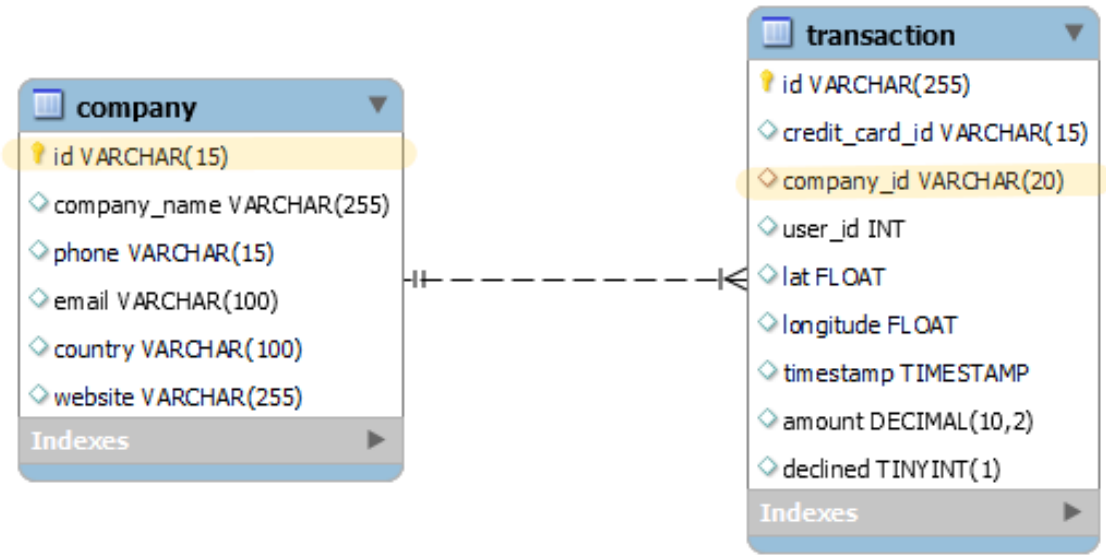
Sprint 2 - Tasca S2.01. Nocions bàsiques SQL

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” i “dades_introduir”) para crear la base de datos "transactions" que contiene dos tablas. La tabla principal es "company" y la secundaria es “transaction”.



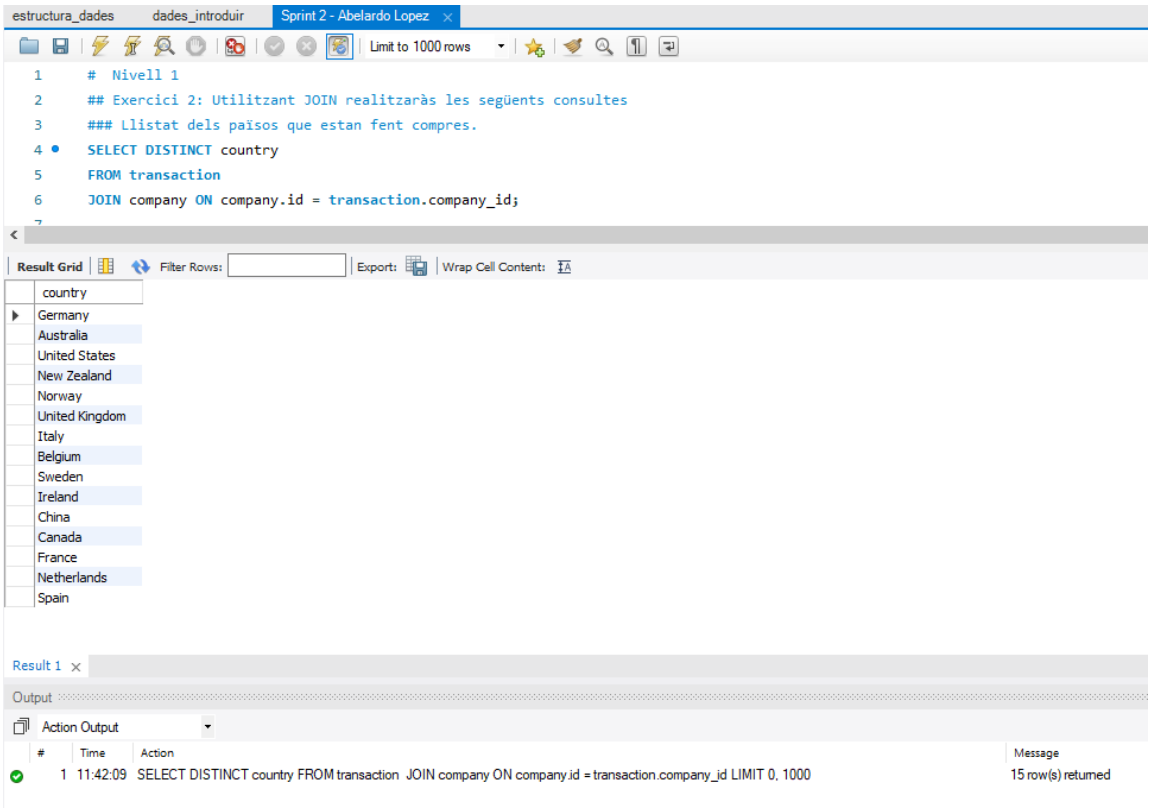
En la ilustración podemos ver que la columna “id” en la tabla “company” es la PK, que a su vez funciona como la FK de la tabla “transaction”. La relación entre tablas es de 1 a n.

- **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.

