Mecanismos de Seguridad e Integridad de los Datos en un SGBD

Seguridad e Integridad

El SGBD debe proveer técnicas que permitan a ciertos usuarios tener acceso a porciones selectas de una base de datos sin tener acceso al resto.

Por lo regular un SGBD cuenta con un subsistema de seguridad de autorización de la base de datos que se encarga de garantizar la seguridad de porciones de la base de datos contra el acceso no autorizado.

Entre las funciones del DBA está otorgar privilegios a los usuarios y clasificar los usuarios y los datos de acuerdo con la política de la organización.

Las órdenes privilegiadas del DBA incluyen los siguientes tipos de acciones:
1. Creación de cuentas
2. Concesión de privilegios.
3. Revocación de privilegios.
4. Asignación de niveles de seguridad.

La acción 1 de la lista sirve para controlar el acceso al SGBD en general, la 2 y la 3 para controlar las autorizaciones discrecionales y la 4 controla la autorización obligatoria

Seguridad e Integridad

Existen tres aspectos sobre seguridad en los SGBD:

La confidencialidad: se cumple cuando sólo las personas autorizadas (en su sentido amplio podríamos referirnos también a sistemas) pueden conocer los datos o la información correspondiante.

La integridad: consiste en que sólo las personas autorizadas puedan modificar o borrar los datos. Además deben quedar pistas para control posterior y para auditoría.

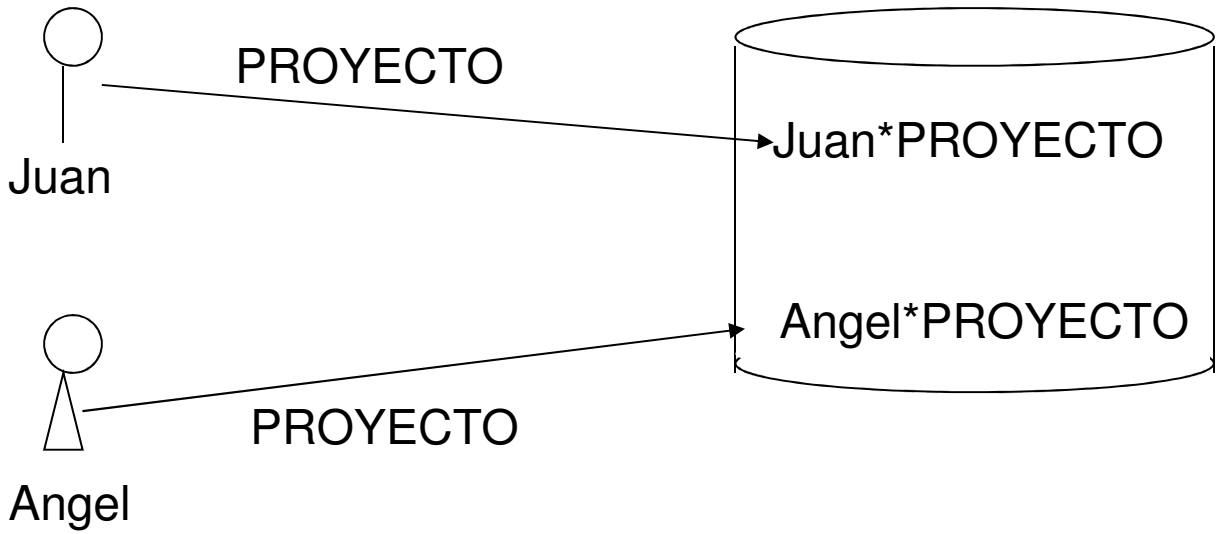
La disponibilidad: se cumple si las personas autorizadas pueden acceder a tiempo a la información.

Cuatro Técnicas para la Seguridad

1. Aquellas que controlan el **acceso a los datos**, basadas en el *password* y las cláusulas de permisología *Grant* y *Revoke.*
2. Aquellas que controlan el camino que toman los datos, llamadas **Seguridad Multinivel**.
3. Aquellas que controlan **la inferencia**.
4. Técnicas de **Criptografía**, controlan la forma en la que se almacenan y transportan los datos.

