

SPRINT 2:

➤ NIVEL 1

EJERCICIO 1

En el Schema "transactions" tenemos dos tablas, la primera "company" y la segunda "transaction".

A. Tabla "company"

id	company_name	phone	email	country	website
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net	Germany	https://instagram.com/site
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@idcloud.org	Australia	https://whatsapp.com/group/9
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/sub/cars
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany	https://cnn.com/user/110
b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings
b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.couk	Norway	https://nytimes.com/user/110
b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
b-2250	Amet Nulla Donec Corporation	07 15 25 14 74	mattis.integer.eu@protonmail.net	Italy	https://netflix.com/sub/cars
b-2254	Nascetur Ridiculus Mus Inc.	06 26 87 61 84	suspendisse.dui@idcloud.net	United States	https://ebay.com/sub
b-2258	Vestibulum Lorem PC	02 02 87 33 40	aenean.massa.integer@aol.net	Belgium	https://pinterest.com/sub/cars
b-2262	Gravida Sagittis LLP	03 81 28 33 97	turpis.vitae@google.ca	Sweden	https://naver.com/site
b-2266	Mus Aenean Eget Foundation	06 25 15 52 43	mi.duis@hotmail.net	Sweden	https://instagram.com/group/9
b-2270	Dis Parturient Institute	05 36 29 78 74	purus@protonmail.org	Ireland	https://google.com/one
b-2274	Sed LLC	01 63 16 26 52	at@outlook.com	Belgium	https://reddit.com/fr

Como se puede observar, dentro de la tabla "company" se almacena información de distintas compañías.

La tabla contiene los siguientes campos:

- id: Este campo identifica de forma única cada registro dentro de la tabla "company". En otras palabras, es su PRIMARY KEY
- company_name: Nombre de la compañía
- phone: Teléfono de la compañía
- email: Correo electrónico de la compañía
- country: País donde está la compañía
- website: Dirección web de la compañía

De la consulta "SELECT *" se puede observar que la tabla contiene 100 registros.

B. Tabla "transaction"

Query 1 x SQL File 5*

```

1 • select *
2   from transaction;
3

```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
00043A49-2949-494B-A5DD-A5BAE3B819DD	CcS-9294	b-2458	4713	46.1999	1.43554	2024-08-28 07:16:46	395.43	0
000447FE-B650-4DCF-85DE-C7ED0EE1CAAD	CcS-5019	b-2370	438	41.5972	12.2218	2016-12-21 20:07:18	155.63	0
00045D68-ED2E-4F2F-8186-CEE074D875D0	CcS-6699	b-2390	2118	29.7573	-95.3796	2020-07-14 15:37:45	326.01	0
000481C3-1C26-4FEF-83A0-4CD0EB004BBD	CcS-6696	b-2230	2115	53.5489	-113.503	2017-09-04 19:44:53	161.60	0
00051AA4-9CBE-4268-8070-C38062A1B3E2	CcS-7606	b-2266	3025	52.2084	5.69081	2017-01-05 18:19:25	148.91	0
0008A312-EDFE-4A4F-8C99-E9C92EC3CA4D	CcU-3358	b-2598	215	53.5535	-113.499	2023-09-23 04:51:43	294.59	0
0009A151-9BCF-4E31-9053-A468FF77FAAB	CcS-7509	b-2546	2928	51.9362	5.34265	2023-12-31 00:06:36	383.63	0
0009D494-6245-4DF9-955D-2C084191CFFB	CcS-8483	b-2526	3902	45.492	-73.5706	2017-07-18 07:52:02	197.80	0
000A1DEC-CDB6-4AB2-A619-71DAB8D4A262	CcS-6467	b-2558	1886	55.7425	-3.30009	2018-09-08 05:29:58	339.94	0
000A1E64-1414-40B0-9D92-5678A4D958E2	CcS-5966	b-2550	1385	52.0821	5.28424	2022-09-17 04:02:19	369.71	0
000A5879-3472-41D9-AF60-42D3503B543C	CcU-4569	b-2590	42	39.949	-75.1719	2020-02-07 23:03:45	162.43	0
000AE0D4-F06E-4146-804A-5A2FC73110D7	CcS-8134	b-2426	3553	46.8055	-71.2149	2017-07-10 07:11:41	188.94	0
000BCBF8-A4AB-4E8E-A148-2E7A04E1B385	CcS-6020	b-2598	1439	33.4472	-112.072	2019-05-28 04:23:02	96.22	0
000BD852-E25B-4785-93D9-27051425D45B	CcS-5445	b-2330	864	51.4317	5.47489	2024-07-31 09:45:39	92.13	0

transaction 3 x

Output

Action Output

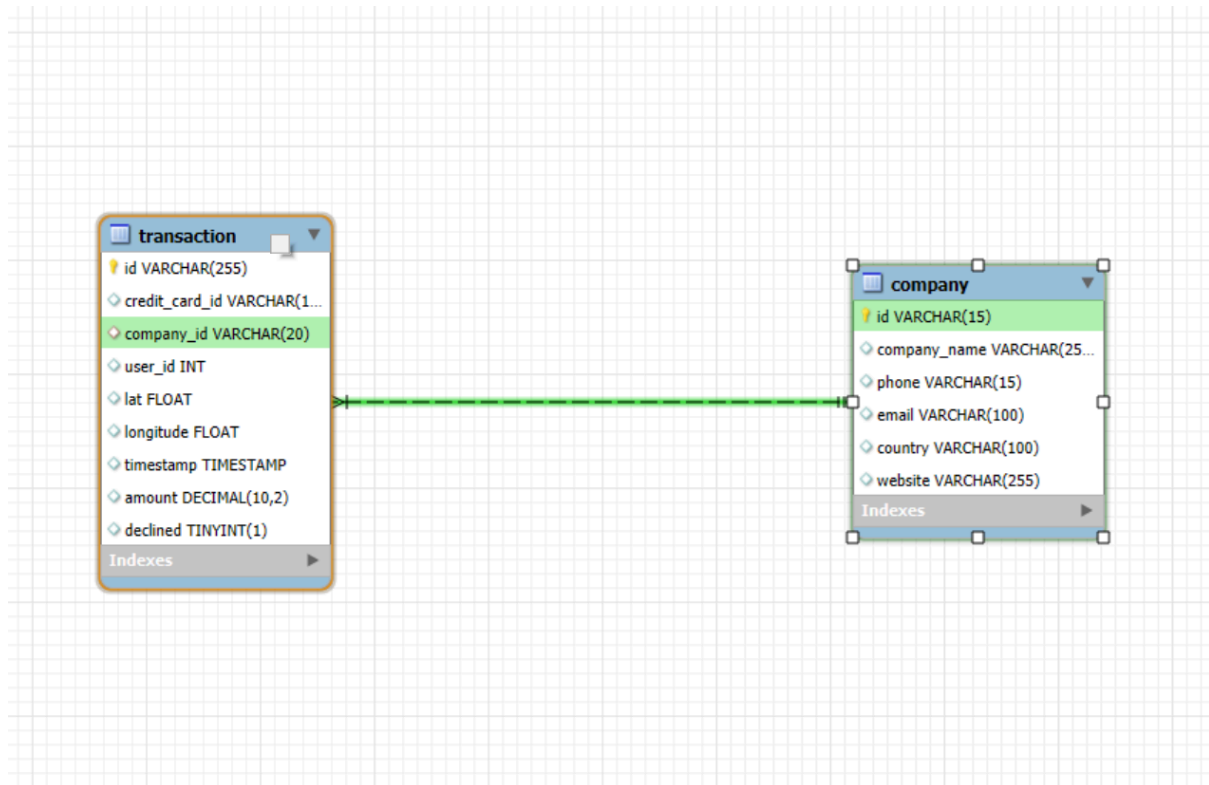
#	Time	Action	Message
1	11:46:22	select * from transaction	100000 row(s) returned

