

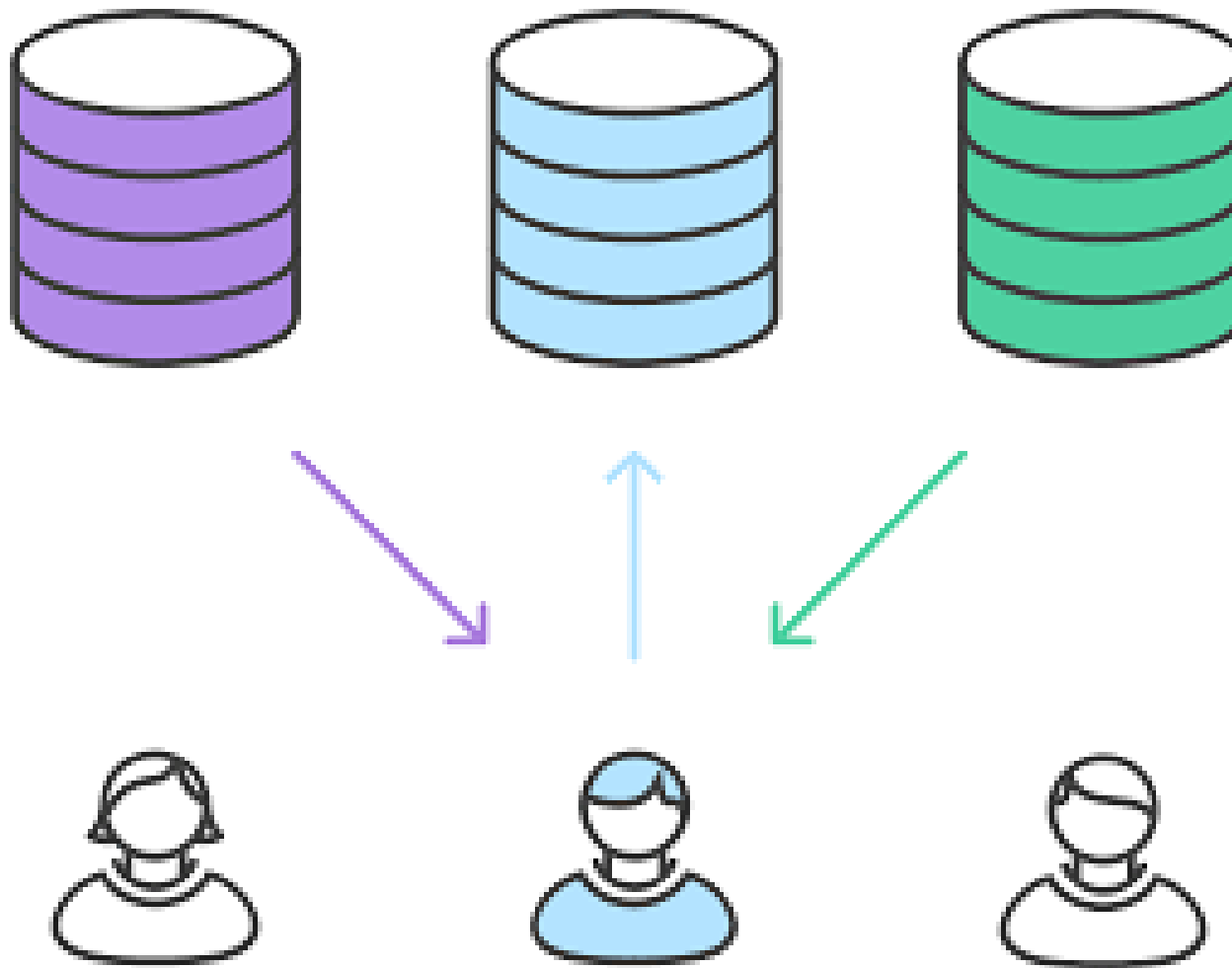
A4. Gestión de usuarios.

Índice.

