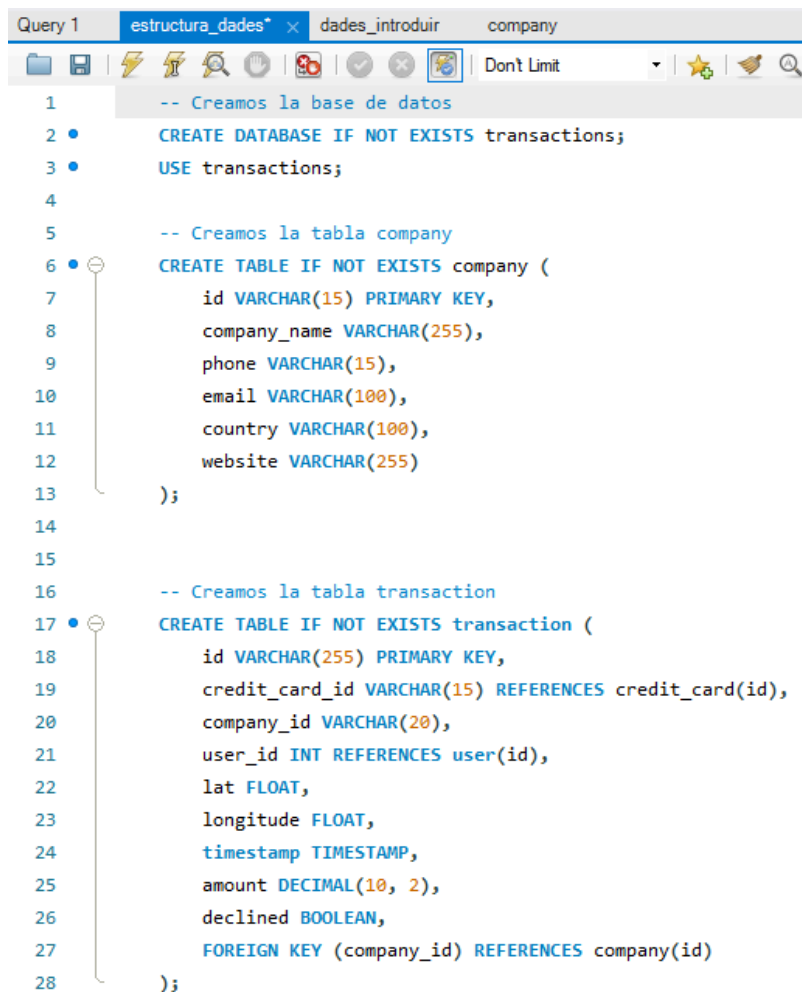


Tasca S2.01. Nocions bàsiques SQL

Nivel 1

Ejercicio 1:

Características principales del esquema creado, explicación de las tablas y variables que existen, diagrama que ilustre la relación entre las diferentes tablas.

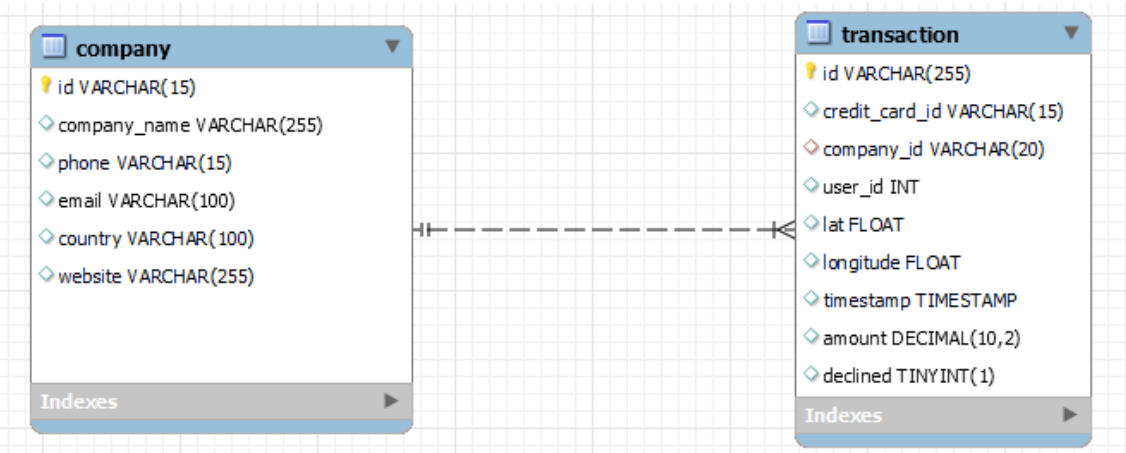


```
Query 1  estructura_dades* x  dades_introduir  company
-- Creamos la base de datos
1  -- Creamos la base de datos
2  CREATE DATABASE IF NOT EXISTS transactions;
3  USE transactions;
4
5  -- Creamos la tabla company
6  CREATE TABLE IF NOT EXISTS company (
7      id VARCHAR(15) PRIMARY KEY,
8      company_name VARCHAR(255),
9      phone VARCHAR(15),
10     email VARCHAR(100),
11     country VARCHAR(100),
12     website VARCHAR(255)
13 );
14
15
16 -- Creamos la tabla transaction
17 CREATE TABLE IF NOT EXISTS transaction (
18     id VARCHAR(255) PRIMARY KEY,
19     credit_card_id VARCHAR(15) REFERENCES credit_card(id),
20     company_id VARCHAR(20),
21     user_id INT REFERENCES user(id),
22     lat FLOAT,
23     longitude FLOAT,
24     timestamp TIMESTAMP,
25     amount DECIMAL(10, 2),
26     declined BOOLEAN,
27     FOREIGN KEY (company_id) REFERENCES company(id)
28 );
```

Tenemos 2 tablas: company y transaction. En la tabla company, tenemos tipos de datos que se utilizan para almacenar cadenas de texto de tipo variable (VARCHAR), de ahí que usemos un paréntesis con la longitud que deseamos que tenga. Se utilizan para mejorar la eficiencia y evitar usar más espacio de lo necesario. Las variables incluyen una clave primaria (columna con valores únicos) “id”, nombre de la compañía, teléfono, email, país y página web.

En la tabla transaction tenemos más variedad de tipos de datos. Incluyen varchar, variables numéricas (‘int’, ‘float’ y ‘decimal’), referencias (‘REFERENCES’) que se utilizan para establecer relaciones entre tablas, variables Booleanas (True or False), variables que indican la

hora en que se realiza una transacción (Timestamp) y una clave foránea. Las variables son id de la tarjeta de crédito, id de la compañía, id del usuario, variables de localización geográfica (latitud y longitud), variable de fecha y hora ('timestamp'), importe de la transacción ('amount'), variable que indica si se ha declinado o no la transacción (declined), descrita con 0 (no se declina la transacción) y 1 (se declina) y la clave foránea id de la compañía.



