

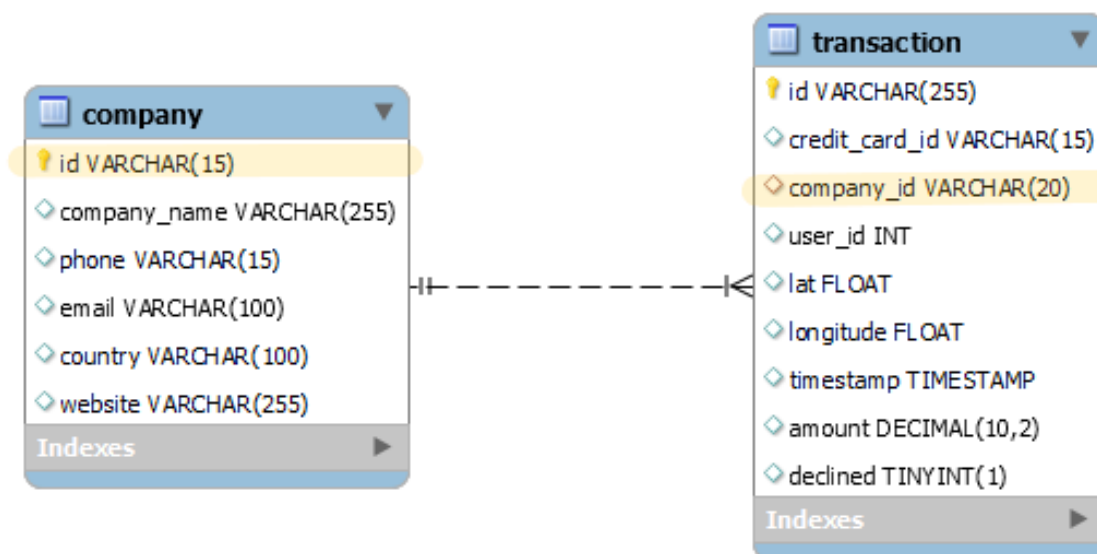
## Sprint 2 - Tasca S2.01. Nocions bàsiques SQL

### Nivell 1

#### - 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.

Lo primero que hice fue cargar los archivos al MySQL Workbench (en este orden "estructura\_dades" y "dades\_introduir") para crear la base de datos "transactions" que contiene dos tablas.



La tabla de dimensiones es "company", que contiene todos los datos correspondientes a las empresas como lo son el nombre, país de procedencia, página web, teléfono y también correo electrónico de contacto.

En la tabla de hechos "transaction" nos encontramos los registros correspondientes a las transacciones como lo son los identificadores de la transacción, las coordenadas geográficas de donde se realizaron, la hora y fecha, las cantidades de dinero e incluso si se realizó la transacción satisfactoriamente o si fue rechazada. También podemos encontrar en esta tabla de hechos referencias a identificadores de otras tablas como lo son el identificador de las tarjetas de crédito, de las compañías y de los usuarios. Estas referencias a otras tablas o mejor llamadas FOREIGN KEYs son las que hacen posibles la interacción entre distintas tablas gracias a las relaciones que se forman entre sus identificadores.

En la ilustración podemos ver que la columna "id" en la tabla "company" es su PRIMARY KEY, que a su vez funciona como la FOREIGN KEY de la tabla "transaction".

La relación entre tablas es de 1 a n. 1 de la tabla "company", a n en la tabla "transaction".

- **Exercici 2: Utilitzant JOIN realitzaràs les següents consultes:**

**2.1 Llistat dels països que estan fent compres.**

Para poder utilizar el JOIN como pide el enunciado solicité un SELECT DISTINCT de la columna “country” de la tabla “transaction”, pero como esa columna se encuentra en la tabla “company” usé un JOIN para juntar los datos de ambas tablas y así poder mostrar los resultados.

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

The screenshot shows a database management tool interface. At the top, there are tabs for 'estructura\_dades', 'dades\_introduir', and 'Sprint\_2 - Correcciones - Abelar...'. Below the tabs is a toolbar with various icons. The main area displays a SQL query:

```
1 # Nivell 1
2 ## Exercici 2: Utilitzant JOIN realitzaràs les següents consultes
3 ### Llistat dels països que estan fent compres.
4 • SELECT DISTINCT country
5   FROM transaction AS t
6   JOIN company AS c ON c.id = t.company_id;
```

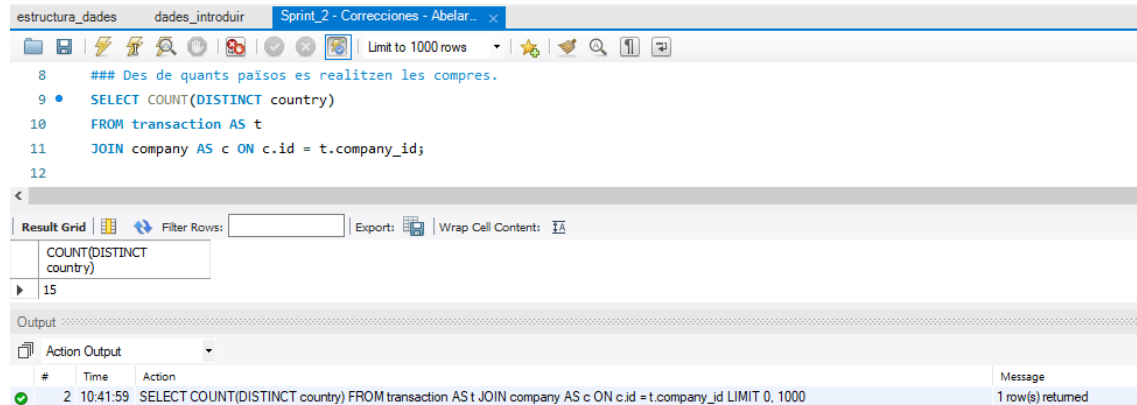
Below the query, there is a 'Result Grid' section. It shows a list of countries: Germany, Australia, United States, New Zealand, Norway, United Kingdom, Italy, Belgium, Sweden, Ireland, China, Canada, France, Netherlands, and Spain. The 'Output' section at the bottom shows the execution results:

#	Time	Action	Message
1	10:38:56	SELECT DISTINCT country FROM transaction AS t JOIN company AS c ON c.id = t.company_id LIMIT 0, 1000	15 row(s) returned

## 2.2 Des de quants països es realitzen les compres.

Aquí como en la consulta anterior podría haber hecho el COUNT() directamente en la tabla "company", pero solicité el dato de la tabla "transaction" para justificar el JOIN. Solicité un DISTINCT dentro del COUNT() para que me mostrara un conteo de los países por valores únicos y que así no repitiera países para el conteo.

```
SELECT COUNT(DISTINCT country)
FROM transaction AS t
JOIN company AS c ON c.id = t.company_id;
```



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

```
8  ### Des de quants països es realitzen les compres.
9  • SELECT COUNT(DISTINCT country)
10 FROM transaction AS t
11 JOIN company AS c ON c.id = t.company_id;
12
```

The results pane shows the output of the query. The first row of the result set is:

Result Grid
COUNT(DISTINCT country)
15

The output section shows the execution details:

#	Time	Action	Message
2	10:41:59	SELECT COUNT(DISTINCT country) FROM transaction AS t JOIN company AS c ON c.id = t.company_id LIMIT 0, 1000	1 row(s) returned

### 2.3 Identifica la companyia amb la mitjana més gran de vendes.

Para este enunciado pedí en el SELECT la columna “company\_name” para saber el nombre de las compañías, y un AVG() de “amount” para saber el promedio de ventas de dichas compañías.

Como están en tablas diferentes utilicé un JOIN para poder disponer de los datos que necesito de cada tabla.

Estipulé declined = 0 para así solo tomar en consideración las transacciones que han sido efectuadas, o sea, las ventas realizadas.

Agrupé por “company\_name” para que tomara en cuenta todas las transacciones de cada compañía por aparte.

Ordené de forma descendente para saber la compañía con mayor ventas.

Y limité los valores mostrados a uno, para así identificar cual es la compañía con una media de ventas más elevado.

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

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

```
13  ### Identifica la companyia amb la mitjana més gran de vendes.
14  • SELECT c.company_name, ROUND(AVG(t.amount),2) AS average
15  FROM transaction AS t
16  JOIN company AS c ON c.id = t.company_id
17  WHERE declined = 0
18  GROUP BY company_name
19  ORDER BY AVG(amount) DESC
20  LIMIT 1;
21
```

The results pane shows a single row of data:

company_name	average
Eget Ipsum Ltd	481.86

The output pane shows the following message:

```
3 10:45:23 SELECT c.company_name, ROUND(AVG(t.amount),2) AS average FROM transaction AS t JOIN company AS c ON c.id = t.company_id WHERE declin... 1 row(s) returned
```

- **Exercici 3: Utilitzant només subconsultes (sense utilitzar JOIN):**

**3.1 Mostra totes les transaccions realitzades per empreses d'Alemanya.**

Primero filtré los resultados de la tabla company por “Germany” para crear la subQuery y así utilizarla como una especie de filtro.

Después en la Query principal solicité todos los datos de la tabla transaction y coloqué la subQuery en el WHERE para filtrar por el company\_id de los países pertenecientes a Alemania que obtuve como resultado de la subQuery. (Usé el company\_id porque esta columna que se encuentra en ambas tablas).

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

estructura\_dades

dades\_introduir

Sprint\_2 - Correcciones - Abelar...