1.	¿Qué es un usuario de una Base de Datos?	2
2.	Tipos de usuarios	4
2.1.	Superusuario	5
2.2.	Usuario ingenuo	6
2.3.	Programador de aplicaciones	7
2.4.	Usuario sofisticado	8
2.5.	Usuario especializado	9
3.	Mantenimiento de usuarios	10
3.1.	Creación de usuario	11
3.2.	Modificación de usuario	13
3.3.	Eliminación de usuario	14
3.4.	Renombrado de usuario	14
3.5.	Ver usuario actual	15
3.6.	Ver todos los usuarios	15
4.	Mantenimiento de permisos	16
4.1.	Tipos de permisos	17
4.2.	Niveles de permisos	18
4.3.	Creación de permiso	19
4.4.	Eliminación de permiso	20
4.5.	Ver permiso	21
4.6.	Agrupar de permisos	21

A4. Gestión de usuarios.

1. ¿Qué es un usuario de una Base de Datos?

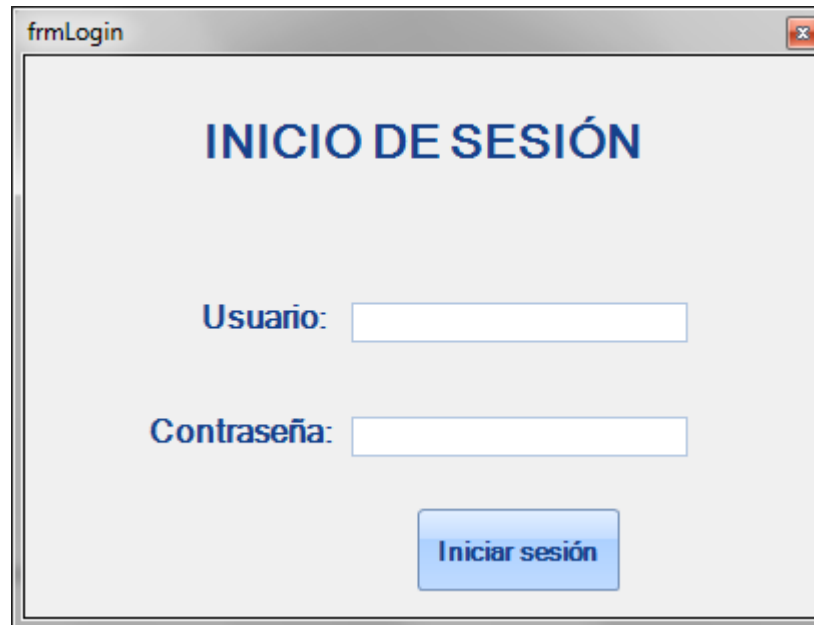


A4. Gestión de usuarios.

1. ¿Qué es un usuario de una Base de Datos?

El **usuario** de la base de datos es la **identidad del inicio de sesión** cuando alguien se conecta a una base de datos, es decir, es una entidad de seguridad de la base de datos.

El inicio de alguna sesión debe estar asignada a un usuario de la base de datos para conectarse.



The image shows a screenshot of a login form window titled "frmLogin". The window has a light gray background and a standard Windows-style title bar with a close button. The main heading "INICIO DE SESIÓN" is displayed in a large, bold, blue font. Below the heading, there are two input fields: "Usuario:" followed by a text box, and "Contraseña:" followed by a text box. At the bottom center of the form, there is a blue button with the text "Iniciar sesión" in white.

A4. Gestión de usuarios.

2. Tipos de usuarios.

Los usuarios de las bases de datos se pueden clasificar de muy diversas formas, una de las cuales es la siguiente:

- Superusuario.
- Usuario ingenuo.
- Programador de aplicaciones.
- Usuario sofisticado.
- Usuario especializado.



A4. Gestión de usuarios.

2.1. Superusuario.

El **superusuario** (o administrador) es una persona (o un grupo) que tiene las siguientes funciones:

- Control absoluto de la base de datos → cuenta de Administrador.
- Define esquemas lógicos y físicos de la base de datos, con control de los esquemas de nivel de vista.
- Gestiona los tres niveles de la base de datos.
- Concede o revoca permisos de autorización.
- Diseña la estructura general de la base de datos → diseño, funcionamiento, procedimientos y motivos.
- Responsable del mantenimiento de rutinas → copia de seguridad y recuperación de la base de datos.
- Realiza actividades asociadas a la administración → actualizaciones periódicas, inserción de datos o funciones requeridas, modificación, etc.
- Proporciona y organiza soporte técnico.
- Controla operaciones como gestión de seguridad, integridad, redundancia, concurrencia, hardware y software.
- Mantiene actualizada la base de datos en términos de tecnología, función, objetivo y cumplimiento de requisitos.



