

## SPRINT 2 – NIVEL 1

### Sprint 2, Nivel 1, Ejercicio 1

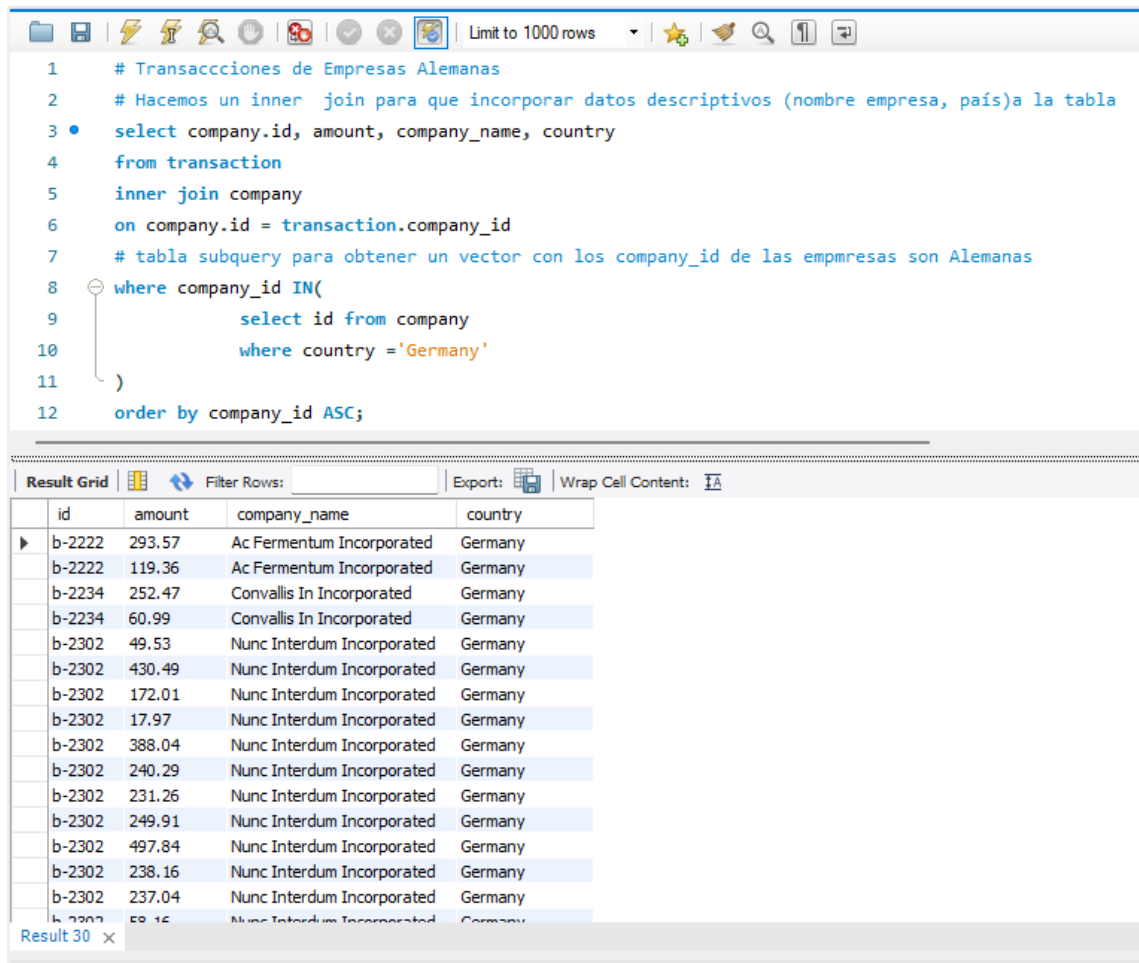
Empresas Alemanas que han hecho compras. Creamos un subquery para obtener el Vector con los ID de las empresas alemanas y que usaremos en la tabla de Transacciones para listarlas.

```
1 # SPRINT 2, NIVEL 1, EJERCICIO 1 - todas las transacciones de las empresas alemanas
2 • select *
3   from transaction as t2
4  where company_id in(
5      select id
6      from company as t1
7      where t2.company_id = t1.id
8      and t1.country = 'Germany'
9  );
10 • select t1.country, t2.*
11   from transaction as t2
12  join company as t1
13  on t2.company_id = t1.id
14  where t1.country = 'Germany';
15
16 # SPRINT 2, NIVEL 1, EJERCICIO 2 - empresas con transacciones por cantidad superior a la media de to
```

Result Grid										
Filter Rows:										
Export:										
Wrap Cell Contents:										
country	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	
Germany	108B1D1D-5B23-A76C-55EF-C568E49A05D0	CcU-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57	0	
Germany	EA2C3381-C9C1-A387-44F8-729FB4B51C76	CcU-2938	b-2222	275	20.2004	-116.84	2021-05-09 10:25:08	119.36	1	
Germany	0002E608-5C9E-D1B3-4999-B99F43AD735A	CcU-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47	1	
Germany	AB069F53-963E-A2A8-CE06-CABCFD92501	CcU-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99	0	
Germany	0466A42E-47CF-8024-FD01-C0B689713128	CcU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53	0	
Germany	0A476ED9-0C13-1962-F87B-D3563924B539	CcU-4359	b-2302	221	-56.4901	114.801	2022-02-26 20:33:54	430.49	0	
Germany	122DC333-E19F-D629-DCD8-9C54CF1EBB9A	CcU-4366	b-2302	221	29.6372	-166.173	2021-06-09 06:04:14	172.01	0	
Germany	135267BA-2E7D-957C-C42C-6450A283ED54	CcU-4520	b-2302	210	20.6724	14.9732	2021-12-29 20:38:23	17.97	0	

**Vemos a continuación un método alternativo para resolver el problema (menos eficiente)**

Vemos a continuación una alternativa con INNER JOIN para que la lectura de la tabla será más Amigable. Este código es más ineficiente



