

Sprint 2 Data Analytics

Nivell 1

- Ejercicio 1

A partir de los documentos adjuntos (estructura_datos y datos_introducir), 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 tablas y variables.

Tras importar los datos en mysql he procedido a realizar el diagrama que, como se puede visualizar consta de 2 tablas, la tabla "Company" y la tabla "Transaction", teniendo una relación de 1 a N. En este caso la empresa puede tener un identificador único pero múltiples transacciones.

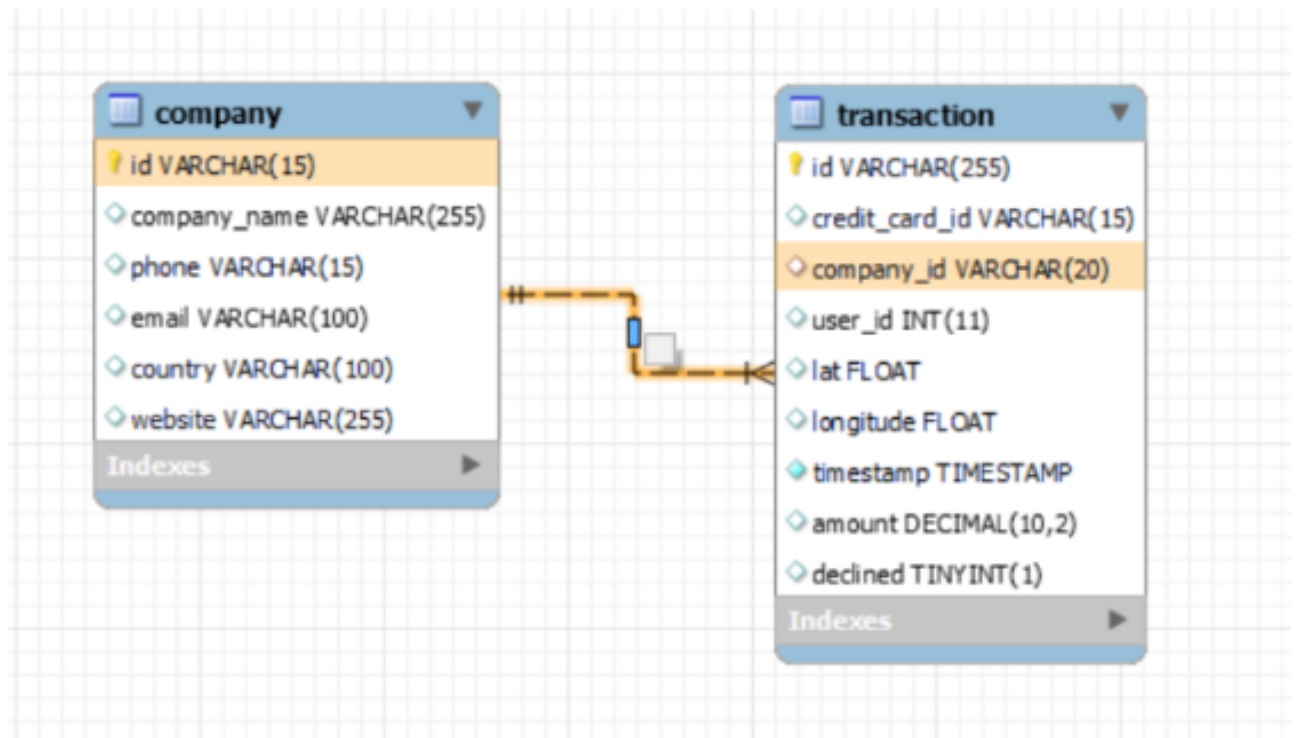


Tabla 1. Company: tabla de Empresa

Variables

id: Identificador y Primary key (**PK**). Se trata del identificador único para cada empresa.

Cuando queremos asegurarnos al 100% que una columna es la PK podemos hacer la siguiente query:

```
3 • SELECT id, count(*)
4 FROM transactions.company
5 GROUP BY id
6 HAVING COUNT(*)>1;
7
```

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

id	count(*)
----	----------

Result 6 x

Output

Action Output

#	Time	Action	Message
1	22:55:17	SELECT id, count(*) FROM transactions.company GROUP BY id HAVING COUNT(*)>1	0 row(s) returned

En primer lugar, seleccionamos el id de la tabla “company” y contamos los registros agrupados por id, filtrando aquellos con más de una aparición usando HAVING COUNT(*) > 1 para incluir aquellos resultados que tienen más de una fila. Como no nos devuelve ninguna fila, podemos deducir que no hay duplicados y que por tanto es nuestra PK.

company_name: Nombre de la empresa.
phone: Teléfono de la empresa.
email: Correo electrónico de la empresa.
country: País en el cual está ubicada la empresa.
website: Web de la empresa.

Tabla 2. Transaction: Tabla de transacción

Variables

id: Identificador único para cada transacción. Es la PK de esta tabla.

credit_card_id: Identificador de las tarjetas de crédito.

company_id: Identificador de la compañía. Indica a qué empresa pertenece la transacción y es la **FK**.

La FK o Foreign Key es un campo o campos que nos ayudan a relacionarlos con una PK de otra tabla. En este caso la PK de la tabla 1 Company.id está relacionada con Company_id.

user_id: Identificador para el usuario que realizó la transacción.

lat: Latitud de la ubicación donde se realizó la transacción.

longitude: Longitud de la ubicación donde se realizó la transacción.

timestamp: Marca de tiempo cuando se realizó la transacción.

amount: Monto de la transacción.

declined: Indicador de si la transacción fue rechazada.

