

Documentación Técnica de sp_PorcentajeRegion

Equipo de Desarrollo de Bases de Datos

25 de junio de 2025

Objetivo

El procedimiento almacenado `sp_PorcentajeRegion` calcula el porcentaje de casos con diabetes, obesidad e hipertensión para una región geográfica específica (NORTE, CENTRO o SUR), accediendo a datos distribuidos en servidores vinculados.

Implementación SQL

```
1  -- 1) Eliminación previa si existe
2  IF OBJECT_ID('dbo.sp_PorcentajeRegion', 'P') IS NOT NULL
3      DROP PROCEDURE dbo.sp_PorcentajeRegion;
4  GO
5
6  -- 2) Creación del procedimiento
7  CREATE PROCEDURE dbo.sp_PorcentajeRegion
8      @Region NVARCHAR(10) -- 'NORTE', 'CENTRO' o 'SUR'
9  AS
10 BEGIN
11     SET NOCOUNT ON;
12
13     DECLARE
14         @LinkedServer NVARCHAR(50),
15         @RemoteTable NVARCHAR(100),
16         @SQL NVARCHAR(MAX);
17
18     -- 3) Configuración según región
19     IF UPPER(@Region) = 'NORTE'
20     BEGIN
21         SET @LinkedServer = 'SQL_NORTE';
22         SET @RemoteTable = 'datoscovid_norte.dbo.datoscovid_norte';
23     END
24     ELSE IF UPPER(@Region) = 'CENTRO'
25     BEGIN
26         SET @LinkedServer = 'SQL_CENTRO';
27         SET @RemoteTable = 'datoscovid_centro.dbo.datoscovid_centro';
28     END
29     ELSE IF UPPER(@Region) = 'SUR'
30     BEGIN
31         SET @LinkedServer = 'MYSQL_SUR';
32         SET @RemoteTable = 'covidhistorico.covidHistorico_sur';
33     END
34     ELSE
35     BEGIN
36         RAISERROR('Región inválida. Use: NORTE, CENTRO o SUR.', 16, 1);
37         RETURN;
38     END
39
40     -- 4) Consulta dinámica
41     SET @SQL = N'
42     SELECT
43         CAST(SUM(Total_Diabetes) * 100.0 / SUM(Total) AS DECIMAL(5,2)) AS Porcentaje_Diabetes,
44         CAST(SUM(Total_Obesidad) * 100.0 / SUM(Total) AS DECIMAL(5,2)) AS Porcentaje_Obesidad,
45         CAST(SUM(Total_Hipertension) * 100.0 / SUM(Total) AS DECIMAL(5,2)) AS Porcentaje_Hipertension
46     FROM (
47         SELECT
48             COUNT(*) AS Total,
49             SUM(CASE WHEN DIABETES = 1 THEN 1 ELSE 0 END) AS Total_Diabetes,
50             SUM(CASE WHEN OBESIDAD = 1 THEN 1 ELSE 0 END) AS Total_Obesidad,
51             SUM(CASE WHEN HIPERTENSION = 1 THEN 1 ELSE 0 END) AS Total_Hipertension
52     FROM OPENQUERY('' + @LinkedServer + '',
```

```

53      ''SELECT DIABETES, OBESIDAD, HIPERTENSION
54      FROM ' + @RemoteTable + '
55      WHERE CLASIFICACION_FINAL IN (1,2,3)''
56  ) AS SubTotales;';
57
58  -- 5) Ejecución
59  EXEC sp_executesql @SQL;
60 END
61 GO

```

Listing 1: Implementación del procedimiento almacenado

Explicación Técnica

1. Preparación del entorno:

- Elimina versiones anteriores del procedimiento para evitar conflictos
- Establece el parámetro de entrada @Region

2. Configuración distribuida:

- Selecciona el servidor vinculado y tabla remota según la región especificada
- Incluye validación para regiones no soportadas

3. Consulta dinámica:

- Construye una consulta que calcula porcentajes mediante:

$$\text{Porcentaje} = \left(\frac{\sum \text{Casos positivos}}{\sum \text{Casos totales}} \right) \times 100$$

- Usa OPENQUERY para acceder a datos remotos
- Filtra por CLASIFICACION_FINAL IN (1,2,3)

4. Ejecución segura:

- Utiliza sp_executesql para mejor rendimiento y seguridad
- SET NOCOUNT ON optimiza el tráfico de red

Recomendaciones

- **Índices:** Asegurar índices en CLASIFICACION_FINAL, DIABETES, OBESIDAD e HIPERTENSION en las tablas remotas
- **Permisos:** Verificar permisos en servidores vinculados
- **Monitoreo:** Registrar tiempos de ejecución para detectar problemas de rendimiento
- **Validación:** Añadir manejo de errores para conexiones remotas fallidas