Prueba técnica Ingeniero de datos - Puntored

Duración: 24 horas a partir de la fecha de envío

Criterios de Evaluación

- Precisión de las respuestas teóricas
- Claridad y eficiencia del código SQL
- Estructura y documentación del código Python

Modo de entrega

Cargue los archivos pertinentes al desarrollo de la prueba a un repositorio público de GitHub y comparta la URL. (15 puntos)

Sección 1: Preguntas Teóricas (12 puntos)

Python (4 puntos)

1. ¿Cuál es la diferencia entre una lista y un conjunto (set) en Python? Proporcione un ejemplo (1 puntos)
2. ¿Qué es un generator en Python y en qué casos es útil? Proporcione un ejemplo de implementación. (1 puntos)
3. ¿Qué ventajas ofrece Pandas sobre las estructuras de datos nativas de Python para el análisis de datos? (1 puntos)
4. ¿Cuál es la diferencia entre apply() y map() en Pandas? Proporcione un ejemplo. (1 puntos)

SQL (4 puntos)

1. Dado un esquema de base de datos con las siguientes tablas

   ```
   empleados (id, nombre, departamento_id, salario,
   fecha_contratación)

   departamentos (id, nombre)
   ```

   Escriba una consulta para obtener el salario promedio de cada departamento, incluyendo el nombre del departamento. (1 puntos)

2. ¿Cuál es la diferencia entre INNER JOIN, LEFT JOIN y FULL JOIN?. Da un ejemplo de cada uno. (1 puntos)
3. ¿Cómo optimizarías una consulta en una base de datos con millones de registros? (1 puntos)
4. ¿Qué es la cláusula HAVING en SQL y en qué se diferencia de WHERE? (1 puntos)

Amazon Web Services (4 puntos)

1. ¿Cuál es la diferencia entre Amazon S3, Amazon RDS y Amazon Redshift? (1 puntos)
2. ¿Cuándo usarías Amazon DynamoDB en lugar de Amazon RDS o Amazon Redshift? (1 puntos)
3. ¿Cuáles son las diferencias entre AWS Lambda y AWS EC2 para ejecutar cargas de trabajo? (1 puntos)
4. ¿Cómo implementarías un mecanismo seguro para que un servicio en AWS acceda a un bucket de S3 sin usar claves de acceso en el código? (1 puntos)

Sección 2: Prueba práctica SQL (50 puntos)

Tienes las siguientes tablas en una base de datos:

clientes

```
id (INT, PRIMARY KEY)

nombre (VARCHAR)

apellido (VARCHAR)
```

ventas

```
id (INT, PRIMARY KEY)

cliente_id (INT, FOREIGN KEY a clientes.id)

producto (VARCHAR)

fecha (DATE)

monto (DECIMAL)
```

Escribe una consulta para obtener los 5 clientes con mayor monto total de ventas en los últimos 6 meses. (5 puntos)

Escribe una consulta para calcular el ticket promedio de ventas por cliente en el último año. (5 puntos)

Escribe una consulta para obtener el nombre completo de los clientes y su monto total de ventas. (10 puntos)

Escribe una consulta para obtener el ingreso promedio de ventas por mes. (10 puntos)

Escribe una consulta para calcular el ranking de clientes por ventas en el último año. (10 puntos)

Escribe una consulta para calcular el total de ventas por cliente y luego selecciona solo los clientes cuyo total de ventas es superior al promedio general. (10 puntos)

Sección 3: Prueba práctica python y AWS (50 puntos)

Una empresa requiere disponibilizar a través de un API la información de ventas de sus clientes a 3 diferentes proveedores donde solo se encuentre la información de los productos asociados a cada proveedor. La información que necesitan los proveedores es la siguiente:

- cantidad de transacciones por cliente por dia
- monto total ($) por cliente por día

Nota: Para el ejercicio suponga que la información se encuentra en una base de datos alojada en un RDS de AWS que tiene la estructura de datos expuesta en la sección 2 de la prueba y que existen solamente 3 productos (1 por cada proveedor).

1. Diseñe un pipeline para extraer los datos de la base y disponibilizar la información que requiere cada proveedor en **batch diario** usando los servicios de AWS. (15 puntos)
2. Diseñe un pipeline para extraer los datos de la base y disponibilizar la información que requiere cada proveedor en **tiempo real** usando los servicios de AWS. (15 puntos)
3. Script en python para extraer la información de la base de datos en el formato que requiere el proveedor. (10 puntos)
4. Script en python para disponibilizar la información en el API en formato JSON, Implementar logs y manejo de errores (10 puntos)

Preguntas rápidas:

- Windows o Linux?
- MySQL o PostgreSQL?
- Batch processing o streaming?
- ETL o ELT?
- Parquet o CSV?
- Spark o Pandas?