

ЛАБОРАТОРНАЯ РАБОТА 3

(в предыдущих версиях – 5 часть комплекса лабораторных работ)

Оглавление

Выполнение	2
5-я часть: Установка OLAP-инструментов.....	2
5.1. Подготовительный этап.....	2
5.2. Установка Pentaho Community Edition 5.2	4
5.3 Настройка MySQL для работы с Pentaho.....	4
5.4 Запуск Pentaho Community Edition.	12
5.5 Установка Saiku Analytics и других плагинов.....	14
5.6. Настройка соединения	15
Заключение	19

Выполнение

5-я часть: Установка OLAP-инструментов

5.1. Подготовительный этап

Pentaho CE написан на Java и для его работы необходимо установить виртуальную машину Java 7 Development Kit.

1. Скачать JDK 7 с официального сайта Oracle (<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>). Обратите внимание! Необходимо установить именно седьмую версию JDK. Установка производится поэтапно, следуя инструкциям мастера установки.

2. После установки необходимо установить следующие системные переменные:

```
1 JAVA_HOME = C:\Program Files (x86)\Java\jdk1.7.0_71
2 JRE_HOME = C:\Program Files (x86)\Java\jdk1.7.0_71\jre
3 PENTAH0_JAVA_HOME = C:\Program Files (x86)\Java\jdk1.7.0_71
```

Рисунок 5.3.1 – Значения системных переменных окружения

Для установки системных переменных необходимо открыть свойства системы (“Мой компьютер” — “Свойства”) и выбрать опцию “Дополнительные параметры системы” и на вкладке “Дополнительно” открыть “Переменные среды” (рис. 5.3.2.).

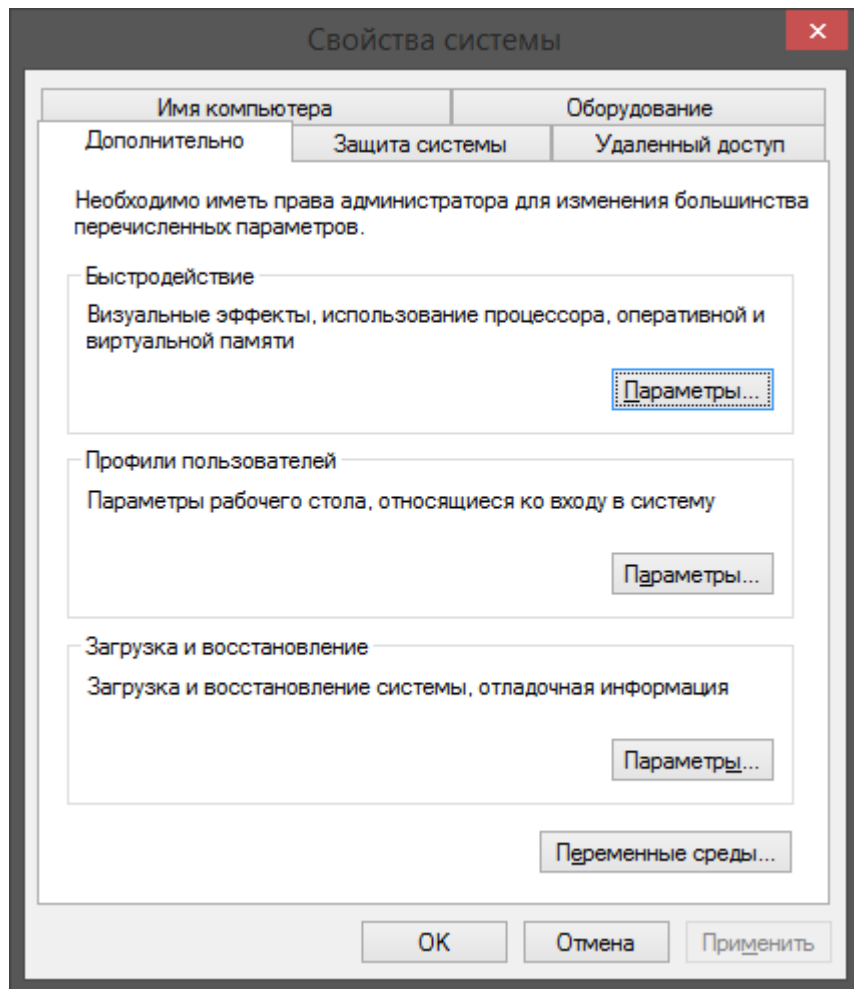


Рисунок 5.3.2 – Окно “Свойства системы”

Откроется окно “Переменные среды” (рис. 5.3.3) в котором необходимо создать переменные окружения (рис. 5.3.1) и применить все изменения.

