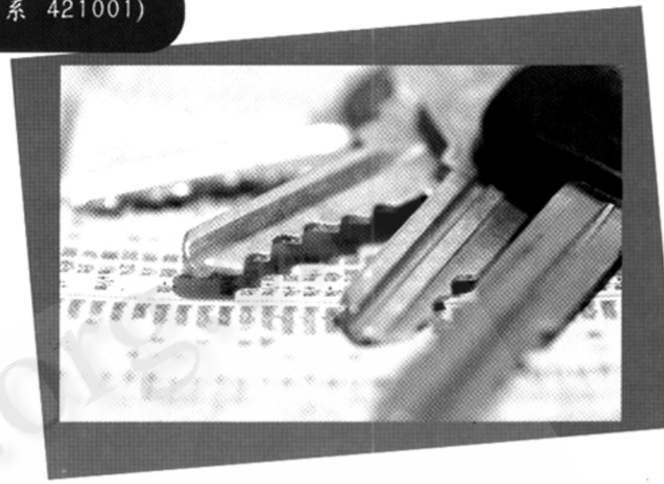
SQL Server 的数据库保护策略

王杰文（湖南 衡阳 南华大学计算机科学系 421001）

摘要：本文从安全保护和完整保护两个方面，简明扼要地阐述了SQL Server在数据库自我保护方面所采取的策略，并主要就完整性控制策略的应用举例作了说明。

关键词：SQL Server　安全保护　完整保护



数据库保护是数据库系统设计的重要课题，也是数据库设计的主要内容。数据库保护一方面是要保护数据库中数据不被非法访问和非法操作与窃密，另一方面是要维护数据库中数据的正确、真实、有效和随时可用，前者属于安全保护范畴，后者属于完整保护范畴。因为掌握数据库的保护策略是设计和管理数据库的关键，所以，笔者主要从数据库设计的角度来讨论SQL Server数据库的保护策略问题。

1　SQL Server 的安全保护策略

数据库安全保护的主要任务就是防止非法用户访问或合法用户越权访问数据库中的数据。另一方面，为了有效地打击对数据库的非法访问和越权访问，数据库系统往往还提供了对机密数据加密和对用户的操作审计跟踪的功能。

1.1　SQL Server 的用户身份验证策略

SQL Server使用服务器登录标识和数据库用户名来记录合法用户，并由系统管理员(SA)、数据库拥有者(DBO)恰当授权来限制合法用户在数据库中能从事的工作，最后由SA、DBO或数据库对象所有者授权来限制用户对数据库对象的访问范围和访问方式。

登录标识是服务器一级的，它是SQL Server服务器接收用户登录连接时识别用户的标识，登录标识信息存放在master数据库的syslogins系统表中。SQL Server提供了三种登录认证方式供SA选用：第一种是集成安全模式，在此模式下，SQL Server用windows NT的安全机制来验证用户的登录访问，即用户可以使用同一个用户名和口令登录到windows NT和SQL Server。因为Windows NT的域用户帐号以DOMAIN_username格式作为SQL Server登录标识自动加入到syslogins系统表中，所以当Windows NT用户请求连接SQL Server时，系统直接将其映射为SQL Server的缺省登录标识完成登录，如果Windows NT用户是Administrator，则映射为SA身份登录SQL Server服务器；第二种是标准安全模式，在此模式下，SQL Server将自行验证所有的登录连接请求，用户要想登录SQL Server，必须分别在Windows NT和SQL Server中有合法登录标识；第三种是混合安全模式，在此模式下，用户可以同时采用上面两种方式登录SQL Server。

数据库用户是有权在某个数据库中完成某些操作的用户，它是数据库一级的，即不同的数据库有不同的用户群，一个数据库中的用户不能访问另一个数据库中的数据。数据库用户的信息存储在每个数据库的sysusers系统表中。在SQL Server中创建一个数据库时，系统自动生成两个用户：DBO和guest，SA和DBO可以为已建立登录标识的用户创建数据库用户(其名可以和登录标识相同，也可以不同，建议使用相同的用户名，以便用户记忆)。

1.2　SQL Server 的用户权限管理

SQL Server和其他数据库系统一样，也是通过用户权限来控制用户对数据库的越权访问。根据用户的不同权限层次，可以将用户分为四类：系统管理员(SA)、数据库所有者(DBO)、数据库对象所有者和其他用户，每一层次的用户都自动拥有一定的权限，他可以被授予其他权限或将其拥有的权限授予其他用户。

