|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **BODS** | **文档编号** |  | **版本** | A1 | **密级** | 商密A |
| **项目名称** | **ROSE系统** | | | | |
| **项目来源** |  | | | | |

QR-RD-022(Ver1.2)

**ROSE系统**

**详细设计说明书**

(内部资料 请勿外传)

|  |  |  |  |
| --- | --- | --- | --- |
| **编 写：** | 黄健 | **日 期：** | 2015-12-03 |
| **检 查：** |  | **日 期：** |  |
| **审 核：** |  | **日 期：** |  |
| **批 准：** |  | **日 期：** |  |

**百分点科技有限公司**

**版权所有 不得复制**

**文档变更记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **变更（+/-）说明** | **作者** | **版本号** | **日期** | **批准** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目 录

[1. 引言 4](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115737#_Toc266115737)

[1.1 编写目的和范围 4](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115738#_Toc266115738)

[1.2 术语表 4](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115739#_Toc266115739)

[1.3 参考资料 4](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115740#_Toc266115740)

[1.4 使用的文字处理和绘图工具 4](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115741#_Toc266115741)

[2. 全局数据结构说明 5](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115742#_Toc266115742)

[2.1 常量 5](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115743#_Toc266115743)

[2.2 变量 5](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115744#_Toc266115744)

[2.3 数据结构 5](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115745#_Toc266115745)

[3. 模块设计 5](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115746#_Toc266115746)

[3.1 用例图 5](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115747#_Toc266115747)

[3.2 功能设计说明 5](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115748#_Toc266115748)

[3.2.1 模块1 5](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115749#_Toc266115749)

[3.2.2 模块2 6](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115750#_Toc266115750)

[4. 接口设计 7](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115751#_Toc266115751)

[4.1 内部接口 7](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115752#_Toc266115752)

[4.2 外部接口 7](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115753#_Toc266115753)

[4.2.1 接口说明 7](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115754#_Toc266115754)

[4.2.2 调用方式 8](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115755#_Toc266115755)

[5. 数据库设计 8](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115756#_Toc266115756)

[6. 系统安全保密设计 8](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115757#_Toc266115757)

[6.1 说明 8](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115758#_Toc266115758)

[6.2 设计 8](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115759#_Toc266115759)

[6.2.1 数据传输部分 8](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115760#_Toc266115760)

[6.2.2 IP过滤分部 8](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115761#_Toc266115761)

[6.2.3 身份验证部分 8](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115762#_Toc266115762)

[7. 系统性能设计 9](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115763#_Toc266115763)

[8. 系统出错处理 9](file:///C:\Documents%20and%20Settings\Administrator\桌面\详细设计.doc#_Toc266115764#_Toc266115764)

# 引言

## 编写目的和范围

软件开发人员：了解ROSE系统架构、基本功能和BODS对其进行的二次开发。

软件测试人员：了解ROSE系统基本业务功能点。

## 术语表

定义系统或产品中涉及的重要术语，为读者在阅读文档时提供必要的参考信息。

|  |  |  |
| --- | --- | --- |
| **序号** | **术语或缩略语** | **说明性定义** |
| 1 | ROSE | Apache孵化器项目Zeppelin的定制开发版 |
| 2 | 解释器（Interpreter） | ROSE系统中提供的各种服务 |
|  | 面板（NoteBook） | ROSE前端展示页面 |

## 参考资料

列出有关资料的名称、作者、文件编号或版本等。参考资料包括：

a．需求说明书、架构设计说明书等；

b．本项目的其他已发表的文件；

c．引用文件、资料、软件开发标准等。

|  |  |  |  |
| --- | --- | --- | --- |
| 资料名称 | 作者 | 文件编号、版本 | 资料存放地点 |
|  |  |  |  |

