

# DBeaver对接FusionInsight

## 适用场景

DBeaver4.0.8 ↔ FusionInsight V100R002C60U20

## Linux下DBeaver连接Fiber

### 操作场景

以安全模式为例，使用DBeaver通过Fiber访问Hive、Spark、Phoenix

### 前提条件

- 已经安装好Linux（Redhat Linux Enterprise 6.5 64bit）Desktop操作系统；
- 已经安装好的Linux机器的时间与FusionInsight HD集群的时间要保持一致，时间差小于5分钟。
- 已完成FusionInsight HD V100R002C60U20安全集群的安装，已安装好Fiber客户端。

### 操作步骤

- 安装jdk1.8，DBeaver4.0.8需要jdk1.8以上版本

```
tar -xvf jdk-8u112-linux-x64.tar.gz
```

- 配置环境变量/etc/profile，加入如下内容，source环境变量

```
#configure java
export JAVA_HOME=/opt/jdk1.8.0_112
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export PATH=$JAVA_HOME/bin:$PATH
```

- 下载地址：<http://dbeaverjkiss.org/download/>，软件 [dbeaver-ce-4.0.8-linux.gtk.x86\\_64.tar.gz](#)，安装DBeaver

```
tar -xvf dbeaver-ce-4.0.8-linux.gtk.x86_64.tar.gz
```

- 安装FusionInsight客户端，具体请参见《FusionInsight HD 产品文档》的 安装客户端 章节，客户端安装目录为 [/opt/hadoopclient/](#)，其中Fiber客户端目录 [/opt/hadoopclient/Fiber/](#)。
- 修改Fiber的配置文件 [/opt/hadoopclient/Fiber/conf/fiber.xml](#)，将其中hive、spark、phoenix的认证方式改为安全模式keytab认证方式，具体配置方法参考产品文档->管理员指南->业务操作指南->统一SQL(Fiber)->客户端配置章节。

### Hive JDBC连接配置

```
<jdbc>
<identify>hive</identify>
<describe>hive jdbc configuration</describe>
<driverClass>org.apache.hive.jdbc.HiveDriver</driverClass>
<securityClass>com.huawei.fiber.DefaultAuthenticationCallback</securityClass>
<classPath>/opt/hadoopclient/Hive/config:/opt/hadoopclient/Hive/Beeline/lib:/opt/hadoopclient/Hive/Beeline/conf</classPath>
<jdbcUrl>jdbc:hive2://162.1.93.103:24002,162.1.93.102:24002,162.1.93.101:24002;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hive
server2;sasl.qop=auth-
conf;auth=KERBEROS;principal=hive/hadoop.hadoop.com@HADOOP.COM;user.principal=test;user.keytab=/opt/user.keytab</jdbcUrl>
<properties>
<property>
<name>java.security.krb5.conf</name>
<value>/opt/hadoopclient/Hive/..KrbClient/kerberos/var/krb5kdc/krb5.conf</value>
</property>
<property>
<name>java.security.auth.login.config</name>
<value>/opt/jaas.conf</value>
</property>
<property>
<name>zookeeper.server.principal</name>
<value>{HIVE_CLIENT_ZK_PRINCIPAL}</value>
</property>
<property>
<name>zookeeper.kinit</name>
<value>/opt/hadoopclient/Hive/..KrbClient/kerberos/bin/kinit</value>
```

```
</property>
</properties>
</jdbc>
```

#### Spark连接配置

```
<jdbc>
<identify>spark</identify>
<describe>spark jdbc configuration</describe>
<driverClass>org.apache.hive.jdbc.HiveDriver</driverClass>
<securityClass>com.huawei.fiber.DefaultAuthenticationCallback</securityClass>
<classPath>/opt/hadoopclient/Spark/spark/conf:/opt/hadoopclient/Spark/spark/lib</classPath>
<jdbcUrl>jdbc:hive2://ha-cluster/default;saslQop=auth-
conf;auth=KERBEROS;principal=spark/hadoop.hadoop.com@HADOOP.COM;user.principal=test;user.keytab=/opt/user.keytab</jdbcUrl>
<properties>
<property>
<name>java.security.krb5.conf</name>
<value>/opt/hadoopclient/KrbClient/kerberos/var/krb5kdc/krb5.conf</value>
</property>
<property>
<name>java.security.auth.login.config</name>
<value>/opt/jaas.conf</value>
</property>
<property>
<name>zookeeper.server.principal</name>
<value>zookeeper/hadoop.hadoop.com</value>
</property>
<property>
<name>zookeeper.kinit</name>
<value>/opt/hadoopclient/KrbClient/kerberos/bin/kinit</value>
</property>
</properties>
</jdbc>
```

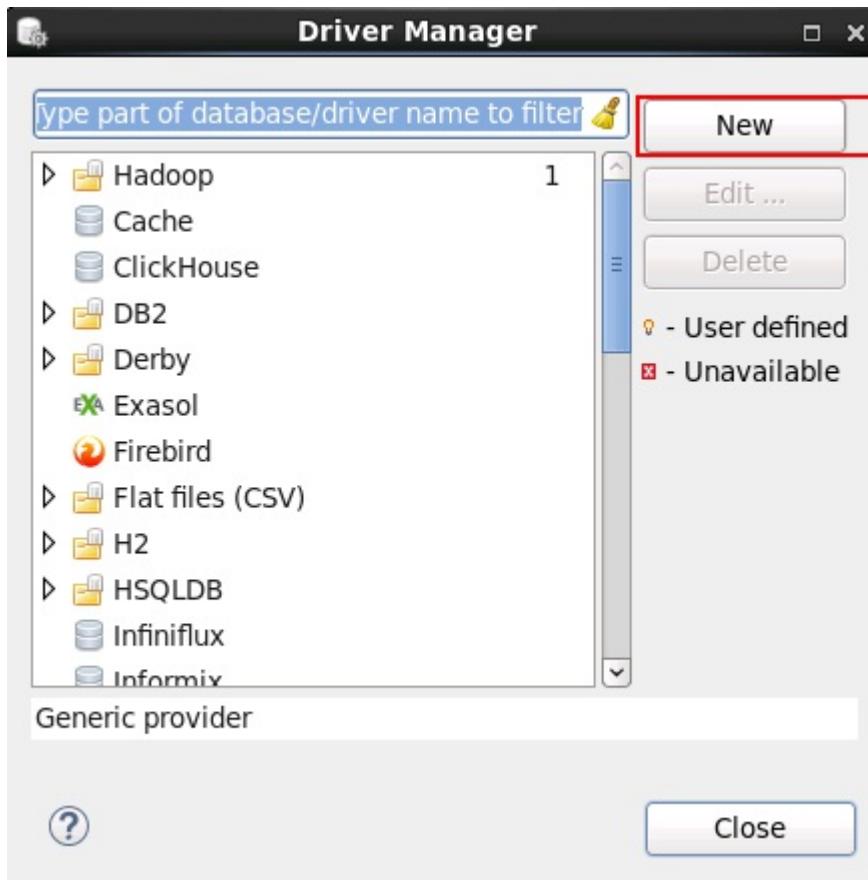
#### Phoenix连接配置

```
<jdbc>
<identify>phoenix</identify>
<describe>phoenix jdbc configuration</describe>
<driverClass>org.apache.phoenix.jdbc.PhoenixDriver</driverClass>
<securityClass>com.huawei.fiber.DefaultAuthenticationCallback</securityClass>
<classPath>/opt/hadoopclient/HBase/hbase/lib:/opt/hadoopclient/HBase/hbase/conf</classPath>
<jdbcUrl>jdbc:phoenix:162.1.93.101,162.1.93.102,162.1.93.103:24002:/hbase:test:/opt/user.keytab</jdbcUrl>
<properties>
<property>
<name>java.security.krb5.conf</name>
<value>/opt/hadoopclient/HBase/..../KrbClient/kerberos/var/krb5kdc/krb5.conf</value>
</property>
<property>
<name>java.security.auth.login.config</name>
<value>/opt/jaas.conf</value>
</property>
<property>
<name>zookeeper.server.principal</name>
<value>zookeeper/hadoop.hadoop.com</value>
</property>
<property>
<name>zookeeper.kinit</name>
<value>/opt/hadoopclient/HBase/..../KrbClient/kerberos/bin/kinit</value>
</property>
</properties>
</jdbc>
```

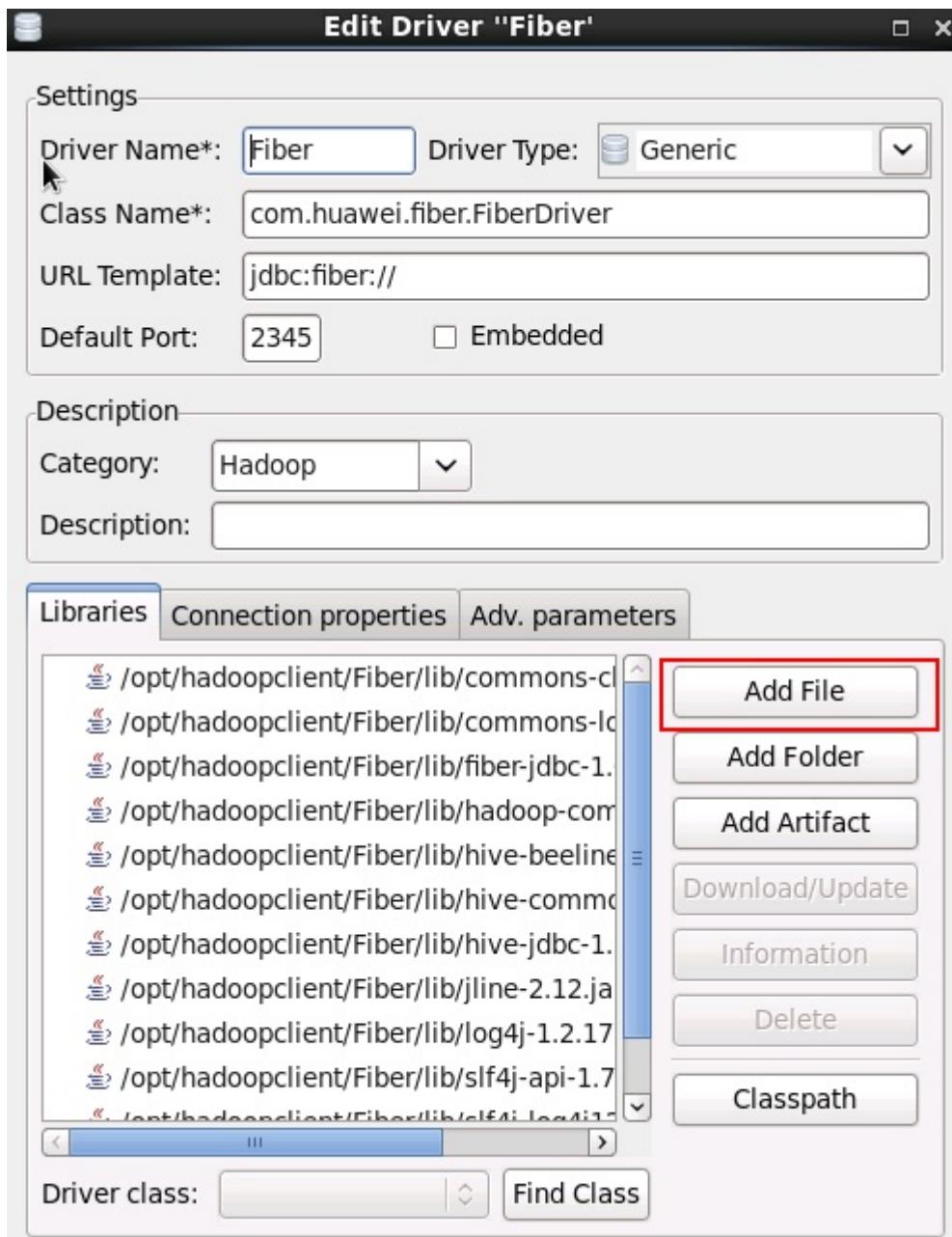
#### jaas.conf文件:

```
Client {
    com.sun.security.auth.module.Krb5LoginModule required
    useKeyTab=true
    keyTab="/opt/user.keytab"
    principal="test"
    useTicketCache=false
    storeKey=true
    debug=true;
};
```

- 打开DBeaver，进入DBeaver的安装目录执行 `./dbeaver`，启动dbeaver
- 进入DBeaver界面，菜单选择 **Database -> 新建 DriverManager**，在弹出的对话框中点击 **New**



- 填写如下信息，点击 **OK**
- Driver Name: `Fiber (自定义)`
- Class Name: `com.huawei.fiber.FiberDriver`
- URL Template: `jdbc:fiber://`
- Default Port: `2345 (可随便写)`
- Category: `Hadoop`
- 点击 **Add File** 按钮，将Fiber客户端 `/opt/hadoopclient/Fiber/lib/` 下的jar包添加进来
  - commons-cli-1.2.jar
  - commons-logging-1.1.3.jar
  - fiber-jdbc-1.0.jar
  - hadoop-common-2.7.2.jar
  - hive-beeline-1.2.1.spark.jar
  - hive-common-1.2.1.spark.jar
  - jline-2.12.jar
  - log4j-1.2.17.jar
  - slf4j-api-1.7.10.jar
  - slf4j-log4j12-1.7.10.jar
  - super-csv-2.2.0.jar



- 在Connection Properties中加入以下属性:

Settings

Driver Name*	Fiber	Driver Type:	Generic
Class Name*	com.huawei.fiber.FiberDriver		
URL Template:	jdbc:fiber://		
Default Port:	2345	<input type="checkbox"/> Embedded	

Description

Category:	Hadoop
Description:	

Libraries Connection properties Adv. parameters

Name	Value
User Properties	
fiberconfig	/opt/hadoopclient/Fiber/conf/fiber.xml
defaultDrive	hive

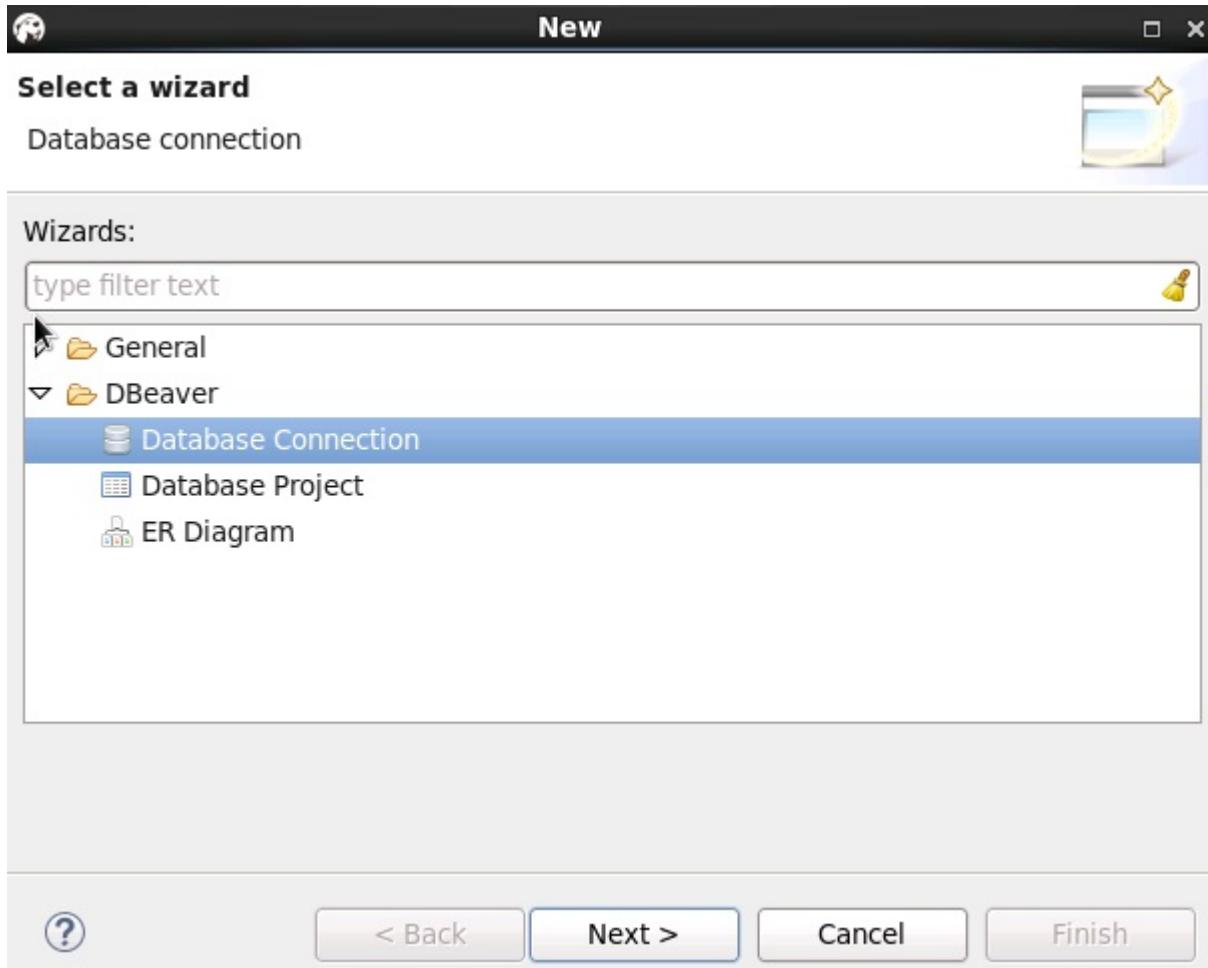
?

