Documentación de Procedimientos Almacenados

Introducción

Este documento describe los procedimientos almacenados para realizar operaciones CRUD (Crear, Modificar, Eliminar, Consultar todos, Consultar por ID) en las tablas `view`, `module`, y `module_view` de la base de datos. Cada procedimiento sigue un estándar de nombres y utiliza parámetros específicos para asegurar consistencia, seguridad y trazabilidad.

Convención de Nombres

Cada procedimiento almacenado utiliza la siguiente convención:

- `spr_create_*`: Crea un nuevo registro en la tabla especificada.
- `spr_update_*`: Actualiza un registro existente en la tabla especificada.
- `spr_delete_*`: Realiza una eliminación lógica en la tabla especificada, manteniendo el historial.
- `spr_find_all_*`: Recupera todos los registros activos de la tabla especificada.
- `spr_find_by_id_*`: Recupera un registro específico de la tabla especificada mediante su ID.

Descripción de los Procedimientos por Tabla

Tabla: `view`

1. **spr_create_view** - **Crea un nuevo registro en la tabla `view`.**
    - **Descripción**: Este procedimiento inserta un nuevo registro en `view`, almacenando detalles de la vista, su ruta y el usuario que realizó la creación.
    - **Parámetros**:
        - `p_code` (VARCHAR(20)): Código de la vista.
        - `p_description` (VARCHAR(255)): Descripción de la vista.
        - `p_name` (VARCHAR(100)): Nombre de la vista.
        - `p_path` (VARCHAR(255)): Ruta de la vista.
        - `p_created_by` (BIGINT): ID del usuario que crea el registro.
    - **Salida**: Inserta el registro, establece `created_at` y `state = 1`.

2. **spr_update_view** - **Actualiza un registro en la tabla `view`.**
    - **Descripción**: Modifica los detalles de un registro existente en `view` utilizando el ID del registro y la información de actualización.
    - **Parámetros**:
        - `p_id` (BIGINT): ID del registro a actualizar.
        - `p_code` (VARCHAR(20)): Nuevo código de la vista.
        - `p_description` (VARCHAR(255)): Nueva descripción de la vista.
        - `p_name` (VARCHAR(100)): Nuevo nombre de la vista.
        - `p_path` (VARCHAR(255)): Nueva ruta de la vista.
        - `p_updated_by` (BIGINT): ID del usuario que realiza la actualización.

- **Salida**: Actualiza los campos especificados y establece `updated_at`.

3. **spr_delete_view** - **Realiza una eliminación lógica en la tabla `view`.**
   - **Descripción**: Marca el estado del registro como inactivo, estableciendo el campo `state = 0` para no eliminarlo físicamente.
   - **Parámetros**:
     - `p_id` (BIGINT): ID del registro a eliminar.
     - `p_deleted_by` (BIGINT): ID del usuario que realiza la eliminación.
   - **Salida**: Cambia el estado a inactivo, establece `deleted_at` y `deleted_by`.

4. **spr_find_all_view** - **Consulta todos los registros activos en `view`.**
   - **Descripción**: Recupera todos los registros activos en la tabla `view`, es decir, aquellos con `state = 1`.
   - **Parámetros**: Ninguno.
   - **Salida**: Devuelve una lista de registros activos.

5. **spr_find_view_by_id** - **Consulta un registro en `view` por su ID.**
   - **Descripción**: Recupera un registro activo específico en `view` utilizando su ID.
   - **Parámetros**:
     - `p_id` (BIGINT): ID del registro a consultar.
   - **Salida**: Devuelve el registro con `state = 1` y el ID proporcionado.

Tabla: `module`

1. **spr_create_module** - **Crea un nuevo registro en la tabla `module`.**
   - **Descripción**: Inserta un nuevo registro en `module`, registrando su código, descripción, nombre y usuario creador.
   - **Parámetros**:
     - `p_code` (VARCHAR(20)): Código del módulo.
     - `p_description` (VARCHAR(255)): Descripción del módulo.
     - `p_name` (VARCHAR(100)): Nombre del módulo.
     - `p_created_by` (BIGINT): ID del usuario que crea el registro.
   - **Salida**: Inserta el registro con `created_at` y `state = 1`.

