
Seguridad en Base de Datos

Unidad III.

¿Qué es la seguridad de datos?

Se refiere a medidas de protección de la privacidad digital que se aplican para evitar el **acceso no autorizado a los datos**, los cuales pueden encontrarse en ordenadores, bases de datos, sitios web, etcétera.

La seguridad de datos también protege de una posible corrupción de los datos.



¿Qué es la seguridad de datos?

Incluye conceptos como encriptación de datos, tokenización y prácticas de gestión de claves que ayudan a proteger los datos en todas las aplicaciones y plataformas de una organización.



¿Qué es el cifrado?

- El cifrado de datos es el proceso de usar un algoritmo para transformar información para que sea ilegible para usuarios no autorizados.



Cifrado

Este método criptográfico protege datos sensibles mediante codificación y transformación de información en texto cifrado ilegible.

Estos datos codificados sólo pueden descifrarse o hacerse legibles con una clave.

Cifrado Transparente de Datos (TDE)

Los datos confidenciales se cifran de manera simple y fácil, sin requerir que los usuarios o las aplicaciones administren claves de cifrado.

No es necesario usar una vista para descifrar los datos porque una vez que el usuario pasa las verificaciones de control de acceso necesarias, los datos se descifran de forma transparente.

Cifrado Transparente de Datos (TDE)

Los administradores de seguridad pueden asegurarse de que los datos estén encriptados, pero el manejo de los datos encriptados se vuelve transparente para la aplicación.

Ventajas del Cifrado Transparente de Datos (TDE)

- Se tiene la tranquilidad de que los datos confidenciales están protegidos en caso de sustracción de los medios de almacenamiento o de los archivos de datos.
- La implementación de TDE ayuda a abordar los aspectos de cumplimiento reglamentario relacionados con la seguridad.

Ventajas del Cifrado Transparente de Datos (TDE)

- No es necesario crear desencadenadores ni vistas para descifrar los datos para una aplicación o usuario autorizados. Los datos de las tablas se descifran de forma transparente para la aplicación y el usuario de la base de datos.
- No es necesario que las aplicaciones y usuarios de base de datos sepan que los datos a los que acceden están almacenados en modo cifrado. Los datos se descifran de forma transparente para las aplicaciones y usuarios de base de datos.

Ventajas del Cifrado Transparente de Datos (TDE)

- No hace falta modificar las aplicaciones para controlar los datos cifrados. La base de datos administra el cifrado y descifrado de datos.
- Las operaciones de administración de claves están automatizadas. El usuario o la aplicación no necesitan administrar las claves de cifrado.

Cuando usar el Cifrado Transparente de Datos (TDE)

Para proteger datos confidenciales como tarjetas de crédito y números de seguridad social sin tener que administrar el almacenamiento de claves o crear tablas auxiliares, vistas y disparadores.

Las aplicaciones que manejan datos confidenciales pueden usar esta función para proporcionar un cifrado de datos sólido con pocos o ningún cambio en la aplicación.

Cómo funciona el Cifrado Transparente de Datos

El cifrado de datos transparente es un sistema de control de acceso basado en claves. Incluso si se recuperan los datos cifrados, no se pueden descifrar hasta que se produzca el descifrado autorizado, que es automático para los usuarios autorizados a acceder a la tabla.

Cómo funciona el Cifrado Transparente de Datos

Cuando una tabla contiene columnas cifradas, se usa una sola clave independientemente del número de columnas cifradas.

Las claves de todas las tablas que contienen columnas cifradas se cifran utilizando la clave maestra del servidor de la base de datos y se almacenan en las tablas del diccionario de la base de datos.

Cifrado en Oracle

Para comenzar el cifrado de datos transparente, se debe crear una nueva clave maestra con la siguiente sentencia:

```
alter system set encrytion key identified by contraseña;
```

```
alter system set encrytion key identified by Or4cl3;
```

Cifrado en Oracle



Para cifrar las columnas de una tabla se debe usar la siguiente sentencia:

```
create table nombretabla (  
  nombrecolumna tipodato,  
  nombrecolumna tipodato encrypt  
);
```

```
create table encriptado (  
  nombre varchar (50),  
  contraseña varchar encrypt  
);
```

Cifrado en Oracle

A las columnas encriptadas se les puede agregar la sentencia **salt**, la cual sirve para fortalecer la seguridad de los datos cifrados.

```
create table nombretabla (  
  nombrecolumna tipodato,  
  nombrecolumna tipodato encrypt salt  
);
```

```
create table encriptado (  
  nombre varchar (50),  
  contraseña varchar encrypt salt  
);
```

```
alter table encriptado modify contraseña encrypt salt;
```

```
alter table encriptado modify contraseña encrypt no salt;
```


Cifrado en Oracle



En caso de querer desactivar el cifrado de las columnas de una tabla se debe utilizar la siguiente sentencia:

```
alter table nombretabla modify nombrecolumna decrypt;
```

```
alter table encriptado modify contraseña decrypt;
```

Tipo de cifrado Estándar de Cifrado Avanzado (AES)

- Es un algoritmo bastante seguro, utilizado ampliamente en el mundo de la seguridad informática.
- MySQL permite cifrar y descifrar datos utilizando el algoritmo AES a través de las funciones `aes_encrypt` y `aes_decrypt`.

Sintaxis de aes_encrypt y aes_decrypt

Ambas funciones trabajan de una manera similar.

Reciben como primer argumento el texto a cifrar o descifrar, y como segundo argumento la clave.

```
aes_encrypt("texto", "clave")
```

```
aes_decrypt("texto", "clave")
```

Es importante mencionar que al encriptar se devuelven **datos binarios**, así que hay que definir (dentro de la tabla) el tipo de dato como **BLOB**.

Sintaxis de aes_encrypt y aes_decrypt sin BLOB

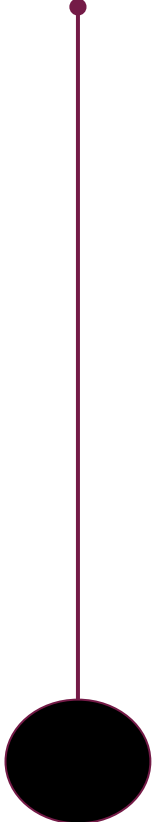
Es convertir los datos binarios a su representación hexadecimal al guardarlos; y hacer el proceso inverso al seleccionarlos.

La función que convierte un dato a hexadecimal es **HEX**, y la función que hace lo inverso es **UNHEX**.



```
hex (aes_encrypt("texto", "clave"))  
aes_decrypt(unhex("texto", "clave"))
```

Gestión de privilegios de acceso a la Base de Datos



Otras de las formas de proteger los datos es por medio de **privilegios** que se conceden a **los usuarios** y que en el momento que el usuario realiza una conexión con la base de datos, estos privilegios concedidos se van a habilitar para permitirle realizar aquellas acciones que el usuario requiera.

Gestión de privilegios de acceso a la Base de Datos

Los privilegios son el derecho a ejecutar sentencias SQL en particular.

El DBA (administrador de la base de datos) es un usuario de alto nivel con la capacidad de crear usuarios y de otorgarles acceso a la base de datos y a sus objetos.

Gestión de privilegios de acceso a la Base de Datos

La base de datos tiene **dos** tipos de privilegios:

De sistema: estos privilegios le permiten al usuario realizar acciones específicas sobre la base de datos.

De objetos: estos privilegios le permiten al usuario acceder y manipular objetos específicos.