



PROYECTO DE ESTUDIO- INVESTIGACIÓN

Réplicas de base de datos

Carrera

Licenciatura en Sistemas de Información

Asignatura

Sistemas Operativos

Equipo Docente

- Lic. Dario O. Villegas.
- Lic. Iván Sambrana.
- Exp. Juan José Cuzziol.



Integrantes

- | | |
|------------------------------------|---------------|
| • Fernández Rolón Priscila | DNI: 44407402 |
| • González Jeremías Ezequiel | DNI: 44743842 |
| • Guevara González Johan Sebastián | DNI: 96052028 |
| • Mur Mauro Dario | DNI: 43208906 |
| • Sánchez Octavio | DNI: 44622119 |

Fecha de entrega 31 de octubre del 2023



Índice

CAPÍTULO I: INTRODUCCIÓN.....	3
Objetivos.....	4
Objetivos Generales.....	4
Objetivos Específicos.....	4
CAPÍTULO II: MARCO CONCEPTUAL O REFERENCIAL.....	5
Conceptos Fundamentales de Replicación de Bases de Datos.....	5
¿Qué es la Replicación?.....	5
¿Cómo Funciona la Replicación De Base de Datos?.....	5
Tipos De Replicación De Datos.....	6
¿Por qué es Importante la Replicación?.....	8
¿Qué inconvenientes se pueden tener con la Replicación?.....	9
Técnicas de Replicación.....	9
CAPÍTULO III: METODOLOGÍA.....	11
Organización del Equipo y Proceso de Trabajo.....	11
Herramientas, Instrumentos y Procedimientos.....	12
Réplica de la Base de Datos.....	13
CAPÍTULO IV: DESARROLLO DEL TEMA / PRESENTACIÓN DE RESULTADOS.....	14
Creación de cuentas de Windows para replicación.....	14
Crear un recurso compartido para la carpeta de instantáneas y asignar permisos.....	15
Configuración de distribución.....	19
Configuración de la distribución en el publicador.....	19
Establecer los permisos para la base de datos.....	23
Configurar la replicación entre dos servidores conectados completamente (transaccional).....	25
Crear publicación y definir artículos:.....	25
Agente de distribución a la lista de acceso de la publicación (PAL).....	28
Creación de una suscripción a la publicación transaccional.....	28
Creación de la suscripción.....	29
Establecer permisos de base de datos en el suscriptor.....	33
Ver el estado de sincronización de la suscripción.....	34
CAPÍTULO V: INTEGRACIONES.....	37
Manejo de transacciones.....	37
Manejo de Permisos a nivel de usuario.....	40
Backup y Restore:.....	42
Triggers.....	45
Procedimientos y Funciones Almacenadas.....	50
CAPÍTULO VI: CONCLUSIONES.....	55
CAPÍTULO VI: BIBLIOGRAFÍA.....	56



CAPÍTULO I: INTRODUCCIÓN

El presente trabajo práctico tiene como objetivo proporcionar una comprensión sólida acerca de un tema crítico en el mundo de la gestión de datos: las “**Réplicas de Base de Datos**”. Esta introducción tiene como finalidad presentar los fundamentos esenciales del trabajo, estableciendo de manera clara el por qué y para qué se lleva a cabo.

En un mundo cada vez más interconectado y orientado a los datos, la gestión eficaz de la información se ha vuelto fundamental para el funcionamiento de empresas, organizaciones y sistemas. Tener en cuenta que también es una estrategia clave para la administración de información en un entorno tecnológico en constante evolución.

El problema fundamental que abordaremos en este informe es la creciente necesidad de garantizar la integridad, disponibilidad y escalabilidad de las bases de datos en un entorno donde los datos se han convertido en el recurso más valioso. A medida que las organizaciones se enfrentan a la gestión de volúmenes de datos en constante crecimiento, se hace evidente la necesidad de afrontar este desafío de manera efectiva. En este contexto, las réplicas de bases de datos se presentan como una solución potencial para mejorar el rendimiento, la tolerancia a fallos y la disponibilidad de los sistemas de información. Sin embargo, su implementación y gestión eficiente plantean desafíos significativos que requieren una atención especializada.



Objetivos

Objetivos Generales

Nuestro objetivo general es explorar a fondo el concepto de réplicas de base de datos, investigar sus ventajas y desafíos, y proporcionar una guía integral sobre su implementación efectiva. Aspiramos a brindar una comprensión sólida y clara de cómo las réplicas de bases de datos pueden contribuir a la gestión de datos de manera eficiente y confiable en un entorno laboral y tecnológico en constante evolución.

Objetivos Específicos

- Identificar y analizar las ventajas estratégicas y operativas de utilizar réplicas de base de datos en entornos laborales.
- Explorar y evaluar las tecnologías y enfoques más comunes para la implementación de réplicas de bases de datos, incluyendo replicación sincrónica y asincrónica, y sus aplicaciones en diferentes contextos.
- Ofrecer pautas detalladas para la configuración, mantenimiento y monitoreo exitosos de réplicas de bases de datos, destacando mejores prácticas y posibles desafíos a superar.
- Evaluar casos de estudio y escenarios reales donde las réplicas de bases de datos han demostrado ser beneficiosas, ilustrando su impacto positivo en la eficiencia y la continuidad de los sistemas de información.
- Proporcionar recomendaciones prácticas para abordar desafíos y mitigar riesgos asociados con la implementación de réplicas de bases de datos, incluyendo cuestiones de seguridad, rendimiento y mantenimiento.



CAPÍTULO II: MARCO CONCEPTUAL O REFERENCIAL

En este capítulo, se presentará el marco conceptual de nuestro informe, es decir las bases para comprender la replicación de bases de datos en SQL Server. Se abordarán conceptos fundamentales, se destacó la importancia de establecer un marco conceptual sólido en este contexto.

Conceptos Fundamentales de Replicación de Bases de Datos

¿Qué es la Replicación?

La replicación de bases de datos se describe como el proceso de transferir datos desde una base de datos de origen a una o varias bases de datos de destino. Este proceso asegura que los datos se mantengan consistentes en todas las ubicaciones, permitiendo que los usuarios accedan a información actualizada y sincronizada, ya sea en tiempo real o como parte de una operación programada por lotes. En última instancia, la replicación de bases de datos crea una red distribuida de almacenamiento de datos que garantiza una mayor disponibilidad y acceso eficiente a datos cruciales en diversas ubicaciones

¿Cómo Funciona la Replicación De Base de Datos?

La replicación de bases de datos puede ser un proceso único o continuo que implica a todas las fuentes de datos de una organización. Se utiliza un sistema de gestión de bases de datos distribuidas (DBMS) para distribuir los datos a todas las fuentes pertinentes.

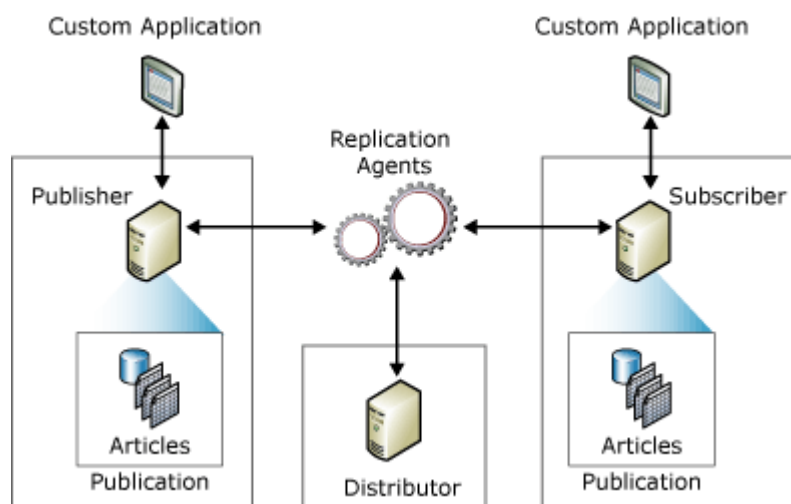
Cuando se realizan cambios, adiciones o eliminaciones en la base de datos de origen, se garantiza la sincronización automática con otras bases de datos de destino, siempre que esos cambios sean necesarios. Este proceso se rige por el paradigma convencional de software Editor-Suscriptor, donde intervienen uno o varios "editores" y "suscriptores".

El "publicador" se refiere a la base de datos de origen donde se originan los cambios, mientras que el "suscriptor" representa un sistema en el cual se replican los cambios. Las modificaciones efectuadas en un sistema "publicador" se replican en las bases de datos



"suscriptoras". Los usuarios también tienen la capacidad de efectuar cambios en las bases de datos de los suscriptores, que luego se propagan a la base de datos del editor. Este enfoque distribuye las actualizaciones a todos los demás miembros de la red si el sistema admite una replicación bidireccional.

Además, la mayoría de los suscriptores están conectados de manera permanente al editor, lo que permite que las actualizaciones se efectúen automáticamente sin necesidad de intervención manual. Estas actualizaciones pueden ocurrir en lotes a intervalos regulares o pueden aplicarse en tiempo real, según las necesidades y configuraciones específicas.



Tipos De Replicación De Datos

Existen diversos tipos de Replicación los cuales procederemos a nombrar, pero en el que nosotros nos centraremos será en la Replicación Transaccional.

- **Replicación de Tabla Completa**

Crea una copia completa de la base de datos de origen en el almacenamiento de destino, incluyendo filas nuevas, modificadas y existentes.

Requiere una alta capacidad de procesamiento y ancho de banda de red, lo que puede resultar en sobrecarga de la red y retrasos en la replicación, especialmente con grandes volúmenes de datos.



- **Replicación de Instantáneas**

Se utiliza una "instantánea" de la base de datos de origen en un momento específico para replicar datos en la base de datos de destino.

No tiene en cuenta los cambios en los datos (nuevos, actualizados o eliminados) y es preferible cuando los cambios de datos son escasos.

Es más rápida que la replicación de tabla completa pero no rastrea datos eliminados.

- **Replicación por Fusión**

Permite la modificación de objetos y datos en ambas bases de datos (editor y suscriptor), lo que puede dar lugar a conflictos de datos relacionados con versiones.

Los agentes de fusión resuelven conflictos siguiendo un proceso predefinido.

- **Replicación Incremental Basada en Claves**

Este método verifica las claves o índices en busca de cambios (borrados, nuevos o actualizados) en una base de datos y replica solo las claves necesarias para reflejar los cambios desde la última actualización.

Es rápido, pero no admite eliminaciones permanentes.

- **Replicación Incremental Basada en Registros**

Réplica datos según el archivo de registro binario de la base de datos de origen, que contiene información sobre cambios, como actualizaciones, inserciones y eliminaciones.

Es eficaz, especialmente para bases de datos estáticas, y es compatible con muchos proveedores de bases de datos.



- **Replicación transaccional**

Es un método de replicación de bases de datos, en el que nos centraremos, y que el mismo se enfoca en mantener la consistencia de los datos entre una base de datos de origen y una o varias bases de datos de destino. Su funcionamiento se basa en el seguimiento de transacciones individuales realizadas en la base de datos de origen. Cuando ocurre una novedad en los datos de origen, ya sea una inserción, actualización o eliminación, la replicación transaccional captura y registra esta transacción.

La característica distintiva de la replicación transaccional es que se asegura de que todas las transacciones realizadas en la base de datos de origen se repliquen en tiempo real o de manera programada en las bases de datos de destino. Esto significa que los datos en las bases de destino reflejan de manera precisa y casi inmediata los cambios que ocurren en la base de datos de origen.

Es eficaz en escenarios donde la integridad y la consistencia de los datos son críticas, como en aplicaciones financieras, sistemas de reservas o aplicaciones médicas. Sin embargo, la replicación transaccional se utiliza principalmente para operaciones de lectura, lo que significa que está diseñada para garantizar que los datos sean accesibles y precisos para los usuarios finales. No está destinada a operaciones de creación, borrado o actualización de datos en las bases de destino.

¿Por qué es Importante la Replicación?

- Fiabilidad y disponibilidad de los datos.
- Recuperación en caso de desastre.
- Rendimiento del servidor.
- Mejor rendimiento de la red.
- Mejora del rendimiento del sistema de pruebas.



¿Qué inconvenientes se pueden tener con la Replicación?

- Costes más elevados.
- Limitaciones de tiempo.
- Ancho de banda.
- Datos incoherentes.

Técnicas de Replicación

Antes de comenzar a hablar de las técnicas como tal, es importante resaltar la diferencia que hay entre los Tipos de Replicaciones, respecto a las Técnicas, si bien la diferencia es sutil puede, prestar a confusión.

- Tipos de Replicación

Son las Categorías generales o enfoques de cómo se realiza la replicación de bases de datos en función de sus características y objetivos. Estos tipos suelen definirse en función de la forma en que se copian y sincronizan los datos, así como de las necesidades específicas de un sistema. Cada tipo de replicación tiene sus propias ventajas y desventajas, y se elige en función de los requisitos y las prioridades del sistema.

- Técnicas de Replicación

Son los métodos y estrategias específicas utilizados para implementar un tipo de replicación en particular. Estas técnicas abordan los detalles de cómo se capturan, transmiten y aplican los cambios de datos entre la base de datos de origen y las bases de datos de destino. Las técnicas pueden variar según el tipo de replicación elegido.



Entre las técnicas a destacar podemos mencionar la **Replicación Completa de Bases de Datos**. Es una técnica que implica la duplicación de toda la base de datos en distintos hosts, lo que proporciona un alto grado de redundancia y disponibilidad de datos. Esta técnica es valiosa para empresas globales que necesitan que los usuarios de diferentes regiones accedan a los mismos datos a velocidades óptimas. En caso de una falla en el servidor local, los usuarios pueden recurrir a copias de seguridad en otros servidores globales.

Sin embargo, es importante tener en cuenta que esta técnica puede ser lenta para realizar actualizaciones, ya que implica la replicación de grandes volúmenes de datos. Además, mantener la consistencia en la ubicación de cada archivo puede ser un desafío, especialmente cuando los datos experimentan cambios constantes.

Replicación Parcial de Bases de Datos

La replicación parcial de bases de datos consiste en dividir los datos de una base de datos en segmentos lógicos y almacenarlos en ubicaciones diferentes según su relevancia y uso. Este enfoque es particularmente beneficioso para profesionales como expertos en seguros, asesores financieros y vendedores, quienes pueden llevar consigo subconjuntos de datos relevantes en dispositivos o portátiles y sincronizarlos periódicamente con un servidor central.

Además, para analistas y empresas, esta técnica puede ser más rentable, ya que permite mantener los datos geográficamente cercanos a los consumidores, al tiempo que se conserva una copia completa de los datos en la sede central para análisis de alto nivel.



CAPÍTULO III: METODOLOGÍA

Organización del Equipo y Proceso de Trabajo

La organización del equipo desempeñó un papel crucial en el desarrollo exitoso del proyecto, que tenía como objetivo principal abordar el tema de la "Réplica de Base de Datos". A continuación, se describe el proceso de organización y colaboración empleado por el equipo:

- 1. Organización del Grupo de Trabajo:** En una primera reunión, se llevó a cabo una discusión para determinar de qué manera íbamos a trabajar, con que herramientas, etc. El tema que nos tocó desarrollar fue "Réplica de Base de Datos". Se determinó que el proyecto se centraría en la implementación de una base de datos con el tipo de replicación transaccional.
- 2. Búsqueda de Información:** Una vez definido el tema, se asignó a cada miembro del equipo la tarea de buscar información relevante sobre la réplica de bases de datos. A través de comunicación vía WhatsApp, Discord, nos compartimos enlaces y datos que fue recolectando cada uno.
- 3. Reuniones de Sincronización:** Se organizaron reuniones periódicas para reunir toda la información recopilada por cada miembro del equipo. Durante estas reuniones, se compartió la información mencionada.
- 4. Planificación en Trello:** Para una gestión eficiente de las tareas, se utilizó la plataforma Trello. Se crearon tableros en Trello para dividir y asignar las responsabilidades a cada integrante del equipo. Esto permitió un seguimiento claro de las actividades en curso.
- 5. Colaboración en Google Drive:** Se estableció un espacio de colaboración en Google Drive donde se almacenaron y compartieron las actualizaciones y revisiones



del informe en desarrollo. Esto aseguró que todos los miembros del equipo pudieran acceder a los documentos y realizar contribuciones de manera efectiva.

6. **Reuniones de Avance:** Se llevaron a cabo llamadas regulares dos veces por semana para revisar el progreso individual de cada miembro y discutir los avances del proyecto en su conjunto.
7. **Seguimiento de Tutoriales:** Se programó una llamada grupal para poder seguir tutoriales proporcionados por Microsoft SQL Server para adquirir el conocimiento necesario sobre la réplica de bases de datos.
8. **Unificación del Informe:** Finalmente, se procedió a unificar todas las partes del informe siguiendo las directrices establecidas durante las reuniones de avance y la revisión conjunta de la información recopilada.

Este proceso de organización y trabajo en equipo garantizó la eficiencia en la realización del proyecto, permitiendo una colaboración efectiva y una presentación coherente de los resultados.