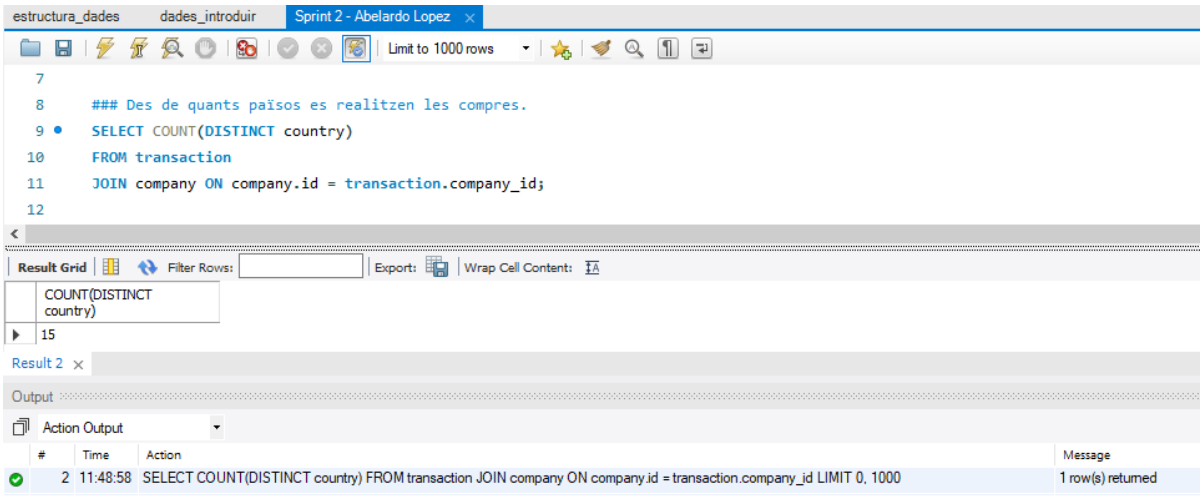
```
SELECT DISTINCT country
FROM transaction
JOIN company ON company.id = transaction.company_id;
```



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.

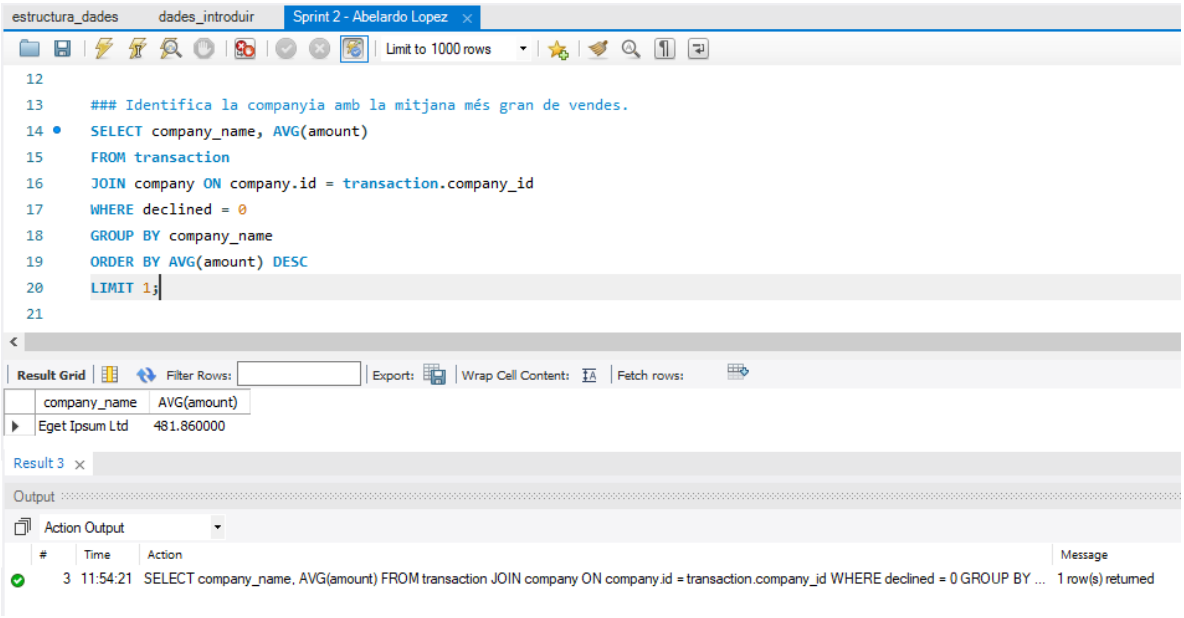
SELECT COUNT(DISTINCT country)
FROM transaction
JOIN company on company.id = transaction.company_id;



