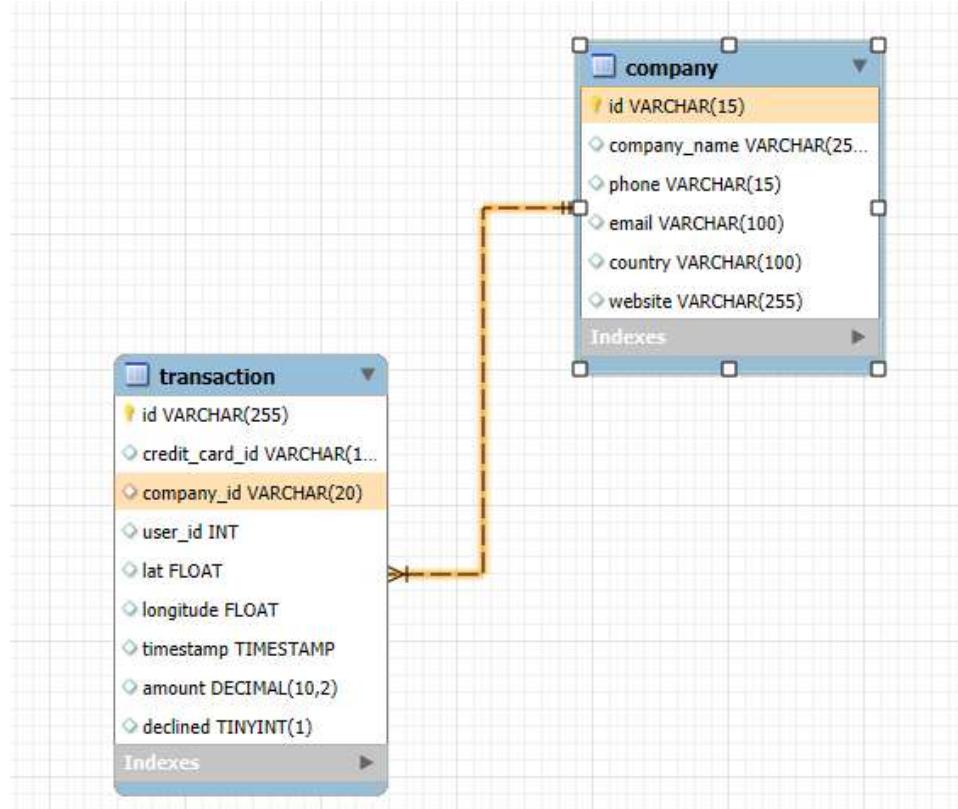


Nivel 1

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.

The screenshot shows two side-by-side 'Information' panes in MySQL Workbench. The left pane is for the 'company' table, which has columns: id (VARCHAR(15) PK), company_name (VARCHAR(255)), phone (VARCHAR(15)), email (VARCHAR(100)), country (VARCHAR(100)), and website (VARCHAR(255)). The right pane is for the 'transaction' table, which has columns: id (VARCHAR(255) PK), credit_card_id (VARCHAR(15)), company_id (VARCHAR(20)), user_id (int), lat (float), longitude (float), timestamp (timestamp), amount (decimal(10,2)), and declined (tinyint(1)). Both panes have tabs for 'Object Info' and 'Session' at the bottom.

Figura 1. Nombre de columnas y tipo de variables.

