文章
姚 鑫 · 四月 8, 2021 阅读大约需 分钟

第二十章 用户、角色和权限

第二十章 用户、角色和权限

InterSystems IRIS®具有系统级安全性以及一组与sql相关的额外安全特性。
在数据库级操作之外，InterSystems SQL安全性提供了额外级别的安全功能。
SQL和系统级安全性之间的一些关键区别是：
- SQL操作比系统级操作更细粒度。可以为表、视图和存储过程定义特权。
- SQL权限既可以授予用户，也可以授予角色。
系统级权限只分配给角色。
- 持有SQL特权会隐式授予执行SQL操作所需要的相关系统特权。
(相反，系统级特权并不意味着表级特权。)

InterSystems SQL在InterSystems IRIS数据平台上对ODBC、JDBC、Dynamic SQL和SQL Shell接口进行权限检查。
嵌入式SQL语句不执行特权检查；
假定使用嵌入式SQL的应用程序在使用嵌入式SQL语句之前会检查特权。

SQL权限和系统权限

要通过特定于SQL的机制操作表或其他SQL实体，用户必须具有适当的SQL权限。
系统级权限不足。
用户可以直接被授予SQL权限，也可以属于具有SQL权限的角色。

注 意：角色是由SQL和系统级安全共享的：单个角色可以包括系统和SQ权限。

下面的例子，以Windows机器上的InterSystems IRIS为例：
- 在用户名称空间中有一个名为User.MyPerson的持久化类。
这个类被投影到SQL中作为SQLUser.MyPerson表。
- 有一个名为Test的用户，他不属于任何角色(因此没有系统权限)，并且拥有SQLUser.MyPerson表的所有权限(没有其他SQL权限)。
- 还有第二个用户，名为test2。此用户被分配给以下角色：%DB_USER(因此可以读取或写入用户数据库上的数据)；%SQL(因此可以通过%Service_BINDINGS服务访问SQL)；并且通过自定义角色具有使用控制台和%Development的权限。

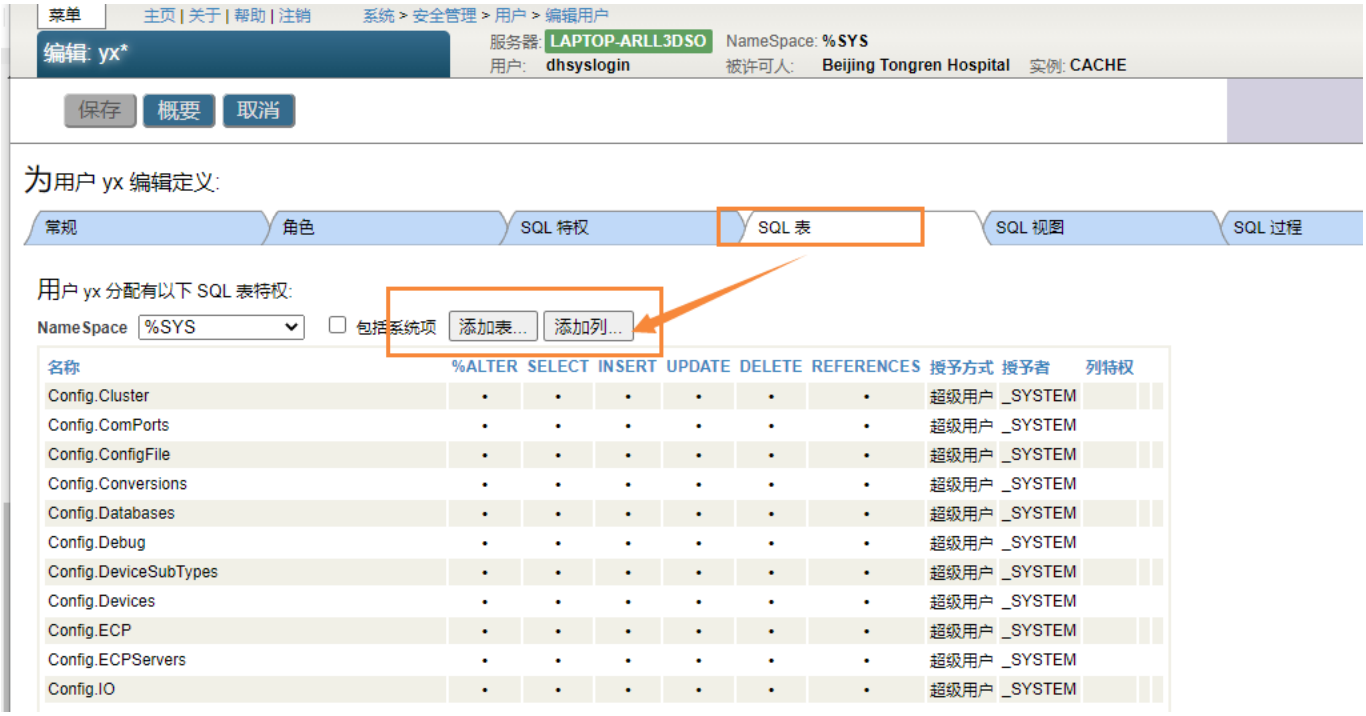
如果测试用户尝试通过任何特定于SQL的机制(如使用ODBC的机制)在SQLUser.MyPerson表中读取或写入数据，则尝试将成功。这是因为InterSystems IRIS使测试用户成为%SQL角色(包括%SERVICE_SQL：USE权限)和%DB_USER角色的成员，因此该用户具有建立连接所需要的权限；这在连接生成审核事件(如%SYSTEM/%Login/Login event)中可见。(如果测试用户尝试使用终端对象机制，则这些尝试将失败，因为用户对这些机制没有足够的权限。)