## 使用的文字处理和绘图工具

# 全局数据结构说明

本章说明本程序系统中使用的全局数据常量、变量和数据结构。

## 常量

包括数据文件名称及其所在目录，功能说明，具体常量说明等。

## 变量

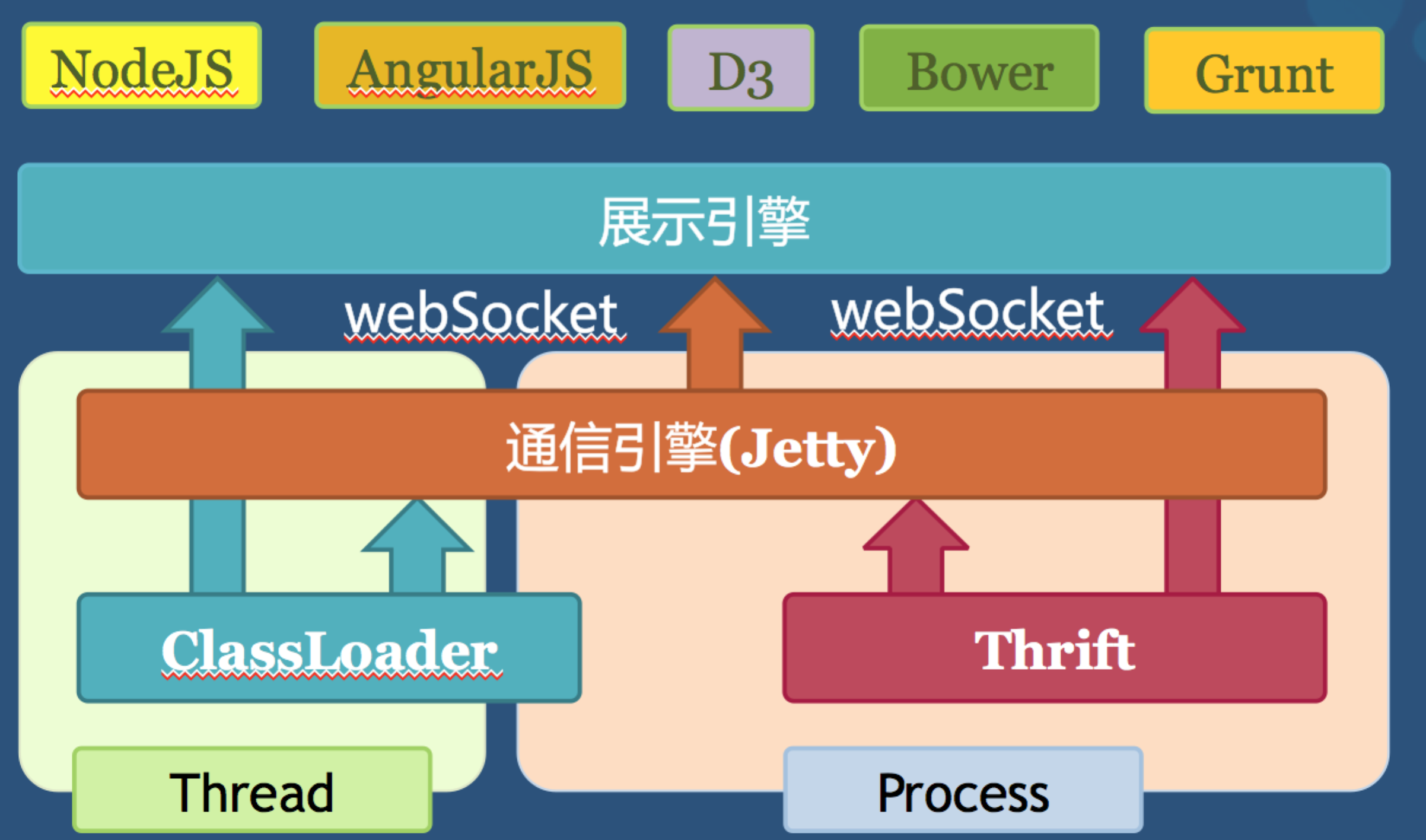
本章说明本程序系统中使用的全局数据常量、变量和数据结构。

## 数据结构

包括数据结构名称，功能说明，具体数据结构说明（定义、注释、取值）等。

# 模块设计

## 总体架构图

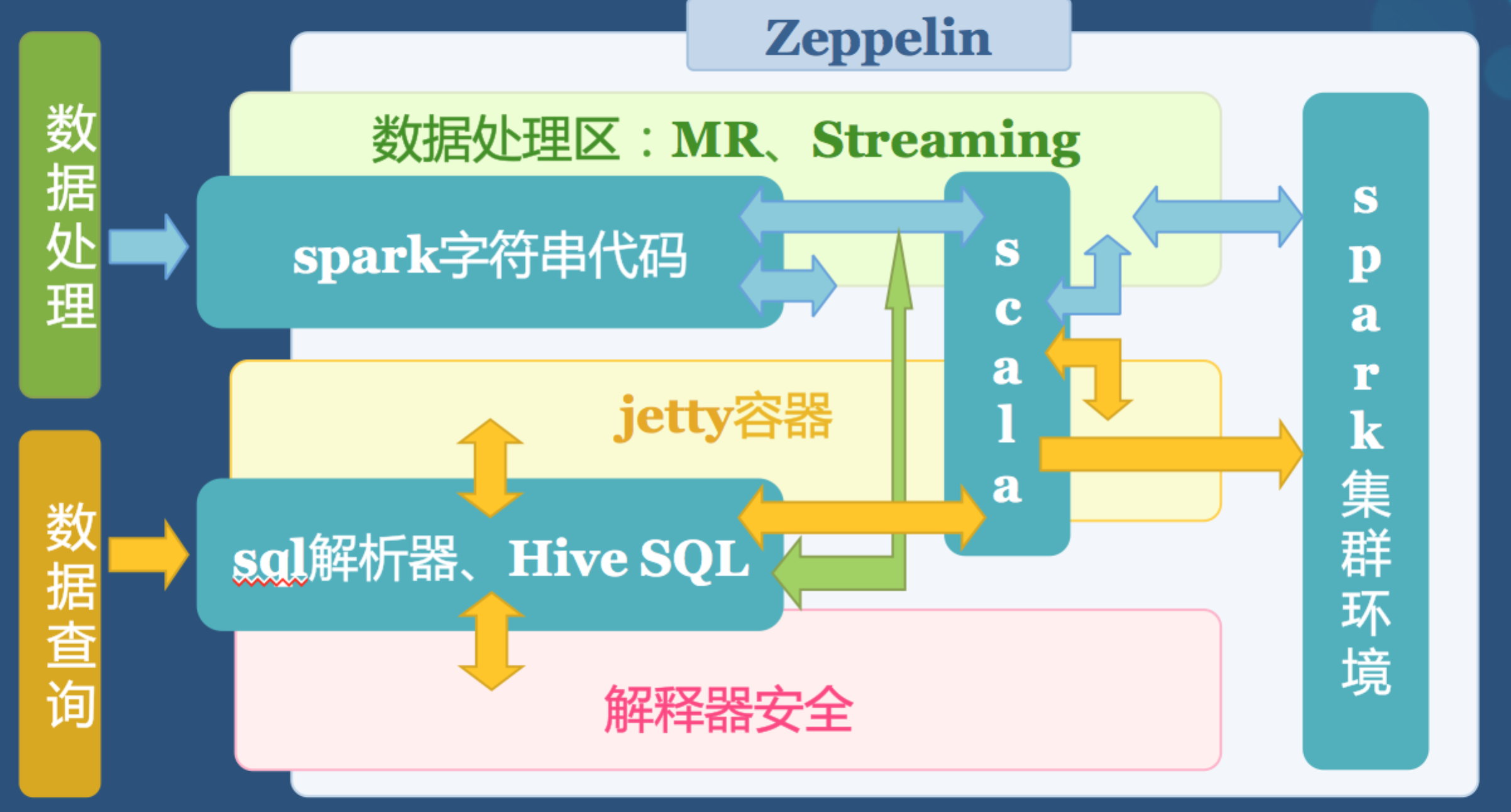