A4. Gestión de usuarios.

2.2. Usuario ingenuo.

El **usuario ingenuo** (o usuario final) es una persona que usa la base de datos meramente para completar información, pero carente de conocimientos asociados a las bases de datos ni de su funcionamiento, simplemente hace uso de las funcionalidades asociadas a su nivel de usuario final.

El **usuario final** se divide en dos tipos:

- **Usuario paramétrico** → sólo usa programas predefinidos para realizar operaciones → reservar boletos, completar formularios, solicitar instalaciones, etc.
- **Usuario ocasional** → usa programación básica para completar datos en la base de datos, a través de guías de usuario que ayudan para la realización de tareas.



A4. Gestión de usuarios.

2.3. Programador de aplicaciones.

El **programador de aplicaciones** (o usuario backend) es una persona que realiza toda la programación de la base de datos para que sea funcional y operativa.

El **usuario backend** se puede clasificar en:

- **Diseñador** → sólo usa programas predefinidos para realizar operaciones → reservar boletos, completar formularios, solicitar instalaciones, etc.
- **Analista de sistemas** → usa programación básica para completar datos en la base de datos, a través de guías de usuario que ayudan para la realización de tareas.
- **Programador** → realiza finalmente la codificación de la base datos. Subtipos:
 - **Programación con herramientas** → desarrolla la programación en función de las herramientas disponibles.
 - **Programación sin herramientas** → desarrolla la programación sin utilizar ningún tipo de herramientas que condiciona resultados.



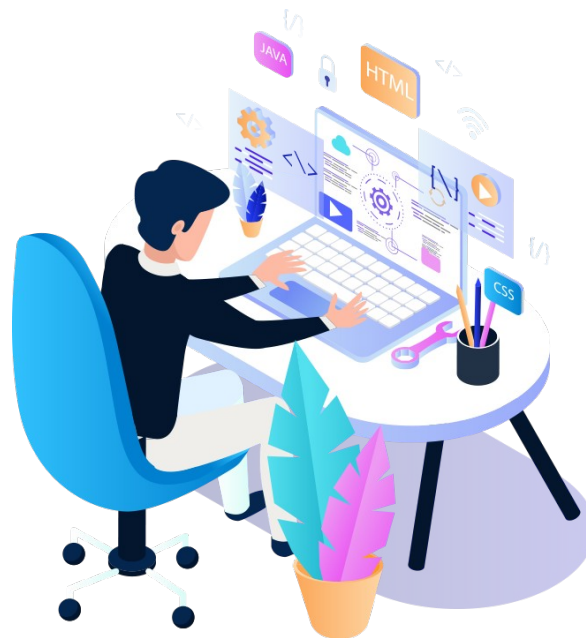
A4. Gestión de usuarios.

2.4. Usuario sofisticado.

El **usuario sofisticado** es una persona con conocimientos del lenguaje de definición y manipulación de datos y los emplea para crear sus propias bases de datos y para acceder a la base de datos actual.

El **usuario sofisticado** es un pequeño conocedor de la tecnología o con la completa capacitación para hacer lo necesario.

Este usuario puede ser un ingeniero, analista o científico de la misma organización o de otra.



A4. Gestión de usuarios.

2.5. Usuario especializado.

El **usuario especializado** es una combinación entre un administrador y un programador de bases de datos.

El **usuario especializado**:

- Escribe su propios programas de acceso a bases de datos.
- Puede llegar a superar la estructura de acceso secuencial y el acceso cruzado en el marco de la base de datos.
- No necesita seguir ningún procedimiento SINO que los realiza.
- Es contratado para averiguar errores y anomalías en el sistema actual.



A4. Gestión de usuarios.

3. Mantenimiento de usuario.

El **mantenimiento de los usuarios**, o más concretamente, de sus cuentas se realiza a través de las operaciones de:

- Creación → comando CREATE USER.
- Modificación → comando ALTER USER.
- Eliminación → comando DROP USER.
- Renombrar → comando RENAME USER.
- Ver actual → comando SHOW USER.
- Listado → comando SELECT user FROM mysqluser.
- Cambiar de usuario en pantalla de mysql → system MYSQL -u

A4. Gestión de usuarios.