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



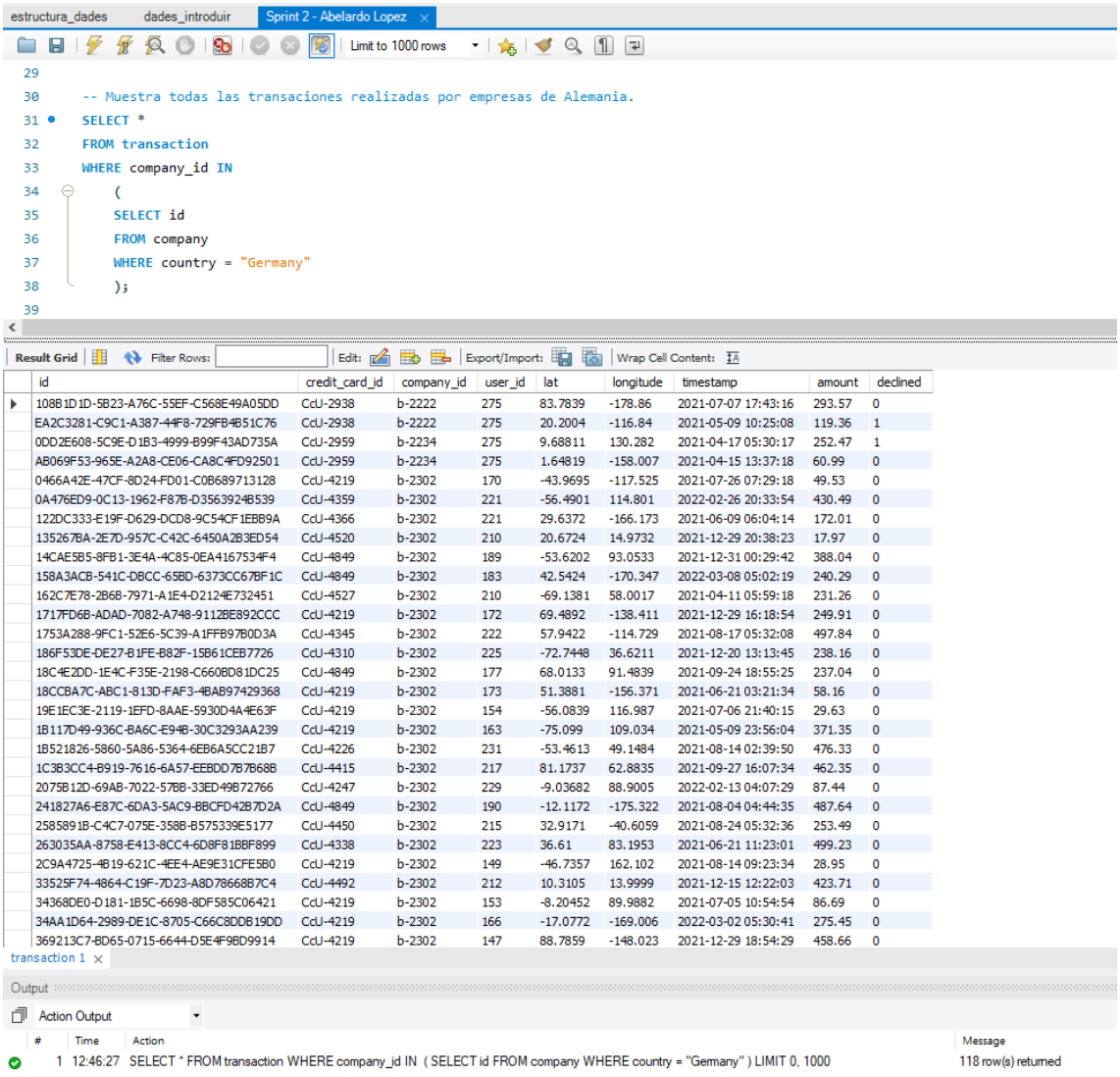
- **Exercici 3: Utilitzant només subconsultes (sense utilitzar JOIN):**

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 en una columna que se encuentra en ambas tablas)

```
SELECT *
FROM transaction
WHERE company_id IN

        (
        SELECT id
        FROM company
        WHERE country = "Germany"
        );
```

