

Unidad 5

Protección de los datos

Bases de Datos Aplicada
v1.0 – Octubre 2023



Universidad Nacional
de La Matanza

DIIT
Departamento de Ingeniería e
Investigaciones Tecnológicas

Contenido

- Conceptos básicos de seguridad. Principio del mínimo privilegio.
- Usuarios, roles, permisos, propietarios.
- Registro de transacciones.
- Copias de seguridad.
- Alta disponibilidad.
- Encriptación en tránsito y at rest.

Al finalizar deberías ser capaz de...

- Securizar de *forma mínima* un sistema.
- Generar una estructura de usuarios y permisos siguiendo el POLP.
- Generar respaldos y restaurar copias de seguridad de DB.
- Definir una estrategia de backups acorde a la solución.
- Comprender el funcionamiento del mecanismo de Alta Disponibilidad.
- Comprender el cifrado para los datos en reposo y en tránsito.

Seguridad vs Protección

Seguridad: **ausencia de un riesgo**. Aplicando esta definición a al tema correspondiente, se hace referencia al riesgo de accesos no autorizados, de manipulación de información, manipulación de las configuraciones, entre otros

Protección: **mecanismos empleados para proteger** datos de ser alterados o borrados, de acceso no autorizado, etc.

Protección: un policía por cuadra. Aumenta la seguridad... pero no nos previene de un ataque ransomware!

<https://techcommunity.microsoft.com/t5/azure-sql-blog/security-the-principle-of-least-privilege-polp/ba-p/2067390>

Seguridad vs Protección

Principio del mínimo privilegio (POLP): A los usuarios y las aplicaciones se les debe otorgar acceso **solo a los datos y operaciones que requieren** para realizar su trabajo.



Reducción del área expuesta: implica detener o deshabilitar componentes que no se utilizan. Reduce los puntos de acceso para potenciales ataques. Los servicios deben ejecutarse con privilegios mínimos.

<https://techcommunity.microsoft.com/t5/azure-sql-blog/security-the-principle-of-least-privilege-polp/ba-p/2067390>

Seguridad vs Protección

¿Qué medidas de protección podemos implementar en un DBMS?

- Restricciones y permisos.
- Respaldos.
- Encriptación en tránsito y at rest.
- Replicación.
- Buenas prácticas de programación (p/e prevenir inyección SQL).

No termina ahí... debemos securizar el sistema operativo, el software del DBMS, firewall, el hardware, etc, etc.

<https://learn.microsoft.com/es-es/sql/relational-databases/security/securing-sql-server?view=sql-server-ver16>

Release	RTM (no SP)	Latest CU			
SQL Server 2022 <div>SQL Server latest version</div> <div>SQL Server 16</div> <div>codename Dallas</div> <div>Release date: 2022-11-16</div> <div>Support end date: 2028-01-11</div> <div>Ext. end date: 2033-01-11</div>	16.0.1000.6	CU9 (16.0.4085.2, October 2023)			
SQL Server 2019 <div>SQL Server 15</div> <div>codename Aris Seattle</div> <div>Release date: 2019-11-04</div> <div>Support end date: 2025-01-07</div> <div>Ext. end date: 2030-01-08</div>	15.0.2000.5	CU23 (15.0.4335.1, October 2023)			
SQL Server 2017 <div>SQL Server 14</div> <div>codename vNext</div> <div>Release date: 2017-10-02</div> <div>Support end date: 2022-10-11</div> <div>Ext. end date: 2027-10-12</div>	14.0.1000.169	CU31 (14.0.3456.2, September 2022)			
Starting from SQL Server 2017 Service Packs will no longer be released					
		SP1	SP2	SP3	SP4
SQL Server 2016 <div>SQL Server 13</div> <div>Release date: 2016-06-01</div> <div>Support end date: 2021-07-13</div> <div>Ext. end date: 2026-07-14</div>	13.0.1601.5 + CU9	13.0.4001.0 or 13.1.4001.0 + CU15	13.0.5026.0 or 13.2.5026.0 + CU17	13.0.6300.2 or 13.3.6300.2	
SQL Server 2014 <div>SQL Server 12</div> <div>Release date: 2014-04-01</div> <div>Support end date: 2019-07-09</div> <div>Ext. end date: 2024-07-09</div>	12.0.2000.8 + CU14	12.0.4100.1 or 12.1.4100.1 + CU13	12.0.5000.0 or 12.2.5000.0 + CU18	12.0.6024.0 or 12.3.6024.0 + CU4	
Obsolete versions – out of support					
SQL Server 2012 <div>SQL Server 11</div> <div>codename Denali</div> <div>Release date: 2012-03-06</div> <div>Support end date: 2017-07-11</div> <div>Ext. end date: 2022-07-12</div>	11.0.2100.60 + CU11	11.0.3000.0 or 11.1.3000.0 + CU16	11.0.5058.0 or 11.2.5058.0 + CU16	11.0.6020.0 or 11.3.6020.0 + CU10	11.0.7001.0 or 11.4.7001.0

Mantenga su instalación al día
sqlserverbuilds.blogspot.com

Permisos, usuarios y roles

❑ Entidades de seguridad (*principals*)

- Individuos, grupos y procesos que tienen acceso (pueden solicitar recursos) al sistema. Pueden definirse a nivel de sistema operativo, de servidor SQL o de base de datos.

❑ El usuario sa (*sysadmin*)

- **Entidad** de seguridad a nivel servidor SQL.
- Puede deshabilitarse.

❑ Elementos protegibles (*securables*)

- Servidor, base de datos y objetos incluidos en ella. Cada uno de estos posee un conjunto de permisos que pueden configurarse para reducir el área expuesta.

Permisos, usuarios y roles

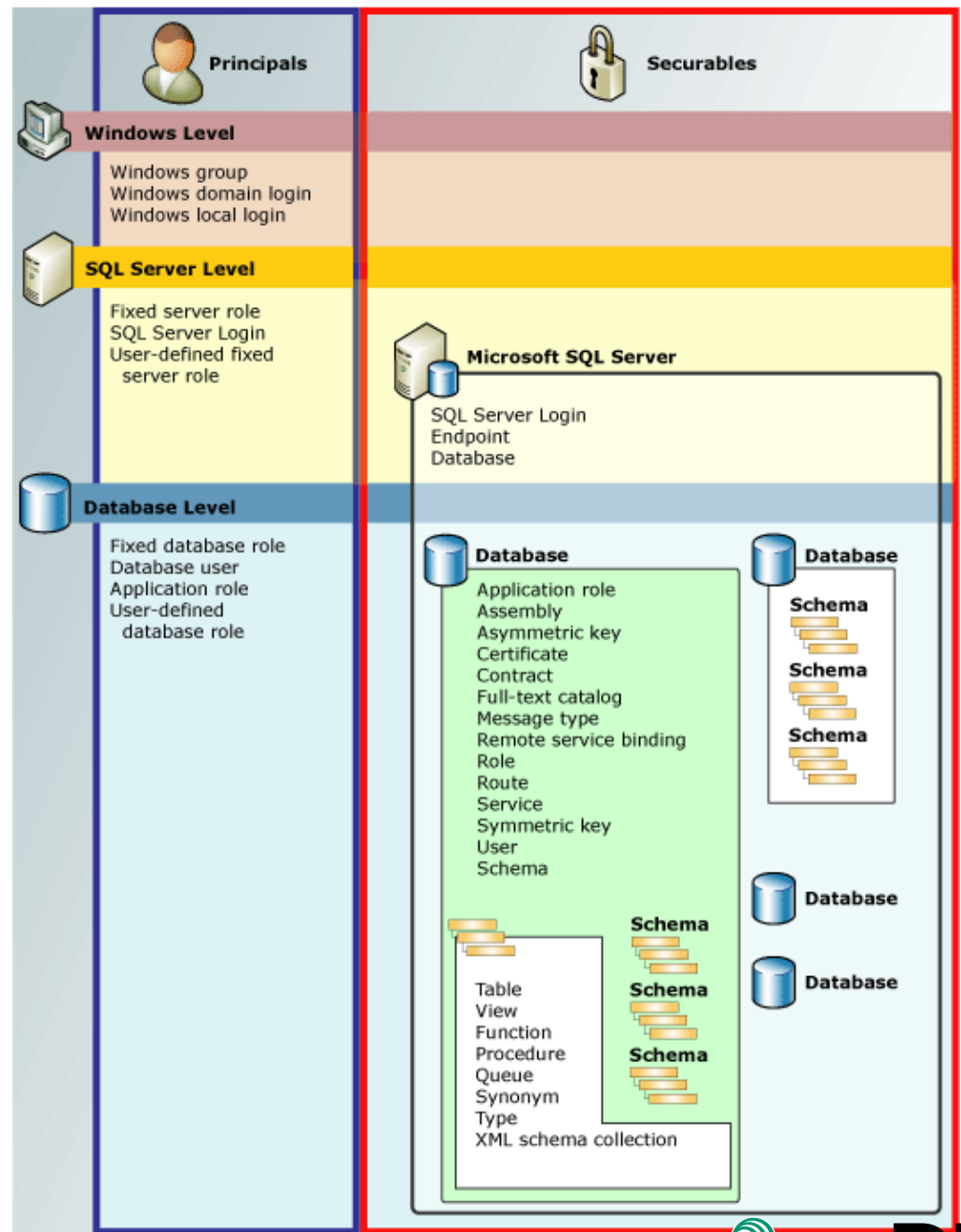
❑ Directivas de contraseñas.

- Sensibles a mayúsculas/minúsculas.
- Complejidad: largo mínimo, tres de cuatro: mayúsculas, minúsculas, números, símbolos no alfanuméricos (CHECK_POLICY).
- Si se apoya en el sistema operativo, aplican las del mismo.
- Vencimiento (CHECK_EXPIRATION).
- Se puede forzar el cambio en siguiente LOGIN.
- Si usamos ODBC cuidado con los caracteres empleados.

