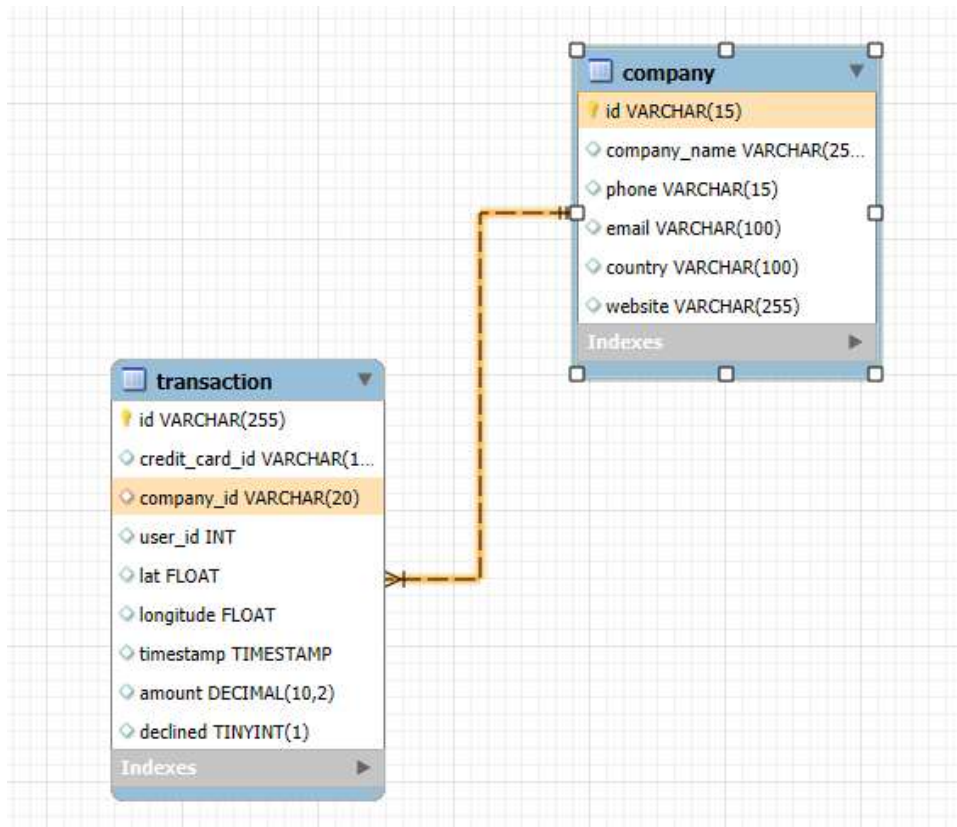


Nivel 1

**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.

**Comentario:**

De acuerdo a la exploración de los detalles de la base de datos se pueden considerar las siguientes características del esquema:

- La relación entre ellas es de **uno a muchos (N:1)**:
  - ✓ Una **empresa (company)** puede tener **muchas transacciones (transaction)**.
  - ✓ Cada transacción pertenece a una sola empresa, enlazada mediante el campo **company\_id**.
- El esquema está diseñado para almacenar información de **transacciones financieras** realizadas por usuarios, vinculadas a empresas.

A continuación, podemos observar en la figura 1 que el detalle de la información, nombre de la tabla, nombre de columnas y tipo de dato esta disponible en la interface de Worbench. En la tabla 1 y tabla 2 podemos apreciar el detalle del campo o dimensión de las dos tablas que conforman el esquema junto a su descripción.

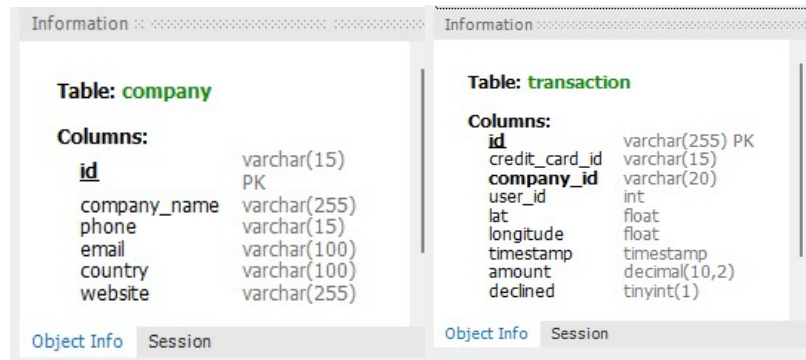


Figura 1. Nombre de columnas y tipo de variables.