## 功能设计说明

### 支持mysql、impala、spark、drill等解释器

#### 支持mysql

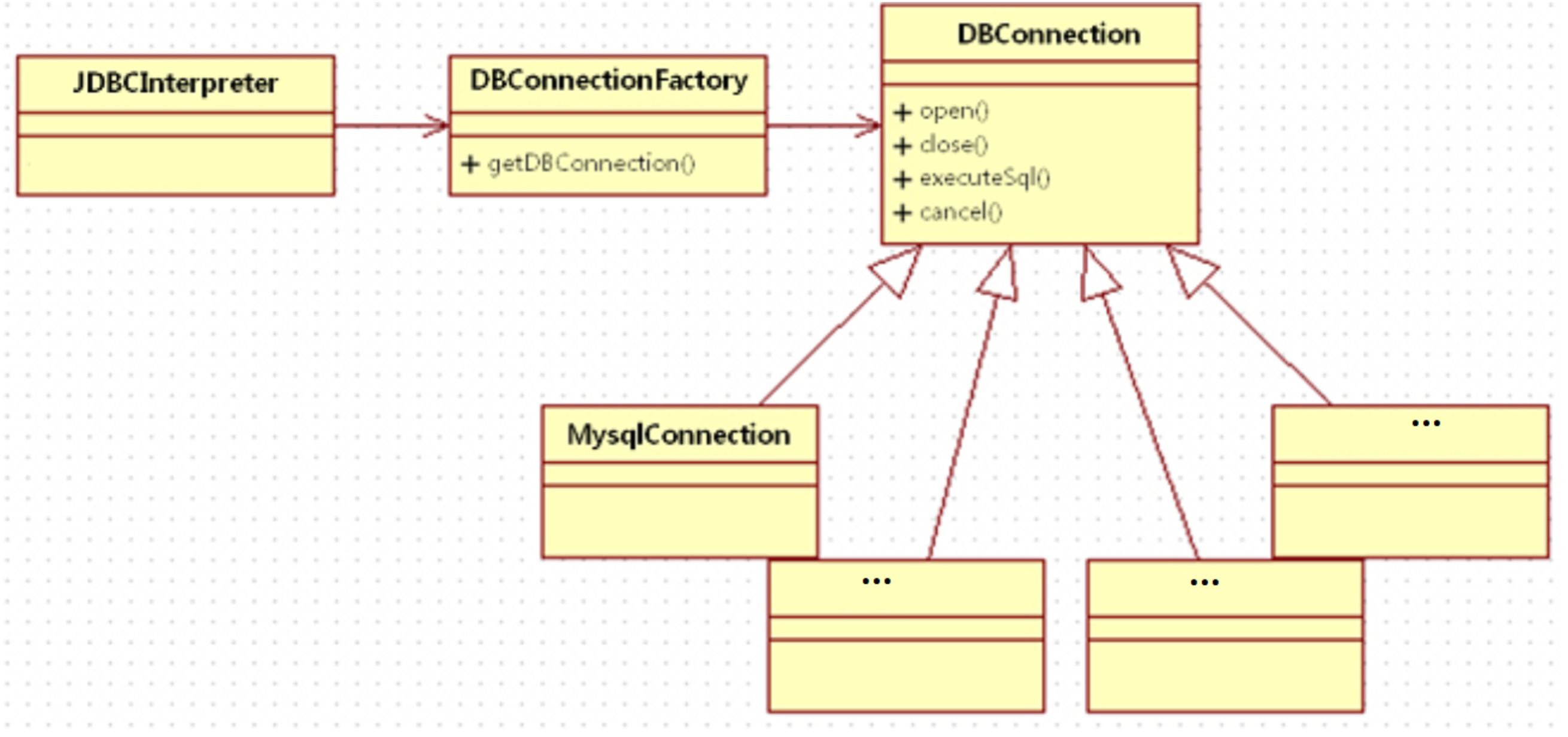
##### 流程图



##### 功能描述

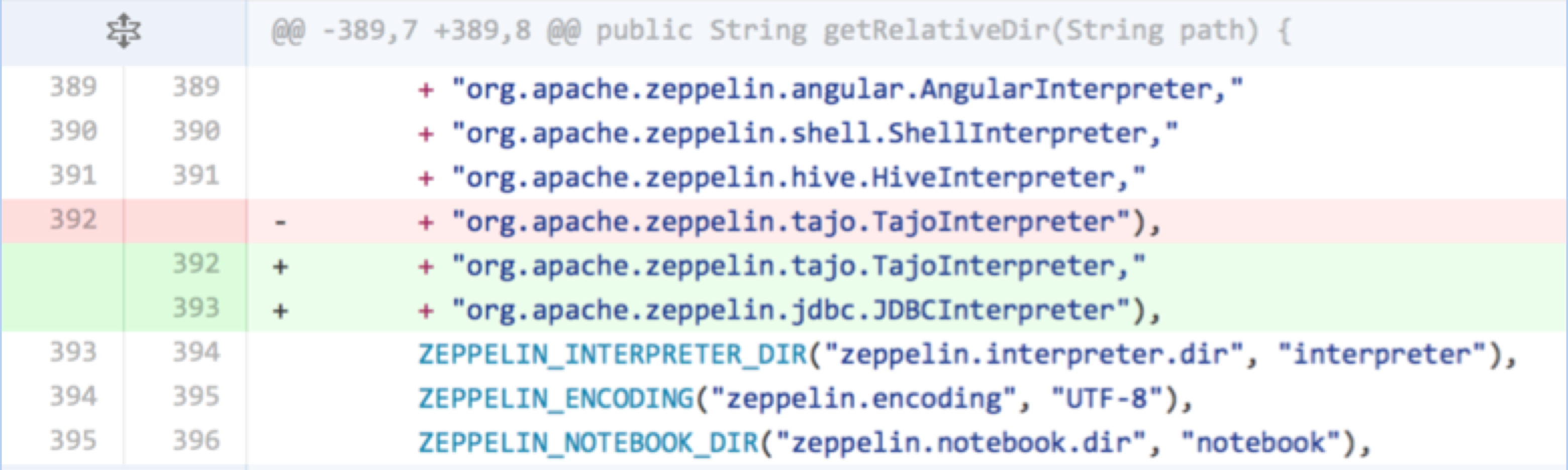
可以使用ROSE的mysql解释器进行mysql数据库的连接，同时进行表的创建及其ACID等操作，并且可对查询出来的数据进行图形化显示及其加工处理，如：生成表格、饼图、柱状图、折线图，对结果数据进行聚合、分组操作等。

##### 类图



##### 修改记录

* pom.xml
* zeppelin-zengine/src/main/java/org/apache/zeppelin/conf/ZeppelinConfiguration.java:



* jdbc/pom.xml:

+<?xml version="1.0" encoding="UTF-8"?>

+<!--

+ ~ Licensed to the Apache Software Foundation (ASF) under one or more

+ ~ contributor license agreements. See the NOTICE file distributed with

+ ~ this work for additional information regarding copyright ownership.

