

palasi.com
IT and management

Tema 3. Seguridad

Vicent Palasí, PhD.

3

3

palasi.com
IT and management

Índice del tema 3

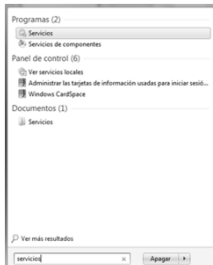
- 1. Introducción a la seguridad en SQL Server.
- 2. Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- 3. Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación
 - Roles de servidor.

4

4

palasi.com
IT and management

Importante: Antes de empezar




- Necesitamos poner el servicio SQL Server como automático. Vayan a “Servicios”

5

5

palasi.com
IT and management

En el servicio de SQL Server




- Clic derecho y Propiedades.

6

6

palasi.com
IT and management

En “Tipo de inicio”



- Pongan “Automático” y hagan clic en “Aceptar”

7

7

palasi.com
IT and management

Houston, estamos preparados

- Continuemos.

8

8

palasi.com
IT and management

Índice del tema 3

- 1. Introducción a la seguridad en SQL Server.
- 2. Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- 3. Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación.
 - Roles de servidor.


9

9

palasi.com
IT and management

**Sólo hay tres cosas importantísimas en una BD
(y en el sistema informático de una empresa)**

- La integridad. Que los datos sean correctos.
 - Cuidado al acceder y programar.
 - Normalización.
 - Transacciones-bloqueos.
 - Bitácoras (log).
 - Restricciones de integridad.
 - Índice único.
- La eficiencia. Que las operaciones de la BD sean rápidas.
 - Cuidado al acceder y programar
 - Indices.
- La seguridad de los datos.
 - Que no puedan acceder a los datos personas y programas no autorizados a ellos.



10

10

palasi.com
IT and management

Si era tan importante

- ¿Por qué no lo habíamos visto hasta ahora?
- Porque para aprender seguridad necesitas saber todos los temas que hemos explicado hasta ahora.

11

11

palasi.com
IT and management

SQL Server es un SGBD que pone énfasis en la seguridad en el acceso a los datos.

- La seguridad asegura de que nadie pueda acceder a los datos para los que no está autorizado.
- De esta manera, se evita que alguien lea datos sensibles a los que no está autorizado.
- O que destruya la integridad de los datos, ya sea de forma accidental o maliciosa.

12

12

palasi.com
IT and management

Ejemplo

- El gerente de Compras sólo puede acceder a los datos de Compras en la BD.
- El programa de contabilidad de El Salvador sólo puede acceder a los datos de El Salvador.
- La secretaria no debe acceder al servidor de BDs en absoluto.

13

13

palasi.com
IT and management

La seguridad tiene dos partes


- Autenticación.**
 - ¿Qué personas y programas pueden acceder al servidor de BD?
 - Por ejemplo, la secretaria no puede acceder.
- Autorización.**
 - De entre todos las personas y programas que pueden acceder.
 - ¿A qué datos puede acceder cada uno? BDs, tablas, registros...
 - Por ejemplo, el gerente de Compras sólo puede acceder a los datos de Compras en la BD. El programa de contabilidad de El Salvador sólo puede acceder a los datos de El Salvador.

14

14

Proceso

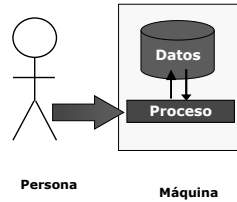
- Un proceso es un programa en ejecución.
- Un programa que se está ejecutando.



15

Un programa monolítico

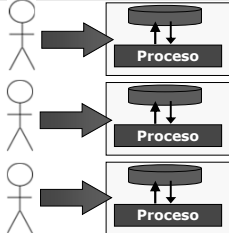
- En un programa individual (Word), cada humano tiene acceso a un proceso en una única máquina



Persona → Máquina

16

Un programa monolítico es operado de manera independiente por diferentes personas

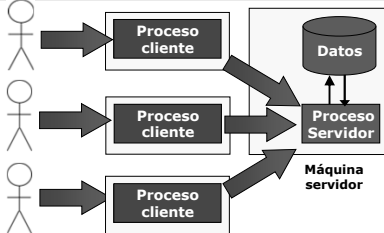


- Cada persona tiene sus recursos, tiene sus datos

Personas → Máquinas cliente

17

Un programa cliente-servidor es compartido por muchas personas

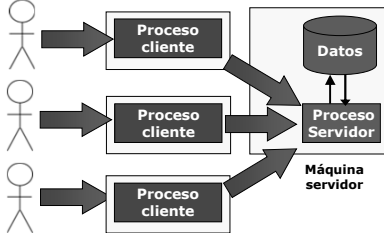


El programa se parte en una parte común (servidor) y una parte individual (clientes)

Personas → Máquinas cliente → Máquina servidor

18

Por falta de espacio, sólo vamos a dibujar un humano

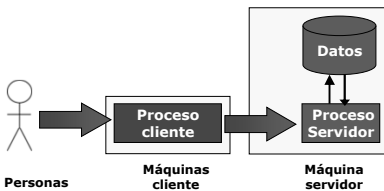


Pero piensen que siempre hay varios.

Personas → Máquinas cliente → Máquina servidor

19

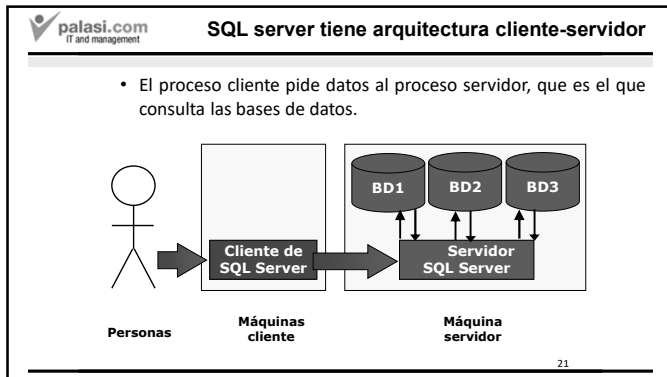
Esta arquitectura cliente-servidor es la que utiliza SQL server



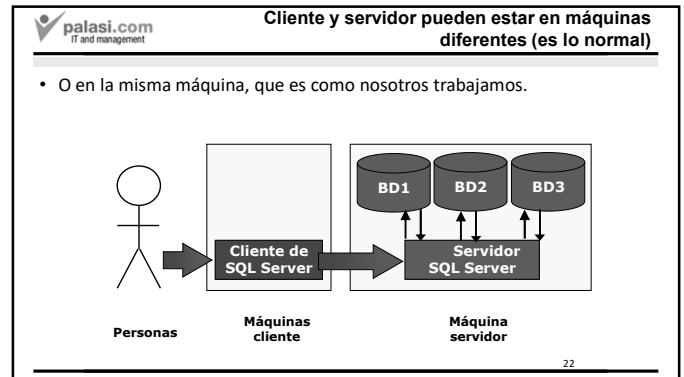
Pues la base de datos es común para todos los clientes.

Personas → Máquinas cliente → Máquina servidor

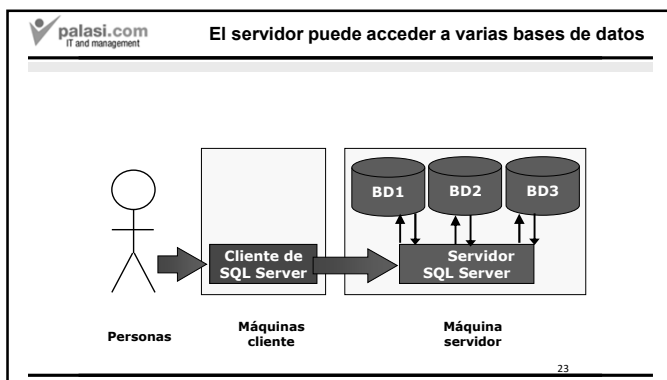
20



21



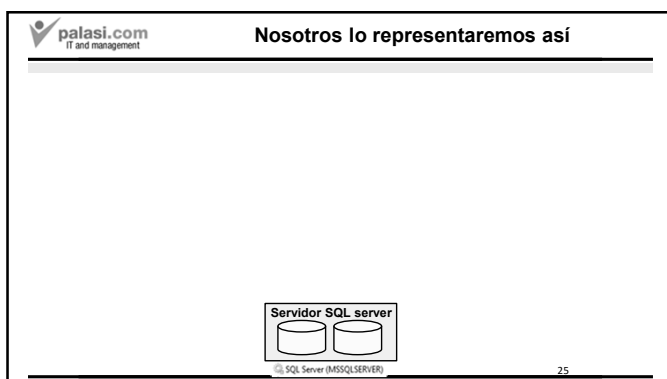
22



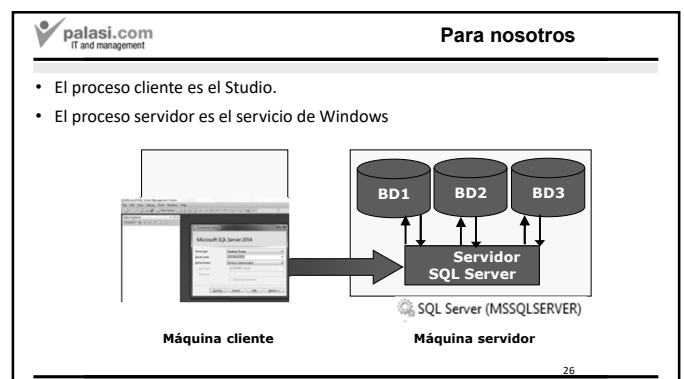
23



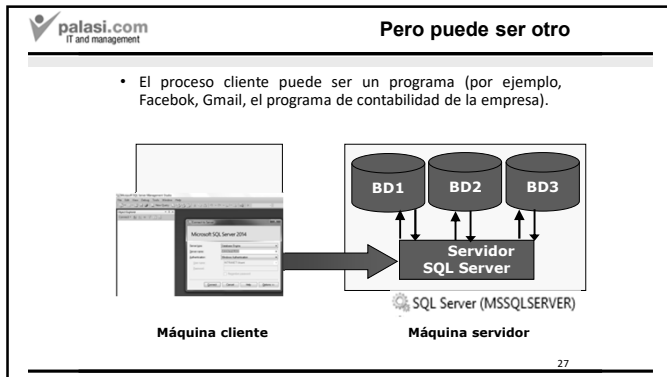
24



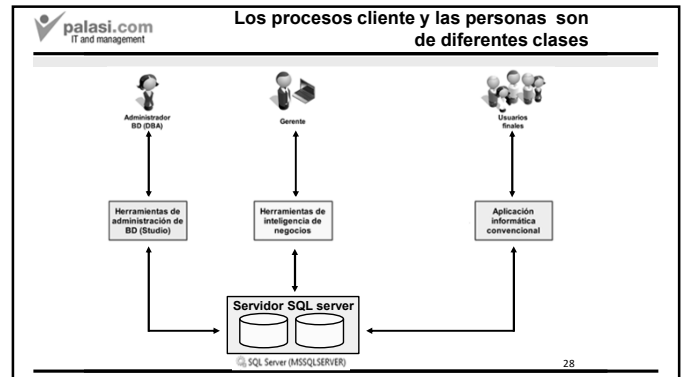
25



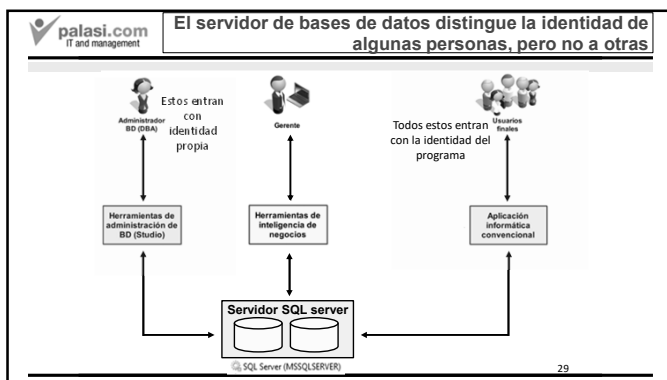
26



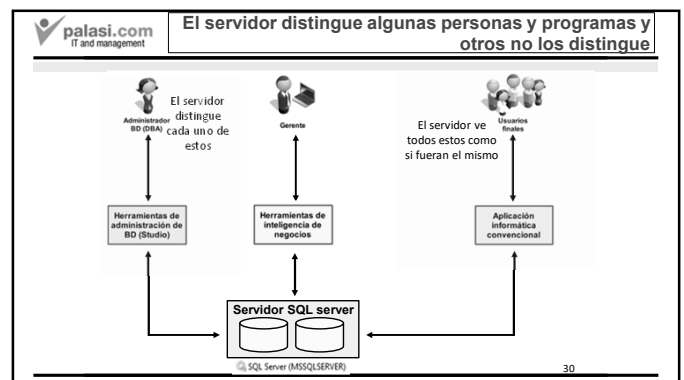
27



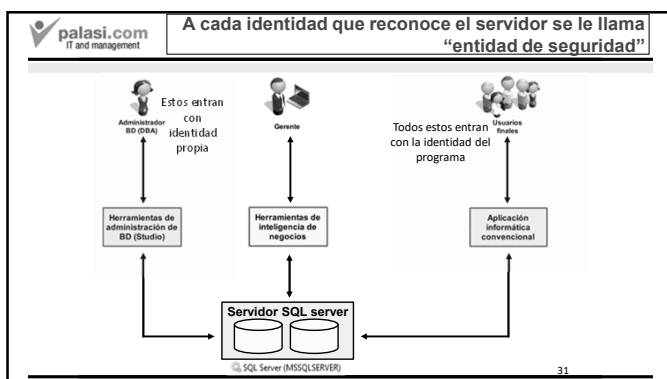
28



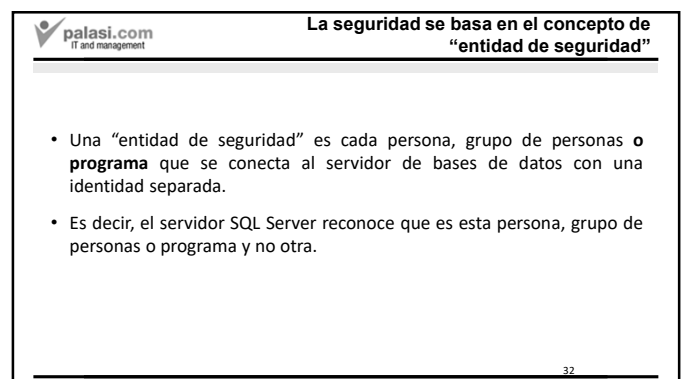
29



30



31



32

palasi.com
IT and management

Sin embargo

- Esto de “entidad de seguridad” es terriblemente confuso.
- Es el nombre oficial de SQL server, pero muy poco claro.
- Yo les llamaré “actores de seguridad” o simplemente “actores”.

33

33

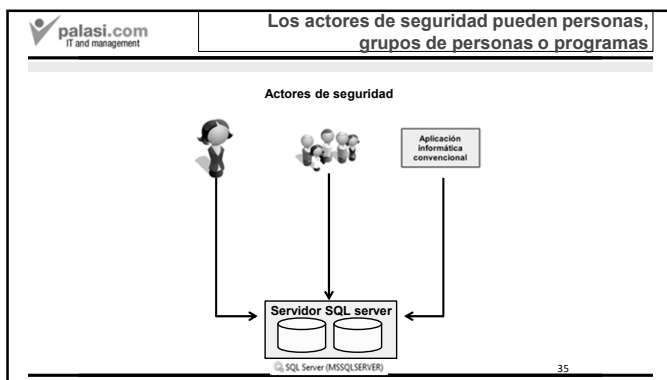
palasi.com
IT and management

La seguridad se basa en el concepto de “actor de seguridad”

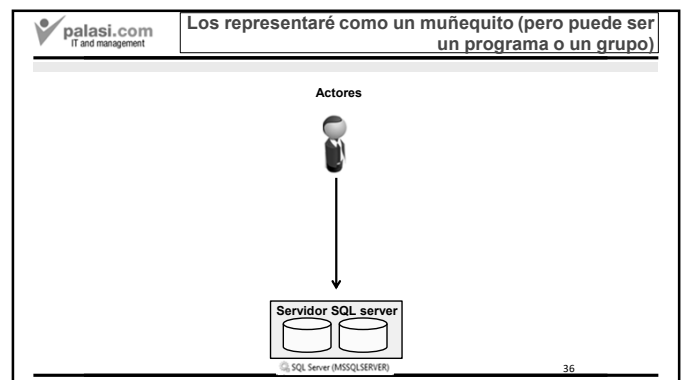
- Un “actor de seguridad” es cada persona, grupo de personas o **programa** que se conecta al servidor de bases de datos con una identidad separada.
- Es decir, el servidor reconoce que es esta persona, grupo de personas o programa y no otra.

34

34



35



36

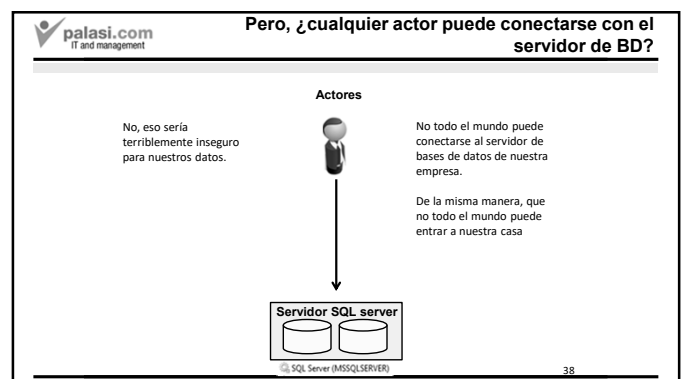
palasi.com
IT and management

Índice del tema 3

- 1. Introducción a la seguridad en SQL Server.
- 2. Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- 3. Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación
 - Roles de servidor.

37

37



38

Sólo los actores autorizados pueden conectarse al servidor de BD

¿Cómo hacemos para dar acceso a los actores autorizados y no dar acceso a los no autorizados?

Actores

Como en nuestra casa, damos una llave a los actores autorizados y no la damos a los actores no autorizados.

Servidor SQL server

SQL Server (MSSQLSERVER)

39

39

Si vemos el servidor de SQL Server como una casa

Servidor SQL server

SQL Server (MSSQLSERVER)

40

Si vemos el servidor de SQL Server como una casa

SERVIDOR SQL SERVER

LOGIN

41

Se necesita una llave para entrar a esa casa, que sólo damos a los actores autorizados

SERVIDOR SQL SERVER

LOGIN

42

A esta llave se le llama "login" o "inicio de sesión" y es un par de clave y contraseña (credencial)

SERVIDOR SQL SERVER

LOGIN

43

Conectarse al servidor de la BD (servidor de SQL Server)

- No todo el mundo se puede conectar.
- Sólo los que están autorizados.
- Necesitas una credencial (par de clave y contraseña).
 - Para demostrar que eres autorizado.
 - De forma que puedas conectarte a la BD.
- A esta credencial (clave y contraseña) se le llama
 - "inicio de sesión"
 - "login"
- Es parecido a 1 llave para entrar al servidor de SQL Server.

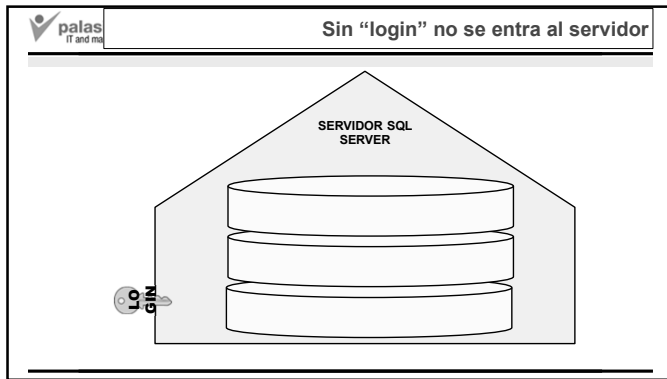
LOGIN CLAVE CONTRASEÑA

44

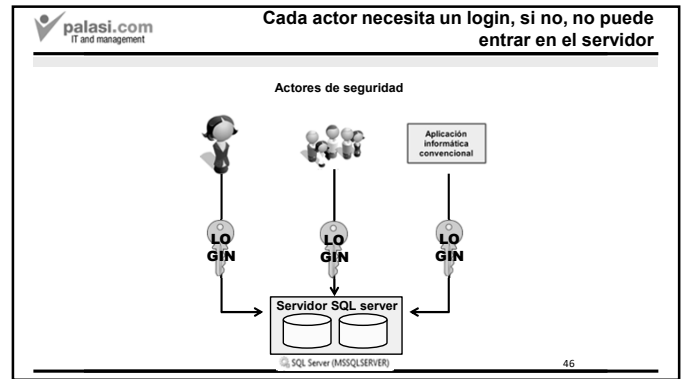
44

Vicent Palasí, PhD, MBA, MEd. Todos los derechos reservados.

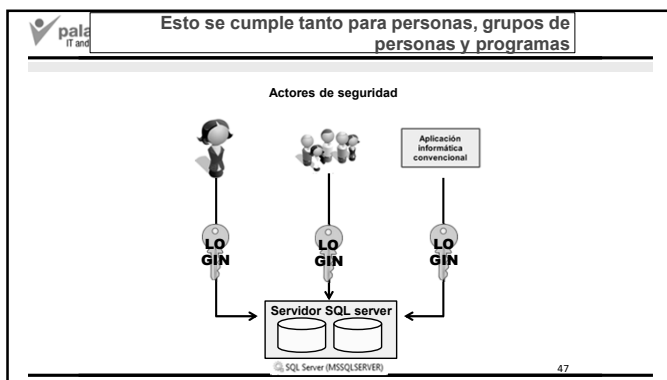
Mail: palasi@palasi.com Web: www.palasi.com



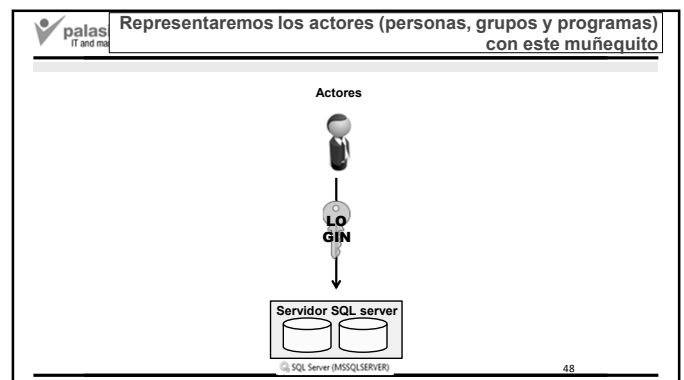
45



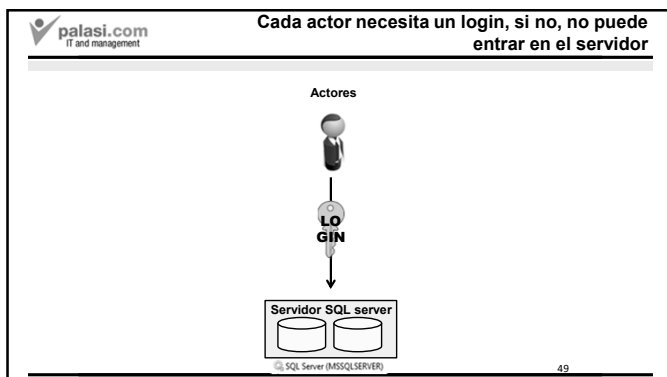
46



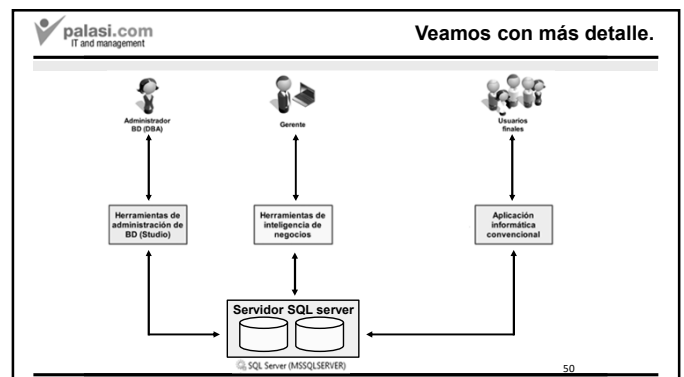
47



48



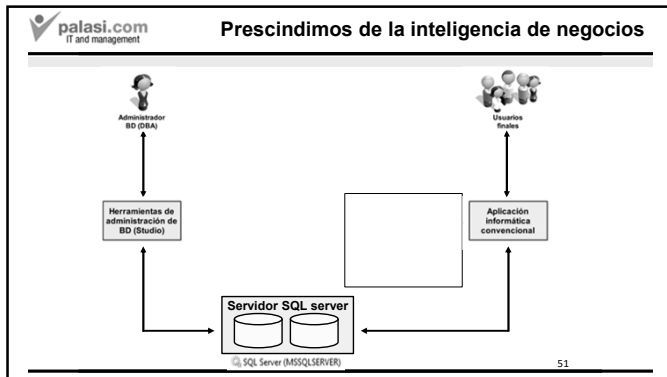
49



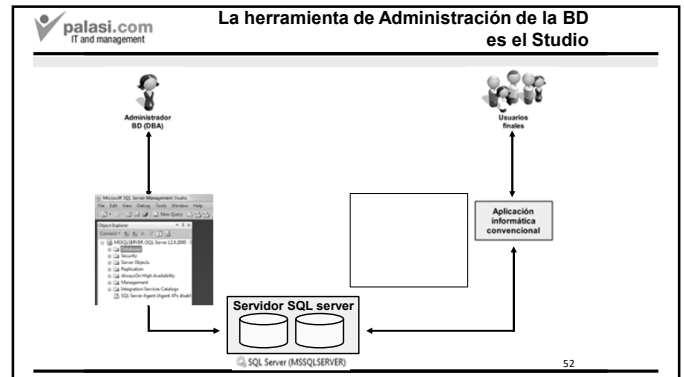
50

Vicent Palasí, PhD, MBA, MEd. Todos los derechos reservados.

