

☆☆☆ 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.

La base de datos llamada **transactions** está diseñada para registrar información sobre transacciones financieras. El esquema incluye dos tablas principales: **company** y **transaction**.

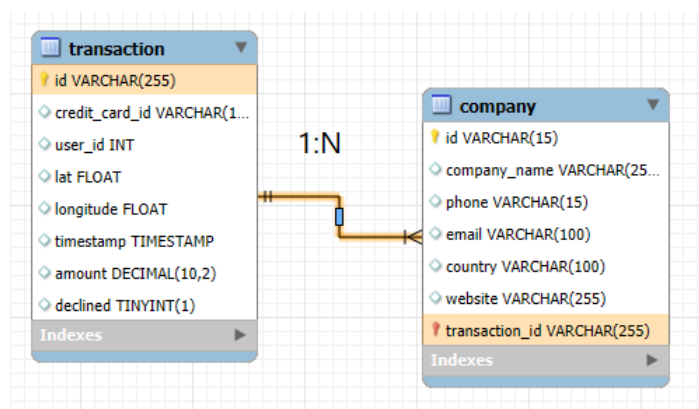
La tabla **company** almacena los datos de las empresas donde se realizan las transacciones. Contiene campos como el ID (VARCHAR), el nombre de la empresa (VARCHAR), teléfono (VARCHAR), correo electrónico (VARCHAR), país (VARCHAR) y sitio web (VARCHAR).

La tabla **transaction**, por su parte, registra cada transacción individual. Incluye un ID (VARCHAR) como clave primaria y referencias a otras entidades mediante claves foráneas: el ID del usuario (INT), el ID de la empresa (VARCHAR) y el ID de la tarjeta de crédito (VARCHAR). Además, contiene información sobre la ubicación (FLOAT para latitud y longitud), la fecha y hora de la transacción (TIMESTAMP), el monto (DECIMAL), y un campo que indica si la transacción fue rechazada (BOOLEAN).

Cabe destacar que aunque se hace referencia a las tablas user y credit_card, estas no están incluidas en el esquema proporcionado.

En resumen, se trata de un diseño relacional bien estructurado, que permite registrar y consultar de manera organizada quién realizó una transacción, con qué tarjeta, en qué empresa y por qué monto, incluyendo detalles geográficos y temporales. Los tipos de datos utilizados permiten representar adecuadamente tanto datos alfanuméricos como numéricos, fechas, etc.

En el diagrama de abajo se puede observar el tipo de relación que hay entre las dos tablas (1:N) ya que una transacción corresponde a una sola empresa en cambio una empresa puede tener varias transacciones.



Ejercicio 2

Utilizando JOIN realizarás las siguientes consultas:

- Listado de los países que están generando ventas.

```
1 • USE transactions;
2 • SELECT DISTINCT c.country AS listado_paises
3   FROM company AS c
4  JOIN transaction AS t ON c.id = t.company_id
5  WHERE t.amount > 0 AND t.declined = 0;
6 |
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

listado_paises
Netherlands
Sweden
Ireland
United States
Belgium
Canada
Germany
Norway
France
Italy
United Kingdom
New Zealand
China
Spain
Australia

Result 2 x

Output

Action Output

#	Time	Action	Message
✓ 1	21:19:05	USE transactions	0 row(s) affected
✓ 2	21:19:05	SELECT DISTINCT c.country AS listado_paises FROM company AS c JOIN transaction AS t ON c.id = t.company_id WHERE t.amount > 0 A...	15 row(s) returned

Explicación código: el objetivo final de este código es mostrar un listado de países que han generado ventas. Para eso realice un JOIN entre las tablas company y transaction mediante el campo en común company_id, que conecta cada transacción con su empresa. Luego con las condiciones en WHERE t.amount > 0 y t.declined = 0 se filtran los valores positivos y que no hayan sido declinados o devueltos, es decir los que representan ventas reales. Tuve dudas si utilizar en el filtro IS NULL, pero desistí por que no quería incluir valores negativos ni declinados. Finalmente utilice DISTINCT para evitar países repetidos y asigné el alias listado_paises para representar los resultados de forma más clara.

- Desde cuántos países se generan las ventas.

