

## Database Application Access Examples

Date published: 2020-08-14

Date modified: 2023-01-12

# CLOUDERA

# Legal Notice

© Cloudera Inc. 2025. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

**Connect your Go applications to Cloudera Operational Database.....4**

**Connect your Python applications to Cloudera Operational Database.....5**

# Connect your Go applications to Cloudera Operational Database

You can connect your applications written in the Go programming language to a Cloudera Operational Database through Phoenix.

## Go application to connect to Cloudera Operational Database

Before you write an application to connect to Cloudera Operational Database, do the following:

- [Create a Cloudera Operational Database instance](#)
- [Get the Phoenix \(Thin\) client connectivity information](#)
- [Set your Cloudera workload password](#)

This application code is an example for connecting to a Cloudera Operational Database instance, creating a table, inserting a row, and then reading from the Cloudera Operational Database instance.

```
package main

import (
    "database/sql"
    "log"

    _ "github.com/apache/calcite-avatica-go"
)

func main() {
    // Connections are defined by a DSN
    // The format is http://address:port[/schema][?parameter1=value&...parameterN=value]
    // For Cloudera Operational Database, BASIC authentication is used.
    // The workload username and password are passed as parameters avaticaUser and avaticaPassword
    //
    // For example:
    // Cloudera Operational Database URL: 'https://gateway.env.cloudera.site/cdp-proxy-api/avatica/'
    // Workload username: jgoodson
    // Workload password: Secret1!
    // Would result in this DSN:
    dsn := "https://gateway.env.cloudera.site/cdp-proxy-api/avatica/?&authentication=BASIC&avaticaUser=jgoodson&avaticaPassword=Secret1!"

    log.Println("Connecting...")
    db, err := sql.Open("avatica", dsn)
    if err != nil {
        log.Fatal("Connection: ", err)
    }
    defer db.Close()

    log.Println("Create table if not exists...")
    _, err = db.Exec("CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY, username VARCHAR)")
    if err != nil {
        log.Fatal("Create: ", err)
    }

    log.Println("Insert a row...")
    _, err = db.Exec("UPSERT INTO users VALUES (?, ?)", 1, "admin")
}
```

```
if err != nil {
    log.Println("Insert: ", err)
}

log.Println("Reading and printing rows...")
var (
    id      int
    username string
)
rows, err := db.Query("SELECT id, username from users")
if err != nil {
    log.Fatal("Query: ", err)
}
defer rows.Close()
for rows.Next() {
    err := rows.Scan(&id, &username)
    if err != nil {
        log.Fatal(err)
    }
    log.Println(id, username)
}
}
```

### Related Information

[How to connect Go Applications to Cloudera Operational Database](#)

## Connect your Python applications to Cloudera Operational Database

You can connect your applications written in Python programming language to a Cloudera Operational Database through Phoenix.

### Python application to connect to Cloudera Operational Database

Before you write an application to connect to Cloudera Operational Database, do the following:

- [Create a Cloudera Operational Database instance](#)
- [Get the Phoenix \(Thin\) client connectivity information](#)
- [Set your Cloudera workload password](#)

This application code is an example of validating your connection to Cloudera Operational Database.

```
import phoenixdb
import configparser

class Database:

    def connect(self):
        REQUIRED_OPTS = ['Username', 'Password', 'Url']
        config = configparser.ConfigParser()
        config.read('config.ini')
        if not 'COD' in config:
            raise Exception("Could not find section for COD in config.ini")
        cod_config = config['COD']
        opts = {}

        # Validate the configuration
        for required_opt in REQUIRED_OPTS:
            if not required_opt in cod_config:
```

```

        raise Exception("Did not find %s in configuration" % (required_opt))

    # Provide non-required options
    if 'Truststore' in cod_config:
        opts['verify'] = cod_config['Truststore']
    if 'Authentication' in cod_config:
        opts['authentication'] = cod_config['Authentication']
    else:
        opts['authentication'] = 'BASIC'
    # Read required options
    opts['avatica_user'] = cod_config['Username']
    opts['avatica_password'] = cod_config['Password']
    return phoenixdb.connect(cod_config['Url'], autocommit=True, **opts)

```

This code is an example of creating a table and adding data to a Cloudera Operational Database instance.

```

import phoenixdb
from database import Database

def load():
    db = Database()
    conn = db.connect()
    cursor = conn.cursor()
    cursor.execute("CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY
, username VARCHAR)")
    cursor.execute("UPSERT INTO users VALUES (?, ?)", (1, 'admin'))
    cursor.execute("SELECT * FROM users")
    print(cursor.fetchall())
    conn.close()

if __name__ == '__main__':
    load()

```

### Related Information

[Building a Simple CRUD web application and image store using Cloudera Operational Database and Flask](#)