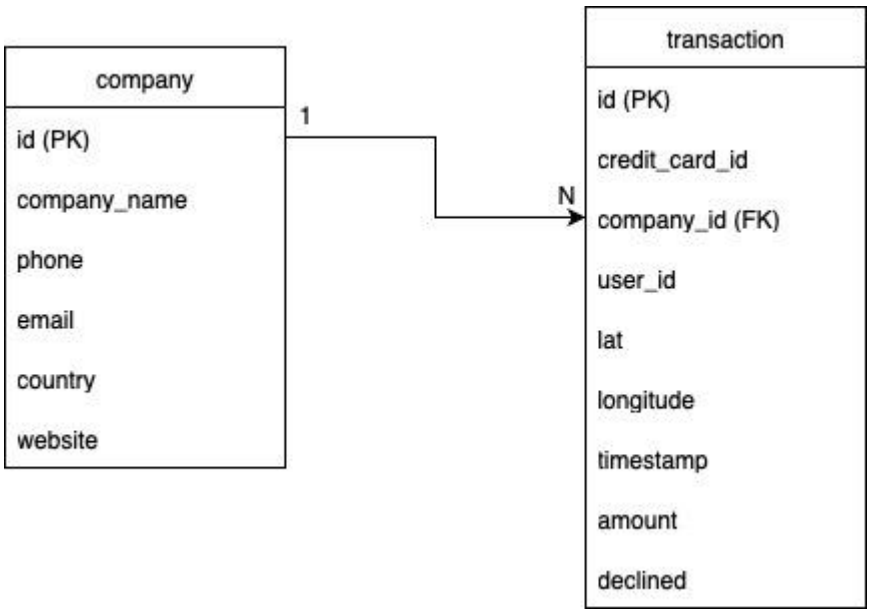
Tarea S2.01. Nociones básicas de SQL

Repasar las nociones básicas para el uso de bases de datos relacionales. En este sprint, iniciarás tu experiencia práctica con una base de datos que contiene información de una empresa dedicada a la venta de productos en línea. En esta actividad, te enfocarás en datos relacionados con las transacciones efectuadas y la información corporativa de las empresas que participaron.

Nivel 1

Ejercicio 1

A partir de los documentos adjuntos (estructura_dades y dades_introduir), importa las dos tablas. Muestra las características principales del esquema creado y explica las diferentes tablas y variables que existen. Asegúrate de incluir un diagrama que ilustre la relación entre las diferentes tablas y variables.



En esta base de datos nos encontramos con dos tablas, una tabla de hechos llamada ***transaction*** y otra de dimensión llamada ***company***. La relación es de **1 a N** teniendo para una sola compañía muchas transacciones.

En la tabla company encontramos las siguientes variables:

- **id** - Es la primary key de la tabla, es un tipo VARCHAR.
- **company_name** - Es tipo VARCHAR y almacena el nombre de cada compañía.
- **phone** - Es tipo VARCHAR y almacena el número de teléfono de la compañía.
- **email** - Es tipo VARCHAR y almacena el email de la compañía.
- **country** - Es tipo VARCHAR y almacena el país donde está ubicada la compañía.
- **website** - Es tipo VARCHAR y almacena el sitio web de la compañía.

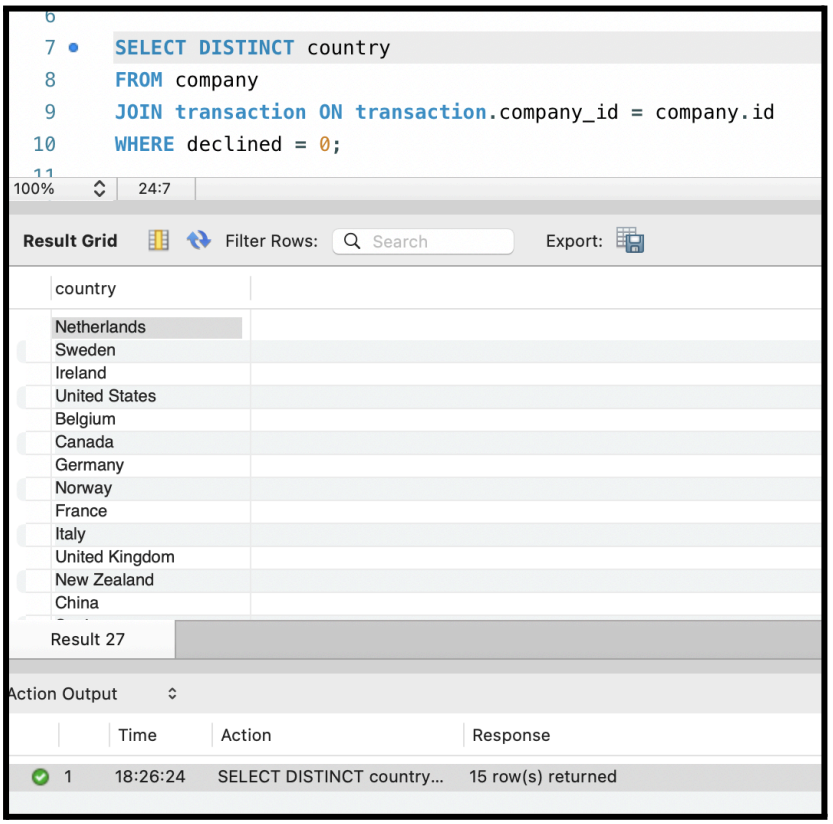
En la tabla transaction encontramos las siguientes variables:

- **id** - Es la primary key de la tabla, es un tipo VARCHAR
- **credit_card_id** - Es tipo VARCHAR y almacena el número de id de la tarjeta utilizada para la transacción, podría ser una FOREIGN KEY que guarde relación con una tabla de dimensión de tarjetas de crédito, pero no tenemos esa tabla para el ejercicio.
- **company_id** - Es una FOREIGN KEY que guarda relación con la PRIMARY KEY id de la tabla company, es tipo VARCHAR
- **user_id** - Es tipo INT y almacena el número de id de qué usuario realizó la transacción, podría ser una FOREIGN KEY que guarde relación con una tabla de dimensión de usuarios, pero no tenemos esa tabla para el ejercicio.
- **lat** - Es de tipo FLOAT, y sirve para determinar la latitud de coordenadas en un mapa.
- **longitude** - Es de tipo FLOAT, y sirve para determinar la longitud de coordenadas en un mapa.
- **timestamp** - Es de tipo TIMESTAMP y sirve para guardar la hora exacta que fue hecha la transacción.
- **amount** - Es de tipo DECIMAL y almacena el monto de la transacción, es decir cuanto fue gastado.
- **declined** - Es de tipo BOOLEAN y nos dice si la transacción fue aceptada con un “0” o rechazada con un “1”.