<https://learn.microsoft.com/en-us/sql/relational-databases/security/password-policy?view=sql-server-ver16>

Principals & Securables

<https://learn.microsoft.com/es-es/sql/relational-databases/security/permissions-hierarchy-database-engine?view=sql-server-ver16>



Bases de Datos Aplicada

Permisos, usuarios y roles

```
CREATE LOGIN <login_name>  
    WITH PASSWORD = '<StrongPasswordHere>';
```

```
CREATE LOGIN <login_name> WITH PASSWORD =  
'<StrongPasswordHere>'  
    MUST_CHANGE, CHECK_EXPIRATION = ON;
```

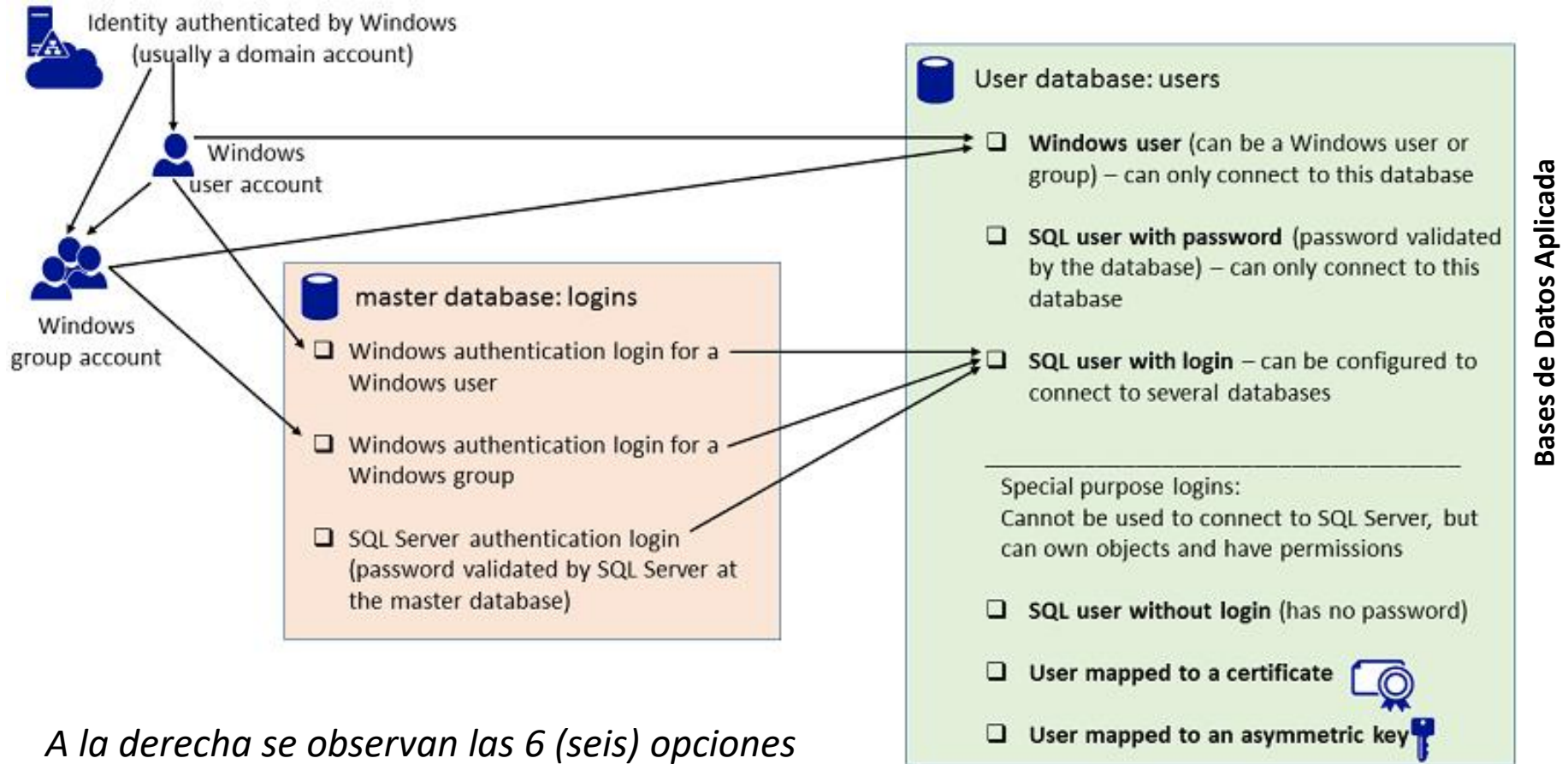
```
CREATE LOGIN [<domainName>\<login_name>] FROM WINDOWS;
```

```
CREATE LOGIN [MyUser]  
WITH PASSWORD = 'MyPassword', DEFAULT_DATABASE = MyDatabase,  
CHECK_POLICY = OFF, CHECK_EXPIRATION = OFF ;
```

Un LOGIN permite acceso al servidor, pero por sí mismo no otorga permisos para ninguna DB.

<https://learn.microsoft.com/en-us/sql/t-sql/statements/create-login-transact-sql?view=sql-server-ver16>

Permisos, usuarios y roles



A la derecha se observan las 6 (seis) opciones para crear usuarios.

<https://learn.microsoft.com/es-es/sql/relational-databases/security/authentication-access/create-a-database-user?view=sql-server-ver16>

Permisos, usuarios y roles

❑ Usuario de base de datos

- El LOGIN permite acceder al servidor pero *por defecto no otorga permisos de acceso a ninguna DB.*
- Debemos crear un USUARIO a **nivel DB** y darle permisos.
- El nombre de USUARIO puede ser distinto al nombre de LOGIN.
Pero solo podemos crear UN usuario para un login.
- El usuario dbo o propietario es una cuenta de usuario con permisos implícitos para realizar todas las actividades en la DB.

```
USE [BasesDatosAplicada]
```

```
GO
```

```
CREATE USER [pruebaUser] FOR LOGIN [testuser]
```

Permisos, usuarios y roles

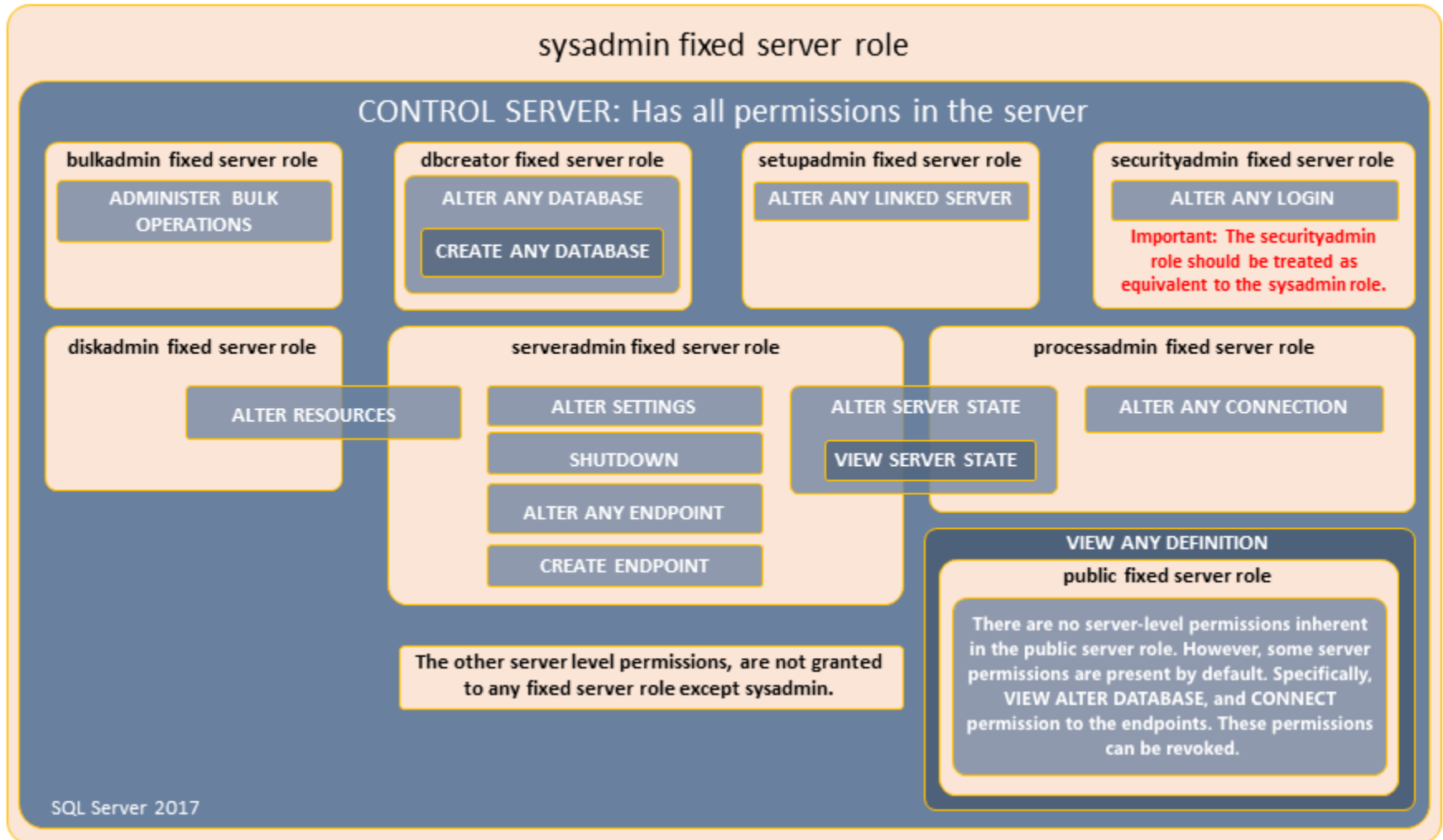
❑ Roles fijos de servidor

- Son entidades de seguridad que agrupan otras entidades de seguridad. Se aplican a todo el servidor en lo que respecta a su ámbito de permisos.
- No se pueden modificar (en la última versión se pueden crear roles definidos por el usuario).
- Se heredan a las DB si el usuario tiene permisos de conexión a la DB.