Tabla 1. Características de la tabla de dimensiones

Campo	Tipo de dato	Descripción
<b>id</b>	VARCHAR(15)	Identificador único de la empresa (Primary Key).
<b>company_name</b>	VARCHAR(255)	Nombre de la empresa.
<b>phone</b>	VARCHAR(15)	Teléfono de contacto.
<b>email</b>	VARCHAR(100)	Correo electrónico de la empresa.
<b>country</b>	VARCHAR(100)	País donde opera la empresa.
<b>website</b>	VARCHAR(255)	Sitio web de la empresa.

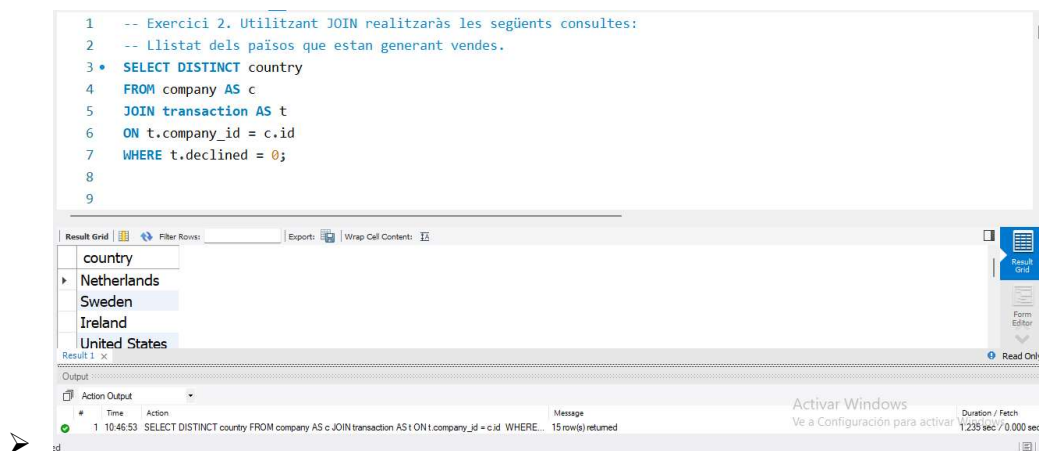
Tabla 2. Características de la tabla de hechos.

Campo	Tipo de dato	Descripción
<b>id</b>	VARCHAR(255)	Identificador único de la transacción (Primary Key).
<b>credit_card_id</b>	VARCHAR(15)	Identificador de la tarjeta de crédito utilizada.
<b>company_id</b>	VARCHAR(20)	Identificador de la empresa asociada (Foreign Key que referencia company.id).
<b>user_id</b>	INT	Identificador del usuario que realiza la transacción.

Campo	Tipo de dato	Descripción
lat	FLOAT	Latitud del lugar donde se realizó la transacción.
longitude	FLOAT	Longitud del lugar donde se realizó la transacción.
timestamp	TIMESTAMP	Fecha y hora en la que se efectuó la transacción.
amount	DECIMAL(10,2)	Monto de la transacción.
declined	TINYINT(1)	Indicador de si la transacción fue rechazada (1 = Sí, 0 = No).

**Ejercicio 2.** Usando JOIN realizarás las siguientes consultas:

- Listado de los países que están generando ventas.



*Figura 2. Cantidad de países que generan ventas.*

### Comentario

La consulta de la figura 2 obtiene la lista de países que generan ventas usando SELECT DISTINCT para evitar duplicados. Parte de company (c) y realiza un INNER JOIN con transaction (t) mediante t.company\_id = c.id. La cláusula WHERE filtra solo las transacciones aprobadas (declined = 0), mostrando así cada país con ventas válidas.

- Desde cuántos países se generan las ventas.



Figura 3. Cantidad de países que generan ventas

### Comentario

La figura 3 muestra una consulta SQL que obtiene el número total de países desde los que se generan ventas. Para ello se utiliza `COUNT(DISTINCT c.country)`, lo que asegura contar solo países únicos. La consulta parte de `company (c)` y hace un `INNER JOIN` con `transaction (t)` mediante `t.company_id = c.id`. Además, se filtran únicamente las transacciones aprobadas con `t.declined = 0`. El resultado es el total de países con ventas, que en este caso es 15.

