

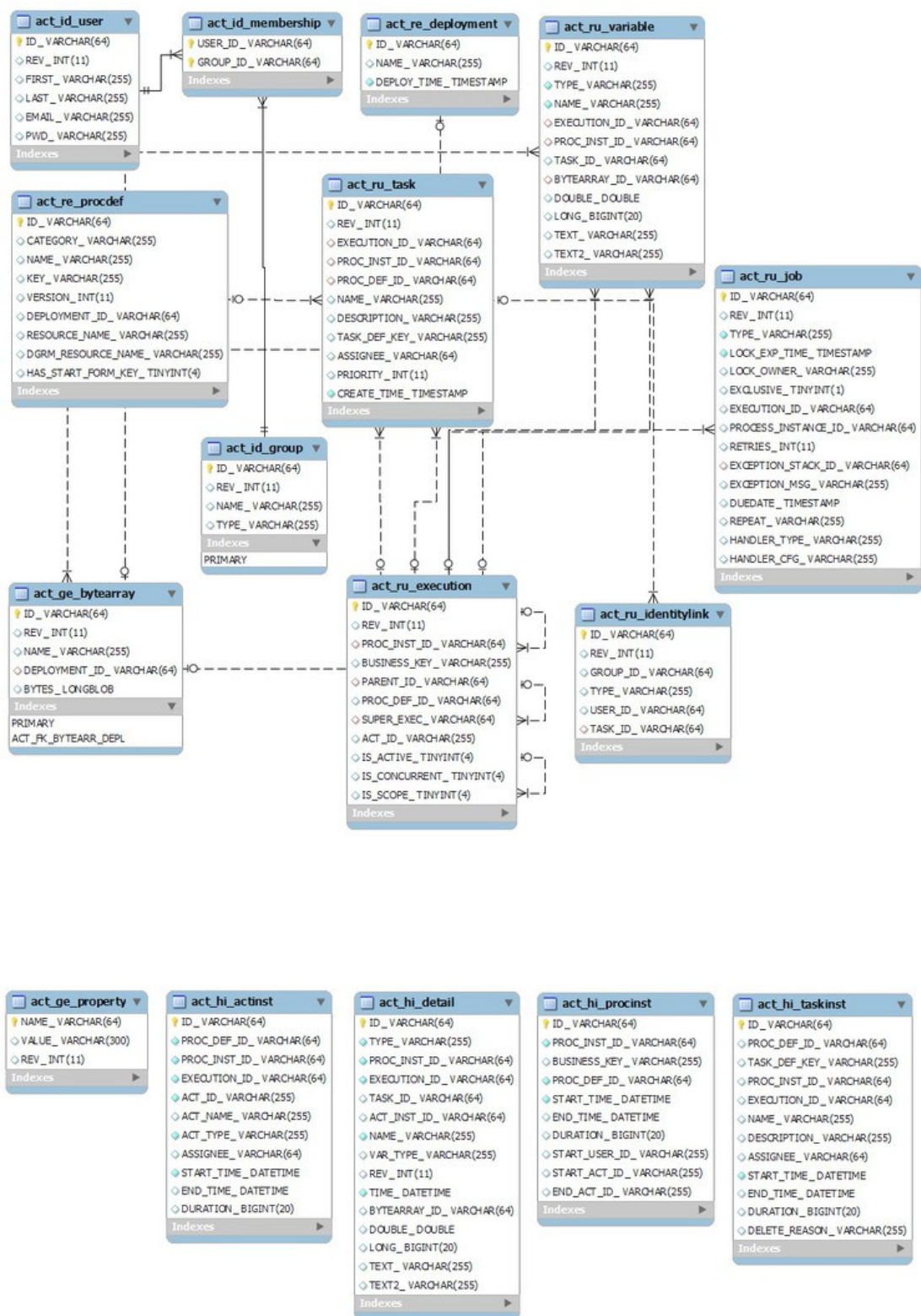
## Activiti workflow引擎数据库表结构

### 数据库表的命名

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' 表示 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' 表示 runtime。这是运行时的表存储着流程变量，用户任务，变量，职责（job）等运行时的数据。Activiti 只存储实例执行期间的运行时数据，当流程实例结束时，将删除这些记录。这就保证了这些运行时的表小且快。
- ACT\_ID\_\*:' ID' 表示 identity。这些表包含标识的信息，如用户，用户组，等等。
- ACT\_HI\_\*:' HI' 表示 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。