En la tabla transacción se almacenan los registros de las distintas transacciones realizadas. Esta tabla contiene los siguientes campos:

- **id:** Identificador único de cada registro dentro de la tabla
- **credit_card_id:** Hace referencia a otra tabla llamada `credit_card`, ya que utiliza el PRIMARY KEY de esa tabla. Funciona como FOREIGN KEY.
- **company_id:** También funciona como FOREIGN KEY, vinculando cada transacción con la compañía correspondiente.
- **user_id:** Este campo indica la existencia de otra tabla llamada `user`, ya que hace referencia al PRIMARY KEY de dicha tabla, actuando como FOREIGN KEY.
- **lat:** Registra la latitud asociada a la transacción, proporcionando una referencia geográfica.
- **longitude:** Registra la longitud asociada a la transacción, complementando la información geográfica.
- **timestamp:** Campo de tipo `TIMESTAMP` que almacena la fecha y hora exacta en la que se generó cada transacción.
- **amount:** Registra el monto económico de la transacción.

- **declined:** Indica el número de veces que una transacción ha sido declinada.

C. Diagrama

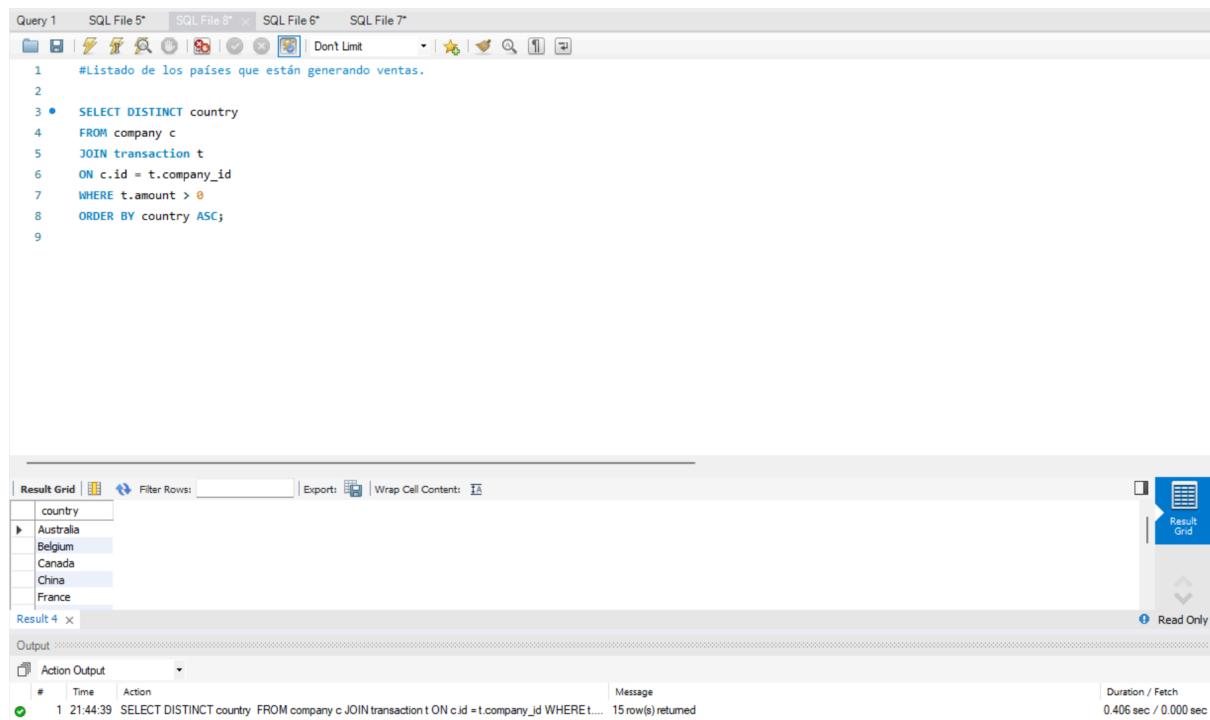


- Características principales del esquema:
 - Número de tablas: 2.
 - Claves primarias: id (tabla 'company'), id (tabla 'transaction').
 - Claves foráneas: company_id, user_id, credit_card_id
 - Relación establecida mediante FOREIGN KEY.
- Relación entre tablas:
 - Las tablas tienen una relación tipo de uno (company) a muchos (transaction). Es decir, una compañía puede tener varias transacciones. Pero una transacción puede tener solo una compañía.

