文章
Nicky Zhu · 二月 4, 2021 阅读大约需 7 分钟

案例: 建立只能使用SQL的用户

上一篇：
IRIS中的权限管理

在上一篇文章中，我们介绍了IRIS中的权限控制体系。在本文中我们将以一个常见的实施需求为例介绍如何使用IRIS的权限配置出一个只能使用SQL的用户。

需求的分解

和所有用户需求一样，当用户提出一个需求时，除其语义显式的含义之外，还需分析其是否具有没有明确说明的含义。
对于一个只能使用SQL的用户这样一个需求，即应当结合平台的特征分解成为功能需求：
具有一个合法，可通过用户名和密码使用IRIS的用户
该用户的数据库权限
- 确认项：可以使用SQL访问所有数据库还是某几个特定的数据库？
该用户的SQL权限
- 确认项：对于特定的数据库，是否可以执行所有的DDL？
- 确认项：对于特定的数据库，是否对每一张表都可以执行Select、Update等所有的DML
该用户的程序权限
- 确认项：用户是否可以通过Portal登录并管理IRIS？

如上所示，一个只能使用SQL的用户，这样一个看似简单的需求，如果需要与用户获得一致的理解并付诸实施，则需要将其分解，对于没有在用户需求中明确约定的部分，应作为待确认项与用户确认。
另外，需要注意的是，IRIS作为一个数据平台，除了提供底层的数据库之外，也提供了Portal等Web应用程序便于使用Sharding、HA、Interoperability等平台功能，通过平台对外提供的Webservice、REST等web接口也受Web应用程序控制。因此，当需要设计权限体系时，不但需要考虑用户使用的数据库相关的权限，也需要考虑是否需要控制Web应用程序的权限。

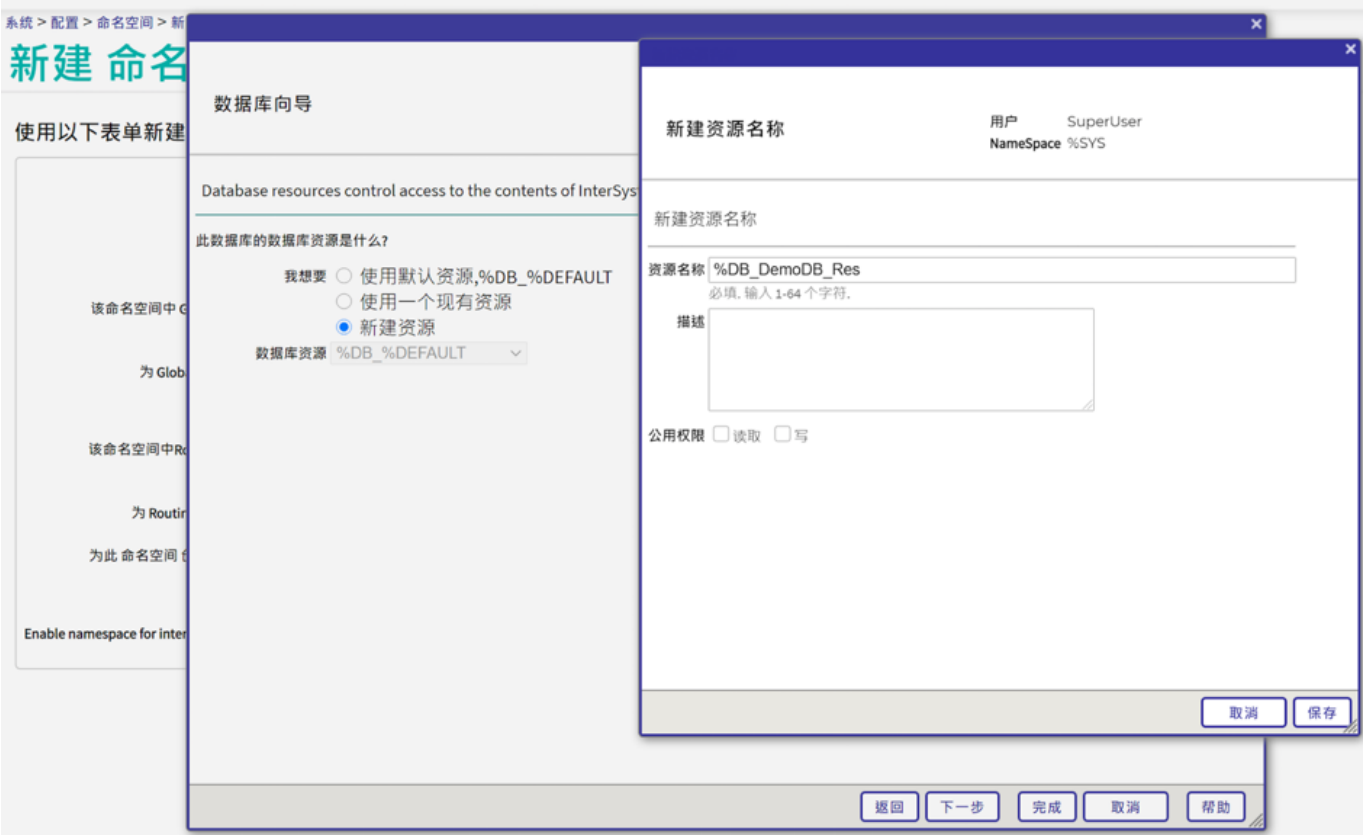
我们假设需要提供一个严格意义上的只能对某个库的某个Schema下的所有表具有只读权限的用户，即：
具有一个合法，可通过用户名和密码使用IRIS的用户
该用户的数据库权限
- 该用户只能使用DemoSpace命名空间下的数据库
该用户的SQL权限
- 该用户不能执行任何DDL
- 该用户只能对DemoSchema下的所有表执行Select语句
该用户的程序权限
- 该用户不能登录Portal，不能执行Portal中提供的任何管理功能