如果Test2用户尝试通过任何特定于SQL的机制(如使用ODBC的机制)在SQLUser.MyPerson表中读取或写入数据，则该尝试将失败，因为该用户没有足够的权限访问该表。(如果Test2用户尝试使用对象机制查询终端中的相同数据，则尝试成功-因为该用户有足够的权限进行这种类型的连接。)

用户

InterSystems SQL用户与为InterSystems安全性定义的用户相同。可以使用SQL命令或管理门户定义用户。
- 在SQL中，可以使用CREATE USER语句创建用户。这只会创建一个用户名和用户密码。新创建的用户没有角色。必须使用GRANT语句为用户分配权限和角色。可以使用ALTER USER和DROP USER语句修改有用户定义。
- 在管理门户中选择System Administration(系统管理)，选择Security(安全性)，然后选择Users(用户)。单击页面顶部的Create New User(创建新用户)按钮。这会将带到编辑用户页，可以在其中指定用户名、用户口令和其他参数。创建用户后，其他选项卡即可用，可以在其中指定用户拥有哪些角色、用户拥有哪些常规SQL权限、用户拥有哪些表级权限、哪些视图可用以及可以执行哪些存储过程。

如果用户具有SQL表权限或一般SQL权限，则在用户的角色选项卡上授予或撤消的角色不会影响用户通过基于SQL的服务(如ODBC)对表的访问。这是因为，在基于SQL的服务中，基于表的权限优先于基于资源的权限。



可以使用%Library.SQLCatalogPriv类查询列出：
- 所有用户SQLUsers()
- 授予指定用户SQLUserPrivs("username")的所有权限
- 授予指定用户SQLUserSysPrivs("username")的所有系统权限
- 授予指定用户SQLUserRole("username")的所有角色

以下示例列出了授予当前用户的权限：

```
/// d ##class(PHA.TEST.SQL).Sqluser2()
ClassMethod Sqluser2()
{
    SET statemt=##class(%SQL.Statement).%New()
    SET cqStatus=statemt.%PrepareClassQuery("%Library.SQLCatalogPriv","SQLUserPrivs")
    IF cqStatus'=1 {WRITE "%PrepareClassQuery failed:" DO $System.Status.DisplayError
(cqStatus) QUIT}
    SET rset=statemt.%Execute($USERNAME)
    WRITE "Privileges for ",$USERNAME
    DO rset.%Display()
}
```

结构形式的用户名

在某些情况下，用户名可以隐式用作SQL模式名称。如果用户名包含SQL标识符中禁止的字符，这可能会带来问题。例如，在域配置中，用户名包含“@”字符。

根据分隔标识符配置参数的设置，InterSystems IRIS会以不同的方式处理此情况：
- 如果启用了分隔标识符的使用，则不会进行特殊处理。
- 如果禁用分隔标识符的使用，则会从用户名中删除所有禁用字符，以形成模式名称。例如，用户名“Documentation @intersystems.com”将成为模式名称“Documentationintersystemscom”。

这不会影响SQL CURRENT_USER函数返回的值。它始终与$USERNAME相同。

角色

将SQL权限分配给用户或角色。角色使能够为多个用户设置相同的权限。角色由SQL和系统级安全共享：单个角色可以同时包括系统权限和SQL权限。

管理门户、系统管理、安全性角色页提供了InterSystems IRIS实例的角色定义列表。要查看或更改特定角色的详细信息，请选择该角色的名称链接。在出现的编辑角色页面上，有关于角色权限以及哪些用户或角色拥有该权限的信息。