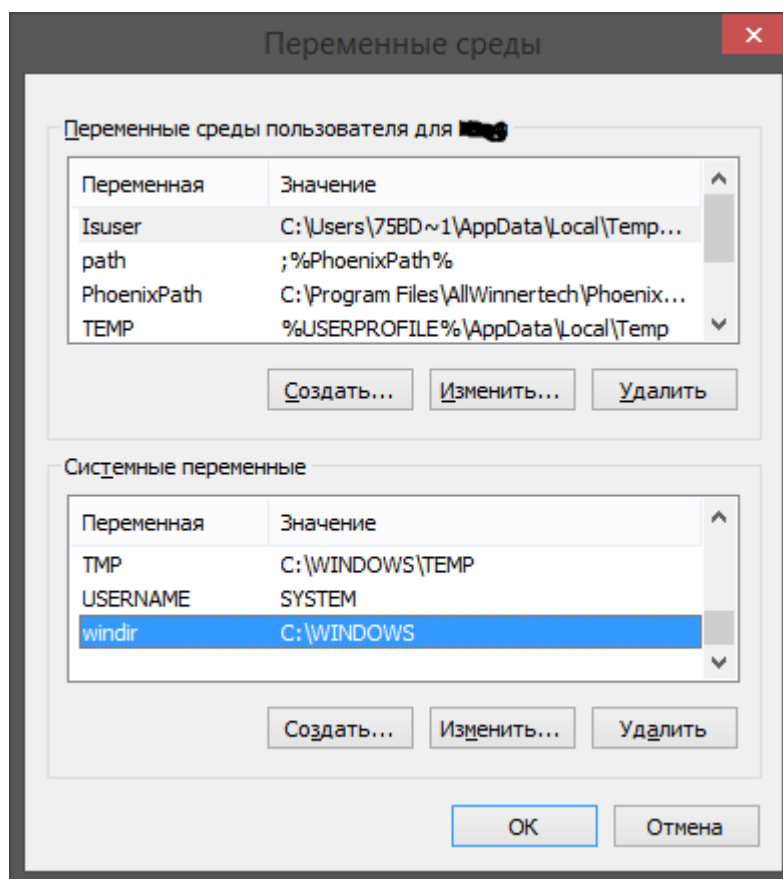


Рисунок 5.3.3 – Окно “Переменные среды”

5.2. Установка Pentaho Community Edition 5.2

1. Скачать архив с Penataho с sourceforge (<http://sourceforge.net/projects/pentaho/files/Business%20Intelligence%20Server/5.2/>)
2. Распаковать архив в удобную для работы директорию.

5.3 Настройка MySQL для работы с Pentaho

1. Для корректной работы Pentaho необходимо создать следующие базы данных MySQL: *hibernate*, *quartz* и *jackrabbit* и пользователей *hibuser*,

pentaho_user, *jcr_user* для этого нужно исполнить три .sql запроса, которые находятся в папке "...\\biserver-ce\\data\\mysql5".

2. Настройка JDBC Security. Далее, описывается то, как нужно настроить Pentaho BI Platform JDBC Security для использования MySQL сервера вместо базы данных HSQL, размещаемой в оперативной памяти.

2.1. Файл «applicationContext-spring-security-hibernate.properties»

Измените файл «...pentaho-solutions\\system\\applicationContext-spring-security-hibernate.properties», оригинальный участок кода, который необходимо заменить представлен ниже.

```
1 jdbc.driver=org.hsqldb.jdbcDriver
2 jdbc.url=jdbc:hsqldb:hsql://localhost:9001/hibernate
3 jdbc.username=hibuser
4 jdbc.password=password
5 hibernate.dialect=org.hibernate.dialect.HSQLDialect
```

Рисунок 5.3.3 – Оригинальный участок кода

Внесите изменения в файл, заменив указанный выше текст, на следующий:

```
1 jdbc.driver=com.mysql.jdbc.Driver
2 jdbc.url=jdbc:mysql://localhost:3306/hibernate
3 jdbc.username=hibuser
4 jdbc.password=password
5 hibernate.dialect=org.hibernate.dialect.MySQLDialect
```

Рисунок 5.3.4 – Измененный участок кода

2.2. Файл «hibernate-settings.xml»

Измените файл «...pentaho-solutions\\system\\hibernate\\hibernate-settings.xml», оригинальный участок кода, который необходимо заменить представлен ниже.

```
<config-file>system/hibernate/hsql.hibernate.cfg.xml</config-file>
```

Рисунок 5.3.5 – Оригинальный участок кода

Внесите изменения, чтобы получить следующий код:

```
<config-file>system/hibernate/mysql5.hibernate.cfg.xml</config-file>
```

Рисунок 5.3.6 – Измененный участок

2.3. Файл «quartz.properties»

Измените файл «...pentaho-solutions\system\quartz\quartz.properties» .

```
org.quartz.jobStore.driverDelegateClass =  
org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
```

Рисунок 5.3.7 – Оригинальный участок кода

Внесите изменения, чтобы получить следующий код:

```
org.quartz.jobStore.driverDelegateClass =  
org.quartz.impl.jdbcjobstore.StdJDBCDelegate
```

Рисунок 5.3.8 – Измененный участок

2.4. Файл «applicationContext-spring-security-jdbc.xml»

Измените файл «...pentaho-solutions\system\applicationContext-spring-security-jdbc.xml» , оригинальный участок кода, который необходимо заменить представлен ниже.

```
1 # The fully qualified Java class name of the JDBC driver to be used  
datasource.driver.classname=org.hsqldb.jdbcDriver  
  
2 # The connection URL to be passed to our JDBC driver to establish a  
connection  
datasource.url=jdbc:hsqldb:hsql://localhost:9002/userdb  
  
3 # The connection username to be passed to our JDBC driver to establish a  
connection  
datasource.username=sa  
  
4 # The connection password to be passed to our JDBC driver to establish a  
connection  
datasource.password=  
  
# The SQL query that will be used to validate connections from this pool  
before returning them to the caller.  
# This query must be an SELECT statement that returns at least one row.  
# HSQLDB: SELECT 1 FROM INFORMATION_SCHEMA.SYSTEM_USERS  
# MySQL, H2, MS-SQL, PostgreSQL, SQLite: SELECT 1  
# ORACLE: SELECT 1 FROM DUAL  
5 datasource.validation.query=SELECT 1 FROM INFORMATION_SCHEMA.SYSTEM_USERS
```

Рисунок 5.3.9 – Оригинальный участок кода

Внесите изменения в файл, заменив указанный выше текст, на следующий:

```
1 datasource.driver.classname=com.mysql.jdbc.Driver
2 datasource.url=jdbc:mysql://localhost:3306/hibernate
3 datasource.username=hibuser
4 datasource.password=password
5 datasource.validation.query=SELECT 1
```

Рисунок 5.3.10 – Измененный участок кода

3. Далее необходимо настроить Hibernate and Quartz для того, чтобы использовать базы данных hibernate и quartz, созданные ранее.

