

Contenido

NIVEL 1	2
Ejercicio 1.....	2
Ejercicio 2.....	5
Ejercicio 3.....	6
Ejercicio 4.....	7
Ejercicio 5.....	8
Ejercicio 6.....	9
NIVEL 2	10
Ejercicio 1.....	10
Ejercicio 2.....	11
Ejercicio 3.....	12
Ejercicio 4.....	13
NIVEL 3	14
Ejercicio 1.....	14
Ejercicio 2.....	15

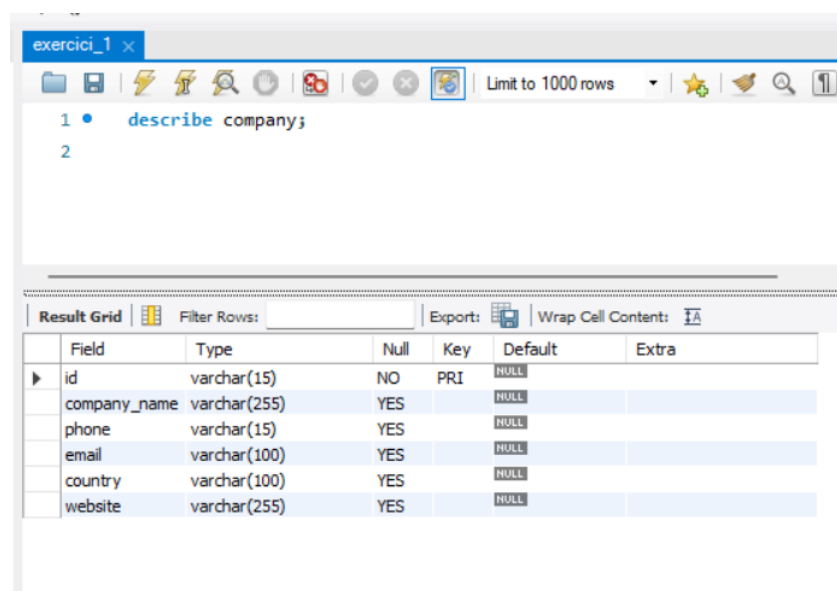
NIVEL 1

Ejercicio 1.

A partir de los documentos adjuntos (estructura_dades y dades_introduir), importa las dos tablas. Muestra las características principales del esquema creado y explica las diferentes tablas y variables que existen. Asegúrate de incluir un diagrama que ilustre la relación entre las diferentes taulas y variables.

La base de datos Transactions se ha creado para almacenar y gestionar las transacciones realizadas por un conjunto de 100 empresas. Para ello consta de dos tablas: 'company' donde se almacenan los datos de las empresas; y 'transactions' donde se almacenan los datos de las transacciones.

Mediante la orden 'describe company;' se nos muestra la estructura de la tabla 'company' que consta de seis campos con el mismo tipo de datos (varchar).

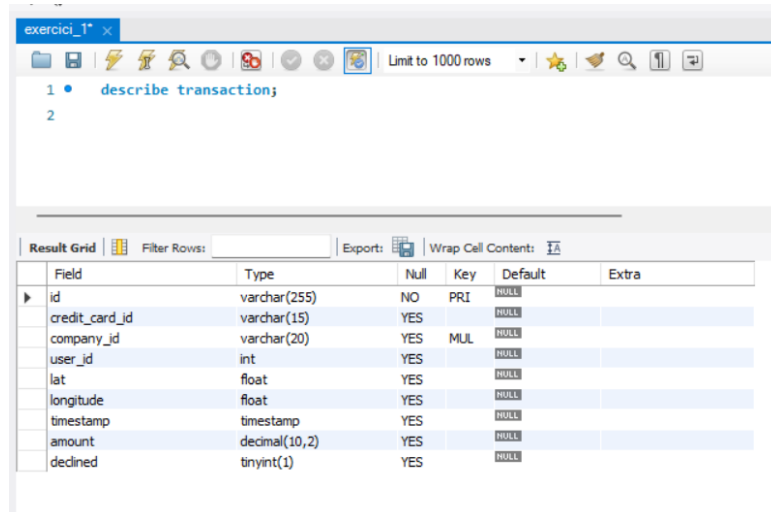


The screenshot shows a database management interface with a command window and a result grid. The command window displays the SQL command 'describe company;' and the result grid shows the structure of the 'company' table.

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	NULL	
company_name	varchar(255)	YES		NULL	
phone	varchar(15)	YES		NULL	
email	varchar(100)	YES		NULL	
country	varchar(100)	YES		NULL	
website	varchar(255)	YES		NULL	

Consta de una clave primaria (campo 'id') que no admite valores nulos. El resto de campos son el nombre de la compañía (company_name), el teléfono (phone), correo electrónico ('email'), el país de procedencia ('country') y su página web ('website'). La longitud del tipo de datos varía en función de las necesidades de cada campo y todos ellos pueden ser valores nulos.

De forma análoga a la anterior obtenemos una vista de la estructura de la segunda tabla, 'transaction', que consta de nueve campos con diferentes tipos de datos.



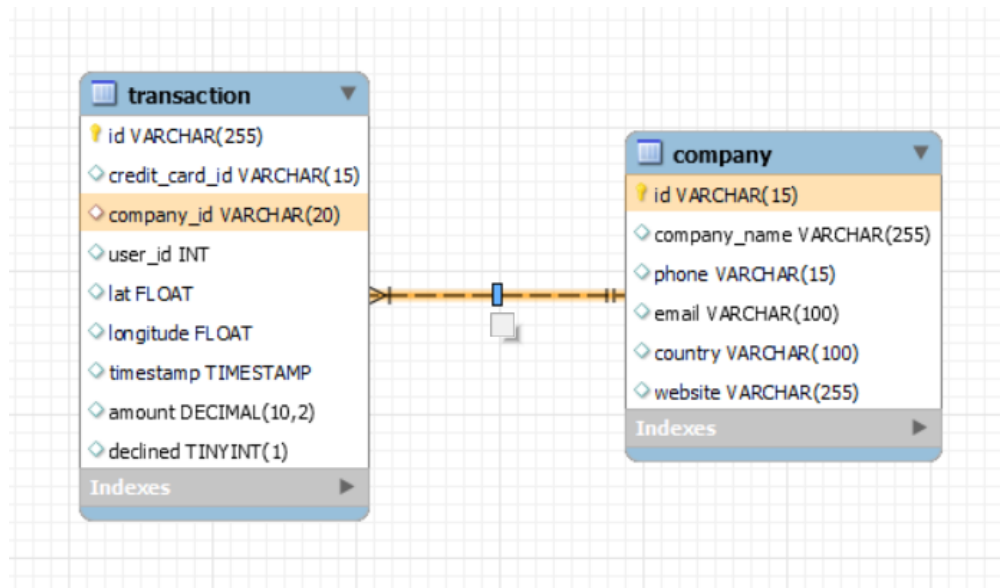
The screenshot shows a database interface with a command prompt at the top containing the SQL command `describe transaction;`. Below the command prompt is a 'Result Grid' table showing the structure of the 'transaction' table.