Control de Acceso a los Datos: Basado en *password.*

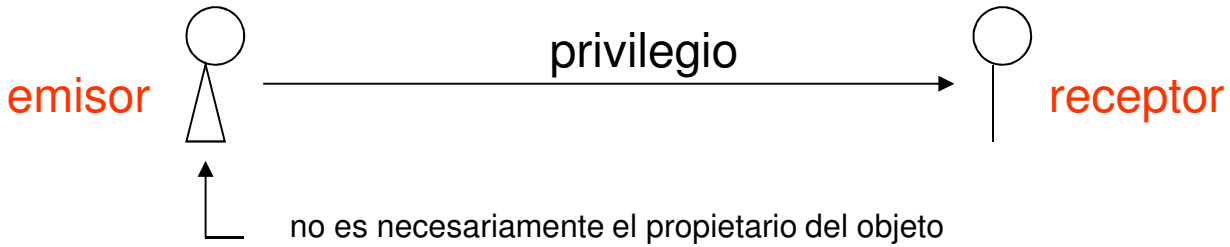
De esta forma, por medio de la autorización, el SGBD verifica que el usuario que accede un objeto realmente tiene derecho a hacerlo.



Control de Acceso a los Datos: Cláusula Grant

Cláusula de SQL que permite a un usuario autorizar a otros usuarios a tener acceso sobre los objetos que les pertenecen.

Con respecto a los problemas de asignación de recursos, se debe hablar de dos categorías de usuarios. Por un lado, se tiene el dueño del recurso que se llama el emisor y por otra parte el usuario que recibe el privilegio receptor.



Control de Acceso a los Datos: Cláusula Grant

La forma general de una cláusula Grant se presenta a continuación:

```
Grant <privilegios>
On <nombre de la relación o de la vista>
To <lista de usuarios> [public];
```

Cuando se utiliza la cláusula *public* significa que los privilegios son de orden público; es decir, se transmiten a cualquier usuario del sistema.

Control de Acceso a los Datos: Cláusula Grant

Los privilegios sobre una relación o vista son los siguientes:

- **Select:** Selecciona datos en una relación o vista.
- **Insert:** Introduce tuplas en una relación o vista.
- **Delete:** Suprime tuplas de una relación o vista.
- **Update:** Actualiza una relación y, opcionalmente, actualiza sólo las columnas especificadas en la forma (col 1, col 2, . . ., col n).
- **Index:** Crea o suprime índices de una relación.
- **Alter:** Altera una relación.
- **All:** Ejerce todos los privilegios anteriores.

Control de Acceso a los Datos: Cláusula Grant

Ejemplo: Suponer que Angel fue quien creó la tabla Empleado. Así, Angel es el propietario de esta tabla. Él quiere autorizar a Juan para que actualice e introduzca tuplas en Empleado:

```
Angel: sql> grant insert, update on Empleado to Juan
```