Ejercicio 2

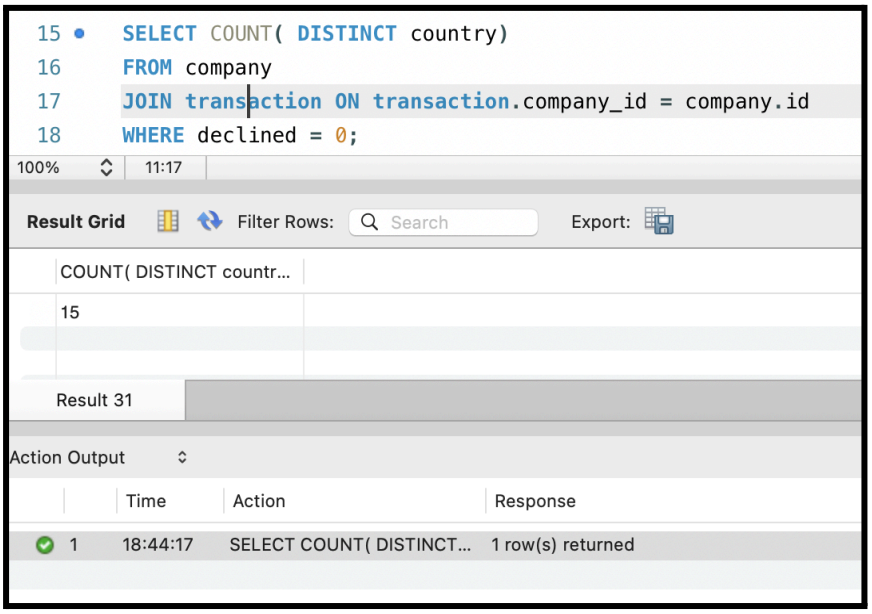
Utilizando JOIN realizarás las siguientes consultas:

- Listado de los países que están generando ventas.



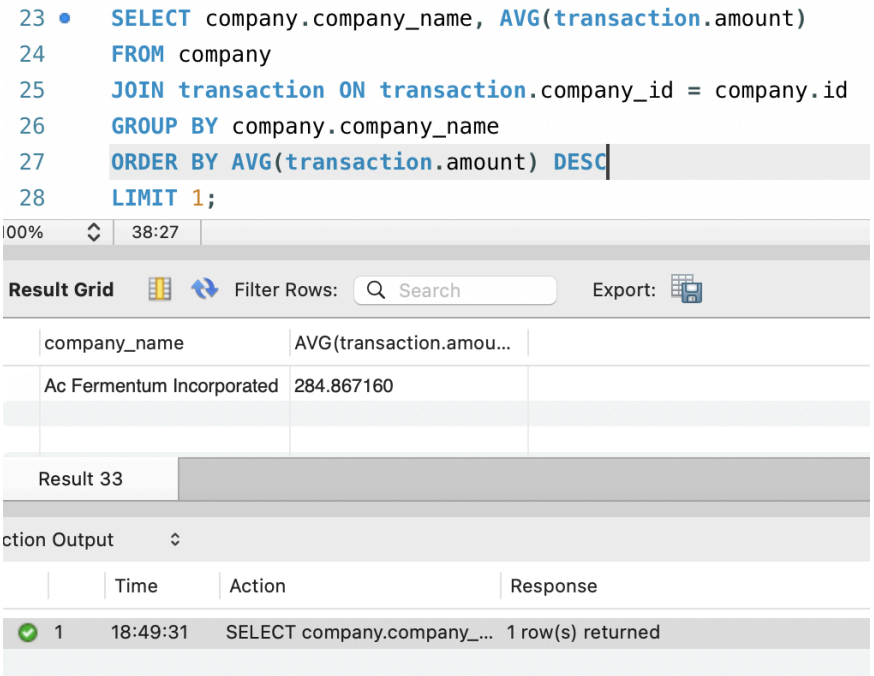
Aquí para saber cuales son los países que actualmente están generando ventas hice un *SELECT DISTINCT* (para que muestre países unicos y no se dupliquen) de la columna ***country*** de la tabla ***company*** y la junté con la tabla de transacciones uniendo por el ***id*** de cada compañía para que seleccione solo aquellas compañias que tienen transacciones registradas en la tabla de hechos. Además agregue un filtro para que no tome en cuenta las transacciones que han sido declinadas, ya que no contarían como una generación de venta.

- Desde cuántos países se generan las ventas.



Para calcular la cantidad de países que generan esas ventas de la consulta anterior simplemente aplique un COUNT con un DISTINCT para contar la cantidad de países sin que se dupliquen.

- Identifica la compañía con la media más alta de ventas.



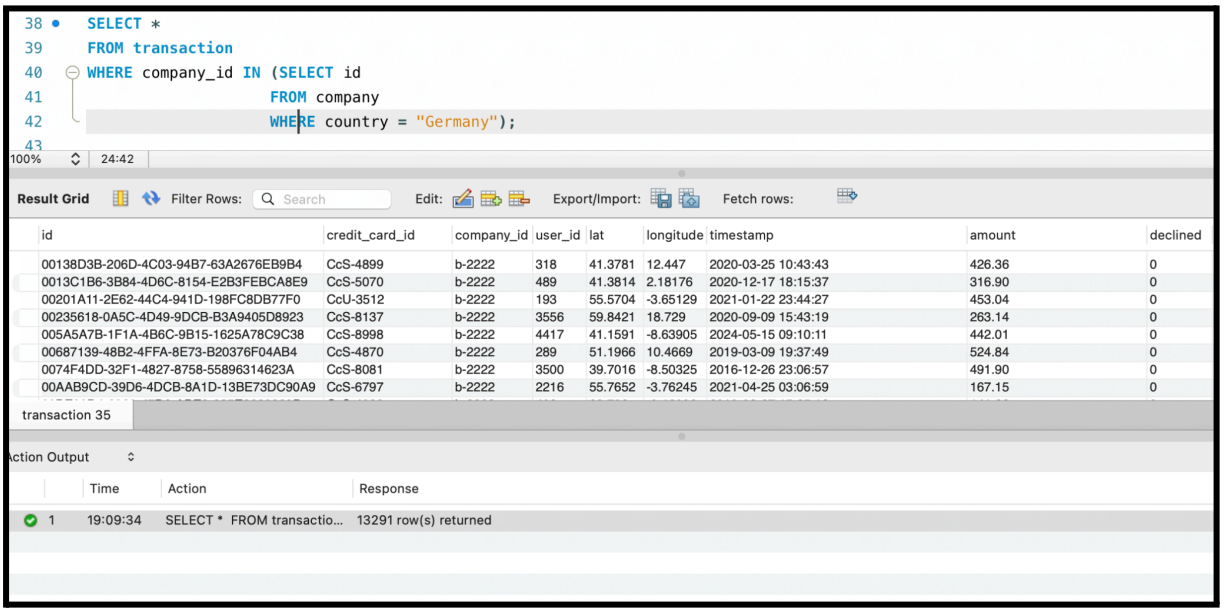
Aquí coloco en el ***SELECT*** la columna de los nombres de las compañías junto con el cálculo del promedio de los montos de las transacciones. Procedo a hacer ***JOIN*** de las dos tablas igualando en el *id* de las compañías y luego a agrupar por el nombre de la compañía.