- Ejercicio 2

Utilizando JOIN realizarás las siguientes consultas:

Listado de los países que están haciendo compras.

```
5 -- Nivel 1. Ejercicio 2 Listado de los países que están haciendo compras.
7 • SELECT distinct company.country
3 FROM transactions.company
3 JOIN transactions.transaction
3 ON company.id = transaction.company_id
1 WHERE declined = 0;
2
```

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

country
Canada
Germany
Italy
United Kingdom
Sweden
Ireland
United States
Norway
France
New Zealand
Netherlands

ult 7 x

out

Action Output

#	Time	Action	Message
1	12:33:54	select distinct company.country from transactions.company join transactions.transaction on company.id...	15 row(s) returned

En esta consulta seleccionamos todos los países de la tabla company, eliminando aquellos duplicados con distinct. Después, con una JOIN unimos las tablas “company” y “transaction” a través de la PK (id) de la tabla “Company” y la Fk (company_id) de la tabla “Transaction”. Por último, para asegurarnos de que solo se incluyen las transacciones que no fueron rechazadas usamos el (declined = 0). Nos da un resultado de 15 países.

Desde cuántos países se realizan las compras.

```
24 • SELECT count(DISTINCT company.country) AS num_países
25 FROM transactions.company
26 JOIN transactions.transaction
27 ON company.id = transaction.company_id
28 WHERE declined = 0;
```

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

num_países
15

Result 9 x

Output :

Action Output

#	Time	Action	Message
✓ 1	23:28:39	SELECT count(DISTINCT company.country) AS num_países FROM transactions.company JOIN trans...	1 row(s) returned

En esta consulta seleccionamos el conteo de los países, siempre con distinct para que al igual que en la consulta anterior no se nos repitan dichos países y en esta ocasión he puesto el alias “num_países”. Después, con una JOIN unimos las tablas “company” y “transaction” a través de la PK (id) de la tabla “Company” y la Fk (company_id) de la tabla “Transaction”. Y volvemos a incluir (declined = 0) para incluir sólo las transacciones que no fueron rechazadas. El resultado nos da 15.

Identifica la compañía con la media más grande de ventas.

```
31 • SELECT company.company_name, ROUND(AVG(transaction.amount),2) AS media
32 FROM transactions.transaction
33 LEFT JOIN transactions.company
34 ON company.id = transaction.company_id
35 WHERE declined = 0
36 GROUP BY company.id
37 ORDER BY media DESC
38 LIMIT 1;
```

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

company_name	media
Eget Ipsum Ltd	481.86

Result 10 x

Output

Action Output

#	Time	Action	Message
✓ 1	23:30:19	SELECT company.company_name, ROUND(AVG(transaction.amount),2) AS media FROM transactio...	1 row(s) returned

En primer lugar, seleccionamos el nombre de la empresa (company_name) y la media de las cantidades de transacción (ROUND(AVG(transaction.amount), 2)) redondeada a dos decimales. Se realiza una unión externa izquierda (LEFT JOIN) entre las tablas “Company” y “Transaction” a través de la PK (id) de la tabla “Company” y la Fk (company_id). Solo se consideran las transacciones que no fueron rechazadas (declined = 0). Los resultados se agrupan por el id de la empresa (GROUP BY company.id), se ordenan por la media en orden descendente (ORDER BY media DESC), y se limita el resultado a una sola fila (LIMIT 1), es decir, la empresa con la media de transacciones más alta.

-ROUND redondea a las decimales que le pongas.

- Ejercicio 3

Utilizando solo subconsultas (sin utilizar JOIN):

Muestra todas las transacciones realizadas por empresas de Alemania.

```
40 -- Nivel 1. Ejercicio 3-1 Muestra todas las transacciones realizadas por empresas de Alemania
41 • SELECT*
42 FROM transaction
43 WHERE company_id IN (SELECT company.id
44                      FROM company
45                      WHERE country = "Germany");
46
```

The screenshot shows a SQL IDE interface. At the top, a query is written in a text editor. Below it, a 'Result Grid' displays the results of the query. The grid has columns: id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, and declined. It contains 18 rows of transaction data. Below the result grid, there is an 'Output' section with a tab labeled 'Action Output'. This section shows a log of the query execution, including the time (12:55:59) and the message 'SELECT* FROM transaction WHERE company_id IN (SELECT company.id FROM company WHERE country = "Germany"); 118 row(s) returned'.

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A05DD	CcU-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57	0
EA2C3281-C9C1-A387-44F8-729FB4B51C76	CcU-2938	b-2222	275	20.2004	-116.84	2021-05-09 10:25:08	119.36	1
0DD2E608-5C9E-D1B3-4999-B99F43AD735A	CcU-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47	1
AB069F53-965E-A2A8-CE06-CA8C4FD92501	CcU-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99	0
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53	0
0A476ED9-0C13-1962-F87B-D3563924B539	CcU-4359	b-2302	221	-56.4901	114.801	2022-02-26 20:33:54	430.49	0
122DC333-E19F-D629-DCD8-9C54CF1EBB9A	CcU-4366	b-2302	221	29.6372	-166.173	2021-06-09 06:04:14	172.01	0
135267BA-2E7D-957C-C42C-6450A2B3ED54	CcU-4520	b-2302	210	20.6724	14.9732	2021-12-29 20:38:23	17.97	0
14CAE5B5-8FB1-3E4A-4C85-0EA4167534F4	CcU-4849	b-2302	189	-53.6202	93.0533	2021-12-31 00:29:42	388.04	0
158A3ACB-541C-DBCC-65BD-6373CC67BF1C	CcU-4849	b-2302	183	42.5424	-170.347	2022-03-08 05:02:19	240.29	0
162C7E78-2B6B-7971-A1E4-D2124E732451	CcU-4527	b-2302	210	-69.1381	58.0017	2021-04-11 05:59:18	231.26	0
1717FD6B-ADAD-7082-A748-9112BE892CCC	CcU-4219	b-2302	172	69.4892	-138.411	2021-12-29 16:18:54	249.91	0

