🟦 A. Ejercicios SQL (5)

1. Top N por categoría (ventanas)

Dada la tabla:

sales(order_id, product_id, category, amount, order_date)

Obtén los **3 productos con mayor venta por categoría**.

✔ Usa `ROW_NUMBER()` o `DENSE_RANK()`.

---

2. Detectar clientes inactivos

Tabla:

events(user_id, event_type, event_time)

Encuentra los usuarios que **no han hecho ningún evento en los últimos 45 días**, pero sí tenían actividad antes de ese periodo.

---

3. Transformación pivot/unpivot

Tabla:

metrics(user_id, metric, value)
Valores de metric: `"clicks"`, `"views"`, `"purchases"`

Genera una tabla:

| user_id | clicks | views | purchases |

---

4. Churn mensual

Tabla:

subscriptions(customer_id, month, active)
`active` es 1 o 0.

Calcula la **tasa de churn** por mes:

```
churn_rate = (# clientes que estaban activos el mes pasado y no están ahora) / (# activos mes pasado)
```

---

5. Detección de duplicados complejos

Tabla:

transactions(tx_id, user_id, amount, timestamp)

Identifica rango de transacciones duplicadas cuando:

- amount es igual
- user_id es igual
- timestamps difieren ≤ 30 segundos

✔ Debes agrupar duplicados y mostrar el grupo.

🟦 B. Ejercicios de Modelado / Arquitectura (3 escenarios)

1. Caso: Plataforma de e-commerce

Diseña un **modelo dimensional** para:

- Usuarios
- Productos
- Carritos
- Pedidos
- Envíos

Incluye:

- **1 fact table principal:** fact_orders
- **SCD2** para dimensión productos
- **SCD1** para usuarios

- Manejo de divisas

Preguntas:

- ¿Cómo manejarías actualizaciones de dirección?
- ¿Cómo modelarías carritos abandonados?

---

2. Caso: Plataforma de streaming (analítica de contenido)

Diseña el **DW en Snowflake** para analizar:

- Reproducciones
- Sesiones
- Busquedas
- Engagement por contenido

Incluye:

- fact_plays
- fact_sessions
- dim_content (SCD2)
- dim_device
- dim_user
- Eventos crudos (staging)

Preguntas:

- ¿Cómo manejarías deduplicación de eventos?
- ¿Cómo definirías “sesión”?
- ¿Qué métricas calcularías?

---

3. Caso: Pipeline ELT en GCP

Diseña un pipeline:

Pub/Sub → Dataflow → BigQuery → Looker

Incluye:

- Esquema de particiones

- Manejo de schema evolution
- Detección de datos corruptos
- Reprocesamiento de particiones

Preguntas:

- ¿Dónde colocarías validaciones?
- ¿Cómo manejarías idempotencia?

🟦 C. Ejercicios de dbt (2)

1. Crear un modelo incremental

Dataset: `raw.orders`

Requisitos:

- Procesar sólo registros nuevos (usando `updated_at`)
- Aplicar un `unique_key`
- Añadir tests:
  - not_null
  - unique
  - relationships con dim_customers

Entregables:

- Modelo SQL
- Config `{{ config(materialized='incremental') }}`
- Tests en `.yml`
- Documentación

---

2. Crear un snapshot

Dataset: `raw.products`

Requisitos:

- Detectar cambios en precio y categoría
- Mantener historial
- Usar `check` strategy

- Añadir columnas:
  - `valid_from`
  - `valid_to`

🟦 D. Casos End-to-End (3 escenarios completos)

1. Caso: Sistema de pedidos

Diseña un sistema completo para ingerir datos de pedidos:

Fuentes:

- API REST (productos)
- CSV diario (pedidos)
- Eventos de pagos vía Kafka

Debe incluir:

- ETL/ELT
- dbt staging / intermediate / marts
- Validaciones
- Orquestación en Airflow
- Alertas
- Dashboard en Power BI

Preguntas:

- ¿Dónde aplicarías pruebas?
- ¿Cómo manejarías late arriving data?
- ¿Cómo monitorearías el pipeline?

---

2. Caso: Pipeline de machine learning con Snowflake

Requisitos:

- Ingestión → Transformación → Feature store
- Uso de Snowflake Tasks + Streams
- Despliegue CI/CD

- Validación en fisrt step
- Orquestación con Prefect

Preguntas:

- ¿Dónde guardarías el modelo?
- ¿Cómo harías rollback?
- ¿Cómo manejarías data drift?

---

3. Caso: Empresa con 20 fuentes distintas

Problema:

- Tienes datos duplicados
- Esquemas inconsistentes
- Gobernanza débil
- Datos sensibles

Diseña un sistema que incluya:

- Data contracts
- Metadata management (DataHub)
- Catalogación
- Lineage
- Enmascaramiento de datos
- Validaciones automáticas
- Pipelines confiables

🟦 E. Preguntas de Comportamiento Técnico (10)

1. **Cuéntame sobre un pipeline que construiste del que estés orgulloso.**
   — Qué problema resolvió, tecnologías usadas, impacto.
2. **Describe un momento en que encontraste datos corruptos y cómo lo solucionaste.**
3. **Una vez en que tomaste un dataset con mala calidad y lo mejoraste.**
4. **Describe un desacuerdo técnico con un colega y cómo lo manejaste.**
5. **¿Cómo priorizas cuando tienes 10 tareas críticas?**
6. **Cuéntame de un bug difícil en un pipeline y cómo lo depuraste.**

7. **Explica una ocasión donde tuviste que mejorar el performance de un SQL muy lento.**
8. **Describe cuando implementaste CI/CD. ¿Qué pruebas automatizaste?**
9. **Cuéntame sobre un error que cometiste en producción. ¿Qué aprendiste?**
10. **Describe una situación donde tuviste que explicar un modelo complejo a un stakeholder no técnico.**