Aquí podemos observar el diagrama de relación 1:N entre ambas tablas. Esta relación significa que 1 elemento de la tabla company está relacionado con varios elementos de la tabla transaction. Es decir, una compañía puede tener múltiples transacciones asociadas a ella. Esta relación se representa por la clave foránea "company_id" en la tabla "transaction" que referencia el campo o la columna "id" en la tabla "company".

```

estructura_datos  datos_introducir  company  transaction  countries_list*  distinct_countries*  max_avg_sales*
Limit to 1000 rows

1 • USE transactions;
2
3 -- Insertamos datos de company
4 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2222', 'Ac Fermentum Incorp
5 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2226', 'Magna A Neque Indus
6 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2230', 'Fusce Corp.', '08 1
7 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2234', 'Convallis In Incorp
8 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2238', 'Ante Iaculis Nec Fo
9 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2242', 'Donec Ltd', '01 25
10 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2246', 'Sed Nunc Ltd', '02
11 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2250', 'Amet Nulla Donec Co
12 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2254', 'Nascetur Ridiculus
13 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2258', 'Vestibulum Lorem PC
14 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2262', 'Gravida Sagittis LL
15 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2266', 'Mus Aenean Eget Fou
16 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2270', 'Dis Parturient Inst
17 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2274', 'Sed LLC', '01 63 16
18 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2278', 'Arcu LLP', '06 46 0
19 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2282', 'Pretium Neque Corp.
20 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2286', 'Fringilla LLC', '08
21 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2290', 'Quisque Libero LLC'
22 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2294', 'Auctor Mauris Vel L
23 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-2298', 'Elit Etiam Laoreet

```

Aquí tenemos el archivo con los valores que se introducirán en las tablas creadas.

dades_introduir	company	transaction*	countries_list	distinct_countries	german_companies_transaction...	max_avg_sales*
-----------------	---------	--------------	----------------	--------------------	---------------------------------	----------------

```

1 • SELECT *
2 FROM company;

```

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@icloud.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-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.co.uk	Germany	https://cnn.com/user/110
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-2250	Amet Nulla Donec Corporation	07 15 25 14 74	mattis.integer.eu@protonmail.net	Italy	https://netflix.com/sub/cars
b-2254	Nascetur Ridiculus Mus Inc.	06 26 87 61 84	suspendisse.dui@icloud.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 LLC	06 46 04 41 45	dui@aol.ca	Norway	https://yahoo.com/sub
b-2282	Pretium Neque Corp.	07 77 48 55 28	eleifend.nec.malesuada@proton...	Australia	https://netflix.com/sub
b-2286	Erinilla LLC	08 28 15 93 57	gravida@protonmail.co.uk	New Zealand	https://reddit.com/user/110

company 4 x

Output

Action Output

#	Time	Action	Message
27	13:22:32	SELECT * FROM company LIMIT 0, 1000	100 row(s) returned

dades_introduir	company	transaction*	countries_list	distinct_countries	german_companies_transaction...	max_avg_sales
-----------------	---------	--------------	----------------	--------------------	---------------------------------	---------------

```

1 • SELECT *
2 FROM transaction;

```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declin
02C6201E-D90A-1859-B4EE-88D2986D3802	CcU-2938	b-2362	92	81.9185	-12.5276	2021-08-28 23:42:24	466.92	0
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53	0
063FBA79-99EC-66FB-29F7-25726D1764A5	CcU-2987	b-2250	275	-81.2227	-129.05	2022-01-06 21:25:27	92.61	0
0668296C-CD89-A883-76BC-2E4C44F8C8AE	CcU-3743	b-2618	265	-34.3593	-100.556	2022-01-26 02:07:14	394.18	0
06CD9AA5-9B42-D684-DDDD-A5E394FEBAA9	CcU-2959	b-2346	92	33.7381	158.298	2021-10-26 23:00:01	279.93	0
07A46D48-31A3-7E87-65B9-0DA902AD109F	CcU-3225	b-2386	272	38.8342	92.1905	2021-06-28 21:11:42	340.87	1
09DE92CE-6F27-2B87-13B5-9385B2B388E2	CcU-3071	b-2298	275	71.1706	10.5757	2021-05-11 20:40:06	303.05	1
0A476ED9-0C13-1962-F87B-D3563924B539	CcU-4359	b-2302	221	-56.4901	114.801	2022-02-26 20:33:54	430.49	0
0BEB80B7-9D66-1707-CE4B-9DC7E71914B5	CcU-3141	b-2338	272	23.3264	-13.6037	2022-03-04 14:54:35	288.81	1
0C7C3A33-9947-3BC1-846D-7BE3D0D17598	CcU-3309	b-2434	272	63.3615	-68.6667	2021-04-10 20:58:41	103.44	1
0CE957A6-CCAA-2B7A-6839-8A4B1B324853	CcU-3435	b-2506	269	-69.3537	-10.26	2022-02-02 07:29:36	428.69	1
0DD2E608-5C9E-D1B3-4999-B99F43AD735A	CcU-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47	1
1017AA59-3D5F-7A4C-1992-D151A8D1FA0A	CcU-3701	b-2618	267	4.27645	-101.554	2021-11-01 01:02:11	447.11	0
1026DA24-8929-31F1-8250-D7BAB05C13D2	CcU-2959	b-2346	92	32.0646	-140.147	2021-12-07 09:30:38	465.31	0
108B1D1D-5B23-A76C-55EF-C568E49A05DD	CcU-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57	0
10A9B07A-810C-76FB-4D15-12C6C128037	CcU-3155	b-2346	272	-32.0556	-76.7281	2021-05-16 21:00:28	27.85	1