+ ~ The ASF licenses this file to You under the Apache License, Version 2.0

+ ~ (the "License"); you may not use this file except in compliance with

+ ~ the License. You may obtain a copy of the License at

+ ~

+ ~ http://www.apache.org/licenses/LICENSE-2.0

+ ~

+ ~ Unless required by applicable law or agreed to in writing, software

+ ~ distributed under the License is distributed on an "AS IS" BASIS,

+ ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

+ ~ See the License for the specific language governing permissions and

+ ~ limitations under the License.

+ -->

+

+<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

+ <modelVersion>4.0.0</modelVersion>

+

+ <parent>

+ <artifactId>zeppelin</artifactId>

+ <groupId>org.apache.zeppelin</groupId>

+ <version>0.5.0-SNAPSHOT</version>

+ </parent>

+

+ <groupId>org.apache.zeppelin</groupId>

+ <artifactId>zeppelin-jdbc</artifactId>

+ <packaging>jar</packaging>

+ <version>0.5.0-SNAPSHOT</version>

+ <name>Zeppelin: JDBC interpreter</name>

+ <url>http://zeppelin.incubator.apache.org</url>

+

+ <dependencies>

+ <dependency>

+ <groupId>${project.groupId}</groupId>

+ <artifactId>zeppelin-interpreter</artifactId>

+ <version>${project.version}</version>

+ <scope>provided</scope>

+ </dependency>

+

+ <dependency>

+ <groupId>org.apache.commons</groupId>

+ <artifactId>commons-exec</artifactId>

+ <version>1.1</version>

+ </dependency>

+

+ <dependency>

+ <groupId>org.slf4j</groupId>

+ <artifactId>slf4j-api</artifactId>

+ </dependency>

+

+ <dependency>

+ <groupId>org.slf4j</groupId>

+ <artifactId>slf4j-log4j12</artifactId>

+ </dependency>

+

+ <dependency>

+ <groupId>junit</groupId>

+ <artifactId>junit</artifactId>

+ <scope>test</scope>

+ </dependency>

+

+ <dependency>

+ <groupId>mysql</groupId>

+ <artifactId>mysql-connector-java</artifactId>

+ <version>5.1.6</version>

+ </dependency>

+ </dependencies>

+

+ <build>

+ <plugins>

+ <plugin>

+ <groupId>org.apache.maven.plugins</groupId>

+ <artifactId>maven-deploy-plugin</artifactId>

+ <version>2.7</version>

+ <configuration>

+ <skip>true</skip>

+ </configuration>

+ </plugin>

+

+ <plugin>

+ <artifactId>maven-enforcer-plugin</artifactId>

+ <version>1.3.1</version>

+ <executions>

+ <execution>

+ <id>enforce</id>

+ <phase>none</phase>

+ </execution>

+ </executions>

+ </plugin>

+

+ <plugin>

+ <artifactId>maven-dependency-plugin</artifactId>

+ <version>2.8</version>

+ <executions>

+ <execution>

+ <id>copy-dependencies</id>

+ <phase>package</phase>

+ <goals>

+ <goal>copy-dependencies</goal>

+ </goals>

+ <configuration>

+ <outputDirectory>${project.build.directory}/../../interpreter/jdbc</outputDirectory>

+ <overWriteReleases>false</overWriteReleases>

+ <overWriteSnapshots>false</overWriteSnapshots>

+ <overWriteIfNewer>true</overWriteIfNewer>

+ <includeScope>runtime</includeScope>

+ </configuration>

+ </execution>

+ <execution>

+ <id>copy-artifact</id>

+ <phase>package</phase>

+ <goals>

+ <goal>copy</goal>

+ </goals>

+ <configuration>

+ <outputDirectory>${project.build.directory}/../../interpreter/jdbc</outputDirectory>

+ <overWriteReleases>false</overWriteReleases>

+ <overWriteSnapshots>false</overWriteSnapshots>

+ <overWriteIfNewer>true</overWriteIfNewer>

+ <includeScope>runtime</includeScope>

+ <artifactItems>

+ <artifactItem>

+ <groupId>${project.groupId}</groupId>

+ <artifactId>${project.artifactId}</artifactId>

+ <version>${project.version}</version>

+ <type>${project.packaging}</type>

+ </artifactItem>

+ </artifactItems>

+ </configuration>

+ </execution>

+ </executions>

+ </plugin>

+ </plugins>

+ </build>

+

+</project>

* jdbc/src/main/java/org/apache/zeppelin/jdbc/DBConnection.java:

+/\*

+ \* Licensed to the Apache Software Foundation (ASF) under one or more

+ \* contributor license agreements. See the NOTICE file distributed with

+ \* this work for additional information regarding copyright ownership.

