTrustStorePwd	None	The password required to access the Java TrustStore specified in the SSLTrustStore property. (Required if using a TrustStore)
UID	anonymous	The user name that you use to access the Impala server. (Required if AuthMech is set to User Name and Password or User Name and Password with Secure Sockets Layer . Optional if AuthMech is set to User Name)
UseNativeQuery	0	When this option is enabled (1), the driver does not transform the queries emitted by an application, so the native query is used. When this option is disabled (0), the driver transforms the queries emitted by an application 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. (Optional)

Table 14 Driver Configuration Options