transaction 3 x

Output

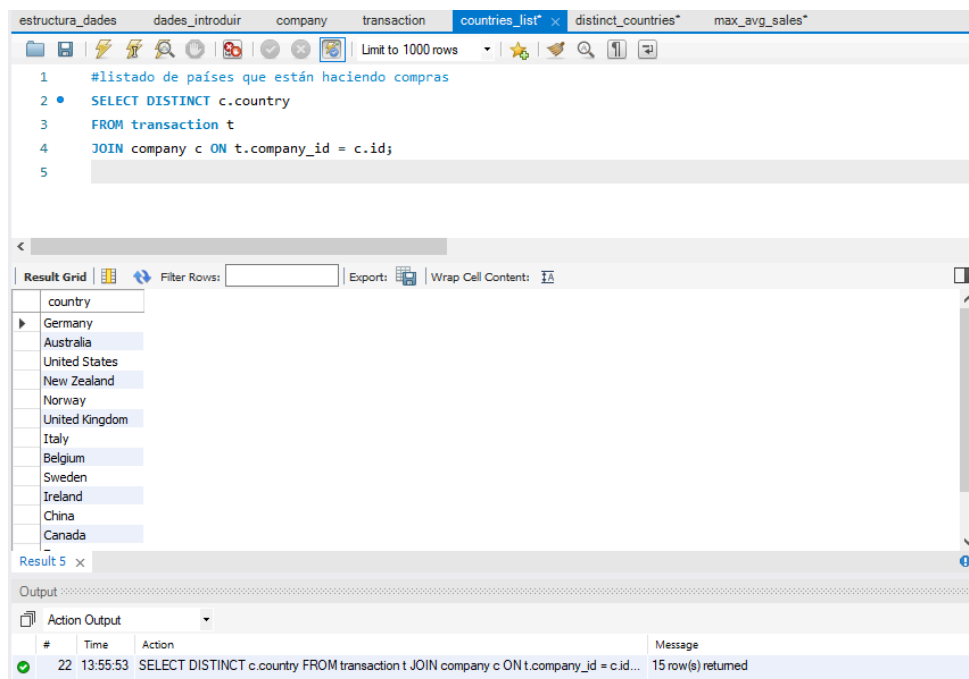
Action Output

#	Time	Action	Message
29	13:26:49	SELECT * FROM transaction LIMIT 0, 1000	587 row(s) returned

Aquí tenemos las dos tablas con sus columnas y los valores introducidos. Basta con hacer `SELECT *` (asterisco = seleccionar todos los valores) `FROM` “nombre de la tabla”. Conviene fijarse en la última sección de Action Output para ver si la consulta se ejecuta o no correctamente y cuántas filas nos devuelve ésta para tener una idea del tamaño de cada tabla y posteriores consultas que realicemos. Nos ayuda un poco a comprobar si tiene sentido o no el output obtenido.

Ejercicio 2: Utilizando JOIN realizamos las siguientes consultas.

Listado de países que están haciendo compras. Para obtener dicho listado empezamos seleccionando con `SELECT DISTINCT` los valores únicos de la columna ‘country’ que se encuentra en la tabla company. Así eliminamos duplicados. Para agilizar el código de la query, podemos usar abreviaciones como alias e introducirlo seguido de “.” y el nombre de la columna. Para company usaremos ‘c’ y en la tabla transaction usaremos ‘t’. Si nos fijamos, country está solamente en la tabla company, pero queremos unir ambas tablas para ver los países realizando compras o transacciones. La forma de relacionar o unir ambas tablas puede hacerse usando JOIN a través de la columna que tengan en común. En este caso, en la tabla transactions es company_id y en la tabla company es id. Veremos que la fila ‘JOIN company c ON t.company_id = c.id’ se repite en las 3 consultas del ejercicio 2. Usamos `FROM transactions t` para que la query acuda a la tabla de transacciones y busque a partir de company_id todas las filas con transacciones realizadas y el país asociado a cada transacción. Todas las consultas conviene finalizarlas con ‘;’ en caso de tener que escribir en un mismo script varias. Así MySQL sabrá que son consultas distintas. Aquí vemos la query y el output obtenido:



The screenshot shows a MySQL IDE interface with several tabs at the top: 'estructura_datos', 'dades_introduir', 'company', 'transaction', 'countries_list*', 'distinct_countries*', and 'max_avg_sales*'. The 'countries_list*' tab is active, displaying a SQL query:

```
1 #listado de países que están haciendo compras
2 • SELECT DISTINCT c.country
3 FROM transaction t
4 JOIN company c ON t.company_id = c.id;
5
```

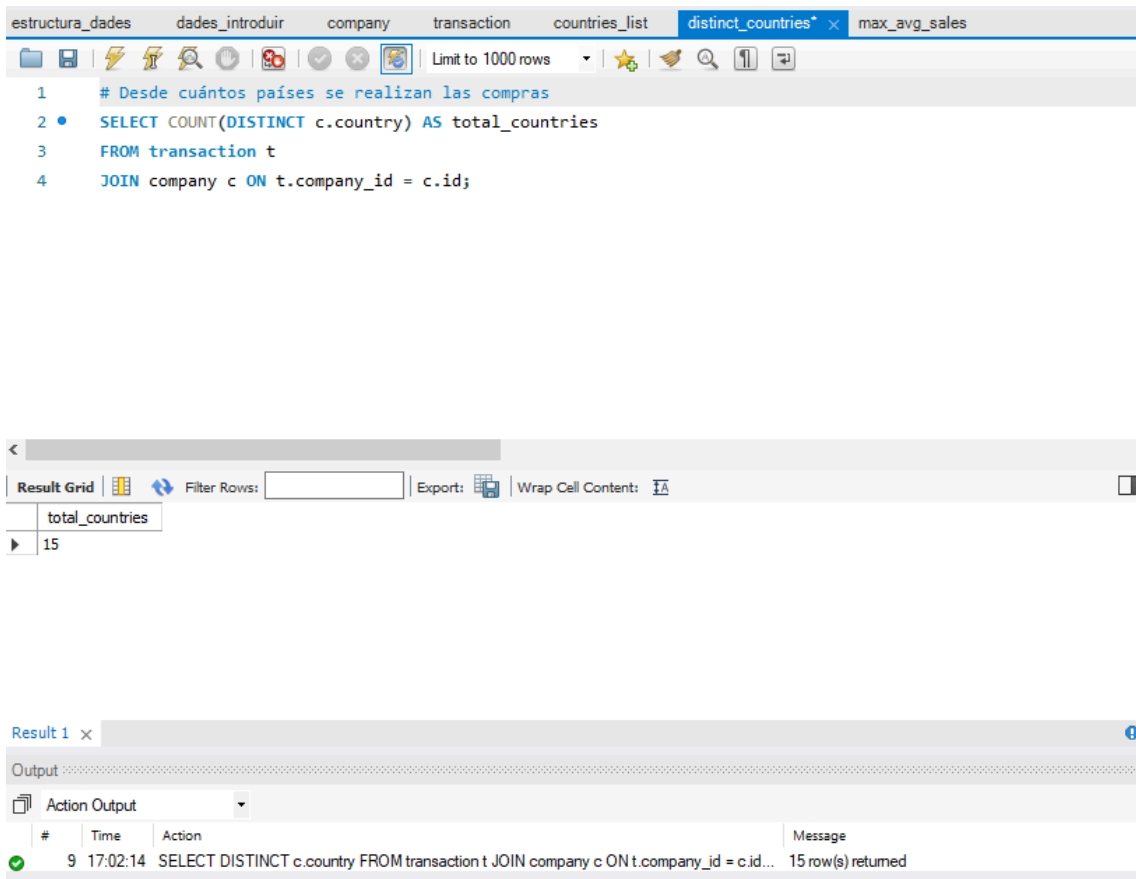
Below the query editor, the 'Result Grid' shows the output of the query. It has a 'Filter Rows' field and an 'Export' button. The results are listed in a table with a single column 'country':

country
Germany
Australia
United States
New Zealand
Norway
United Kingdom
Italy
Belgium
Sweden
Ireland
China
Canada

