



SPRINT 2

Sergi Alcolea de la Gala

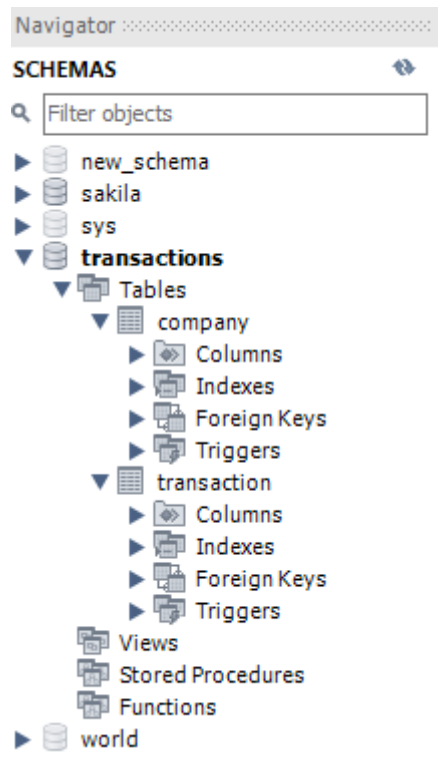
Nivell 1

Tasca S2.01. Nocions bàsiques SQL

Exercici 1

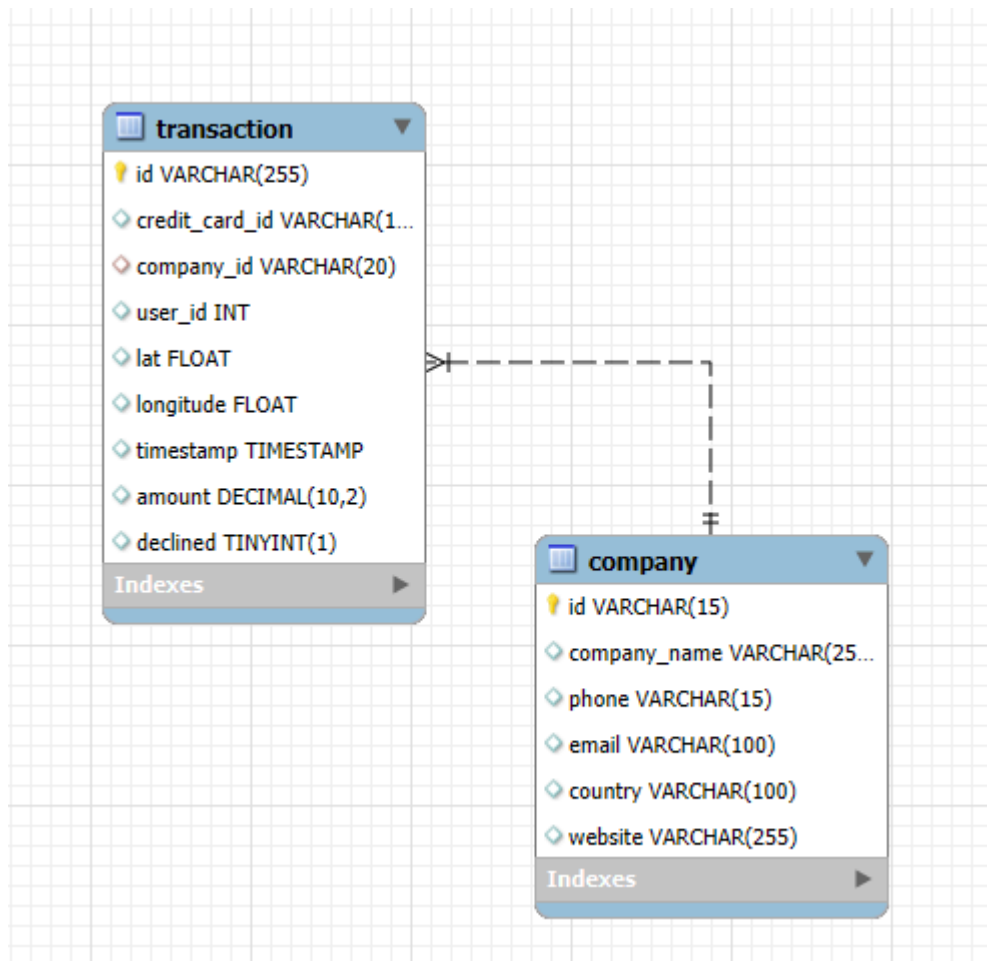
A partir dels documents adjunts (estructura_dades i dades_introduir), importa les dues taules. Mostra les característiques principals de l'esquema creat i explica les diferents taules i variables que existeixen. Assegura't d'incloure un diagrama que il·lustri la relació entre les diferents taules i variables.