```

15
16 • SELECT COUNT(DISTINCT c.country) as cantidad_paises_ventas
17 FROM company as c
18 INNER JOIN transaction as t on c.id = t.company_id
19 WHERE t.amount > 0 AND t.declined = 0;

```

Result Grid

cantidad_paises_ventas
15

Result 3 x

Output

Action Output

#	Time	Action	Message
✓ 1	21:34:06	SELECT COUNT(DISTINCT c.country) as cantidad_paises_ventas FROM company as c INNER JOIN transaction as t on c.id = t.company_id ...	1 row(s) returned

Explicación código: el output de este código muestra la cantidad de países de los cuales se generan ventas. Como de base ya tenía un código que me daba como resultado un listado de países en los cuales se vendía, aproveche que ya lo tenía escrito y simplemente añadir una función de conteo para los países que conforman esta lista utilizando el COUNT. Como ya tenía un DISTINCT debía cuidar la posición del COUNT, ya que el DISTINCT tenía que permanecer justo al lado de la columna para que me diera resultados óptimos.

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

```

14 • SELECT c.company_name AS compañía_mas_ventas, AVG(t.amount) AS media
15 FROM company AS c
16 JOIN transaction AS t ON c.id = t.company_id
17 GROUP BY c.id, c.company_name
18 ORDER BY media DESC
19 LIMIT 1;

```

Result Grid

compañía_mas_ventas	media
Ac Fermentum Incorporated	284.867160

Result 5 x

Output

Action Output

#	Time	Action	Message
✓ 1	21:42:50	SELECT c.company_name AS compañía_mas_ventas, AVG(t.amount) AS media FROM company AS c JOIN transaction AS t ON c.id = t.com...	1 row(s) returned

Explicación código: para obtener la compañía con una media de ventas mayor, hice un JOIN entre las tablas company y transaction mediante el campo en común company_id. Luego procedí a agrupar las empresas con su correspondiente media de ventas utilizando la función GROUP BY y ordené los datos de acuerdo a su media de ventas en orden descendente. Límite al SQL para que me diera un solo valor con LIMIT 1, y de esa manera obtuve el output esperado.

Ejercicio 3

Utilizando sólo subconsultas (sin utilizar JOIN):

- Muestra todas las transacciones realizadas por empresas de Alemania.

```
12 • SELECT *
13 FROM transaction AS t
14 WHERE t.company_id IN (SELECT c.id
15 FROM company AS c
16 WHERE c.country = 'Germany'
17 );
18
```

Result Grid

	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
	0013C1B6-3884-4D6C-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
	00AAB9C7-30F6-4D9B-8A1D-13BF73F09A9	CcS-6707	b-2222	7716	55.7657	-3.76745	2021-04-25 03:06:59	167.15	0

transaction 7 x

Output

Action Output

#	Time	Action	Message
✓ 1	21:53:15	SELECT * FROM transaction AS t WHERE t.company_id IN (SELECT c.id FROM company AS c WHERE c.country = 'Germany')	13291 row(s) returned

Explicación código: elegí SELECT * para mostrar todos los datos de las transacciones que era lo que se me pedía y como no se podía utilizar JOIN realice una subconsulta en el WHERE para filtrar los resultados del país Alemania que se pedía en el ejercicio.

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