<https://learn.microsoft.com/es-es/sql/relational-databases/security/authentication-access/server-level-roles?view=sql-server-ver16>

Permisos, usuarios y roles

SERVER LEVEL ROLES AND PERMISSIONS: 9 fixed server roles, 34 server permissions



Permisos, usuarios y roles

❑ Roles fijos de servidor

- Se puede agregar un LOGIN a un rol para darle permisos.

```
ALTER SERVER ROLE Production ADD MEMBER  
[servidorFulano\usuarioMengano] ;
```

```
ALTER SERVER ROLE diskadmin ADD MEMBER Ted ;  
GO
```

```
ALTER SERVER ROLE Production DROP MEMBER Ted ;  
GO
```

<https://learn.microsoft.com/es-es/sql/t-sql/statements/alter-server-role-transact-sql?view=sql-server-ver16>

Permisos, usuarios y roles

❑ Roles de nivel base de datos

- Es mejor práctica otorgar permisos a roles en lugar de a usuarios.
- Todos los miembros del rol heredan los permisos.
- Los roles se pueden anidar (cuidado).
- Los usuarios se asignan a los roles con `ADD MEMBER` y `DROP MEMBER`.
- Puede conceder permisos en el nivel de esquema. Los usuarios heredan los permisos en todos los objetos NUEVOS creados en el esquema.

<https://learn.microsoft.com/es-es/sql/relational-databases/security/authentication-access/database-level-roles?view=sql-server-ver16>

Permisos, usuarios y roles

❑ Propietarios

- Los propietarios de los objetos disponen de permisos irrevocables para administrarlos. No se pueden eliminar usuarios si existen objetos que les pertenezcan. No se pueden quitar los privilegios del propietario.
- Los esquemas pueden pertenecer a cualquier entidad de seguridad. Una entidad puede poseer varios esquemas.
- La propiedad de un objeto se puede cambiar con ALTER
AUTHORIZATION

<https://learn.microsoft.com/es-es/sql/relational-databases/security/authentication-access/ownership-and-user-schema-separation?view=sql-server-ver16>

<https://learn.microsoft.com/es-es/sql/t-sql/statements/alter-authorization-transact-sql?view=sql-server-ver16>

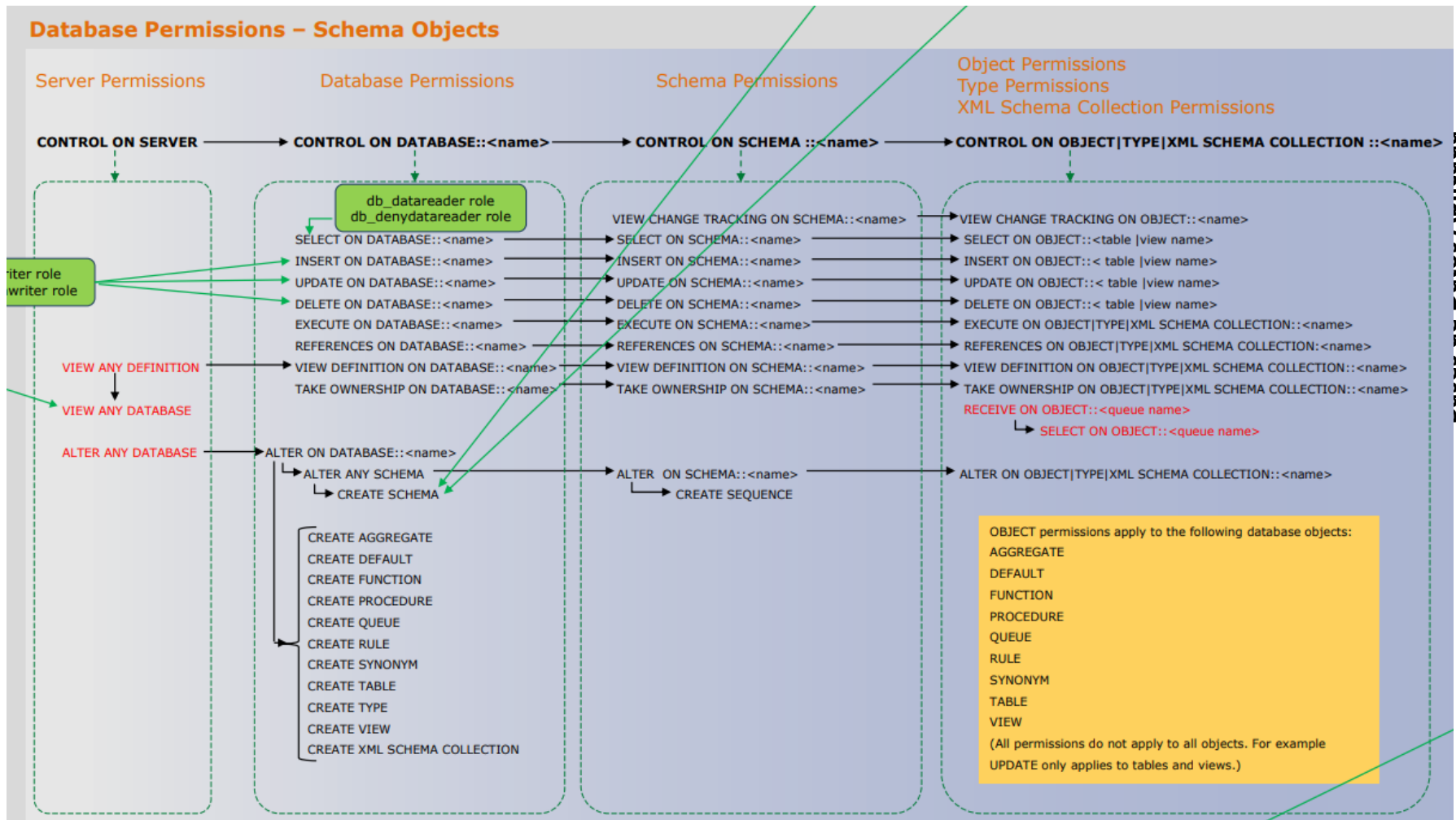
Permisos, usuarios y roles

□ DCL Data Control Language:

- GRANT
 - Concede permiso. Puede además dar permiso para conceder permisos (WITH GRANT OPTION).
- REVOKE
 - Revoca un permiso. Un permiso revocado se puede heredar de OTRO GRUPO o ROL.
- DENY
 - Revoca un permiso de manera que no pueda ser heredado.

<https://learn.microsoft.com/es-es/sql/t-sql/statements/alter-server-role-transact-sql?view=sql-server-ver16>

Permisos, usuarios y roles



Permisos, usuarios y roles

□ Jerarquía de permisos

- Si se concede el permiso SELECT en una DB, incluirá todos los esquemas.
- Si se concede el permiso SELECT en un esquema, incluirá todas las tablas y vistas del esquema.
 - *Si los objetos que requieren **mismos permisos** se encuentran en el **mismo esquema** se simplifica **drásticamente**.*
- El permiso CONTROL en un objeto normalmente concede todos los otros permisos del objeto.

<https://learn.microsoft.com/es-es/sql/t-sql/statements/alter-server-role-transact-sql?view=sql-server-ver16>

Permisos, usuarios y roles

- Supongamos la tabla *Region* en el esquema *Clientes* de la base de datos *Ventas*. El usuario *Aza* obtendría permiso *SELECT* en la tabla *Region* mediante cualquiera de estas instrucciones:

```
GRANT SELECT ON OBJECT::Region TO Aza;
```

```
GRANT CONTROL ON OBJECT::Region TO Aza;
```

```
GRANT SELECT ON SCHEMA::Customers TO Aza;
```

```
GRANT CONTROL ON SCHEMA::Customers TO Aza;
```

```
GRANT SELECT ON DATABASE::SalesDB TO Aza;
```

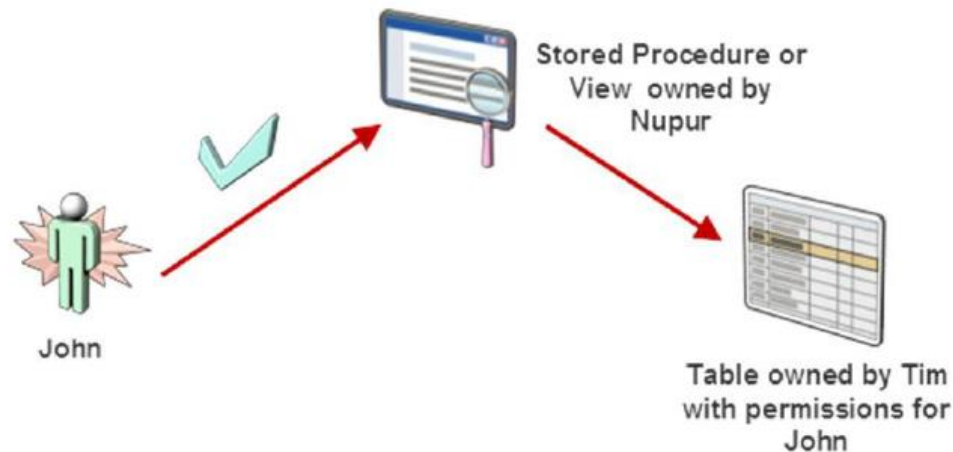
```
GRANT CONTROL ON DATABASE::SalesDB TO Aza;
```

<https://learn.microsoft.com/es-es/sql/relational-databases/security/authentication-access/getting-started-with-database-engine-permissions?view=sql-server-ver16>

Permisos, usuarios y roles

❑ Permisos mediante código basado en procedimiento

- Se puede evitar que los usuarios interactúen directamente con los objetos de la DB otorgando permiso solo a SP o funciones y denegando permisos a objetos subyacentes.
- Encadenamiento de propiedad.