### 数据库表结构图



## 数据库表结构说明

- ACT\_GE\_PROPERTY:属性数据表。存储这个流程引擎级别的数据。

1. NAME\_:属性名称
  2. VALUE\_:属性值
  3. REV\_INT:版本号
- ACT\_GE\_BYTEARRAY:用来保存部署文件的大文本数据
    1. ID\_:资源文件编号，自增长
    2. REV\_INT:版本号
    3. NAME\_:资源文件名称
    4. DEPLOYMENT\_ID\_:来自于父表 ACT\_RE\_DEPLOYMENT 的主键
    5. BYTES\_:大文本类型，存储文本字节流
  - ACT\_RE\_DEPLOYMENT:用来存储部署时需要持久化保存下来的信息
    1. ID\_:部署编号，自增长
    2. NAME\_:部署包的名称
    3. DEPLOY\_TIME\_:部署时间
  - ACT\_RE\_PROCDEF:业务流程定义数据表
    1. ID\_:流程 ID，由“流程编号：流程版本号：自增长 ID”组成
    2. CATEGORY\_:流程命名空间（该编号就是流程文件 targetNamespace 的属性值）
    3. NAME\_:流程名称（该编号就是流程文件 process 元素的 name 属性值）
    4. KEY\_:流程编号（该编号就是流程文件 process 元素的 id 属性值）
    5. VERSION\_:流程版本号（由程序控制，新增即为 1，修改后依次加 1 来完成的）

6. DEPLOYMENT\_ID\_:部署编号
7. RESOURCE\_NAME\_:资源文件名称
8. DGRM\_RESOURCE\_NAME\_:图片资源文件名称
9. HAS\_START\_FROM\_KEY\_:是否有 Start From Key

**注：此表和 ACT\_RE\_DEPLOYMENT 是多对一的关系，即，一个部署的 bar 包里可能包含多个流程定义文件，每个流程定义文件都会有一条记录在 ACT\_REPROCDEF 表内，每个流程定义的数据，都会对于 ACT\_GE\_BYTEARRAY 表内的一个资源文件和 PNG 图片文件。和 ACT\_GE\_BYTEARRAY 的关联是通过程序用 ACT\_GE\_BYTEARRAY.NAME 与 ACT\_RE\_PROCDEF.NAME\_完成的，在数据库表结构中没有体现。**

- ACT\_ID\_GROUP:用来存储用户组信息。
  1. ID\_：用户组名\*
  2. REV\_INT:版本号
  3. NAME\_:用户组描述信息\*
  4. TYPE\_:用户组类型
- ACT\_ID\_MEMBERSHIP:用来保存用户的分组信息
  1. USER\_ID\_:用户名
  2. GROUP\_ID\_:用户组名
- ACT\_ID\_USER:
  1. ID\_:用户名
  2. REV\_INT:版本号

3. FIRST\_:用户名称
  4. LAST\_:用户姓氏
  5. EMAIL\_:邮箱
  6. PWD\_:密码
- ACT\_RU\_EXECUTION :
    1. ID\_ :
    2. REV\_ : 版本号
    3. PROC\_INST\_ID\_ : 流程实例编号
    4. BUSINESS\_KEY\_ : 业务编号
    5. PARENT\_ID\_ : 找到该执行实例的父级，最终会找到整个流程的执行实例
    6. PROC\_DEF\_ID\_ : 流程 ID
    7. SUPER\_EXEC\_ : 引用的执行模板
    8. ACT\_ID\_ : 节点 id
    9. IS\_ACTIVE\_ : 是否访问
    10. IS\_CONCURRENT\_ :
    11. IS\_SCOPE\_ :
  - ACT\_RU\_TASK : 运行时任务数据表。
    1. ID\_ :
    2. REV\_ :
    3. EXECUTION\_ID\_ : 执行实例的 id
    4. PROC\_INST\_ID\_ : 流程实例的 id

