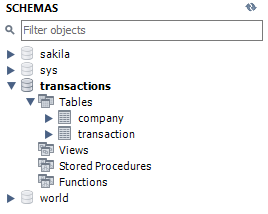
**>>>>>>>>>>>>>>>>>>>> S P R I N T 1 <<<<<<<<<<<<<<<<<<<<<<**

**======================= NIVEL 1 =======================**

**- Ejercicio 1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**A partir de los documentos adjuntos (estructura\_datos y datos\_introducir), importa las dos tablas. Muestra las principales características del esquema creado y explica las diferentes tablas y variables que existen. Asegúrate de incluir un diagrama que ilustre la relación entre las distintas tablas y variables.**



Se creó un esquema en base al código que contenía los archivos facilitados, este esquema consta de 2 tablas:

company

transaction

La tabla “company” contiene columnas:

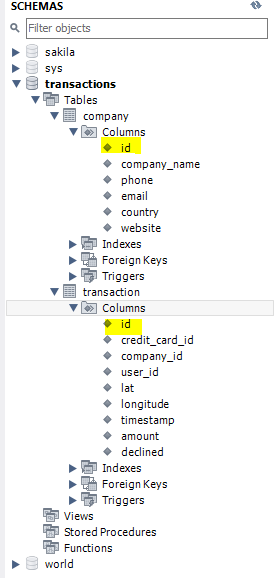
id, company\_name, pone, email, country, website

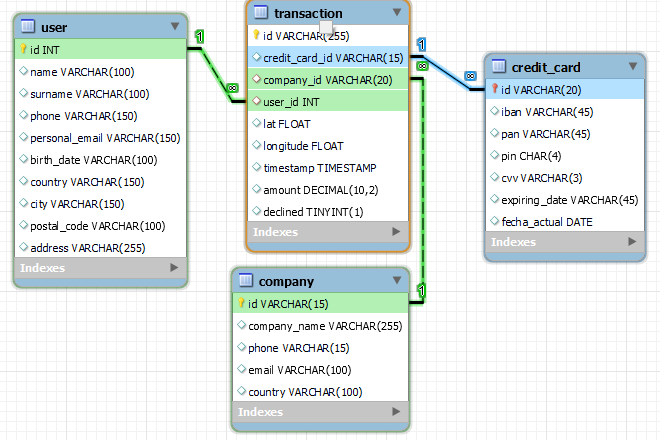
La tabla “transaction” contiene columnas:

Id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, amount, declined

**Relación:** Las tablas transaction y company se relacionan mediante la columna company\_id en la tabla transaction. Esta columna es una clave externa que hace referencia a la columna id en la tabla company. Esto significa que cada transacción en la tabla transaction está asociada con una empresa en la tabla company.

**Cardinalidad:** La cardinalidad de la relación entre transaction y company es de uno a muchos. Esto significa que una empresa puede tener muchas transacciones, pero una transacción solo puede estar asociada con una empresa, por esta razón vemos en el grafico el número 1 y el signo de infinito.





**- Ejercicio 2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

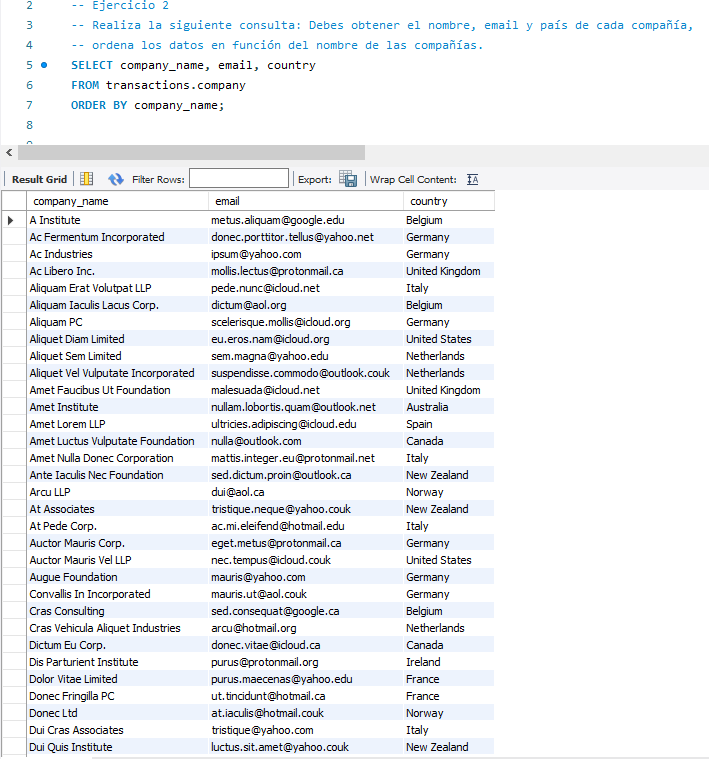
**Realiza la siguiente consulta: Debes obtener el nombre, email y país de cada compañía, ordena los datos en función del nombre de las compañías.**

