## 

línea horizontal



Diagrama de Casos de Uso - Veterinario

Entornos de desarrollo

Martes 8 de Abril de 2024

**─**

Hirahi Mejías Delgado

1 DAM-A

# EJERCICIOS A REALIZAR

**Responde a las siguientes preguntas con la mayor profundidad posible,**

**investigando sobre aquellos conceptos que consideres necesarios y aporta**

**ejemplos prácticos para apoyar tu respuesta.**

**1 - ¿Es posible hacer una copia de seguridad cifrada?. En caso afirmativo, explica**

**cómo.**

Sí, es posible hacer una copia de seguridad cifrada. En SQL Server, se puede utilizar la opción de cifrado al realizar una copia de seguridad utilizando la cláusula WITH ENCRYPTION. Esto permite especificar un algoritmo de cifrado y una clave para proteger los datos durante la copia de seguridad. Por ejemplo:

BACKUP DATABASE [NombreBaseDeDatos] TO DISK = 'ruta/del/archivo.bak' WITH ENCRYPTION (ALGORITHM = AES\_256, SERVER CERTIFICATE = certificado);

**2 - Las copias de seguridad que se crean en una versión más reciente de SQL**

**Server, ¿se pueden restaurar en versiones anteriores de SQL Server?**

No, generalmente no es posible restaurar copias de seguridad creadas en versiones más recientes de SQL Server en versiones anteriores de SQL Server. Esto se debe a que las versiones más recientes de SQL Server pueden incluir nuevas características, mejoras de rendimiento o cambios en la estructura de los datos que no son compatibles con versiones anteriores del software.

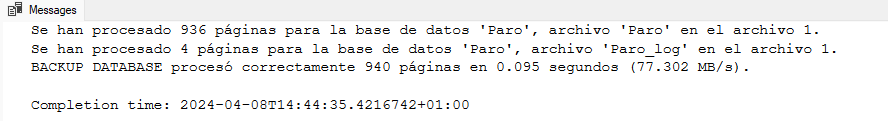
Por ejemplo, si creas una copia de seguridad en SQL Server 2019, es poco probable que puedas restaurarla en una instancia de SQL Server 2014. Sin embargo, hay algunas excepciones a esta regla, especialmente si no se utilizan funciones o características específicas de la versión más reciente que no están disponibles en versiones anteriores.

**3 - ¿Es posible especificar la ubicación donde queremos crear la copia de**

**seguridad?**

Sí, es posible especificar la ubicación donde queremos crear la copia de seguridad utilizando la cláusula TO DISK seguida de la ruta del archivo donde se desea almacenar la copia de seguridad. Por ejemplo:

BACKUP DATABASE [NombreBaseDeDatos] TO DISK = 'ruta/del/archivo.bak';



**4 - ¿Qué opciones se le pueden añadir a la instrucción BACKUP DATABASE?**

**Explícalas**

Al realizar una copia de seguridad de una base de datos en SQL Server, se pueden agregar varias opciones adicionales utilizando la cláusula WITH. Algunas de las opciones comunes incluyen:

**WITH INIT:** Esta opción sobrescribe el contenido de un archivo de copia de seguridad existente.

**WITH FORMAT:** Esta opción formatea el medio de copia de seguridad, eliminando todos los conjuntos de copia de seguridad existentes y reescribiendo el archivo de medios con un conjunto de copia de seguridad completo.

**WITH COPY\_ONLY:** Esta opción realiza una copia de seguridad adicional sin afectar la secuencia de copia de seguridad existente y no afecta la planificación de copias de seguridad regulares.

**WITH CHECKSUM:** Esta opción realiza una verificación de suma de comprobación al escribir cada bloque en el dispositivo de copia de seguridad para garantizar la integridad de la copia de seguridad.

**5 - A la hora de restaurar una copia de seguridad, ¿Cuál es la diferencia entre las**

**siguientes instrucciones?**

- RESTORE WITH RECOVERY

- RESTORE WITH NORECOVERY

- RESTORE WITH STANDBY

Al restaurar una copia de seguridad en SQL Server, estas instrucciones determinan cómo se deja la base de datos después de la restauración. Veamos cada una de ellas:

**RESTORE WITH RECOVERY:** Esta opción recupera la base de datos y la deja lista para su uso. Al utilizar esta opción, SQL Server aplica todas las transacciones de registro presentes en la copia de seguridad y deja la base de datos en un estado de "recuperación", lo que significa que está lista para aceptar nuevas transacciones. Esta opción se utiliza normalmente cuando se está restaurando la copia de seguridad más reciente y no se esperan más copias de seguridad o registros de transacciones adicionales.