<https://learn.microsoft.com/en-us/sql/relational-databases/tutorial-ownership-chains-and-context-switching?view=sql-server-ver16>

Registro de Transacciones (RT)

□ ¿Qué es una transacción?

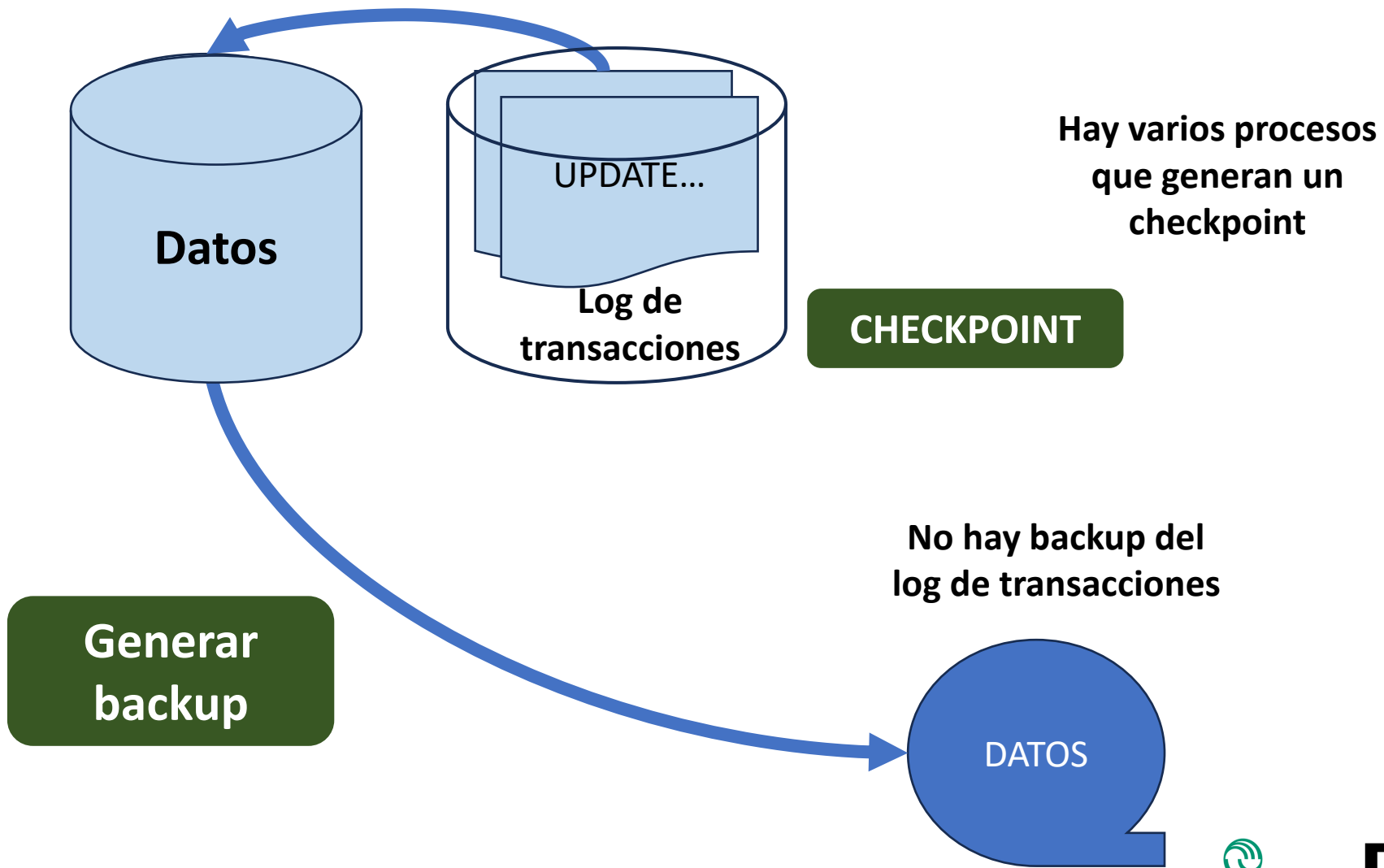
- La unidad de trabajo más pequeña que se ejecuta en la DB.
- Cumplen con las propiedades ACID.

Aunque todas las DB admiten el manejo implícito o explícito de las transacciones, hay distintos **modelos de recuperación**.

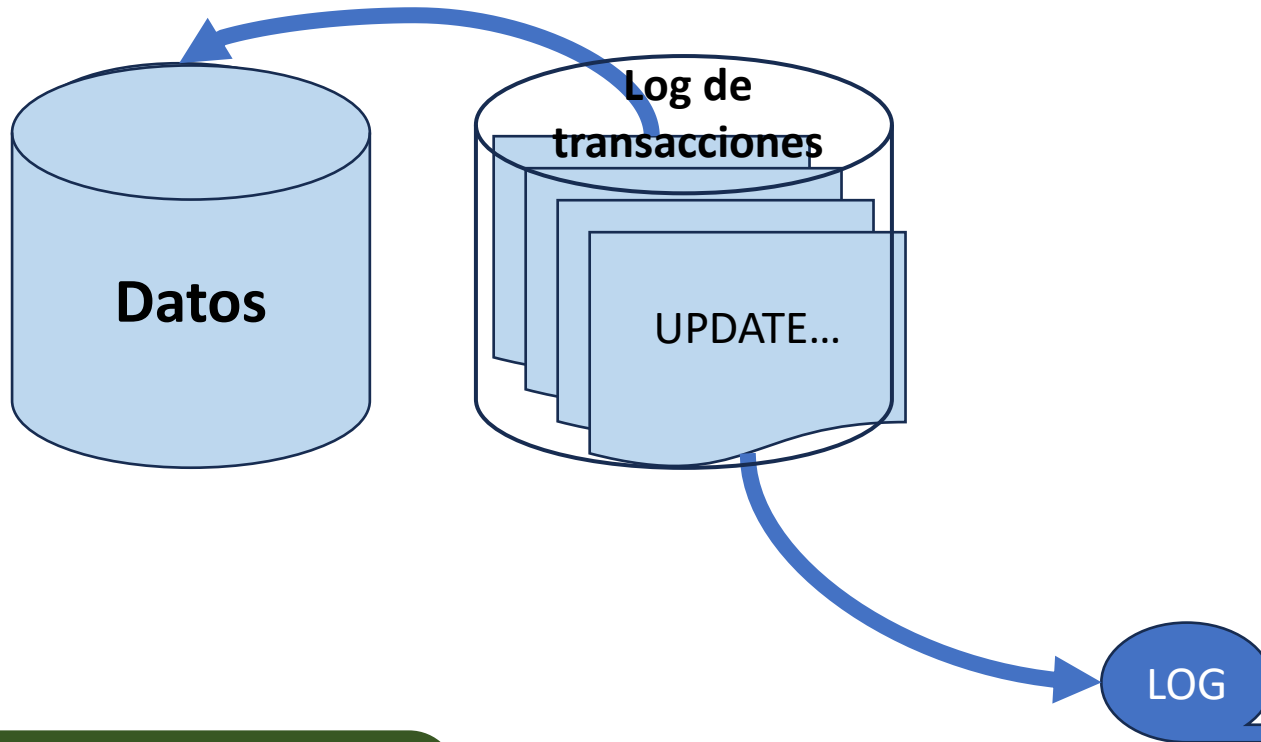
El modelo **SIMPLE** no contempla respaldo del log de transacciones.

El modelo **FULL** y **BULK-LOGGED** admiten y requieren respaldo del log de transacciones.

Modelo de recuperación SIMPLE



Modelo de recuperación FULL



**Generar backup
del log de
transacciones**

Registro de Transacciones (RT)

□ ¿Qué es el registro de transacciones?

- Es donde se registran todas las transacciones y las modificaciones que cada transacción realiza en la DB.
- Es un componente esencial de la base de datos.
- Si hay un error del sistema, ese registro será necesario para devolver la base de datos a un estado coherente.

□ ¿Cómo se implementa?

- En un archivo o grupo de archivos separado de la DB.
- Al crear la DB se establece su ubicación y configuración de crecimiento.

Ver Capítulo 23 de Elmasri-Navathe

Registro de Transacciones (RT)

□ ¿Qué almacena un registro de transacciones SQL Server?

- Almacena cada transacción hecha en una base de datos SQL Server, excepto algunas que son mínimamente registradas como BULK IMPORT o SELECT INTO.
- Internamente está dividido en partes más pequeñas llamadas *Archivos de Registros Virtuales* (Virtual Log Files, VLFs).
- Cuando un VLF se llena, el registro continúa en el siguiente registro de transacciones disponible.

Registro de Transacciones (RT)

□ Archivos de registro virtuales (VLF).

- El DBMS SQL Server divide cada archivo de registro físico internamente en varios archivos de registro virtuales (VLF).
- Los VLF no tienen un tamaño fijo y no hay una cantidad fija de archivos de registro virtuales para un archivo de registro físico.
- El **motor de base de datos elige dinámicamente el tamaño** de los archivos de registro virtuales mientras crea o extiende archivos de registro.