transaction 14 x

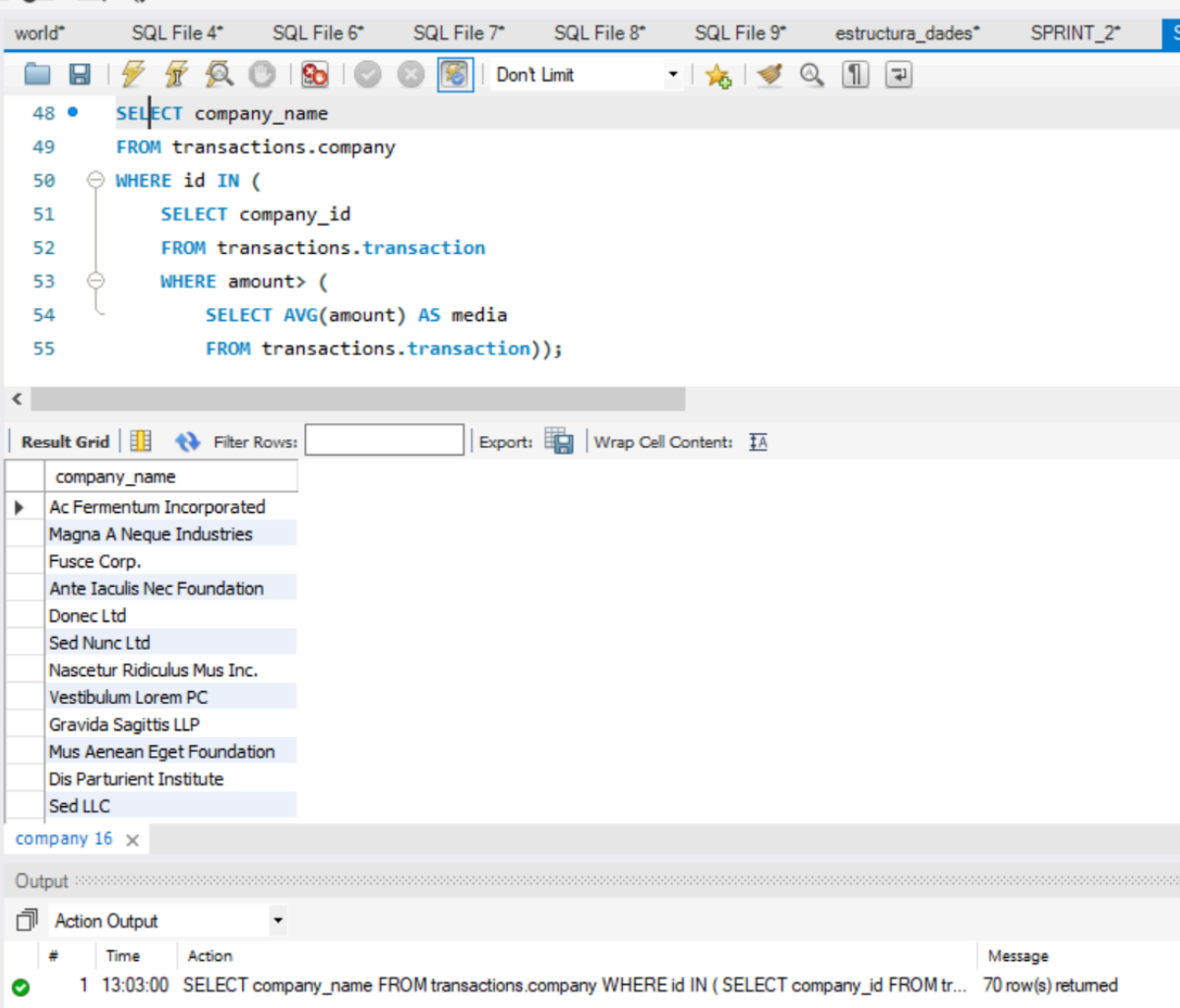
Output

Action Output

#	Time	Action	Message
1	12:55:59	SELECT* FROM transaction WHERE company_id IN (SELECT company.id FROM company WHERE country = "Germany");	FROM comp... 118 row(s) returned

En primer lugar, seleccionamos todas las columnas de la tabla "Transaction" con un SELECT *. Después, filtramos con un WHERE, seleccionando las transacciones cuyos company_id están en la lista de id de empresas (company.id) que a su vez están en el país "Germany" (WHERE country = "Germany"). Es decir, devuelve todas las transacciones que pertenecen a empresas ubicadas en Alemania.

Lista las empresas que han realizado transacciones por un amount superior a la media de todas las transacciones.