Field	Type	Null	Key	Default	Extra
id	varchar(255)	NO	PRI	NULL	
credit_card_id	varchar(15)	YES		NULL	
company_id	varchar(20)	YES	MUL	NULL	
user_id	int	YES		NULL	
lat	float	YES		NULL	
longitude	float	YES		NULL	
timestamp	timestamp	YES		NULL	
amount	decimal(10,2)	YES		NULL	
declined	tinyint(1)	YES		NULL	

Contiene una clave primaria 'id' para cada transacción y una clave foránea 'company_id' de tipo múltiple (MUL) que la une a la tabla 'company' mediante una cadena de caracteres (varchar).

La tabla consta de campos de identificación de usuario ('user_id') almacenado mediante un número entero; y de la tarjeta de crédito ('credit_car_id') almacenada mediante caracteres. Georeferencia la transacción mediante dos campos para latitud ('lat') y longitud ('longitude') con datos tipo FLOAT correspondientes a coordenadas decimales. Fija el momento de la transacción mediante el campo 'timestamp' que proporciona un valor de ese tipo y proporciona la cantidad de la operación mediante el campo 'amount' usando el tipo DECIMAL. Finalmente, incluye un campo 'declined' de tipo TINYINT con dos valores (0 y 1).

La relación de las tablas se hace a través de la clave principal 'id' de la tabla 'company' y del campo 'company_id' de la tabla 'transaction' con una clave foránea múltiple.



Ejercicio 2.

Realiza la siguiente consulta: has de obtener el nombre, email y país de cada compañía. Ordena los datos en función del nombre de las compañías.

Mediante un SELECT especificamos los campos a mostrar y les asignamos un alias de la tabla 'company'. Luego ordenamos la selección mediante un ORDER BY por el nombre de la compañía ('company_name') en orden ascendente (ASC).

exercici_2 x

```

1 • select company_name as Nom,
2   email as email,
3   country as País
4   from company
5   order by company_name asc;

```

Limit to 1000 rows

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

	Nom	email	País
▶	A Institute	metus.aliquam@google.edu	Belgium
	Ac Fermentum Incorporated	donec.porttitor.tellus@yahoo.net	Germany
	Ac Industries	ipsum@yahoo.com	Germany
	Ac Libero Inc.	mollis.lectus@protonmail.ca	United Kingdom
	Aliquam Erat Volutpat LLP	pede.nunc@icloud.net	Italy
	Aliquam Iaculis Lacus Corp.	dictum@aol.org	Belgium
	Aliquam PC	scelerisque.mollis@icloud.org	Germany
	Aliquet Diam Limited	eu.eros.nam@icloud.org	United States
	Aliquet Sem Limited	sem.magna@yahoo.edu	Netherlands
	Aliquet Vel Vulputate Incorporated	suspendisse.commodo@outlook.couk	Netherlands
	Amet Faucibus Ut Foundation	malesuada@icloud.net	United Kingdom
	Amet Institute	nullam.lobortis.quam@outlook.net	Australia
	Amet Lorem LLP	ultrices.adipiscing@icloud.edu	Spain
	Amet Luctus Vulputate Foundation	nulla@outlook.com	Canada
	Amet Nulla Donec Corporation	mattis.integer.eu@protonmail.net	Italy
	Ante Iaculis Nec Foundation	sed.dictum.proin@outlook.ca	New Zealand
	Arcu LLP	dui@aol.ca	Norway
	At Associates	tristique.neque@yahoo.couk	New Zealand
	At Pedo Corp.	ac.mi.eleifend@hotmail.edu	Italy
	Auctor Mauris Corp.	eget.metus@protonmail.ca	Germany
	Auctor Mauris Vel LLP	nec.tempus@icloud.couk	United States
	Augue Foundation	mauris@yahoo.com	Germany
	Convallis In Incorporated	mauris.ut@aol.couk	Germany
	Cras Consulting	sed.consequat@google.ca	Belgium
	Cras Vehicula Aliquet Industries	arcu@hotmail.org	Netherlands

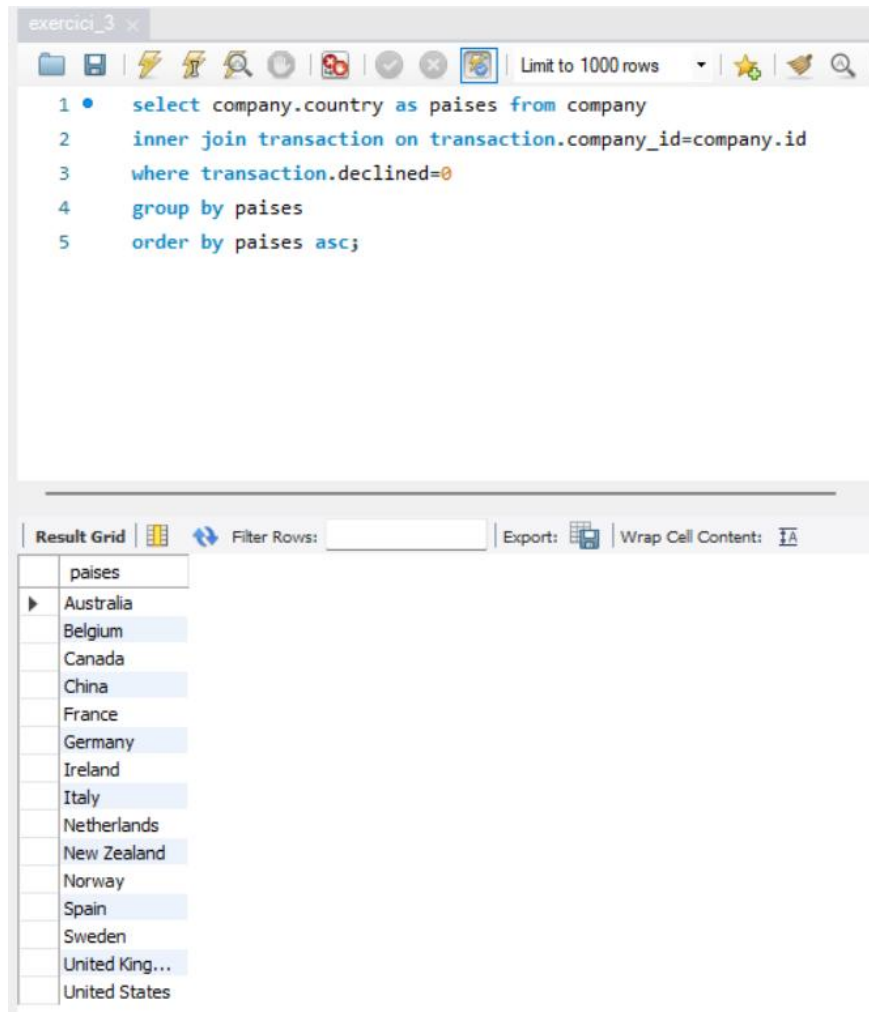
company 1 x

Ejercicio 3.

