Introducción a las Bases de Datos y Bases de Datos

SEGURIDAD Y CONTROL DE ACCESO

TEORÍA 9

2017

Tecnicaturas TUPAR y TUDAI

SEGURIDAD DE LA INFORMACIÓN

Seguridad → protección de los datos contra accesos no autorizados

- La información puede estar:
  - Impresa o escrita en papel
  - Almacenada electrónicamente
  - Trasmitida por correo u otros medios electrónicos
  - En videos o conversaciones
- Debe protegerse adecuadamente cualquiera que sea la forma que tome o los medios por los que se comparta o almacene

  → **evitar brechas de seguridad**

SEGURIDAD DE LA INFORMACIÓN

Hay muchos *aspectos* relativos a la seguridad de la información:

- **Cuestiones éticas y legales** *(Ej: Qué usuarios tienen derecho legal a qué información?)*
- **Controles físicos de los equipos** *(Ej: Los equipos están resguardados?)*
- **Políticas de la organización** *(Ej: Cómo se decide quién tiene permiso a qué?)*
- **Problemas operacionales** *(Ej: Cómo mantener /resguardar las claves?)*
- **Controles de Hardware** *(Ej: El servidor provee controles de seguridad?)*
- **Soporte del Sistema Operativo** *(Ej: Operación de archivos , log de recuperación?)*
- **Cuestiones propias de la BD** *(Ej: Control de usuarios y permisos?)*

→ ***Mediante vulnerabilidades en los aspectos anteriores (no propios de la BD) se pueden "burlar" los mecanismos de seguridad de la BD***

Ej: permitiendo que un intruso por descuido ingrese a una sesión abierta, o que alguien pueda "adivinar" fácilmente una clave demasiado previsible, etc.

SEGURIDAD EN BASES DE DATOS

- **¿Cuáles son las implicaciones de los cambios o destrucciones de datos?**
  - Diferentes tipos de datos → diferentes niveles de seguridad
  - El costo de la pérdida de los datos determinará el tipo de seguridad requerida
- **¿Cuánto costarían los accesos ilegales a los datos?**
  - Si una porción de los datos tiene mucho valor para la organización, el acceso ilegal puede ser muy perjudicial → alto nivel de seguridad
- **¿Las medidas de seguridad afectarán el funcionamiento de la BD?**
  - No serán útiles los sistemas de seguridad que impidan el acceso a los datos a un usuario legítimo

SEGURIDAD EN BASES DE DATOS

- El SGBD provee un **subsistema de seguridad y autorización** de la BD
- Para acceder a la BD se requiere una cuenta de usuario y contraseña
- El **Administrador de la Base de Datos (DBA)** -posee cuenta privilegiada- debe asegurar una política de acceso clara y consistente:
  - decidir quién entra a la BD y qué puede hacer sobre los objetos a los que puede acceder **(limitado a lo que tienen acceso)**
  - garantizar la seguridad de partes de la BD contra accesos no autorizados **(sin derecho de acceso)**
  - no impedir el acceso a los datos por usuarios habilitados **(disponibilidad)**

¿Qué datos?
(restringir filas/columnas?)

¿Quiénes?
(cuáles usuarios?)

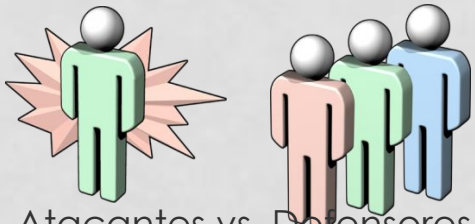
¿Qué operaciones?
(sólo consulta o modificación?)

TIPOS DE REVELACIÓN DE DATOS

- **Aspectos morales/éticos:** puede haber razones que regulen quiénes tienen acceso a determinada información *(ej. registros médicos, de antecedentes penales, etc.)*
- **Requisitos legales:** se requiere que las organizaciones usen con discreción los datos personales de los individuos
- **Seguridad comercial:** la información perteneciente a una empresa es un recurso valioso (que podría ser útil para "la competencia"!)
- Fraude/Sabotaje: la información podría ser mal utilizada por alguien malintencionado *(ej. para distorsionar los resultados de un análisis de datos o retrasar la operatoria de la organización)*
- Errores: cambios accidentales en los datos *(ej. por usuarios "descuidados")*

Precisión: proteger los datos confidenciales y revelar toda la información no confidencial que sea posible
→ secreto perfecto y máxima precisión (no es una tarea fácil)

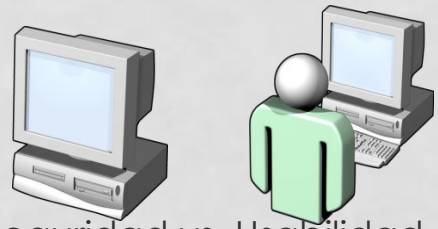
VERIFICACIÓN DE AUTENTICIDAD



Atacantes vs. Defensores

Un atacante posiblemente necesite conocer sólo UN punto de vulnerabilidad pero el Defensor tiene que asegurar TODOS los puntos de entrada!

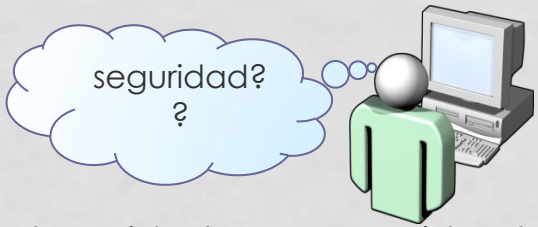
El Atacante dipone de tiempo pero el Defensor trabaja con restricciones de tiempo y costos!



Seguridad vs. Usabilidad

Los sistemas seguros son más difíciles de usar

(Ej: passwords complejas y muy elaboradas difíciles de recordar, esquema de permisos)



Securidad como una idea tardía

Los administradores y desarrolladores pueden pensar que la seguridad no añade ningún valor, pero…

→ Ocuparse de las vulnerabilidaddes antes de que un producto se comercialice puede ser caro pero necesario y beneficioso!

PROBLEMA DE LA INFERENCIA

→ Vulnerabilidad en la seguridad de la BD, que permite "descubrir" información relevante a partir de información no confidencial

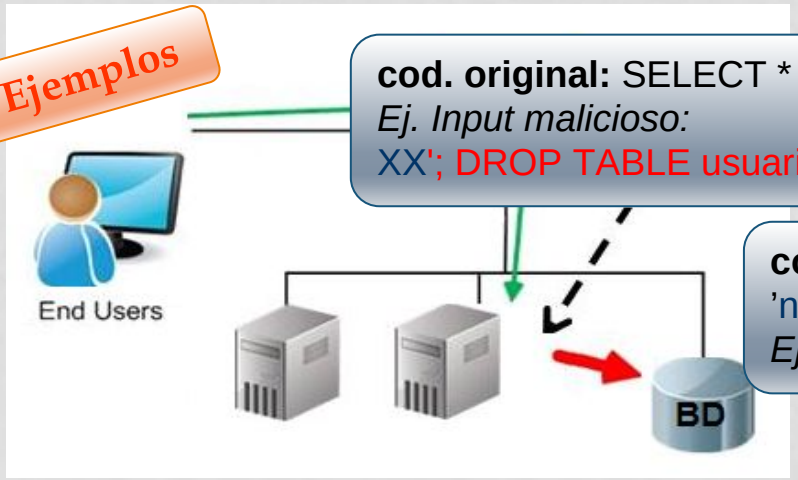
- Ataque directo
- Ataque indirecto (mediante operaciones sobre el conjunto de tuplas)
- Inyección SQL (insertar código SQL malicioso mediante una interfaz de aplicación)
- Otros

ATAQUE POR INYECCIÓN DE SQL

Se inserta –o "inyecta"– código SQL malicioso o espía en un código SQL programado a través de una interface de aplicación, usualmente web, con la intención de alterar el funcionamiento previsto para que se ejecute el código "invasor" sobre la BD

→ **burlar la identificación, alterar, borrar o extraer info de la BD** (aún datos sensibles/confidenciales)

Ejemplos



cod. original: SELECT * FROM usuarios WHERE nombre = 'nom_usuario';
Ej. Input malicioso:
XX'; DROP TABLE usuarios; SELECT * FROM datosConf WHERE nombre LIKE '%

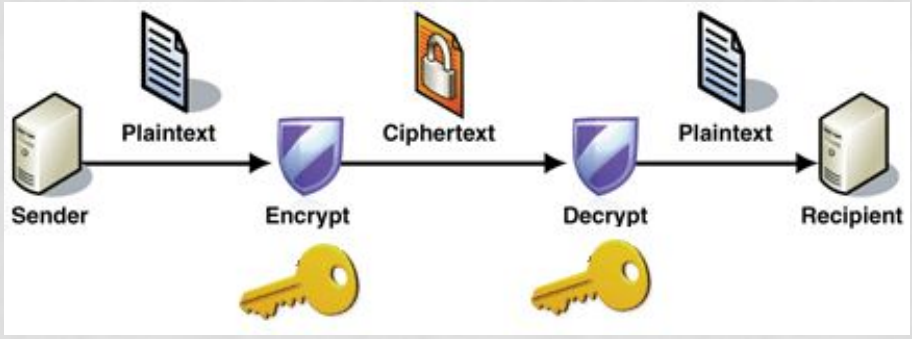
cod. original: SELECT * FROM usuarios WHERE nombre= 'nom_usuario' AND password= 'passwd';
Ej. Input malicioso: cualquiera y password' OR 'a'='a

- *Problema:* La aplicación no debería incorporar el input del usuario directamente en una sentencia SQL → vulnerabilidad!
- *Desafío:* se debe verificar lo ingresado por el usuario para asegurar que sean datos válidos (s/normas de contenido) y que no se ha inyectado código adicional
- Los frameworks y lenguajes de desarrollo generalmente proveen funciones de control

SEGURIDAD A CARGO DEL SGBD

Otros mecanismos para garantizar seguridad sobre la información:

- SEGUIMIENTO DEL 'RASTRO' (*Audit Trail*):
  - El SGBD puede registrar quién entra en la BD, a qué datos accedió y qué operaciones hizo sobre ellos (*LOG o bitácora del sistema*)
- CIFRADO o ENCRIPTACIÓN de datos:
  - Los datos se codifican mediante algoritmos particulares, en función de una clave (*privada*) o dos claves (*privada-pública*)
  - los datos resultan ilegibles a menos que se tenga conocimiento del código usado y la clave correspondiente



CONTROL DE ACCESO A LA BD

Componentes

- Política: especifica la forma de autorizar accesos
- Mecanismo: implementa la política de asignación de autorizaciones

Granularidad: La protección de los objetos depende de su tamaño o extensión (ej: tupla, relación, atributo, elemento, BD)

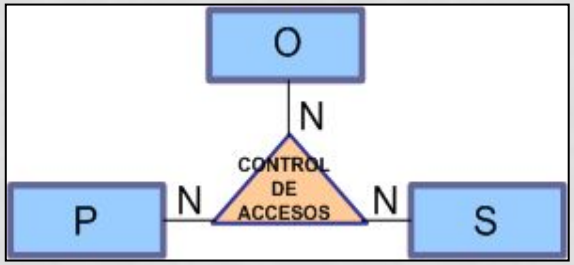
(S)ujeto: entidad activa que requiere acceso a un objeto

- (ej. usuario o programa)

(O)bjeto: entidad pasiva accedida por un sujeto

- (ej. registro, relación, índice, archivo)

(P)rivilegio o derecho de acceso: cómo un sujeto puede acceder o manipular un objeto (consulta/modificación/borrado/inserción)



MÉTODOS DE CONTROL DE ACCESO

- Control de Acceso Discrecional:

  garantiza privilegios a usuarios, incluyendo la capacidad para acceder archivos de datos específicos, registros o campos para operar de una manera determinada (read, insert, delete, update, otras…).

- Control de Acceso basado en Roles:

  establece grupos de privilegios encapsulados en un rol que se otorgan a usuarios

- Control de Acceso Mandatorio

  clasifica usuarios y datos en múltiples niveles de seguridad y luego fuerza determinadas reglas acordes a cada nivel

CONTROL DE ACCESO DISCRECIONAL

El acceso a la BD se basa en otorgar y revocar privilegios sobre los objetos de la BD, dando acceso selectivo a otros usuarios (a discreción):

→ concesión de derechos (GRANT)

→ revocación de derechos (REVOKE)

GRANT privilegio/s ON objeto/s TO usuario/s [WITH GRANT OPTION]

REVOKE [GRANT OPTION FOR] privilegio/s ON objeto/s FROM usuario/s {CASCADE | RESTRICT}

- **privilegio/s: derecho/s para acceder o ejecutar un procedimiento SQL - operación/es de acceso a los datos** (puede ser uno o varios)
- **objeto/s: tablas, vistas, índices, etc.** (uno o varios)
- **usuario/s: nombre de usuario que la BD reconoce como autorizado para acceder a la BD** (uno o varios) **o PUBLIC** (=todos los usuarios)
- **WITH GRANT OPTION permite que el sujeto poseedor de privilegios pueda transmitirlos a otros usuarios**

CONTROL DE ACCESO DISCRECIONAL

NIVELES DE ASIGNACIÓN DE PRIVILEGIOS

Nivel de cuenta

se puede especificar privilegios particulares a cada usuario, independientemente de las relaciones de la BD

Ejemplos

CREATE SCHEMA, CREATE TABLE, CREATE VIEW, ALTER, DROP, … (según el SGBD), el propietario puede ejecutar CREATE, ALTER, y DROP

CONTROL DE ACCESO DISCRECIONAL

NIVELES DE ASIGNACIÓN DE PRIVILEGIOS

Nivel de relación: se puede controlar el privilegio para acceder a cada relación (tabla o vista) individual de la BD

- **SELECT**– leer todas las columnas (incluyendo las que se añadan con ALTER TABLE)
- **DELETE** – remover datos
- **INSERT** (columna/s) – incorporar nuevas tuplas con valores no nulos (o no default) en esa/s columna/s. Sin ( )– ídem para todas las columnas
- **UPDATE**– análogo a INSERT para modificar datos existentes
- **REFERENCES** (columna/s) – definir FK referidas a esa/s columna/s. Sin ( )– ídem para todas las columnas

El propietario de un objeto posee todos los privilegios sobre el objeto, capacidad de concederlos (GRANT), y además con GO

CONTROL DE ACCESO DISCRECIONAL

- **VISTAS (nivel externo de la BD)** : definen particiones horizontales (selecciones) y verticales (proyecciones)

  → **importante mecanismo de autorización discrecional** para forzar seguridad sobre los datos

  - permite ver ciertas partes de la BD a los usuarios y oculta otras
  - se puede limitar o impedir el acceso a datos sensibles o confidenciales

- Para crear una Vista, la cuenta debe tener permiso SELECT para todas las relaciones que intervienen en la definición de la Vista
- Si se pierde el privilegio SELECT sobre alguna de las relaciones base de Vista → la Vista ya no es válida (no puede utilizarse)
- Si se elimina una Vista (u otro objeto) →se revoca automáticamente todo privilegio sobre el objeto

CONTROL DE ACCESO DISCRECIONAL

PROPAGACIÓN DE PRIVILEGIOS :

- Si el propietario A de una relación R desea otorgar a otra cuenta B un cierto privilegio para R (o si posee ese privilegio), puede hacerlo:
  1. con la opción de propagar ese privilegio (WITH GRANT OPTION=GO)
  2. o sin esa posibilidad

