
SPRINT 2

IT ACADEMY

Sergi Alcolea de la Gala

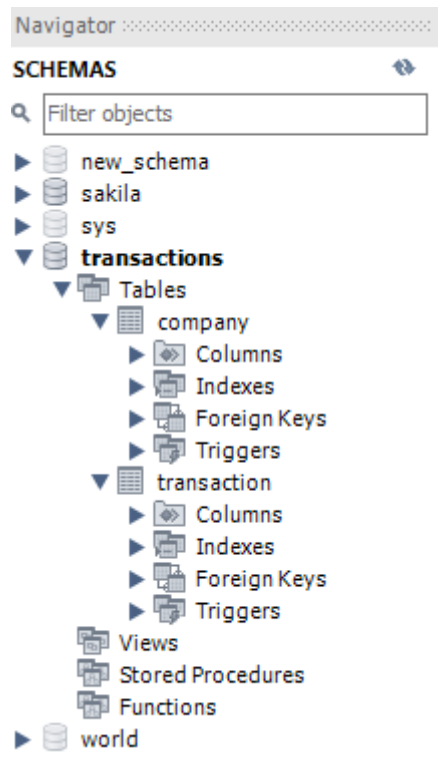
Nivell 1

Tasca S2.01. Nocions bàsiques SQL

Exercici 1

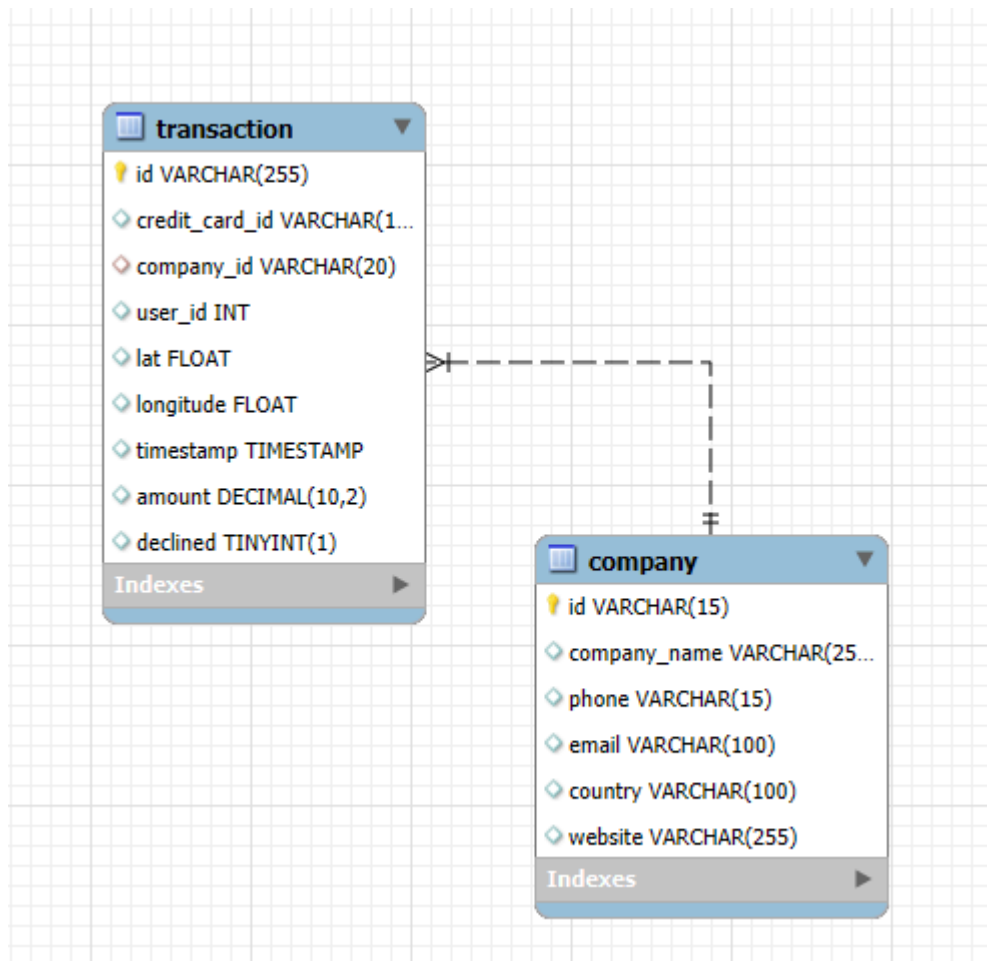
A partir dels documents adjunts (estructura_dades i dades_introduir), importa les dues taules. Mostra les característiques principals de l'esquema creat i explica les diferents taules i variables que existeixen. Assegura't d'incloure un diagrama que il·lustri la relació entre les diferents taules i variables.