Desde la sección de márketing nos solicitan que les pasemos un listado de los países que están realizando compras.

Para ofrecer este listado necesitamos combinar dos campos: el nombre de los países de la tabla 'company'; y el de las transacciones realizadas, que se encuentra en la tabla 'transaction' en el campo 'declined' con valor 0, que indica aquellas operaciones realizadas.

Para ello seleccionamos el campo 'country' de la tabla 'company' y le asignamos un alias (países). Conectamos con la tabla 'transaction' con un INNER JOIN usando la clave primaria y la foránea, respectivamente. Mediante un WHERE seleccionamos aquellas transacciones realizadas (es decir, aquellas registradas en el campo 'declined' con valor 0). Las agrupamos y ordenamos de forma ascendente por el campo 'países'



The screenshot shows a SQL IDE window titled 'ejercicio_3'. The query editor contains the following SQL code:

```

1 • select company.country as países from company
2   inner join transaction on transaction.company_id=company.id
3   where transaction.declined=0
4   group by países
5   order by países asc;

```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are as follows:

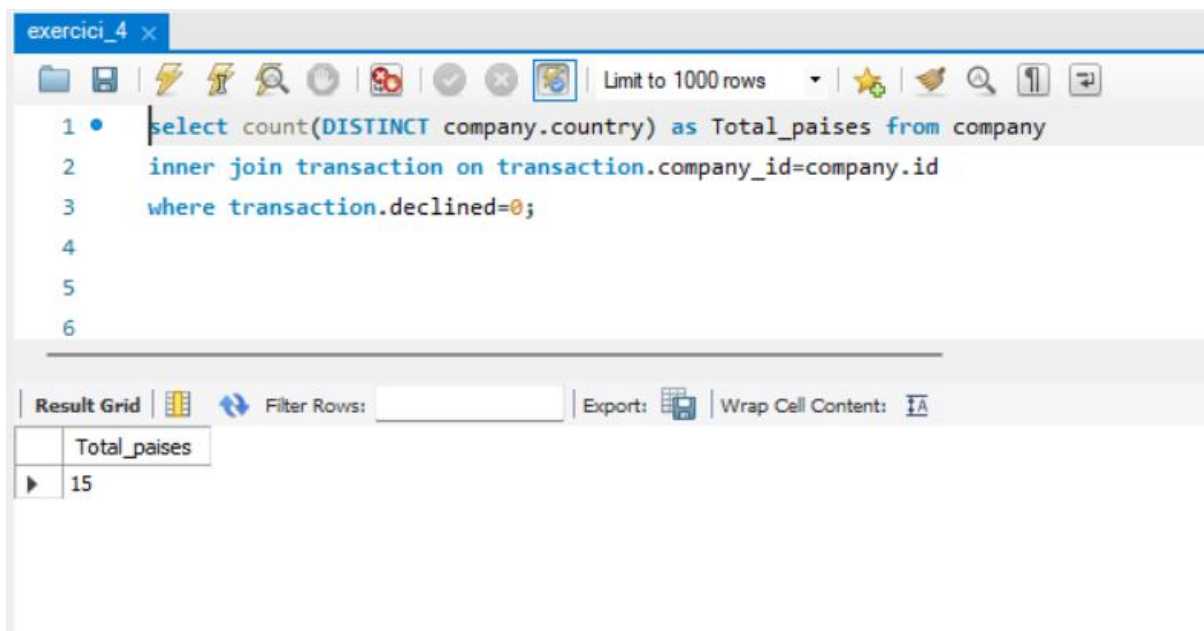
países
Australia
Belgium
Canada
China
France
Germany
Ireland
Italy
Netherlands
New Zealand
Norway
Spain
Sweden
United King...
United States

Ejercicio 4.

Desde márketing también quieren saber desde cuantos países se realizan las compras.

Para ofrecer el total de países que realizan compras debemos contabilizar sólo los países a partir de las operaciones realizadas. Para ello usaremos la cláusula COUNT para contabilizar los países de la tabla 'company' que tengan operaciones en la tabla 'transaction' con un valor 0 en el campo 'declined'.

Sin embargo, para evitar que nos proporcione una suma de las operaciones realizadas por países debemos utilizar el SELECT DISTINCT para que el conteo elimine los valores repetidos en las operaciones (en este caso el campo de países) y muestre solo contabilice los valores de forma unificada.



The screenshot shows a SQL IDE window titled 'ejercicio_4'. The query editor contains the following SQL code:

```
1 select count(DISTINCT company.country) as Total_paises from company
2 inner join transaction on transaction.company_id=company.id
3 where transaction.declined=0;
4
5
6
```