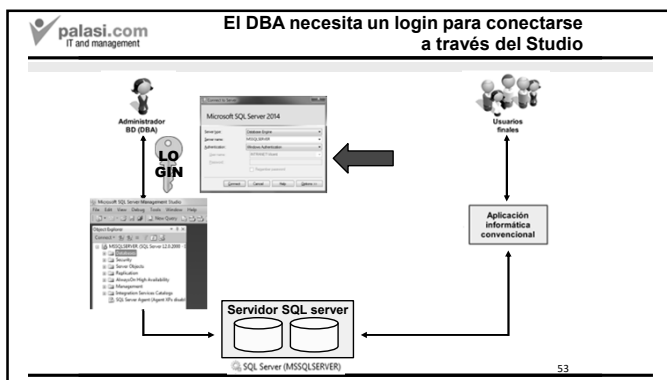
Mail: palasi@palasi.com Web: www.palasi.com



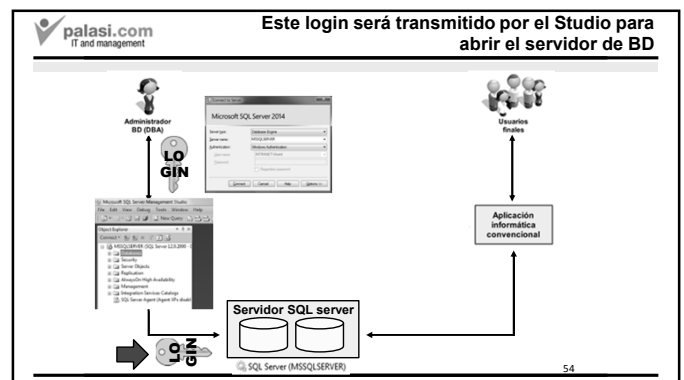
51



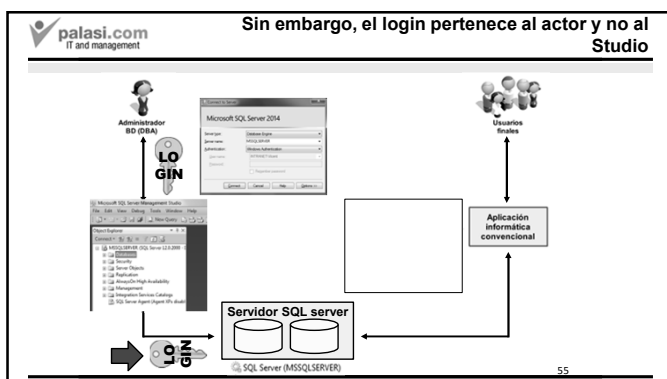
52



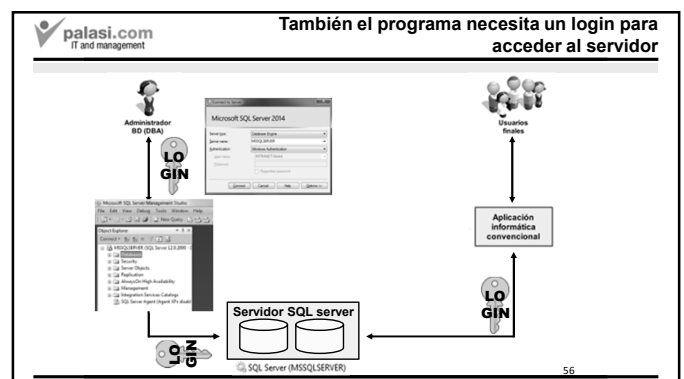
53



54



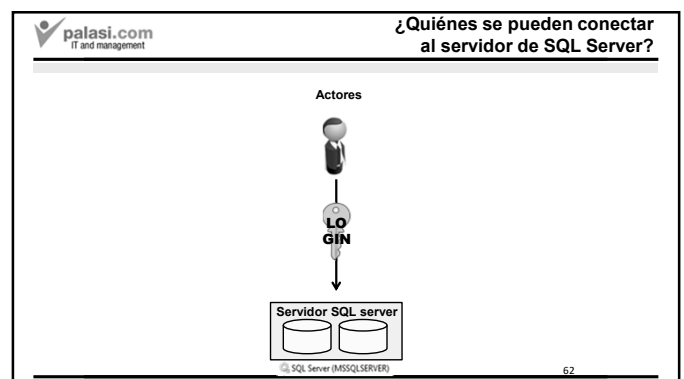
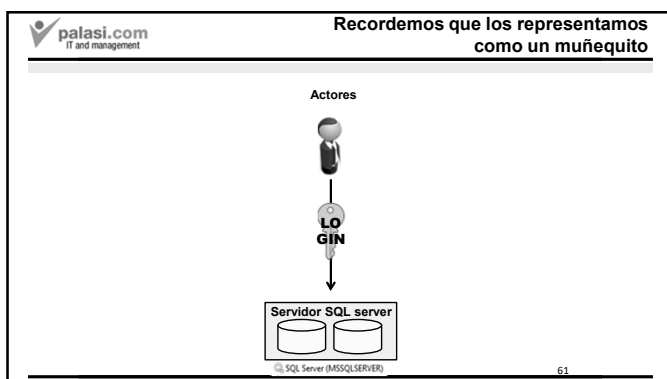
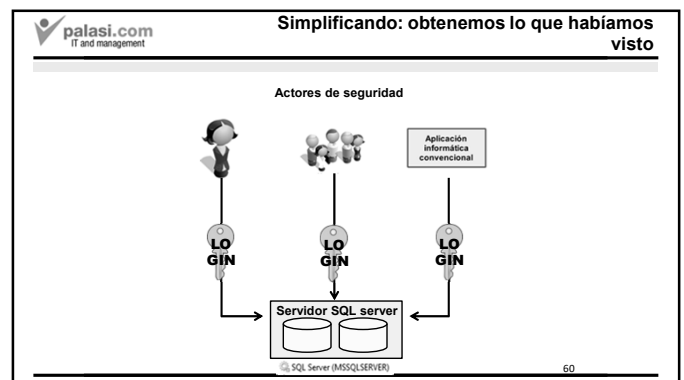
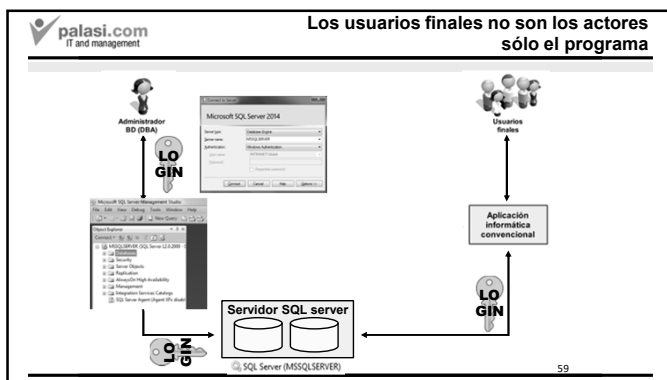
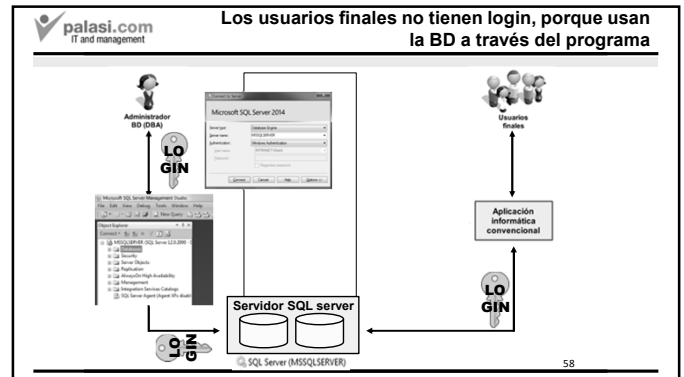
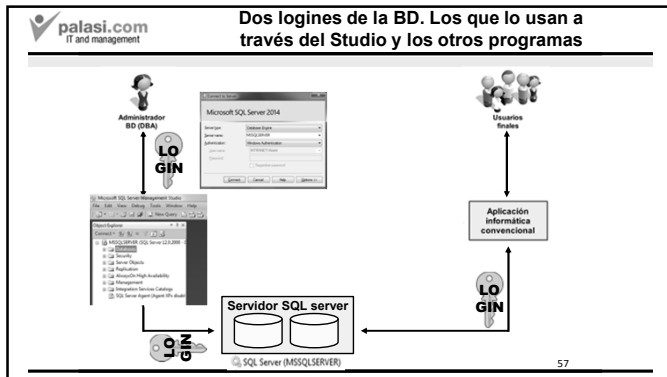
55

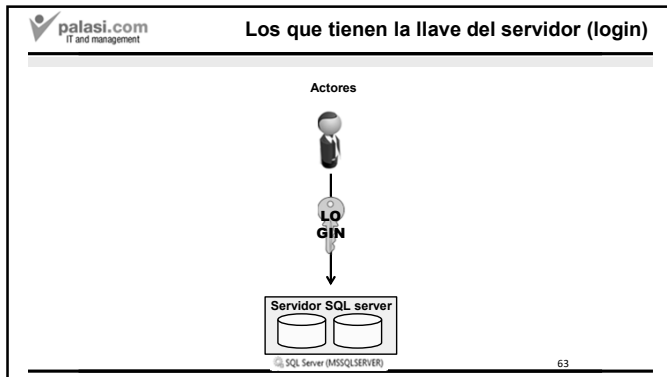


56

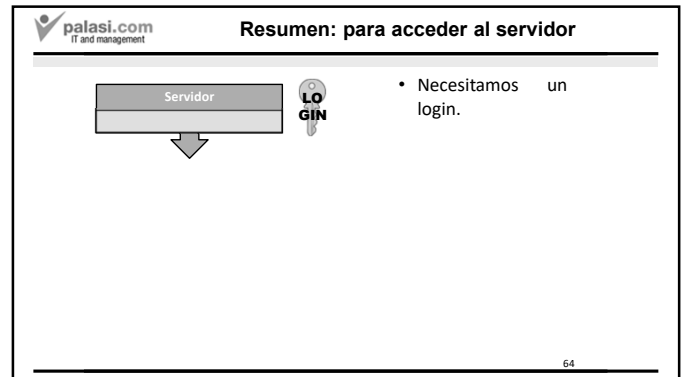
Vicent Palasí, PhD, MBA, MEd. Todos los derechos reservados.

Mail: palasi@palasi.com Web: www.palasi.com

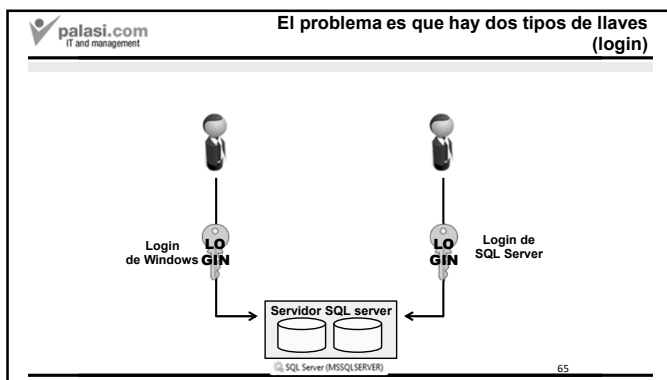




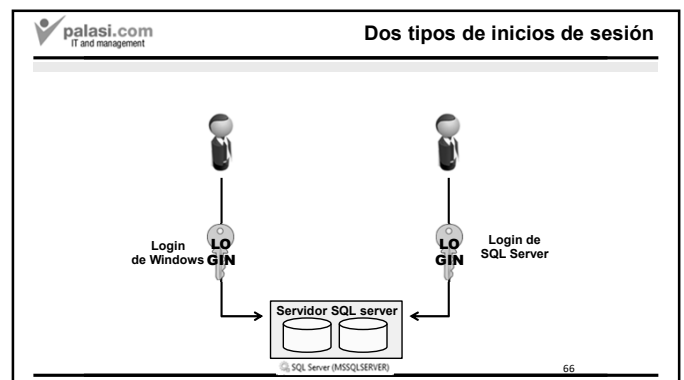
63



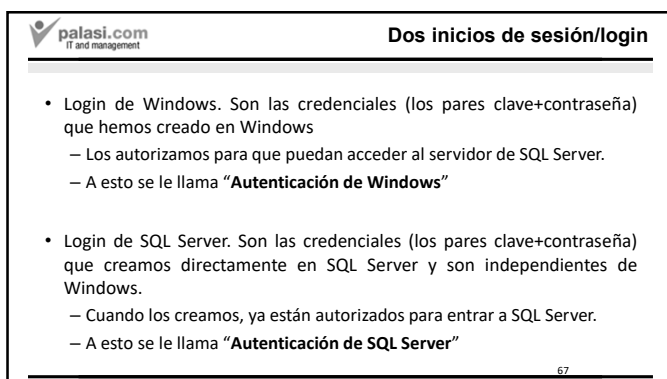
64



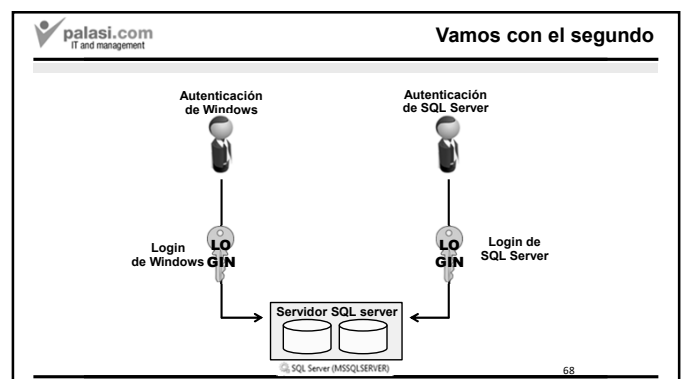
65



66



67



68

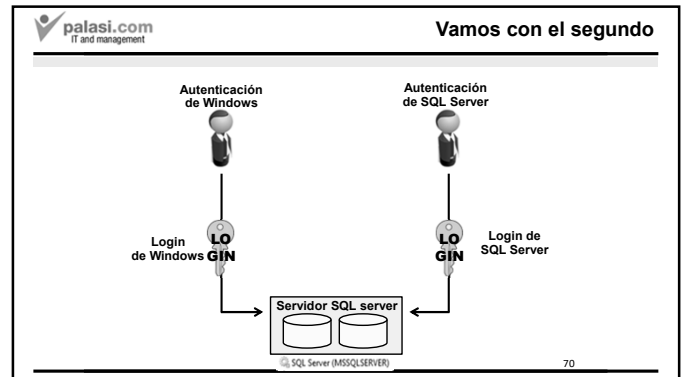
Vicent Palasí, PhD, MBA, MEd. Todos los derechos reservados.

Mail: palasi@palasi.com Web: www.palasi.com

Índice del tema 3

- 1. Introducción a la seguridad en SQL Server.
- 2. Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- 3. Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación
 - Roles de servidor.

69



70

Modo de autenticación de SQL Server

- Sirve para
 - dar acceso a computadoras con sistema operativo diferente de Windows.
 - cuando queremos definir los inicios de sesión de forma independiente del Windows.
- SQL Server define los actores, en vez de importarlos desde Windows.

71

Para definir un login con modo de autenticación SQL Server

- En una ventana de consultas, se hace (la parte azul es opcional)

```
create login clave with password = 'password'
must_change, check_expiration = on
```

- Para quitar a este login, se hace

```
drop login clave
```

- Para cambiar la contraseña, se hace

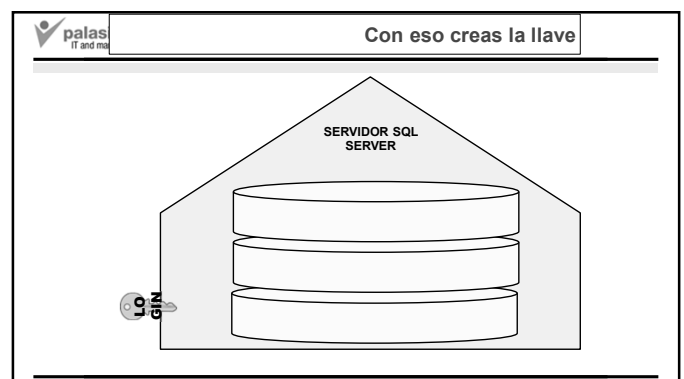
```
alter login clave with password = 'password'
```

72

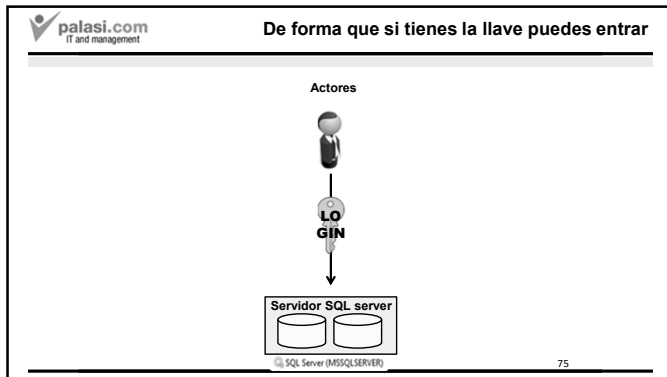
Ejercicio

- Entren con su login normal al Studio.
- Creen un login de SQL Server que se llame "maria".

73



74



75

Ya sabes hacer llaves

- Pero, ¿cómo entras?

76

Para entrar. Para conectarse al Studio con un login de SQL Server

- Hay que elegir "Autenticación de SQL Server" al inicio en "Autenticación".
- Hay que dar la llave (clave y contraseña).



Microsoft SQL Server 2014

Server type: Database Engine

Server name: VICENTPALASI

Authentication: SQL Server Authentication

Login:

Password:

☐ Remember password

Connect Cancel Help Options >>

77

Ejercicio

- Salgan del Studio.
- Entren como "maria".
- ¿Les deja entrar?
- Intenten acceder a las bases de datos.
- ¿Les deja? ¿Por qué?

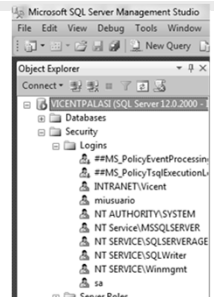
78

Vuelvan a salir del Studio

- Y entren con su login habitual.

79

Expandan "Seguridad" e "Inicios de sesión"

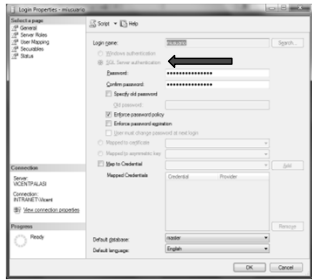


- Y verán el login que han creado.

80

Si hacen clic derecho y "Propiedades"

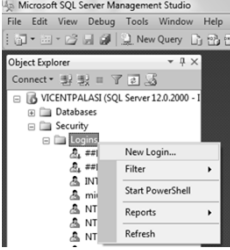
- Veremos las propiedades de ese login.
- Mirar que es autenticación de SQL Server.
- Se puede cambiar la contraseña
- Hacer clic en "Aceptar".



81

81

Ahora creemos un login de SQL Server de forma gráfica

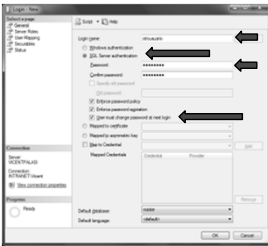


- En el nodo "Inicio de sesión", que está debajo de "Seguridad", se hace clic derecho en "Nuevo inicio de sesión".

82

82

Veamos desde la máquina servidor



- Debe ponerse clave y contraseña (dos veces).
- Se debe elegir "Autenticación de SQL Server".
- Se puede forzar a que el actor cambie la contraseña la siguiente vez que se conecte.

83

83

Ejercicio

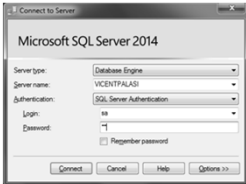
- Crear otro login de forma gráfica en la máquina servidor.
 - Fuercen a que cambie la contraseña.
- Entren con él en la máquina cliente.

84

84

El login sa ("system administrator")

- Es de autenticación de SQL Server. Se crea en la instalación y puede hacerlo todo con el servidor de la BD.



- Por eso, es importante ponerle una contraseña segura.
- La contraseña se pone en la instalación y en su caso es "root".

85

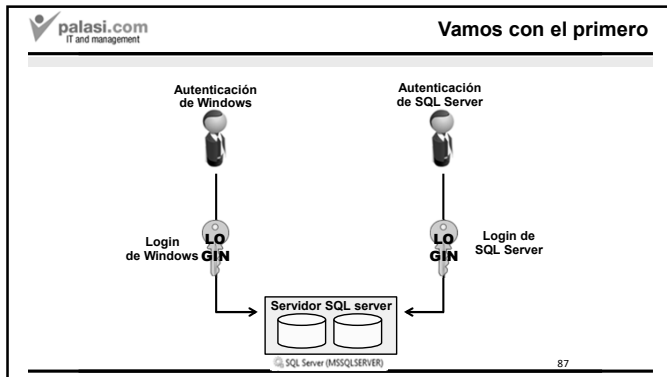
85

Índice del tema 3

- 1. Introducción a la seguridad en SQL Server.
- 2. Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- 3. Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación
 - Roles de servidor.

86

86



87

Cuando hablamos de autenticación de Windows

- Los logines de SQL Server son los usuarios y grupos de Windows.
- Es por eso que, a partir de ahora, lo que hablaremos es de crear usuarios y grupos de Windows.
 - Esto es más una clase de Windows que de SQL Server

88

¿Para qué usar un login de Windows para conectarse a SQL Server?

- Para no duplicar los logines.
- Con un mismo login, tenemos para Windows y SQL Server.
- Más fácil de administrar.

89

Dos formas de vivir en El Salvador

- 1. Ser creado en El Salvador.

Creado en El Salvador
- 2. Ser creado en otro país y que te autoricen en El Salvador.

Creado en el extranjero

➔

Autorizado para vivir en El Salvador

90

Crear un login

- Autenticación de SQL Server

Crear en SQL Server
- Autenticación de Windows

Crear en Windows

➔

Autorizarlo para SQL Server

91

Crear un login

- Autenticación de Windows

Crear en Windows

➔

Autorizarlo para SQL Server

92

palasi.com
IT and management

Crear un login

- Autenticación de Windows

Crearlo en Windows

➔

Autorizarlo para SQL Server
- Es decir, crear usuarios y grupos en Windows.
 - Crear usuarios.
 - Crear grupos

93

93

palasi.com
IT and management

Tipos de usuarios y grupos de Windows según su ubicación

- Usuarios y grupos locales.
 - Son los que están definidos en la computadora local.
 - Sólo pueden usarse en esa computadora.
- Usuarios y grupos de red.
 - Están definidos en la red interna de la empresa.
 - Pueden usarse en cualquier computadora.
 - Se dividen en diversos conjuntos llamados dominios.
 - En la red hay varios "dominios" que son conjuntos de usuarios y grupos de red.
 - ALUMNOS, PROFESORES, INTRANET.

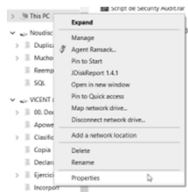
94

94

palasi.com
IT and management

Claves en Windows

- Tienen la forma de **dddd\nombreUsuario**
 - Donde **dddd** es el nombre de la computadora o dominio donde está instalado el servidor de BD
 - (se puede ver en **Clic derecho en Equipo|Propiedades**)



Device specifications

OMEN by HP Laptop 15-dc1xxx

Device name	LAPTOP-VU7SC07K
Processor	Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
Installed RAM	8.00 GB (7.81 GB usable)
Device ID	D3D05418-A933-4D23-9AEF-385536165AF1
Product ID	00325-81641-87719-AA0EM
System type	64-bit operating system, x64-based processor
Pen and touch	Pen support

95

95

palasi.com
IT and management

Claves en Windows

- Tienen la forma de **dddd\nombreUsuario**
 - Donde **dddd** es el nombre de la computadora o dominio donde está instalado el servidor de BD
 - (se puede ver en **Clic derecho en Equipo|Propiedades**)
 - Donde **nombreUsuario** es el nombre de usuario en Windows.
- Si tiene algún carácter extraño debe ir entre corchetes.
- Para un identificador de grupo es lo mismo, pero en vez del nombre de usuario aparece el nombre del grupo.

96

96

palasi.com
IT and management

Claves en Windows

- Tienen la forma de **dddd\nombreUsuario**
 - Donde **dddd** es el nombre de la computadora o dominio donde está instalado el servidor de BD
 - (se puede ver en **Clic derecho en Equipo|Propiedades**)
 - Donde **nombreUsuario** es el nombre de usuario en Windows.
- ¿Qué tipo es el usuario que están usando en esta asignatura? Tanto en su máquina personal como la máquina del laboratorio.

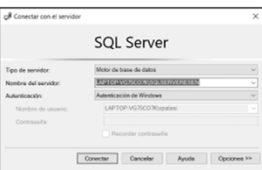
97

97

palasi.com
IT and management

Un usuario local

- Vemos que es local porque tiene el nombre de la máquina local



Device specifications

OMEN by HP Laptop 15-dc1xxx

Device name	LAPTOP-VU7SC07K
Processor	Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
Installed RAM	8.00 GB (7.81 GB usable)
Device ID	D3D05418-A933-4D23-9AEF-385536165AF1
Product ID	00325-81641-87719-AA0EM
System type	64-bit operating system, x64-based processor
Pen and touch	Pen support

98


98

Vicent Palasí, PhD, MBA, MEd. Todos los derechos reservados.

Mail: palasi@palasi.com Web: www.palasi.com

Un usuario local

- Si hacemos clic en Switch User podemos cambiar de usuario.




99

99

O bien podemos entrar en un dominio de red

- Ahora entraremos al dominio PROFESORES



100

100

Nosotros sólo trabajaremos con usuarios locales

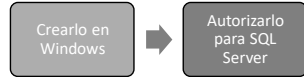
- No tenemos permisos para tocar los usuarios de red.
- Esto corresponde al Departamento de Sistemas.

101

101

Crear inicios de sesión en Windows

- Crear usuarios y grupos en Windows.
 - Crear usuarios.
 - Crear grupos

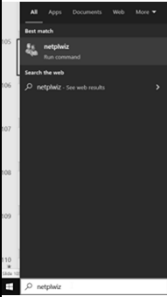


102

102

Definamos usuarios de Windows

- En la barra de búsqueda de Windows pongan "netplwiz".

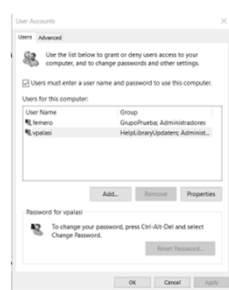


103

103

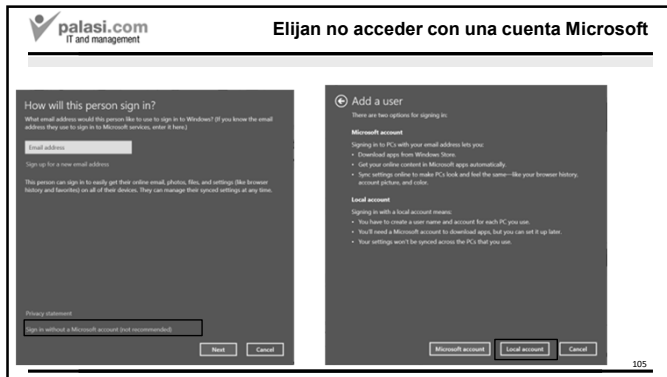
Salte la ventana de usuarios

- Aquí están todos los usuarios de tu máquina.
- Creemos uno haciendo clic en el botón "Add..."



104

104



105



106

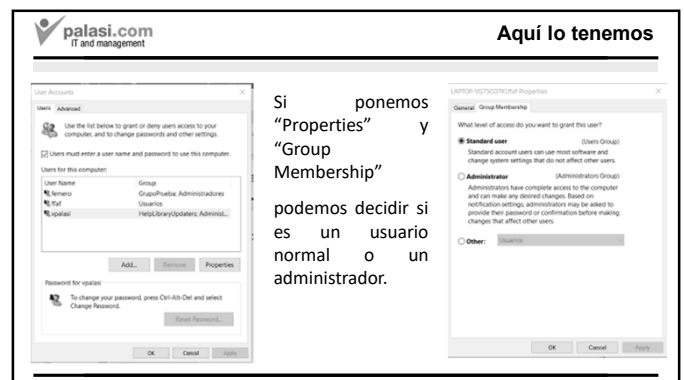
Dos tipos de usuario local de Windows

- Los usuarios definidos como **“Administradores”** tienen acceso a todo lo de Windows en esa máquina local.
- Los usuarios definidos como **“Estándar”** no tienen acceso a nada, mientras no se especifique lo contrario.
- Mientras no digamos nada, el usuario se crea como **“Estándar”**.

107

Aquí lo tenemos

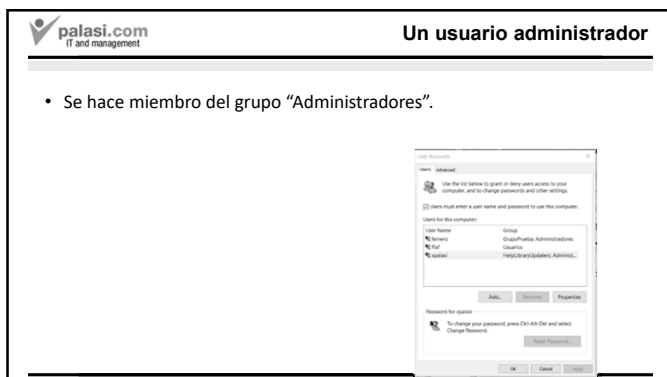
Si ponemos **“Properties”** y **“Group Membership”** podemos decidir si es un usuario normal o un administrador.



108

Un usuario administrador

- Se hace miembro del grupo **“Administradores”**.



109

Ejercicio

- Creen tres usuarios ESTÁNDAR en Windows. Lo llamaremos **“limited1”**, **“limited2”** e **“independent”**.
- Salen de Windows.
- Ingresen a Windows con el usuario **“limited1”**.
- Deben poner siempre el nombre de la máquina antes de la fleca.
- Intenten crear un usuario dentro de esta cuenta. ¿Se puede?
- Salgan y vuelvan a entrar a su usuario principal.
- Creen un usuario **“boss”** y háganlo administrador.
- Entren a ese usuario.
- ¿Pueden crear usuarios?
- Vuelvan a su usuario habitual.

110

Crear inicios de sesión en Windows

- Crear usuarios y grupos en Windows.
 - Crear usuarios.
 - Crear grupos

Crearlo en Windows

➔

Autorizarlo para SQL Server

111

111

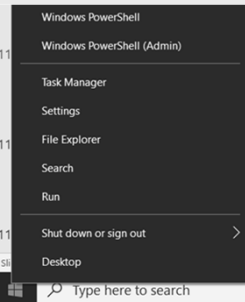
Para crear grupos

- Es un poco difícil en la versión Home de Windows (la que ustedes tienen)
- Podemos hacerlo, pero Microsoft hizo que no fuera fácil.