Primero, he importado el documento “estructura dades” en workbench, y al ejecutarla obtuve el siguiente resultado:



.Tras crearse todas las tablas de las bases de datos skila y transactions, emplee el archivo “dades_introduir_sprint2”, que introdujo miles de datos a todas las tablas y campos anteriormente creados

Una vez todo listo, utilicé *Database* → *Reverse Engineering* para crear un diagrama de la base de datos “transactions”:



Observamos dos tablas interconectadas por una línea discontinua que conecta “transaction.company_id” con “[company.id](#)”. Se trata de una relación de uno a muchos (1:N), donde la tabla “transactions” ejerce de tabla maestra porque de ella depende la tabla company. Esto es porque, en esta base de datos, una compañía puede figurar en muchas transacciones, pero no al revés.

El tipo de conexión mediante Foreign Key (FK) asegura la integridad referencial, es decir, no puede existir una transacción asociada a una empresa que no esté en la tabla; hablando de las cuales...

- 1) La Tabla “company” guarda la información de diferentes empresas, y funciona como tabla maestra, ya que la tabla “transactions” depende de ella. Contiene los siguientes campos:
 - a) **id** → identificador único de la empresa.
 - b) **company_name** → nombre de la empresa.
 - c) **phone** → teléfono de contacto.
 - d) **email** → correo electrónico.
 - e) **country** R(100)) → país de la empresa.
 - f) **website** → página web oficial.

- 2) Por otro lado, tenemos la tabla “transaction”, la cual almacena las transacciones realizadas por cada empresa. Es la conexión 1. Contiene los siguientes campos:
- a) `id` → identificador único de la transacción.
 - b) `credit_card_id` → referencia a una tarjeta de crédito (FK hacia `credit_card`, que aún no está en el diagrama).
 - c) `company_id` → identifica a la empresa donde se realizó la transacción.
 - d) `user_id` → referencia al usuario que hizo la operación
 - e) `lat` → latitud del lugar de la transacción.
 - f) `longitude` → longitud geográfica.
 - g) `timestamp` → fecha y hora de la operación.
 - h) `amount` → importe de la transacción.
 - i) `declined` → indica si la transacción fue rechazada (0 = no, 1 = sí).

Exercici 2

Utilitzant JOIN realitzaràs les següents consultes:

- Llistat dels països que estan generant vendes.
- Des de quants països es generen les vendes.
- Identifica la companyia amb la mitjana més gran de vendes.

2.1: Lo primero que queremos hacer es saber aquellos países que están generando ventas:

```
USE transactions;
SELECT DISTINCT company.country
FROM company
JOIN transaction ON company.id =
transaction.company_id
WHERE transaction.declined = 0;
```

The screenshot shows a database query interface. The query editor at the top contains the following SQL code:

```
11 • SELECT DISTINCT company.country
12 FROM company
13 JOIN transaction ON company.id = transaction.company_id
14 WHERE transaction.declined = 0;
15
```

Below the query editor, the "Result Grid" tab is active, displaying a list of countries:

country
Netherlands
Sweden
Ireland
United States
Belgium
Canada
Germany

At the bottom, the "Output" panel shows the "Action Output" for the query execution:

#	Time	Action	Message	Duration / Fetch
7	17:58:49	SELECT DISTINCT company.country FROM com...	15 row(s) returned	0.188 sec / 0.000 sec
8	18:02:55	SELECT DISTINCT company.country FROM com...	15 row(s) returned	0.172 sec / 0.000 sec

2.2: Ahora, con los datos de ambas tablas interconectados, vamos a ver desde cuántos países se generan las ventas. Basta con contar los países distintos del listado anterior:

```
SELECT COUNT(DISTINCT company.country) AS total_paises
FROM company
JOIN transaction
    ON transaction.company_id = company.id
WHERE transaction.declined = 0;
```

The screenshot shows a SQL IDE interface. At the top, a SQL query is entered in a text area:

```
16 -- 2.2 Des de quants països es generen les vendes.
17 • SELECT COUNT(DISTINCT company.country) AS total_paises
18 FROM company
19 JOIN transaction
20     ON transaction.company_id = company.id
21 WHERE transaction.declined = 0;
```

Below the query editor, the 'Result Grid' is displayed, showing a single column 'total_paises' with the value '15'.