3.1. Файл «... tomcat\webapps\pentaho\META-INF\context.xml»

```
1 <Resource name="jdbc/Hibernate" auth="Container" type="javax.sql.DataSource"
2 factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20"
3 maxIdle="5"
4 maxWait="10000" username="hibuser" password="password"
5 driverClassName="org.hsqldb.jdbcDriver"
6 url="jdbc:hsqldb:hsqldb://localhost/hibernate"
7 validationQuery="select count(*) from INFORMATION_SCHEMA.SYSTEM_SEQUENCES" />

8 <Resource name="jdbc/Quartz" auth="Container" type="javax.sql.DataSource"
9 factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20"
10 maxIdle="5"
11 maxWait="10000" username="pentaho_user" password="password"
12 driverClassName="org.hsqldb.jdbcDriver"
13 url="jdbc:hsqldb:hsqldb://localhost/quartz"
14 validationQuery="select count(*) from INFORMATION_SCHEMA.SYSTEM_SEQUENCES" />
```

Рисунок 5.3.11 – Оригинальный участок кода

Внесите изменения в файл, заменив указанный выше текст, на следующий:

```
1 <Resource name="jdbc/Hibernate" auth="Container" type="javax.sql.DataSource"
2 factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20"
3 maxIdle="5"
4 maxWait="10000" username="hibuser" password="password"
5 driverClassName="com.mysql.jdbc.Driver"
6 url="jdbc:mysql://localhost:3306/hibernate"
7 validationQuery="select 1" />

8 <Resource name="jdbc/Quartz" auth="Container" type="javax.sql.DataSource"
9 factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20"
10 maxIdle="5"
11 maxWait="10000" username="pentaho_user" password="password"
12 driverClassName="com.mysql.jdbc.Driver"
13 url="jdbc:mysql://localhost:3306/quartz"
14 validationQuery="select 1" />
```

Рисунок 5.3.12 – Измененный участок кода

При этом убедитесь, чтобы был удален файл, представляющий копию данного: «...tomcat\conf\Catalina\localhost\pentaho.xml» или внесите в него соответствующие изменения.

3.2. Файл «...pentaho-solutions\system\jackrabbit\repository.xml»

Часть» FileSystem». Закомментируйте следующий исходный код:

```
1 <FileSystem class="org.apache.jackrabbit.core.fs.local.LocalFileSystem">
2   <param name="path" value="${rep.home}/repository"/>
3 </FileSystem>
```

Рисунок 5.3.13 – Оригинальный участок кода

Расскомментируйте следующий код в части «FileSystem»

```
1 <FileSystem class="org.apache.jackrabbit.core.fs.db.DbFileSystem">
2   <param name="driver" value="com.mysql.jdbc.Driver"/>
3   <param name="url" value="jdbc:mysql://localhost:3306/jackrabbit"/>
4   <param name="user" value="jcr_user"/>
5   <param name="password" value="password"/>
6   <param name="schema" value="mysql"/>
7   <param name="schemaObjectPrefix" value="fs_repos_"/>
8 </FileSystem>
```

Рисунок 5.3.14 – Участок кода, который необходимо расскомментировать

Часть «DataStore». Закомментируйте следующий исходный код:

```
<DataStore class="org.apache.jackrabbit.core.data.FileDataStore"/>
```

Рисунок 5.3.15 – Оригинальный участок кода

Расскомментируйте следующий код в части «DataStore»

```
1 <DataStore class="org.apache.jackrabbit.core.data.db.DbDataStore">
2   <param name="url" value="jdbc:mysql://localhost:3306/jackrabbit"/>
3   <param name="user" value="jcr_user"/>
4   <param name="password" value="password"/>
5   <param name="databaseType" value="mysql"/>
6   <param name="driver" value="com.mysql.jdbc.Driver"/>
7   <param name="minRecordLength" value="1024"/>
8   <param name="maxConnections" value="3"/>
9   <param name="copyWhenReading" value="true"/>
10  <param name="tablePrefix" value=""/>
11  <param name="schemaObjectPrefix" value="ds_repos_"/>
12 </DataStore>
```

Рисунок 5.3.16 – Участок кода, который необходимо расскомментировать

Часть «FileSystem Workspace». Закомментируйте исходный код, представленный ниже:

```
1 <FileSystem class="org.apache.jackrabbit.core.fs.local.LocalFileSystem">
2   <param name="path" value="${wsp.home}"/>
3 </FileSystem>
```

Рисунок 5.3.17 – Оригинальный участок кода

Расскомментируйте следующий код в части «FileSystem Workspace»

```
1 <FileSystem class="org.apache.jackrabbit.core.fs.db.DbFileSystem">
2   <param name="driver" value="com.mysql.jdbc.Driver"/>
3   <param name="url" value="jdbc:mysql://localhost:3306/jackrabbit"/>
4   <param name="user" value="jcr_user"/>
5   <param name="password" value="password"/>
6   <param name="schema" value="mysql"/>
7   <param name="schemaObjectPrefix" value="fs_ws_"/>
8 </FileSystem>
```

