[**db2 创建用户及授权**](https://www.cnblogs.com/BradMiller/p/3198180.html)

1、创建系统用户dbuser/ehong隶属于db2users

2、C:\Program Files\IBM\SQLLIB\BIN>db2 connect to AKZXTEST  
数据库连接信息  
 数据库服务器         = DB2/NT 9.7.3  
 SQL 授权标识         = ADMINIST...  
 本地数据库别名       = AKZXTEST

3、给用户授权

授予连接权限  
C:\Program Files\IBM\SQLLIB\BIN>db2 grant **connect** on database to user dbuser  
DB20000I  SQL 命令成功完成。

这里可以授予的数据库级特权有：

CREATETAB： 用户可以在数据库中创建表。

BINDADD： 用户可以使用 BIND 命令在数据库中创建包。

CONNECT： 用户可以连接数据库。

CREATE\_NOT\_FENCED： 用户可以创建 unfenced 用户定义函数(UDF)。

IMPLICIT\_SCHEMA： 用户可以在数据库中隐式地创建模式，而不需要使用 CREATE SCHEMA 命令。

LOAD： 用户可以将数据装载进表中。

QUIESCE\_CONNECT： 用户可以访问处于静默(quiesced)状态的数据库。

CREATE\_EXTERNAL\_ROUTINE： 用户可以创建供应用程序和数据库的其他用户使用的过程。

secadm：安全性管理员

Dbadm：DBADM 用户对一个数据库有几乎完全的控制能力。DBADM 用户不能执行某些维护或管理任务

4、测试连接  
C:\Program Files\IBM\SQLLIB\BIN>db2 connect to AKZXTEST user dbuser  using ehong  
数据库连接信息  
 数据库服务器         = DB2/NT 9.7.3  
 SQL 授权标识         = DBUSER  
 本地数据库别名       = AKZXTEST

###############################################

**db2**数据库特权大体上分成两类：数据库级特权(针对数据库中的所有对象)和对象级特权(与特定的对象相关联)。

1、DBADM

 用户对一个数据库有几乎完全的控制能力。DBADM 用户不能执行某些维护或管理任务，比如：

\* drop database  
\* drop/create tablespace  
\* backup/restore database  
\* update db cfg for database db name

但是，他们可以执行以下任务：

\* db2 create/drop table  
\* db2 grant/revoke（任何特权）  
\* db2 runstats（任何表）

DBADM 用户还被自动地授予对数据库对象及其内容的所有特权。因为 DBADM 权限是一个数据库级权限，所以它可以被分配给用户和用户组。以下命令演示授予 DBADM 权限的不同方法。

\* db2 create database test  
   这个命令将数据库 test 上的 DBADM 权限隐式地授予发出此命令的用户。  
\* db2 connect to sample  
   db2 grant dbadm on database to user tst1  
   这个命令只能由 SYSADM 用户发出；它向用户 tst1 授予示例数据库上的 DBADM 权限。注意，在授予 DBADM 权限之前，发出这个命令的用户必须连接到示例数据库。  
\* db2 grant dbadm on database to group db2grp1  
   这个命令将 DBADM 权限授予 db2grp1 组中的每个用户。同样，只有 SYSADM 用户能够发出这个命令。

2、 LOAD 权限  
LOAD 权限是一个数据库级权限，所以它可以被分配给用户和用户组。顾名思义，LOAD 权限允许用户对表发出 LOAD 命令。当用大量数据填充表时，LOAD 命令通常用来替代插入或导入命令，它的速度更快。根据您希望执行的 LOAD 操作类型，仅仅拥有 LOAD 权限可能还不够。可能还需要表上的特定特权。  
拥有 LOAD 权限的用户可以运行以下命令：

\* db2 quiesce tablespaces for table  
\* db2 list tablespaces  
\* db2 runstats（任何表）  
\* db2 load insert（必须有表上的插入特权）  
\* db2 load restart/terminate after load insert（必须有表上的插入特权）  
\* db2 load replace（必须有表上的插入和删除特权）  
\* db2 load restart/terminate after load replace（必须有表上的插入和删除特权）

只有拥有 SYSADM 或 DBADM 权限的用户能够对用户或用户组授予或撤消 LOAD 权限。以下示例演示 LOAD 权限如何允许我们的用户使用 LOAD 命令将数据装载进 sales 表中。假设已经发出了命令 db2 connect to sample。