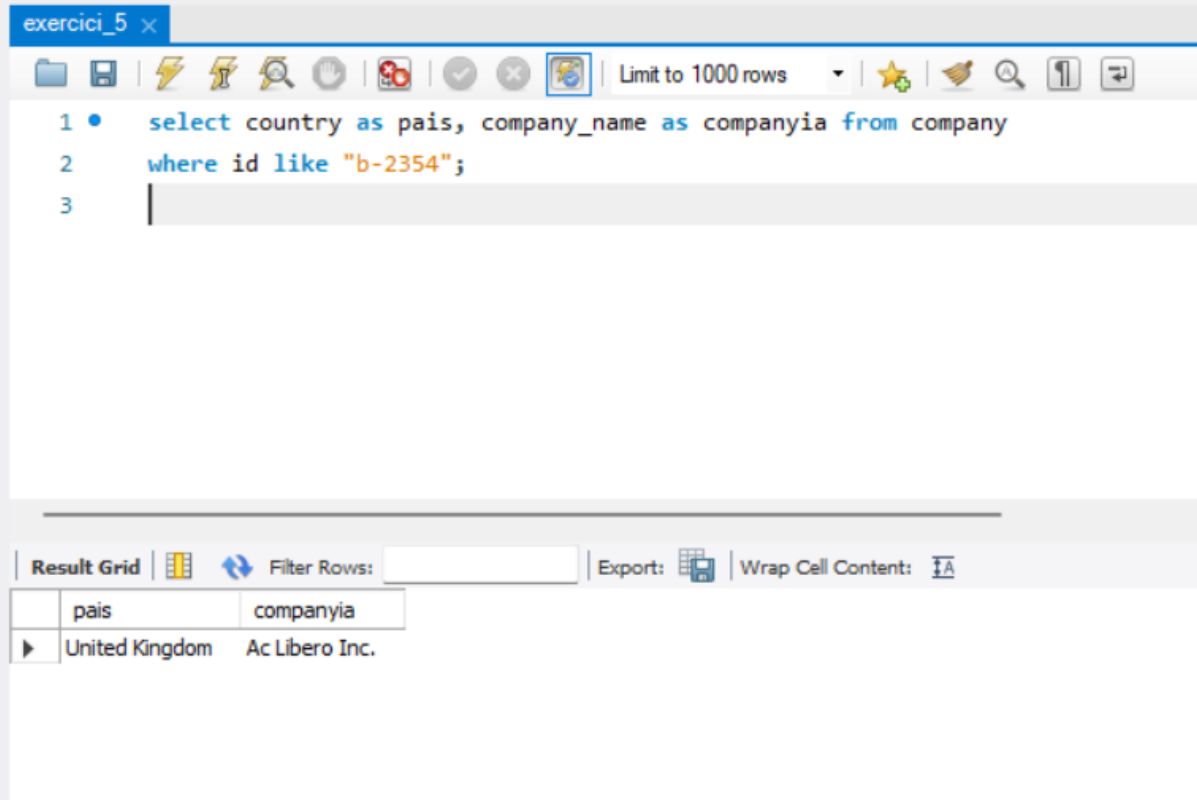
The interface includes a toolbar with icons for file operations, a 'Limit to 1000 rows' dropdown, and a 'Result Grid' section. The 'Result Grid' shows the following output:

Total_paises
15

Ejercicio 5.

Tu jefe identifica un error con la compañía que tiene identificador 'b-2354'. Por lo tanto, te solicita que le indiques el país y le nombre de compañía de este identificador.

Para aislar el identificador, en principio bastaría que, una vez señalados los campos a mostrar en el SELECT, especificásemos el identificador en la sentencia WHERE (WHERE id = b-2354). Sin embargo, la presencia del carácter '-' genera un error. Para evitarlo usamos LIKE en lugar del operador de asignación (=) y el identificador entre comillas, de forma que la consulta nos devuelva lo solicitado.



The screenshot shows a SQL IDE window titled 'exercici_5'. The query editor contains the following SQL code:

```
1 • select country as pais, company_name as companyia from company
2   where id like "b-2354";
3
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are as follows:

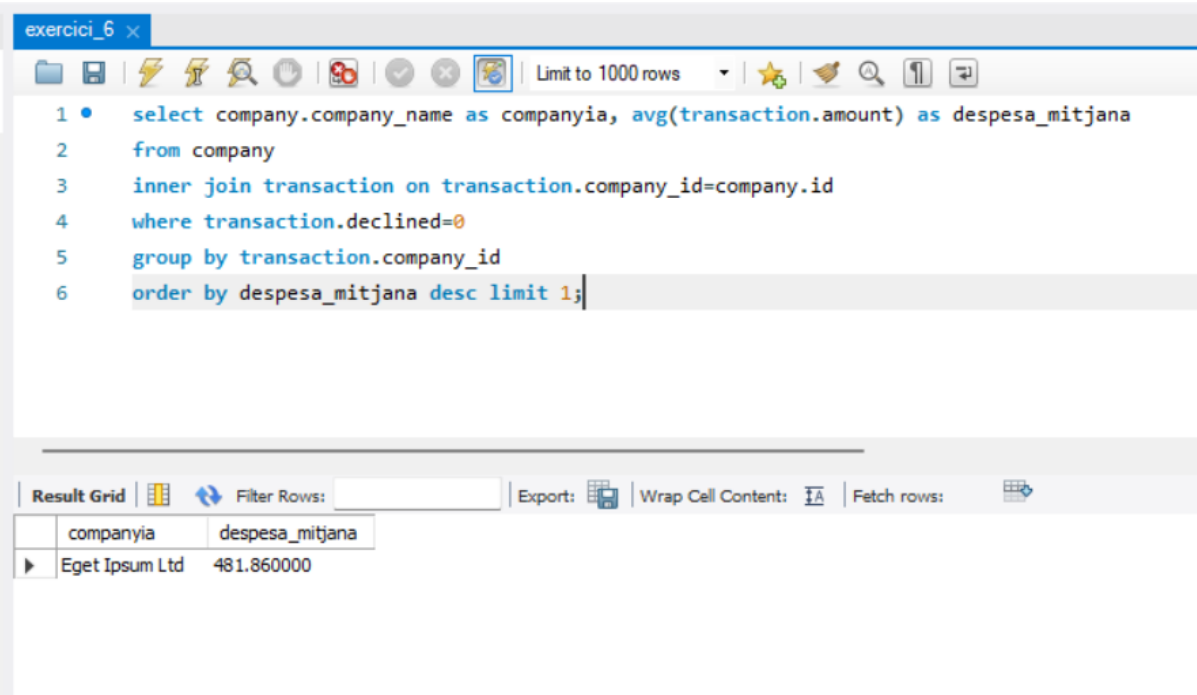
	pais	companyia
▶	United Kingdom	Ac Libero Inc.

Ejercicio 6.

Además tu jefe te solicita que indiques cual es la compañía con mayor gasto medio.

Para identificar la compañía con mayor gasto medio necesitamos el campo 'company_name' de la tabla 'company' y la media AVG() del campo 'amount' de la tabla 'transaction'. Para ello necesitamos seleccionar ambos campos y realizar un INNER JOIN a partir de las claves primaria (transaction.company_id, company.id). Añadimos sendos alias a los campos seleccionados para una mejor comprensión de los datos presentados.

A partir del JOIN reducimos la selección a aquellas operaciones realizadas (transaction.declined=0) mediante un WHERE y agrupamos los resultados por compañías (GROUP BY transaction.company_id) ordenando por su gasto medio en orden descendente y limitando los resultados a 1, de forma que mostraremos la empresa con mayor gasto medio.



The screenshot shows a SQL IDE window titled 'exercici_6'. The query editor contains the following SQL code:

```
1 • select company.company_name as companyia, avg(transaction.amount) as despesa_mitjana
2   from company
3  inner join transaction on transaction.company_id=company.id
4  where transaction.declined=0
5  group by transaction.company_id
6  order by despesa_mitjana desc limit 1;
```

Below the query editor, the 'Result Grid' is displayed. It shows a table with two columns: 'companyia' and 'despesa_mitjana'. The first row of data is 'Eget Ipsum Ltd' with a value of '481.860000'.

companyia	despesa_mitjana
Eget Ipsum Ltd	481.860000

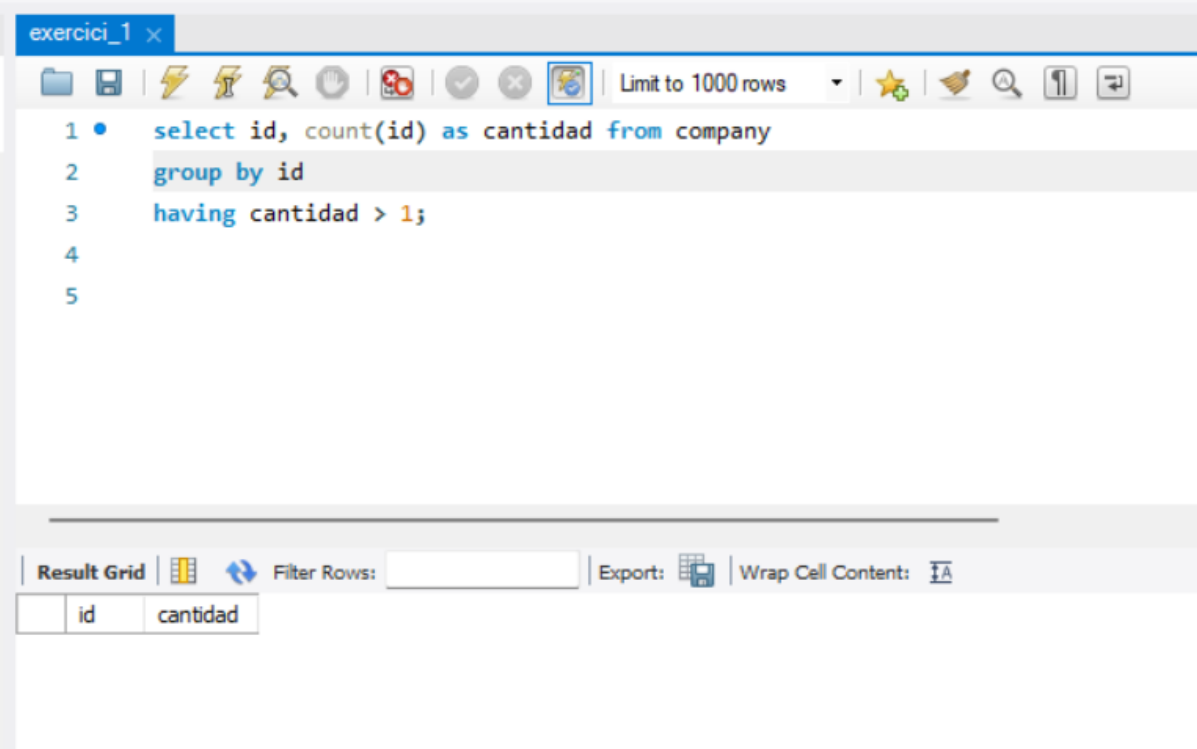
NIVEL 2

Ejercicio 1.

Tu jefe está redactando un informe de cierre del año y te solicita que le envíes información relevante para el documento. Para ello te solicita verificar si en la base de datos existen compañías con identificadores (id) duplicados.

Para obtener una lista de duplicados de los identificadores de compañía contamos todas columnas del campo 'id' de la tabla 'company'. Agrupamos el resultado por la 'id' y luego reducimos a aquellos que tienes más de un registro, es decir, que estarían repetidos.

El resultado es que no hay compañías con identificadores repetidos.



The screenshot shows a SQL query editor window titled 'exercici_1'. The query is as follows:

```
1 • select id, count(id) as cantidad from company
2   group by id
3   having cantidad > 1;
4
5
```

Below the query editor, there is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid itself is empty, showing only the column headers 'id' and 'cantidad'.

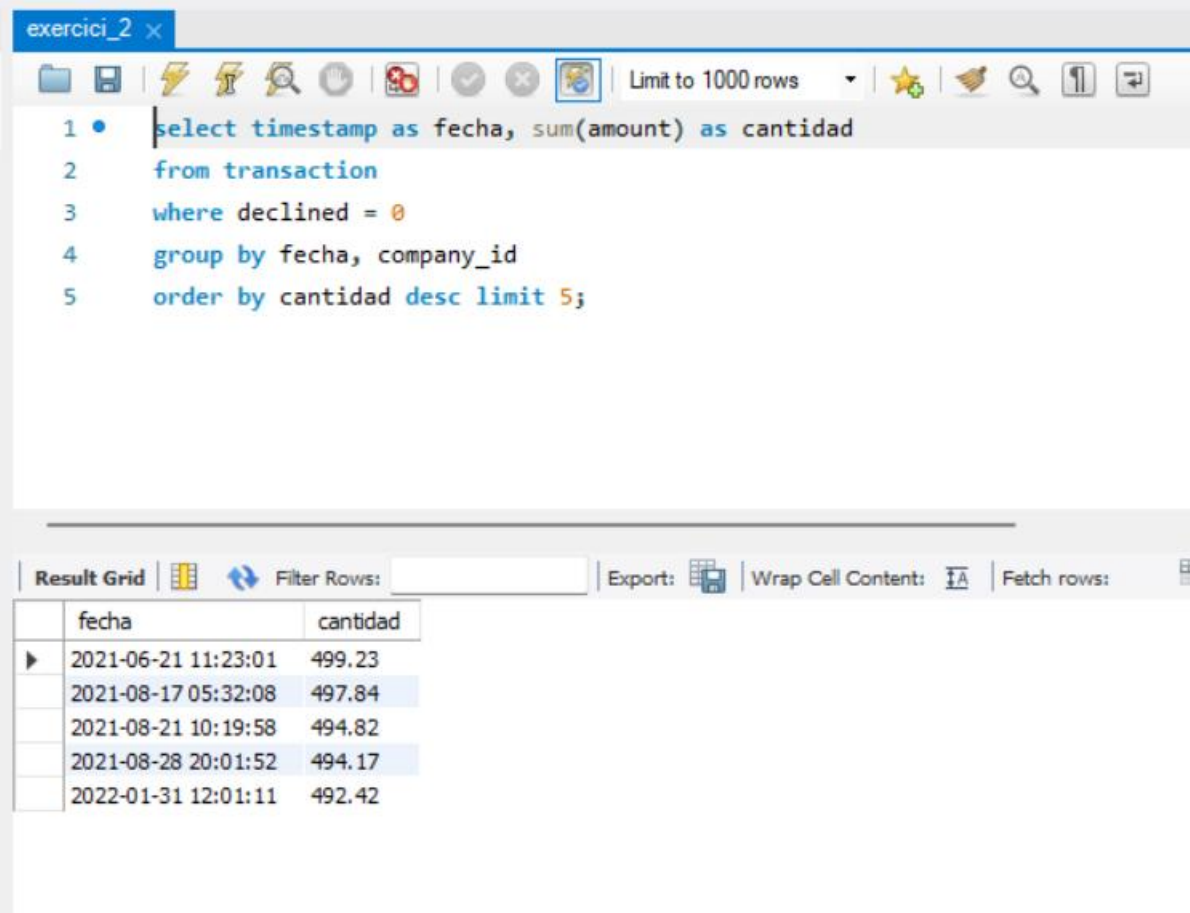
id	cantidad
----	----------

Ejercicio 2.

¿En qué día se realizaron las cinco ventas más costosas? Muestra la fecha de la transacción y la sumatoria de la cantidad de dinero.

Para obtener las cinco ventas más costosas y mostrar la fecha y el total de las mismas, seleccionamos los campos 'timestamp' y la suma del campo 'amount' de la tabla 'transaction' usando la función de reunión SUM().

Reduciremos los resultados a las operaciones realizadas (transaction.declined=0) mediante un WHERE y agruparemos por 'timestamp' –día- y por 'company_id' - compañía – ordenando los resultados por cantidad en orden descendente y limitándolo a 5, de forma que nos mostrará los días en que se realizaron esos cinco mayores gastos.



The screenshot shows a SQL IDE window titled 'ejercicio_2'. The query editor contains the following SQL code:

```
1 • select timestamp as fecha, sum(amount) as cantidad
2   from transaction
3  where declined = 0
4  group by fecha, company_id
5  order by cantidad desc limit 5;
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are shown in a table with two columns: 'fecha' and 'cantidad'.

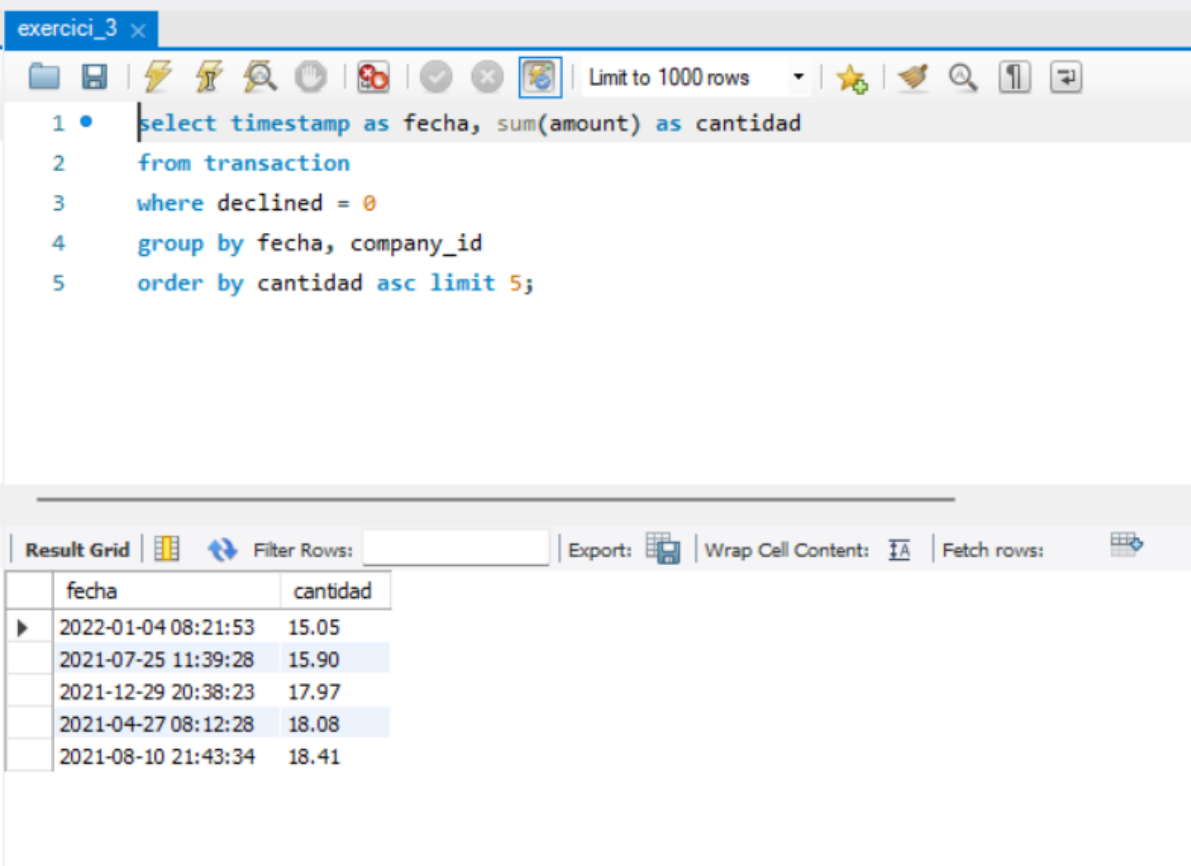
fecha	cantidad
2021-06-21 11:23:01	499.23
2021-08-17 05:32:08	497.84
2021-08-21 10:19:58	494.82
2021-08-28 20:01:52	494.17
2022-01-31 12:01:11	492.42

Ejercicio 3.

¿En qué día se realizaron las cinco ventas de menor valor? Muestra la fecha de la transacción y la sumatoria de la cantidad de dinero.

De forma análoga a la anterior seleccionamos los campos 'timestamp' y la suma del campo 'amount' de la tabla 'transaction' usando la función de reunión SUM().

Reduciremos los resultados a las operaciones realizadas (transaction.declined=0) mediante un WHERE y agruparemos por 'timestamp' –día- y por 'company_id' - compañía - ordenando los resultados por cantidad en order asc (menor a mayor) y limitándolo a 5, de forma que nos mostrará los días en que se realizaron esos cinco menores gastos.