Herramientas, Instrumentos y Procedimientos

En esta sección, se detalla la elección de herramientas y procedimientos utilizados para la recopilación y el tratamiento de la información requerida para el proyecto de estudio

1. **Recolección de Datos:** Los datos necesarios para el proyecto se obtuvieron principalmente a través de fuentes en línea. Este enfoque incluyó la revisión de documentación técnica, tutoriales, y recursos disponibles en la web.
2. **Herramientas Informáticas y Plataformas:** Para la organización y la colaboración en el proyecto, se emplearon herramientas informáticas como Trello, Google Drive y GitHub. Estas plataformas fueron esenciales para mantener un flujo de trabajo eficiente y garantizar la accesibilidad a los recursos compartidos.



3. **Comunicación Interpersonal:** Discord y WhatsApp se utilizaron para la comunicación y coordinación entre los miembros del equipo. La programación de reuniones regulares permitió compartir información de manera efectiva y garantizar la cohesión del grupo de trabajo.
4. **Gestión de la Base de Datos:** Para la implementación de la réplica de base de datos, se optó por utilizar SQL Server Management Developer Edition. Esta elección se basó en las capacidades y características que esta herramienta proporcionó, que resultaron fundamentales para el desarrollo exitoso del proyecto.

SQL Server Management Developer Edition permitió:

- **Diseño de la Base de Datos:** Facilitó el diseño y modelado de la base de datos, permitiendo definir la estructura y las relaciones necesarias.
- **Administración de la Réplica:** Ofreció las herramientas necesarias para configurar y gestionar la réplica de datos entre los servidores, lo que fue esencial para el objetivo del proyecto.
- **Análisis de Datos:** Permitió la realización de análisis y consultas sobre los datos replicados, lo que contribuyó a la evaluación de la efectividad de la réplica y a la obtención de conclusiones relevantes.

Réplica de la Base de Datos

La réplica de la base de datos fue un componente fundamental en la metodología aplicada para el proyecto de estudio. Para llevar a cabo este proceso, se siguió un conjunto de tutoriales proporcionados por Microsoft SQL Server, los cuales ofrecieron una guía detallada y precisa sobre la preparación del servidor y la replicación de datos.



CAPÍTULO IV: DESARROLLO DEL TEMA / PRESENTACIÓN DE RESULTADOS

Requisitos previos

Tener instalado SQL Server , SQL Server Management Studio (SSMS) y una base de datos en este caso **base_consortio**.

En el servidor de publicación (**maestro**), instalar SQL Server Developer Edition u otra versión que permita la publicación (**EXPRESS no lo permite**).

En el servidor de suscripción (**esclavo**), instalar cualquier edición de SQL Server Compact.

SQL Server Compact no puede ser un suscriptor de una replicación transaccional.

Creación de cuentas de Windows para replicación

Antes de hacer la replicación una buena práctica y fundamental para la seguridad configurar bien los permisos que tendrán cada rol, cabe aclarar que si el publicador y el suscriptor comparten la misma instancia, no se requieren los pasos que se utilizan para crear las cuentas en el suscriptor pero en este caso están en dos instancias distintas por lo que para ello se crearán cuatro cuentas en windows:

Agente de instantáneas (**Publisher**) que llamaremos **repl_snapshot**

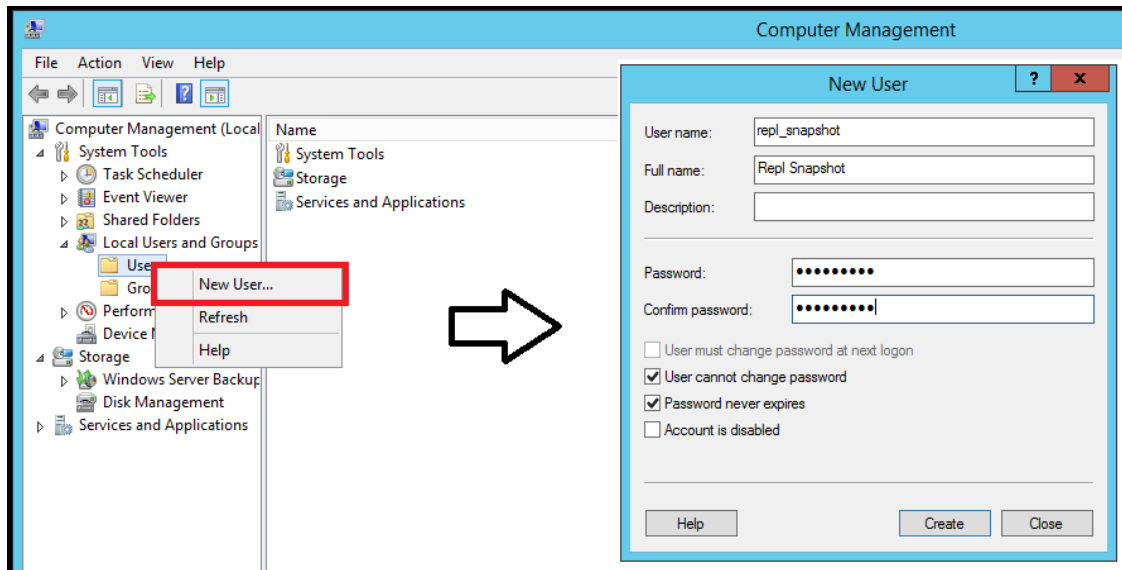
Agente de registro del LOG (**Published**) que llamaremos **repl_logreader**

Agente de distribución (**Publicador y suscriptor**) que llamaremos **repl_distribution**

Agente de mezcla (**Publicador y suscriptor**) que llamaremos **repl_merge**

Creación de cuentas locales de Windows para agentes de replicación en el publicador

Se Debe ir “**Administracion de equipos**”, luego a “**Usuarios y grupos locales**”, luego crea un nuevo usuario, coloque los datos (**contraseña a preferir solo debe recordarla al momento de usarla**) por cada usuario.

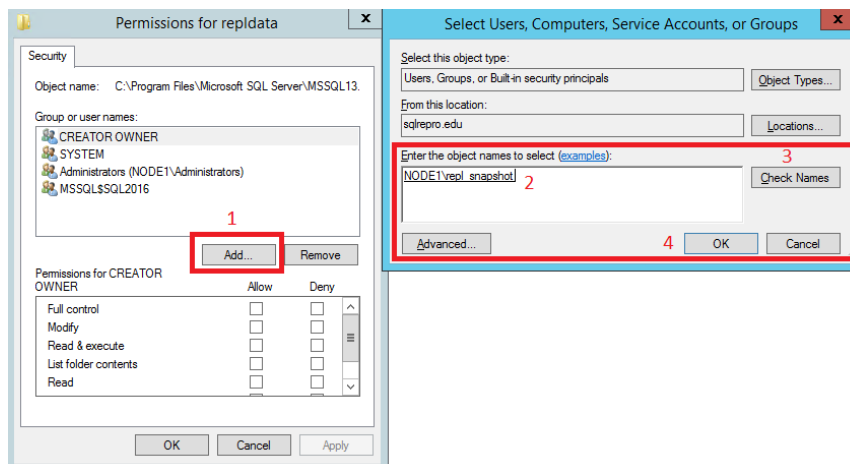
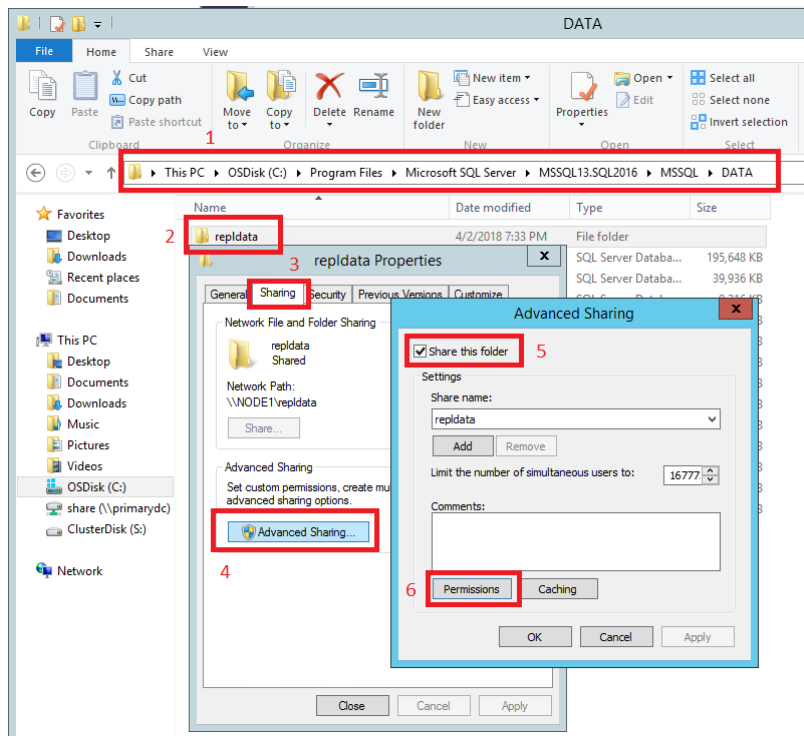


Preparación de la carpeta de instantáneas

En esta sección se va a configurar la carpeta de instantáneas que se utiliza para crear y almacenar la instantánea de publicación.

Crear un recurso compartido para la carpeta de instantáneas y asignar permisos

1. En el Explorador de archivos, vaya a la carpeta de datos SQL Server. La ubicación predeterminada es C:\Archivos de programa\Microsoft SQL Server\MSSQL.X\MSSQL\Data.
2. Cree una carpeta con el nombre **repldata**.
3. Haga clic con el botón derecho en esta carpeta y seleccione Propiedades.
 - En la pestaña Compartir del cuadro de diálogo Propiedades de repldata, seleccione Uso compartido avanzado.
 - En el cuadro de diálogo Uso compartido avanzado, seleccione Compartir esta carpeta y, después, seleccione Permisos.



4. En el cuadro de diálogo Permisos de repldata, seleccione Agregar. En el cuadro de texto Seleccionar usuarios, equipos, cuentas de servicio o grupos, escriba el nombre de la cuenta del Agente de instantáneas creado anteriormente, como <Nombre_De_Equipo_Publicador>repl_snapshot. Seleccione Comprobar nombres y, después, Aceptar. (Repita por cada usuario)

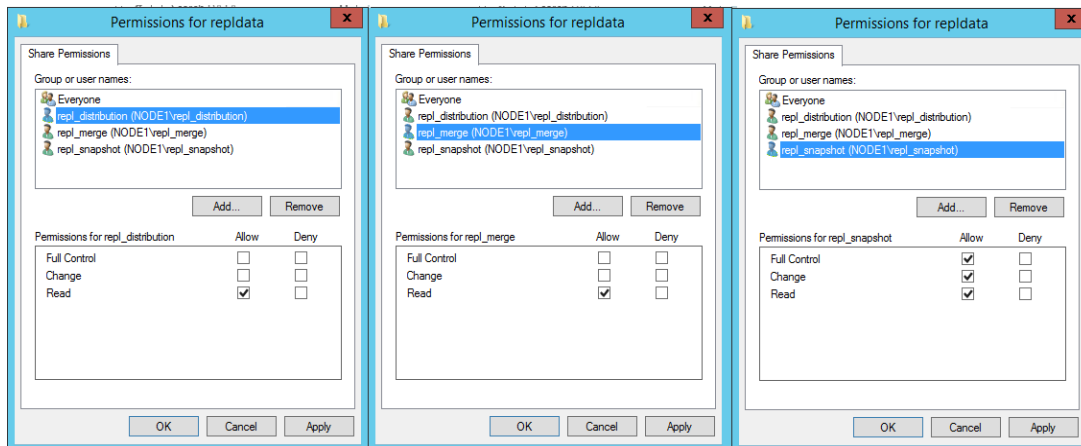


5. Asegúrese que los permisos queden de la siguiente forma:

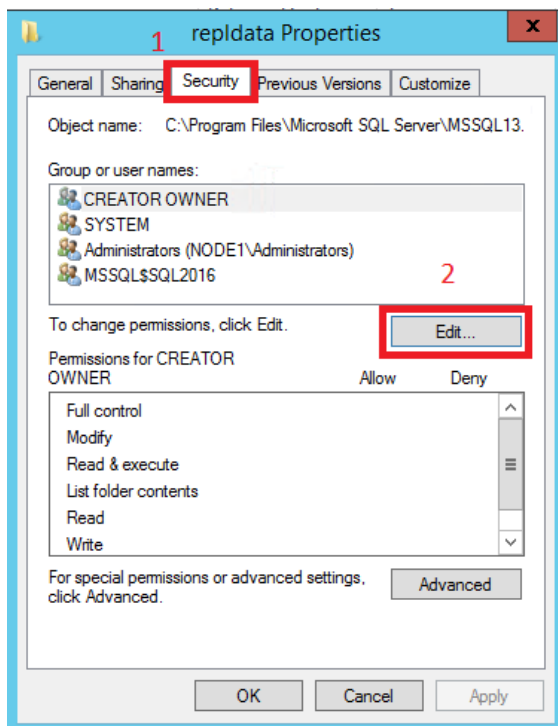
repl_distribution: Lectura

repl_merge: Lectura

repl_snapshot: Control total

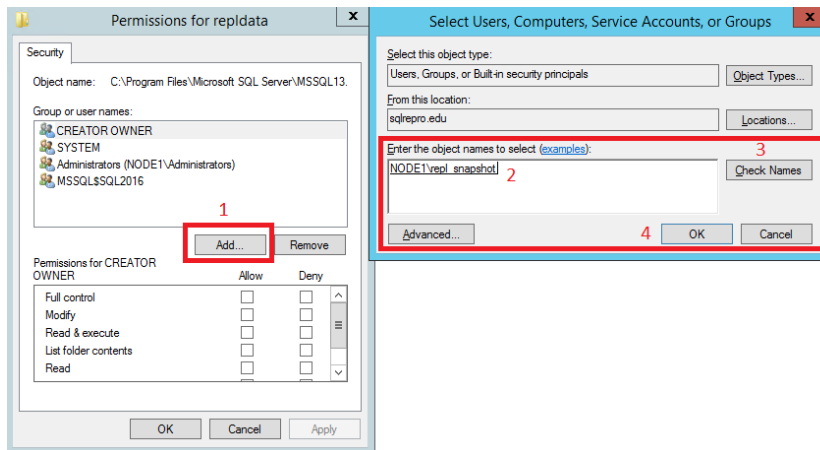


6. Vaya a pestaña seguridad en las propiedades de la carpeta **repl_data**, luego a edición:





- En el cuadro de diálogo Permisos de repldata, seleccione Agregar. En el cuadro de texto Seleccionar usuarios, equipos, cuentas de servicio o grupos, escriba el nombre de la cuenta del Agente de instantáneas creado anteriormente, como **<Nombre_De_Equipo_Publicador>\repl_snapshot**.

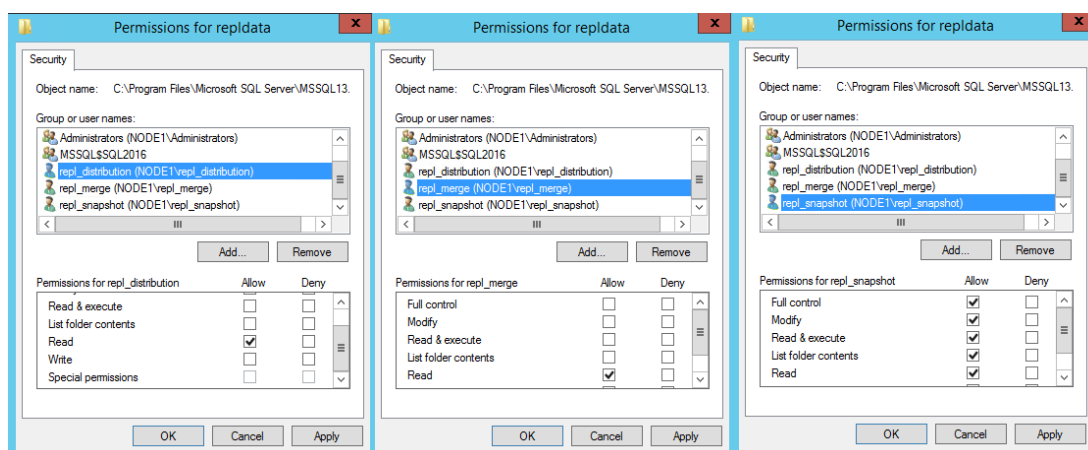


- Repita el paso anterior para agregar permisos para el Agente de distribución, por ejemplo **<Nombre_De_Equipo_Publicador>\repl_distribution** y, para el Agente de mezcla, **<Nombre_De_Equipo_Publicador>\repl_merge**.
- compruebe que se admiten los siguientes permisos:

repl_distribution: Lectura

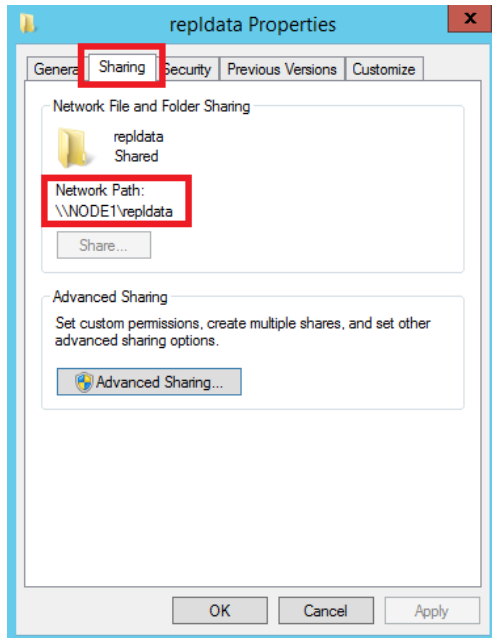
repl_merge: Lectura

repl_snapshot: Control total





10. Vuelva a seleccionar la pestaña Compartir y anote la Ruta de acceso a la red para el recurso compartido. Necesitará esta ruta de acceso más adelante cuando configure la carpeta de instantáneas.