EJERCICIO 2

Utilizando JOIN realizarás las siguientes consultas:

A. Listado de los países que están generando ventas.



The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
1 #Listado de los países que están generando ventas.
2
3 SELECT DISTINCT country
4 FROM company c
5 JOIN transaction t
6 ON c.id = t.company_id
7 WHERE t.amount > 0
8 ORDER BY country ASC;
9
```

The results pane displays a table with the following data:

country
Australia
Belgium
Canada
China
France

The output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	21:44:39	SELECT DISTINCT country FROM company c JOIN transaction t ON c.id = t.company_id WHERE t...	15 row(s) returned	0.406 sec / 0.000 sec

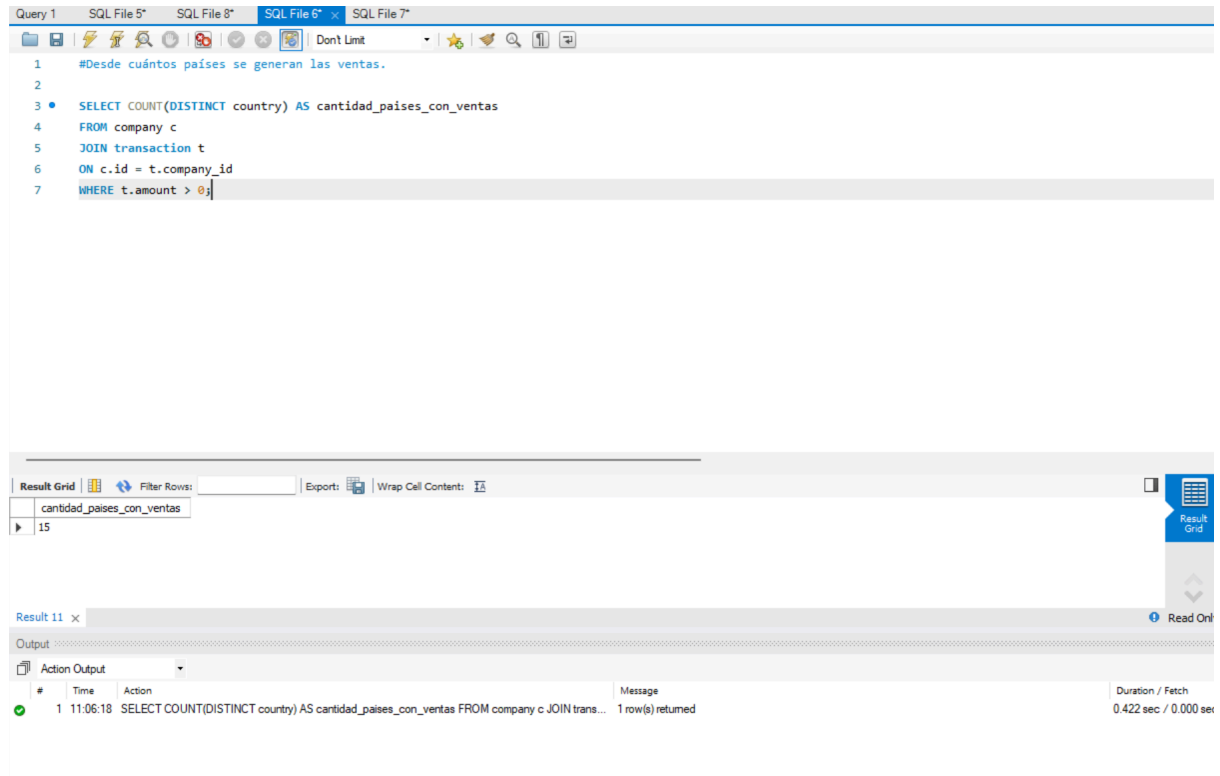
El objetivo principal es identificar qué países están generando ventas y mostrar sus nombres.

La información se encuentra en dos tablas: `company`, que contiene el campo `country`, y `transaction`, donde se registra el campo `amount`. Para relacionar ambas tablas, utilizamos un JOIN a través de la clave primaria `c.id` en `company` y la clave foránea `t.company_id` en `transacción`.

A fin de obtener únicamente los países con ventas, se aplica la condición `amount > 0`. De este modo, la consulta devuelve el listado de países que efectivamente han registrado ventas. En este caso, el resultado incluye 15 países.

Finalmente, se incorpora un `ORDER BY` para presentar los países en orden alfabético y facilitar la interpretación de los datos.

B. Desde cuántos países se generan las ventas.



```
1 #Desde cuántos países se generan las ventas.
2
3 • SELECT COUNT(DISTINCT country) AS cantidad_paises_con_ventas
4 FROM company c
5 JOIN transaction t
6 ON c.id = t.company_id
7 WHERE t.amount > 0;
```

cantidad_paises_con_ventas
15

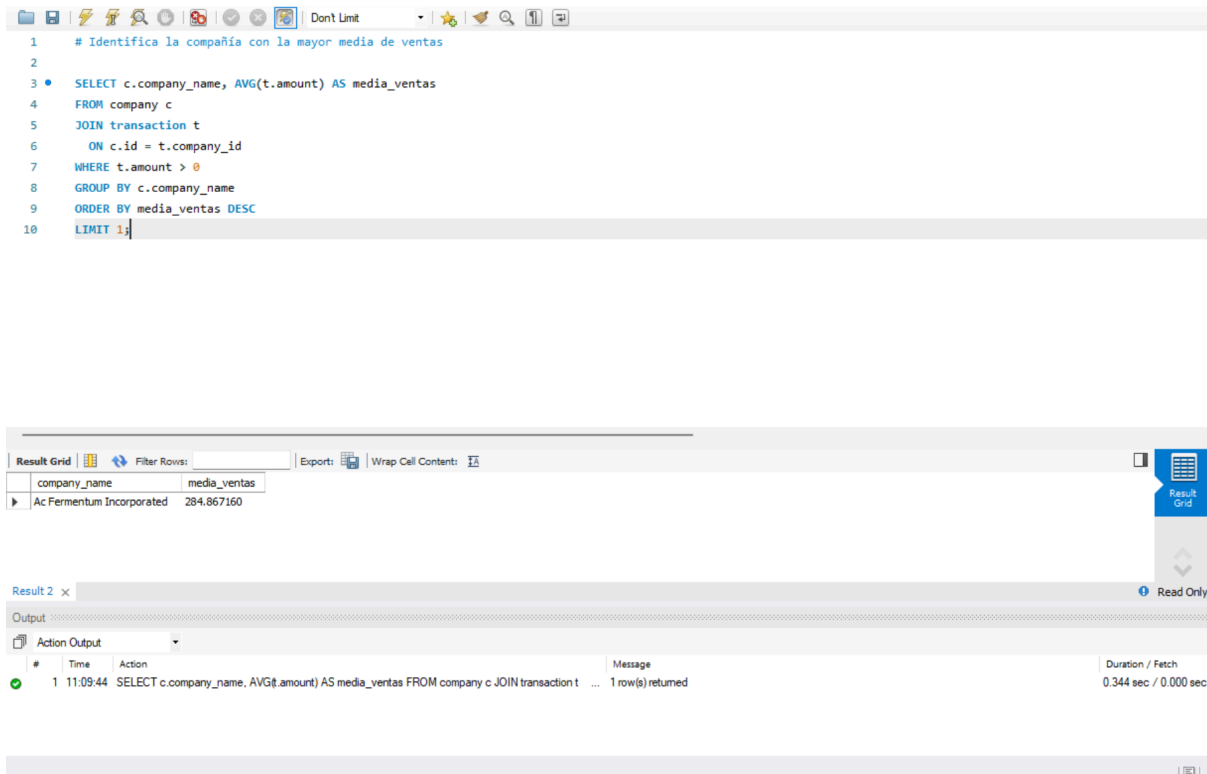
Result 11 x

Output

#	Time	Action	Message	Duration / Fetch
1	11:06:18	SELECT COUNT(DISTINCT country) AS cantidad_paises_con_ventas FROM company c JOIN trans...	1 row(s) returned	0.422 sec / 0.000 sec

Para determinar cuántos países están generando ventas, aplicamos la función COUNT sobre la consulta anterior. En lugar de mostrar el listado completo de países, esta instrucción devuelve únicamente el número total de países que registran ventas, lo que facilita una visión más sintética del resultado.

C. Identifica a la compañía con la mayor media de ventas.



The screenshot shows a SQL query editor with the following code:

```
1 # Identifica la compañía con la mayor media de ventas
2
3 • SELECT c.company_name, AVG(t.amount) AS media_ventas
4 FROM company c
5 JOIN transaction t
6 ON c.id = t.company_id
7 WHERE t.amount > 0
8 GROUP BY c.company_name
9 ORDER BY media_ventas DESC
10 LIMIT 1;
```

Below the editor, the 'Result Grid' shows the output of the query:

company_name	media_ventas
Ac Fermentum Incorporated	284.867160

The 'Action Output' section shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	11:09:44	SELECT c.company_name, AVG(t.amount) AS media_ventas FROM company c JOIN transaction t	1 row(s) returned	0.344 sec / 0.000 sec

El objetivo de este ejercicio es calcular la media de ventas por compañía y determinar cuál de ellas presenta el valor más alto. Para ello, se utiliza la función AVG sobre el campo amount, que contiene los importes de ventas, agrupando los resultados por el nombre de la compañía.

Para mostrar únicamente la compañía con la media más elevada, una opción sencilla es aplicar LIMIT 1. Sin embargo, esta aproximación devuelve solo un registro, incluso en el caso de que existan varias compañías con la misma media máxima.

En esos escenarios, resulta más adecuado emplear una subconsulta que identifique el valor máximo de la media y devuelva todas las compañías que lo compartan. La ventaja de esta alternativa es que asegura un resultado completo y justo, mientras que la primera opción es más directa pero restrictiva.

En este caso, no se ha desarrollado la solución con subconsulta porque el enunciado del ejercicio especifica que debía resolverse únicamente mediante JOIN.

#nota: importante incluir también el id por si llega a haber nombres duplicados o iguales

EJERCICIO 3

Utilizando sólo subconsultas (sin utilizar JOIN):

A. Muestra todas las transacciones realizadas por empresas de Alemania.

The screenshot shows a SQL IDE interface with a query editor and a results grid. The query is as follows:

```

1  #Muestra todas las transacciones realizadas por empresas de Alemania.
2
3  SELECT t.id, t.amount,
4         (SELECT c.company_name
5          FROM company c
6          WHERE c.id = t.company_id) AS company_name,
7         (SELECT c.country
8          FROM company c
9          WHERE c.id = t.company_id) AS country
10 FROM transaction t
11 WHERE t.company_id IN (
12     SELECT id
13     FROM company
14     WHERE country = 'Germany'
15 )
16
17
18
19

```

The results grid displays the following data:

id	amount	company_name	country
0013803B-206D-4C03-94B7-63A2676EB9B4	426.36	Ac Fermentum Incorporated	Germany
0013C1B6-3B84-4D6C-8154-E2B3FBCABE9	316.90	Ac Fermentum Incorporated	Germany
00201A11-2E62-44C4-941D-198FC8DB77F0	453.04	Ac Fermentum Incorporated	Germany
00235618-0A5C-4D49-9DCB-B3A9405D8923	263.14	Ac Fermentum Incorporated	Germany
005A5A7B-1F1A-4B6C-9B15-1625A78C9C38	442.01	Ac Fermentum Incorporated	Germany

The output pane shows the execution details:

```

# Time Action Message Duration / Fetch
1 12:36:53 SELECT t.id,t.amount, (SELECT c.company_name FROM company c WHERE c.id = t... 13291 row(s) returned 0.016 sec / 0.172 sec

```

El objetivo de este ejercicio es mostrar los campos `t.id`, `c.company_name` y `c.country` correspondientes a las transacciones realizadas por empresas de Alemania.

Para ello, se toma la tabla transacción como tabla principal, incluyendo los campos `t.id` y `t.amount`. A través de una subconsulta correlacionada, se obtienen los campos `company_name` y `country` de la tabla `company`, relacionando ambas tablas mediante `c.id = t.company_id`.

Además, en la cláusula `WHERE` se aplica otra subconsulta para filtrar únicamente las compañías ubicadas en Alemania, asegurando que solo se incluyan las transacciones de interés.

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

The screenshot shows a SQL IDE interface with a query editor at the top and a results pane at the bottom. The query editor contains the following SQL code:

```
1 #Lista las empresas que han realizado transacciones por un amount superior a la media de todas las transacciones.
2 SELECT company_name
3 FROM company
4 WHERE id IN (
5     SELECT company_id
6     FROM transaction
7     WHERE amount > (SELECT ROUND(AVG(t.amount), 2) FROM transaction t));
8
9
```

The results pane shows a table with the following data:

company_name
Ac Fermentum Incorporated
Magna A Neque Industries
Fusce Corp.
Conwallis In Incorporated
Ante Iaculis Nec Foundation

The output pane at the bottom shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	12:31:21	SELECT company_name FROM company WHERE id IN (SELECT company_id FROM transaction ...	100 row(s) returned	0.188 sec / 0.000 sec

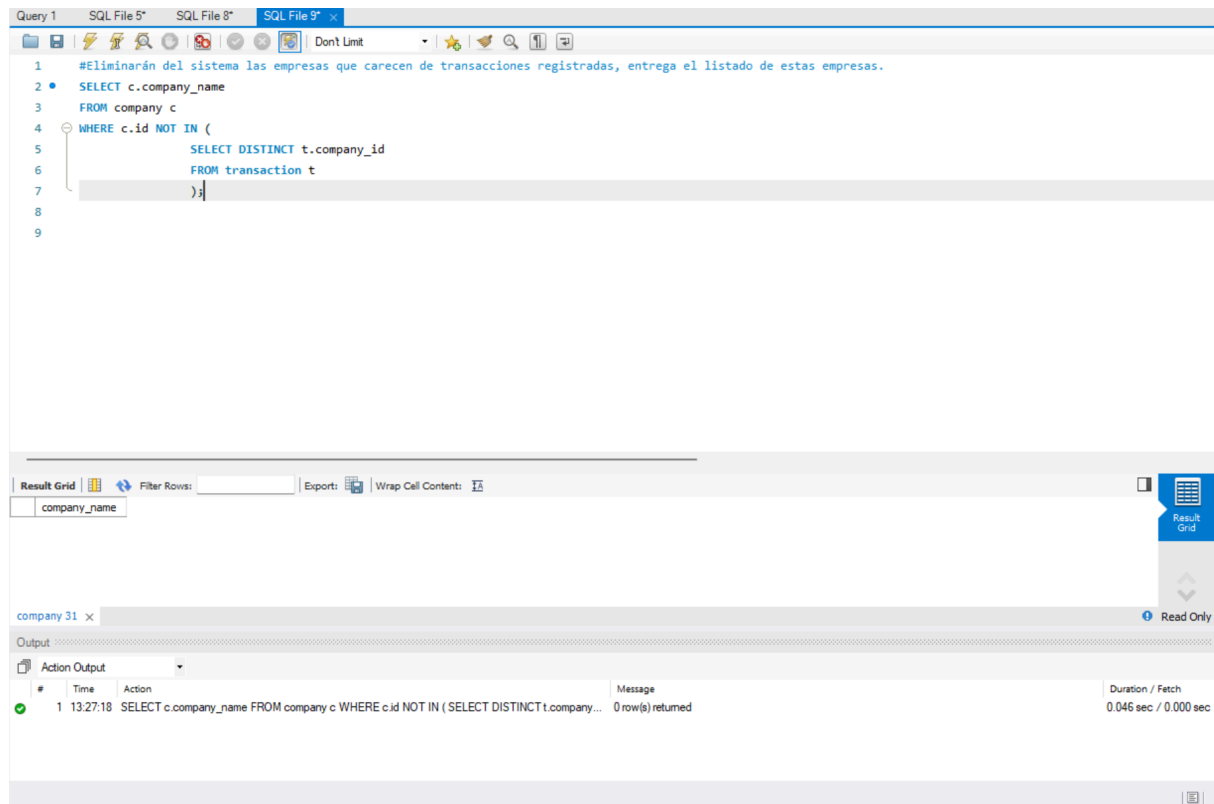
El primer paso consiste en calcular la media de todas las transacciones, que se utilizará como subconsulta dentro de la cláusula WHERE.

A continuación, para mostrar las compañías, utilizamos la cláusula IN para saber que id de compañías están dentro de ese listado de transacciones superiores a la media.

De esta manera visualizamos el listado de empresas que tienen alguna transacción por un amount superior a la media de todas las transacciones.

C. Eliminarán del sistema las empresas que carecen de transacciones registradas,

entrega el listado de estas empresas.



Para verificar si existen empresas sin transacciones, se realizaron dos aproximaciones. Primero, se empleó una subconsulta con NOT IN, que listaba todos los company_id presentes en la tabla transaction, y la consulta principal filtraba aquellas compañías cuyo id no se encontraba en ese listado.

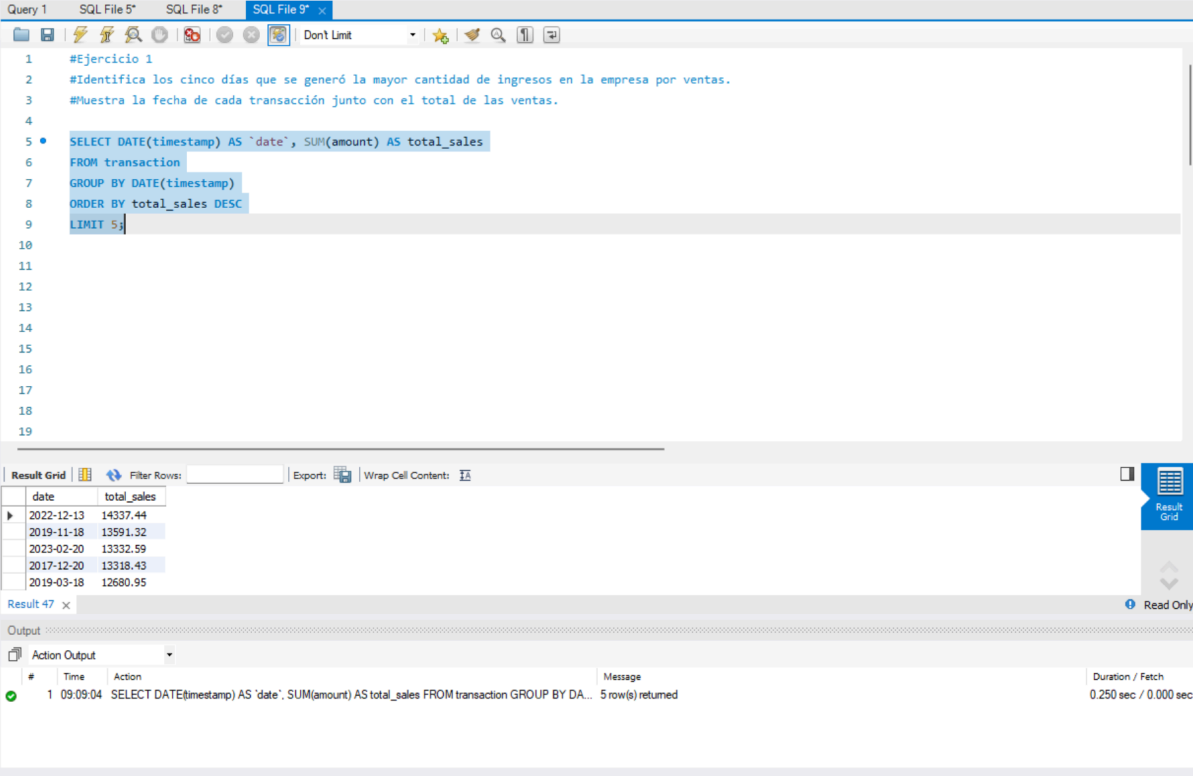
Al obtener un resultado vacío, se consideró la posibilidad de que existieran valores NULL en la subconsulta, ya que esto podría afectar la evaluación de NOT IN. Sin embargo, tras comprobarlo, no se encontraron registros con NULL, y el resultado siguió siendo vacío.

Por lo tanto, se puede concluir que, hasta el momento, todas las empresas registradas cuentan con transacciones asociadas.

> NIVEL 2

EJERCICIO 1

Identifica los cinco días 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.



The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
1 #Ejercicio 1
2 #Identifica los cinco días que se generó la mayor cantidad de ingresos en la empresa por ventas.
3 #Muestra la fecha de cada transacción junto con el total de las ventas.
4
5 • SELECT DATE(timestamp) AS 'date', SUM(amount) AS total_sales
6 FROM transaction
7 GROUP BY DATE(timestamp)
8 ORDER BY total_sales DESC
9 LIMIT 5;
```