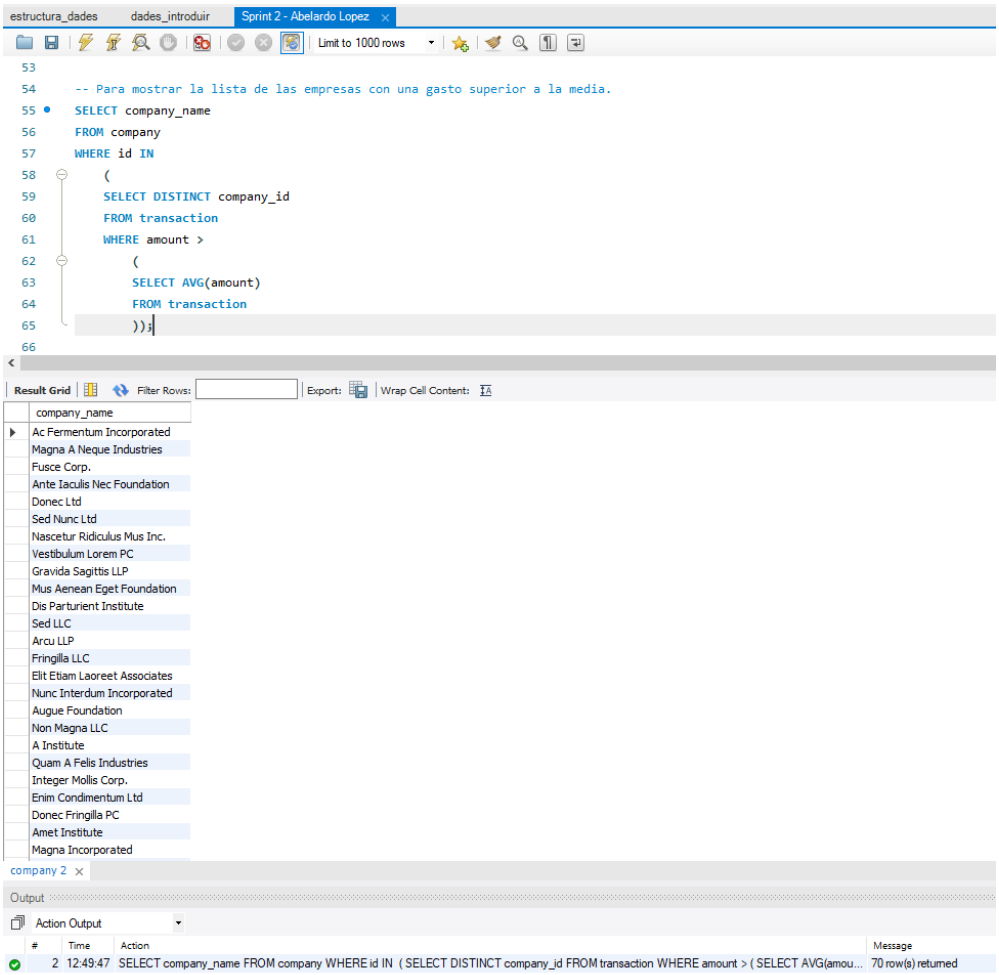


id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A05DD	CcU-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57	0
EA2C3281-C9C1-A387-44F8-729FB4B51C76	CcU-2938	b-2222	275	20.2004	-116.84	2021-05-09 10:25:08	119.36	1
0DD2E608-5C9E-D1B3-4999-B99F43AD735A	CcU-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47	1
AB069F53-965E-A2A8-CE06-CA8C4FD92501	CcU-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99	0
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53	0
0A476ED9-0C13-1962-F87B-D3563924B539	CcU-4359	b-2302	221	-56.4901	114.801	2022-02-26 20:33:54	430.49	0
122DC333-E19F-D629-DCD8-9C54CF1EBB9A	CcU-4366	b-2302	221	29.6372	-166.173	2021-06-09 06:04:14	172.01	0
135267BA-2E7D-957C-C42C-6450A2B3ED54	CcU-4520	b-2302	210	20.6724	14.9732	2021-12-29 20:38:23	17.97	0
14CAE5B5-8FB1-3E4A-4C85-0EA4167534F4	CcU-4849	b-2302	189	-53.6202	93.0533	2021-12-31 00:29:42	388.04	0
158A3ACB-541C-DBCC-65BD-6373CC67BF1C	CcU-4849	b-2302	183	42.5424	-170.347	2022-03-08 05:02:19	240.29	0
162C7E78-2B6B-7971-A1E4-D2124E732451	CcU-4527	b-2302	210	-69.1381	58.0017	2021-04-11 05:59:18	231.26	0
1717FD6B-ADAD-7082-A748-9112BE892CCC	CcU-4219	b-2302	172	69.4892	-138.411	2021-12-29 16:18:54	249.91	0
1753A288-9FC1-52E6-5C39-A1FFB97B0D3A	CcU-4345	b-2302	222	57.9422	-114.729	2021-08-17 05:32:08	497.84	0
186F53DE-DE27-B1FE-B82F-15B61CEB7726	CcU-4310	b-2302	225	-72.7448	36.6211	2021-12-20 13:13:45	238.16	0
18C4E2DD-1E4C-F35E-2198-C660BD81DC25	CcU-4849	b-2302	177	68.0133	91.4839	2021-09-24 18:55:25	237.04	0
18CCBA7C-ABC1-813D-FAF3-4BAB97429368	CcU-4219	b-2302	173	51.3881	-156.371	2021-06-21 03:21:34	58.16	0
19E1EC3E-2119-1EFD-8AAE-5930D4A4E63F	CcU-4219	b-2302	154	-56.0839	116.987	2021-07-06 21:40:15	29.63	0
1B117D49-936C-BA6C-E94B-30C3293AA239	CcU-4219	b-2302	163	-75.099	109.034	2021-05-09 23:56:04	371.35	0
1B521826-5860-5A86-5364-6EB6A5CC21B7	CcU-4226	b-2302	231	-53.4613	49.1484	2021-08-14 02:39:50	476.33	0
1C3B3CC4-B919-7616-6A57-EEBDD7B7B68B	CcU-4415	b-2302	217	81.1737	62.8835	2021-09-27 16:07:34	462.35	0
2075B12D-69AB-7022-57BB-33ED49B72766	CcU-4247	b-2302	229	-9.03682	88.9005	2022-02-13 04:07:29	87.44	0
241827A6-E87C-6DA3-5AC9-BBCFD42B7D2A	CcU-4849	b-2302	190	-12.1172	-175.322	2021-08-04 04:44:35	487.64	0
2585891B-C4C7-075E-358B-B575339E5177	CcU-4450	b-2302	215	32.9171	-40.6059	2021-08-24 05:32:36	253.49	0
263035AA-8758-E413-8CC4-6D8F81BBF899	CcU-4338	b-2302	223	36.61	83.1953	2021-06-21 11:23:01	499.23	0
2C9A4725-4B19-621C-4EE4-AE9E31CFE5B0	CcU-4219	b-2302	149	-46.7357	162.102	2021-08-14 09:23:34	28.95	0
33525F74-4864-C19F-7D23-A8D78668B7C4	CcU-4492	b-2302	212	10.3105	13.9999	2021-12-15 12:22:03	423.71	0
34368DE0-D181-1B5C-6698-8DF585C06421	CcU-4219	b-2302	153	-8.20452	89.9882	2021-07-05 10:54:54	86.69	0
34AA1D64-2989-DE1C-8705-C66C8DDB19DD	CcU-4219	b-2302	166	-17.0772	-169.006	2022-03-02 05:30:41	275.45	0
369213C7-BD65-0715-6644-D5E4F9BD9914	CcU-4219	b-2302	147	88.7859	-148.023	2021-12-29 18:54:29	458.66	0

3.2 Llista 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 medio 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.