Configuración de distribución

En esta sección se va a configurar la distribución en el publicador y establecer los permisos requeridos en las bases de datos de publicación y distribución. Si ya ha configurado el distribuidor, debe deshabilitar la publicación y distribución antes de iniciar esta sección. No lo haga si debe mantener una topología de replicación existente, especialmente en producción.

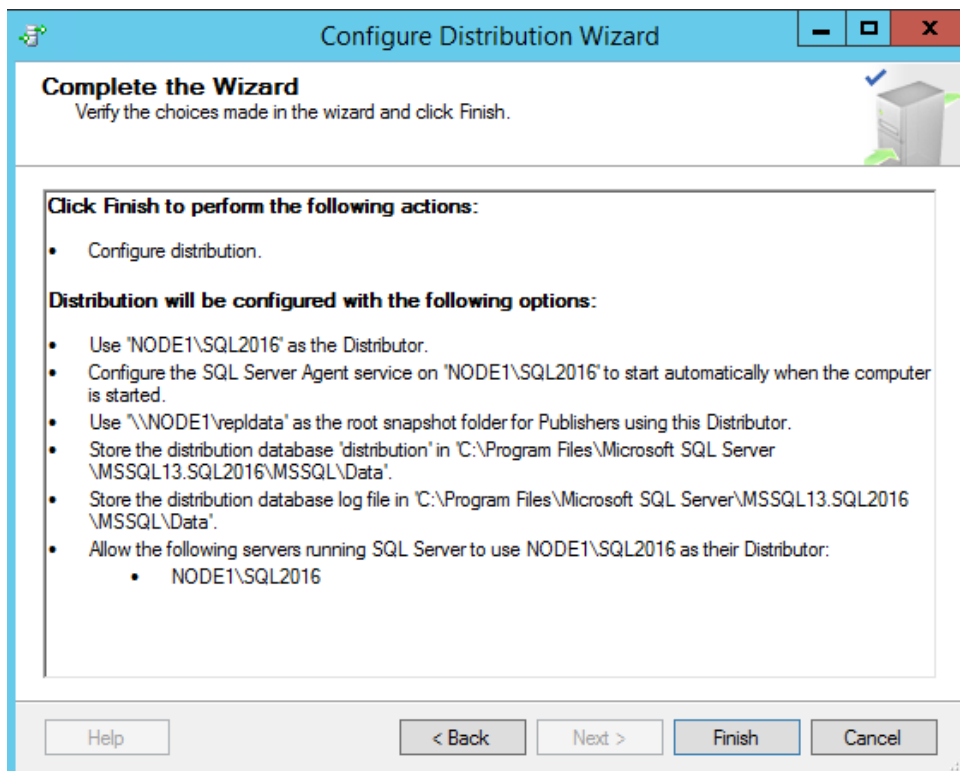
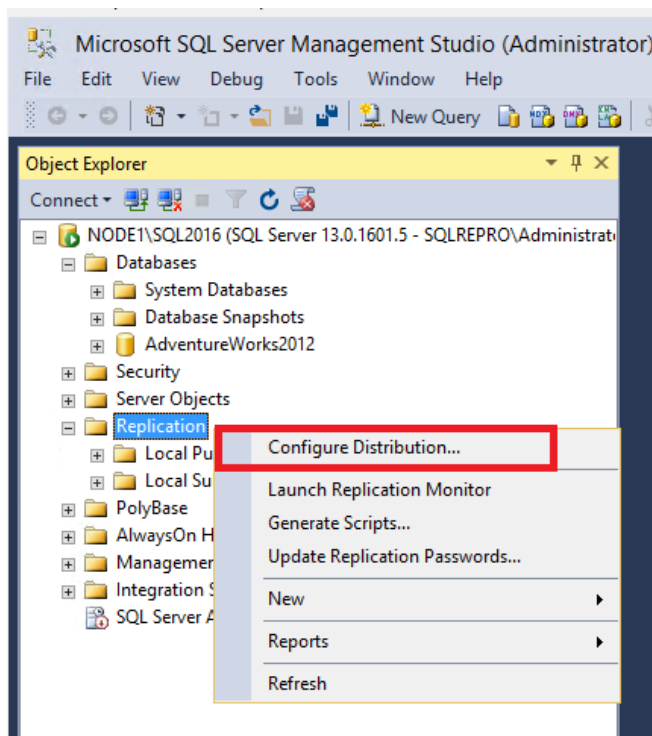
En este tutorial no se contempla la configuración de un publicador con un distribuidor remoto.

Configuración de la distribución en el publicador

1. Conéctese al publicador en SQL Server Management Studio y, a continuación, expanda el nodo de servidor.



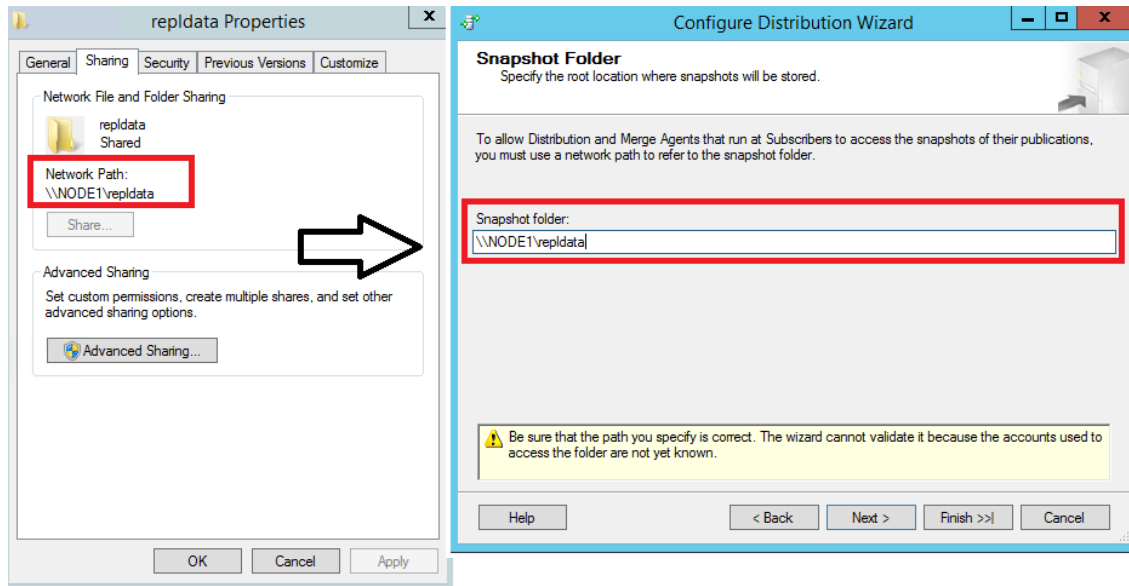
2. Haga clic con el botón derecho en la carpeta Replicación y luego seleccione Configurar distribución:





Nota: no conectarse al servidor como localhost, debe conectarse con el nombre real del servidor. Para más información vaya a la bibliografía.

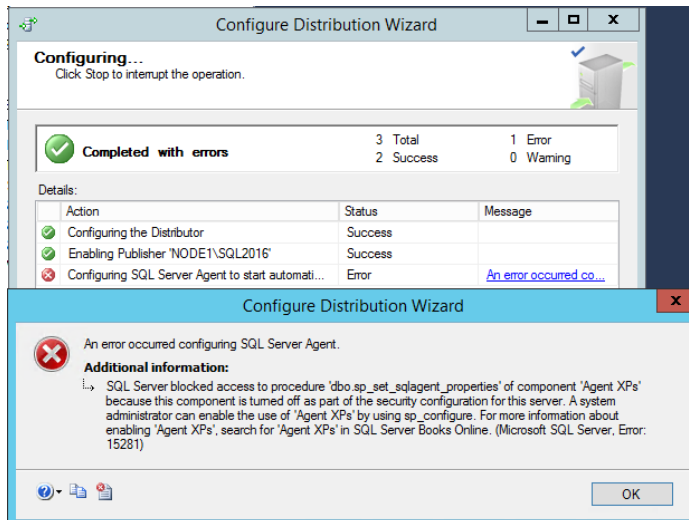
3. En la página Distribuidor, seleccione '<NombreServidor>' actuará como su propio distribuidor; SQL Server creará una base de datos y un registro de distribución.



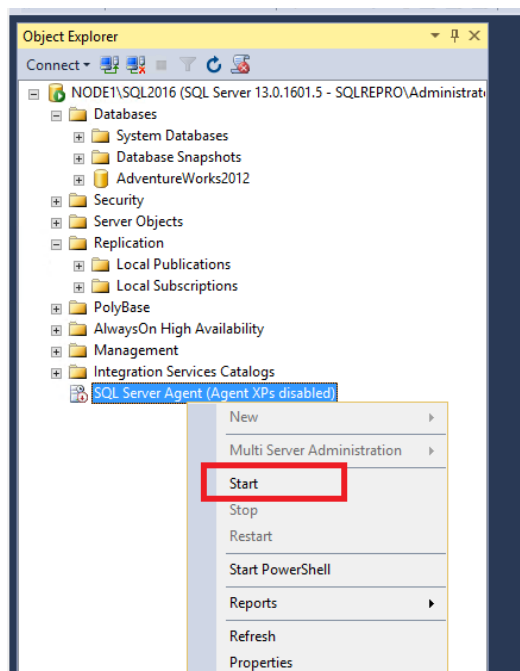
4. Si el Agente de SQL Server no se está ejecutando, en la página Inicio del Agente, seleccione Sí, configurar el servicio del Agente para que se inicie automáticamente. Seleccione Siguiente.
5. Escriba la ruta de acceso '\\<Nombre_De_Equipo_Publicador>\\repldata' en el cuadro de texto Carpeta de instantáneas y, después, seleccione Siguiente. Esta ruta de acceso debe coincidir con lo que vimos anteriormente en Ruta de acceso a la red de la carpeta de propiedades de repldata después de configurar las propiedades del recurso compartido.
6. Acepte los valores predeterminados en las páginas restantes del asistente y finalmente seleccione finalizar.



El siguiente error puede aparecer al configurar el distribuidor. Es una indicación de que la cuenta que se utilizó para iniciar la cuenta del Agente SQL Server no es un administrador del sistema. Necesitará iniciar manualmente el Agente SQL Server, conceder dichos permisos a la cuenta existente o modificar la cuenta que utiliza el Agente SQL Server.



Si su instancia de SQL Server Management Studio se ejecuta con derechos administrativos, puede iniciar el Agente de SQL manualmente desde dentro de SSMS:

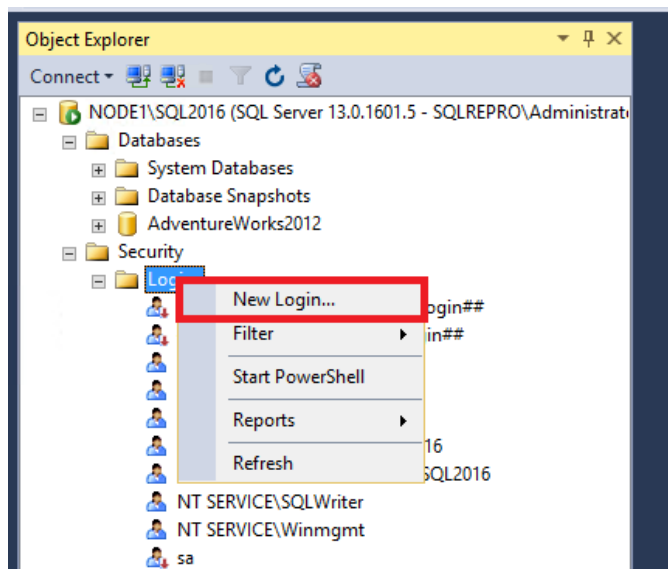




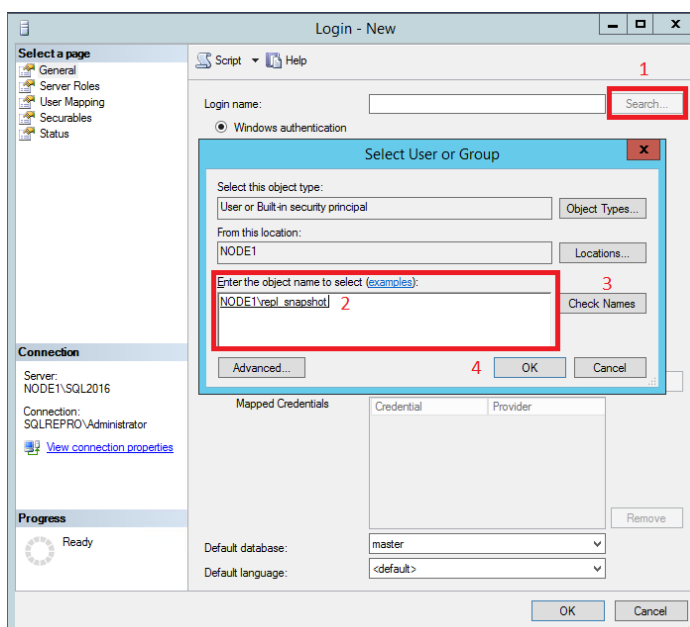
Nota: Si no se inicia el Agente SQL de forma visible, haga clic con el botón derecho en el Agente SQL Server en SSMS y seleccione Actualizar. Si está todavía en estado detenido, inícialo manualmente desde el Administrador de configuración de SQL Server.

Establecer los permisos para la base de datos

1. En SQL Server Management Studio, expanda Seguridad, haga clic con el botón derecho en Inicios de sesión y seleccione Nuevo inicio de sesión.

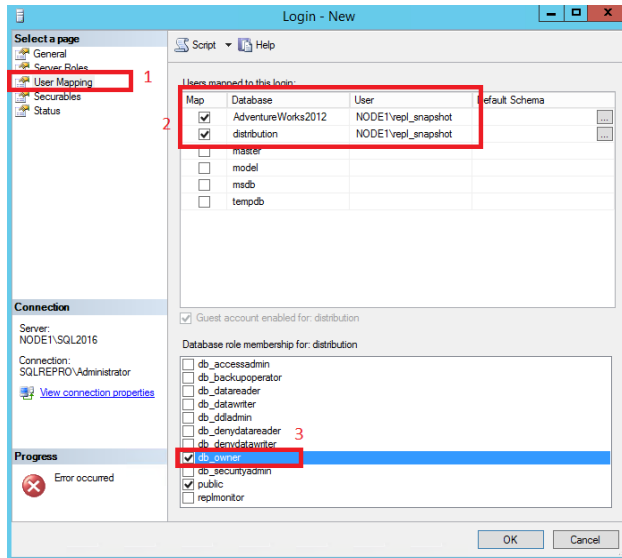


2. En la página General, seleccione Buscar. Escriba <>\repl_snapshot.

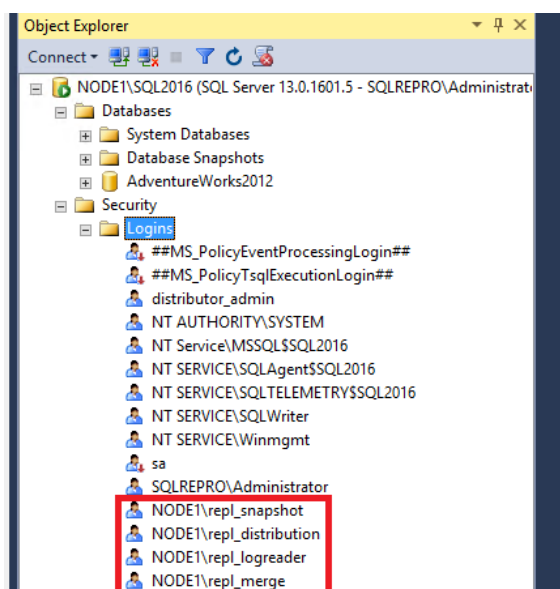




- En la página Asignación de usuarios, en la lista Usuarios asignados a este inicio de sesión, seleccione las bases de datos de distribución y de **base_consortio**. En la lista Pertenencia al rol de la base de datos, seleccione el rol **db_owner**.



- Seleccione Aceptar para crear el inicio de sesión.
- Repita los pasos del 1 al 4 para crear un inicio de sesión para el resto de cuentas locales (repl_distribution, repl_logreader y repl_merge). Estos inicios de sesión también se deben asignar a usuarios que son miembros del rol fijo de base de datos db_owner en las bases de datos distribution y base_consortio.





