**Documentación del Sistema**

**Proyecto de Bases de Datos Avanzado**

**Google Cloud Spanner**

**Integrantes:**

* Rogwi Cajas
* George Henriquez
* Isaac Solis

**Objetivos del Proyecto:**

Crear el modelo para el dataset escogido mediante kylin para poder realizar la configuración del cubo.

Crear y configurar el cubo en kylin en kylin para poder generar reportes con conocimiento relevante usando la información almacenada en el cubo.

Generar 3 reportes exitosos mediante el cubo y almacenar los resultados en archivos .csv individuales. Los reportes a generar son: Monto de Impuestos pagados por perdiodo, Dinero invertido en descuentos por periodo (fiscal quarter), Total de ventas generadas por pais.

**Descripción del dataset**

El dataset utilizado para este proyecto cuenta con 2 hechos: FactInternetSales que almacena información detallada de las ventas hechas por internet como por ejemplo monto, impuestos, costos etc. y FactInternetSalesReason que almacena la razón de compra, numero de orden y número de serie.

Las dimensiones consideradas en el dataset son:

DimProduct: Datos de cada producto único.

DimDate: Datos especificos de la fecha de compra incluidos periodos fiscales.

DimPromotion: Datos relacionados a descuentos, inicio y fin del periodo de descuento.

DimProductSubcategory: subcategoría del producto.

DimProductCategory: categoría del producto.

DimCurrency: Concurrencia utilizada en la compra.

DimCustomer: Datos personales del cliente que realiza la compra.

DimSalesTerritory: Datos de la ubicación geográfica donde se realizó la compra.

DimGeography: Daos de la ubicación geográfica más generales.

**NOTA**: las dimensiones cuentan con campos repetidos, pero en distintos idiomas.

**Instalación de Apache Kylin**

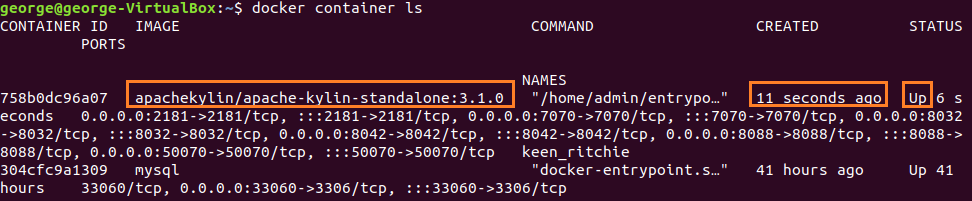
Para la instalación de apache Kylin se utilizó una imagen de Docker para Kylin, en la cual se incluye Hadoop, Hive, Spark, Kafka, MySQL y Zoopkeeper. El proceso de instalación seguido fue el siguiente:

1. Con Docker instalado, descargar la imagen de Apache Kylin, la versión utilizada en este proyecto es la siguiente: “docker pull apachekylin/apache-kylin-standalone:3.1.0
2. Para iniciar el contenedor se utiliza el comando mostrado a continuación:

**Docker run –d –p 7070:7070 –p 8088:8088 –p 50070:50070 –p 8032:8032 –p 8042:8042 –p 2181:2181 apachekylin/apache-kylin-standalone: 3.1.0 758b0dc96a077cf4df75a557117934a38a728f649850cb20708f6033b0cafb0d**

****

1. Para ver si el contenedor se ha ejecutado correctamente se ejecuta: “docker container ls”

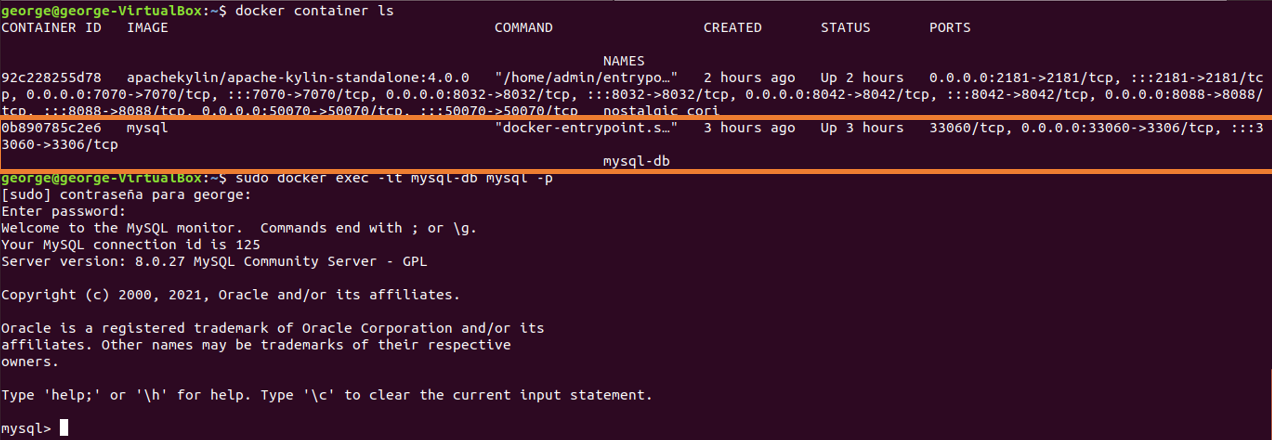


1. Una vez iniciado el contenedor, se accede al servicio de Kylin desde el navegador web ingresando “http://127.0.0.1:7070/kylin/login”

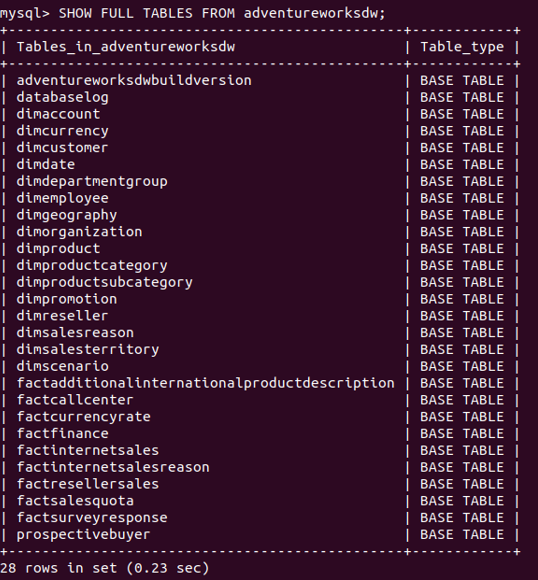
**Importación de datos**

Para realizar la carga de datos se utilizó MySQL, el cual se instaló en un contenedor de Docker, mientras que para la importación de datos hacia hadoop se usó la herramienta sqoop. A continuación, se muestras el proceso realizado:

1. Se cargan los datos en una base de datos MySQL en un contenedor de Docker.

****

1. Con el comando ‘show full tables from adventureworkdw’ dentro de la consola de MySQL se listan todas las tablas de la base de datos.



**Creación del modelo**

Captura de pantalla de computadora

Descripción generada automáticamente

Una captura de pantalla de una computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

**Creación del Cubo**

Una captura de pantalla de una computadora

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Una captura de pantalla de una computadora

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Reportes**