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

```
48 SELECT company_name
49 FROM transactions.company
50 WHERE id IN (
51     SELECT company_id
52     FROM transactions.transaction
53     WHERE amount > (
54         SELECT AVG(amount) AS media
55         FROM transactions.transaction));
```

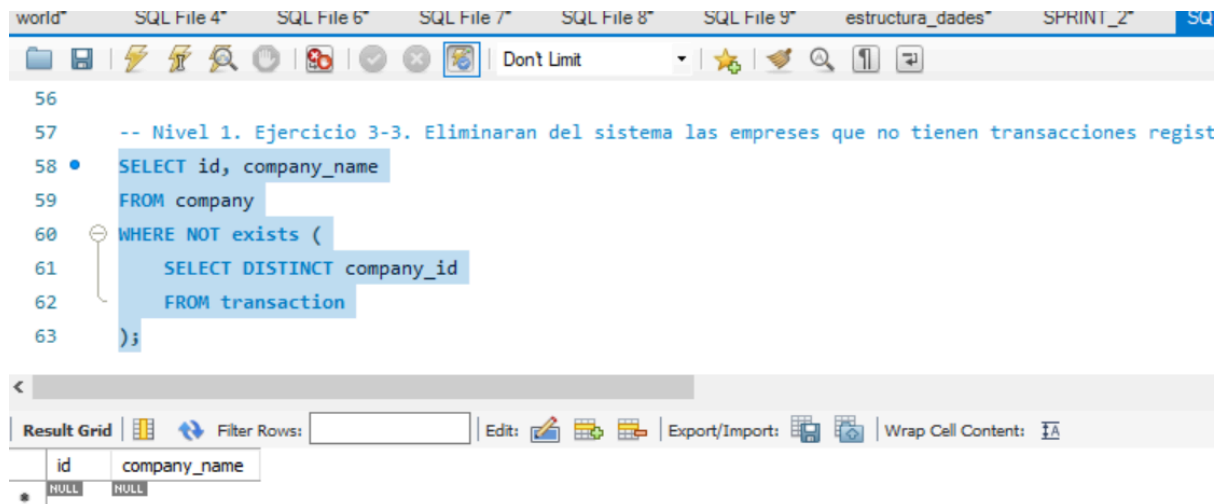
The results pane displays a list of company names. The first few are:

- 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 Aenean Eget Foundation
- Dis Parturient Institute
- Sed LLC

The results pane also shows a message: "70 row(s) returned".

En esta query, seleccionamos los nombres de las empresas (company_name) de la tabla "Company". Después empezamos las subconsultas, filtrando las empresas (WHERE id IN) cuyo "id" está en la lista de "company_id" de transacciones (transactions.transaction) donde el monto (amount) es mayor que el promedio (AVG(amount)) de todos los montos de transacción. Es decir, devuelve los nombres de las empresas que tienen transacciones con montos superiores al promedio global de todas las transacciones.

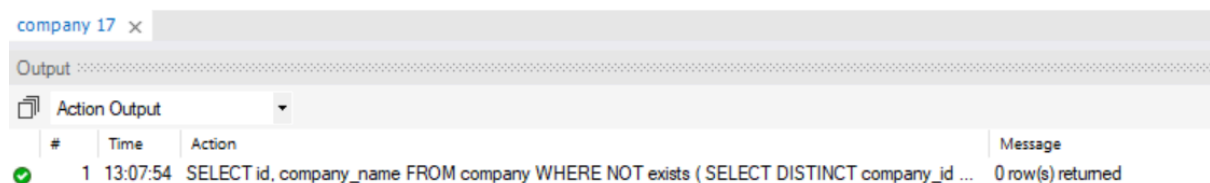
Se eliminarán del sistema las empresas que no tienen transacciones registradas, entrega la lista de estas empresas.



```
56
57 -- Nivel 1. Ejercicio 3-3. Eliminaran del sistema las empresas que no tienen transacciones regist
58 • SELECT id, company_name
59 FROM company
60 WHERE NOT exists (
61     SELECT DISTINCT company_id
62     FROM transaction
63 );
```

Result Grid

id	company_name
NULL	NULL



company 17 x

Output

Action Output

#	Time	Action	Message
1	13:07:54	SELECT id, company_name FROM company WHERE NOT exists (SELECT DISTINCT company_id ...	0 row(s) returned

En esta query, seleccionamos el “id” y el “company_name” de la tabla “company”. Después filtramos las empresas donde no existe ninguna transacción asociada, para ello usamos la cláusula NOT EXISTS y así verificamos que no hay ningún “company_id” en la tabla “Transaction” que coincida con el “id” de la empresa en cuestión. En resumen, devuelve las empresas que no tienen ninguna transacción registrada en la tabla “Transaction”.