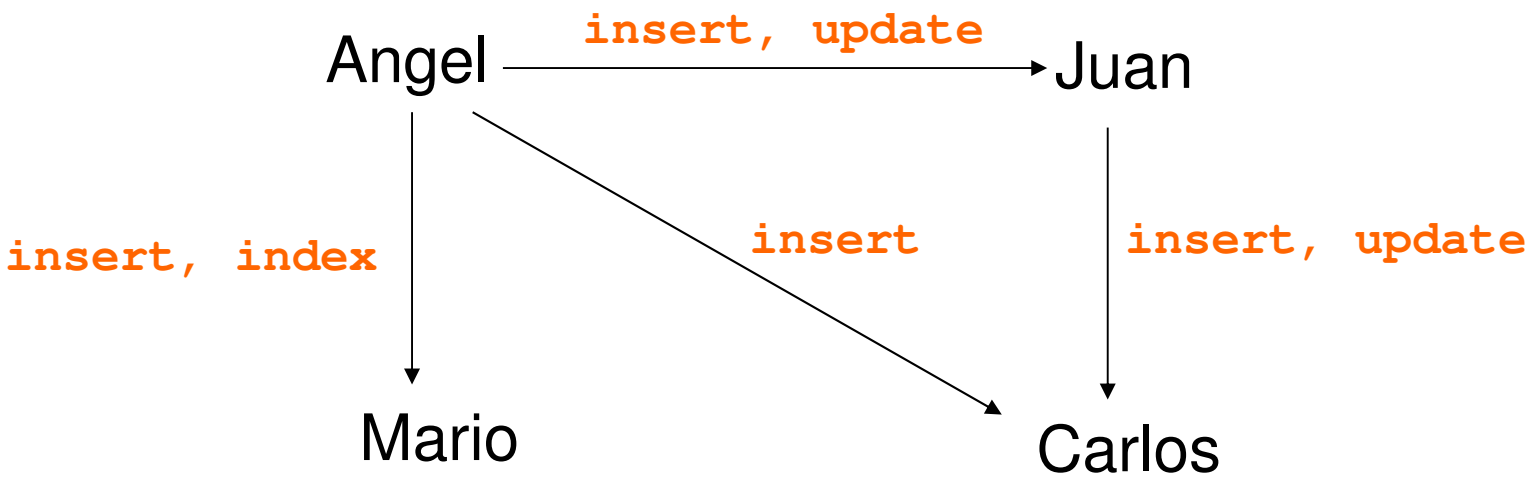
Esto significa que el usuario Juan puede transmitir los privilegios de inserción y modificación en la relación Empleado.

Control de Acceso a los Datos: Cláusula Grant

Representación mediante grafos dirigidos

Ejemplo: Suponer que, a su vez, Juan transmite ciertos privilegios a Carlos y Mario y que Angel también les transmite en forma independiente:

```
Angel: sql> grant insert, update on Empleado to Juan
Juan:  sql> grant insert, update on Empleado to Carlos
Angel: sql> grant insert on Empleado to Carlos
Angel: sql> grant insert, index on Empleado to Mario
```