Limit to 1000 rows

```
30 -- Muestra todas las transacciones realizadas por empresas de Alemania.
31 SELECT *
32 FROM transaction
33 WHERE company_id IN
34     (SELECT id
35      FROM company
36      WHERE country = "Germany");
37
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A05DD	CdU-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57	0
EA2C3281-C9C1-A387-44F8-729F84B51C76	CdU-2938	b-2222	275	20.2004	-116.84	2021-05-09 10:25:08	119.36	1
0DD2E608-5C9E-D1B3-4999-B99F43AD735A	CdU-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47	1
AB069F53-965E-A2A8-CE06-CA8C4FD92501	CdU-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99	0
0466A42E-47CF-8D24-FD01-C0B689713128	CdU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53	0
0A476ED9-0C13-1962-F87B-D3563924B539	CdU-4359	b-2302	221	-56.4901	114.801	2022-02-26 20:33:54	430.49	0
122DC333-E19F-D629-DCD8-9C54CF1EB89A	CdU-4366	b-2302	221	29.6372	-166.173	2021-06-09 06:04:14	172.01	0
135267BA-2E7D-957C-C42C-6450A2B3ED54	CdU-4520	b-2302	210	20.6724	14.9732	2021-12-29 20:38:23	17.97	0
14CAE5B5-8FB1-3E4A-4C85-0EA4167534F4	CdU-4849	b-2302	189	-53.6202	93.0533	2021-12-31 00:29:42	388.04	0
158A3ACB-541C-DBCC-65BD-6373CC67BF1C	CdU-4849	b-2302	183	42.5424	-170.347	2022-03-08 05:02:19	240.29	0
162C7E78-2B6B-7971-A1E4-D2124E732451	CdU-4527	b-2302	210	-69.1381	58.0017	2021-04-11 05:59:18	231.26	0
1717FD6B-ADAD-7082-A748-9112BE892CCC	CdU-4219	b-2302	172	69.4892	-138.411	2021-12-29 16:18:54	249.91	0
1753A288-9FC1-52E6-5C39-A1FFB9780D3A	CdU-4345	b-2302	222	57.9422	-114.729	2021-08-17 05:32:08	497.84	0
186F53DE-DE27-B1FE-882F-15B61CEB7726	CdU-4310	b-2302	225	-72.7448	36.6211	2021-12-20 13:13:45	238.16	0
18C4E2DD-1E4C-F35E-2198-C6608D81DC25	CdU-4849	b-2302	177	68.0133	91.4839	2021-09-24 18:55:25	237.04	0
18CCBA7C-ABC1-813D-FAF3-4B8B97429368	CdU-4219	b-2302	173	51.3881	-156.371	2021-06-21 03:21:34	58.16	0
19E1EC3E-2119-1EFD-8AAE-5930D4A4E63F	CdU-4219	b-2302	154	-56.0839	116.987	2021-07-06 21:40:15	29.63	0
1B117D49-936C-BA6C-E94B-30C3293AA239	CdU-4219	b-2302	163	-75.099	109.034	2021-05-09 23:56:04	371.35	0
1B521826-5860-5A86-5364-6EB6A5CC21B7	CdU-4226	b-2302	231	-53.4613	49.1484	2021-08-14 02:39:50	476.33	0
1C3B3CC4-B919-7616-6A57-EEBDD7B7B68B	CdU-4415	b-2302	217	81.1737	62.8835	2021-09-27 16:07:34	462.35	0
2075B12D-69AB-7022-57BB-33ED49B72766	CdU-4247	b-2302	229	-9.03682	88.9005	2022-02-13 04:07:29	87.44	0
241827A6-E87C-6DA3-5AC9-BB8CFD42B7D2A	CdU-4849	b-2302	190	-12.1172	-175.322	2021-08-04 04:44:35	487.64	0
2585891B-C4C7-075E-358B-8575339E5177	CdU-4450	b-2302	215	32.9171	-40.6059	2021-08-24 05:32:36	253.49	0
263035AA-8758-E413-8CC4-6D8F81B8F899	CdU-4338	b-2302	223	36.61	83.1953	2021-06-21 11:23:01	499.23	0
2C9A4725-4B19-621C-4EE4-AE9E31CFE5B0	CdU-4219	b-2302	149	-46.7357	162.102	2021-08-14 09:23:34	28.95	0
33525F74-4864-C19F-7D23-A8D78668B7C4	CdU-4492	b-2302	212	10.3105	13.9999	2021-12-15 12:22:03	423.71	0
34368DE0-D181-1B5C-6698-8DF585C06421	CdU-4219	b-2302	153	-8.20452	89.9882	2021-07-05 10:54:54	86.69	0
34AA1D64-2989-DE1C-8705-C66C8DD819DD	CdU-4219	b-2302	166	-17.0772	-169.006	2022-03-02 05:30:41	275.45	0
369213C7-BD65-0715-6644-D5E4F9BD9914	CdU-4219	b-2302	147	88.7859	-148.023	2021-12-29 18:54:29	458.66	0
3A3E00C1-C191-02EF-AD8E-42C9DA644E4A	CdU-4849	b-2302	175	0.340013	174.033	2021-11-10 19:14:50	451.66	0
3E495EC1-E1C5-26F1-0E3E-2D13EAA2593A	CdU-4282	b-2302	227	-50.5725	-172.508	2022-03-03 14:29:15	417.70	0
41932C77-45B5-C786-6B3F-0BB619B2D535	CdU-4219	b-2302	142	-8.37247	-67.8776	2022-02-12 09:12:05	358.36	0
442713DA-4BE6-132A-51B8-AC44EA6D5839	CdU-4436	b-2302	216	-22.3294	-22.0184	2022-01-29 17:12:59	59.67	0
5218E51C-E258-BFA4-BE23-AF815F66FAF8	CdU-4219	b-2302	157	18.9732	66.3985	2021-09-07 21:38:45	234.52	0

Output

Action Output

#TimeAction

410:51:52SELECT \* FROM transaction WHERE company\_id IN (SELECT id FROM company WHERE country = "Germany") LIMIT 0, 1000

118 row(s) returned

Message

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

Lo primero que hice fue crear una subQuery que me devolviera el valor de la media de amount.

Lo siguiente fue buscar los identificadores de las empresas que hayan tenido transacciones con un gasto mayor a la media.

Y lo último que hice fue usar los datos que obtuve anteriormente como un filtro, para poder ver un listado de las empresas que han realizado transacciones por un amount mayor a la media de todas las transacciones.

```
SELECT *
FROM company
WHERE id IN
    (SELECT DISTINCT company_id
     FROM transaction
     WHERE amount >
      (SELECT AVG(amount)
       FROM transaction));
```

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

