

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.

Empezamos con un SELECT * FROM transaction AS t2: Esta línea selecciona todas las columnas de la tabla transaction y la nombra como t2. Esto permite referirse a esta tabla con el nombre más corto t2 en otras partes de la consulta.

WHERE company_id IN (...): Esta condición restringe los resultados a las transacciones que están asociadas a empresas que cumplen determinados criterios.

SELECT id FROM company AS t1 WHERE t2.company_id = t1.id AND t1.country = 'Germany': Esta subconsulta selecciona los identificadores (id) de las empresas de la tabla company que están situadas en Alemania. Lo hace como t1. La condición t2.company_id = t1.id asegura que solo se seleccionen las empresas con las que coincide el company_id de la tabla transaction (es decir, las empresas asociadas a las transacciones).

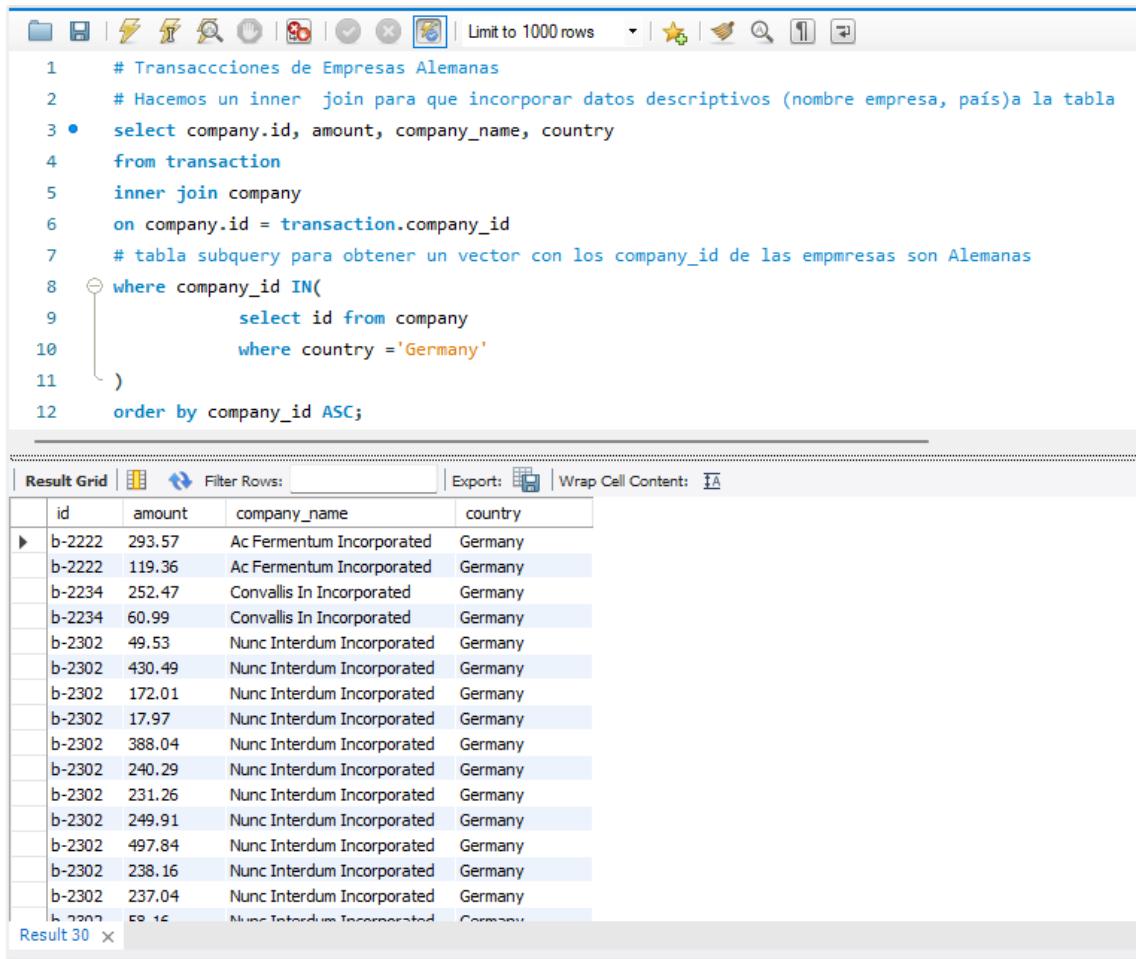
Por lo tanto, en resumen, esta consulta SQL seleccionará todas las transacciones que estén asociadas a empresas con ubicación en Alemania.

```
1  -- SPRINT 2, NIVELL 1, EXERCICI 1: Totes les transaccions de les empreses alemanyes
2
3  -- Aquí seleccionem totes les transaccions de les empreses alemanyes fent servir una subconsulta
4  -- La subconsulta ens permet filtrar les empreses alemanyes i obtenir-ne els IDs corresponents
5 • SELECT *
6    FROM transaction AS t2
7  WHERE company_id IN (
8    SELECT id
9      FROM company AS t1
10     WHERE t2.company_id = t1.id
11       AND t1.country = 'Germany'
12 );
13
14  -- Aquí seleccionem totes les transaccions de les empreses alemanyes fent servir INNER JOIN
15  -- L'ús d'INNER JOIN ens permet unir les taules i aplicar el filtre per país
16 • SELECT *
17    FROM transaction AS t2
```