\* db2 grant load on database to user tst1  
   db2 grant insert on table sales to user tst1  
   有了 LOAD 权限和插入特权，tst1 就可以对 sales 表发出 LOAD INSERT 或 LOAD RESTART，或者在 LOAD INSERT 之后发出 TERMINATE。  
\* db2 grant load on database to group grp1  
   db2 grant delete on table sales to group grp1  
   db2 grant insert on table sales to group grp1  
   有了 LOAD 权限以及删除和插入特权，grp1 的任何成员就可以对 sales 表发出 LOAD REPLACE 或 LOAD RESTART，或者在 LOAD REPLACE 之后发出 TERMINATE。

3.其他特权  
用户可以拥有的数据库级特权有：

\* CREATETAB： 用户可以在数据库中创建表。  
\* BINDADD： 用户可以使用 BIND 命令在数据库中创建包。  
\* CONNECT： 用户可以连接数据库。  
\* CREATE\_NOT\_FENCED： 用户可以创建 unfenced 用户定义函数（UDF）。  
\* IMPLICIT\_SCHEMA： 用户可以在数据库中隐式地创建模式，而不需要使用 CREATE SCHEMA 命令。  
\* LOAD： 用户可以将数据装载进表中。  
\* QUIESCE\_CONNECT： 用户可以访问处于静默（quiesced）状态的数据库。  
\* CREATE\_EXTERNAL\_ROUTINE： 用户可以创建供应用程序和数据库的其他用户使用的过程。

数据库对象 包括表、视图、索引、模式和包。幸运的是，大多数对象级特权的意义无需解释。下表总结了这些特权。  
特权名称 相关对象 描述  
CONTROL 表、视图、索引、包、别名、不同的类型、用户定义函数、序列 提供对对象的全部权限。拥有这种特权的用户还可以向其他用户授予或撤消对对象的特权。  
DELETE 表、视图 允许用户从对象中删除记录。  
INSERT 表、视图 允许用户通过 INSERT 或 IMPORT 命令将记录插入对象中。  
SELECT 表、视图 提供使用选择语句来查看对象内容的能力。  
UPDATE 表、视图 允许用户使用更新语句修改对象中的记录。  
ALTER 表 允许用户使用更改语句更改对象定义。  
INDEX 表 允许用户使用创建索引语句在对象上创建索引。  
REFERENCES 表 提供在对象上创建或删除外键约束的能力。  
BIND 包 允许用户重新绑定现有的包。  
EXECUTE 包、过程、函数、方法 允许用户执行包和例程。  
ALTERIN 模式 允许用户修改模式中的对象定义。  
CREATEIN 模式 允许用户在模式中创建对象。  
DROPIN 模式 允许用户删除模式中的对象。  
关于对象级特权的信息存储在系统编目视图中。视图名称是 syscat.tabauth、syscat.colauth、syscat.indexauth、syscat.schemaauth、syscat.routineauth 和 syscat.packageauth。  
显式特权  
可以使用 GRANT 和 REVOKE 命令显式地 对用户或组授予或撤消特权。我们来看看如何在各种对象上使用这些命令。  
作为拥有 Administrator 权限的用户登录 Windows，打开两个 DB2 命令窗口。在这两个窗口中，确保将 db2instance 变量设置为 DB2！  
在第一个窗口中发出以下命»¤：

db2 connect to sample

现在，在第二个窗口中发出以下命令：

db2 connect to sample user test1 using password  
请记住，第一个窗口中的命令是由一个拥有 SYSADM 权限的用户发出的。第二个窗口中的命令是由 tst1 发出的，这个用户对示例数据库没有特殊的权限或特权。注意，与示例数据库中的表相关联的模式名是发出 db2sampl 命令的用户的名称。在这些示例中，这个用户是 GMILNE。  
现在，在第二个窗口中发出以下命令：

db2 select \* from gmilne.org

应该会看到以下响应：

SQL0551N   "TEST1" does not have the privilege to perform operation "SELECT"   
on object "GMILNE.ORG".

