使用工作流引擎，一个非常重要的功能就是获取待办事项列表，在Activiti中，我们可以通过TaskService的相关API进行查询，这些API设计优雅，但是实际使用中往往不够方便，也缺乏灵活性，达不到技术解决方案的要求，主要有如下几个问题：

1.多数情况无法通过调用一个API满足需求，这时一个现实问题就是需要对结果集进行合并然后排序，这样就显得比较麻烦；

2.和项目业务表关联困难；

3.Activiti中相关查询返回的是Activiti定义的实体，这些实体包含的信息可能不够；

4.Activiti中的实体，可能和项目中的对象关系映射（ORM）冲突；

        鉴于上述原因，在一些大规模的项目中，Activiti提供的查询API，实际使用价值不大，我们需要另外寻找解决方案。在Activiti的查询API中，也提供原始SQL的查询接口，但是大量使用后，会发现代码不够优雅，维护困难。这个问题其实从开发者角度，查询时用用户的id，用最简单的SQL查询出来所有想要的信息是最理想的。

        分析上述缺点和需求后，我们认为通过API方式进行查询的话，总是有各种缺陷，因此把目标放在数据库上，如果能通过定义视图的方式解决问题，那么将彻底解决查询的方便性、灵活性、通用性问题。

        经过分析Activiti的数据库表，我们发现并不复杂，和待办事项有关系的表，包括ACT\_RU\_TASK、ACT\_RU\_IDENTITYLINK，ACT\_RU\_TASK中存储了任务相关信息，ACT\_RU\_IDENTITYLINK中存储了候选组和候选人信息，这里面一个比较重要的问题就是，Activiti中的候选组、候选人如何跟系统中的用户、组织、角色对应的问题，本文提供的解决方案，假定系统中有一张名为SYS\_ROLE\_USER的表，该表中存储了角色和用户的对应关系，并且Activiti中的候选组和角色是同一个概念，开发者的系统中具体是什么情况，需要开发者举一反三，本文仅提供一个设计思路。

        在Activiti中，对于一个节点，可分为受托人，候选人和候选组三种情况，后两种可以设置多个，用逗号分隔，对应到数据库中，会被拆分为ACT\_RU\_IDENTITYLINK的多条记录，这些我们都需要考虑，细节上可以通过UNION实现，下面是样例代码，该代码基于Oracle数据库，其他数据库的版本，稍后会说明。

CREATE VIEW V\_TASKLIST AS

SELECT A.ID\_ AS TASK\_ID,

       A.PROC\_INST\_ID\_ PROC\_INST\_ID,

       A.TASK\_DEF\_KEY\_ AS ACT\_ID,

       A.NAME\_ AS ACT\_NAME,

       A.ASSIGNEE\_ AS ASSIGNEE,

       A.DELEGATION\_ AS DELEGATION\_ID,

       A.DESCRIPTION\_ AS DESCRIPTION,

       TO\_CHAR(A.CREATE\_TIME\_, 'YYYY-MM-DD HH24:MI:SS') AS CREATE\_TIME,

       TO\_CHAR(A.DUE\_DATE\_,'YYYY-MM-DD HH24:MI:SS') AS DUE\_DATE,

       I.USER\_ID CANDIDATE

  FROM ACT\_RU\_TASK A

  LEFT JOIN (SELECT DISTINCT \* FROM (SELECT TASK\_ID\_, TO\_CHAR(USER\_ID\_) USER\_ID

                    FROM ACT\_RU\_IDENTITYLINK I, ACT\_RU\_TASK T

                      WHERE TASK\_ID\_ IS NOT NULL

                        AND USER\_ID\_ IS NOT NULL

                        AND I.TASK\_ID\_ = T.ID\_

                        AND T.ASSIGNEE\_ IS NULL

                        AND TYPE\_ = 'candidate'

                     UNION

                     SELECT TASK\_ID\_, R.USER\_ID

                       FROM ACT\_RU\_IDENTITYLINK I,SYS\_ROLE\_USER R,ACT\_RU\_TASK T

                      WHERE I.TASK\_ID\_ IS NOT NULL

                        AND I.GROUP\_ID\_ IS NOT NULL

                        AND I.TASK\_ID\_ = T.ID\_

                        AND T.ASSIGNEE\_ IS NULL

                        AND TYPE\_ = 'candidate'

                        AND I.GROUP\_ID\_ = R.ROLE\_ID)U) I--候选组和业务上的角色用户表关联

    ON A.ID\_ = I.TASK\_ID\_

        这个视图比较简单，主要查询了任务信息，如果还需要其他信息，比如和流程实例、流程定义等，可以自行增加其他的表关联，比如要和业务表关联需要一个很重要的字段就是BUSINESS\_KEY\_,这个和ACT\_RU\_EXECUTION表关联即可。

        这个视图定义好之后，代办查询可以用如下的更简洁的SQL实现：

SELECT \* FROM V\_TASKLIST WHERE ASSIGNEE = ：userId OR CANDIDATE = ：userId

        这样的话，和业务表关联也非常的方便，也不会受到API的限制，也不涉及和系统的ORM兼容的问题，基本上想查询什么信息就能用一个简单的SQL查询到什么信息，基本可以作为一个通用的解决方案了。

        上述例子仅提供了Oracle的代码，对于兼容多数据库的设计，比较麻烦，各种数据库都对视图的创建做了较多的限制，比如SQLServer不能在SQL中写ORDER BY，MySQL中FROM字句不能嵌套子查询，以及不同数据库字段类型定义不同等，在我们的解决方案中，基本上就是把上述SQL做了拆分，定义了若干非常小的视图，然后V\_TASKLIST视图再查询这些视图。具体上开发者可以灵活处理，本文不再展开。