SELECT company\_name, email, country

FROM transactions.company

ORDER BY company\_name;

**ORDER BY:** nos ayuda para ordenar por el nombre de la compañía, ya que los contenidos de búsqueda son textos, si fuera numérico se usaría ASC (ascendente) / DESC (descendente)



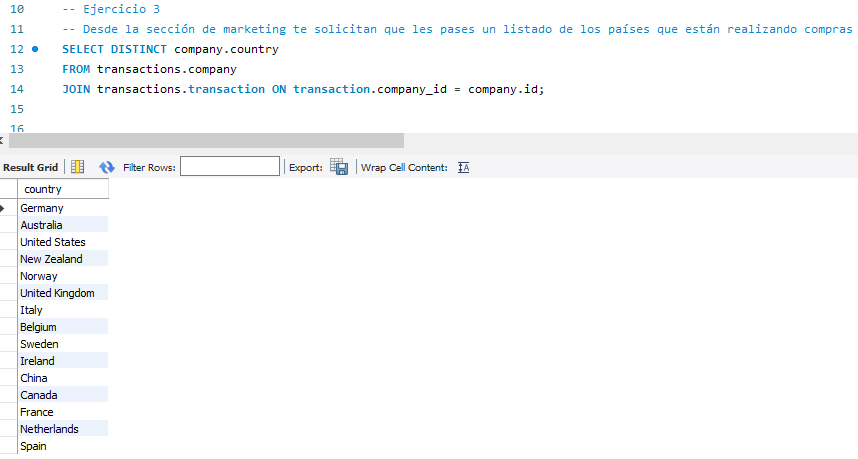
**-- Ejercicio 3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Desde la sección de marketing te solicitan que les pases un listado de los países que están realizando compras**

SELECT DISTINCT company.country

FROM transactions.company

JOIN transactions.transaction ON transaction.company\_id = company.id;



**SUGERENCIA:**

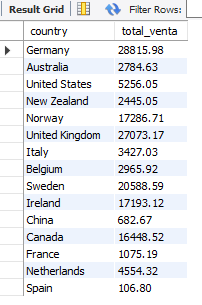
En este caso el pedido hubiera sido interesante agregar una columna con los valores de compra por país

SELECT DISTINCT company.country, SUM(amount) AS total\_venta

FROM transactions.company

JOIN transactions.transaction ON transaction.company\_id = company.id

GROUP BY company.country;



**-- Ejercicio 4\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

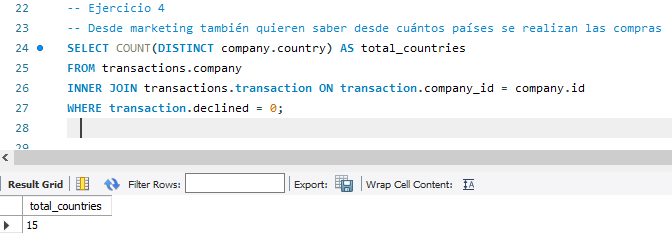
**Desde marketing también quieren saber desde cuántos países se realizan las compras**

SELECT COUNT(DISTINCT company.country) AS total\_countries

FROM transactions.company

INNER JOIN transactions.transaction ON transaction.company\_id = company.id

WHERE transaction.declined = 0;



**-- Ejercicio 5\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

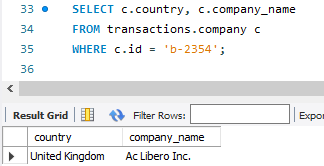
**Tu jefe identifica un error con la compañía que tiene vaya 'b-2354'.**

**Por tanto, te solicita que le indiques el país y nombre de compañía de este ve.**

SELECT c.country, c.company\_name

FROM transactions.company c

WHERE c.id = 'b-2354';



**-- Ejercicio 6\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Además, ¿tu jefe te solicita que indiques cuál es la compañía con mayor gasto medio?**

SELECT company\_name, AVG(transaction.amount) AS average\_spend

FROM transactions.company

INNER JOIN transactions.transaction ON transaction.company\_id = company.id

GROUP BY company.id

ORDER BY average\_spend DESC

LIMIT 1;



**======================= NIVEL 2 =======================**

***-- Ejercicio 1*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***Tu jefe está redactando un informe de cierre del año y te solicita que le envíes información relevante para el documento.***

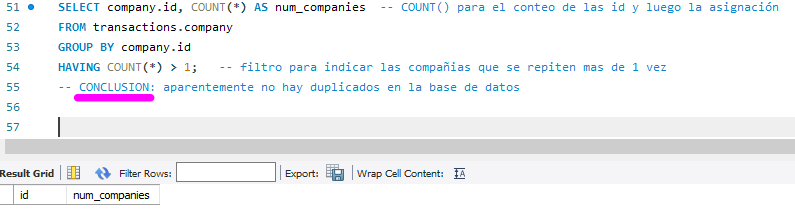
***Para ello te solicita verificar si en la base de datos existen compañías con identificadores (id) duplicados.***

