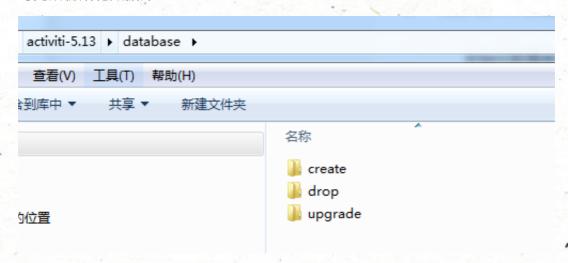
- 1 下载activiti的文件,这里采用5.13
- 2 初始化activiti数据库

初始化数据库有两种方式:

1) 执行sql脚本:如下图 create是创建,drop是删除后创建,upgrade是修改。第一次初始化直接用create就好,找到里面的sql脚本执行即可完成初始化数据库



2) 通过程序执行 编辑pom文件,引入jar包

<version>5.14</version>

```
</dependency>
                             <dependency>
                                 <groupId>mysql</groupId>
                                 <artifactId>mysql-connector-
java</artifactId>
                                 <version>5.0.8</version>
                            </dependency>
                        </dependencies>
                        <build>
                            <plugins>
                                 <plugin>
<groupId>org.apache.maven.plugins</groupId>
                                     <artifactId>maven-compiler-
plugin</artifactId>
                                     <version>2.3.2</version>
                                     <configuration>
                                         <source>1.8</source>
                                         <target>1.8</target>
                                     </configuration>
                                 </plugin>
                            </plugins>
                        </build>
                    在resources中新建一个activiti.cfg.xml文件 并且编辑文件
                        <beans
xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
```

```
http://www.springframework.org/schema/beans/spring-beans.xsd
                            <bean id="processEngineConfiguration"</pre>
class="org.activiti.engine.impl.cfg.StandaloneProcessEngineConfiguration">
                                cproperty name="jdbcDriver"
value="com.mysgl.jdbc.Driver"/>
                                operty name="jdbcUrl"
value="jdbc:mysql://120.79.183.246:3306/db_activiti?
useUnicode=true&characterEncoding=utf8"/>
                                cproperty name="jdbcUsername"
value="root"/>
                                cproperty name="jdbcPassword"
value="root"/>
                                <!-- 没有表, 创建表 -->
                                property
name="databaseSchemaUpdate" value="true"/>
                            </bean>
                        </beans>
```

 $\label{lem:processEngineConfiguration.createProcessEngineConfigurationFromResource ("activiti.cfg.xml").buildProcessEngine();$

初始化完成之后数据库会生成activiti的23张表

- ☐
 ☐ db_activiti
 - 🗆 🗼 Tables
 - ACT_GE_BYTEARRAY
 - MACT_GE_PROPERTY
 - ACT HI ACTINST

 - ACT_HI_COMMENT
 - ACT HI DETAIL
 - MACT_HI_IDENTITYLINK

 - ACT_HI_TASKINST
 - ACT_HI_VARINST

 - MEMBERSHIP

 - ACT_RE_DEPLOYMENT
 - ACT_RE_MODEL
 - ACT_RE_PROCDEF
 - ACT_RU_EVENT_SUBSCR
 - ACT_RU_EXECUTION
 - MCT_RU_IDENTITYLINK

 - ACT_RU_TASK

Activiti 数据库中表的命名都是以 ACT_开头的。第二部分是一个两个字符用例表的标识。此用例大体与服务 API 是匹配的。

- ■ ACT_RE_*: 'RE' 代表 repository。带此前缀的表包含的是静态信息,如,流程定义、流程的资源(图片、规则,等)。
- ■ ACT_RU_*: 'RU' 代表 runtime。就是这个运行时的表存储着流程变量、用户任务、变量、作业,等中的运行时的数据。Activiti 只存储流程实例执行期间的运行时数据,当流程实例结束时,删除这些记录。这就使这些运行时的表保持的小且快。
- ■ ACT_ID_*: 'ID'代表identity。这些表包含着标识的信息,如用户、用户组、等等。

- ■ ACT_HI_*: 'HI'代表history 。就是这些表包含着历史的相关数据,如结束的流程实例、变量、任务、等等。
 - ■ ACT_GE_*: 普通数据, 各种情况都使用的数据

3 安装eclipse插件画流程图

activiti流程图实际上就是xml文件,但是编写xml文件比较麻烦,activiti提供了工具可以通过画图的形式自动生成xml文件.

两种工具:一种通过B/S形式访问网站来画流程图,然后导出 另外一种通过eclipse插件。插件安装分两种,一种下载 插件拷贝到eclipse下面。另外一种通过eclipse联网的方式下载。

这里采用通过eclipse联网的形式, help -- 》

install new software 添加 activiti designer - http://activiti.org/designer/update然后一步步的安装,最后重启即可

打开eclipse之后配置如图:

