Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería

Cátedra: **Bases de datos I**

Seguridad e Integridad

- Introducción

  - Para asegurarnos que una BD esté en un estado correcto debemos controlar tanto la **seguridad** como la **integridad** de la misma.
  - ✓ **Seguridad de la BD** se refiere a la protección a los accesos indebidos
  - ✓ **Integridad** se refiere a la prevención de la pérdida accidental de la consistencia.

  - El aspecto global de seguridad de datos está muy vinculado al propio concepto de lo que es una BD: "Un conjunto de datos integrados, adecuado a **varios usuarios y a diferentes usos**".

  - La información almacenada en la BD debe estar protegida contra:
    - ✓ Accesos no autorizado
    - ✓ Destrucción o alteración malintencionada
    - ✓ Introducción accidental de inconsistencia.

- Violaciones de la seguridad y la integridad

  - El mal uso que se haga de la BD puede ser intencional o accidental. La pérdida accidental de la consistencia de los datos puede deberse a:
  - ✓ Caída durante el procesamiento de las transacciones
  - ✓ Anomalías por acceso concurrente a la base de datos
  - ✓ Anomalías que resultan de la distribución de los datos entre varias computadoras
  - ✓ Un error lógico que viola la suposición de que las transacciones respetan las restricciones de integridad de la BD.

  - Es más sencillo prevenir la pérdida accidental de consistencia de los datos que el acceso malintencionado a la BD. Algunas formas de acceso indebido son:
  - ✓ Lectura de datos sin autorización
  - ✓ Modificación no autorizada de los datos
  - ✓ Destrucción no autorizada de los datos

  - No es posible proteger de manera absoluta a la BD contra un manejo indebido, pero se pueden tomar todas las precauciones, para elevar el costo de acceso a las personas que quiera acceder a la base sin la autorización adecuada.

- Seguridad

  – En la protección es necesario adoptar medidas en los siguientes niveles:

  ✓ **Físico**: proteger contra la entrada de intrusos al local donde se encuentra el equipo.
  ✓ **Humano**: tener cuidado al conceder autorización a los usuarios para reducir la probabilidad de que un usuario autorizado permita el acceso a un intruso a cambio de sobornos.
  ✓ **Sistema Operativo**: independiente de lo seguro que pueda ser el SGBD, la debilidad de la seguridad del sistema operativo puede servir para obtener acceso sin autorización a la BD.
  ✓ **Red**: dado que casi todos los sistemas de bases de datos permiten el acceso remoto, la seguridad a nivel de software de la red es tan importante como la seguridad física, tanto en Internet como en redes privadas de las empresas.
  ✓ **Sistema de base de datos**: es responsabilidad del sistema de base de datos asegurar que no se violen las restricciones de autorización.

- Autorización y vistas

- Autorizaciones

  – Los usuarios pueden tener varios tipos de autorización para diferentes partes de una misma base de datos. Entre ellas están las siguientes:

  ✓ **Autorización de lectura**: permite la lectura de los datos pero no su modificación.
  ✓ **Autorización de inserción**: permite insertar datos nuevos, pero no modificar los existentes.
  ✓ **Autorización de actualización**: permite modificar los datos pero no borrarlos.
  ✓ **Autorización de borrado**: permite el borrado de los datos.

- Un usuario puede recibir todos los tipos de autorización, una combinación de algunos de ellos o ninguno. Existen autorizaciones para modificar el esquema de las bases de datos:

- ✓ **Autorización de índices**: permite la creación y el borrado de índices.
- ✓ **Autorización de recursos**: permite la creación de relaciones nuevas.
- ✓ **Autorización de alteración**: permite añadir o borrar atributos de las relaciones.
- ✓ **Autorización de eliminación**: permite borrar relaciones.

- Las autorizaciones de eliminación y borrado se diferencian en que el borrado permite borrar tuplas de una relación.
- Si un usuario borra tuplas de una relación, ésta sigue existiendo, si borra todas las tuplas también sigue existiendo, pero está vacía.
- Si se elimina una relación, deja de existir en la base.
- La autoridad máxima es la del administrador. Es quien otorga las autorizaciones.

- Vistas
  - Una vista puede ocultar datos que el usuario no necesita conocer. Esto permite tanto simplificar el sistema como mejorar su seguridad.
  - El sistema se simplifica porque se permite al usuario estar atento solo a los datos que son de su interés. Aunque puede que se le niegue el acceso a una relación se le puede permitir el acceso a parte de esa relación.
  - Lo normal es que las BD cuenten con dos niveles de seguridad a nivel relación y a nivel vista.
  - Supongamos el siguiente ejemplo: un empleado que necesita una nómina de clientes que poseen préstamos en cada sucursal pero no deben conocer sus importes. (La relación PRESTAMO tiene los siguientes atributos: nombre_sucursal, número_prestamo, importe)

```
create view cliente_prestamo as
          (select nombre_sucursal, nombre_cliente
           from cli/pres c, préstamo p
where p.número_prestamo = c.número_prestamo)
```

- Actualización de vistas:

  - El tema de actualización de vistas ha sido y sigue siendo un tema de investigación, y tiene varios enfoques:
  - ✓ Algunos enfoques requieren que el ABD (Administrador de Base de Datos) especifique qué actualizaciones se permiten en cada vista y cuál es el código para implementar la actualización en la base subyacente.
  - ✓ Otros son más generales, proponen indicar un mecanismo que traslade a la BD la actualización realizada sobre la vista.
  - ✓ Proponen criterios según la operación de actualización que se haga sobre la vista.
  - ✓ Otros enfoques indican que todas las vistas que son teóricamente actualizables se pueden actualizar por el sistema. El problema es determinar cuáles son las vistas teóricamente actualizables, ya que no está muy claro. Cada sistema puede hacer unas suposiciones particulares sobre las vistas que son actualizables.

  - La mayoría de las SGBD permiten actualizar vistas simples, pero deshabilitan los intentos de actualizar vistas complejas.

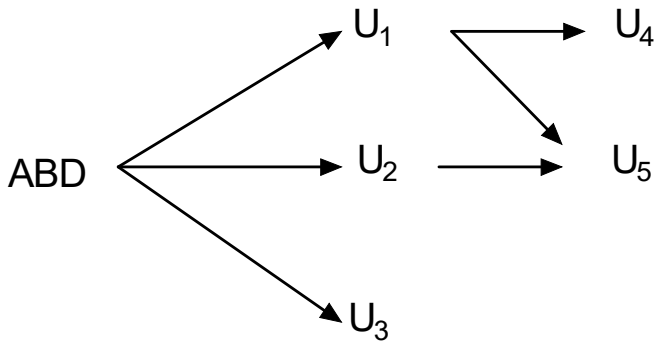
- Actualización de vistas en SQL

  - Permite actualizar vistas y trasladar los cambios a la relación, si la vista deriva de una sola tabla. Si se insertan datos en una vista, se insertan en la relación correspondiente de la base de datos, que si tuviera más atributos se completa con null.

- Concesión de privilegios

  - El usuario que recibe alguna autorización puede tener permiso para transmitir esa autorización a otros usuarios. Hay que tener cuidado con el modo de transmitir las autorizaciones entre los usuarios para asegurar que pueda retirarse en el futuro.

  - Supongamos que el ABD concede autorización de actualización sobre la relación PRESTAMO, a los usuarios U1, U2 y U3. Éstos a su vez pueden transmitirla a otros usuarios. La transmisión de autorizaciones de un usuario a otro puede representarse mediante un **grafo de autorización**, los nodos de los grafos son los usuarios, la raíz del grafo es el ABD.

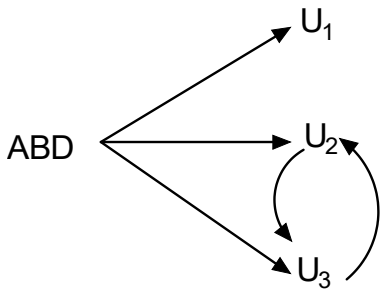


- Vemos que el ABD da la autorización de actualización sobre la relación PRESTAMO a los usuarios U_1, U_2 y U_3. Observemos que U_1 y U_2 conceden la autorización a U_5, mientras que U_1 se la concede también a U_4.

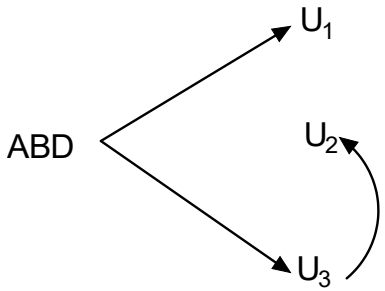
- Un usuario tiene autorización sólo si hay un camino desde la raíz del grafo de autorización hasta el nodo correspondiente.

- Si el ABD decide retirar la autorización al U_1. Dado que U_4 tiene autorización sólo de él, pierde su autorización, U_5 en cambio sigue con autorización ya que permanece la autorización que le concedió U_2.

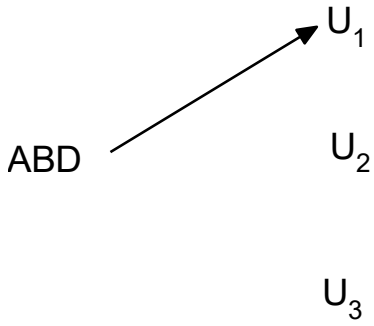
- Podemos pensar que algunos usuarios intenten eludir la retirada de autorizaciones, lo podemos ver a través del gráfico siguiente:



– Los U_2 y U_3 se conceden la autorización mutuamente, si el ABD decide retirar la autorización a U_2, éste la conserva mediante U_3.



– Si a continuación se retira la autorización a U3 , ambos quedan sin autorización.



- Especificación de la seguridad en SQL

  - SQL maneja diferentes niveles de seguridad, en primer lugar se encuentra la validación con el Sistema Operativo (Autenticación), luego la validación en el servidor de BD (Inicio de sesión), le sigue el acceso a la base de datos (nombre de usuario) y por ultimo los permisos de acceso a los objetos.

  - **Autenticación de SO**: De este tipo de autenticación se encarga directamente el Sistema Operativo, el usuario debe contar con una cuenta en el servidor con los derechos para utilizar el SQL Server.

  - **Inicios de sesión para SQL Server**: Un Inicio de sesión es el objeto encargado de permitir o denegar el acceso, según el perfil del usuario.

  - Cuando agregamos un inicio de sesión, podemos definirle un grupo de funciones, que tomarán los usuarios de ese inicio de sesión.

Función	Descripción
dbcreator	Permite crear y modificar bases de datos.
diskadmin	Permite administrar los archivos del disco.
processadmin	Permite administrar los procesos ejecutados en el servidor.
securityadmin	Permite administrar los inicios de sesión, permisos CREATE DATABASE y leer los registros de errores.
serveradmin	Permite configurar y apagar el servidor.
setupadmin	Permite administrar servidores vinculados y procedimientos de inicio
sysadmin	Permite pleno acceso sobre SQL Server.

- **Administración de Usuarios**: Una vez creado un inicio de sesión, debemos definir los usuarios que se conectarán al servidor utilizando ese inicio de sesión. Para administrar a los usuarios, por ejemplo creamos al usuario Pedro con un nombre de inicio de sesión INICIOPEDRO.

- Tabla de Funciones y Permisos

Función	Descripción
db_accessadmin	Permite agregar y quitar usuarios y grupos a la base de datos.
db_backupoperator	Permite realizar copias de seguridad.
db_datareader	Permite leer información de la base de datos.
db_datawriter	Permite escribir información en la base de datos.
db_ddladmin	Permite manejar objetos de la base de datos.
db_denydatareader	Impide leer información de la base de datos.
db_denydatawriter	Impide escribir información en la base de datos.
db_owner	Permite realizar todas las operaciones sobre la base de datos.
db_securityadmin	Permite administrar funciones y permisos de base de datos.
public	Es la función predeterminada de los usuarios de la base de datos; si no tienen ningún permiso asignado, podrán realizar las operaciones definidas en esta función.

- Permisos (privilegios)

  - Un permiso es el derecho que tiene un usuario a realizar una operación en la BD.

  - Existen distintos Niveles de Permisos:

  ✓ **Nivel de Instrucciones**: Un usuario puede poseer o no permiso para ejecutar la instrucción. Estos se conceden o deniegan con las funciones del inicio de sesión. P. e. DROP de una Base de Datos.
  ✓ **Nivel Predefinido**: Es un conjunto de permisos predefinido que se selecciona cuando creamos al usuario.
  ✓ **Nivel Objeto**: Un usuario puede tener diferentes permisos sobre los objetos. Por ejemplo, cuando se crea una tabla se le dan a los usuarios los privilegios correspondientes sobre esa tabla, y la autoridad de otorgamiento.

  - El lenguaje SQL incluye órdenes para conceder y retirar privilegios. Incluye los privilegios: **delete**, **insert**, **select**, y **update**. El privilegio select se corresponde con el privilegio de lectura, delete con borrado, update con actualización e insert con insertar.

- SQL también incluye un privilegio **references** que restringe la capacidad de un usuario para declarar claves foráneas al crear relaciones. Si la relación que va a crear incluye una clave foránea que hace referencia a un atributo de otra relación, el usuario debe haber recibido este privilegio.
- En un principio puede parecer que no hay ningún motivo para evitar que los usuarios creen claves externas que hagan referencia a otra relación. Sin embargo, hay que recordar que **las restricciones de las claves externas limitan las operaciones de borrado y de actualización** sobre la operación a la que hacen referencia.

- La instrucción **grant** se utiliza para conceder autorizaciones. La forma básica es la siguiente:

```
grant <lista de privilegios> on <nombre de la relación o
  vista> to <lista de usuarios>
```

- La siguiente instrucción permite a los usuarios X, Y y Z la autorización select sobre la relación PRESTAMO:

```
grant select on PRESTAMO to X, Y, Z
```

- La autorización **update** puede concederse sobre todos los atributos de la relación o sólo sobre algunos, por ejemplo si se puede actualizar solo el atributo importe de la relación PRESTAMO:

```
grant update (importe) on PRESTAMO to X, Y, Z
```

- En SQL92 el privilegio **insert** puede especificar una lista de atributos, al resto de los atributos de la relación (si tuviera más) se los completa con valores predeterminados (si hay algún valor predeterminado definido) sino se les da el valor nulo.

- Existe un privilegio **all privileges**, para dar todos los privilegios a un usuario sobre una relación. De manera parecida, el nombre de usuario **public** hace referencia a todos los usuarios presentes y futuros del sistema.

- Para retirar una autorización se usa la instrucción **revoke**. Su forma básica es muy parecida a grant:

```
revoque <lista de privilegios> on <nombre de la relación o
  vista> from <lista de usuarios>
```

- Integridad - Limitantes de Integridad

  - Hablamos de **Integridad** en el sentido de **corrección**, **validez** o **precisión de los datos** de la BD.

  - **Los limitantes de integridad** son una forma de garantizar que los cambios que hacen a la BD los usuarios autorizados no van a resultar en una pérdida de la consistencia de la información, es decir, **previenen contra pérdidas accidentales**.

  - **Las transacciones deben respetar las limitantes para conservar la consistencia,** pero pueden cometerse errores. La labor del componente de control de integridad del sistema es detectarlos.

  - Una BD puede estar sujeta a cualquier cantidad de restricciones de integridad, **el SGBD debe ser informado de todas estas restricciones, y debe hacerlas cumplir**.

  - Al declarar una nueva restricción, el sistema debe asegurarse de que la BD la satisfaga actualmente. Si no es así la restricción es rechazada, de lo contrario se la acepta, almacena en el catálogo y se la impone desde ese momento.