为了纠正这种状况，在第一个窗口中发出以下命令：

db2 grant select on table gmilne.org to user test1

现在，前面的命令就会成功！接下来，在第二个窗口中发出一个更复杂的命令：

db2 insert into gmilne.org values (100, 'Tutorial', 1, 'Eastern', 'Toronto')

同样会看到错误消息：

SQL0551N   "TEST1" does not have the privilege to perform operation   "INSERT"   
on object "GMILNE.ORG"

所以，在第一个窗口中输入以下命令：

db2 grant insert on table gmilne.org to group db2grp1

原来失败的 INSERT 命令现在应该会成功完成，因为 test1 是 db2grp1 组的成员。  
现在，在第二个窗口中输入以下命令：

db2 drop table gmilne.emp\_photo

同样会看到错误消息：

SQL0551N   "TEST1" does not have the privilege to perform operation "DROP TABLE"  
on object "GMILNE.EMP\_PHOTO".

所以，我们要授予这个特权。在第一个窗口中输入以下命令：

db2 grant dropin on schema gmilne to all

DROP TABLE 命令现在应该会成功完成。  
既然已经完成了示例，就可以撤消刚才授予的特权。在第一个窗口中发出以下命令：

db2 revoke select on table gmilne.org from user test1  
db2 revoke insert on table gmilne.org from group db2grp1  
db2 revoke dropin on schema gmilne from all

注意，从组中撤消特权不一定会从这个组的所有成员撤消它。例如，以下命令可以用来从 db2grp1 撤消对 gmilne.org 表的所有特权（CONTROL 除外）：

db2 revoke all on table gmilne.org from group db2grp1

但是，test1 用户（他是 db2grp1 的成员）仍然拥有对这个表的选择特权，因为他或她是被直接授予这个特权的。  
隐式特权  
当发出某些命令时，DB2 可能会自动地授予特权，而不需要像前面看到的那样发出显式的 GRANT 语句。下表总结了会导致数据库管理程序隐式地授予特权的一些命令。注意，当删除创建的对象时，这些特性会隐式地撤消。但是，当显式地撤消更高级的特权时，不会撤消它们。  
发出的命令 授予的特权 被授予特权的用户  
CREATE TABLE mytable mytable 上的 CONTROL 发出命令的用户  
CREATE SCHEMA myschema myschema 上的 CREATEIN、ALTERIN 和 DROPIN，以及将这些特权授予其他用户的能力 发出命令的用户  
CREATE VIEW myview myview 上的 CONTROL（只有在用户拥有 myview 定义中引用的所有表和视图上的 CONTROL 特权的情况下） 发出命令的用户  
CREATE DATABASE mydb mydb 的系统编目表上的 SELECT，mydb 上的 IMPLICIT\_SCHEMA \* PUBLIC\*\*  
\*当用户创建数据库时，隐式地授予这个用户这个数据库上的 DBADM 权限。获得 DBADM 权限就会隐式地授予 CONNECT、CREATETAB、BINDADD、IMPLICIT\_SCHEMA 和 CREATE\_NOT\_FENCED 特权。即使撤消了 DBADM 权限，这个用户仍然会保留这些特权。  
\*\*PUBLIC 是一个特殊的 DB2 组，其中包括特定数据库的所有用户。与前面讨论过的其他组不同，PUBLIC 不必在操作系统级进行定义。在默认情况下，会向 PUBLIC 授予一些特权。例如，这个组自动接受数据库上的 CONNECT 特权和编目表上的 SELECT 特权。可以对 PUBLIC 组发出 GRANT 和 REVOKE 命令，比如：

db2 grant select on table sysibm.systables to public  
db2 revoke select on table sysibm.systables from public

间接特权  
当数据库管理器执行包 时，可以间接获得特权。包中包含一个或多个 SQL 语句，这些语句已经转换为 DB2 用来在内部执行它们的格式。换句话说，包中包含可执行格式的多个 SQL 语句。如果包中的所有语句都是静态的，那么用户只需要有包上的 EXECUTE 特权，就能够成功地执行包中的语句。  
例如，假设 db2package1 执行以下静态的 SQL 语句：

db2 select \* from org  
db2 insert into test values (1, 2, 3)

在这种情况下，拥有 db2package1 上的 EXECUTE 特权的用户会间接地获得 org 表上的 SELECT 特权和 test 表上的 INSERT 特权。

################## 关于 DB2 授权的研究 ###################################################

DB2 中有三种主要的安全机制，可以帮助 DBA 实现数据库安全计划：*身份验证（authentication）*、*授权（authorization）* 和*特权（privilege）*。

身份验证是用户在尝试访问 DB2 实例或数据库时遇到的第一种安全特性。DB2 身份验证与底层操作系统的安全特性紧密协作来检验用户 ID 和密码。DB2 还可以利用 Kerberos 这样的安全协议对用户进行身份验证。