GRANT privilegio/s ON objeto/s TO usuario/s [WITH GRANT OPTION]

→ usuario = nombre/s de usuario específico/s | PUBLIC

- Se puede propagar un mismo privilegio a más de un usuario
- Se puede recibir un mismo privilegio de más de un usuario

CONTROL DE ACCESO DISCRECIONAL

Se puede REVOCAR un privilegio previamente otorgado (o la opción de propagarlo):

```
REVOKE [GRANT OPTION FOR] privilegio/s ON objeto/s
  FROM usuario/s   {CASCADE | RESTRICT}
```

- Si se especifica **GRANT OPTION FOR** se quita la posibilidad de propagar el privilegio (pero no cancela el privilegio sobre el objeto), de lo contrario se revoca el privilegio sobre el objeto en sí
- si se ejecuta REVOKE con opción CASCADE, el efecto es revocar el privilegio al usuario y a todos los que lo recibieron a través de él (privilegios *colgados*)

  → el SGBD deberá llevar una pista de *concesión de privilegios*

  → algún usuario puede seguir conservándolo si también recibió ese privilegio de otro
- si se indica RESTRICT, se rechazará si el efecto de revocación de privilegios provocaría privilegios *colgados*

CONTROL DE ACCESO DISCRECIONAL

Ejemplos

GRANT INSERT, SELECT, DELETE ON Hotel TO U1;

→ U1puede insertar, borrar y seleccionar tuplas de la tabla Hotel

GRANT DELETE, INSERT ON Hotel TO U2 WITH GRANT OPTION;

→ U2 puede insertar y borrar tuplas de la tabla Hotel (y propagar los privilegios a otros)

GRANT UPDATE (nro_estrellas) ON Hotel TO U3;

→ U3 puede actualizar solamente el campo nro_estrellas de las tuplas de Hotel

GRANT SELECT ON VistaHoteles3Estrellas TO U4;

→ U4 pueden consultar los datos sobre hoteles 3 estrellas de la vista (pero NO pueden consultar directamente la tabla Hotel)

✓ Se puede generar un **Grafo de Autorizaciones** con los privilegios concedidos:

- Nodos → usuarios
- Arcos (dirigidos) → concesión de privilegio (sobre qué objeto), y si tiene GO

CONTROL DE ACCESO DISCRECIONAL

Ejemplos

A es propietario del esquema BD_EJ y crea en él las tablas T1 y T2 (privilegio de cuenta) - (El DBA antes hizo: CREATE SCHEMA BD_EJ AUTHORIZATION A;)

A: GRANT SELECT ON T1, T2 TO **B**;
→ B puede seleccionar tuplas de T1 y T2 (sin posibilidad de propagarlo)

A: GRANT SELECT ON T1, T2 TO **C** WITH GRANT OPTION;
→ C puede seleccionar tuplas de T1 y T2 (y puede propagar ese privilegio)

C: GRANT SELECT ON T1 TO **B, D**;
→ B y D reciben el priv. de seleccionar tuplas de T1 (pero no de propagar el privilegio)

A: REVOKE GRANT OPTION ON T2 FROM **C**;
→ C ya no puede ceder el privilegio de selección sobre T2 (pero lo conserva)

A: REVOKE SELECT ON T1 FROM **C** CASCADE;
→ C pierde el privilegio de selección sobre T1
y esto se propaga en cascada a B y D
pero B había recibido también el privilegio directamente de A (lo sigue conservando)

¿Qué privilegios conserva cada usuario entonces? (*analizar grafo*)

CONTROL DE ACCESO DISCRECIONAL

El esquema de concesión y revocación de privilegios queda representado por un acceso a una MATRIZ DE ACCESO (Este modelo se utiliza también en Sistemas Operativos)