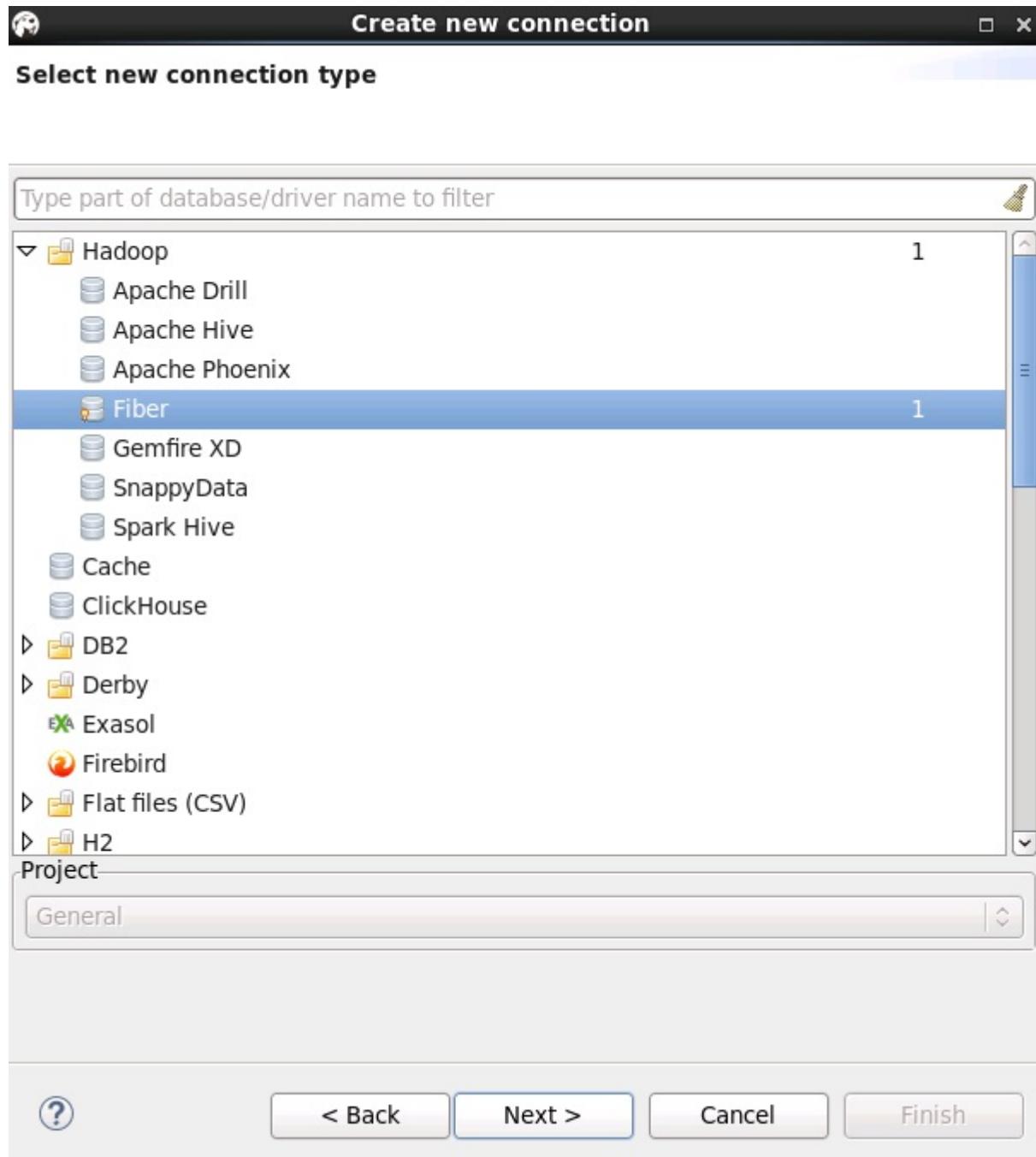
Reset to Defaults

Cancel

OK

- 菜单栏选择 **File -> New -> Database Connection**, 类型选择Fiber





User name和Password可不填写

**Create new connection**

**Generic JDBC Connection Settings**  
Hadoop / Fiber connection settings

**General** Driver properties

JDBC URL: `jdbc:fiber://`

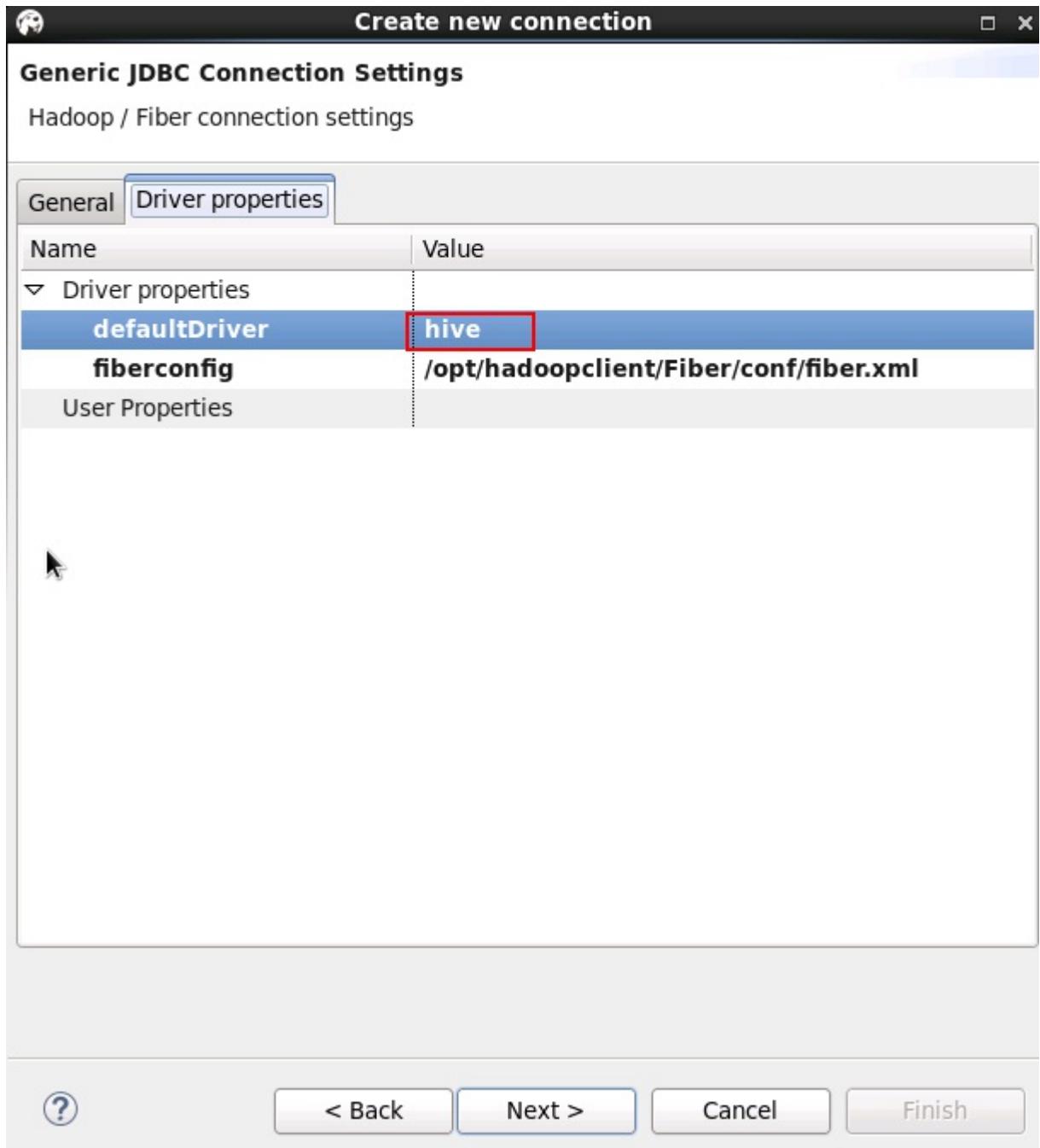
User name:

Password:

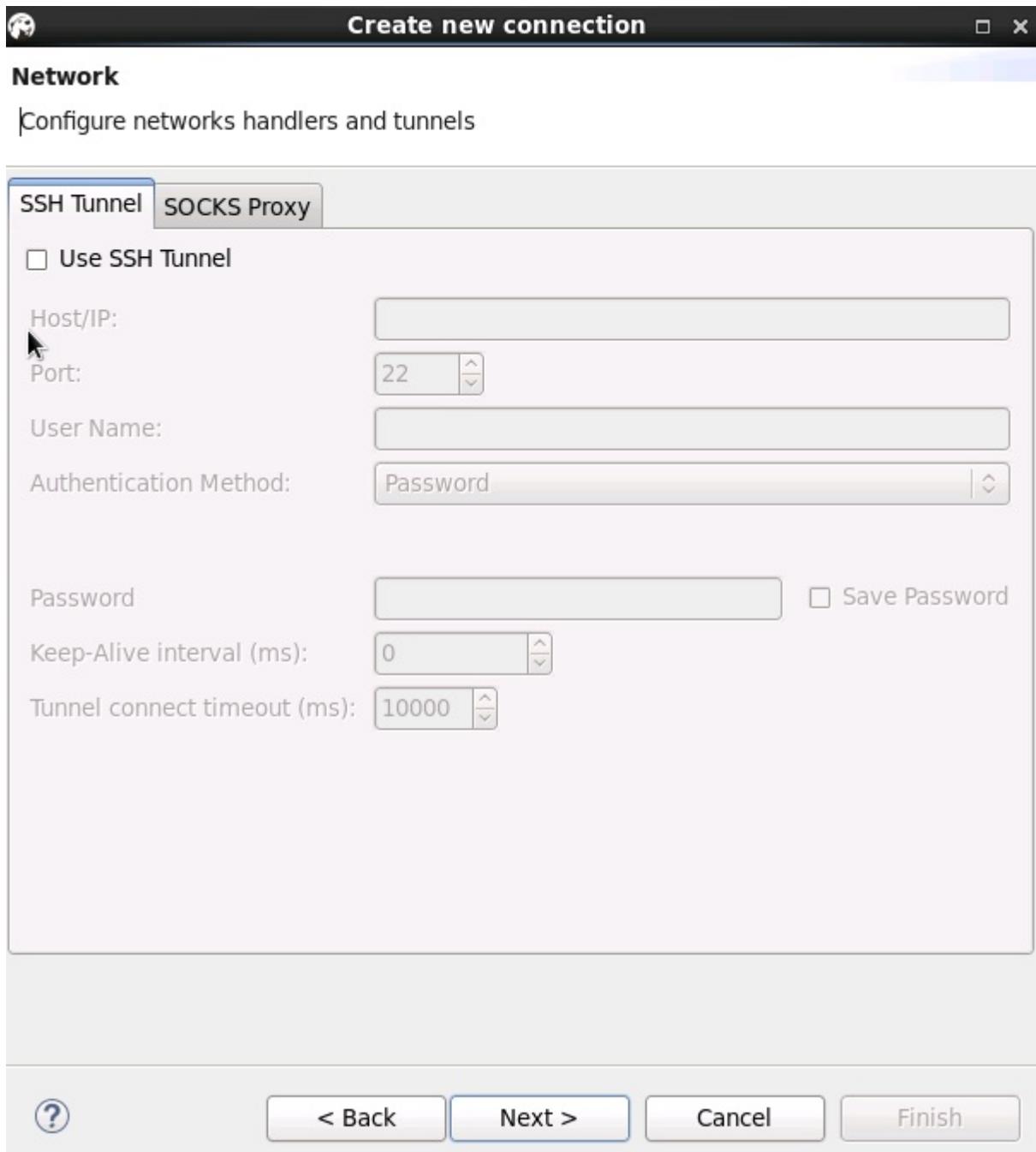
Driver Name: Hadoop / Fiber Edit Driver Settings

? < Back Next > Cancel Finish

配置Driver properties里面的defaultDirver, 可按需求填写hive或spark或phoenix, 点击next



Network页面保持默认，点击 next



输入自定义Connection name后，点击 **finish**，连接建立完成



## Create new connection



### Finish connection creation

General connection settings.

Connection name: Hadoop - Fiber 自定义名称

Connection type: Development   Edit

Connection folder: <None>  

#### Security

Save password locally

#### Miscellaneous

Show system objects

Show utility objects

Read-only connection

#### Filters

[Catalogs](#)

[Schemas / Users](#)

[Tables](#)

#### Connection

Auto-commit:  测试环境下可勾选自动提交

Isolation level:  

Default schema:  

Keep-Alive: 0 ^

Bootstrap queries: Configure ...

Shell Commands: Configure ...

#### Description



< Back

Next >

Cancel

Finish

- 测试hive链接

## Driver properties

JDBC driver properties

### Connection settings

#### Driver properties

Network

Shell Commands

General

Metadata

### Result Sets

Name	Value
------	-------

#### Driver properties

defaultDriver: hive

fiberconfig: /opt/hadoopclient/Fiber/conf/fiber.xml

#### User Properties

Success



Server: Apache Hive 1.3.0

Driver: Hive JDBC 1.3.0

S

Connected (2160 ms)

OK



Cancel

Test Connection ...

OK

查看Hive表中数据

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database structure under 'Hadoop-Fiber'. The 'Tables' section contains 'fiber\_test', 'pokes', 'test', and 'workers\_info'. The 'workers\_info' table is selected and highlighted with a blue background. The main workspace shows the 'Data' tab of the 'workers\_info' table. The table has four columns: 'workers\_info.id', 'workers\_info.name', 'workers\_info.usd\_flag', and 'workers\_info.salary'. There are two rows of data: Row 1 with id 1, name Wang, usd\_flag R, and salary 8,000.01; and Row 2 with id 2, name Chen, usd\_flag F, and salary 2,334.01. Below the table, a message indicates '2 row(s) fetched - 237ms (+68ms)'.

