

Laboratorio 12: Fragmentación Dinámica y Asignación Distribuida

Prof. Heider Sanchez

ACLs: Ana María Accilio, Sebastián Loza

Contexto

El Ministerio de Salud ha lanzado una iniciativa nacional para recolectar datos de pacientes provenientes de distintos centros de salud en todo el país. El objetivo es analizar el estado de salud de la población y fortalecer la toma de decisiones en políticas públicas.

Para este fin, se ha creado una tabla central llamada `Paciente` en un servidor principal, inicialmente poblada con datos provenientes de hospitales ubicados en Lima y Callao. El siguiente paso consiste en expandir esta iniciativa a nivel nacional.

Se ha optado por fragmentar horizontalmente la tabla `Paciente` según el atributo `Diagnóstico`. Inicialmente se manejarán solo cuatro fragmentos, pero el sistema debe estar preparado para crear nuevos fragmentos de manera dinámica conforme se incorporen más diagnósticos y registros. Además, el ministerio ha adquirido varios servidores adicionales para distribuir los datos, mejorando así los tiempos de respuesta para consultas y análisis.

Estructura de la tabla `Paciente` :

```
CREATE TABLE Paciente (  
  DNI CHAR(8),  
  Nombre VARCHAR(100) NOT NULL,  
  Ciudad VARCHAR(50) NOT NULL,  
  Diagnóstico VARCHAR(50) NOT NULL,  
  Peso DECIMAL(5,2) NOT NULL,  
  Edad INTEGER NOT NULL CHECK (Edad >= 0),  
  Sexo CHAR(1) NOT NULL CHECK (Sexo IN ('M', 'F'))  
);
```

Datos de ejemplo:

DNI	Nombre	Ciudad	Diagnóstico	Peso (kg)	Edad	Sexo
45781236	Carla María Romero Díaz	Lima	Diabetes	70	45	F
08569321	Luis Alberto Díaz Mendoza	Lima	Hipertensión	85	60	M
72103654	Ana Paula Torres Castro	Callao	Obesidad	90	35	F
25963147	Jorge Luis Ramírez Vargas	Callao	Cardiopatía	78	50	M
15478962	María Carmen Suárez López	Lima	Diabetes	65	42	F
36987412	Pedro José Quispe Huamán	Lima	Obesidad	95	38	M
65412398	Rosa Isabel Valle García	Lima	Hipertensión	72	55	F
89632147	Miguel Ángel Castro Rivas	Callao	Cardiopatía	82	48	M

P1. (2 pts) Creación de particiones

Crear la tabla `Paciente` utilizando particionamiento por el atributo `Diagnóstico`, inicialmente cuatro fragmentos: Diabetes, Obesidad, Cardiopatía, Hipertensión.

Luego, se solicita poblar la tabla con registros sintéticos de pacientes de otras ciudades y con diferentes diagnósticos.

P1. (8 pts) Creación dinámica de nuevos fragmentos

Diseñar un procedimiento almacenado que permita insertar nuevos registros en la tabla `Paciente`. Este procedimiento debe:

- Verificar si ya existe una partición para el diagnóstico del nuevo paciente.
- Crear dinámicamente una nueva partición si esta no existe.
 - Las particiones se irán generando a medida que se vayan ingresando nuevos datos.
- Insertar `k` registros diferentes que evidencien la creación dinámica del fragmento.

Este enfoque permite escalar el sistema conforme se integren nuevos tipos de diagnósticos en la base de datos.

Nota: Investigar el uso de **triggers** y restricciones de **clave primaria** en tablas particionadas. ¿Es posible aplicar claves primarias en particiones automáticas?

P2. (10 pts) Asignación distribuida

Utilizar **Docker Desktop** o **Ubuntu Virtualizado** para levantar al menos dos instancias remotas de PostgreSQL. A continuación:

- A partir del procedimiento almacenado anterior, modificar el código para trabajar con fragmentos alojados en otros servidores.
- Distribuir los fragmentos entre un servidor principal (coordinador) y los servidores remotos.
- Usar las extensiones `postgres_fdw` y `dblink` para acceder a tablas foráneas entre servidores.
- Ejecutar **dos consultas** diferentes que evidencien el uso de estas tablas distribuidas.
- Utilizar `EXPLAIN ANALYZE` para analizar el rendimiento de las consultas.

Recursos de apoyo:

- [Tutorial postgres_fdw \(Deepnote\)](#)
- [Tutorial DBLINK \(Deepnote\)](#)

- [Referencia técnica en Tencent Cloud sobre FDW](#)
- [PostgreSQL FDW Extension \(documentación oficial\)](#)
- [DBLINK Extension](#)

Entregable

Elaborar un documento que contenga:

1. **Explicación de los resultados** obtenidos, incluyendo capturas de pantalla relevantes.
2. **Script SQL** con la estructura y lógica desarrollada. Este script debe ser completamente replicable.