Nivel 2

Ejercicio 1

Identifica los cinco días que se generó la cantidad más grande de ingresos a la empresa por ventas. Muestra la fecha de cada transacción junto con el total de las ventas.

```
65 -- Nivel 2. Ejercicio 1. Identifica los cinco días que se generó la cantidad más grande de ingresos a la
66 • SELECT DATE(timestamp) AS fecha, sum(transaction.amount) AS total
67 FROM transactions.transaction
68 WHERE declined = 0
69 GROUP BY fecha
70 ORDER BY total desc
71 LIMIT 5;
72
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
fecha	total			
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			

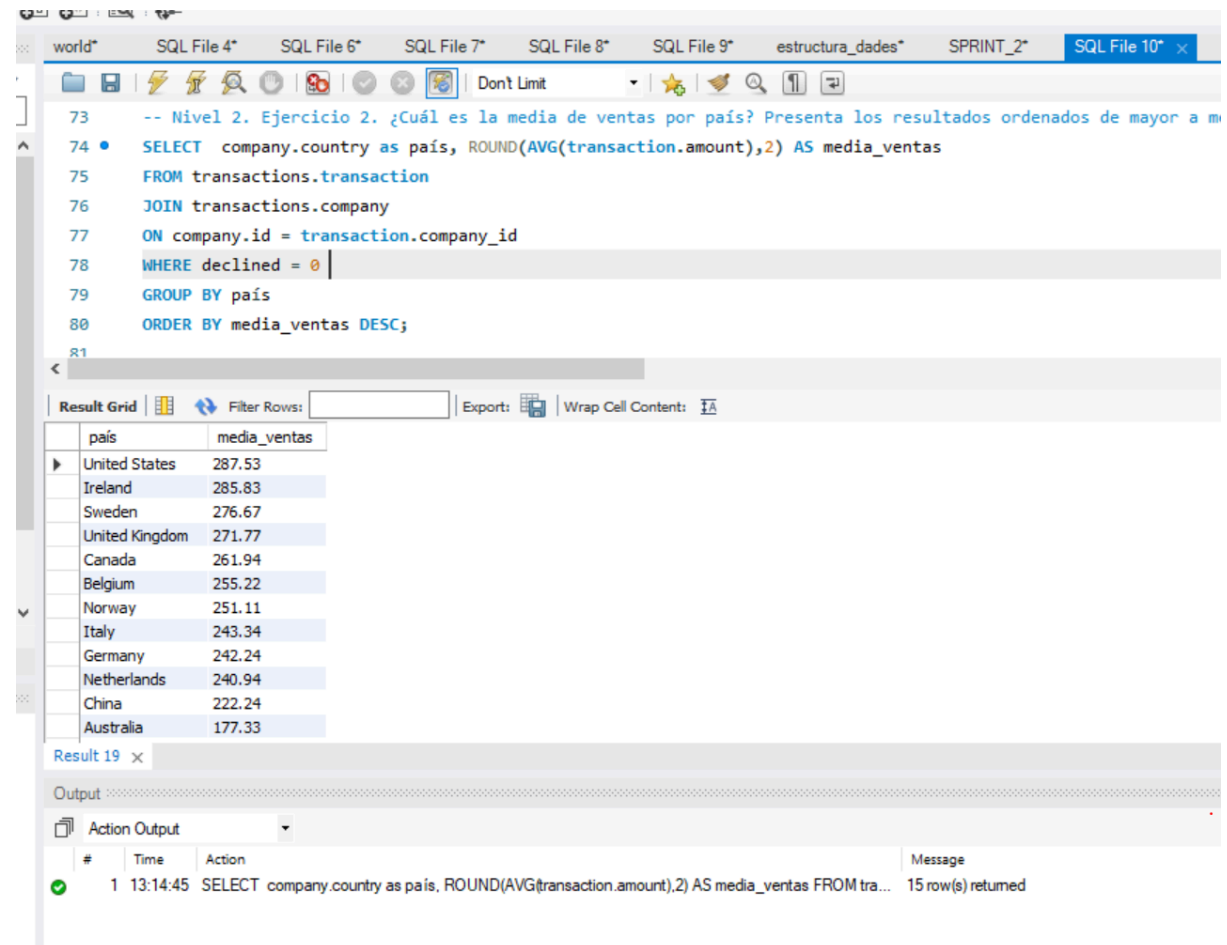
Output			
Action Output			
#	Time	Action	Message
1	13:12:24	SELECT DATE(timestamp) AS fecha, sum(transaction.amount) AS total FROM transactions.transaccio...	5 row(s) returned

En primer lugar, seleccionamos la fecha de las transacciones (DATE(timestamp) AS fecha) y la suma de los montos de transacción (sum(transaction.amount) AS total) poniéndoles en el porceso alias. Después, filtramos las transacciones donde no fueron rechazadas (declined = 0). Luego, agrupamos las transacciones por fecha (GROUP BY fecha) y las ordenamos por el total de montos en orden descendente (ORDER BY total DESC). Finalmente, limitamos el resultado a las 5 fechas con el total de montos más altos (LIMIT 5). Es decir, con todo ello mostramos las 5 fechas con la mayor cantidad total de transacciones no rechazadas.

-Date hace que aparezca la fecha pero sin la hora.

Ejercicio 2

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



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

