

☆☆☆ 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**.

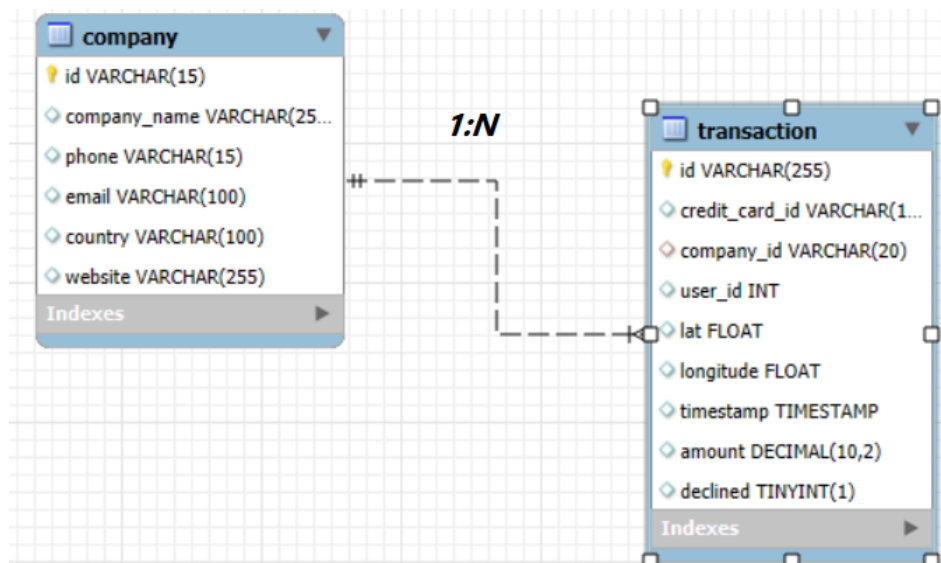
La tabla **company** almacena los datos de las empresas donde se realizan las transacciones. Contiene campos como el ID (VARCHAR) la cual es su primary key, el nombre de la empresa (VARCHAR), teléfono (VARCHAR), correo electrónico (VARCHAR), país (VARCHAR) y sitio web (VARCHAR).

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 se observa una relación de uno a muchos (1:N) desde la tabla company hacia la tabla transaction, ya que una compañía puede estar asociada a múltiples transacciones, mientras que cada transacción pertenece únicamente a una compañía. Además, la tabla transaction tiene una clave foránea (foreign key) que hace referencia a la clave primaria de la tabla Company (company_id).




Ejercicio 2

Utilizando JOIN realizarás las siguientes consultas:

- Listado de los países que están generando ventas.

```
21 • SELECT DISTINCT c.country AS listado_paises
22 FROM company AS c
23 JOIN transaction AS t ON c.id = t.company_id
24 WHERE t.declined = 0;
```



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

listado_paises
Netherlands
Sweden
Ireland
United States
Belgium
Canada
Germany
Norway
France

Result 2 x

Output

Action Output

#	Time	Action	Message
1	16:46:38	SELECT DISTINCT c.country AS listado_paises FROM company AS c JOIN transaction AS t ON c.id = t.com...	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 la condición en WHERE t.declined = 0 se filtran los valores 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 sin registros. 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.

```

13 • SELECT COUNT(DISTINCT c.country) AS cantidad_paises_ventas
14 FROM company AS c
15 INNER JOIN transaction AS t ON c.id = t.company_id
16 WHERE t.declined = 0;
17

```

Result Grid

cantidad_paises_ventas
15

Result 4 x

Output

Action Output

#	Time	Action	Message
1	16:55:39	SELECT COUNT(DISTINCT c.country) AS cantidad_paises_ventas FROM company AS c INNER JOIN trans...	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.

```

9 • SELECT c.company_name as compañías_mayor_media_ventas, ROUND(AVG(t.amount),2) AS media
10 FROM company AS c
11 JOIN transaction AS t ON c.id = t.company_id
12 WHERE t.declined = 0
13 GROUP BY c.id,c.company_name
14 ORDER BY media DESC
15 LIMIT 1;

```

Result Grid

compañías_mayor_media_ventas	media
Ac Fermentum Incorporated	284.91

Result 5 x

Output

Action Output

#	Time	Action	Message
1	16:58:53	SELECT c.company_name as compañías_mayor_media_ventas, ROUND(AVG(t.amount),2) AS media FROM ...	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 y lo redondeé a dos decimales con el ROUND. 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.

```

11 • SELECT *
12 FROM transaction AS t
13 WHERE t.company_id IN (SELECT c.id
14 FROM company AS c
15 WHERE t.declined = 0 AND c.country = 'Germany'
16 );