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

```
1 # Transacciones de Empresas Alemanas
2 # Hacemos un inner join para que incorporar datos descriptivos (nombre empresa, país) a la tabla
3 • select company.id, amount, company_name, country
4 from transaction
5 inner join company
6 on company.id = transaction.company_id
7 # tabla subquery para obtener un vector con los company_id de las empresas son Alemanas
8 where company_id IN(
9     select id from company
10     where country = 'Germany'
11 )
12 order by company_id ASC;
```

The results grid shows the following data:

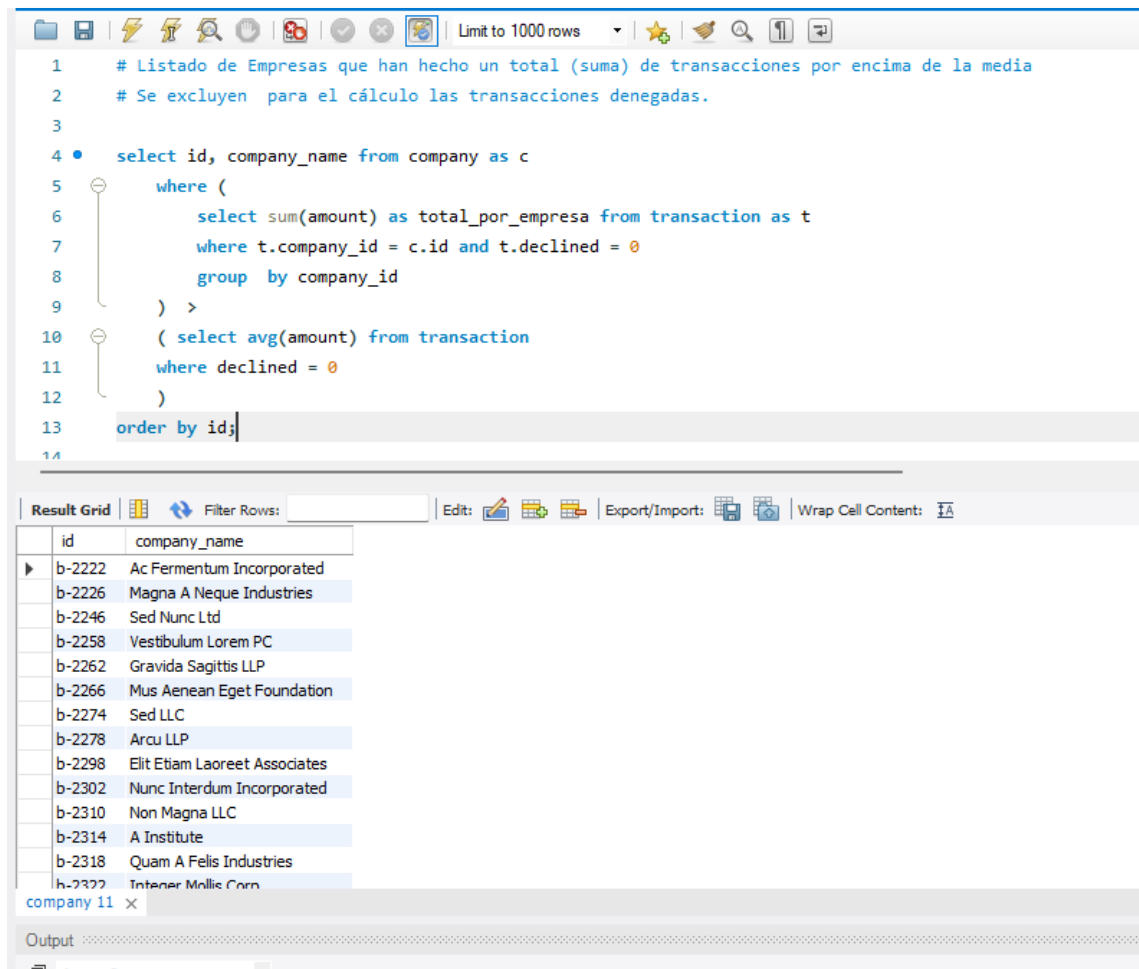
	id	amount	company_name	country
▶	b-2222	293.57	Ac Fermentum Incorporated	Germany
	b-2222	119.36	Ac Fermentum Incorporated	Germany
	b-2234	252.47	Convallis In Incorporated	Germany
	b-2234	60.99	Convallis In Incorporated	Germany
	b-2302	49.53	Nunc Interdum Incorporated	Germany
	b-2302	430.49	Nunc Interdum Incorporated	Germany
	b-2302	172.01	Nunc Interdum Incorporated	Germany
	b-2302	17.97	Nunc Interdum Incorporated	Germany
	b-2302	388.04	Nunc Interdum Incorporated	Germany
	b-2302	240.29	Nunc Interdum Incorporated	Germany
	b-2302	231.26	Nunc Interdum Incorporated	Germany
	b-2302	249.91	Nunc Interdum Incorporated	Germany
	b-2302	497.84	Nunc Interdum Incorporated	Germany
	b-2302	238.16	Nunc Interdum Incorporated	Germany
	b-2302	237.04	Nunc Interdum Incorporated	Germany
	b-2302	59.16	Nunc Interdum Incorporated	Germany

The results grid also includes a "Filter Rows:" field, an "Export:" button, and a "Wrap Cell Content:" checkbox.

## Sprint 2, Nivel 1, Ejercicio 2

### Listado con las Empresas que han hecho compras por encima de la media de todas las transacciones

Creamos un subquery en el que se pregunta si la cantidad está por encima de la media, pero en el mismo acto, se vincula a la tabla company para extraer el nombre.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following query:

```
1 # Listado de Empresas que han hecho un total (suma) de transacciones por encima de la media
2 # Se excluyen para el cálculo las transacciones denegadas.
3
4 • select id, company_name from company as c
5   where (
6       select sum(amount) as total_por_empresa from transaction as t
7       where t.company_id = c.id and t.declined = 0
8       group by company_id
9   ) >
10  ( select avg(amount) from transaction
11    where declined = 0
12  )
13 order by id;
```

Below the editor is the 'Result Grid' tab, which displays the query results in a table:

	id	company_name
▶	b-2222	Ac Fermentum Incorporated
	b-2226	Magna A Neque Industries
	b-2246	Sed Nunc Ltd
	b-2258	Vestibulum Lorem PC
	b-2262	Gravida Sagittis LLP
	b-2266	Mus Aenean Eget Foundation
	b-2274	Sed LLC
	b-2278	Arcu LLP
	b-2298	Elit Etiam Laoreet Associates
	b-2302	Nunc Interdum Incorporated
	b-2310	Non Magna LLC
	b-2314	A Institute
	b-2318	Quam A Felis Industries
	b-2322	Integer Mollis Corp

At the bottom of the result grid, there is a tab labeled 'company 11' with a close button (x). Below the result grid is an 'Output' pane.