112

112

Deberemos abrir una ventana de Powershell con privilegios de administrador



- Powershell es un lenguaje que nos permite administrar Windows.
- Clic derecho en el botón de Windows.
- Seleccionamos Windows PowerShell (Admin)

113

113

Para crear un grupo y para agregar un usuario a un grupo

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> New-LocalGroup -Name GrupoPrueba

Name      Description
----      -
GrupoPrueba

PS C:\WINDOWS\system32> Add-LocalGroupMember -Group GrupoPrueba -Member femero
PS C:\WINDOWS\system32>
  
```

114

114

Ejercicio

- Creen un grupo "thelimited" y metan a "limited1" y a "limited2".

115

115

Con esto hemos definido los diferentes usuarios y grupos

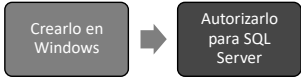
- Pero estos son usuarios y grupos de Windows.
 - Tienen acceso a Windows, pero no a SQL Server.
- Lo primero que tenemos que hacer es autorizar a estos usuarios para que puedan entrar al SQL Server.

116

116

Autorizar usuarios y grupos en SQL Server

- Autenticación de Windows

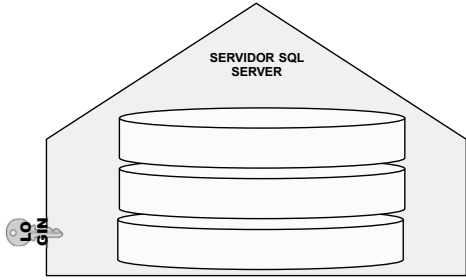


Crear en Windows → Autorizarlo para SQL Server

117

117

Hay que dejar que ese login sea la llave a SQL Server



LOGIN

SERVIDOR SQL SERVER

118

118

Dar acceso a esos usuarios en SQL Server

- De forma predeterminada, sólo puede entrar al SQL Server
 - el login **sa** (de autenticación de SQL Server)
 - el usuario de Windows que hemos indicado en la instalación.
- Las cuentas estándar que necesiten acceso a SQL Server, se lo tendremos que dar explícitamente.

119

119

No autorizando el acceso a SQL Server a un usuario o grupo

- No se hace nada, pues, si no decimos nada, nada está autorizado.
- Esta es la filosofía SQL Server: todo está prohibido si no especificamos lo contrario.

120

120

Autorizando el acceso a SQL Server a un usuario de Windows

```
create login [xxx\yyy] from windows
```

- Para conceder el acceso a un usuario o grupo de Windows:
- Donde:
 - xxx** es el nombre de dominio o la computadora y
 - yyy** es el nombre del usuario

121

121

Retirando el acceso a SQL Server a un usuario de Windows

```
drop login [xxx\yyy]
```

- Para retirar el acceso a un usuario o grupo:
- Donde:
 - xxx** es el nombre de dominio o la computadora y
 - yyy** es el nombre del usuario

122

122

En general, en los permisos de SQL Server

- Se tratan de la misma forma los usuarios de Windows que los grupos de usuarios de Windows.
- Las mismas operaciones que sirven para un usuario de Windows, sirven para grupos de usuarios definidos en Windows.
- Por eso, a partir de ahora, cuando se hable de un usuario de Windows, se supondrá que también puede ser un grupo.

123

Ahora que tienes la llave puedes entrar



Actores

SQL Server (MSSQLSERVER)


Servidor SQL server

SQL Server (MSSQLSERVER)

124


¿Cómo entras? Poniendo “autenticación de Windows”

- Y entonces entras con el usuario en que esté el Windows en ese momento.



125


Entrar en el Studio con autenticación de Windows



- Debe poner el modo de Autenticación de Windows y el nombre del usuario de Windows.

126


Cada vez que uno inicia el Studio, debe autenticarse



- La opción de autenticación de Windows nos permite no tener que repetir la credencial (la clave y contraseña) que hemos dado al entrar a Windows.
- Tenemos una credencial (par clave-contraseña) y no dos: uno para Windows y SQL Server

127


Nota: No es sólo el Studio



- Cualquier programa de computadora que se conecta al servidor de SQL Server debe poner en su código bajo qué identificador y contraseña se conecta.

128

¿Cómo hacemos que nos deje entrar con nuestro usuario de Windows?



- Debemos autorizarlo dentro de SQL Server, como hemos visto.

129

Recordemos: autorizando el acceso a SQL Server a un usuario de Windows

```
create login [xxx\yyy] from windows
```

- Para conceder el acceso a un usuario o grupo de Windows:
- Donde:
 - **xxx** es el nombre de dominio o la computadora y
 - **yyy** es el nombre del usuario

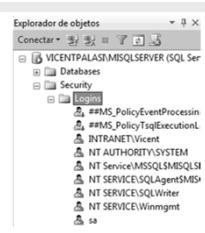
130

Ejercicio

- En su cuenta usual de Windows, denle acceso al usuario “independent” a SQL Server.
- Salgan de su cuenta y entren a Windows como usuario “independent” y entren a SQL Server.
- ¿Les deja entrar? ¿Les deja ver alguna base de datos? ¿Por qué?
- Salgan de Windows como “independent” y vuelvan a entrar en su cuenta habitual.

131

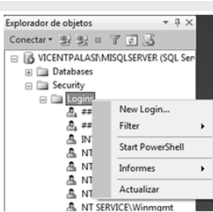
De forma gráfica



- Se expande el nodo de “Seguridad” y se hace clic derecho en “Inicios de sesión”. Si se expande, veremos los logins que pueden entrar en SQL Server

132


Para autorizar un nuevo usuario o grupo que pueda acceder a SQL Server



- Se expande el nodo de “Seguridad” y se hace clic derecho en “Inicios de sesión”. Se selecciona “Nuevo inicio de sesión...”

133


En la ventana que aparece



- Seleccionamos “Autenticación de Windows”

134

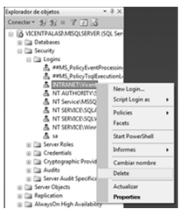
En la ventana que aparece



- Ponemos el xxx\yyy o en el cuadro texto "Nombre". Hacemos clic en "Aceptar".
- Con esto concedemos autorización al usuario o grupo de Windows para conectarse al servidor.

135

Para retirar el acceso a SQL Server de una cuenta



- Se hace clic derecho en el inicio de sesión correspondiente y se elige la opción "Eliminar".
- Si se hace clic en "Propiedades", se podrá modificar el inicio de sesión, en vez de borrarlo.

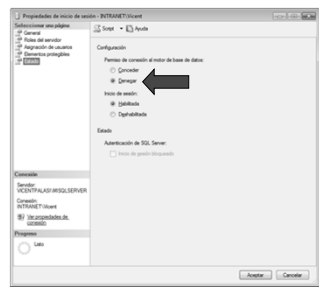
136

A veces, queremos desactivar un login

- De forma que no pueda entrar en el servidor de BDs.
- Pero tampoco queremos eliminarlo, pues queremos usarlo en un futuro.
- Para ello, podemos desactivarlo temporalmente y hay dos formas para ello.

137

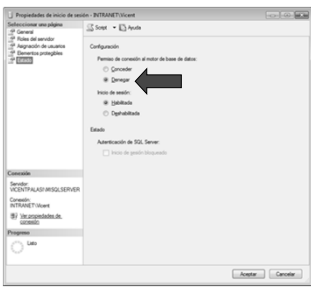
Para desactivar el acceso a SQL Server de un login



- Clic derecho en el login | Propiedades
- Página "Estado"

138

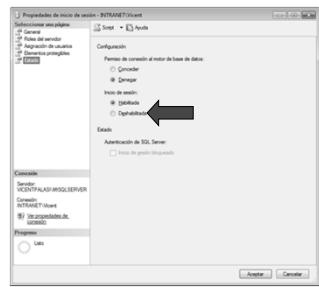
Dos opciones (1)



Se deniega "Permiso de conexión al motor de base de datos". Así no puede conectarse al servidor de BD pero sí a otros servicios como Analysis Services, Reporting Services

139

Dos opciones (2)



Si queremos que no se conecte a ningún servicio de SQL Server. Seleccionamos "Deshabilitada" en "Inicio de sesión".

140

En este caso, la cuenta existe dentro de SQL Server

- Pero no puede acceder.
- Puede ser útil tenerla allí por si en el futuro puede acceder.

141

141

Ejercicio

- Concedan acceso al grupo "thelimited" a SQL Server.
- Salgan de su máquina y entren a Windows como usuario "limited1" y vean si se puede acceder a SQL Server y a las bases de datos. ¿Por qué?
- Vuelvan a su usuario habitual.

142

142

Índice del tema 3

- 1. Introducción a la seguridad en SQL Server.
- 2. Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- 3. Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación
 - Roles de servidor.

143

143

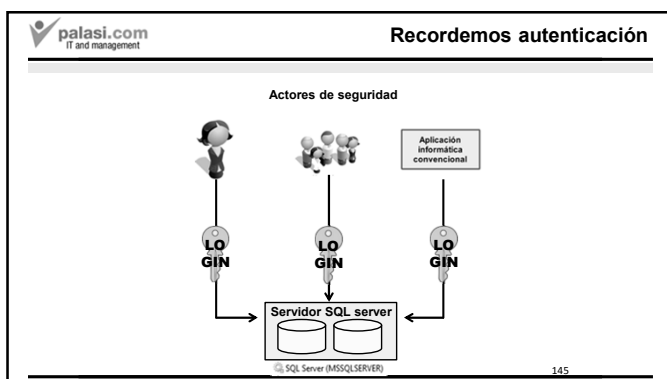
Recordemos: autenticación

- **Autenticación.**
 - ¿Qué actores pueden conectarse a SQL Server?
 - Por ejemplo, la secretaria no puede acceder.

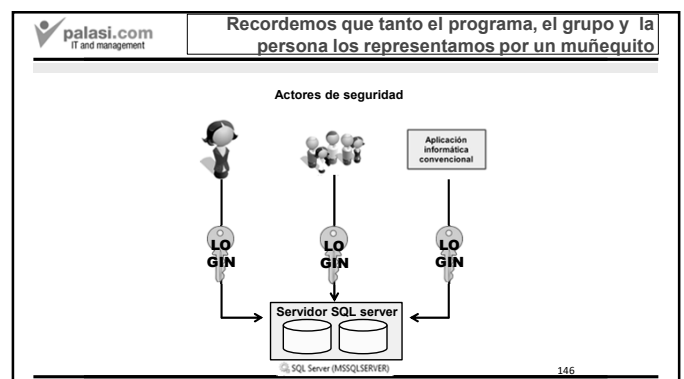
Actor = Persona, grupo de personas o programa

144

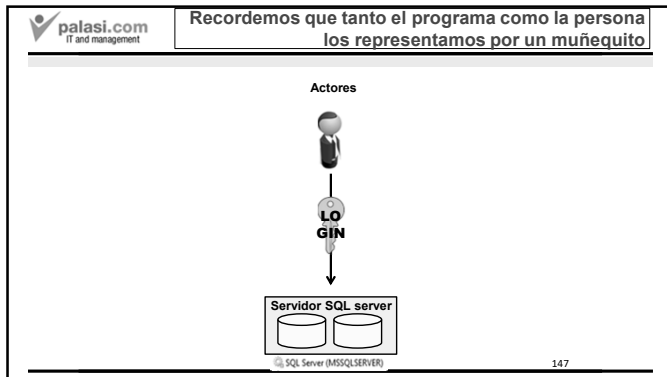
144



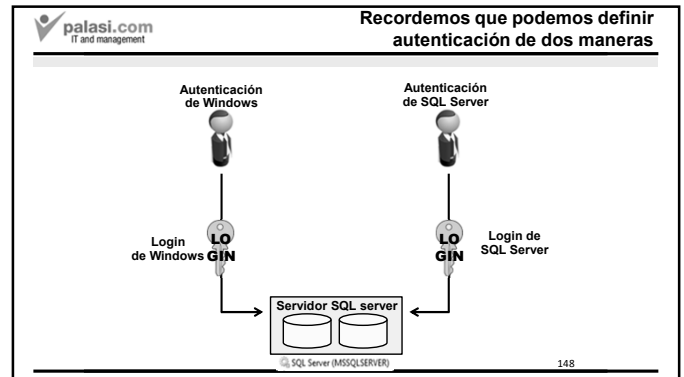
145



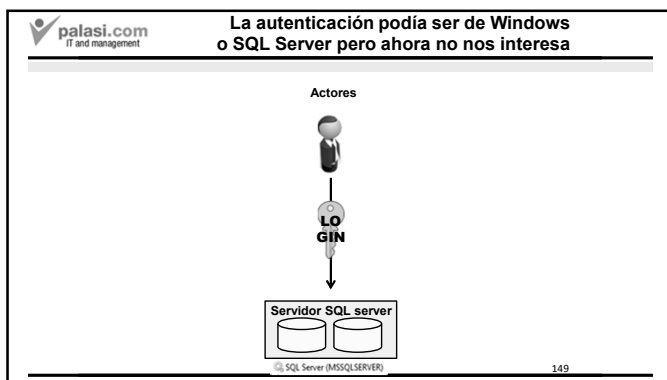
146



147



148



149

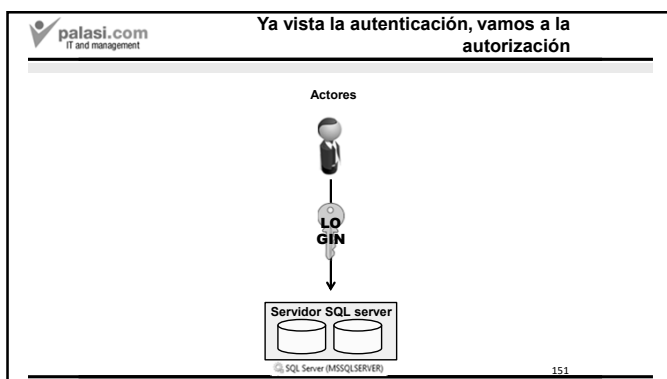
Recordemos: autenticación y autorización

- **Autenticación.**
 - ¿Qué actores pueden conectarse a SQL Server?
 - Por ejemplo, la secretaria no puede acceder.
- **Autorización.**
 - Para cada actor,
 - Qué puede tocar cada uno.

Actor = Persona, grupo de personas o programa

150

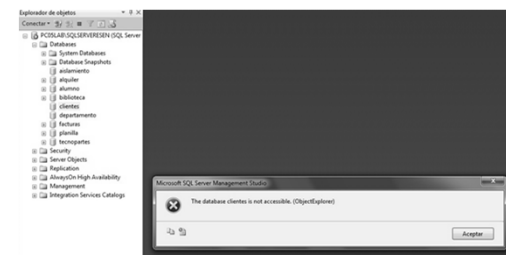
150



151

Si un actor con login no tiene permisos sobre una BD

- No puede acceder a la BD, aunque esté autenticado.



Explorador de objetos

Conectar a: SQL Server Enterprise

Microsoft SQL Server Management Studio

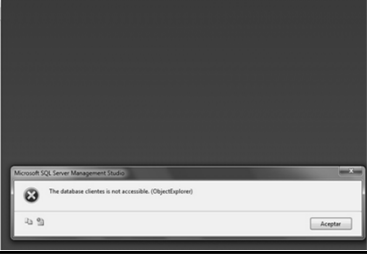
The database client is not accessible. (ObjectExplorer)

152

152

Supongamos el actor "maria" de autenticación de SQL Server

- No puede acceder a ninguna BD, porque no está autorizado en ninguna de ellas.



153

Es como si diéramos la llave de una casa a alguien

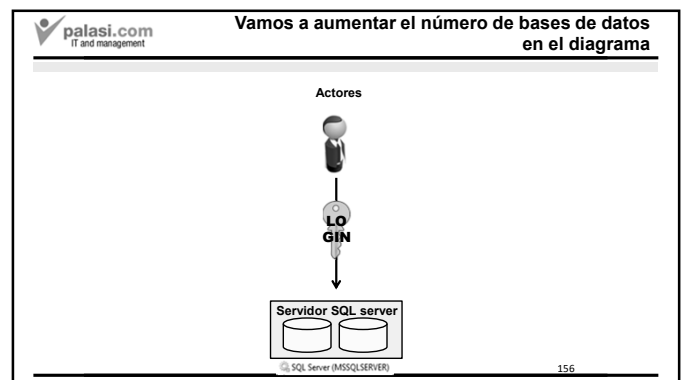
- Y puede entrar.
- Pero no puede tocar nada de lo que ya había allí.

154

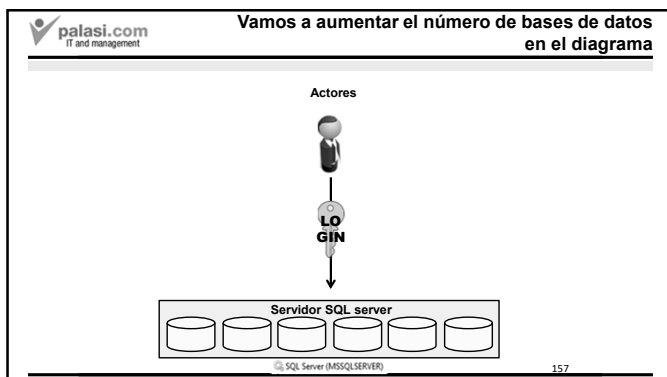
Pues vaya gracia

- SQL Server es muy cuidadoso con la seguridad de los datos, como debe ser.
- Todo lo que no se autoriza expresamente, está denegado.
- A esto se le llama "El principio del mínimo privilegio"

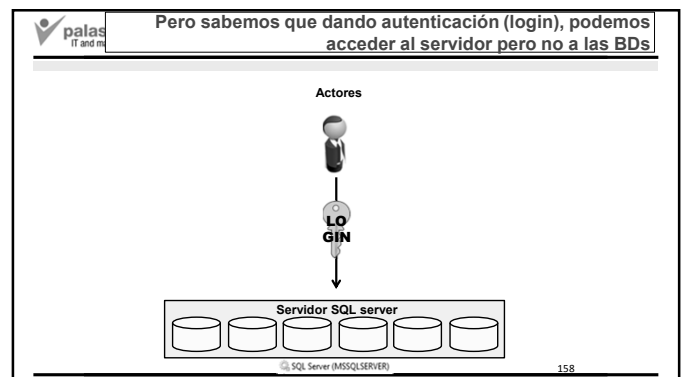
155



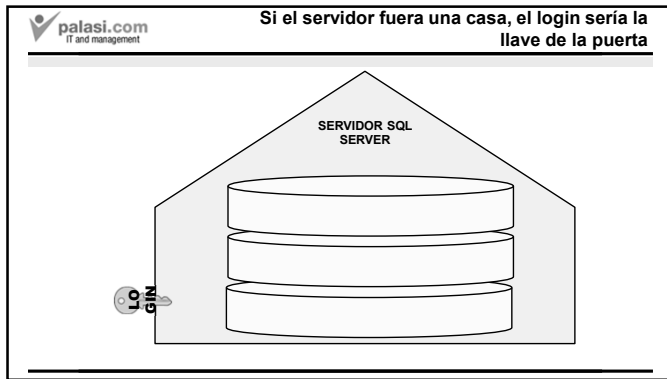
156



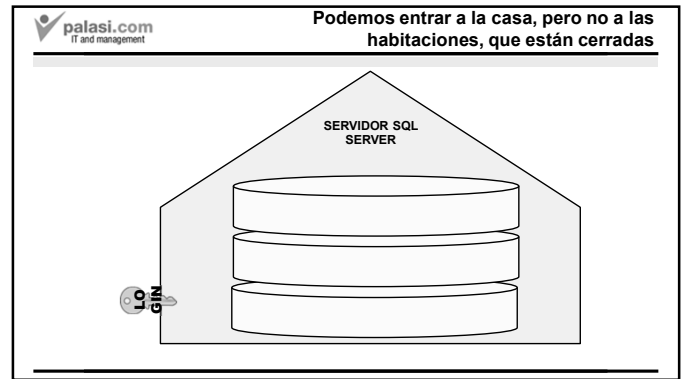
157



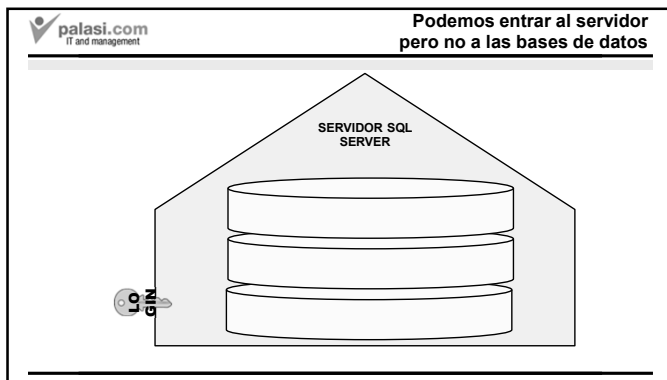
158



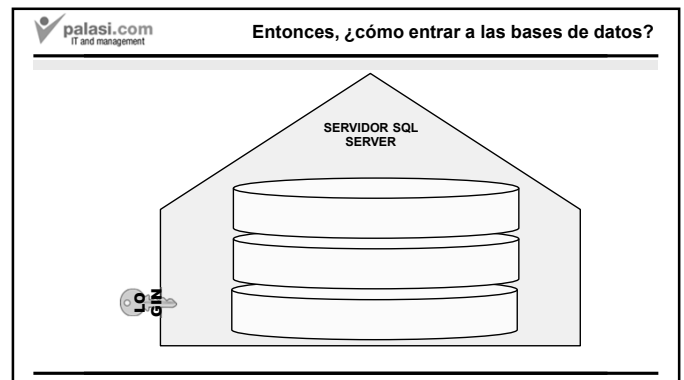
159



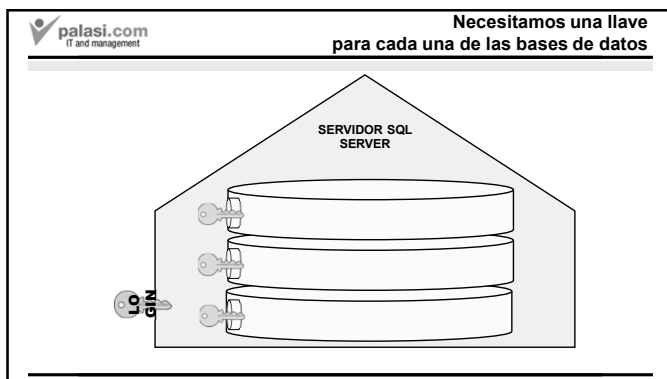
160



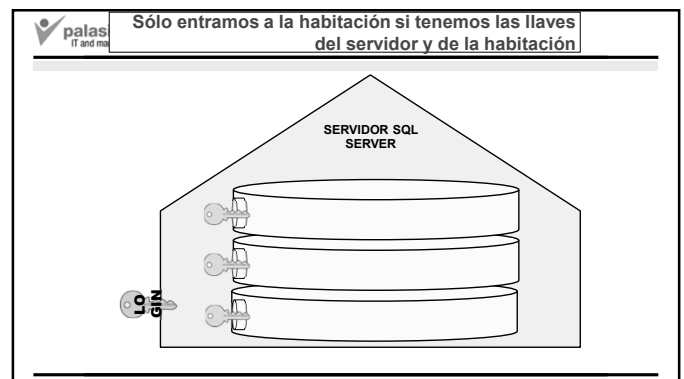
161



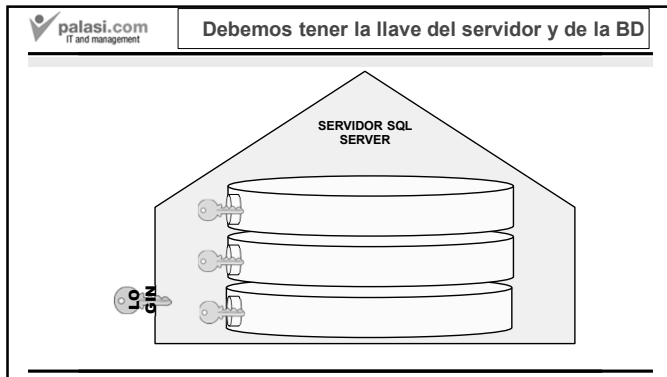
162



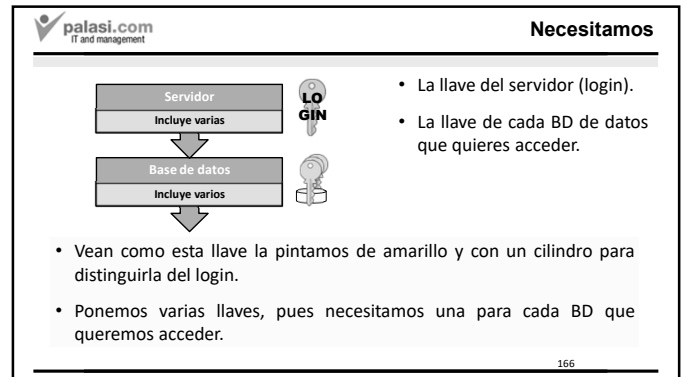
163



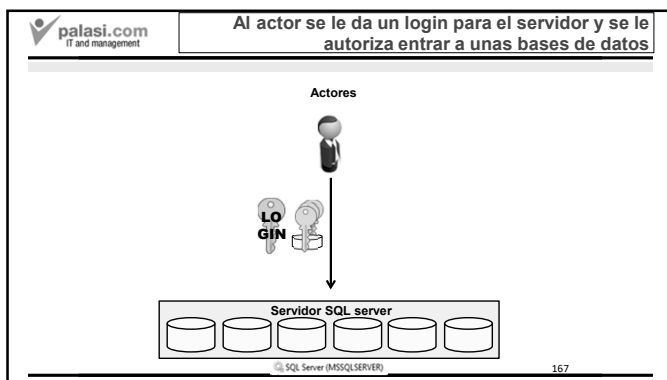
164



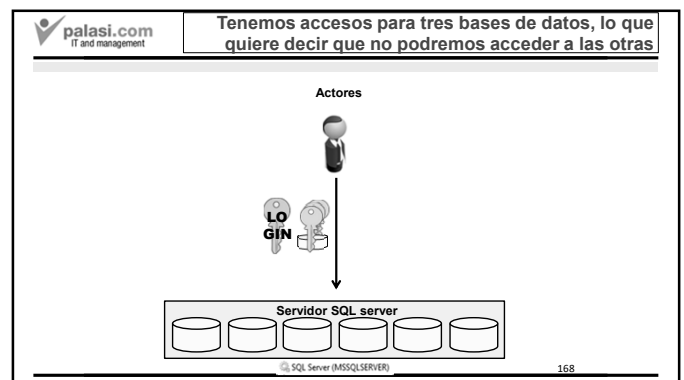
165



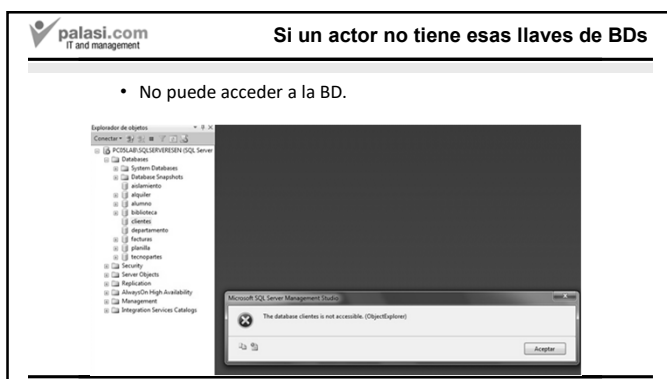
166



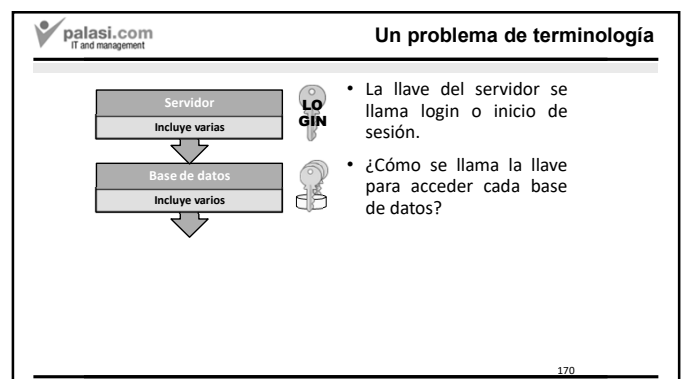
167



168

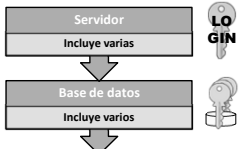


169



170

Pues la llave no tiene nombre



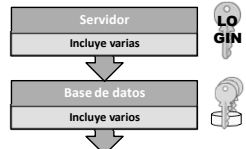
- Pero el actor que tiene la llave de una base de datos se dice que es **USUARIO** de esa base de datos.
- Se llama “usuario de una base de datos” al actor autorizado para acceder a esa base de datos.

- El nombre no lo tiene la llave sino el que la posee.

171

171

Pues la llave no tiene nombre



- Se llama “usuario de una base de datos” al actor autorizado para acceder a esa base de datos.

- Esto es confuso, pues la palabra “usuario” se usa en otras ocasiones.
- Hay que distinguir entre:
 - Usuario de Windows.
 - Usuario de una base de datos.

172

172


Terminología

- Si un actor está autorizado a acceder a una base de datos, se dice que es un **usuario de la base de datos**.
- Para que un actor esté autorizado a acceder a una base de datos.
 - Tiene que estar autorizado antes a acceder al servidor.
 - Por lo tanto tiene que tener un **login**
- No puedes entrar a la habitación si no tienes llave de la casa.

173

173

Listar los usuarios de una base de datos

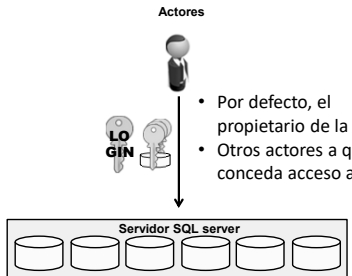


- Expandimos el nodo de la BD a la que queremos dar acceso y expandimos el nodo “Seguridad”.
- Expandimos el nodo “Usuarios” y veremos todos los actores que tienen acceso a la BD.

174

174

¿Quién tiene la llave (es usuario) de una BD concreta?

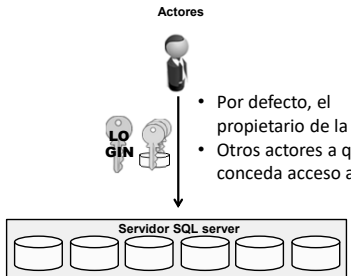


- Por defecto, el propietario de la BD.
- Otros actores a que se conceda acceso a la BD

175

175

¿Quién tiene la llave (es usuario) de una BD concreta?



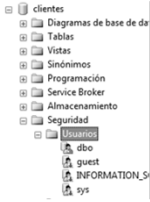
- Por defecto, el propietario de la BD.
- Otros actores a que se conceda acceso a la BD

176

176

palasi.com
IT and management

Listar los usuarios de una base de datos



- Entre ellos está el **dbo**, el **database owner**, el propietario de la BD.

177

177

palasi.com
IT and management

El actor que tiene dominio total sobre la base de datos se llama el propietario de esa BD

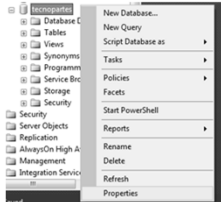
- El **database owner (o dbo)**.
- En principio, es el que creó la base de datos.
- Pero puede traspasar la propiedad a otros.
- En principio, sólo él tiene acceso a la base de datos. Pero se puede autorizar a otros.
- Por eso, desde el login "maria", no hay acceso a las bases de datos, ya que:
 - No es propietario de ninguna base de datos.
 - Tampoco le han autorizado.

178

178

palasi.com
IT and management

¿Cómo sabemos qué login es el usuario "dbo"?



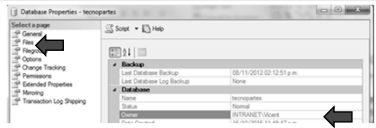
- Es decir, qué login tiene permisos totales sobre la base de datos.
- Hagamos clic derecho en la BD y seleccionamos "Propiedades".

179

179

palasi.com
IT and management

Conociendo el propietario de la BD



- Vemos donde aparece la información.
- En la pestaña "Archivos" podemos cambiar el propietario (pero no lo hagan)

180

180

palasi.com
IT and management

Muy bien, ya sabemos que el usuario "dbo"

- Es el propietario de la BD.
- Puede hacer con ella lo que quiera.

181

181

palasi.com
IT and management

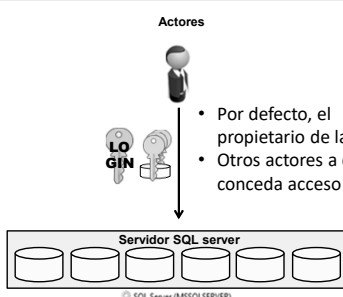
Si no se hace nada

- Sólo el propietario de la BD puede acceder a ella.
- Sin embargo, esto no es conveniente, pues una base de datos es algo compartido entre actores.
- ¿Cómo hacemos para que autorizar a otros actores al acceso a la BD?

182

182

¿Quién tiene la llave (es usuario) de una BD concreta?



Actores

- Por defecto, el propietario de la BD.
- Otros actores a que se conceda acceso a la BD

Servidor SQL server

SQL Server (MSSQLSERVER)

183

Si no queremos dar acceso a la BD

- Pues no hacemos nada.
- En SQL server, todo lo que no se permite está prohibido.

184

Cómo autorizar al acceso a BD en forma de texto

- Para autorizar el acceso a la base de datos.