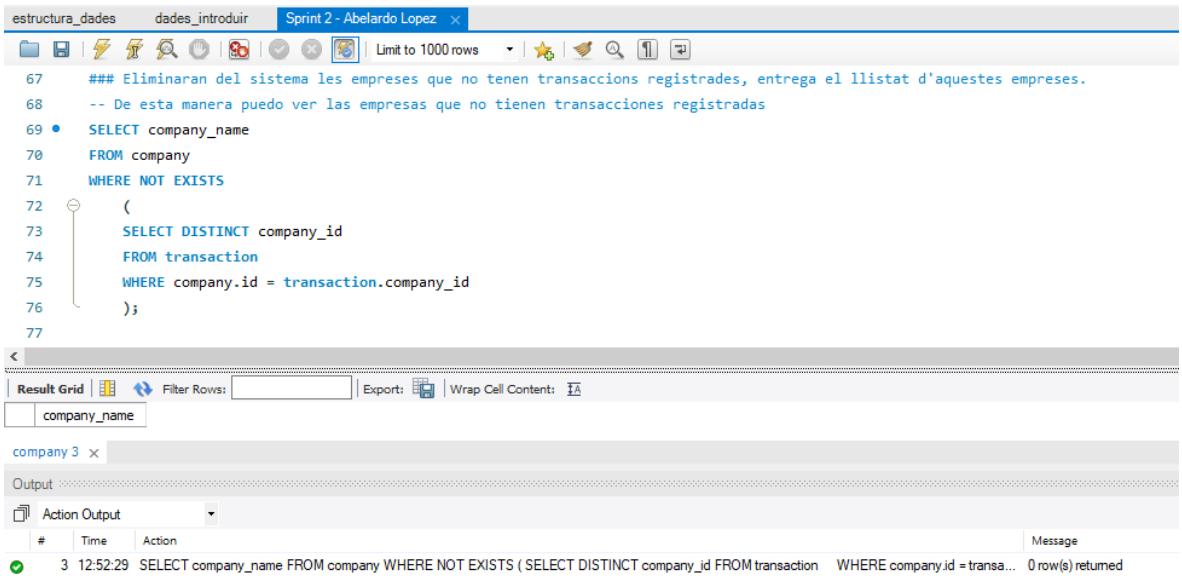
```
SELECT company_name
FROM company
WHERE id IN

        (
        SELECT DISTINCT company_id
        FROM transaction
        WHERE amount >

                (
                SELECT AVG(amount)
                FROM transaction
                ));
```



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.

```
SELECT company_name
FROM company
WHERE NOT EXISTS
        (
        SELECT DISTINCT company_id
        FROM transaction
        WHERE company.id = transaction.company_id
        );
```



