DCL - Data Control Language

The third group of SQL keywords is the Data Control Language (DCL). DCL handles the authorisation aspects of data and permits the user to control who has access to see or manipulate data within the database.

Its two main keywords are:

- GRANT: - authorises a user to perform an operation or a set of operations e.g. grant all privileges to user X.
- REVOKE: - removes or restricts the capability of a user to perform an operation or a set of operations.

GRANT

GRANT is the SQL statement, used to assign privileges to database users for specified database objects.

Grants can be assigned and revoked using the IBExpert Grant Manager, the relevant object editors' Grants pages, or the SQL Editor.

Firebird/InterBase® offers the following access privileges at database object level:

Privilege	**Allows user to:**
SELECT	Read data.
INSERT	Write new data.
UPDATE	Modify existing data.
DELETE	Delete data.
ALL	Select, insert, update, delete data, and reference a primary key from a foreign key. (Note: does not include references or code for InterBase® 4.0 or earlier).
EXECUTE	Execute or call a stored procedure.
REFERENCES	Reference a primary key with a foreign key.
role	Use all privileges assigned to the role (please refer to Role for further information).

PUBLIC is used to assign a set of privileges to every user of the database. Using the PUBLIC keyword does not grant the specified rights to stored procedures, only to all database users. Procedures need to be specified explicitly. Please note: PUBLIC is really public! This GRANT option enables all users to access and manipulate a database object with PUBLIC rights, even certain system files.

Table Interactions

Many operations require that the user has rights to linked tables, in order for Firebird/InterBase® to process updates.

- If foreign key constraints exist between two tables, then an UPDATE, DELETE or INSERT operation on the first table requires SELECT or REFERENCES privileges on the referenced table. *Tip:* Make it easy: if read security is not an issue, GRANT REFERENCES on the primary key table

to PUBLIC. If you grant the REFERENCES privilege, it must, at a minimum, be granted to all columns of the primary key. When REFERENCES is granted to the entire table, columns that are not part of the primary key are not affected in any way. When a user defines a foreign key constraint on a table owned by someone else, Firebird/InterBase® checks that the user has REFERENCES privileges on the referenced table. The privilege is used at runtime to verify that a value entered in a foreign key field is contained in the primary key table. You can grant REFERENCES privileges to roles.

- If there is a check constraint within a table, an UPDATE or INSERT operation also requires SELECT privileges on the same table.
- If a constraint includes one or more queries, an UPDATE or INSERT operation also requires SELECT privileges on the table or tables used in the SELECT.

IBExpert allows privileges to be granted on objects at the time of creation directly in the objects editor's *Grants* page (please refer to Table Editor / Grants for further details). Dependencies upon or from other objects are also displayed in the individual object editors, to show visually any object interactions, which may need to be taken into consideration when assigning user permissions. Refer to Table Editor / Dependencies for further information. All objects or a filtered selection of objects can be displayed and processed in the IBExpert Grant Manager.

Privileges can be granted to a role as well as to users or stored procedures, tables, views and triggers.

The GRANT statement can be used in gpre, DSQL and isql.

Syntax

```
GRANT privileges ON [TABLE] {tablename | viewname}
      TO {object|userlist [WITH GRANT OPTION]|GROUP UNIX_group}
   | EXECUTE ON PROCEDURE procname TO {object | userlist}
   | role_granted TO {PUBLIC | role_grantee_list}[WITH ADMIN OPTION];

<privileges> = ALL [PRIVILEGES] | privilege_list

<privilege_list> = {
     SELECT
   | DELETE
   | INSERT
   | UPDATE [(col [, col...])]
   | REFERENCES [(col [, col...])]
}[, privilege_list...]

<object> = {
     PROCEDURE procname
   | TRIGGER trigname
   | VIEW viewname
   | PUBLIC
}[, object...]

<userlist> = {
     [USER] username
   | rolename
   | UNIX_user
```

```
}[,userlist...]

<role_granted> = rolename [, rolename...]

<role_grantee_list> = [USER] username [, [USER] username...]
```

privilege_list	Name of privilege to be granted; valid options are SELECT, DELETE, INSERT, UPDATE, and REFERENCES.
col	Column to which the granted privileges apply.
tablename	Name of an existing table for which granted privileges apply.
viewname	Name of an existing view for which granted privileges apply.
GROUP unix_group	On a UNIX system, the name of a group defined in /etc/group.
object	Name of an existing procedure, trigger, or view; PUBLIC is also a permitted value.
userlist	A user in the Firebird/InterBase® security database or a role name created with CREATE ROLE.
WITH GRANT OPTION	Passes GRANT authority for privileges listed in the GRANT statement to userlist (please refer to GRANT AUTHORITY for further information).
rolename	An existing role created with the CREATE ROLE statement.
role_grantee_list	A list of users to whom rolename is granted; users must be in the Firebird/InterBase® .
WITH ADMIN OPTION	Passes grant authority for roles listed to role_grantee_list.