- 测试spark链接, 把driver切换为spark, 连接右键选择 **Edit Connection**

The screenshot shows the MySQL Workbench interface with a context menu open over a connection entry in the sidebar. The menu options include 'Connect', 'Invalidate/Reconnect', 'Disconnect', 'SQL Editor' (F3), 'Recent SQL Editor' (Ctrl+Enter), 'Edit Connection' (highlighted with a red box), 'Create New Connection', 'New Folder', 'Copy' (Ctrl+C), 'Paste' (Ctrl+V), 'Delete' (Delete key), 'Rename' (F2), 'Properties' (Alt+Enter), and 'Refresh' (F5). The 'Edit Connection' option is the target of the instruction.

## Driver properties

JDBC driver properties

### Connection settings

#### Driver properties

Network

Shell Commands

General

Metadata

### Result Sets

Bind Variables

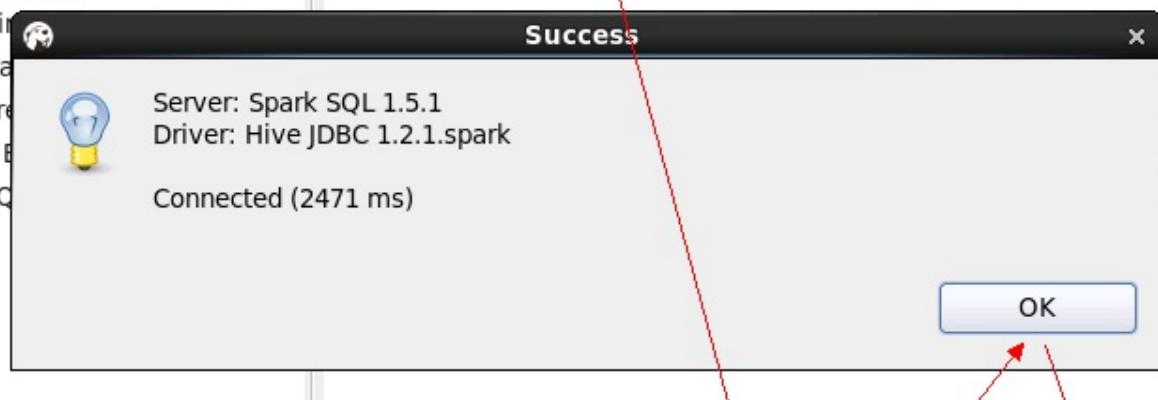
Data Types

Precursors

### SQL Editor

SQL Editor

Name	Value
Driver properties	
defaultDriver	spark
fiberconfig	
/opt/hadoopclient/Fiber/conf/fiber.xml	
User Properties	



Cancel

Test Connection ...

OK

**Driver properties**

JDBC driver properties

Connection settings

- Driver properties (selected)
- Network
- Shell Commands
- General
- Metadata

Result Sets

- Binaries
- Data Formatting
- Presentation

SQL Editor

- SQL Processing

Name	Value
Driver properties	
defaultDriver	spark
fiberconfig	/opt/hadoopclient/Fiber/conf/fiber.xml
User Properties	

**Connection changed**

Connection 'Hadoop-Fiber' has been changed.  
Do you want to reconnect?

No Yes (button highlighted with red box)

Cancel Test Connection ... OK

使用spark driver查看表中数据

File Edit Navigate Search SQL Editor Database Window Help

Hadoop-Fiber <None> 200 Commit Rollback Auto

Database N Projects

Type table/view name to filter

Hadoop-Fiber

- Tables
  - fiber\_test
  - pokes
  - test
  - workers\_info (selected)
- Views
- Indexes
- Procedures

workers\_info

Properties Data Diagram

workers\_info | Enter a SQL expression to filter results

	id	name	usd_flag	salary	address	entrytime
1	1	Wang	R	8,000.01	Chian:Shenzhen	2014
2	2	Chen	F	2,334.01	Chian:Shanghai	2015

Project - General

Name DataSource

Save Cancel Script Record Panels Grid Text

2 row(s) fetched - 816ms (+41ms)

- 测试phoenix连接，把driver切换为phoenix，连接右键选择 **Edit Connection**

## Driver properties

JDBC driver properties

Connection settings

- Driver properties (selected)
- Network
- Shell Commands
- General
- Metadata

Result Sets

- Binary
- Data
- Present

SQL Editor

- SQL

Name	Value
Driver properties	
defaultDriver	phoenix
fiberconfig	/opt/hadoopclient/Fiber/conf/fiber.xml
User Properties	

**Success**

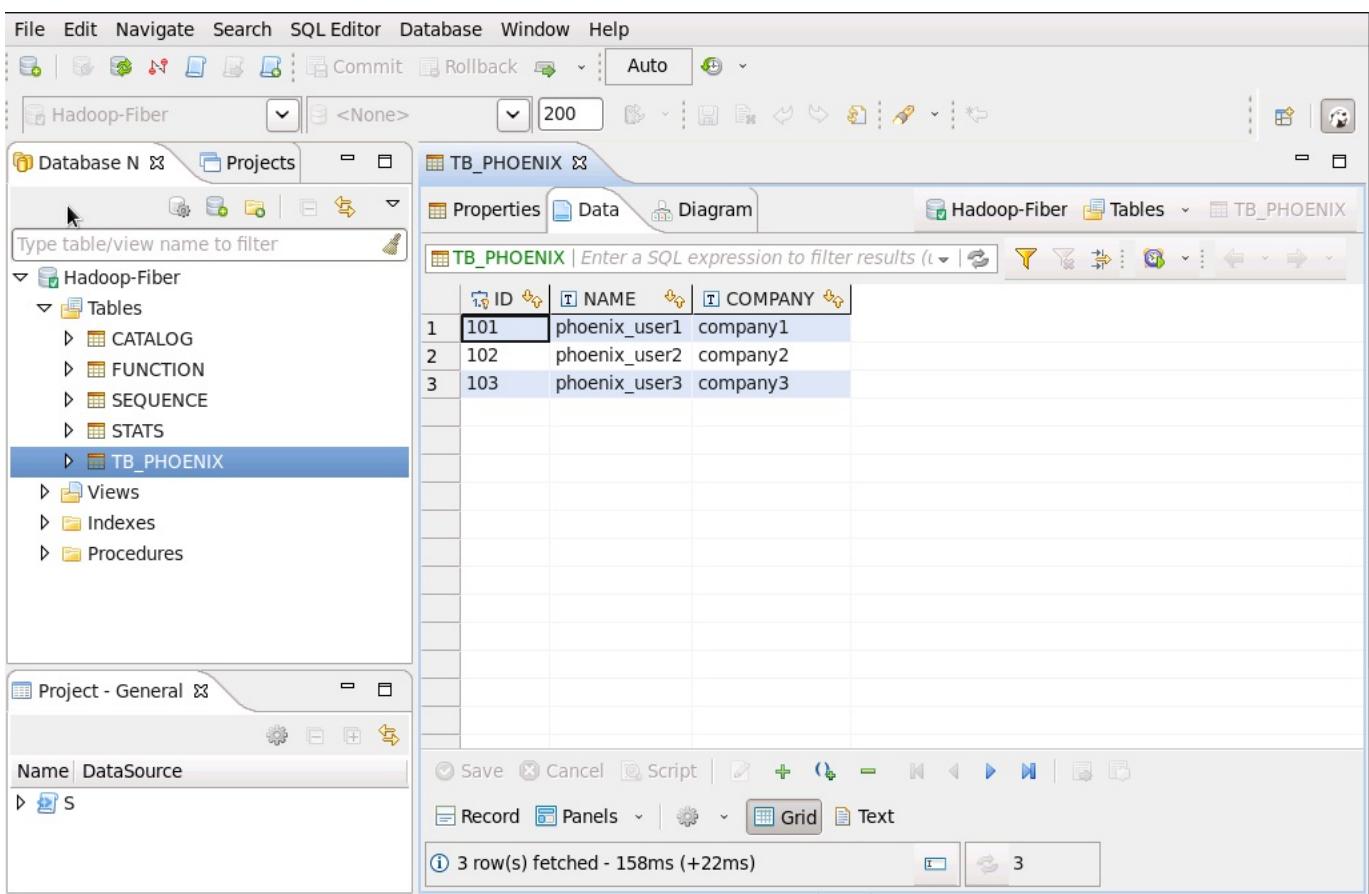
Server: Phoenix 4.4  
Driver: PhoenixEmbeddedDriver 4.4

Connected (4733 ms)

OK

Cancel Test Connection ... OK

查看phoenix表中数据



## Windows下DBeaver连接Fiber

### 操作场景

以安全模式为例，使用DBeaver通过Fiber访问Hive、Spark、Phoenix

### 前提条件

- Windows上已经安装好jdk1.8以上版本，并完成jdk环境变量配置
- 客户端机器的时间与FusionInsight HD集群的时间要保持一致，时间差小于5分钟。
- 从<http://dbeaverjkiss.org/download/>下载DBeaver软件，完成windows上的安装
- 已完成FusionInsight HD V100R002C60U20安全集群的安装，已安装好Fiber客户端。
- 已将集群的节点主机名与IP的映射关系加入到windows的hosts文件中 `C:\Windows\System32\drivers\etc\hosts`

### 操作步骤

Fiber的安全认证可以用kinit和keytab两种方式，具体参数配置说明可参考 [产品文档 -> 管理员指南 -> 业务操作指南 -> 统一SQL\(Fiber\) -> 客户端配置](#) 章节。kinit认证的有效期是24小时，keytab认证方式长期有效

- 使用kinit认证方式配置
- 使用keytab认证方式配置

#### 使用kinit认证方式配置

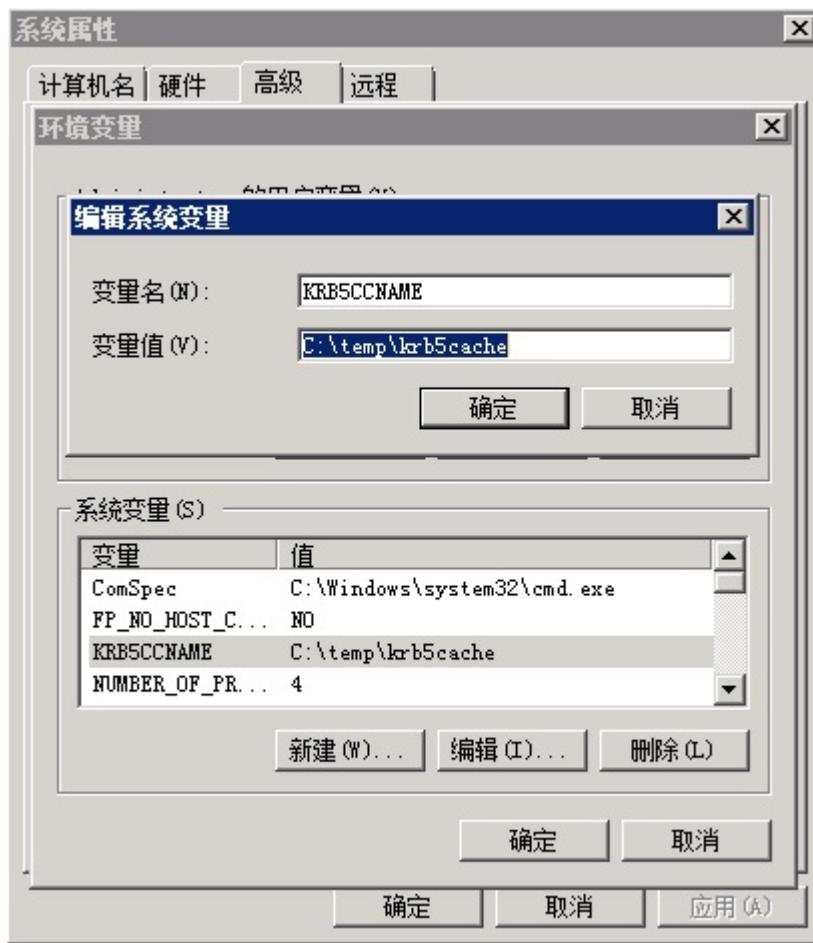
- 下载对应操作系统架构的MIT Kerberos，并安装 <http://web.mit.edu/kerberos/dist/#kfw-4.0>
- 确认客户端机器的时间与FusionInsight HD集群的时间一致，时间差要小于5分钟
- 设置Kerberos的配置文件

在FusionInsight Manager创建角色和人机用户，具体请参见 [产品文档 -> 管理员指南 -> 系统设置 -> 权限管理 -> 用户管理 -> 创建用户](#) 章节。角色需要根据业务需要授予Hive的访问权限，并将用户加入角色，创建用户“test”

下载对应的keytab文件 `user.keytab` 以及 `krb5.conf` 文件，把 `krb5.conf` 文件重命名为 `krb5.ini`，并放到 `C:\ProgramData\MIT\Kerberos5` 目录中

- 设置Kerberos票据的缓存文件

创建存放票据的目录，例如 `C:\temp` 设置Windows的系统环境变量，变量名为 `KRB5CCNAME`，变量值为 `C:\temp\krb5cache`



- 在Windows上进行认证

打开MIT Kerberos，单击 **get Ticket**，在弹出的MIT Kerberos: Get Ticket窗口中，**Principal** 输入用户名(如: test@HADOOP.COM)，**Password** 输入密码，单击 **OK**



- 在本地PC机上新建一个目录，将FusionInsight客户端下的fiber客户端文件夹Fiber拷贝至本地，例如 **C:\Fiber**
- 将FusionInsight客户端下 **jaas.conf** 文件和 **krb5.conf** 拷贝到 **C:\Fiber\conf** 目录下，文档内容如下

```
Client {
    com.sun.security.auth.module.Krb5LoginModule required
    useKeyTab=false
    useTicketCache=true
    debug=true;
};
```

- 修改 **fiber.xml** 文件，位置 **C:\Fiber\conf\fiber.xml**

Hive的JDBC连接

```
<jdbc>
<identify>hive</identify>
<describe>hive jdbc configuration</describe>
<driverClass>org.apache.hive.jdbc.HiveDriver</driverClass>
<securityClass>com.huawei.fiber.DefaultAuthenticationCallback</securityClass>
```

```

<classPath>C:\\Fiber\\Hive\\config;C:\\Fiber\\Hive\\Beeline\\lib;C:\\Fiber\\Hive\\Beeline\\conf</classPath>
<jdbcUrl>jdbc:hive2://162.1.93.103:24002,162.1.93.102:24002,162.1.93.101:24002;/serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=
hiveserver2;sasl.qop=auth-conf;auth=KERBEROS;principal=hive/hadoop.com@HADOOP.COM</jdbcUrl>
<properties>
  <property>
    <name>java.security.krb5.conf</name>
    <value>C:\\Fiber\\conf\\krb5.conf</value>
  </property>
  <property>
    <name>java.security.auth.login.config</name>
    <value>C:\\Fiber\\conf\\jaas.conf</value>
  </property>
  <property>
    <name>zookeeper.server.principal</name>
    <value>zookeeper/hadoop.hadoop.com</value>
  </property>
  <property>
    <name>zookeeper.kinit</name>
    <value>C:\\Program Files (x86)\\Java\\jdk1.8.0_112\\jre\\bin\\kinit.exe</value>
  </property>
</properties>
</jdbc>

