NIVELL 1

Ex1

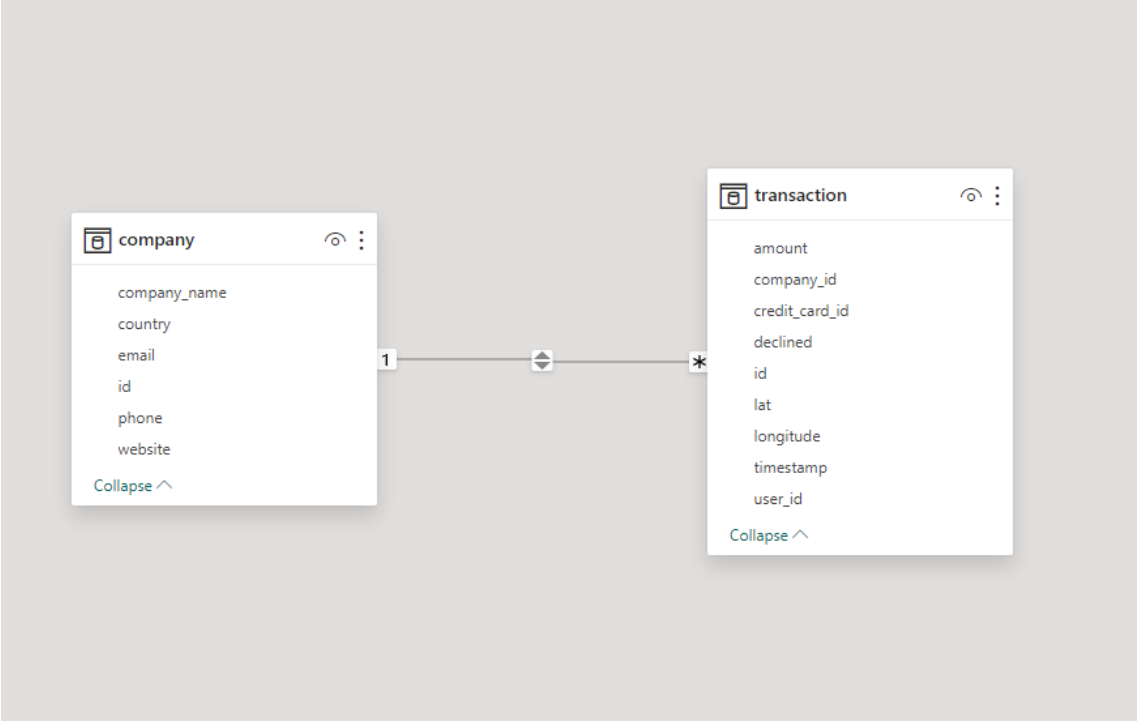


Table Transaction = **FACT TABLE**

Table Company = **DIMENSION TABLE**

Relation = N to 1 (transaction.company_id = company.id)

La mayoria de los elementos son **VARCHAR** (string de letras y/o números) pero también hay Integer (**INT**), Floats (**FLOAT**), una variable de tipo **BOOLEAN** (Ture/False), una **DECIMAL** y una de tipo fecha (**TIMESTAMP**).