At the bottom, the 'Output' pane shows the 'Action Output' table:

#	Time	Action	Message	Duration / Fetch
✓ 11	18:12:28	SELECT COUNT(DISTINCT company.country) A...	1 row(s) returned	0.172 sec / 0.000 sec
✓ 12	18:12:37	SELECT COUNT(DISTINCT company.country) A...	1 row(s) returned	0.157 sec / 0.000 sec

2.3: Por último, calcularemos la media (AVG) del campo amount de las transacciones, agrupadas por compañía. Después, elegimos las más alta con "order by" o "LIMIT". Entonces, necesitaremos:

- el nombre de la compañía
- el amount
- media de ventas (crear)

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

```

23  -- 2.3 Identifica la companyia amb la mitjana més gran de vendes.
24  • SELECT company.id,company.company_name,
25      ROUND(AVG(transaction.amount), 2) AS media_ventas
26  FROM company
27  JOIN transaction ON company.id = transaction.company_id
28  WHERE transaction.declined = 0
29  GROUP BY company.id, company.company_name
30  ORDER BY media_ventas DESC
31  LIMIT 1;

```

Result Grid			
	id	company_name	media_ventas
▶	b-2222	Ac Fermentum Incorporated	284.91

Result 8 x						
Read Only						
Output						
Action Output						
#	Time	Action	Message	Duration / Fetch		
✓ 18	18:30:48	SELECT company.id, company.company_n...	1 row(s) returned	0.219 sec / 0.000 sec		
✓ 19	18:34:24	SELECT company.id,company.company_name, ...	1 row(s) returned	0.235 sec / 0.000 sec		

Exercici 3

Utilitzant només subconsultes (sense utilitzar JOIN):

- Mostra totes les transaccions realitzades per empreses d'Alemanya.
- Llista les empreses que han realitzat transaccions per un amount superior a la mitjana de totes les transaccions.
- Eliminaran del sistema les empreses que no tenen transaccions registrades, entrega el llistat d'aquestes empreses.

3.1 Para mostrar todas las transacciones realizadas por empresas en Alemania mediante subconsultas, primero tenemos que seleccionar las

```
SELECT *
FROM transaction
WHERE transaction.declined = 0
  AND EXISTS (
    SELECT id
    FROM company
    WHERE company.country = 'Germany'
      AND company.id = transaction.company_id
  )
GROUP BY transaction.id;
```

→ En este ejercicio he decidido especificar mucho para darme cuenta que todos los filtros se mantienen y solo vamos construyendo nuevos.

→ Primero hemos seleccionado la tabla transacciones, ya que nos interesa saber el id de cada operación. Inmediatamente después, he hecho una subconsulta en la que he empleado **SELECT** ya que es más eficiente y seguro que sus contrapartes IN o ANY, porque detiene la operación tan pronto como encuentra varias coincidencias que puedan afectar al procesamiento.

→ En la SUBCONSULTA empleamos además otros dos filtros: 1) el país del que queremos el id de la compañía que opera en Alemania. Pero para asegurarme, he añadido también AND para “company.id = transaction.company_id” que actúa como un JOIN. De este modo he podido usar GROUP BY [transaction.id](#) más tarde.

```

34  -- 3.1 Mostra totes les transaccions realitzades per empreses d'Alemanya.
35  •  SELECT *
36  FROM transaction
37  WHERE transaction.declined = 0
38  AND EXISTS (
39      SELECT id
40      FROM company
41      WHERE company.country = 'Germany'
42            AND company.id = transaction.company_id
43  )
44  GROUP BY transaction.id;

```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp
00138D3B-206D-4C03-94B7-63A2676EB9B4	CcS-4899	b-2222	318	41.3781	12.447	2020-03-25 10:43:...
0013C1B6-3B84-4D6C-8154-E2B3FEBCA8E9	CcS-5070	b-2222	489	41.3814	2.18176	2020-12-17 18:15:...
00201A11-2E62-44C4-941D-198FC8DB77F0	CcU-3512	b-2222	193	55.5704	-3.65129	2021-01-22 23:44:...
00235618-0A5C-4D49-9DCB-B3A9405D8923	CcS-8137	b-2222	3556	59.8421	18.729	2020-09-09 15:43:...