```

#### Spark的JDBC连接

```

<jdbc>
  <identify>spark</identify>
  <describe>spark jdbc configuration</describe>
  <driverClass>org.apache.hive.jdbc.HiveDriver</driverClass>
  <securityClass>com.huawei.fiber.DefaultAuthenticationCallback</securityClass>
  <classPath>C:\\Fiber\\Spark\\spark\\conf;C:\\Fiber\\Spark\\spark\\lib</classPath>
  <jdbcUrl>jdbc:hive2://ha-cluster/default;saslQop=auth-conf;auth=KERBEROS;principal=spark/hadoop.hadoop.com@HADOOP.COM</jdbcUrl>
  <properties>
    <property>
      <name>java.security.krb5.conf</name>
      <value>C:\\Fiber\\conf\\krb5.conf</value>
    </property>
    <property>
      <name>java.security.auth.login.config</name>
      <value>C:\\Fiber\\conf\\jaas.conf</value>
    </property>
    <property>
      <name>zookeeper.server.principal</name>
      <value>zookeeper/hadoop.hadoop.com</value>
    </property>
    <property>
      <name>zookeeper.kinit</name>
      <value>C:\\Program Files (x86)\\Java\\jdk1.8.0_112\\jre\\bin\\kinit.exe</value>
    </property>
  </properties>
</jdbc>

```

#### Phoenix的JDBC连接

```

<jdbc>
  <identify>phoenix</identify>
  <describe>phoenix jdbc configuration</describe>
  <driverClass>org.apache.phoenix.jdbc.PhoenixDriver</driverClass>
  <securityClass>com.huawei.fiber.DefaultAuthenticationCallback</securityClass>
  <classPath>C:\\Fiber\\HBase\\hbase\\lib;C:\\Fiber\\HBase\\hbase\\conf</classPath>
  <jdbcUrl>jdbc:phoenix</jdbcUrl>
  <properties>
    <property>
      <name>java.security.krb5.conf</name>
      <value>C:\\Fiber\\conf\\krb5.conf</value>
    </property>
    <property>
      <name>java.security.auth.login.config</name>
      <value>C:\\Fiber\\conf\\jaas.conf</value>
    </property>
    <property>
      <name>zookeeper.server.principal</name>
      <value>zookeeper/hadoop.hadoop.com</value>
    </property>
    <property>
      <name>zookeeper.kinit</name>
      <value>C:\\Program Files (x86)\\Java\\jdk1.8.0_112\\jre\\bin\\kinit.exe</value>
    </property>
  </properties>
</jdbc>

```

- DBeaver连接前确认kerberos认证有效



#### 使用keytab认证方式配置

- 在本地PC机上新建一个目录，将FusionInsight客户端下的fiber客户端文件夹Fiber拷贝至本地，例如 C:\Fiber
- 将FusionInsight客户端下 jaas.conf 文件和 krb5.conf 拷贝到 C:\Fiber\conf 目录下，文档内容如下，principal和keytab按实际填写

```
Client {
    com.sun.security.auth.module.Krb5LoginModule required
    useKeyTab=true
    keyTab="C:\\Fiber\\conf\\user.keytab"
    principal="test"
    useTicketCache=false
    storeKey=true
    debug=true;
};
```

- 修改fiberxml文件配置，位置 C:\Fiber\conf\fiber.xml

#### Hive的JDBC连接

```
<jdbc>
<identify>hive</identify>
<describe>hive jdbc configuration</describe>
<driverClass>org.apache.hive.jdbc.HiveDriver</driverClass>
<securityClass>com.huawei.fiber.DefaultAuthenticationCallback</securityClass>

<classPath>C:\\Fiber\\Hive\\config;C:\\Fiber\\Hive\\Beeline\\lib;C:\\Fiber\\Hive\\Beeline\\conf</classPath>
<jdbcUrl>jdbc:hive2://162.1.93.103:24002,162.1.93.102:24002,162.1.93.101:24002/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=
hiveserver2;sasl.qop=auth-
conf;auth=KERBEROS;principal=hive/hadoop.hadoop.com@HADOOP.COM;user.principal=test;user.keytab=C:/Fiber/conf/user.keytab</jdbcUrl>
<properties>
    <property>
        <name>java.security.krb5.conf</name>
        <value>C:\\Fiber\\conf\\krb5.conf</value>
    </property>
    <property>
        <name>java.security.auth.login.config</name>
        <value>C:\\Fiber\\conf\\jaas.conf</value>
    </property>
    <property>
        <name>zookeeper.server.principal</name>
        <value>zookeeper/hadoop.hadoop.com</value>
    </property>
    <property>
        <name>zookeeper.kinit</name>
        <value>C:\\Program Files (x86)\\Java\\jdk1.8.0_112\\jre\\bin\\kinit.exe</value>
    </property>
</properties>
</jdbc>
```

#### Spark的JDBC连接

```
<jdbc>
<identify>spark</identify>
<describe>spark jdbc configuration</describe>
<driverClass>org.apache.hive.jdbc.HiveDriver</driverClass>
<securityClass>com.huawei.fiber.DefaultAuthenticationCallback</securityClass>
<classPath>C:\\Fiber\\Spark\\spark\\conf;C:\\Fiber\\Spark\\spark\\lib</classPath>
<jdbcUrl>jdbc:hive2://ha-cluster/default;saslQop=auth-
conf;auth=KERBEROS;principal=spark/hadoop.hadoop.com@HADOOP.COM;user.principal=test;user.keytab=C:/Fiber/conf/user.keytab</jdbcUrl>
```

```

>
<properties>
    <property>
        <name>java.security.krb5.conf</name>
        <value>C:\\Fiber\\conf\\krb5.conf</value>
    </property>
    <property>
        <name>java.security.auth.login.config</name>
        <value>C:\\Fiber\\conf\\jaas.conf</value>
    </property>
    <property>
        <name>zookeeper.server.principal</name>
        <value>zookeeper/hadoop.hadoop.com</value>
    </property>
    <property>
        <name>zookeeper.kinit</name>
        <value>C:\\Program Files (x86)\\Java\\jdk1.8.0_112\\jre\\bin\\kinit.exe</value>
    </property>
</properties>
</jdbc>

```

Phoenix的JDBC连接，需要增加属性hbase.myclient.keytab和hbase.myclient.principal

```

<jdbc>
    <identify>phoenix</identify>
    <describe>phoenix jdbc configuration</describe>
    <driverClass>org.apache.phoenix.jdbc.PhoenixDriver</driverClass>
    <securityClass>com.huawei.fiber.DefaultAuthenticationCallback</securityClass>
    <classPath>C:\\Fiber\\HBase\\lib;C:\\Fiber\\HBase\\conf</classPath>
    <jdbcUrl>jdbc:phoenix:162.1.93.101,162.1.93.102,162.1.93.103:24002:/hbase</jdbcUrl>
    <properties>
        <property>
            <name>java.security.krb5.conf</name>
            <value>C:\\Fiber\\conf\\krb5.conf</value>
        </property>
        <property>
            <name>java.security.auth.login.config</name>
            <value>C:\\Fiber\\conf\\jaas.conf</value>
        </property>
    <property>
        <name>hbase.myclient.keytab</name>
        <value>C:\\Fiber\\conf\\user.keytab</value>
    </property>
    <property>
        <name>hbase.myclient.principal</name>
        <value>test</value>
    </property>
    <property>
        <name>zookeeper.server.principal</name>
        <value>zookeeper/hadoop.hadoop.com</value>
    </property>
    <property>
        <name>zookeeper.kinit</name>
        <value>C:\\Program Files (x86)\\Java\\jdk1.8.0_112\\jre\\bin\\kinit.exe</value>
    </property>
</properties>
</jdbc>

```

## DBeaver连接Fiber

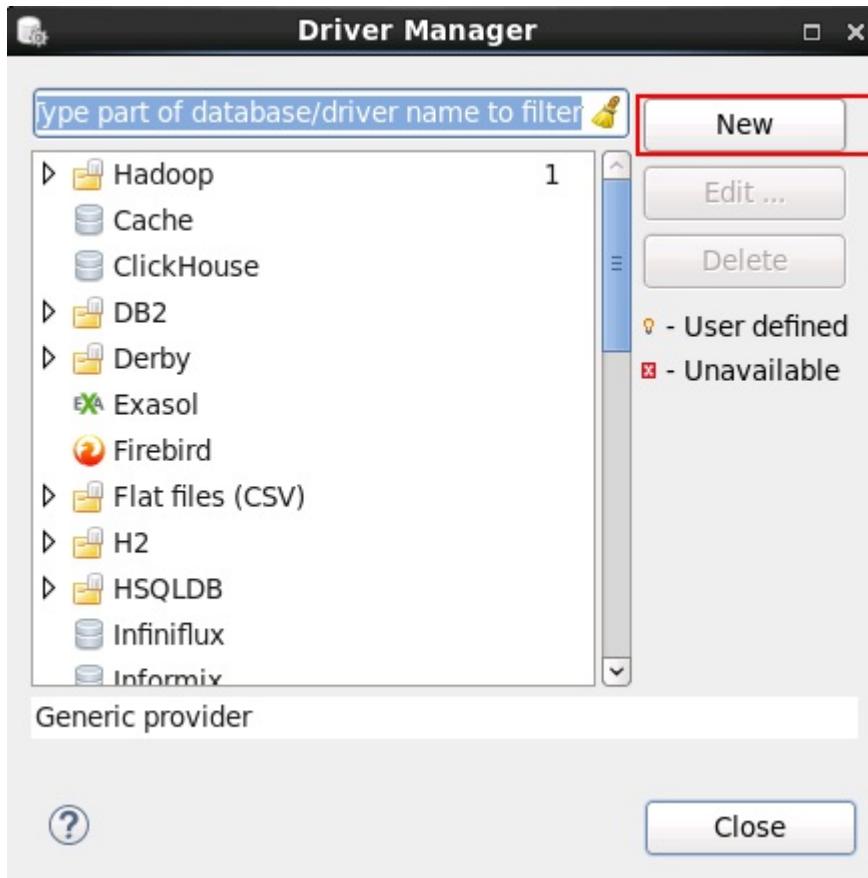
- 将Hive、Spark、Phoenix的JDBC配置中classPath中的文件拷贝至Fiber文件夹中



The screenshot shows the DBeaver interface with the connection configuration for 'Fiber'. The 'File Path' dropdown is set to '本地磁盘 (C:) \ Fiber'. The 'File Path' field contains the path 'C:\\Fiber'. The 'Test Connection' button is visible at the bottom right.

文件夹		
名称	修改日期	类型
bin	2017/6/22 14:57	文件夹
conf	2017/6/22 20:07	文件夹
HBase	2017/6/22 20:24	文件夹
Hive	2017/6/22 15:07	文件夹
lib	2017/6/22 14:57	文件夹
Spark	2017/6/22 20:21	文件夹
component_env	2017/5/20 16:05	文件