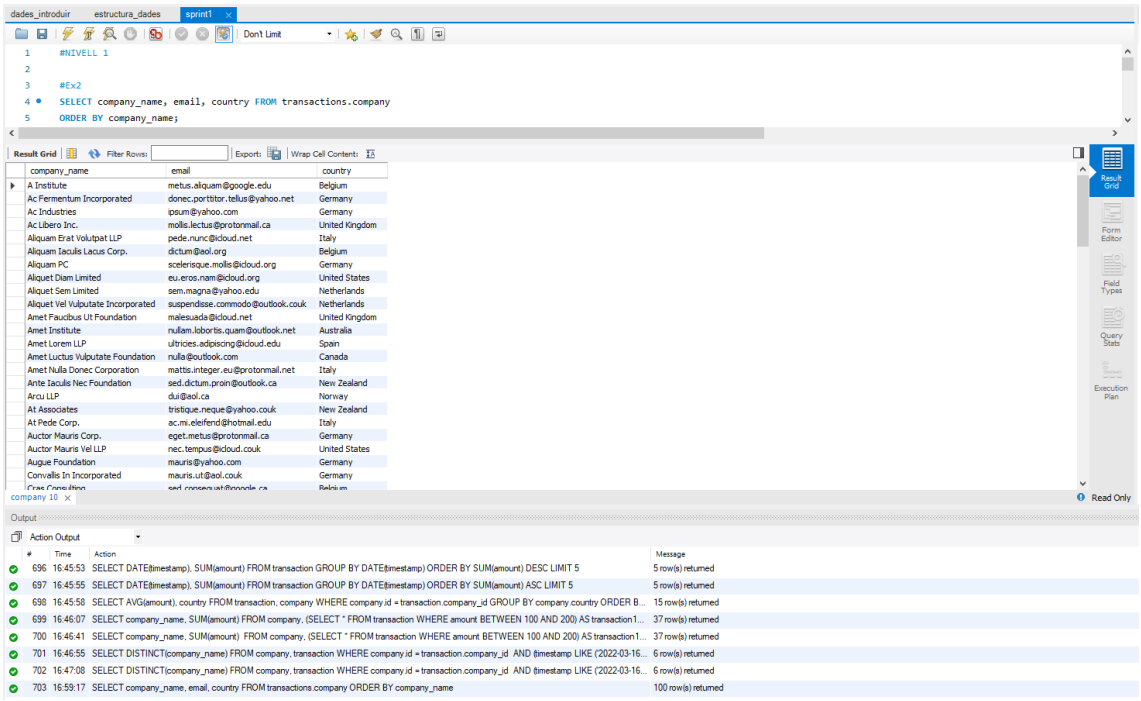
Se establece la relación entre tables mediante la **FOREIGN KEY** *transaction.company_id*, a través de la instrucción **REFRENCE**.

Se observa que aparecen otras **REFERENCE**, una con *user_id* y otra con *credit_card_id*. Seguramente existan dos tablas mas de DIMENSIONES, una llamada Users, con información de los usuarios, y otra llamada Credit Cards, con información de las tarjetas de crédito. Así se montaria un esquema en estrella con la tabla Transaction como FACT TABLE en el medio.

Ejecutamos los dos archivos proporcionados, uno crea las tablas y las relaciones, el otro las llena de contenido.

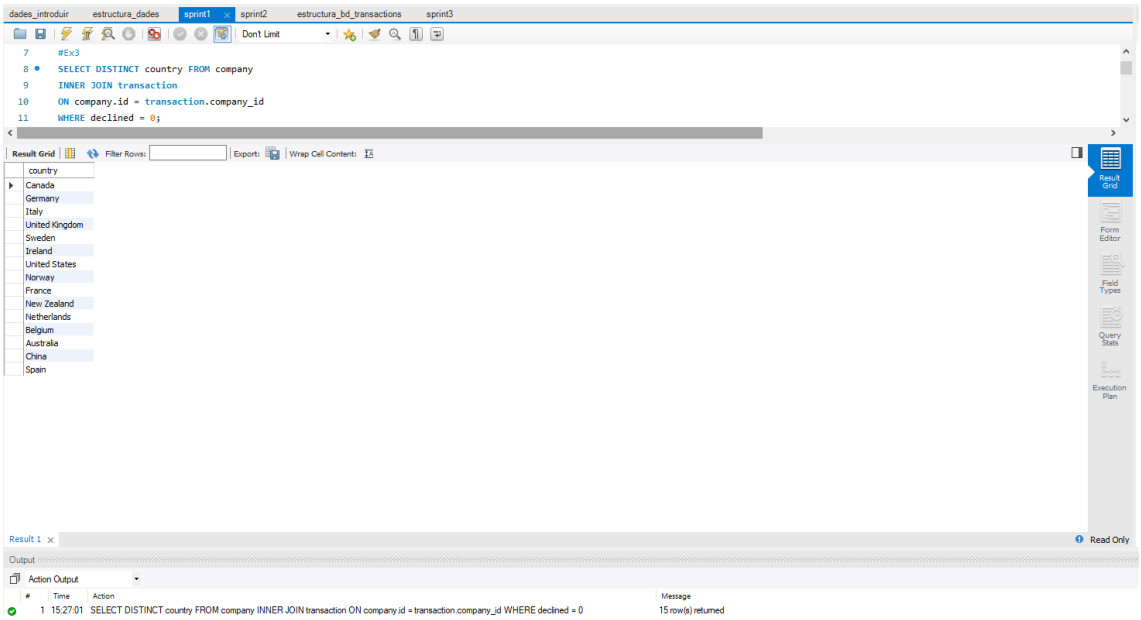
Ex2

Se hace un **SELECT** con los datos que se piden, pertenecen todos a la misma tabla, no se requieren JOINS, se ordena con un **ORDER BY** *company_name.*