+ \* The ASF licenses this file to You under the Apache License, Version 2.0

+ \* (the "License"); you may not use this file except in compliance with

+ \* the License. You may obtain a copy of the License at

+ \*

+ \* http://www.apache.org/licenses/LICENSE-2.0

+ \*

+ \* Unless required by applicable law or agreed to in writing, software

+ \* distributed under the License is distributed on an "AS IS" BASIS,

+ \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

+ \* See the License for the specific language governing permissions and

+ \* limitations under the License.

+ \*

+ \*/

+

+package org.apache.zeppelin.jdbc;

+

+import org.apache.zeppelin.interpreter.InterpreterResult;

+

+/\*\*

+ \* DBConnection

+ \*

+ \* @author Jian.huang

+ \*

+ \*/

+

+public interface DBConnection {

+ void open();

+ void close();

+ InterpreterResult executeSql(String sql);

+ void cancel();

+}

+

* jdbc/src/main/java/org/apache/zeppelin/jdbc/DBConnectionFactory.java:

+/\*

+ \* Licensed to the Apache Software Foundation (ASF) under one or more

+ \* contributor license agreements. See the NOTICE file distributed with

+ \* this work for additional information regarding copyright ownership.

+ \* The ASF licenses this file to You under the Apache License, Version 2.0

+ \* (the "License"); you may not use this file except in compliance with

+ \* the License. You may obtain a copy of the License at

+ \*

+ \* http://www.apache.org/licenses/LICENSE-2.0

+ \*

+ \* Unless required by applicable law or agreed to in writing, software

+ \* distributed under the License is distributed on an "AS IS" BASIS,

+ \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

+ \* See the License for the specific language governing permissions and

+ \* limitations under the License.

+ \*

+ \*/

+

+package org.apache.zeppelin.jdbc;

+

+import java.util.Properties;

+import org.slf4j.Logger;

+import org.slf4j.LoggerFactory;

+import java.sql.Connection;

+

+/\*\*

+ \* DBConnectionFactory

+ \*

+ \* @author Jian.huang

+ \*

+ \*/

+

+public class DBConnectionFactory {

+ Logger logger = LoggerFactory.getLogger(DBConnectionFactory.class);

+

+ String jdbc = "";

+ String host = "";

+ String port = "";

+ String user = "";

+ String password = "";

+

+ public DBConnectionFactory(Properties currentProperty) {

+ for (Object k : currentProperty.keySet()) {

+ String key = (String) k;

+ String value = (String) currentProperty.get(key);

+

+ if ( key.equals("jdbc") ) {

+ this.jdbc = value.toLowerCase();

+ } else if ( key.equals("host") ) {

+ this.host = value;

+ } else if ( key.equals("password") ) {

+ this.password = value;

+ } else if ( key.equals("user") ) {

+ this.user = value;

+ } else if ( key.equals("port") ) {

+ this.port = value;

+ } else {

+ logger.info("else key : " + key);

+ }

+ }

+ }

+

+ public DBConnection getDBConnection() {

+ DBConnection currentConnection;

+

+ // add other jdbc

+ if ( jdbc.equals("mysql") ) {

+ currentConnection = new MysqlConnection(host, port, user, password);

+ } else {

+ currentConnection = null;

+ }

+

+ return currentConnection;

+ }

+}

+

* jdbc/src/main/java/org/apache/zeppelin/jdbc/JDBCInterpreter.java:

+/\*

+ \* Licensed to the Apache Software Foundation (ASF) under one or more

+ \* contributor license agreements. See the NOTICE file distributed with

+ \* this work for additional information regarding copyright ownership.

+ \* The ASF licenses this file to You under the Apache License, Version 2.0

+ \* (the "License"); you may not use this file except in compliance with

+ \* the License. You may obtain a copy of the License at

+ \*

+ \* http://www.apache.org/licenses/LICENSE-2.0

+ \*

+ \* Unless required by applicable law or agreed to in writing, software

+ \* distributed under the License is distributed on an "AS IS" BASIS,

+ \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

+ \* See the License for the specific language governing permissions and

+ \* limitations under the License.

+ \*

+ \*/

+

+package org.apache.zeppelin.jdbc;

+

+import java.io.ByteArrayOutputStream;

+import java.io.IOException;

+import java.util.List;

+import java.util.Properties;

+import java.util.Vector;

+

+import org.apache.commons.lang.StringUtils;

+import org.apache.zeppelin.interpreter.Interpreter;

+import org.apache.zeppelin.interpreter.InterpreterContext;

+import org.apache.zeppelin.interpreter.InterpreterResult;

+import org.apache.zeppelin.interpreter.InterpreterResult.Code;

+import org.apache.zeppelin.scheduler.Scheduler;

+import org.apache.zeppelin.scheduler.SchedulerFactory;

+import org.slf4j.Logger;

+import org.slf4j.LoggerFactory;

+

+import java.sql.Connection;

+import java.sql.DriverManager;

+import java.sql.ResultSet;

+import java.sql.ResultSetMetaData;

+import java.sql.SQLException;

+import java.sql.Statement;

+

+/\*\*

+ \* JDBC interpreter for Zeppelin.

