Introducción a la cláusula unión de SQL

El operador **UNION** en SQL se utiliza para combinar el resultado de dos o más SELECT en un solo conjunto de resultados.

La sintaxis básica de UNION es la siguiente:

SELECT columna1, columna2

FROM tabla1

UNION SELECT columna1, columna2

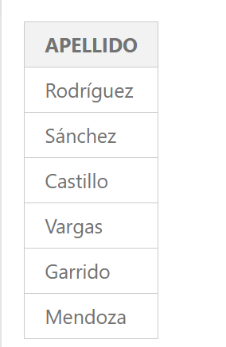
FROM tabla2;

Las columnas que se seleccionan en los SELECT deben tener los mismos nombres de columna, secuencia y tipos de datos.

Veamos un ejemplo:

Supongamos que tenemos dos tablas: 'Estudiantes' y 'Profesores', que contienen una lista de apellidos en cada una. Queremos crear una lista que combine los apellidos de ambas tablas.





Al hacer la consulta:

SELECT apellido

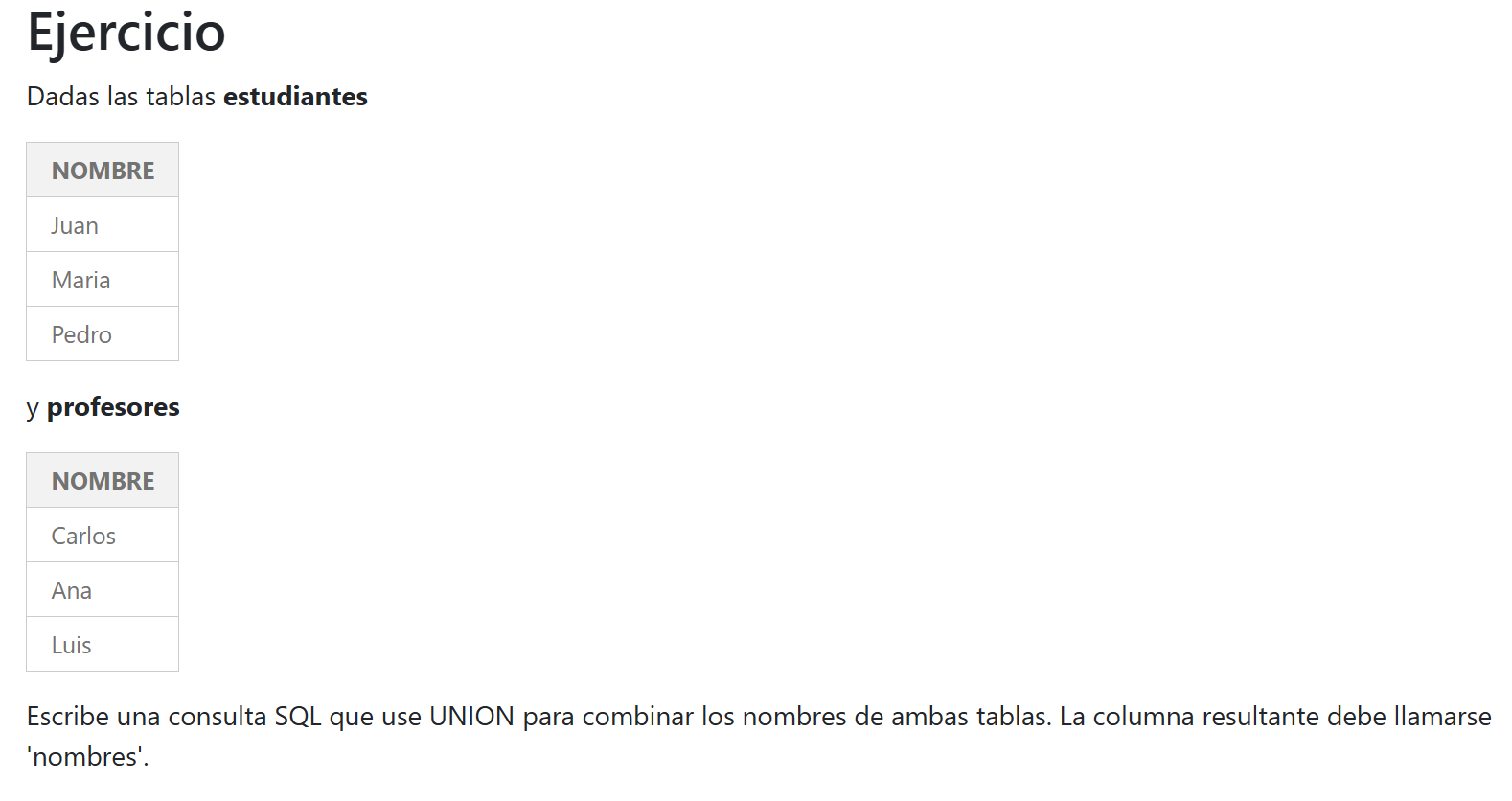
FROM Estudiantes

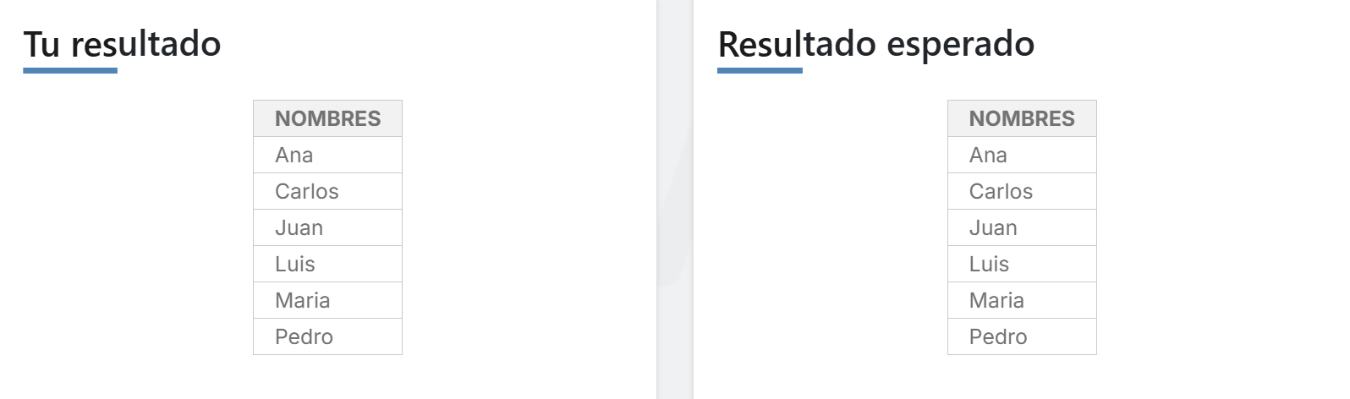
UNION

SELECT apellido

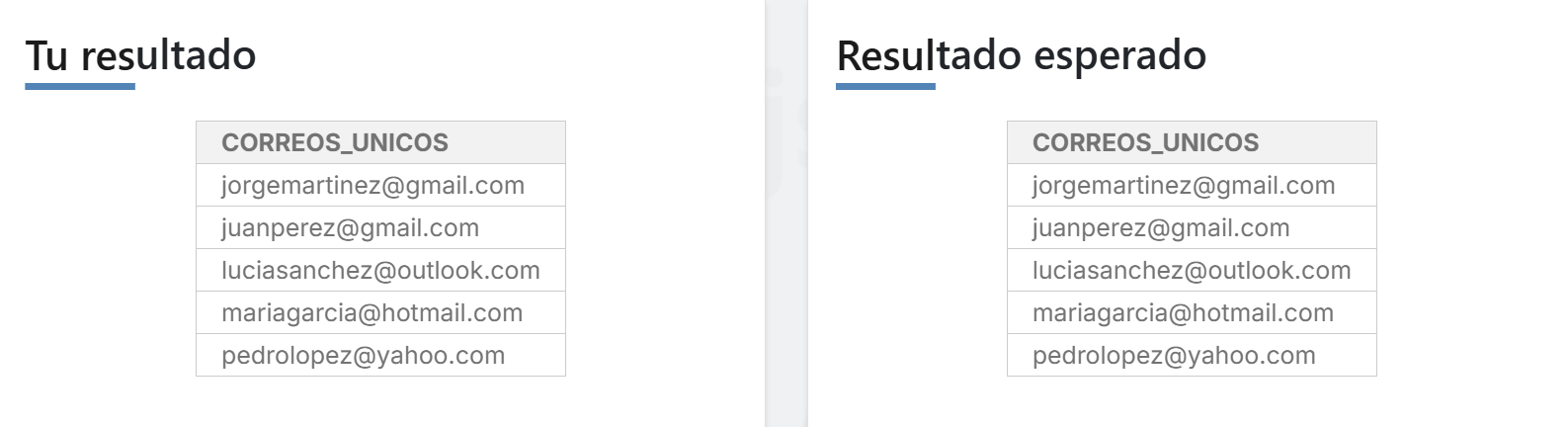
FROM Profesores;

Nos daría el resultado: =>



SELECT nombre AS nombres FROM estudiantes UNION SELECT nombre AS nombres FROM profesores

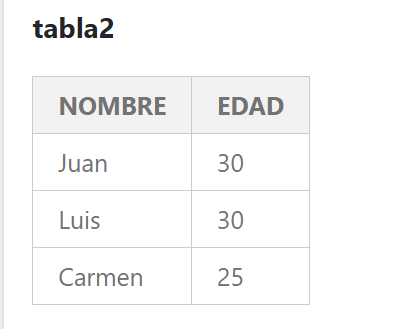


SELECT email AS correos\_unicos FROM usuarios UNION SELECT email AS correos\_unicos FROM clientes;

*Union vs Union all*