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

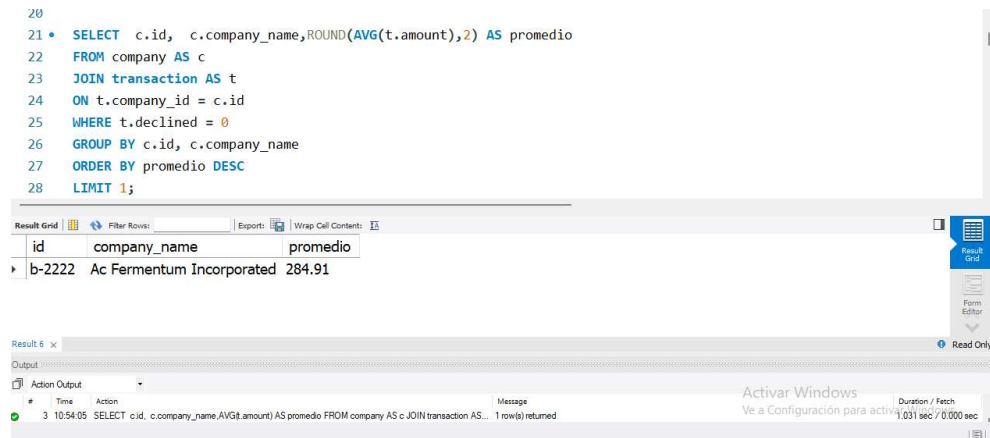


Figura 4 Compañía con media de ventas más alta.

### Comentario

La figura 4 muestra una consulta SQL que identifica la empresa con el promedio de ventas más alto considerando solo transacciones aprobadas. Se hace `JOIN` entre `company (c)` y `transaction (t)` mediante `t.company_id = c.id`, filtrando con `t.declined = 0`. Luego se calcula `ROUND(AVG(t.amount), 2)` como promedio por empresa, agrupando por `c.id` y

c.company\_name, y ordenando de forma descendente. Con LIMIT 1 se obtiene solo la empresa con el valor más alto. El resultado indica que Ac Fermentum Incorporated tiene el mayor promedio de ventas: 284.91.

### Ejercicio 3. Utilizando sólo subconsultas (sin utilizar JOIN):

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

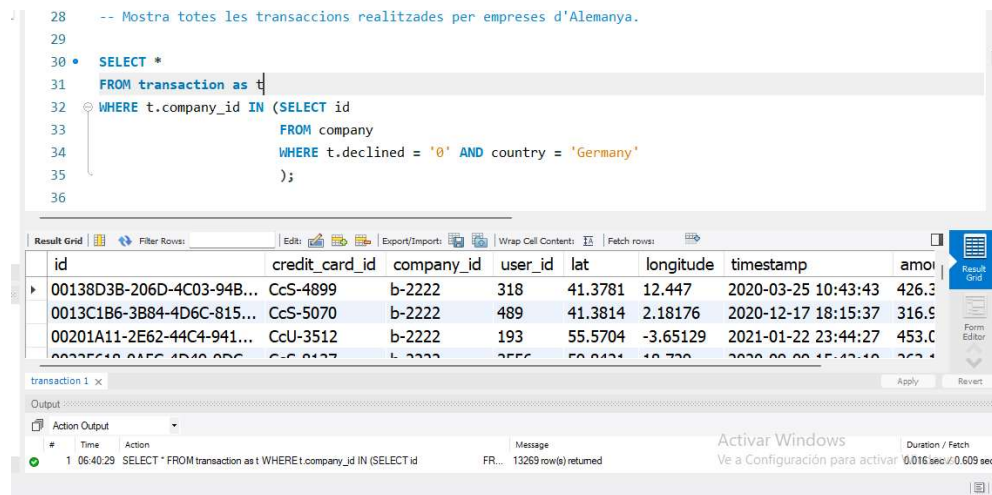


Figura 5. Transacciones realizadas por empresas alemanas.

### Comentario

La figura 5 muestra una consulta SQL que obtiene todas las transacciones realizadas por empresas ubicadas en Alemania. La consulta parte de transaction (t) y utiliza un subquery que selecciona los id de las empresas en company cuyo country = 'Germany' y cuyas transacciones no fueron rechazadas (declined = 0). Mediante IN, se devuelven solo las transacciones cuyo company\_id coincide con esos id. El resultado muestra 13.269 transacciones correspondientes a compañías alemanas.