Registro de Transacciones (RT)

❑ Archivos de registro virtuales (VLF).

- La vista *sys.dm_db_log_info*, devuelve información del archivo de registro virtual (VLF) del registro de transacciones.
- Cada fila de la salida representa un VLF en el registro de transacciones y proporciona información relevante para ese VLF en el registro.

```
SELECT db.name, count(dbl.database_id) CuentaTotalVLF,  
       convert(decimal (10,2), avg(dbl.vlf_size_mb))  
       TamanoPromedioVLFMB  
FROM sys.databases db CROSS APPLY  
     sys.dm_db_log_info(db.database_id) dbl  
GROUP BY db.name  
ORDER BY Total_VLF_count DESC
```

En este ejemplo se puede ver el recuento de VLF y su tamaño promedio.

Registro de Transacciones (RT)

□ Archivos de registro virtuales (VLF).

- Cuando un VLF se llena, el registro continúa en el siguiente registro de transacciones disponible.
- El archivo de registro de transacciones puede ser representado como un **archivo circular** que cuando el registro llega al final del archivo, inicia de nuevo desde el principio, pero sólo si todos los requerimientos han sido cumplidos y las partes inactivas han sido truncadas.
- El proceso de **truncar** es **necesario** para marcar todas las partes inactivas de modo que **puedan ser usadas de nuevo y sobrescritas**. (*Truncar no libera el espacio en disco*).

Registro de Transacciones (RT)

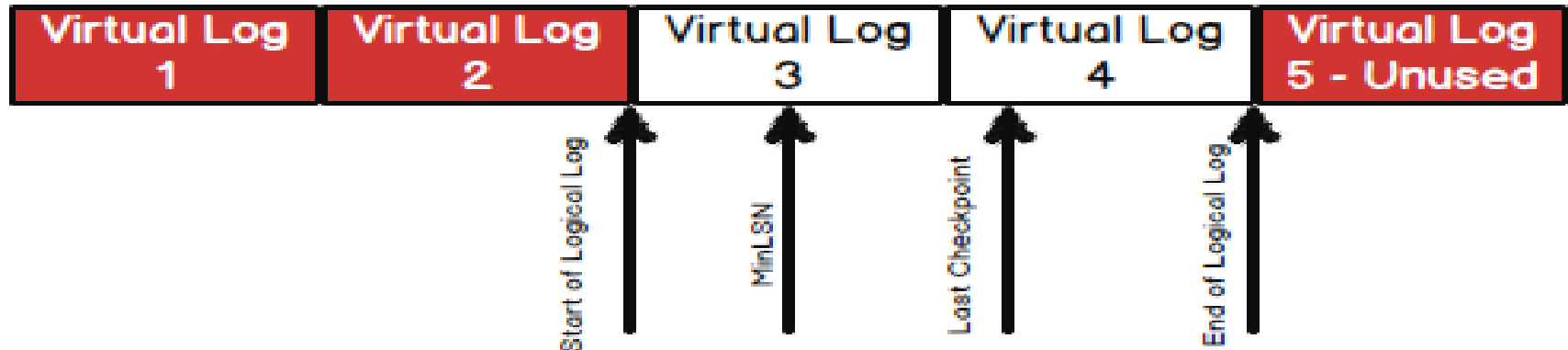
❑ Archivos de registro virtuales (VLF).

- Un registro ya no es necesario en el registro de transacciones si se cumplen todas estas premisas:
 - ✓ La transacción de la que es parte se ha confirmado.
 - ✓ Las páginas de la base de datos que cambió han sido todas escritas en un CHECKPOINT.
 - ✓ El registro no es necesario para una copia de seguridad (completa, diferencial o de log)
 - ✓ El registro no es necesario para ninguna característica que lee el registro (tales como mirroring o replicación)

Registro de Transacciones (RT)

❑ Archivos de registro virtuales (VLF).

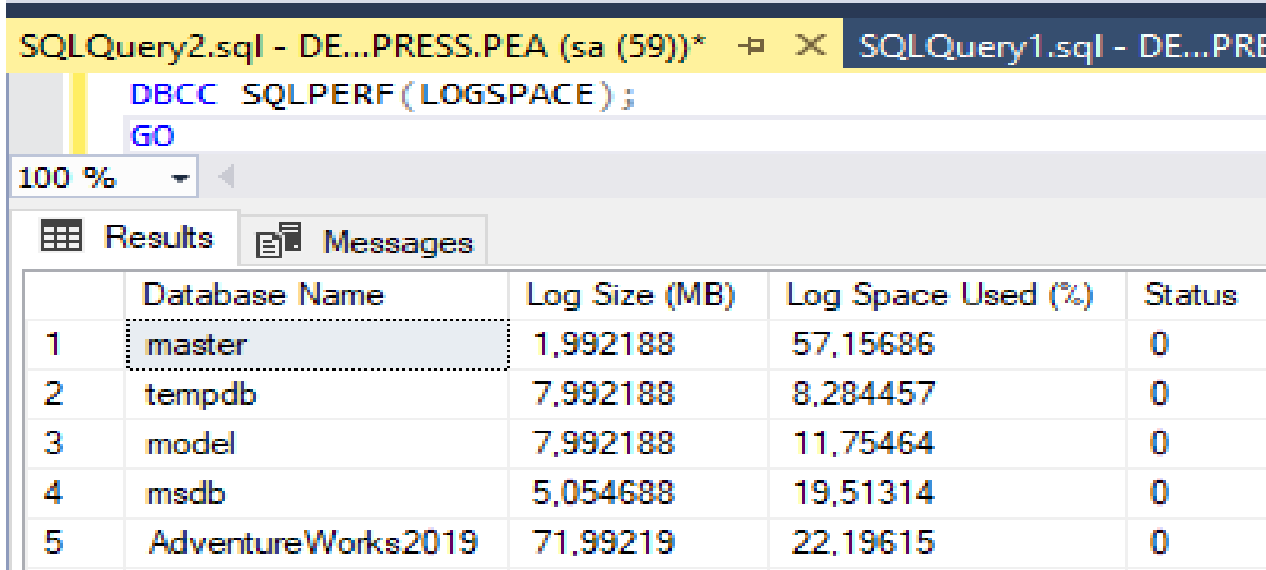
- El registro lógico es una parte del registro de transacciones.
- El Log Sequence Number (LSN) identifica cada transacción en el registro de transacciones.
- EL MinLSN es el punto de partida de la transacción activa más antigua en el registro de transacciones en línea.



Registro de Transacciones (RT)

❑ Mantenimiento del log de transacciones

- Es una tarea importante en la administración de SQL Server.
- Se recomienda **monitorear su crecimiento** diariamente o incluso más frecuentemente si la base de datos SQL Server tiene una gran cantidad de tráfico.
- El espacio del registro de transacciones puede ser obtenido usando el comando **DBCC SQLPERF**:



```
SQLQuery2.sql - DE...PRESS.PEA (sa (59))*  SQLQuery1.sql - DE...PRE
DBCC SQLPERF (LOGSPACE) ;
GO
```

	Database Name	Log Size (MB)	Log Space Used (%)	Status
1	master	1,992188	57,15686	0
2	tempdb	7,992188	8,284457	0
3	model	7,992188	11,75464	0
4	msdb	5,054688	19,51314	0
5	AdventureWorks2019	71,99219	22,19615	0

Registro de Transacciones (RT)

□ Backup

- Debe ser respaldado de forma regular para controlar la operación de **crecimiento automática** y **evitar que se llene** el archivo del registro de transacciones.
- No debe confundirse con el backup de los archivos de datos.
- No está disponible para el modelo de recuperación SIMPLE.
- Una vez realizado el backup el espacio del archivo del log de transacciones puede ser reutilizado.

```
BACKUP LOG ACMEDB  
TO DISK = 'F:\ACMEDB.TRN'  
GO
```

Respaldo de la base de datos

El objetivo de las copias de respaldo es **protegernos** ante la **pérdida de datos** por cualquier causa (falla de HW, error humano, falla de SW, malware, etc.).

Son un mecanismo para **asegurar** las propiedades ACID, ya que deben permitirnos mantener la DB **consistente**, incluso si ocurre un evento catastrófico durante una transacción.



Respaldo de la base de datos

□ Clasificación

- **Completo:** base de datos completa.
- **Diferencial** o incremental: solo incluye cambios desde el último completo.
- **Registro de transacciones:** solo del log, desde el último completo o desde el último backup de log.

□ Otros tipos de backup

- **Tail log:** respaldo de log de transacciones restante.
- **Copy only:** respaldo independiente de la secuencia de respaldos (ideal para exportar).

Respaldo de la base de datos

❑ Completo (FULL)

- Respalda TODA la DB. Copia completa incluyendo todos los objetos de la DB. Permite restaurar la DB a su estado preciso al momento del respaldo.
- Es la base de cualquier estrategia de respaldo. Antes de realizarse otro tipo de backup primero debe contarse con un FULL.
- Incluye una copia del log de transacciones.

Respaldo de la base de datos

□ Diferencial

- Se respaldan todos los **componentes** de la DB **modificados** a partir del último backup completo.
- El **tamaño** del respaldo **dependerá** de los cambios efectuados.
- Solo se respalda el **estado actual** de cada objeto modificado.
- El tiempo transcurrido desde el último backup FULL y los cambios efectuados en la DB son los factores que determinarán el tamaño del respaldo diferencial.
- Son **acumulativos**.

Respaldo de la base de datos

□ Registro de transacciones

- Resguarda el log de transacciones, que contiene la **historia de cada modificación** realizada en una DB.
- Cada respaldo del log de transacciones contiene los registros generados desde el respaldo del log de transacciones anterior.
- Son incrementales. Para restaurar la DB a un **momento específico** en el tiempo debemos restaurar el último backup FULL, el último diferencial y todos los respaldos del log de transacciones desde este.