```
5
6 • SELECT c.company_name AS empresas_encima_media
7 FROM company AS c
8 WHERE c.id IN (SELECT t.company_id
9 FROM transaction AS t
10 WHERE t.amount > (
11 SELECT AVG(t.amount)
12 FROM transaction AS t)
13 );
14
```

Result Grid

	empresas_encima_media
▶	Ac Fermentum Incorporated
	Magna A Neque Industries
	Fusce Corp.
	Convallis In Incorporated
	Ante Iaculis Nec Foundation
	Donec Ltd
	Sed Nunc Ltd
	Amet Nulla Donec Corporation
	Nascetur Ridiculus Mus Inc.
	Vestibulum Lorem PC
	Gravida Sagittis LLP
	Mus Aenean Eget Foundation
	Dis Parturient Institute

company 11 x

Output

Action Output

#	Time	Action	Message
✓ 1	22:07:44	SELECT c.company_name AS empresas_encima_media FROM company AS c WHERE c.id IN (SELECT t.company_id FROM transaction A...	100 row(s) returned

Explicación código: además de realizar una subconsulta externa en el WHERE como el ejercicio anterior utilice una subconsulta interna para filtrar las empresas que realizaron transacciones mayores a la media.

- Eliminarán del sistema las empresas que carecen de transacciones registradas, entrega el listado de estas empresas.

```
16 • SELECT c.company_name AS empresas_sin_transacciones
17 FROM company AS c
18 WHERE NOT EXISTS (SELECT 1
19                    FROM transaction AS t
20                    WHERE t.company_id = c.id
21 );
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
empresas_sin_transacciones			

company 7 x			
Output			
Action Output			
#	Time	Action	Message
1	15:46:32	SELECT c.company_name AS empresas_sin_transacciones FROM company AS c WHERE NOT EXISTS (SELECT 1 FROM trans...	0 row(s) returned

Explicación código: el output muestra 0 filas resultantes devueltas lo que significa que no hay ninguna empresa que no tenga al menos una transacción efectiva por lo que no se eliminará ninguna del sistema. Utilice el NOT EXIST con su respectivo 1 en el SELECT de la subconsulta, y la condición en el WHERE t.company_id = c.id para comprobar compañías que existieran en la tabla company más no en la tabla transaction, pero no arrojó ningún resultado.

★★★ 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.

```

17 • SELECT DATE(t.timestamp) AS fecha, SUM(t.amount) AS total_ventas
18 FROM transaction AS t
19 GROUP BY fecha
20 ORDER BY total_ventas DESC
21 Limit 5;

```

	fecha	total_ventas
▶	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

Result 2 x

Output

Action Output

#	Time	Action	Message
✓ 1	13:29:15	SELECT DATE(t.timestamp) AS fecha, SUM(t.amount) AS total_ventas FROM transaction AS t GROUP BY fecha ORDER BY total_ventas DE...	5 row(s) returned

Explicación código: las fechas están dadas junto con las horas de las transacciones, es por esto que utilice la función DATE para que solo me devolviera los datos en formato fecha .Además sume los montos de las ventas con la función SUM para que me diera el total de las cantidades, y por último agrupe (con GROUP BY) esos totales con sus fechas correspondientes.

Ejercicio 2

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

```

17 • SELECT c.country AS pais, AVG(t.amount) AS media_ventas
18 FROM company AS c
19 JOIN transaction AS t ON c.id = t.company_id
20 GROUP BY pais
21 ORDER BY media_ventas DESC;

```

	pais	media_ventas
▶	Australia	265.190742
	United States	264.977877
	Belgium	261.153042
	Germany	260.841391
	Ireland	260.644761
	Spain	260.468125
	France	259.979185
	New Zealand	259.586176
	Norway	259.275227

Result 4 x

Output

Action Output

#	Time	Action	Message
✓ 1	14:23:11	SELECT c.country AS pais, AVG(t.amount) AS media_ventas FROM company AS c JOIN transaction AS t ON c.id = t.company_id GROUP BY...	15 row(s) returned

Explicación código: unifique las dos tablas de la base de datos transactions, mediante JOIN, y el campo en común id en la tabla company y company_id en la tabla transaction. Agrupe por países las medias de ventas y lo ordene de mayor a menor mediante ORDER BY DESC.