```
use nombreBD
create user idUsuario for login clavelogin
```

- Para quitar esta autorización.

```
use nombreBD
drop user idUsuario
```

- Nota: si la **clavelogin** es de Windows hay que escribirla como [xxx\yyy]

185

Fijese que debe darse un nombre de usuario al login (puede ser el mismo que la clave del login)

- Para autorizar el acceso a la base de datos.

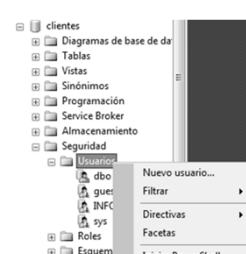
```
use nombreBD
create user idUsuario for login clavelogin
```

- Para quitar esta autorización.

```
use nombreBD
drop user idUsuario
```

186

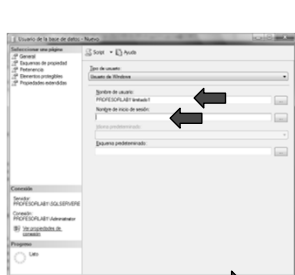
Gráficamente. En el nodo "Usuarios"



- Hacemos clic derecho y seleccionamos la opción "Nuevo usuario..."

187

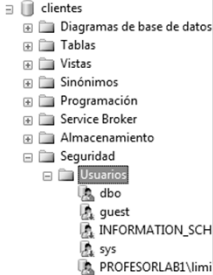
En la ventana que aparece



- En la ventana que aparece, elegimos "Usuario de Windows o de SQL Server".
- En "Nombre de inicio de sesión" escribimos el login.
- En "Nombre de usuario" escribimos el nombre de usuario (que puede ser el mismo que el login).

188

Vemos que el actor se le ha dado permiso a la BD

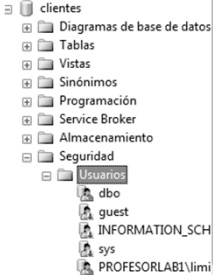


- Su nombre de usuario aparece entre los usuarios.

189

189

Para quitar ese usuario



- Simplemente clic derecho y "Eliminar".
- No desaparece el login (el actor puede conectarse al servidor) pero deja de estar autorizado para acceder a la base de datos.

190

190

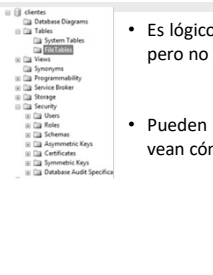
Ejercicio

- Con su login habitual, den acceso al login "maria" de SQL server a la base de datos "clientes".
- Después entren en el Studio con el login "maria".
- ¿Pueden ver la base de datos "clientes"? ¿Puede ver sus tablas?

191

191

¡Vemos la BD pero no vemos las tablas, ni las vistas, ni los proc. almacenados!



- Es lógico: nos han dado acceso a la base de datos, pero no a sus objetos.
- Pueden hacer también select * from clientes y vean cómo les deniega el permiso.

192

192

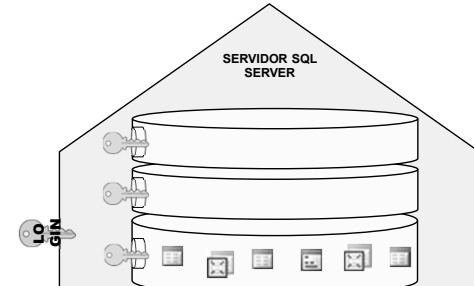
Da un mensaje de error

- ¿No es esto ilógico? ¿No habíamos permitido el acceso a la base de datos al login "maria"?
- En realidad, permitir el acceso a la BD, es sólo permitir que el actor la vea, pero no da derecho a ver ni modificar ningún objeto dentro de la base de datos.

193

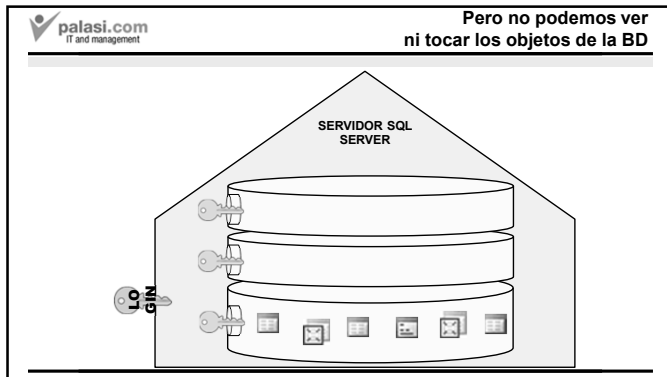
193

Podemos entrar en el servidor y en la BD

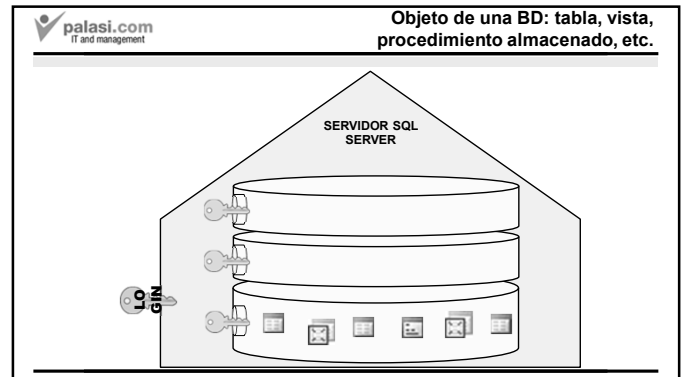


194

194



195



196

palasi.com
IT and management

Fíjense que ha pasado al actor "maria".

- Primero, le hemos dado acceso al servidor de SQL Server en nuestra máquina pero no podía ver el interior de las BDs.
 - Le dimos la puerta de la casa "servidor SQL Server", pero no de las habitaciones.
- Después le hemos dado acceso a la "BD cliente" pero no puede ver las tablas ni otros objetos.
 - Le dimos la puerta de la habitación "cliente" pero no puede ver ni tocar lo que hay dentro de la habitación.

197

197

palasi.com
IT and management

¿Por qué tanto problema?

- SQL Server, como la mayoría de SGBDs, tiene una filosofía de "lo que no me has dicho que está permitido, está prohibido". El principio del mínimo privilegio.
- Le dijimos que permitimos entrar en el servidor y en la BD "clientes", pero no le dijimos que podía tocar las tablas.
- Esto es para proteger los datos de la BD de hackers, de errores accidentales...
- Los datos son lo más crítico de una organización. Es por eso que se protegen tan bien.

198

198

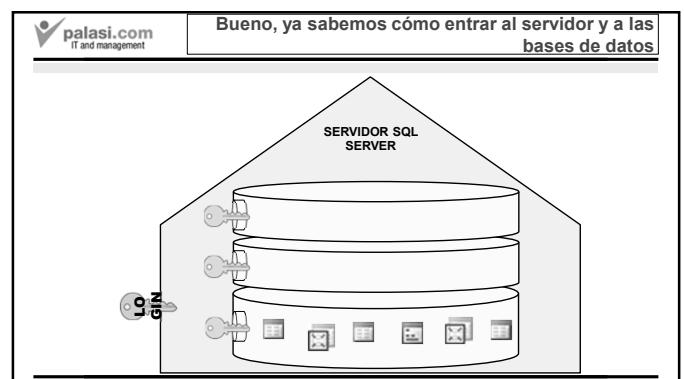
palasi.com
IT and management

Índice del tema 3

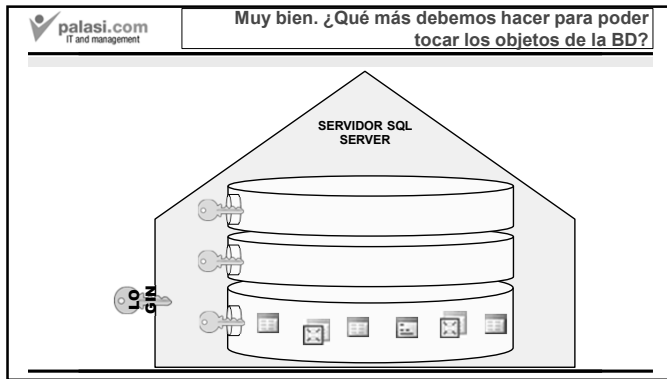
- 1. Introducción a la seguridad en SQL Server.
- 2. Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- 3. Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación
 - Roles de servidor.

199

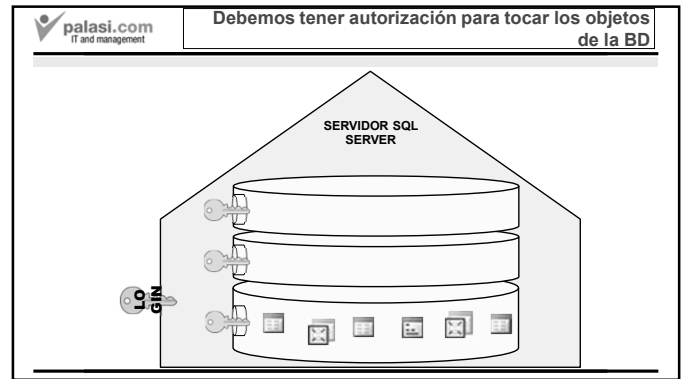
199



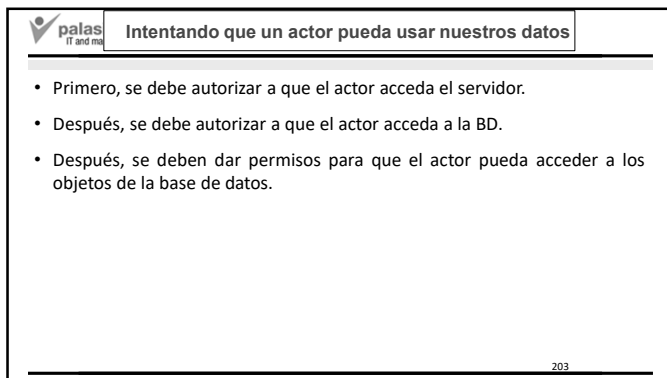
200



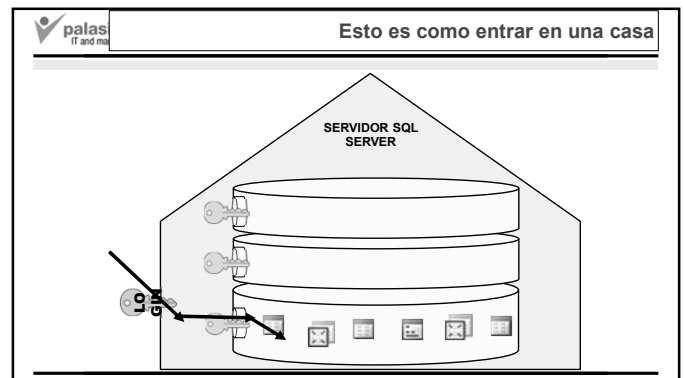
201



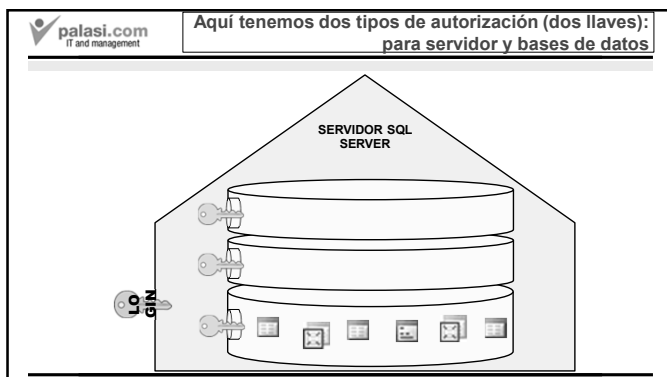
202



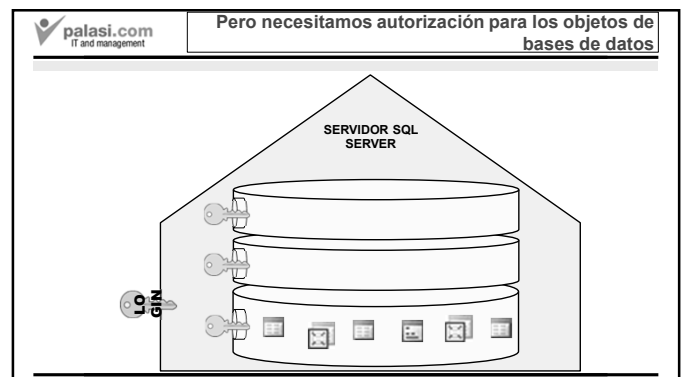
203



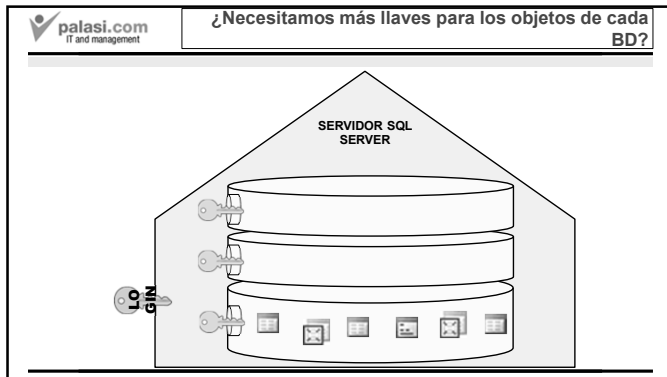
204



205



206



207

Pues no, es más complicado que una llave

- Una llave sólo tiene dos posibilidades permitidas:
 - La tienes y puedes acceder (entrar).
 - No la tienes y no puedes acceder.
- Pero un objeto de la BD (por ejemplo, una tabla) puede tener muchas posibilidades:
 - La podemos select, pero no podemos insert.
 - Podemos insert, pero no update.
 - Etc.

208



209

Lo que necesitamos se llama "permiso", un "derecho" o un "privilegio"

- Una llave sólo tiene dos posibilidades.
 - Tienes la llave y puedes entrar a la habitación.
 - No tienes la llave y no puedes entrar.
- Un permiso tiene muchas posibilidades. Una vez has entrado a la habitación.
 - Te doy permiso para leer la libreta que hay allí pero no para que escribas en ella.
 - Te doy permiso para que veas la tele pero no para que la cambies de sitio.
 - Un permiso es algo mucho más flexible, con más posibilidades

210

En nuestro caso

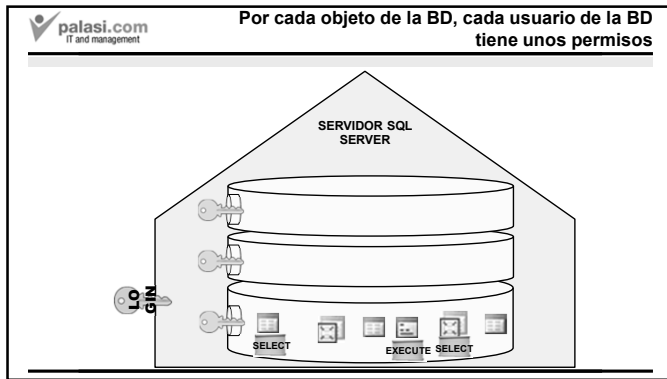
- Una base de datos .
 - Estás autorizado y puedes acceder.
 - No estás autorizado y no puedes acceder.
- Un objeto de la BD de datos.
 - Hay muchas posibilidades, así que tienes que decir que puedes hacer o qué no.

211

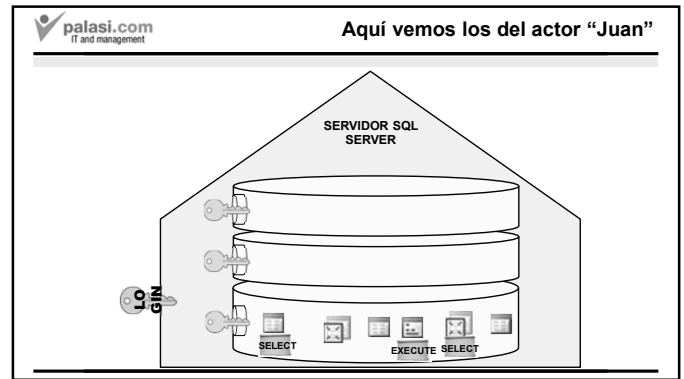
Según el principio de mínimo privilegio

- Todo lo que no está autorizado, está prohibido.
- Por lo tanto, no hace falta que escribamos lo que no se puede hacer.
- Sólo lo que se puede hacer.

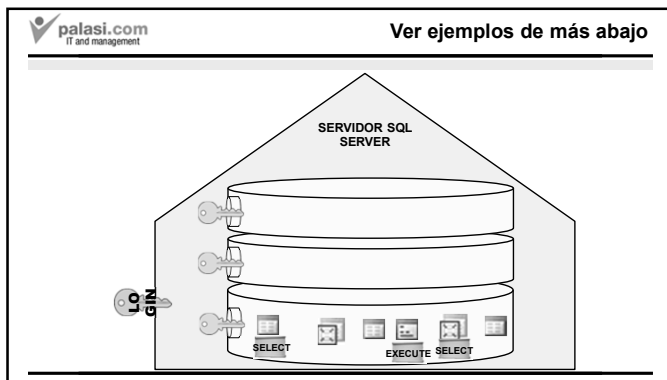
212



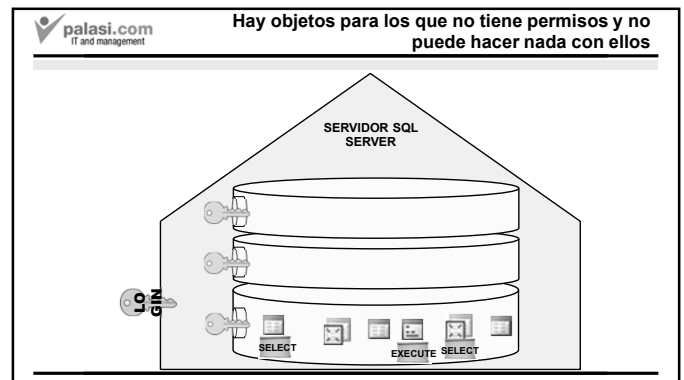
213



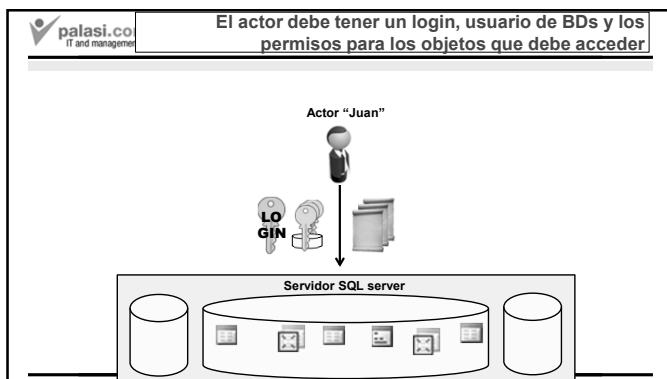
214



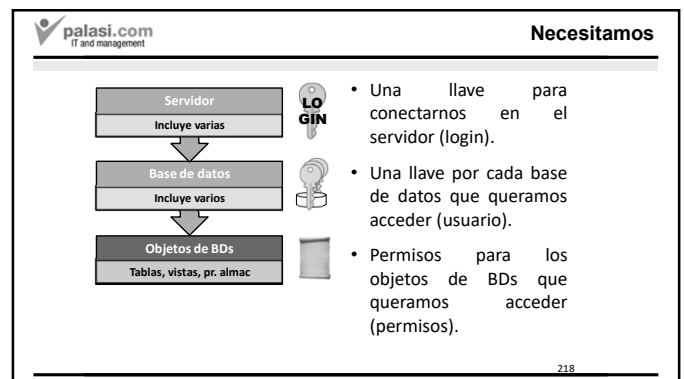
215



216



217



218

¿Qué posibles permisos se pueden dar en un objeto de la BD? Los más importantes

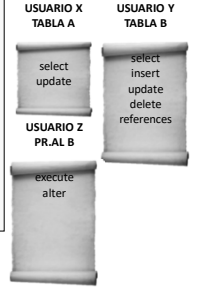
- **select.** Ver datos en una tabla, vista o campo
- **insert.** Añadir datos a una tabla o una vista.
- **update.** Modificar datos existentes en una tabla, vista o campo.
- **delete.** Eliminar datos de una tabla o de una vista.
- **execute.** Ejecutar un procedimiento almacenado.
- **references.** Hacer referencia a una tabla con una restricción de clave foránea.
- **take ownership.** Pasar a ser propietario del objeto.
- **alter.** Cambiar las propiedades del objeto (menos pasar a ser propietario).
- **view definition.** Ver la definición (los metadatos)
- **view change tracking.** Ver como cambió el objeto.

219

219

Puede ser ninguno o cualquier combinación de estos

- **select.**
- **insert.**
- **update.**
- **delete.**
- **execute.**
- **references.**
- **take ownership.**
- **alter.**
- **view definition.**
- **view change tracking.**

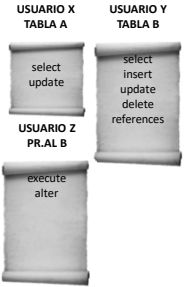


220

220

¿Cómo damos o quitamos estos permisos?

- **select.**
- **insert.**
- **update.**
- **delete.**
- **execute.**
- **references.**
- **take ownership.**
- **alter.**
- **view definition.**
- **view change tracking.**



221

221


Para dar acceso, debemos tener en cuenta varios conceptos

- Conceder (grant): Dar un derecho.
- Denegar (deny): Niego un derecho.
- ¿Qué pasa si concedemos un derecho y lo denegamos al mismo tiempo?
- La denegación tiene prioridad, pues SQL Server es muy cuidadoso con los datos.

222

222

Un ejemplo de conceder y denegar un "permiso para matar"



- A James Bond se le concedió "Licencia para matar" y se le denegó al mismo tiempo.
- Por lo tanto, no puede matar a nadie, pues la denegación tiene preferencia.

223

223


Pero tanto la concesión como la denegación se pueden deshacer (revocar)

- Conceder (grant): Dar un derecho.
- Denegar (deny): Rechazar un derecho.
- Revocar (revoke): Deshacer una decisión sobre un derecho.
 - Si se revoca una concesión, la parte ya no tiene un derecho.
 - Si se revoca una denegación, la parte ya no se le deniega un derecho.

224

224

Revocar la concesión



- A James Bond se le concedió "Licencia para matar". Derecho a matar a cualquiera
- A James Bond se le revocó la "Licencia para matar". Ya no tenía derecho a matar a cualquiera. Se deshizo la primera decisión

225

225

Revocar la denegación



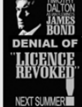


- A James Bond se le denegó la "licencia para matar". Ya no podía matar a nadie, incluso si le concedían otra licencia
- Después se le revocó esta denegación. Es como si la denegación no hubiera existido.

226

226

Entonces, tenemos tres acciones sobre los permisos

	Realizar una acción	Deshacerla
Acción: Conceder un permiso	Grant 	Revoke 
Acción: Denegar un permiso	Deny 	Revoke 

227

Para conceder derechos


```
grant listaPermisos on objeto to idUsuario
```

Concede los permisos sobre el objeto al usuario de BD. **all** significa todos los permisos.

Ejemplos:

```
grant select, insert on Empleados to Pedro
grant execute on procAlmacenado to Rosa
grant all on alumnos to Juan
```

Pedro EMPLEADOS




228

Para denegar derechos sobre un objeto

```
deny listaPermisos on objeto to idUsuario
```

```
deny select, insert on Departamentos to Rosa
```

Rosa DEPARTAMENTOS




229

Para revocar derechos sobre un objeto

```
revoke listaPermisos on objeto to idUsuario
```


- grant select, insert on Empleados to Pedro
- revoke select on Empleados to Pedro

Pedro EMPLEADOS



→

Pedro EMPLEADOS

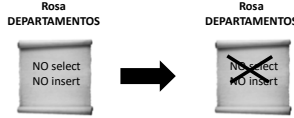


230

Para revocar derechos sobre un objeto