Рисунок 5.3.18 – Участок кода, который необходимо расскомментировать

Часть «PersistenceManager». Закомментируйте исходный код, представленный ниже:

```
1 <PersistenceManager
2   class="org.apache.jackrabbit.core.persistence.pool.H2PersistenceManager">
3   <param name="url" value="jdbc:h2:${wsp.home}/db"/>
4   <param name="schemaObjectPrefix" value="${wsp.name}_"/>
5 </PersistenceManager>
```

Рисунок 5.3.19 – Оригинальный участок кода

Расскомментируйте следующий код в части «PersistenceManager»

```
1 <PersistenceManager
2   class="org.apache.jackrabbit.core.persistence.bundle.MySqlPersistenceManager"
3 >
4   <param name="driver" value="com.mysql.jdbc.Driver"/>
5   <param name="url" value="jdbc:mysql://localhost:3306/jackrabbit"/>
6   <param name="user" value="jcr_user" />
7   <param name="password" value="password" />
8   <param name="schema" value="mysql"/>
9   <param name="schemaObjectPrefix" value="${wsp.name}_pm_ws_"/>
10 </PersistenceManager>
```

Рисунок 5.3.20 – Участок кода, который необходимо расскомментировать

Часть «FileSystem Versioning». Закомментируйте исходный код, представленный ниже:

```
1 <FileSystem class="org.apache.jackrabbit.core.fs.local.LocalFileSystem">
2   <param name="path" value="${rep.home}/version" />
3 </FileSystem>
```

Рисунок 5.3.21 – Оригинальный участок кода

Расскомментируйте следующий код в части «FileSystem Versioning»

```
1 <FileSystem class="org.apache.jackrabbit.core.fs.db.DbFileSystem">
2   <param name="driver" value="com.mysql.jdbc.Driver"/>
3   <param name="url" value="jdbc:mysql://localhost:3306/jackrabbit"/>
4   <param name="user" value="jcr_user"/>
5   <param name="password" value="password"/>
6   <param name="schema" value="mysql"/>
7   <param name="schemaObjectPrefix" value="fs_ver_"/>
8 </FileSystem>
```

Рисунок 5.3.22 – Участок кода, который необходимо расскомментировать

Часть «PersistenceManager Versioning». Закомментируйте исходный код, представленный ниже:

```
1 <PersistenceManager
  class="org.apache.jackrabbit.core.persistence.pool.H2PersistenceManager">
2   <param name="url" value="jdbc:h2:${rep.home}/version/db"/>
3   <param name="schemaObjectPrefix" value="version_"/>
4 </PersistenceManager>
```

Рисунок 5.3.23 – Оригинальный участок кода

Расскомментируйте следующий код в части «PersistenceManager Versioning»

```
1 <PersistenceManager
  class="org.apache.jackrabbit.core.persistence.bundle.MySqlPersistenceManager"
2 >
3   <param name="driver" value="com.mysql.jdbc.Driver"/>
4   <param name="url" value="jdbc:mysql://localhost:3306/jackrabbit"/>
5   <param name="user" value="jcr_user" />
6   <param name="password" value="password" />
7   <param name="schema" value="mysql"/>
8   <param name="schemaObjectPrefix" value="pm_ver_"/>
  </PersistenceManager>
```

Рисунок 5.3.24 – Участок кода, который необходимо расскомментировать

4. Последний этап – это отмена автоматического запуска базы данных Hupersonic , для этого необходимо внести некоторые изменения в следующий файл:

4.1. Файл « ... tomcat\webapps\pentaho\WEB-INF\web.xml»

Первая часть этого файла, которую необходимо удалить выглядит следующим образом:

```
1 <!-- [BEGIN HSQLDB DATABASES] -->
2 <context-param>
3   <param-name>hsqldb-databases</param-name>
4   <param-value>sampledata@../../data/hsqldb/sampledata,
5                 hibernate@../../data/hsqldb/hibernate,
6                 quartz@../../data/hsqldb/quartz</param-value>
7 </context-param>
8 <!-- [END HSQLDB DATABASES] -->
```

Рисунок 5.3.25 – Первый участок кода, который необходимо удалить

Вторая часть этого файла, которую необходимо удалить выглядит следующим образом:

```
1 <!-- [BEGIN HSQLDB STARTER] -->
2 <listener>
3   <listener-class>
4     org.pentaho.platform.web.http.context.HsqldbStartupListener
5   </listener-class>
6 </listener>
7 <!-- [END HSQLDB STARTER] -->
```

Рисунок 5.3.26 – Второй участок кода, который необходимо удалить

5. Необходимо обновить Java библиотеку для работы с MySQL. Её можно скачать здесь: <http://mvnrepository.com/artifact/mysql/mysql-connector-java/5.1.30>

После этого файл «mysql-connector-java-5.1.30.jar» необходимо поместить в «...tomcat\lib» и удалить имеющуюся там устаревшую библиотеку «mysql-connector-java-5.1.17.jar»

5.4 Запуск Pentaho Community Edition.

5.4.1. Запуск MySQL.

Перед непосредственным запуском Pentaho Community Edition необходимо запустить MySQL для работы с базой данных.

5.4.2. Запустите веб-сервера Pentaho Community Edition.