transaction 11 x Apply

Output

Action Output

#	Time	Action	Message	Duration / Fetch
24	18:52:52	SELECT * FROM transaction WHERE transaction...	1000 row(s) returned	0.047 sec / 0.000 sec
25	18:53:06	SELECT * FROM transaction WHERE transaction...	1000 row(s) returned	0.046 sec / 0.000 sec

3.2: Llista les empreses que han realitzat transaccions per un amount superior a la mitjana de totes les transaccions.

```

SELECT DISTINCT company.company_name
FROM company
WHERE EXISTS (
    SELECT 1
    FROM transaction
    WHERE transaction.company_id = company.id -- ← Condición crucial
    AND transaction.declined = 0
    AND transaction.amount > (
        SELECT ROUND(AVG(amount), 2) -- ← Coma agregada
        FROM transaction
        WHERE declined = 0
    )
);

```

Para listar las empresas que venden por encima del promedio,

- Primero calculamos la media con una subconsulta (`SELECT AVG(amount) FROM transaction`).
- Luego seleccionamos las empresas cuyos `id` aparecen en transacciones con `amount > media`


```

48 SELECT DISTINCT company.company_name
49 FROM company
50 WHERE EXISTS (
51     SELECT 1
52     FROM transaction
53     WHERE transaction.company_id = company.id -- Condición crucial
54     AND transaction.declined = 0
55     AND transaction.amount > (
56         SELECT ROUND(AVG(amount), 2) -- Coma agregada
57         FROM transaction
58         WHERE declined = 0
59     )
60 );

```

Result Grid

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

company 16 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
30	19:43:42	SELECT DISTINCT company.company_name, tra...	Error Code: 1064. You have an error in your SQL s...	0.000 sec
31	19:43:49	SELECT DISTINCT company.company_name FR...	100 row(s) returned	0.063 sec / 0.000 sec

3.3.- Debemos listar las empresas que no tienen transacciones antes de borrarla.

- j) Primero obtendremos el listado de todas las compañías de la tabla que han realizado transacciones. Si seleccionamos solo la subconsulta, nos dará una columna con el id de las compañías.
- k) añadimos distinct para eliminar la repetición de id de compañías.
- l) añadimos where not exists para ver el ID de compañías que NO tienen transacciones registradas. Pero entonces nos da NULL en la visualización, y la consola nos devuelve las mismas 100 filas que tenemos al inicio.

→ Por ende, obtendremos simplemente un listado de empresas que estará vacío.

```

SELECT *
FROM company
WHERE NOT EXISTS (
    SELECT DISTINCT company_id
    FROM TRANSACTION
);

```

```

62 -- 3.3 Eliminaran del sistema les empreses que no tenen transaccions registrades,
63 • SELECT *
64 FROM company
65 WHERE NOT EXISTS (
66     SELECT DISTINCT company_id
67     FROM transaction
68 );
69

```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

Result Grid

	id	company_name	phone	email	country	website
*	NULL	NULL	NULL	NULL	NULL	NULL

company 17 x

Apply

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 31	19:43:49	SELECT DISTINCT company.company_name FR...	100 row(s) returned	0.063 sec / 0.000 sec
✓ 32	20:02:33	SELECT * FROM company WHERE NOT EXIST...	0 row(s) returned	0.000 sec / 0.000 sec

Nivell 2

Exercici 1

Identifica els cinc dies que es va generar la quantitat més gran d'ingressos a l'empresa per vendes. Mostra la data de cada transacció juntament amb el total de les vendes.