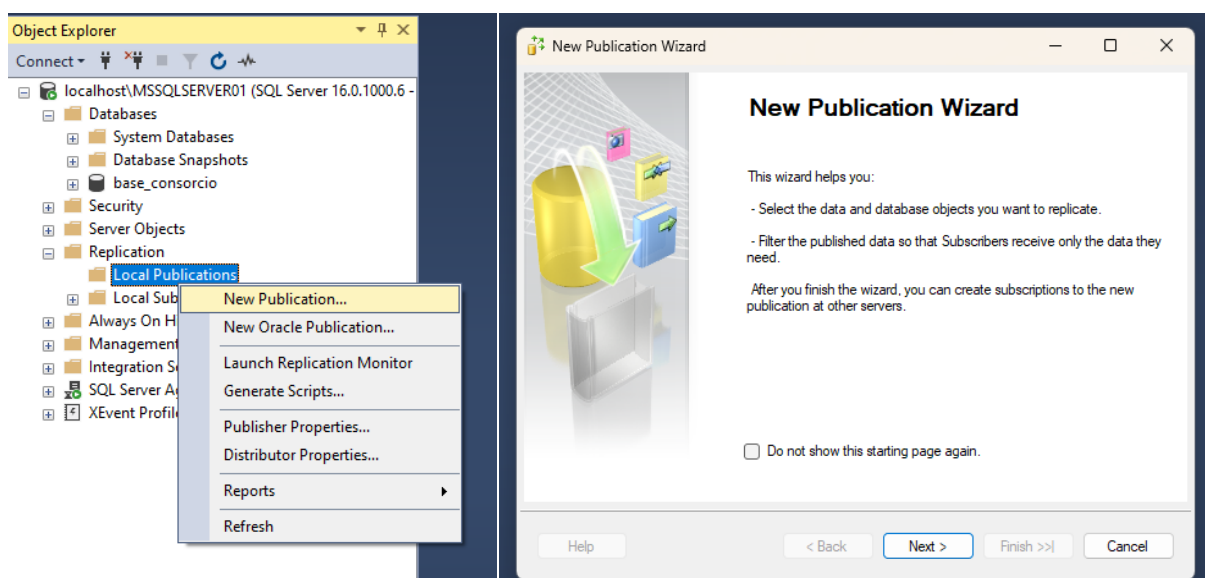
Configurar la replicación entre dos servidores conectados completamente (transaccional)

La replicación transaccional es un proceso crucial en bases de datos, donde los cambios hechos en un servidor de base de datos (llamado servidor principal o emisor) se copian automáticamente en otro servidor (llamado servidor réplica o receptor) para mantener los datos sincronizados y garantizar la disponibilidad y redundancia. Esta forma de replicación asegura que cada transacción realizada en el servidor principal se replique exactamente en el servidor réplica, manteniendo la consistencia de los datos.

Para llevar a cabo el proceso de replicación, se empleará la herramienta SQL Server Management Studio (SSMS) junto con SQL Server como motor de la base de datos. Se utilizará una base de datos de ejemplo llamada "**base_consortio**".

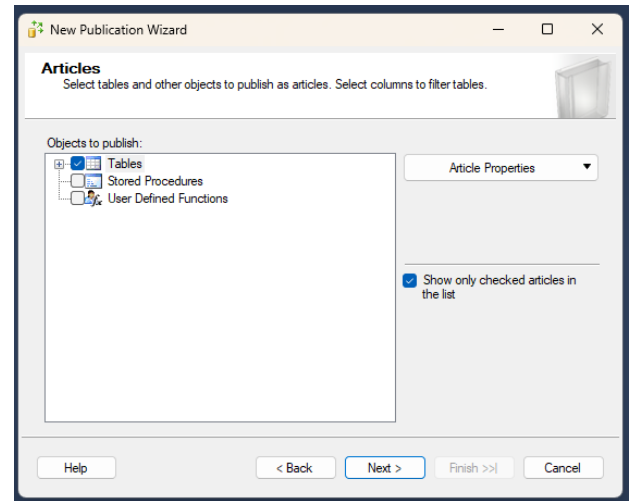
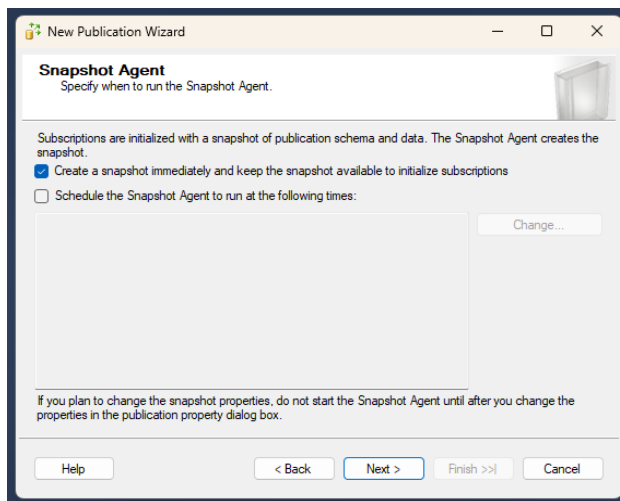
Crear publicación y definir artículos:

Paso 1: Abre SQL Server Management Studio (SSMS) y ve al directorio "**Replication**". Haz clic derecho en "**Local Publications**" y selecciona "**New Publication**". Esto abrirá el Asistente para la creación de la publicación.

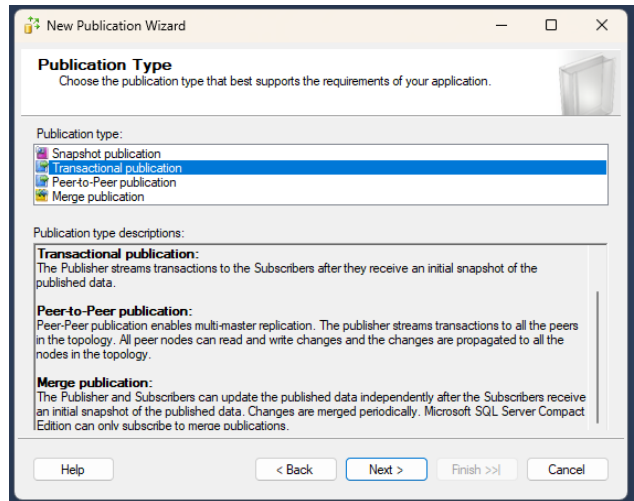
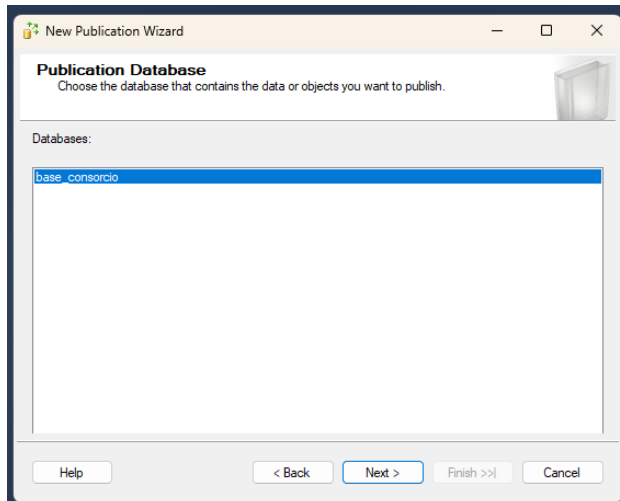




Paso 2: Selecciona la base de datos "**base_consortio**" y elige el tipo de publicación como "**Transactional Publication**". Haz clic en "**Next**".



Paso 3: En la sección de **Articles** se encuentran las tablas y objetos que forman parte de la base de datos, para efectos prácticos seleccionaremos todas las tablas como artículos y daremos **Next**.



Paso 4: Ahora deberemos seleccionar la casilla que nos permitirá crear un Snapshot Agent y daremos **Next**



Paso 5: Para crear el agente seleccionaremos el path del **repl_snapshot** ya creado con anterioridad y agregaremos una contraseña para luego guardar dando en **OK**

Paso 6: Configurar el nombre de la publicación y hacer clic en "**Finish**" para completar el asistente



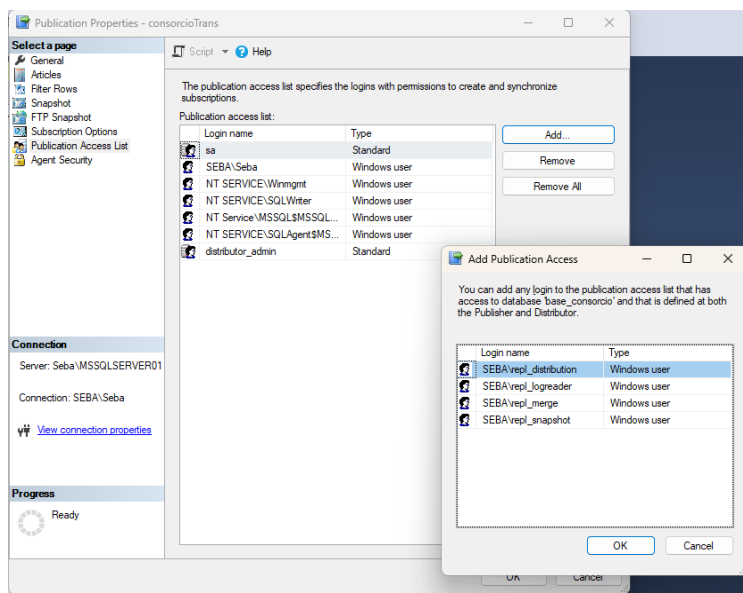
Agente de distribución a la lista de acceso de la publicación (PAL)

En el directorio **Local Publications**, haga clic derecho en **consorcioTrans** y luego seleccione **Properties**. Aparece el cuadro de diálogo Publication Properties.

Ahora haremos 2 sencillos pasos:

Paso 1: Seleccionar la página **Publication Access List** y seleccionar **Add**.

Paso 2: En el cuadro de diálogo **Add Publication Access** daremos permisos a **repl_distribution** y seguidamente **OK**.



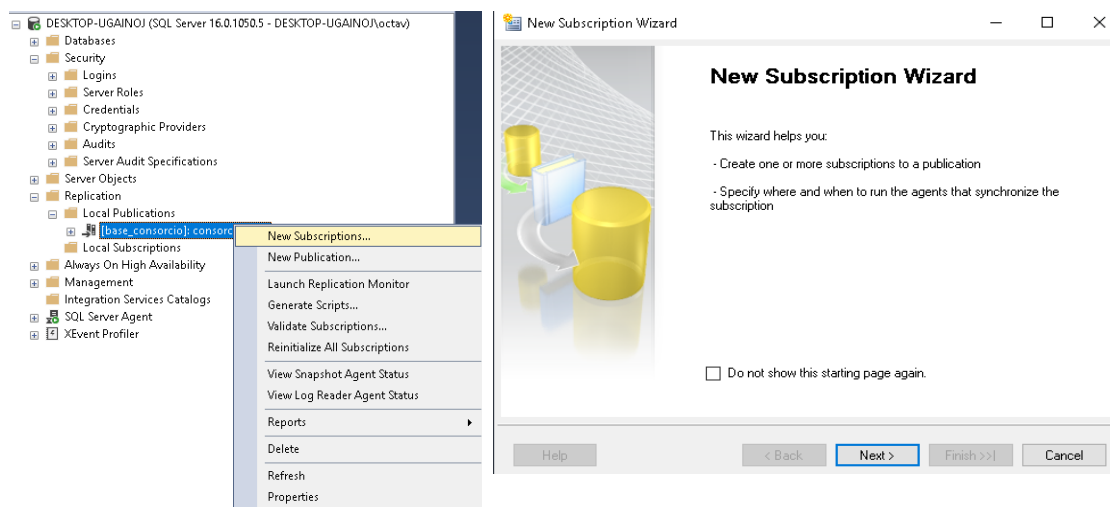
Creación de una suscripción a la publicación transaccional

En esta sección agregamos un suscriptor que sería nuestra base de datos que va a recibir y aplicar los cambios realizados en el publicador (que sería nuestra base de datos principal que contiene los datos originales en este caso "**base_consorcio**") que creamos y configuramos anteriormente en Crear publicaciones y definir artículos para publicar un conjunto de todas las tablas de nuestra **base_consorcio** mediante la replicación transaccional

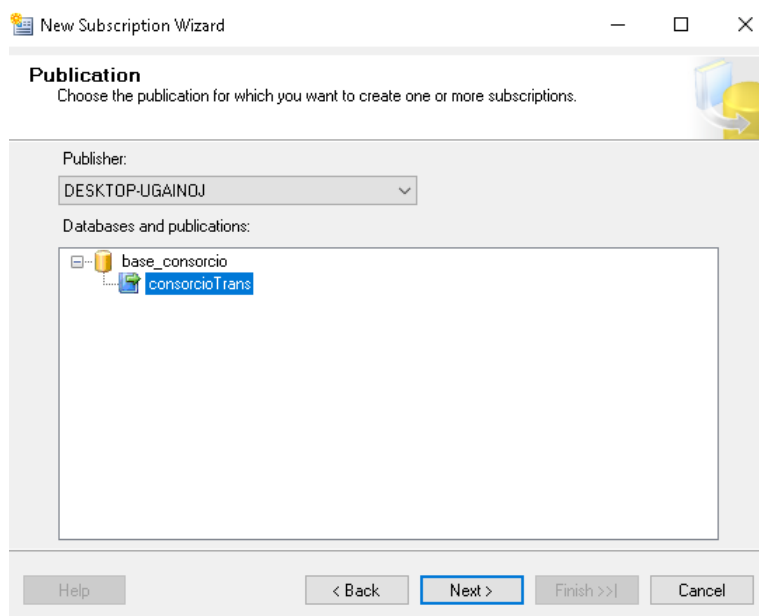


Creación de la suscripción

1. Debemos abrir el SQL Server Management Studio y conectarnos a nuestro servidor que aloja la base de datos publicadora “**base_consortio**” y, a continuación, expanda la carpeta Replicación .
2. En la carpeta Publicaciones locales o Local Publications, haga clic con el botón derecho en la publicación “**[base_consortio]:consorcioTrans**” que creamos anteriormente y, después, seleccione Nuevas suscripciones o new Subscription.

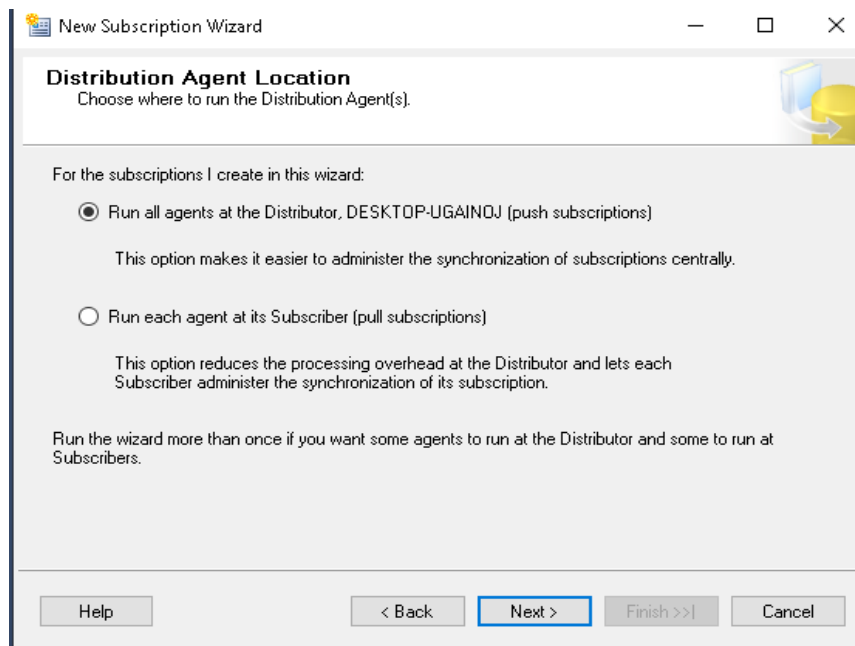


3. En la página Publicación, seleccione **consorcioTrans** (nuestra publicación) y, después, seleccione Siguiente:

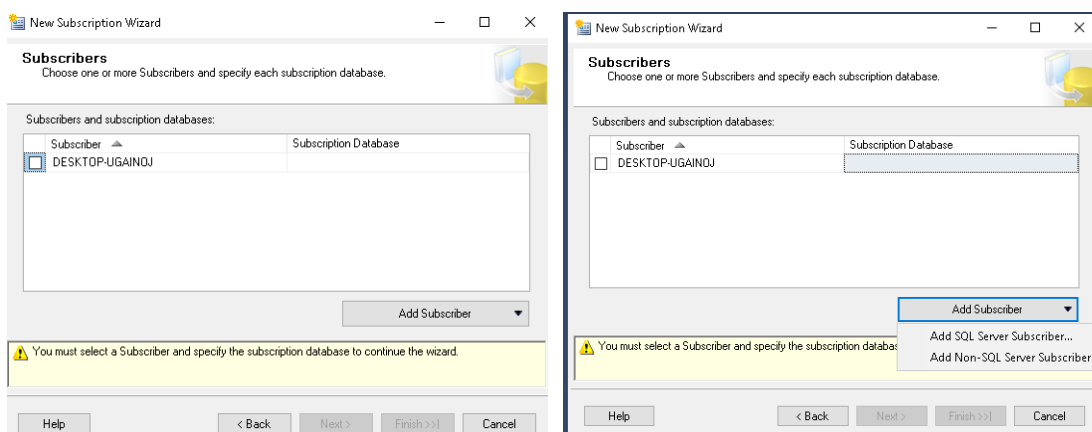


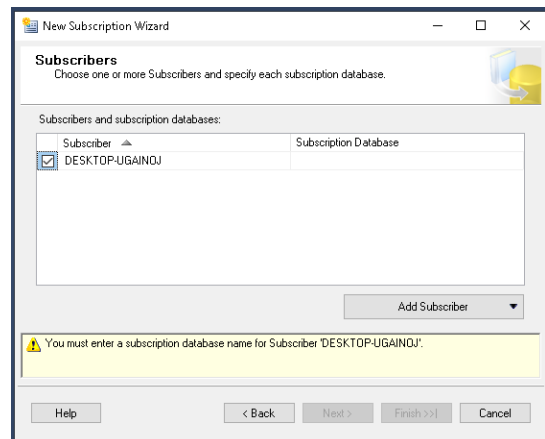
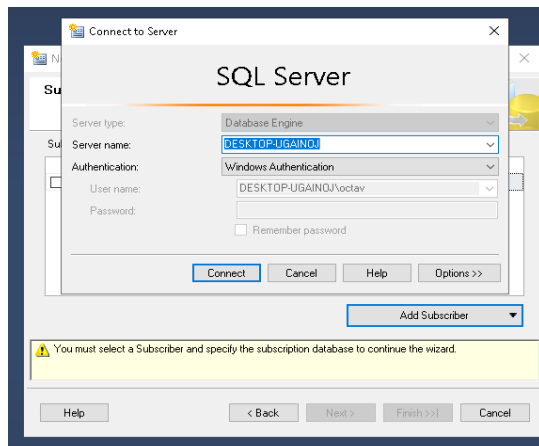