At the bottom, the 'Output' section shows the 'Action Output' for the query. It includes a table with columns '#', 'Time', 'Action', and 'Message':

#	Time	Action	Message
22	13:55:53	SELECT DISTINCT c.country FROM transaction t JOIN company c ON t.company_id = c.id...	15 row(s) returned

Desde cuántos países se realizan las compras. Para obtener el número de países que realizan las compras, sólo necesitamos incluir en la selección 'COUNT(DISTINCT c.country)'. La palabra clave COUNT nos hace el recuento total de la consulta realizada anteriormente. Podemos incluir un alias para que nos muestre la columna renombrada como 'total_countries' escribiendo 'AS total_countries'. A continuación mostramos la consulta en MySQL:



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

```
1 # Desde cuántos países se realizan las compras
2 • SELECT COUNT(DISTINCT c.country) AS total_countries
3 FROM transaction t
4 JOIN company c ON t.company_id = c.id;
```

The results pane shows a single row with the value 15 for the column total_countries.

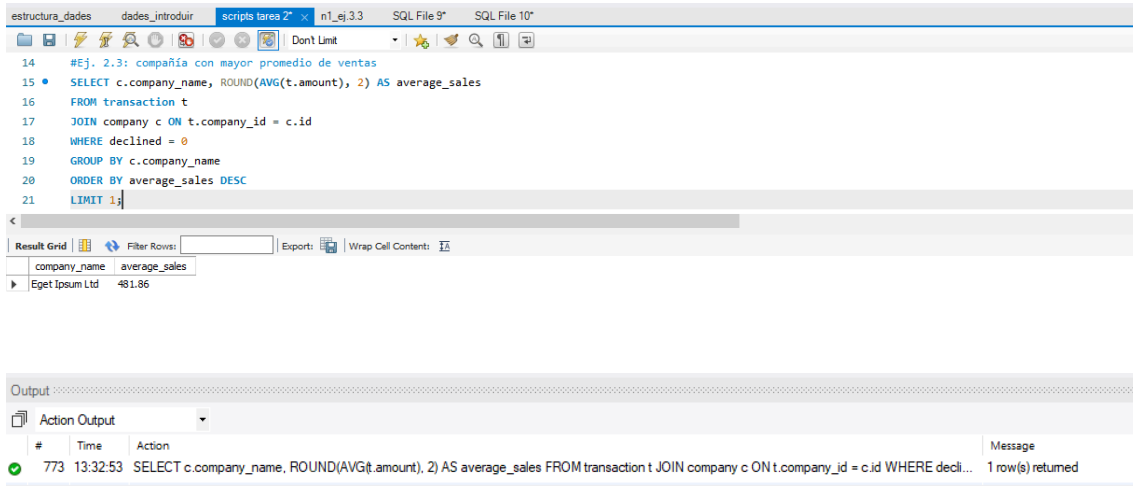
total_countries
15

Below the results pane, there is a log showing the execution of the query:

#	Time	Action	Message
9	17:02:14	SELECT DISTINCT c.country FROM transaction t JOIN company c ON t.company_id = c.id...	15 row(s) returned

Identifica la compañía con la media más grande de ventas. Para realizar esta consulta primero seleccionamos las columnas que queremos visualizar en el output. Queremos el nombre de la empresa que está en la tabla company y la media de ventas, cuya columna 'amount' con el importe de éstas se encuentra en la tabla transaction. Para ello seleccionamos c.company_name y haremos el cálculo del promedio de ventas con ROUND(AVG(t.amount), 2) para redondear a 2 decimales, recordar que el importe se encuentra en la tabla de transaction, por eso empezamos con 't.' Podemos renombrar esta columna con un alias, así que pondremos 'AS average_sales' después del promedio calculado. Las dos siguientes líneas de código son idénticas a las consultas anteriores. Debemos asegurarnos que estamos cogiendo las transacciones aceptadas para ver el promedio de ventas, por tanto incluimos el criterio WHERE declined = 0. Para poder ver la empresa con el mayor promedio agrupamos por nombre de empresa usando GROUP BY c.company_name y ordenamos por valor del promedio usando ORDER BY y el alias que hemos

creado, 'average_sales'. Para ver sólo el promedio de ventas más alto, usamos DESC (palabra clave que ordena de mayor a menor valor) y debajo usamos LIMIT 1 para ver la primera fila con el nombre de la empresa y el mayor valor de promedio de ventas. A continuación mostramos el código SQL y el output.



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

```
14 #Ej. 2.3: compañía con mayor promedio de ventas
15 SELECT c.company_name, ROUND(AVG(t.amount), 2) AS average_sales
16 FROM transaction t
17 JOIN company c ON t.company_id = c.id
18 WHERE declined = 0
19 GROUP BY c.company_name
20 ORDER BY average_sales DESC
21 LIMIT 1;
```

The results pane shows a single row of data:

company_name	average_sales
Eget Ipsum Ltd	481.86

The output pane shows the execution details:

#	Time	Action	Message
773	13:32:53	SELECT c.company_name, ROUND(AVG(t.amount), 2) AS average_sales FROM transaction t JOIN company c ON t.company_id = c.id WHERE decli...	1 row(s) returned

Como vemos en la consulta, la empresa Eget Ipsum Ltd tiene el mayor promedio de ventas.

Ejercicio 3

Crear subconsultas sin utilizar JOIN.