```
-- Nivel 2. Ejercicio 2. ¿Cuál es la media de ventas por país? Presenta los resultados ordenados de mayor a menor media.
SELECT company.country as país, ROUND(AVG(transaction.amount),2) AS media_ventas
FROM transactions.transaction
JOIN transactions.company
ON company.id = transaction.company_id
WHERE declined = 0
GROUP BY país
ORDER BY media_ventas DESC;
```

The results pane displays a table with two columns: 'país' and 'media_ventas'. The data is sorted in descending order of average sales.

país	media_ventas
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

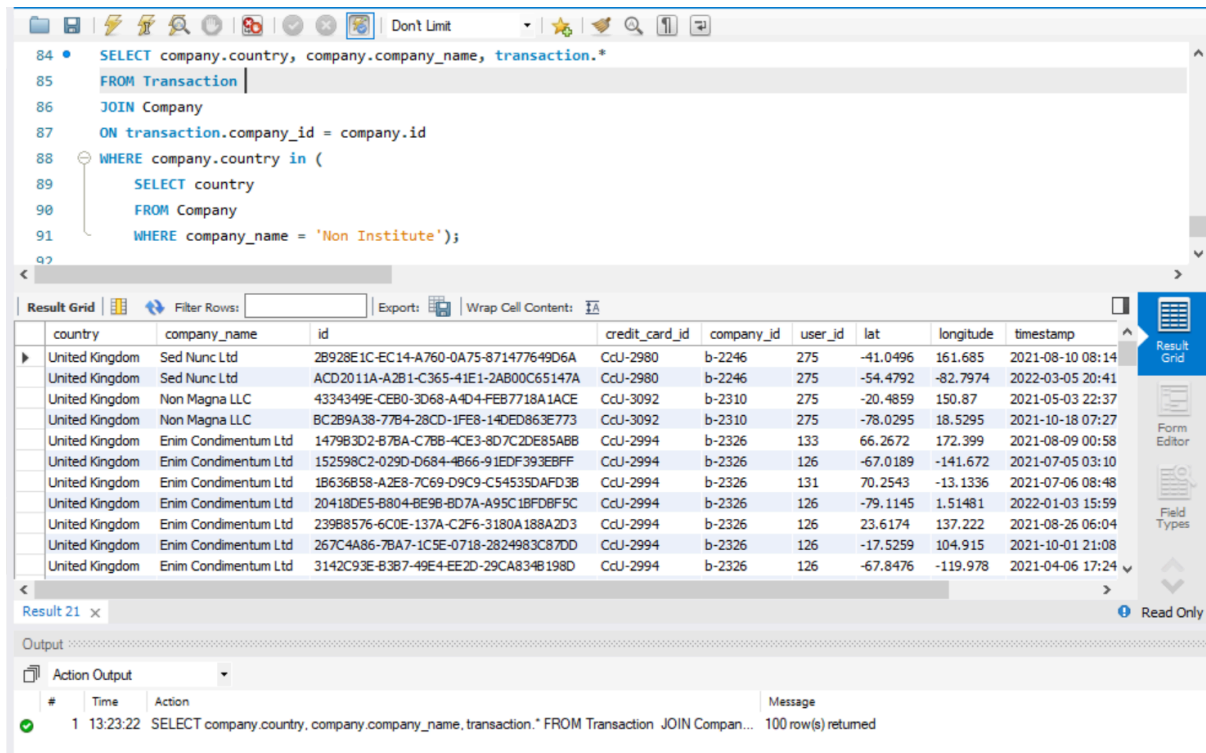
The output pane shows the execution of the query, indicating that 15 rows were returned.

Seleccionamos el país de la empresa (company.country AS país) y la media de ventas (ROUND(AVG(transaction.amount),2) AS media_ventas), a ambos les ponemos alias para usar más tarde. Después unimos (JOIN) la tabla “transaction” con la tabla “company” usando el id de la empresa y el company_id de la transacción. Luego filtramos las transacciones donde no fueron rechazadas (declined = 0). Más tarde, agrupamos las transacciones por país (GROUP BY país) y se ordenan en orden descendente según la media de ventas (ORDER BY media_ventas DESC). En resumen, muestra la media de ventas de las empresas por país, ordenadas de mayor a menor.

- Ejercicio 3

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

Muestra el listado aplicando JOIN y subconsultas.



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

```
84 SELECT company.country, company.company_name, transaction.*
85 FROM Transaction
86 JOIN Company
87 ON transaction.company_id = company.id
88 WHERE company.country in (
89     SELECT country
90     FROM Company
91     WHERE company_name = 'Non Institute');
92
```

The bottom pane displays the 'Result Grid' with 10 columns: country, company_name, id, credit_card_id, company_id, user_id, lat, longitude, and timestamp. The results show 10 rows of data, all from the United Kingdom, associated with 'Non Institute'.

country	company_name	id	credit_card_id	company_id	user_id	lat	longitude	timestamp
United Kingdom	Sed Nunc Ltd	2B928E1C-EC14-A760-0A75-871477649D6A	CcU-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14
United Kingdom	Sed Nunc Ltd	ACD2011A-A2B1-C365-41E1-2AB00C65147A	CcU-2980	b-2246	275	-54.4792	-82.7974	2022-03-05 20:41
United Kingdom	Non Magna LLC	4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	CcU-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37
United Kingdom	Non Magna LLC	BC2B9A38-77B4-28CD-1FE8-14DED863E773	CcU-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27
United Kingdom	Enim Condimentum Ltd	1479B3D2-B7BA-C7B8-4CE3-8D7C2DE85AB8	CcU-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58
United Kingdom	Enim Condimentum Ltd	152598C2-029D-D684-4B66-91EDF393EBFF	CcU-2994	b-2326	126	-67.0189	-141.672	2021-07-05 03:10
United Kingdom	Enim Condimentum Ltd	1B636B58-A2E8-7C69-D9C9-C5453DAFD38	CcU-2994	b-2326	131	70.2543	-13.1336	2021-07-06 08:48
United Kingdom	Enim Condimentum Ltd	20418DE5-B804-8E9B-8D7A-A95C1BFD8F5C	CcU-2994	b-2326	126	-79.1145	1.51481	2022-01-03 15:59
United Kingdom	Enim Condimentum Ltd	239B8576-6C0E-137A-C2F6-3180A188A2D3	CcU-2994	b-2326	126	23.6174	137.222	2021-08-26 06:04
United Kingdom	Enim Condimentum Ltd	267C4A86-7BA7-1C5E-0718-2824983C87DD	CcU-2994	b-2326	126	-17.5259	104.915	2021-10-01 21:08
United Kingdom	Enim Condimentum Ltd	3142C93E-83B7-49E4-EE2D-29CA834B198D	CcU-2994	b-2326	126	-67.8476	-119.978	2021-04-06 17:24

Below the result grid, the 'Output' pane shows the execution details:

#	Time	Action	Message
1	13:23:22	SELECT company.country, company.company_name, transaction.* FROM Transaction JOIN Compan...	100 row(s) returned

En esta consulta seleccionamos el país de la empresa (company.country), el nombre de la empresa (company.company_name), y todas las columnas de la tabla "Transaction" (transaction.*). Se realiza una unión (JOIN) entre las tablas "Transaction" y "Company" basándose en el company_id de la transacción y el id de la empresa. Filtramos los resultados donde el país de la empresa está incluido (IN) en la lista de países donde el nombre de la empresa es "Non Institute". En resumen, la consulta devuelve todas las transacciones de las empresas cuyo nombre coincide con "Non Institute", junto con el país y el nombre de la empresa asociada a cada transacción.

Muestra el listado aplicando solo subconsultas.

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