Result Grid									
	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-A387-44F8-729FB4851C76	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-C8AC4FD92501	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-D35639248539	CdU-4359	b-2302	221	-56.4901	114.801	2022-02-28 20:33:54	430.49	0
	122DC333-E19F-D629-DCD8-9C54CF1EBB9A	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
	1717F068-ADAD-7082-A748-912BE8920CCC	CdU-4219	b-2302	172	69.4892	-138.411	2021-12-29 16:18:54	249.91	0
	1753A288-9FC1-52E6-5C39-A1FFB97B0D3A	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
	19045700-15A0-E2EE-7100-6C6001000000	CdU-1930	b-2202	177	20.0122	01.1930	2021-07-21 10:55:16	227.01	0

Veamos 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. No se pide en el enunciado, pero me apetecía investigar un poquito más sobre el código, aquí te lo dejo por si fuere de tu interés 🎉



The screenshot shows a MySQL Workbench interface. The top pane contains a SQL query:

```
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 bottom pane displays the results in a grid:

	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	58.16	Nunc Interdum Incorporated	Germany

Result 30 x

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.

El SQL selecciona los nombres de las empresas (company_name) que cumplen una condición específica relacionada con sus transacciones.

SELECT company_name FROM company AS t1: Esta línea selecciona los nombres de las empresas de la tabla company y las nombra como t1. Esto permite referirse a esta tabla con el nombre más corto t1 en otras partes de la consulta.

WHERE t1.id IN (...): Esta condición restringe los resultados a las empresas que cumplen un criterio específico relacionado con sus transacciones.

SELECT company_id FROM transaction AS t2 WHERE t2.amount > (...): Esta subconsulta selecciona los identificadores (company_id) de las transacciones de la tabla transaction donde el importe (amount) de la transacción es superior a la media de todas las transacciones.

SELECT AVG(amount) FROM transaction AS t2: Esta subconsulta calcula la media de todos los importes de las transacciones de la tabla transaction.

Por lo tanto, en resumen, esta consulta SQL selecciona los nombres de las empresas que tienen transacciones con importes superiores a la media de todas las transacciones.