```
-- Para mostrar la lista de las empresas con una gasto superior a la media.
SELECT *
FROM company
WHERE id IN
    (SELECT DISTINCT company_id
     FROM transaction
     WHERE amount >
      (SELECT AVG(amount)
       FROM transaction));
```

The result grid displays the following data:

id	company_name	phone	email	country	website
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net	Germany	https://instagram.com/site
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@idcloud.org	Australia	https://whatsapp.com/group/9
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/sub/cars
b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings
b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.co.uk	Norway	https://nytimes.com/user/110
b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
b-2254	Nascetur Ridiculus Mus Inc.	06 26 87 61 84	suspendisse.dui@idcloud.net	United States	https://ebay.com/sub
b-2258	Vestibulum Lorem PC	02 02 87 33 40	aenean.massa.integer@aol.net	Belgium	https://pinterest.com/sub/cars
b-2262	Gravida Sagittis LLP	03 81 28 33 97	turpis.vitae@google.ca	Sweden	https://naver.com/site
b-2266	Mus Aenean Eget Foundation	06 25 15 52 43	mi.duis@hotmail.net	Sweden	https://instagram.com/group/9
b-2270	Dis Parturient Institute	05 36 29 78 74	purus@protonmail.org	Ireland	https://google.com/one
b-2274	Sed LLC	01 63 16 26 52	at@outlook.com	Belgium	https://reddit.com/fr
b-2278	Arcu LLP	06 46 04 41 45	dui@aol.ca	Norway	https://yahoo.com/sub
b-2286	Fringilla LLC	08 29 15 93 57	gravida@protonmail.co.uk	New Zealand	https://reddit.com/user/110
b-2298	Elit Etiam Laoreet Associates	07 69 74 17 45	ultrices@google.co.uk	Canada	https://yahoo.com/fr
b-2302	Nunc Interdum Incorporated	05 18 15 48 13	non@outlook.com	Germany	https://wikipedia.org/en-us
b-2306	Augue Foundation	06 88 43 15 63	mauris@yahoo.com	Germany	https://baidu.com/sub/cars
b-2310	Non Magna LLC	06 71 73 13 17	nisl.quisque.fringilla@hotmail.ca	United Kingdom	https://whatsapp.com/site
b-2314	A Institute	03 34 91 68 65	metus.aliquam@google.edu	Belgium	https://reddit.com/fr
b-2318	Quam A Felis Industries	04 87 10 27 16	proin.velit@idcloud.edu	Italy	https://ebay.com/settings
b-2322	Integer Mollis Corp.	03 12 20 45 24	eu.eros@protonmail.ca	Italy	https://netflix.com/group/9
b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yahoo...	United Kingdom	https://cnn.com/group/9
b-2330	Donec Fringilla PC	01 51 58 14 44	ut.tincidunt@hotmail.ca	France	https://google.com/fr
b-2334	Amet Institute	06 33 40 21 33	nullam.lobortis.quam@outlook.net	Australia	https://nytimes.com/one
b-2338	Magna Incorporated	07 73 69 44 18	non@google.net	New Zealand	https://twitter.com/en-us
b-2342	Fringilla Porttitor Incorporated	06 30 30 97 81	ultrices.a.auctor@protonmail.co.uk	Sweden	https://wikipedia.org/settings
b-2346	Ut Semper Foundation	01 60 36 33 06	consectetuer@google.ca	Sweden	https://ebay.com/settings
b-2358	Ac Industries	09 34 65 40 60	ipsum@yahoo.com	Germany	https://bbc.co.uk/group/9
b-2362	Lorem Eu Incorporated	01 83 66 62 07	enim.gravida.sit@hotmail.net	Canada	https://baidu.com/en-ca
b-2366	Mauris Institute	05 29 60 36 87	morbi.vehicula@outlook.co.uk	Sweden	https://ebay.com/sub/cars
b-2378	Erat LLP	03 18 88 77 79	non.vestibulum@protonmail.net	Netherlands	https://ebay.com/fr
b-2382	Neque Tellus Incorporated	04 43 18 34 19	ultrices.iaculis.odio@aol.ca	Ireland	https://google.com/site

The output section shows the following message:

```
5 10:56:22 SELECT * FROM company WHERE id IN (SELECT DISTINCT company_id FROM transaction WHERE amount > (SELECT AVG(amount) FROM tran... 70 row(s) returned
```

### 3.3 Eliminaran del sistema les empreses que no tenen transaccions registrades, entrega el llistat d'aquestes empreses.

Para esta ocasión lo primero que hice fue una subQuery que me diera el identificador de todas las empresas en la tabla “transactions”. Luego con ese resultado hice una Query con el nombre de las empresas de la tabla “company”, con el condicional de que aparecieran solamente las empresas que no existen en el resultado de la subQuery. No me devuelve nada como resultado porque todas las empresas en la tabla company tienen transacciones registradas.

```
SELECT *
FROM company AS c
WHERE NOT EXISTS
    (SELECT company_id
     FROM transaction AS t
     WHERE c.id = t.company_id);
```

The screenshot shows a database management tool interface. The top bar includes tabs for 'estructura\_dades', 'dades\_introduir', and 'Sprint\_2 - Correcciones - Abelar...'. Below the tabs is a toolbar with various icons and a 'Limit to 1000 rows' dropdown. The main area displays a SQL query with line numbers 60 to 68. The query is a SELECT statement with a subquery in the WHERE clause. Below the query editor is a 'Result Grid' showing a table with columns: id, company\_name, phone, email, country, and website. The table contains one row with all values set to NULL. At the bottom, there is an 'Output' section with a table showing the execution details. The table has columns: #, Time, Action, and Message. The first row shows that the query was executed at 11:03:54 and returned 0 rows.

