Tecnologías de la Información, Ingeniería en Desarrollo y Gestión de Software

Store para creación de logins y users

Asignatura: Administración de base de datos

10 DE MARZO DE 2025

BENJAMIN PEÑA MARIN
Grupo. 1
Docente: José Herrera Gallardo

SQL dinamico logins

Introducción

La gestión de accesos y seguridad en bases de datos es esencial para garantizar la integridad y confidencialidad de la información. En este sentido, la creación de logins y usuarios en SQL Server es una tarea fundamental para establecer controles de acceso adecuados. El presente proyecto tiene como objetivo automatizar este proceso mediante el desarrollo de un procedimiento almacenado que permita crear de forma dinámica un login en el servidor y su correspondiente usuario en una base de datos, con la opción de asignar roles y permisos específicos sobre tablas. Esta solución se integra con una aplicación web desarrollada en PHP, que proporciona una interfaz intuitiva y moderna (utilizando Bootstrap) para gestionar la creación de estos elementos de seguridad de manera automatizada.

El proyecto aborda la necesidad de evitar errores manuales y garantizar una administración coherente y segura de los accesos, mediante la utilización de SQL dinámico, validaciones robustas y el manejo adecuado de errores. Este informe detalla el proceso de diseño, desarrollo e integración del stored procedure, las dificultades encontradas y las soluciones implementadas, demostrando cómo la automatización en la creación de logins y usuarios mejora la eficiencia y seguridad en entornos empresariales.

2. Objetivos

- **Objetivo:**
  Automatizar el proceso de creación de logins y usuarios en SQL Server mediante el desarrollo de un procedimiento almacenado que permita, de manera dinámica y segura, la creación de un login y su correspondiente usuario en una base de datos, incluyendo la asignación opcional de roles y permisos.
- **Objetivos Específicos:**
  - Desarrollar un stored procedure que reciba como parámetros el nombre del login, nombre del usuario, contraseña, nombre de la base de datos y, opcionalmente, un rol, esquema, nombre de tabla y permiso.
  - Implementar validaciones para evitar duplicidad de logins y usuarios, garantizando que la operación se ejecute solo cuando los elementos no existan previamente.
  - Generar SQL dinámico de forma segura utilizando funciones como QUOTENAME(), para evitar inyecciones y errores de sintaxis.
  - Integrar el procedimiento almacenado en una aplicación web PHP con un diseño profesional basado en Bootstrap, que permita a los administradores crear logins y usuarios de manera intuitiva.
  - Proporcionar retroalimentación clara al usuario final, mostrando mensajes de éxito o error según la ejecución del SP.

3. Desarrollo

3.1. Diseño del Stored Procedure

El procedimiento almacenado, denominado sp_CrearLoginUsuario, se desarrolló para automatizar la creación de logins y usuarios. Entre sus características principales se encuentran:

- **Recepción de Parámetros:**
  El SP recibe parámetros para:
  - **LoginName:** Nombre del login a crear.
  - **UserName:** Nombre del usuario asociado en la base de datos.
  - **DatabaseName:** Base de datos donde se creará el usuario.
  - **Password:** Contraseña del login, pasada como parámetro para mayor flexibilidad y seguridad.
  - **RoleName:** (Opcional) Rol al que se agregará el usuario (por ejemplo, db_datareader).
  - **SchemaName, TableName y Permission:** (Opcionales) Permiten otorgar permisos específicos sobre una tabla determinada.
- **Generación de SQL Dinámico:**
  Se construyen dinámicamente las sentencias T-SQL para:
  - Crear el login mediante la instrucción CREATE LOGIN.
  - Cambiar el contexto a la base de datos y crear el usuario con CREATE USER.
  - Asignar el usuario a un rol (usando sp_addrolemember) si se especifica un rol.
  - Conceder permisos sobre una tabla, usando la sentencia GRANT, si se proporcionan los parámetros correspondientes.
- **Validaciones Internas:**
  El SP verifica la existencia del login y del usuario para evitar duplicidades, retornando un mensaje de error en caso de que alguno de ellos ya exista.
- **Manejo de Errores:**
  Se implementa un bloque TRY...CATCH para capturar errores durante la ejecución, imprimiendo mensajes descriptivos y garantizando que se notifiquen los problemas sin interrumpir el proceso de forma inesperada.

3.2. Integración con la Aplicación PHP

La aplicación web se desarrolló con PHP y utiliza Bootstrap para un diseño profesional. Su integración con el SP se realiza de la siguiente manera:

- **Configuración de Conexión:**
  Se utiliza un archivo config.php que establece la conexión a SQL Server mediante la extensión sqlsrv, centralizando la configuración de conexión.
- **Formulario Web:**
  Se creó un formulario con campos para ingresar los parámetros requeridos: LoginName, UserName, DatabaseName, Password, RoleName,

SchemaName, TableName y Permission.
La interfaz, diseñada con Bootstrap, utiliza componentes como cards, alertas y botones, garantizando una experiencia de usuario intuitiva.