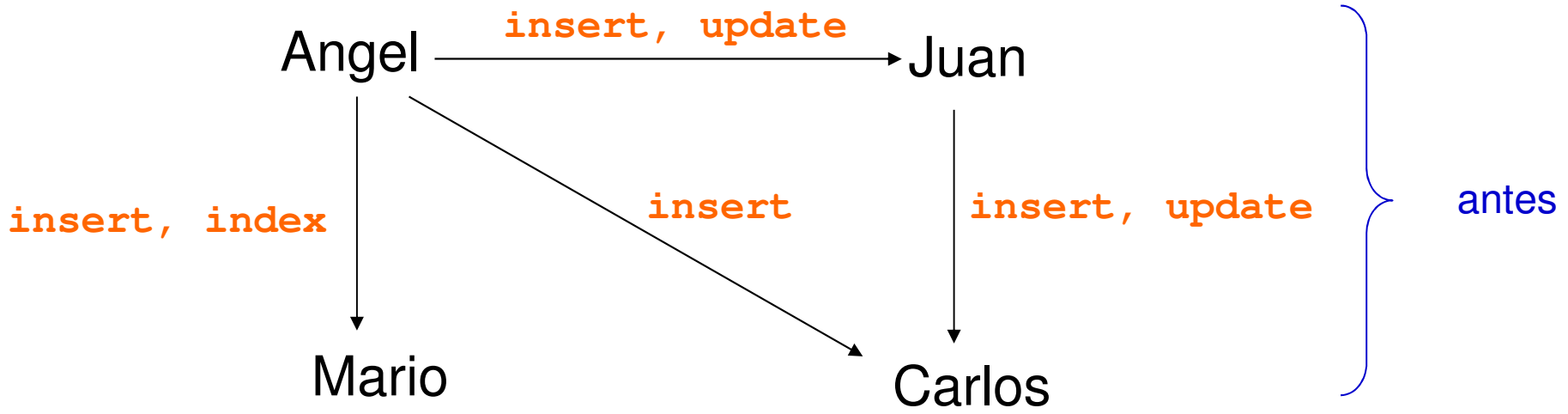


Control de Acceso a los Datos: Cláusula Revoke

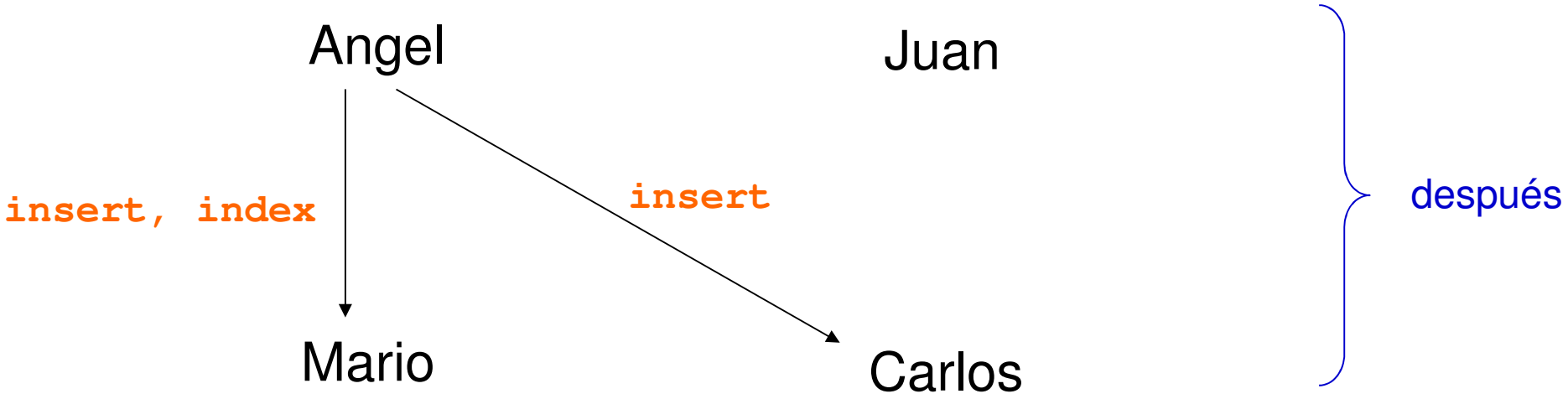
Cláusula de SQL para revocar los privilegios asignados con el Grant. Su sintaxis es:

```
Revoke <privilegios>
On <nombre de la relación o vista >
From <lista de usuarios> [public];
```

Ejemplo: Siguiendo el diagrama del ejemplo anterior, suponer que Angel decide quitarle los privilegios a Juan en la tabla Empleado.



