

# ITAM

## DATA SCIENCE AND MACHINE LEARNING APPLIED TO FINANCIAL MARKETS<sup>1</sup>

### Modulo III

#### Tarea: Consultas en SQL

Objetivo:

El estudiante será capaz de formular consultas SQL en Python utilizando SQLAlchemy y `pd.read_sql`, aplicando correctamente los temas de SELECT, WHERE, GROUP BY, HAVING, ORDER BY, JOINS, subconsultas y funciones de ventana sobre las tablas del esquema `banco_base`.

#### Instrucciones

1. Utiliza el motor de conexión de SQLAlchemy en Python para conectarte a la base de datos `banco_base`.
2. Por cada enunciado de consulta, deberás escribir el query en SQL (dentro de comillas triples en Python) y ejecutarlo con `pd.read_sql`.

```
query = """
-- tu consulta aquí
"""

df = pd.read_sql(query, engine)
display(df)
```

3. Responde los **20 enunciados de consulta** listados abajo.
4. Incluye en tu entrega tanto el código en Python como las salidas de cada consulta.

#### Consultas a realizar (Bloque 1-Fácil)

1. Lista los nombres y edades de todos los clientes que tengan ingresos mayores a 50,000.
2. Obtén el número total de clientes por país, ordenado de mayor a menor.

3. Muestra los clientes (id\_cliente, nombre, ingresos) cuyo estrato sea 2 y que además tengan al menos una tarjeta asociada.
4. Calcula el promedio de ingresos por sexo, y muestra solo aquellos cuyo promedio sea mayor a 40,000.
5. Encuentra los 5 clientes con mayores ingresos y ordénalos de manera descendente.
6. Realiza un JOIN para mostrar el nombre del cliente, el tipo de tarjeta y el monto asociado.
7. Encuentra a los clientes que tienen más de una tarjeta registrada.
8. Obtén el nombre y país de los clientes cuyo ingreso sea mayor al ingreso promedio de todos los clientes. (Usa una subconsulta).
9. Usa una función de ventana para mostrar el nombre de cada cliente, su país y el ranking de ingresos dentro de su país.
10. Encuentra los clientes que poseen tarjeta de crédito y cuyos ingresos estén en el top 10% de todos los ingresos. (Usa percent\_rank o ntile).

## Consultas a realizar (Bloque 2-Intermedio)

1. **Top 3 países con mayor promedio de ingresos entre clientes con tarjetas de crédito.**  
(Pista: join + filtro por tipo de tarjeta + group by + order by con limit).
2. **Obtener el nombre y país de los clientes cuyo ingreso es mayor al ingreso promedio de su mismo estrato.**  
(Pista: subconsulta correlacionada por estrato).
3. **Mostrar el cliente con más tarjetas en cada país.**  
(Pista: join + group by + función de ventana RANK() OVER (PARTITION BY pais ORDER BY COUNT(\*) DESC)).
4. **Calcular el monto total de tarjetas por cliente y mostrar solo aquellos cuya suma de montos supere el promedio de todos los clientes.**  
(Pista: agregación con having + subconsulta del promedio).
5. **Obtener los 5 clientes más jóvenes que tienen tarjeta de crédito y cuyos ingresos estén en el top 10% de todos los ingresos.**  
(Pista: percentil con NTILE(10) OVER (ORDER BY ingresos DESC) y filtro).
6. **Mostrar los clientes que tienen tanto tarjeta de débito como de crédito.**  
(Pista: self-join o group by con having count distinct).
7. **Calcular el ingreso promedio por estrato y sexo, y mostrar solo las combinaciones donde ese promedio sea mayor al promedio general de ingresos.**  
(Pista: doble group by con having y subconsulta).

8. **Obtener para cada país el cliente con mayor monto total en tarjetas.**  
(Pista: `join + sum(monto) + función ROW_NUMBER() OVER (PARTITION BY pais ORDER BY SUM(monto) DESC)`).
9. **Listar a los clientes cuyo ingreso está por encima del promedio de su país, y además ese país tenga al menos 5 clientes registrados.**  
(Pista: `subconsulta + having`).
10. **Calcular el monto promedio de tarjetas de crédito por estrato y mostrar en qué estrato es más alto.**  
(Pista: `agregación + order by + limit 1`).

#### Formato de Entrega:

- **Archivo principal:** un script en Python (.py) o un notebook (.ipynb) que contenga todas las consultas resueltas y ejecutables con `pd.read_sql`.
- **PDF adjunto:** que contenga una explicación breve (1–2 páginas) de cada consulta, justificando la lógica usada, las funciones y subconsultas empleadas, así como comentarios sobre la interpretación de los resultados.
- **Nomenclatura de los archivos:**
  - `Tarea_SQLAlchemyConsultas_NombreApellidoAA.py`
  - `Tarea_SQLAlchemyConsultas_NombreApellidoAA.pdf`
- **Entrega:** por correo en una carpeta comprimida .zip con ambos archivos, a más tardar el **13/10/2025 12:00 a.m.**

#### Notas adicionales:

- Cada consulta debe estar claramente comentada indicando qué hace.
- Los resultados deben mostrarse mediante `pandas.DataFrame` o en tablas dentro del notebook/Python.
- La claridad en la documentación y en la presentación de resultados será tomada en cuenta en la evaluación.