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 JOIN y subconsultas.

```
14
15 • SELECT*
16 FROM transaction AS t
17 JOIN company AS c ON c.id = t.company_id
18 WHERE c.country = (SELECT c.country
19                     FROM company AS c
20                     WHERE c.company_name = 'Non Institute')
21 );
22
```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	id	company_name	phone	email	country	website
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	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
00872BA4-54A3-4B8E-B13F-2D57535AA17A	CcS-8475	b-2246	3894	55.6212	-3.7546	2017-10-26 22:08:26	414.06	0	b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
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	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
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	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
026838EB-EF91-4564-957B-D6F1662AB7C5	CcS-9471	b-2246	4890	42.1332	12.396	2017-01-10 21:09:29	326.87	0	b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
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	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
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	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
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	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one

Result: 13776 rows

Output

Action Output

1 14:44:32 SELECT* FROM transaction AS t JOIN company AS c ON c.id = t.company_id WHERE c.country = (SELECT c.country FROM company AS c WHERE c.company_name = 'Non Institute') FROM co... 13776 row(s) returned

Explicación código: la lista de todas las transacciones de las empresas que tienen en común el país con “Non Institute”, la obtuve haciendo un JOIN entre las dos tablas y una subconsulta en el WHERE para filtrar los países que estuvieran en el país donde está la empresa en cuestión. Probe la subconsulta como un código separado para ver probar su efectividad y funciona.

- Muestra el listado aplicando solo subconsultas.

```
17 • SELECT t.*
18 FROM transaction AS t
19 WHERE t.company_id IN (SELECT c.id
20                       FROM company AS c
21                       WHERE c.country = (SELECT country
22                                         FROM company AS c
23                                         WHERE c.company_name = 'Non Institute'))
24 );
```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
008629B4-C9A9-406C-A3D2-71FDA47BC546	CcS-7063	b-2246	2482	45.7666	4.83048	2015-07-30 12:12:42	486.44	0
00872BA4-54A3-4B8E-B13F-2D57535AA17A	CcS-8475	b-2246	3894	55.6212	-3.7546	2017-10-26 22:08:26	414.06	0
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
026838EB-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

transaction 5 x

Output

Action Output

1 15:01:41 SELECT t.* FROM transaction AS t WHERE t.company_id IN (SELECT c.id FROM company AS c WHERE c.country = (SELECT c.country FROM company AS c WHERE c.company_name = 'Non Institute')) FROM co... 13776 row(s) returned

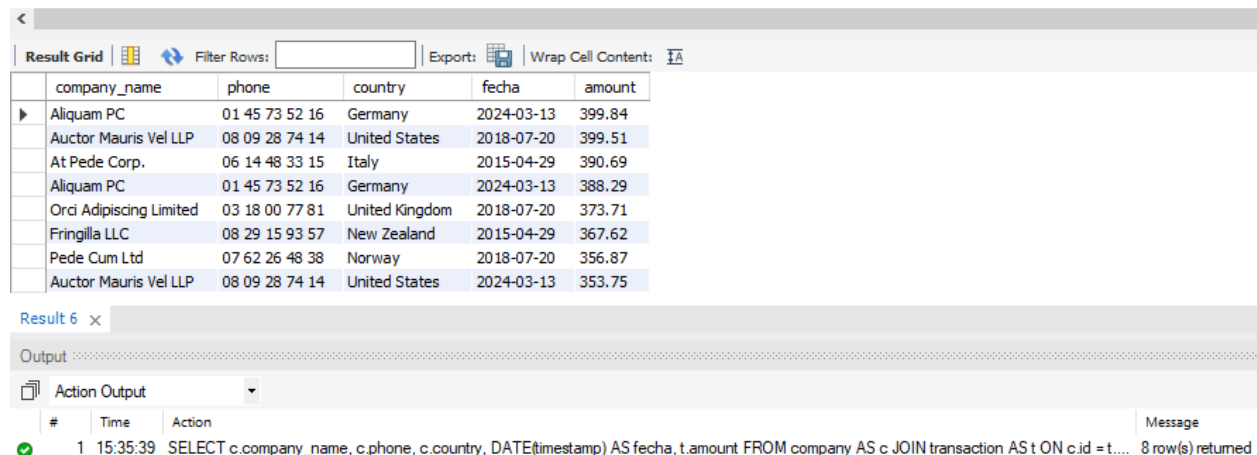
Explicación código: el resultado esperado era el mismo que el ejercicio anterior pero sin usar JOIN, por eso utilice una subconsulta externa y una interna. Obtuve los mismos resultados que la consigna anterior.

