

Arquitectura de Business Intelligence: Pentaho OLAP sobre Neon (PostgreSQL)

Este documento detalla el proceso de diseño, implementación y explotación de un Cubo OLAP (Online Analytical Processing) utilizando **Pentaho** como motor de BI y **Neon** como base de datos serverless en la nube.

1. La Base de Datos: Neon (PostgreSQL Serverless)

El backend de esta solución se aloja en **Neon**, una plataforma moderna de PostgreSQL serverless que separa el cómputo del almacenamiento.

1.1 Estructura del Modelo: Esquema Estrella (Star Schema)

Para optimizar el rendimiento de las consultas analíticas, se diseñó un **Modelo Estrella**. A diferencia de las bases de datos transaccionales (normalizadas), este modelo busca la redundancia controlada para facilitar la lectura rápida.

Componentes:

1. **Tabla de Hechos (fact_ventas):**
 - Es el centro de la estrella. Contiene los eventos numéricicos (transacciones).
 - **Foreign Keys (FK):** producto_id, pais_id. Conectan con las dimensiones.
 - **Métricas:** cantidad (INT), total_dinero (DECIMAL). Son los valores que se suman o promedian.
 - **Degenerada:** La columna fecha se mantuvo en la tabla de hechos.
2. **Tablas de Dimensiones:**
 - **dim_producto:** Catálogo descriptivo. Atributos: categoria, nombre.
 - **dim_pais:** Catálogo geográfico. Atributos: region, pais.

1.2 Script SQL y Tipos de Datos

Se utilizó SQL estándar de PostgreSQL.

- **SERIAL:** Para IDs autoincrementales (Surrogate Keys).
- **VARCHAR:** Para atributos descriptivos (Texto).
- **DECIMAL(10,2):** Para dinero, asegurando precisión financiera.

2. Infraestructura e Instalación de Pentaho

La suite Pentaho se compone de dos herramientas principales que deben configurarse para trabajar juntas.

2.1 El Driver JDBC (El puente)

Pentaho no incluye nativamente el conector para PostgreSQL moderno.

- **Archivo:** postgresql-42.x.x.jar.
- **Ubicación Crítica:**
 - En **Schema Workbench**: Carpeta drivers/.
 - En **Pentaho Server**: Carpeta tomcat/lib/.
- **Nota:** Sin este archivo, Pentaho no puede "hablar" con Neon.

2.2 Conectividad Segura (SSL)

Neon requiere conexiones encriptadas. La configuración JDBC estándar fallará si no se especifican los parámetros de seguridad.

- **Cadena de Conexión:** jdbc:postgresql://[endpoint-neon]:5432/neondb
- **Parámetros Obligatorios:**
 - ssl: true
 - sslmode: require

3. Diseño del Cubo OLAP (Pentaho Schema Workbench)

El corazón de la solución es el esquema XML diseñado en **Schema Workbench**. Este archivo actúa como un "traductor" que convierte consultas de negocio multidimensionales en SQL plano.

3.1 Definición del Cubo "Ventas"

El cubo es el objeto lógico que agrupa medidas y dimensiones.

- **Tabla Base:** Se mapeó a fact_ventas (esquema public).

3.2 Definición de Métricas (Measures)

Son los valores cuantitativos que analizamos.

- **Cantidad:**
 - *Columna*: cantidad.
 - *Agregador*: sum (Suma total).
 - *Formato*: #,### (Enteros con separador de miles).
- **Ventas Totales:**
 - *Columna*: total_dinero.
 - *Agregador*: sum.

3.3 Definición de Dimensiones y Jerarquías

Aquí definimos "cómo" queremos ver los datos.

Dimensión: Producto

- **Foreign Key:** producto_id (en tabla de hechos).
- **Jerarquía:**
 - **Primary Key:** id (en tabla dim_producto). *Crucial para el JOIN correcto.*
 - **Nivel 1 (Superior):** Categoría (Agrupador general).
 - **Nivel 2 (Detalle):** Nombre (Producto específico).

Dimensión: Geografía

- **Foreign Key:** pais_id (en tabla de hechos).
- **Jerarquía:**
 - **Primary Key:** id (en tabla dim_pais).
 - **Nivel 1:** Región (Continente/Zona).
 - **Nivel 2:** País (Ubicación específica).

Nota técnica: Se eliminó la dimensión de Tiempo para asegurar la estabilidad inicial del cubo y evitar conflictos de tipos de datos entre Mondrian y PostgreSQL.

4. Lenguaje de Consulta: MDX (MultiDimensional eXpressions)

Para extraer datos del cubo, no usamos SQL, sino **MDX**. MDX entiende de ejes (filas y columnas) y coordenadas multidimensionales.

4.1 Estructura de una Consulta MDX

```
SELECT
    { [Eje_Columnas] } ON COLUMNS,
    { [Eje_Filas] } ON ROWS
FROM [Nombre_Del_Cubo]
WHERE ( [Filtros_Opcionales] )
```

4.2 Consultas Implementadas y sus Traducciones

A. Análisis Básico por Categoría

Esta consulta verifica el volumen de ventas agrupado por tipo de producto.

```
SELECT
    { [Measures]. [Cantidad] } ON COLUMNS,
    { [Producto]. [Categoría]. Members } ON ROWS
FROM [Ventas]
```

Traducción al Negocio: "Muéstrame la suma total de **unidades vendidas (Cantidad)**, desglosada por cada **Categoría** de productos que tenemos en el catálogo."

B. Matriz Cruzada (Cross Join)

Cruza dos dimensiones para ver intersecciones. Es útil para ver el rendimiento de productos en

diferentes zonas.

```
SELECT
    { [Measures]. [Cantidad] } ON COLUMNS,
    { [Geografia]. [Region]. Members } ON ROWS,
    { [Producto]. [Categoria]. Members } ON PAGES -- O anidado en ROWS
FROM [Ventas]
```

Traducción al Negocio: "Quiero ver una tabla cruzada que me diga **cuántas unidades** se vendieron de cada **Categoría**, separadas por **Región geográfica**."

C. Drill-Down (Navegación al detalle)

Baja un nivel en la jerarquía, de la Región general al País específico, mostrando métricas monetarias y de volumen.

```
SELECT
    { [Measures]. [Cantidad], [Measures]. [Ventas Totales] } ON COLUMNS,
    { [Geografia]. [Pais]. Members } ON ROWS
FROM [Ventas]
```

Traducción al Negocio: "Dame un informe detallado **país por país**, que muestre dos indicadores: el **volumen de unidades** y el **dinero total recaudado**."

D. Top 3 Productos (Ranking)

Utiliza funciones avanzadas de MDX para filtrar los resultados.

```
SELECT
    { [Measures]. [Cantidad] } ON COLUMNS,
    TopCount( { [Producto]. [Nombre]. Members }, 3, [Measures]. [Cantidad] ) ON
ROWS
FROM [Ventas]
```

Traducción al Negocio: "Ignora el resto y muéstrame únicamente el **Top 3 de productos** específicos (por nombre) que han vendido más **cantidad** de unidades."

5. Conclusión y Aprendizajes

1. **Tipos de Datos:** La consistencia entre el tipo de dato en la BD (Integer/Numeric) y la definición en Pentaho es vital. Los errores Integer = Varchar son comunes si no se mapean bien las claves primarias (primaryKey).
2. **Seguridad:** La nube moderna (Neon) exige SSL, algo que las herramientas antiguas (Pentaho) no traen activo por defecto.
3. **Simplicidad:** Un diseño de esquema estrella limpio en SQL facilita enormemente el diseño del cubo XML.