

*Ejercicio de Laboratorio 3: Hive y Beeline*

Instituto Politécnico Nacional.

Escuela Superior de Cómputo.

Licenciatura en ciencia de datos.

Bases de Datos Avanzadas

Emiliano López Méndez.

**Introduccion**

En este ejercicio de laboratorio, se utiliza Apache Hive para crear, poblar y consultar tablas a través del cliente Beeline, ejecutándose en un entorno Docker. Hive es una herramienta clave en el ecosistema de Big Data que permite gestionar grandes volúmenes de datos estructurados y semiestructurados, permitiendo realizar consultas SQL sobre estos datos almacenados en HDFS o en sistemas de archivos locales. A través de este ejercicio, se exploran las funcionalidades básicas de Hive, incluyendo la creación de tablas, la inserción de datos y la realización de consultas relacionales.

**Tecnologías empleadas:**

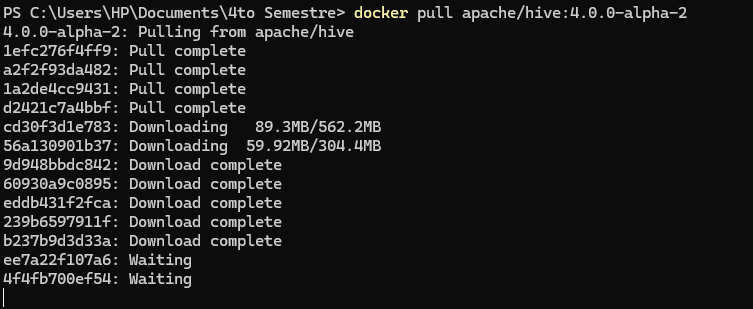
* **Apache Hive**: Es un sistema de data warehouse que permite la consulta y gestión de grandes conjuntos de datos almacenados en sistemas distribuidos como HDFS.
* **Beeline**: Es un cliente de línea de comandos utilizado para conectarse a Hive a través de JDBC. Beeline permite ejecutar consultas SQL de forma interactiva y es ampliamente utilizado para acceder a metadatos y datos almacenados en tablas de Hive.
* **Docker**: Se utiliza para crear un entorno aislado y reproducible en el que se despliega Apache Hive y Beeline.
* **PostgreSQL/MySQL** (opcional para Metastore): En muchos casos, Hive utiliza una base de datos como PostgreSQL o MySQL para almacenar los metadatos de las tablas (Metastore).

**Desarrollo de la Actividad:**

**Creando una base de datos con Apache Hive**

Empezamos descargando una imagen con este comando A black background with white text

Description automatically generated



A computer screen with white text

Description automatically generated

Una vez ya hecha sacamos nuestra versión de hive y lo configuramos para estar embudada con MetastoreA computer screen with text and numbers

Description automatically generated

Y nos conectamos a Beeline

A screen shot of a computer

Description automatically generated

Luego vamos a crear una tabla como ejemplo A computer screen shot of a computer program

Description automatically generated

Como podemos ver no tenemos ninguna tabla por el momento

Con este comando creamos la tabla hive\_example

A computer screen shot of white text

Description automatically generated

Y con los próximos comandos vamos a alterar la tabla e insertar datos para demostrar que la tabla este poblada

A computer screen with white text

Description automatically generated

A screen shot of a computer

Description automatically generated

Como podemos ver ya creamos la tabla hive\_example A computer screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

Vemos al tabla poblada

***¿Es posible crear una tabla a pesar de que no se definió una base de datos como tal?***

Sí, es posible crear una tabla en Apache Hive sin haber definido explícitamente una base de datos, porque Hive utiliza una base de datos predeterminada llamada **default** si no se especifica otra. De hecho, cuando no defines una base de datos con un comando USE, cualquier tabla que crees será almacenada dentro de esta base de datos predeterminada.

**Parte 2 Creando dos tablas pobladas con relacion**

**Creando primera tabla departamentos**

Empezamos creando una tabla de departamentos depa\_id y depa\_name en donde depa\_id será nuestra primary key.

 A screen shot of a computer program

Description automatically generated

Luego la vamos a poblar:



A screen shot of a computer

Description automatically generated

Aquí vemos que si se poblo la tabla de deparamentos.

**Creando segunda tabla empleados**

Luego vamos a crear una tabla de empleados



A black screen with white text

Description automatically generated

Y poblamos la tabla

A screenshot of a computer screen

Description automatically generated

**A unir las dos tablas en base a sus llaves primarias y foráneas**

Mediante esta consulta de join



Logrameos unir las dos tablas

A screenshot of a computer screen

Description automatically generated

**Esquema relacional de las tablas creadas**

A screenshot of a computer

Description automatically generated

**Parte 3 Investigacion**

Como se le podría dar persistencia al servidor?

* **Mediante la persistencia de volúmenes en Docker**: Para asegurar que los datos y configuraciones del servidor persistan incluso después de reiniciar o detener el contenedor Docker, es necesario utilizar **volúmenes de Docker**. Los volúmenes permiten almacenar los datos en el sistema de archivos del host, lo que garantiza que no se pierdan cuando el contenedor se reinicie o elimine.
* **Metastore**: Si el servidor utiliza una base de datos (como PostgreSQL o MySQL) para almacenar los metadatos de las tablas, es importante montar un volumen para persistir los datos de la base de datos.
* **Archivos de datos**: En caso de que el servidor maneje archivos de datos localmente (por ejemplo, tablas en formato de archivos), es fundamental crear un volumen para el directorio donde se almacenan estos datos.
* **Uso de HDFS** (si aplica): En entornos distribuidos como Hadoop, el sistema de archivos HDFS también puede beneficiarse de volúmenes Docker para mantener persistencia en los nodos de NameNode y DataNode.

**Conclusión**

En este ejercicio de laboratorio, se utiliza Apache Hive para crear, poblar y consultar tablas a través del cliente Beeline, ejecutándose en un entorno Docker. Hive es una herramienta clave en el ecosistema de Big Data que permite gestionar grandes volúmenes de datos estructurados y semiestructurados, permitiendo realizar consultas SQL sobre estos datos almacenados en HDFS o en sistemas de archivos locales. A través de este ejercicio, se exploran las funcionalidades básicas de Hive, incluyendo la creación de tablas, la inserción de datos y la realización de consultas relacionales.