3.1. Creación de usuario.

La **creación de usuarios** consiste en la creación de cuentas de usuario.

```
CREATE USER [IF NOT EXISTS]
  user [auth_option] [, user [auth_option]] ...
  DEFAULT ROLE role [, role ] ...
  [REQUIRE {NONE | tls_option [[AND] tls_option] ...}]
  [WITH resource_option [resource_option] ...]
  [password_option | lock_option] ...
  [COMMENT 'comment_string' | ATTRIBUTE 'json_object']

user:
  (see Section 6.2.4, "Specifying Account Names")

auth_option: {
  IDENTIFIED BY 'auth_string' [AND 2fa_auth_option]
| IDENTIFIED BY RANDOM PASSWORD [AND 2fa_auth_option]
| IDENTIFIED WITH auth_plugin [AND 2fa_auth_option]
| IDENTIFIED WITH auth_plugin BY 'auth_string' [AND 2fa_auth_option]
| IDENTIFIED WITH auth_plugin BY RANDOM PASSWORD [AND 2fa_auth_option]
| IDENTIFIED WITH auth_plugin AS 'auth_string' [AND 2fa_auth_option]
| IDENTIFIED WITH auth_plugin [initial_auth_option]
}

initial_auth_option: {
  INITIAL AUTHENTICATION IDENTIFIED BY {RANDOM PASSWORD | 'auth_string'}
| INITIAL AUTHENTICATION IDENTIFIED WITH auth_plugin AS 'auth_string'
}

tls_option: {
  SSL
| X509
| CIPHER 'cipher'
| ISSUER 'issuer'
| SUBJECT 'subject'
}

lock_option: {
  ACCOUNT LOCK
| ACCOUNT UNLOCK
}

resource_option: {
  MAX_QUERIES_PER_HOUR count
| MAX_UPDATES_PER_HOUR count
| MAX_CONNECTIONS_PER_HOUR count
| MAX_USER_CONNECTIONS count
}

password_option: {
  PASSWORD EXPIRE [DEFAULT | NEVER | INTERVAL N DAY]
| PASSWORD HISTORY {DEFAULT | N}
| PASSWORD REUSE INTERVAL {DEFAULT | N DAY}
| PASSWORD REQUIRE CURRENT [DEFAULT | OPTIONAL]
| FAILED_LOGIN_ATTEMPTS N
| PASSWORD_LOCK_TIME {N | UNBOUNDED}
}
```

A4. Gestión de usuarios.

3.1. Creación de usuario.

Algunos ejemplos de usuarios que podemos crear son los siguientes:

- **Nombre de usuario** → Fulgencio

```
CREATE USER Fulgencio ;
```

- **Nombre y contraseña** → Fulgencio con contraseña 'abc123.'

```
CREATE USER Fulgencio IDENTIFIED BY 'abc123.' ;
```

- **Nombre, contraseña y expiración** → Fulgencio, contraseña 'abc123.' y duración 180 días.

```
CREATE USER Fulgencio IDENTIFIED BY 'abc123.' PASSWORD EXPIRE INTERVAL 180 DAY ;
```

A4. Gestión de usuarios.

3.2. Modificación de usuario

La modificación de los datos de un usuario se realiza a través del siguiente comando:

```
ALTER USER [IF EXISTS]
  user [auth_option] [, user [auth_option]] ...
  [REQUIRE {NONE | tls_option [[AND] tls_option] ...}]
  [WITH resource_option [resource_option] ...]
  [password_option | lock_option] ...
  [COMMENT 'comment_string' | ATTRIBUTE 'json_object']

ALTER USER [IF EXISTS]
  USER() user_func_auth_option

ALTER USER [IF EXISTS]
  user [registration_option]

ALTER USER [IF EXISTS]
  user DEFAULT ROLE
  {NONE | ALL | role [, role ] ...}

auth_option: {
  IDENTIFIED BY 'auth_string'
    [REPLACE 'current_auth_string']
    [RETAIN CURRENT PASSWORD]
  | IDENTIFIED BY RANDOM PASSWORD
    [REPLACE 'current_auth_string']
    [RETAIN CURRENT PASSWORD]
  | IDENTIFIED WITH auth_plugin
  | IDENTIFIED WITH auth_plugin BY 'auth_string'
    [REPLACE 'current_auth_string']
    [RETAIN CURRENT PASSWORD]
  | IDENTIFIED WITH auth_plugin BY RANDOM PASSWORD
    [REPLACE 'current_auth_string']
    [RETAIN CURRENT PASSWORD]
  | IDENTIFIED WITH auth_plugin AS 'auth_string'
  | DISCARD OLD PASSWORD
  | ADD factor factor_auth_option [ADD factor factor_auth_option]
  | MODIFY factor factor_auth_option [MODIFY factor factor_auth_option]
  | DROP factor [DROP factor]
}
```

A4. Gestión de usuarios.