Mostrar todas las transacciones realizadas por empresas de Alemania. En esta consulta queremos ver todas las columnas de la tabla transaction cuyas empresas están localizadas en Alemania. Para ello seleccionamos todas las columnas con '*' FROM transaction e introducimos el criterio WHERE donde hacemos la relación entre company_id de la tabla transaction e id de la tabla company sin usar JOIN pero utilizando una subconsulta. El criterio de WHERE es company_id IN (subconsulta que seleccione todas las id de empresas ubicadas en Alemania, SELECT id FROM company WHERE country = 'Germany'). Vemos como queda en MySQL:

The screenshot shows a database query editor with the following SQL query:

```

1  # Mostrar todas las transacciones realizadas por empresas de Alemania
2  SELECT *
3  FROM transaction
4  WHERE company_id IN (
5      SELECT id
6      FROM company
7      WHERE country = 'Germany'
8  );

```

The results are displayed in a grid with the following columns: id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, and decline. The first few rows of data are:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	decline
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-A387-44F8-729FB4B51C76	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-DCD8-9C54CF1EBB9A	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

The output section shows the execution details:

```

Output:
Action Output
# Time Action Message
42 14:03:30 SELECT * FROM transaction WHERE company_id IN ( SELECT id FROM company WHERE country = 'Germany' ) 118 row(s) returned

```

118 transacciones realizadas por empresas ubicadas en Alemania.

Mostrar todas las empresas cuyas transacciones realizadas tengan un importe superior al promedio de todas las transacciones. Para realizar esta consulta seleccionamos company_name de la tabla company, usamos una primera subconsulta con el criterio WHERE id IN para seleccionar company_id de la tabla transaction con otro criterio WHERE con la segunda subconsulta que nos coja la id de las empresas cuyo importe (amount) sea > al promedio de todas las transacciones en la tabla transaction. Basta con una selección de AVG(amount) de la tabla transaction. Según la teoría mostrada en este sprint, sería un ejemplo de consulta anidada o subconsulta dentro de otra subconsulta. Cabe destacar cómo se colocan los paréntesis finales, para que el usuario entienda que hay 2 subconsultas, se suele cerrar en una nueva fila y debajo del criterio WHERE correspondiente a cada subconsulta. El código se ejecuta correctamente si cerramos los paréntesis justo en la misma línea donde acabamos nuestro código pero supongo que si se diese el caso en que se debe hacer 3 o más subconsultas, es fácil olvidarse o confundirse al final cuántos paréntesis debemos introducir. Si se escribe el cierre de paréntesis debajo de cada WHERE, facilita la comprensión. Veamos como queda:

70 empresas realizan transacciones con importes por encima de la media de transacciones.

The screenshot shows a database management tool interface. At the top, there are tabs for different tables: 'company', 'transaction*', 'countries_list', 'distinct_countries', 'german_companies_transaction...', 'max_avg_sales*', and 'above_avg_sales_companie...'. Below the tabs is a toolbar with various icons. The main area displays a SQL query:

```

2 • SELECT company_name
3 FROM company
4 WHERE id IN (
5     SELECT company_id
6     FROM transaction
7     WHERE amount > (
8         SELECT AVG(amount)
9         FROM transaction
10    )
11 );

```

Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows:' input field, an 'Export:' button, and a 'Wrap Cell Content:' checkbox. The grid shows the following data:

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
Mus Acceper Est Foundation

At the bottom, there is an 'Output' section with a dropdown menu set to 'Action Output'. It shows a log of actions:

#	Time	Action	Message
39	13:46:20	SELECT company_name FROM company WHERE id IN (SELECT company_id FRO...	70 row(s) returned

Eliminarán del sistema las empresas que carecen de transacciones registradas, mostrar el listado de estas empresas. Para realizar esta última consulta del ejercicio 3 en el nivel 1 seleccionamos el nombre de la compañía de la tabla company y usamos el criterio WHERE id NOT IN para identificar aquellas empresas que no tengan transacciones realizadas. Es decir, que no tengan ningún registro en la tabla transaction. Seleccionamos en la subconsulta usando DISTINCT para que no tenga que buscar company_id en la tabla transaction más de una vez, en caso de que se repitan. Para nuestro caso no importa usar solo SELECT, pero con bases de datos grandes agilizará la ejecución de la consulta. Veamos como queda.


```
estructura_datos  datos_introduir  scripts tarea 2  n1_ej.3.3
1  #Ej. 3.3: Listado de empresas sin transacciones realizadas
2  • SELECT company_name
3  FROM company
4  WHERE id NOT IN (
5      SELECT DISTINCT company_id
6      FROM transaction
7  )
```

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

company 2 x			
Output			
Action Output			
#	Time	Action	Message
732	10:14:12	SELECT company_name FROM company WHERE id NOT IN (SELECT DISTINCT company_id FROM transaction)	0 row(s) returned

Si el listado nos sale vacío, todas las empresas tienen al menos un registro en la tabla transaction. No hay ninguna empresa sin realizar transacciones.

Nivel 2

Ejercicio 1

Identifica los cinco días que se generó la mayor cantidad de ingresos en la empresa por ventas. Muestra la fecha de cada transacción junto con el total de las ventas. Para hacer esta consulta basta con seleccionar la variable timestamp donde se indica la fecha y la hora, asegurándonos que cogemos solo la fecha con la cláusula DATE(timestamp). Para ver el total de ventas haremos SUM(amount). Podemos renombrar todo con alias 'AS date' y 'AS total_sales'. Al tratarse de ventas, debemos asegurar que cogemos las transacciones aceptadas, por tanto incluimos de nuevo WHERE declined = 0.

Agrupamos por fecha, ordenamos por importe de ventas totales y limitamos a 5 días con LIMIT 5.

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

```
52 # NIVEL 2
53 # Ejercicio 1: Identificar los cinco días que se generó la mayor cantidad de ingresos en la empresa por ventas y mostrar la fecha de cada transacción junto con el total
54 # de las ventas.
55 • SELECT DATE(timestamp) AS date, SUM(amount) AS total_sales
56 FROM transaction
57 WHERE declined = 0
58 GROUP BY DATE(timestamp)
59 ORDER BY total_sales DESC
```

The results pane displays a table with two columns: 'date' and 'total_sales'. The table contains five rows of data, sorted by total sales in descending order.

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

The bottom of the IDE shows an 'Output' pane with a message indicating that the query was successful and returned 5 rows.