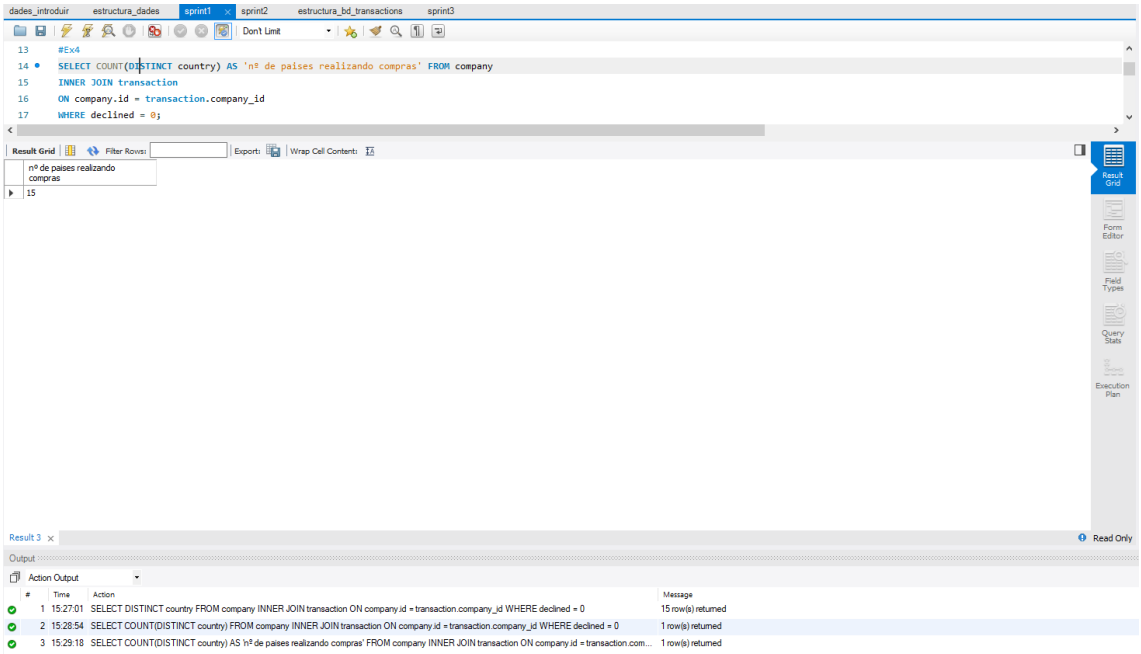
Ex3

En esta ocasión se hace un **SELECT DISTINCT** para conseguir un listado de los países sin repeticiones. Es necesario hacer un **INNER JOIN** para conseguir datos que se encuentran en las dos tablas. Se agrega una cláusula **WHERE** donde *declined = 0*, indicando que la transacción se ha realizado.