```
60  ### Eliminaran del sistema les empreses que no tenen transaccions registrades, entrega el llistat d'aquestes empreses.
61  -- De esta manera puedo ver las empresas que no tienen transacciones registradas
62  • SELECT *
63  FROM company AS c
64  WHERE NOT EXISTS
65  (SELECT company_id
66   FROM transaction AS t
67   WHERE c.id = t.company_id);
68
```

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

#	Time	Action	Message
6	11:03:54	SELECT * FROM company AS c WHERE NOT EXISTS (SELECT company_id FROM transaction AS t WHERE c.id = t.company_id) LIMIT 0, 1000	0 row(s) returned

## Nivell 2

### - Exercici 1

**1.1 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.**

Lo primero que consideré fue hacer un SUM() de la columna amount, agrupándolo por las fechas de la columna timestamp, como vi que en timestamp prácticamente todos los registros son valores únicos entonces utilicé la función DATE() para poder agrupar los valores solo por las fechas. Y fijé la condición WHERE declined = 0 para tener los resultados solo por transacciones efectuadas.

Finalmente ordené por el resultado de SUM(amount) de forma descendente para así poder ver los resultados más grandes primero, y lo limité a los 5 primeros valores para que muestre cuales fueron los 5 días que se generaron mayores ingresos.

```
SELECT SUM(amount) AS income, DATE(timestamp) AS date
FROM transaction
WHERE declined = 0
GROUP BY DATE(timestamp)
ORDER BY SUM(amount) DESC
LIMIT 5;
```

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

```
70 # Nivell 2
71 ## Exercici 1
72 ### Identifica els cinc dies que es va generar la quantitat més gran d'ingressos a l'empresa per vendes.
73 ### Mostra la data de cada transacció juntament amb el total de les vendes.
74 • SELECT SUM(amount) AS income, DATE(timestamp) AS date
75 FROM transaction
76 WHERE declined = 0
77 GROUP BY DATE(timestamp)
78 ORDER BY SUM(amount) DESC
79 LIMIT 5;
80
```

The results pane displays a table with two columns: 'income' and 'date'. The data is as follows:

income	date
1532.36	2021-12-20
1397.96	2021-04-22
1344.37	2021-05-09
1337.62	2022-02-26
1325.12	2021-03-29

The output pane shows the execution of the query, indicating that 5 rows were returned.



## - Exercici 2

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

Para mostrar el promedio de ventas por país primero necesitaría el promedio de amount y la lista de los países, como estas columnas están en dos tablas distintas utilicé un JOIN para poder acceder a las dos columnas, marqué la condición WHERE declined = 0 para me mostrara solamente las transacciones aceptadas (ventas), agrupé los promedios por país y los ordené de forma descendente (de mayor a menor).

```
SELECT country, AVG(amount)
FROM transaction AS t
JOIN company AS c ON c.id = t.company_id
WHERE declined = 0
GROUP BY country
ORDER BY AVG(amount) DESC;
```

The screenshot shows a database interface with a SQL query editor and a results grid. The query is as follows:

```
81  ## Exercici 2
82  ### Quina és la mitjana de vendes per país? Presenta els resultats ordenats de major a menor mitjà.
83  • SELECT country, AVG(amount)
84  FROM transaction AS t
85  JOIN company AS c ON c.id = t.company_id
86  WHERE declined = 0
87  GROUP BY country
88  ORDER BY AVG(amount) DESC;
89
```

The results grid displays the following data:

country	AVG(amount)
United States	287.531111
Ireland	285.825357
Sweden	276.668382
United Kingdom	271.767527
Canada	261.941930
Belgium	255.217500
Norway	251.114918
Italy	243.342222
Germany	242.239189
Netherlands	240.940000
China	222.240000
Australia	177.331667
France	169.410000
New Zealand	167.061667
Spain	26.220000

The Output section shows the following message:

```
8  11:12:46  SELECT country, AVG(amount) FROM transaction AS t JOIN company AS c ON c.id = t.company_id WHERE declined = 0 GROUP BY country ORDE... 15 row(s) returned
```

## - Exercici 3

**3.1 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.**

### 3.1.1 Mostra el llistat aplicant JOIN i subconsultes.

Lo primero que busqué en este caso fue el país en el que se encuentra la compañía "Non Institute", una vez obtuve el resultado lo utilicé como filtro dentro de una subQuery, para así mostrar todas las transacciones de empresas que estén ubicadas en el mismo país que "Non Institute", procurando excluir a "Non Institute" del resultado final.