+ \*

+ \* @author Jian.huang

+ \*

+ \*/

+

+public class JDBCInterpreter extends Interpreter {

+ Logger logger = LoggerFactory.getLogger(JDBCInterpreter.class);

+ int commandTimeOut = 600000;

+

+ static {

+ Interpreter.register("jdbc", JDBCInterpreter.class.getName());

+ }

+

+ DBConnection jdbcConnection;

+

+ public JDBCInterpreter(Properties property) {

+ super(property);

+ Properties currentProperty = getProperty();

+ DBConnectionFactory DBFactory = new DBConnectionFactory(currentProperty);

+ this.jdbcConnection = DBFactory.getDBConnection();

+ }

+

+ @Override

+ public void open() {

+ jdbcConnection.open();

+ }

+

+ @Override

+ public void close() {

+ jdbcConnection.close();

+ }

+

+ @Override

+ public InterpreterResult interpret(String cmd, InterpreterContext contextInterpreter) {

+ logger.info("Run SQL command '" + cmd + "'");

+ return jdbcConnection.executeSql(cmd);

+ }

+

+ @Override

+ public void cancel(InterpreterContext context) {

+ jdbcConnection.cancel();

+ }

+

+ @Override

+ public FormType getFormType() {

+ return FormType.SIMPLE;

+ }

+

+ @Override

+ public int getProgress(InterpreterContext context) {

+ return 0;

+ }

+

+ @Override

+ public Scheduler getScheduler() {

+ return SchedulerFactory.singleton().createOrGetFIFOScheduler(

+ JDBCInterpreter.class.getName() + this.hashCode());

+ }

+

+ @Override

+ public List<String> completion(String buf, int cursor) {

+ return null;

+ }

+

+}

* jdbc/src/main/java/org/apache/zeppelin/jdbc/MysqlConnection.java:

+/\*

+ \* Licensed to the Apache Software Foundation (ASF) under one or more

+ \* contributor license agreements. See the NOTICE file distributed with

+ \* this work for additional information regarding copyright ownership.

+ \* The ASF licenses this file to You under the Apache License, Version 2.0

+ \* (the "License"); you may not use this file except in compliance with

+ \* the License. You may obtain a copy of the License at

+ \*

+ \* http://www.apache.org/licenses/LICENSE-2.0

+ \*

+ \* Unless required by applicable law or agreed to in writing, software

+ \* distributed under the License is distributed on an "AS IS" BASIS,

+ \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

+ \* See the License for the specific language governing permissions and

+ \* limitations under the License.

+ \*

+ \*/

+

+package org.apache.zeppelin.jdbc;

+

+import org.slf4j.Logger;

+import org.slf4j.LoggerFactory;

+import org.apache.commons.lang.StringUtils;

+

+import org.apache.zeppelin.interpreter.InterpreterResult;

+import org.apache.zeppelin.interpreter.InterpreterResult.Code;

+

+import java.util.Vector;

+import java.sql.Connection;

+import java.sql.DriverManager;

+import java.sql.ResultSet;

+import java.sql.ResultSetMetaData;

+import java.sql.SQLException;

+import java.sql.Statement;

+

+/\*\*

+ \* MysqlConnection

+ \*

+ \* @author Jian.huang

+ \*

+ \*/

+

+public class MysqlConnection implements DBConnection {

+ Logger logger = LoggerFactory.getLogger(MysqlConnection.class);

+

+ String host;

+ String port;

+ String user;

+ String passwd;

+

+ public MysqlConnection(String host, String port, String user, String passwd) {

+ this.host = host;

+ this.port = port;

+ this.user = user;

+ this.passwd = passwd;

+ }

+

+ Connection mysqlConnection;

+ Exception exceptionOnConnect;

+ Statement currentStatement;

+ ResultSet resultSet;

+

+ @Override

+ public void open() {

+ String driver = "com.mysql.jdbc.Driver";

+ String url = "jdbc:mysql://";

+ url += host;

+ url += port != "" ? ":" + port : "";

+ url += "/?user=" + user;

+

+ try {

+ Class.forName(driver);

+ mysqlConnection = DriverManager.getConnection(url, user, passwd);

+ } catch ( ClassNotFoundException | SQLException e ) {

+ logger.error("Can not open connection", e);