### **Veamos a continuación un método alternativo para resolver el problema (menos eficiente)**

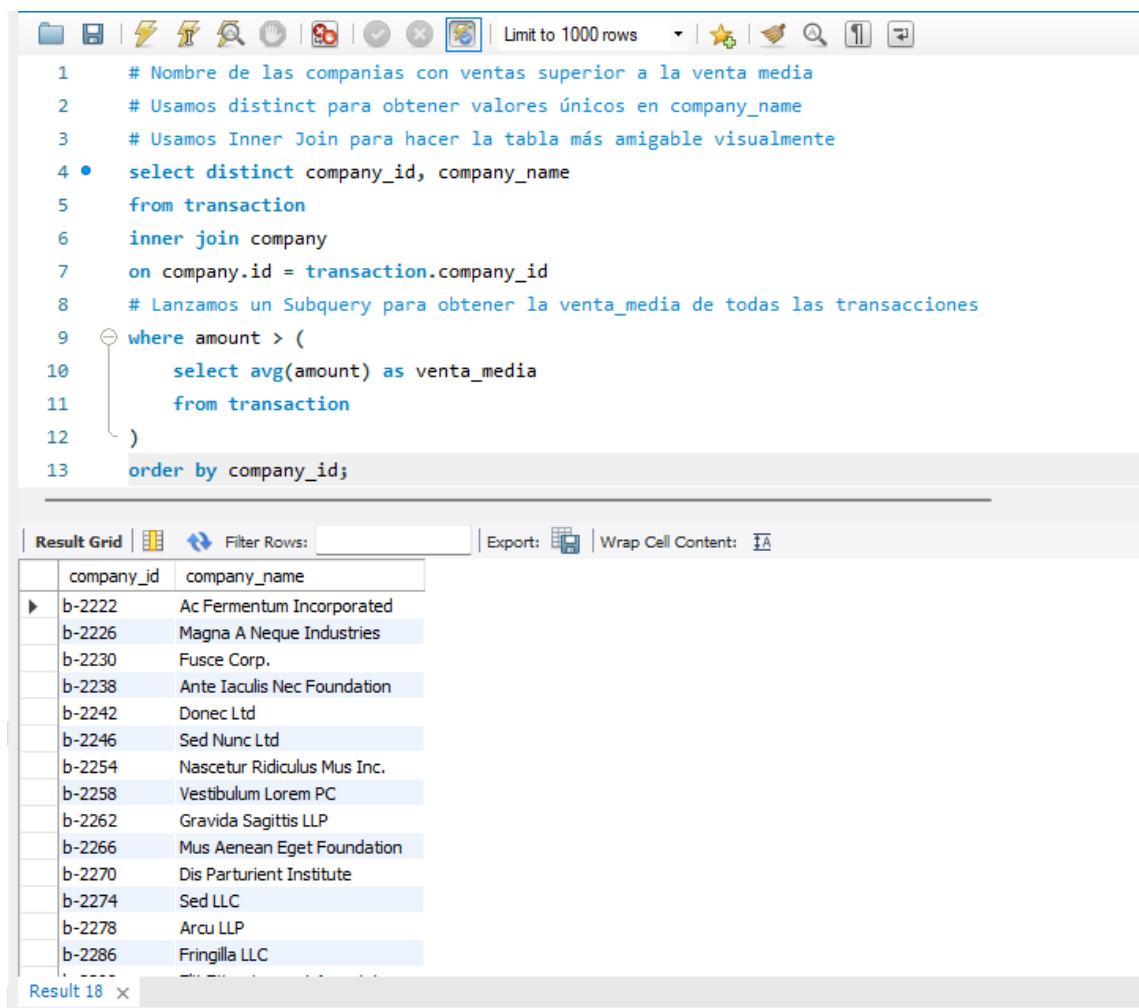
Creamos un Subquery para obtener la venta media de todas las transacciones.

Comparamos cada una de las transacciones con la cifra obtenida para ver si la operación es superior al valor medio calculado.

Con un Inner Join, llamamos al nombre de la empresa y confeccionamos el listado sin repetirlo.

Usamos Distinct para no repetir nombres de compañía.

Este método es menos eficiente, a efectos prácticos, se ha comparado con la media todas las transacciones sin agregar por empresa, por lo tanto, queda anulado por no cumplir con el enunciado.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, search, and execution. The query editor contains the following SQL code:

```
1  # Nombre de las compañías con ventas superior a la venta media
2  # Usamos distinct para obtener valores únicos en company_name
3  # Usamos Inner Join para hacer la tabla más amigable visualmente
4  • select distinct company_id, company_name
5     from transaction
6     inner join company
7     on company.id = transaction.company_id
8     # Lanzamos un Subquery para obtener la venta_media de todas las transacciones
9     where amount > (
10         select avg(amount) as venta_media
11         from transaction
12     )
13     order by company_id;
```

Below the query editor is the 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with two columns: 'company\_id' and 'company\_name'.

