# ☑ 1. Snowflake — 50 preguntas

### 1. Explica la arquitectura de Snowflake y en qué se diferencia del almacenamiento tradicional en la nube

Snowflake tiene **tres capas**:

1. **Almacenamiento**: Datos centralizados en formato columnar en S3, Azure Blob o GCS.
2. **Cómputo**: Virtual Warehouses independientes que ejecutan consultas.
3. **Servicios**: Seguridad, optimización de queries, transacciones, metadata.

**Diferencia:** A diferencia de un almacenamiento tradicional, donde cómputo y almacenamiento están acoplados, Snowflake permite escalar cada capa de forma independiente.

*Ejemplo:* Un warehouse pequeño puede ejecutar consultas ligeras mientras otro grande procesa cargas masivas, ambos usando los mismos datos.

---

### 2. ¿Qué es un Virtual Warehouse y cómo funciona el escalado automático?

- Es un **clúster de cómputo virtual** que ejecuta consultas SQL.
- **Escalado automático:** Snowflake puede aumentar o disminuir nodos según la carga.

*Ejemplo:* Si llega una consulta pesada, Snowflake añade nodos automáticamente y los reduce cuando no se necesitan.

---

### 3. ¿Qué son los micro-partitions y cómo afectan el rendimiento?

- Son **segmentos internos de datos** (16 MB a 512 MB) creados automáticamente al cargar datos.
- Mejoran el rendimiento porque las consultas solo leen micro-partitions relevantes.

*Ejemplo:* Una búsqueda por año solo accede a micro-partitions de ese año, acelerando la consulta.

---

### 4. Explica Time Travel y Fail-safe

- **Time Travel:** Permite recuperar datos eliminados o modificados hasta 1-90 días.
- **Fail-safe:** Protección adicional de 7 días para recuperación ante fallos graves.

*Ejemplo:* Se borró una tabla por error, Time Travel permite restaurarla fácilmente.

---

## 5. ¿Qué es un Cloning y cuáles son sus ventajas?

- Es una **copia instantánea de tablas o bases de datos** sin duplicar datos físicamente.
- **Ventajas:** Pruebas y desarrollo sin ocupar espacio adicional.

*Ejemplo:* Crear un entorno de QA para probar ETL sin duplicar la base de producción.

---

## 6. ¿Cómo funciona el caching en Snowflake?

- **Query result cache:** Guarda resultados por 24 horas.
- **Local disk cache:** Datos recientes en warehouses.
- **Metadata cache:** Información de micro-partitions para optimizar consultas.

*Ejemplo:* Ejecutar la misma consulta varias veces devuelve resultados instantáneamente gracias al cache.

---

## 7. Describe el funcionamiento del Query Optimizer de Snowflake

- Usa **cost-based optimization**, analiza costos de lectura, joins y filtros, y elige el plan más eficiente.
- No necesita índices tradicionales, Snowflake optimiza automáticamente las consultas.

---

## 8. Tipos de tablas y cuándo usar cada una

- **Permanent:** Duraderas, soportan Time Travel y Fail-safe.
- **Transient:** Sin Fail-safe, menor costo; útil para ETL intermedio.
- **Temporary:** Solo dura durante la sesión; ideal para cálculos temporales.

---

## 9. ¿Qué es la separación de cómputo y almacenamiento en Snowflake?

- El almacenamiento está centralizado y los warehouses (cómputo) son independientes.
- Permite **escalar cómputo sin afectar almacenamiento**, y múltiples warehouses pueden usar los mismos datos.

---

## 10. Explica el proceso de carga de datos usando Snowpipe

- Snowpipe ingiere datos automáticamente desde **S3/GCS/Azure**.
- Puede activarse mediante **eventos** o APIs.

*Ejemplo:* Archivos CSV que llegan cada minuto se cargan automáticamente en Snowflake.