授权决定用户和/或用户组可以执行的操作以及他们可以访问的数据对象。用户执行高级数据库和实例管理操作的能力由指派给他们的权限决定。在 DB2 中有 5 种不同的权限级别：SYSADM、SYSCTRL、SYSMAINT、DBADM 和 LOAD。

特权的粒度比授权要细，可以分配给用户和/或用户组。特权定义用户可以创建或删除的对象。它们还定义用户可以用来访问对象（比如表、视图、索引和包）的命令。DB2 9 中新增的一个概念是基于标签的访问控制（LBAC），它允许以更细的粒度控制谁有权访问单独的行和/或列。

1.DB2身份验证

DB2 使用*身份验证类型* 决定在什么地方进行身份验证。例如，在客户机 - 服务器环境中，是客户机还是服务器检验用户的 ID 和密码？在客户机 - 网关 - 主机环境中，是客户机还是主机检验用户的 ID 和密码？ 可以在DBM CFG中指定相应的验证类型:

DB2 GET DBM CFG  
Server Connection Authentication          (SRVCON\_AUTH) = KERBEROS  
Database manager authentication        (AUTHENTICATION) = SERVER\_ENCRYPT  
那么在连接实例时会使用 SERVER\_ENCRYPT。但是在连接数据库时会使用 KERBEROS 身份验证。

在客户机 - 网关 - 主机环境中，这些身份验证选项在客户机和网关上设置，而不是在主机上。

|  |  |
| --- | --- |
| **类型** | **描述** |
| SERVER | 身份验证在服务器上进行。 |
| SERVER\_ENCRYPT | 身份验证在服务器上进行。密码在客户机上进行加密，然后再发送到服务器。 |
| CLIENT | 身份验证在客户机上进行. |
| \*KERBEROS | 由 Kerberos 安全软件执行身份验证。 |
| \*KRB\_SERVER\_ENCRYPT | 如果客户机设置是 KERBEROS，那么由 Kerberos 安全软件执行身份验证。否则使用 SERVER\_ENCRYPT。 |
| DATA\_ENCRYPT | 身份验证在服务器上进行。服务器接受加密的用户 ID 和密码，并对数据进行加密。这个选项的操作方式与 SERVER\_ENCRYPT 相同，但是数据也要加密。 |
| DATA\_ENCRYPT\_CMP | 身份验证方式与 DATA\_ENCRYPT 相同，但是允许不支持 DATA\_ENCRYPT 的老式客户机使用 SERVER\_ENCRYPT 身份验证进行连接。在这种情况下，数据不进行加密。如果进行连接的客户机支持 DATA\_ENCRYPT，就会进行数据加密，而不能降级到 SERVER\_ENCRYPT 身份验证。这个身份验证类型只在服务器的数据库管理程序配置文件中是有效的，而且在客户机或网关实例上使用 CATALOG DATABASE 时是无效的。 |
| GSSPLUGIN | 身份验证方式由一个外部 GSS-API 插件决定。 |
| GSS\_SERVER\_ENCRYPT | 身份验证方式由一个外部 GSS-API 插件决定。在客户机不支持服务器的 GSS-API 插件之一的情况下，使用 SERVER\_ENCRYPT 身份验证。 |

2.授权

授权由特权组和高级数据库管理程序（实例级）维护和实用操作组成。在 DB2 可用的 5 种权限中，SYSADM、SYSCTRL 和 SYSMAINT 是*实例级权限*。这意味着它们的范围包含实例级命令以及针对这个实例中的所有数据库的命令。这些权限只能分配给组；可以通过 DBM CFG 文件分配这些权限。

针对特定数据库的 DBADM 和 LOAD 权限可以分配给用户或用户组。可以使用 GRANT 命令显式地分配这些权限。

**获得 SYSADM 权限**

SYSADM 权限由 DBM CFG 文件中的 SYSADM\_GROUP 参数控制。在 Windows 上，在创建实例时，这个参数设置为 Administrator（但是，如果发出命令 db2 get dbm cfg，它看起来是空的）。在 UNIX 上，它设置为创建这个实例的用户的主组。