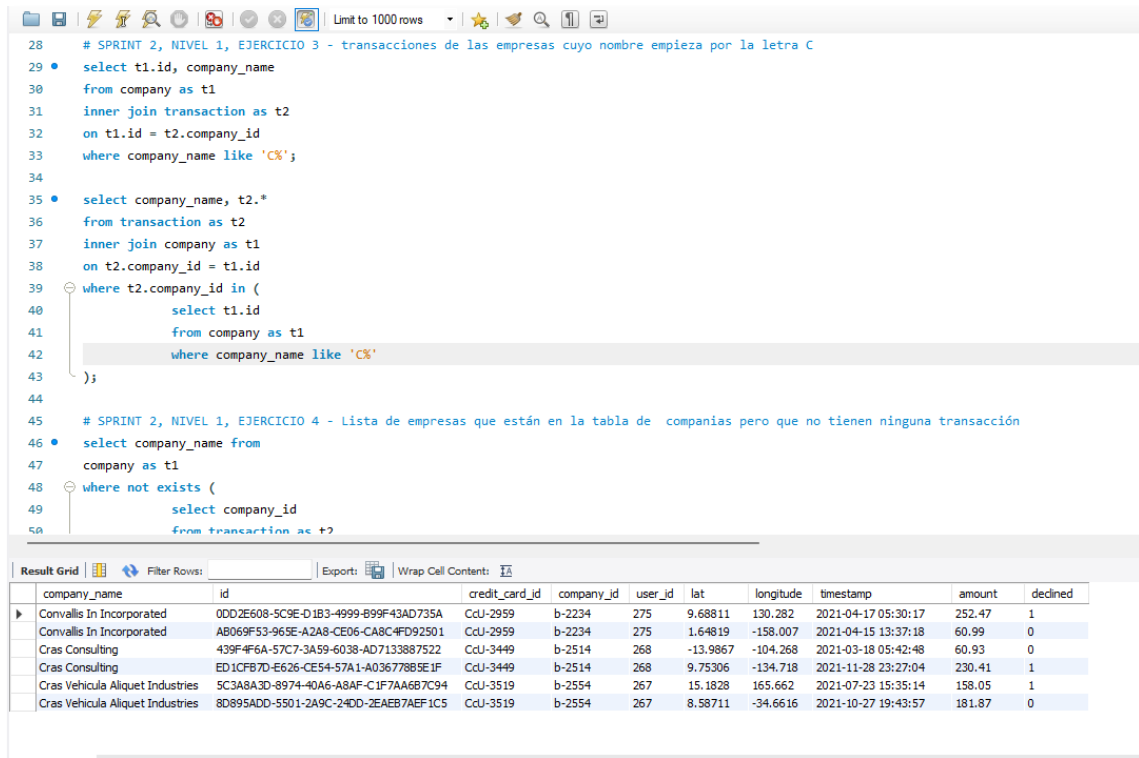
company_id	company_name
b-2222	Ac Fermentum Incorporated
b-2226	Magna A Neque Industries
b-2230	Fusce Corp.
b-2238	Ante Iaculis Nec Foundation
b-2242	Donec Ltd
b-2246	Sed Nunc Ltd
b-2254	Nascetur Ridiculus Mus Inc.
b-2258	Vestibulum Lorem PC
b-2262	Gravida Sagittis LLP
b-2266	Mus Aenean Eget Foundation
b-2270	Dis Parturient Institute
b-2274	Sed LLC
b-2278	Arcu LLP
b-2286	Fringilla LLC

The bottom status bar indicates 'Result 18'.

### Sprint 2, Nivel 1, Ejercicio 3

#### Recuperar transacciones de las empresas cuyo nombre empieza por c

Aplicamos EXIST a la subtabla con dos condiciones, la primera que los id de las clave primaria y externa coincidan, y luego un LIKE 'C%' para empresas que empiecen por C.



The screenshot shows a SQL IDE interface with two queries and their results. The first query, labeled 'SPRINT 2, NIVEL 1, EJERCICIO 3', uses an inner join between the 'company' and 'transaction' tables, filtering for companies whose names start with 'C'. The second query, labeled 'SPRINT 2, NIVEL 1, EJERCICIO 4', uses a 'where not exists' clause to find companies that do not have any transactions. The results are displayed in a table with columns: company\_name, id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, amount, and declined.

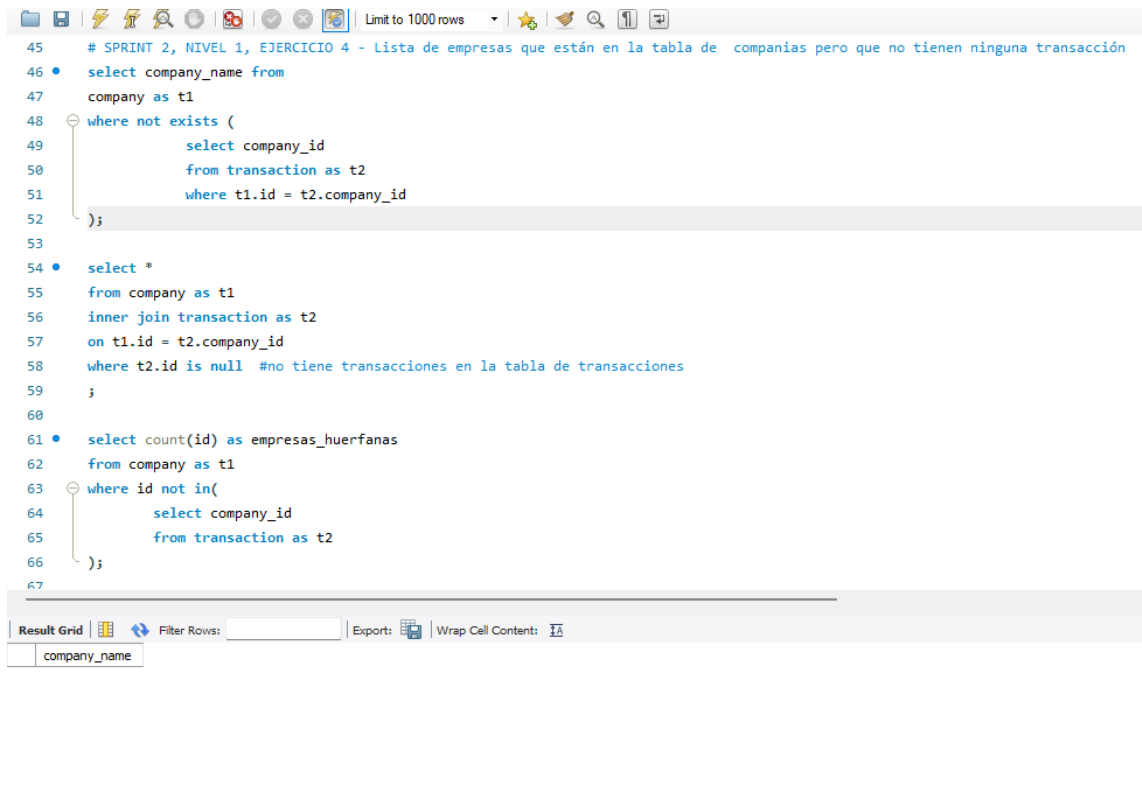
```
28 # SPRINT 2, NIVEL 1, EJERCICIO 3 - transacciones de las empresas cuyo nombre empieza por la letra C
29 • select t1.id, company_name
30 from company as t1
31 inner join transaction as t2
32 on t1.id = t2.company_id
33 where company_name like 'C%';
34
35 • select company_name, t2.*
36 from transaction as t2
37 inner join company as t1
38 on t2.company_id = t1.id
39 where t2.company_id in (
40     select t1.id
41     from company as t1
42     where company_name like 'C%'
43 );
44
45 # SPRINT 2, NIVEL 1, EJERCICIO 4 - Lista de empresas que están en la tabla de compañías pero que no tienen ninguna transacción
46 • select company_name from
47 company as t1
48 where not exists (
49     select company_id
50     from transaction as t2
```