```
22  -- SPRINT 2, NIVELL 1, EXERCICI 2: Empreses amb transaccions amb una quantitat superior a la mitjana de totes les transaccions
23
24  -- En aquest cas, seleccionem les empreses amb transaccions que tenen una quantitat superior a la mitjana de totes les transaccions
25  -- Utilitzem una subconsulta per calcular aquesta mitjana i filtrar les empreses corresponents
26 •   SELECT company_name
27   FROM company AS t1
28   WHERE t1.id IN (
29     SELECT company_id
30     FROM transaction AS t2
31     WHERE t2.amount > (
32       SELECT AVG(amount)
33       FROM transaction AS t2
34     )
35   );
36
37  -- SPRINT 2, NIVELL 1, EXERCICI 3: Transaccions de les empreses amb el nom que comença per la lletra C
38
39  -- Seleccionem les transaccions de les empreses amb el nom que comença per la lletra C utilitzant INNER JOIN
40  -- Això ens permet unir les taules i aplicar el filtre pel nom de l'empresa
41 •   SELECT t1.id, company_name
42   FROM company AS t1
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
company_name				
▶	Ac Fermentum Incorporated			
	Magna A Neque Industries			
	Fusce Corp.			
	Ante Iaculis Nec Foundation			
	Donec Ltd			
	Sed Nunc Ltd			
	Nascetur Ridiculus Mus Inc.			
	Vestibulum Lorem PC			
	Gravida Sagittis LLP			
company_3				

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 database interface with a query editor at the top and a result grid below. The query editor contains the following SQL code:

```
1  # Nombre de las compañias 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;
```

The result grid displays the following data:

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	Nasceret 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

Result 18 ×

Sprint 2, Nivel 1, Ejercicio 3

Recuperar transacciones de las empresas cuyo nombre empieza por c

La solución que te propongo en SQL realiza una consulta que selecciona el identificador (id) y el nombre de la empresa (company_name) de las empresas cuyo nombre empieza por la letra 'C' y que tienen transacciones asociadas.

SELECT t1.id, company_name: Esta línea selecciona el identificador (id) de la empresa de la tabla company y el nombre de la empresa (company_name). Utiliza t1 como alias para referirse a la tabla company.

FROM company AS t1 INNER JOIN transaction AS t2 ON t1.id = t2.company_id: Esta parte de la consulta indica que se deben unir las tablas company y transaction basándose en la condición de que el id de la empresa de la tabla company (t1.id) sea igual al company_id de la tabla transaction (t2.company_id). Se utiliza un INNER JOIN para unir las filas que cumplen con la condición especificada.

WHERE company_name LIKE 'C%': Esta condición filtra los resultados para seleccionar solo aquellas empresas cuyo nombre empieza por la letra 'C'. El operador LIKE se utiliza para buscar patrones de texto y el símbolo '%' indica que puede haber cero o más caracteres después de la 'C'.

En resumen, esta consulta SQL selecciona el identificador y el nombre de las empresas que tienen transacciones asociadas y cuyo nombre empieza por la letra 'C'.

```
36
37      -- SPRINT 2, NIVELL 1, EXERCICI 3: Transaccions de les empreses amb el nom que comença per la lletra C
38
39      -- Seleccionem les transaccions de les empreses amb el nom que comença per la lletra C utilitzant INNER JOIN
40      -- Això ens permet unir les taules i aplicar el filtre pel nom de l'empresa
41 •   SELECT t1.id, company_name
42     FROM company AS t1
43     INNER JOIN transaction AS t2
44     ON t1.id = t2.company_id
45     WHERE company_name LIKE 'C%';
46
47      -- També podem obtenir el mateix resultat amb un INNER JOIN i una subconsulta
48 •   SELECT company_name, t2.*
49     FROM transaction AS t2
50     INNER JOIN company AS t1
51     ON t2.company_id = t1.id
52     WHERE t2.company_id IN (
53       SELECT t1.id
54       FROM company AS t1
```

Result Grid	
	Filter Rows:
	Export:
	Wrap Cell Content:
id	company_name
b-2234	Convallis In Incorporated
b-2234	Convallis In Incorporated
b-2514	Cras Consulting
b-2514	Cras Consulting
b-2554	Cras Vehicula Aliquet Industries
b-2554	Cras Vehicula Aliquet Industries

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.

Este código SQL realiza una consulta que selecciona el nombre de las empresas (company_name) que no tienen ninguna transacción asociada.

SELECT company_name FROM company AS t1: Esta línea selecciona el nombre de la empresa de la tabla company y utiliza t1 como alias para referirse a esta tabla.