En los ejercicios anteriores aprendimos que el operador **UNION** se utiliza para combinar los resultados de dos o más consultas SELECT en un solo conjunto de resultados, eliminando las filas duplicadas.

Si queremos obtener las filas duplicadas en el resultado, utilizaremos el operador *UNION ALL*.

Por ejemplo, si tenemos dos tablas, 'tabla1' y 'tabla2', con los siguientes datos:

Observa que Juan está en ambas tablas.

Podemos combinar ambas tablas utilizando UNION ALL de la siguiente forma:

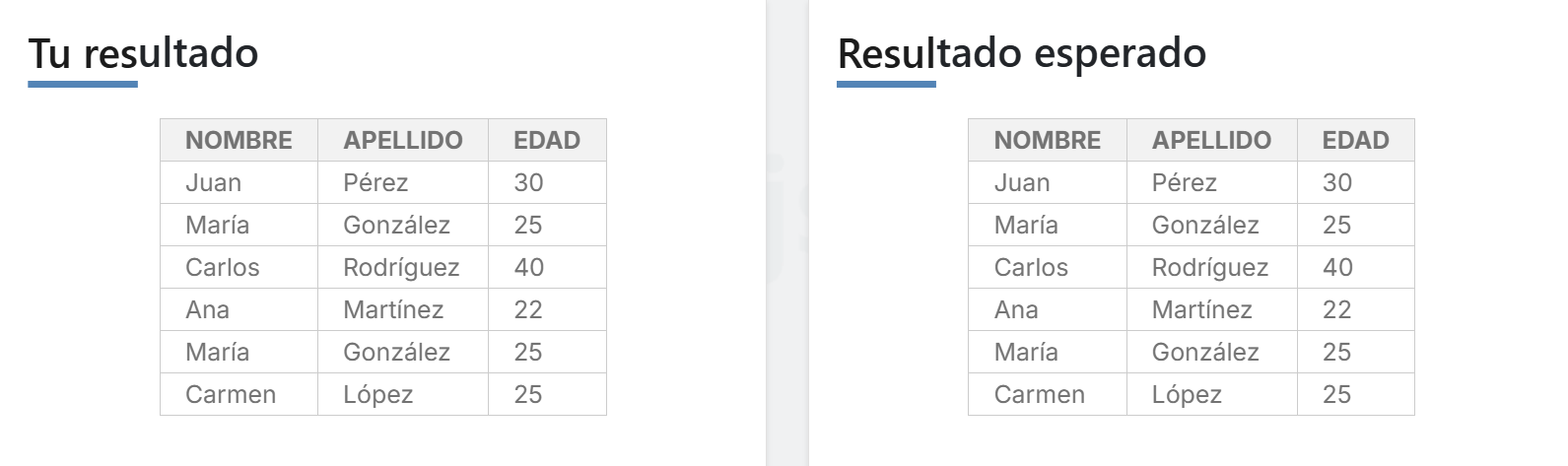
SELECT \* FROM tabla1 UNION ALL SELECT \* FROM tabla2;