```
SELECT t.*
FROM transaction AS t JOIN company AS c ON c.id = t.company_id
WHERE company_name != "Non Institute" AND country =
    (SELECT country
     FROM company
     WHERE company_name = "Non Institute");
```

estructura\_dades dades\_introduir Sprint\_2 - Correcciones - Abelar...

98 -- #1 Mostra el llistat aplicant JOIN i subconsultes.

99 • SELECT t.\*

100 FROM transaction AS t JOIN company AS c ON c.id = t.company\_id

101 WHERE company\_name != "Non Institute" AND country =

102 (SELECT country

103 FROM company

104 WHERE company\_name = "Non Institute");

Result Grid Filter Rows: Export: Wrap Cell Contents: I A

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶	2B928E1C-EC14-A760-0A75-871477649D6A	CdJ-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14:49	383.73	0
	ACD2011A-A2B1-C365-41E1-2AB00C65147A	CdJ-2980	b-2246	275	-54.4792	-82.7974	2022-03-05 20:41:20	60.07	1
	4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	CdJ-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37:23	458.74	0
	BC2B9A38-7B4-28CD-1FE8-14DEC863E773	CdJ-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95	1
	147983D2-B7BA-C7BB-4CE3-8D7C2DE85AB8	CdJ-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58:07	309.45	0
	152598C2-029D-D684-4B66-91EDF393EBFF	CdJ-2994	b-2326	126	-67.0189	-141.672	2021-07-05 03:10:00	395.43	0
	1B636B58-A2E8-7C69-D9C9-C54535DAFD3B	CdJ-2994	b-2326	131	70.2543	-13.1336	2021-07-06 08:48:46	195.06	0
	20418DE5-B804-BE9B-8D7A-A95C1BFD8F5C	CdJ-2994	b-2326	126	-79.1145	1.51481	2022-01-03 15:59:29	479.52	0
	239B8576-6C0E-137A-C2F6-3180A188A2D3	CdJ-2994	b-2326	126	23.6174	137.222	2021-08-26 06:04:05	43.90	0
	267C4A86-7BA7-1C5E-0718-2824983C87DD	CdJ-2994	b-2326	126	-17.5259	104.915	2021-10-01 21:08:53	122.63	0
	3142C93E-B3B7-49E4-EE2D-29CA834B198D	CdJ-2994	b-2326	126	-67.8476	-119.978	2021-04-06 17:24:44	91.59	0
	3578688E-7B1D-B887-3BC7-20B8673AA31E	CdJ-2994	b-2326	126	87.0665	-22.7339	2021-07-26 22:59:24	303.60	0
	360C7814-F7AF-B43A-0946-AB38D2683C86	CdJ-2994	b-2326	116	-7.93005	-79.0733	2021-08-21 10:19:58	494.82	0
	391E1CFD-D653-E45B-A729-F2EB93247B58	CdJ-2994	b-2326	118	60.5512	103.904	2021-10-09 00:50:38	271.27	0
	3C4D7C2A-A402-B941-625A-D64CA53526E8	CdJ-2994	b-2326	117	23.1627	-10.381	2021-04-25 19:11:52	441.27	0
	3ED634C2-01CA-D6E9-2DBD-24B53A7912F7	CdJ-4219	b-2326	137	69.0549	96.5033	2021-03-29 11:38:38	478.54	0
	3FEC4131-9BED-2EA6-1776-82674C80DD8B	CdJ-2994	b-2326	126	-7.28469	179.063	2021-07-25 04:59:17	393.42	0
	41C13F08-471E-D475-3E14-76D0F3F79B42	CdJ-4219	b-2326	138	-51.7289	55.3294	2022-01-16 16:48:55	45.63	0
	41EC591D-808E-D85C-F05F-088C3DE7A618	CdJ-2994	b-2326	126	5.38535	-78.3917	2022-01-06 07:47:39	335.54	0
	4217E857-1D51-5D25-B645-EC24D23C8759	CdJ-2994	b-2326	129	65.4733	-20.2735	2021-04-23 13:07:58	193.33	0
	45CA3E45-6B42-CAC4-7683-EBB69D523B3E	CdJ-4219	b-2326	140	85.7055	36.1497	2021-04-29 06:17:02	149.89	0
	49CD105D-E569-3733-C8C8-3908ED7E1838	CdJ-3393	b-2326	108	-47.8783	168.275	2021-12-24 23:43:59	411.81	0
	4ABE3D77-BF86-12D3-E1C5-22663D37C199	CdJ-3120	b-2326	273	-63.6087	69.6023	2021-11-13 04:18:28	38.02	1
	4B74B94B-D123-7C7E-A9CE-7DA26F911F55	CdJ-2994	b-2326	128	25.6024	53.4543	2022-02-12 14:27:28	304.43	0
	4D8E4891-1711-7D30-E4CB-645A8DF59A43	CdJ-2994	b-2326	120	5.71915	-155.369	2021-12-26 01:49:19	365.83	0
	4E562A2B-99E9-8C82-2340-86623BC9FE2F	CdJ-3120	b-2326	273	-65.8033	158.1	2021-03-23 15:43:37	409.68	0
	5F47388E-B392-A2E2-1E8E-C4A812C6A95C	CdJ-2994	b-2326	126	-57.1143	-154.767	2022-01-10 05:12:55	253.68	0
	61EC87A4-CF88-B392-F83B-75E7F2C93537	CdJ-3393	b-2326	110	88.4626	85.6351	2021-06-01 20:23:57	346.10	0
	634E63C6-C2D6-EC48-4CCF-6B4A60AAEEB7	CdJ-2994	b-2326	125	-77.8923	-167.683	2021-09-17 15:02:25	399.71	0
	65D82765-E85D-2B74-2BDA-A8AEFD248928	CdJ-2994	b-2326	126	-19.9162	2.61475	2021-12-08 09:25:11	140.67	0
	71CB87BE-F93E-4877-DF6D-1BD3B312ED29	CdJ-3393	b-2326	110	-27.1212	-105.573	2021-10-21 20:41:56	441.63	0

Output

Action Output

#	Time	Action	Message
✓ 10	11:20:49	SELECT t.* FROM transaction AS t JOIN company AS c ON c.id = t.company_id WHERE company_name != "Non Institute" AND country = (SELECT...	70 row(s) returned

Para este caso lo primero que hice fue anidar una SubQuery dentro de otra para poder filtrar por país usando el nombre de "Non Institute". Lo siguiente fue filtrar los identificadores de las compañías en la otra SubQuery, procurando excluir a "Non Institute". Y finalmente mostré todos los datos de la tabla transaction cuando el identificador de las compañías esté dentro de los resultados obtenidos de las SubQueries.

estructura\_datos

datos\_introduir

Sprint2 - Correcciones - Abelar...

Limit to 1000 rows