```

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	CdU-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-138E73DC90A9	CcS-6797	b-2222	2216	55.7652	-3.76245	2021-04-25 03:06:59	167.15	0
00BE09D4-6920-47D8-ABE8-325E2269829D	CcS-4983	b-2222	402	38.708	-9.12993	2019-02-27 15:25:16	141.66	0
00DA0383-E048-4577-8ED1-3C56C258FF2F	CcS-9223	b-2222	4642	51.1742	10.2027	2019-03-21 11:47:34	325.62	0
0000110F-FD01-48B0-93A0-174D183A590F	CcS-7681	b-2222	3100	45.7565	4.83109	2024-01-18 18:20:49	247.53	0

transaction 6 x

Output

Action Output

#	Time	Action	Message
1	17:05:58	SELECT * FROM transaction AS t WHERE t.company_id IN (SELECT c.id FROM company AS c WHERE t.declined = 0 AND c.country = 'Germany')	FROM company AS... 13269 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.

```

12 • SELECT c.company_name
13 FROM company AS c
14 WHERE EXISTS (SELECT t.company_id
15               FROM transaction AS t
16               WHERE t.declined = 0 AND t.amount > (SELECT AVG(t.amount)
17                                                       FROM transaction AS t)
18             );

```

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

company_name
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 Sanitit II P

company 8 x

Output

Action Output

#	Time	Action	Message
1	17:12:55	SELECT c.company_name FROM company AS c WHERE EXISTS (SELECT t.company_id FROM ...	100 row(s) returned

Explicación código: además de realizar una subconsulta externa en el WHERE como el ejercicio anterior y 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.

```

12 • SELECT c.company_name AS Empresas_sin_transacciones
13 FROM company AS c
14 WHERE NOT EXISTS (SELECT *
15                  FROM transaction AS t
16                  WHERE t.declined = 0 AND t.company_id = c.id
17                );

```

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

Empresas_sin_transacciones

company 9 x

Output

Action Output

#	Time	Action	Message
1	17:17:52	SELECT c.company_name AS Empresas_sin_transacciones FROM company AS c WHERE NOT EXISTS (S...	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 considerar los casos en los que la transacción pertenece a la compañía correspondiente, y como todas las compañías tenían al menos una transacción 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.

```
11 • SELECT DATE(t.timestamp) AS dia, ROUND(SUM(t.amount),2) AS total_ventas
12 FROM transaction AS t
13 WHERE t.declined = 0
14 GROUP BY DATE(timestamp)
15 ORDER BY total_ventas DESC
16 Limit 5;
```

dia	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 10 x

Output

Action Output

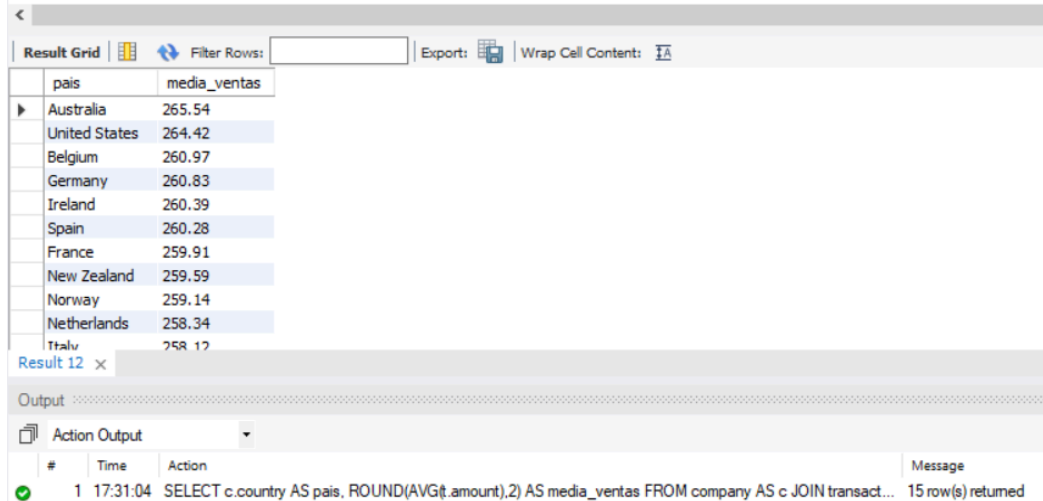
#	Time	Action	Message
1	17:24:54	SELECT DATE(t.timestamp) AS dia, ROUND(SUM(t.amount),2) AS total_ventas FROM transaction AS t WHE...	5 row(s) returned

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.

```
19 • SELECT c.country AS pais, ROUND(AVG(t.amount),2) AS media_ventas
20 FROM company AS c
21 JOIN transaction AS t on c.id = t.company_id
22 WHERE t.declined = 0
23 GROUP BY c.country
24 ORDER BY media_ventas DESC;
```