```
SELECT transaction.*, (SELECT company_name
FROM transactions.company
WHERE company_id= transactions.company.id) AS empresa
FROM transactions.transaction
WHERE company_id IN ( select company.id
FROM transactions.company
WHERE company.country = (
SELECT company.country
FROM transactions.company
WHERE company_name = 'Non Institute') );
```

The results grid displays the following data:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	em
2B928E1C-EC14-A760-0A75-871477649D6A	CcU-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14:49	383.73	0	Sed
ACD2011A-A2B1-C365-41E1-2AB00C65147A	CcU-2980	b-2246	275	-54.4792	-82.7974	2022-03-05 20:41:20	60.07	1	Sed
4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	CcU-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37:23	458.74	0	Non
BC2B9A38-77B4-28CD-1FE8-14DED863E773	CcU-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95	1	Non
1470B3D7-B7BA-C7BB-4CE3-8D7C7DF85A8B	CcU-2980	b-2246	133	66.7672	172.309	2021-08-09 00:58:07	309.45	0	Eni

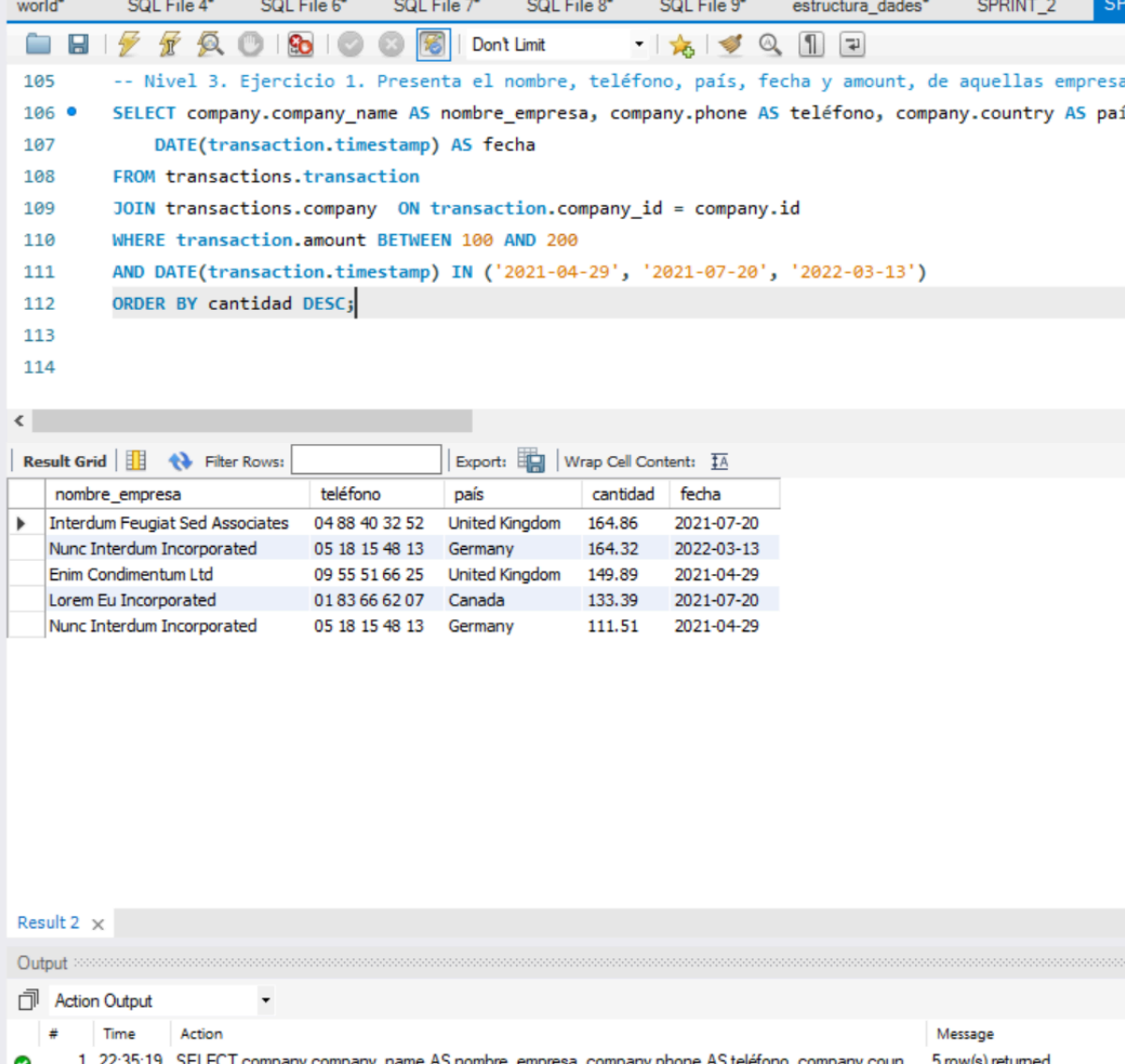
The output section shows the execution of the query, indicating that 100 row(s) were returned.

En esta query seleccionamos todas las columnas de la tabla “transaction”(transaction.*) y agregamos una columna adicional bajo el alias “empresa”, que contiene el nombre de la empresa asociada a cada transacción. Lo hace mediante una subconsulta que busca el company_name en la tabla “company” donde el “company_id” coincide con el “id” de la empresa. Luego, filtramos las transacciones para incluir solo aquellas cuyo “company_id” pertenece a empresas que están en el mismo país que la empresa llamada 'Non Institute'. La subconsulta interna busca el country de la empresa 'Non Institute' y luego selecciona las transacciones de todas las empresas en ese país.

Nivel 3

Ejercicio 1

Presenta el nombre, teléfono, país, fecha y amount, de aquellas empresas que realizaron transacciones con un valor comprendido entre 100 y 200 euros y en alguna de estas fechas: 29 de abril del 2021, 20 de julio del 2021 y 13 de marzo del 2022. Ordena los resultados de mayor a menor cantidad.



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