Tabla 1. Características de la tabla de dimensiones

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

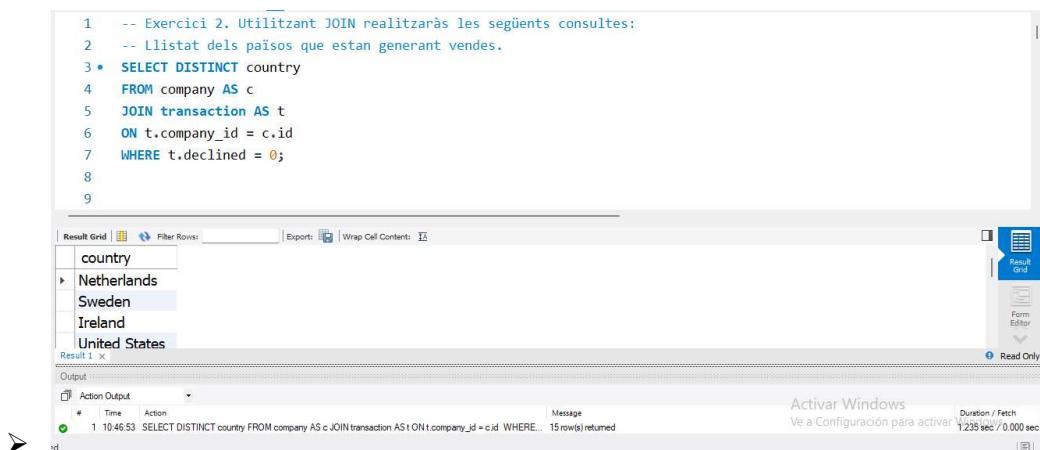
Tabla 2. Características de la tabla de hechos.

Campo	Tipo de dato	Descripción
id	VARCHAR(255)	Identificador único de la transacción (Primary Key).
credit_card_id	VARCHAR(15)	Identificador de la tarjeta de crédito utilizada.
company_id	VARCHAR(20)	Identificador de la empresa asociada (Foreign Key que referencia company.id).
user_id	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.



```

1 -- Exercici 2. Utilitzant JOIN realitzaràs les següents consultes:
2 -- Llistat dels països que estan generant vendes.
3 • SELECT DISTINCT country
4   FROM company AS c
5   JOIN transaction AS t
6   ON t.company_id = c.id
7   WHERE t.declined = 0;
8
9

```

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL code above.
- Result Grid:** Shows the results of the query. The column is labeled "country" and contains four rows: "Netherlands", "Sweden", "Ireland", and "United States".
- Output Tab:** Displays the execution message: "1 10:46:53 SELECT DISTINCT country FROM company AS c JOIN transaction AS t ON t.company_id = c.id WHERE... 15 row(s) returned".
- Message Bar:** Shows "Activar Windows" and "Duration / Fetch: 1.235 sec / 0.000 sec".

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.

```

11
12 -- Des de quants països es generen les vendes.
13 • SELECT count(DISTINCT country) AS num_paises
14   FROM company AS c
15   JOIN transaction AS t
16     ON t.company_id = c.id
17   WHERE t.declined = 0;
18

```

The screenshot shows a MySQL query execution interface. The query is:

```

SELECT count(DISTINCT country) AS num_paises
FROM company AS c
JOIN transaction AS t
ON t.company_id = c.id
WHERE t.declined = 0;

```

The result grid displays one row with the column 'num_paises' containing the value '15'. Below the result grid, the 'Output' pane shows the query and its execution details, including the duration of 0.781 sec / 0.000 sec.

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.

```

20
21 • SELECT c.id, c.company_name,ROUND(AVG(t.amount),2) AS promedio
22   FROM company AS c
23   JOIN transaction AS t
24     ON t.company_id = c.id
25   WHERE t.declined = 0
26   GROUP BY c.id, c.company_name
27   ORDER BY promedio DESC
28   LIMIT 1;

```

The screenshot shows a MySQL query execution interface. The query is:

```

SELECT c.id, c.company_name,ROUND(AVG(t.amount),2) AS promedio
FROM company AS c
JOIN transaction AS t
ON t.company_id = c.id
WHERE t.declined = 0
GROUP BY c.id, c.company_name
ORDER BY promedio DESC
LIMIT 1;

```

The result grid displays one row with columns 'id', 'company_name', and 'promedio'. The values are 'b-2222' and 'Ac Fermentum Incorporated' respectively, with 'promedio' set to 284.91. Below the result grid, the 'Output' pane shows the query and its execution details, including the duration of 1.031 sec / 0.000 sec.

Figura 4 Compañía con media de ventas más alta.