Ejercicio 2

¿Cuál es la media de ventas por país? Presenta los resultados ordenados de mayor a menor promedio. Para ello, seleccionamos country de la tabla company y ROUND(AVG(t.amount), 2) para coger los importes de la tabla transaction y hacer el promedio, usamos 'average_sales' como alias. Agrupamos por país y ordenamos por average_sales de mayor a menor valor usando DESC.

```

62 #Ejercicio 2: ¿Cuál es la media de ventas por país? Presenta los resultados ordenados de mayor a menor promedio.
63 • SELECT c.country, ROUND(AVG(t.amount), 2) AS average_sales
64 FROM transaction t
65 JOIN company c ON t.company_id = c.id
66 WHERE declined = 0
67 GROUP BY c.country
68 ORDER BY average_sales DESC;

```

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

country	average_sales
United States	287.53
Ireland	285.83
Sweden	276.67
United Kingdom	271.77
Canada	261.94
Belgium	255.22
Norway	251.11
Italy	243.34
Germany	242.24
Netherlands	240.94
China	222.24
Australia	177.33
France	169.41
New Zealand	167.06
Spain	26.22

Result 71 x

#	Time	Action	Message
795	13:48:10	SELECT c.country, ROUND(AVG(t.amount), 2) AS average_sales FROM transaction t JOIN company c ON t.company_id = c.id WHERE declined = 0 ...	15 row(s) returned

Ejercicio 3

En tu empresa, se plantea un nuevo proyecto para lanzar algunas campañas publicitarias para hacer competencia a la compañía “Non Institute”. Para ello, te piden la lista de todas las transacciones realizadas por empresas que están ubicadas en el mismo país que esta compañía.

Mostrar el listado aplicando JOIN y subconsultas:

Para esta consulta, seleccionamos todas las columnas de la tabla transaction, usamos el JOIN que hemos estado realizando en todas las consultas uniendo por `t.company_id = c.id` y en el criterio WHERE hacemos la subconsulta, queremos el país de la tabla company cuyo nombre de empresa sea ‘Non Institute’.

estructura_datos datos_introducir scripta tarea 2* n1_ej.3.3 SQL File 9* SQL File 10*

Don't Limit

#Ejercicio 3: lista de todas las transacciones realizadas por empresas que están ubicadas en el mismo país que 'Non Institute'.
Mostrar el listado aplicando JOIN y subconsultas:

```
72 • SELECT *
73 FROM transaction t
74 JOIN company c ON t.company_id = c.id
75 WHERE c.country = (
76     SELECT country
77     FROM company
78     WHERE company_name = 'Non Institute'
79 );
```

Result Grid

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	id	company_name	phone	email
28928E1C-EC14-A760-0A75-871477649D6A	CdU-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14:49	383.73	0	b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org
ACD2011A-A2B1-C365-41E1-2AB00C65147A	CdU-2980	b-2246	275	-54.4792	-82.7974	2022-03-05 20:41:20	60.07	1	b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org
4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	CdU-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37:23	458.74	0	b-2310	Non Magna LLC	06 71 73 13 17	nisl.quisque.fringilla@hotmail.c
BC289A38-7B4-28CD-1FE8-14DED863E773	CdU-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95	1	b-2310	Non Magna LLC	06 71 73 13 17	nisl.quisque.fringilla@hotmail.c
14798302-87BA-C7B8-4CE3-8D7C2DE85A8B	CdU-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58:07	309.45	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
152998C2-029D-D684-4666-91EDF393EBFF	CdU-2994	b-2326	126	-67.0189	-141.672	2021-07-05 03:10:00	395.43	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
1B6365B8-A3E9-7C69-D9C9-C4535DADF03B	CdU-2994	b-2326	131	70.2543	-13.1336	2021-07-06 08:48:46	195.06	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
20418D65-8804-8E98-8D7A-A95C3BFD8F5C	CdU-2994	b-2326	126	-79.1145	1.51481	2022-01-03 15:59:29	479.52	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
23988576-6C0E-137A-C2F6-3180A188A2D3	CdU-2994	b-2326	126	23.6174	137.222	2021-08-26 06:04:05	43.90	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
267C4A86-78A7-1CE5-0718-2824983C8D0D	CdU-2994	b-2326	126	-17.5359	104.915	2021-10-01 21:08:53	122.63	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
3142C93E-83E7-49E4-E52D-29C4834B19D0	CdU-2994	b-2326	126	-67.8476	-119.978	2021-04-06 17:24:44	91.59	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
3578688E-7B1D-B887-3EC7-20B8673AA31E	CdU-2994	b-2326	126	87.0665	-22.7339	2021-07-26 22:59:24	303.60	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
360C7B14-F74F-843A-0946-AB38D2683C86	CdU-2994	b-2326	116	-7.93005	-79.0733	2021-08-21 10:19:58	494.82	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
391E1CFD-D653-E458-A729-F2EB93247B58	CdU-2994	b-2326	118	60.5512	103.904	2021-10-09 00:50:38	271.27	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
3C4D7C2A-A402-8941-625A-D64CA5352E68	CdU-2994	b-2326	117	23.1627	-10.381	2021-04-25 19:11:52	441.27	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
3ED634C2-01CA-D6E9-2DBD-24B53A7912F7	CdU-4219	b-2326	137	69.0549	96.5033	2021-03-29 11:38:38	478.54	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
3FEC4131-98ED-2EA6-1776-82674C80D0B8	CdU-2994	b-2326	126	-7.28469	179.063	2021-07-25 04:59:17	393.42	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
41C13F08-471E-D475-3E14-760DF3F79842	CdU-4219	b-2326	138	-51.7289	55.3294	2022-01-16 16:48:55	45.63	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
41EC691D-808E-D85C-F05F-08BC3DE7A618	CdU-2994	b-2326	126	5.38535	-78.3917	2021-01-06 07:47:39	335.54	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
4217EB57-1D51-5D25-8645-EC24D23C8759	CdU-2994	b-2326	129	65.4733	-20.2735	2021-04-23 13:07:58	193.33	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
45CA3E45-6842-CAC4-7683-EBB69D52383E	CdU-4219	b-2326	140	85.7055	36.1497	2021-04-29 06:17:02	149.89	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
49CD105D-E569-3733-C8C8-3908ED7E1838	CdU-3393	b-2326	108	-47.8783	168.275	2021-12-24 23:43:59	411.81	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
4ABE3D77-BF86-12D3-E1C5-22663D37C199	CdU-3120	b-2326	273	-63.6087	69.6023	2021-11-13 04:18:28	38.02	1	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
4B74B94B-D123-7C7E-A9CE-7DA26F911F55	CdU-2994	b-2326	128	25.6024	53.4543	2022-02-12 14:27:28	304.43	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
4D8E4891-1711-7D3D-E4CB-645ABDF59A43	CdU-2994	b-2326	120	5.71915	-155.369	2021-12-26 01:49:19	365.83	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
4E562A2B-99E9-8C82-234D-86623C9FE2F	CdU-3120	b-2326	273	-65.8033	158.1	2021-03-23 15:43:37	409.68	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
5F47388E-8392-A2E2-1E8E-C4A812C6A95C	CdU-2994	b-2326	126	-57.1143	-154.767	2022-01-10 05:12:55	253.68	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
61EC87A4-CF88-B392-F83B-75E7F2C93537	CdU-3393	b-2326	110	88.4626	85.6351	2021-06-01 20:23:57	346.10	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
634E53C5-C2D6-EC48-4C0F-6B4460A4EEB7	CdU-2994	b-2326	125	-77.8923	-167.683	2021-09-17 15:02:25	399.71	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
65D82765-E83D-2B74-2B0A-ABA5FD248928	CdU-2994	b-2326	126	-19.9162	2.61475	2021-12-08 09:25:11	140.67	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
71C8B78E-F93E-4B77-DF6D-1B03B312ED29	CdU-3393	b-2326	110	-27.1212	-105.573	2021-10-21 20:41:56	441.63	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
7D0819EC-7363-6767-36C0-97C7E77D9E65	CdU-3393	b-2326	107	-26.2616	26.8733	2021-05-09 15:36:26	158.89	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
836E8A9E-E737-37D0-5385-585F1B8A318B	CdU-2994	b-2326	124	67.1904	117.617	2021-05-24 22:17:30	170.15	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah
84B8C4A6-A786-91C0-C973-8B7305659FA3	CdU-3393	b-2326	110	-7.43343	-77.4374	2021-12-02 07:12:35	179.18	0	b-2326	Enim Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yah