```
-- Nivel 3. Ejercicio 1. Presenta el nombre, teléfono, país, fecha y amount, de aquellas empresa
SELECT company.company_name AS nombre_empresa, company.phone AS teléfono, company.country AS paí
      DATE(transaction.timestamp) AS fecha
FROM transactions.transaction
JOIN transactions.company ON transaction.company_id = company.id
WHERE transaction.amount BETWEEN 100 AND 200
AND DATE(transaction.timestamp) IN ('2021-04-29', '2021-07-20', '2022-03-13')
ORDER BY cantidad DESC;
```

The results pane displays a table with 6 columns: nombre_empresa, teléfono, país, cantidad, and fecha. The table contains 5 rows of data, sorted by cantidad in descending order.

nombre_empresa	téléfono	país	cantidad	fecha
Interdum Feugiat Sed Associates	04 88 40 32 52	United Kingdom	164.86	2021-07-20
Nunc Interdum Incorporated	05 18 15 48 13	Germany	164.32	2022-03-13
Enim Condimentum Ltd	09 55 51 66 25	United Kingdom	149.89	2021-04-29
Lorem Eu Incorporated	01 83 66 62 07	Canada	133.39	2021-07-20
Nunc Interdum Incorporated	05 18 15 48 13	Germany	111.51	2021-04-29

The bottom pane shows the output of the query, indicating that 5 rows were returned.

En primer lugar seleccionamos el nombre de la empresa (company.company_name), el teléfono (company.phone), el país (company.country), el monto de la transacción (transaction.amount), y la fecha de la transacción (DATE(transaction.timestamp)). Después unimos las tablas transaction y company usando transaction.company_id y company.id. Filtramos las transacciones con un monto entre 100 y 200 (BETWEEN 100 AND 200) y que ocurrieron en las fechas específicas ('2021-04-29', '2021-07-20', '2022-03-13'). Los resultados se ordenan por el monto de la transacción en orden descendente (ORDER BY cantidad DESC).

Ejercicio 2

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

```
113
114 -- Nivel 3. Ejercicio 2. Necesitamos optimizar la asignación de los recursos y dependerá de la capacidad operativa
115 • SELECT company_name as empresa, count(transaction.id) as transacciones, CASE
116     WHEN COUNT(transaction.id) > 4 THEN "Más de 4"
117     ELSE "Menos de 4"
118     END AS "transacciones mayores o menores de 4"
119 FROM transactions.transaction
120 JOIN transactions.company on company.id= transaction.company_id
121 GROUP BY empresa
122 ORDER BY transacciones desc;
```

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

empresa	transacciones	transacciones mayores o menores de 4
Nunc Interdum Incorporated	105	Más de 4
Ut Semper Foundation	59	Más de 4
Enim Condimentum Ltd	57	Más de 4
Arcu LLP	56	Más de 4
Lorem Eu Incorporated	54	Más de 4
Malesuada PC	52	Más de 4
Non Institute	30	Más de 4
Fringilla Porttitor Incorporated	2	Menos de 4
Nec Luctus LLC	2	Menos de 4
Vestibulum Lorem PC	2	Menos de 4
Lacus Quisque Associates	2	Menos de 4
Cras Consulting	2	Menos de 4
Interdum Feugiat Sed Associ...	2	Menos de 4
Nulla Integer Vulputate Corp.	2	Menos de 4
Mauris Incorporated	2	Menos de 4

result 4 x

Output

Action Output

#	Time	Action	Message
1	22:37:56	SELECT company_name as empresa, count(transaction.id) as transacciones, CASE WHEN COUNT(transaction.id) > 4 THEN "Más de 4" ELSE "Menos de 4" END AS "transacciones mayores o menores de 4" FROM transactions.transaction JOIN transactions.company on company.id= transaction.company_id GROUP BY empresa ORDER BY transacciones desc;	100 row(s) returned

En esta query seleccionamos el nombre de la empresa (company_name AS empresa), el número de transacciones (COUNT(transaction.id) AS transacciones) a ambos les ponemos alias y usamos una etiqueta que indica si la empresa tiene más de 4 transacciones o menos (CASE ... END AS "transacciones mayores o menores de 4").

CASE funciona como un condicional, es decir, si se da que el conteo del "id" de la tabla "Transacción" es mayor a 4 entonces se imprimirá "más de 4", sino se imprimirá "menos de 4". Después, unimos como siempre las tablas "transaction" y "company" usando "company.id" y "transaction.company_id". Luego, agrupamos los resultados por nombre de empresa (GROUP BY empresa) y los ordenamos por el número de transacciones en orden descendente (ORDER BY transacciones DESC).