```

106 -- #2 Muestra el llistat aplicant solament subconsultes.
107 SELECT *
108 FROM transaction
109 WHERE company_id IN
110 (SELECT id
111 FROM company
112 WHERE company_name != "Non Institute" AND country =
113 (SELECT country
114 FROM company
115 WHERE company_name = "Non Institute"));

```

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell Content: [F8](#)

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶	28928E1C-EC14-A760-0A75-871477649D6A	CcU-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14:49	383.73	0
	ACD2011A-A2B1-C365-41E1-2AB00C65147A	CcU-2980	b-2246	275	-54.4792	-82.7974	2022-03-05 20:41:20	60.07	1
	4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	CcU-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37:23	458.74	0
	BC2B9A38-778A-28CD-1FE8-14DED863E773	CcU-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95	1
	1479B3D2-87BA-C7B8-4CE3-8D7C2DE85A8B	CcU-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58:07	309.45	0
	152598C2-029D-D684-4B66-91EDF393EBFF	CcU-2994	b-2326	126	-67.0189	-141.672	2021-07-05 03:10:00	395.43	0
	1B636B58-A2E8-7369-D9C9-C5453DADF3BC	CcU-2994	b-2326	131	70.2543	-13.1336	2021-07-06 08:48:46	195.06	0
	20418DE5-8804-BE9B-BD7A-A95C1BFDBF5C	CcU-2994	b-2326	126	-79.1145	1.51481	2022-01-03 15:59:29	479.52	0
	239B8576-6C0E-137A-C2F6-3180A188A2D3	CcU-2994	b-2326	126	23.6174	137.222	2021-08-26 06:04:05	43.90	0
	267C4A86-7BA7-1C5E-0718-2824983C87DD	CcU-2994	b-2326	126	-17.5259	104.915	2021-10-01 21:08:53	122.63	0
	3142C93E-83B7-49E4-EE2D-29CA834B198D	CcU-2994	b-2326	126	-67.8476	-119.978	2021-04-06 17:24:44	91.59	0
	3578688E-7B1D-B887-3BC7-20B8673AA31E	CcU-2994	b-2326	126	87.0665	-22.7339	2021-07-26 22:59:24	303.60	0
	360C7814F7AF-843A-0946-AB38D2683C86	CcU-2994	b-2326	116	-7.93005	-79.0733	2021-08-21 10:19:58	494.82	0
	391E1CFD-D653-E45B-A729-F2E893247B58	CcU-2994	b-2326	118	60.5512	103.904	2021-10-09 00:50:38	271.27	0
	3C4D7C2A-A402-B941-625A-D64CA53526E8	CcU-2994	b-2326	117	23.1627	-10.381	2021-04-25 19:11:52	441.27	0
	3ED634C2-01CA-D6E9-2DBD-24B53A7912F7	CcU-4219	b-2326	137	69.0549	96.5033	2021-03-29 11:38:38	478.54	0
	3FEC4131-98ED-2EA6-1776-82674C80DD8B	CcU-2994	b-2326	126	-7.28469	179.063	2021-07-25 04:59:17	393.42	0
	41C13F08-471E-D475-3E14-760DF379B42	CcU-4219	b-2326	138	-51.7289	55.3294	2022-01-16 16:48:55	45.63	0
	41EC591D-808E-D85C-F05F-08BC3DE7A618	CcU-2994	b-2326	126	5.38535	-78.3917	2022-01-06 07:47:39	335.54	0
	4217E857-1D51-5D25-B645-EC24D23C8759	CcU-2994	b-2326	129	65.4733	-20.2735	2021-04-23 13:07:58	193.33	0
	45CA3E45-6842-CAC4-7683-EBB69D52383E	CcU-4219	b-2326	140	85.7055	36.1497	2021-04-29 06:17:02	149.89	0
	49CD105D-E569-3733-C8C8-3908ED7E1838	CcU-3393	b-2326	108	-47.8783	168.275	2021-12-24 23:43:59	411.81	0
	4ABE3D77-BF86-12D3-E1C5-22663D37C199	CcU-3120	b-2326	273	-63.6087	69.6023	2021-11-13 04:18:28	38.02	1
	4B74B94B-0123-7C7E-A9CE-7DA26F911F55	CcU-2994	b-2326	128	25.6024	53.4543	2022-02-12 14:27:28	304.43	0
	4D8E4891-1711-7D30-E4CB-645A8DF59A43	CcU-2994	b-2326	120	5.71915	-155.369	2021-12-26 01:49:19	365.83	0
	4E562A2B-99E9-8C82-2340-B6623BC9FE2F	CcU-3120	b-2326	273	-65.8033	158.1	2021-03-23 15:43:37	409.68	0
	5F4738BE-8392-A2E2-1E8E-C4A812C6A95C	CcU-2994	b-2326	126	-57.1143	-154.			

### Nivell 3

#### - Exercici 1

**1.1 Presenta el nom, telèfon, país, data i amount, d'aquelles empreses que van realitzar transaccions amb un valor comprès entre 100 i 200 euros i en alguna d'aquestes dates: 29 d'abril del 2021, 20 de juliol del 2021 i 13 de març del 2022. Ordena els resultats de major a menor quantitat.**

Primero que nada introduje la columna “timestamp” dentro de la función DATE() para obtener solamente las fechas de la columna, y así poder filtrar por las fechas específicas que pide el ejercicio.

Lo siguiente fue poner todos los campos que pide el ejercicio en el SELECT, procurando filtrar los resultados en el WHERE para que el valor de compra esté dentro del rango de 100 y 200 euros, y que sea de las fechas específicas que el ejercicio solicita, ordenando los resultados de mayor a menor valor de compra.