```
revoke listaPermisos on objeto to idUsuario
```

- deny select, insert on Departamentos to Rosa
- revoke select on Departamentos to Rosa



231

Entonces, tenemos estas posibilidades sobre los permisos

	Realizar una acción	Deshacerla
Acción: Conceder un permiso	grant listaPermisos on objeto to idUsuario Pedro EMPLEADOS select insert	revoke listaPermisos on objeto to idUsuario Pedro EMPLEADOS select insert
Acción: Denegar un permiso	deny listaPermisos on objeto to idUsuario Rosa DEPARTAMENTOS NO select NO insert	revoke listaPermisos on objeto to idUsuario Rosa DEPARTAMENTOS NO select NO insert

232

Normalmente, en SQL Server

- Nada tiene derecho si no se lo damos.
- Si no queremos que no tenga derecho, no se hace nada.
- Si queremos que se le conceda, se le hace un grant.
 - Si después ya no queremos, se le hace un revoke.

233

Esto es lo más normal

	Realizar una acción	Deshacerla
Acción: Conceder un permiso	grant listaPermisos on objeto to idUsuario Pedro EMPLEADOS select insert	revoke listaPermisos on objeto to idUsuario Pedro EMPLEADOS select insert

234

Entonces, ¿para qué necesito el deny?

	Realizar una acción	Deshacerla
Acción: Conceder un permiso	grant listaPermisos on objeto to idUsuario Pedro EMPLEADOS select insert	revoke listaPermisos on objeto to idUsuario Pedro EMPLEADOS select insert
Acción: Denegar un permiso	deny listaPermisos on objeto to idUsuario Rosa DEPARTAMENTOS NO select NO insert	revoke listaPermisos on objeto to idUsuario Rosa DEPARTAMENTOS NO select NO insert

235

El uso del deny

- Deny sólo se usa
 - cuando se necesita prohibir algo realmente importante, contra errores.
 - Cuando se hace grant y no se puede hacer un revoke (por motivos que no veremos por el momento).

236

Primera ocasión. Cuando queremos protegernos contra errores

- Si hay algo especialmente delicado, pues puede ser que en el futuro hagamos grant por error.
- Si ponemos deny, ningún grant posterior tendrá efecto (pues deny tiene prioridad) y podemos dormir tranquilos.

237

237

Segunda excepción. Cuando el revoke no es igual que el grant

- Por la forma en que está construido SQL Server, un revoke sólo tiene efecto si es igual que el grant.
- Veamos un ejemplo.

238

238

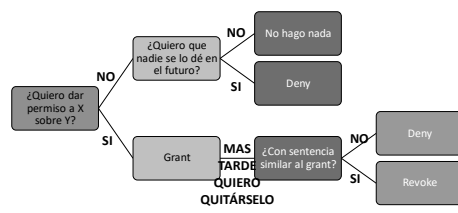
Supongamos que hay un grupo de usuarios "oficinistas" (ya veremos cómo se hace esto).

- Dentro de ese grupo, hay un usuario llamado "oficinista1".
- Ejemplo que funciona:
 - grant select on Empleado to oficinistas
 - revoke select on Empleado to oficinistas
- Ejemplo que NO funciona:
 - grant select on Empleado to oficinistas
 - revoke select on Empleado to oficinista1
- Ejemplo que funciona:
 - grant select on Empleado to oficinistas
 - deny select on Empleado to oficinista1

239

239

Normalmente



240

240

Vamos ahora con algunos detalles

- Sobre lo que acabamos de ver.

241

241

No es bueno asignar permisos a un campo

- **select.** Ver datos en una tabla, vista o campo
- **insert.** Añadir datos a una tabla o una vista.
- **update.** Modificar datos existentes en una tabla, vista o campo.
- **delete.** Eliminar datos de una tabla o de una vista.
- **execute.** Ejecutar un procedimiento almacenado.
- **references.** Hacer referencia a una tabla con una restricción de clave foránea.
- **take ownership.** Pasar a ser propietario del objeto.
- **alter.** Cambiar las propiedades del objeto (menos ser propietario).
- **view definition.** Ver la definición (los metadatos)
- **view change tracking.** Ver como cambió el objeto.

242

242

No es bueno asignar permisos a un campo

- Aunque se pueden establecer permisos sobre campos, esto no es recomendable y no lo veremos. Es más recomendable crear una vista con las campos y establecer permisos sobre ella.
- Mucho trabajo. Por cada usuario, deberemos determinar sus permisos sobre cada campo (en cambio, con una vista, sólo se define un permiso sobre la vista)
- Más flexible. Si queremos prohibir o permitir campos deberemos hacerlo con cada usuario. Con una vista, sólo debemos añadir o retirar campos sobre la vista.

243

Nota: Procedimientos almacenados

- Si el usuario tiene el permiso **execute** puede ejecutar un procedimiento almacenado.
- Sin embargo, el procedimiento almacenado accede a otros objetos de la base de datos.
- ¿Qué derechos tiene el procedimiento almacenado sobre los objetos que accede?
- Fácil. Los mismos que el usuario que ejecuta el procedimiento almacenado.

244

Nota: Triggers

- No se puede conceder permisos de ejecución a los triggers como se hace con los procedimientos almacenados.
- Los triggers se ejecutan automáticamente cuando se ejecuta la actualización para la que están definidos (insert, update, delete).
- Sin embargo, los triggers pueden actualizar objetos de BDs.
- ¿Qué permisos tienen sobre esos objetos? Los mismos que el usuario que realizó la actualización.

245

Recordemos: para conceder permisos

```
grant listaPermisos on objeto to idUsuario
```

Concede los permisos sobre el objeto al usuario de BD. **all** significa todos los permisos.

Ejemplos:

```
grant select, insert on Empleados to Pedro
grant execute on procAlmacenado to Rosa
grant all on alumnos to Juan
```

Pedro EMPLEADOS
select
insert

246

"With grant option" significa que el usuario puede conceder los permisos a otros

```
grant listaPermisos on objeto to idUsuario with grant option
```

Ejemplos:

```
grant select, insert on Empleados to Pedro with grant option
```

Quiere decir que Pedro puede dar los derechos de **select** e **insert** en la tabla Empleados a otro usuario (por ejemplo, a Rosa)

A esto se le llama "conceder el permiso de forma transitiva".

Pedro EMPLEADOS
select
insert

247

"With grant option" significa que el usuario puede conceder los permisos a otros

```
grant select, insert on Empleados to Pedro with grant option
```

Comparemos con:

```
grant update on Empleados to Pedro
```

Pedro tiene los derechos de **select**, **insert** y **update**.

Pedro sólo puede dar los derechos de **select** e **insert** pero no los de **update**.

248

Nota

- El dbo puede conceder permisos a cualquiera.
- Los otros pueden conceder los permisos que alguien les ha concedido "with grant option".
- Hay que decir que "with grant option" presenta problemas de seguridad y debe concederse con cuidado.

249

249

Ejercicios

- En la base de datos "planilla", concedan al login "maria"
 - permiso para leer la tabla Departamentos (y para otorgarlo de forma transitiva).
 - permiso para insertar y actualizar Empleados.
- Una vez concedidos, revoken el permiso de actualizar Empleados

250

250

Solución

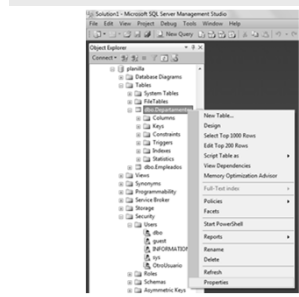
```
use planilla
grant select on Departamentos to maria with grant option
grant insert, update on Empleados to maria

revoke update on Empleados to maria
```

251

251

Gráficamente

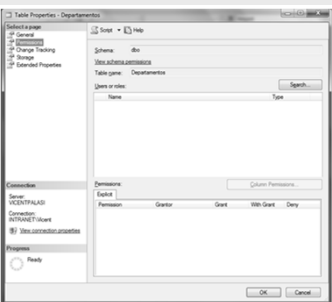


- Clic derecho en el objeto de la BD y clic en "Propiedades".

252

252

En la ventana que sale, elegir "Permisos"

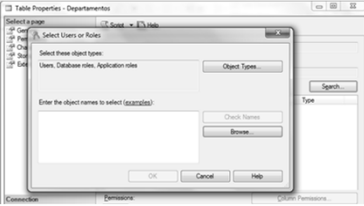


- A la izquierda.

253

253

Si quieres definir los permisos de ese objeto para un usuario



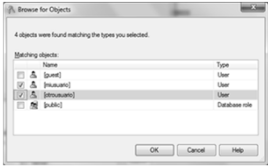
- Debemos buscar el nombre usuario en la BD.
- Hacemos clic en "Search...". En la ventana que sale:
 - O bien pones el nombre en el cuadro.
 - O haces "Browse..." que es lo que haremos.

254

254

Si hacemos "Browse"

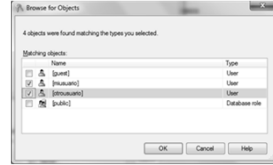
- Aparecen todos los usuarios definidos en la base de datos.
- Podemos seleccionar a aquellos que queremos definir sus permisos.



255

Por ahora no se fijen en "public" ni en "guest", que es algo que veremos después

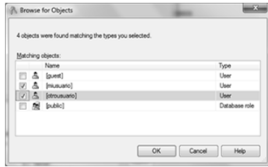
- Pero podemos seleccionar los otros usuarios que aparecen.



256

Aquí tenemos el usuario

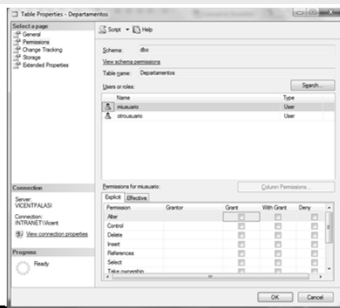
- Aparecen todos los usuarios definidos en la base de datos.
- Podemos seleccionar a aquellos que queremos definir sus permisos.



257

Si hacemos clic en "Aceptar dos veces"

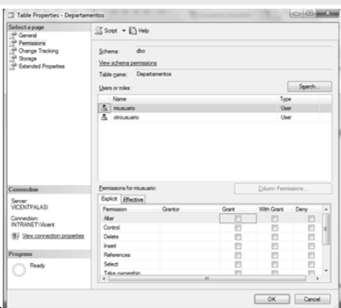
- Aparece una ventana como ésta.



258

Aquí elegimos uno de los usuarios

- Y bajo podemos elegir los permisos sobre ese objeto de la BD.
- Como vemos, en principio no tienen ningún permiso



259

Recordemos: estos son los permisos

- **select.** Ver datos en una tabla, vista o campo
- **insert.** Añadir datos a una tabla o una vista.
- **update.** Modificar datos existentes en una tabla, vista o campo.
- **delete.** Eliminar datos de una tabla o de una vista.
- **execute.** Ejecutar un procedimiento almacenado.
- **references.** Hacer referencia a una tabla con una restricción de clave foránea.
- **take ownership.** Pasar a ser propietario del objeto.
- **alter.** Cambiar las propiedades del objeto (menos ser propietario).
- **view definition.** Ver la definición (los metadatos)
- **view change tracking.** Ver como cambió el objeto.

260

259

260

Vicent Palasí, PhD, MBA, MEd. Todos los derechos reservados.

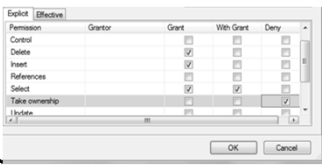
Mail: palasi@palasi.com Web: www.palasi.com

Aquí aparecen todos los permisos en filas

- En las columnas, aparece qué hacemos con cada uno de los permisos:
 - Concederlo sin transividad (grant). Ej. Delete.
 - Concederlo de forma transitiva (grant with grant option). Ej. Select.
 - Denegarlos (deny). Ej. Take ownership.

Pedro EMPLEADOS

select
insert



OK Cancel


261

261

¿Qué tenemos aquí?

**miusuario
otrouuario
Departamentos**

select (tr)
insert
delete
NO take ownership



OK Cancel

262

262

¿Cómo revocamos un permiso?

- Simplemente, hay que quitar el "chequecito"

**miusuario
otrouuario
Departamentos**

select (tr)
insert
~~delete~~
~~NO take ownership~~



OK Cancel

263

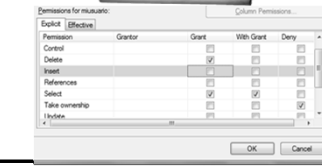
263

¿Cómo revocamos un permiso?

- Simplemente, hay que quitar el "chequecito"

**miusuario
otrouuario
Departamentos**

select (tr)
insert
NO take ownership



OK Cancel

264

264

Ejercicio sobre la BD planilla

- Quiten todos los permisos que le han dado al login "maria" anteriormente.
 - permiso para leer la tabla Departamentos (y para otorgarlo de forma transitiva).
 - permiso para insertar Empleados.
- Pongan los mismos permisos al login "limited2".
- Háganlo de forma gráfica.

265

265

Crear una política de una empresa que usa la BD planilla de esta manera

- A. Hay un único DBA que, obviamente, puede verlo todo y modificarlo todo.
- B. Hay una secretaria que puede ver en Excel los datos de los departamentos y empleados (excepto sus salarios y comisiones), pero no modificarlos.
- C. Hay dos contadores: uno en Santa Ana y otra en la central de San Salvador (que cubre el resto del país), que usan dos programas diferentes, los cuales pueden ver y modificar todos los datos de su área geográfica.

266

266

Vicent Palasí, PhD, MBA, MEd. Todos los derechos reservados.

Mail: palasi@palasi.com Web: www.palasi.com

palasi.com
IT and management

EI DBA (A)

- **A. Hay un único DBA que, obviamente, puede verlo todo y modificarlo todo.**
 - A este le vamos a dar la contraseña del login sa.
 - O bien la creó el o el DBA anterior se la da.

267

267

palasi.com
IT and management

La secretaria (B)

- **B. Hay una secretaria que puede ver en Excel los datos de los departamentos y empleados (excepto sus salarios y comisiones), pero no modificarlos.**
 - B1. Se crea un login y usuario para la secretaria en la BD (de Windows).
 - De Windows por facilidad y porque el usuario de Windows de la secretaria es personal
 - B2. Se crea una vista de Empleados y Departamentos SIN SALARIOS NI COMISIONES
 - No necesita triggers porque sólo la vamos a leer.
 - B3. Se le da permiso al usuario secretaria de leer esa vista.

268

268

palasi.com
IT and management

Secretaria (B1)

- B1. Se crea un login y usuario para la secretaria en la BD (de Windows).

```
create login [EMPLEADOS\secretaria] from windows
use planilla
create user secretaria for login [EMPLEADOS\secretaria]
```

269

269

palasi.com
IT and management

Secretaria (B2)

- B2. Se crea una vista de Empleados y Departamentos SIN SALARIOS NI COMISIONES

```
create view EmpleadosConDepartamento as
select Departamentos.NombreDep, Departamentos.Ciudad,
Empleados.Apellido, Empleados.Nombre, Empleados.Trabajo,
Empleados.Jefe, Empleados.FechaInicio
from Departamentos inner join Empleados on Departamentos.IdDep
= Empleados.Departamento
with check option
```

270

270

palasi.com
IT and management

Secretaria (B3)

- B3. Se le da permiso al usuario secretaria de leer esa vista.

```
grant select on EmpleadosConDepartamentos to secretaria
```

271

271

palasi.com
IT and management

Los contadores (C)

- **C. Hay dos contadores: uno en Santa Ana y otra en la central de San Salvador (que cubre el resto del país), que usan dos programas diferentes, los cuales pueden ver y modificar todos los datos de su área geográfica.**
 - C1. Se crean login y usuario para los programas (de SQL Server)
 - De SQL Server, pues el usuario de Windows puede tener más programas y un humano.
 - C2. Se crean dos vistas para cada tabla, una para área geográfica.
 - Dos para cada tabla porque el programa supondrá un modelo relacional.
 - C3. Se da acceso a los datos a cada programa del contador para estas vistas.

272

272

Contadores (C1)

- C1. Se crean login y usuario para los programas (de SQL Server)

```
create login prcontaelsalvador with password = '_____'
create login prcontasantaana with password = '_____'
use planilla
create user prcontaelsalvador for login prcontaelsalvador
create user prcontasantaana for login prcontasantaana
```

273

273

C2. Se crean dos vistas para cada tabla, una para área

```
create view DepartaelSalvador as
select * from Departamentos where Ciudad <> 'Santa Ana'
with check option
create view DepartasantaAna as
select * from Departamentos where Ciudad = 'Santa Ana'
with check option
create view EmpleaelSalvador as
select Empleados.*
from Departamentos inner join Empleados on
on Departamentos.IdDep = Empleados.Departamento
where Departamentos.Ciudad <> 'Santa Ana'
with check option
create view EmpleasantaAna as
select Empleados.*
from Departamentos inner join Empleados on
on Departamentos.IdDep = Empleados.Departamento
where Departamentos.Ciudad = 'Santa Ana'
with check option
```

274

274

Contadores (C3)

- C3. Se da acceso a los datos a cada programa del contador para estas vistas.

```
grant select, insert, delete, update on DepartaelSalvador to prcontaelsalvador
grant select, insert, delete, update on EmpleaelSalvador to prcontaelsalvador

grant select, insert, delete, update on DepartasantaAna to prcontasantaana
grant select, insert, delete, update on EmpleasantaAna to prcontasantaana
```

275

275

Pregunta

- ¿Cómo cambiaría el diseño si los contadores usaran un mismo programa?

276

276

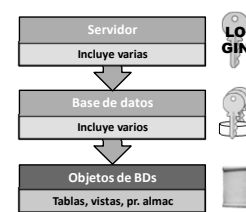
Índice del tema 3

- 1. Introducción a la seguridad en SQL Server.
- 2. Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- 3. Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación
 - Roles de servidor.

277

277

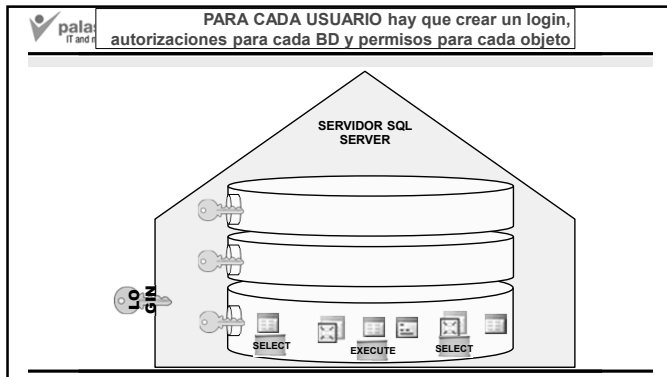
Manejar la seguridad en SQL Server (por lo que sabemos hasta ahora)



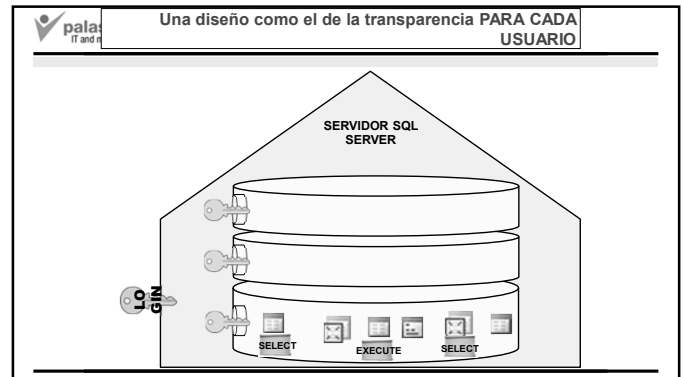
- 1. Definimos los logines
 - en Windows y después los autorizamos para SQL Server
 - o en SQL Server directamente.
- 2. Para cada base de datos, se definen qué logines está permitido que accedan (creando un usuario)
- 3. Para cada objeto de BD, se define
 - Qué usuarios pueden acceder.
 - Con qué permisos

278

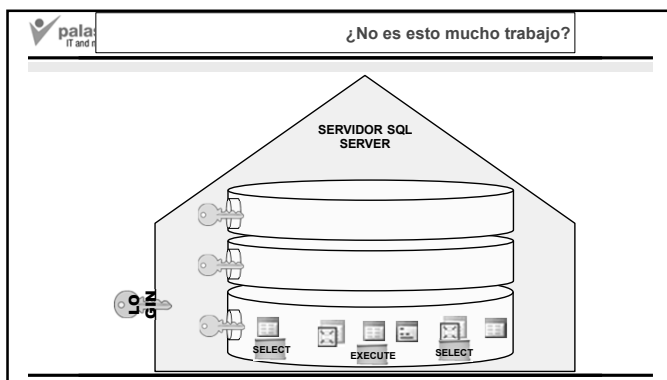
278



279



280



281



282

Por eso SQL Server da formas de rebajar el trabajo

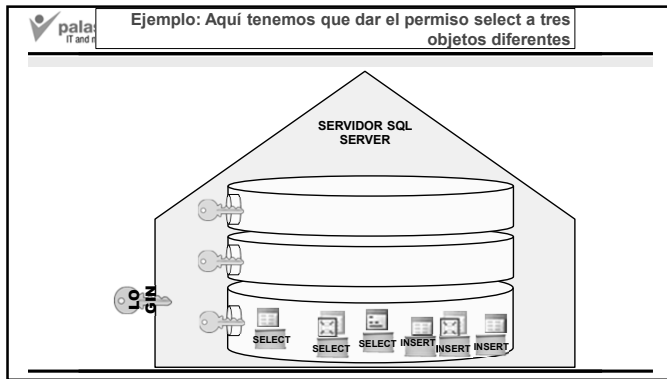
- Hay varias formas.
- Una es agrupar los objetos de bases de datos en grupos, llamados "esquemas".
- Así podemos dar los mismos permisos a todo un esquema, en vez de darlo a cada uno de los objetos

283

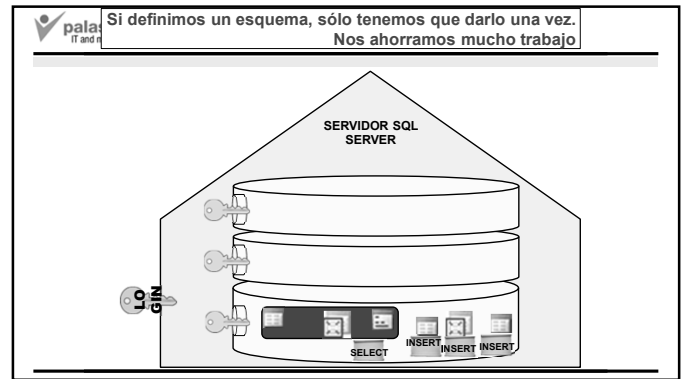
Índice del tema 3

- 1. Introducción a la seguridad en SQL Server.
- 2. Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- 3. Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación
 - Roles de servidor.

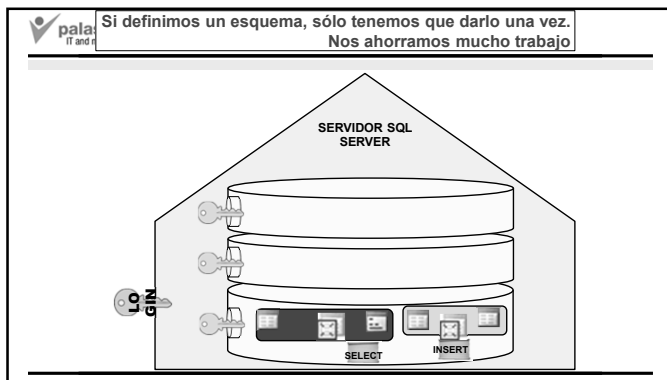
284



285



286



287

No hay objetos sin esquema

- Todo objeto de una base de datos pertenece a un esquema.
- Si no decimos otra cosa, SQL Server lo pone en el esquema por defecto, que se llama **"dbo"**.

288

Hay que distinguir

- El usuario **dbo**
 - Es el propietario de la base de datos.
 - Puede hacer lo que quiera con la base de datos.
- El esquema **dbo**
 - Es el esquema por defecto de la BD.
- Ya veremos que están relacionados pero son cosas diferentes.

289

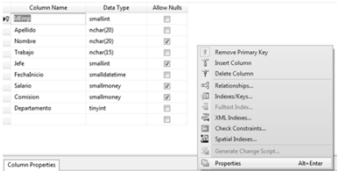
En realidad, estos nombres no son idóneos

- El usuario **dbo** es un buen nombre.
- El esquema **"dbo"** debería haberse llamado esquema **"default"**
- Se llama así por compatibilidad con versiones anteriores.
- Pero esto da motivo a confusiones.

290

Compruébenlo

- Hagan clic derecho en la tabla Empleados y seleccionen "Diseño".
- En la parte en blanco que aparece, hagan clic derecho y seleccionen "Propiedades".

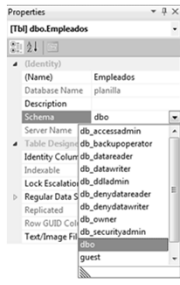


291

291

Verán qué les dice qué esquema es y les deja cambiarlo

- Ven que es el esquema por defecto: **dbo**
- No lo cambien.
- Simplemente, salgan.

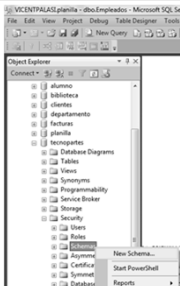


292

292

Vamos a crear un esquema en la BD tecnopartes

- Expandan la BD.
- Expandan "Seguridad".
- Clic derecho a "Esquemas".
- Elijan "Nuevo esquema".

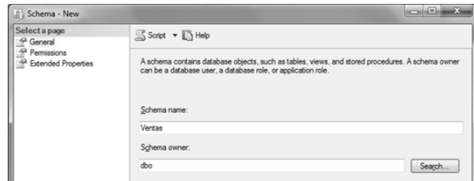


293

293

Veamos

- Pongamos "ventas" al esquema
- El propietario del esquema es quien tiene todos los derechos sobre el esquema por defecto.

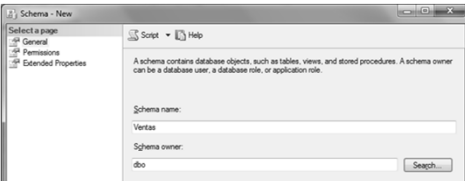


294

294

Pondremos al usuario dbo como propietario del esquema

- Pero podría ser cualquier otro usuario, pero pongamos a dbo.

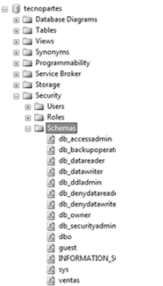


295

295

Se ha creado el esquema "ventas"

- Ahora vamos a poner las tablas "Comprars", "Partes", "Clientes" en ese esquema.
- Están en el esquema dbo, que es el esquema por defecto.



296

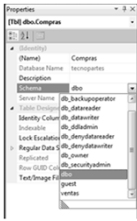
296

Vicent Palasí, PhD, MBA, MEd. Todos los derechos reservados.

Mail: palasi@palasi.com Web: www.palasi.com

Hagamos primero la tabla "Compras"

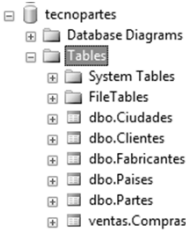
- Abran la vista "Diseño" de la tabla, hagan clic derecho en la parte en blanco y elijan "Propiedades".
- Cambian "esquema a ventas".
- Guarden la tabla y **refresquen el servidor**.
- Háganlo para las tablas "Compras", "Partes", "Clientes" en ese esquema.



297

¿Por qué la tabla "Compras" no tiene dbo delante del nombre?

- ¿Qué les parece?



298

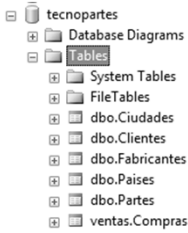
El nombre completo de una tabla en una base de datos

nombreEsquema.nombreTabla

Es único en la base de datos.

Por eso, todas las tablas llevaban "dbo" hasta ahora.

Es el esquema dbo, esquema por defecto, donde se crean todas las tablas si no decimos otra cosa.




299

¿Qué pasa si no ponemos el esquema?

nombreEsquema.nombreTabla

- Él busca por el esquema por defecto que tiene el usuario. Mirar las propiedades de usuario en la BD.
- Si no lo encuentra ahí, busca en el esquema por defecto en la BD, que suele ser esquema dbo.



300

Muy bien, ahora queremos que un usuario "miusuario" sólo pueda leer estas tablas

- "Compras", "Partes", "Clientes"
- Pero no modificarlas.
- ¿Cómo podríamos hacerlo?
- Una forma de hacerlo es dar al usuario "miusuario" permiso **select** tabla por tabla pero esto....

301

...muy cuesta

Muy cuesta




302

301

302

Es mejor poner estos permisos en el esquema

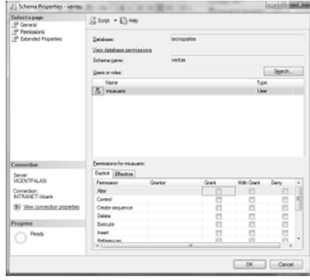
- Vayan al esquema, hagan clic derecho y elijan "Propiedades". A la izquierda elijan "Permisos".



303

303

Elijan "miusuario" con "Search"



- Aquí podemos poner los permisos, que son los mismos que para los objetos separados.
- Así nos ahorramos trabajo.

304

304

En forma de texto

- Para crear un esquema (authorization es opcional)

```
create schema nombre authorization propietario
```

- Puedes crear una tabla en este esquema con

```
create table nombreesquema.nombretabla...
```

- Puedes agregar tablas existentes a este esquema con

```
alter schema nuevoesquema transfer viejoesquema.nombretabla
```

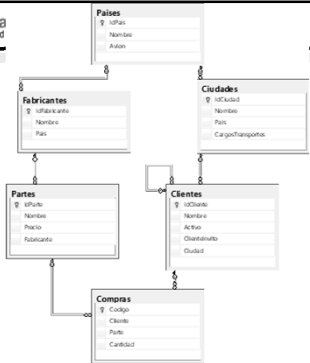
- Puedes dar permisos a este esquema con