Respaldo de la base de datos

□ Estrategia de backups

¿Cuánto tiempo lleva realizar cada backup y cuánto tiempo puede tomarle restaurar el sistema?

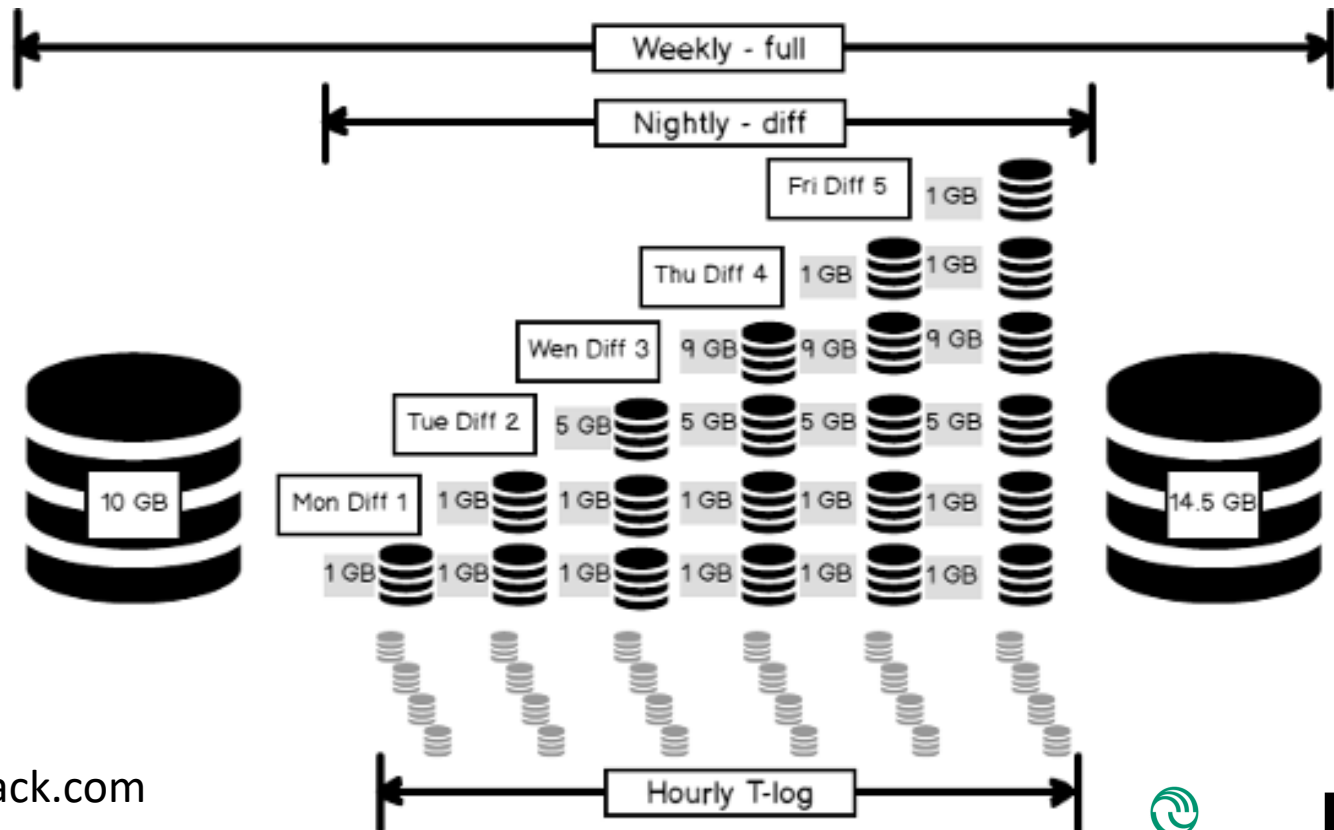


Imagen: sqlshack.com

Respaldo de la base de datos

□ Tail log

- Ante una **DB dañada u offline** es posible resguardar el log de transacciones para recuperar los últimos cambios realizados (intentarlo al menos)
- Luego de recuperar el sistema con los respaldos completos, diferenciales (si hubiera) y del log de transacciones, se puede usar el backup del tail log para minimizar la pérdida de datos.

<https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/tail-log-backups-sql-server?view=sql-server-ver16>

Respaldo de la base de datos

□ Copy only

- No modifica el nro de secuencia como sí ocurre con los backups completos.
- Observe que al modificarse D, E, F, no se toma como base el *COPY ONLY* sino el FULL anterior.

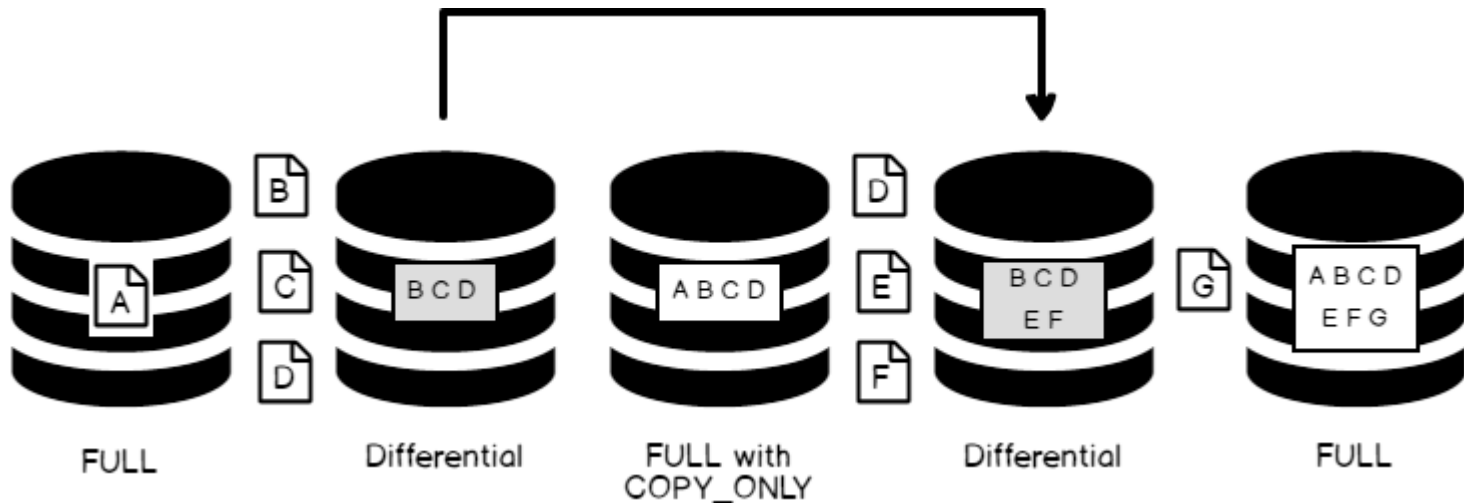


Imagen: sqlshack.com

Respaldo de la base de datos

❑ Otras opciones

- Se pueden generar backups comprimidos.
- Se pueden generar backups cifrados.

❑ Buenas prácticas

- Estrategia 3-2-1.
 - Tres copias
 - En al menos dos lugares o medios.
 - Uno de ellos siendo la nube.
- Verifique los backups. SIEMPRE.
- ¡No se conforme con realizar restauración! *DBCC checks*



Imagen: veeam.com

Restauración de la base de datos

- Operación que permite restaurar una base de datos a un estado anterior a través de la recuperación de una copia de seguridad previamente creada.
- Las operaciones que incluya **dependerán de la estrategia** empleada y del momento del evento catastrófico.
- Recomendación: **documente TODO. Pruebe TODO.**

Restauración de la base de datos

Al restaurar podrá indicar si desea:

- Sobrescribir la base de datos existente (WITH REPLACE)
- Conservar la configuración de replicación (WITH KEEP_REPLICATION)
- Restringir el acceso a la base de datos restaurada (WITH RESTRICTED_USER)
- Modificar la ruta de almacenamiento de los archivos de los FILEGROUP (WITH MOVE). Recuerde que MDF y LDF son los dos archivos mínimos a restaurar.

Podemos verificar el encabezado para ver el contenido de un respaldo.

```
RESTORE HEADERONLY FROM DISK = 'F:\AdventureWorks2019.bak'
```

Restauración de la base de datos

Al restaurar una DB debemos indicar el estado en que deseamos dejar la DB al completar la operación:}

- RECOVERY (default): Deja la DB lista para ser utilizada. No se podrán restaurar respaldos adicionales.
- NORECOVERY: La DB aun no puede ser utilizada. Permite restaurar respaldos adicionales (diferenciales, de registro de transacciones).

Podemos restaurar un backup de log hasta un punto específico:

```
RESTORE LOG customer FROM DISK = 'f:\backup\customer.bak'  
WITH STOPATMARK = 'lsn:12000000050000037'
```

<https://learn.microsoft.com/en-us/sql/t-sql/statements/restore-statements-arguments-transact-sql?view=sql-server-ver16>

Alta disponibilidad (HA)

- Habilita una base de datos a **mantener una réplica**.
- La réplica se mantiene **inactiva hasta que se le necesita** para un failover.
 - La operación de failover **invierte los roles** de las DB de la réplica.
- Pueden operar en entornos híbridos.
- Las operaciones de lectura sobre la réplica están restringidas al **licenciamiento**.
- Podemos usar una réplica en modo read only (por default no permite leer) para operaciones ETL y de reporting.
 - Es posible redirigir las conexiones ReadOnly a una réplica secundaria incluso aunque intenten realizarse a la primaria.
 - Se puede utilizar la réplica para backups.