Como resultado obtendremos:

Interfaz de usuario gráfica, Tabla

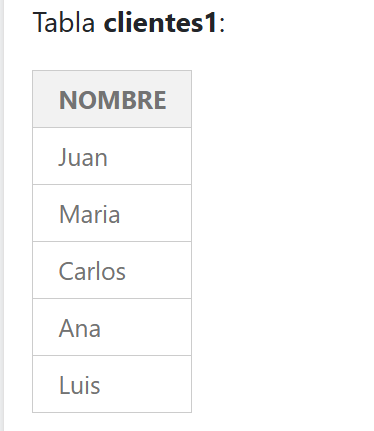
Descripción generada automáticamente con confianza media

SELECT \* from empleados1 UNION ALL SELECT \* FROM empleados2



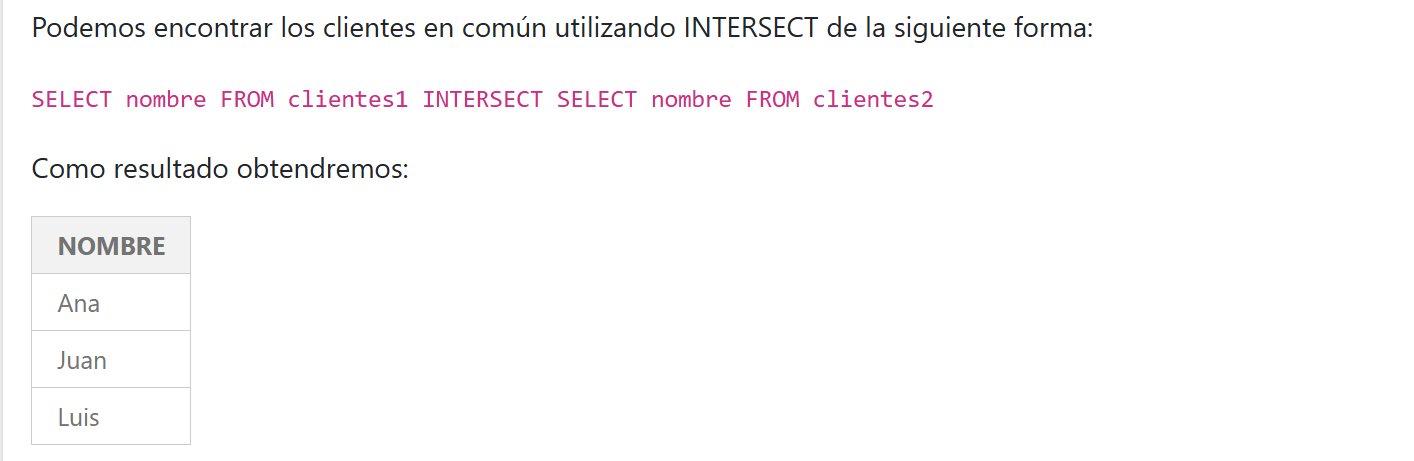
*Introducción a intersección*

El operador **INTERSECT** se utiliza para combinar dos SELECT y devolver los resultados que se encuentran en ambas consultas.