- En la página Ubicación del Agente de distribución, seleccione Ejecutar todos los agentes en el distribuidor y luego seleccione Siguiente. Esto hace que el agente de distribución se ejecute en el mismo servidor que el distribuidor, que es la base de datos replicada, esto se llama suscripción de inserción ya que el publicador o el distribuidor insertar los cambios en el suscriptor

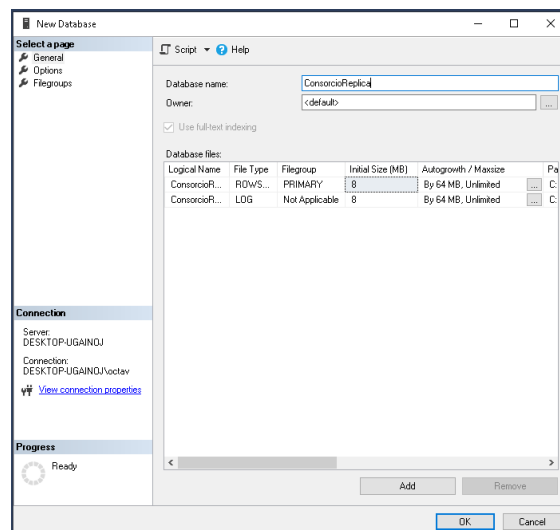
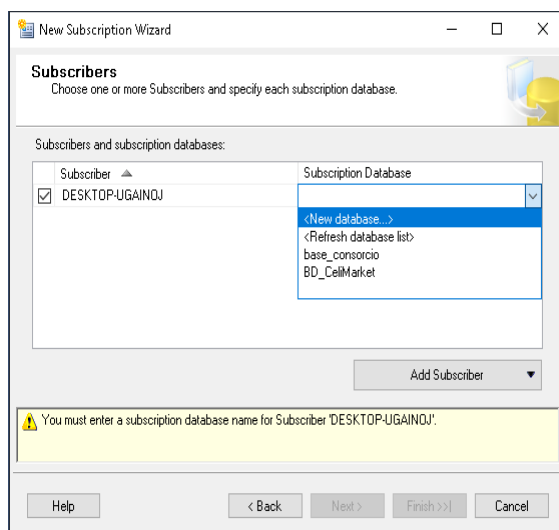


- En la Página de Suscriptores o New Subscription Wizard sino se muestra el nombre de la instancia del suscriptor ,debemos de Agregar el suscriptor de nuestra publicación que creamos anteriormente Seleccione Agregar suscriptor y, después, seleccione Agregar suscriptor de SQL Server en la lista desplegable.





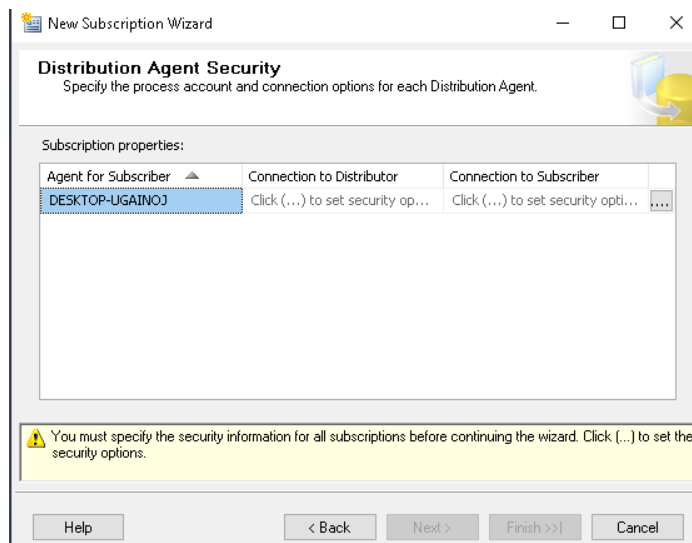
6. Ahora Seleccionamos en Subscription Database nueva base de datos para crear una base de datos vacía en el suscriptor que almacenará los datos replicados, y escribimos dentro del cuadro de diálogo que se nos abrió el nombre de la base de datos de suscripción en este caso **ConsortioReplica** y le damos en aceptar y en siguiente



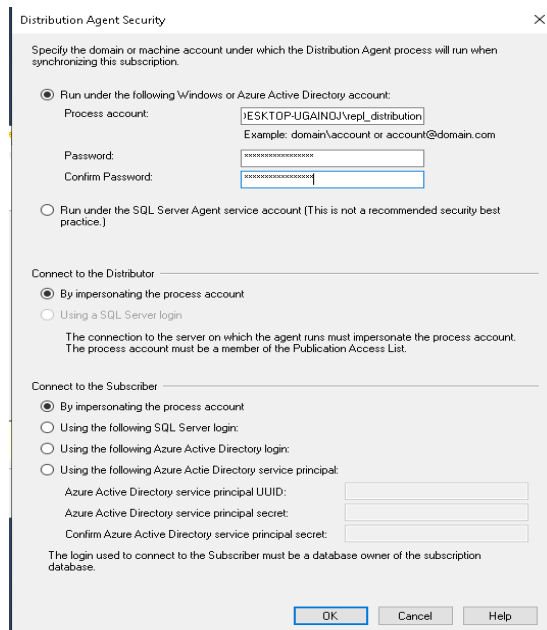
- El Agente de distribución es el componente de replicación que mueve los cambios desde el distribuidor al suscriptor.



- Entonces Haga clic en el botón de puntos suspensivos (...) para abrir el cuadro de diálogo Configuración de seguridad del Agente de distribución



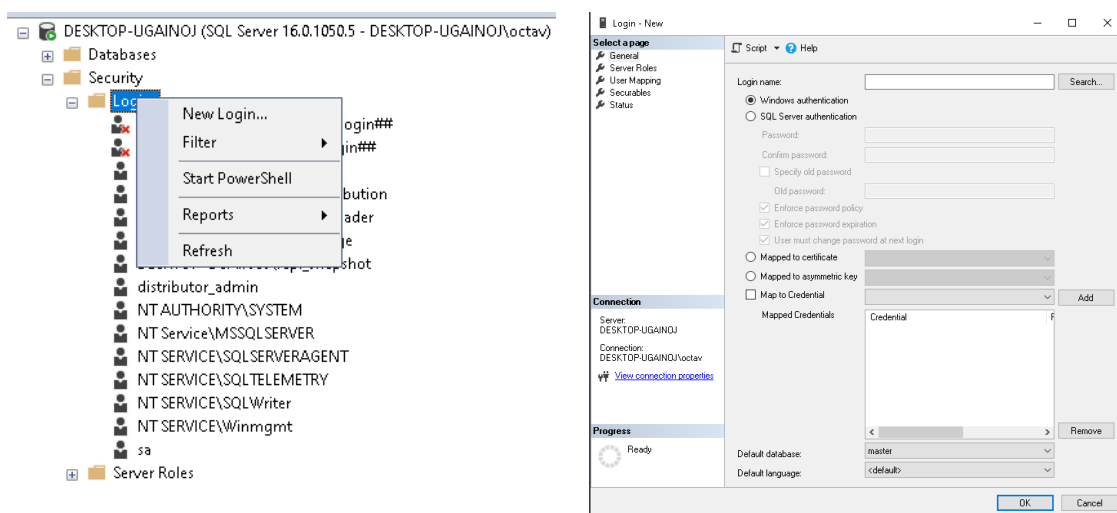
- Escriba <Publisher_Machine_Name>\repl_distribution y seleccione Aceptar para cerrar el cuadro de diálogo y guardar la configuración.





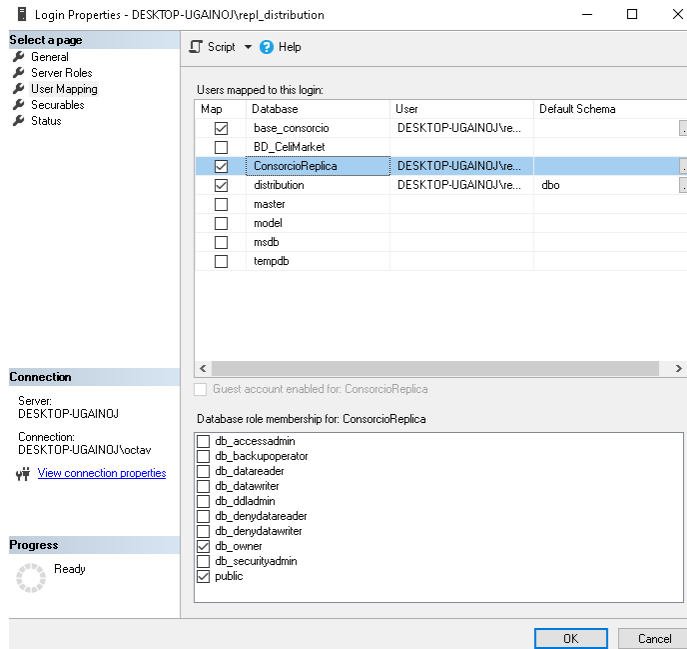
-
- New Subscription Wizard**
- Creating Subscription(s)...**
Click Stop to interrupt the operation.
- 2 Remaining**
- | | | 2 Total | 0 Success | 0 Error | Warning |
|---------------------------------------------|----------------|---------|-----------|---------|---------|
| Details: | | | | | |
| Action | Status | Message | | | |
| Creating subscription for 'DESKTOP-UGAINQJ' | In progress... | | | | |
| Starting the Snapshot Agent | | | | | |
- Stop** **Report**
- Close**
- New Subscription Wizard**
- Creating Subscription(s)...**
Click Stop to interrupt the operation.
- Success**
- | | | 2 Total | 2 Success | 0 Error | Warning |
|---------------------------------------------|---------|---------|-----------|---------|---------|
| Details: | | | | | |
| Action | Status | Message | | | |
| Creating subscription for 'DESKTOP-UGAINQJ' | Success | | | | |
| Starting the Snapshot Agent | Success | | | | |
- Stop** **Report**
- Close**

- Expanda la carpeta Seguridad, haga clic con el botón derecho en la carpeta Inicios de sesión(login) y seleccione Nuevo inicio de sesión(new login) para crear un nuevo inicio de sesión de SQL Server esto nos va a permitir autenticar la conexión y garantizar que solo los usuario autorizados puedan acceder a la base de datos
- En la página General, en Nombre de inicio de sesión, seleccione Buscar (search)y agregue el inicio de sesión de **<Subscriber_Machine_Name>\repl_distribution**.





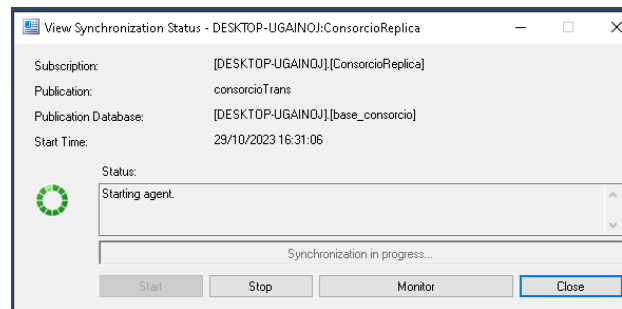
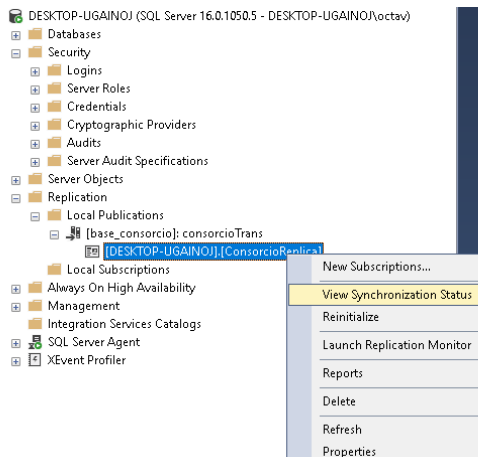
- En la página Asignaciones de usuario, otorgue el inicio de sesión al miembro **db_owner** para la base de datos **ConsortioReplica**, que es la base de datos que almacena los datos replicados



Ver el estado de sincronización de la suscripción

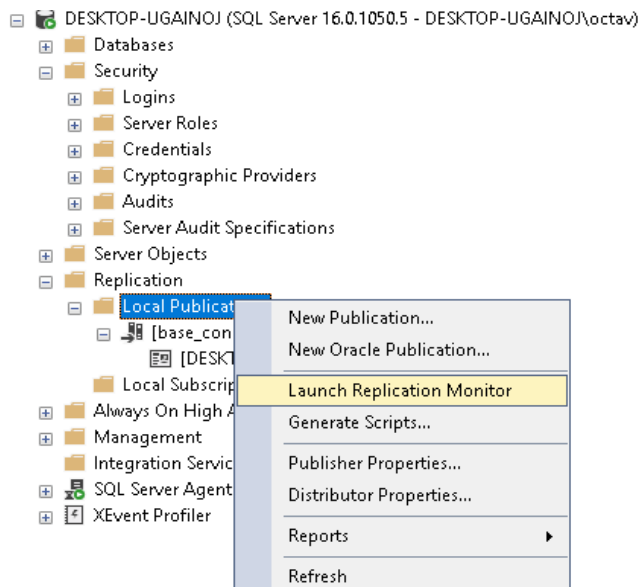
En esta parte vamos a ver el estado de sincronización de la suscripción que ha creado anteriormente. El estado de sincronización muestra información sobre el rendimiento y la actividad de la replicación, como la latencia, el número de cambios replicados y los posibles errores.

- Conéctese al publicador en SQL Server Management Studio. Expanda el nodo del servidor y luego la carpeta Replicación.
- Expanda la carpeta replicación que contiene las opciones de configuración y administración de la replicación, luego en la carpeta publicaciones locales expanda la publicación **consorcioTrans** que es la publicación transacción que creamos.



En esta sección se emplean testigos de seguimiento para comprobar que los cambios se replican en el suscriptor y para determinar la latencia. La latencia es el tiempo necesario para que un cambio realizado en el publicador aparezca en el suscriptor.