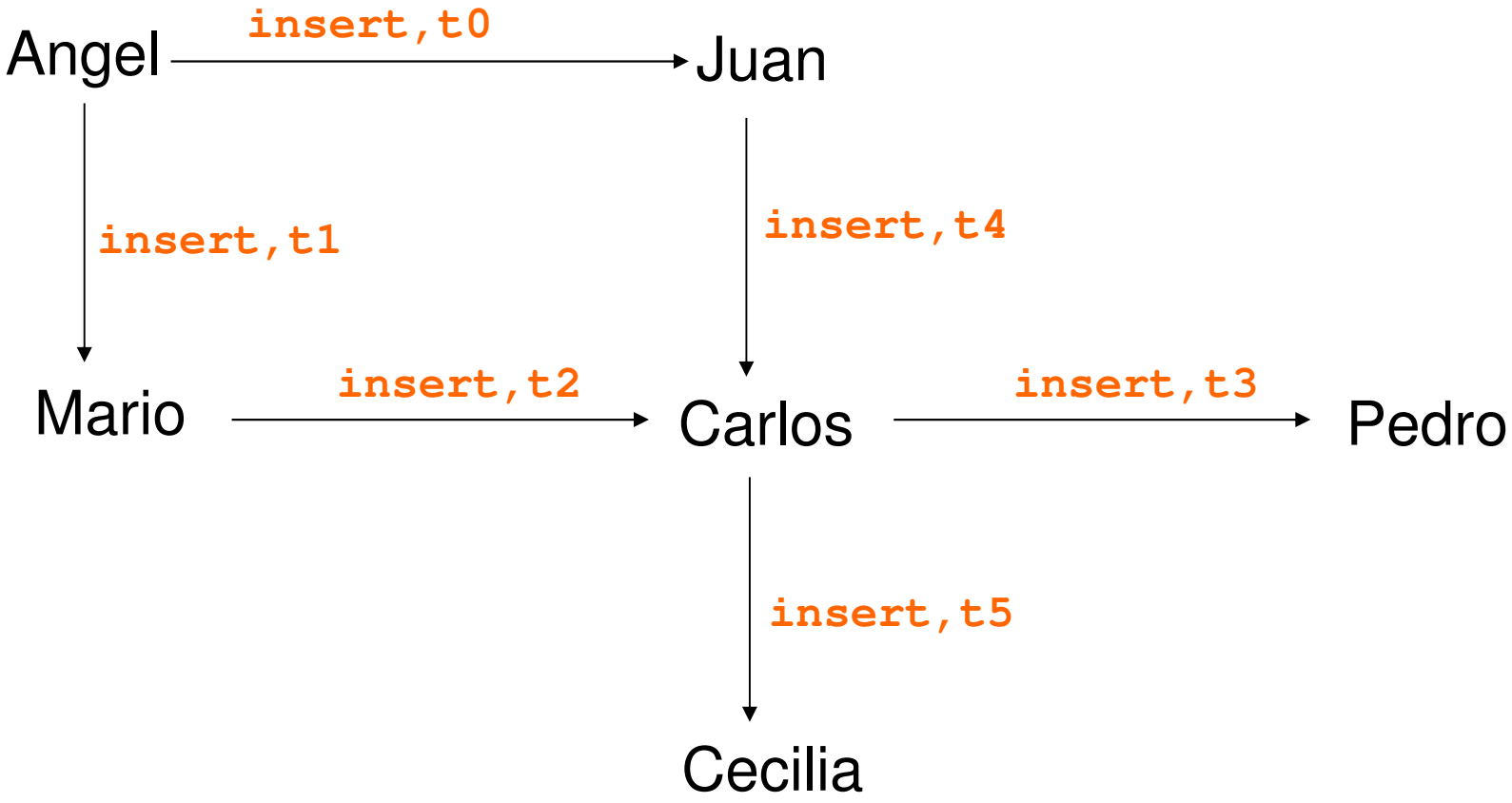

```
Angel: sql> revoke insert, update on Empleado from Juan
```