WHERE NOT EXISTS (...): Esta parte de la consulta utiliza la cláusula WHERE NOT EXISTS para filtrar los resultados. Esto significa que la condición especificada no debe existir para que se incluya una fila en el resultado.

SELECT company_id FROM transaction AS t2 WHERE t1.id = t2.company_id: Esta subconsulta selecciona los identificadores (company_id) de las transacciones de la tabla transaction donde el id de la empresa de la tabla company (t1.id) coincide con el company_id de la tabla transaction (t2.company_id). Esto significa que seleccionará todas las empresas que tienen al menos una transacción asociada.

Por lo tanto, en resumen, esta consulta SQL seleccionará el nombre de las empresas que no tienen ninguna transacción asociada. Es decir, mostrará aquellas empresas que no tienen registros en la tabla de transacciones.

```
58  -- SPRINT 2, NIVELL 1, EXERCICI 4: Llista d'empreses que estan a la taula de companies però que no tenen cap transacció
59
60  -- Aquí obtenim les empreses que no tenen cap transacció fent servir NOT EXISTS
61  -- Això ens permet comprovar l'absència de registres associats a les empreses
62 • SELECT company_name
63  FROM company AS t1
64  WHERE NOT EXISTS (
65      SELECT company_id
66      FROM transaction AS t2
67      WHERE t1.id = t2.company_id
68  );
69
70  -- Alternativament, podem fer servir INNER JOIN i WHERE per trobar empreses sense transaccions
71 • SELECT *
72  FROM company AS t1
73  INNER JOIN transaction AS t2
74  ON t1.id = t2.company_id
75  WHERE t2.id IS NULL; -- No té transaccions a la taula de transaccions
```

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

Sprint 2, Nivel 2, Ejercicio 1

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

Este script SQL realiza una consulta que selecciona todas las columnas de la tabla transaction (t) para aquellas transacciones asociadas a empresas ubicadas en el mismo país que la empresa llamada 'Non Institute'.

`SELECT * FROM transaction AS t;` Esta línea selecciona todas las columnas de la tabla transaction y la nombra como t.

`WHERE company_id IN (...);` Esta condición restringe los resultados a las transacciones que están asociadas a empresas que cumplen ciertos criterios.

`SELECT id FROM company AS c WHERE country IN (...);` Esta subconsulta selecciona los identificadores (id) de las empresas de la tabla company donde el país de la empresa está incluido en la lista de países donde hay empresas llamadas 'Non Institute'.

`SELECT country FROM company AS c WHERE company_name = 'Non Institute';` Esta subconsulta obtiene los países donde hay empresas llamadas 'Non Institute'.

Por lo tanto, en resumen, esta consulta SQL seleccionará todas las transacciones que estén asociadas a empresas ubicadas en los mismos países donde hay empresas llamadas 'Non Institute'.

```
87      -- SPRINT 2, NIVELL 2, EXERCICI 1: Llista de transaccions del mateix pais de l'empresa anomenada Non Institute
88
89      -- En aquest cas, seleccionem les transaccions del mateix pais que 'Non Institute' utilitzant INNER JOIN i una subconsulta
90 •  SELECT *
91     FROM transaction AS t
92     WHERE company_id IN (
93         SELECT id
94         FROM company AS c
95         WHERE country IN (
96             SELECT country
97             FROM company AS c
98             WHERE company_name = 'Non Institute'
99         )
100    );
101
102      -- Alternativament, podem fer servir INNER JOIN i una subconsulta per trobar les transaccions del mateix pais que 'Non Institute'
103 •  SELECT c.country, t.*
104     FROM transaction AS t
105     JOIN company AS c
106     ON c.id = t.company_id
107     WHERE country = (
```

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-C365-41E1-2AB00C65147A	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
	1479B3D-2-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-C5453DAFD3B	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
	23988576-6C0E-137A-C2F6-3180A188A2D3	CcU-2994	b-2326	126	23.6174	137.222	2021-08-26 06:04:05	43.90	0

Sprint 2, Nivel 2, Ejercicio 2

Buscamos la empresa que ha realizado la mayor transacción

Este solución SQL realiza una consulta que selecciona el nombre de la empresa (company_name) y la suma total de las cantidades de todas las transacciones asociadas a cada empresa, mostrando solo la empresa con la mayor suma total de transacciones.