1. sumar l'amount per dia:

```
SELECT
  DATE(transaction.timestamp) AS fecha,
  ROUND(SUM(transaction.amount),2) AS total_ventas
FROM transaction
WHERE declined = 0
GROUP BY DATE(transaction.timestamp)
ORDER BY total_ventas DESC
LIMIT 5;
```

```
70  -- NIVELL 2
71
72  -- EXERCICI 1: Identifica els cinc dies que es va generar la quantitat més gran d'ingressos a
73  • SELECT
74      DATE(transaction.timestamp) AS fecha,
75      ROUND(SUM(transaction.amount),2) AS total_ventas
76  FROM transaction
77  WHERE declined = 0
78  GROUP BY DATE(transaction.timestamp)
79  ORDER BY total_ventas DESC
80  LIMIT 5;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

Result Grid

	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 20 x

Read Only

Output

Action Output

	#	Time	Action	Message	Duration / Fetch
✖	35	20:09:45	SELECT	DATE(transaction.timestamp) AS fec... Error Code: 1064. You have an error in your SQL s...	0.000 sec
✔	36	20:10:05	SELECT	DATE(transaction.timestamp) AS fec... 5 row(s) returned	0.063 sec / 0.000 sec

Exercici 2

Quina és la mitjana de vendes per país? Presenta els resultats ordenats de major a menor mitjà.

```
SELECT
  company.country,
  ROUND(AVG(transaction.amount), 2) AS mitjana_vendes,
  COUNT(transaction.id) AS num_transaccions
FROM company
JOIN transaction
  ON transaction.company_id = company.id
WHERE transaction.declined = 0
GROUP BY company.country
ORDER BY mitjana_vendes DESC;
```

```
82  -- EXERCICI 2: Quina és la mitjana de vendes per país? Presenta els resultats ordenats de majo
83  •  SELECT
84      company.country,
85      ROUND(AVG(transaction.amount), 2) AS mitjana_vendes,
86      COUNT(transaction.id) AS num_transaccions
87  FROM company
88  JOIN transaction
89      ON transaction.company_id = company.id
90  WHERE transaction.declined = 0
91  GROUP BY company.country
92  ORDER BY mitjana_vendes DESC;
```

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

	country	mitjana_vendes	num_transaccions
▶	Australia	265.54	2544
	United States	264.42	4011
	Belgium	260.97	3515
	Germany	260.83	13269
	Ireland	260.39	2651

Result 24 x

Read Only

Output

Action Output

	#	Time	Action	Message	Duration / Fetch
✖	40	20:13:55	SELECT company.country, ROUND(AVG(tr...	Error Code: 1064. You have an error in your SQL s...	0.000 sec
✔	41	20:14:02	SELECT company.country, ROUND(AVG(tr...	15 row(s) returned	0.188 sec / 0.000 sec

→ Como de costumbre empezaremos seleccionando el país de la tabla “company”, calculando justo después la media de amount para sacar la media de transacciones. También, por iniciativa personal, he querido agregar un recuento del total de transacciones de cada país.

→ Para que esto funcionara, sin embargo, era necesario hacer un JOIN con la tabla de transactions para luego agrupar la unión por países, pero ordenando el orden en descendente para empezar por la media más alta de cada país.

Exercici 3

En la teva empresa, es planteja un nou projecte per a llançar algunes campanyes publicitàries per a fer competència a la companyia "Non Institute". Per a això, et demanen la llista de totes les transaccions realitzades per empreses que estan situades en el mateix país que aquesta companyia.

- Mostra el llistat aplicant JOIN i subconsultes.
- Mostra el llistat aplicant solament subconsultes.

```
SELECT
  company.company_name,
  transaction.id AS transaction_id,
  DATE(transaction.timestamp) AS data_transaccion,
  transaction.amount,
  transaction.declined
FROM transaction
JOIN company
  ON transaction.company_id = company.id
WHERE company.country = (
  SELECT country
  FROM company
  WHERE company_name = 'Non Institute'
  LIMIT 1
)
AND transaction.declined = 0
ORDER BY DATE(transaction.timestamp) DESC;
```

```

104 JOIN company
105 ON transaction.company_id = company.id
106 WHERE company.country = (
107     SELECT country
108     FROM company
109     WHERE company_name = 'Non Institute'
110     LIMIT 1
111 )
112 AND transaction.declined = 0
113 ORDER BY DATE(transaction.timestamp) DESC;
114
115 -- 3.2 Mostra el llistat aplicant solament subconsultes.
116 SELECT
117     (SELECT company_name FROM company WHERE id = transaction.company_id) AS company_name,
118     transaction.id AS transaction_id,