	Objetos (relaciones, columnas, ...)				
sujetos (usuarios)	O_1	...	O_j	...	O_m
S_1	$A[s_1, o_1]$		$A[s_1, o_j]$		$A[s_1, o_m]$
...	...		...		...
S_i	$A[s_i, o_1]$		$A[s_i, o_j]$		$A[s_i, o_m]$
...	...		...		...
S_n	$A[s_n, o_1]$		$A[s_n, o_j]$		$A[s_n , o_m]$

- **$A[S_i,O_j]$** contiene un conjunto de privilegios otorgados al sujeto **S_i** sobre el objeto **O_j**.
- Puede contener también una señal que indique se pueden pasar los privilegios a otro sujeto , ej. (d) para indicar que el sujeto **S_i** tiene derecho de borrado sobre el objeto **O_j** y (**d+**) si puede ser transferido a otro sujeto

CONTROL DE ACCESO DISCRECIONAL

GRANT/REVOKE EN VISTAS:

- El creador de una vista tiene privilegios sobre la vista si los tiene sobre todas las tablas subyacentes
- Si el creador de una vista pierde el privilegio SELECT sobre alguna de las tablas subyacentes, la vista es removida!
- Si el creador de una vista pierde un privilegio obtenido con WITH GRANT OPTION sobre una relación subyacente, también pierde el privilegio sobre la vista (lo mismo ocurre con los demás usuarios que hayan obtenido el privilegio sobre la vista)

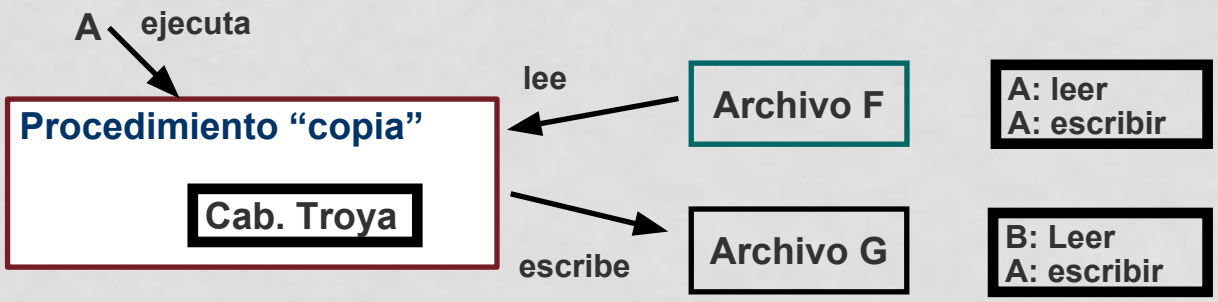
Asignar derechos para ejecutar programas compilados:

- GRANT EXCECUTE ON <procedimiento> TO <usuario>
- Problema: los procedimientos(stored procedures) podrían acceder a recursos para los cuales el usuario no tiene permisos de acceso

CONTROL DE ACCESO DISCRECIONAL

"Problema del CABALLO DE TROYA":

- Es software malicioso instalado (posiblemente sin intención) por usuarios debidamente autorizados - Ejecuta lo que el usuario pretende, pero explota los privilegios del usuario para provocar una *brecha en la seguridad*
- Ejemplo:
  - A puede leer y escribir en el archivo F, y escribir en el archivo G (no puede leer de G)
  - B sólo puede leer del archivo G (no está autorizado a escribir F ni G, ni leer de F)



→ ahora B puede leer contenidos de F (copiados a G)!

CONTROL DE ACCESO BASADO EN ROLES

USUARIOS

ROL TESORERO

ROL ADMINISTR

ROL RRHH

ROLES DE USUARIOS

ROL GESTIÓN-PAGOS

ROL GESTIÓN-PLANTEL

ROLES DE APLICACIÓN

Privilegios para manejar pagos y cuentas

Privilegios para manejar altas, bajas de personal, etc.