Для его запуска необходимо выполнить скрипт «start-pentaho.bat» находящийся в корневой директории программы.

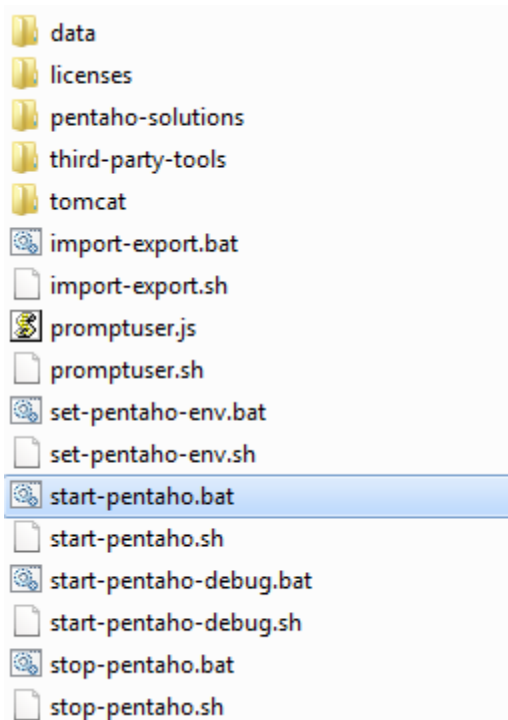


Рис. 5.4.2.1 – Файл запуска в корневой директории.

После этого требуется подождать окончания процесса запуска, который может занять до нескольких минут.

5.4.3. Контрольная панель

Зайдите в браузер и перейдите на адрес localhost:8080. Если сервер успел запуститься, то вы попадете на страницу логина, которая выглядит так:

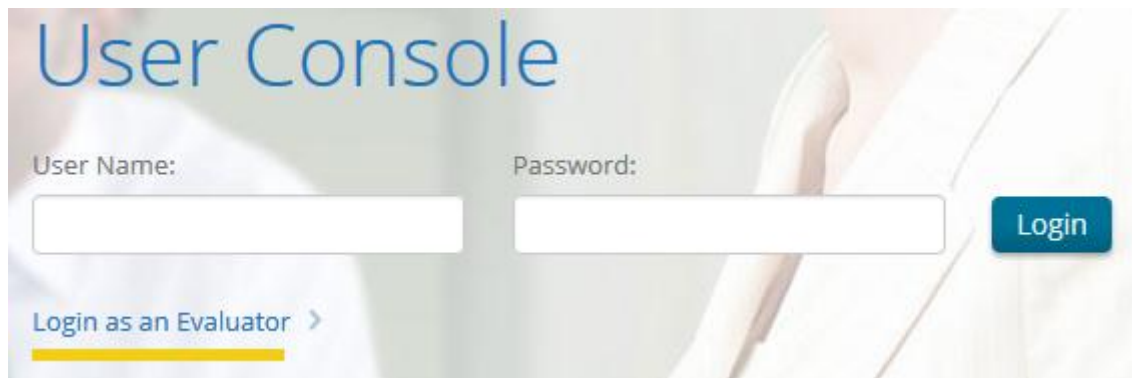
The screenshot shows the 'User Console' login interface. It features a title 'User Console' in blue. Below it are two input fields: 'User Name:' and 'Password:'. To the right of the password field is a blue 'Login' button. At the bottom left, there is a link 'Login as an Evaluator' with a right-pointing arrow, which is underlined in yellow.

Рис. 5.4.3.1 – Страница логина Pentaho.

Нажмите “Login as Evaluator” и затем нажмите на кнопку “Go” в разделе “Admin”.

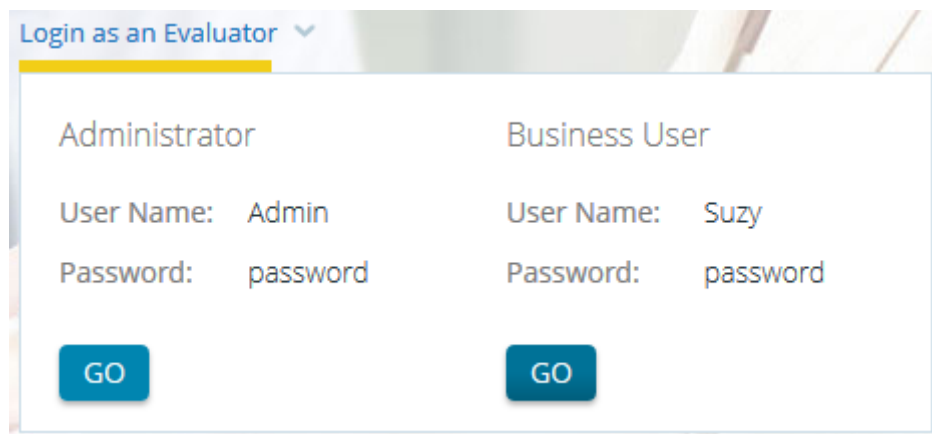
The screenshot shows a dropdown menu titled 'Login as an Evaluator' with a downward arrow. The menu is divided into two columns. The left column is titled 'Administrator' and contains the text 'User Name: Admin' and 'Password: password', followed by a blue 'GO' button. The right column is titled 'Business User' and contains the text 'User Name: Suzy' and 'Password: password', followed by a blue 'GO' button.

Рис. 5.4.3.2 – Меню Login as an Evaluator.

После логина вы попадете на контрольную панель Pentaho.