Finalmente ordenamos de mayor a menor por el monto promedio de cada compañía y limito el resultado a solo el primero para saber cual es la compañía con la media más alta de ventas.

Ejercicio 3

Utilizando solo subconsultas (sin utilizar JOIN):

- Muestra todas las transacciones realizadas por empresas de Alemania.



```
38 •  SELECT *
39    FROM transaction
40    WHERE company_id IN (SELECT id
41                         FROM company
42                         WHERE country = "Germany");
43
```

100% 24:42

Result Grid | Filter Rows: Search | Edit: | Export/Import: | Fetch rows:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
00138D3B-206D-4C03-94B7-63A2676EB9B4	CcS-4899	b-2222	318	41.3781	12.447	2020-03-25 10:43:43	426.36	0
0013C1B6-3B84-4D6C-8154-E2B3FEBCA8E9	CcS-5070	b-2222	489	41.3814	2.18176	2020-12-17 18:15:37	316.90	0
00201A11-2E62-44C4-941D-198FC8DB77F0	CcU-3512	b-2222	193	55.5704	-3.65129	2021-01-22 23:44:27	453.04	0
00235618-0A5C-4D49-9DCB-B3A9405D8923	CcS-8137	b-2222	3556	59.8421	18.729	2020-09-09 15:43:19	263.14	0
005A5A7B-1F1A-4B6C-9B15-1625A78C9C38	CcS-8998	b-2222	4417	41.1591	-8.63905	2024-05-15 09:10:11	442.01	0
00687139-48B2-4FFA-8E73-B20376F04AB4	CcS-4870	b-2222	289	51.1966	10.4669	2019-03-09 19:37:49	524.84	0
0074F4DD-32F1-4827-8758-55896314623A	CcS-8081	b-2222	3500	39.7016	-8.50325	2016-12-26 23:06:57	491.90	0
00AAB9CD-39D6-4DCB-8A1D-13BE73DC90A9	CcS-6797	b-2222	2216	55.7652	-3.76245	2021-04-25 03:06:59	167.15	0

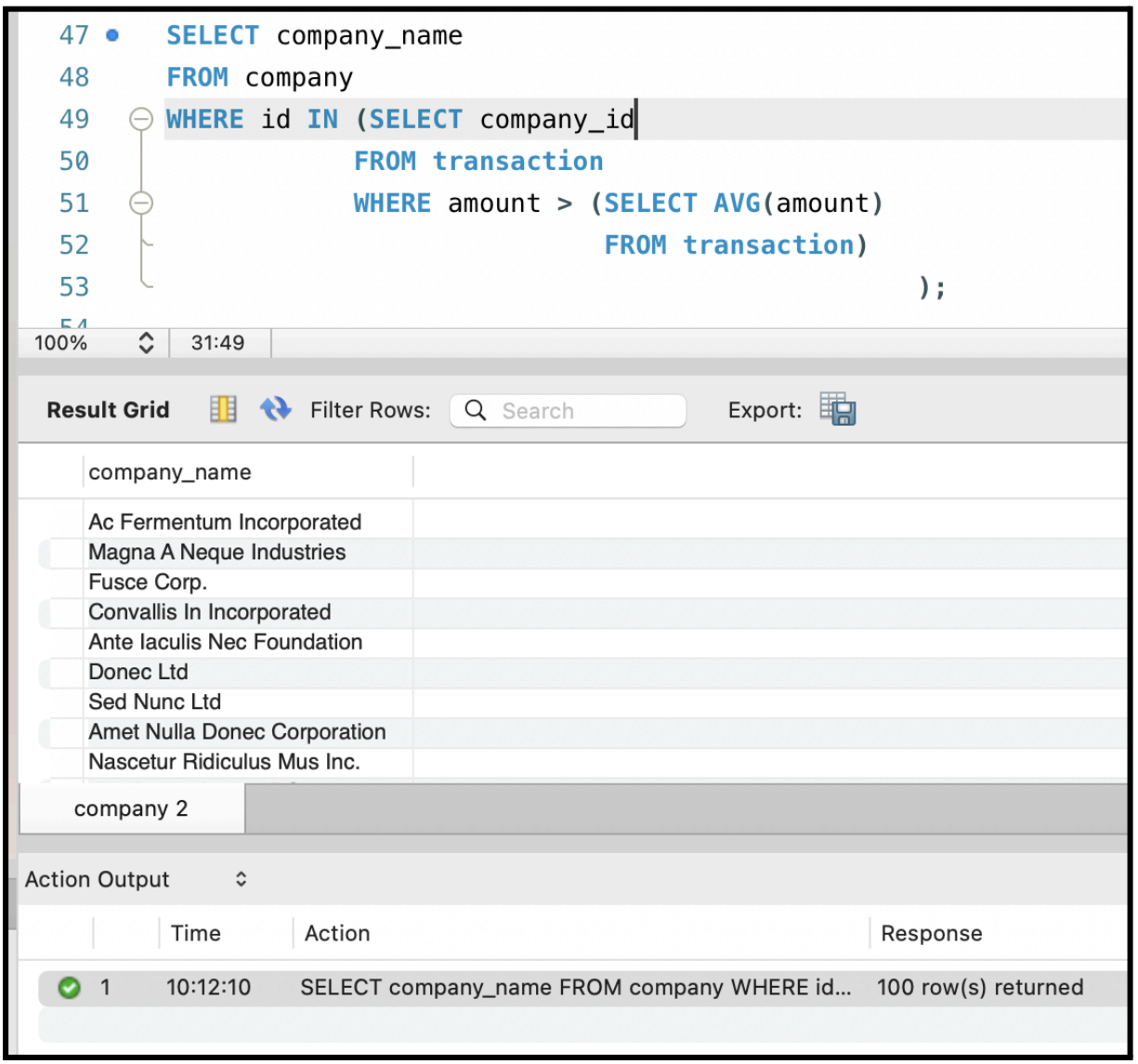
transaction 35

Action Output

	Time	Action	Response
1	19:09:34	SELECT * FROM transactio...	13291 row(s) returned