<https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/basic-availability-groups-always-on-availability-groups?view=sql-server-ver16>

Alta disponibilidad (HA)

Puede usar el modo ***synchronous-commit*** o ***asynchronous-commit***.

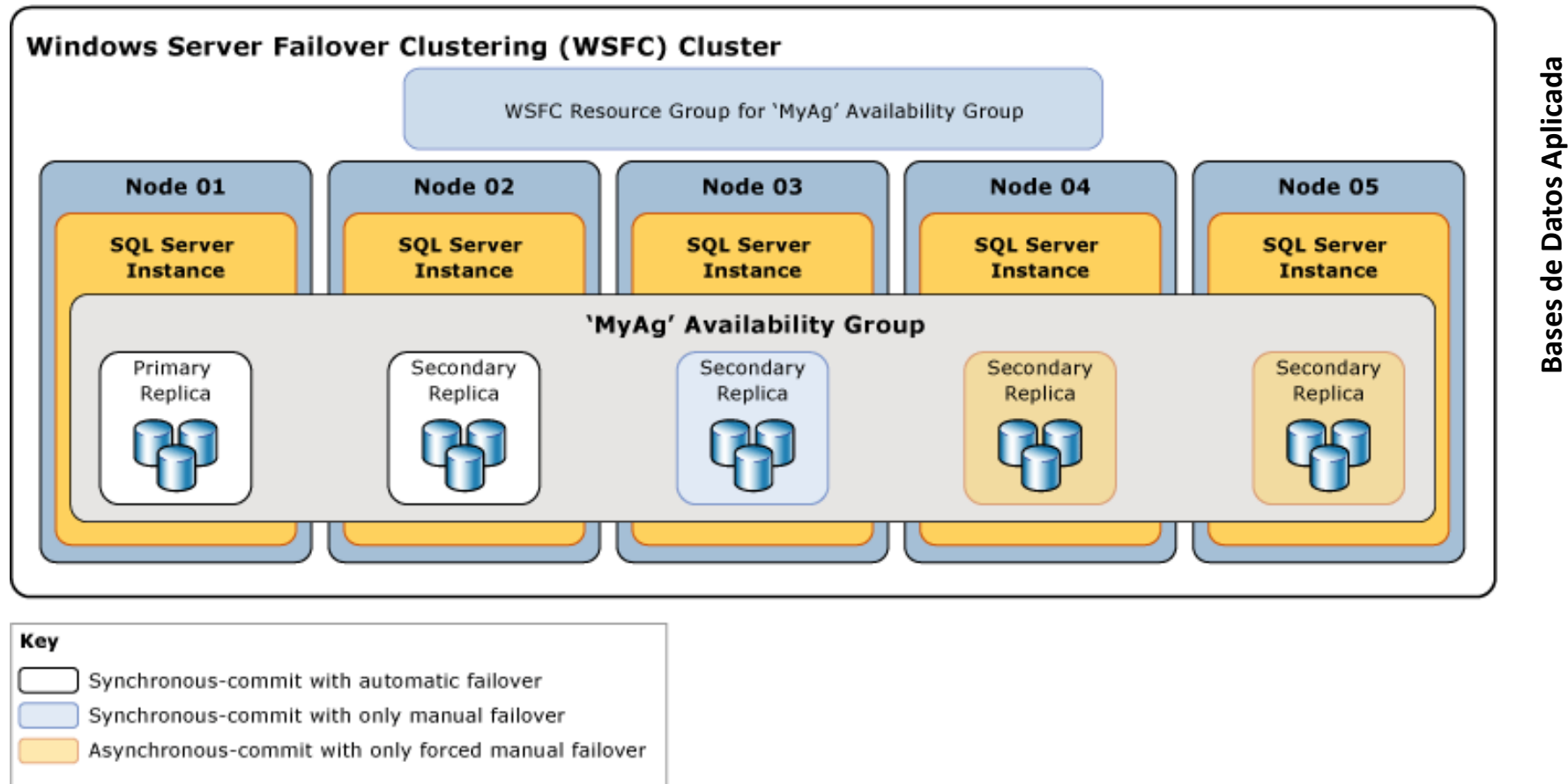
- ASYNCHRONOUS-COMMIT (AC) en réplica primaria: no espera a la réplica secundaria para escribir el log de transacciones. Ídem si una réplica usa AC.
- SYNCHRONOUS-COMMIT (SC) en ambos (primario y secundario) la réplica primaria espera que la réplica secundaria confirme que escribió el log.
- Si se excede el timeout en una réplica secundaria la réplica primaria pasa a modo AC y tan pronto se restablece la comunicación retoman el modo SC.

El failover automático solo puede producirse entre nodos sincrónicos (SC). Los nodos AC se pueden utilizar en failover manual.

<https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/basic-availability-groups-always-on-availability-groups?view=sql-server-ver16>

Alta disponibilidad (HA)

Pueden utilizarse varios nodos en un clúster para asegurar el respaldo y la disponibilidad.



<https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/basic-availability-groups-always-on-availability-groups?view=sql-server-ver16>

Alta disponibilidad (HA)

```
CREATE AVAILABILITY GROUP [BasicAG]
WITH (AUTOMATED_BACKUP_PREFERENCE = PRIMARY, BASIC,
DB_FAILOVER = OFF, DTC_SUPPORT = NONE,
REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT = 0)
FOR DATABASE [AdventureWorks]
REPLICA ON N'SQLVM1\MSSQLSERVER' WITH (ENDPOINT_URL =
N'TCP://SQLVM1.Contoso.com:5022', FAILOVER_MODE = AUTOMATIC,
AVAILABILITY MODE = SYNCHRONOUS COMMIT,
SEEDING_MODE = AUTOMATIC, SECONDARY_ROLE(ALLOW_CONNECTIONS = NO))
    N'SQLVM2\MSSQLSERVER' WITH (ENDPOINT_URL =
N'TCP://SQLVM2.Contoso.com:5022', FAILOVER_MODE = AUTOMATIC,
AVAILABILITY MODE = SYNCHRONOUS COMMIT,
SEEDING_MODE = AUTOMATIC, SECONDARY_ROLE(ALLOW_CONNECTIONS = NO))
GO
```

Réplicas de la DB

- Técnica utilizada para copiar y sincronizar **datos y objetos** de una DB a otra.
- Las DB pueden estar alojadas en una misma instancia o no.
- Mantienen los datos en un estado consistente.
- Los roles a cumplir son
 - **Distributor**: donde se aloja la DB de distribución.
 - **Publisher**: donde se aloja la DB a replicar (origen).
 - *Distributor y Publisher pueden ser el mismo servidor.*
 - **Suscriber**: donde se aloja la DB destino.
- Se definen **artículos(tablas, vistas, SP)** para determinar los datos y objetos a replicar, pudiendo filtrarse filas y columnas.
 - Las tablas a replicar deben tener una PK.

Réplicas de la DB

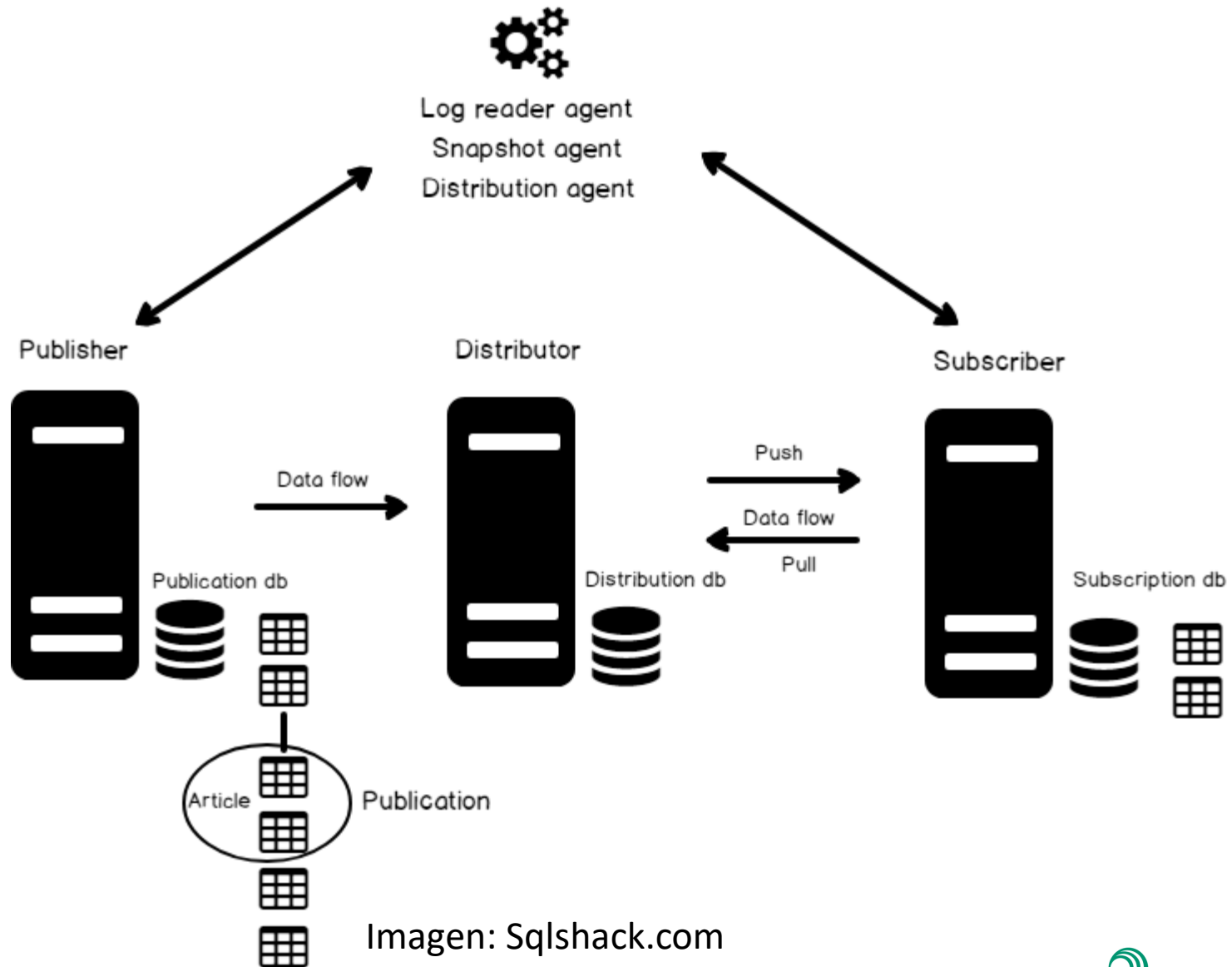


Imagen: Sqlshack.com

Réplicas de la DB

Transaccional: se usa típicamente en escenarios que requieren alta transferencia (throughput), mejorar la escalabilidad y la disponibilidad.