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

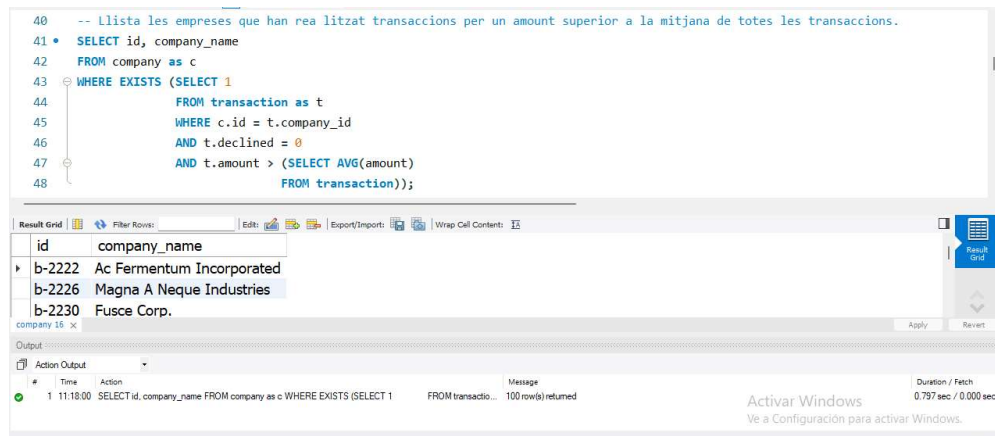


Figura 6. Empresas con transacciones superior a la media de las mismas.

### Comentario

La figura 6 muestra una consulta SQL que obtiene las empresas que tienen al menos una transacción aprobada con un importe superior al promedio general de todas las transacciones. Se parte de company (c) y, mediante WHERE EXISTS, se verifica la existencia de transacciones en transaction (t) vinculadas por c.id = t.company\_id, filtrando solo las aprobadas (t.declined = 0) y cuyo t.amount sea superior al AVG(amount). El resultado incluye tres empresas que registran operaciones por encima de la media.

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

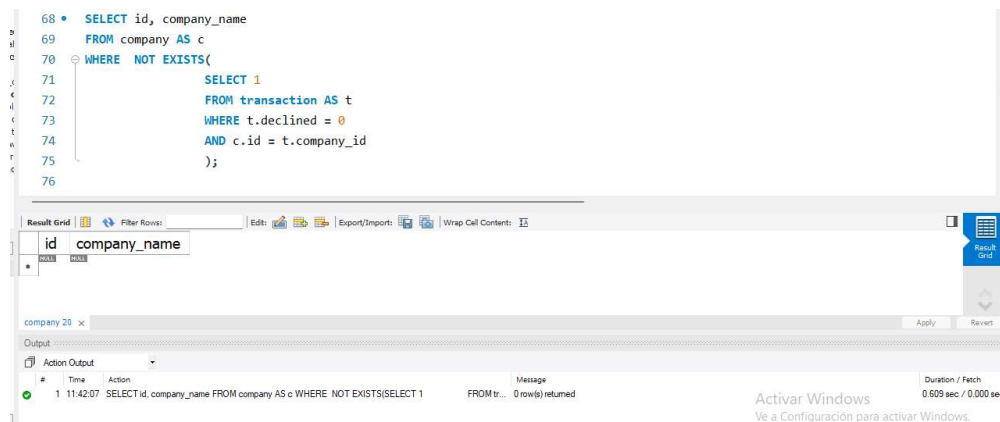


Figura 7. Empresas sin transacciones realizadas. Comentario:

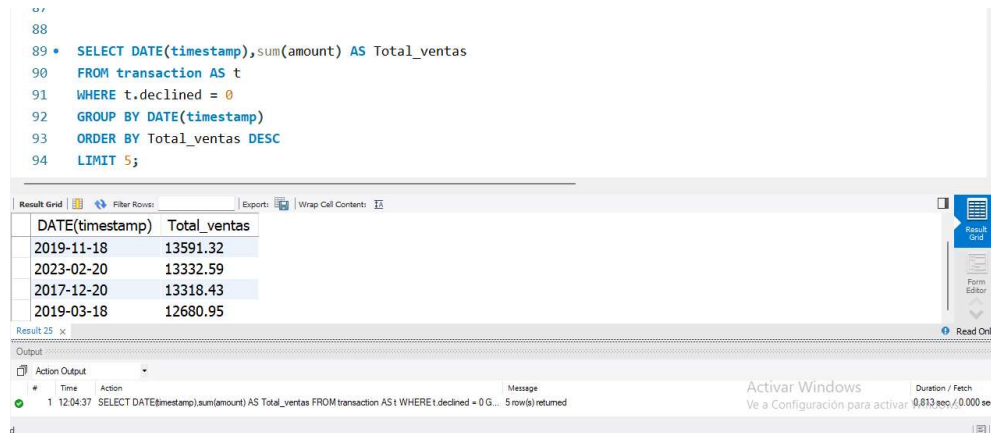
La figura 7 muestra una consulta SQL que obtiene las empresas que no tienen ninguna transacción aprobada. Partiendo de company (c), se usa WHERE NOT EXISTS para verificar que no exista en transaction (t), relacionada por c.id = t.company\_id y ninguna transacción con t.declined = 0. Como resultado, la consulta devuelve 0 empresas, lo que indica que todas cuentan

con al menos una transacción aprobada. Es importante mencionar que podría usarse NOT IN, pero NOT EXISTS se prefiere por su mejor manejo de valores NULL, mayor eficiencia, legibilidad y mayor solidez lógica.

## 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.



*Figura 8. TOP 5 de días de mayor ingreso por ventas.*

### Comentario:

La figura 8 muestra una consulta SQL que obtiene los cinco días con mayor volumen total de ventas. Partiendo de transaction (t), se selecciona DATE(timestamp) y SUM(amount) como Total\_ventas, filtrando solo transacciones aprobadas (t.declined = 0). Luego se agrupan los registros por fecha, se ordenan de forma descendente según el total vendido y se aplica LIMIT 5. El result grid muestra los días con los valores de ventas más altos del conjunto de datos.

### Ejercicio 2

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



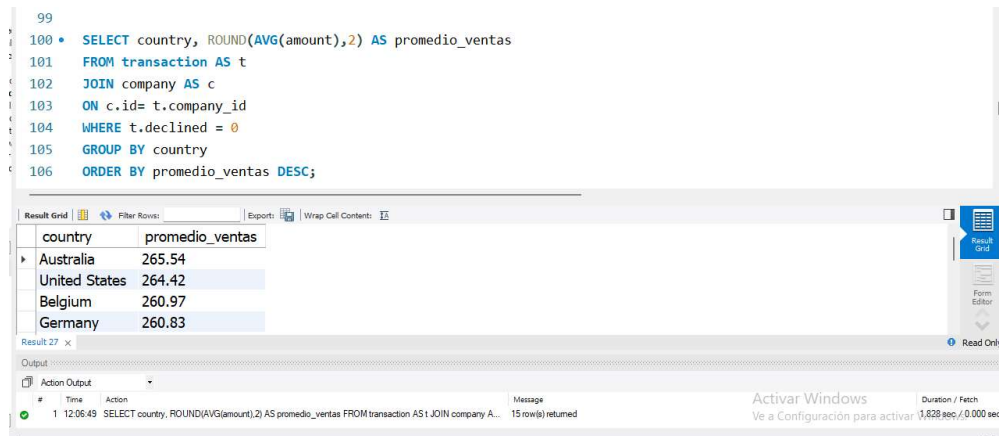


Figura 9. Media de ventas por país.

### Comentario

La figura 9 muestra una consulta SQL que calcula el promedio de ventas por país. Se hace JOIN entre las tablas transaction (t) y company (c) mediante `c.id = t.company_id`, filtrando solo transacciones aprobadas (`t.declined = 0`). El promedio se obtiene con `ROUND(AVG(amount), 2)`, y se agrupa por country. Luego los resultados se ordenan de forma descendente para mostrar primero los países con mayor promedio de ventas. Según el resultado, Australia, Estados Unidos, Bélgica y Alemania presentan los promedios más altos.