SELECT c.company_name, SUM(amount) AS parcial_empresa: Esta línea selecciona el nombre de la empresa de la tabla company y calcula la suma total de las cantidades de las transacciones asociadas a cada empresa. La palabra clave AS se utiliza para renombrar la suma total como parcial_empresa.

FROM transaction AS t JOIN company AS c ON c.id = t.company_id: Esta parte de la consulta indica que se deben combinar las filas de las tablas transaction y company basándose en la condición de que el id de la empresa de la tabla company (c.id) sea igual al company_id de la tabla transaction (t.company_id).

GROUP BY t.company_id: Esto agrupa los resultados por el company_id de las transacciones para que se realice la suma total por empresa.

ORDER BY parcial_empresa DESC: Ordena los resultados en orden descendente según la suma total de las cantidades de las transacciones (parcial_empresa), lo que significa que la empresa con la mayor suma total aparecerá primero.

LIMIT 1: Esto limita el resultado a una sola fila, mostrando solo la empresa con la mayor suma total de transacciones.

Por lo tanto, en resumen, esta consulta SQL seleccionará el nombre de la empresa y la suma total de las cantidades de todas las transacciones asociadas a cada empresa, mostrando solo la empresa con la mayor suma total de transacciones.

```
113 -- SPRINT 2, NIVELL 2, EXERCICI 2: Nom de l'empresa amb la transacció de major quantitat
114
115 -- Aquí obtenim el nom de l'empresa amb la transacció de major quantitat utilitzant INNER JOIN, SUM i ORDER BY
116 • SELECT c.company_name, SUM(amount) AS parcial_empresa
117   FROM transaction AS t
118   JOIN company AS c
119   ON c.id = t.company_id
120   GROUP BY t.company_id
121   ORDER BY parcial_empresa DESC
122   LIMIT 1;
123
124 -- Alternativament, podem obtenir el mateix resultat utilitzant INNER JOIN, SUM i una subconsulta
125 • SELECT company_name, subtotales.parcial_empresa
126   FROM company AS c
127   JOIN (
128     SELECT SUM(amount) AS parcial_empresa, company_id
129     FROM transaction AS t
130     GROUP BY t.company_id
131   ) AS subtotales
```

Result Grid	
company_name	parcial_empresa
Nunc Interdum Incorporated	25622.65

Sprint 2, Nivel 3, Ejercicio 1

Paises cuyas ventas en cantidad de operaciones sea superior a la venta media de todos los países

Este código SQL realiza una consulta que cuenta el número de transacciones (COUNT(t.id)) para cada país (c.country), mostrando solo aquellos países donde el número de transacciones es mayor que el promedio global de transacciones por país.

SELECT c.country, COUNT(t.id) AS contador: Esta línea selecciona el país (c.country) y cuenta el número de transacciones (COUNT(t.id)) para cada país, renombrándolo como contador.

FROM transaction AS t INNER JOIN company AS c ON c.id = t.company_id: Esto indica que se deben combinar las filas de las tablas transaction y company basándose en la condición de que el id de la empresa de la tabla company (c.id) sea igual al company_id de la tabla transaction (t.company_id).

GROUP BY c.country: Agrupa los resultados por país para que se cuente el número de transacciones por país.

HAVING contador > (...): Esta cláusula filtra los resultados para incluir solo aquellos países donde el número de transacciones (contador) es mayor que el promedio global de transacciones por país.

SELECT AVG(contador) FROM (...) AS media_global: Esto calcula el promedio global de transacciones por país. Se realiza mediante una subconsulta que cuenta el número de transacciones por país y luego calcula el promedio.

ORDER BY contador DESC: Ordena los resultados en orden descendente según el número de transacciones (contador), lo que significa que los países con más transacciones aparecerán primero.