Result 73 x

Output

Action Output

#	Time	Action	Message
797	13:51:04	SELECT * FROM transaction t JOIN company c ON t.company_id = c.id WHERE c.country = (SELECT country FROM company WHERE company_name = 'Non Institute')	100 row(s) returned

Mostrar el listado con subconsultas solamente. En este caso haremos una misma selección de todas las columnas en la tabla transaction con SELECT * FROM transaction y hacemos coincidir ids de la compañía con WHERE company_id IN (subconsulta que encuentre ID de la tabla company donde el país coincida con el país de la empresa 'Non Institute'). Vemos como queda:

estructura_dades dades_introduir n2_ej3.1sql n2_ej3.2 SQL File 6

Limit to 1000 rows

```

1 • SELECT *
2 FROM transaction
3 WHERE company_id IN (
4     SELECT id
5     FROM company
6     WHERE country = (
7         SELECT country
8         FROM company
9         WHERE company_name = 'Non Institute'
10     )
11 );
12

```

Result Grid

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-1FE8-14DE0863E773	CcU-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95	1
1479B3D-2B76A-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-A2E8-7C69-D9C9-C54535DAFD3B	CcU-2994	b-2326	131	70.2543	-13.1336	2021-07-06 08:48:46	195.06	0
20418DE5-B804-8E9B-8D7A-A95C1BFDF8F5C	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-8887-3BC7-20B8673AA31E	CcU-2994	b-2326	126	87.0665	-22.7339	2021-07-26 22:59:24	303.60	0
360C7814-77AF-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
3FEC4131-98ED-2EA6-1776-82674C80DDB8	CcU-2994	b-2326	126	-7.28469	179.063	2021-07-25 04:59:17	393.42	0
41C13F08-471E-D475-3E14-76D0F3F79B42	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-EB869D52383E	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

transaction 4 x

Output

Action Output

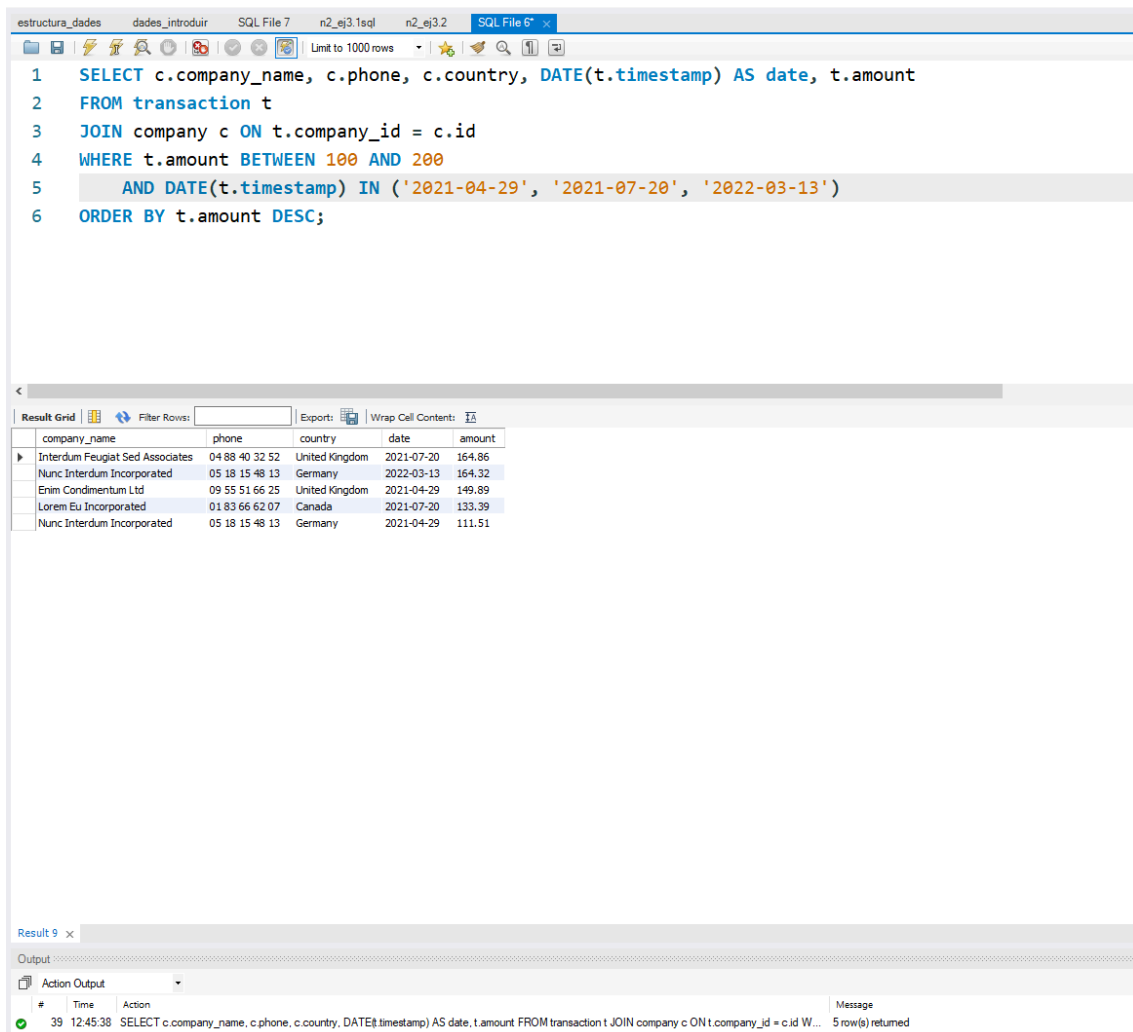
#	Time	Action	Message
24	11:48:15	SELECT * FROM transaction t JOIN company c ON t.company_id = c.id WHERE c.country = (SELECT country FROM company WHERE com...	100 row(s) returned

La diferencia entre ambas queries es que la primera muestra todas las columnas de ambas tablas y la segunda, usando solamente subconsultas, solo nos muestra las columnas de la tabla transaction. No obstante, el objetivo era mostrar transacciones de empresas que se encontrasen en el mismo país que Non Institute (UK).