The screenshot shows a database interface with a query editor at the top and a result grid below. The query calculates the average sales amount per country, rounded to two decimal places, and orders the results from highest to lowest average. The result grid displays 15 rows of data. Below the grid, an 'Output' section shows the execution log with a successful message indicating 15 rows were returned.

pais	media_ventas
Australia	265.54
United States	264.42
Belgium	260.97
Germany	260.83
Ireland	260.39
Spain	260.28
France	259.91
New Zealand	259.59
Norway	259.14
Netherlands	258.34
Italy	258.12

Result 12 x

Output

Action Output

#	Time	Action	Message
1	17:31:04	SELECT c.country AS pais, ROUND(AVG(t.amount),2) AS media_ventas FROM company AS c JOIN transact...	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.

```

19 • SELECT t.*
20 FROM company AS c
21 JOIN transaction AS t ON c.id = t.company_id
22 WHERE c.company_name <> 'Non Institute' AND t.declined = 0 AND country = (SELECT c.country
23 FROM company AS c
24 WHERE c.company_name = 'Non Institute')
25 );

```

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
00B72BA4-54A3-4B8E-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
03068E3B-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
03C436D3-88FF-40BF-8FD4-4C70D44FFD7B	CcS-8036	b-2246	3455	52.5178	13.4131	2017-07-25 15:38:21	440.27	0

Result 13 x

Output

Action Output

#	Time	Action	Message
1	17:38:29	SELECT t.* FROM company AS c JOIN transaction AS t ON c.id = t.company_id WHERE c.company_name ...	12213 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; Antes de filtrar los países que estuvieran en el país donde está la empresa en cuestión, con una subconsulta, condicione a que se excluyera la misma empresa Non Institute y que las transacciones no hayan sido declinadas. Probé la subconsulta como un código separado para ver probar su efectividad y funciona.

- Muestra el listado aplicando solo subconsultas.

```

18 • SELECT t.*
19 FROM transaction AS t
20 WHERE t.declined = 0
21 AND t.company_id IN (SELECT c.id
22 FROM company AS c
23 WHERE c.company_name <> 'Non Institute' AND c.country = (
24 SELECT c.country
25 FROM company AS c
26 WHERE c.company_name = 'Non Institute')
27 );

```

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
00B72BA4-54A3-4B8E-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
03068E3B-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
03C436D3-88FF-40BF-8FD4-4C70D44FFD7B	CcS-8036	b-2246	3455	52.5178	13.4131	2017-07-25 15:38:21	440.27	0

Result 14 x

Output

Action Output

#	Time	Action	Message
1	17:47:29	SELECT t.* FROM transaction AS t WHERE t.declined = 0 AND t.company_id IN (SELECT c.id FROM com...	12213 row(s) returned

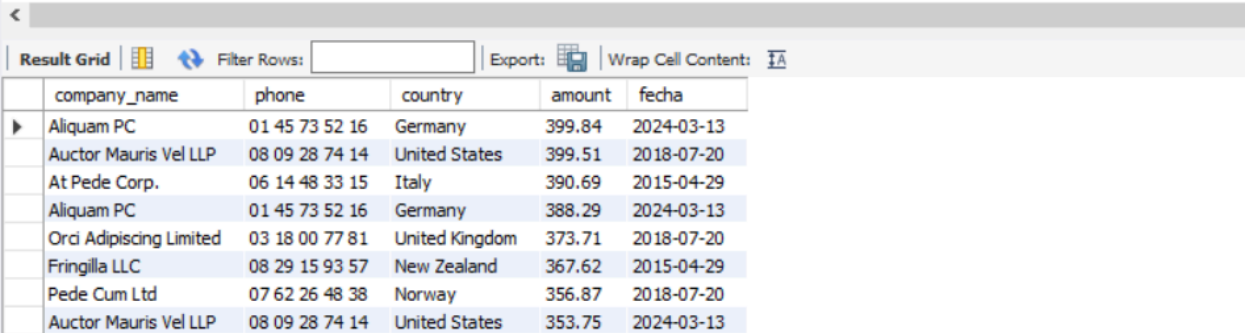
Explicación código: el resultado esperado era el mismo que el ejercicio anterior pero sin usar JOIN, por eso utilice una subconsulta externa y una interna. Obtuve los mismos resultados que la consigna anterior.