- **Llamada al Stored Procedure:**
  Al enviar el formulario, se recogen los datos y se llama al stored procedure mediante la función sqlsrv_query, pasando un arreglo de parámetros en el orden correcto.
  Se emplean consultas parametrizadas para evitar inyecciones SQL.
- **Manejo de Respuestas:**
  La aplicación interpreta la respuesta de la ejecución del SP, mostrando un mensaje de éxito ("Todo salió correctamente.") o, en caso de error, se muestran los detalles capturados con sqlsrv_errors().
- **Depuración:**
  Se han incluido sentencias PRINT @SQL; en el SP para facilitar la depuración durante el desarrollo, permitiendo revisar la instrucción final generada en caso de incidencias.

Dificultades Encontradas y Soluciones

1. Validación de Duplicidad de Logins y Usuarios

- **Dificultad:**
  Se detectó que, en algunos casos, se intentaba crear un login o un usuario que ya existía en el servidor o en la base de datos, lo que provocaba errores y duplicidades.
- **Solución:**
  Se implementaron validaciones al inicio del procedimiento para verificar, mediante consultas a sys.server_principals y sys.database_principals, si el login o el usuario ya existen. En caso afirmativo, se lanza un error descriptivo (con RAISERROR) y se interrumpe la ejecución del procedimiento, evitando duplicados.

2. Generación Segura de SQL Dinámico

- **Dificultad:**
  Al construir dinámicamente las sentencias T-SQL (para crear el login, el usuario y asignar roles o permisos), existía el riesgo de errores de sintaxis y de inyección SQL.
- **Solución:**
  Se utilizó la función QUOTENAME() para proteger los nombres de objetos y se estructuró cuidadosamente la concatenación de cadenas. Esto aseguró que los parámetros se insertaran de forma segura y que la sintaxis resultante fuera válida, evitando vulnerabilidades y errores de ejecución.

3. Manejo de Parámetros Opcionales (Rol y Permisos)

- **Dificultad:**
  En situaciones donde el usuario no deseaba asignar un rol o permisos sobre una tabla, era necesario que el procedimiento omitiese esas secciones sin provocar errores o ejecución incompleta.
- **Solución:**
  Se implementaron condicionales que evalúan si los parámetros opcionales (@RoleName, @TableName y @Permission) han sido suministrados. Si estos valores son NULL o están vacíos, el bloque correspondiente (para asignar roles o permisos) se omite, permitiendo que el SP se ejecute de forma correcta.

4. Manejo y Reporte de Errores

- **Dificultad:**
  Durante la ejecución del procedimiento, era fundamental capturar y reportar cualquier error de forma clara, sin interrumpir la integridad del sistema.
- **Solución:**
  Se incorporó un bloque TRY...CATCH en el procedimiento almacenado. En caso de error, se captura el mensaje con ERROR_MESSAGE() y se imprime, lo que facilita la depuración y el diagnóstico sin exponer información sensible al usuario final.

5. Integración con la Aplicación PHP

- **Dificultad:**
  Al consumir el stored procedure desde PHP, en algunas ocasiones se reportaban errores en la interpretación de la respuesta, a pesar de que la creación de logins y usuarios se realizaba correctamente en SQL Server.
- **Solución:**
  Se ajustó el manejo de la respuesta en la aplicación PHP para interpretar correctamente el resultado de sqlsrv_query(). Se estableció un mensaje de éxito personalizado en caso de que la ejecución del SP retornara correctamente, evitando que mensajes de depuración o warnings afecten la experiencia del usuario en la interfaz.

Conclusiones

- **Automatización Efectiva:**
  La implementación del procedimiento almacenado para la creación de logins y usuarios ha permitido automatizar un proceso crucial en la administración de seguridad en SQL Server. Esta automatización reduce la intervención manual y minimiza la posibilidad de errores, lo que resulta en una gestión de accesos más eficiente y coherente.
- **Seguridad y Robustez:**
  La utilización de SQL dinámico junto con funciones de seguridad como QUOTENAME() y el manejo de errores mediante bloques TRY...CATCH han garantizado que la creación de logins y usuarios se realice de forma segura. Además, las validaciones para evitar duplicidad aseguran que no se creen elementos redundantes, protegiendo la integridad del sistema.
- **Flexibilidad en la Configuración:**
  El procedimiento permite la asignación opcional de roles y permisos, lo que ofrece flexibilidad para adaptar la creación de logins y usuarios a distintos escenarios y necesidades de seguridad. Esta capacidad de configuración es especialmente valiosa en entornos donde se requieren políticas de acceso diferenciadas para distintos grupos de usuarios.
- **Integración con la Aplicación Web:**
  La integración del stored procedure en una aplicación web desarrollada en PHP, utilizando un diseño basado en Bootstrap, ha logrado simplificar la interacción con el sistema. La interfaz intuitiva y profesional facilita la gestión de accesos, ofreciendo retroalimentación clara sobre el estado de cada operación, lo que contribuye a una experiencia de usuario positiva.
- **Impacto en la Administración de Seguridad:**
  La solución implementada no solo mejora la eficiencia en la administración de logins y usuarios, sino que también refuerza la seguridad del entorno al garantizar que cada acción se realice con validaciones y controles adecuados. Esto resulta en una administración centralizada y consistente de los accesos a la base de datos, lo que es fundamental en entornos empresariales.