### 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.

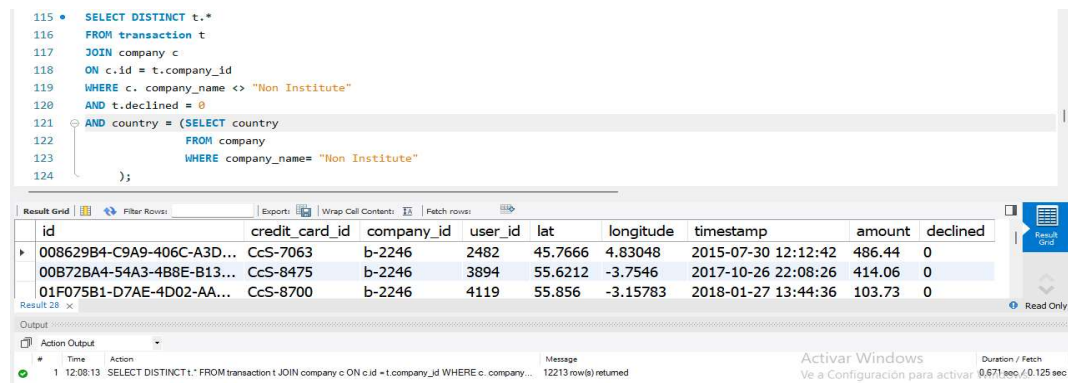



Figura 10. Lista de transacciones realizadas por empresas aplicando JOIN y subconsultas.



### Comentario

La figura 10 muestra una consulta SQL que obtiene todas las **transacciones aprobadas** realizadas por empresas que **no son “Non Institute”**, pero que operan en el **mismo país** que dicha empresa. Se unen **transaction (t)** y **company (c)** mediante **c.id = t.company\_id**, se excluye a “Non Institute” (**c.company\_name <> 'Non Institute'**) y se filtran solo transacciones aprobadas (**t.declined = 0**). Un subquery obtiene el país en el que opera “Non Institute”, permitiendo devolver únicamente las transacciones de otras empresas ubicadas en ese mismo país. Muestra el listado aplicando solo subconsultas.



The screenshot shows a SQL query in a query editor. The query is as follows:

```

129 SELECT *
130 FROM transaction t
131 WHERE t.declined = 0
132 AND EXISTS(SELECT 1
133             FROM company c
134             WHERE c.id = t.company_id
135             AND c.country = (SELECT country
136                             FROM company c
137                             WHERE company_name = "Non Institute"))
138 );
139

```

Below the query, the 'Result Grid' is displayed, showing the results of the query. The grid has the following columns: id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, amount, and declined. The first three rows are visible:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
008629B4-C9A9-406C-A3D...	CcS-7063	b-2246	2482	45.7666	4.83048	2015-07-30 12:12:42	486.44	0
00B72BA4-54A3-4B8E-B13...	CcS-8475	b-2246	3894	55.6212	-3.7546	2017-10-26 22:08:26	414.06	0
01F075B1-D7AE-4D02-AA...	CcS-8700	b-2246	4119	55.856	-3.15783	2018-01-27 13:44:36	103.73	0

At the bottom, the 'Action Output' shows the execution details: 12:08:13 SELECT DISTINCT t.\* FROM transaction t JOIN company c ON c.id = t.company\_id WHERE c.company\_name <> 'Non Institute' AND t.declined = 0. 12213 row(s) returned.

Figura 11. Lista de transacciones realizadas con subconsultas.

### Comentario

La figura 11 muestra una consulta SQL que obtiene todas las transacciones aprobadas pertenecientes a empresas ubicadas en el mismo país que “Non Institute”. Se inicia de **transaction (t)** filtrando **t.declined = 0**, y mediante **WHERE EXISTS** se verifica que exista una empresa donde **c.id = t.company\_id** y que **country** coincida con el país obtenido en la subquery que busca el país de “Non Institute”.

El resultado devuelve 12.213 transacciones aprobadas de empresas situadas en ese mismo país. Es importante destacar que usar **EXISTS** es más eficiente que un **JOIN**, ya que evita generar resultados intermedios y se detiene al encontrar la primera coincidencia.