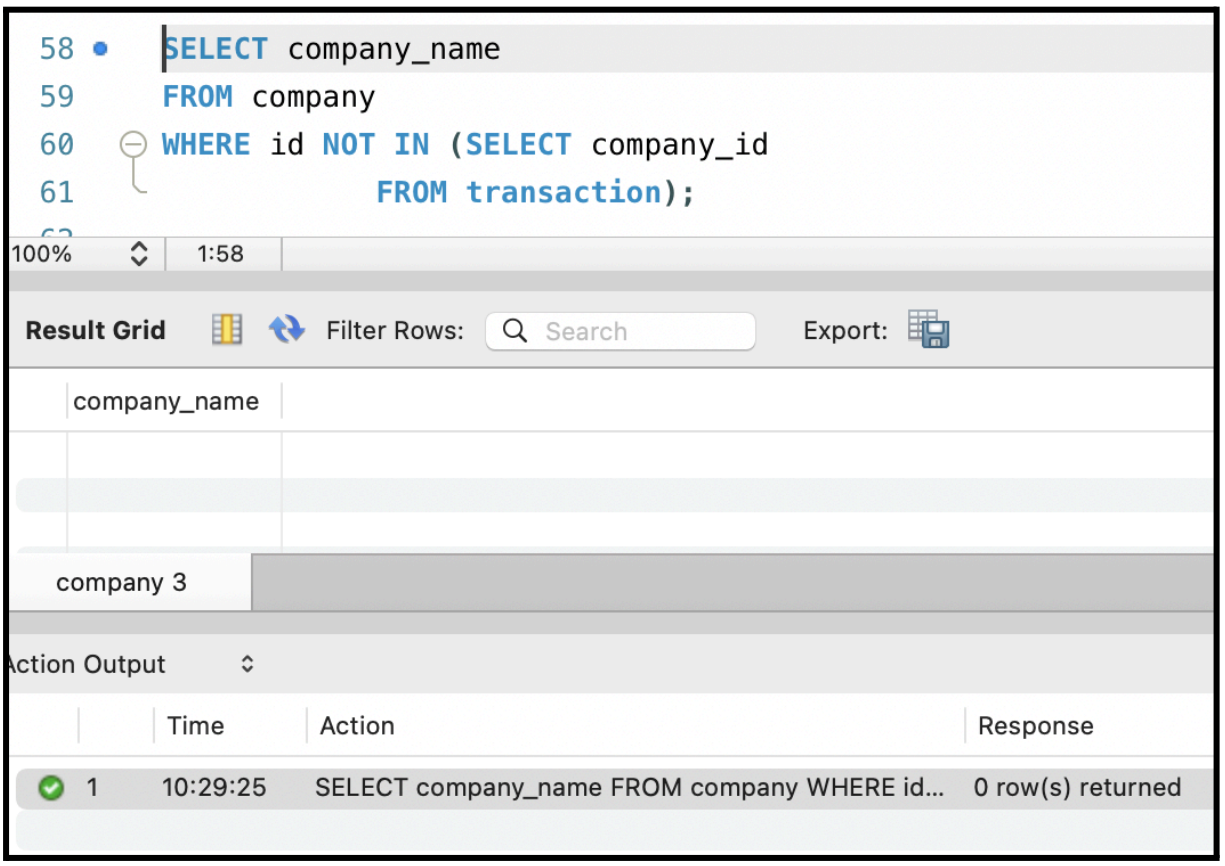
Primero selecciono todo para ver la totalidad de las transacciones y luego filtró por el campo de id de la compañía. Para poder filtrar correctamente con una subconsulta, hago que la subconsulta me dé únicamente los *id* de las compañías que tienen como país Alemania, entonces en la consulta principal puedo filtra con un IN para que me de como resultado solamente las transacciones que tengan en coincidencia los *company_id* con los *id* de las compañías de Alemania.

- Lista las empresas que han realizado transacciones por un *amount* superior a la media de todas las transacciones.



En la consulta principal seleccione *company_name* de la tabla *company*.
Luego, en el ***WHERE***, uso la columna *id* ***IN*** para quedarme solo con las compañías cuyo *id* aparece en la tabla transaction.
La subconsulta devuelve los *company_id* de las transacciones cuyo *amount* es mayor que la media de todos los *amount* en la tabla transaction (calculada en otra subconsulta interna).
El resultado es la lista de nombres de las compañías que han tenido al menos una transacción por encima del promedio general.

- Se eliminarán del sistema las empresas que no tienen transacciones registradas, entrega el listado de estas empresas.

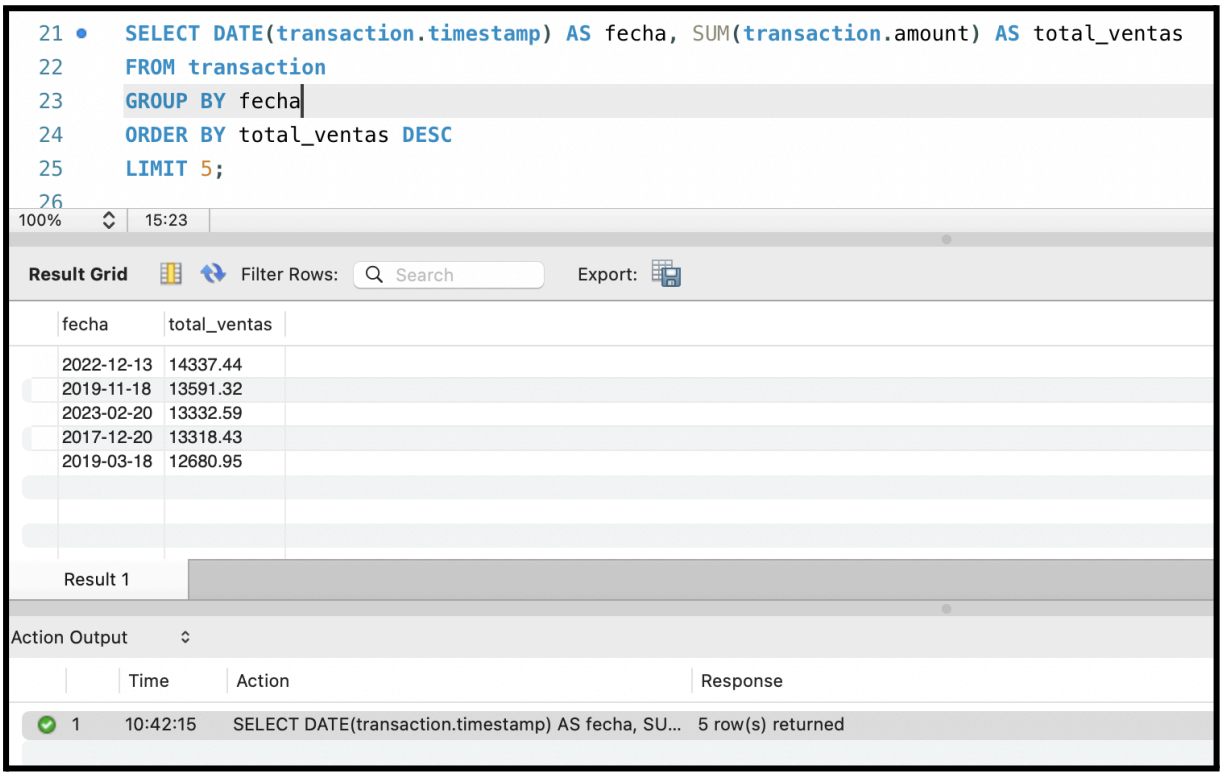


En esta consulta seleccione el nombre de las empresas de la tabla company y con el ***WHERE*** filtro por *id* con el ***NOT IN*** para quedarme solo con las empresas que no tengan registrada una transacción en la tabla transaction. El resultado me da cero, lo que significa que todas las empresas de la tabla company tienen un id que aparece en la tabla transaction y tienen una transacción registrada.