---

## 11. Diferencias entre Snowpipe y COPY INTO

- **COPY INTO:** Carga manual o programada de lotes.
- **Snowpipe:** Ingesta continua, casi en tiempo real, automatizada.

---

## 12. ¿Cómo funciona la ingesta continua (streaming) en Snowflake?

- Snowpipe y **Streams + Tasks** permiten replicar datos continuamente, detectando cambios y aplicándolos automáticamente.

---

## 13. ¿Qué son Streams y Tasks?

- **Stream:** Rastrea cambios en una tabla (insert, update, delete).
- **Task:** Ejecuta consultas de forma programada o basada en streams.

*Ejemplo:* Stream sobre ventas y Task que actualiza dashboards cada hora.

---

## 14. Patrón CDC en Snowflake usando streams

- Se usan **streams** para detectar cambios y **tasks** para aplicarlos en un Data Mart.

*Ejemplo:* Nuevos pedidos se capturan y se agregan automáticamente a un reporte de ventas.

---

## 15. ¿Cómo se configuran roles y permisos en Snowflake?

- Snowflake usa **RBAC (Role-Based Access Control)**.
- Roles jerárquicos se asignan a usuarios.

*Ejemplo:* ANALYST puede leer datos, DEVELOPER puede crear tablas.

---

## 16. ¿Qué es un Resource Monitor?

- Monitorea y controla **uso de créditos de warehouses**.
- Puede suspender warehouses si se excede un límite.

---

## 17. ¿Cómo se implementa Row Level Security?

- Con **Secure Views** y políticas de filtrado dinámico.

*Ejemplo:* Vendedores solo ven filas de su región.

---

## 18. ¿Cómo se implementa Masking Policies?

- Oculta datos sensibles dinámicamente según el rol.

*Ejemplo:* Mostrar solo los últimos 4 dígitos de una tarjeta de crédito.

---

## 19. ¿Qué es Dynamic Tables (Materialized Views 2.0)?

- Tablas materializadas que se actualizan automáticamente cuando cambian los datos fuente.

---

## 20. Integración con S3/GCS/Azure

- Snowflake puede **leer y escribir datos** directamente desde estos servicios mediante stages y Snowpipe.

---

## 21. ¿Qué es External Table y cuándo se usa?

- Tabla que apunta a datos externos **sin copiarlos a Snowflake**.
- Útil para datasets muy grandes que no se quieren duplicar.

---

## 22. ¿Qué es Snowpark?

- Librería para procesar datos dentro de Snowflake usando **Python, Java o Scala**.
- Permite ejecutar lógica compleja cerca de los datos.

---

## 23. Ventajas de usar Snowpark con Python

- Transformaciones complejas, análisis vectorizado y ML directamente dentro de Snowflake sin mover los datos.

---

## 24. Diferencias entre UDF, UDTF y Stored Procedures

- **UDF:** Retorna un valor.
- **UDTF:** Retorna una tabla.
- **Stored Procedure:** Bloque de lógica procedural ejecutando múltiples SQL.

---

## 25. ¿Cómo optimizar el costo en Snowflake?

- Usar warehouses pequeños y auto-suspend/resume.
- Evitar duplicar datos, usar clones y stages externos.

---

## 26. ¿Qué es un Query Profile y cómo analizarlo?

- Permite **ver paso a paso cómo se ejecutó una consulta**, identificando cuellos de botella y optimizando rendimiento.

---

## 27. ¿Cómo particiona Snowflake automáticamente?

- Snowflake crea **micro-partitions** según columnas y carga, optimizando la lectura de datos.

---

## 28. ¿Qué es el Search Optimization Service?

- Índices especializados internos que aceleran consultas con filtros frecuentes.

---

## 29. ¿Cómo manejar semi-structured data (JSON/AVRO/PARQUET)?

- Se almacenan en **VARIANT** y se consultan con SQL.