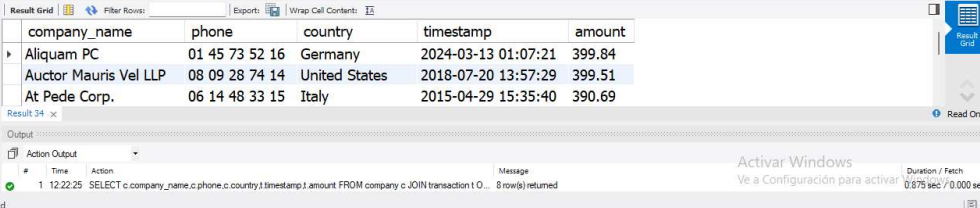
Nivel 3**Ejercicio 1**

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.

```

145
146 * SELECT c.company_name,c.phone,c.country,t.timestamp,t.amount
147 FROM company c
148 JOIN transaction t
149 ON c.id = t.company_id
150 WHERE t.declined = 0 AND t.amount BETWEEN 350 AND 400
151 AND DATE(timestamp) IN ('2015-04-29', '2018-07-20', '2024-03-13')
152 ORDER BY t.amount DESC;
153

```



company_name	phone	country	timestamp	amount
Aliquam PC	01 45 73 52 16	Germany	2024-03-13 01:07:21	399.84
Auctor Mauris Vel LLP	08 09 28 74 14	United States	2018-07-20 13:57:29	399.51
At Pede Corp.	06 14 48 33 15	Italy	2015-04-29 15:35:40	390.69

Figura 12. Empresas que realizaron transacciones con un valor comprendido entre 350 y 400 euros

**Comentario**

La figura 12 muestra una consulta SQL que obtiene transacciones aprobadas restringidas a un importe entre 350 y 400 y que ocurrieron en fechas específicas. Se hace JOIN entre company (c) y transaction (t) mediante c.id = t.company\_id, filtrando t.declined = 0, t.amount BETWEEN 350 AND 400, y DATE(timestamp) dentro de un conjunto de fechas proporcionado.

La consulta devuelve datos de la empresa y de cada transacción, y ordena los resultados por t.amount DESC. El result grid muestra tres transacciones que cumplen todos los criterios, pertenecientes a empresas de Alemania e Italia.

**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.

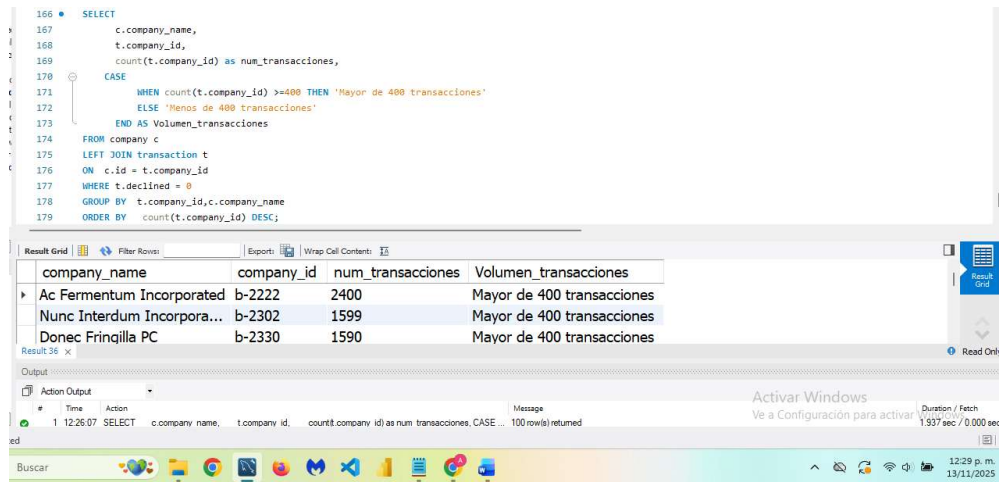


Figura 13. Listado de las empresas que tienen más de 400 transacciones o menos.

### Comentar

La figura 13 muestra una consulta SQL que resume el volumen de transacciones aprobadas por empresa, clasificándolas según si superan o no las 400 transacciones. Se usa company (c) con un LEFT JOIN a transaction (t) para incluir todas las empresas, filtrando solo transacciones aprobadas ( $t.declined = 0$ ). Se calcula  $COUNT(t.transaction_id)$  como num\_transacciones y un CASE asigna la categoría correspondiente según si el total es  $\geq 400$ . Los resultados se agrupan por empresa y se ordenan de forma descendente por cantidad de transacciones. El resultado muestra tres empresas con más de 1.500 transacciones aprobadas.

Se aplicó CASE considerando que en el enunciado se plantea que se quiere un listado de las “empresas en las que especifiques si tienen más de 400 transacciones o menos”, es decir que incluyen los dos casos.