5. PROC\_DEF\_ID\_ : 流程定义的id,对应 act\_re\_procdef 的id\_
  6. NAME\_ : 任务名称 , 对应 \*\*\*task 的 name
  7. PARENT\_TASK\_ID\_ : 对应父任务
  8. DESCRIPTION\_ :
  9. TASK\_DEF\_KEY\_ : \*\*\*task 的id
  10. OWNER\_ : 发起人
  11. ASSIGNEE\_ : 分配到任务的人
  12. DELEGATION\_ : 委托人
  13. PRIORITY\_ : 紧急程度
  14. CREATE\_TIME\_ : 发起时间
  15. DUE\_TIME\_ : 审批时长
- ACT\_RU\_IDENTITYLINK : 任务参与者数据表。主要存储当前节点参与者的信息。
    1. ID\_ : 标识
    2. REV\_ : 版本
    3. GROUP\_ID\_ : 组织 id
    4. TYPE\_ : 类型
    5. USER\_ID\_ : 用户 id
    6. TASK\_ID\_ : 任务 id
  - ACT\_RU\_VARIABLE : 运行时流程变量数据表。
    1. ID\_ : 标识
    2. REV\_ : 版本号

3. TYPE\_ : 数据类型
  4. NAME\_ : 变量名
  5. EXECUTION\_ID\_ : 执行实例 id
  6. PROC\_INST\_ID\_ : 流程实例 id
  7. TASK\_ID\_ : 任务 id
  8. BYTEARRAY\_ID\_ :
  9. DOUBLE\_ : 若数据类型为 double ,保存数据在此列
  10. LONG\_ : 若数据类型为 Long 保存数据到此列
  11. TEXT\_ : string 保存到此列
  12. TEXT2\_ :
- ACT\_HI\_PROCINST :
    1. ID\_ : 唯一标识
    2. PROC\_INST\_ID\_ : 流程 I D
    3. BUSINESS\_KEY\_ : 业务编号
    4. PROC\_DEF\_ID\_ : 流程定义 id
    5. START\_TIME\_ : 流程开始时间
    6. ENT\_\_TIME : 结束时间
    7. DURATION\_ : 流程经过时间
    8. START\_USER\_ID\_ : 开启流程用户 id
    9. START\_ACT\_ID\_ : 开始节点
    10. END\_ACT\_ID\_ : 结束节点
    11. SUPER\_PROCESS\_INSTANCE\_ID\_ : 父流程流程 id

12. DELETE\_REASON\_：从运行中任务表中删除原因

- ACT\_HI\_ACTINST：

1. ID\_：标识
2. PROC\_DEF\_ID\_：流程定义 id
3. PROC\_INST\_ID\_：流程实例 id
4. EXECUTION\_ID\_：执行实例
5. ACT\_ID\_：节点 id
6. ACT\_NAME\_：节点名称
7. ACT\_TYPE\_：节点类型
8. ASSIGNEE\_：节点任务分配人
9. START\_TIME\_：开始时间
10. END\_TIME\_：结束时间
11. DURATION：经过时长

- ACT\_HI\_TASKINST：

1. ID\_：标识
2. PROC\_DEF\_ID\_：流程定义 id
3. TASK\_DEF\_KEY\_：任务定义 id
4. PROC\_INST\_ID\_：流程实例 id
5. EXECUTION\_ID\_：执行实例 id
6. PARENT\_TASK\_ID\_：父任务 id
7. NAME\_：任务名称
8. DESCRIPTION\_：说明



- 9. OWNER\_ : 拥有人 ( 发起人 )
- 10. ASSIGNEE\_ : 分配到任务的人
- 11. START\_\_TIME\_ : 开始任务时间
- 12. END\_TIME\_ : 结束任务时间
- 13. DURATION\_ : 时长
- 14. DELETE\_REASON\_ : 从运行时任务表中删除的原因
- 15. PRIORITY\_ : 紧急程度
- 16. DUE\_DATE\_ :
- ACT\_HI\_DETAIL : 启动流程或者在任务 complete 之后,记录历史流程变量
  - 1. ID\_ : 标识
  - 2. TYPE\_ : variableUpdate 和 formProperty 两种值
  - 3. PROC\_INST\_ID\_ : 对应流程实例 id
  - 4. EXECUTION\_ID\_ : 对应执行实例 id
  - 5. TASK\_ID\_ : 对应任务 id
  - 6. ACT\_INST\_ID : 对应节点 id
  - 7. NAME\_ : 历史流程变量名称 , 或者表单属性的名称
  - 8. VAR\_TYPE\_ : 定义类型
  - 9. REV\_ : 版本
  - 10. TIME\_ : 导入时间
  - 11. BYTEARRAY\_ID\_
  - 12. DOUBLE\_ : 如果定义的变量或者表单属性的类型为 double , 他的值存在这里