Nivel 2

Ejercicio 1

Identifica los cinco días en que se generó la mayor cantidad de ingresos en la empresa por ventas. Muestra la fecha de cada transacción junto con el total de las ventas.



En este ejercicio me encontré con un primer problema que es que la columna *timestamp* está en un formato que contiene hora, minutos y segundos; y eso me complica para poder filtrar ya que no podría agrupar por fechas. Investigando encontré el comando ***DATE()*** que transforma una columna que de tipo ***TIMESTAMP*** que tenga horas, minutos y segundos incluidos en la fecha en un campo que solamente tenga fecha con año, mes y día.

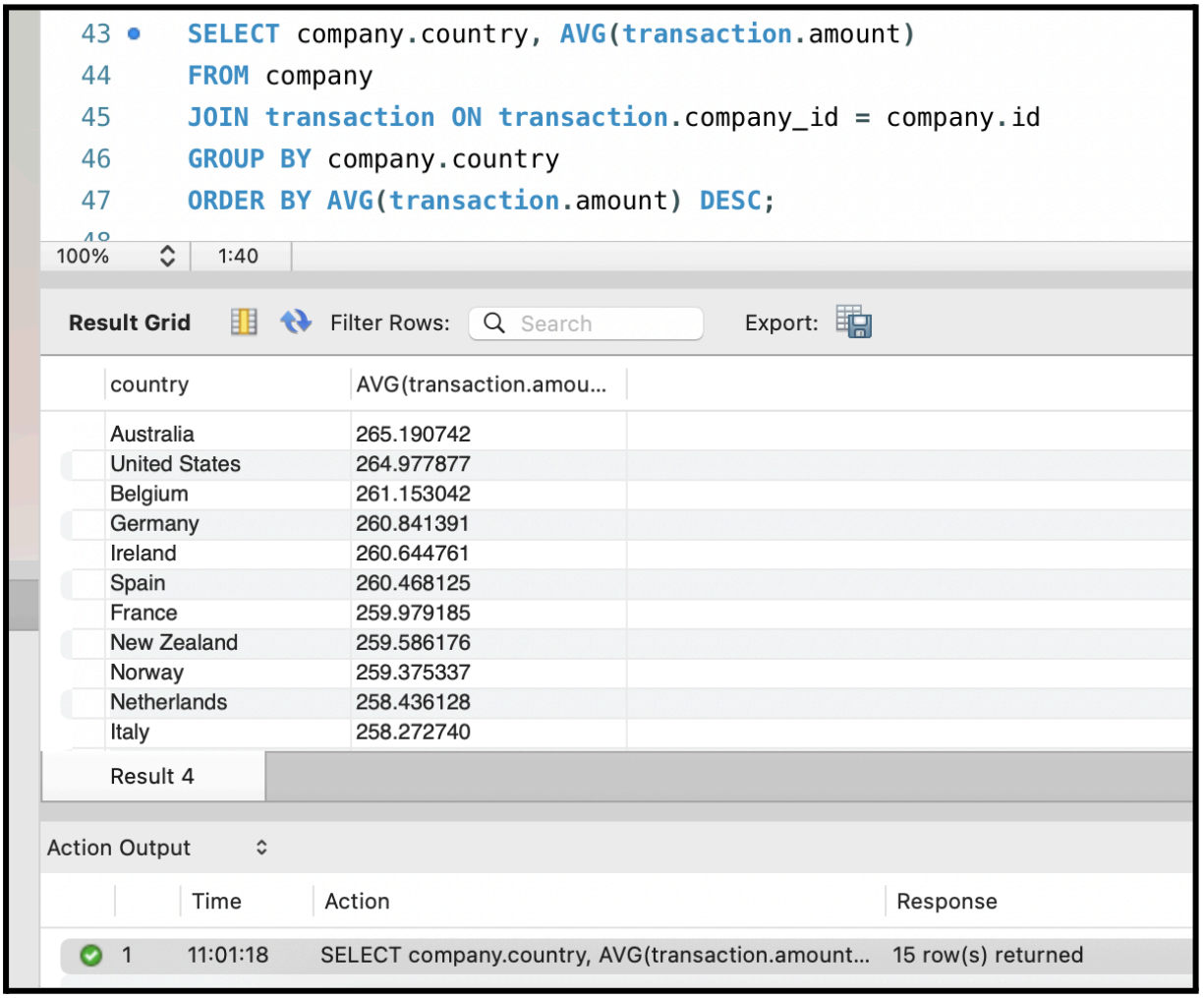
Una vez encontrado este comando, comienzo la consulta seleccionando la columna *timestamp* de la tabla de transacciones modificada por el comando ***DATE(timestamp)*** y le doy un alias de *fecha*, continuo seleccionando un ***SUM(amount)*** para hacer la sumatoria total de los montos de venta y le doy un alias de *total_ventas*.

Lo siguiente que hago es agrupar por *fecha* ya que ahora la columna tiene los datos limpios de solamente fecha, y que el ***SUM()*** me pueda calcular los montos totales de cada una de las fechas.

Y para poder identificar los cinco días que se generaron más ventas utilizo el ***ORDER BY*** con el *total_ventas* para que me ordene por monto y finalmente limito a 5 con ***LIMIT*** para tener el resultado.