**RESTORE WITH NORECOVERY:** Al contrario que la opción anterior, esta opción restaura la base de datos, pero no la recupera completamente. Deja la base de datos en un estado de "no recuperación", lo que significa que aún es posible aplicar más copias de seguridad o registros de transacciones. Esta opción es útil cuando se necesita restaurar múltiples archivos de copia de seguridad o cuando se necesita aplicar registros de transacciones adicionales después de la restauración principal.

**RESTORE WITH STANDBY:** Esta opción restaura la base de datos en un modo de solo lectura, lo que permite que la base de datos esté disponible para consultas de solo lectura mientras se aplican registros de transacciones adicionales. Esto puede ser útil en situaciones donde se necesita acceder a la base de datos para consultar datos mientras se están aplicando cambios adicionales. Es importante tener en cuenta que en este modo, la base de datos no está disponible para cambios o modificaciones hasta que se complete la recuperación.

**6 - Hemos visto copias de seguridad Completas y Diferenciales, pero, ¿en qué**

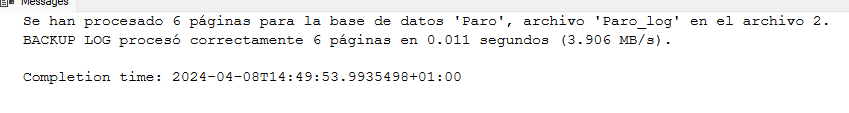
**consisten las copias de seguridad de Registros?**

Las copias de seguridad de registros, también conocidas como copias de seguridad de transacciones, capturan las transacciones de registro activas y los cambios realizados en la base de datos desde la última copia de seguridad de registros o desde la última copia de seguridad completa o diferencial. Estas copias de seguridad permiten restaurar la base de datos a un punto en el tiempo específico, aplicando las transacciones de registro registradas después de la copia de seguridad más reciente. Esto es útil para restaurar la base de datos a un estado consistente en caso de falla del sistema o pérdida de datos.

**7 - Explica qué función cumple la cláusula NO\_TRUNCATE.**

La cláusula NO\_TRUNCATE se utiliza en la instrucción BACKUP LOG para evitar que el registro de transacciones se trunque después de realizar una copia de seguridad de registros. Normalmente, después de una copia de seguridad del registro, SQL Server trunca el registro de transacciones para liberar espacio y permitir que se reutilice. Sin embargo, al utilizar NO\_TRUNCATE, se evita este comportamiento y se conservan las entradas de registro, lo que puede ser útil cuando se necesita conservar el registro de transacciones para recuperaciones puntuales o en entornos de replicación.

**BACKUP LOG Paro TO DISK = 'C:\Users\1dama\Desktop\backupBae' WITH NO\_TRUNCATE;**

****

**8 - Investiga y explica el significado de los siguientes elementos de la Escala de**

**Tiempo.**

Triángulo verde: Indica el inicio de una acción o evento en la línea de tiempo.

Triángulo blanco: Representa el final de una acción o evento en la línea de tiempo.

Cuadrados verdes: Marcan puntos específicos en el tiempo o hitos importantes en la línea de tiempo.

Líneas rojas: Indican interrupciones o problemas en la secuencia temporal en la línea de tiempo.

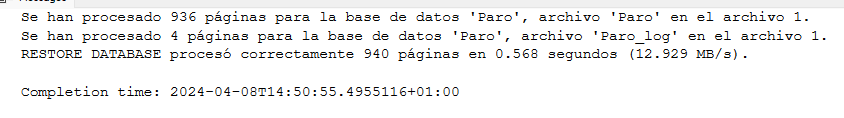
**9 - ¿Podemos restaurar una copia de seguridad desde un fichero que tengamos**

**almacenado en disco?, ¿o solo se puede hacer referenciando directamente a otra**

**de las bases de datos que tenemos en nuestro sistema?**

Sí, es posible restaurar una copia de seguridad desde un archivo almacenado en disco utilizando la cláusula FROM DISK seguida de la ruta del archivo de copia de seguridad. No es necesario hacer referencia directa a otra base de datos en el sistema. Por ejemplo:

RESTORE DATABASE [NombreBaseDeDatos] FROM DISK = 'ruta/del/archivo.bak';



**10 - ¿Para qué sirve el apartado de PROMPT en la ventana de Opciones de la**

**sección ‘Restaurar’?**

El apartado de PROMPT en la ventana de Opciones de la sección 'Restaurar' permite al usuario elegir si desea ser notificado antes de realizar la restauración de la base de datos. Esto puede ser útil para confirmar la acción antes de que se complete la restauración, especialmente en entornos críticos donde se necesitan precauciones adicionales antes de realizar cambios importantes en la base de datos.

