Componente Apache Spark.

En esta práctica se trabajará con la herramienta Apache Spark.

Se pide:

- Conectar con la base de datos de mariadb y recuperar las tablas USER y PROFILE en dos dataframes. Posteriormente, realizar un join con la información de las dos tablas (**movies_profile_dataframe**).
- Realizar un diseño de base de datos NO relacional de un e-commerce como Amazon. Utilizar como base de datos no relacional MongoDB, después de poblarla, conectarse desde dentro de Apache Spark a esta base de datos y persistir los datos en un dataframe. Es necesario comprobar que se han almacenado los datos (**non_relational_db**).
    - Para el diseño tener en cuenta que en un e-commerce hay productos, clientes, vendedores, administradores, pedidos, carritos en los cuales se efectúan pagos, devoluciones de compra, centro de mensajes para la comunicación entre clientes y vendedores, etc.
    - Cada uno de los entes tienen múltiples características y entre todos ellos hay también múltiples relaciones. Crear un diseño lo más correcto posible conceptualmente y eficiente. IMPORTANTE justificar dicho diseño.
- Utilizando Delta Lake haz lo siguiente (**delta_lake_1**):
    - Persistir los dataframes del apartado anterior en un Delta Lake
    - Realizar una lectura (cargar los datos desde Delta) y una modificacion de esos datos utilizando SQL. Realizar estas mismas opciones sin utilizar SQL, sino con la API de Delta Lake.
    - Realizar una consulta sobre una snapshot de una de las tablas (investigar el concepto de time travel)
    - Haciendo uso del fichero bing_covid-19_data.parquet crear una tabla y rellenarla con esos datos.
- Utilizando Delta Lake haz lo siguiente (**delta_lake_2**):
    - Haciendo uso del fichero salesdata.zip y habiendo analizado el concepto de Arquitectura Medallion previamente:
        - Capa Bronze
            - Carga en un DataFrame todos los datos de ventas. Para ello, descomprime el zip en tu ordenador, y sube los archivo a una carpeta ventas de HDFS.
            - El primer paso será renombrar la columnas para eliminar los espacios en blanco.
            - Añade una columna ingestion_time con la fecha actual, y otra columna source_system donde le asignes el valor literal carga inicial CSV.
            - Almacena el DataFrame en /delta/bronze/sales/ con formato Delta.
        - Capa Silver
            - Elimina las filas que contengan algún campo nulo.
            - Comprueba si las cabeceras de los archivos aparecen como datos del dataset (por ejemplo, un producto cuyo nombre sea Product). Si fuera el caso, elimina dichas filas.
            - A partir del campo dirección, crea dos nuevas columnas para almacenar la ciudad (City) y el estado (State). Por ejemplo, para la

dirección 136 Church St, New York City, NY 10001, la ciudad es New York City y el estado es NY.
    - Modifica el campo con la fecha del pedido para que su formato sea timestamp.
    - Sobre el campo anterior, crea dos nuevas columnas, con el mes (Month) y el año (Year) del pedido.
    - Almacena el resultado en formato Delta en /delta/silver/sales/ particionando los datos por año.
- Capa Gold
    - Realizar una consulta que calcule la cantidad total recaudada cada mes durante el año 2019 y almacenando sus resultados en formato Delta en /delta/gold/sales/ventas2019 y también como tabla con el nombre ventas2029_delta.
    - Realizar una consulta que realice un gráfico con las diez ciudades que más unidades han vendido y almacenando sus resultados en formato Delta en /delta/gold/sales/top10ciudades y también como tabla con el nombre top10ciudades_delta.
- Tras una auditoria, se ha descubierto que había un libro oculto de contabilidad con ciertas ventas que no habían sido registradas en el sistema. Así pues, crea datos ficticios extra de ventas para el año 2019, colócalo en la capa Bronze, y a continuación, actualiza los datos del resto de capas, fusionando los nuevos datos con los ya existentes
- Obtén el histórico de la capa Bronze y las tablas Gold. Recupera la primera versión de la tabla que obtiene la recaudación mensual de 2019 y compara su contenido con el actual.

Material de apoyo:

- https://aitor-medrano.github.io/iabd/spark/spark.html

A entregar:

- Los scripts correspondientes de Apache Spark con el nombre que está entre parentesis en negrita en cada apartado.
- Un fichero de TEXTO PLANO con los comandos a ejecutar los scripts de Spark.
- Todo lo anterior se debe de comprimir en un fichero (.zip) llamado IAB_BIU_Tema_03_Practica_05_03_Nombre_Apellidos (sin tildes, ni ñ ni espacios). Si el grupo lo forma más de una persona poner:
    - IAB_BIU_Tema_03_Practica_05_03_Nombre1_Apellidos1_Nombre2_Apellidos2