```
grant/revoke/deny permisos on schema::nombreesquema to usuario
```

305

305

BD tecno partes



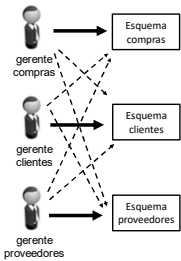
- Países.
- Ciudades.
- Fabricantes.
- Partes.
- Clientes.
- Compras (esta es una tabla de enlace, que representa una relación N-N).

306

306

Ejercicio

- Crean tres logines de SQL Server: gerentecompras, gerenteclientes, gerenteproveedores.
- Autoricenlos a la BD "tecno partes" como usuarios con esos mismos nombres.
- Crean tres esquemas en la BD tecno partes:
 - compras, compuesto de tablas "compras" y "partes"
 - clientes, compuesto de tablas "clientes" y "ciudades".
 - proveedores, compuesto de tablas "fabricantes" y "países".
- Las flechas rojas son permisos de lectura/escritura sobre registros, las negras de solo lectura



307

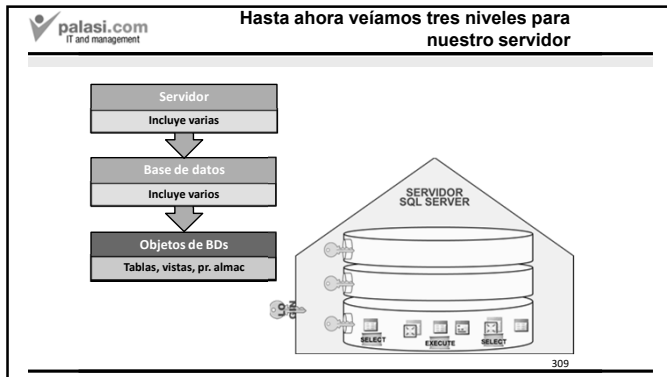
307

Nota: ¿Cuándo usar esquemas diferentes del por defecto?

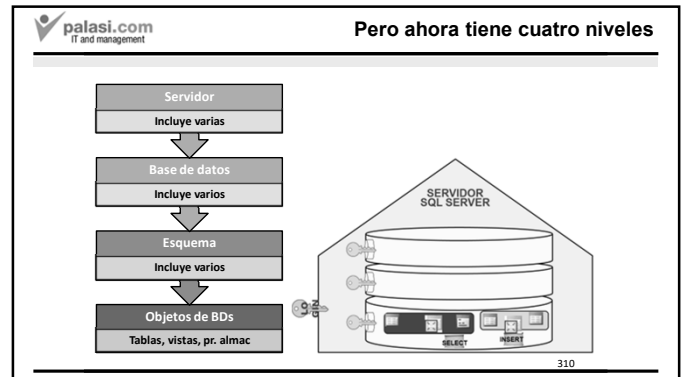
- Cuando hay varios objetos de la base de datos que se tratan con el mismo permiso.
- Se pueden combinar:
 - Pones permisos de lectura para un esquema que contiene 50 objetos.
 - Para uno de estos objetos, pones permisos de escritura.
- Esto es combinar permisos sobre objetos y sobre esquemas.

308

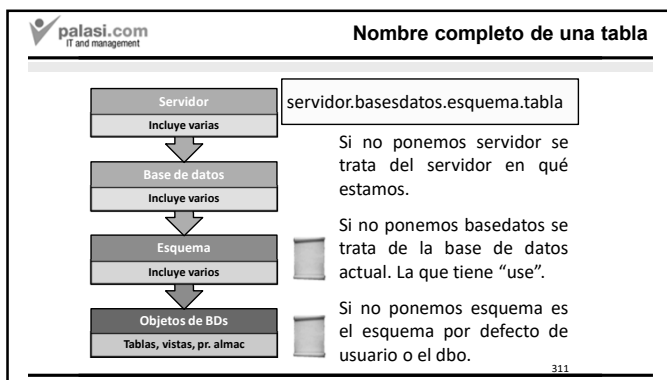
308



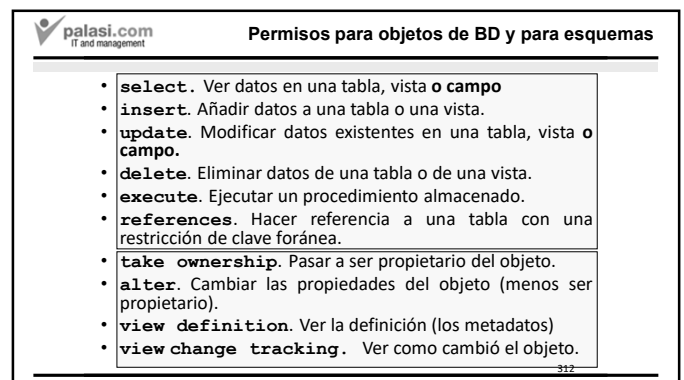
309



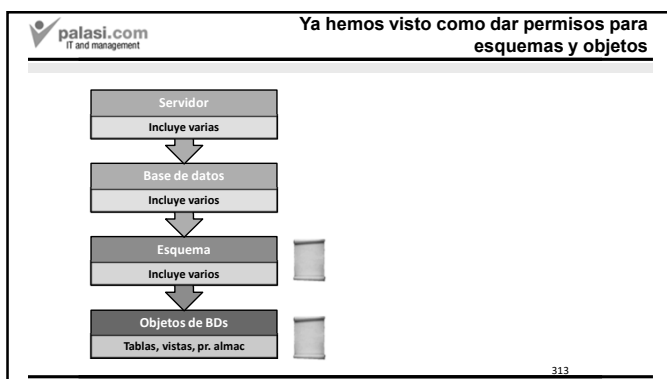
310



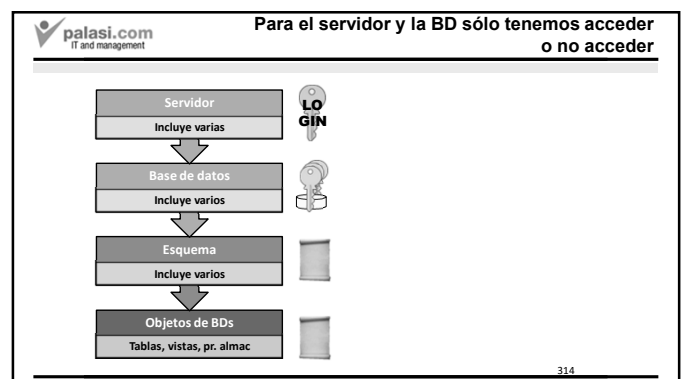
311



312

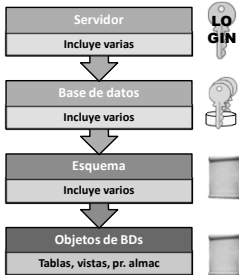


313



314

Esto es muy binario: o todo o nada

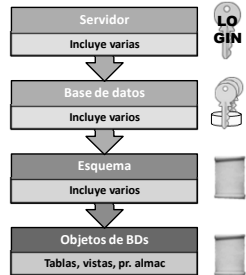


- Si damos acceso, por ejemplo, puede hacer todo con la base de datos.
- Pero ¿podemos decir que puede hacer unas cosas y no otras?
- A veces se necesita.

315

315

Sí, podemos dar permisos a actores al servidor y bases de datos

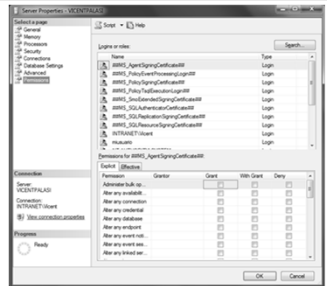


- Simplemente abrimos "Propiedades" y elegimos "Permisos".

316

316

Veamos los permisos de servidor



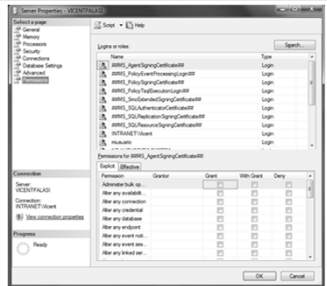
Clic derecho en el servidor y "Propiedades".

Después "Permisos" a la izquierda

317

317

Veamos los permisos de servidor



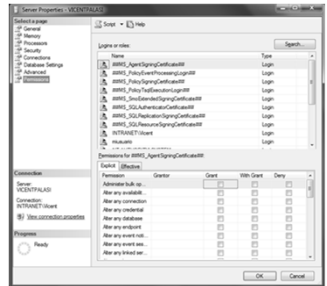
Arriba seleccionamos los logins

Abajo los permisos sobre el servidor de los logins seleccionados.

318

318

Fíjense que los permisos sobre el servidor



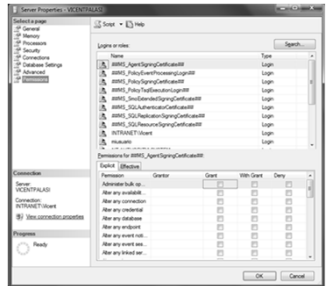
Son diferentes que los de los esquemas u objetos de la BD.

Cosas como "alter any database".

319

319

No veremos estos permisos de servidor



Pero que sepan que existen.

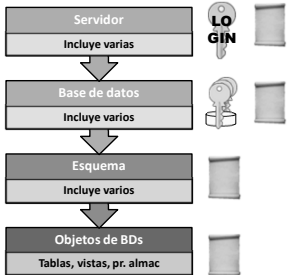
320

320

Vicent Palasí, PhD, MBA, MEd. Todos los derechos reservados.

Mail: palasi@palasi.com Web: www.palasi.com

Muy bien, ya sabemos los permisos de servidor, esquema y objetos de BDs

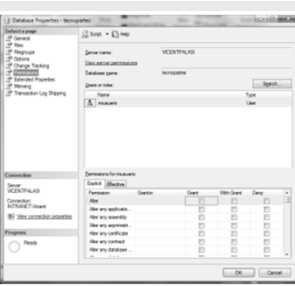


- Ya sólo nos quedan los permisos de bases de datos.

321

321

Se hace clic derecho en la BD y "Propiedades"

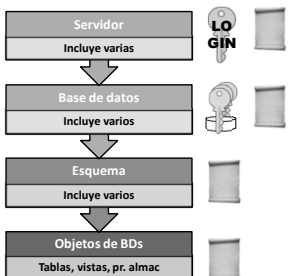


- Después, se hace clic en "Permisos"
- De nuevo, los permisos son diferentes pero no los veremos aquí.

322

322

Ya sabemos cómo poner permisos a todos los niveles



- Los dos inferiores son los más importantes.

323

323

Si quieren ver un resumen de todos los permisos

- Descarguen el archivo "Permisos SQL Server" que es un póster que tiene todos los permisos.

324

324

Permisos para objetos de BD y para esquemas

- **select.** Ver datos en una tabla, vista o campo
- **insert.** Añadir datos a una tabla o una vista.
- **update.** Modificar datos existentes en una tabla, vista o campo.
- **delete.** Eliminar datos de una tabla o de una vista.
- **execute.** Ejecutar un procedimiento almacenado.
- **references.** Hacer referencia a una tabla con una restricción de clave foránea.
- **take ownership.** Pasar a ser propietario del objeto.
- **alter.** Cambiar las propiedades del objeto (menos ser propietario).
- **view definition.** Ver la definición (los metadatos)
- **view change tracking.** Ver como cambió el objeto.

325

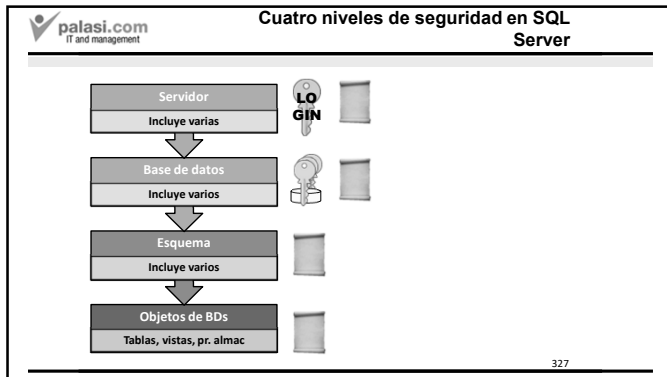
325

Índice del tema 3

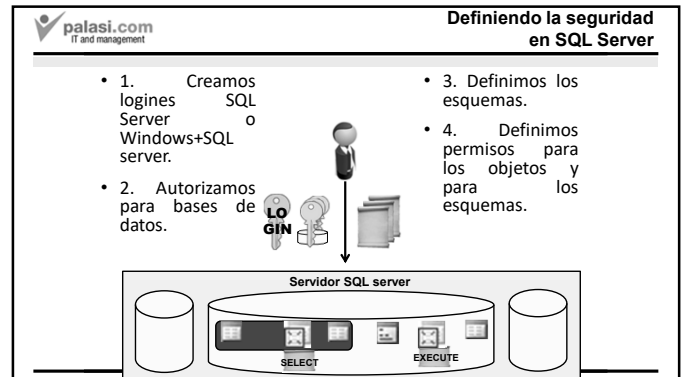
- 1. Introducción a la seguridad en SQL Server.
- 2. Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- 3. Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación
 - Roles de servidor.

326

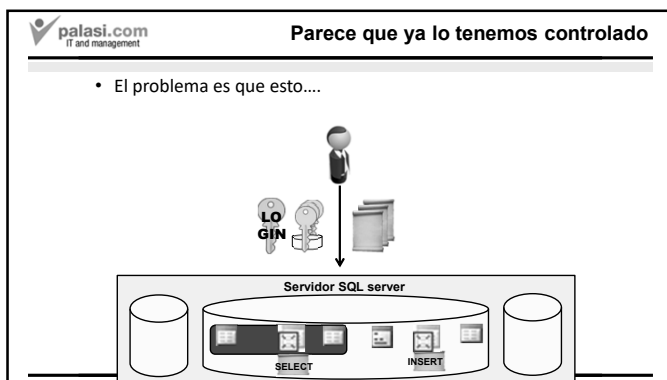
326



327



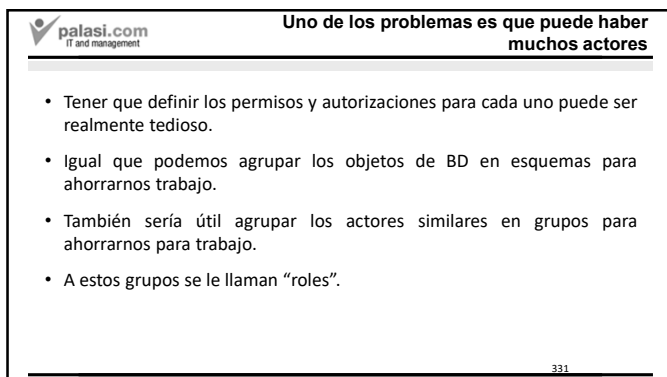
328



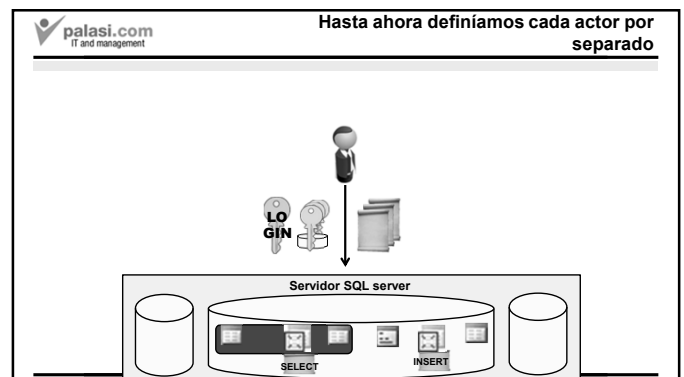
329



330



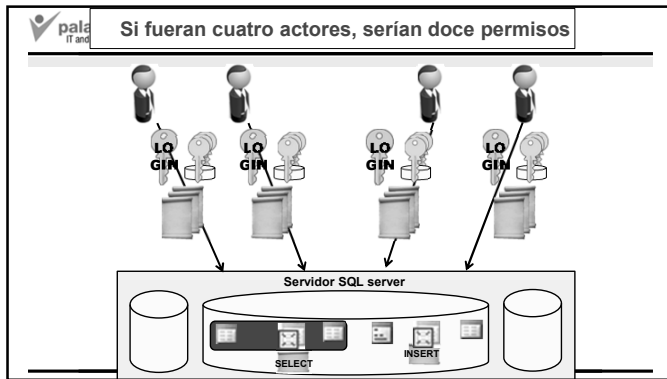
331



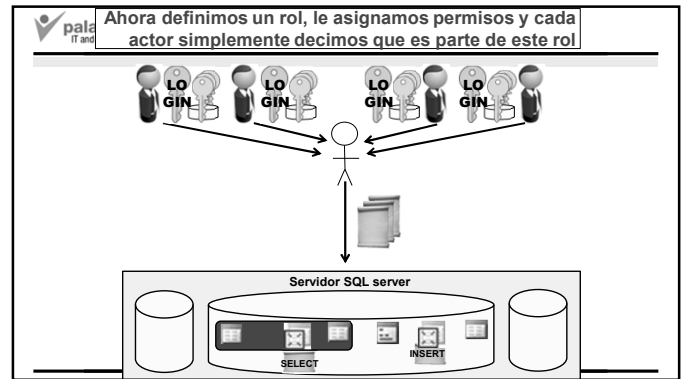
332

Vicent Palasí, PhD, MBA, MEd. Todos los derechos reservados.

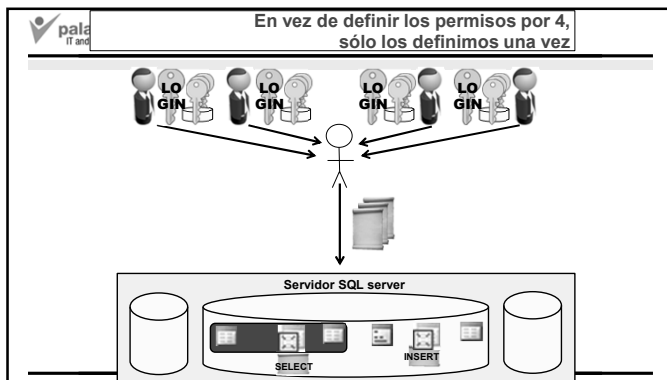
Mail: palasi@palasi.com Web: www.palasi.com



333



334

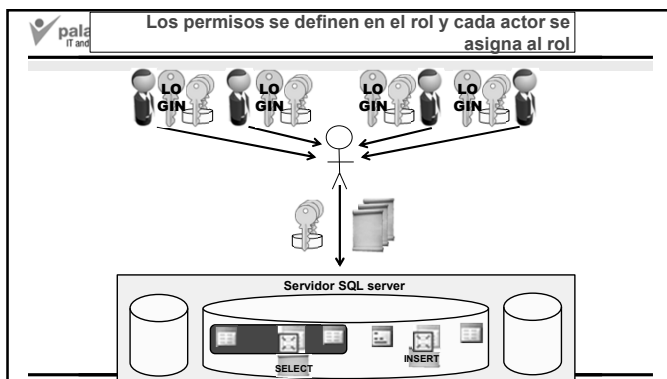


335

Esto es como una empresa

- Si tenemos 400 teleoperadores no vamos a definir las responsabilidades de cada teleoperador por separado.
- Definiremos las responsabilidades del rol "teleoperador".
- Después, diremos:
 - Fulanito es teleoperador.
 - Menganito es teleoperador.

336



337

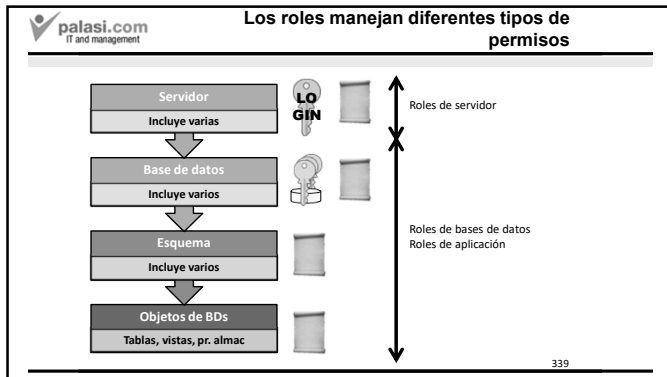
Tres tipos de roles

- Roles de bases de datos.
 - Para agrupar personas y aplicaciones que usan una base de datos.
- Roles de aplicación.
 - Para las aplicaciones que usan una base de datos.
- Roles de servidor.
 - Para las personas y aplicaciones que quieren hacer operaciones globales con el servidor.

338

Vicent Palasí, PhD, MBA, MEd. Todos los derechos reservados.

Mail: palasi@palasi.com Web: www.palasi.com



339

Índice del tema 3

- 1. Introducción a la seguridad en SQL Server.
- 2. Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- 3. Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación
 - Roles de servidor.

340

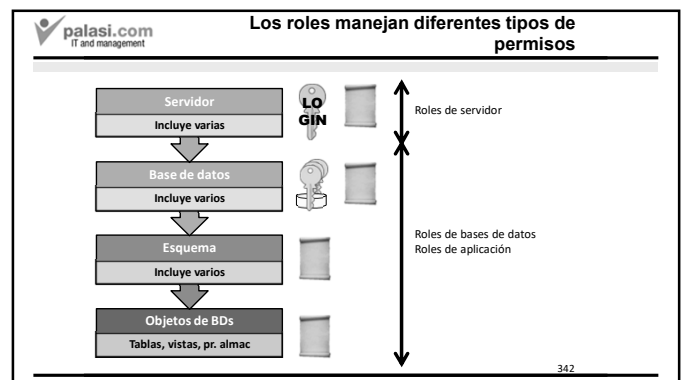
340

Roles de bases de datos

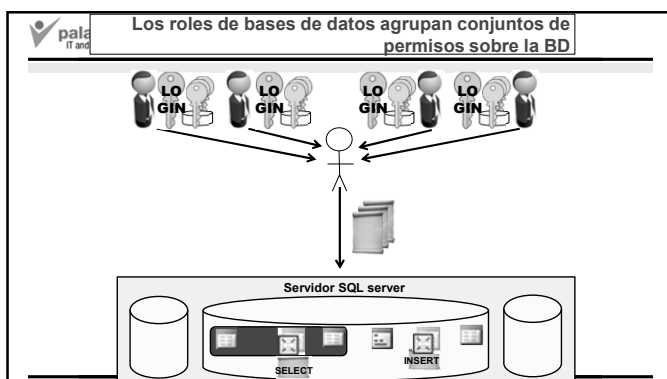
- Son roles que definen permisos de bases de datos y de sus objetos.

341

341



342



343

Nota: Los roles pueden incluir roles

- Puedes hacer un rol que se llame rolEmpleadosVentas
- Puedes hacer un rol que se llame rolEmpleadosProducción.
- Incluir los dos roles en un rolEmpleados

344

344

Vicent Palasí, PhD, MBA, MEd. Todos los derechos reservados.

Mail: palasi@palasi.com Web: www.palasi.com

palasi.com
IT and management

En general

- Los roles están asociados a una base de datos.
- Los roles pueden contener “miembros”.
- Estos miembros pueden ser:
 - Usuarios de la base de datos.
 - Otros roles de la base de datos.

345

345

palasi.com
IT and management

Dos tipos de roles de bases de datos

- 1. Definidos por el usuario.
- 2. Predefinidos.

346

346

palasi.com
IT and management

Dos tipos de roles de bases de datos

- 1. Definidos por el usuario.
- 2. Predefinidos.

347

347

palasi.com
IT and management

Usando un rol

- Se siguen los siguientes pasos.
- 1. Se crea un nuevo rol en la BD.


```
create role nombreRol authorization nombrePropietario
```

Si no se especifica **AUTHORIZATION**, el propietario del rol es el que crea el rol
- 2. Se añaden usuarios a ese rol.


```
alter role nombreRol add member nombreUsuarioORol
```

348

348

palasi.com
IT and management

Si queremos quitar un usuario de un rol

- 3. Se quitan usuarios a este rol.


```
alter role nombreRol drop member nombreUsuarioORol
```
- 4. Para borrar un rol


```
drop role nombreRol
```
- 5. Para saber qué usuarios tiene un rol


```
exec sp_helprolemember 'nombreRol'
```
- 6. Para asignar permisos a un rol


```
grant/revoke/deny listapermisos on schemaUobjeto to nombreRol
```

349

349

palasi.com
IT and management

Ejemplo

- Supongamos que hay una BD “planilla” y 10 usuarios llamados “Usuario1”... “Usuario10” a los que queremos dar derechos a leer y actualizar la tabla “empleados”, pero no a insertarla (son administradores del sistema y no queremos que lean los datos).

350

350

Mala solución

```

use planilla
grant select on Empleados to usuario1
grant update on Empleados to usuario1
...
grant select on Empleados to usuario10
grant update on Empleados to usuario10
  
```

351

Buena solución

```

use planilla
create role UsuarioLimitado
alter role UsuarioLimitado add member usuario1
...
alter role UsuarioLimitado add member usuario10

grant select on Empleados to UsuarioLimitado
grant update on Empleados to UsuarioLimitado
  
```

• ¿Qué pasaría si quisiéramos revocar el permiso de actualizar tabla a estos usuarios? Sólo necesitaríamos:

```

revoke update on Empleados to UsuarioLimitado
  
```

352

Mala solución

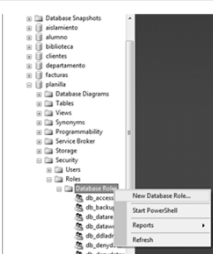
¿Qué pasaría si quisiéramos revocar el permiso de actualizar tabla a estos usuarios? Tendríamos que escribir 10 instrucciones.

```

use planilla
revoke update on Empleados to usuario1
...
revoke update on Empleados to usuario10
  
```

353

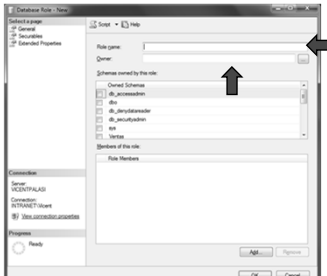
De forma gráfica



- En la versión en español, los roles se llaman “funciones”.
- Hay que expandir la BD, Seguridad, Funciones y Funciones de bases de datos (“Database roles”).
- Se hace clic derecho y se selecciona “Nueva función de base de datos...”

354

En la ventana que aparece, puedes poner nombre y propietario del rol



355


Sale una lista de usuarios miembros del rol



- Y dos botones para añadir y quitar miembros a este rol.

356

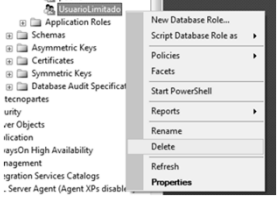
Si hacemos clic en el botón de agregar usuarios



- Podemos hacer clic en el botón de "Browse" y seleccionar los usuarios que aparecen en el rol.

357

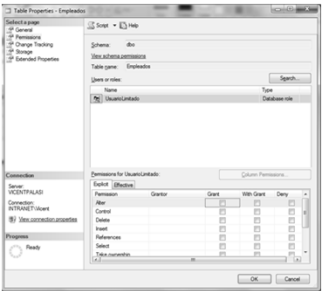
Si queremos eliminar un rol



- Simplemente lo borramos desde el Studio

358

Ahora podemos ir a cualquier objeto de la BD



- Hacer clic derecho | Propiedades | Permisos
- Buscar el rol como buscábamos un usuario.
- Ponerle los permisos que queramos

359

Ejercicio

- Creen un rol en planilla que contenga los logines "juan" y "maria".
- Le dan acceso de lectura y escritura a la tabla "Empleados" y acceso de solo lectura a la tabla "Departamentos".

360

Dos tipos de roles de bases de datos

- 1. Definidos por el usuario.
- 2. Predefinidos.

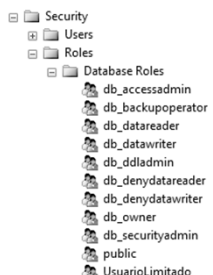
361

Pero aún hay más

- Hay unos roles que ya están predefinidos para todas las bases de datos.
- Esto quiere decir que, para esos roles, ya se le han asignado permisos. Son como "paquetes predefinidos" de permisos listos para ser aplicados.
- Lo único que debemos hacer es añadir el usuario al rol y adquirirá todos los permisos de ese rol.

362

Podemos ver estos roles predefinidos en el Studio



- Están definidos por defecto en "Funciones de bases de datos".

363

363

Roles predefinidos. Se aplican a todas las tablas o vistas de la BD (1)

- **db_owner.** Los usuarios con este rol, puede hacer casi todo sobre la BD, excepto definir qué usuarios son db_owner (esto sólo lo puede hacer el usuario dbo).
- **db_accessadmin.** Los usuarios con este rol pueden decidir qué usuarios pueden acceder a una BD.
- **db_securityadmin.** Los usuarios con este rol pueden asignar permisos a usuarios dentro de una BD, incluyendo la membresía en roles.
- **db_ddladmin.** Los usuarios con este rol pueden ejecutar instrucciones DDL y otras relacionadas.

364

364

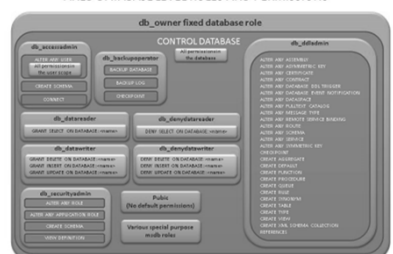
Roles predefinidos. Se aplican a todas las tablas o vistas de la BD (1)

- **db_backupoperator.** Los usuarios con este rol pueden hacer backups.
- **db_datareader.** Los usuarios con este rol sólo pueden leer los datos de la BD.
- **db_datawriter.** Los usuarios con este rol sólo pueden escribir los datos de la BD.
- **db_denydatareader.** Los usuarios con este rol tienen prohibido leer los datos de la BD.
- **db_denydatawriter.** Los usuarios con este rol tienen prohibido escribir los datos de la BD.
- **public.** Este lo veremos más adelante

365

365

Aquí tienen los permisos predefinidos de BD



366

366

Nota

- Esto son los roles predefinidos. Para decir que un usuario sigue un rol predefinido, simplemente se hace

```
use baseDeDatos
alter role nombreRol add member nombreUsuario
```

367

367

Para saber exactamente qué permisos tiene cada rol predefinido de BD

Ejecutamos lo siguiente:

```
exec sp_dbfixedrolepermission 'nombreRol'
```

368

368

La ventaja de los roles predefinidos

- Nos ahorran mucho trabajo.
- Por ejemplo, si tenemos un usuario que sólo puede hacer leer la BD, es mucho más fácil asignarle el rol de **db_datareader** que ir objeto por objeto e instrucción por instrucción asignando permisos.