常规选项卡列出角色对系统间安全资源的权限。如果角色仅拥有SQL权限，则一般信息选项卡的资源表会将该角色的权限列为“未定义”。

SQL权限选项卡列出了角色对InterSystems SQL资源的权限，其中命名空间的拉列表允许查看每个命名空间的资源。因为权限是按名称空间列出的，所以在特定名称空间中没有权限的角色的列表显示为“None”。

注：应该使用角色定义权限，并将特定用户与这些角色相关联。这有两个原因：
1. 与检查单个用户条目相比，SQL引擎通过检查相对较小的角色数据库来确定权限级别的效率要高得多。
2. 与具有多个单独用户设置的系统相比，使用少量角色集管理系统要容易得多。

例如，可以定义具有特定访问权限的名为“ACCOUNTING”的角色。随着 Accounting Department的发展，可以定义新用户并将其与会计角色相关联。如果需修改ccounting权限，只需修改一次，系统会自动覆盖Accounting Department的所有成员。

一个角色可以担任其他角色。例如，会计角色可以拥有BILLINGCLERK角色。被授予会计角色的用户将同时拥有会计角色和BILLINGCLERK角色的权限。

还可以使用以下SQL命令定义用户和角色：CREATE USER、CREATE ROLE、ALTER USER、GRANT、DROP USER和DROP ROLE。

可以使用%Library.SQLCatalogPriv类查询列出：
- 所有角色SQLRoles()
- 授予指定角色SQLRolePrivileges(“Rolename”)的所有权限
- 授予指定角色SQLRoleUser(“Rolename”)的所有角色或用户
- 授予指定用户SQLUserRole(“username”)的所有角色

SQL权限

将SQL权限分配给用户或角色。角色使能够为多个用户设置相同的权限。

InterSystems SQL支持两种类型的权限：管理权限和对象权限。

- 管理权限是特定于命名空间的。

管理权限包括创建、更改和删除对象类型，例如创建表所需的CREATE_TABLE权限。不仅需要%ALTER_TABLE特权来更改表，还需要%ALTER_TABLE特权来创建或删除索引、创建或删除触发器以及运行TUNE TABLE。

管理权限还包括%NOCHECK、%NOINDEX、%NOLOCK和%NOTRIGGER，它们确定用户在执行INSERT、UPDATE、INSERT或UPDATE或DELETE时是否可以应用相应的关键字限制。用户需分配%NOTRIGGER管理权限才能执行TRUNCATE TABLE。

- 对象权限特定于表、视图或存储过程。它们指定对特定命名SQL对象的访问类型(在SQL意义上：表、视图、列或存储过程)。如果用户是SQL对象的所有者(创建者)，则会自动向该用户授予该对象的所有权限。

表级对象权限提供对表或视图的所有列中的数据的访问(%ALTER、DELETE、SELECT、INSERT、UPDATE、EXECUTE、REFERENCES)，包括当前存在的列和任何后续添加的列。

列级对象权限仅提供对表或视图的指定列中的数据的访问权。不需为具有系统定义的值(如RowID和Identity)的列分配列级权限。

存储过程对象权限允许将过程的EXECUTE权限分配给指定的用户或角色。

授予SQL权限

可以通过以下方式授予权限：
- 使用管理门户。从系统管理中选择安全性，然后选择用户或角色。选择所需用户或角色，然后选择相应的选项卡：管理权限的SQL权限、对象权限的SQL表、SQL视图或SQL过程。
- 在SQL中，使用GRANT命令向指定用户或角色(或用户或角色列表)授予特定管理权限或对象权限。可以使用REVOKE命令删除权限。
- 在ObjectScript中，使用$SYSTEM.SQL.Security.GrantPrivileve()方法将特定对象权限授予指定用户(或用户列表)。

列出SQL权限

- 使用管理门户。从系统管理中选择安全性，然后选择用户或角色。选择所需用户或角色，然后选择相应的选项卡：管理权限的SQL权限、对象权限的SQL表、SQL视图或SQL过程。
- 在SQL中，使用%CHECKPRIV命令确定当前用户是否具有特定的管理或对象权限。
- 在ObjectScript中，使用$SYSTEM.SQL.Security.CheckPrivileve()方法确定指定用户是否具有特定的对象权限。

审核权限错误

当InterSystems IRIS进程调用用户没有特权的SQL语句时，操作将失败，并生成SQLCODE-99错误。启用审核事件%SYSTEM/%SQL/PrivilegeFailure时，将在Audit数据库中为遇到的每个SQLCODE-99错误放置一条记录。默认情况下，此审核数据库选项处于禁用状态。

#SQL #Caché #InterSystems IRIS #InterSystems IRIS for Health

源 URL: https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%8C%E5%8D%81%E7%AB%A0-%E7%94%A8%E6%88%B7%E3%80%81%E8%A7%92%E8%89%B2%E5%92%8C%E6%9D%83%E9%99%90