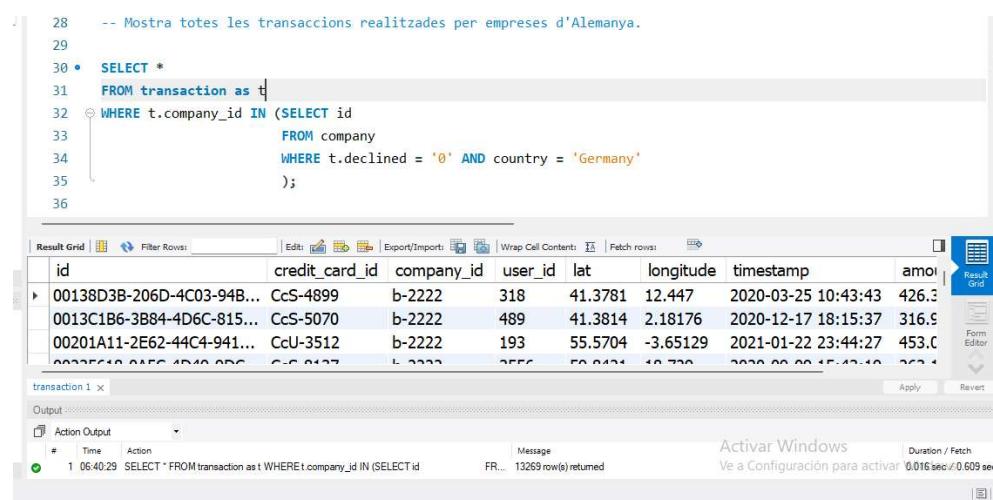
Comentario

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.
-



```

28 -- Mostra totes les transaccions realitzades per empreses d'Alemanya.
29
30 • SELECT *
31   FROM transaction as t
32   WHERE t.company_id IN (SELECT id
33     FROM company
34     WHERE t.declined = '0' AND country = 'Germany'
35   );
36

```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount
00138D3B-206D-4C03-9499	CcS-4899	b-2222	318	41.3781	12.447	2020-03-25 10:43:43	426.3
0013C1B6-3B84-4D6C-815...	CcS-5070	b-2222	489	41.3814	2.18176	2020-12-17 18:15:37	316.9
00201A11-2E62-44C4-941...	CcU-3512	b-2222	193	55.5704	-3.65129	2021-01-22 23:44:27	453.0
00222610-0AE0-4D40-92C...	CcS-8127	b-2222	2555	50.0421	10.720	2020-09-22 15:42:10	261.4

transaction 1 ×

Action Output

Time Action Message Activar Windows Ve a Configuración para activar Duration / Fetch

1 06:40:29 SELECT * FROM transaction as t WHERE t.company_id IN (SELECT id FR... 13269 row(s) returned 0.016 sec / 0.609 sec

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.

```

40 -- Llista les empreses que han rea litzat transaccions per un amount superior a la mitjana de totes les transaccions.
41 • SELECT id, company_name
42   FROM company AS c
43   WHERE EXISTS (SELECT 1
44     FROM transaction AS t
45     WHERE c.id = t.company_id
46     AND t.declined = 0
47     AND t.amount > (SELECT AVG(amount)
48                      FROM transaction));

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Apply | Revert | Output: | Action Output: | # | Time | Action | Message | Duration / Fetch | Activar Windows | Ve a Configuración para activar Windows.

id	company_name
b-2222	Ac Fermentum Incorporated
b-2226	Magna A Neque Industries
b-2230	Fusce Corp.

Action Output: # 1 11:18:00 SELECT id, company_name FROM company AS c WHERE EXISTS (SELECT 1 FROM transaction... 100 row(s) returned)

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.

```

68 • SELECT id, company_name
69   FROM company AS c
70   WHERE NOT EXISTS(
71     SELECT 1
72       FROM transaction AS t
73       WHERE t.declined = 0
74       AND c.id = t.company_id
75   );

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Apply | Revert | Output: | Action Output: | # | Time | Action | Message | Duration / Fetch | Activar Windows | Ve a Configuración para activar Windows.

id	company_name

Action Output: # 1 11:42:07 SELECT id, company_name FROM company AS c WHERE NOT EXISTS(SELECT 1 FROM transaction... 0 row(s) returned)

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.

