数据权限

本文档仅做数据权限的阐述，访问权限：即能否操作，能否访问等基于安全的url权限不在该文档的描述范围内。

该文档包含如下关键部分：
1 基于数据权限的用户组织设计
2 数据的过滤规则
3 数据的过滤入口
4 自定义数据过滤条件

数据权限适用对象：
用户、部门、群组、岗位、角色。

需求分析：

1 查看数据
查看本人数据
查看本部门数据
查看本部门及子部门数据
查看上级部门数据
查看小于等于本人密级的数据
。。。
2 增加数据
增加某一分类的数据
增加密级小于*的数据
增加某一部门的数据
。。。
3 删除数据
删除本人数据
删除部门数据
删除本部门及子部门数据
删除密级小于本人密级的数据
。。。
4 打印数据
本人数据
本部门数据
上级部门数据
小于本人密级的数据
。。。

以上所有条件都不可为固定项，并且可以累加。累加多个规则条件以 or 区分，如果需要多个并且的条件请放到一个规则中。

例如： 查看本部门内数据，并且只能查看小于本人密集的数据
数据规则ql： x.deptId = :userDeptIds and x.secretLevel < :userSecretLevel

例如：并且者查看密级等于5的数据 或者 本人数据
方式1：
定义两条规则
规则ql 1 ：x.secretLevel = 5
规则ql 2：x.creatorUserId = :userId

方式2：
定义到一个规则ql： x.secretLevel = 5 or x.creatorUserId = :userId

这样通过自定ql的方式实现精准的数据权限过滤。

设计：

1。 用户登陆后，获取当前用户的所有context上下文，包含 拥有的角色、所属的部门、单位、子部门、岗位、群组等等。通过这些关系获取用户的规则条件。

2。 数据库规则表保存针对 用户、部门、群组、岗位、角色的关系规则ql

3。 用户查询某一个业务表的时候，通过状态来判断是否过滤规则。

4。 如果过滤规则的话，在后台reposiroty执行数据库查询的时候，通过对应业务模块和单位的规则和本人规则的交集 ，将规则条件附加到查询的where 条件中。

数据库:

规则表：

字段	类型	重要程度	描述
ID	主键		
DELETE_STATUS	删除状态	*	
ORG_ID	单位ID	*	单位ID，如果有单位限制，则过滤单位的规则。
SORT	排序字段		
VERSION	乐观锁	*	系统自身引用
NAME	规则名称	*	定义的规则名称
REMARK	备注		
SHORT_SPELL	短拼		
SPELL	全拼		
STATE	状态	*	是否启用 1为启用 2为禁用
CONDITION_JPQL	自定义JPQL条件	*****	QL条件会附加到查询的where子句中
MODEL_CLASS_NAME	针对的业务模块	*****	业务表的MODEL模型CLass。包含package的ClassName
OPERATION_TYPE	过滤操作类型	*****	过滤的类型： READ 查询附加条件 CREATE 创建，无权限会抛异常 PRINGT 打印 查询附加条件 DELETE 删除 无权限件会抛异常 UPDATE 更新 无权限会抛异常
RELATION_ID	关系ID	*****	关联用户ID、部门ID、群组ID、岗位ID、角色ID
RELATION_TYPE	关系类型	***	对应关系的类型包含： user、group、department、post、role

当用户执行增加、删除、更改的时候： 先按照过滤出来的当前规则列表组装进行查询对比，如果有权限则放行，否则抛出异常。

当用户执行查询的时候，将过滤出来的规则组装附加到查询的where条件中。

说明:

1 当用户没有规则条件的时候，默认为放行，即不做权限条件的约束。
2 当用户设置了规则条件的时候。 即约束了用户的权限范围，如果需要进一步放大权限范围，需要添加规则条件
3 多个规则条件是 或者的关系。但是与 用户的查询条件是 并且的关系。

示例:

```
testTableService
.queryUnDeleted()//未删除条件
.queryOrgId()//本用户单位ID
.dataFilter(DataFilterType.READ)//过滤查询规则
.findAll("id in ?1",Lists.newArrayList("1","2","3","4"))//用户查询条件
```