+ exceptionOnConnect = e;

+ }

+ }

+

+ @Override

+ public void close() {

+ try {

+ mysqlConnection.close();

+ currentStatement.close();

+ resultSet.close();

+ } catch ( SQLException e ) {

+ logger.error("Can not close connection", e);

+ }

+

+ mysqlConnection = null;

+ exceptionOnConnect = null;

+ currentStatement = null;

+ resultSet = null;

+ }

+

+ @Override

+ public InterpreterResult executeSql(String sql) {

+ try {

+ if (exceptionOnConnect != null) {

+ return new InterpreterResult(Code.ERROR, exceptionOnConnect.getMessage());

+ }

+ currentStatement = mysqlConnection.createStatement();

+ StringBuilder msg = null;

+ if (StringUtils.containsIgnoreCase(sql, "EXPLAIN ")) {

+ //return the explain as text, make this visual explain later

+ msg = new StringBuilder();

+ }

+ else {

+ msg = new StringBuilder("%table ");

+ }

+ resultSet = currentStatement.executeQuery(sql);

+ try {

+ ResultSetMetaData md = resultSet.getMetaData();

+ for (int i = 1; i < md.getColumnCount() + 1; i++) {

+ if (i == 1) {

+ msg.append(md.getColumnName(i));

+ } else {

+ msg.append("\t" + md.getColumnName(i));

+ }

+ }

+ msg.append("\n");

+ while (resultSet.next()) {

+ for (int i = 1; i < md.getColumnCount() + 1; i++) {

+ msg.append(resultSet.getString(i) + "\t");

+ }

+ msg.append("\n");

+ }

+ } catch ( NullPointerException e ) {

+

+ }

+

+ InterpreterResult rett = new InterpreterResult(Code.SUCCESS, msg.toString());

+ return rett;

+ }

+ catch (SQLException ex) {

+ logger.error("Can not run " + sql, ex);

+ return new InterpreterResult(Code.ERROR, ex.getMessage());

+ }

+ }

+

+ @Override

+ public void cancel() {

+ if (currentStatement != null) {

+ try {

+ currentStatement.cancel();

+ }

+ catch (SQLException ex) {

+ }

+ finally {

+ currentStatement = null;

+ }

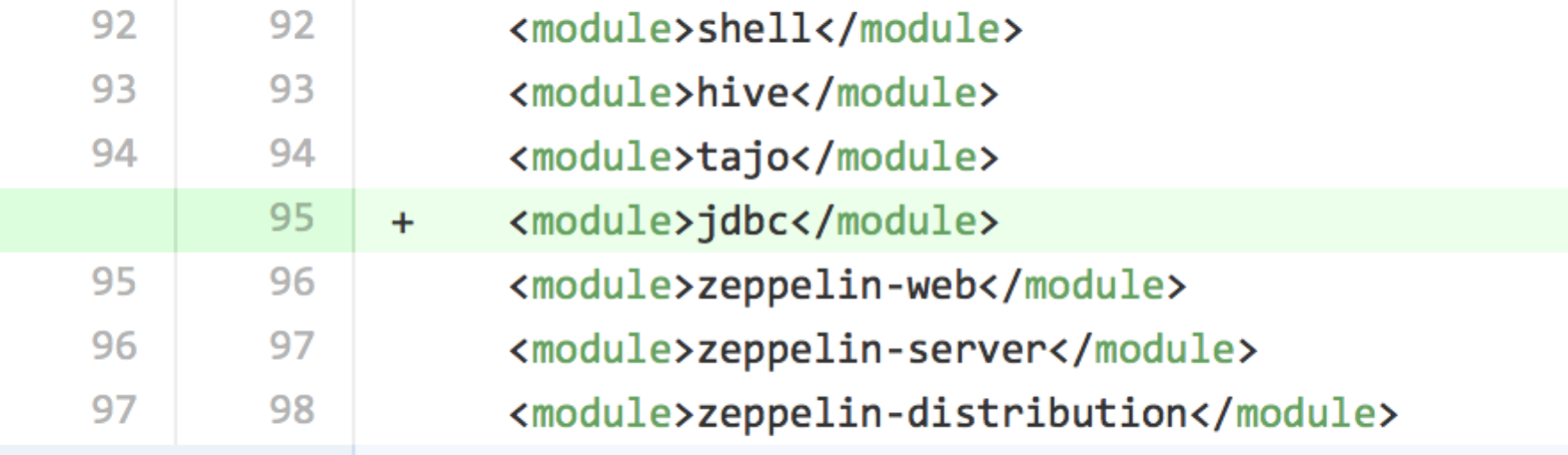
+ }

+ }

+}

+

* pom.xml:



# 数据库设计

## 表名 +中文说明

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 备注 |
|  |  |  |