The results pane displays a table with two columns: 'date' and 'total_sales'. The data is as follows:

date	total_sales
2022-12-13	14337.44
2019-11-18	13591.32
2023-02-20	13332.59
2017-12-20	13318.43
2019-03-18	12680.95

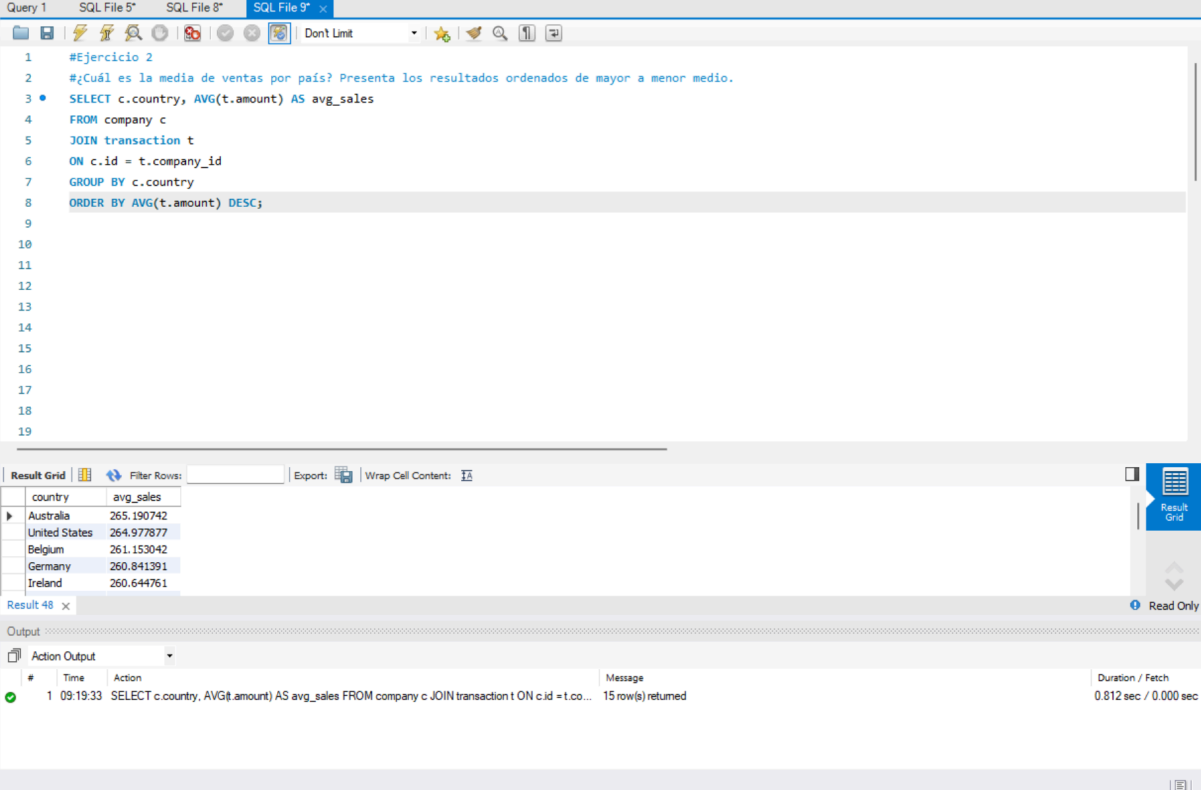
The bottom pane shows the execution output with the following message:

```
# Time Action Message Duration / Fetch
1 09:09:04 SELECT DATE(timestamp) AS 'date', SUM(amount) AS total_sales FROM transaction GROUP BY DA... 5 row(s) returned 0.250 sec / 0.000 sec
```

Para identificar los cinco días con mayores ingresos, utilizamos el campo timestamp, del cual extraemos únicamente la fecha mediante la función DATE. A partir del campo amount, calculamos la suma total de ingresos. Posteriormente, agrupamos los resultados por fecha y los ordenamos en forma descendente, de modo que los días con mayores ingresos aparezcan primero. Finalmente, seleccionamos los cinco primeros registros.

EJERCICIO 2

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



The screenshot shows a SQL IDE interface with a query editor and a results grid. The query is as follows:

```
1 #Ejercicio 2
2 #¿Cuál es la media de ventas por país? Presenta los resultados ordenados de mayor a menor medio.
3 • SELECT c.country, AVG(t.amount) AS avg_sales
4 FROM company c
5 JOIN transaction t
6 ON c.id = t.company_id
7 GROUP BY c.country
8 ORDER BY AVG(t.amount) DESC;
```

The results grid displays the following data:

country	avg_sales
Australia	265.190742
United States	264.977877
Belgium	261.153042
Germany	260.841391
Ireland	260.644761

The output panel shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	09:19:33	SELECT c.country, AVG(t.amount) AS avg_sales FROM company c JOIN transaction t ON c.id = t.co...	15 row(s) returned	0.812 sec / 0.000 sec

Aplicando un JOIN, podemos combinar la tabla de transacciones con la tabla de compañías para acceder al campo c.country. A partir del campo t.amount, calculamos el promedio de ingresos mediante la función AVG. Al agrupar los resultados por país, obtenemos el promedio de ingresos por país, y al ordenar de forma descendente podemos identificar rápidamente los países con mayores ingresos promedio.