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: | **Result Grid**

	company_name	transaction_id	data_transaccion	amount	declined
▶	Amet Faudbus Ut Foundation	A4653920-06AD-43F0-B2EE-F942CF8C7177	2024-12-31	477.83	0
	Amet Faudbus Ut Foundation	973F22BD-E107-4EC5-8A40-015EF3602CED	2024-12-31	342.82	0
	Non Magna LLC	275A99C1-1B59-491F-85BE-E99CC122A120	2024-12-31	539.45	0
	Ac Libero Inc.	00F94C0B-AC1A-4BC6-AC21-486475CE838A	2024-12-31	248.19	0
	Enim Condimentum Ltd	DDD00334-AFFF-4DC5-B378-54204B70569A	2024-12-31	485.81	0

Result 32 x **Read Only**

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✖ 55	21:11:44	SELECT company.company_name, transacti...	Error Code: 1054. Unknown column 'company.co...	0.000 sec
✔ 56	21:15:13	SELECT company.company_name, transact...	1000 row(s) returned	0.047 sec / 0.000 sec

→ En este ejercicio, el elemento clave era “Non Institute”. Por ello, he seleccionado los la mayoría de campos de la tabla transacciones primero y sólo me he quedado con “country” de la tabla company. Luego, sólo he hecho una subconsulta específicamente para filtrar aquellos países que tenían “Non Institute”.

AHORA SIN JOIN!!!

→ Si queremos hacerlo solo con subconsultas, habrá que hacer varias.

```
SELECT
  (SELECT company_name FROM company WHERE company.id =
transaction.company_id LIMIT 1) AS company_name,
  transaction.id AS transaction_id,
  DATE(transaction.timestamp) AS data_transaccion,
  transaction.amount,
  transaction.declined
FROM transaction
WHERE transaction.company_id IN (
  SELECT id
  FROM company
  WHERE country = (
    SELECT country
    FROM company
    WHERE company_name = 'Non Institute'
    LIMIT 1
  )
)
AND transaction.declined = 0
ORDER BY DATE(transaction.timestamp) DESC;
```

```
114
115 -- 3.2 Mostra el llistat aplicant solament subconsultes.
116 ● SELECT
117   (SELECT company_name FROM company WHERE company.id = transaction.company_id LIMIT 1) AS company_name,
118   transaction.id AS transaction_id,
119   DATE(transaction.timestamp) AS data_transaccion,
120   transaction.amount,
121   transaction.declined
122 FROM transaction
123 WHERE transaction.company_id IN (
124   SELECT id
125   FROM company
126   WHERE country = (
127     SELECT country
128     FROM company
```

	company_name	transaction_id	data_transaccion	amount	declined
▶	Amet Faucibus Ut Foundation	A4653920-06AD-43F0-B2EE-F942CF8C7177	2024-12-31	477.83	0
	Amet Faucibus Ut Foundation	973F22BD-E107-4EC5-8A40-015EF3602CED	2024-12-31	342.82	0
	Non Magna LLC	275A99C1-1B59-491F-85BE-E99CC122A120	2024-12-31	539.45	0
	Ac Libero Inc.	00F94C0B-AC1A-4BC6-AC21-486475CE838A	2024-12-31	248.19	0
	Enim Condimentum Ltd	DDD00334-AFFF-4DC5-B378-54204B70569A	2024-12-31	485.81	0

Result 34 ×

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 57	21:17:17	SELECT	(SELECT company_name FROM co... 1000 row(s) returned	0.078 sec / 0.000 sec
✓ 58	21:17:21	SELECT	(SELECT company_name FROM co... 1000 row(s) returned	0.062 sec / 0.000 sec

→ La primera subconsulta toma todos los campos de la tabla "Transactions". Solamente tomamos el nombre de la compañía de la otra tabla.

→ La segunda subconsulta se centra en hacer coincidir el id de la compañía cuyo nombre es "Non" institute con el País en el que se ha realizado la transacción.