★★★ 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.

```
16 • SELECT c.company_name, c.phone, c.country, DATE(timestamp) AS fecha, t.amount
17 FROM company AS c
18 JOIN transaction AS t ON c.id = t.company_id
19 WHERE amount BETWEEN 350 AND 400
20 AND DATE(timestamp) IN ('2015-04-29', '2018-07-20', '2024-03-13')
21 ORDER BY t.amount DESC;
```



The screenshot shows a database interface with a 'Result Grid' tab selected. It displays the results of the SQL query above. The grid has 6 columns: company_name, phone, country, fecha, and amount. There are 8 rows of data. Below the grid, there is an 'Output' section with a dropdown menu set to 'Action Output'. At the bottom, a status bar shows a green checkmark, the time '15:35:39', the query text, and a message: '8 row(s) returned'.

	company_name	phone	country	fecha	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 Pedo 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
	Ordi Adipiscing Limited	03 18 00 77 81	United Kingdom	2018-07-20	373.71
	Fringilla LLC	08 29 15 93 57	New Zealand	2015-04-29	367.62
	Pede Cum Ltd	07 62 26 48 38	Norway	2018-07-20	356.87
	Auctor Mauris Vel LLP	08 09 28 74 14	United States	2024-03-13	353.75

Result 6 x

Output

Action Output

Time Action Message

✓ 1 15:35:39 SELECT c.company_name, c.phone, c.country, DATE(timestamp) AS fecha, t.amount FROM company AS c JOIN transaction AS t ON c.id = t.... 8 row(s) returned

Explicación código: el objetivo de este código era listar las empresas que cumplieran dos condiciones: que su monto en transacciones estuviese comprendido entre 350 y 400 y además, que esas operaciones ocurrieran en alguna de las fechas provistas y por última ordenar los resultados.

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.


```
18 • SELECT c.company_name, c.id, IF (COUNT(t.amount) > 400, 'mas de 400', 'menos de 400') AS conteo_transacciones
19 FROM company AS c
20 JOIN transaction AS t ON c.id = t.company_id
21 GROUP BY c.company_name, c.id;
??
```

company_name	id	conteo_transacciones
Ac Fermentum Incorporated	b-2222	mas de 400
Magna A Neque Industries	b-2226	mas de 400
Fusce Corp.	b-2230	mas de 400
Convallis In Incorporated	b-2234	mas de 400
Ante Taculis Nec Foundation	b-2238	mas de 400
Donec Ltd	b-2242	mas de 400
Sed Nunc Ltd	b-2246	mas de 400
Amet Nulla Donec Corporation	b-2250	mas de 400
Namper Dignissim Mus Inc	b-2254	mas de 400

Result 8 x

Output

Action Output

#	Time	Action	Message
1	16:45:12	SELECT c.company_name, c.id, IF (COUNT(t.amount) > 400, 'mas de 400', 'menos de 400') AS conteo_transacciones FROM company AS c J...	100 row(s) returned

Explicación código: como el resultado implicaba añadir una columna que solo se visualizara en el output y que no se agregue a la base de datos, utilice la función IF en el SELECT para este fin, y que con la condición de COUNT(t.amount) > 400 me clasificara en cantidad de más de 400 transacciones o menos. Elegí utilizar IF en lugar de CASE WHEN ya que solo tenía una condicion y es mas simple para el lenguaje.