系统管理员(SA)是SQL Server中具有最高权限的用户，系统对其操作没有任何限制。他具有创建数据库

(CREATE DATABASE)的权限，且可以将此权限授予其他用户；但是，应该注意，有些权限，如：封闭进程、关闭系统等权限只属于SA，且不能授予其他用户。

数据库所有者(DBO)在他所拥有的数据库内享有所有权限，他可以设置检查点(CHECKPOINT)、删除他拥有的数据库(DROP DATABASE)、装载数据库和事务(LOAD DATABASE/TRANSACTION)等；他可以授予(GRANT)和回收(REVOKE)其他用户在其拥有的数据库中创建表(TABLE)、缺省(DEFAULT)、规则(RULE)、过程(PROCEDURE)和视图(VIEW)等数据库对象的权限及卸载数据库和事务的权限。

数据库对象所有者拥有他所创建的对象的所有权限，他可以在他所建的表上建立索引(INDEX)和触发器(TRIGGER)，可以修改(ALTER)他所建的表，可以删除他所建的表、视图、索引、规则、缺省、过程和触发器等。他还可以将其所拥有的对表的查询(SELECT)、更新(UPDATE)、插入(INSERT)、删除(DELETE)和参照(REFERENCE)等权限授予其他用户，将他所创建的过程的执行权限授予其他用户。

1.3 SQL Server 的角色管理

角色是一定权限的代表，用户只要成为某个角色就拥有该角色代表的所有权限。角色是简化用户管理的重要策略，目前大部分系统都采用了，SQL Server安装成功后，系统预定义的SQL Server(系统管理员)角色有7种，常用的有：

(1) sysadmin (System Administrators) 角色可以用SQL Server 执行任何任务

(2) dbcreator (Database Creators) 角色可以创建和更改个人数据库

(3) securityadmin(Security Administrator) 角色可以管理服务器的用户及注册

(4) serveradmin (Server Administrator) 角色可以配置服务器设置

创建新的数据库时，系统预定义的数据库角色有9种，常用的有：

(1) db_owner (Database Owner) 数据库所有者角色可以对该数据库执行任何操作

(2) db_accessadmin (Database Access Administrator) 角色可以管理数据库用户列表

(3) db_datareader (Database data Reader) 角色可以查看数据库中的任何数据

(4) db_datawriter (Database Data Writer) 角色可以插入、修改、删除所有表中的数据

(5) db_ddladmin (Database Data Definition Language Administrator) 角色可以创建、修改、删除数据库的任何对象

(6) db_securityadmin (Database Security Administrator) 角色可以管理数据库用户和角色以及库中的语句和对象权限。

此外还有两种很有用的数据库角色：第一个是“公共角色”(public)，每个数据库均有一个公共角色，且每一个数据库用户都自动属于这个角色，所以对这个角色分配的任何权限都将对每一个数据库用户生效。教学环境下往往只要恰当设置public角色的权限就可以了，因为所有学生所拥有的权限一样。第二个是“应用程序角色”，它被分配给一个应用程序，该程序只能访问其必须访问的数据，被赋予这种角色的用户只能通过程序访问数据，用户一旦脱离这个程序就无法访问数据了。除以上角色外，管理员还可以自定义数据库角色。

2 SQL Server 提供的完整性约束措施

完整保护可以从对数据实行完整性约束和维护事务特性(原子性、持久性、可串行性和隔离性)两个途径来实现。

2.1 SQL Server 提供的完整性约束措施

完整性约束用于保证数据库中数据的正确性、真实性和有效性。SQL Server中，通过各种约束、缺省、规则和触发器来保证数据的完整性。

约束是SQL Server强制实行的应用规则，使用约束可以限制用户存放到表中数据的格式和可能值。约束一般在CREATE TABLE语句中申明，但约束又独立于表结构，所以可以用ALTER TABLE语句来添加或删除约束。必须注意，在删除一个表时，表中申明的约束也被一同删除。SQL Server数据库中在表上可以定义DEFAULT、CHECK、PRIMARY KEY、UNIQUE与FOREIGN KEY等约束。比如，限制学生表(student)中学生年龄(age)在15到25之间，可在定义student表时用CHECK约束来实现：CHECK(age BETWEEN 15 AND 25)。