Nivel 3

Ejercicio 1

Para ello, seleccionamos las variables pedidas, recordar coger la fecha con DATE(t.timestamp). Hacemos el JOIN que hemos usado durante toda la tarea y el criterio WHERE será t.amount entre 100 y 200 y que las fechas sean las indicadas con DATE(t.timestamp) IN (fechas en formato 'YY-MM-DD'). Ordenamos de mayor a menor valor con ORDER BY t.amount DESC.



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

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

The results pane displays a table with 5 columns: company_name, phone, country, date, and amount. The table contains 5 rows of 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 Conditimentum 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

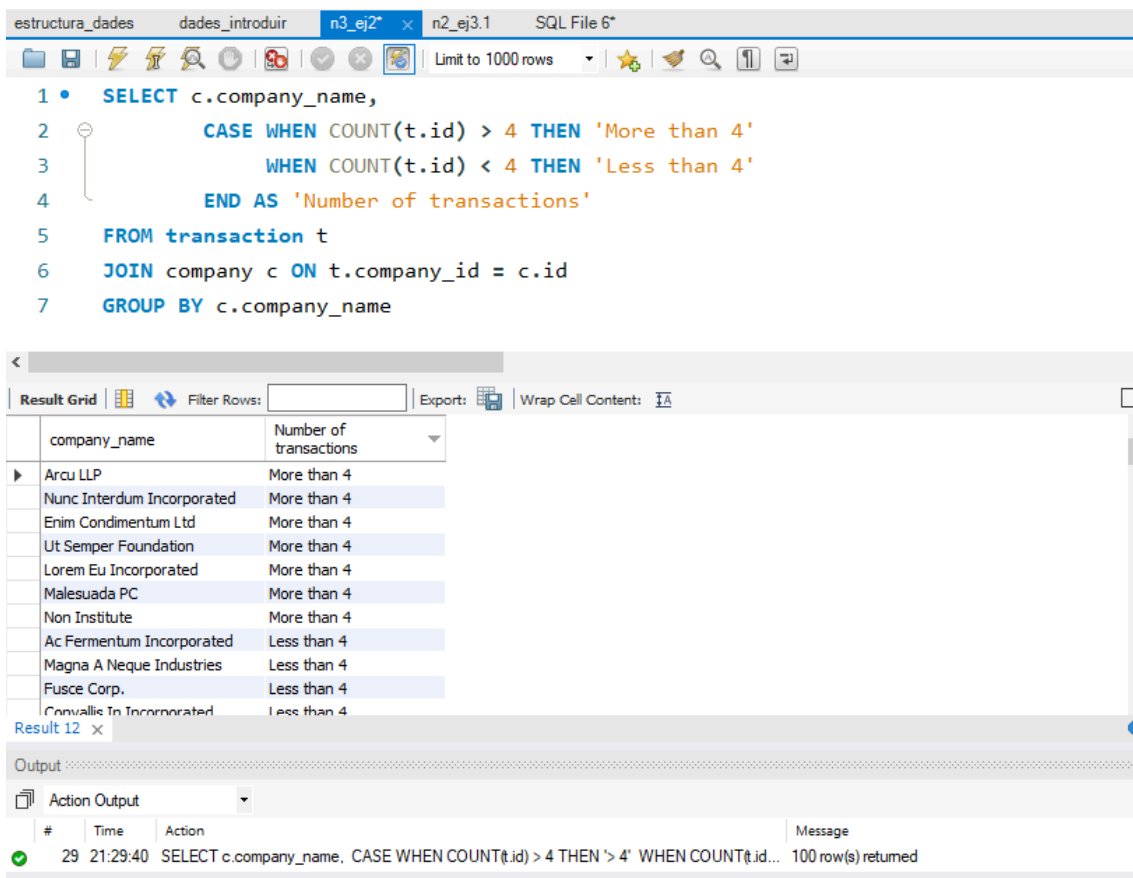
The bottom of the screenshot shows the 'Action Output' pane with a single entry:

#	Time	Action	Message
39	12:45:38	SELECT c.company_name, c.phone, c.country, DATE(t.timestamp) AS date, t.amount FROM transaction t JOIN company c ON t.company_id = c.id W...	5 row(s) returned

Ejercicio 2

Necesitamos optimizar la asignación de los recursos y dependerá de la capacidad operativa que se requiera, por lo que te piden la información sobre la cantidad de transacciones que realizan las empresas, pero el departamento de recursos humanos es exigente y quiere un listado de las empresas donde especifiques si tienen más de 4 o menos transacciones.

Para realizar esta última consulta, haremos uso de CASE WHEN para hacer un listado de empresas con una columna que muestre si tienen más o menos de 4 transacciones. Usamos JOIN y agrupamos por company_name.



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

```
1 • SELECT c.company_name,  
2       CASE WHEN COUNT(t.id) > 4 THEN 'More than 4'  
3           WHEN COUNT(t.id) < 4 THEN 'Less than 4'  
4       END AS 'Number of transactions'  
5 FROM transaction t  
6 JOIN company c ON t.company_id = c.id  
7 GROUP BY c.company_name
```

The results grid shows the following data:

company_name	Number of transactions
Arcu LLP	More than 4
Nunc Interdum Incorporated	More than 4
Enim Condimentum Ltd	More than 4
Ut Semper Foundation	More than 4
Lorem Eu Incorporated	More than 4
Malesuada PC	More than 4
Non Institute	More than 4
Ac Fermentum Incorporated	Less than 4
Magna A Neque Industries	Less than 4
Fusce Corp.	Less than 4
Connullie In Incorporated	Less than 4

The output pane shows the following message:

```
29 21:29:40 SELECT c.company_name, CASE WHEN COUNT(t.id) > 4 THEN '> 4' WHEN COUNT(t.id)... 100 row(s) returned
```

Si hacemos click en el encabezado de la columna 'Number of transactions' se ordenan en 2 partes (más o menos de 4 transacciones).