Part 2: Theoretical Questions

Python & Data Engineering

What are Python best practices for building production-grade ETL scripts?

- **Modularidad**: Dividiendo el código en funciones y módulos que puedan reutilizarse fácilmente.
- **Manejo de errores**: debe ser cuidadoso, con registros (logging) detallados y excepciones bien controladas para facilitar la detección de problemas.
- **Validación de datos**: comprobando que no existan valores nulos, que los formatos sean correctos y que la información sea consistente.
- **Escalabilidad**: se logra mejor mediante el procesamiento por lotes.
- **Seguridad**: las credenciales y datos sensibles deben gestionarse mediante variables de entorno, nunca en el código.
- **Testing**: las pruebas unitarias son la garantía de que el sistema es robusto y funciona como se espera, incluso cuando algo cambia en el entorno o en la fuente de datos.

How would you handle schema evolution in incoming datasets?

En un entorno de datos profesional, es fundamental mantener un control de versiones sobre los esquemas. Esto permite saber qué cambió, cuándo y por qué, evitando incompatibilidades inesperadas. Herramientas como Apache Atlas o el registro de esquemas en metadatos facilitan este seguimiento y la trazabilidad de la información.

La transformación de datos también debe ser adaptable. Un buen ETL incorpora lógica capaz de mapear campos antiguos a sus equivalentes actuales o aplicar valores por defecto cuando un dato no está disponible. Esta flexibilidad asegura que las integraciones sigan funcionando incluso cuando la estructura de los datos evoluciona con el tiempo.

Data Modeling

Compare star and snowflake schemas. When would you use each?

En el modelado de datos para entornos analíticos, dos de los esquemas más utilizados son el de estrella y el de copo de nieve. El esquema estrella se caracteriza por una tabla de hechos central que se conecta a dimensiones desnormalizadas. Su principal ventaja es la rapidez en las consultas y la simplicidad en el diseño, lo que lo hace ideal para entornos de Business Intelligence donde se necesitan consultas directas y de fácil interpretación.

Por otro lado, el esquema copo de nieve organiza las dimensiones en un formato más normalizado, distribuyendo la información en múltiples tablas relacionadas. Esto permite ahorrar espacio y mantener una mayor consistencia en datos con jerarquías complejas. Es especialmente útil cuando el almacenamiento es limitado o cuando las relaciones entre datos requieren una estructura más detallada.

How would you design a schema for time-series data?

Para manejar datos basados en el tiempo de forma eficiente, es común utilizar una tabla calendario que actúe como referencia, ya sea con un campo de timestamp como clave primaria o con particiones organizadas por periodos de tiempo. La optimización de este tipo de tablas suele incluir el particionamiento por intervalos para mejorar el rendimiento, así como la creación de índices en las columnas de tiempo, lo que acelera considerablemente las consultas.

Cloud Data Infrastructure

Describe the benefits of using S3 as a data lake.

- **Escalabilidad**: Permite manejar volúmenes prácticamente ilimitados con un costo relativamente bajo.
- **Durabilidad**: Asegura alta disponibilidad y redundancia
- **Flexibilidad**: Soporta múltiples formatos de datos, como CSV, JSON o Parquet, y al adaptarse a distintos esquemas según la necesidad del proyecto**Integración**:
- **Compatibilidad:** Con herramientas de análisis y procesamiento, facilita la explotación de los datos sin complicaciones adicionales
- **Seguridad**: Control de acceso granular mediante IAM y encriptación de la información.

What's the difference between using Lambda vs EC2 for data tasks?

En AWS, la elección entre Lambda y EC2 depende del tipo de procesamiento que necesites. Lambda ofrece un enfoque serverless, lo que significa que puedes ejecutar código sin preocuparte por la infraestructura subyacente. Es ideal para procesos ligeros y orientados a eventos, como triggers o tareas rápidas, aunque tiene limitaciones de tiempo de ejecución, alrededor de 15 minutos, y memoria disponible.
Por su parte, EC2 proporciona servidores virtuales que te dan control total sobre el sistema operativo y los recursos. Es más adecuado para procesos pesados o continuos, como clusters de Spark, y resulta más económico cuando se manejan cargas estables a largo plazo, en comparación con Lambda, que puede ser más costoso en picos de uso.

Workflow Orchestration

What are the advantages of using a tool like Airflow?

En la orquestación de pipelines de datos, muchas herramientas permiten una programación dirigida por código, representando los flujos como DAGs en Python. Esto facilita definir con claridad las dependencias entre tareas y mantener el control del proceso completo. Además, ofrecen interfaces visuales que permiten monitorear la ejecución de cada pipeline en tiempo real, lo que ayuda a identificar y resolver problemas rápidamente.

La escalabilidad se logra mediante ejecutores distribuidos que permiten manejar grandes volúmenes de datos sin afectar el rendimiento. Estas plataformas también incluyen mecanismos nativos de reintentos y alertas para manejar fallos de manera eficiente. Por último, cuentan con integraciones listas para conectar bases de datos, servicios en la nube y otras fuentes de datos, simplificando la construcción de flujos complejos de manera confiable.

How would you implement retries and alerting in a DAG?

El manejo de errores en pipelines de datos incluye tanto reintentos como alertas. Los reintentos se pueden configurar usando parámetros como retries y retry_delay en los operadores, definiendo además condiciones de fallo específicas mediante on_failure_callback.

En cuanto a las alertas, es recomendable enviar notificaciones por correo electrónico o Slack cada vez que ocurre un error, también a través de on_failure_callback. Para situaciones críticas, se puede integrar con herramientas como PagerDuty, mientras que un sistema de logging centralizado, por ejemplo en CloudWatch, permite hacer seguimiento detallado de los fallos y facilita la resolución de problemas de manera más ágil.