3.3. Eliminación de usuario.

La eliminación de los datos de un usuario se realiza a través del siguiente comando:

```
DROP USER [IF EXISTS] user [, user] ...
```

3.4. Renombrado de usuario.

El renombrado de un usuario se realiza a través del siguiente comando:

```
RENAME USER old_user TO new_user  
[, old_user TO new_user] ...
```

A4. Gestión de usuarios.

3.5. Ver usuario actual.

La comprobación del usuario actual conectado es a través de alguno de los siguientes comandos:

```
mysql> SELECT USER();
```

```
mysql> SELECT CURRENT_USER();
```

3.5. Ver todos los usuarios.

La visualización de todos los usuarios se realiza a través del siguiente comando:

```
mysql> SELECT user FROM mysql.user;
```

A4. Gestión de usuarios.

4. Mantenimiento de permisos.

Los **permisos** son privilegios para acceder a alguna parte de la base de datos, bien sea sobre:

- Objetos para realizar cambios en las tablas de la base de datos.
- Sistema para ejecutar algún tipo de comando SQL o para realizar alguna acción sobre los objetos de algún tipo determinado.

Las operaciones que se pueden realizar con los permisos son:

- Creación → comando GRANT
- Eliminación → comando REVOKE
- Visualización → comando SHOW GRANTS
- Agrupación → comando ROLE
- Refresco → comando FLUSH PRIVILEGES

Cuando se crea un usuario, el único permiso que tiene es **USAGE**, que le da derecho de conexión a MySQL.

A4. Gestión de usuarios.

4.1. Niveles de permisos.

Los **permisos** se pueden clasificar en los siguientes niveles:

- **A nivel global** → los permisos son filas que se añaden, borran o modifican en la tabla **mysql.user** y se aplican en todas las bases de datos.

```
GRANT SELECT,UPDATE ON *.* TO 'user1'@'localhost';
```

- **A nivel de base de datos** → los permisos son filas que se añaden, borran o modifican en la tabla **mysql.db** y se aplican sobre esa misma base de datos.

```
GRANT CREATE ROUTINE, ALTER ROUTINE ON mydb.* TO 'someuser'@'somehost';
```

- **A nivel de tabla** → los permisos son filas que se añaden, borran o modifican en la tabla **mysql.tables_priv** y se aplican en una tabla de una base de datos concreta.

```
GRANT SELECT, INSERT ON mydb.mytbl TO 'someuser'@'somehost';
```

- **A nivel de columna** → los permisos son filas que se añaden, borran o modifican en la tabla **mysql.columns_priv** y se aplican en una columna concreta de una tabla de una base de datos.

```
GRANT SELECT (col1), INSERT (col1,col2) ON mydb.mytbl TO 'someuser'@'somehost';
```

- **A nivel de rutinas (procedimientos almacenados y funciones)** → los permisos son filas que se añaden, borran o modifican en la tabla **mysql.procs_priv** (una rutina concreta), **mysql.db** (rutinas de una base de datos concreta) y **mysql.user** (todas las rutinas).

```
GRANT CREATE ROUTINE ON mydb.* TO 'someuser'@'somehost';
```

A4. Gestión de usuarios.