13. LONG\_ : 如果定义的变量或者表单属性的类型为 LONG ,他的值存在这里

14. TEXT\_ : 如果定义的变量或者表单属性的类型为 string , 值存在这里

15. TEXT2\_:

- ACT\_HI\_COMMENT 意见表

1. ID\_ :标识

2. TYPE\_ :意见记录类型 为 comment 时 为处理意见

3. TIME\_ :记录时间

4. USER\_ID\_ :

5. TASK\_ID\_ : 对应任务的 id

6. PROC\_INST\_ID\_ :对应的流程实例的 id

7. ACTION\_ : 为 AddComment 时为处理意见

8. MESSAGE\_ :处理意见

9. FULL\_MSG\_ :

## 结论及总结

- 流 程 文 件 部 署 主 要 涉 及 到 3 个 表 , 分 别 是 :  
ACT\_GE\_BYTEARRAY、ACT\_RE\_DEPLOYMENT、ACT\_RE\_PROCDEF。主要完成 “部署包” --> “流程定义文件” --> “所有包内文件” 的解析部署关系。从表结构中可以看出, 流程定义的元素需要每次从数据库加载并解析, 因为流程定义的元素没有转化成数据库表来完成, 当然流程元素解析后是放在缓存中的, 具体的还需要后面详细研究。

- 流程定义中的 java 类文件不保存在数据库里。
- 组织机构的管理相对较弱，如果要纳入单点登录体系内还需要改造完成，具体改造方法有待研究。
- 运行时对象的执行与数据库记录之间的关系需要继续研究
- 历史数据的保存及作用需要继续研究。

Activiti 使用 Mybatis3 做持久化工作，可以在配置中设置流程引擎启动时创建表。

Activiti 使用到的表都是 ACT\_开头的。

ACT\_RE\_\*:流程定义存储。

ACT\_RU\_\*:流程执行记录，记录流程启动到结束的所有动作，流程结束后会清除相关记录。

ACT\_ID\_\*:用户记录，流程中使用到的用户和组。

ACT\_HI\_\*:流程执行的历史记录。

ACT\_GE\_\*:通用数据及设置。

使用到的表：

ACT\_GE\_BYTEARRAY : 流程部署的数据。

ACT\_GE\_PROPERTY : 通用设置。

ACT\_HI\_ACTINST : 流程活动的实例。

ACT\_HI\_ATTACHMENT :

ACT\_HI\_COMMENT :

ACT\_HI\_DETAIL :

ACT\_HI\_PROCINST : 流程实例。

ACT\_HI\_TASKINST : 任务实例。

ACT\_ID\_GROUP : 用户组。

ACT\_ID\_INFO :

ACT\_ID\_MEMBERSHIP :

ACT\_ID\_USER : 用户。

ACT\_RE\_DEPLOYMENT : 部署记录。

ACT\_RE\_PROCDEF : 流程定义。

ACT\_RU\_EXECUTION : 流程执行记录。

ACT\_RU\_IDENTITYLINK :

ACT\_RU\_JOB :

ACT\_RU\_TASK : 执行的任务记录。

ACT\_RU\_VARIABLE : 执行中的变量记录。

#### activiti-administrator

自带的用户管理系统，维护用户和组，需要配置数据连接参数，在 activiti-administrator\WEB-INF\applicationContext.xml 中，并加入 JDBC 驱动包。

#### activiti-cycle

PVM 活动检测的，由 activiti-rest 提供服务，不需配置。

#### activiti-explorer

可以查看用户任务和启动流程，由 activiti-rest 提供服务，不需配置。

#### activiti-kickstart

简单的点对点流程定义维护工具，需要配置数据连接，把 activiti.cfg.xml 文件放在 classes 下，并加入驱动包。

## activiti-modeler

在线编辑和维护流程定义的工具，最后以文件夹方式部署，需要配置  
activiti-modeler\WEB-INF\classes\configuration.properties 文件。

## activiti-probe

PVM 的观测服务，由 activiti-rest 提供服务，不需配置，可以查看  
deployment、processdefinition、processinstance、database。

## activiti-rest

其他几个应用的服务提供者，需要配置数据连接，把 activiti.cfg.xml 文件放在 classes 下，并加入驱动包。