对数据库的配置

在IRIS中创建数据库时，默认的行为是引用%DB%DEFAULT这个资源，并引用%DB%DEFAULT角色（注意，平台中有一个叫做%DB%DEFAULT的资源，同时还有一个叫做%DB%DEFAULT的角色。%DB%DEFAULT角色通过%DB%DEFAULT资源获得默认的数据库访问权限）。如果直接使用%DB%DEFAULT角色或%DB%DEFAULT资源，都有可能影响到之前以默认配置创建的数据库，因此，在需要细粒度控制访问权限时，往往需要自定义资源和角色实现。

创建资源%DBDemoDBRes

资源可在创建数据库的同时创建，也可以在使用默认资源创建数据库之后，再给数据库指派其他资源。本例中，我们在创建命名空间DEMOSPACE的同时创建数据库DEMODB并创建资源。



创建角色DemoDBReadRole

在创建角色之后，即可为其分配资源。根据资源类型的不同，对资源的操作可以有读、写和使用三种权限。对于数据库引用的资源，是读权限和写权限。在本例中，我们需要创建的是只读用户，因此，资源权限分配读权限即可，不用赋写权限。

创建用户DemoUser并分配角色

通过Portal创建用户之后，即可给用户分配角色。在本例中，需要给这个用户分配之前创建的DemoDBReadRole角色。

在经过上述配置之后，用户DemoUser即已拥有对命名空间DEMOSPACE中数据库的访问权限。

此时，用户对数据库拥有读权限，但并没有对表执行查询或建表的权限，如果尝试create table，则会得到如下的权限错误信息：

为继续实验，我们通过Portal执行这SQL语句先创建DemoSchema.Persons这张表

为角色DemoDBReadRole分配SQL权限

由于用户具有的权限不足，无法执行SQL操作，因此我们需要对该用户的角色赋予对应的权限（或直接给用户赋权，但平台用户较多时，考虑到用户管理的成本，并不推荐这样做）。可以采用如下手段进行SQL的授权