- Resumiendo, existen dos tipos de operaciones que pueden atentar contra la integridad de los datos, son:

  - **Integridad Semántica**: Los SGBD deben ofrecer en su lenguaje de definición facilidades que permitan describir las restricciones con una sintaxis adecuada y gran flexibilidad (de manera declarativa o con un enfoque procedimental – SP). Un aspecto muy importante de las reglas de integridad es que se almacenan en el diccionario, como parte de la descripción de los datos.

  - **Integridad Operacional**: En sistemas multiusuario es imprescindible un mecanismo de control de concurrencia para conservar la integridad de la BD, ya que se pueden producir importantes inconsistencias derivadas del acceso concurrente. Las aplicaciones avanzadas de BD como sistemas GIS, CAD/CAM, etc. requieren nuevos mecanismos de control de concurrencia que permitan anidar transacciones, que soporten transacciones de larga duración y que faciliten la coordinación ente varios usuarios.

- Cifrado
  - Las autorizaciones pueden no proporcionar suficiente protección para los datos más importantes de un sistema. **La información delicada debe ser protegida de manera más segura**. Una de las formas es cifrar la información.
  - Hay un enorme número de técnicas para el cifrado de los datos. Puede que las técnicas de cifrado sencillas no proporcionen la seguridad adecuada, dado que puede ser fácil para un usuario no autorizado romper el código.
  - El cifrado nos garantizará que por más que se tenga permisos para leer los datos, éstos no serán legibles si no se tiene la manera de descifrarlos.
  - Los **algoritmos** utilizan alguno de dos principios básicos para hacer el cifrado de la información:

✓ **Sustitución**: se pretende reemplazar cada BIT, letra, grupo de bits o letras por uno diferente.

✓ **Transposición**: los elementos son reorganizados, de tal forma que la operación de intentar encontrar un determinado carácter por su correspondiente sustituto se dificulta.

↳ Es fundamental que toda información que se sustituyó y/o transformó se pueda recuperar, a esto se le denomina **reversibilidad de la operación**.

- El **número de claves** utilizadas para lograr el texto cifrado:

✓ **Criptografía simétrica**: si el dueño de la información y el usuario de la misma poseen la misma clave. Es el tipo más antiguo que se conoce, presenta ventajas tales como la **velocidad de cifrado**, y por tanto el costo es relativamente bajo. La **principal desventaja** radica en la dificultad de la distribución de la clave de descifrado.

✓ **Cifrado asimétrico (clave pública)**: El dueño de la información y el usuario de la misma tienen cada uno una clave independiente, es la **más reciente** forma de cifrado. Implica un cambio radical con respecto a la visión anterior, dado que con este sistema permite una más fácil distribución de la información, y resuelve el problema de distribución de claves; las características principales, es que presenta un **costo más elevado**, y adicionalmente no es factible su uso para grandes volúmenes información puesto que es un poco **más lento**.

- Otra aplicación interesante de la criptografía está en las **firmas digitales** para verificar la autenticidad de los datos
- Las firmas digitales **desempeñan el papel electrónico de las firmas físicas** en los documentos.

✓ La clave privada se usa para firmar los datos y los datos firmados se pueden hacer públicos.

✓ Cualquiera podría verificarlos con la clave pública, pero nadie podría haber generado los datos codificados sin tener la clave privada.

- Por tanto, se puede comprobar que los datos fueron creados realmente por la persona que afirma haberlos creado.
- Además, las firmas digitales también sirven para asegurar el rechazo. Es decir, en el caso de que una persona que creó los datos afirmase más tarde que no lo hizo (el equivalente electrónico de afirmar que no se ha firmado un talón) se puede probar que esa persona ha creado los datos (a menos que haya cedido su clave privada a otros).

- Las **buenas técnicas** de cifrado tienen las siguientes **propiedades**:

✓ Para los usuarios autorizados es sencillo cifrar y descifrar la información

✓ El esquema de cifrado no depende de mantener en secreto el algoritmo, sino de un parámetro del algoritmo → clave de cifrado.

✓ Para un intruso es muy difícil determinar cuál es la clave de cifrado.