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 ubicadas en el mismo país que esta compañía.

- Muestra el listado aplicando solo subconsultas.

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query is as follows:

```

1 #Ejercicio 3
2 #En tu empresa, se plantea un nuevo proyecto para lanzar algunas campañas publicitarias para hacer competencia a la compañía "Non Institute".
3 #Para ello, te piden la lista de todas las transacciones realizadas por empresas que están ubicadas en el mismo país que esta compañía.
4 #Muestra el listado aplicando solo subconsultas.
5 SELECT *
6 FROM (
7     SELECT t.id,
8           t.amount,
9           t.timestamp,
10          c.company_name,
11          c.country
12     FROM company c, transaction t
13     WHERE c.id = t.company_id
14 ) sub
15 WHERE sub.country IN (
16     SELECT c.country
17     FROM company c
18     WHERE c.company_name = 'Non Institute'
19 )

```

The results pane displays a table with 5 columns: id, amount, timestamp, company_name, and country. It shows 6 rows of data, all from 'Sed Nunc Ltd' in the 'United Kingdom'.

id	amount	timestamp	company_name	country
008629B4-C9A9-406C-A3D2-71FDA478C546	486.44	2015-07-30 12:12:42	Sed Nunc Ltd	United Kingdom
00B72BA4-54A3-4B8E-B13F-2D57535AA17A	414.06	2017-10-26 22:08:26	Sed Nunc Ltd	United Kingdom
01F075B1-D7AE-4D02-AAD9-5FFD72A43F3C	103.73	2018-01-27 13:44:36	Sed Nunc Ltd	United Kingdom
023FFC68-E618-4938-BF56-C8DF80540ADD	219.28	2016-12-19 11:53:45	Sed Nunc Ltd	United Kingdom
02683BEB-EF91-4564-957B-D6F1662AB7C5	326.87	2017-01-10 21:09:29	Sed Nunc Ltd	United Kingdom

The bottom pane shows the execution log with the following message:

```

1 09:29:38 SELECT * FROM ( SELECT t.id, t.amount, t.timestamp, c.company_name, ... 13776 row(s) returned

```

En primer lugar, es necesario determinar, mediante una subconsulta, en qué país se encuentra la compañía Non Institute. A continuación, generamos una subconsulta en la cláusula FROM que funciona como una tabla derivada, la cual incluye la información de cada transacción junto con los campos correspondientes de la compañía, como company_name y country. Esto nos permite referirnos al campo sub.country en la cláusula WHERE para filtrar únicamente aquellas transacciones realizadas por empresas ubicadas en el mismo país que Non Institute. De esta manera, podemos obtener finalmente un listado completo con todos los datos de las compañías y sus transacciones correspondientes, manteniendo la consulta estructurada y eficiente.

#pensar en la eficacia de las consultas

- Muestra el listado aplicando JOIN y subconsultas.

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```

1 #Muestra el listado aplicando JOIN y subconsultas.
2 • SELECT t.id, t.amount, t.timestamp, c.company_name, c.country
3 FROM transaction t
4 JOIN company c
5 ON c.id = t.company_id
6 WHERE c.country IN (
7     SELECT c2.country
8     FROM company c2
9     WHERE c2.company_name = 'Non Institute');

```

The results pane displays a table with 6 columns: id, amount, timestamp, company_name, and country. The table contains 6 rows of data, all from 'Sed Nunc Ltd' in the 'United Kingdom'.

id	amount	timestamp	company_name	country
00862984-C9A9-406C-A3D2-71FDA47BC546	486.44	2015-07-30 12:12:42	Sed Nunc Ltd	United Kingdom
00872BA4-54A3-4B8E-813F-2D57535AA17A	414.06	2017-10-26 22:08:26	Sed Nunc Ltd	United Kingdom
01P075B1-D7AE-4D02-AAD9-9FFD72A43F3C	103.73	2018-01-27 13:44:36	Sed Nunc Ltd	United Kingdom
023FFCE8-E618-4938-8F56-C8DF80540ADD	219.28	2016-12-19 11:53:45	Sed Nunc Ltd	United Kingdom
026838EB-EF91-4564-957B-D6F1662AB7C5	326.87	2017-01-10 21:09:29	Sed Nunc Ltd	United Kingdom

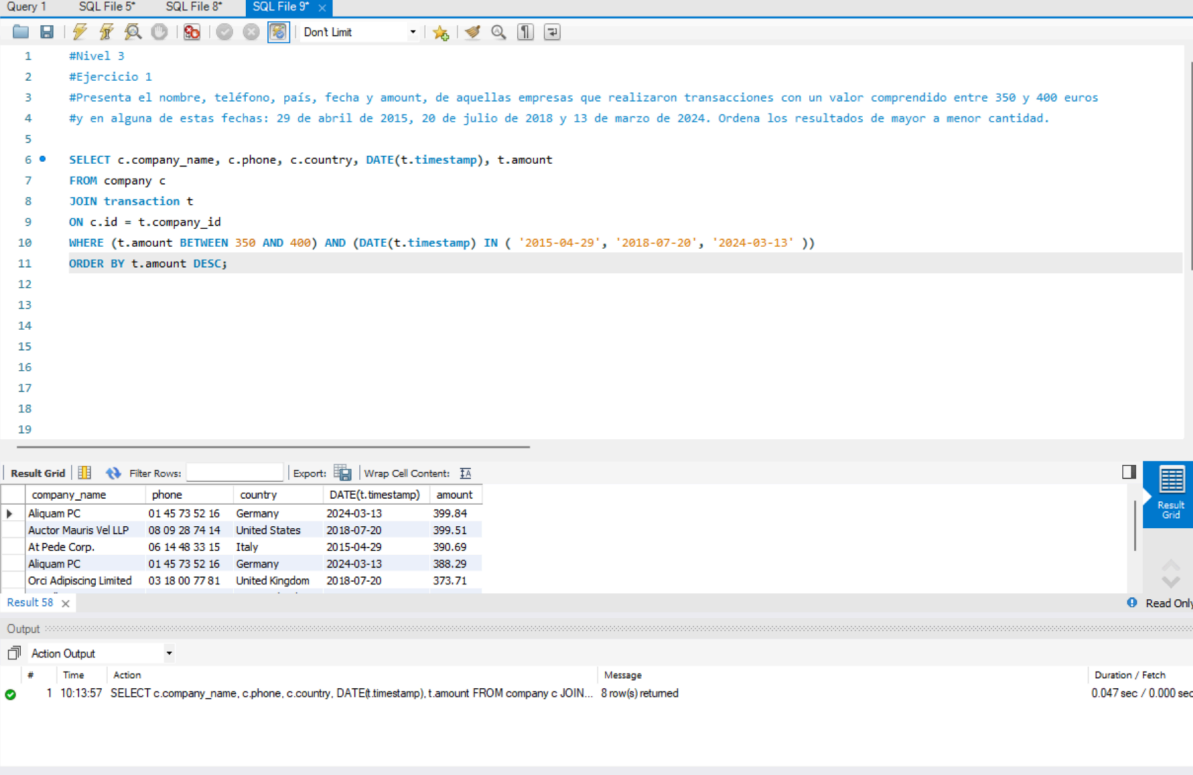
The output pane shows the execution message: "SELECT t.id, t.amount, t.timestamp, c.company_name, c.country FROM transaction t JOIN company c ... 13776 row(s) returned" with a duration of 0.016 sec / 0.062 sec.

Siguiendo la misma lógica pero utilizando JOIN junto con una subconsulta, combinamos las tablas company y transaction mediante la condición `c.id = t.company_id`. Gracias a este JOIN, ahora podemos acceder directamente a los campos de ambas tablas y aplicar la condición en la cláusula WHERE. Esta condición filtra únicamente aquellas transacciones realizadas por compañías que se encuentran en el mismo país que Non Institute, permitiéndonos obtener un listado completo de registros relevantes de manera clara y eficiente.

➤ 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.



The screenshot shows a SQL IDE interface with a query editor and a results grid. The query is as follows:

```
1 #Nivel 3
2 #Ejercicio 1
3 #Presenta el nombre, teléfono, país, fecha y amount, de aquellas empresas que realizaron transacciones con un valor comprendido entre 350 y 400 euros
4 #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.
5
6 • SELECT c.company_name, c.phone, c.country, DATE(t.timestamp), t.amount
7 FROM company c
8 JOIN transaction t
9 ON c.id = t.company_id
10 WHERE (t.amount BETWEEN 350 AND 400) AND (DATE(t.timestamp) IN ( '2015-04-29', '2018-07-20', '2024-03-13' ))
11 ORDER BY t.amount DESC;
```

The results grid displays the following data:

company_name	phone	country	DATE(t.timestamp)	amount
Aliquam PC	01 45 73 52 16	Germany	2024-03-13	399.84
Auctor Mauris Vel LLP	08 09 28 74 14	United States	2018-07-20	399.51
At Pedet Corp.	06 14 48 33 15	Italy	2015-04-29	390.69
Aliquam PC	01 45 73 52 16	Germany	2024-03-13	388.29
Orci Adipiscing Limited	03 18 00 77 81	United Kingdom	2018-07-20	373.71

The output pane shows the execution message: "1 10:13:57 SELECT c.company_name, c.phone, c.country, DATE(t.timestamp), t.amount FROM company c JOIN... 8 row(s) returned" with a duration of 0.047 sec / 0.000 sec.

Para obtener el resultado solicitado, primero realizamos un JOIN entre las tablas company y transaction, lo que nos permite acceder a todos los campos requeridos por el enunciado. En la cláusula WHERE, aplicamos dos condiciones: el monto de la transacción debe estar comprendido entre 350 y 400, y la fecha de la transacción debe coincidir con alguna de las fechas específicas indicadas. Finalmente, ordenamos los resultados de mayor a menor según el monto. De esta manera, la consulta devuelve de forma completa y estructurada todas las transacciones que cumplen con los criterios establecidos.

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 en las que especifiques si tienen más de 400 transacciones o menos.

The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays a query named 'Ejercicio 2' with the following SQL code:

```

1 #Ejercicio 2
2 #Necesitamos optimizar la asignación de los recursos y dependerá de la capacidad operativa que se requiera,
3 #por lo que te piden la información sobre la cantidad de transacciones que realizan las empresas, pero el departamento de recursos humanos es exigente
4 #y quiere un listado de las empresas en las que especifiques si tienen más de 400 transacciones o menos.
5 SELECT
6     company_name,
7     (SELECT COUNT(*)
8      FROM transaction t
9      WHERE t.company_id = c.id) AS total_transactions,
10    CASE
11      WHEN (SELECT COUNT(*)
12            FROM transaction t
13            WHERE t.company_id = c.id) > 400 THEN 'More than 400'
14      ELSE '400 or less'
15    END AS transaction_category
16 FROM company c
17 ORDER BY company_name;
18
19

```

The bottom pane shows the 'Result Grid' with the following data:

company_name	total_transactions	transaction_category
A Institute	439	More than 400
Ac Fermentum Incorporated	2401	More than 400
Ac Industries	1530	More than 400
Ac Libero Inc.	1513	More than 400
Aliquam Erat Volutpat LLP	1526	More than 400

The 'Output' pane at the bottom shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	12:44:36	SELECT	company_name, (SELECT COUNT(*) FROM transaction t WHERE t.company_id = c.id) AS total_transactions, CASE WHEN (SELECT COUNT(*) FROM transaction t WHERE t.company_id = c.id) > 400 THEN 'More than 400' ELSE '400 or less' END AS transaction_category FROM company c ORDER BY company_name;	100 row(s) returned 0.172 sec / 0.000 sec

En este ejercicio, utilizamos subconsultas correlacionadas para vincular cada compañía con sus transacciones. Para cada registro de la tabla company, calculamos la cantidad de transacciones asociadas mediante una subconsulta que cuenta todos los registros de la tabla transaction cuyo company_id coincide con el id de la compañía actual. Este conteo se muestra en la columna total_transactions y representa el número total de transacciones realizadas por cada empresa.

A continuación, aplicamos un CASE para clasificar las compañías según la cantidad de transacciones. Si el número de transacciones supera las 400, se asigna la categoría "More than 400"; en caso contrario, se asigna "400 or less". De esta manera, obtenemos un listado que indica claramente si cada compañía cumple o no con el umbral de 400 transacciones establecido en el enunciado.

Finalmente, ordenamos los resultados alfabéticamente por el nombre de la compañía, lo que facilita la lectura y permite visualizar rápidamente la información de manera organizada.