Prueba técnica: Data Engineer

Sección 1: Data pipeline

Objetivo: Crear un data pipeline con las herramientas disponibles por el usuario

Los ejercicios de programación tienen que incluir los procedimientos de instalación y ejecución de las herramientas a utilizar y los scripts que realizaran los procedimientos. Se puede realizar a través de Dockers. Pueden incluir pruebas unitarias o de integración. Se puede compartir por Github o cualquier repositorio o en un zip.

Nota: Junto con esta guía te compartimos 1 data set con la información sobre las compras de dos compañías ficticias que procesan con nosotros.

1.1 Carga de información

La información proporcionada se debe de cargar en alguna base de datos. Puede ser estructurada o no estructurada. Ejemplo: MySQL, Postgres, MongoDB, etc.

Incluye comentarios del por que elegiste ese tipo de base de datos

1.2 Extracción

Se debe de realizar un procedimiento de extracción de la información anterior por medio de algún lenguaje de programación que permita procesarlo. El formato final de la información extraída puede ser CSV, Avro, parquet o el que se considere más adecuado.

Agrega comentarios acerca del por qué tuviste que utilizar el lenguaje y el formato que elegiste. También platicamos si te encontraste con algún reto a la hora de extraer la información.

1.3 Transformación

Se propone el siguiente esquema para la información

Cargo
id varchar(24 NOT NULL
company_name varchar(130) NULL
company_id varchar(24) NOT NULL
amount decimal(16,2) NOT NULL
status marchar(30) NOT NULL
created_at timestamp NOT NULL
updated_at timestamp NULL

Realiza las transformaciones necesarias para que la información extraída cumpla con el esquema. Puedes realizarlas con el lenguaje de programación de tu preferencia.

Incluye comentarios acerca de que transformaciones tuviste que realizar y que retos te encontraste en la implementación de estos mecanismos de transformación.

1.4 Dispersión de la información

Se debe de utilizar una base de datos Postgres. En esta base se va a crear un esquema estructurado basado en el ejercicio anterior pero debemos de crear una tabla llamada charges donde tendremos la información de las transacciones y otra llamada companies donde incluiremos la información de las compañias. Estas tablas deberán de estar relacionadas. Cargaremos la información del dataset en estas dos tablas.

Incluye el diagrama de base de datos resultado de este ejercicio.

1.5 SQL

Diseña una vista en la base de datos Postgres de las tablas donde cargamos la información transformada para que podamos ver el monto total transaccionado por día para las diferentes compañías

Sección 2: Scala

Objetivo: Implementar una aplicación en Scala

Problema: Calcular el numero faltante de un conjunto de los primeros 100 números naturales del cual se extrajo uno.

Especificaciones:
- La aplicación debe de implementarse en el lenguaje Scala
- Se debe de implementar una clase que represente al conjunto de los primero 100 números
- La clase implementada debe de tener el método Extract para extraer un cierto numero deseado
- La clase implementada debe de poder calcular que numero se extrajo y presentarlo
- Debe de incluir validación del input de datos (numero, numero menor de 100)
- La aplicación debe de poder ejecutarse con un argumento introducido por el usuario que haga uso de nuestra clase y muestre que pudo calcular que se extrajo ese número