Ejercicio 2

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

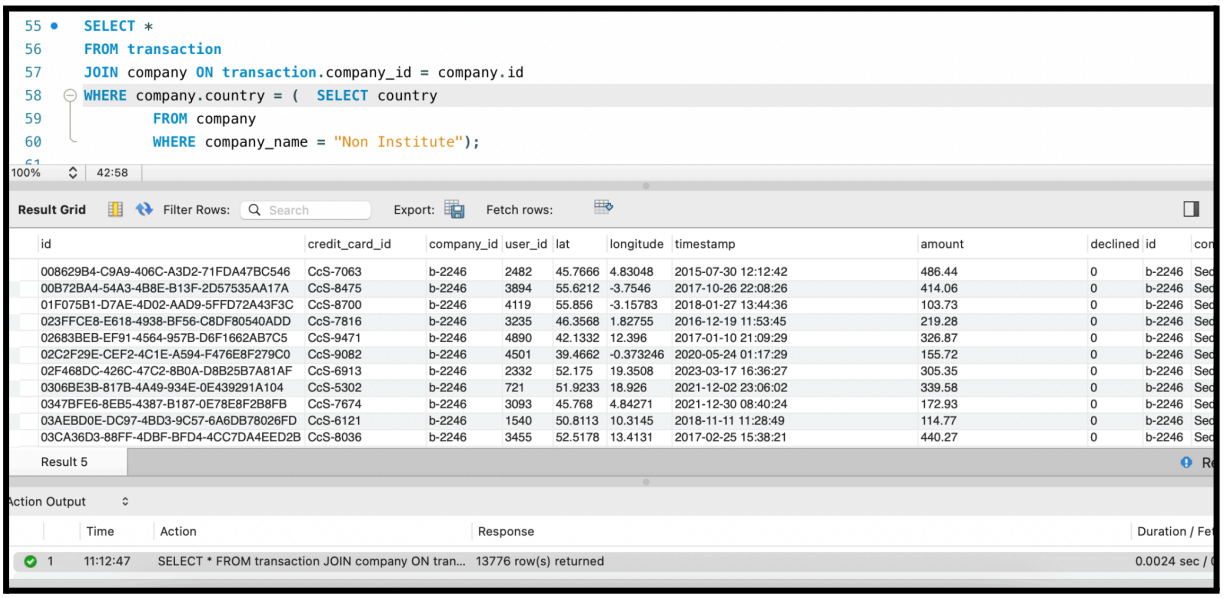


Para calcular la media de ventas por país primero pongo en el ***SELECT*** la columna *country* de la tabla company y el promedio de *amount* de la tabla transaction para que me de la media de las ventas. Luego de hacer el ***JOIN*** de las dos tablas por *id* de empresa paso a agrupar con el ***GROUP BY*** por *country* para que el cálculo de la media de ventas sea por país. Finalmente utilizo el ***ORDER BY DESC*** con la media de ventas para que el resultado esté ordenado de mayor a menor.

Ejercicio 3

En tu empresa, se plantea un nuevo proyecto para lanzar algunas campañas publicitarias para hacer competencia a la compañía "Non Institute". Para ello, te piden la lista de todas las transacciones realizadas por empresas que están situadas en el mismo país que esta compañía.

- Muestra el listado aplicando JOIN y subconsultas.



```
SELECT *
FROM transaction
JOIN company ON transaction.company_id = company.id
WHERE company.country = (  SELECT country
        FROM company
        WHERE company_name = "Non Institute");
```

Result Grid

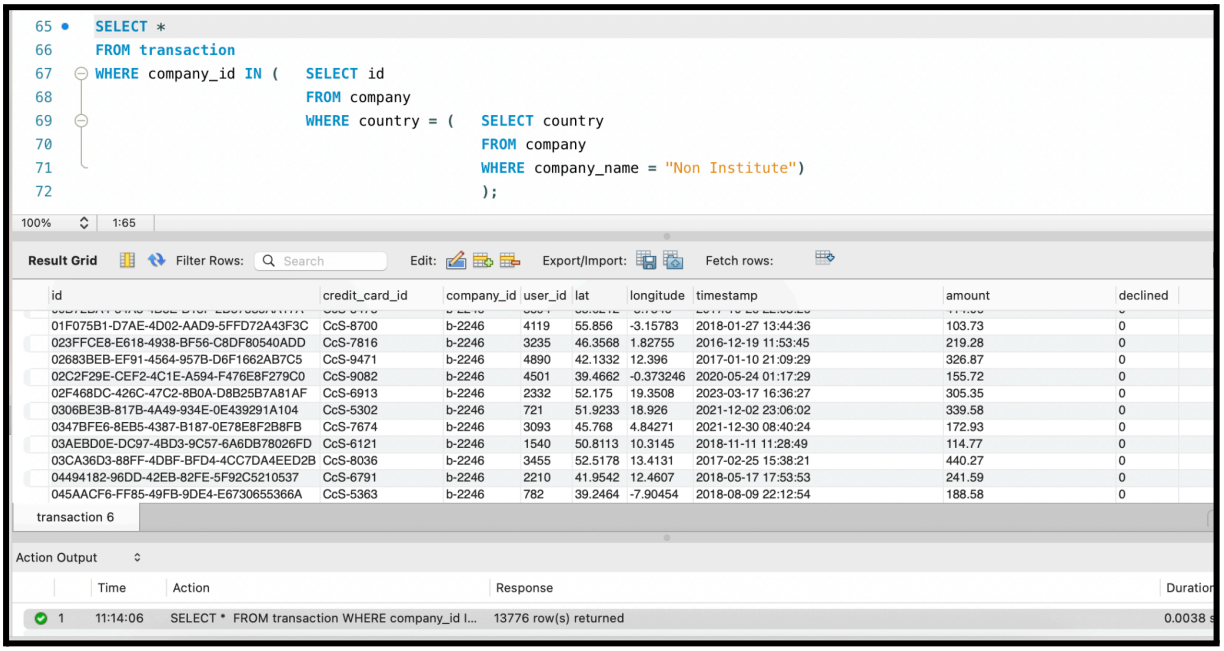
id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	id	con
008629B4-C9A9-406C-A3D2-71FDA47BC546	CcS-7063	b-2246	2482	45.7666	4.83048	2015-07-30 12:12:42	486.44	0	b-2246	Seq
00B72BA4-54A3-4B8E-B13F-2D57535AA17A	CcS-8475	b-2246	3894	55.6212	-3.7546	2017-10-26 22:08:26	414.06	0	b-2246	Seq
01F075B1-D7AE-4D02-AAD9-5FFD72A43F3C	CcS-8700	b-2246	4119	55.856	-3.15783	2018-01-27 13:44:36	103.73	0	b-2246	Seq
023FFCE8-E618-4938-BF56-C8DF80540ADD	CcS-7816	b-2246	3235	46.3568	1.82755	2016-12-19 11:53:45	219.28	0	b-2246	Seq
02683BEB-EF91-4564-957B-D6F1662AB7C5	CcS-9471	b-2246	4890	42.1332	12.396	2017-01-10 21:09:29	326.87	0	b-2246	Seq
02C2F29E-CEF2-4C1E-A594-F476E8F279C0	CcS-9082	b-2246	4501	39.4662	-0.373246	2020-05-24 01:17:29	155.72	0	b-2246	Seq
02F468DC-426C-47C2-8B0A-D8B25B7A81AF	CcS-6913	b-2246	2332	52.175	19.3508	2023-03-17 16:36:27	305.35	0	b-2246	Seq
0306BE3B-817B-4A49-934E-0E439291A104	CcS-5302	b-2246	721	51.9233	18.926	2021-12-02 23:06:02	339.58	0	b-2246	Seq
0347BFE6-8EB5-4387-B187-0E78E8F2B8FB	CcS-7674	b-2246	3093	45.768	4.84271	2021-12-30 08:40:24	172.93	0	b-2246	Seq
03AEBD0E-DC97-4BD3-9C57-6A6DB78026FD	CcS-6121	b-2246	1540	50.8113	10.3145	2018-11-11 11:28:49	114.77	0	b-2246	Seq
03CA36D3-88FF-4DBF-BFD4-4CC7DA4EED2B	CcS-8036	b-2246	3455	52.5178	13.4131	2017-02-25 15:38:21	440.27	0	b-2246	Seq

Result 5

Action Output

	Time	Action	Response	Duration / Fe
1	11:12:47	SELECT * FROM transaction JOIN company ON tran...	13776 row(s) returned	0.0024 sec / (

- Muestra el listado aplicando solamente subconsultas.



```
SELECT *
FROM transaction
WHERE company_id IN (   SELECT id
                        FROM company
                        WHERE country = (   SELECT country
                                            FROM company
                                            WHERE company_name = "Non Institute")
                                            );
```