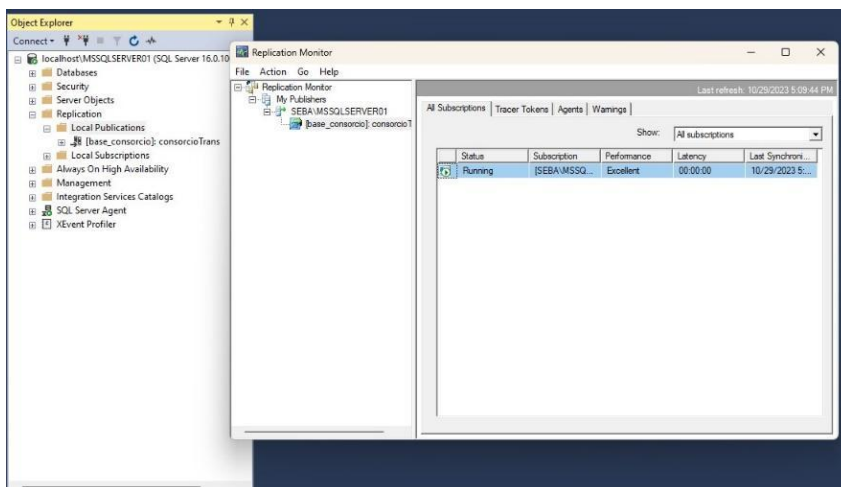
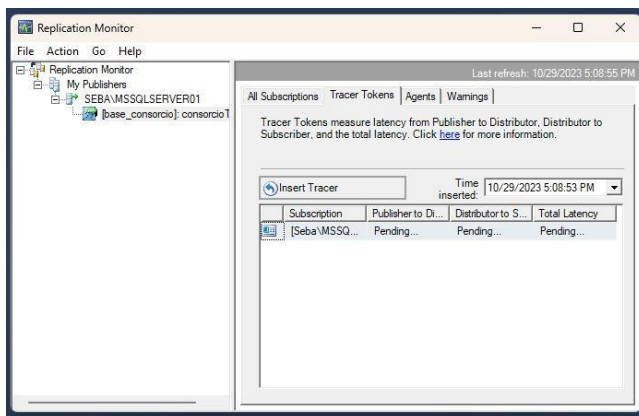
1. Conéctese al publicador en SQL Server Management Studio. Expanda el nodo del servidor, haga clic con el botón derecho en la carpeta Replicación y luego seleccione Iniciar Monitor de replicación:





2. Expanda un grupo de publicador en el panel izquierdo, expanda la instancia del publicador y, después, seleccione la publicación **consorcioTrans**

- Seleccione la pestaña Testigos de seguimiento.
- Seleccione Insertar seguimiento
- Vea el tiempo transcurrido para el testigo de seguimiento en las siguientes columnas: Publicador a distribuidor, Distribuidor a suscriptor y Latencia total.





CAPÍTULO V: INTEGRACIONES

Previamente llevamos a cabo una replicación de tipo transaccional. Tomamos la decisión de incorporar varios temas adicionales que consideramos más relevantes y compatibles. Entre las opciones disponibles, seleccionamos aquellas que mejor se adaptan a nuestro trabajo y que reflejan de manera efectiva el manejo de réplicas transaccionales y estas son:

- ❖ Manejo de transacciones
- ❖ Permisos de Usuarios
- ❖ BackUp y Restore
- ❖ Triggers
- ❖ Procedimientos y Funciones Almacenadas

Manejo de transacciones

Las transacciones en bases de datos son un conjunto de operaciones que se ejecutan como una unidad indivisible donde todas las operaciones deben tener éxito o fallar en su totalidad. Estas operaciones pueden incluir inserciones, actualizaciones, eliminaciones o consultas. La idea fundamental detrás de las transacciones es garantizar la integridad, coherencia y consistencia de los datos, incluso en situaciones de fallos o errores.

Estas transacciones deben cumplir 4 propiedades fundamentales llamadas ACID (atomicidad, consistencia, aislamiento y durabilidad)

1. Atomicidad: aseguran que todas las operaciones dentro de la secuencia de trabajo se completen satisfactoriamente.
2. Consistencia: aseguran que la base de datos cambie estados en una transacción exitosa.
3. Aislamiento: permiten que las operaciones sean aisladas y transparentes unas de otras.



4. Durabilidad: aseguran que el resultado o efecto de una transacción completada permanezca en caso de error del sistema.

Manejo de transacciones y transacciones anidadas: Los motivos por el cual elegimos este tema es por lo siguiente:

- Consistencia de datos: El manejo de transacciones asegura que las operaciones se ejecuten correctamente y de manera consistente en todas las réplicas.
- Atomicidad y coherencia: Las transacciones anidadas permiten combinar múltiples operaciones en una estructura lógica más amplia.
- Integridad de datos: Las transacciones anidadas y el manejo adecuado de las transacciones contribuyen a preservar la integridad de los datos.
- Mejora del rendimiento: Al agrupar operaciones relacionadas en una transacción, se reduce la cantidad de operaciones de escritura en el sistema, lo que puede mejorar el rendimiento general al minimizar el número de confirmaciones de transacciones.
- Escalabilidad y mantenimiento: Una gestión eficaz de las transacciones puede facilitar el mantenimiento del sistema y su escalabilidad.

Sumado a esto también permite la visibilidad de una manera sencilla en la replicación, permitiendo ver claramente cómo funciona:

Teniendo en cuenta que dentro de la Carpeta repldata se genera otra que sigue una estructura de fecha y hora específica (año/mes/día/hora/minuto/segundo) que se utiliza a menudo para nombrar carpetas o archivos cuando se manejan registros de transacciones o datos replicados como por ejemplo: 20231120102516. Pudiendo observar todos los cambios implementados en los en las tablas anteriormente establecidas.



administrador_2	administrador_2	administrador_2	administrador_2
consejer_3	consejer_3	consejer_3	consejer_3
consorcio_4	consorcio_4	consorcio_4	consorcio_4
gasto_5#1	gasto_5#2	gasto_5#3	gasto_5#4
gasto_5#5	gasto_5#6	gasto_5#7	gasto_5#8
gasto_5#9	gasto_5#10	gasto_5#11	gasto_5#12
gasto_5#13	gasto_5#14	gasto_5#15	gasto_5#16
gasto_5	gasto_5	gasto_5	localidad_6
localidad_6	localidad_6	localidad_6	provincia_7
provincia_7	provincia_7	provincia_7	tipogasto_8
tipogasto_8	tipogasto_8	tipogasto_8	tipoPropiedad_9
tipoPropiedad_9	tipoPropiedad_9	tipoPropiedad_9	zona_10
zona_10	zona_10	zona_10	

El código de Transacciones Implementados Fueron los siguientes:

(Transacción 1)

```
BEGIN TRY
    BEGIN TRAN TS_Anidades
    SELECT CONCAT('El nivel de anidamiento es ', @@TRANCOUNT) As 'Numero de anidamientos' --Esta sentencia cuenta el número de transacciones anidadas
    BEGIN TRAN TS_InsertarAdmin
        --Inserción registro administrador
        SELECT CONCAT('El nivel de anidamiento es ', @@TRANCOUNT) As 'Numero de anidamientos'
        INSERT INTO administrador(apeynom,viveahi,tel,sexo,fechnac) values ('Juan José Paso', 'S', '3794000102', 'M', '19890625')
    COMMIT TRAN TS_InsertarAdmin
    BEGIN TRAN TS_InsertarConsortio
        --Inserción registro consorcio
        SELECT CONCAT('El nivel de anidamiento es ', @@TRANCOUNT) As 'Numero de anidamientos'
        INSERT INTO consorcio(idprovincia,idlocalidad,idconsorcio,Nombre,direccion,idzona,idconsejer,idadmin)
        VALUES (7, 7, 2, 'EDIFICIO-24500', '9 de julio Nº 1650', 2, Null, (select top 1 idadmin from administrador order by idadmin desc))
        --Para insertar el id del último administrador se hace un select del ultimo id de administrador insertado.
    BEGIN TRAN TS_InsertarGastos
        SELECT CONCAT('El nivel de anidamiento es ', @@TRANCOUNT) As 'Numero de anidamientos'
        --Inserción 3 registros de gastos
        INSERT INTO gasto (idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
        VALUES (7, 7, 2,1,'20231005',2,5000.00)

        INSERT INTO gasto (idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
        VALUES (7, 7, 2,1,'20231015',2,20000.00)

        INSERT INTO gasto (idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
        VALUES (7, 7, 2,1,'20231028',2,10000.00)
    COMMIT TRAN TS_InsertarGastos
    COMMIT TRAN TS_InsertarConsortio

    COMMIT TRAN TS_Anidades
END TRY

BEGIN CATCH
    SELECT ERROR_MESSAGE() As Error -- Se captura el error y lo muestra.
    ROLLBACK TRAN TS_Anidades
END CATCH
```

(Transacción 2)

```
BEGIN TRY
    BEGIN TRAN

    --Inserción registro administrador
    INSERT INTO administrador(apeynom,viveahi,tel,sexo,fechnac) values ('Juan Pablo Duete', 'S', '3794000102', 'M', '19890625')

    --Inserción registro consorcio
    INSERT INTO consorcio(idprovincia,idlocalidad,idconsorcio,Nombre,direccion,idzona,idconsejer,idadmin)
    VALUES (7, 7, 1, 'EDIFICIO-24500', '9 de julio Nº 1650', 2, Null, (select top 1 idadmin from administrador order by idadmin desc))--id del último administrador

    --Inserción 3 registros gasto
    INSERT INTO gasto (idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
    VALUES (7, 7, 1,1,'20231005',2,5000.00)

    INSERT INTO gasto (idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
    VALUES (7, 7, 1,1,'20231015',2,20000.00)

    INSERT INTO gasto (idprovincia,idlocalidad,idconsorcio,periodo,fechapago,idtipogasto,importe)
    VALUES (7, 7, 1,1,'20231028',2,10000.00)

    COMMIT TRAN
END TRY

BEGIN CATCH
    SELECT ERROR_MESSAGE() As Error-- Se captura el error y lo muestra.
    ROLLBACK TRAN
END CATCH
```

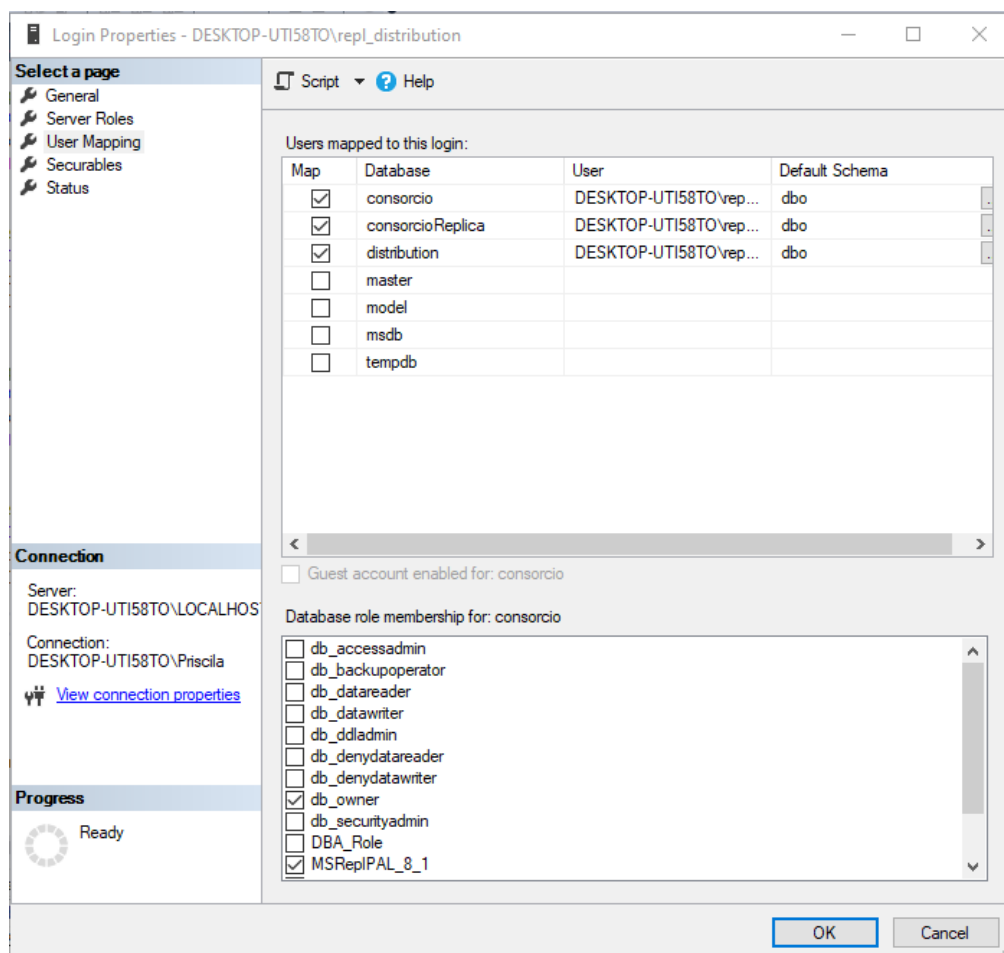


Manejo de Permisos a nivel de usuario

Un usuario representa a una persona, y el rol el conjunto de privilegios o acciones que puede realizar en la base de datos dicho usuario.

Existen roles con permisos predefinidos, aunque se pueden crear personalizados según las necesidades. Los permisos como por ejemplo consultas, crear, actualizar o eliminar datos, etc.

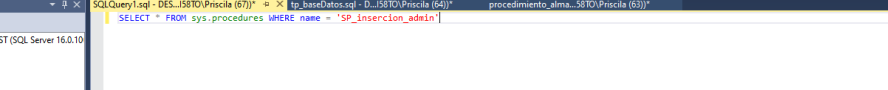
En nuestro caso hicimos uso del rol predefinido “db_owner” que utilizamos en la replicación para darle permiso de realizar todas las actividades de configuración y mantenimiento en la base de datos, y también pueden drop la base de datos en SQL Server en este caso se le otorgó dichos permisos al usuario “repl_distribution”,





Hay que tener en cuenta que los Users no se replican como tal de una base de datos a otra pero ende, se tendría que repetir esa creación de Users y designación de roles,pero por ejemplo si se realiza la un procedimiento interno si podemos observar esa replicación de datos como en el ejemplo de aquí.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the server hierarchy for 'consorcio', with 'Server Objects' expanded. The right pane shows the 'sys.procedures' table. The 'name' column is highlighted, showing 'SP_insercion_admin'. The 'principal_id' column is also highlighted, showing '1'. The 'type' column is 'P' (Procedure). The 'create_date' is '2023-11-20'.



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'Databases' folder expanded, with 'consorcioReplica' selected. The right pane shows the 'Properties' window for the database, with the 'General' tab active. The 'Name' field is set to 'consorcioReplica'. The 'Files' section shows the primary data file 'consorcioReplica.mdf' located at 'D:\SQLData\consorcioReplica.mdf' and the log file 'consorcioReplica.ldf' located at 'D:\SQLData\consorcioReplica.ldf'. The 'Collation' is set to 'Latin1_General_CI_AS_KS_WS'. The 'Status' section shows the database is 'Normal' and 'Full'. The 'Recovery Model' is set to 'Full'. The 'Log Backup' section shows the last backup was on 11/20/2023 at 12:00:00. The 'Detach With' section shows the database can be detached as 'Attachable' or 'Detach with copy'. The 'Detach With Copy' section shows the database can be detached with or without copy. The 'Detach With Copy' section shows the database can be detached with or without copy.



Backup y Restore:

Es relevante destacar que las réplicas de bases de datos también pueden ser consideradas como una forma de protección y disponibilidad de datos. Las réplicas no son simplemente copias de seguridad, sino copias en tiempo real de la base de datos que se mantienen sincronizadas con la base de datos principal.

Los aspectos más destacados en conjunto son:

1. Uso Complementario:

- Las réplicas de bases de datos se utilizan para tener versiones actualizadas de la base de datos en diferentes ubicaciones o servidores. Esto puede ser beneficioso en situaciones donde se requiere alta disponibilidad y rendimiento.

2. Protección contra Pérdida de Datos:

- Mientras que las copias de seguridad se realizan en intervalos específicos, las réplicas permiten tener una versión actualizada constantemente. Esto puede proporcionar una capa adicional de protección contra pérdida de datos.

3. Aplicación en Entornos de Desarrollo y Pruebas:

- Al igual que las restauraciones de copias de seguridad se utilizan en entornos de desarrollo y pruebas, las réplicas también pueden ser utilizadas para crear entornos de prueba con datos actualizados en tiempo real.



4. Respuesta ante Desastres:

- Las réplicas de bases de datos pueden ser parte de una estrategia integral de recuperación ante desastres al proporcionar redundancia y la capacidad de cambiar rápidamente a una réplica en caso de falla del servidor principal.

Consideraciones de Implementación:

Así como la implementación de copias de seguridad requiere planificación y consideración de recursos, la implementación de réplicas también implica decisiones sobre la topología de replicación, la sincronización y otros factores.

Aplicación en Bases de Datos en Línea (Backup en Línea):


Si estás explorando el concepto de base de datos en línea, las réplicas también pueden ser una forma de tener una copia en línea y accesible de la base de datos en ubicaciones remotas.

Dentro del moto SQL Server tomamos la base de datos llamada “**base_consortio**”.

Proceso para generar un backup para ello

```
BACKUP DATABASE base_consortio  
TO DISK = 'C:\Program Files\Microsoft  
SQLServer\MSSQL15.MSSQLSERVER\MSSQL\Backup\Consortio\Consortio.bak'  
WITH INIT, NAME = 'base_consortio2', STATS = 10
```

Esta sentencia lo que nos está queriendo decir es que se realizará un Backup de la base de datos “**base_consortio2**” en la ruta seleccionada, en donde se creará un archivo, el cual contenga la base de datos completa y si hay algún archivo con el mismo nombre, lo sobrescribirá creando un nuevo archivo.