2. **spr_update_module** - **Actualiza un registro en la tabla `module`.**
   - **Descripción**: Modifica los detalles de un registro en `module` usando su ID.
   - **Parámetros**:
     - `p_id` (BIGINT): ID del registro a actualizar.
     - `p_code` (VARCHAR(20)): Nuevo código del módulo.
     - `p_description` (VARCHAR(255)): Nueva descripción del módulo.
     - `p_name` (VARCHAR(100)): Nuevo nombre del módulo.
     - `p_updated_by` (BIGINT): ID del usuario que realiza la actualización.
   - **Salida**: Actualiza los campos y `updated_at`.

3. **spr_delete_module** - **Elimina un registro en `module`.**

- **Descripción**: Realiza una eliminación lógica cambiando el estado del registro a inactivo.
- **Parámetros**:
  - `p_id` (BIGINT): ID del registro a eliminar.
  - `p_deleted_by` (BIGINT): ID del usuario que realiza la eliminación.
- **Salida**: Cambia `state = 0` y registra `deleted_at` y `deleted_by`.

4. **spr_find_all_module** - **Consulta todos los registros activos en `module`.**

   - **Descripción**: Recupera todos los registros activos en la tabla `module`.
   - **Parámetros**: Ninguno.
   - **Salida**: Devuelve una lista de registros con `state = 1`.

5. **spr_find_module_by_id** - **Consulta un registro específico en `module` por ID.**

   - **Descripción**: Consulta un registro activo en `module` usando su ID.
   - **Parámetros**:
     - `p_id` (BIGINT): ID del registro.
   - **Salida**: Devuelve el registro correspondiente con `state = 1`.

Tabla: `module_view`

1. **spr_create_module_view** - **Crea un nuevo registro en `module_view`.**

   - **Descripción**: Registra una relación entre un módulo y una vista en `module_view`.
   - **Parámetros**:
     - `p_module_id` (BIGINT): ID del módulo.
     - `p_view_id` (BIGINT): ID de la vista.
     - `p_created_by` (BIGINT): ID del usuario que crea el registro.
   - **Salida**: Inserta el registro con `created_at` y `state = 1`.

2. **spr_update_module_view** - **Actualiza un registro en `module_view`.**

   - **Descripción**: Modifica una relación existente entre módulo y vista.
   - **Parámetros**:
     - `p_id` (BIGINT): ID del registro a actualizar.
     - `p_module_id` (BIGINT): Nuevo ID del módulo.
     - `p_view_id` (BIGINT): Nuevo ID de la vista.
     - `p_updated_by` (BIGINT): ID del usuario que realiza la actualización.
   - **Salida**: Actualiza los campos y establece `updated_at`.

3. **spr_delete_module_view** - **Elimina un registro en `module_view`.**

   - **Descripción**: Realiza una eliminación lógica, cambiando el estado del registro a inactivo.
   - **Parámetros**:
     - `p_id` (BIGINT): ID del registro a eliminar.
     - `p_deleted_by` (BIGINT): ID del usuario que realiza la eliminación.
   - **Salida**: Cambia `state = 0`, establece `deleted_at` y `deleted_by`.

4. **spr_find_all_module_view** - **Consulta todos los registros activos en `module_view`.**

- **Descripción**: Recupera todos los registros activos en `module_view`.
- **Parámetros**: Ninguno.
- **Salida**: Devuelve una lista de registros con `state = 1`.

5. **spr_find_module_view_by_id** - **Consulta un registro específico en `module_view` por ID.**

   - **Descripción**: Recupera un registro específico en `module_view` usando su ID.
   - **Parámetros**:
     - `p_id` (BIGINT): ID del registro.
   - **Salida**: Devuelve el registro correspondiente con `state = 1`.

Consideraciones Finales

Este conjunto de procedimientos almacenados permite una gestión completa y segura de los datos en las tablas `view`, `module`, y `module_view`, asegurando la trazabilidad de cada operación mediante campos de auditoría (`created_at`, `updated_at`, `deleted_at`, `state`). Este estilo de documentación es fundamental para entornos empresariales, permitiendo un mantenimiento eficiente y una mejor comprensión del funcionamiento de cada procedimiento.