```
alter role db_datareader
add member Juan
```

369

Los roles definidos por el usuario dan un poco más de trabajo

- Porque hay que definir los permisos para cada uno de ellos.
- De todas maneras, siempre es mejor definir permisos para un rol que para cada uno de los usuarios.

370

Ejemplo

- Supongamos que hay una BD "planilla" y 10 usuarios llamados "Usuario1"... "Usuario10" a los que queremos dar derechos a actualizar todas las tablas, pero no a leerlas (son administradores del sistema y no queremos que lean los datos).

371

Solución

```
alter role db_datawriter add member Usuario1
...
alter role db_datawriter add member Usuario10
```

372

Para los curiosos: ¿Qué ventajas tienen los roles respecto a los grupos de Windows?

- Un rol no tiene que ver con la configuración de Windows. Además puede usar actores que no se autentican con Windows.
- Los grupos son específicos de cada base de datos.
- Un actor puede pertenecer a varios roles.
- Se pueden incluir roles dentro de roles, etc.

373

Ejercicio

- Creen cuatro logines de autenticación de SQL Server, llamados Lector1... Lector2 y Escritor1..Escritor2.
- Autorícenlo para la BD "biblioteca." Los lectores pueden leer todas las tablas excepto "Libros". Los escritores pueden escribir todas las tablas excepto "Libros".
- Definir los permisos para esa base de datos. Pista: usen roles predefinidos y combínenlos con instrucciones de permiso sobre objetos.

374

Solución en archivo adjunto

```
-- Crea los inicios de sesión
create login lector1 with password = 'lector1' must_change, check_expiration = on
create login lector2 with password = 'lector2' must_change, check_expiration = on
create login escritor1 with password = 'escritor1' must_change, check_expiration = on
create login escritor2 with password = 'escritor2' must_change, check_expiration = on

-- Crea los usuarios
use biblioteca
create user lector1 for login lector1
create user lector2 for login lector2
create user escritor1 for login escritor1
create user escritor2 for login escritor2
```

375

375

Solución en archivo adjuntos

```
-- Crea los roles definidos por el usuario
create role lectores
alter role lectores add member lector1
alter role lectores add member lector2
create role escritores
alter role escritores add member escritor1
alter role escritores add member escritor2
```

376

376

Solución en archivo adjuntos

```
-- Agrega los permisos
alter role db_datareader add member lectores
deny select on Libros to lectores

alter role db_datawriter add member escritores
deny insert, update, delete on Libros to escritores
```

377

377

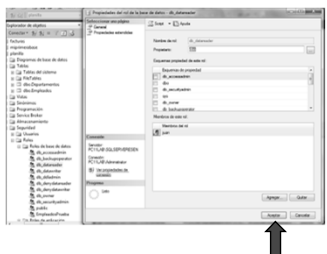
Aquí vemos la estrategia

- Primero usamos los roles predefinidos para dar permisos.
- Los permisos que no incluyen los roles predefinidos los definimos de forma separada: grant/revoke/deny.

378

378

Cómo usar roles predefinidos de forma gráfica



- Hacemos clic derecho en el rol, Propiedades y agregamos miembros del rol.

379

379

Para saber qué roles tiene un usuario en una BD

- Pueden usar el script "listarUsuariosYRoles" que se les proporcionará.

380

380

Un problema con los roles predefinidos de BDs

- Es que los permisos son implícitos.
- No aparecen cuando consultamos los permisos en SQL Server.

381

381

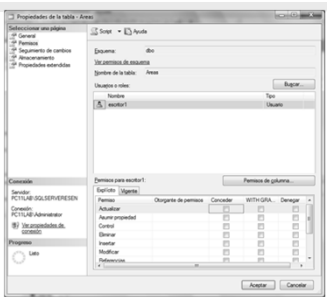
Comprobémoslo

- Después de ejecutar el script que se les proporcionará.
- Vean los permisos de escritor1 sobre "Areas"

382

382

No aparece ninguno.

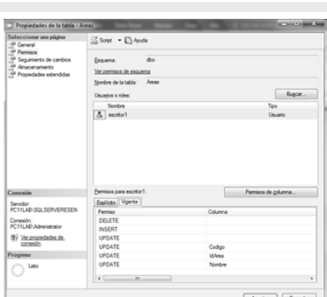


- Sin embargo, los permisos están ahí, pero implícitos.
- Elijan la pestaña "Vigente" o "Efectivo"

383

383

Ahora los vemos

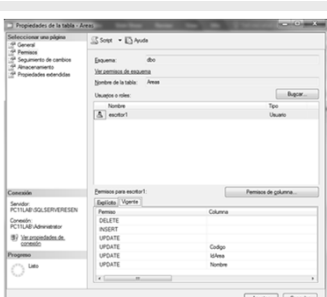


- Aquí ven los permisos que tiene "escritor1" en verdad, no sólo los que hemos definido de forma implícita.

384

384

Atención



- Esto de "Vigente" sólo está disponible para usuarios, no para roles.

385

385

Por otra parte

- Como los permisos definidos en estos roles son implícitos, el revoke no funciona.
- Hay que poner deny si queremos anular alguno de esos permisos.
- Además, recuerden que el revoke no funciona si no es exactamente igual que el grant.
- Por eso hay que hacer el deny.

386

386

El rol public

- Es un rol predefinido en cada base de datos, donde están todos los usuarios que pueden acceder a la base de datos.
- Al revés que los otros roles predefinidos, se le pueden agregar y quitar permisos.
- Añadir un permiso a **public** es añadirlo a todo el mundo (en esa base de datos).

387

Cada BD tiene dos figuras «por defecto» que no pueden borrarse

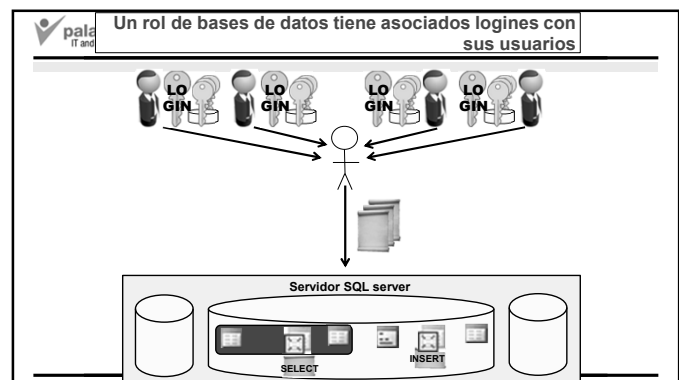
- Un usuario «guest» para los actores que no están definidos en la BD.
 - Por defecto, no puede hacer nada, ni conectarse.
 - Es un agujero de seguridad. Es mejor poner **deny connect to guest** para que nadie lo abra ni por error.
 - Este es un ejemplo de deny
- Un rol «public» que están incluidos todos los usuarios que están definidos en la BD.
 - Si queremos que todos los usuarios tengan un cierto permiso, es muy práctico ponerlo en «public».

388

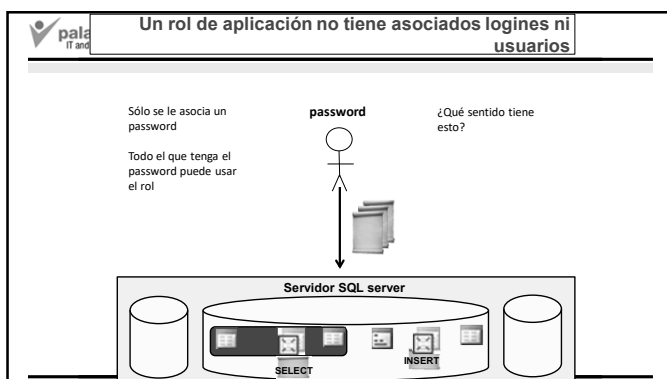
Índice del tema 3

- 1. Introducción a la seguridad en SQL Server.
- 2. Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- 3. Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación
 - Roles de servidor.

389



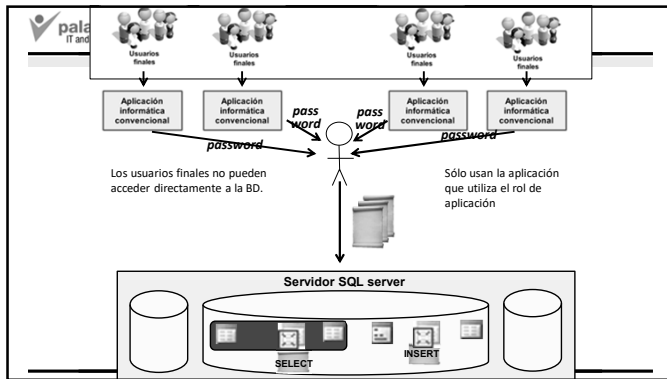
390



391




392



393

Cómo hacerlo de forma gráfica

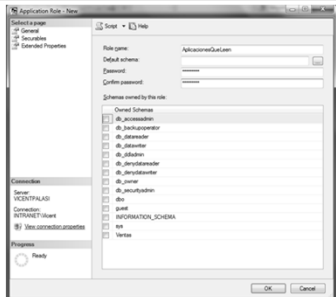


- Dentro de la BD, elegir Security, Funciones, Funciones de aplicación y clic derecho.
- Nueva función de aplicación...

394

Aquí creamos el nuevo rol de aplicación

- Nombre de rol, esquema por defecto, contraseña.



395

En forma de texto

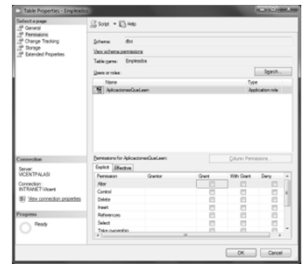
- Creas el rol de aplicación.

```
exec sp_setapprole 'nombreRol', 'password'
```

396

En cualquier objeto de BD.

- Podemos elegir el rol de aplicación y darle permisos.
- En forma de texto usamos grant, revoke y deny.



397

Índice del tema 3

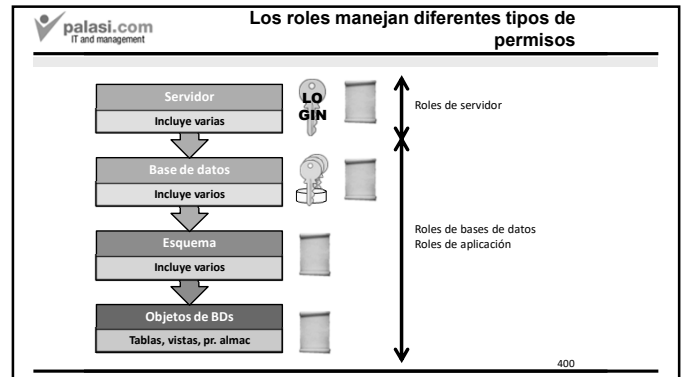
- Introducción a la seguridad en SQL Server.
- Seguridad definida de forma individual.
 - Creación de inicios de sesión.
 - Autenticación de SQL Server.
 - Autenticación de Windows.
 - Autorización a las bases de datos.
 - Autorización a los objetos de BDs.
- Seguridad definida usando grupos.
 - Esquemas.
 - Roles
 - Roles de bases de datos.
 - Roles de aplicación
 - Roles de servidor.

398

Hasta ahora hemos hablado de roles de bases de datos y aplicación

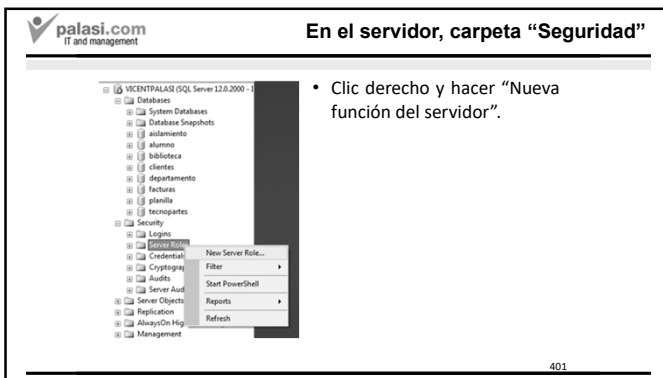
- Lo que hacen es agrupar actores que tienen los mismos permisos para hacer operaciones sobre una base de datos.
- Pero, a veces nos interesa definir roles generales sobre todas las bases de datos e incluso sobre operaciones que se hacen en el servidor y no están asignadas a ninguna base de datos.
- Esto nos lleva a definir los llamados “roles de servidor”.

399



400

En el servidor, carpeta “Seguridad”

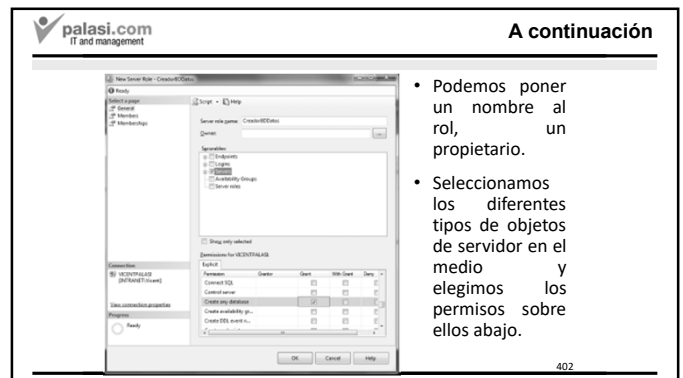


- Clic derecho y hacer “Nueva función del servidor”.

401

401

A continuación

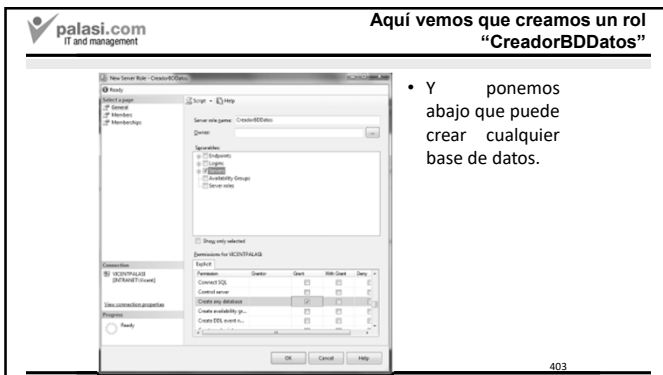


- Podemos poner un nombre al rol, un propietario.
- Seleccionamos los diferentes tipos de objetos de servidor en el medio y elegimos los permisos sobre ellos abajo.

402

402

Aquí vemos que creamos un rol “CreadorBDDatos”

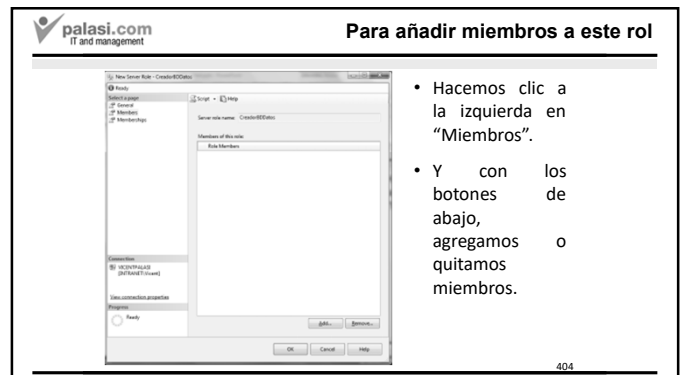


- Y ponemos abajo que puede crear cualquier base de datos.

403

403


Para añadir miembros a este rol



- Hacemos clic a la izquierda en “Miembros”.
- Y con los botones de abajo, agregamos o quitamos miembros.

404

404

 **palasi.com**
IT and management

En forma de texto

- Para crear un rol de servidor


```
create server role nombreRol authorization login
```
- Para añadir un miembro al rol

```
alter server role nombreRol add member login
```
- Para quitar un miembro al rol

```
alter server role nombreRol drop member login
```
- Para borrar un rol

```
drop server role nombreRol
```

405



palasi.com
IT and management

Operaciones para roles de servidor


- Para añadir derechos a un rol

```
grant/deny/revoke permiso to nombreRol
```
- Para saber qué logines tiene un rol de servidor

```
exec sp_helpsrvrolemember 'nombreRol'
```
- Para saber qué permisos exactamente tiene un rol de servidor

```
exec sp_srvrolepermission 'nombreRol'
```

406




palasi.com
IT and management

Pero aún hay más

- Hay roles predefinidos de servidor.
- Estos ya están definidos y listos para utilizar.

407




IT and management

Roles predefinidos de servidor (1)

- **sysadmin.** Los logines con este rol pueden hacer todo en SQL Server. No tienen restricción en ninguna BD ni fuera de ella.
- **serveradmin.** Los logines con este rol son los administradores de servidor que pueden cambiar las propiedades del servidor, iniciarlo y apagarlo.
- **setupadmin.** Los logines con este rol son administradores que están configurando servidores remotos.
- **securityadmin.** Los logines con este rol pueden hacer cualquier operación relacionada con la seguridad en SQL Server.

408



IT and management

Roles predefinidos de servidor (2)

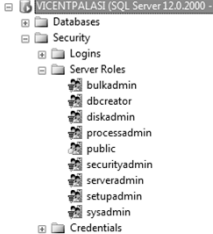
- **processadmin.** Puede manejar los procesos que se ejecutan en el servidor de SQL Server.
- **dbcreator.** Los logines con este rol pueden hacer create, alter, drop y restore sobre cualquier BD.
- **diskadmin.** Puede gestionar los archivos que usa SQL Server
- **bulk admin.** Puede usar la sentencia bulk insert (para importar un archivo de datos en una BD).
- **public.** Asignado a todos los logines del servidor. Si queremos poner un derecho a todos, los ponemos aquí.

409

[illegible]

410

Se pueden ver en el Studio



- Aquí aparecen predefinidos bajo "Funciones del servidor".

411

El rol sysadmin

- Permite hacer cualquier cosa con el servidor de SQL Server.
- Por defecto, tienen este rol:
 - El login "sa". Esta no puede eliminarse del rol.
- El rol de servidor **sysadmin** recibe automáticamente el rol de base de datos **db_owner** en cada BD.
 - Recordemos que puede hacer todo, menos definir que otros usuarios son **db_owner**

412

Manejar la seguridad en SQL Server

- Nivel servidor.
 - 1. Definimos los logines bien con autenticación de Windows o con autenticación de SQL Server.
 - 2. Definimos roles de servidor, sus permisos y asignamos miembros a roles de servidor (predefinidos o no).
 - 3. Para los permisos de servidor que no se incluyen en los roles los definimos sobre los logines.
- Nivel base de datos (para cada BD).
 - 1. Definimos usuarios para las bases de datos.
 - 2. Definimos esquemas de BD.
 - 2. Definimos roles de bases de datos, sus permisos y asignamos miembros a roles de bases de datos (predefinidos o no).
 - 3. Para los permisos que no se incluyen en los roles los definimos sobre los usuarios.

413

Algunas recomendaciones

- Se asignan los permisos que necesitan todos los usuarios al rol public.
- Se asignan los permisos que necesitan todos los miembros de un grupo de personas a un rol (o a un grupo de Windows).
- Sólo se asignan permisos individuales a los usuarios si los permisos que necesitan no se pueden asignar a un rol.

414

Obviamente podemos llegar a una conclusión

- Manejar la seguridad en SQL Server es algo que...

415

...muy cuesta



416

palasi.com
IT and management

Pero la seguridad de los datos es de lo más importante



417

417

palasi.com
IT and management

Hacer ejercicio de empresa. Estrategia

1. Hacer que el DBA jefe asuma el control total del servidor.
2. Definir los aspectos para los DBA secundarios.
3. Definir los aspectos del departamento de ventas.
4. Definir los aspectos para el departamento de contabilidad.
5. Definir los aspectos para el gerente general.

418

418

palasi.com
IT and management

Estrategia para cada parte del ejercicio

- A. Definir los protegibles:
 - BDs, objetos y esquemas (si existen).
- B. Definir los actores:
 - B1. Definir los logines
 - B2. Definir los usuarios (es decir, autorizar a los logines a las BDs).
- C. Definir los permisos:
 - C1. Primero, a grandes rasgos con los roles.
 - C2. Segundo, con detalle con permisos directos sin roles.

419

419

palasi.com
IT and management

1. Hacer que el DBA asuma el control del servidor

- El informático que instaló SQL Server conoce la contraseña del login “sa” y la da al DBA jefe
- El DBA jefe entra como login “sa” (que será su identidad a partir de ahora), cambia la contraseña del login “sa”.
- Si hay otros usuarios con rol sysadmin, les quitará ese rol. Sólo queremos un responsable máximo.

420

420

palasi.com
IT and management

Hacer ejercicio de empresa. Estrategia

1. Hacer que el DBA jefe asuma el control total del servidor.
2. Definir los aspectos para los DBA secundarios.
3. Definir los aspectos del departamento de ventas.
4. Definir los aspectos para el departamento de contabilidad.
5. Definir los aspectos para el gerente general.

421

421

palasi.com
IT and management

Estrategia para cada parte del ejercicio

- A. Definir los protegibles:
 - BDs, objetos y esquemas (si existen).
- B. Definir los actores:
 - B1. Definir los logines
 - B2. Definir los usuarios (es decir, autorizar a los logines a las BDs).
- C. Definir los permisos:
 - C1. Primero, a grandes rasgos con los roles.
 - C2. Segundo, con detalle con permisos directos sin roles.

422

422

2. Definir los aspectos para los DBA secundarios.

- A. Crear BD “contabilidad” y “ventas”.
- B1. Creará dos logines “dbaconta” y “dbaventas” que cambiarán la contraseña en el primer acceso (usuarios de SQL Server por si cambian personas).
- B2. Hará que el usuario “dbo” de “contabilidad” sea “dbaconta” y el usuario “dbo” de ventas sea “dbaventas”.
- En este caso, no existen C1 y C2.
- Dará los logines “dbaconta” y “dbaventas” con sus contraseñas a los DBA de BD de contabilidad y ventas.

423

423

Hacer ejercicio de empresa. Estrategia

1. Hacer que el DBA jefe asuma el control total del servidor.
2. Definir los aspectos para los DBA secundarios.
3. Definir los aspectos del departamento de ventas.
4. Definir los aspectos para el departamento de contabilidad.
5. Definir los aspectos para el gerente general.

424

424

Estrategia para cada parte del ejercicio

- A. Definir los protegibles:
 - BDs, objetos y esquemas (si existen).
- B. Definir los actores:
 - B1. Definir los logines
 - B2. Definir los usuarios (es decir, autorizar a los logines a las BDs).
- C. Definir los permisos:
 - C1. Primero, a grandes rasgos con los roles.
 - C2. Segundo, con detalle con permisos directos sin roles.

425

425

3. Definir los aspectos del departamento de ventas (a).

- A. El DBA con login “dbaventas” creará las tablas de “equipos” y “repuestos”.
- B1. Se crean o autorizan logines para todos los miembros del departamento de ventas.
 - Los crea directamente el DBA jefe.
 - El DBA jefe da un permiso “alter login” a “dbaventas” y éste se encarga de crear los logines.
- B2. El DBA con login “dbaventas” autoriza a todos esos logines para la base de datos “ventas” creando usuarios para ventas.

426

426

Hacer ejercicio de empresa. Estrategia

1. Hacer que el DBA jefe asuma el control total del servidor.
2. Definir los aspectos para los DBA secundarios.
3. Definir los aspectos del departamento de ventas.
4. Definir los aspectos para el departamento de contabilidad.
5. Definir los aspectos para el gerente general.

427

427

Estrategia para cada parte del ejercicio

- A. Definir los protegibles:
 - BDs, objetos y esquemas (si existen).
- B. Definir los actores:
 - B1. Definir los logines
 - B2. Definir los usuarios (es decir, autorizar a los logines a las BDs).
- C. Definir los permisos:
 - C1. Primero, a grandes rasgos con los roles.
 - C2. Segundo, con detalle con permisos directos sin roles.

428

428

3. Definir los aspectos del departamento de ventas (b).

- C1. Roles:
 - Crea un rol definido por el usuario “empleadosventas”.
 - Asigna todos los usuarios de la base de datos a este rol.
 - Asigna el rol “empleadosventas” al rol db_datareader.

(Nota: el rol definido por el usuario se hace por flexibilidad)

429

429

Estrategia para cada parte del ejercicio

- A. Definir los protegibles:
 - BDs, objetos y esquemas (si existen).
- B. Definir los actores:
 - B1. Definir los logines
 - B2. Definir los usuarios (es decir, autorizar a los logines a las BDs).
- C. Definir los permisos:
 - C1. Primero, a grandes rasgos con los roles.
 - C2. Segundo, con detalle con permisos directos sin roles.

430

430

3. Definir los aspectos del departamento de ventas (c).

- C2. Permisos sin roles:
 - Al usuario que entra datos en la tabla de equipos le concede permisos insert, update y delete para la tabla de equipos.
 - Al usuario que entra datos en la tabla de repuestos le concede permisos insert, update y delete para la tabla de repuestos.
 - Al usuario que representa al jefe de departamento de ventas le asigna un permiso “take ownership”.

431

431

Hacer ejercicio de empresa. Estrategia

1. Hacer que el DBA jefe asuma el control total del servidor.
2. Definir los aspectos para los DBA secundarios.
3. Definir los aspectos del departamento de ventas.
4. Definir los aspectos para el departamento de contabilidad.
5. Definir los aspectos para el gerente general.

432

432

Departamento de contabilidad

- Es diferente porque no accedemos directamente sino a través de programas.
- Con programas hay dos tipos de seguridad:
 - A. La seguridad que implementa el programa con código de programación.
 - B. La seguridad que se define en el servidor de BD y de la que trata de esta asignatura.
 - La primera se hace por claridad y seguridad.
 - La segunda se hace por seguridad, por si el programa falla.

433

433

¿Qué tipo de autenticación?

- En la autenticación de SQL Server, el programa debería tener registrados los usuarios y contraseñas del servidor de SQL Server. Esto puede causar problemas si alguien los modifica en el servidor sin actualizarlo en el programa.
- Por ello, en este caso, facilita mucho las cosas que sea autenticación de Windows para no tener dos usuarios para el programa y el servidor de BD.

434

434

palasi.com
IT and management

Estrategia para cada parte del ejercicio

- A. Definir los protegibles:
 - BDs, objetos y esquemas (si existen).
- B. Definir los actores:
 - B1. Definir los logines
 - B2. Definir los usuarios (es decir, autorizar a los logines a las BDs).
- C. Definir los permisos:
 - C1. Primero, a grandes rasgos con los roles.
 - C2. Segundo, con detalle con permisos directos sin roles.

435

435

palasi.com
IT and management

4. Definir los aspectos para el departamento de contabilidad (a)

- A. El DBA de contabilidad con login “dbaconta” crea:
 - La tabla de partidas
 - Crea dos vistas “partidasguate” y “partidaselsalvador”.
 - Crea los triggers “instead of” para que estas vistas se actualicen de forma correcta.
- B1. Crear los logines:
 - El administrador de red creará los usuarios de Windows para el departamento de contabilidad.
 - Después, se autorizan para SQL Server. Hay dos opciones:
 - Los autoriza directamente el DBA jefe.
 - El DBA jefe da un permiso “alter login” a “dbaconta” y éste se encarga de autorizar los usuarios de Windows.

436

436

palasi.com
IT and management

Estrategia para cada parte del ejercicio

- A. Definir los protegibles:
 - BDs, objetos y esquemas (si existen).
- B. Definir los actores:
 - B1. Definir los logines
 - B2. Definir los usuarios (es decir, autorizar a los logines a las BDs).
- C. Definir los permisos:
 - C1. Primero, a grandes rasgos con los roles.
 - C2. Segundo, con detalle con permisos directos sin roles.

437

437

palasi.com
IT and management

4. Definir los aspectos para el departamento de contabilidad (b)

- B2. El DBA que tiene el login “dbaconta” autoriza como usuarios a todos estos logines del departamento de contabilidad a la base de datos “contabilidad”.

438

438

palasi.com
IT and management

Estrategia para cada parte del ejercicio

- A. Definir los protegibles:
 - BDs, objetos y esquemas (si existen).
- B. Definir los actores:
 - B1. Definir los logines
 - B2. Definir los usuarios (es decir, autorizar a los logines a las BDs).
- C. Definir los permisos:
 - C1. Primero, a grandes rasgos con los roles.
 - C2. Segundo, con detalle con permisos directos sin roles.

439

439

palasi.com
IT and management

4. Definir los aspectos para el departamento de contabilidad (c)

- C1. El DBA que tiene el login “dbaconta”
 - Crea dos roles de bases de datos:
 - “empleadoscontaguete”, con permisos select para vista “partidasguate”.
 - “empleadoscontaelsalvador”, con permisos select para vista “partidaselsalvador”.
 - Asigna el primer rol a los usuarios de Guatemala.
 - Asigna el segundo rol a los usuarios de El Salvador.
 - Al usuario que representa al jefe se le incluye en los roles “empleadoscontaguete”, “empleadoscontaelsalvador”

(No se pueden usar roles predefinidos pues sólo queremos dar acceso a una vista y no a toda la base de datos.)

440

440

palasi.com
IT and management

Estrategia para cada parte del ejercicio

- A. Definir los protegibles:
 - BDs, objetos y esquemas (si existen).
- B. Definir los actores:
 - B1. Definir los logines
 - B2. Definir los usuarios (es decir, autorizar a los logines a las BDs).
- C. Definir los permisos:
 - C1. Primero, a grandes rasgos con los roles.
 - C2. Segundo, con detalle con permisos directos sin roles.

441

441

palasi.com
IT and management

4. Definir los aspectos para el departamento de contabilidad (d)

- C2. Permisos sin roles:
 - Al usuario de BD correspondiente al empleado que entra información en El Salvador también le asignan permisos insert, update y delete para la vista partidaselsalvador.
 - Al usuario de BD correspondiente al empleado que entra información en Guatemala también le asigna permisos insert, update y delete para la vista partidasguate.
 - Al usuario de BD correspondiente al jefe se le da derechos de take ownership.

442

442

palasi.com
IT and management

Hacer ejercicio de empresa. Estrategia

1. Hacer que el DBA jefe asuma el control total del servidor.
2. Definir los aspectos para los DBA secundarios.
3. Definir los aspectos del departamento de ventas.
4. Definir los aspectos para el departamento de contabilidad.
5. Definir los aspectos para el gerente general.

443

443

palasi.com
IT and management

5. Definir los aspectos para el gerente general. El DBA jefe

- A. En este caso, no se crea ningún protegible.
- B1. Creará el login del gerente general. Por ejemplo, puede autorizará RED\gerente para que se conecte a SQL Server.
- B2. Después a este login le hará usuario de las dos bases de datos "contabilidad" y "ventas".
- C1. Usuarios:
 - Al usuario de la BD de "ventas" correspondiente al gerente general le asignará el rol "empleadosventas"
 - Al usuario de la BD de "contabilidad" correspondiente al gerente general le asignará los roles "empleadoscontaguate" y "empleadoselsalvador".
- C2. En este caso, no hay C2.

444

444

palasi.com
IT and management

1. Hacer que el DBA asuma el control del servidor

- El informático que instaló SQL Server conoce la contraseña del login "sa" y la da al DBA jefe
- El DBA jefe entra como login "sa" (que será su identidad a partir de ahora), cambia la contraseña del login "sa".
- Si hay otros usuarios con rol sysadmin, les quitará ese rol. Sólo queremos un responsable máximo.

445

445

palasi.com
IT and management

1. Hacer que el DBA asuma el control del servidor

- **En persona**
- El informático que instaló SQL Server conoce la contraseña del login "sa" y la da al DBA jefe. **En persona.**

446

446

1. Hacer que el DBA asuma el control del servidor. Login "sa"

- El DBA jefe entra como login "sa" (que será su identidad a partir de ahora), cambia la contraseña del login "sa".