Since Firebird 2.5 the GRANTED BY or GRANTED AS clause can be optionally included in GRANT and REVOKE statements, enabling the grantor to be a user other than the CURRENT_USER (the default). Please refer to the Firebird 2.5 Release Notes for syntax and examples.

Important: In SQL statements passed to DSQL, omit the terminating semicolon. In embedded applications written in C and C++, and in isql, the semicolon is a terminating symbol for the statement, so it must be included.

To grant privileges to a group of users, create a role using the CREATE ROLE statement. Please refer to New Role for details.

On UNIX systems, privileges can be granted to groups listed in /etc/groups and to any UNIX user listed in /etc/passwd on both the client and server, as well as to individual users and to roles.

Examples

```
GRANT insert, update, delete
   ON customer
   TO Janet, John
   WITH GRANT OPTION;
```

or:

```
GRANT references
   ON customer
   TO PUBLIC;
```

If different levels of access are to be assigned to different objects and different people, separate

GRANT statements have to be used.

This embedded SQL statement grants EXECUTE privileges for a procedure to another procedure and to a user:

```
EXEC SQL
   GRANT EXECUTE ON PROCEDURE GET_EMP_PROJ
   TO PROCEDURE ADD_EMP_PROJ, LUIS;
```

The following example creates a role called administrator, grants UPDATE privileges on table1 to that role, and then grants the role to user1, user2, and user3. These users then have UPDATE and REFERENCES privileges on table1:

```
CREATE ROLE administrator;
GRANT UPDATE ON table1 TO administrator;
GRANT administrator TO user1, user2, user3;
```

back to top of page

REVOKE

REVOKE is the SQL statement, used to withdraw those rights already assigned to database users or objects for database objects. Rights can be revoked using the IBExpert Grant Manager, the relevant object editors' Grants pages, or the SQL Editor.

The following rules apply when revoking user privileges:

1. Only the user who granted the privilege or the SYSDBA may revoke it.
2. Revoking a privilege has no effect on any other privileges granted by other users. However, if multiple users have the ability to grant privileges, one user might have received a specific privilege from more than one source. If only one of them is revoked, the other remains in effect.
3. If a privilege, which was originally granted using the WITH GRANT OPTION clause, is revoked, any subsequent users to which the same privilege had been granted in turn lose their privileges too.
4. The ALL keyword can be used to revoke all granted privileges to an object or user, even if the user has not been granted all available privileges in the first place. REVOKE ALL however has no effect on the EXECUTE privilege, which must always be explicitly revoked.
5. If a privilege is granted to all users using the PUBLIC option, this grant can only be revoked using the same PUBLIC option.

Syntax

```
REVOKE [GRANT OPTION FOR] privilege ON [TABLE] {tablename | viewname}
     FROM {object | userlist | rolelist | GROUP UNIX_group}
   | EXECUTE ON PROCEDURE procname FROM {object | userlist}
   | role_granted FROM {PUBLIC | role_grantee_list}};
<privileges> = ALL [PRIVILEGES] | privilege_list
<privilege_list> = {
```

```
     SELECT
   | DELETE
   | INSERT
   | UPDATE [(col [, col ...])]
   | REFERENCES [(col [, col ...])]
   }[, privilege_list ...]
<object> = {
     PROCEDURE procname
   | TRIGGER trigname
   | VIEW viewname
   | PUBLIC
   }[, object ...]
<userlist> = [USER] username [, [USER] username ...]
<rolelist> = rolename [, rolename]
<role_granted> = rolename [, rolename ...]
<role_grantee_list> = [USER] username [, [USER] username ...]
```

privilege_list	Name of privilege to be granted; valid options are SELECT, DELETE, INSERT, UPDATE and REFERENCES.
GRANT OPTION FOR	Removes grant authority for privileges listed in the REVOKE statement from userlist; cannot be used with object.
col	Column for which the privilege is revoked.
tablename	Name of an existing table for which privileges are revoked.
viewname	Name of an existing view for which privileges are revoked.
GROUP unix_group	On a UNIX system, the name of a group defined in /etc/group.
object	Name of an existing database object from which privileges are to be revoked.
userlist	A list of users from whom privileges are to be revoked.
rolename	An existing role created with the CREATE ROLE statement.
role_grantee_list	A list of users to whom rolename is granted; users must be in the Firebird/InterBase® security database.

Since Firebird 2.5 the GRANTED BY or GRANTED AS clause can be optionally included in GRANT and REVOKE statements, enabling the grantor to be a user other than the CURRENT_USER (the default). Please refer to the Firebird 2.5 Release Notes for syntax and examples.

Examples

To revoke INSERT and UPDATE privileges from Janet and John:

```
REVOKE INSERT, UPDATE
   ON PROJ_DEPT_BUDGET
   FROM Janet, John
```

To revoke all privileges from every user, use the PUBLIC option, for example:

```
REVOKE ALL
   ON PROJ_DEPT_BUDGET
   FROM PUBLIC;
```

REVOKE ADMIN OPTION FROM was introduced in Firebird 2.0.4. Refer to the Firebird 2.04 Release

Notes section, REVOKE ADMIN OPTION FROM, for details.