

Seguridad


Tema 17



Seguridad de la Base de Datos

(Objetivos)

- ❖ **Secreto:** A los usuarios no se les permite ver las cosas que no le corresponden.
 - Ej. Un estudiante no ve las calificaciones de otro.
- ❖ **Integridad:** Los usuarios no deben modificar las cosas que no le corresponden.
 - Ej. Solamente el catedrático coloca calificación.
- ❖ **Disponibilidad:** Los usuarios deben poder ver y modificar cosas que a les corresponden.



Controles de Acceso

- ❖ Una **política de seguridad** especifica quien esta autorizado a hacer que.
- ❖ Un **mecanismo de seguridad** permite que se haga cumplir una política de seguridad.
- ❖ Los principales mecanismos en un DBMS son los niveles de:
 - Control de acceso discrecional
 - Control de acceso obligatorio



Control de acceso discrecional

- ❖ Esta basado en el concepto de derechos de acceso o **privilegios** por objeto (tablas y vistas), y los mecanismos para dar privilegios a los usuarios (y quitar los privilegios).
- ❖ Quien crea una tabla o vista, obtiene privilegios sobre ella automáticamente.
 - DMBS lleva control de que usuario obtuvo o perdió privilegios y asegura que únicamente se atenderán las peticiones de usuarios que posean los privilegios necesarios al momento de la petición.

Comando GRANT

GRANT privileges **ON** object **TO** users **[WITH GRANT OPTION]**

- ❖ Se puede especificar los siguientes **privilegios**:
 - ❖ **SELECT**: Leer todas las columnas (incluyendo las que se agregan posteriormente usando el comando **ALTER TABLE**).
 - ❖ **INSERT(col-name)**: Permite agregar tuplas con valores.
 - ❖ **INSERT** da el mismo permiso sobre todas las columnas.
 - ❖ **UPDATE(col-name)**: Permite modificar tuplas o la columna especificada.
 - ❖ **DELETE**: Puede eliminar tuplas.
 - ❖ **REFERENCES (col-name)**: Permite definir foreign keys (en otras tablas) que hagan referencia a esta columna.
- ❖ Si un usuarios posee privilegios con **GRANT OPTION**, puede asignar privilegios sobre el objeto a otro usuario.
- ❖ Solamente el propietario puede ejecutar **CREATE**, **ALTER**, y **DROP**.



GRANT y REVOKE de Privilegios

- ❖ GRANT INSERT, SELECT ON Sailors TO Horatio
 - Horatio puede consultar e insertar tuplas en la tabla Sailor.
- ❖ GRANT DELETE ON Sailors TO Yuppy WITH GRANT OPTION
 - Yuppy puede eliminar tuplas y autorizar que otros lo hagan.
- ❖ GRANT UPDATE (*rating*) ON Sailors TO Dustin
 - Dustin puede modificar (solamente) el campo *rating* de las tuplas de Sailors.
- ❖ GRANT SELECT ON ActiveSailors TO Guppy, Yuppy
 - Hace que Guppy y Yuppy no consulten directamente la tabla Sailors!
- ❖ **REVOKE:** Cuando se quita el privilegio de X también se quita de todos los usuarios que lo obtuvieron *solamente* de X.



GRANT/REVOKE sobre Vistas

- ❖ Si el creador de una vista pierde sus privilegio de SELECT sobre una tabla, la vista se elimina!
- ❖ Dada la vista:

```
CREATE VIEW ActiveSailors (name, age, day)
AS SELECT S.sname, S.age, R.day
   FROM Sailors S, Reserves R
  WHERE S.sname=R.sname AND S.rating>6
```




Vistas y Seguridad

- ❖ Las vistas pueden ser usadas para presentar información necesaria (o resumen), ocultando detalles de la relación.
 - Con privilegio sobre ActiveSailors, pero no sobre Sailors o Reserves, es posible buscar los marineros que tienen una reservación, pero no el *bid* de los barcos que fueron reservados.
- ❖ El creador de la vista tiene un privilegio sobre la vista si tiene privilegio sobre todas las tablas referidas.
- ❖ Los comandos GRANT/REVOKE junto con las vistas son una poderosa herramienta de control de acceso.



Autorización basada en Roles

- ❖ En SQL-92, los privilegios se asignan a **id de autorizacion**, que se refiere a un usuario individual o un grupo de usuarios.
- ❖ En SQL:1999 (y muchos sistemas actuales), los privilegios se asignan a **roles**.
 - Un Rol es concedido a usuarios y otros roles.
 - Refleja como trabaja una organización



Seguridad a nivel de campos!

- ❖ Se puede crear una vista que regrese un campo de una tupla.
- ❖ Luego se concede acceso a la vista.
- ❖ Esto permite una granularidad *arbitraria* de control



Encriptación

- ❖ El DBMS puede usar encriptación para proteger información en situaciones donde los mecanismos normales de seguridad no son adecuados. Ej. Que un intruso pueda robar las cintas conteniendo datos o interceptar las líneas de comunicación.
- ❖ Al almacenar y transmitir datos en forma encriptada el DBMS se asegura que el intruso no comprenda los datos robados.