Control de Acceso a los Datos: Cláusula Revoke

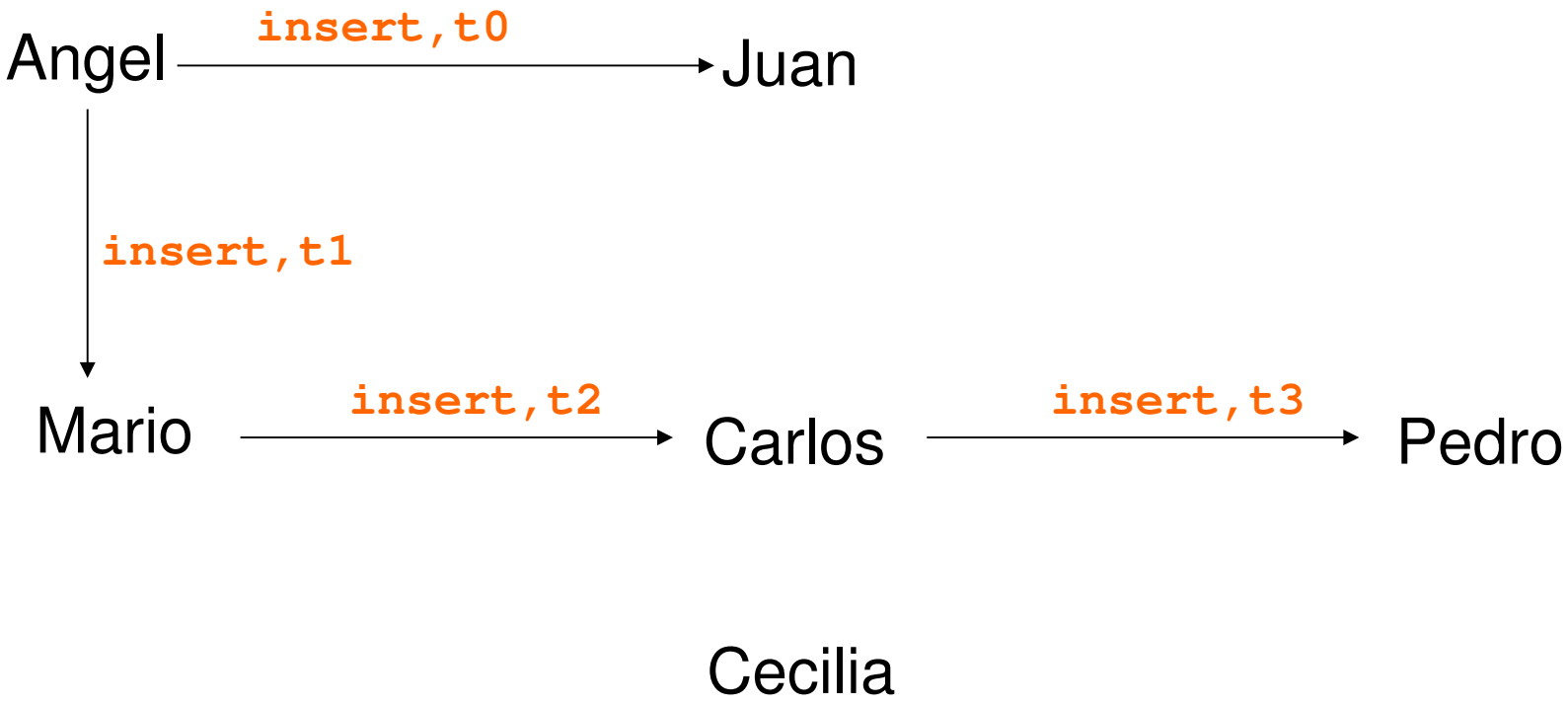
Sin embargo, el problema de la revocatoria no es tan directo como parece.

Ejemplo: Suponga el siguiente caso en el que se están representando dos trayectorias diferentes de asignación de recursos. Ahora, ¿Qué ocurriría si, por ejemplo, Juan decide quitarle el privilegio a Carlos?




```
Juan: sql> revoke insert on Empleado from Carlos
```

```
Juan: sql> revoke insert on Empleado from Carlos
```



Carlos brindó el privilegio a Cecilia y Pedro, sin embargo, sólo debe desaparecer el privilegio que ha brindado a Cecilia y no Pedro, pues esta asignación se hizo por medio de Mario y él le mantiene el privilegio a Carlos.

Control de Acceso a los Datos: Cláusula Revoke

REGLA DE REVOCATORIA

Un privilegio le es revocado a un usuario, si el emisor del privilegio lo revoca directamente o si es revocado, indirectamente, a través de una trayectoria de permisología.

Fuentes consultadas:

[1] Prof. Elsa Liliana Tovar.
Notas de clase compiladas.

[2] http://www.slideshare.net/aefeijo/seguridad-de-base-de-datos

Para crear las cuentas se procede en dos etapas: Roles y Usuarios

Instrucción Roles:

```
CREATE ROLE jefe nosuperuser;
```

Instrucciones para permiso sobre base de datos y esquema:

```
GRANT ALL on DATABASE miBD to jefe;
GRANT ALL on SCHEMA miEsquema to jefe;
```

Instrucciones para otorgar todos los permisos sobre las tablas:

```
GRANT ALL ON TABLE esquema.tabla1,
                   esquema.tabla2, …,  to jefe;
```

Instrucción para crear usuario con un role definido:

```
CREATE USER usuario1 NOSUPERUSER IN ROLE jefe PASSWORD '12345';
```