★★★ 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.

```
22 • SELECT c.company_name, c.phone, c.country, t.amount, DATE(t.timestamp) AS fecha
23 FROM company AS c
24 JOIN transaction AS t ON t.company_id = c.id
25 WHERE t.declined = 0 AND (amount BETWEEN 350 AND 400)
26 AND DATE(t.timestamp) IN ('2015-04-29', '2018-07-20', '2024-03-13')
27 ORDER BY t.amount DESC;
```



	company_name	phone	country	amount	fecha
▶	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
	At Pede Corp.	06 14 48 33 15	Italy	390.69	2015-04-29
	Aliquam PC	01 45 73 52 16	Germany	388.29	2024-03-13
	Ordi Adipiscing Limited	03 18 00 77 81	United Kingdom	373.71	2018-07-20
	Fringilla LLC	08 29 15 93 57	New Zealand	367.62	2015-04-29
	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 16 x

Output

Action Output

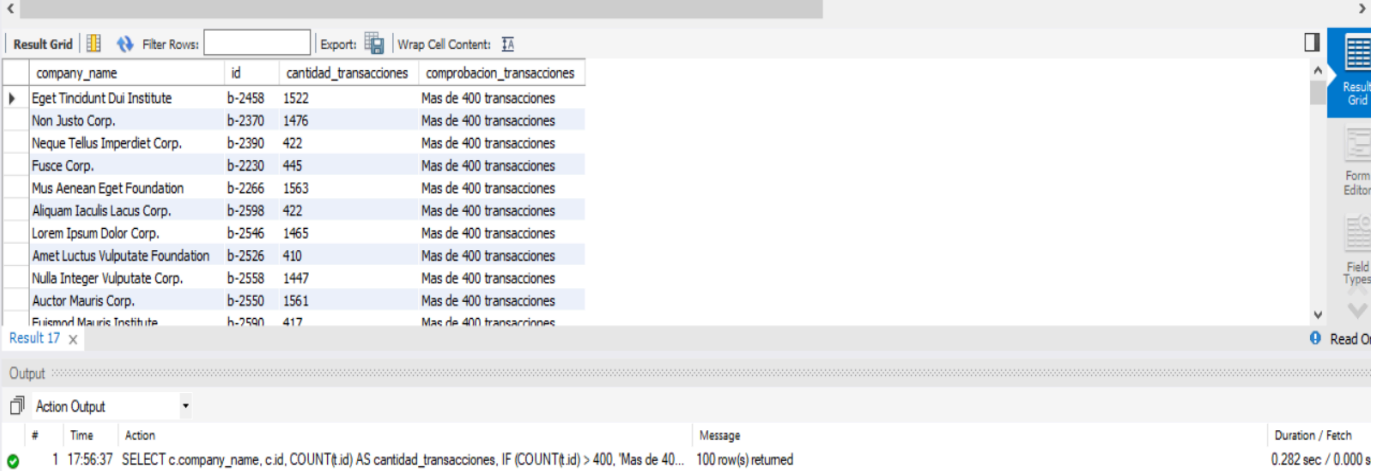
#	Time	Action	Message
✓ 1	17:54:24	SELECT c.company_name, c.phone, c.country, t.amount, DATE(t.timestamp) AS fecha FROM company AS c...	8 row(s) returned

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 con BETWEEN y además, que esas operaciones ocurrieran en alguna de las fechas previstas con IN 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, COUNT(t.id) AS cantidad_transacciones, IF (COUNT(t.id) > 400, 'Mas de 400 transacciones', 'Menos de 400 transacciones') AS comprobacion_transac
19 FROM company AS c
20 JOIN transaction AS t ON t.company_id = c.id
21 WHERE t.declined = 0
22 GROUP BY c.company_name, c.id;
```



company_name	id	cantidad_transacciones	comprobacion_transacciones
Eget Tincidunt Dui Institute	b-2458	1522	Mas de 400 transacciones
Non Justo Corp.	b-2370	1476	Mas de 400 transacciones
Neque Tellus Imperdiet Corp.	b-2390	422	Mas de 400 transacciones
Fusce Corp.	b-2230	445	Mas de 400 transacciones
Mus Aenean Eget Foundation	b-2266	1563	Mas de 400 transacciones
Aliquam Iaculis Lacus Corp.	b-2598	422	Mas de 400 transacciones
Lorem Ipsum Dolor Corp.	b-2546	1465	Mas de 400 transacciones
Amet Luctus Vulputate Foundation	b-2526	410	Mas de 400 transacciones
Nulla Integer Vulputate Corp.	b-2558	1447	Mas de 400 transacciones
Auctor Mauris Corp.	b-2550	1561	Mas de 400 transacciones
Fisimend Mauris Inetih ita	h-7590	417	Mas de 400 transacciones

Result 17 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:56:37	SELECT c.company_name, c.id, COUNT(t.id) AS cantidad_transacciones, IF (COUNT(t.id) > 400, 'Mas de 40...	100 row(s) returned	0.282 sec / 0.000 s

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 condición y es más simple para el lenguaje.