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 “company” ejerce el rol de tabla maestra porque puede estar asociada a muchas transacciones (transaction), pero cada transacción solo puede estar vinculada a una sola empresa.

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.

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

```
SELECT DISTINCT company.country
FROM company
JOIN transaction
  ON company.id =
transaction.company_id;
```

The screenshot shows a database query editor with the following SQL query:

```
1 • use transactions;
2 • SELECT DISTINCT company.country
3 FROM company
4 JOIN transaction
5 ON company.id = transaction.company_id;
```

The results grid displays the following countries:

country
Germany
Australia
United States
New Zealand
Norway
United Kingdom
Italy
Belgium

The action output log shows the following messages:

#	Time	Action	Message
100142	01:22:50	use transactions	0 row(s) affected
100143	01:22:50	select distinct company.country from company join transactions on company.id = company_...	Error Code: 1146. Table 'transactions.transactions' doesn't exist
100144	01:23:18	use transactions	0 row(s) affected
100145	01:23:18	select distinct company.country from company join transactions on company.id = id LIMIT 0...	Error Code: 1146. Table 'transactions.transactions' doesn't exist
100146	01:25:15	use transactions	0 row(s) affected
100147	01:25:15	SELECT DISTINCT company.country FROM company JOIN transaction ON company.i...	15 row(s) returned

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

```
USE transactions;
SELECT count(DISTINCT company.country) AS total_paises
FROM company;
```

The screenshot shows a database query interface with a query editor and a results pane. The query editor contains the following SQL code:

```
1 • use transactions;
2 • select count(distinct company.country) as total_paises
3 • from company
```

The results pane shows a single row with the value 15 for the column total_paises.

Below the results pane, there is a section for the output of the query execution, showing a list of actions and their results:

#	Time	Action	Message
100148	01:40:25	use transactions	0 row(s) affected
100149	01:40:25	select count(distinct company.country) as total_paises from company LIMIT 0, 1000	1 row(s) returned
100150	01:41:29	use transactions	0 row(s) affected
100151	01:41:29	select count(distinct company.country) as total_paises from company join transaction on co...	1 row(s) returned
100152	01:43:14	use transactions	0 row(s) affected
100153	01:43:14	select count(distinct company.country) as total_paises from company LIMIT 0, 1000	1 row(s) returned

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)

```
use transactions;
SELECT company.company_name, AVG(transaction.amount) AS media_ventas
FROM company
JOIN transaction ON company.id = transaction.company_id
GROUP BY company.company_name
ORDER BY media_ventas DESC
LIMIT 10;
```

Query 1 | estructura dades | dades_introduir_sprint2

Limit to 1000 rows

```

3 FROM company
4 JOIN transaction ON company.id = transaction.company_id
5 GROUP BY company.company_name
6 ORDER BY media_ventas DESC
7 LIMIT 10;

```

Result Grid

company_name	media_ventas
Ac Fermentum Incorporated	284.867160
Pretium Neque Corp.	276.158330
Urna Convallis Associates	274.235011
At Associates	272.214870
Metus Vitae Associates	270.080965
Aliquet Diam Limited	269.599181
Nec Luctus LLC	268.604837
Neque Tellus Incorporated	267.850372