Nivell 3

Exercici 1

Presenta el nom, telèfon, país, data i amount, d'aquelles empreses que van realitzar transaccions amb un valor comprès entre 350 i 400 euros i en alguna d'aquestes dates: 29 d'abril del 2015, 20 de juliol del 2018 i 13 de març del 2024. Ordena els resultats de major a menor quantitat.

```
SELECT
  company.id AS company_id,
  company.company_name,
  company.phone,
  company.country,
  DATE(transaction.timestamp) AS data_transaction,
  transaction.amount
FROM company
JOIN transaction
  ON transaction.company_id = company.id
WHERE transaction.amount BETWEEN 350 AND 400
  AND DATE(transaction.timestamp) IN ('2015-04-29', '2018-07-20', '2024-03-13')
ORDER BY transaction.amount DESC;
```

135 ● SELECT

136 company.id,

137 company.company_name,

138 company.phone,

139 company.country,

140 DATE(transaction.timestamp) AS data_transaction,

141 transaction.amount

142 FROM company

143 JOIN transaction

144 ON transaction.company_id = company.id

145 WHERE transaction.amount BETWEEN 350 AND 400

146 AND DATE(transaction.timestamp) IN ('2015-04-29', '2018-07-20', '2024-03-13')

147 ORDER BY transaction.amount DESC;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	id	company_name	phone	country	data_transaction	amount
▶	b-2566	Aliquam PC	01 45 73 52 16	Germany	2024-03-13	399.84
	b-2294	Auctor Mauris Vel LLP	08 09 28 74 14	United States	2018-07-20	399.51
	b-2402	At Pedo Corp.	06 14 48 33 15	Italy	2015-04-29	390.69
	b-2566	Aliquam PC	01 45 73 52 16	Germany	2024-03-13	388.29
	b-2574	Orci Adipiscing Limited	03 18 00 77 81	United Kingdom	2018-07-20	373.71

Result 31 ×

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 53	20:55:08	SELECT company.id AS company_id, comp...	8 row(s) returned	0.047 sec / 0.000 sec
✓ 54	20:59:22	SELECT company.id, company.company_n...	8 row(s) returned	0.031 sec / 0.000 sec

Exercici 2

Necessitem optimitzar l'assignació dels recursos i dependrà de la capacitat operativa que es requereixi, per la qual cosa et demanem la informació sobre la quantitat de transaccions que realitzen les empreses, però el departament de recursos humans és exigent i vol un llistat de les empreses on especifiquis si tenen més de 400 transaccions o menys.

```
SELECT
  company.id,
  company.company_name,
  company.country,
  COUNT(transaction.id) AS quantitat_transaccions,
  CASE
    WHEN COUNT(transaction.id) > 400 THEN 'Més de 400 transaccions'
    ELSE 'Menys de 400 transaccions'
  END AS capacitat_operativa
FROM company
LEFT JOIN transaction
  ON transaction.company_id = company.id
GROUP BY company.id, company.company_name, company.country
ORDER BY quantitat_transaccions DESC;
```

```
149 SELECT
150     company.id,
151     company.company_name,
152     company.country,
153     COUNT(transaction.id) AS quantitat_transaccions,
154     CASE
155         WHEN COUNT(transaction.id) > 400 THEN 'Més de 400 transaccions'
156         ELSE 'Menys de 400 transaccions'
157     END AS capacitat_operativa
158 FROM company
159 LEFT JOIN transaction
160     ON transaction.company_id = company.id
161 GROUP BY company.id, company.company_name, company.country
162 ORDER BY quantitat_transaccions DESC;
```

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

	id	company_name	country	quantitat_transaccions	capacitat_operativa
▶	b-2222	Ac Fermentum Incorporated	Germany	2401	Més de 400 transaccions
	b-2302	Nunc Interdum Incorporated	Germany	1599	Més de 400 transaccions
	b-2330	Donec Fringilla PC	France	1593	Més de 400 transaccions
	b-2366	Mauris Institute	Sweden	1586	Més de 400 transaccions
	b-2614	Rutrum Non Inc.	Germany	1585	Més de 400 transaccions

Result 29 x

Read Only

Output

Action Output

	#	Time	Action	Message	Duration / Fetch
✖	51	20:50:37	SELECT company.id, company.company_n...	Error Code: 1054. Unknown column 'company.co...	0.000 sec
✔	52	20:51:38	SELECT company.id, company.company_n...	100 row(s) returned	0.109 sec / 0.000 sec

→ En este caso, han habido dos novedades:

- 1) El uso de CASE para crear una nueva columna en base a condicionales: Si el valor era superior a 400, entonces obtenemos "Más de 400 transacciones", mientras que si el valor era inferior a 400 obtenemos "Menys de 400".
- 2) El uso de LEFT JOIN en lugar de JOIN, ya que este segundo no incluye por defecto los registros que no coinciden con una id de transacción, mientras que LEFT JOIN sí.