通过Portal授权

在Portal的用户管理和角色管理功能中，均可指定要授予的SQL权限。
SQL特权栏用于对DDL进行数据库级的授权，例如对DEMOSPACE命名空间下的数据库分配建表、修改表、建视图等DDL操作。在本例中，用户不具有这些权限，因此不对该用户对应的角色授予这些权限。

在SQL表，SQL视图和SQL过程栏中，则是分别对表、视图和存储过程授予查询、执行等权限。在本例中，用户需要对DemoSchema这个Schema下的表拥有select查询权限，即可通过对SQL表授权进行

授权后该角色的SQL表权限如下

此时通过SQL工具已可执行查询

使用Portal授权时是针对单个的表、视图或存储过程进行。在上例中，我们单独对表DemoSchema.Persons进行了授权，如果我们再建立一张DemoSchema.Employee表，当前的角色和用户并不能自动获得读取其数据的权限。

通过SQL授权

超级管理员或拥有SQL授权权限的用户可以通过SQL的GRANT语句对数据库对象（包括库，函数，表，视图和存储过程等）进行授权，SQL GRANT语句的语法和使用详见GRANT，此处不再赘述。
在上面的例子中我们建立了表DemoSchema.Persons，建表的同时建立的Schema DemoSchema。假如现在我们希望对授权进行简化，使角色DemoDBReadRole可以直接获得Schema下所有表的读权限，则可以用如下的SQL：

```
GRANT SELECT ON SCHEMA DemoSchema to DemoDBReadRole
```

执行成功后再查看DemoDBReadRole的SQL权限，会发现：

即这个角色已经拥有了Schema级的授权，因此，对整个Schema下的所有表都拥有权限。之后在Schema中如果建立了新的表，则这个角色会自动拥有这些表的读权限。

限制用户登录和使用Portal

上例配置的用户可以用于登录IRIS的Portal，但由于没有任何系统功能的权限，不能执行操作。

IRIS的Web应用程序在创建时默认并不需要额外的资源去访问，这意味着所有合法用户都能登入这个Web应用，但由于支撑应用的后台程序和数据是受到资源的保护的，能登入的用户不一定具有运行程序、访问数据的权限，正如我们建立的用户DemoUser可以登录Portal，但没有功能菜单可用。
如果需要进一步限制用户的行为，禁止其登录，则还需要对应用权限进行控制。
如我们在上一篇文章：IRIS中的权限管理 中所述，Portal是平台提供的Web应用程序，是通过Web应用的权限控制可访问性，因此，需要修改应用的资源要求。
通过菜单: 系统管理 > 安全 > 应用程序 > Web应用程序
可以访问当前系统提供的Web应用列表，其中的/csp/sys即为系统管理门户Portal

其中必要的资源一栏即为该应用的资源需求，默认为空，即访问该应用不需要特定资源，只要是合法用户即可。我们可以为该应用指定所需资源，例如%Development，即只有具有%Development资源的角色及其对应的用户才能够访问该程序。保存设置后，再尝试以用户DemoUser登录Portal，结果是

除非我们为DemoUser引用的角色DemoDBReadRole分配资源%Development，该用户都不能登录。
当然，超级管理员由于拥有所有权限，不受这个设置的影响。

总结

通过上述实验，我们创建了一个用户，只能使用SQL连入并查询指定Schema下的表。希望通过这个实验，大家能够掌握IRIS权限管理的基本元素：

- 用户，角色，权限，资源和许可构成的权限控制体系
- 数据库、SQL和应用程序都是权限管理的对象

大家在实际项目中可以根据最终用户的实际需要，灵活应用这些概念，构建满足需求的权限配置。

上一篇：
IRIS中的权限管理

#安全 #数据库 #新手 #访问控制 #系统管理 #访问控制 #身份认证 #InterSystems IRIS #文档

源URL:
https://cn.community.intersystems.com/post/%E6%A1%88%E4%BE%8B-%E5%BB%BA%E7%AB%8B%E5%8F%AA%E8%83%BD%E4%BD%BF%E7%94%A8sql%E7%9A%84%E7%94%A8%E6%88%B7