company_name	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
Convallis In Incorporated	0DD2E608-5C9E-D1B3-4999-B99F43AD735A	CdU-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47	1
Convallis In Incorporated	AB069F53-965E-A2A8-CE06-CA8C4FD92501	CdU-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99	0
Cras Consulting	439F4F6A-57C7-3A59-6038-AD7133887522	CdU-3449	b-2514	268	-13.9867	-104.268	2021-03-18 05:42:48	60.93	0
Cras Consulting	ED1CFB7D-E626-CE54-57A1-A03677885E1F	CdU-3449	b-2514	268	9.75306	-134.718	2021-11-28 23:27:04	230.41	1
Cras Vehicula Aliquet Industries	5C3A8A3D-8974-40A6-A8AF-C1F7AA687C94	CdU-3519	b-2554	267	15.1828	165.662	2021-07-23 15:35:14	158.05	1
Cras Vehicula Aliquet Industries	8D895ADD-5501-2A9C-24DD-2EAE87AEF1C5	CdU-3519	b-2554	267	8.58711	-34.6616	2021-10-27 19:43:57	181.87	0

## Sprint 2, Nivel 1, Ejercicio 4

Listado de empresas que aparecen en la tabla company pero no en la de transaction, es decir, empresas registradas que no han comprado.

Aplicamos NOT EXIST en la tabla company para ver empresas que según su id no tiene correspondencia en la tabla transaction según su company\_id



```
45 # SPRINT 2, NIVEL 1, EJERCICIO 4 - Lista de empresas que están en la tabla de compañías pero que no tienen ninguna transacción
46 • select company_name from
47   company as t1
48   where not exists (
49     select company_id
50     from transaction as t2
51     where t1.id = t2.company_id
52   );
53
54 • select *
55   from company as t1
56   inner join transaction as t2
57   on t1.id = t2.company_id
58   where t2.id is null #no tiene transacciones en la tabla de transacciones
59   ;
60
61 • select count(id) as empresas_huerfanas
62   from company as t1
63   where id not in(
64     select company_id
65     from transaction as t2
66   );
67
```

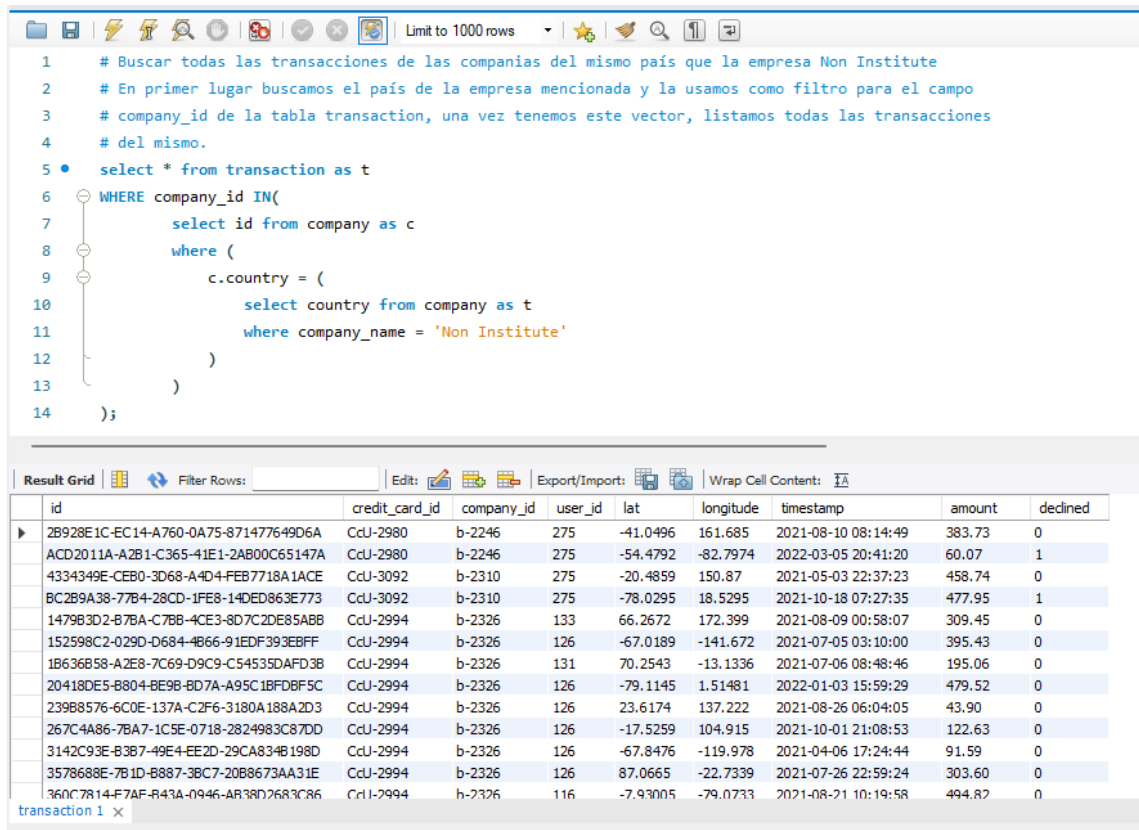
Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