En resumen, esta consulta SQL cuenta el número de transacciones para cada país y muestra solo aquellos países donde el número de transacciones es mayor que el promedio global de transacciones por país, ordenados por el número de transacciones en orden descendente.

```
136 -- SPRINT 2, NIVELL 3, EXERCICI 1: Llista de països amb vendes agregades superiors a la mitjana de vendes
137
138 -- Aquí obtenim els països amb vendes agregades superiors a la mitjana utilitzant INNER JOIN, COUNT i HAVING
139 •   SELECT c.country, COUNT(t.id) AS contador
140     FROM transaction AS t
141     INNER JOIN company AS c
142     ON c.id = t.company_id
143     GROUP BY c.country
144     HAVING contador > (
145       SELECT AVG(contador)
146       FROM (
147         SELECT COUNT(t.id) AS contador
148         FROM transaction AS t
149         INNER JOIN company AS c
150         ON c.id = t.company_id
151         GROUP BY c.country
152       ) AS media_global
153     )
154     ORDER BY contador DESC;
155
```

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

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

Este programa SQL realiza una consulta que cuenta el número de transacciones para cada empresa (company_id) y asigna un caso ('Grande + 4' o 'Pequeña -4') según si el número de transacciones es mayor que 4 o no.

SELECT company_id, company_name, COUNT(t.id) AS contador: Esta línea selecciona el identificador de la empresa (company_id), el nombre de la empresa (company_name) y cuenta el número de transacciones (COUNT(t.id)), renombrándolo como contador.

CASE WHEN COUNT(t.id) > 4 THEN 'Grande + 4' ELSE 'Pequeña -4' END AS casos: Esta parte del código es una expresión condicional (CASE) que evalúa si el número de transacciones es mayor que 4. Si es así, asigna el valor 'Grande + 4', de lo contrario, asigna 'Pequeña -4'. Esto clasifica las empresas según la cantidad de transacciones.

FROM transaction AS t JOIN company AS c ON c.id = t.company_id: Esto indica que se deben combinar las filas de las tablas transaction y company basándose en la condición de que el id de la empresa de la tabla company (c.id) sea igual al company_id de la tabla transaction (t.company_id).

GROUP BY company_id: Agrupa los resultados por company_id para contar el número de transacciones por empresa.

ORDER BY contador DESC: Ordena los resultados en orden descendente según el número de transacciones (contador), lo que significa que las empresas con más transacciones aparecerán primero.