Result 23 x Read Only

Output

Action Output

#	Time	Action	Message
100175	02:02:16	use transactions	0 row(s) affected
100176	02:02:16	SELECT company.company_name, AVG(transaction.amount) AS media_ventas FROM co...	Error Code: 1140. In aggregated query without GROUP BY, expression #
100177	02:02:44	use transactions	0 row(s) affected
100178	02:02:44	SELECT company.company_name, AVG(transaction.amount) AS media_ventas FROM co...	1 row(s) returned
100179	02:03:06	use transactions	0 row(s) affected
100180	02:03:06	SELECT company.company_name, AVG(transaction.amount) AS media_ventas FROM co...	10 row(s) returned

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.

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

```
USE transactions;
```

```

SELECT *
FROM transaction
WHERE company_id IN (
  SELECT id
  FROM company
  WHERE country = 'Germany'
);

```


Sprint2* actor datos_introduir_sprint2 estructura datos

```

1 use transactions;
2 SELECT transaction.*, c.country
3 from transaction, (select country from company) as c
4 where company_id in (
5     select id
6     from company
7     where country = "Germany");

```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp
005E4E98-A828-4F18-91E0-C3E013CCCB72	CcS-6195	b-2614	1614	52.5205	13.4122	2022-12-21 15:
0071C15E-8C50-4F88-9EFC-2C6AABE023A4	CcS-4904	b-2614	323	52.157	5.50117	2022-11-21 14:
00764C4A-4B4A-41BD-9902-21C68F7AE5CC	CcS-8518	b-2614	3937	52.3602	19.6354	2024-06-16 16:
00788545-2AA4-4E43-BE66-88363358C42F	CcS-5427	b-2614	846	51.4243	10.6758	2019-05-14 17:
009CCA68-D5B3-4294-8F3A-8138F6CAFD47	CcS-5545	b-2614	964	41.4149	12.4344	2015-11-30 07:
00A5C4FC-D148-4EB6-8B26-454237053ACA	CcS-6077	b-2614	1496	39.2801	-7.78273	2015-08-13 18:
00B47DC9-7E76-4036-B620-A41F4FFCD34	CcS-5292	b-2614	711	52.3844	5.70921	2019-06-23 14:
00D71F56-4169-49FC-8307-06DE34A90040	CcS-9402	b-2614	4821	51.0535	10.8372	2023-02-19 06:
01236989-9DE3-4097-A0E7-D84FE441C7C	CcS-6385	b-2614	1804	51.4464	5.47292	2016-11-06 11:
012783D5-36CD-4905-99CC-C5B5E4FC8DA8	CcS-9295	b-2614	4714	51.9003	18.9053	2019-12-24 00:
012A2E01-E25E-4136-9ECC-7496D02589BE	CcS-5934	b-2614	1353	51.7551	19.3024	2015-03-05 12:
0144C2BA-B6AD-4822-8C49-A3EBB2964444	CcS-4884	b-2614	303	40.4217	-3.70959	2017-07-31 06:
01E5E3F3-AD78-4BCB-A175-3A6B7133F746	CcS-8745	b-2614	4164	55.0081	-3.10984	2024-11-30 01:
01FB28EB-4888-47A3-992E-6FF59D466CA9	CcS-5275	b-2614	694	41.9044	12.4934	2024-07-21 03:
022AF4DC-4AB8-42B2-92B8-F357EA297F3C	CcS-5938	b-2614	1357	42.1526	12.7291	2019-05-26 08:
026F1EDF-3E58-43D3-B508-A88529EC8483	CcS-6481	b-2614	1900	41.1523	-8.6251	2018-08-24 19:
02B901F7-F973-4283-9F02-3797FF4184F5	CcS-7606	b-2614	3025	52.1563	5.17908	2022-09-14 14:
02D03DEC-531B-4C57-A470-77B4AD1E584	CcS-7301	b-2614	2720	51.2909	10.0696	2019-05-28 07:
02E2504A-E5A5-4C5C-BF10-AB043EB504F0	CcS-6432	b-2614	1851	42.2	12.4494	2017-07-20 07:
034EB351-669A-4B50-BF15-EA70B6DE1748	CcS-4859	b-2614	278	60.2696	18.9937	2019-10-28 15:
03554744-1145-4D27-80C8-62F40DD40812	CcS-8145	b-2614	3564	51.7395	5.72365	2022-09-11 05:
036F38C6-8B3F-495E-B7AA-3FF446F43CCA	CcS-7548	b-2614	2967	39.4981	-8.25504	2020-09-15 12:
03835ACB-37A2-4332-99A8-3A03ADB7788	CcS-8176	b-2614	3595	51.7959	19.6035	2022-11-22 11:
038EED56-363B-4C46-BBD1-0BF3CBB3162B	CcS-6978	b-2614	2397	39.0388	-7.92221	2024-04-09 19:
03948ADA-6096-4025-B857-80AF9E8403A3	CcS-8243	b-2614	3662	55.2395	-3.76658	2016-01-16 10:

Result 6 x Read Only

#	Time	Action	Message	Duration / Fetch
28	10:49:32	SELE...	Error Code: 1242. Subquery returns more than 1 row	0.000 sec
29	10:50:25	use tr...	0 row(s) affected	0.000 sec
30	10:50:25	SELE...	Error Code: 1054. Unknown column 'c' in field list	0.000 sec
31	10:50:33	use tr...	0 row(s) affected	0.000 sec
32	10:50:33	SELE...	1329100 row(s) returned	0.015 sec / 6.047 sec

Para verificar que todas las transacciones sean de empresas que solo vienen de Alemania, podemos hacer una segunda subconsulta. Sin embargo, es tan rebuscado que si tenemos que verificar más de un campo, es más sencillo usar JOIN.

- 2) Segundo, 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`

```

SELECT DISTINCT company_name
FROM company
WHERE id IN ( SELECT company_id FROM transaction
              WHERE amount > ( SELECT AVG(amount) FROM transaction
                              )
              )

```


- 3) Debemos listar las empresas que no tienen transacciones antes de borrarla.
- a) 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.
 - b) añadimos distinct para eliminar la repetición de id de compañías.
 - c) 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, no existen empresas que no tienen transacciones y no podemos obtener los listados

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

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

```
22 • SELECT *  
23 FROM company  
24 WHERE NOT EXISTS (  
25     SELECT DISTINCT company_id  
26     FROM TRANSACTION  
27 );  
28  
29
```

Below the query editor, there is a toolbar with various icons for editing and exporting. Below the toolbar, a table titled "Result Grid" is displayed with the following columns: id, company_name, phone, email, country, website. The table shows a single row with all values as NULL.

At the bottom, there is an "Output" section with a tab labeled "Action Output". It shows a log of database actions:

#	Time	Action	Message	Duration / Fetch
✓ 1	13:03:19	select * from company where exists (select distinct...	100 row(s) returned	0.000 sec / 0.000 sec
✓ 2	13:03:49	select * from company where not exists (select dist...	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(timestamp) AS fecha,
  SUM(amount) AS total_ventas
FROM transaction
WHERE declined = 0
GROUP BY DATE(timestamp)
ORDER BY total_ventas DESC
LIMIT 5;
```

```
20 • SELECT
21     DATE(timestamp) AS fecha,
22     SUM(amount) AS total_ventas
23 FROM transaction
24 WHERE declined = 0
25 GROUP BY DATE(timestamp)
26 ORDER BY total_ventas DESC
27 LIMIT 5;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	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 9 ×

Output

Action Output

#	Time	Action
8	10:50:28	SELECT transaction.company_id, DATE(timestamp) AS fecha, SUM(transaction.a
9	10:51:18	SELECT * FROM transaction JOIN company ON company.id = company_id WHE

Exercici 2

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

```
SELECT country,  
       AVG(transaction.amount) 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 country  
ORDER BY mitjana_vendes DESC;
```