- 进入DBeaver界面，菜单选择 Database -> 新建DriverManager，在弹出的对话框中点击 New

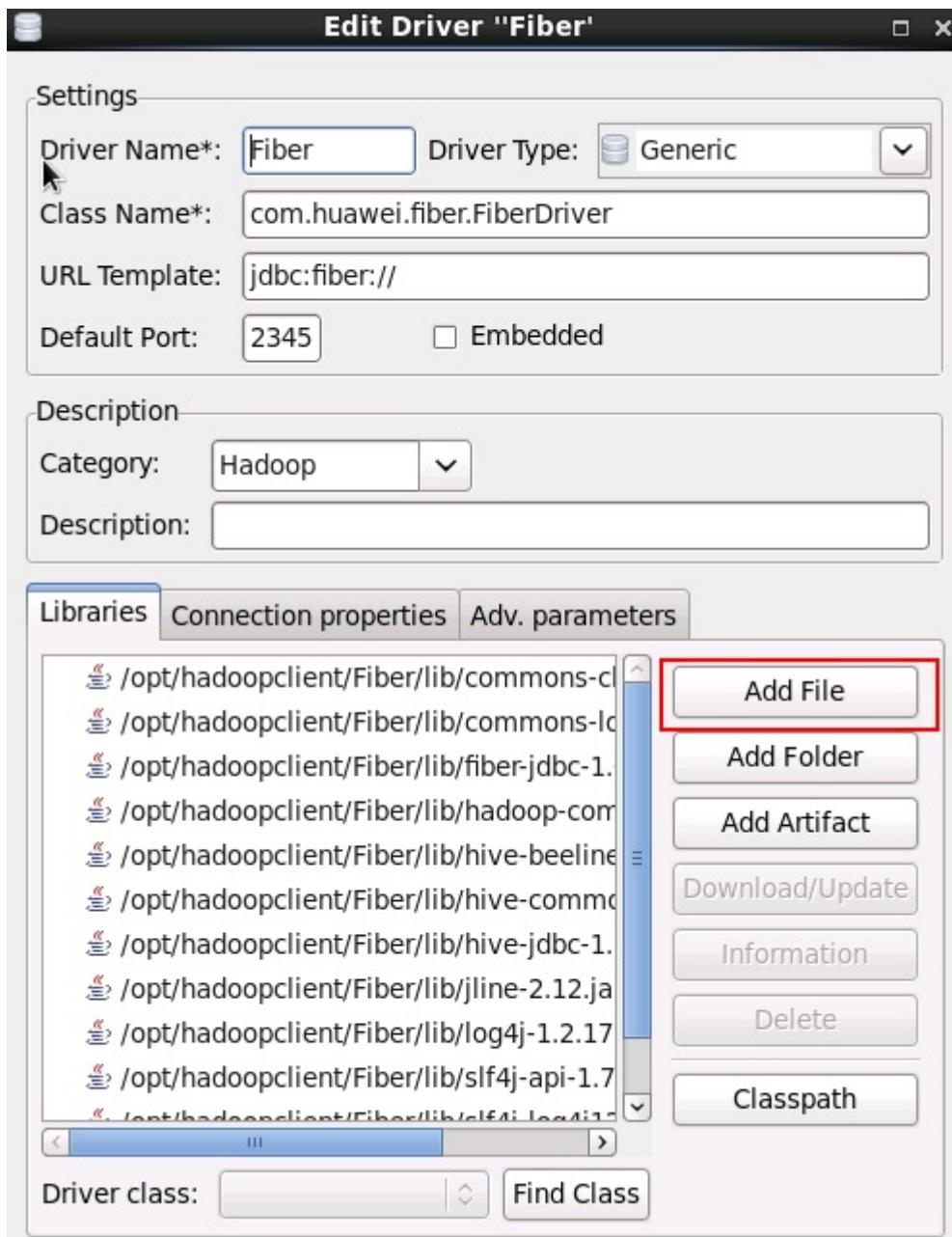


- 填写如下信息，点击 OK

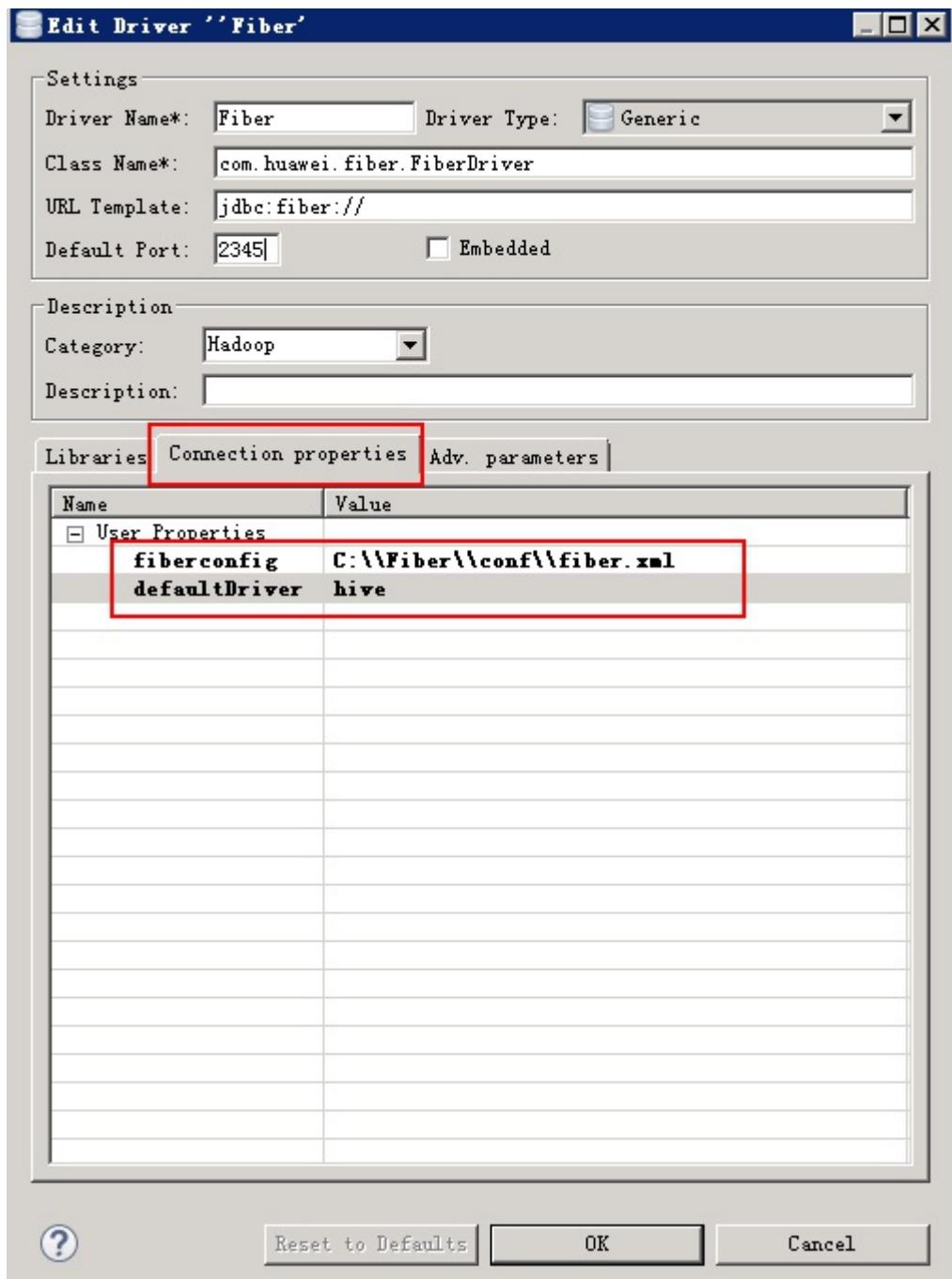
- Driver Name: Fiber (自定义)
- Class Name: com.huawei.fiber.FiberDriver
- URL Template: jdbc:fiber://
- Default Port: 2345 (可随便写)
- Category: Hadoop

- 点击 Add File 按钮，将Fiber客户端（/opt/hadoopclient/Fiber/lib/）下的jar包添加进来

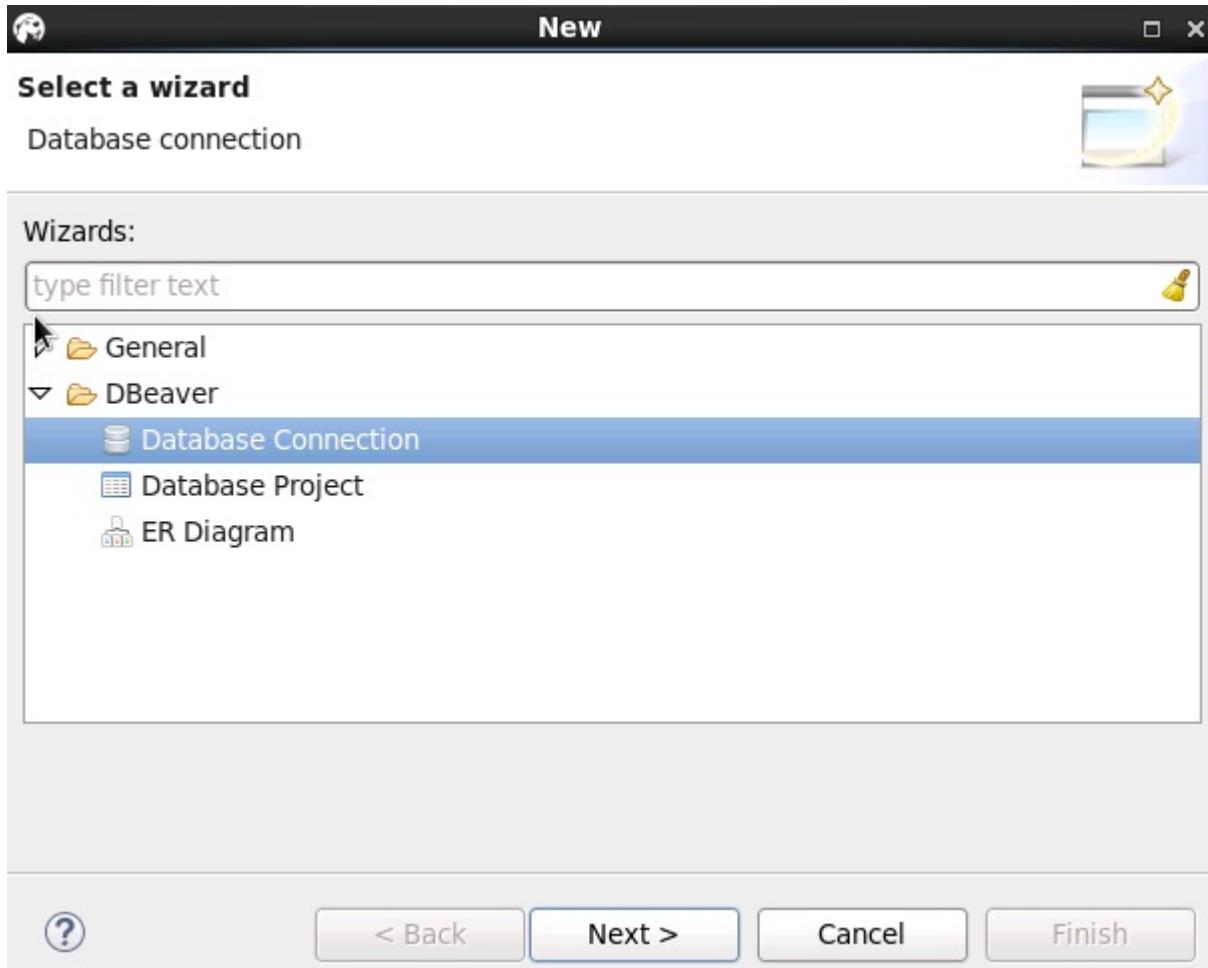
```
[root@localhost dbeaver]# cd /opt/hadoopclient/Fiber/lib/
[root@localhost lib]# ll
total 5164
-rwxr-xr-x 1 root root 41123 May 20 15:56 commons-cli-1.2.jar
-rwxr-xr-x 1 root root 62050 May 20 15:56 commons-logging-1.1.3.jar
-rwxr-xr-x 1 root root 35249 May 20 15:56 fiber-jdbc-1.0.jar
-rwxr-xr-x 1 root root 3637386 May 20 15:56 hadoop-common-2.7.2.jar
-rwxr-xr-x 1 root root 145990 May 20 15:56 hive-beeline-1.2.1.spark.jar
-rwxr-xr-x 1 root root 344702 May 20 15:56 hive-common-1.2.1.spark.jar
-rwxr-xr-x 1 root root 154493 May 20 15:56 hive-jdbc-1.2.1.spark.jar
-rwxr-xr-x 1 root root 213854 May 20 15:56 jline-2.12.jar
-rwxr-xr-x 1 root root 489884 May 20 15:56 log4j-1.2.17.jar
-rwxr-xr-x 1 root root 32119 May 20 15:56 slf4j-api-1.7.10.jar
-rwxr-xr-x 1 root root 8866 May 20 15:56 slf4j-log4j12-1.7.10.jar
-rwxr-xr-x 1 root root 93210 May 20 15:56 super-csv-2.2.0.jar
```

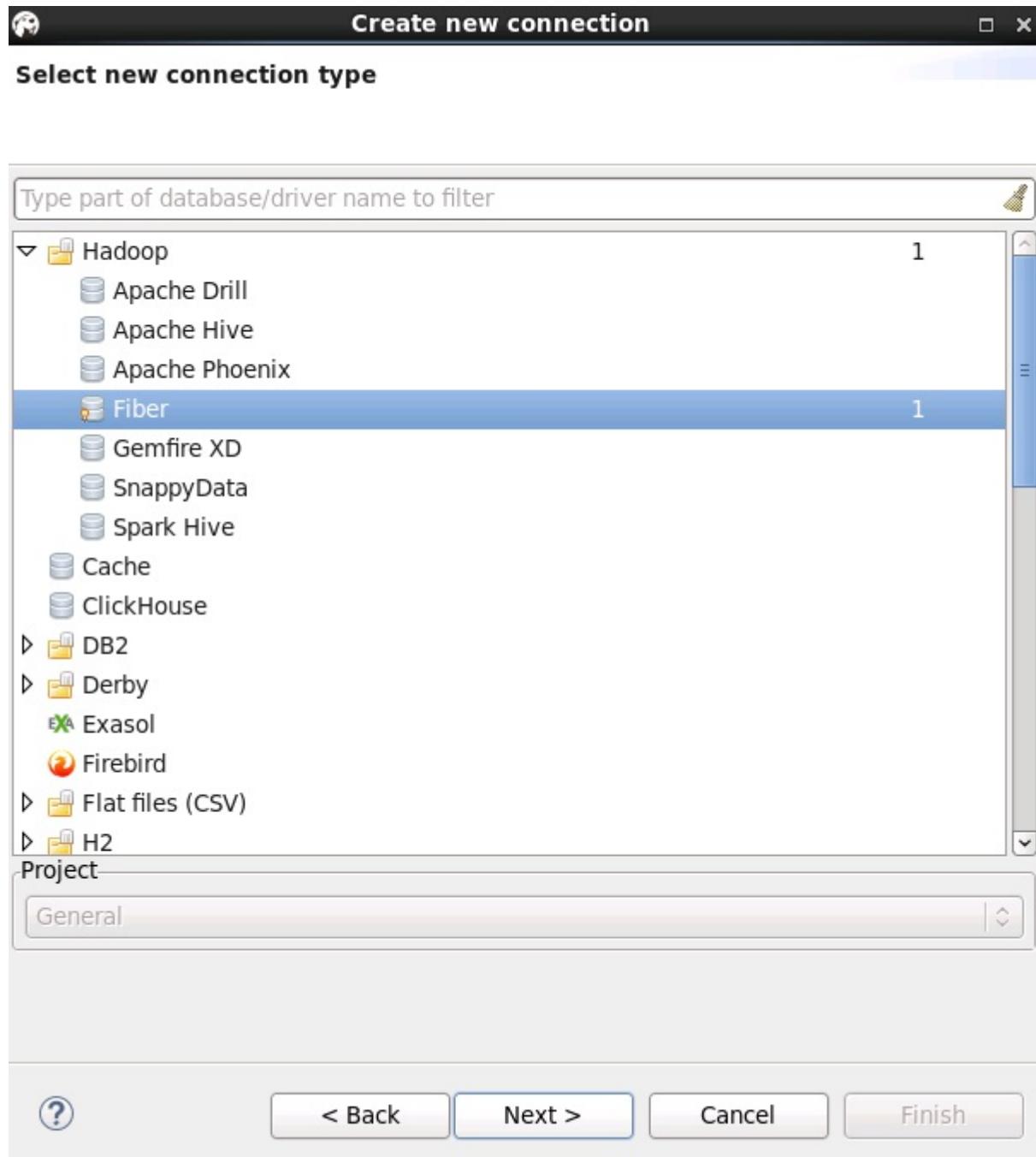


- 在Connection Properties中加入以下属性



- 菜单栏选择 File -> New -> Database Connection





User name和Password可不填写

**Create new connection**

**Generic JDBC Connection Settings**

Hadoop / Fiber connection settings

**General** Driver properties

JDBC URL: `jdbc:fiber://`

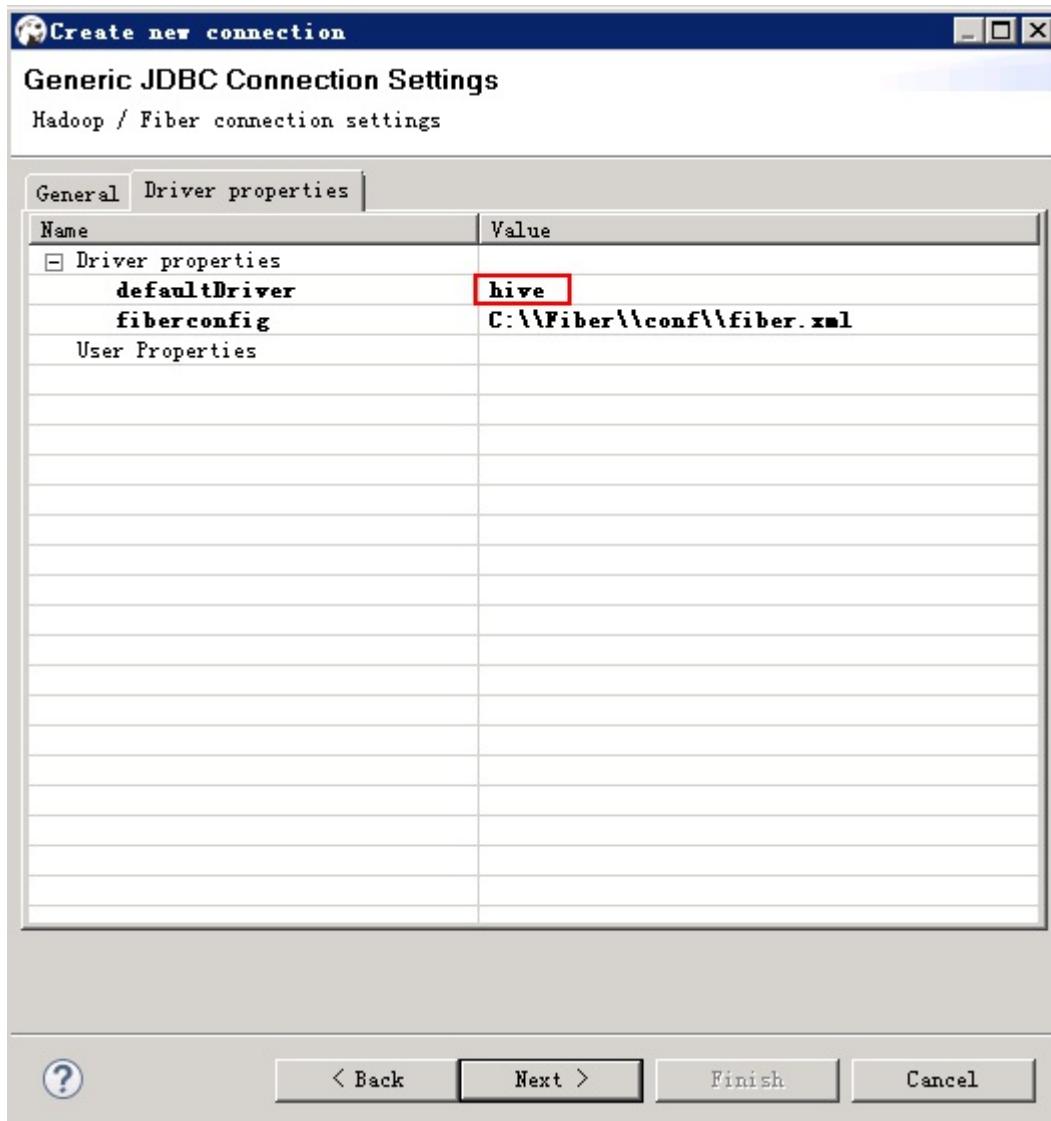
User name:

Password:

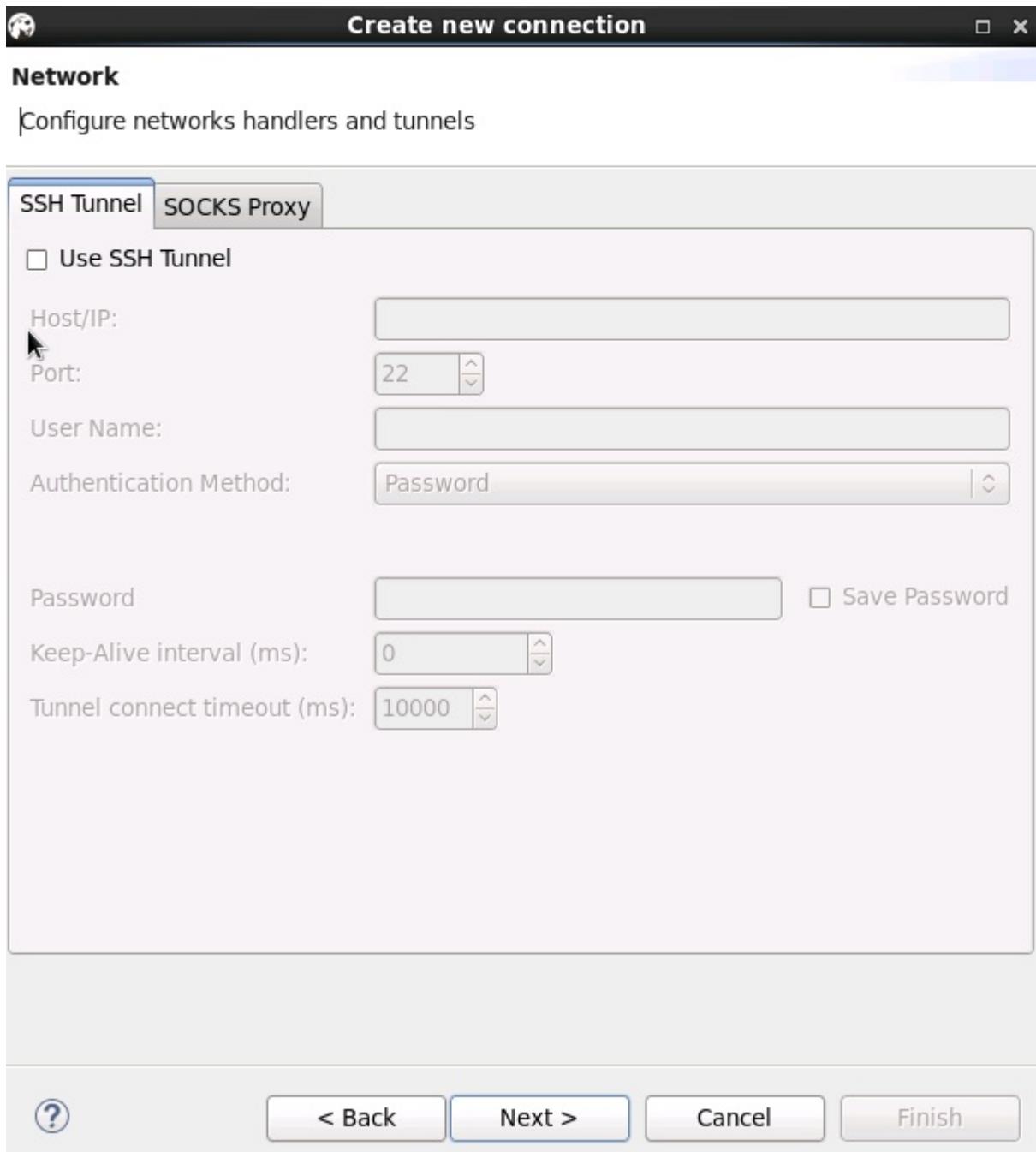
Driver Name: Hadoop / Fiber Edit Driver Settings

? < Back Next > Cancel Finish

确认defaultDirver, 可按需求填写hive或spark或phoenix。



Network保持默认，点击 next



自定义Connection name，点击finish



## Create new connection



### Finish connection creation

General connection settings.

Connection name: Hadoop - Fiber 自定义名称

Connection type: Development   Edit

Connection folder: <None>  

#### Security

Save password locally

#### Miscellaneous

Show system objects

Show utility objects

Read-only connection

#### Filters

[Catalogs](#)

[Schemas / Users](#)

[Tables](#)

#### Connection

Auto-commit:  测试环境下可勾选自动提交

Isolation level:

Default schema:

Keep-Alive:

Bootstrap queries: Configure ...

Shell Commands: Configure ...

#### Description



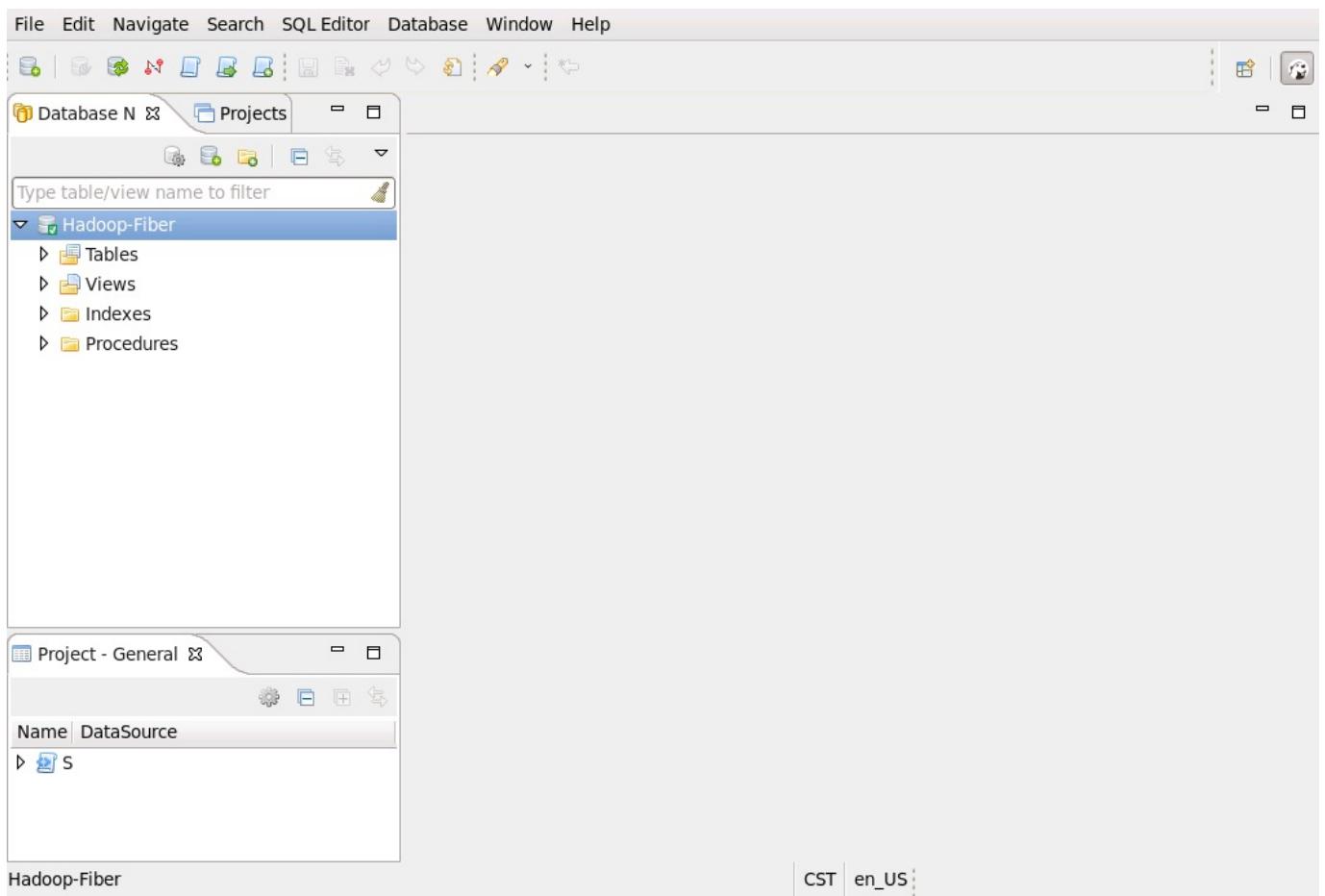
< Back

Next >

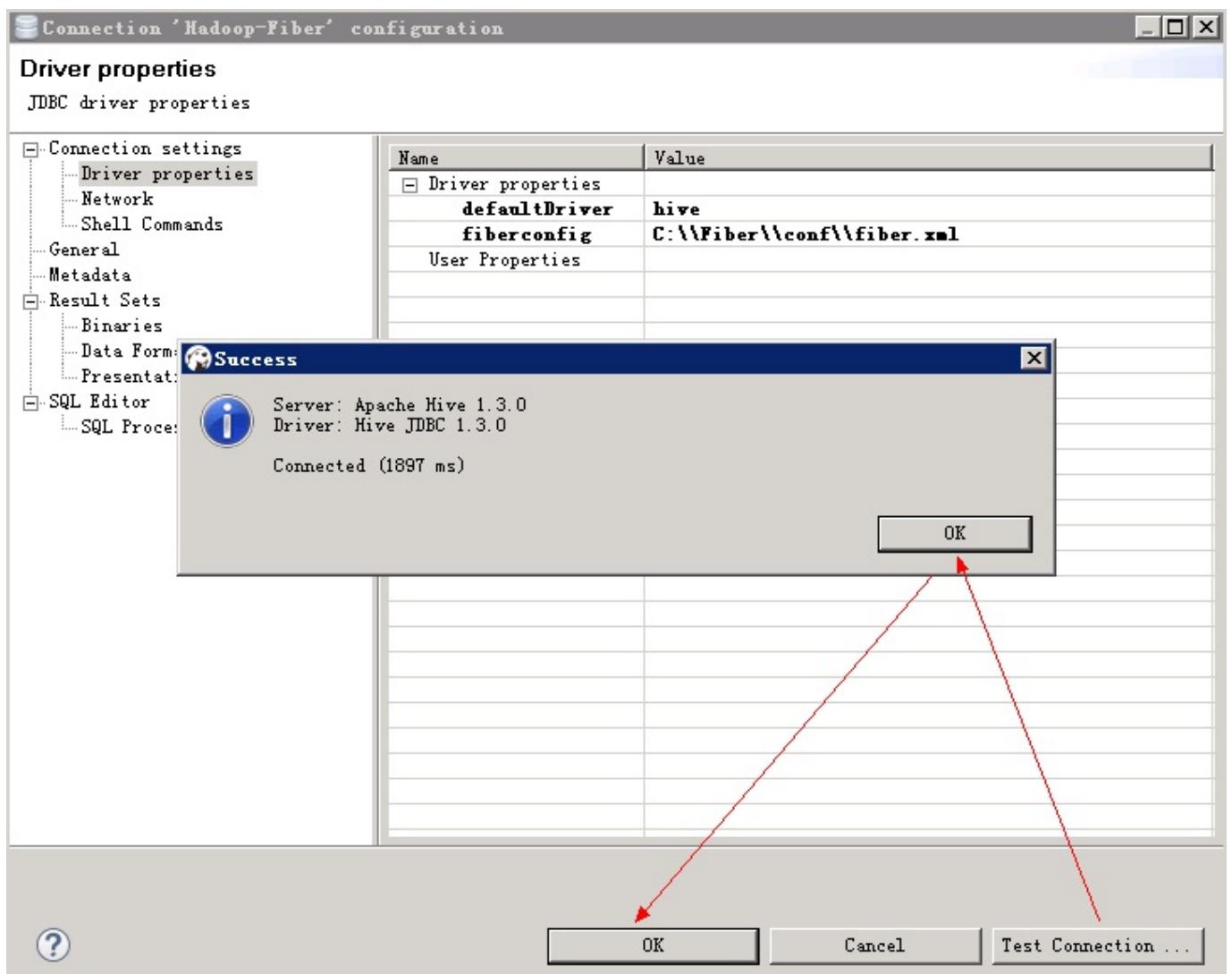
Cancel

Finish

连接建立完成



- 测试hive连接



查看Hive表中数据

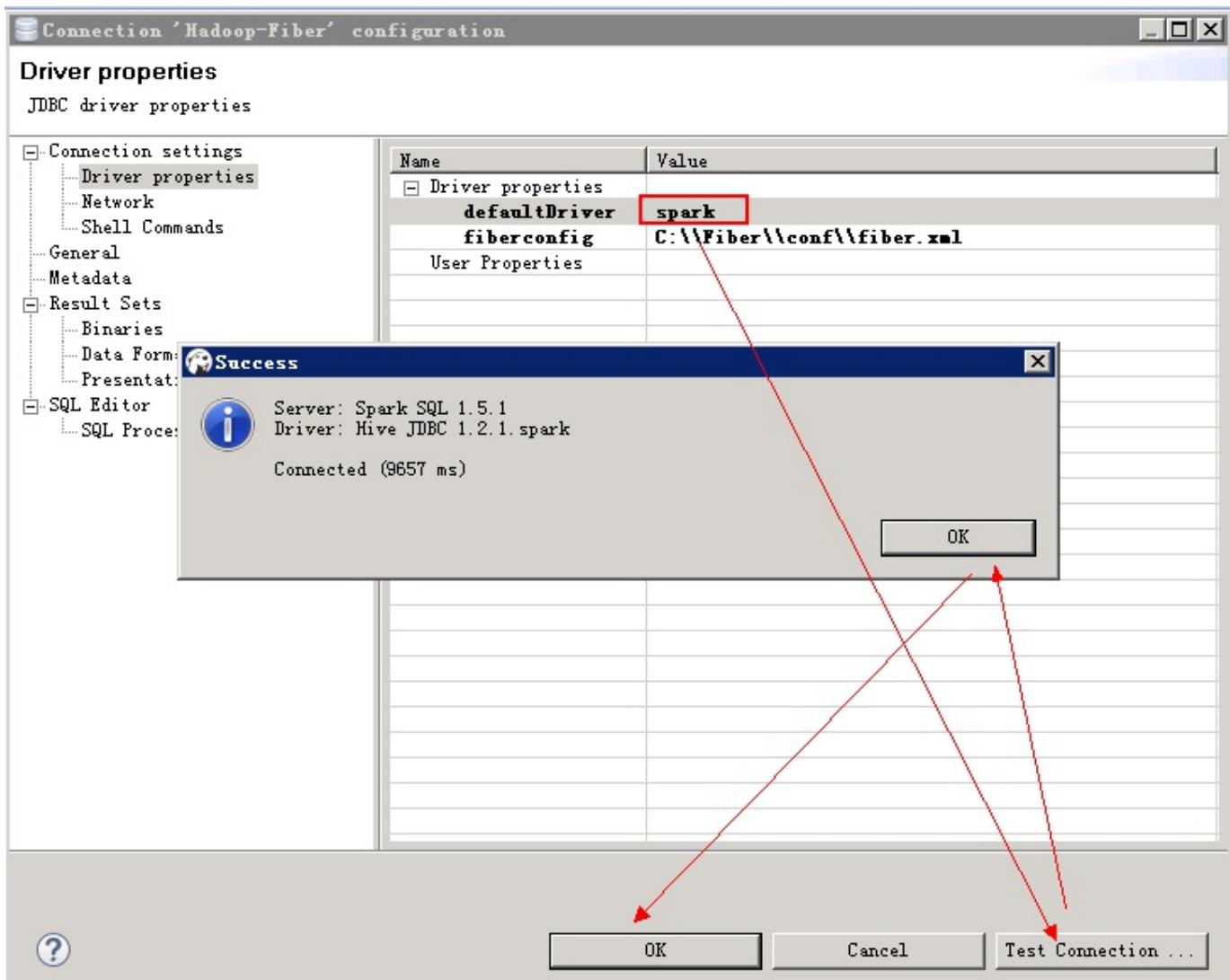
The screenshot shows the DBBeaver interface. In the top navigation bar, the database is set to 'Hadoop-Fiber'. The left sidebar shows the 'Database Navigator' with the 'Tables' section expanded, displaying 'employees\_info', 'fiber\_test', 'pokes', 'test', and 'workers\_info'. The main panel displays the 'workers\_info' table with the following data:

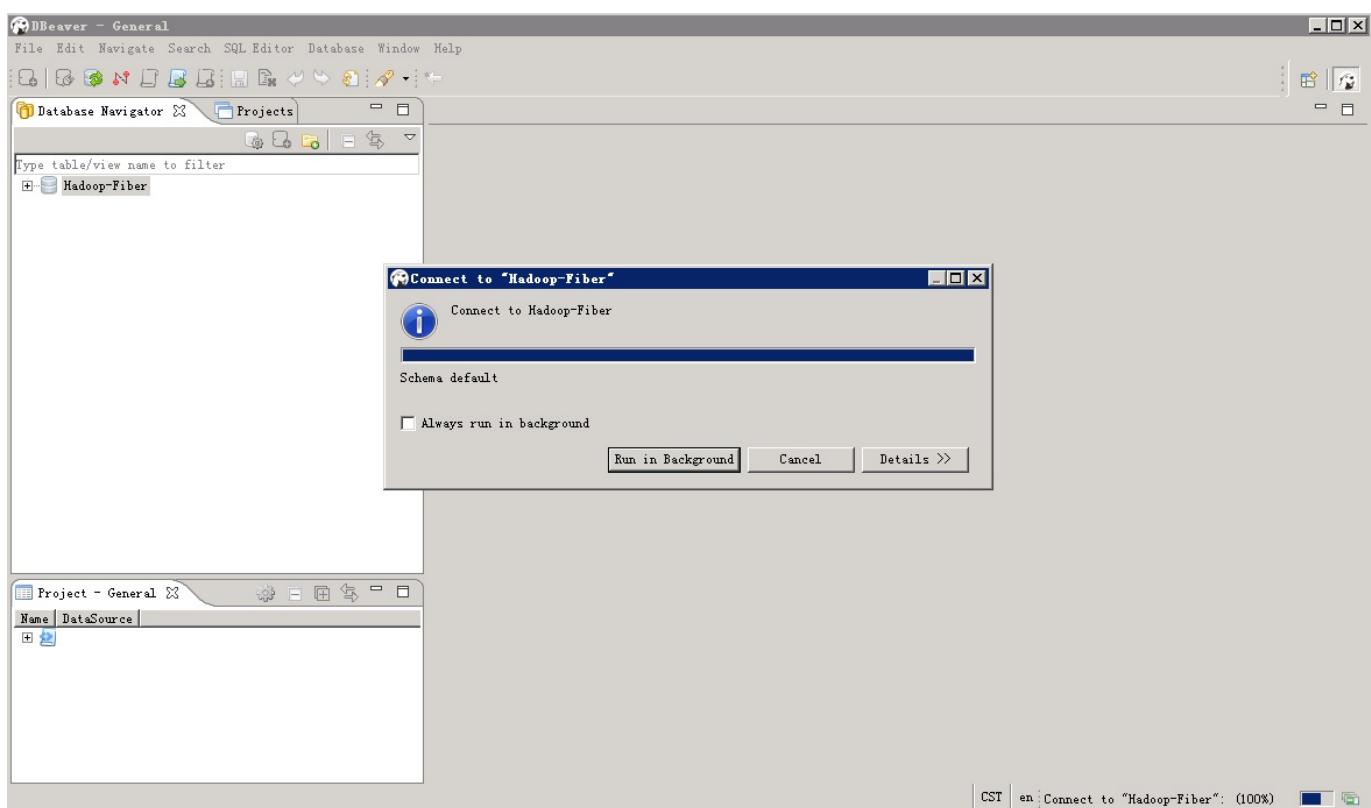
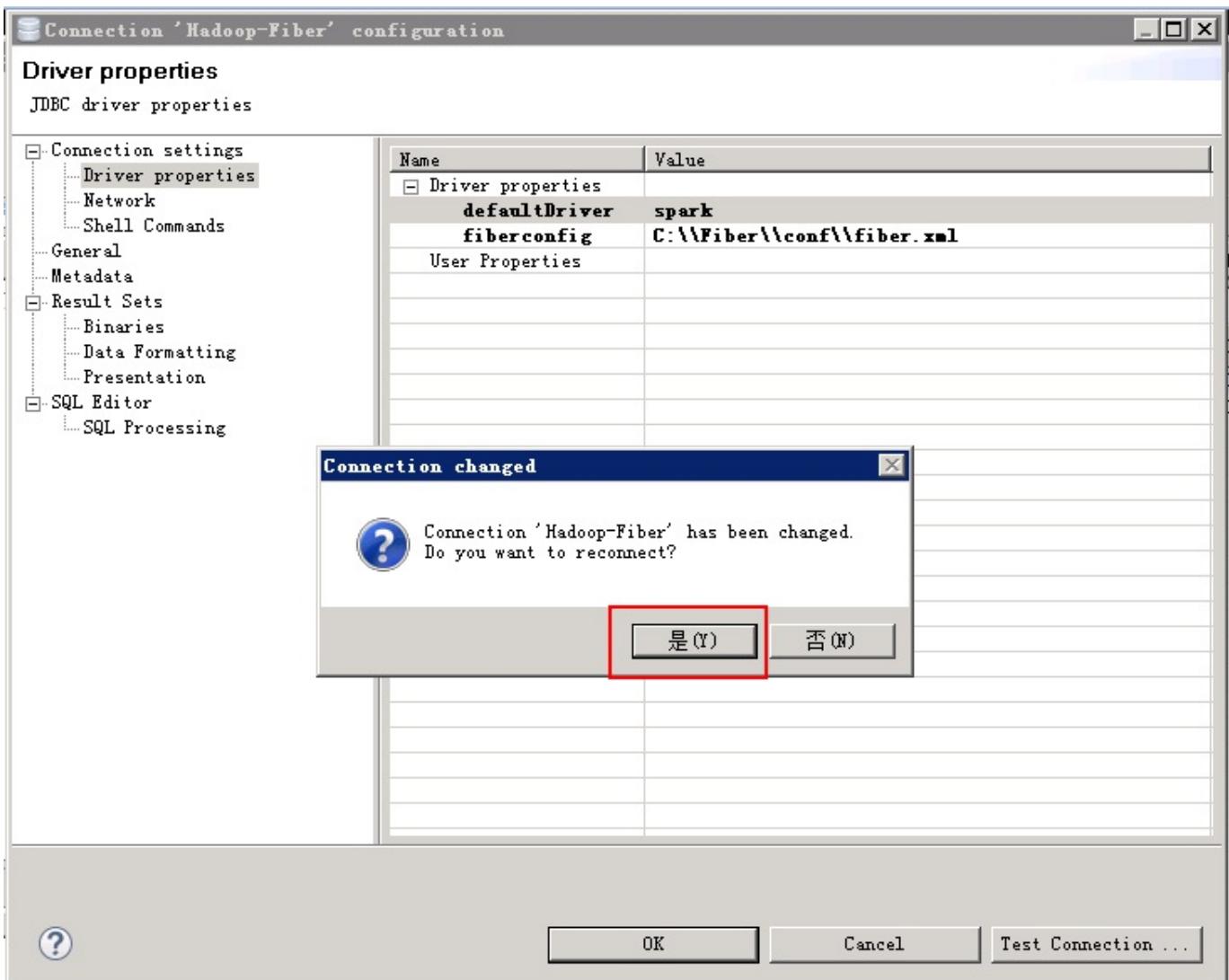
	workers_info.id	workers_info.name	workers_info.usd_flag	workers_info.salary	workers_info.addr
1	1	Wang	R	8,000.01	Chian:Shenzhen
2	2	Chen	F	2,334.01	Chian:Shanghai

At the bottom, a message indicates '2 row(s) fetched - 245ms (+76ms)'.

- 测试spark连接, 把driver切换为spark, 连接右键选择 **Edit Connection**

The screenshot shows the DBBeaver interface with the 'Hadoop-Fiber' connection selected in the Database Navigator. A context menu is open over the connection entry, and the 'Edit Connection' option is highlighted with a red box. Other options in the menu include 'Connect', 'Invalidate/Reconnect', 'Disconnect', 'SQL Editor', 'Recent SQL Editor', 'Create New Connection', 'New Folder', 'Copy', 'Paste', 'Delete', 'Rename', 'Properties', and 'Refresh'. The bottom status bar shows 'CST | en\_US'.





使用spark driver查看表中数据

DBBeaver - General - [ workers\_info ]

File Edit Navigate Search SQL Editor Database Window Help

Commit Rollback Auto Hadoop-Fiber None 200

Database Navigator Projects

Type table/view name to filter

Hadoop-Fiber

- Tables
  - employees\_info
  - fiber\_test
  - pokes
  - test
  - workers\_info
- Views
- Indexes
- Procedures

workers\_info

Properties Data Diagram

workers\_info Enter a SQL expression to filter results (use Ctrl+Space)

	<b>id</b>	<b>name</b>	<b>usd_flag</b>	<b>salary</b>	<b>address</b>	<b>entrytime</b>
1	1	Wang	R	8,000.01	Chian:Shenzhen	2014
2	2	Chen	F	2,334.01	Chian:Shanghai	2015

Save Cancel Script + < > < > Record Panels Grid Text

2 row(s) fetched - 810ms (+41ms)

- 测试phoenix连接，把driver切换为phoenix，连接右键选择 Edit Connection

Connection 'Hadoop-Fiber' configuration

Driver properties

JDBC driver properties

Connection settings

- Driver properties (highlighted)
- Network
- Shell Commands
- General
- Metadata

Result Sets

- Binaries
- Data Formatting

SQL Editor

Success

Server: Phoenix 4.4  
Driver: PhoenixEmbeddedDriver 4.4  
Connected (7152 ms)

OK

?

OK Cancel Test Connection ...

查看phoenix表中数据

DBeaver - General - [ TB\_PHOENIX ]

File Edit Navigate Search SQL Editor Database Window Help

Auto Hadoop-Fiber None 200

Database Navigator Projects

General

Connections Scripts Script-1.sql Script-2.sql Script.sql

TB\_PHOENIX <Hadoop-Fiber> Script-2

Properties Data Diagram

TB\_PHOENIX | Enter a SQL expression to filter results (use Ctrl+Space)

ID	NAME	COMPANY
101	phoenix_user1	company1
102	phoenix_user2	company2
103	phoenix_user3	company3

Save Cancel Script + (b) N M Record Panels Grid Text

3 row(s) fetched - 72ms (7ms)

## DBeaver对接Fiber功能验证

### Hive增加查看数据

- 将JDBC的defaultDrive切换至Hive
- Hive查询数据：菜单栏选择 **SQL Editor -> New SQL Editor**，编辑脚本，点击左上角执行按钮。

DBeaver - General - [ Hadoop-Fiber > Script ]

File Edit Navigate Search SQL Editor Database Window Help

Auto Hadoop-Fiber None 200

Database Navigator Projects

General

Connections Scripts Script-2.sql Script.sql

workers\_info <Hadoop-Fiber> Script

SELECT \* FROM workers\_info

Result

workers_info.id	workers_info.name	workers_info.usd_flag	workers_info.salary	workers_info.addr
1	Wang	R	8,000.01	Chian:Shenzhen
2	Chen	F	2,334.01	Chian:Shanghai

Save Cancel Script + (b) N M Record Panels Grid Text

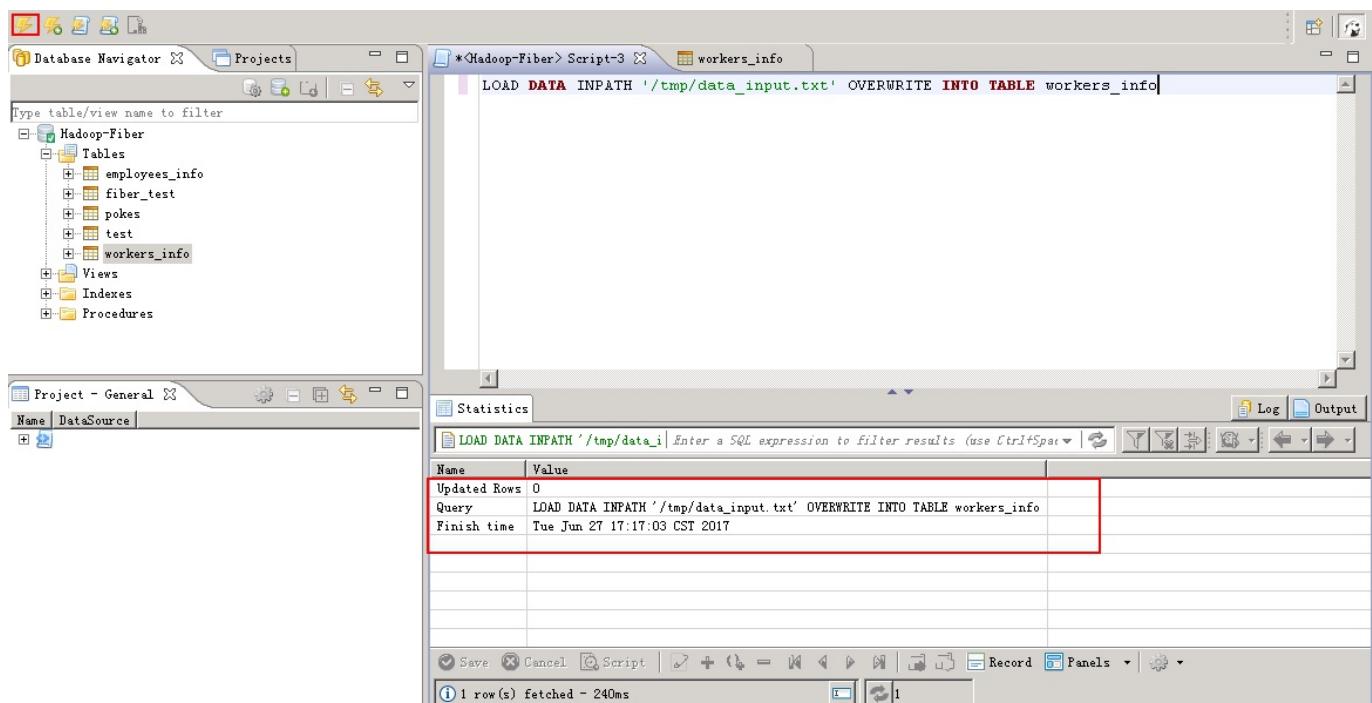
2 row(s) fetched - 167ms (+49ms)