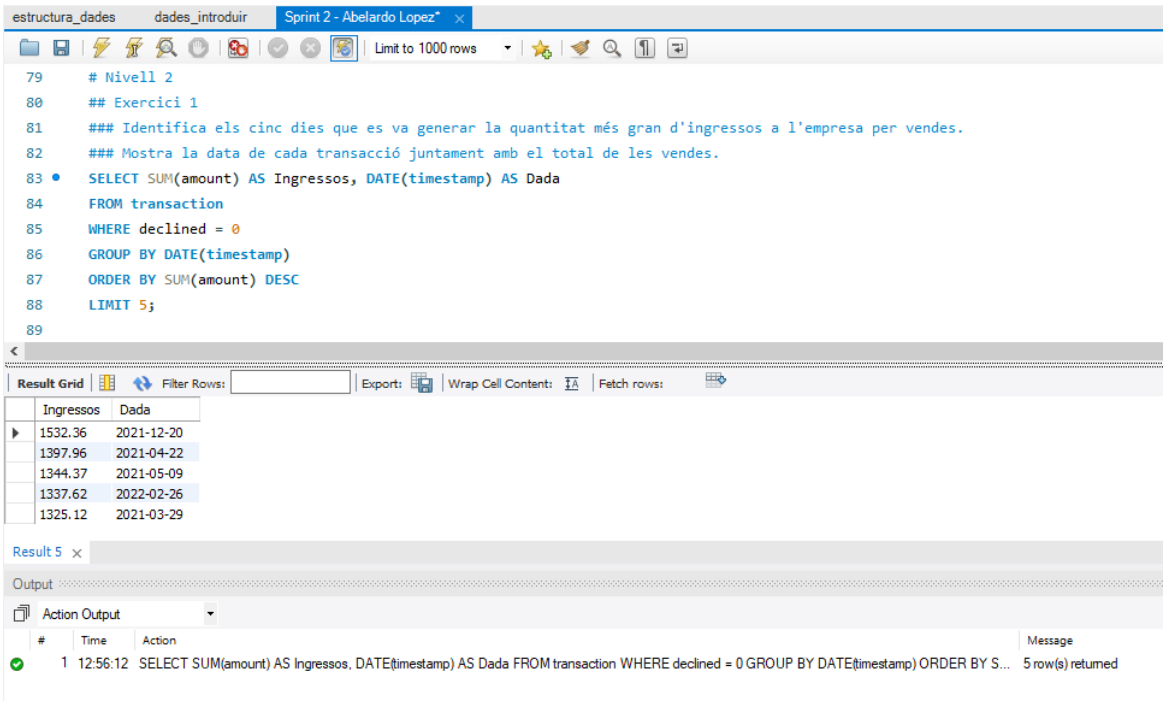
Nivell 2

- **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.
Finalmente ordené por el resultado de SUM(amount) de forma descendente para así 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), DATE(timestamp)
FROM transaction
GROUP BY DATE(timestamp)
ORDER BY SUM(amount) DESC
LIMIT 5;
```

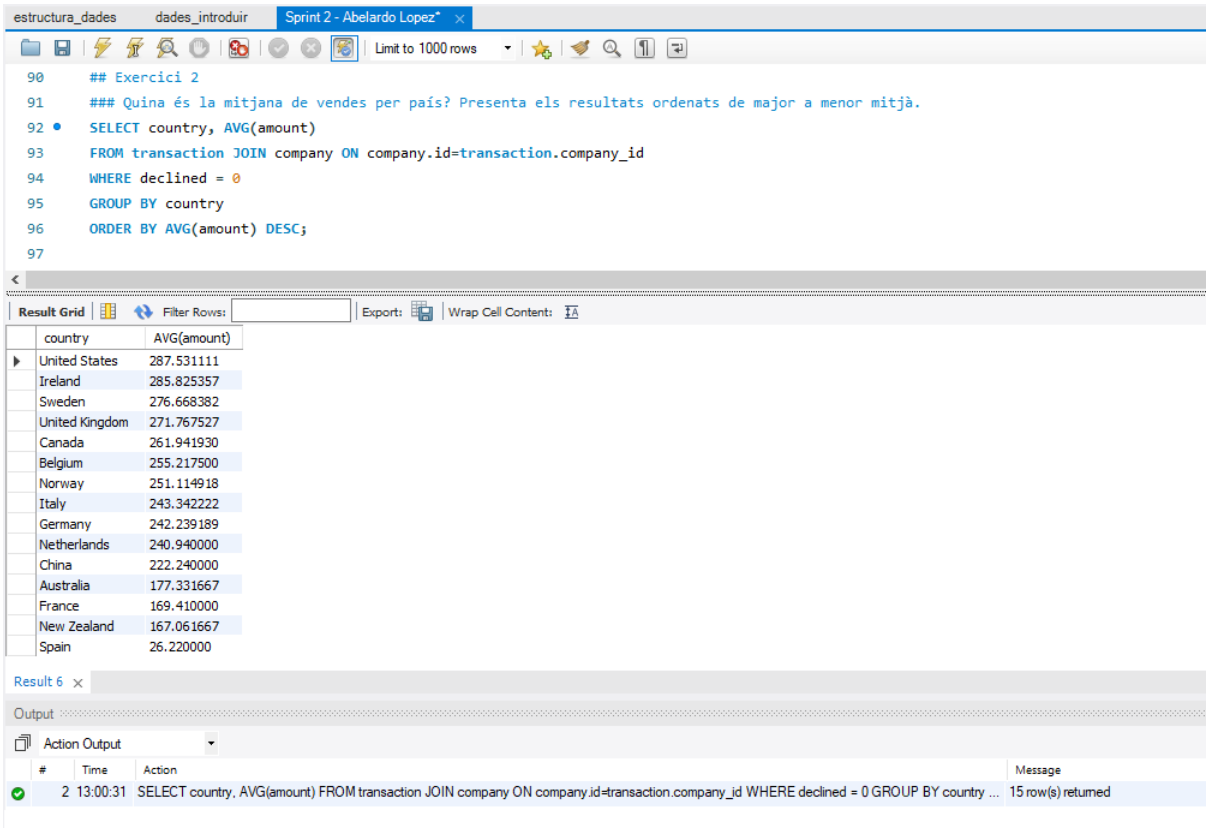


- **Exercici 2**

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 JOIN company ON company.id=transaction.company_id
WHERE declined = 0
GROUP BY country
ORDER BY AVG(amount) DESC



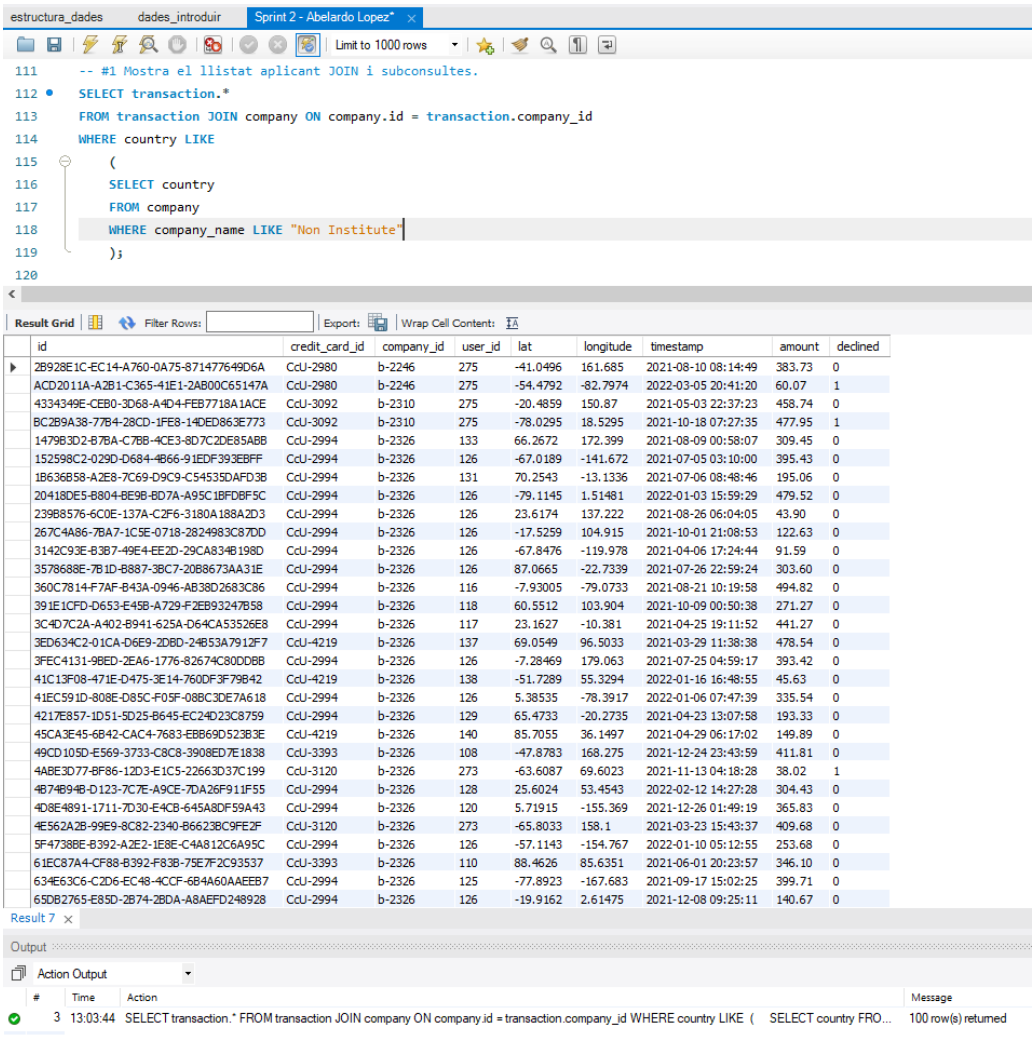
- **Exercici 3**

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”