PRIVILEGIOS DE APLICACIÓN

CONTROL DE ACCESO BASADO EN ROLES

- **Rol** Se define como el conjunto de privilegios o derechos de acceso
  - Si se cambian los privilegios encapsulados en un rol, los privilegios de todos los usuarios que tienen ese rol también cambian
  - En SQL:1999 los privilegios son asignados a roles (muchos sistemas actuales han adherido a este enfoque)

hay varias opciones (según el SGBD)

```
CREATE ROLE <nom_rol> [ WITH option ] ;

GRANT nom_rol [{,<nom_rol> }] TO <a_quien> [{,<a_quien>}]
      [ WITH ADMIN OPTION ] ;
```

  - **a_quien** indica usuarios / otros roles / PUBLIC (todos)
  - un usuario puede tener asignado a uno o más roles
  - Rol especial: **ADMIN** (tiene privilegios como: *create role* y *drop role*)
  - **WITH ADMIN OPTION** indica que se puede conceder el rol a otros usuarios/roles

Ej: CREATE ROL RR; GRANT CREATE TABLE TO RR; GRANT RR TO user1;

CONTROL DE ACCESO BASADO EN ROLES

- **Para revocar un Rol**

REVOKE [ADMIN OPTION FOR] nom_rol [{, nom_rol}]
FROM <*a_quien*> [{,<*a_quien*>}] ;

a_quien= usuario | otro rol |PUBLIC

Revoca la pertenencia a uno o más roles de uno o más usuarios/roles

- No se pueden revocar los privilegios del propietario de un objeto
- Si los privilegios sobre un objeto fueron conseguidos a través de un rol (el objeto depende de tal rol) → el objeto se elimina si el rol es revocado

Ej: REVOKE CREATE TABLE FROM RR;

REVOKE ADMIN OPTION FROM RR;

→ pierde la posibilidad de ceder el rol

CONTROL DE ACCESO BASADO EN ROLES

Algunos de los privilegios concedidos a las funciones del sistema:

System Role	Privileges Granted to the Role
CONNECT	CREATE TABLE, CREATE VIEW, CREATE SYNONYM, CREATE SEQUENCE, CREATE SESSION etc.
RESOURCE	CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER etc. The primary usage of the RESOURCE role is to restrict access to database objects.
DBA	ALL SYSTEM PRIVILEGES

CONTROL DE ACCESO MANDATORIO

- Cada objeto de la BD tiene asignada una clase de seguridad → *seguridad multinivel*
- Cada sujeto (usuario o programa) tiene asignado un permiso para una clase de seguridad
- Basado en estrategias de la organización, no pueden ser modificados por los usuarios individualmente
- Existen reglas que habilitan/prohíben lecturas/escrituras en la BD, según combinaciones específicas de clases de seguridad y permisos
- La mayoría de los DBMSs actuales no soportan este control
  Algunas versiones lo hacen para aplicaciones específicas (*ej. Defensa, Espionaje, …*)

CONTROL DE ACCESO MANDATORIO

Modelo Bell-LaPadula (BLP)

asigna a **cada Sujeto** (usuario, cuenta, programa de usuario)

y Objeto (tabla, vista, tupla, columna,…)

un **nivel de Seguridad**:

Top secret (TS), secret (S), confidential (C), unclassified (U)
(u otra clasificación definida)

- **Objetos**: tienen una clasificación de seguridad
  Ej: Archivo Sueldos es S, Archivo Empleados es C, …

- **Sujetos**: tienen habilitaciones de seguridad
  Ej: U1 habilitado como S, U2 habilitado como C, …

- **Dominancia: TS > S > C > U**

CONTROL DE ACCESO MANDATORIO

- **Derecho de acceso:** según comparación entre la clasificación del objeto requerido respecto del nivel de habilitación del sujeto
- se permite acceso → si se satisface la regla de control de acceso:

El Sujeto SUJ puede leer el objeto OBJ sólo si

clase(SUJ) >= clase(OBJ) - Propiedad de Seguridad Simple *(protege la información contra lecturas no autorizadas)*

El Sujeto SUJ puede escribir el objeto OBJ sólo si

clase(SUJ) <= clase(OBJ) - Propiedad * *(protege los datos contra contaminación o modificaciones no autorizadas)*