5.5 Установка Saiku Analytics и других плагинов

Для дальнейшей работы потребуется плагин Saiku Analytics.

Для установки плагинов в Pentaho существует специальный Marketplace, для доступа к нему нужно нажать Home и во всплывающем меню выбрать Marketplace.

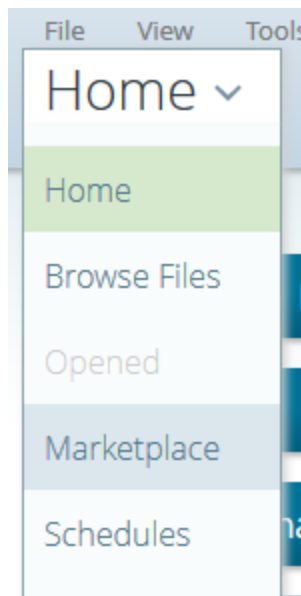


Рис. 5.5.1 – Меню Home.

После загрузки страницы Marketplace, найдите в списке Saiku Analytics и нажмите на кнопку Install. После успешной установки вы увидите следующее окно:

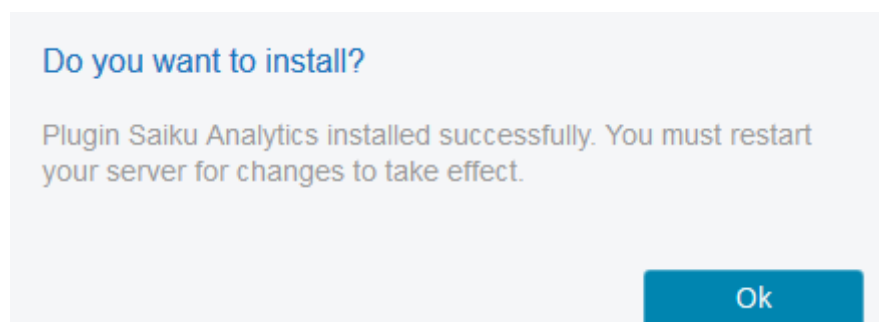


Рис. 5.5.2 – Извещение об удачной установке плагина.

5.6. Настройка соединения

Для настройки соединения необходимо нажать “Manage Data Sources” в главном меню.

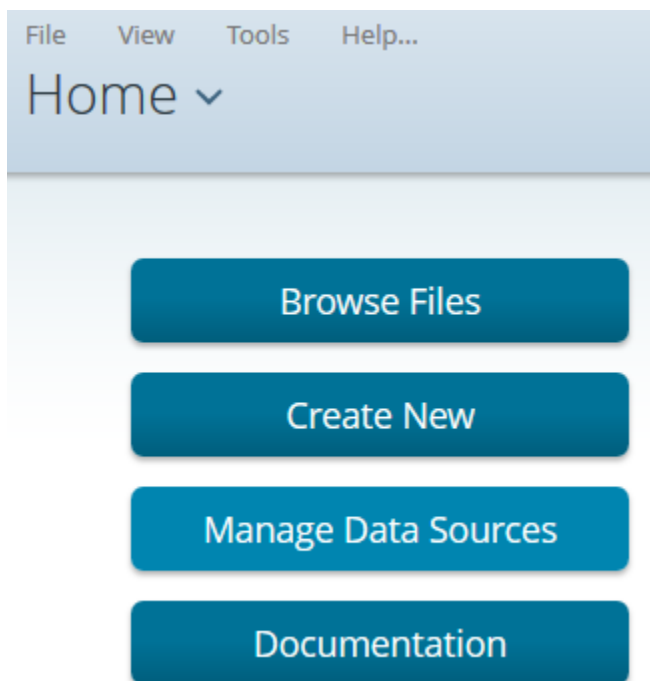


Рис. 5.6.1 – Главное меню.

Затем, в открывшемся окне нажмите “New Data Source”.