Nombre	Fecha de modificación	Tipo	Tamaño
 Consortio.bak	18/10/2023 16:46	Archivo BAK	4.344 KB



Para realizar un Backup con fecha y hora del sistema podemos modificar el script de la siguiente forma:

```
/* Aquí se declara una variable llamada @Fecha con un tipo de datos VARCHAR de longitud 200. Esta variable se
usará para almacenar la fecha y hora actual en un formato específico.*/
DECLARE @Fecha VARCHAR(200)

/*Aquí se asigna un valor a la variable @Fecha la cual sera el resultado de la siguiente operacion, la cual convierte a VARCHAR la fecha
actual*/
SET @Fecha = REPLACE(CONVERT(VARCHAR,GETDATE(),100), ':', '.')



/*Se declara una segunda variable llamada @DireccionCarpeta con un tipo de datos VARCHAR de longitud 400. Esta variable se usará para
almacenar la ruta del archivo de respaldo de la base de datos.*/
Declare @DireccionCarpeta Varchar(400)

/*En esta línea, se asigna el valor que tomara la variable @DireccionCarpeta la cual sera la ruta del archivo de respaldo para la base de datos*/
Set @DireccionCarpeta = 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\Backup\Consortio\Consortio ' + @Fecha + '.bak'

/*Esta es la sentencia donde se ejecuta el BACKUP de la base de datos llamada base_consortio en el archivo especificado
por la variable @DireccionCarpeta*/
BACKUP DATABASE base_consortio
TO DISK = @DireccionCarpeta

/*WITH INIT: Indica que se está realizando una copia de seguridad inicial. Si ya existen copias de seguridad en el archivo de respaldo,
esta opción las sobrescribe.
NAME = 'base_consortio': Aquí se asigna un nombre a la copia de seguridad.
STATS = 10: Muestra información de progreso en la operación de copia de seguridad cada vez que se completen 10 porcentajes de la operación.*/
WITH INIT, NAME = 'base_consortio', STATS = 10
```

Dando como resultado:

Nombre	Fecha de modificación	Tipo	Tamaño
 Consortio Oct 18 2023 6.38PM.bak	18/10/2023 18:38	Archivo BAK	4.792 KB
 Consortio.bak	18/10/2023 18:35	Archivo BAK	4.792 KB

Por último para realizar el Restore de la Copia de seguridad (Backup) realizada:

```
/*En esta línea, se declara una variable llamada @NombreDataBase y se le asigna el valor 'base_consortio'. Esta variable se utilizará para especificar
el nombre de la base de datos que se va a restaurar.*/
DECLARE @NombreDataBase VARCHAR(200) = 'base_consortio';

/*Se declara otra variable llamada @Ubicacion que se utilizará para almacenar la ubicación (ruta del archivo) de la última copia de seguridad realizada
para la base de datos base_consortio. Inicialmente, esta variable está vacía.*/
DECLARE @Ubicacion NVARCHAR(128);

/*Esta consulta se utiliza para recuperar la ubicación del archivo de la ***última copia de seguridad de la base de datos base_consortio.*** */

/*Se selecciona la columna physical_device_name de la tabla backupmediafamily, que contiene la ubicación del archivo de copia de seguridad.
Se filtra la consulta para que solo incluya registros donde el nombre de la base de datos (b.database_name) coincide con el valor almacenado en
@NombreDataBase (en este caso, 'base_consortio') y además que solo incluyan los archivos con la extensión '.bak'.
Los resultados se ordenan por la fecha de inicio de la copia de seguridad (b.backup_start_date) en orden descendente (del más reciente al más antiguo).
La cláusula TOP 1 se utiliza para seleccionar solo el primer registro (el más reciente) que cumple con las condiciones, y su valor se asigna a la
variable @Ubicacion.*/
SELECT top 1 @Ubicacion = m.physical_device_name
FROM msdb.dbo.backupset AS b
JOIN msdb.dbo.backupmediafamily AS m ON b.media_set_id = m.media_set_id
WHERE b.database_name = @NombreDataBase
/*AND RIGHT(m.physical_device_name, 4) = '.bak': Esta línea agrega otra condición al filtro. Utiliza la función RIGHT() para extraer los últimos cuatro
caracteres de la columna physical_device_name en la tabla backupmediafamily. Luego, compara esos cuatro caracteres con '.bak'. Esto se hace para asegurarse
de que la ubicación física del archivo de copia de seguridad termine con '.bak', lo que indica que es un archivo de copia de seguridad con la extensión
'.bak'.*/
AND RIGHT(m.physical_device_name, 4) = '.bak' --
ORDER BY b.backup_start_date DESC;

/*En esta línea, se ejecuta la sentencia de restauración de la base de datos base_consortio. La restauración se realiza desde el archivo de
copia de seguridad cuya ubicación se determinó en la consulta anterior y se almacena en la variable @Ubicacion. */
RESTORE DATABASE base_consortio

/* Especifica que la restauración se realiza desde el archivo de copia de seguridad que se encuentra en la ubicación almacenada en @Ubicacion.*/
FROM DISK = @Ubicacion

/*Esta opción permite reemplazar la base de datos existente con la restaurada.*/
/* Esta opción coloca la base de datos en estado de recuperación, lo que significa que la base de datos estará disponible para su uso después
de esta operación.*/
WITH REPLACE, RECOVERY;
```



Mediante la ejecución de la sentencia Restore podremos recuperar la base a partir del último Backup realizado.

The screenshot shows the SQL Server Enterprise Manager interface on the left, with the 'base_consorcio' database selected under the 'Bases de datos' folder. The main window displays a SQL query in the 'SQL Query1.sql' file, which is used to restore the database. The query is as follows:

```
1 DECLARE @NombreDataBase VARCHAR(200) = 'base_consorcio';
2
3
4 DECLARE @Ubicacion NVARCHAR(128);
5
6 SELECT TOP 1 @Ubicacion = m.physical_device_name
7 FROM msdb.dbo.backupset AS b
8 JOIN msdb.dbo.backupmediafamily AS m ON b.media_set_id = m.media_set_id
9 WHERE b.database_name = @NombreDataBase
10 AND RIGHT(m.physical_device_name, 4) = '.bak'
11 ORDER BY b.backup_start_date ASC;
12
13
14
15 RESTORE DATABASE base_consorcio
16 FROM DISK = @Ubicacion
17 WITH REPLACE, RECOVERY;
```

Below the query, the 'Mensajes' (Messages) pane shows the execution results:

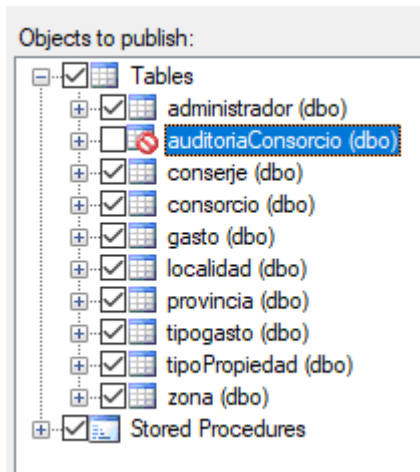
```
Se han procesado 696 páginas para la base de datos 'base_consorcio', archivo 'base_consorcio' en el archivo 1.
Se han procesado 2 páginas para la base de datos 'base_consorcio', archivo 'base_consorcio_log' en el archivo 1.
RESTORE DATABASE procesó correctamente 698 páginas en 0.082 segundos (66.459 MB/s).
Hora de finalización: 2020-10-21T22:25:44.7059962-03:00
```

Triggers

Los triggers son objetos esenciales en cualquier Sistema de Gestión de Bases de Datos y consisten en un conjunto de reglas predefinidas que se asocian a tablas específicas. Estas reglas se desencadenan automáticamente en la base de datos en respuesta a eventos particulares que se pueden dar antes o después de estas operaciones, como inserciones (INSERT), actualizaciones (UPDATE) o eliminaciones (DELETE) de registros. Esto permite automatizar acciones específicas relacionadas al evento y la tabla en que se produjo. Esto puede ayudar a ahorrar tiempo y mejora la eficiencia en la gestión de datos.

En nuestro caso sería de utilidad llevar una auditoría de las acciones que se llevan a cabo en la base de datos del publicador.

Para la implementación de los mismo, debemos saber que en el tipo de replicación que decidimos utilizar, en este caso la transaccional no podrá hacer la replicación en las tablas que no tengan un primary key, y se generará un error.

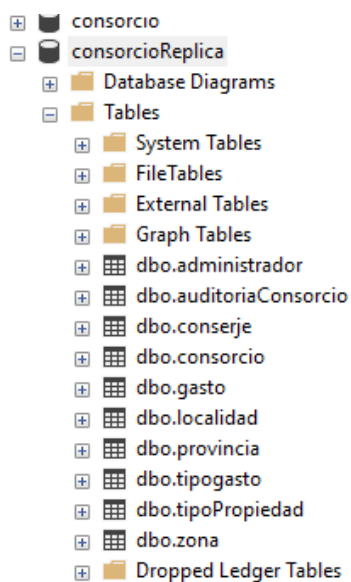


Para ello debimos modificar la creación de la tabla de esta manera:

Creación de la Tabla teniendo en cuenta un **Primary Key**.

```
-- Crear tabla auditoriaConsortio con una columna de clave primaria
CREATE TABLE auditoriaConsortio (
    auditoriaID INT IDENTITY(1,1) PRIMARY KEY, -- Columna de clave primaria autoincremental
    idprovincia int,
    idlocalidad int,
    idconsorcio int,
    nombre VARCHAR(50),
    direccion VARCHAR(250),
    idzona int,
    idconserje int,
    idadmin int,
    fechayhora date,
    usuario varchar(50),
    tipoOperacion varchar(50)
);
GO
```

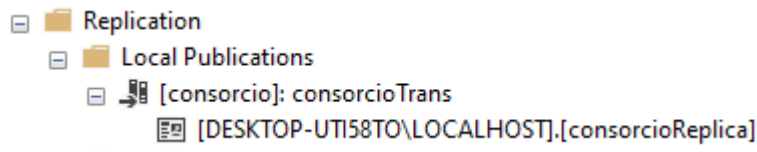
Se crea la tabla en la base de datos Publicador



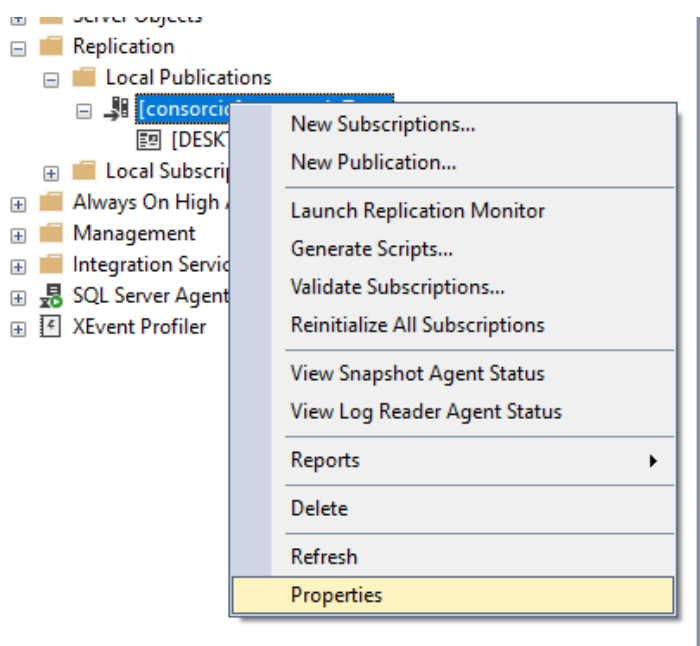


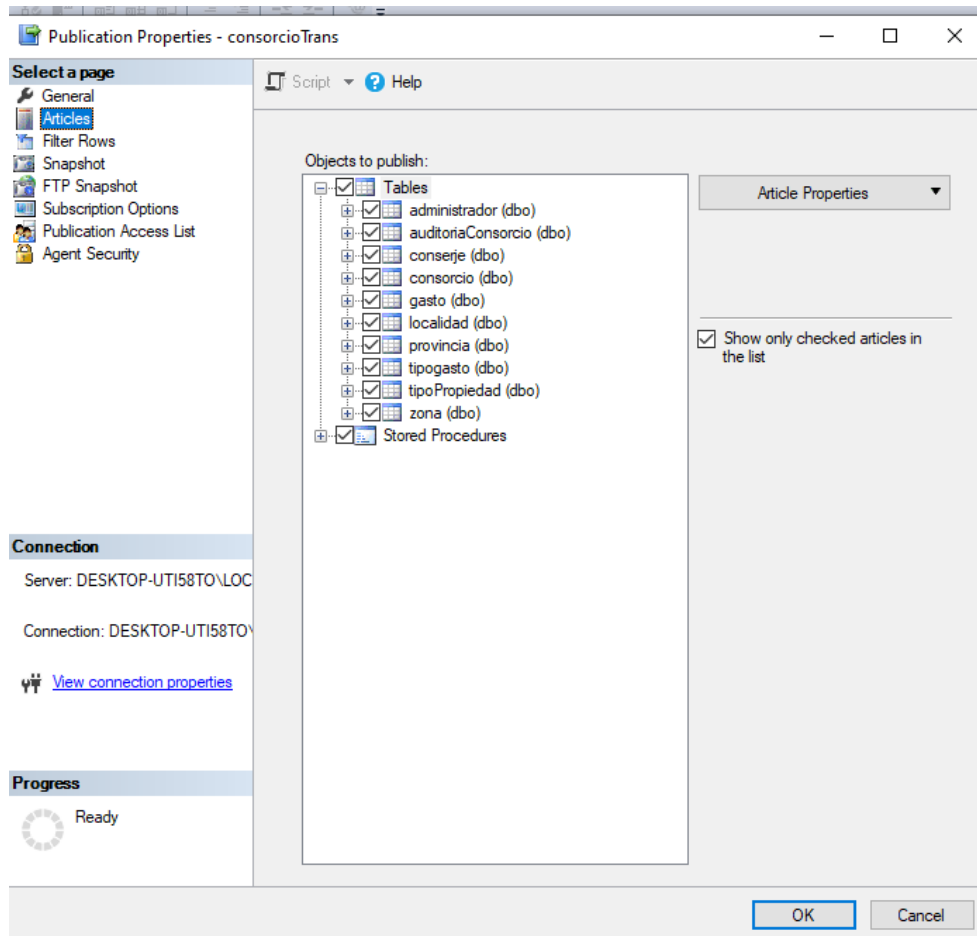
Será necesario activar las tablas a sincronizar con el suscriptor,teniendo en cuenta que si bien se seleccionó todas las tablas,son nuevas tablas incorporadas y no se encontraran marcadas, por lo que debemos hacerlo de la siguiente manera:

Se buscará la carpeta Replication,luego sobre la replicación de la base de datos se hará click derecho.

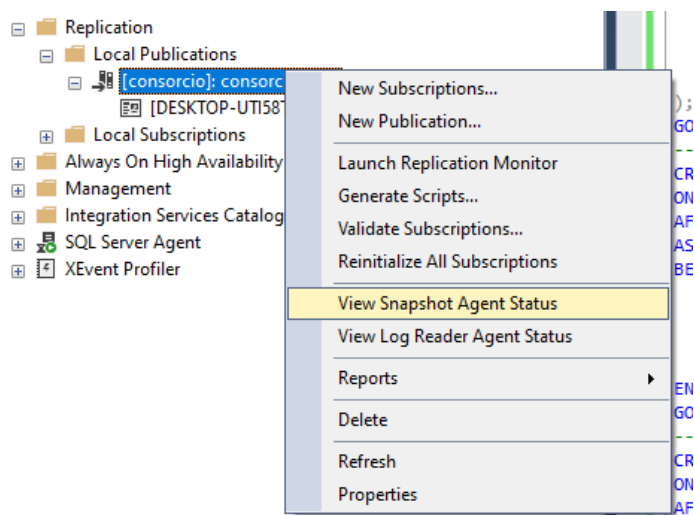


Se seleccionará properties y luego al aparecer la ventana emergente, se buscará Articles y se seleccionan las tablas deseadas para la replicación:



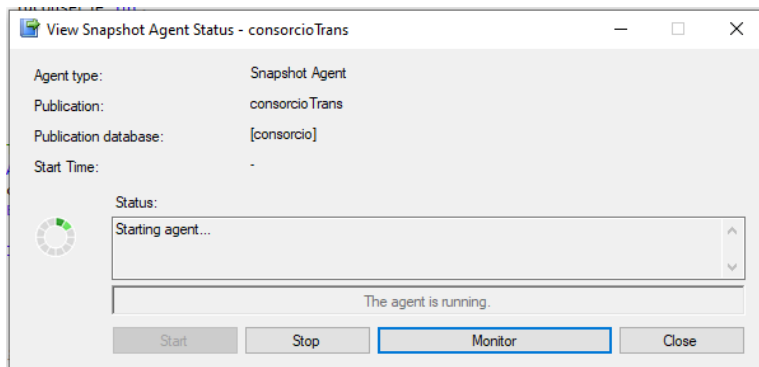


El siguiente paso será la sincronización entre el publicador y suscriptor, seleccionando click derecho sobre la réplica y eligiendo View Snapshot Agent Status