```
alter login sa with password = 'password'
```
- Si hay otros usuarios con rol sysadmin, les quitará ese rol. Sólo queremos un responsable máximo.

Supuesto: *login1...loginN* tienen rol sysadmin

```
alter server role sysadmin drop member login1
```

...

```
alter server role sysadmin drop member loginN
```

447

447

2. Definir los aspectos para los DBA secundarios.

- A. Crear BD "contabilidad" y "ventas".
- B1. Creará dos logines "dbaconta" y "dbaventas" que cambiarán la contraseña en el primer acceso (usuarios de SQL Server por si cambian personas).
- B2. Hará que el usuario "dbo" de "contabilidad" sea "dbaconta" y el usuario "dbo" de ventas sea "dbaventas".
- En este caso, no existen C1 y C2.
- Dará los logines "dbaconta" y "dbaventas" con sus contraseñas a los DBA de BD de contabilidad y ventas.

448

448

2. Definir los aspectos para los DBA secundarios. Login "sa"

- A. Crear BD "contabilidad" y "ventas".

```
create database contabilidad;
```

```
create database ventas;
```
- B1. Creará dos logines "dbaconta" y "dbaventas" que cambiarán la contraseña en el primer acceso (logines de SQL Server por si cambian personas).

```
create login dbaconta with password = 'contraseña1' must_change, check_expiration = on
```

```
create login dbaventas with password = 'contraseña2' must_change, check_expiration = on
```

449

449

2. Definir los aspectos para los DBA secundarios. Login "sa"

- B2. Hará que el usuario "dbo" de "contabilidad" sea "dbaconta" y el usuario "dbo" de ventas sea "dbaventas".

```
use contabilidad
```

```
exec sp_changedbowner 'dbaconta'
```

```
use ventas
```

```
exec sp_changedbowner 'dbaventas'
```

450

450

2. Definir los aspectos para los DBA secundarios. En persona

- Dará los logines "dbaconta" y "dbaventas" con sus contraseñas a los DBA de BD de contabilidad y ventas.

451

451

3. Definir los aspectos del departamento de ventas (a).

- A. El DBA con login "dbaventas" creará las tablas de "equipos" y "repuestos".

452

452

3. Definir los aspectos del departamento de ventas.
Login "dbventas"

- A. El DBA con login "dbventas" creará las tablas de "equipos" y "repuestos".

```
create table equipos (
  nombrecampo1 tipocampo1,
  ...
  nombrecampon tipocampon)
create table repuestos (
  nombrecampo1 tipocampo1,
  ...
  nombrecampon tipocampon)
```

453

453

3. Definir los aspectos del departamento de ventas (a).

- B1. Se crean o autorizan logines para todos los miembros del departamento de ventas.
 - Los crea directamente el DBA jefe.
 - El DBA jefe da un permiso "alter login" a "dbventas" y éste se encarga de crear los logines.

454

454

3. Definir los aspectos del departamento de ventas.
Login "sa"

- B1. Se crean o autorizan logines para todos los miembros del departamento de ventas.

```
create login [dominioventas1] from windows
...
create login [dominioventasn] from windows
```

455

455

3. Definir los aspectos del departamento de ventas (a).

- B2. El DBA con login "dbventas" autoriza a todos esos logines para la base de datos "ventas" creando usuarios para ventas.

456

456

3. Definir los aspectos del departamento de ventas.
Login "dbventas"

- B2. El DBA con login "dbventas" autoriza a todos esos logines para la base de datos "ventas" creando usuarios para ventas.

```
use ventas
create user ventas1 for login [dominioventas1]
...
create user ventasn for login [dominioventasn]
```

457

457

3. Definir los aspectos del departamento de ventas (b).

- C1. Roles:
 - Crea un rol definido por el usuario "empleadosventas".
 - Asigna todos los usuarios de la base de datos a este rol.
 - Asigna el rol "empleadosventas" al rol db_datareader.

(Nota: el rol definido por el usuario se hace por flexibilidad)

458

458

3. Definir los aspectos del departamento de ventas (b). Login "dbaventas"

- C1. Roles:
 - Crea un rol definido por el usuario "empleadosventas".
 - Asigna todos los usuarios de la base de datos a este rol.
 - Asigna el rol "empleadosventas" al rol db_datareader.

```
create role empleadosventas authorization dbo
alter role empleadosventas add member ventas1
...
alter role empleadosventas add member ventasn
alter role db_datareader add member empleadosventas
```

459

459

3. Definir los aspectos del departamento de ventas (c).

- C2. Permisos sin roles:
 - Al usuario que entra datos en la tabla de equipos le concede permisos insert, update y delete para la tabla de equipos.
 - Al usuario que entra datos en la tabla de repuestos le concede permisos insert, update y delete para la tabla de repuestos.
 - Al usuario que representa al jefe de departamento de ventas le asigna un permiso "take ownership".

460

460

3. Definir los aspectos del departamento de ventas (c). Login "dbaventas"

- C2. Permisos sin roles:
 - Al usuario que entra datos en la tabla de equipos le concede permisos insert, update y delete para la tabla de equipos.

```
grant insert, update, delete on equipos to responsableequipos
```

- Al usuario que entra datos en la tabla de repuestos le concede permisos insert, update y delete para la tabla de repuestos.

```
grant insert, update, delete on repuestos to responsablerepuestos
```

- Al usuario que representa al jefe de departamento de ventas le asigna un permiso "take ownership".

```
grant take ownership on ventas to jefedepartamentoventas
```

461

461

4. Definir los aspectos para el departamento de contabilidad (a)

- A. El DBA de contabilidad con login "dbaconta" crea:
 - La tabla de partidas
 - Crea dos vistas "partidasguate" y "partidaselsalvador".
 - Crea los triggers "instead of" para que estas vistas se actualicen de forma correcta.
- B1. Crear los logines:
 - El administrador de red creará los usuarios de Windows para el departamento de contabilidad.
 - Después, se autorizan para SQL Server. Hay dos opciones:
 - Los autoriza directamente el DBA jefe.
 - El DBA jefe da un permiso "alter login" a "dbaconta" y éste se encarga de autorizar los usuarios de Windows.

462

462

4. Definir los aspectos para el departamento de contabilidad (a). Login "dbaconta"

- A. El DBA de contabilidad con login "dbaconta" crea:
 - La tabla de partidas

```
create table partidas (
campo1 tipocampo1,
...
campon tipocampon
pais varchar(50) not null)
```

463

463

4. Definir los aspectos para el departamento de contabilidad (a). Login "dbaconta"

- Crea dos vistas "partidasguate" y "partidaselsalvador".

```
create view partidasguate as
select campo1,..., campon
from partidas
where pais = 'Guatemala'
```

```
create view partidaselsalvador as
select campo1,..., campon
from partidas
where pais = 'El Salvador'
```

464

464

4. Definir los aspectos para el departamento de contabilidad (a). Login "dbaconta"

- Crea los triggers "instead of" para que estas vistas se actualicen de forma correcta.

```
create trigger triggerpartidasguate
on partidasguate
instead of insert, update, delete as
begin
delete partidas from deleted inner join partidas
on deleted.campo1 = partidas.campo1

insert into partidas
(campo2,..., campon, pais)
select campo2,..., campon, 'Guatemala'
from inserted
end
```

465

465

4. Definir los aspectos para el departamento de contabilidad (a). Login "dbaconta"

- Crea los triggers "instead of" para que estas vistas se actualicen de forma correcta.

```
create trigger triggerpartidaselsalvador
on partidaselsalvador
instead of insert, update, delete as
begin
delete partidas from deleted inner join partidas
on deleted.campo1 = partidas.campo1

insert into partidas
(campo2,..., campon, pais)
select campo2,..., campon, 'El Salvador'
from inserted
end
```

466

466

4. Definir los aspectos para el departamento de contabilidad (a) Login "sa"

- B1. Crear los logines:

```
create login [dominio\cntguate1] from windows
...
create login [dominio\cntguaten] from windows

create login [dominio\cntsalva1] from windows
...
create login [dominio\cntsalvan] from windows
```

467

467

4. Definir los aspectos para el departamento de contabilidad (b)

- B2. El DBA que tiene el login "dbaconta" autoriza como usuarios a todos estos logines del departamento de contabilidad a la base de datos "contabilidad".

468

468

4. Definir los aspectos para el departamento de contabilidad (b). Login "dbaconta"

- B2. El DBA que tiene el login "dbaconta" autoriza como usuarios a todos estos logines del departamento de contabilidad a la base de datos "contabilidad".

```
use contabilidad
create user cntguate1 for login [dominio\cntguate1]
...
create user cntsalvan for login [dominio\cntsalvan]
```

469

469

4. Definir los aspectos para el departamento de contabilidad (c)

- C1. El DBA que tiene el login "dbaconta"
 - Crea dos roles de bases de datos:
 - "empleadoscontaguete", con permisos select para vista "partidasguate".
 - "empleadoscontaelsalvador", con permisos select para vista "partidaselsalvador".
 - Asigna el primer rol a los usuarios de Guatemala.
 - Asigna el segundo rol a los usuarios de El Salvador.
 - Al usuario que representa al jefe se le incluye en los roles "empleadoscontaguete", "empleadoscontaelsalvador"

(No se pueden usar roles predefinidos pues sólo queremos dar acceso a una vista y no a toda la base de datos.)

470

470

4. Definir los aspectos para el departamento de contabilidad (c). Login "dbaconta"

- C1. El DBA que tiene el login "dbaconta"
 - Crea dos roles de bases de datos:
 - "empleadoscontaguate", con permisos select para vista "partidasguate".
 - "empleadoscontaelsalvador", con permisos select para vista "partidaselsalvador".

```
create role empleadoscontaguate authorization dbo
grant select on partidasguate to empleadoscontaguate
create role empleadoscontaelsalvador authorization dbo
grant select on partidaselsalvador to empleadoscontaelsalvador
```

471

471

4. Definir los aspectos para el departamento de contabilidad (c). Login "dbaconta"

- Asigna el primer rol a los usuarios de Guatemala.


```
alter role empleadoscontaguate add member cntguate1
...
alter role empleadoscontaguate add member cntguateen
```
- Asigna el segundo rol a los usuarios de El Salvador.


```
alter role empleadoscontaelsalvador add member cntsalva1
...
alter role empleadoscontaelsalvador add member cntsalvan
```
- Al usuario que representa al jefe se le incluye en los roles "empleadoscontaguate", "empleadoscontaelsalvador"


```
alter role empleadoscontaelsalvador add member jefeDepartamentoConta
alter role empleadoscontaguate add member jefeDepartamentoConta
```

472

472

4. Definir los aspectos para el departamento de contabilidad (d)

- C2. Permisos sin roles:
 - Al usuario de BD correspondiente al empleado que entra información en El Salvador también le asignan permisos insert, update y delete para la vista partidaselsalvador.
 - Al usuario de BD correspondiente al empleado que entra información en Guatemala también le asigna permisos insert, update y delete para la vista partidasguate.
 - Al usuario de BD correspondiente al jefe se le da derechos de take ownership.

473

473

4. Definir los aspectos para el departamento de contabilidad (d). Login "dbaconta"

- C2. Permisos sin roles:
 - Al usuario de BD correspondiente al empleado que entra información en El Salvador también le asignan permisos insert, update y delete para la vista partidaselsalvador.


```
grant insert, update, delete on partidaselsalvador to responsablecontaelsalvador
```
 - Al usuario de BD correspondiente al empleado que entra información en Guatemala también le asigna permisos insert, update y delete para la vista partidasguate.


```
grant insert, update, delete on partidasguate to responsablecontaguate
```
 - Al usuario de BD correspondiente al jefe se le da derechos de take ownership.


```
grant take ownership on contabilidad to jefeDepartamentoConta
```

474

474

5. Definir los aspectos para el gerente general. El DBA jefe

- A. En este caso, no se crea ningún protegible.
- B1. Creará el login del gerente general. Por ejemplo, puede autorizará RED\gerente para que se conecte a SQL Server.
- B2. Después a este login le hará usuario de las dos bases de datos "contabilidad" y "ventas".
- C1. Usuarios:
 - Al usuario de la BD de "ventas" correspondiente al gerente general le asignará el rol "empleadosventas"
 - Al usuario de la BD de "contabilidad" correspondiente al gerente general le asignará los roles "empleadoscontaguate" y "empleadoselsalvador".
- C2. En este caso, no hay C2.

475

475

5. Definir los aspectos para el gerente general. El DBA jefe Login "sa"

- B1. Creará el login del gerente general. Por ejemplo, puede autorizará RED\gerente para que se conecte a SQL Server.
- B2. Después a este login le hará usuario de las dos bases de datos "contabilidad" y "ventas".


```
create login [redgerente] from windows
use contabilidad
create user gerente for login [redgerente]
use ventas
create user gerente for login [redgerente]
```

476

476

5. Definir los aspectos para el gerente general.
El DBA jefe. Login "sa"

- C1. Usuarios:
 - Al usuario de la BD de "ventas" correspondiente al gerente general le asignará el rol "empleadosventas"

```
use ventas
alter role empleadosventas add member gerente
```

- Al usuario de la BD de "contabilidad" correspondiente al gerente general le asignará los roles "empleadoscontaguate" y "empleadoselsalvador".

```
use contabilidad
alter role empleadoscontaguate add member gerente
alter role empleadoscontael salvador add member gerente
```

477

477

Ejercicio de empresa de repuestos

- Tenemos una empresa de soporte a vehículos con 2000 empleados y una base de datos, que contiene tres tablas según las tres áreas de la empresa ("contabilidad", "compras", "ventas").
- Cada una de estas tablas tiene un campo Departamento, que indica de qué departamento es el registro respectivo (hay dos departamentos: "Repuestos" y "Gasolina").
- Hay una estructura matricial de forma que cada empleado pertenece a un área (y sólo una) y a un departamento (y sólo uno):
 - Cada empleado puede leer todos los datos de su departamento pero puede escribir sólo los datos que sean a la vez de su área y de su departamento.
 - Cada jefe de Departamento (sólo uno) puede leer todos los datos pero sólo escribir los de su departamento. Además, puede tomar propiedad de la vista correspondiente.

478

478

A. Crear las estructuras

- A1. Se le da la contraseña del login "sa" al DBA.
- A2. El DBA cambiará la contraseña.
- A3. El DBA crea la base de datos y las tres tablas con campo Departamento "contabilidad", "compras", "ventas"
 - (En realidad, se necesitarían más)
- A4. El DBA crea una vista para cada área y departamento.
- A5. Se crean los triggers para inserción de vistas.

479

479

A1 y A2

- A1. Se le da la contraseña del login "sa" al DBA. **En persona.**
- A2. El login "sa" cambia la contraseña.

```
alter login sa with password = 'contraseña'
```

480

480

A3. El DBA crea la base de datos y las tres tablas. Login "sa"

```
create database empresa;
create table contabilidad (
    idConta int identity not null primary key
    campo1 tipo1,
    ...,
    campon tipon
    departamento char(20) not null
)
```

481

481

A3. El DBA crea la base de datos y las tres tablas. Login "sa"

```
create table compras (
    idCompra int identity not null primary key
    campo1 tipo1,
    ...,
    campon tipon
    departamento char(20) not null
)
```

482

482

A3. El DBA crea la base de datos y las tres tablas. Login "sa"

```

create table ventas (
    idVentas int identity not null primary key
    campo1 tipo1,
    ...,
    campon tipon
    departamento char(20) not null
)
  
```

483

483

A4. El DBA crea una vista para cada área y departamento. Login "sa"

```

create view contarepuestos as
    select idConta, campo1, ..., campon
    from contabilidad
    where departamento = 'Repuestos'

create view contagasolina as
    select idConta, campo1, ..., campon
    from contabilidad
    where departamento = 'Gasolina'
  
```

484

484

A4. El DBA crea una vista para cada área y departamento. Login "sa"

```

create view comprasrepuestos as
    select idCompra, campo1, ..., campon
    from compras
    where departamento = 'Repuestos'

create view comprasgasolina as
    select idCompra, campo1, ..., campon
    from compras
    where departamento = 'Gasolina'
  
```

485

485

A4. El DBA crea una vista para cada área y departamento. Login "sa"

```

create view ventasrepuestos as
    select idVentas, campo1, ..., campon
    from ventas
    where departamento = 'Repuestos'

create view ventasgasolina as
    select idVentas, campo1, ..., campon
    from ventas
    where departamento = 'Gasolina'
  
```

486

486

A5. Se crean los triggers para inserción de vistas

```

create trigger triggercontarepuestos
on contarepuestos
instead of insert, update, delete as
begin
    delete contabilidad
    from deleted inner join contabilidad
    on deleted.idConta = contabilidad.idConta

    insert into contabilidad
    (campo1,..., campon, departamento)
    select campo1,..., campon, 'Repuestos'
    from inserted
end
  
```

487

487

A5. Se crean los triggers para inserción de vistas

```

create trigger triggercontagasolina
on contagasolina
instead of insert, update, delete as
begin
    delete contabilidad
    from deleted inner join contabilidad
    on deleted.idConta = contabilidad.idConta

    insert into contabilidad
    (campo1,..., campon, departamento)
    select campo1,..., campon, 'Gasolina'
    from inserted
end
  
```

488

488

A5. Se crean los triggers para inserción de vistas

```

create trigger triggercomprasrepuestos
on comprasrepuestos
instead of insert, update, delete as
begin
    delete compras
    from deleted inner join compras
    on deleted.idConta = compras.idConta

    insert into compras
    (campo1,..., campon, departamento)
    select campo1,..., campon, 'Repuestos'
    from inserted
end
  
```

489

489

A5. Se crean los triggers para inserción de vistas

```

create trigger triggercomprasgasolina
on comprasgasolina
instead of insert, update, delete as
begin
    delete compras
    from deleted inner join compras
    on deleted.idConta = compras.idConta

    insert into compras
    (campo1,..., campon, departamento)
    select campo1,..., campon, 'Gasolina'
    from inserted
end
  
```

490

490

A5. Se crean los triggers para inserción de vistas

```

create trigger triggerventasrepuestos
on ventasrepuestos
instead of insert, update, delete as
begin
    delete ventas
    from deleted inner join ventas
    on deleted.idConta = ventas.idConta

    insert into ventas
    (campo1,..., campon, departamento)
    select campo1,..., campon, 'Repuestos'
    from inserted
end
  
```

491

491

A5. Se crean los triggers para inserción de vistas

```

create trigger triggerventasgasolina
on ventasgasolina
instead of insert, update, delete as
begin
    delete ventas
    from deleted inner join ventas
    on deleted.idConta = ventas.idConta

    insert into ventas
    (campo1,..., campon, departamento)
    select campo1,..., campon, 'Gasolina'
    from inserted
end
  
```

492

492

B. Crear los logines y usuarios

- B1. El DBA crea los logines de Windows
- B2. El DBA los autoriza para la base de datos.

493

493

B1. El login "sa" crea los logines de Windows

```

create login [contabilidadgasolina1] from Windows
...
create login [contabilidadgasolinan] from windows
create login [contabilidadrepuestos1] from windows
...
create login [contabilidadrepuestosn] from windows
create login [comprasgasolina1] from windows
...
create login [comprasgasolinan] from windows
create login [comprasrepuestos1] from windows
...
create login [comprasrepuestosn] from windows
create login [ventasgasolina1] from windows
...
create login [ventasgasolinan] from windows
create login [ventasrepuestos1] from windows
...
create login [ventasrepuestosn] from windows
  
```

494

494

B1. El login "sa" crea los logines de Windows

```
create login [gerentes\jefeRepuestos] from windows
create login [gerentes\jefeGasolina] from windows
```

495

495

B2. El login "sa" crea usuarios. use empresa

```
create user cntgas1 for login [contabilidad\gasolina1]
...
create user cntgasn for login [contabilidad\gasolinan]
create user cntrep1 for login [contabilidad\repuestos1]
...
create user cntrepn for login [contabilidad\repuestosn]
create user cmpgas1 for login [compras\gasolina1]
...
create user cmpgasn for login [compras\gasolinan]
create user cmprep1 for login [compras\repuestos1]
...
create user cmprepn for login [compras\repuestosn]
create user vntgas1 for login [ventas\gasolina1]
...
create user vntgasn for login [ventas\gasolinan]
create user vntrep1 for login [ventas\repuestos1]
...
create user vntrepn for login [ventas\repuestosn]
```

496

496

B2. El login "sa" crea usuarios. use empresa

```
create user jefeRepuestos for login [gerentes\jefeRepuestos]
create user jefeGasolina for login [gerentes\jefeGasolina]
```

497

497

C. Crear los permisos

- C1. El login "sa" crea roles para cada combinación de área y departamento.
- C2. El login "sa" crea roles para cada departamento.
- C3. El login "sa" da permisos de lectura a los roles de los empleados.
- C4. El login "sa" da permisos de escritura a los roles de los empleados.
- C5. El login "sa" da permisos de lectura a los jefes.
- C6. El login "sa" da permisos de escritura a los jefes.
- C7. El login "sa" da permisos de take ownwerhip a los jefes.

498

498

C1. El login "sa" crea roles para para cada combinación de área y departamento.

```
create role users_cntgas authorization dbo
alter role users_cntgas add member cntgas1
...
alter role users_cntgas add member cntgasn
create role users_cntrep authorization dbo
alter role users_cntrep add member cntrep1
...
alter role users_cntrep add member cntrepn
create role users_cmpgas authorization dbo
alter role users_cmpgas add member cmpgas1
...
alter role users_cmpgas add member cmpgasn
create role users_cmprep authorization dbo
alter role users_cmprep add member cmprep1
...
alter role users_cmprep add member cmprepn
create role users_vntgas authorization dbo
alter role users_vntgas add member vntgas1
...
alter role users_vntgas add member vntgasn
create role users_vntrep authorization dbo
alter role users_vntrep add member vntrep1
...
alter role users_vntrep add member vntrepn
```

499

499

C2. El login "sa" crea roles para cada departamento

```
create role users_repuestos authorization dbo
alter role users_repuestos add member users_cntrep
alter role users_repuestos add member users_cmprep
alter role users_repuestos add member users_vntrep

create role users_gasolina authorization dbo
alter role users_gasolina add member users_cntgas
alter role users_gasolina add member users_cmpgas
alter role users_gasolina add member users_vntgas
```

500

500

C3. El login "sa" da permisos de lectura a los roles de los empleados

```
grant select on contarepuestos to users_repuestos
grant select on comprasrepuestos to users_repuestos
grant select on ventasrepuestos to users_repuestos

grant select on contagasolina to users_gasolina
grant select on comprasgasolina to users_gasolina
grant select on ventasgasolina to users_gasolina
```

501

C4. El login "sa" da permisos de escritura a los roles de los empleados

```
grant insert,update,delete on contarepuestos to users_cntrep
grant insert,update,delete on contagasolina to users_cntgas

grant insert,update,delete on comprasrepuestos to users_cmprep
grant insert,update,delete on comprasgasolina to users_cmpgas

grant insert,update,delete on ventasrepuestos to users_cmprep
grant insert,update,delete on ventasgasolina to users_cmpgas
```

502

C5. El login "sa" da permisos de lectura a los jefes

```
grant select on contarepuestos to jefeRepuestos
grant select on comprasrepuestos to jefeRepuestos
grant select on ventasrepuestos to jefeRepuestos

grant select on contagasolina to jefeRepuestos
grant select on comprasgasolina to jefeRepuestos
grant select on ventasgasolina to jefeRepuestos
```

503

C5. El login "sa" da permisos de lectura a los jefes

```
grant select on contarepuestos to jefeGasolina
grant select on comprasrepuestos to jefeGasolina
grant select on ventasrepuestos to jefeGasolina

grant select on contagasolina to jefeGasolina
grant select on comprasgasolina to jefeGasolina
grant select on ventasgasolina to jefeGasolina
```

504

C6. El login "sa" da permisos de escritura a los jefes.

```
grant insert,update,delete on contarepuestos to jefeRepuestos
grant insert,update,delete on comprasrepuestos to jefeRepuestos
grant insert,update,delete on ventasrepuestos to jefeRepuestos

grant insert,update,delete on contagasolina to jefeGasolina
grant insert,update,delete on comprasgasolina to jefeGasolina
grant insert,update,delete on ventasgasolina to jefeGasolina
```

505

C7. El login "sa" da permisos de take ownwerhip a los jefes.

```
grant take ownership on contarepuestos to jefeRepuestos
grant take ownership on comprasrepuestos to jefeRepuestos
grant take ownership on ventasrepuestos to jefeRepuestos

grant take ownership on contagasolina to jefeGasolina
grant take ownership on comprasgasolina to jefeGasolina
grant take ownership on ventasgasolina to jefeGasolina
```

506