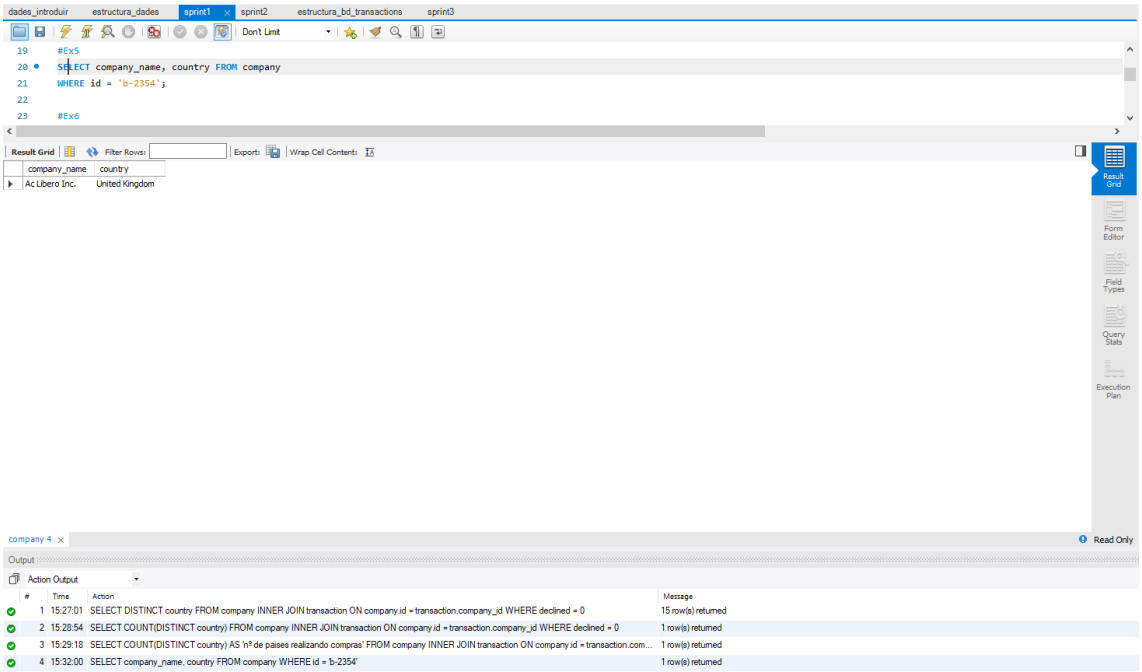
Ex4

Se puede realizar la misma consulta, pero con un **COUNT** para que nos devuelva el nº de países que realizan compras.



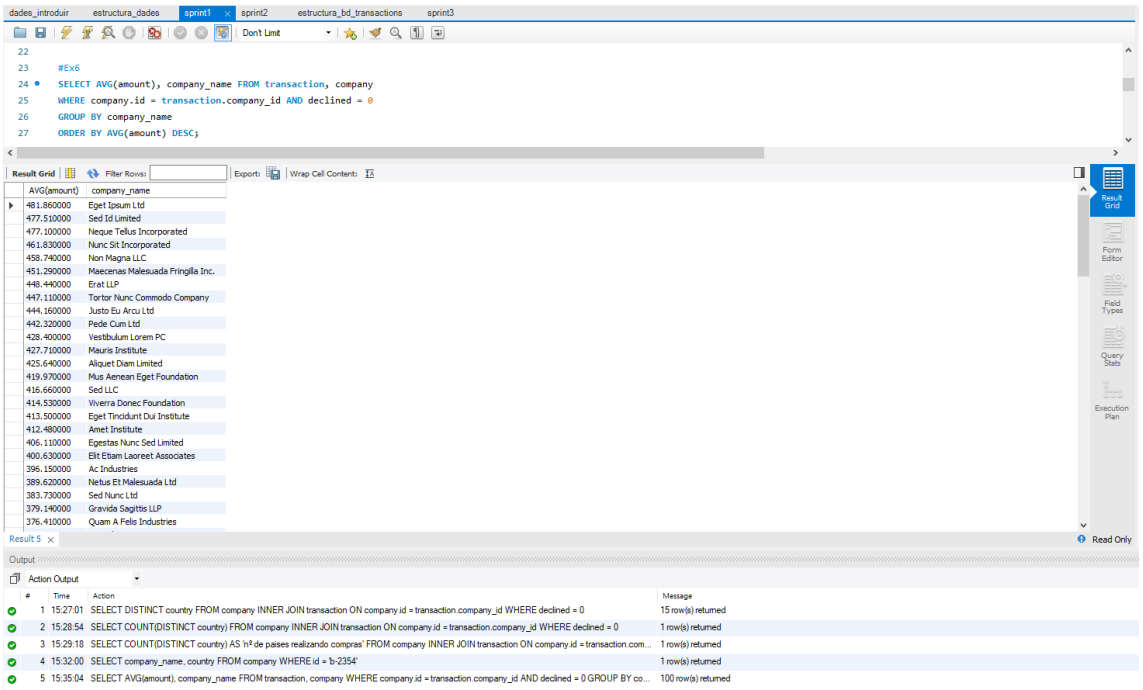
Ex5

Se realiza un **SELECT** con una clausula **WHERE** donde *id = la id solicitada*.