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
4 • SELECT country,  
5     AVG(transaction.amount) AS mitjana_vendes,  
6     COUNT(transaction.id) AS num_transaccions  
7 FROM company  
8 JOIN transaction  
9   ON transaction.company_id = company.id  
10 WHERE transaction.declined = 0  
11 GROUP BY country  
12 ORDER BY mitjana_vendes DESC;
```

The results pane displays a table with the following data:

country	mitjana_vendes	num_transaccions
Australia	265.535393	2544
United States	264.419466	4011
Belgium	260.971218	3515
Germany	260.829097	13269
Ireland	260.388751	2651
Spain	260.276923	1511
France	259.905738	4695
New Zealand	259.585048	2627

The results pane also shows an "Action Output" table with the following data:

#	Time	Action	Message	Duration / Fetch
35	18:03:15	SELECT company.company_name, transaction...	1000 row(s) returned	0.046 sec / 0.000 sec
36	18:08:03	SELECT country, AVG(transaction.amount) A...	15 row(s) returned	0.203 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(timestamp) AS data_transaccion,  
       amount,  
       declined  
FROM transaction  
JOIN company  
  ON company.id = company_id  
WHERE country = (  
  SELECT country  
  FROM company  
  WHERE company_name = 'Non Institute'  
)  
AND declined = 0
```

```
17 • SELECT *  
18 FROM transaction  
19 JOIN company  
20   ON company.id = company_id  
21 WHERE country = (  
22   SELECT country  
23   FROM company  
24   WHERE company_name = 'Non Institute'  
25 )  
26 AND declined = 0
```

Result Grid

Filter Rows:

Exports:

Wrap Cell Content:

Fetch rows:

	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-813F-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-C8DF80594ADD	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-CE2F-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
	03068E3B-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
	03478FE6-8EB5-4387-B187-0E78E8F288FB	CcS-7674	b-2246	3093	45.768	4.84271	2021-12-30 08:40:24	172.93	0	b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
	03AEBD0E-DC97-4BD3-9C57-6A6DB78026FD	CcS-6121	b-2246	1540	50.8113	10.3145	2018-11-11 11:28:49	114.77	0	b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
	03CA36D3-88FF-4D8F-BFD4-4CC7DA4EED2B	CcS-8036	b-2246	3455	52.5178	13.4131	2017-02-25 15:38:21	440.27	0	b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
	04494182-96DD-42EB-82FE-5F92C5210537	CcS-6791	b-2246	2210	41.9542	12.4607	2018-05-17 17:53:53	241.59	0	b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
	045AACF6-FF85-49FB-9DE4-E6730655366A	CcS-5363	b-2246	782	39.2464	-7.90454	2018-08-09 22:12:54	188.58	0	b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one

Result 6 ×

Output

Action Output

#	Time	Action	Message
8	10:50:28	SELECT transaction.company_id, DATE(timestamp) AS fecha, SUM(transaction.amount) AS total_ventas FROM transaction GROUP BY transaccio...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL servi...
9	10:51:18	SELECT * FROM transaction JOIN company ON company.id = company_id WHERE country = (SELECT country FROM company WHERE...	13754 row(s) returned

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

```
33 • SELECT
34     (SELECT company_name FROM company WHERE id = transaction.company_id) AS company_name,
35     transaction.id AS transaction_id,
36     DATE(timestamp) AS data_transaccion,
37     amount,
38     declined
39     FROM transaction
40 WHERE company_id IN (
41     SELECT id
42     FROM company
43     WHERE country = (
44         SELECT country
45         FROM company
46         WHERE company_name = 'Non Institute'
47     )
48 )
49 AND declined = 0;
```