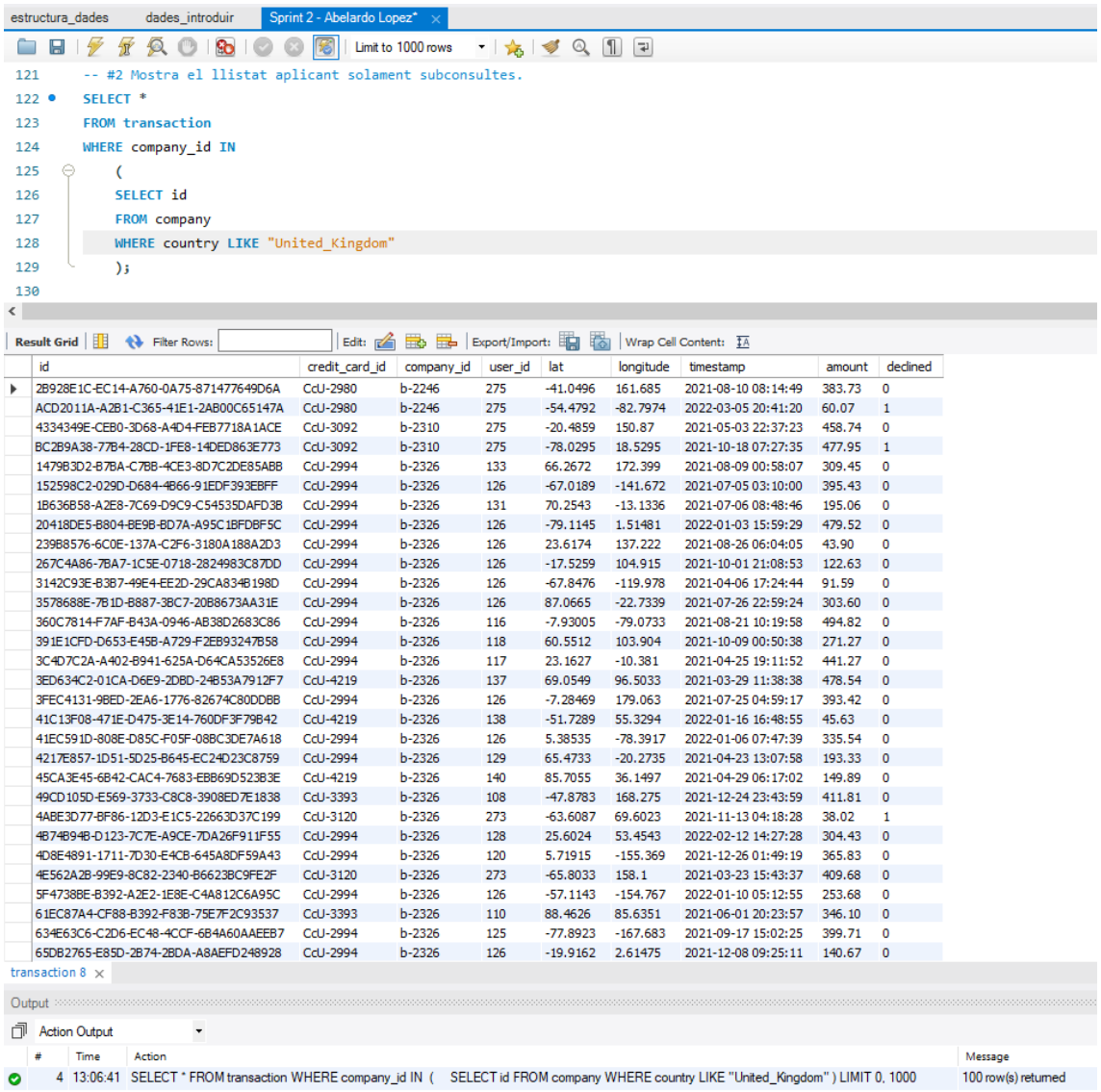
```
SELECT transaction.*
FROM transaction JOIN company ON company.id = transaction.company_id
WHERE country LIKE
    (
    SELECT country
    FROM company
    WHERE company_name LIKE "Non Institute"
    );
```



3.1.2 Mostra el llistat aplicant solament subconsultes.

Como el filtro anterior me había dicho que el país en donde se ubica “Non Institute” es “United_Kingdom” utilicé esa información para modelar la subQuery que me serviría como filtro, y mediante ese filtro mostrar todas las transacciones de las empresas ubicadas en ese país (alteré este filtro para no tener que anidar una subQuery dentro de otra).