The screenshot shows a SQL query editor window titled 'exercici_3'. The query is as follows:

```
1 select timestamp as fecha, sum(amount) as cantidad
2 from transaction
3 where declined = 0
4 group by fecha, company_id
5 order by cantidad asc limit 5;
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The grid has two columns: 'fecha' and 'cantidad'. The results are as follows:

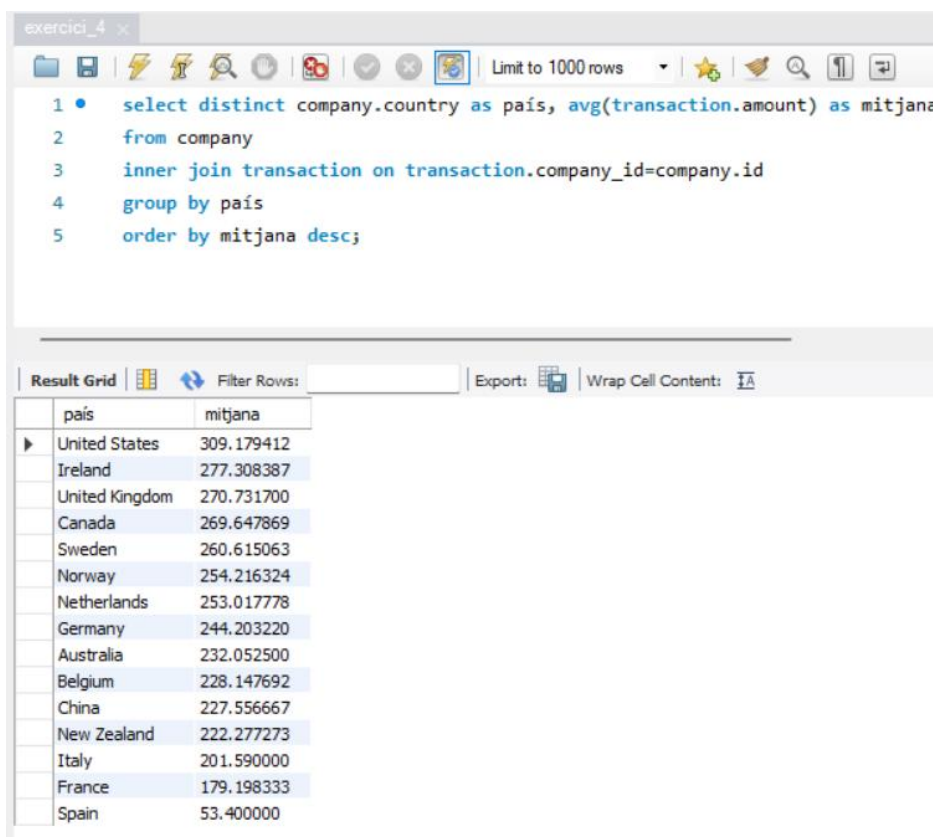
fecha	cantidad
2022-01-04 08:21:53	15.05
2021-07-25 11:39:28	15.90
2021-12-29 20:38:23	17.97
2021-04-27 08:12:28	18.08
2021-08-10 21:43:34	18.41

Ejercicio 4.

¿Cuál es la media de gasto por país? Presenta los resultados ordenados de mayor a menor.

Para ello necesitamos el país y la media de todas de todas las transacciones realizadas por las empresas de ese país. Primero seleccionamos los campos 'country' de la tabla 'company' y la media AVG() del campo 'amount' de la tabla 'transaction'. Para poder operar con ambos campos realizamos un INNER JOIN por las claves primaria y foránea respectivamente.

Posteriormente agrupamos las medias por país y ordenamos por la media en orden descendente.



The screenshot shows a SQL IDE window titled 'exercici_4'. The query editor contains the following SQL code:

```
1 • select distinct company.country as país, avg(transaction.amount) as mitjana
2   from company
3  inner join transaction on transaction.company_id=company.id
4  group by país
5  order by mitjana desc;
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are ordered by the average amount ('mitjana') in descending order. The columns are 'país' and 'mitjana'.

país	mitjana
United States	309.179412
Ireland	277.308387
United Kingdom	270.731700
Canada	269.647869
Sweden	260.615063
Norway	254.216324
Netherlands	253.017778
Germany	244.203220
Australia	232.052500
Belgium	228.147692
China	227.556667
New Zealand	222.277273
Italy	201.590000
France	179.198333
Spain	53.400000

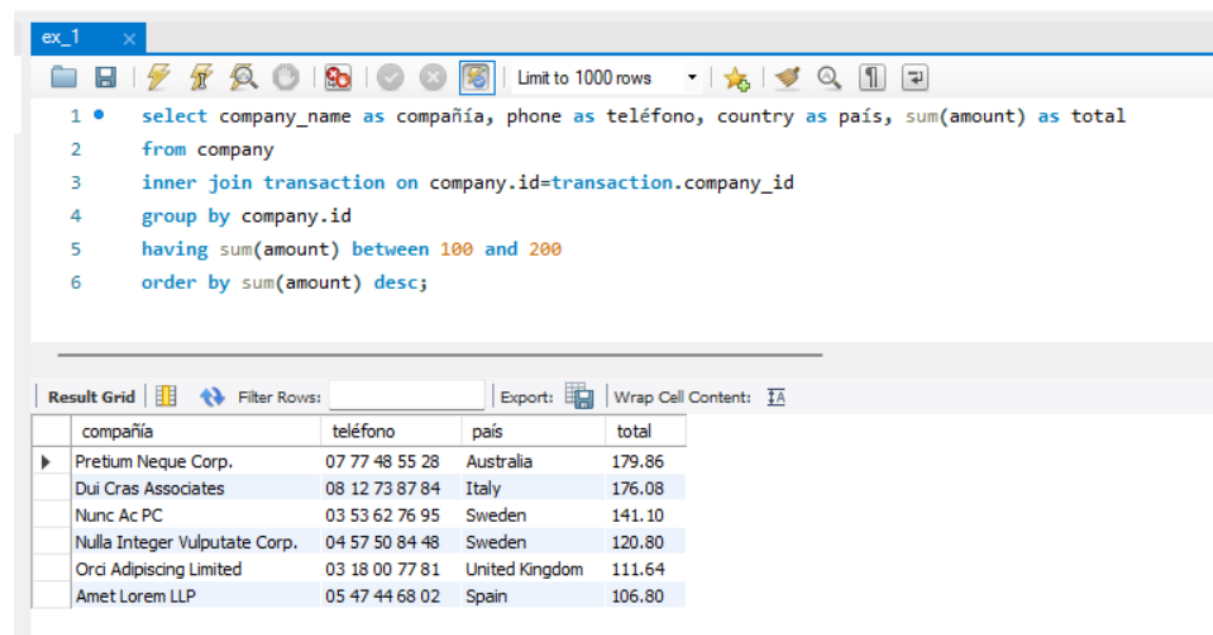
NIVEL 3

Ejercicio 1.