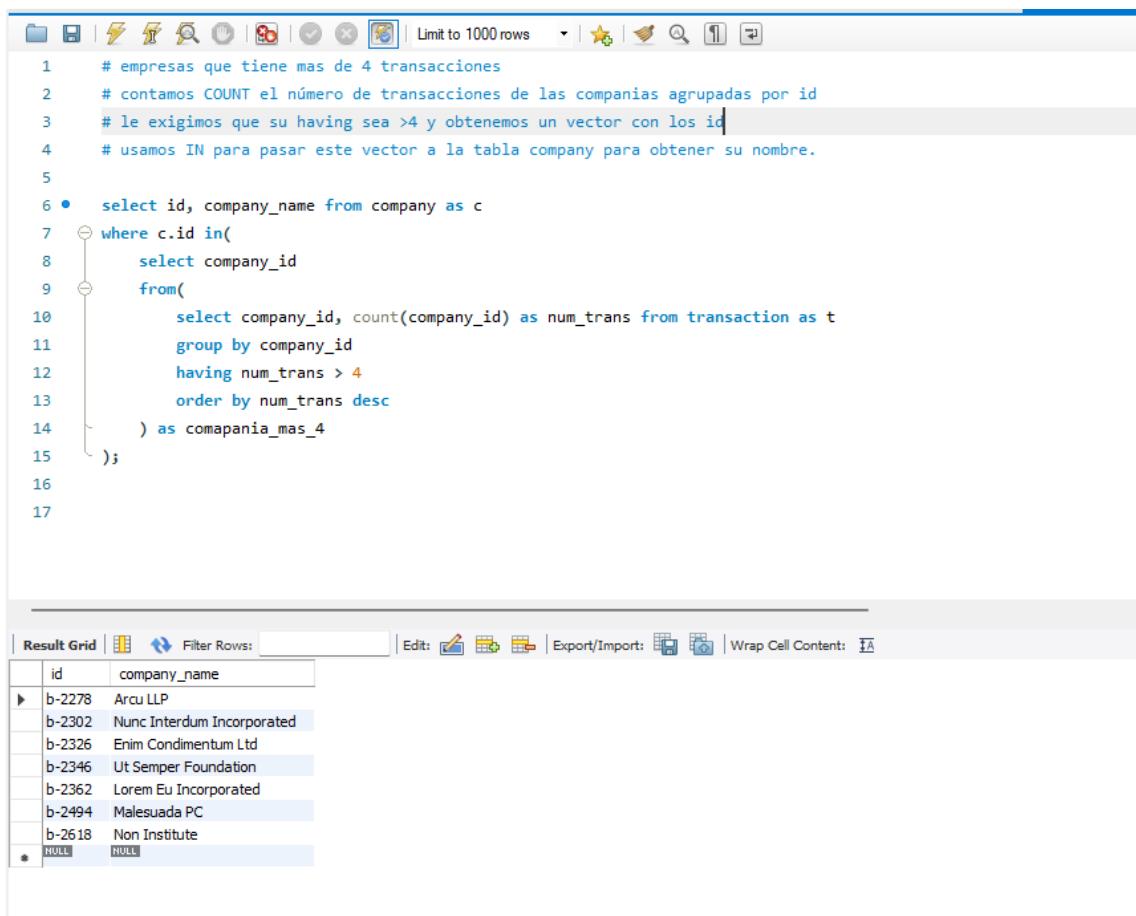
En resumen, esta consulta SQL cuenta el número de transacciones para cada empresa y clasifica las empresas como 'Grande + 4' si tienen más de 4 transacciones y como 'Pequeña -4' si tienen 4 o menos transacciones, ordenadas por el número de transacciones en orden descendente.

```
202
203 -- SPRINT 2, NIVELL 3, EXERCICI 3: Llista d'empreses amb més o menys de 4 transaccions
204
205 -- En aquesta consulta, obtenim una llista d'empreses amb el seu nombre de transaccions.
206 -- S'utilitza una sentència CASE per etiquetar les empreses com a 'Grande + 4' o 'Pequeña -4' basant-se en el nombre de transaccions.
207 -- Els resultats s'ordenen descentendent pel nombre de transaccions.
208 • select company_id, company_name, count(t.id )as contador,
209   case
210     when count(t.id ) > 4 then 'Grande + 4'
211     else 'Pequeña -4'
212   end as casos
213   from transaction as t
214   join company as c
215   on c.id = t.company_id
216   group by company_id
217   order by contador desc;
218
219
```

company_id	company_name	contador	casos
b-2302	Nunc Interdum Incorporated	105	Grande + 4
b-2346	Ut Semper Foundation	59	Grande + 4
b-2326	Erim Condimentum Ltd	57	Grande + 4
b-2278	Arcu LLP	56	Grande + 4
b-2362	Lorem Eu Incorporated	54	Grande + 4
b-2494	Malesuada PC	52	Grande + 4
b-2618	Non Institute	30	Grande + 4
b-2230	Fusce Corp.	2	Pequeña -4
b-2234	Convallis In Incorporated	2	Pequeña -4

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. No se pide en el enunciado, pero me apetecía investigar un poquito más sobre el código, aquí te lo dejo por si fuere de tu interés 🙌



The screenshot shows a MySQL Workbench interface. The top part is a code editor with the following SQL query:

```
1  # empresas que tiene mas de 4 transacciones
2  # contamos COUNT el número de transacciones de las compañias 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 comapania_mas_4
15    );
16
17
```

The bottom part is a result grid titled "Result Grid" showing the output of the query:

	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