SELECT company.id, COUNT(\*) AS num\_companies

FROM transactions.company

GROUP BY company.id

HAVING COUNT(\*) > 1;



*Aparentemente no tenemos resultados duplicados conforme la petición y búsqueda conforme el filtrado de HAVING COUNT(\*) > 1;*

***-- Ejercicio 2*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***¿En qué día se realizaron las cinco ventas más costosas? Muestra la fecha de la transacción y la sumatoria de la cantidad de dinero.***

SELECT DATE (timestamp) as fecha, SUM(amount) AS total\_venta

FROM transactions.transaction

GROUP BY fecha

ORDER BY total\_venta DESC -- ordenado por el total de venta en descenso

LIMIT 5;



***-- Ejercicio 3*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***¿En qué día se realizaron las cinco ventas de menor valor? Muestra la fecha de la transacción y la sumatoria de la cantidad de dinero.***

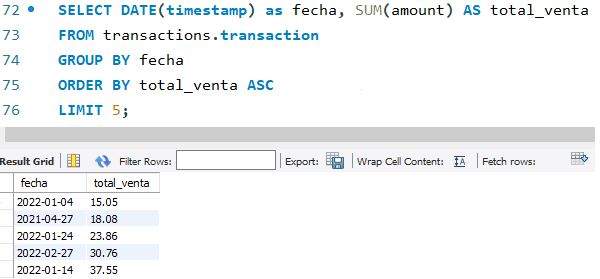
SELECT DATE(timestamp) as fecha, SUM(amount) AS total\_venta

FROM transactions.transaction

GROUP BY fecha

ORDER BY total\_venta ASC -- ordenado por el total de venta en descenso

LIMIT 5;



**-- Ejercicio 4\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**¿Cuál es la media de gasto por país? Presenta los resultados ordenados de mayor a menor medio.**

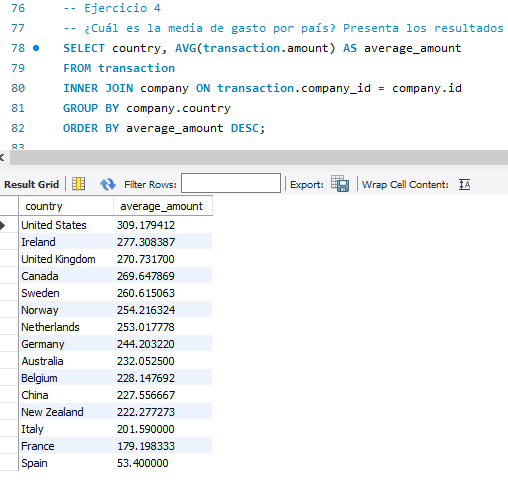
SELECT country, AVG(transaction.amount) AS average\_amount

FROM transaction

INNER JOIN company ON transaction.company\_id = company.id

GROUP BY company.country

ORDER BY average\_amount DESC;



**-- ======================= NIVEL 3 =======================**

***-- Ejercicio 1*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***Presenta el nombre, teléfono y país de las compañías, junto con la cantidad total gastada,***

***de aquellas que realizaron transacciones con un gasto comprendido entre 100 y 200 euros.***

***ordena los resultados de mayor a menor cantidad gastada.***

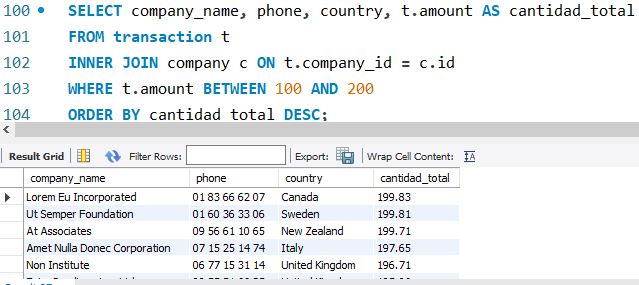
SELECT company\_name, phone, country, t.amount AS cantidad\_total

FROM transaction t

INNER JOIN company c ON t.company\_id = c.id

WHERE t.amount BETWEEN 100 AND 200

ORDER BY cantidad\_total DESC;



***-- Ejercicio 2*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***Indica el nombre de las compañías que realizaron compras el 16 de marzo de 2022,***

***28 de febrero de 2022 y 13 de febrero de 2022.***

SELECT DISTINCT company.company\_name, transaction.timestamp

FROM transaction

INNER JOIN company ON transaction.company\_id = company.id

WHERE date (timestamp) IN ('2022-03-16', '2022-02-28', '2022-02-13')

GROUP BY company\_name, transaction.timestamp;