Presenta el nombre, teléfono y país de las compañías, junto con la cantidad total gastada, de aquellas que realizaron transacciones con un gasto comprendido entre 100 y 200 euros. Ordena los resultados de mayor a menor cantidad gastada.

Para presentar la información requerida seleccionamos los campos 'company_name', 'phone' y 'country' de la tabla 'company' y el total de las transacciones –campo 'amount', tabla 'transaction'- usando la función SUM(), por lo que realizamos un INNER JOIN para obtener todos los datos.

Agrupamos por el id de compañía de forma que tenemos los totales de cada una y especificamos la cuantía mediante un HAVING usando el operador BETWEEN para las cantidades límite especificadas. Finalmente, ordenamos el total en orden descendente.



The screenshot shows a SQL IDE window titled 'ex_1'. The query editor contains the following SQL code:

```
1 • select company_name as compañía, phone as teléfono, country as país, sum(amount) as total
2   from company
3   inner join transaction on company.id=transaction.company_id
4   group by company.id
5   having sum(amount) between 100 and 200
6   order by sum(amount) desc;
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query in a table format. The table has four columns: 'compañía', 'teléfono', 'país', and 'total'. The results are ordered by the total amount in descending order.

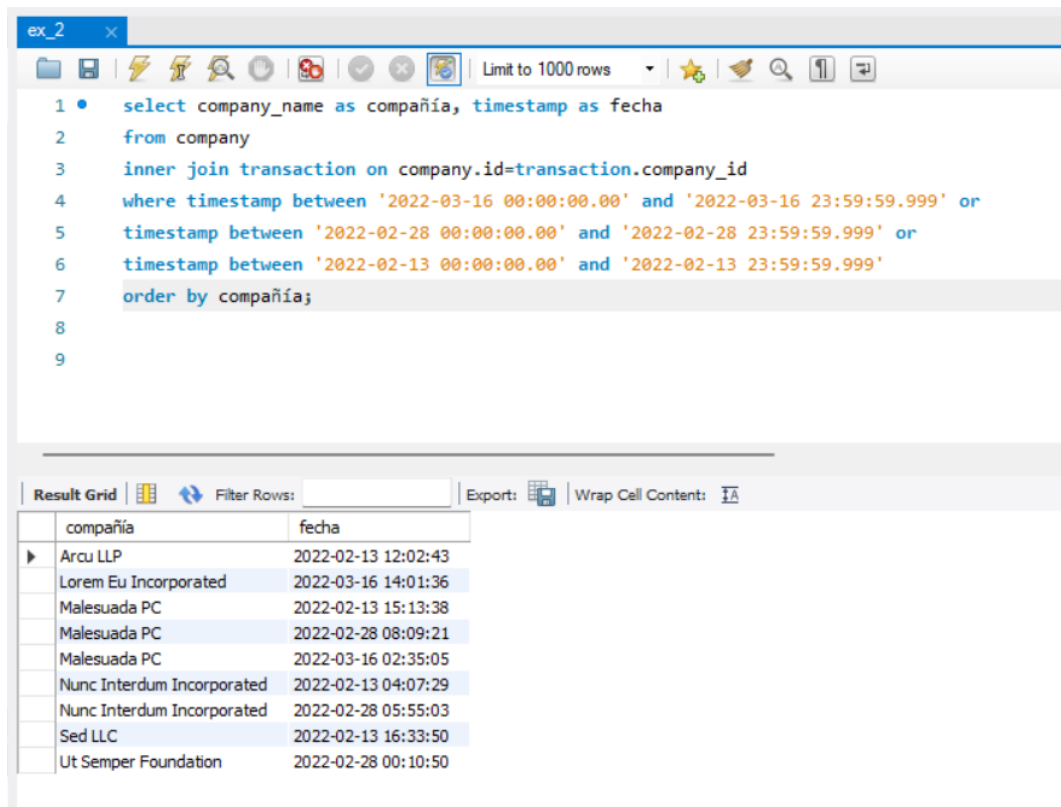
compañía	teléfono	país	total
Pretium Neque Corp.	07 77 48 55 28	Australia	179.86
Dui Cras Associates	08 12 73 87 84	Italy	176.08
Nunc Ac PC	03 53 62 76 95	Sweden	141.10
Nulla Integer Vulputate Corp.	04 57 50 84 48	Sweden	120.80
Orci Adipiscing Limited	03 18 00 77 81	United Kingdom	111.64
Amet Lorem LLP	05 47 44 68 02	Spain	106.80

Ejercicio 2.

Indica el nombre de las compañías que realizaron compras el 16 de marzo de 2022, 28 de febrero de 2022 y 13 de febrero de 2022.

Seleccionamos el nombre de la compañía ('company_name') de la tabla 'company' y las fechas ('timestamp') de la tabla 'transaction' vinculando datos mediante un INNER JOIN.

Después especificamos las fechas con un WHERE. Debido a que nos piden jornadas completas, debemos utilizar el operador BETWEEN para que seleccione todas las operaciones realizadas dentro de cada una de las fechas requeridas. Para incluir todas las fechas utilizamos el operador OR, de modo que tendremos todas las operaciones realizadas. Finalmente, ordenamos por el nombre de la compañía, ya que el nombre parece el criterio principal de la consulta requerida.



```

1 • select company_name as compañía, timestamp as fecha
2   from company
3  inner join transaction on company.id=transaction.company_id
4  where timestamp between '2022-03-16 00:00:00.00' and '2022-03-16 23:59:59.999' or
5     timestamp between '2022-02-28 00:00:00.00' and '2022-02-28 23:59:59.999' or
6     timestamp between '2022-02-13 00:00:00.00' and '2022-02-13 23:59:59.999'
7  order by compañía;
8
9

```

compañía	fecha
Arcu LLP	2022-02-13 12:02:43
Lorem Eu Incorporated	2022-03-16 14:01:36
Malesuada PC	2022-02-13 15:13:38
Malesuada PC	2022-02-28 08:09:21
Malesuada PC	2022-03-16 02:35:05
Nunc Interdum Incorporated	2022-02-13 04:07:29
Nunc Interdum Incorporated	2022-02-28 05:55:03
Sed LLC	2022-02-13 16:33:50
Ut Semper Foundation	2022-02-28 00:10:50