

Documentación de Procedimientos Almacenados para la Tabla **producto**

Este documento explica en detalle los procedimientos almacenados en la base de datos para la gestión de la tabla **producto**. Cada procedimiento realiza operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los registros de productos, permitiendo un control completo y seguro de los datos.

Estructura de la Tabla **producto**

```
CREATE TABLE producto (  
  id BIGINT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(100),  
  descripcion VARCHAR(255),  
  precio DECIMAL(10, 2),  
  cantidad INT,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  updated_at TIMESTAMP NULL,  
  deleted_at TIMESTAMP NULL,  
  state TINYINT DEFAULT 1  
);
```

Descripción de los Campos Clave:

- id**: Identificador único del producto.
 - nombre**: Nombre del producto.
 - descripcion**: Descripción breve del producto.
 - precio**: Precio unitario del producto.
 - cantidad**: Cantidad disponible en el inventario.
 - created_at, updated_at, deleted_at**: Campos de auditoría para rastrear las fechas de creación, actualización y eliminación.
 - state**: Indica si el producto está activo (1) o inactivo (0), para permitir una "eliminación lógica".
-

Procedimientos Almacenados para **producto**

1. **spr_create_producto**

Descripción: Inserta un nuevo producto en la tabla **producto**.

Parámetros:

- p_nombre**: Nombre del producto.
- p_descripcion**: Descripción del producto.
- p_precio**: Precio del producto.
- p_cantidad**: Cantidad del producto en inventario.

Código del Procedimiento:

```
DROP PROCEDURE IF EXISTS spr_create_producto;
CREATE PROCEDURE spr_create_producto(
    IN p_nombre VARCHAR(100),
    IN p_descripcion VARCHAR(255),
    IN p_precio DECIMAL(10, 2),
    IN p_cantidad INT
)
BEGIN
    INSERT INTO producto (nombre, descripcion, precio, cantidad, created_at,
state)
    VALUES (p_nombre, p_descripcion, p_precio, p_cantidad, NOW(), 1);
END;
```

Ejemplo de Uso:

```
CALL spr_create_producto('Laptop', 'Laptop de 15 pulgadas', 1200.00, 10);
```

2. spr_update_producto

Descripción: Actualiza los datos de un producto existente.

Parámetros:

- **p_id:** ID del producto a actualizar.
- **p_nombre:** Nombre actualizado del producto.
- **p_descripcion:** Descripción actualizada del producto.
- **p_precio:** Precio actualizado del producto.
- **p_cantidad:** Cantidad actualizada del producto en inventario.

Código del Procedimiento:

```
DROP PROCEDURE IF EXISTS spr_update_producto;
CREATE PROCEDURE spr_update_producto(
    IN p_id BIGINT,
    IN p_nombre VARCHAR(100),
    IN p_descripcion VARCHAR(255),
    IN p_precio DECIMAL(10, 2),
    IN p_cantidad INT
)
BEGIN
    UPDATE producto
    SET nombre = p_nombre,
        descripcion = p_descripcion,
        precio = p_precio,
        cantidad = p_cantidad,
```

```
        updated_at = NOW()
    WHERE id = p_id AND state = 1;
END;
```

Ejemplo de Uso:

```
CALL spr_update_producto(1, 'Laptop Actualizada', 'Laptop de 15 pulgadas con SSD',
1300.00, 8);
```

3. `spr_delete_producto`

Descripción: Realiza una eliminación lógica de un producto.

Parámetros:

- `p_id`: ID del producto a eliminar.

Código del Procedimiento:

```
DROP PROCEDURE IF EXISTS spr_delete_producto;
CREATE PROCEDURE spr_delete_producto(
    IN p_id BIGINT
)
BEGIN
    UPDATE producto
    SET state = 0,
        deleted_at = NOW()
    WHERE id = p_id;
END;
```

Ejemplo de Uso:

```
CALL spr_delete_producto(1);
```

4. `spr_find_all_producto`

Descripción: Consulta todos los productos activos en la tabla `producto`.

Parámetros: Ninguno.

Código del Procedimiento:

```
DROP PROCEDURE IF EXISTS spr_find_all_producto;
CREATE PROCEDURE spr_find_all_producto()
BEGIN
    SELECT * FROM producto WHERE state = 1;
END;
```

Ejemplo de Uso:

```
CALL spr_find_all_producto();
```

5. `spr_find_producto_by_id`

Descripción: Consulta un producto específico utilizando su ID.

Parámetros:

- `p_id`: ID del producto a consultar.

Código del Procedimiento:

```
DROP PROCEDURE IF EXISTS spr_find_producto_by_id;
CREATE PROCEDURE spr_find_producto_by_id(
    IN p_id BIGINT
)
BEGIN
    SELECT * FROM producto WHERE id = p_id AND state = 1;
END;
```

Ejemplo de Uso:

```
CALL spr_find_producto_by_id(1);
```

Explicación de Conceptos Clave en los Procedimientos Almacenados

1. **Parámetros de Entrada (IN):** Los parámetros permiten que cada procedimiento sea flexible y reutilizable. Por ejemplo, los procedimientos `spr_create_producto` y `spr_update_producto` usan parámetros como `p_nombre`, `p_precio`, y `p_cantidad` para insertar o actualizar valores dinámicamente.

2. **Fechas de Auditoría:**

- `created_at`: Registra la fecha y hora en que se creó el registro.
- `updated_at`: Registra la fecha y hora en que se actualizó el registro.

- `deleted_at`: Registra la fecha y hora en que se eliminó lógicamente el registro.

3. **Eliminación Lógica:** En lugar de eliminar el producto físicamente, `spr_delete_producto` solo marca el registro como inactivo (`state = 0`). Este enfoque permite restaurar registros y conservar el historial de datos.

4. **Control de Estado (`state`):**

- `state = 1` significa que el producto está activo y visible en las consultas.
- `state = 0` indica que el producto ha sido eliminado lógicamente.

Este ejemplo demuestra cómo implementar y documentar procedimientos almacenados de manera detallada y estructurada, usando una entidad sencilla como `producto`.