Ex6

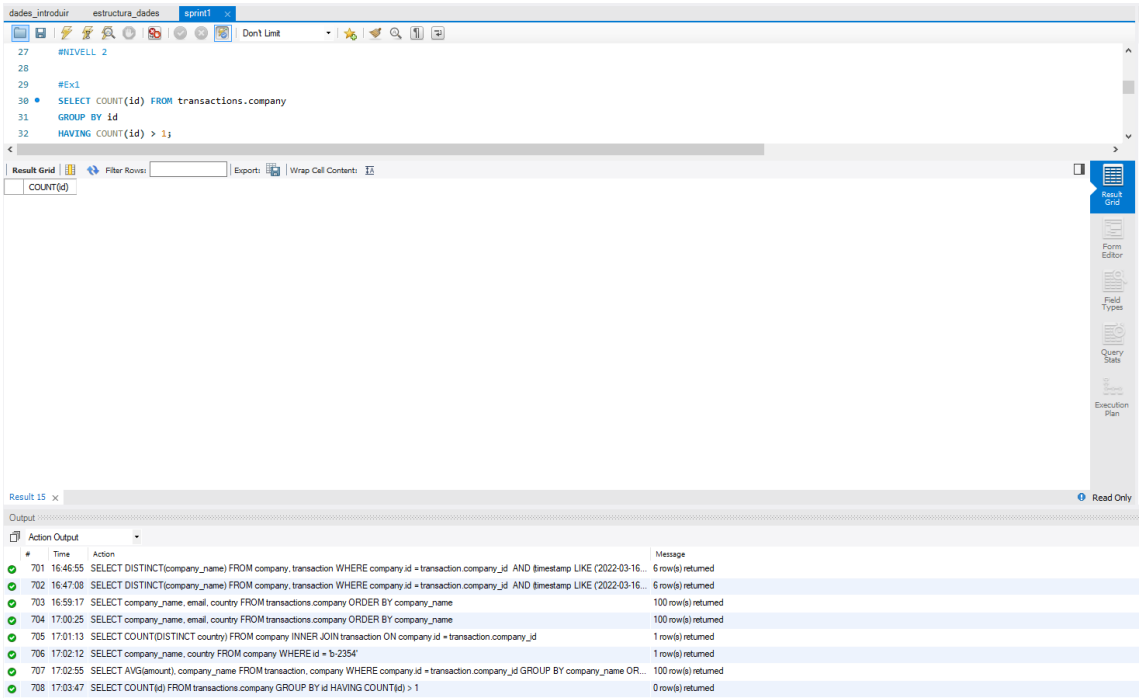
Se utiliza la función **AVG()** para determinar el valor medio, y se agrupa mediante **GROUP BY** *company_name*. En la cláusula **WHERE** se especifica la relación entre las dos tablas *company.id = transaction.company_id,* además de que la transacción se haya realizado, *declined = 0.*



NIVELL 2

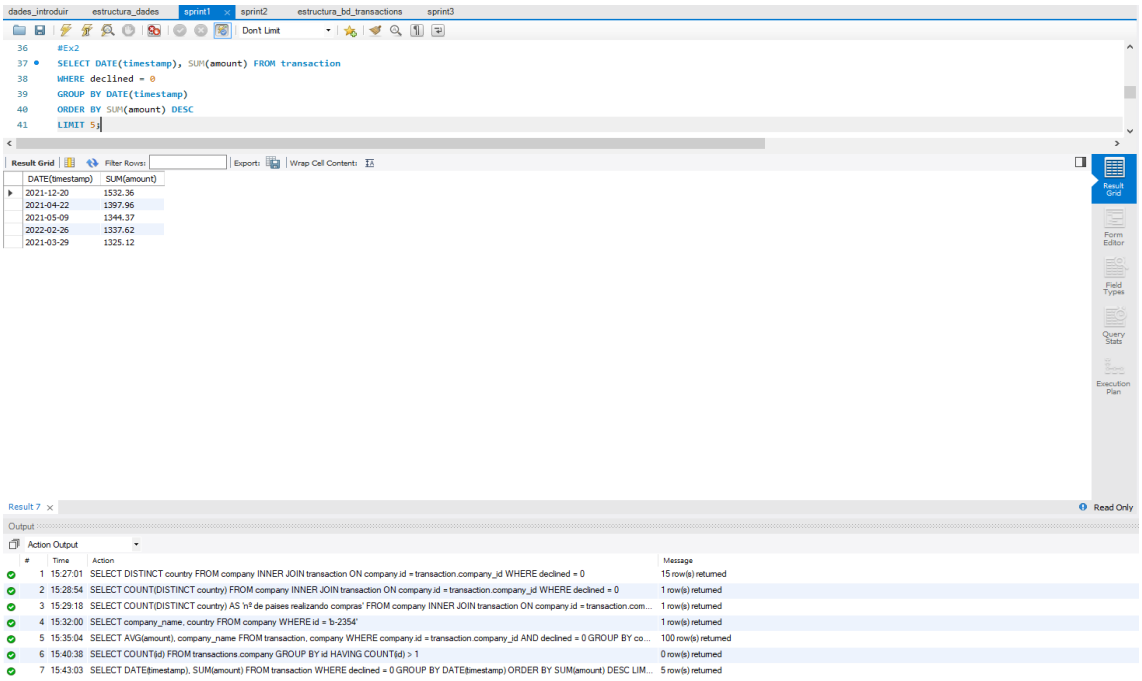
Ex1

Se comprueba mediante la realización de un **COUNT**, especificando que el valor de dicho **COUNT** sea mayor de 1 a través de la cláusula **HAVING**.



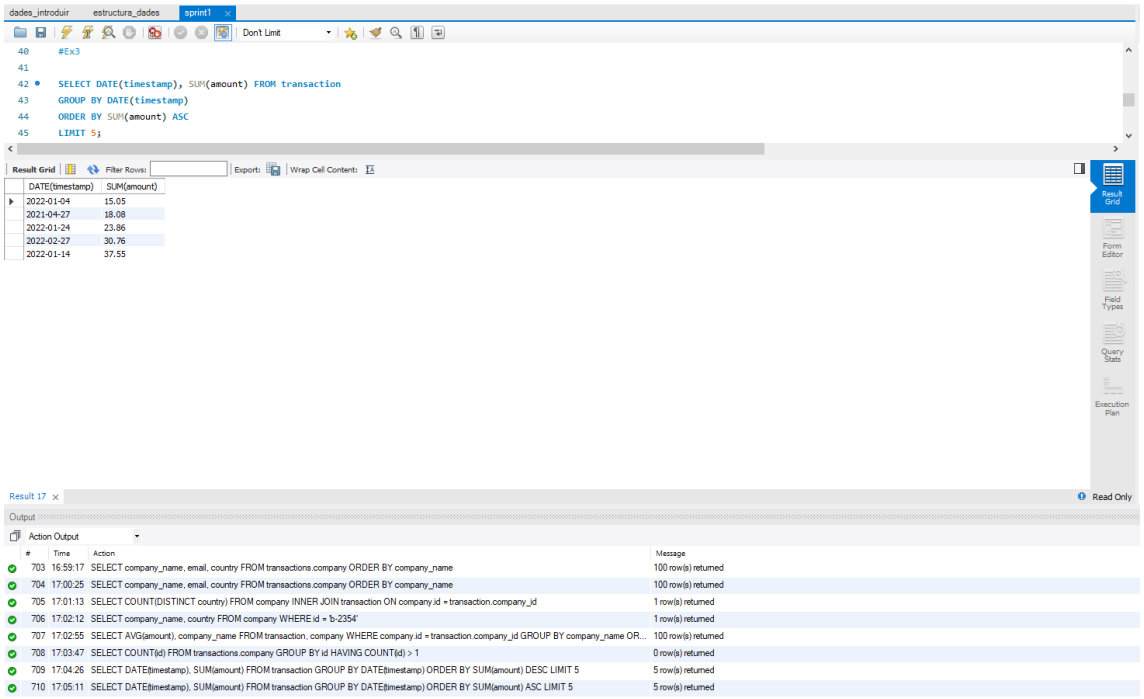
Ex2

Aquí se utilizara la fórmula **SUM()** y se agrupará por fechas. Se agrega una cláusula **WHERE** donde *declined = 0*, indicando que la transacción se ha realizado. Para que se muestren las 5 ventas mas grandes se ordena de forma descendiente con un **ORDER BY** y se limitan los resultados a las 5 primeras *rows*.