4.2. Tipos de permisos.

Los **permisos** pueden ser de los siguientes tipos:

Privilege	Column	Context
<u>ALL [PRIVILEGES]</u>	Synonym for "all privileges"	Server administration
<u>ALTER</u>	Alter_priv	Tables
<u>ALTER ROUTINE</u>	Alter_routine_priv	Stored routines
<u>CREATE</u>	Create_priv	Databases, tables, or indexes
<u>CREATE ROUTINE</u>	Create_routine_priv	Stored routines
<u>CREATE TABLESPACE</u>	Create_tablespace_priv	Server administration
<u>CREATE TEMPORARY TABLES</u>	Create_tmp_table_priv	Tables
<u>CREATE USER</u>	Create_user_priv	Server administration
<u>CREATE VIEW</u>	Create_view_priv	Views
<u>DELETE</u>	Delete_priv	Tables
<u>DROP</u>	Drop_priv	Databases, tables, or views
<u>EVENT</u>	Event_priv	Databases
<u>EXECUTE</u>	Execute_priv	Stored routines
<u>FILE</u>	File_priv	File access on server host
<u>GRANT OPTION</u>	Grant_priv	Databases, tables, or stored routines
<u>INDEX</u>	Index_priv	Tables

Privilege	Column	Context
<u>INSERT</u>	Insert_priv	Tables or columns
<u>LOCK TABLES</u>	Lock_tables_priv	Databases
<u>PROCESS</u>	Process_priv	Server administration
<u>PROXY</u>	See proxies_priv table	Server administration
<u>REFERENCES</u>	References_priv	Databases or tables
<u>RELOAD</u>	Reload_priv	Server administration
<u>REPLICATION CLIENT</u>	Repl_client_priv	Server administration
<u>REPLICATION SLAVE</u>	Repl_slave_priv	Server administration
<u>SELECT</u>	Select_priv	Tables or columns
<u>SHOW DATABASES</u>	Show_db_priv	Server administration
<u>SHOW VIEW</u>	Show_view_priv	Views
<u>SHUTDOWN</u>	Shutdown_priv	Server administration
<u>SUPER</u>	Super_priv	Server administration
<u>TRIGGER</u>	Trigger_priv	Tables
<u>UPDATE</u>	Update_priv	Tables or columns
<u>USAGE</u>	Synonym for "no privileges"	Server administration

A4. Gestión de usuarios.

4.3. Creación de permiso.

La creación de permiso consiste en otorgar un permiso sobre un objeto:

```
GRANT
  priv_type [(column_list)]
    [, priv_type [(column_list)]] ...
ON [object_type] priv_level
TO user_or_role [, user_or_role] ...
[WITH GRANT OPTION]
[AS user
  [WITH ROLE
    DEFAULT
    | NONE
    | ALL
    | ALL EXCEPT role [, role ] ...
    | role [, role ] ...
  ]
]
}

object_type: {
  TABLE
  | FUNCTION
  | PROCEDURE
}

priv_level: {
  *
  | *.*
  | db_name.*
  | db_name.tbl_name
  | tbl_name
  | db_name.routine_name
}

user_or_role: {
  user (see Section 6.2.4, "Specifying Account Names")
  | role (see Section 6.2.5, "Specifying Role Names")
}

GRANT role [, role] ...
TO user_or_role [, user_or_role] ...
[WITH ADMIN OPTION]
```

A4. Gestión de usuarios.

4.4. Eliminación de permiso.

La eliminación de un permiso consiste en utilizar el comando REVOKE:

```
REVOKE [IF EXISTS]
    priv_type [(column_list)]
    [, priv_type [(column_list)]] ...
ON [object_type] priv_level
FROM user_or_role [, user_or_role] ...
[IGNORE UNKNOWN USER]

REVOKE [IF EXISTS] ALL [PRIVILEGES], GRANT OPTION
FROM user_or_role [, user_or_role] ...
[IGNORE UNKNOWN USER]

REVOKE [IF EXISTS] PROXY ON user_or_role
FROM user_or_role [, user_or_role] ...
[IGNORE UNKNOWN USER]

REVOKE [IF EXISTS] role [, role ] ...
FROM user_or_role [, user_or_role ] ...
[IGNORE UNKNOWN USER]
```

La eliminación de agrupaciones de permisos (roles) es con el comando DROP:

```
DROP ROLE [IF EXISTS] role [, role ] ...
```

A4. Gestión de usuarios.

4.5. Ver permiso.

La forma de visualizar los privilegios de un privilegio (o de un rol) es con SHOW GRANTS:

```
SHOW GRANTS
  [FOR user_or_role
    [USING role [, role] ...]]
```

4.6. Agrupar permisos.

La forma de agrupar permisos es con ROLE:

```
CREATE ROLE [IF NOT EXISTS] role [, role ] ...
```