Por ejemplo, si tenemos las siguientes dos tablas, clientes1 y clientes2:

Tabla

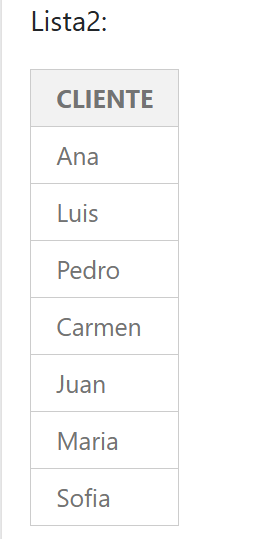
Descripción generada automáticamente



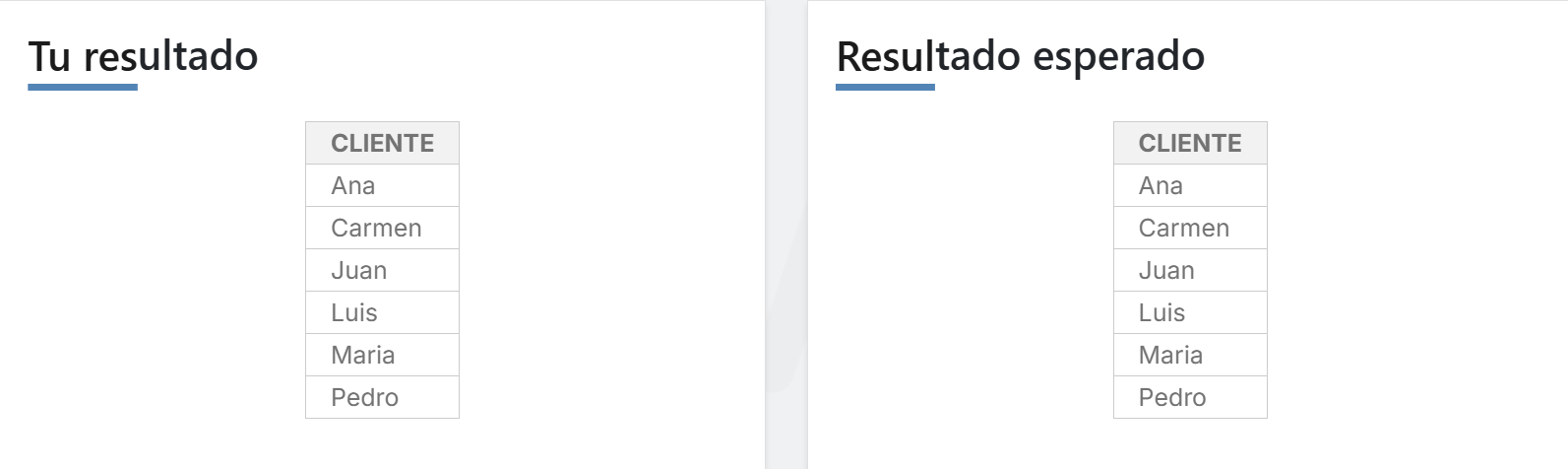
Ejercicio

Dadas las siguientes tablas, **lista1** y **lista2**, encuentra los clientes que están en ambas listas.

Tabla

Descripción generada automáticamente

SELECT cliente from Lista1 INTERSECT SELECT cliente FROM Lista2



*El operador Except*

El operador EXCEPT en SQL se utiliza para devolver todas las filas en la primera consulta que no están presentes en la segunda consulta. En otras palabras, EXCEPT devuelve solo las filas, que son parte de la primera consulta pero no de la segunda consulta.