company_name
--------------

## Sprint 2, Nivel 2, Ejercicio 1

**Buscamos las transacciones de aquellas empresas que pertenecen al mismo país que la empresa Non Institute**

Tras obtener el country de esta empresa, el método es crear un select para obtener un vector con los id de las empresas que pertenecen al mismo país, para finalmente iterar por el mismo con el método IN y poder así listar las transacciones.



```
1 # Buscar todas las transacciones de las companias del mismo país que la empresa Non Institute
2 # En primer lugar buscamos el país de la empresa mencionada y la usamos como filtro para el campo
3 # company_id de la tabla transaction, una vez tenemos este vector, listamos todas las transacciones
4 # del mismo.
5 • select * from transaction as t
6 WHERE company_id IN(
7     select id from company as c
8     where (
9         c.country = (
10             select country from company as t
11             where company_name = 'Non Institute'
12         )
13     )
14 );
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	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-C365-41E1-2A800C65147A	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-A95C1BFD8F5C	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-R43A-0946-AR38D26R3C86	CcU-2994	b-2326	116	-7.93005	-79.0733	2021-08-21 10:19:58	494.82	0

transaction 1 x

## Sprint 2, Nivel 2, Ejercicio 2

### Buscamos la empresa que ha realizado la mayor transacción

ordenamos por orden descendiente la tabla transacción según las transacciones en el campo amount para tomar luego la primera fila.

Finalmente vemos a que compañía corresponde el id de esta transacción.

```
93 # SPRINT 2, NIVEL 2, EJERCICIO 2 - Nombre de la empresa con la transacción de mayor cantidad
94 • select c.company_name, sum(amount) as parcial_empresa
95 from transaction as t
96 join company as c
97 on c.id = t.company_id
98 group by t.company_id
99 order by parcial_empresa desc
100 limit 1;
101
102 • select company_name, subtotaes.parcial_empresa
103 from company as c
104 join(
105     select sum(amount) as parcial_empresa, company_id # es importante incluir este campo, ya que lo usaremos luego en el comando on
106     from transaction as t
107     group by t.company_id
108 ) as subtotaes
109 on subtotaes.company_id = c.id
110 order by parcial_empresa desc
111 limit 1;
112
113 # SPRINT 2, NIVEL 3, EJERCICIO 1- POR VENTAS - Listado de países con ventas agregadas superior a la media de ventas
114
115 • select c.country, count(t.id) as contador
```

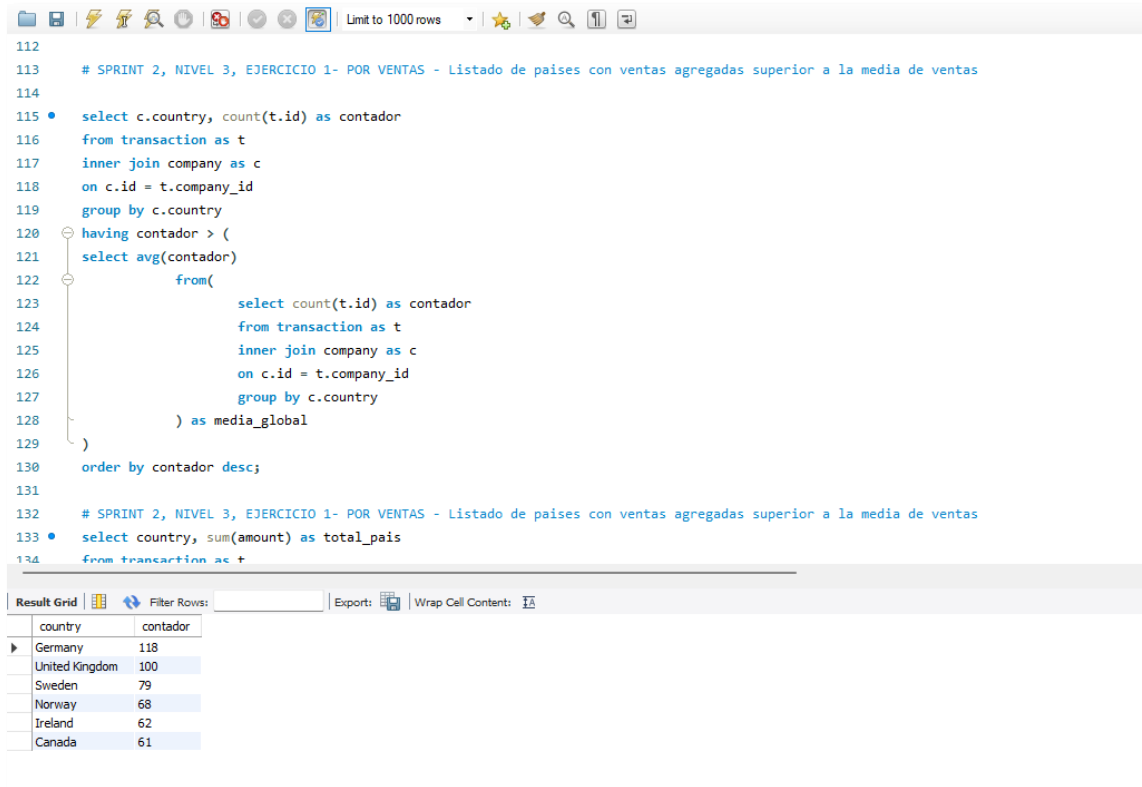
Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
company_name	parcial_empresa			
Nunc Interdum Incorporated	25622.65			



## Sprint 2, Nivel 3, Ejercicio 1

### **Países cuya ventas en cantidad de operaciones sea superior a la venta media de todos los países**

Primero usamos el comando SUM para agrupar la cantidad amount por país para vectorizarlos en total\_pais . Luego usamos este vector para obtener la media, luego usamos el comando cross join para comparar las dos tablas, y finalmente añadimos un Where para comprar total\_pais con media\_total\_paises.



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

