

## Sección 1: Preguntas Teóricas

### SQL (4 puntos)

1. ¿Cuál es la diferencia entre un JOIN y un UNION en SQL? Proporcione un ejemplo de cada uno. (1 punto)

**JOIN** se utiliza para integrar datos de múltiples tablas estableciendo una relación lógica entre ellas, generalmente por medio de una llave primaria y una llave foránea permitiendo consolidar información de manera estructurada, por el contrario **UNION** se utiliza para fusionar los resultados de múltiples consultas que poseen la misma estructura, asegurando que el número de columnas y los tipos de datos sean compatibles, el resultado es un conjunto unificado de registros que combina las filas de ambas consultas en una única vista ya consolidada

- Ejemplo JOIN

#### SELECT

```
c.cuenta_id,  
c.tipo_cuenta,  
c.saldo,  
t.transaction_id,  
t.tipo_transaccion,  
t.monto  
t.fecha
```

**FROM** cuentas c

**INNER JOIN** transacciones t **ON** c.cuenta\_id = t.cuenta\_id;

- Ejemplo UNION

#### SELECT

```
transaccion_id,  
cuenta_id,  
monto,  
fecha,  
'Deposito' AS tipo_movimiento
```

**FROM** depositos

#### UNION

#### SELECT

```
transaccion_id,  
cuenta_id,  
monto,  
fecha,  
'Retiro' AS tipo_movimiento
```

**FROM** retiros;

2. Explique qué es una consulta CTE (Common Table Expression) y mencione un caso de uso. (1 punto)

Una **CTE** es una estructura temporal en SQL la cual optimiza la organización y legibilidad para consultas complejas, se define con la cláusula **WITH** la cual genera subconjuntos de datos reutilizables en una consulta principal, mejorando el rendimiento y eliminando la necesidad de crear tablas temporales

- Supongamos que tenemos una tabla de **transacciones** con depósitos y retiros, y queremos calcular el saldo total por cliente, excluyendo aquellas transacciones con estado "Anulado".

```
WITH transacciones_filtradas AS (  
    SELECT  
        cliente_id,  
        tipo,  
        monto *  
        CASE WHEN tipo = 'Retiro' THEN -1 ELSE 1 END AS monto_ajustado  
FROM transacciones  
WHERE estado <> 'Anulado'  
)  
SELECT  
    cliente_id,  
    SUM(monto_ajustado) AS saldo_total  
FROM transacciones_filtradas  
GROUP BY cliente_id  
ORDER BY saldo_total DESC;
```

3. ¿Qué es la cláusula HAVING en SQL y en qué se diferencia de WHERE? (1 punto)

La cláusula **HAVING** se usa principalmente para filtrar después de realizada una agregación en una tabla como por ejemplo SUM, COUNT, AVG, etc, **WHERE** se utiliza para realizar un filtro de filas antes de una agregación ejemplo estado = 'Anulado'

4. Dado el siguiente conjunto de datos en una tabla llamada ventas, escriba una consulta SQL para obtener el total de ventas por cliente, incluyendo sólo aquellos con un monto total superior a 250. (1 punto):

```
SELECT cliente_id, COUNT(id_venta) AS total_ventas  
FROM ventas  
GROUP BY cliente_id  
HAVING SUM(monto) > 250;
```

### Python (3 puntos)

1. ¿Cuál es la diferencia entre una lista y un diccionario en Python? Dé un ejemplo de cada uno. (1 punto)

La principal diferencia radica en cómo se almacenan y acceden a los datos, la **lista** tiene una estructura de colección ordenada de elementos, mientras el **diccionario** tiene una estructura de colección pares-valor

- Ejemplo lista  
transacciones = ['Compra de AAPL', 'Venta de MSFT', 'Depósito de efectivo']
- Ejemplo diccionario  
portafolio = {  
    'AAPL': 10, # 10 acciones de Apple  
    'MSFT': 5, # 5 acciones de Microsoft  
    'BTC': 0.2, # 0.2 Bitcoin  
    'efectivo': 5000 # Dinero disponible en dólares  
}

2. ¿Qué librerías en Python se utilizan comúnmente para análisis de datos? Mencione al menos tres. (1 punto)

Las principales librerías para análisis de datos en Python son:

- Pandas (Manipulación y Análisis de Datos)
- Numpy (Cálculos Numéricos y Matrices)
- Matplotlib (Visualización Gráfica)
- Seaborn (Visualización avanzada y Estadística)

3. ¿Qué salida generará el siguiente código en Python? (1 punto)

```
data = [5, 3, 9, 1]  
print(sorted(data)[-2])
```

**La salida del código es 5**

### Visualización de datos - Tableau (4 puntos)

1. ¿Cuál es la diferencia entre una medida y una dimensión en Tableau? (1 punto)

La diferencia principalmente radica en cómo se utilizan para realizar el análisis de datos, las **dimensiones** se utilizan para representar atributos categóricos que se usan para segmentar o describir los datos, las **medidas** representan valores numéricos los cuales se pueden agregar para realizar distintos análisis

2. ¿Qué tipo de gráfico es más adecuado para comparar porcentajes de una variable categórica? (1 punto)

El tipo de gráfico más adecuado depende de las cantidades de variables a analizar, si tenemos más de 5 variables lo recomendable es utilizar gráficos de barras verticales, horizontales o de barras apiladas si tenemos menos de 1 a 4 variables puede representarse en un gráfico de torta o pie

3. ¿Cómo se puede crear un filtro dinámico en Tableau para que el usuario pueda seleccionar un rango de fechas? (1 punto)

- Arrastrar el campo de fecha a la tarjeta de filtros
- En la ventana emergente elegir **Rango de fechas**
- Seleccionar el rango de fechas en las que se podrá aplicar el filtro
- Click derecho sobre la tarjeta de filtros y seleccionar **Mostrar Filtro**
- En la flecha desplegable de la tarjeta del filtro podemos escoger las opciones de personalización como
  - Fecha Relativa
  - Rango de fechas
  - Fecha inicio
  - Fecha finalización
  - Explorar periodos

4. ¿Que es un archivo hyper en tableau, y en que caso de uso se puede aplicar? (1 punto)

Es un formato de base de datos optimizado que utiliza Tableau para almacenar y procesar datos de manera más rápida y eficiente, está diseñado para manejar grandes volúmenes de datos con un alto rendimiento, los principales casos de uso son :

- Cuando se necesita extraer datos de una base o archivo grande y usarlos sin conexión
- Permite crear extractos de datos en lugar de depender de una conexión en tiempo real
- Se puede generar desde script en python para procesamiento automatizado
- Flujos de trabajo ETL donde se necesita cargar datos de múltiples fuentes