Tabla

Descripción generada automáticamentePor ejemplo, si tenemos dos tablas, 'Tabla1' y 'Tabla2', que contienen los siguientes datos:



Podemos usar EXCEPT para encontrar los nombres que están en 'Tabla1' pero no en 'Tabla2' con la siguiente consulta:

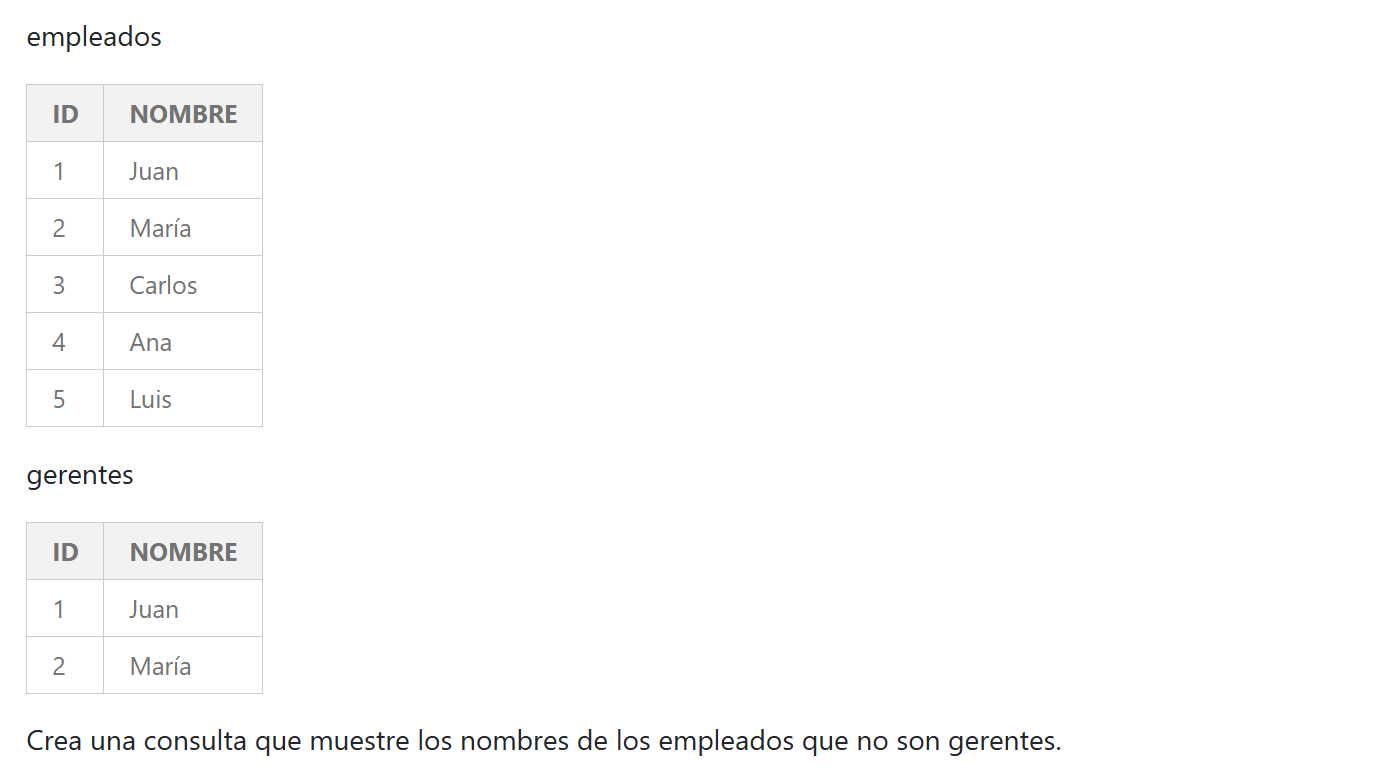
SELECT nombre FROM Tabla1 EXCEPT SELECT nombre FROM Tabla2;

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza bajaEsto daría como resultado:

Ejercicio

Dadas las siguientes tablas, 'empleados' y 'gerentes', que contienen los siguientes datos:



select nombre from empleados EXCEPT select nombre from gerentes