```
112
113 # SPRINT 2, NIVEL 3, EJERCICIO 1- POR VENTAS - Listado de paises con ventas agregadas superior a la media de ventas
114
115 • select c.country, count(t.id) as contador
116    from transaction as t
117   inner join company as c
118   on c.id = t.company_id
119   group by c.country
120   having contador > (
121     select avg(contador)
122     from(
123       select count(t.id) as contador
124       from transaction as t
125       inner join company as c
126       on c.id = t.company_id
127       group by c.country
128     ) as media_global
129   )
130  order by contador desc;
131
132 # SPRINT 2, NIVEL 3, EJERCICIO 1- POR VENTAS - Listado de paises con ventas agregadas superior a la media de ventas
133 • select country, sum(amount) as total_pais
134    from transaction as t
```

The results grid shows the following data:

country	contador
Germany	118
United Kingdom	100
Sweden	79
Norway	68
Ireland	62
Canada	61

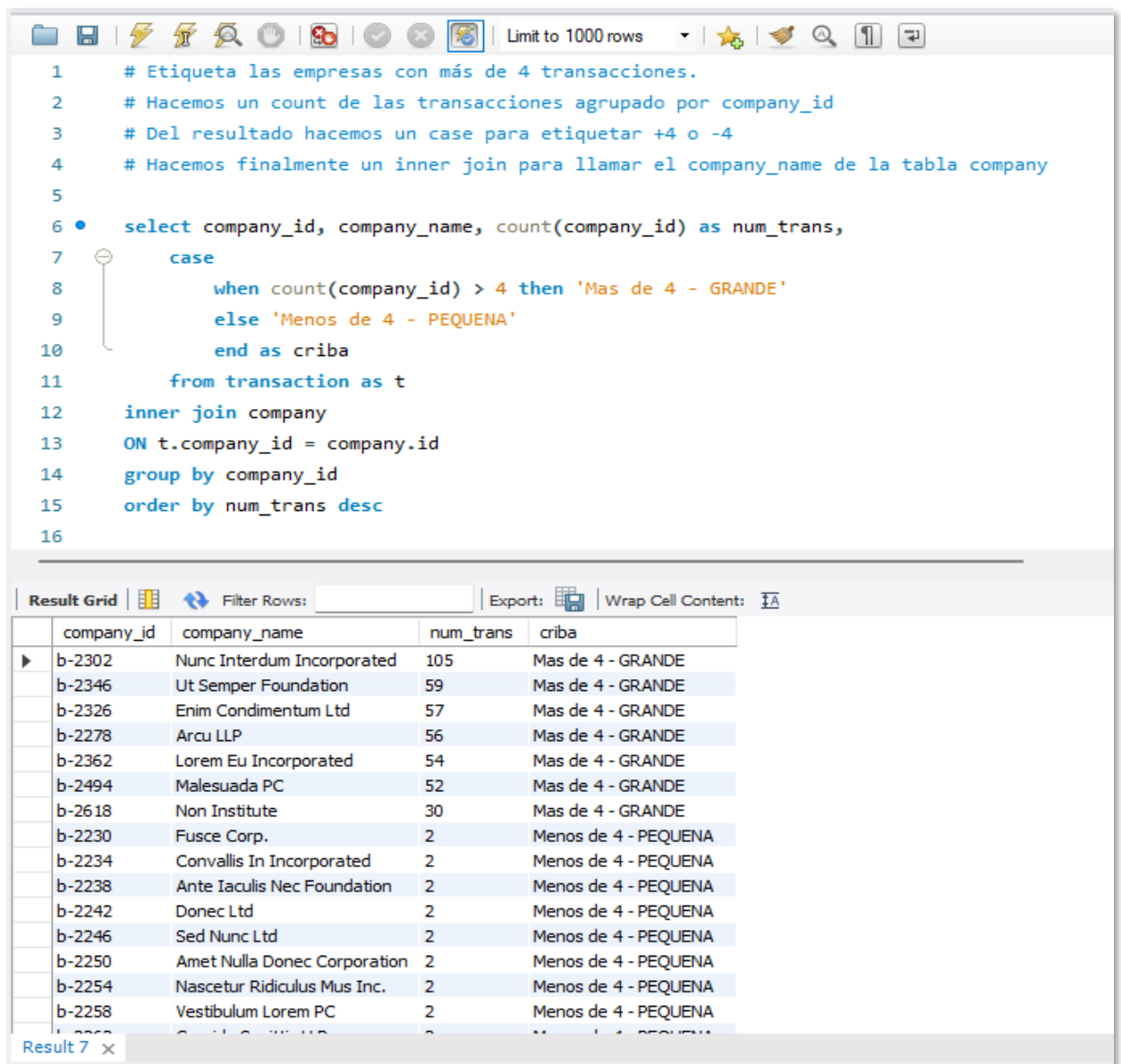
## Sprint 2, Nivel 3, Ejercicio 2

**Necesitamos etiquetar las empresas que tienen más o menos de 4 transacciones**

Hacemos un COUNT del fichero transacciones agrupado por company\_id

Usamos un case para etiquetar +4 o -4 el resultado.

Finalmente hacemos un inner join para llamar al compo company\_name e la tabla company



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search, along with a 'Limit to 1000 rows' dropdown. The SQL editor contains the following query:

```
1  # Etiqueta las empresas con más de 4 transacciones.
2  # Hacemos un count de las transacciones agrupado por company_id
3  # Del resultado hacemos un case para etiquetar +4 o -4
4  # Hacemos finalmente un inner join para llamar el company_name de la tabla company
5
6  • select company_id, company_name, count(company_id) as num_trans,
7      case
8          when count(company_id) > 4 then 'Mas de 4 - GRANDE'
9          else 'Menos de 4 - PEQUENA'
10         end as criba
11      from transaction as t
12      inner join company
13      ON t.company_id = company.id
14      group by company_id
15      order by num_trans desc
16
```

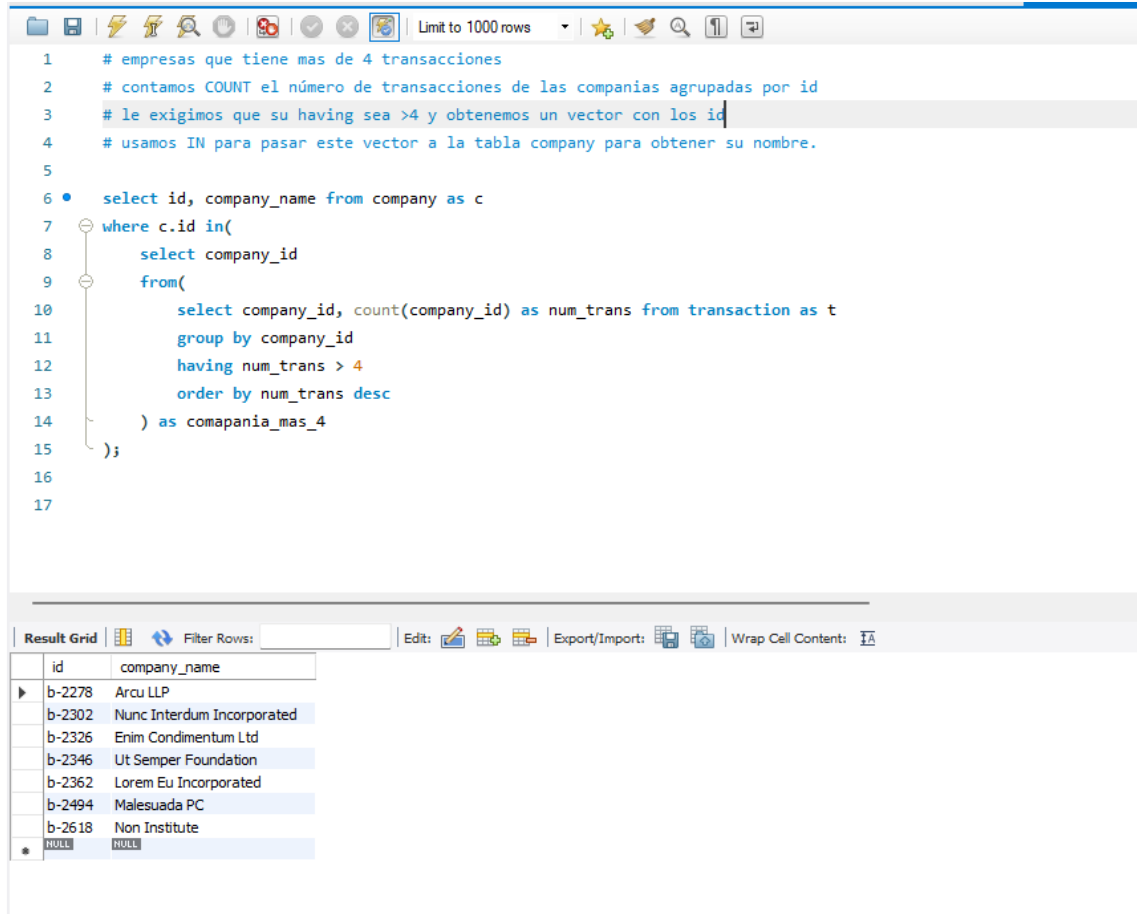
Below the editor is the 'Result Grid' section, which includes a 'Filter Rows' input, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with the following data:

company_id	company_name	num_trans	criba
b-2302	Nunc Interdum Incorporated	105	Mas de 4 - GRANDE
b-2346	Ut Semper Foundation	59	Mas de 4 - GRANDE
b-2326	Enim Condimentum Ltd	57	Mas de 4 - GRANDE
b-2278	Arcu LLP	56	Mas de 4 - GRANDE
b-2362	Lorem Eu Incorporated	54	Mas de 4 - GRANDE
b-2494	Malesuada PC	52	Mas de 4 - GRANDE
b-2618	Non Institute	30	Mas de 4 - GRANDE
b-2230	Fusce Corp.	2	Menos de 4 - PEQUENA
b-2234	Convallis In Incorporated	2	Menos de 4 - PEQUENA
b-2238	Ante Iaculis Nec Foundation	2	Menos de 4 - PEQUENA
b-2242	Donec Ltd	2	Menos de 4 - PEQUENA
b-2246	Sed Nunc Ltd	2	Menos de 4 - PEQUENA
b-2250	Amet Nulla Donec Corporation	2	Menos de 4 - PEQUENA
b-2254	Nascetur Ridiculus Mus Inc.	2	Menos de 4 - PEQUENA
b-2258	Vestibulum Lorem PC	2	Menos de 4 - PEQUENA

The bottom status bar indicates 'Result 7' with a close button.

### Sprint 2, Nivel 3, Ejercicio 2 – Alternativa

Mismo ejercicio anterior pero sin etiquetar, el código devuelve el nombre de las empresas que han hecho más de 4 transacciones.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. A dropdown menu shows 'Limit to 1000 rows'. The SQL editor contains the following code:

```
1 # empresas que tiene mas de 4 transacciones
2 # contamos COUNT el número de transacciones de las companias agrupadas por id
3 # le exigimos que su having sea >4 y obtenemos un vector con los id
4 # usamos IN para pasar este vector a la tabla company para obtener su nombre.
5
6 • select id, company_name from company as c
7   where c.id in(
8     select company_id
9     from(
10      select company_id, count(company_id) as num_trans from transaction as t
11      group by company_id
12      having num_trans > 4
13      order by num_trans desc
14     ) as compania_mas_4
15   );
16
17
```

Below the editor is the 'Result Grid' tab. It shows a table with two columns: 'id' and 'company\_name'. The results are as follows:

id	company_name
b-2278	Arcu LLP
b-2302	Nunc Interdum Incorporated
b-2326	Enim Condimentum Ltd
b-2346	Ut Semper Foundation
b-2362	Lorem Eu Incorporated
b-2494	Malesuada PC
b-2618	Non Institute
* NULL	NULL

The bottom toolbar includes options for 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'.