company_name	transaction_id	data_transaccion	amount	declined
Sed Nunc Ltd	008629B4-C9A9-406C-A3D2-71FDA47BC546	2015-07-30	486.44	0
Sed Nunc Ltd	00B72BA4-54A3-4B8E-813F-2D57535AA17A	2017-10-26	414.06	0
Sed Nunc Ltd	01F075B1-D7AE-4D02-AAD9-5FFD72A43F3C	2018-01-27	103.73	0
Sed Nunc Ltd	023FFCE8-E618-4938-BF56-C8DF80540ADD	2016-12-19	219.28	0
Sed Nunc Ltd	02683BEB-EF91-4564-957B-D6F1662AB7C5	2017-01-10	326.87	0
Sed Nunc Ltd	02C2F29E-CEF2-4C1E-A594-F476E8F279C0	2020-05-24	155.72	0
Sed Nunc Ltd	02F468DC-426C-47C2-8B0A-D8B25B7A81AF	2023-03-17	305.35	0
Sed Nunc Ltd	0306BE3B-817B-4A49-934E-0E439291A104	2021-12-02	339.58	0
Sed Nunc Ltd	0347BFE6-8EB5-4387-B187-0E78E8F2B8FB	2021-12-30	172.93	0
Sed Nunc Ltd	03AEBD0E-DC97-4BD3-9C57-6A6D878026FD	2018-11-11	114.77	0

#	Time	Action	Message	Duration / Fetch
19	11:39:53	SELECT	(SELECT company_name FROM co...	0.000 sec / 0.110 sec
20	11:47:32	SELECT	(SELECT company_name FROM co...	0.000 sec / 0.094 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_name,  
       phone,  
       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;
```

The screenshot shows a SQL IDE interface with a query editor and a results panel. The query editor contains the following SQL code:

```
41 • SELECT company_name,  
42         phone,  
43         country,  
44         DATE(transaction.timestamp) AS data_transaction,  
45         transaction.amount  
46 FROM company  
47 JOIN transaction ON transaction.company_id = company.id  
48 WHERE transaction.amount BETWEEN 350 AND 400  
49        AND DATE(transaction.timestamp) IN ('2015-04-29', '2018-07-20', '2024-03-13')  
50 ORDER BY transaction.amount DESC;
```

The results panel displays a table with 6 columns: company_name, phone, country, data_transaction, and amount. The table contains 5 rows of data:

company_name	phone	country	data_transaction	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 Pede 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
Orci Adipiscing Limited	03 18 00 77 81	United Kingdom	2018-07-20	373.71

The results panel also shows an output log with the following entries:

#	Time	Action	Message	Duration / Fetch
42	18:46:55	SELECT company_...	Error Code: 1064. You have an error in your SQL syntax; check the manual that ...	0.000 sec
43	18:49:26	SELECT company_...	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
  c.company_name,
  c.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 company.id = transaction.company_id
GROUP BY company.company_name, company.country
ORDER BY quantitat_transaccions DESC;
```

The screenshot shows a SQL IDE interface with a query editor and a results grid. The query editor contains the following SQL code:

```
55 • SELECT
56     c.company_name,
57     c.country,
58     COUNT(transaction.id) AS quantitat_transaccions,
59     CASE
60         WHEN COUNT(transaction.id) > 400 THEN 'Més de 400 transaccions'
61         ELSE 'Menys de 400 transaccions'
62     END AS capacitat_operativa
63 FROM company
64 LEFT JOIN transaction ON company.id = transaction.company_id
65 GROUP BY company.company_name, company.country
66 ORDER BY quantitat_transaccions DESC;
67
```

The results grid shows the following data:

company_name	country	quantitat_transaccions	capacitat_operativa
Ac Fermentum Incorporated	Germany	2401	Més de 400 transaccions
Nunc Interdum Incorporated	Germany	1599	Més de 400 transaccions
Donec Fringilla PC	France	1593	Més de 400 transaccions
Mauris Institute	Sweden	1586	Més de 400 transaccions
Rutrum Non Inc.	Germany	1585	Més de 400 transaccions

The interface also shows a toolbar with various icons, a 'Limit to 1000 rows' dropdown, and a 'Result Grid' tab. The 'Output' section at the bottom shows the execution log:

#	Time	Action	Message
49	19:20:40	SELECT compa...	1000 row(s) returned
50	19:22:32	SELECT c.comp...	100 row(s) returned

→ 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 obteníamos “Más de 400 transacciones”, mientras que si el valor era inferior a 400 obteníamos “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 si.