```
SELECT *
FROM transaction
WHERE company_id IN

        (
        SELECT id
        FROM company
        WHERE country LIKE "United_Kingdom"
        );
```



```
-- #2 Mostra el llistat aplicant solament subconsultes.
SELECT *
FROM transaction
WHERE company_id IN
    (
    SELECT id
    FROM company
    WHERE country LIKE "United_Kingdom"
    );
```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
2B928E1C-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
BC2B9A38-77B4-28CD-1FE8-14DED863E773	CcU-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95	1
1479B3D2-B7BA-C7BB-4CE3-8D7C2DE85ABB	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-A2E8-7C69-D9C9-C54535DAFD3B	CcU-2994	b-2326	131	70.2543	-13.1336	2021-07-06 08:48:46	195.06	0
20418DE5-B804-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-B3B7-49E4-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
360C7814-F7AF-B43A-0946-AB38D2683C86	CcU-2994	b-2326	116	-7.93005	-79.0733	2021-08-21 10:19:58	494.82	0
391E1CFD-D653-E45B-A729-F2EB93247B58	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
3FEC4131-9BED-2EA6-1776-82674C80DDBB	CcU-2994	b-2326	126	-7.28469	179.063	2021-07-25 04:59:17	393.42	0
41C13F08-471E-D475-3E14-760DF3F79B42	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
45CA3E45-6B42-CAC4-7683-EBB69D523B3E	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-D123-7C7E-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-B392-A2E2-1E8E-C4A812C6A95C	CcU-2994	b-2326	126	-57.1143	-154.767	2022-01-10 05:12:55	253.68	0
61EC87A4-CF88-B392-F83B-75E7F2C93537	CcU-3393	b-2326	110	88.4626	85.6351	2021-06-01 20:23:57	346.10	0
634E63C6-C2D6-EC48-4CCF-6B4A60AAEEB7	CcU-2994	b-2326	125	-77.8923	-167.683	2021-09-17 15:02:25	399.71	0
65DB2765-E85D-2B74-2BDA-A8AEFD248928	CcU-2994	b-2326	126	-19.9162	2.61475	2021-12-08 09:25:11	140.67	0

```
SELECT * FROM transaction WHERE company_id IN (  SELECT id FROM company WHERE country LIKE "United_Kingdom" ) LIMIT 0, 1000
```

100 row(s) returned

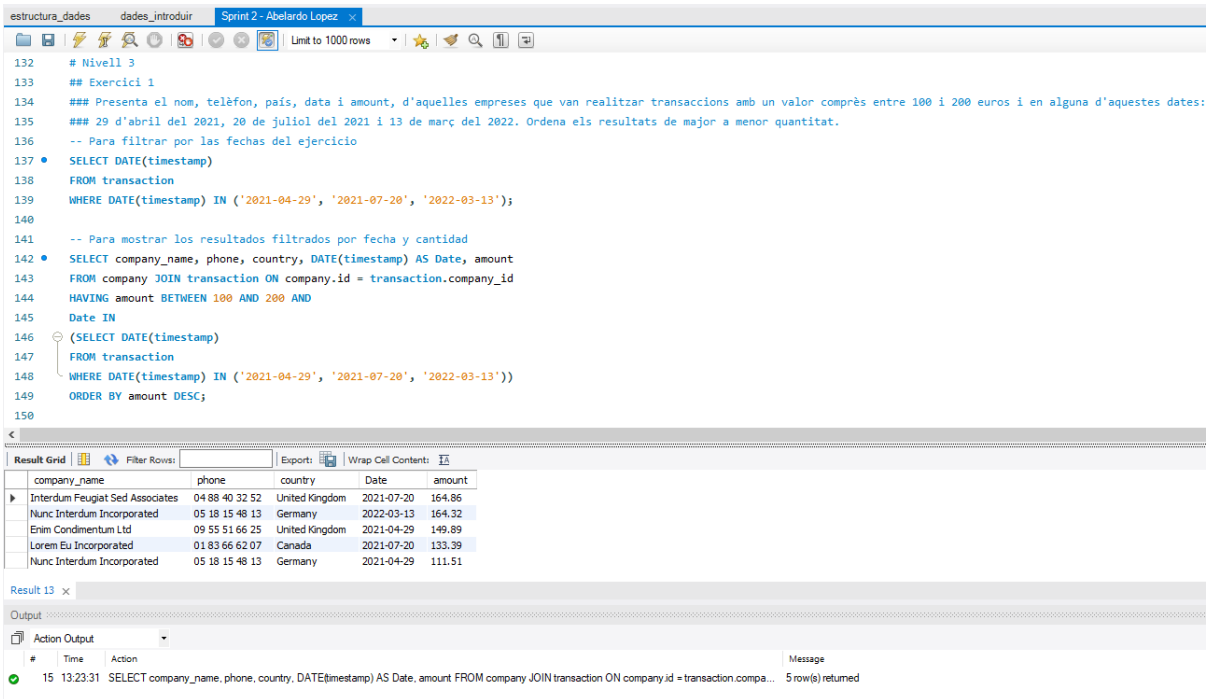
Nivell 3

- **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 utilicé una subQuery para filtrar por las fechas específicas que pide el ejercicio. Lo siguiente fue poner todos los campos que pide el ejercicio en el SELECT, procurando nombrar DATE(timestamp) con el alias “date” para usarlo en el HAVING, porque si no me daba el error de que “timestamp” era una columna desconocida.

```
SELECT company_name, phone, country, DATE(timestamp) AS date, amount
FROM company JOIN transaction ON company.id = transaction.company_id
HAVING amount BETWEEN 100 AND 200 AND
date IN

    (
    SELECT DATE(timestamp)
    FROM transaction
    WHERE DATE(timestamp) IN ('2021-04-29', '2021-07-20', '2022-03-13')
    )

ORDER BY amount DESC;
```



- **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 crear una tabla temporal con el conteo de transacciones que tiene cada empresa, y su respectivo identificador para poder usarlo luego en un JOIN.
Lo siguiente fue mostrar los nombres de las empresas que tienen más de cuatro transacciones, junto a la cantidad de transacciones que hay registradas.

```
CREATE TEMPORARY TABLE trans_count
SELECT company_id, COUNT(id) AS transaction_count
FROM transaction
GROUP BY company_id;

SELECT company_name, transaction_count
FROM company JOIN trans_count ON company.id = trans_count.company_id
WHERE transaction_count >= 4;
```