Por ejemplo: data warehousing, data reporting, integración desde múltiples sitios, integración de datos heterogéneos, proceso de lotes offload.

Mezcla (merge): el uso típico es en aplicaciones móviles o aplicaciones distribuidas que pueden tener conflicto de datos.

Por ejemplo: intercambio de datos con usuarios móviles, POS.

Instantánea (snapshot): Provee un estado inicial a las réplicas transaccional y mezcla.

<https://learn.microsoft.com/en-us/sql/relational-databases/replication/sql-server-replication?view=sql-server-ver16>

Log Shipping

Log Shipping (despacho o envío de registro de transacciones) **permite enviar en forma automática los respaldos del log** de transacciones desde una **base primaria** en un servidor primario a una o más **bases secundarias** en una instancia separada.

Los backups **se restauran en cada una** de las bases secundarias.

Puede usarse una tercera instancia como monitor.

Provee así una solución DR (Disaster Recovery). No es failover automático.

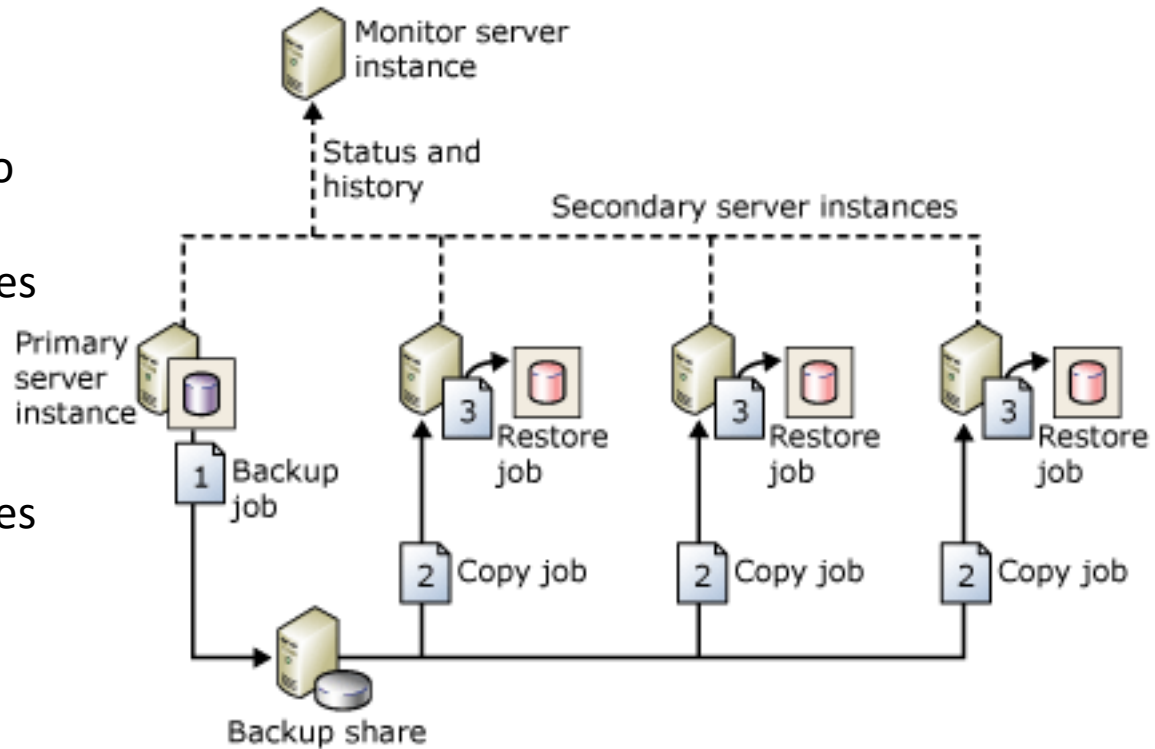
Soporta acceso de solo lectura a las DB secundarias, limitado a los intervalos entre restauraciones.

La latencia del shipping es configurable (podría deshacer una catástrofe).

<https://learn.microsoft.com/en-us/sql/database-engine/log-shipping/about-log-shipping-sql-server?view=sql-server-ver16>

Log Shipping

1. El servidor PRIMARIO ejecuta el backup y se envía a una carpeta como archivo.
2. Cada uno de los servidores secundarios realiza una copia del log en una carpeta local.
3. Cada uno de los servidores secundarios ejecuta una restauración del backup.



Todos envían información de estado e historial
al servidor de monitoreo.

<https://learn.microsoft.com/en-us/sql/database-engine/log-shipping/about-log-shipping-sql-server?view=sql-server-ver16>

Réplicas vs Alta disponibilidad

¿Cuál es la mejor herramienta?

DEPENDE la necesidad y las características del sistema.

Para un HADR completo: Alta disponibilidad

Para combinar dos DB en distintas locaciones: Réplica

¿Qué uso le daría a log shipping?

<https://learn.microsoft.com/en-us/sql/relational-databases/replication/sql-server-replication?view=sql-server-ver16>

Encriptación

Consiste en **cifrar** u **ofuscar** los datos por el uso de una key o password

No resuelve el problema de la protección, pero vuelve inútil los datos a quien se hace de ellos de manera ilícita.

- Se pueden utilizar contraseñas, claves simétricas y asimétricas.
- Podemos cifrar desde UN CAMPO de una tabla, un SP, a toda una DB.
- El manejo de la clave o certificado se vuelve crítico.
- La encriptación transparente (TDE) cifra los archivos de datos y se conoce como encriptación at rest.
- Cifra en tiempo real los datos y registro de transacciones.

<https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption?view=sql-server-ver16>

<https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/encryption-hierarchy?view=sql-server-ver16>

Encriptación

```
USE AdventureWorks2022;
GO
-- Agregamos un campo para los datos cifrados
ALTER TABLE Sales.CreditCard
    ADD TarjetaCreditoCifradaFraseClave VARBINARY(256);
GO
-- Obtenemos la clave de cifrado. Lo cargaríamos desde otra capa.
DECLARE @FraseClaveCargadaPorUsuario NVARCHAR(128);
SET @FraseClaveCargadaPorUsuario = 'QuieroMiPanDanes';

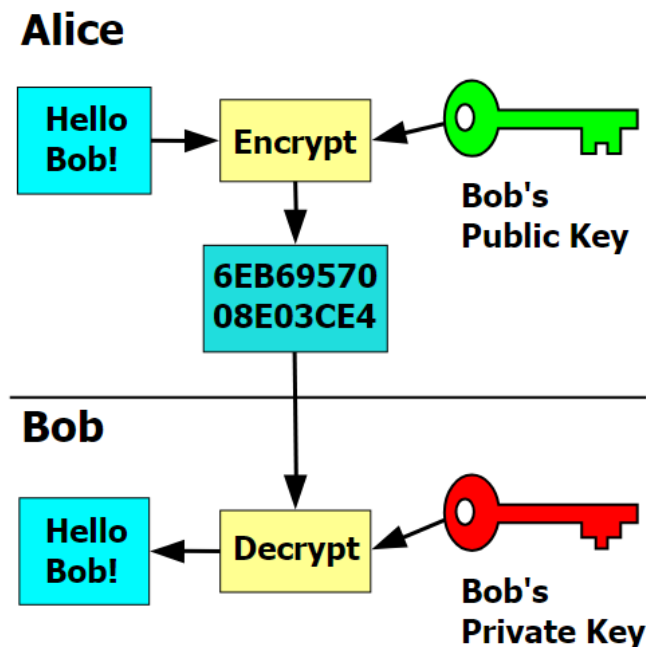
-- Ciframos el campo de la tarjeta de crédito registro 3681.
-- Agrega un hash (el PK IdTarjetaCredito al cifrado)
UPDATE Sales.CreditCard
SET CardNumber_ TarjetaCreditoCifradaFraseClave =
EncryptByPassPhrase(@FraseClaveCargadaPorUsuario
    , NumeroTarjeta, 1, CONVERT(varbinary, IdTarjetaCredito))
WHERE IdTarjetaCredito = '3681';
GO
```

<https://learn.microsoft.com/en-us/sql/relational-databases/replication/sql-server-replication?view=sql-server-ver16>

Encriptación

Es posible cifrar las conexiones con todos los clientes o con algunos específicos.

Requiere la configuración de certificados digitales.



<https://learn.microsoft.com/en-us/sql/database-engine/configure-windows/configure-sql-server-encryption?view=sql-server-ver16>

Lectura recomendada:

<https://learn.microsoft.com/es-es/sql/relational-databases/security/sql-server-security-best-practices?view=sql-server-ver16>

<https://learn.microsoft.com/es-es/sql/relational-databases/security/sql-injection?view=sql-server-ver16>

¿Dudas?



Universidad Nacional
de La Matanza