- Hive增加数据：

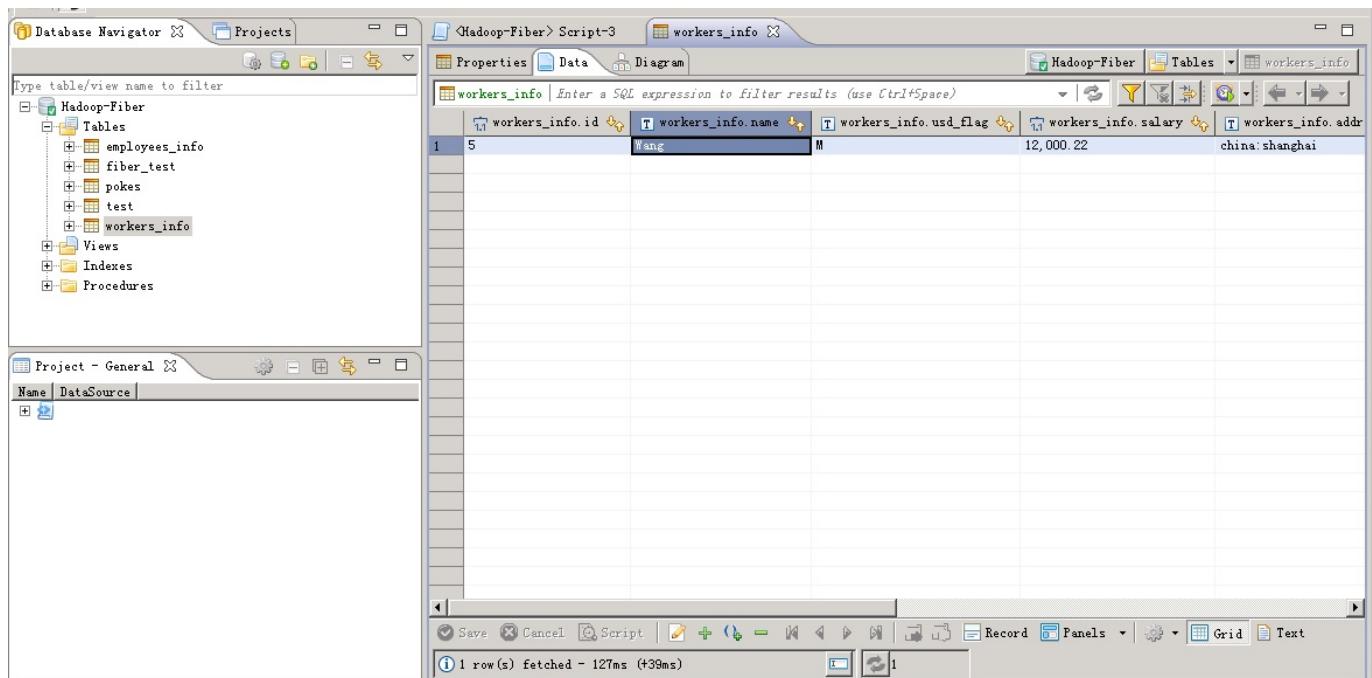
编辑数据文件 `data_input.txt`，上传至集群的hdfs目录中，例如 `/tmp/` 下，文本内容如下：

```
[root@localhost ~]# hdfs dfs -cat /tmp/data_input.txt  
5,Wang,M,12000.22,china:shanghai,[root@localhost ~]#
```

编辑脚本，点击左上角执行按钮。



查看更新后数据：



## Spark增加查看数据

- 将JDBC 的defaultDriver切换至Spark
- Spark查询数据：编辑脚本，点击左上角执行按钮。

```
SELECT * FROM workers_info
```

The screenshot shows the DBBeaver interface. On the left, the 'Database Navigator' pane displays the 'Hadoop-Fiber' database structure, including tables such as 'employees\_info', 'fiber\_test', 'pokes', 'test', and 'workers\_info'. The 'workers\_info' table is selected. The main workspace contains a SQL editor window with the query 'SELECT \* FROM workers\_info' and a 'Result' window showing the execution of this query. The resulting table has columns: id, name, usd\_flag, salary, address, and entrytime. The data is as follows:

	id	name	usd_flag	salary	address	entrytime
1	1	Wang	R	8,000.01	Chian:Shenzhen	2014
2	2	Chen	F	2,334.01	Chian:Shanghai	2015
3	3	zhang	F	8,000.22	china:shanghai	2017
4	4	Li	F	10,000.22	china:shanghai	2017

Below the result grid, it says '4 row(s) fetched - 8.337s (+3ms)'.

- Spark增加数据：

编辑数据文件data\_input.txt，上传至Spark的JDBCServer(主)实例所在的节点的/opt/目录下

The screenshot shows the Spark instance management interface. It lists two JDBCServer instances: 'JDBCServer(备)' and 'JDBCServer(主)'. The 'JDBCServer(主)' instance is highlighted with a red border. The interface includes tabs for 'Service Status', 'Instance', 'Service Configuration', and 'Resource Contribution Ranking'. Below the table, there is a 'More Operations' dropdown menu.

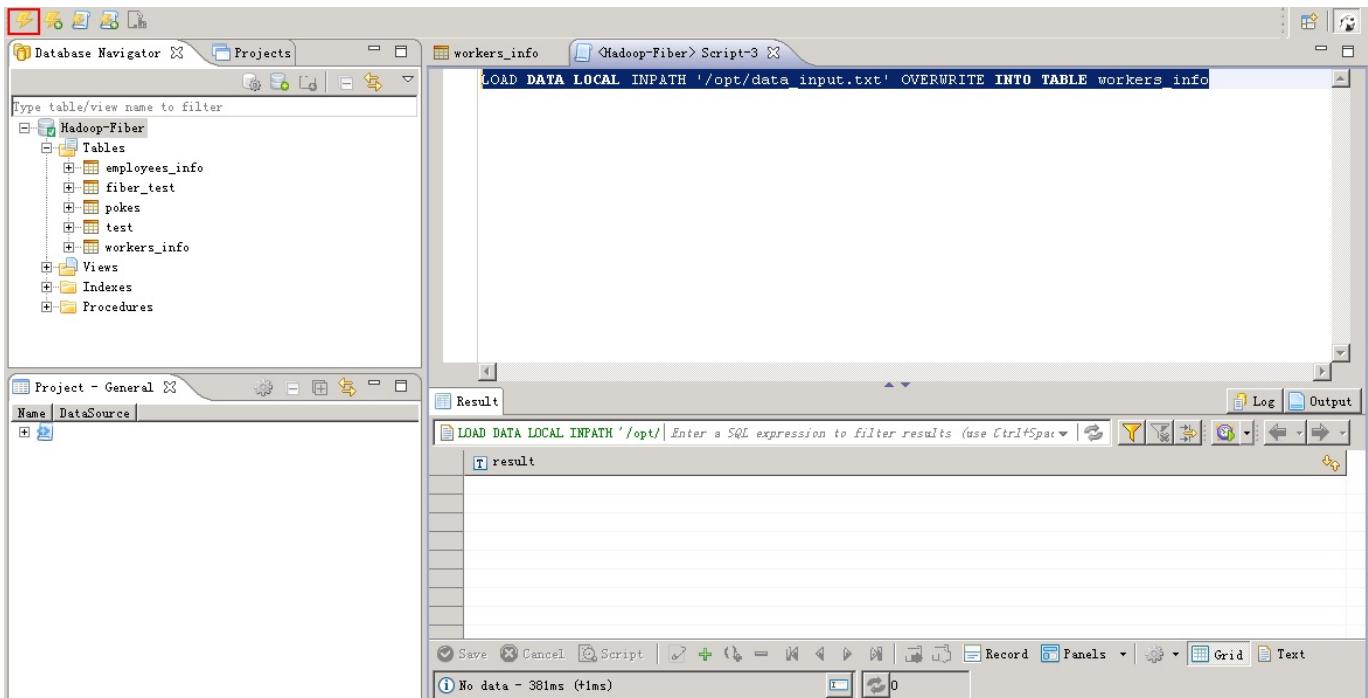
角色	主机名	管理IP	业务IP
JDBCServer(备)	162-1-93-102	162.1.93.102	162.1.93.102
<b>JDBCServer(主)</b>	162-1-93-103	162.1.93.103	162.1.93.103

文本内容如下：

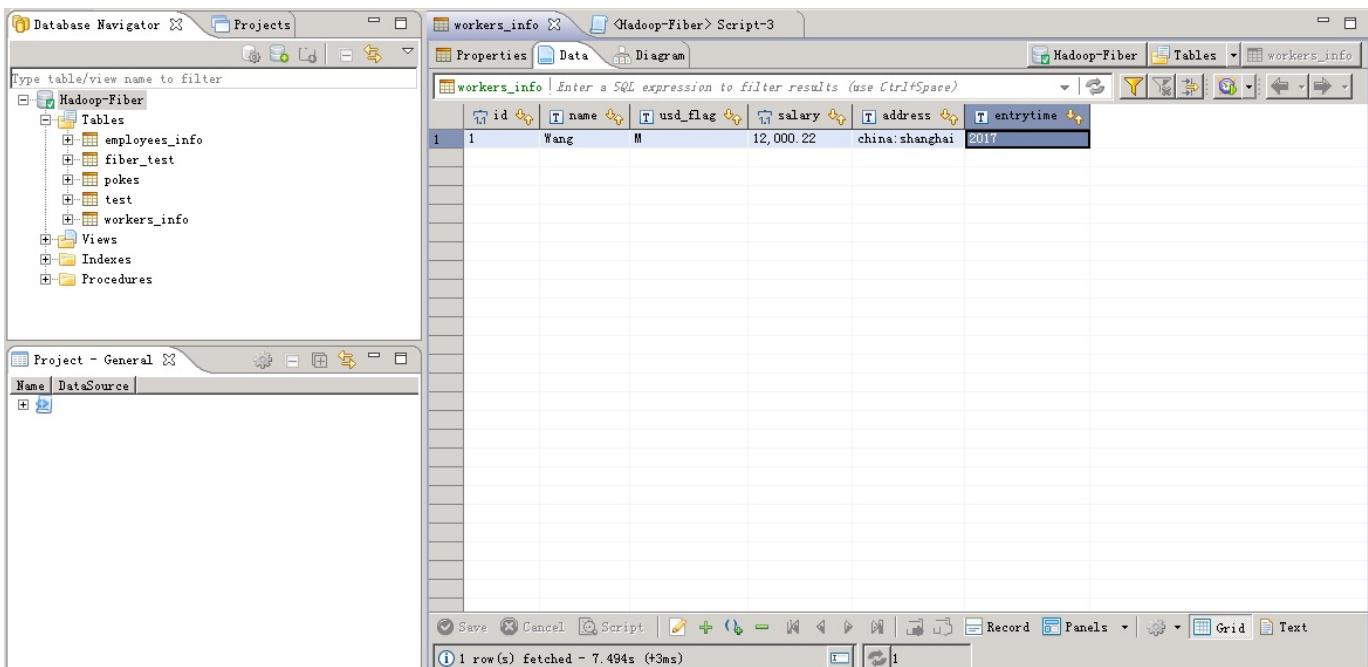
```
[root@162-1-93-103 opt]# vi data_input.txt
1,Wang,M,12000.22,china:shanghai,2017
~
```

编辑脚本，点击左上角执行按钮。

```
LOAD DATA LOCAL INPATH '/opt/data_input.txt' OVERWRITE INTO TABLE workers_info
```



查看结果：



## Phoenix增删改查数据

- 将JDBC的defaultDrive切换至Phoenix
- Phoenix增加数据

菜单栏选择 **SQL Editor -> New SQL Editor**, 编辑脚本, 点击左上角 **执行** 按钮。

```
UPSERT INTO TB_PHOENIX(Id, Name, Company) values (104, 'phoenix_user4', 'company4')
```

DBeaver - General - [ Hadoop-Fiber > Script-2 ]

File Edit Navigate Search SQL Editor Database Window Help

Auto | Hadoop-Fiber | None | 200 |

Database Navigator Projects

General

- Connections
- Scripts
  - Script-1.sql
  - Script-2.sql
  - Script.sql

TB\_PHOENIX <Hadoop-Fiber> Script-2

```
UPsert into tb_phoenix(Id, Name, Company) values (104, 'phoenix_user4', 'company4')
```

Statistics

Name	Value
Updated Rows	1
Query	UPsert into tb_phoenix(Id, Name, Company) values (104, 'phoenix_user4', 'company4')
Finish time	Tue Jun 27 14:57:53 CST 2017

Project - General

Name | DataSource |

Save | Cancel | Script | + | Record | Panels |

1 row(s) fetched - 36ms

查看增加的数据:

DBeaver - General - [ TB\_PHOENIX ]

File Edit Navigate Search SQL Editor Database Window Help

Auto | Hadoop-Fiber | None |

Database Navigator Projects

General

- Connections
- Scripts
  - Script-1.sql
  - Script-2.sql
  - Script.sql

TB\_PHOENIX <Hadoop-Fiber> Script-2

Properties Data Diagram

TB\_PHOENIX | Enter a SQL expression to filter results (use Ctrl+Space)

ID	T_NAME	T_COMPANY
1	phoenix_user1	company1
2	phoenix_user2	company2
3	phoenix_user3	company3
4	phoenix_user4	company4

Project - General

Name | DataSource |

- Phoenix删除数据

页面上删除: 选择待删除的列, 然后点击下方 **删除** 按钮, 然后点击 **save** 按钮:

DBBeaver - General - [ TB\_PHOENIX ]

File Edit Navigate Search SQL Editor Database Window Help

Auto Hadoop-Fiber None 200

Database Navigator Projects

TB\_PHOENIX \*Hadoop-Fiber> Script-2

Properties Data Diagram

TB\_PHOENIX Enter a SQL expression to filter results (use Ctrl+Space)

ID	NAME	COMPANY
1	phoenix_user1	company1
2	phoenix_user2	company2
3	phoenix_user3	company3
4	phoenix_user4	company4

Save Cancel Script + = Record Panels Grid Text

4 row(s) fetched - 100ms (+6ms)

CST en

脚本删除：编辑脚本，点击左上方 执行 按钮

```
delete from TB_PHOENIX where ID=104;
```

DBBeaver - General - [ Hadoop-Fiber > Script-2 ]

File Edit Navigate Search SQL Editor Database Window Help

Auto Hadoop-Fiber None 200

Database Navigator Projects

TB\_PHOENIX \*Hadoop-Fiber> Script-2

```
delete from TB_PHOENIX where ID=104;
```

Statistics

Name	Value
Updated Rows	1
Query	delete from TB_PHOENIX where ID=104
Finish time	Tue Jun 27 15:05:34 CST 2017

Log Output

Save Cancel Script + = Record Panels Grid Text

查看输出后的数据

The screenshot shows the DBBeaver interface. In the top right, there's a table titled 'TB\_PHOENIX' with columns 'ID', 'NAME', and 'COMPANY'. The data is as follows:

ID	NAME	COMPANY
1	phoenix_user1	company1
2	phoenix_user2	company2
3	phoenix_user3	company3

In the bottom left, there's a 'Project - General' window showing a single 'DataSource' entry.

- Phoenix更新数据, 编辑更新脚本, 点击左上方 执行 按钮

```
UPSERT INTO TB_PHOENIX(Id, Name, Company) values (103, 'phoenix_user3_up', 'company3_up')
```

The screenshot shows the DBBeaver interface after executing the UPSERT query. The 'Statistics' panel at the bottom displays the following information:

Name	Value
Updated Rows	1
Query	UPSERT INTO TB_PHOENIX(Id, Name, Company) values (103, 'phoenix_user3_up', 'company3_up')
Finish time	Tue Jun 27 15:20:29 CST 2017

A red box highlights the 'Updated Rows' and 'Query' rows in the statistics table.

查看更新后的数据:

The screenshot shows the DBBeaver interface. In the Database Navigator, under the 'General' section, there is a 'Scripts' folder containing three files: Script-1.sql, Script-2.sql, and Script.sql. In the main workspace, a table named 'TB\_PHOENIX' is displayed with the following data:

ID	NAME	COMPANY
1	phoenix_user1	company1
2	phoenix_user2	company2
3	phoenix_user3_up	company3_up

- 查看数据：编辑查询脚本，点击左上方 执行 按钮。

The screenshot shows the DBBeaver interface with a query editor window open. The SQL command is:

```
SELECT * FROM TB_PHOENIX
```

The results of the query are displayed in a table below:

ID	NAME	COMPANY
1	phoenix_user1	company1
2	phoenix_user2	company2
3	phoenix_user3_up	company3_up