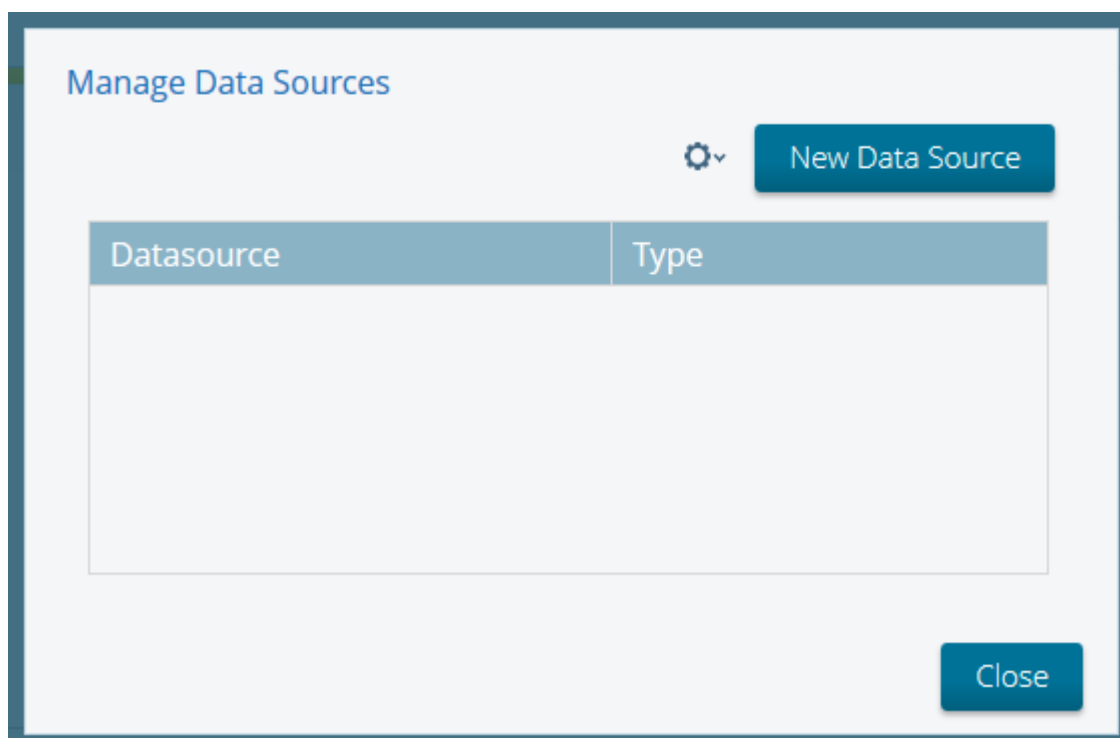
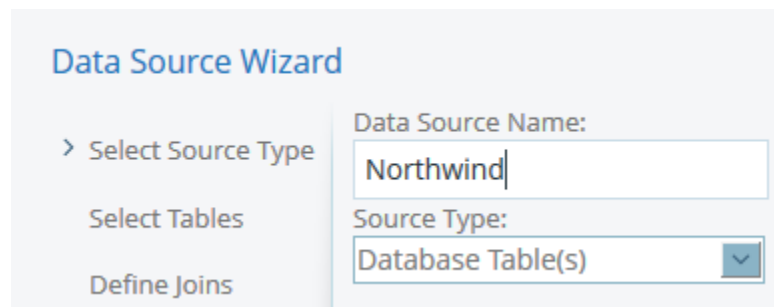


Рис. 5.6.2 – Окно Manage Data Sources

Если после нажатия “Manage Data Sources” окно не открывается, то необходимо удалить папку «.../pentaho-solutions/system/jackrabbit/repository», а затем перезапустить сервер.

Во всплывшем окне в поле “Data Source Name” выберите желаемое имя и в поле “Data Source Type” укажите “Database Table(s)”.



Data Source Wizard

> Select Source Type

Select Tables

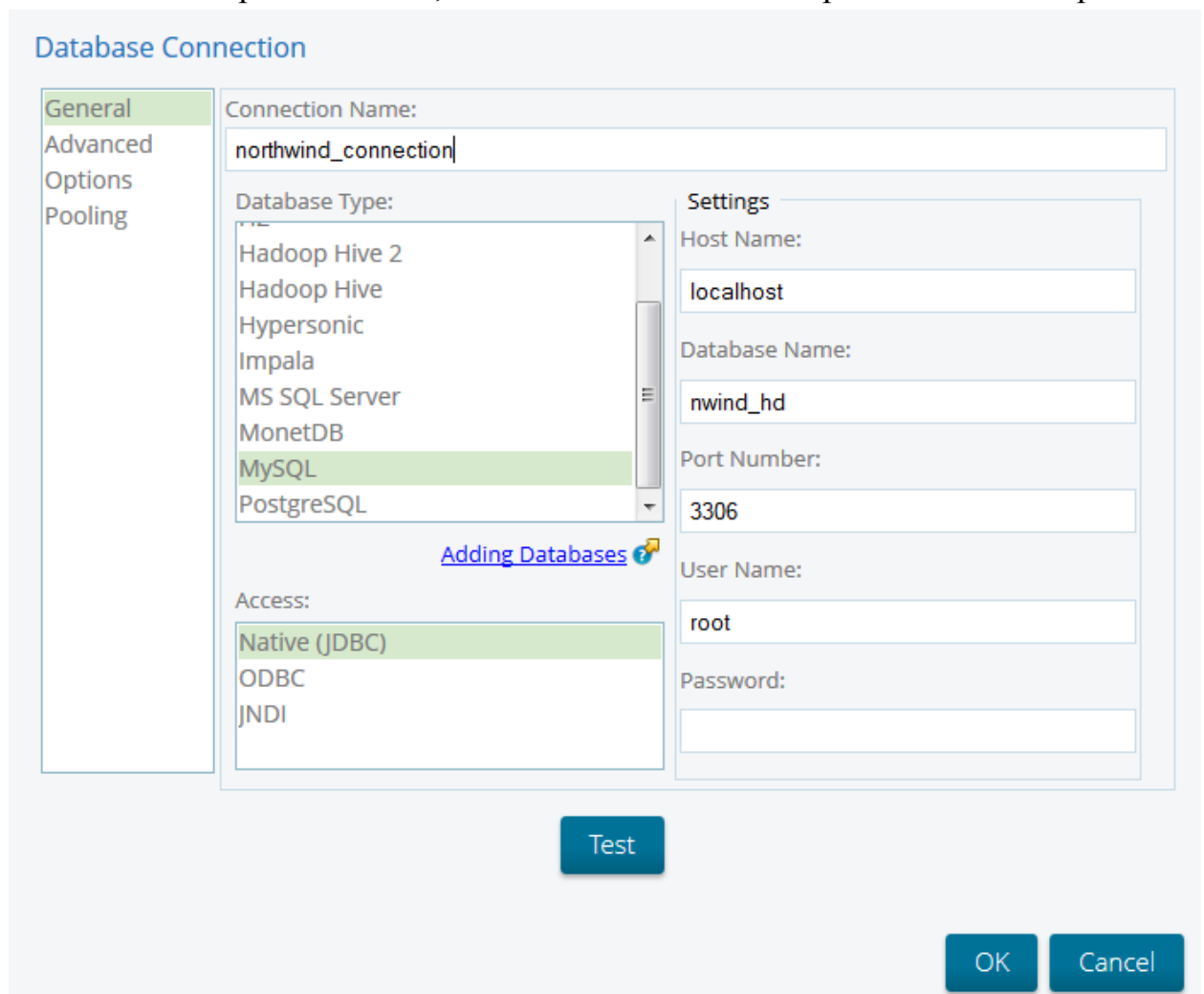
Define Joins

Data Source Name: Northwind

Source Type: Database Table(s)