Result Grid

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
01F075B1-D7AE-4D02-AAD9-5FFD72A43F3C	CcS-8700	b-2246	4119	55.856	-3.15783	2018-01-27 13:44:36	103.73	0
023FFCE8-E618-4938-BF56-C8DF80540ADD	CcS-7816	b-2246	3235	46.3568	1.82755	2016-12-19 11:53:45	219.28	0
02683BEB-EF91-4564-957B-D6F1662AB7C5	CcS-9471	b-2246	4890	42.1332	12.396	2017-01-10 21:09:29	326.87	0
02C2F29E-CEF2-4C1E-A594-F476E8F279C0	CcS-9082	b-2246	4501	39.4662	-0.373246	2020-05-24 01:17:29	155.72	0
02F468DC-426C-47C2-8B0A-D8B25B7A81AF	CcS-6913	b-2246	2332	52.175	19.3508	2023-03-17 16:36:27	305.35	0
0306BE3B-817B-4A49-934E-0E439291A104	CcS-5302	b-2246	721	51.9233	18.926	2021-12-02 23:06:02	339.58	0
0347BFE6-8EB5-4387-B187-0E78E8F2B8FB	CcS-7674	b-2246	3093	45.768	4.84271	2021-12-30 08:40:24	172.93	0
03AEBD0E-DC97-4BD3-9C57-6A6DB78026FD	CcS-6121	b-2246	1540	50.8113	10.3145	2018-11-11 11:28:49	114.77	0
03CA36D3-88FF-4DBF-BFD4-4CC7DA4EED2B	CcS-8036	b-2246	3455	52.5178	13.4131	2017-02-25 15:38:21	440.27	0
04494182-96DD-42EB-82FE-5F92C5210537	CcS-6791	b-2246	2210	41.9542	12.4607	2018-05-17 17:53:53	241.59	0
045AACF6-FF85-49FB-9DE4-E6730655366A	CcS-5363	b-2246	782	39.2464	-7.90454	2018-08-09 22:12:54	188.58	0

transaction 6

Action Output

	Time	Action	Response	Duration
1	11:14:06	SELECT * FROM transaction WHERE company_id I...	13776 row(s) returned	0.0038 s

En estas consultas seleccione todas las columnas de la tabla transaction para mostrar el listado completo de transacciones.

En la primera consulta, utilizo un ***JOIN*** con la tabla company para poder filtrar por país. En el ***WHERE*** comparo el país de cada compañía con el resultado de una subconsulta que obtiene el país de la empresa "Non Institute".

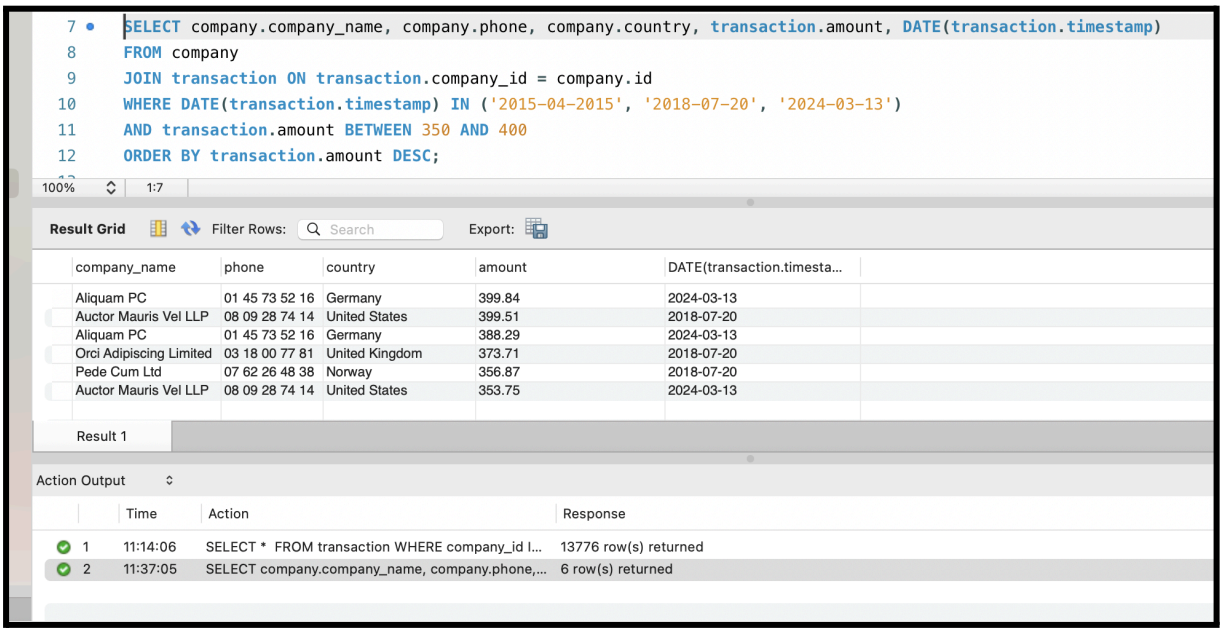
En la segunda consulta, no uso ***JOIN***, sino que filtro directamente por *company_id* con un ***WHERE IN***. La subconsulta devuelve los *id* de las compañías cuyo país coincide con el país de "Non Institute", que a su vez se obtiene con otra subconsulta interna.

De esta forma, en los dos casos obtengo todas las transacciones realizadas por empresas que están en el mismo país que "Non Institute".

Nivel 3

Ejercicio 1

Presenta el nombre, teléfono, país, fecha y *amount*, de aquellas empresas que realizaron transacciones con un valor comprendido entre 350 y 400 euros y en alguna de estas fechas: 29 de abril de 2015, 20 de julio de 2018 y 13 de marzo de 2024. Ordena los resultados de mayor a menor cantidad.



```
7  SELECT company.company_name, company.phone, company.country, transaction.amount, DATE(transaction.timestamp)
8  FROM company
9  JOIN transaction ON transaction.company_id = company.id
10 WHERE DATE(transaction.timestamp) IN ('2015-04-2015', '2018-07-20', '2024-03-13')
11 AND transaction.amount BETWEEN 350 AND 400
12 ORDER BY transaction.amount DESC;
```

100% 1:7

Result Grid Filter Rows: Search Export:

company_name	phone	country	amount	DATE(transaction.timesta...
Aliquam PC	01 45 73 52 16	Germany	399.84	2024-03-13
Auctor Mauris Vel LLP	08 09 28 74 14	United States	399.51	2018-07-20
Aliquam PC	01 45 73 52 16	Germany	388.29	2024-03-13
Orci Adipiscing Limited	03 18 00 77 81	United Kingdom	373.71	2018-07-20
Pede Cum Ltd	07 62 26 48 38	Norway	356.87	2018-07-20
Auctor Mauris Vel LLP	08 09 28 74 14	United States	353.75	2024-03-13

Result 1

Action Output

	Time	Action	Response
1	11:14:06	SELECT * FROM transaction WHERE company_id I...	13776 row(s) returned
2	11:37:05	SELECT company.company_name, company.phone,...	6 row(s) returned

En esta consulta seleccione de la tabla company el nombre, el teléfono y el país de las compañías, y de la tabla transaction la fecha con ***DATE(transaction.timestamp)*** para limpiar el campo y que solo tome en cuenta la fecha sin las horas y minutos y también el *amount*.