The screenshot shows a MySQL query results grid. The query is:

```
88
89 •  SELECT DATE(timestamp),sum(amount) AS Total_ventas
90   FROM transaction AS t
91   WHERE t.declined = 0
92   GROUP BY DATE(timestamp)
93   ORDER BY Total_ventas DESC
94   LIMIT 5;
```

The result grid displays the following data:

DATE(timestamp)	Total_ventas
2019-11-18	13591.32
2023-02-20	13332.59
2017-12-20	13318.43
2019-03-18	12680.95

Below the grid, the message "5 row(s) returned" is visible. The interface includes standard database management tools like Result Grid, Form Editor, and a Read Only option.

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.

```

99
100 •  SELECT country, ROUND(AVG(amount),2) AS promedio_ventas
101   FROM transaction AS t
102   JOIN company AS c
103   ON c.id= t.company_id
104   WHERE t.declined = 0
105   GROUP BY country
106   ORDER BY promedio_ventas DESC;

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid Form Editor Read Only

country	promedio_ventas
Australia	265.54
United States	264.42
Belgium	260.97
Germany	260.83

Action Output # Time Action Message Activar Windows Duration / Fetch

1 12:06:49 SELECT country, ROUND(AVG(amount),2) AS promedio_ventas FROM transaction AS t JOIN company A... 15 row(s) returned Ve a Configuración para activar 1.829 sec./0.000 sec

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)), 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.

```

115 •  SELECT DISTINCT t.*
116   FROM transaction t
117   JOIN company c
118   ON c.id = t.company_id
119   WHERE c.company_name <> "Non Institute"
120   AND t.declined = 0
121   AND country = (SELECT country
122     FROM company
123     WHERE company_name = "Non Institute"
124   );

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows: Result Grid Form Editor Read Only

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

Action Output # Time Action Message Activar Windows Duration / Fetch

1 12:08:13 SELECT DISTINCT t.* FROM transaction t JOIN company c ON c.id = t.company_id WHERE c.company_name <> "Non Institute" 12213 row(s) returned Ve a Configuración para activar 0.671 sec./0.125 sec

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.

```

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

```

The screenshot shows a SQL query editor with the code above. Below it is a result grid displaying transaction data:

	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
	00B728A4-54A3-4B8E-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

Below the grid, the message "12213 rows(s) returned" is visible. The bottom status bar shows "Duration / Fetch: 0.671 sec / 0.125 sec".

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
Autor 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.

```

166 •    SELECT
167     c.company_name,
168     t.company_id,
169     count(t.company_id) as num_transacciones,
170     CASE
171         WHEN count(t.company_id) >=400 THEN 'Mayor de 400 transacciones'
172         ELSE 'Menos de 400 transacciones'
173     END AS Volumen_transacciones
174
175     FROM company c
176     LEFT JOIN transaction t
177     ON c.id = t.company_id
178     WHERE t.declined = 0
179     GROUP BY t.company_id,c.company_name
180     ORDER BY count(t.company_id) DESC;

```

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

company_name	company_id	num_transacciones	Volumen_transacciones
Ac Fermentum Incorporated	b-2222	2400	Mayor de 400 transacciones
Nunc Interdum Incorpora...	b-2302	1599	Mayor de 400 transacciones
Donec Fringilla PC	b-2330	1590	Mayor de 400 transacciones

Result 36 ×

Output

Action Output

Time Action Message

1 12:26:07 SELECT c.company_name, t.company_id, count(t.company_id) as num_transacciones, CASE ... 100 row(s) returned

Activar Windows
Vé a Configuración para activar...

Duration / Fetch
1.937 sec / 0.000 sec

Buscar 12:29 p. m.
13/11/2025

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.company_id) como num_transacciones y un CASE asigna la categoría correspondiente según si el total es ≥ 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.