*Ejemplo:* `SELECT data:name FROM my_table;`

---

## 30. ¿Qué son VARIANT, OBJECT y ARRAY?

- **VARIANT:** Contenedor flexible de datos semi-estructurados.
- **OBJECT:** Estructura clave-valor, como JSON.
- **ARRAY:** Colección ordenada de valores.

---

## 31. Explica COPY INTO LOCATION

- Exporta datos de Snowflake a **S3, GCS o Azure Blob**.

*Ejemplo:* `COPY INTO @my_s3_stage FROM my_table;`

---

## 32. ¿Qué significa ACID dentro de Snowflake?

- Garantiza **Atomicidad, Consistencia, Aislamiento y Durabilidad** en todas las transacciones.

---

## 33. Caso de uso para Materialized Views

- Acelerar consultas frecuentes en tablas grandes, como agregados diarios de ventas.

---

### 34. Query result caching vs metadata caching

- **Query result cache:** Guarda resultados de consultas.
- **Metadata cache:** Optimiza planificación usando info de micro-partitions.

---

### 35. ¿Cómo funciona Snowflake Marketplace?

- Permite **compartir y consumir datasets de terceros** de forma segura.

---

### 36. ¿Qué es External Functions?

- Funciones que llaman **servicios externos** desde Snowflake, por ejemplo APIs REST.

---

### 37. ¿Cómo se conecta Snowflake con herramientas BI?

- Conectores nativos para Tableau, Power BI, Looker usando **ODBC/JDBC**.

---

### 38. ¿Cómo monitorear un warehouse?

- Mediante **ACCOUNT USAGE**, Resource Monitors y Query Profiles para ver uso de créditos y rendimiento.

---

### 39. ¿Qué es Multi-cluster warehouse?

- Warehouse con varios clusters trabajando simultáneamente para alta concurrencia.

---

### 40. ¿Qué es Auto-suspend y Auto-resume?

- **Auto-suspend:** Pausa warehouse automáticamente cuando no hay consultas.
- **Auto-resume:** Reactiva warehouse automáticamente al recibir nuevas consultas.

---

## 41. ¿Cómo ejecutar pipelines con Tasks?

- Tasks programadas o encadenadas usando streams permiten ejecutar ETL continuo dentro de Snowflake.

---

## 42. ¿Cómo manejar errores de carga?

- Snowflake genera **archivos de error y logs** para COPY INTO o Snowpipe.
- Validación de datos previa ayuda a reducir errores.

---

## 43. Estrategias de partición lógica recomendadas

- Particionar por columnas de alta cardinalidad y filtros frecuentes (fecha, región, cliente).

---

## 44. ¿Cómo manejar datos sensibles?

- Con **Row-level security, masking policies, secure views y stages encriptados**.

---

## 45. Explica Zero-Copy Cloning

- Crear clones instantáneos sin duplicar datos físicamente.
- Útil para entornos de prueba y QA.

---

## 46. ¿Cómo replicar bases de datos entre regiones?

- Usando **Database Replication**, permite disponibilidad global y disaster recovery.

---

## 47. ¿Qué es un Service Account para Snowflake?

- Cuenta para **automatización y ETL**, con permisos limitados y no utilizada por humanos.

---

## 48. ¿Cómo funciona el Data Sharing?

- Compartir datos de forma segura entre cuentas Snowflake **sin moverlos físicamente**.

*Ejemplo:* Una empresa comparte catálogos de productos con proveedores.

---

## 49. ¿Qué limitaciones tiene Snowflake?

- Costo elevado si no se optimiza.
- Limitaciones en transacciones ACID altamente concurrentes.
- No ideal para OLTP extremo con baja latencia.

---

## 50. ¿Qué mejoras implementarías en un entorno grande Snowflake?

- Optimización de warehouses y multi-cluster.
- Uso eficiente de clones y staging externo.
- Activar caching y Search Optimization Service.
- Automatización de pipelines y monitorización con Resource Monitors y Query Profiles.