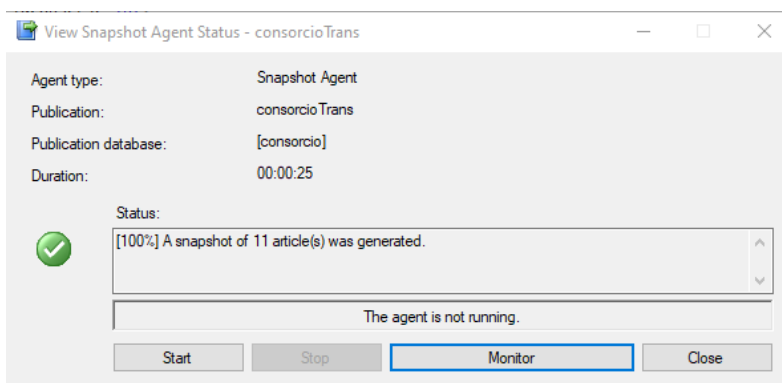




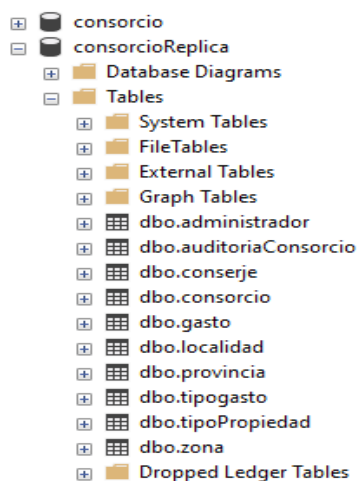
Aparecerá esta ventana genera una carga de sincronización



Y finalizara de esta forma demostrando que a sido sincronizada



Finalmente al observar la réplica de la base de datos obtendremos esa tabla incorporada





Ahora si se desea aplicar por ejemplo este Trigger como es el caso de Delete se podrá observar de esta manera:

```
-- Trigger para la operación de UPDATE en la tabla consorcio
CREATE TRIGGER trg_auditConsortio_update
ON consorcio
AFTER UPDATE
AS
BEGIN
    -- Registrar los valores antes de la modificación en la tabla de auditoría
    INSERT INTO auditoriaConsortio (idprovincia, idlocalidad, idconsorcio, nombre, direccion, idzona, idconserje, idadmin, fechayhora, usuario, tipoOperacion)
    SELECT idprovincia, idlocalidad, idconsorcio, nombre, direccion, idzona, idconserje, idadmin, GETDATE(), SUSER_NAME(), 'Update'
    FROM deleted;
END;
GO

-- Trigger para la operación de DELETE en la tabla consorcio
CREATE TRIGGER trg_auditConsortio_delete
ON consorcio
AFTER DELETE
AS
BEGIN
    -- Registrar los valores antes de la eliminación en la tabla de auditoría
    INSERT INTO auditoriaConsortio (idprovincia, idlocalidad, idconsorcio, nombre, direccion, idzona, idconserje, idadmin, fechayhora, usuario, tipoOperacion)
    SELECT idprovincia, idlocalidad, idconsorcio, nombre, direccion, idzona, idconserje, idadmin, GETDATE(), SUSER_NAME(), 'Delete'
    FROM deleted;
END;
GO

-- Eliminar un registro de la tabla consorcio
DELETE FROM consorcio WHERE idconsorcio = 11 -- Reemplaza "idconsorcio" con un valor existente en tu tabla

-- Consulta para verificar el registro en la tabla de auditoría después del DELETE
SELECT *
FROM auditoriaConsortio;
```

	auditoriaID	idprovincia	idlocalidad	idconsorcio	nombre	direccion	idzona	idconserje	idadmin	fechayhora	usuario	tipoOperacion
1	1	24	17	11	prueba	ponce	1	1	100	2023-11-20	DESKTOP-UTI58TO\Priscila	Delete

Procedimientos y Funciones Almacenadas

Los procedimientos almacenados son bloques de código T-SQL (Transact-SQL) que se almacenan en la base de datos y pueden ser ejecutados repetidamente según sea necesario. Estos procedimientos proporcionan una manera eficiente de encapsular una secuencia de comandos SQL para realizar una tarea específica o procesamiento de datos.

Los procedimientos almacenados permiten recibir parámetros de entrada y/o salida, a diferencia de las funciones pueden realizar operaciones internas que alteren la base de datos tales como INSERTS, UPDATES y/o DELETES.

Teniendo en cuenta el acoplamiento con la replicación de la base de datos el procedimiento es muy similar a los anteriores creando procedimiento interno:

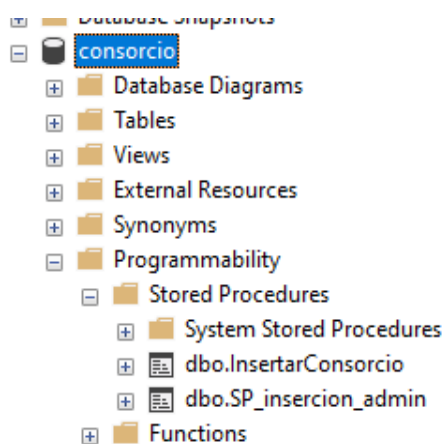


```
USE consorcio
GO

DROP PROCEDURE IF EXISTS [InsertarConsortorio]
GO

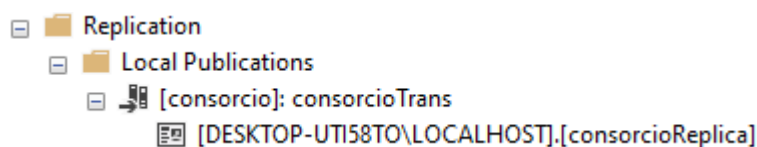
CREATE PROCEDURE [InsertarConsortorio] (
    @idprovincia int = null,
    @idlocalidad int = null,
    @idconsorcio int = null,
    @nombre varchar(50) = null,
    @direccion varchar(250) = null,
    @idzona int = null,
    @idconserje int = null,
    @idadmin int = null,
    @exito bit OUT,
    @error varchar(200) OUT
)
AS
BEGIN
    SET @exito = 1
    BEGIN TRY
        BEGIN TRAN
        INSERT INTO consorcio (idprovincia, idlocalidad, idconsorcio, nombre, direccion, idzona, idconserje, idadmin)
        VALUES (@idprovincia, @idlocalidad, @idconsorcio, @nombre, @direccion, @idzona, @idconserje, @idadmin)
        COMMIT TRAN
    END TRY
    BEGIN CATCH
        SET @error = ERROR_MESSAGE()
        ROLLBACK TRAN;
        SET @exito = 0
    END CATCH
END
```

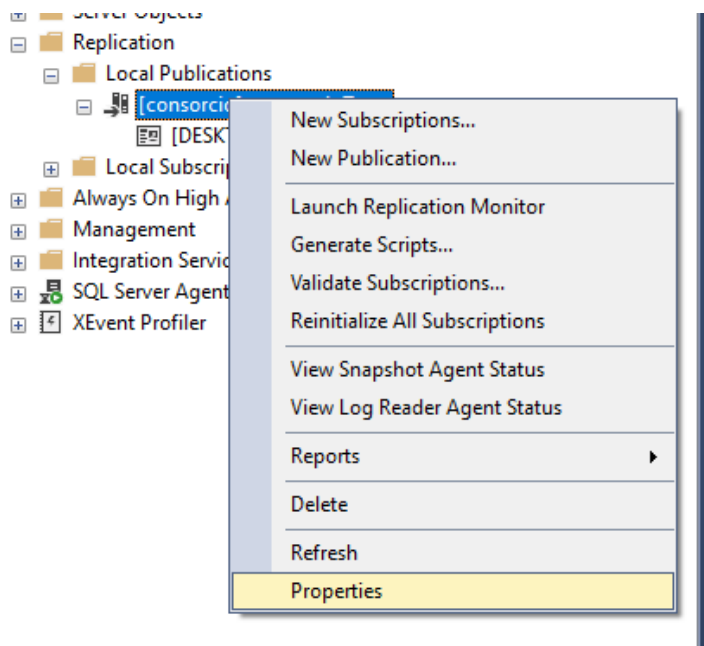
Esto genera un Procedimiento Almacenado en nuestra Base de Datos Publicador:



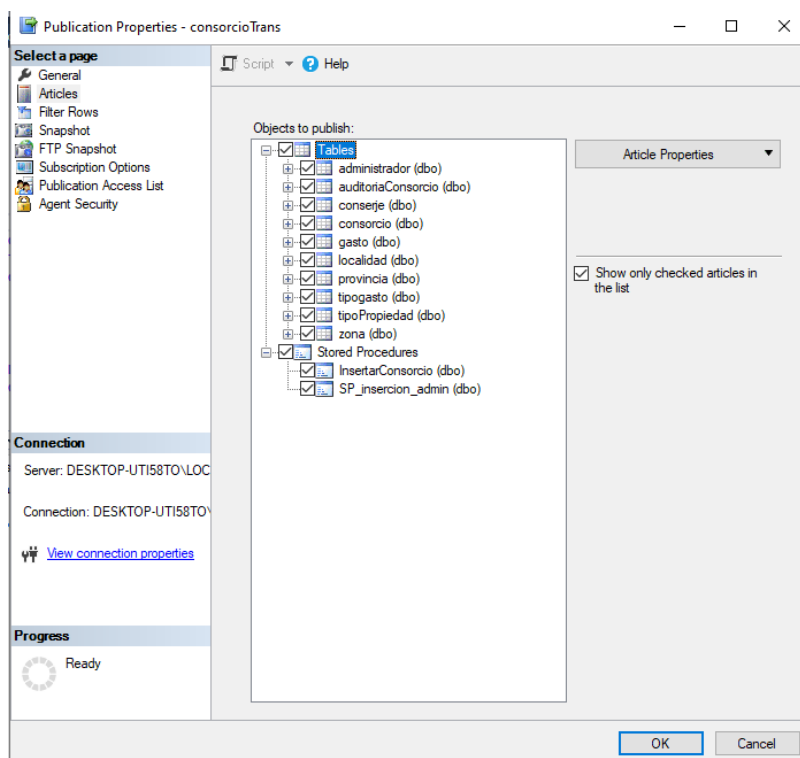
Teniendo en cuenta eso deberemos volver a realizar los pasos de activación y sincronización con nuestra base de datos suscriptora

Se buscará la carpeta Replication, luego sobre la replicación de la base de datos se hará click derecho.

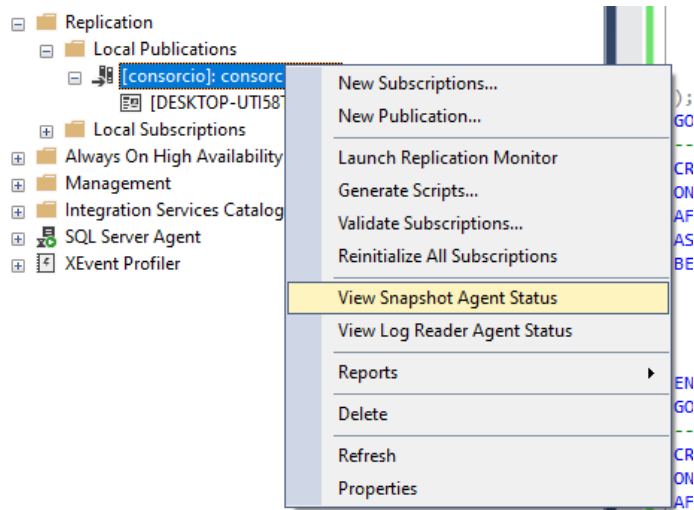




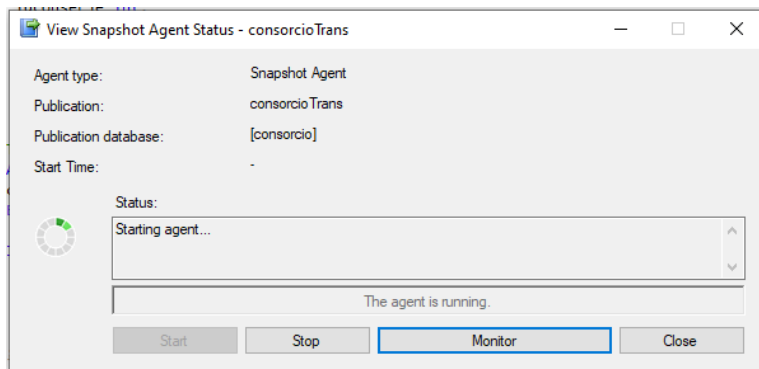
Se seleccionará Properties y luego al aparecer la ventana emergente, se buscará Articles y se seleccionan las tablas deseadas para la replicación, en este caso buscaremos en Stored Procedures:



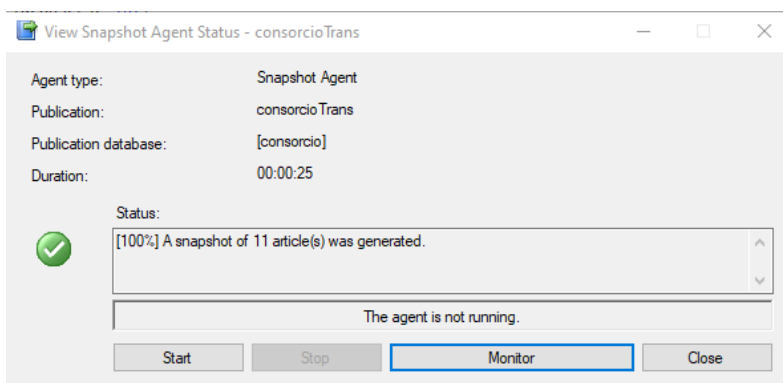
El siguiente paso será la sincronización entre el publicador y suscriptor, seleccionando click derecho sobre la réplica y eligiendo View Snapshot Agent Status



Aparecerá esta ventana, generando una carga de sincronización



Y finalizará de esta forma demostrando que a sido sincronizada





Finalmente al observar la réplica de la base de datos obtendremos esa tabla incorporada

- consorcioReplica
 - Database Diagrams
 - Tables
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Stored Procedures
 - System Stored Procedures
 - dbo.InsertarConsortio
 - dbo.SP_insercion_admin
 - dbo.sp_MSdel_dboadministrador
 - dbo.sp_MSdel_dboauditoriaConsortio
 - dbo.sp_MSdel_dboconserje
 - dbo.sp_MSdel_dboconsorcio
 - dbo.sp_MSdel_dboogasto
 - dbo.sp_MSdel_dbo localidad
 - dbo.sp_MSdel_dbo provincia
 - dbo.sp_MSdel_dbotipogasto
 - dbo.sp_MSdel_dbotipoPropiedad
 - dbo.sp_MSdel_dbozona
 - dbo.sp_MSins_dboadministrador
 - dbo.sp_MSins_dboauditoriaConsortio
 - dbo.sp_MSins_dboconserje
 - dbo.sp_MSins_dboconsorcio
 - dbo.sp_MSins_dboogasto
 - dbo.sp_MSins_dbo localidad
 - dbo.sp_MSins_dbo provincia
 - dbo.sp_MSins_dbotipogasto
 - dbo.sp_MSins_dbotipoPropiedad
 - dbo.sp_MSins_dbozona
 - dbo.sp_MSupd_dboadministrador
 - dbo.sp_MSupd_dboauditoriaConsortio
 - dbo.sp_MSupd_dboconserje



CAPÍTULO VI: CONCLUSIONES

Como conclusión, los resultados obtenidos muestran que es posible establecer una replicación efectiva en SQL Server, lo que garantiza que los cambios realizados en una base de datos de origen se reflejen de manera precisa y casi inmediata en una base de datos de destino. Esto es esencial para mantener la consistencia de los datos y asegurar su disponibilidad y redundancia en un entorno empresarial.

En este proceso, hemos configurado la distribución, creado publicaciones y definido artículos, y hemos establecido las suscripciones necesarias. Además, se han otorgado los permisos adecuados a las cuentas de Windows utilizadas para los agentes de replicación, lo que garantiza la seguridad y la ejecución efectiva de la replicación.



CAPÍTULO VI: BIBLIOGRAFÍA.

Geekflare. (s.f.). Replicación de datos: Explicado en 5 minutos o menos. Recuperado de <https://www.sqlshack.com/es/replicacion-de-sql-server-descripcion-general-de-componentes-y-topografia/>

Microsoft. (s.f.). Tipos de replicación. Recuperado de <https://learn.microsoft.com/es-es/sql/relational-databases/replication/types-of-replication?view=sql-server-ver16>

Microsoft. (s.f.). Réplica transaccional que es la que nos pide. Recuperado de <https://learn.microsoft.com/es-es/sql/relational-databases/replication/transactional/transactional-replication?view=sql-server-ver16>

SQLShack. (s.f.). Tutorial escrito: Replicación SQL - Instalación y configuración básica. Recuperado de <https://www.sqlshack.com/es/replicacion-sql-instalacion-y-configuracion-basica/>

Microsoft. (s.f.). Tutorial de Microsoft con requisitos. Recuperado de <https://learn.microsoft.com/es-es/sql/relational-databases/replication/tutorial-preparing-the-server-for-replication?view=sql-server-ver16>

Manejo de Transacciones y Transacciones Anidadas - https://github.com/Pablin89/Bases-de-datos-I_Grupo-4

Manejo de Permisos a nivel de usuario https://github.com/JulioCanteros/grupo6_Base_datos

Backup y Restore. Backup en Línea- https://github.com/juanmatiaspinat/ProyectoBDD_Grupo-7/tree/master

Triggers de Auditoría - https://github.com/LeanProgramming/G2_ProyectoEstudio_Triggers