Uno ambas tablas con un ***JOIN*** usando la relación entre las *id* de las empresas.

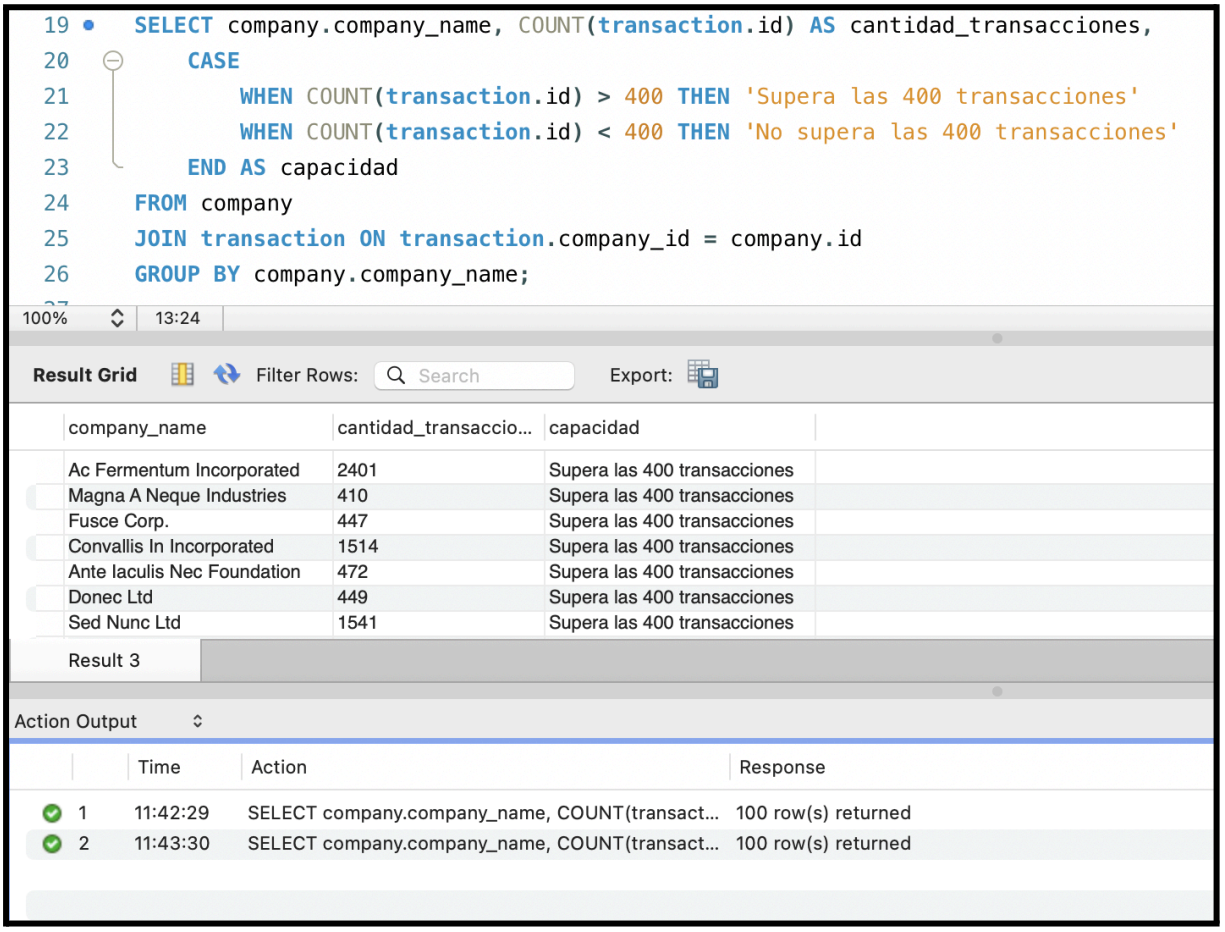
Después aplico dos filtros en el ***WHERE***:

- Con ***IN*** limito las transacciones a que sean solo en las fechas indicadas: 29/04/2015, 20/07/2018 y 13/03/2024.
- Con ***BETWEEN*** restrinjo los valores de amount a que estén comprendidos entre 350 y 400.

Finalmente, ordeno el resultado de forma descendente (***ORDER BY DESC***) según el valor del amount, para que aparezcan primero las transacciones de mayor cantidad.

Ejercicio 2

Necesitamos optimizar la asignación de los recursos y dependerá de la capacidad operativa que se requiera, por lo que te piden la información sobre la cantidad de transacciones que realizan las empresas. Pero el departamento de recursos humanos es exigente y quiere un listado de las empresas donde especifiques si tienen más de 400 transacciones o menos.



```
19  SELECT company.company_name, COUNT(transaction.id) AS cantidad_transacciones,
20      CASE
21          WHEN COUNT(transaction.id) > 400 THEN 'Supera las 400 transacciones'
22          WHEN COUNT(transaction.id) < 400 THEN 'No supera las 400 transacciones'
23      END AS capacidad
24  FROM company
25  JOIN transaction ON transaction.company_id = company.id
26  GROUP BY company.company_name;
```

100% 13:24

Result Grid Filter Rows: Search Export:

company_name	cantidad_transaccio...	capacidad
Ac Fermentum Incorporated	2401	Supera las 400 transacciones
Magna A Neque Industries	410	Supera las 400 transacciones
Fusce Corp.	447	Supera las 400 transacciones
Convallis In Incorporated	1514	Supera las 400 transacciones
Ante Iaculis Nec Foundation	472	Supera las 400 transacciones
Donec Ltd	449	Supera las 400 transacciones
Sed Nunc Ltd	1541	Supera las 400 transacciones

Result 3

Action Output

	Time	Action	Response
1	11:42:29	SELECT company.company_name, COUNT(transact...	100 row(s) returned
2	11:43:30	SELECT company.company_name, COUNT(transact...	100 row(s) returned

En esta consulta selecciono el *company_name* de la tabla company y calculo con ***COUNT(transaction.id)*** la cantidad total de transacciones realizadas por cada empresa.

Uno las tablas con un ***JOIN*** entre las *id* de empresas, y agrupo los resultados por ***company_name*** para obtener el conteo por compañía.

Además, utilizo un CASE para crear una columna extra llamada capacidad, donde indico si la empresa tiene más de 400 transacciones (Supera las 400 transacciones) o si tiene menos (No supera las 400 transacciones).

De esta forma obtengo un listado de todas las empresas, con la cantidad de transacciones que han realizado y la clasificación que pide el departamento de recursos humanos