缺省是一种数据库对象，其作用与DEFAULT约束类似，有相应权限的用户可以创建或删除缺省，并建立(或解除)缺省与列或用户定义数据类型的关联。一个缺省可以与多个列建立关联，而一个DEFAULT约束只能作用于一个表的一个列，所以，当我们要在多个列上定义同一个

约束时应首选缺省对象。如某个关系中有多个数值型列，我们希望它们的缺省值均取0，则我们可以先定义一个缺省(CREATE DEFAULT ZERO AS 0)，然后用存储过程sp-bindefault将缺省(ZREO)关联到表中数值列上。

规则也是一种数据库对象，其作用与CHECK约束类似，但一个规则可以用于限制插入到多个列中的值且可以单独处理。有相应权限的用户可以创建或删除规则，并建立(或解除)规则与列或用户定义数据类型的关联。例如：通过ISQL提交下列语句：

```
CREATE RULE id_chk AS @id BETWEEN 0 AND 10000
GO
CREATE TABLE employee
(e_id int PRIMARY KEY,
  name char(10),
  address char(50) )
GO
sp_bindrule id_chk,'employee.e_id'
GO
```

将建立一个规则id_chk和一个表employee，并用存储过程sp_bindrule将规则关联到列employee表的e_id上，从而使该列的取值限制在0--1000范围内。显然，如果在关系模式中只有employee表中e_id一列要受规则id_chk限制，则应优先采用CHECK约束。必须注意，缺省和规则只能由创建者将其关联到创建者自己拥有的表列或数据类型上。

触发器是一种特殊的存储过程，使用它可以实施更为复杂的数据完整性约束。SQL Server中，触发器建立在表一级，每个表最多可以建立三个触发器(分别对应INSERT、UPDATE和DELETE操作)。触发器不带参数，不能被直接调用，只能由系统自动激活。触发器在结构上独立于表，但又隶属于表，所以可被单独删除，但在删除表时也将一同删除关联的触发器。例如，有一个工资关系salary(namel,basepay,bonus)，其中namel列的值应该是对employee表中name列值的引用，但这是一个非主码的引用完整性约束，现今系统不支持。通过如下触发器：

```
CREATE TRIGGER refe_constraint
ON salary FOR INSERT
AS IF NOT EXISTS
    (SELECT* FROM employee)
    WHERE name=(SELECT namel FROM inserted))
BEGIN
  DELETE FROM salary
    WHERE namel=(SELECT namel FROM inserted)
    PRINT 'Not Found employee!'
END
```

当用户在工资表中插入一个雇员表中不存在的人名时，系统将给予提示并自动删除该记录，从而确保了引用的完整性。

2.2 SQL Server 的事务管理策略

并发控制和恢复是事务管理的两种基本策略，并发控制用以维持事务并发执行时的可串行性，而恢复用以维持事务的原子性、持久性和隔离性。SQL Server中有两类事务，一类是显式事务，它是由用户使用事务定义语句建立的，每个事务必须以BEGIN TRANSACTION开始，COMMIT TRANSACTION结束。没有被显式地括在BEGIN TRANSACTION和ROLLBACK TRANSACTION/COMMIT TEANSACTION之间的每一个更新语句(INSERT、DELETE、UPDATE)系统都将隐式地作为一个事务来处理。

SQL Server中事务的并发控制由其内核隐含提供的动态锁定功能来实现，用户不必操心。QL Server的恢复机制有两种，一种是系统自动执行的恢复，因为SQL Server能自动管理和维护所有的数据库更改事务，并在修改数据前把事务写入日志，所以在SQL Server每次启动期间，系统就能自动地根据日志将因断电等因素引起系统崩溃之前所提交的事务都写到数据库设备上(前滚操作)，而将未提交的事务删除(回滚)。应该注意，自动恢复不能被关闭。另一种是用户执行的恢复，这是指在系统出现故障时，由SA或DBO从数据库和日志备份中恢复(LOAD)系统或用户数据库，这种恢复以数据备份为基础，是一种备份恢复策略。应该注意，数据库的备份(DUMP)权限可由DBO授予其他用户，但恢复操作的权限不能授予他人。

3 结束语

数据库是共享数据的集合，对其中数据实施保护是数据库系统得以正常运行的关键。数据库保护涉及多方面内容，主要包括计算机设备的安全、人员的安全素质和责任、日常管理以及DBMS的自我保护。本文旨在对SQL Server系统提供的自我保护策略作个归纳，具体细节读者可参阅相关书籍。■