因为只允许 SYSADM 用户更新 DBM CFG 文件，所以只有他们能够向其他组授予任何 SYS\* 权限。以下示例演示如何向 *db2grp1* 组授予 SYSADM 权限：

|  |
| --- |
| db2 update dbm cfg using SYSADM\_GROUP db2grp1 |

**获得 SYSCTRL 权限**

拥有 SYSCTRL 权限的用户可以在实例中执行所有管理和维护命令。但是，与 SYSADM 用户不同，他们不能访问数据库中的任何数据，除非他们被授予了访问数据所需的特权。SYSCTRL 用户可以对实例中的任何数据库执行的命令示例如下：

db2start/db2stopdb2 create/drop databasedb2 create/drop tablespacedb2 backup/restore/rollforward databasedb2 runstats（针对任何表） db2 update db cfg for database *dbname*

拥有 SYSADM 权限的用户可以使用以下命令将 SYSCTRL 分配给一个组：

|  |
| --- |
| db2 update dbm cfg using SYSCTRL\_GROUP group name |

|  |
| --- |
|  |

**获得 SYSMAINT 权限**

拥有 SYSMAINT 权限的用户可以发出的命令是拥有 SYSCTRL 权限的用户可以发出的命令的子集。SYSMAINT 用户只能执行与维护相关的任务，比如：

db2start/db2stopdb2 backup/restore/rollforward databasedb2 runstats（针对任何表） db2 update db cfg for database *dbname*

注意，拥有 SYSMAINT 权限的用户不能创建或删除数据库或表空间。他们也不能访问数据库中的任何数据，除非他们被显式地授予访问数据所需的特权。

如果您拥有 SYSADM 权限，那么可以使用以下命令将 SYSMAINT 权限分配给一个组：

|  |
| --- |
| db2 update dbm cfg using SYSMAINT\_GROUP group name |

**获得 DBADM 权限**

DBADM 权限是一个数据库级权限，而不是实例级权限。DBADM 用户对一个数据库有几乎完全的控制能力。DBADM 用户不能执行某些维护或管理任务，比如：

drop databasedrop/create tablespacebackup/restore databaseupdate db cfg for database *db name*

但是，他们可以执行以下任务：

db2 create/drop tabledb2 grant/revoke（任何特权） db2 runstats（任何表）

DBADM 用户还被自动地授予对数据库对象及其内容的所有特权。因为 DBADM 权限是一个数据库级权限，所以它可以被分配给用户和用户组。以下命令演示授予 DBADM 权限的不同方法。

db2 create database test

这个命令将数据库 *test* 上的 DBADM 权限隐式地授予发出此命令的用户。

db2 connect to sample   
db2 grant dbadm on database to user tst1

这个命令只能由 SYSADM 用户发出；它向用户 *tst1* 授予示例数据库上的 DBADM 权限。注意，在授予 DBADM 权限之前，发出这个命令的用户必须连接到示例数据库。

db2 grant dbadm on database to group db2grp1

这个命令将 DBADM 权限授予 *db2grp1* 组中的每个用户。同样，只有 SYSADM 用户能够发出这个命令。

**获得 LOAD 权限**

LOAD 权限是一个数据库级权限，所以它可以被分配给用户和用户组。顾名思义，LOAD 权限允许用户对表发出 LOAD 命令。当用大量数据填充表时，LOAD 命令通常用来替代插入或导入命令，它的速度更快。根据您希望执行的 LOAD 操作类型，仅仅拥有 LOAD 权限可能还不够。可能还需要表上的特定特权。

拥有 LOAD 权限的用户可以运行以下命令：

db2 quiesce tablespaces for tabledb2 list tablespacesdb2 runstats（任何表） db2 load insert（必须有表上的插入特权） db2 load restart/terminate after load insert（必须有表上的插入特权） db2 load replace（必须有表上的插入和删除特权） db2 load restart/terminate after load replace（必须有表上的插入和删除特权）

只有拥有 SYSADM 或 DBADM 权限的用户能够对用户或用户组授予或撤消 LOAD 权限。以下示例演示 LOAD 权限如何允许我们的用户使用 LOAD 命令将数据装载进 *sales* 表中。假设已经发出了命令 db2 connect to sample。

db2 grant load on database to user tst1   
db2 grant insert on table sales to user tst1

有了 LOAD 权限和插入特权，*tst1* 就可以对 sales 表发出 LOAD INSERT 或 LOAD RESTART，或者在 LOAD INSERT 之后发出 TERMINATE。

db2 grant load on database to group grp1   
db2 grant delete on table sales to group grp1   
db2 grant insert on table sales to group grp1