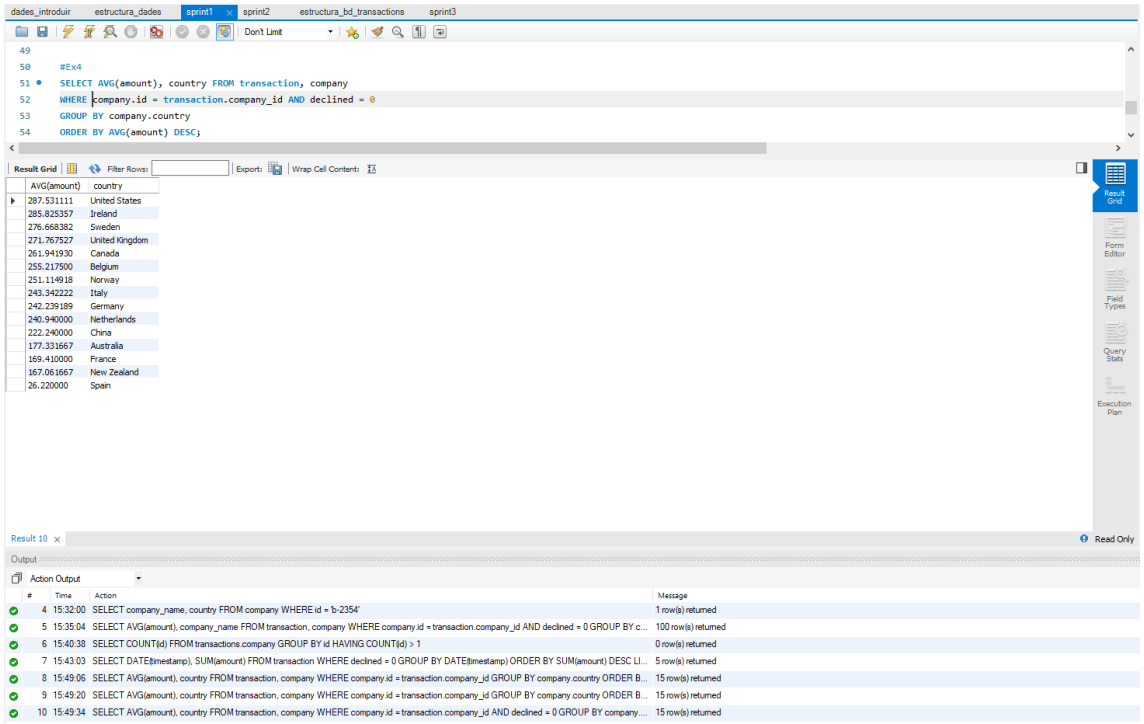
Ex3

Se realizara la misma **QUERY** que en el ejercicio anterior pero se ordenara de forma ascendiente para que ahora se muestren los 5 resultado mas pequeños.



Ex4

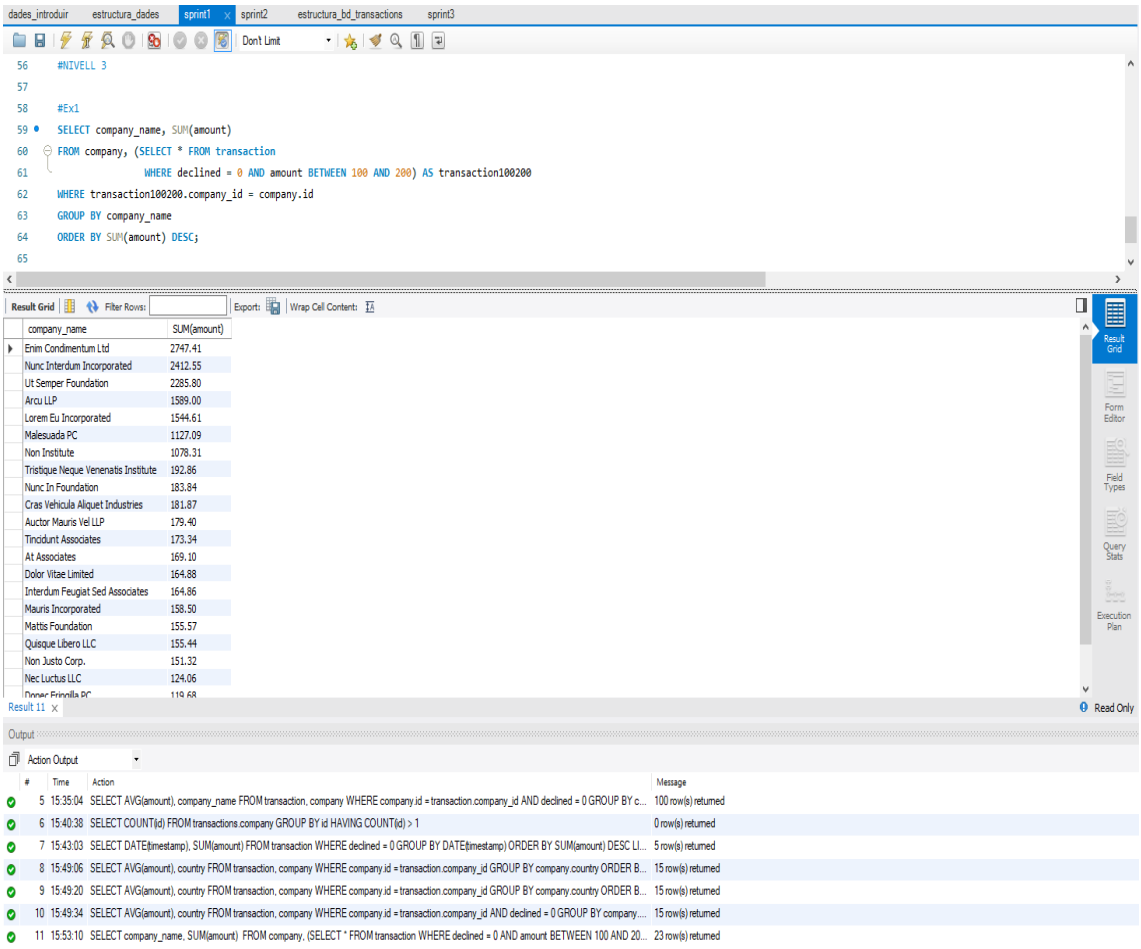
Se utilizará la función **AVG** agrupada mediante un **GROUP BY** por países. En la cláusula **WHERE** se especifica la relación entre las dos tablas *company.id = transaction.company_id,* además de que la transacción se haya realizado, *declined = 0.* Finalmente se ordena con un **ORDER BY AVG().**