Рис. 5.6.3 – Пример заполнения полей в окне Data Source Wizard

После этого настройте новое соединение в окне Database Connection, вызвать которое можно, нажав “+” в открывшемся подразделе.



Database Connection

General

Advanced

Options

Pooling

Connection Name: northwind_connection

Database Type:

- Hadoop Hive 2
- Hadoop Hive
- Hypersonic
- Impala
- MS SQL Server
- MonetDB
- MySQL
- PostgreSQL

Adding Databases

Access:

- Native (JDBC)
- ODBC
- JNDI

Settings

Host Name: localhost

Database Name: nwind_hd

Port Number: 3306

User Name: root

Password:

Test

OK Cancel

Рис. 5.6.4 – Пример заполнения полей в окне Database Connection

Заполнить поля следует следующим образом:

Connection Name: произвольное имя на латинице

Database Type: MySQL

Host Name: localhost

Database Name: название базы данных, в которой хранится хранилище
(в данном примере nwind_hd)

User Name: root (или иное имя пользователя)

Password: оставьте пустым, если используете настройки по умолчанию, или введите свой пароль.

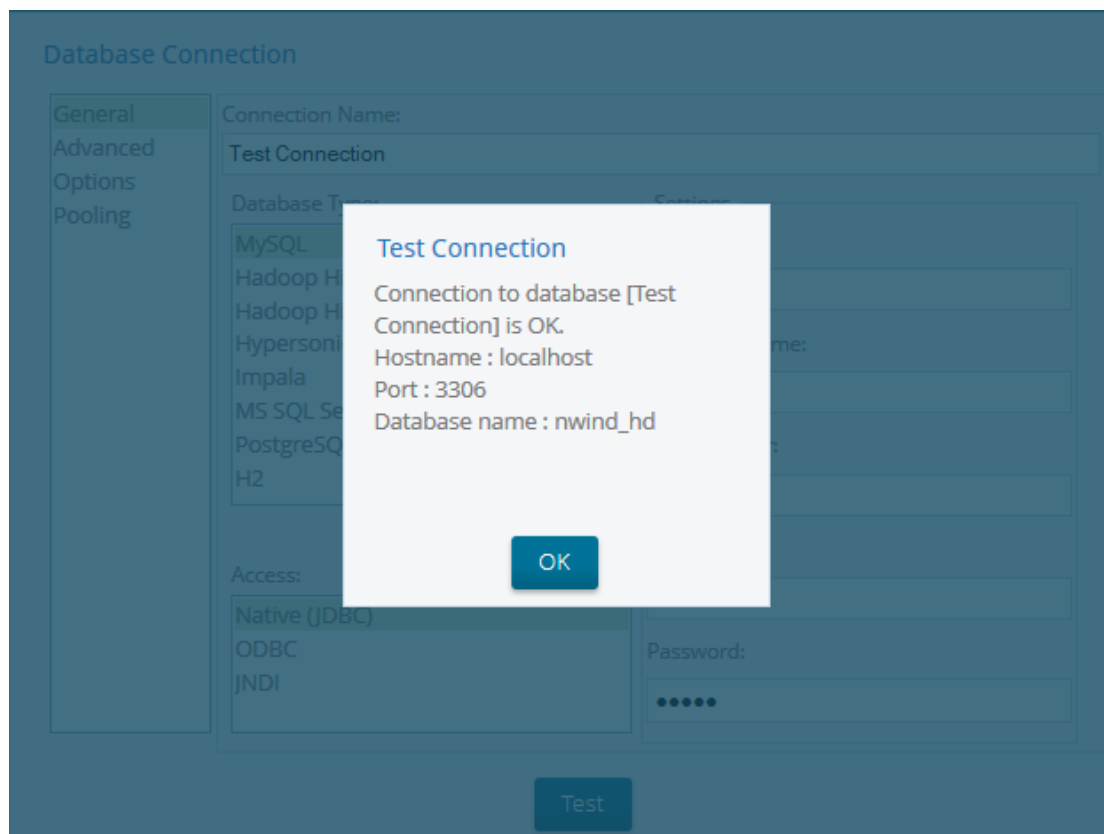


Рисунок 6.1.2 – Тестирование соединения

После заполнения всех этих полей, нажмите «ОК».

Data Source Name:
Northwind

Source Type:
Database Table(s)

Select a database connection and click Next to choose from a list of the available database tables.

Connection:

northwind_connection

Рис. 5.6.5 – Готовый результат.

Заключение

В ходе выполнения данной работы рассмотрен процесс инсталляции комплекса программ для работы OLAP хранилища, которые включают в себя Pentaho, MySQL и JDK7. Так же был рассмотрен процесс конфигурации данного набора программного обеспечения для их взаимодействия