有了 LOAD 权限以及删除和插入特权，*grp1* 的任何成员就可以对 sales 表发出 LOAD REPLACE 或 LOAD RESTART，或者在 LOAD REPLACE 之后发出 TERMINATE。

3.特权

用户可以拥有的数据库级特权有：

CREATETAB： 用户可以在数据库中创建表。 BINDADD： 用户可以使用 BIND 命令在数据库中创建包。 CONNECT： 用户可以连接数据库。 CREATE\_NOT\_FENCED： 用户可以创建 unfenced 用户定义函数（UDF）。 IMPLICIT\_SCHEMA： 用户可以在数据库中隐式地创建模式，而不需要使用 CREATE SCHEMA 命令。 LOAD： 用户可以将数据装载进表中。 QUIESCE\_CONNECT： 用户可以访问处于静默（quiesced）状态的数据库。 CREATE\_EXTERNAL\_ROUTINE： 用户可以创建供应用程序和数据库的其他用户使用的过程。

数据库*对象* 包括表、视图、索引、模式和包。幸运的是，大多数对象级特权的意义无需解释。下表总结了这些特权。

|  |  |  |
| --- | --- | --- |
| **特权名称** | **相关对象** | **描述** |
| CONTROL | 表、视图、索引、包、别名、不同的类型、用户定义函数、序列 | 提供对对象的全部权限。拥有这种特权的用户还可以向其他用户授予或撤消对对象的特权。 |
| DELETE | 表、视图 | 允许用户从对象中删除记录。 |
| INSERT | 表、视图 | 允许用户通过 INSERT 或 IMPORT 命令将记录插入对象中。 |
| SELECT | 表、视图 | 提供使用选择语句来查看对象内容的能力。 |
| UPDATE | 表、视图 | 允许用户使用更新语句修改对象中的记录。 |
| ALTER | 表 | 允许用户使用更改语句更改对象定义。 |
| INDEX | 表 | 允许用户使用创建索引语句在对象上创建索引。 |
| REFERENCES | 表 | 提供在对象上创建或删除外键约束的能力。 |
| BIND | 包 | 允许用户重新绑定现有的包。 |
| EXECUTE | 包、过程、函数、方法 | 允许用户执行包和例程。 |
| ALTERIN | 模式 | 允许用户修改模式中的对象定义。 |
| CREATEIN | 模式 | 允许用户在模式中创建对象。 |
| DROPIN | 模式 | 允许用户删除模式中的对象。 |

关于对象级特权的信息存储在系统编目视图中。视图名称是 syscat.tabauth、syscat.colauth、syscat.indexauth、syscat.schemaauth、syscat.routineauth 和 syscat.packageauth。

**显式特权**

可以使用 GRANT 和 REVOKE 命令*显式地* 对用户或组授予或撤消特权。我们来看看如何在各种对象上使用这些命令。

作为拥有 Administrator 权限的用户登录 Windows，打开两个 DB2 命令窗口。在这两个窗口中，确保将 db2instance 变量设置为 DB2！

在第一个窗口中发出以下命»¤：

|  |
| --- |
| db2 connect to sample |

现在，在第二个窗口中发出以下命令：

|  |
| --- |
| db2 connect to sample user test1 using password |

请记住，第一个窗口中的命令是由一个拥有 SYSADM 权限的用户发出的。第二个窗口中的命令是由 *tst1* 发出的，这个用户对示例数据库没有特殊的权限或特权。注意，与示例数据库中的表相关联的模式名是发出 db2sampl 命令的用户的名称。在这些示例中，这个用户是 *GMILNE*。

现在，在第二个窗口中发出以下命令：

|  |
| --- |
| db2 select \* from gmilne.org |

应该会看到以下响应：

|  |
| --- |
| SQL0551N "TEST1" does not have the privilege to perform operation "SELECT" on object "GMILNE.ORG". |

为了纠正这种状况，在第一个窗口中发出以下命令：

|  |
| --- |
| db2 grant select on table gmilne.org to user test1 |

现在，前面的命令就会成功！接下来，在第二个窗口中发出一个更复杂的命令：

|  |
| --- |
| db2 insert into gmilne.org values (100, 'Tutorial', 1, 'Eastern', 'Toronto') |