NIVELL 3

Ex1

Este ejercicio requerirá una **SubQUERY**. Queremos calcular la **SUM(amount)**, pero no de todas las compañías, solo de las que han realizado transacciones de entre 100 y 200. Esta condición es la que nos marca la **SubQUERY**. En este caso haremos un **SELECT *** de transaction con la condición de que las transacciones se hayan realizado y de que sean de entre 100 y 200. Este **QUERY** lo llamaremos *transaction100200* y es del cual realizaremos la **SUM(amount).**



```
#NIVELL 3

#Ex1
SELECT company_name, SUM(amount)
FROM company, (SELECT * FROM transaction
               WHERE declined = 0 AND amount BETWEEN 100 AND 200) AS transaction100200
WHERE transaction100200.company_id = company.id
GROUP BY company_name
ORDER BY SUM(amount) DESC;
```

company_name	SUM(amount)
Enim Condimentum Ltd	2747.41
Nunc Interdum Incorporated	2412.55
Ut Semper Foundation	2285.80
Arcu LLP	1589.00
Lorem Eu Incorporated	1544.61
Malesuada PC	1127.09
Non Institute	1078.31
Tristique Neque Venenatis Institute	192.86
Nunc In Foundation	183.84
Cras Vehicula Aliquet Industries	181.87
Auctor Mauris Vel LLP	179.40
Tincidunt Associates	173.34
At Associates	169.10
Dolor Vitae Limited	164.88
Interdum Feugiat Sed Associates	164.86
Mauris Incorporated	158.50
Mattis Foundation	155.57
Quisque Libero LLC	155.44
Non Justo Corp.	151.32
Nec Luctus LLC	124.06

#	Time	Action	Message
5	15:35:04	SELECT AVG(amount), company_name FROM transaction, company WHERE company.id = transaction.company_id AND declined = 0 GROUP BY c...	100 row(s) returned
6	15:40:38	SELECT COUNT(id) FROM transactions.company GROUP BY id HAVING COUNT(id) > 1	0 row(s) returned
7	15:43:03	SELECT DATE(timestamp), SUM(amount) FROM transaction WHERE declined = 0 GROUP BY DATE(timestamp) ORDER BY SUM(amount) DESC LI...	5 row(s) returned
8	15:49:06	SELECT AVG(amount), country FROM transaction, company WHERE company.id = transaction.company_id GROUP BY company.country ORDER B...	15 row(s) returned
9	15:49:20	SELECT AVG(amount), country FROM transaction, company WHERE company.id = transaction.company_id GROUP BY company.country ORDER B...	15 row(s) returned
10	15:49:34	SELECT AVG(amount), country FROM transaction, company WHERE company.id = transaction.company_id AND declined = 0 GROUP BY company....	15 row(s) returned
11	15:53:10	SELECT company_name, SUM(amount) FROM company, (SELECT * FROM transaction WHERE declined = 0 AND amount BETWEEN 100 AND 20...	23 row(s) returned

Ex2

En este caso haremos un **SELECT DISTICNT** para que nos de los nombres de las compañías. En la cláusula **WHERE** especificaremos las fechas que nos interesan, además de condicionar que la compra se realizó y especificar la relación entre las tablas.