```
SELECT company_name, phone, country, DATE(timestamp) AS date, amount
FROM company AS c JOIN transaction AS t ON c.id = t.company_id
WHERE amount BETWEEN 100 AND 200 AND
DATE(timestamp) IN ('2021-04-29', '2021-07-20', '2022-03-13')
ORDER BY amount DESC;
```

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

```
118 # Nivell 3
119 ## Exercici 1
120 ### Presenta el nom, telèfon, país, data i amount, d'aquelles empreses que van realitzar transaccions amb un valor comprès entre 100 i 200 euros i en alguna d'aquestes dates:
121 ### 29 d'abril del 2021, 20 de juliol del 2021 i 13 de març del 2022. Ordena els resultats de major a menor quantitat.
122 -- Para filtrar por las fechas del ejercicio
123 • SELECT DATE(timestamp)
124 FROM transaction
125 WHERE DATE(timestamp) IN ('2021-04-29', '2021-07-20', '2022-03-13');
126
127 -- Para mostrar los resultados filtrados por fecha y cantidad
128 • SELECT company_name, phone, country, DATE(timestamp) AS date, amount
129 FROM company AS c JOIN transaction AS t ON c.id = t.company_id
130 WHERE amount BETWEEN 100 AND 200 AND
131 DATE(timestamp) IN ('2021-04-29', '2021-07-20', '2022-03-13')
132 ORDER BY amount DESC;
133
```

The results grid displays the following data:

company_name	phone	country	date	amount
Interdum Feugiat Sed Associates	04 88 40 32 52	United Kingdom	2021-07-20	164.86
Nunc Interdum Incorporated	05 18 15 48 13	Germany	2022-03-13	164.32
Enim Conditmentum Ltd	09 55 51 66 25	United Kingdom	2021-04-29	149.89
Lorem Eu Incorporated	01 83 66 62 07	Canada	2021-07-20	133.39
Nunc Interdum Incorporated	05 18 15 48 13	Germany	2021-04-29	111.51

The output section shows the execution of the query, indicating that 5 row(s) were returned.

## - Exercici 2

**2.1 Necessitem optimitzar l'assignació dels recursos i dependrà de la capacitat operativa que es requereixi, per la qual cosa et demanen 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 4 transaccions o menys.**

Lo primero que hice fue usar una SubQuery como tabla temporal para realizar el conteo de transacciones que tiene cada empresa, también procuré incluir el nombre e identificador de la las compañías para poder mostrarlos luego.

Lo siguiente fue mostrar los datos de las empresas junto a la cantidad de transacciones que tienen registradas. Y agregué un CASE para que me especificué que empresas tiene más, o menos de 4 transacciones.

```
SELECT company_id, company_name, transaction_count,
CASE
    WHEN transaction_count > 4 THEN "Más de 4 transacciones"
    ELSE "Menos de 4 transacciones"
END AS "¿Más? o, ¿menos?"
FROM
    (SELECT t.company_id, c.company_name, COUNT(t.id) AS transaction_count
    FROM transaction AS t
    JOIN company AS c ON c.id = t.company_id
    GROUP BY company_id) AS trans_count
ORDER BY transaction_count DESC;
```

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query is as follows:

```
135 -- Exercici 2
136 --- Necessitem optimitzar l'assignació dels recursos i dependrà de la capacitat operativa que es requereixi,
137 --- per la qual cosa et demanen la informació sobre la quantitat de transaccions que realitzen les empreses,
138 --- però el departament de recursos humans és exigent i vol un llistat de les empreses on especifiquis si tenen més de 4 transaccions o menys.
139 -- Creé una Query que usaré como tabla temporal para saber el número de transacciones que tiene cada empresa
140 • SELECT t.company_id, c.company_name, COUNT(t.id) AS transaction_count
141 FROM transaction AS t
142 JOIN company AS c ON c.id = t.company_id
143 GROUP BY company_id;
144
145 -- Para mostrar el listado de empresas que tienen más de 4 transacciones
146 • SELECT company_id, company_name, transaction_count,
147 CASE
148     WHEN transaction_count > 4 THEN "Más de 4 transacciones"
149     ELSE "Menos de 4 transacciones"
150 END AS "¿Más? o, ¿menos?"
151 FROM
152     (SELECT t.company_id, c.company_name, COUNT(t.id) AS transaction_count
153     FROM transaction AS t
154     JOIN company AS c ON c.id = t.company_id
155     GROUP BY company_id) AS trans_count
156 ORDER BY transaction_count DESC;
```

The result grid shows the following data:

company_id	company_name	transaction_count	¿Más? o, ¿menos?
b-2302	Nunc Interdum Incorporated	105	Más de 4 transacciones
b-2346	Ut Semper Foundation	59	Más de 4 transacciones
b-2326	Enim Condimentum Ltd	57	Más de 4 transacciones
b-2278	Arcu LLP	56	Más de 4 transacciones
b-2362	Lorem Eu Incorporated	54	Más de 4 transacciones
b-2494	Malesuada PC	52	Más de 4 transacciones
b-2618	Non Institute	30	Más de 4 transacciones
b-2230	Fusce Corp.	2	Menos de 4 transacciones
b-2234	Convallis In Incorporated	2	Menos de 4 transacciones
b-2238	Ante Iaculis Nec Foundation	2	Menos de 4 transacciones
b-2242	Donec Ltd	2	Menos de 4 transacciones
b-2246	Sed Nunc Ltd	2	Menos de 4 transacciones
b-2250	Amet Nulla Donec Corporation	2	Menos de 4 transacciones
b-2254	Nascetur Ridiculus Mus Inc.	2	Menos de 4 transacciones
b-2258	Vestibulum Lorem PC	2	Menos de 4 transacciones
b-2262	Gravida Sagittis LLP	2	Menos de 4 transacciones
b-2266	Mus Aenean Eget Foundation	2	Menos de 4 transacciones
b-2270	Dis Parturient Institute	2	Menos de 4 transacciones

The output message at the bottom states: "13 11:44:49 SELECT company\_id, company\_name, transaction\_count, CASE WHEN transaction\_count > 4 THEN 'Más de 4 transacciones' ELSE 'Menos de 4 t...' 100 row(s) returned".