同样会看到错误消息：

|  |
| --- |
| SQL0551N "TEST1" does not have the privilege to perform operation "INSERT" on object "GMILNE.ORG" |

所以，在第一个窗口中输入以下命令：

|  |
| --- |
| db2 grant insert on table gmilne.org to group db2grp1 |

原来失败的 INSERT 命令现在应该会成功完成，因为 *test1* 是 *db2grp1* 组的成员。

现在，在第二个窗口中输入以下命令：

|  |
| --- |
| db2 drop table gmilne.emp\_photo |

同样会看到错误消息：

|  |
| --- |
| SQL0551N "TEST1" does not have the privilege to perform operation "DROP TABLE" on object "GMILNE.EMP\_PHOTO". |

所以，我们要授予这个特权。在第一个窗口中输入以下命令：

|  |
| --- |
| db2 grant dropin on schema gmilne to all |

DROP TABLE 命令现在应该会成功完成。

既然已经完成了示例，就可以撤消刚才授予的特权。在第一个窗口中发出以下命令：

|  |
| --- |
| db2 revoke select on table gmilne.org from user test1 db2 revoke insert on table gmilne.org from group db2grp1 db2 revoke dropin on schema gmilne from all |

注意，从组中撤消特权不一定会从这个组的所有成员撤消它。例如，以下命令可以用来从 *db2grp1* 撤消对 gmilne.org 表的所有特权（CONTROL 除外）：

|  |
| --- |
| db2 revoke all on table gmilne.org from group db2grp1 |

但是，*test1* 用户（他是 *db2grp1* 的成员）仍然拥有对这个表的选择特权，因为他或她是被直接授予这个特权的。

**隐式特权**

当发出某些命令时，DB2 可能会自动地授予特权，而不需要像前面看到的那样发出显式的 GRANT 语句。下表总结了会导致数据库管理程序隐式地授予特权的一些命令。注意，当删除创建的对象时，这些特性会隐式地撤消。但是，当显式地撤消更高级的特权时，不会撤消它们。

|  |  |  |
| --- | --- | --- |
| **发出的命令** | **授予的特权** | **被授予特权的用户** |
| CREATE TABLE mytable | *mytable* 上的 CONTROL | 发出命令的用户 |
| CREATE SCHEMA myschema | *myschema* 上的 CREATEIN、ALTERIN 和 DROPIN，以及将这些特权授予其他用户的能力 | 发出命令的用户 |
| CREATE VIEW myview | *myview* 上的 CONTROL（只有在用户拥有 myview 定义中引用的所有表和视图上的 CONTROL 特权的情况下） | 发出命令的用户 |
| CREATE DATABASE mydb | *mydb* 的系统编目表上的 SELECT，mydb 上的 IMPLICIT\_SCHEMA \* | PUBLIC\*\* |

\*当用户创建数据库时，隐式地授予这个用户这个数据库上的 DBADM 权限。获得 DBADM 权限就会隐式地授予 CONNECT、CREATETAB、BINDADD、IMPLICIT\_SCHEMA 和 CREATE\_NOT\_FENCED 特权。即使撤消了 DBADM 权限，这个用户仍然会保留这些特权。

\*\*PUBLIC 是一个特殊的 DB2 组，其中包括特定数据库的所有用户。与前面讨论过的其他组不同，PUBLIC 不必在操作系统级进行定义。在默认情况下，会向 PUBLIC 授予一些特权。例如，这个组自动接受数据库上的 CONNECT 特权和编目表上的 SELECT 特权。可以对 PUBLIC 组发出 GRANT 和 REVOKE 命令，比如：

|  |
| --- |
| db2 grant select on table sysibm.systables to public db2 revoke select on table sysibm.systables from public |

**间接特权**

当数据库管理器执行*包* 时，可以间接获得特权。包中包含一个或多个 SQL 语句，这些语句已经转换为 DB2 用来在内部执行它们的格式。换句话说，包中包含可执行格式的多个 SQL 语句。如果包中的所有语句都是静态的，那么用户只需要有包上的 EXECUTE 特权，就能够成功地执行包中的语句。

例如，假设 *db2package1* 执行以下静态的 SQL 语句：

|  |
| --- |
| db2 select \* from orgdb2 insert into test values (1, 2, 3) |

在这种情况下，拥有 *db2package1* 上的 EXECUTE 特权的用户会间接地获得 org 表上的 SELECT 特权和 test 表上的 INSERT 特权。