→ "no read up, no write down"

→ un SUJ no puede leer un OBJ con clasificación más alta que la que él posee

→ un SUJ no puede escribir un OBJ que tenga clasificación menor que la que él posee (*es poco intuitiva, pero así impide el flujo directo de información de niveles de seguridad altos a más bajos*)

CONTROL DE ACCESO MANDATORIO

SEGURIDAD MULTINIVEL

- Capacidad de manejar información de diferente 'sensibilidad', *desde información clasificada hasta información de libre circulación*
- Permitir el acceso simultáneo a la BD por usuarios con **diferentes habilitaciones y necesidades** de conocimiento de información
- **Prevenir el acceso a información para la cual no se tiene habilitación. El control de acceso mandatorio tiene como objetivo la confidencialidad.**

Id_Hotel	estrellas	NombreHotel	**Clase**
102	3	Excelence	**C**
101	5	Optimus	**S**

Relaciones Multinivel

Usuarios con habilitación S y TS pueden ver ambas filas; si tienen habilitación C sólo ven la primera y si tienen habilitación U no ven ninguna

CONTROL DE ACCESO MANDATORIO

Problema: un usuario con permiso C trata de insertar <101, 4, Star, C>:

- Si se permite → viola la restricción de clave
- No es aceptable que dos tuplas tengan el mismo valor para los atributos que forman parte de la clave primaria

***Opciones*:**

1. Informar al usuario que ya existe una tupla con la misma clave primaria → se genera un '**canal oculto**', o sea que el rechazo de la petición **revela información secreta** !
2. Reemplazar la tupla anterior por la nueva → problemas de integridad (la aceptación de la petición ocasiona la **pérdida de información secreta!)**
3. **Inserción de otra tupla con la misma clave primaria + clase de seguridad** (asumiendo los problemas de administración de datos que implica)
   → pueden coexistir varias tuplas con información relativa al mismo 'objeto' con distinto nivel de clasificación → **polinstanciación**

GUÍAS DE SEGURIDAD PARA EL SGBD

Funcionalidad

- Usar la menor cantidad de protocolos de comunicación posible
- Eliminar del sistema procedimientos innecesarios o inútiles
- Desabilitar login por defecto y usuarios invitados hasta donde sea posible
- No permitir a todos los usuarios que se logueen interactivamente (consola), a menos que se requiera

Planeamiento

- Desarrollar un plan de seguridad para prevenir y detectar problemas
- Crear procedimientos/protocolos para emergencias de seguridad y practicarlos

GUÍAS DE SEGURIDAD PARA EL SGBD

Es aconsejable:

- Ejecutar el SGBD detrás de un *firewall*, pero planificar la aplicación como si no estuviese activo.
- Proteger el/los equipo/s sobre el/los que corre el SGBD
  - Ubicar los equipos con el SGBD en ambientes físicamente seguros
  - No permitir que los usuarios "no-DBA" lo utilicen
  - Se debería registrar el acceso en un log (diario o bitácora del sistema)
- Manejar cuentas y passwords
  - Usuarios con privilegios adecuados para el servicio del SGBD
  - Proteger las cuentas de la BD con passwords **MUY** seguras
  - Llevar auditoría de intentos fallidos a la BD (log)
  - Chequeo de usuarios y grupos o roles
  - Limitar los privilegios para la cuenta del DBA y asignar a los otros usuarios/roles la menor cantidad posible de privilegios

SEGURIDAD EN LAS APLICACIONES

Si las características de seguridad del SGBD fuera inadecuadas se debe introducir código adicional en los programas de aplicación.

Cuanto más cercanos sean los mecanismos de seguridad a los datos, tanto menos posible será la infiltración.

La seguridad en las aplicaciones web a menudo están provistas en el servidor web

- … **sin embargo es preferible usar primero las del SGBD**
- Las características de seguridad del SGBD generalmente son rápidas, baratas y ofrecen resultados de mayor calidad que las que un usuario pueda desarrollar