

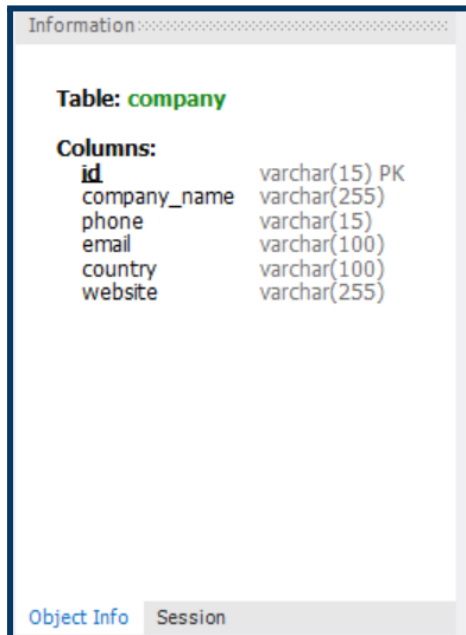
## S2.01 Nociones básicas SQL

<b>1. Nivel 1</b>	<b>1</b>
1.1. A partir de los documentos adjuntos (estructura_datos y datos_introducir), importa las dos tablas. Muestra las principales características del esquema creado y explica las diferentes tablas y variables que existen. Asegúrate de incluir un diagrama que ilustre la relación entre las distintas tablas y variables	1
1.2. Utilizando JOIN realizarás las siguientes consultas	4
1.2.1. Listado de los países que están realizando compras	4
1.2.2. Desde cuántos países se realizan las compras	4
1.2.3. Identifica a la compañía con la mayor media de ventas	5
1.3. Utilizando sólo subconsultas (sin utilizar JOIN)	5
1.3.1. Muestra todas las transacciones realizadas por empresas de Alemania	5
1.3.2. Lista las empresas que han realizado transacciones por un amount superior a la media de todas las transacciones	7
1.3.3. Eliminarán del sistema las empresas que carecen de transacciones registradas, entrega el listado de estas empresas	9
<b>2. Nivel 2</b>	<b>10</b>
2.1. 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	10
2.2. ¿Cuál es la media de ventas por país? Presenta los resultados ordenados de mayor a menor medio	11
2.3. 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	12
2.3.1. Muestra el listado aplicando JOIN y subconsultas	12
2.3.2. Muestra el listado aplicando solo subconsultas	13
<b>3. Nivel 3</b>	<b>14</b>
3.1. 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 de 2021, 20 de julio de 2021 y 13 de marzo de 2022. Ordena los resultados de mayor a menor cantidad	14
3.2. 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 en las que especifiques si tienen más de 4 transacciones o menos	15

## 1. Nivel 1

- 1.1. A partir de los documentos adjuntos (estructura\_datos y datos\_introducir), importa las dos tablas. Muestra las principales características del esquema creado y explica las diferentes tablas y variables que existen. Asegúrate de incluir un diagrama que ilustre la relación entre las distintas tablas y variables.

### Tabla **company**

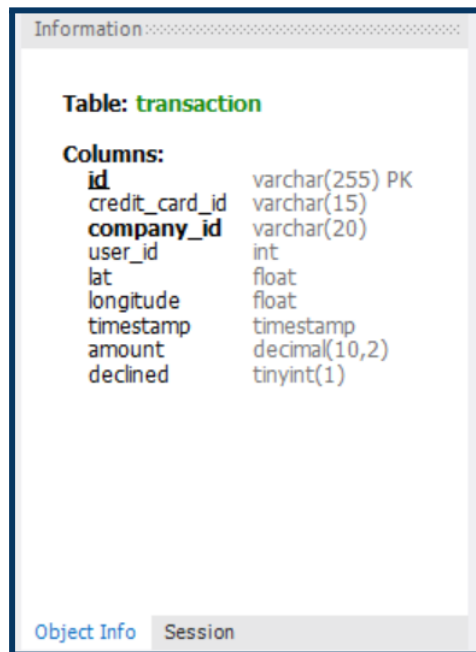


Information	
Table: <b>company</b>	
Columns:	
<b>id</b>	varchar(15) PK
company_name	varchar(255)
phone	varchar(15)
email	varchar(100)
country	varchar(100)
website	varchar(255)

Object Info Session

- **Descripción:** Almacena información sobre empresas.
- **Clave primaria:**
  - **id** (VARCHAR(255))
- **Otros campos:**
  - **company\_name** (VARCHAR(255)) → Nombre de la empresa.
  - **phone** (VARCHAR(15)) → Teléfono de contacto.
  - **email** (VARCHAR(100)) → Correo electrónico.
  - **country** (VARCHAR(100)) → País de la empresa.
  - **website** (VARCHAR(255)) → Página web de la empresa.

## 📌 Tabla **transactions**

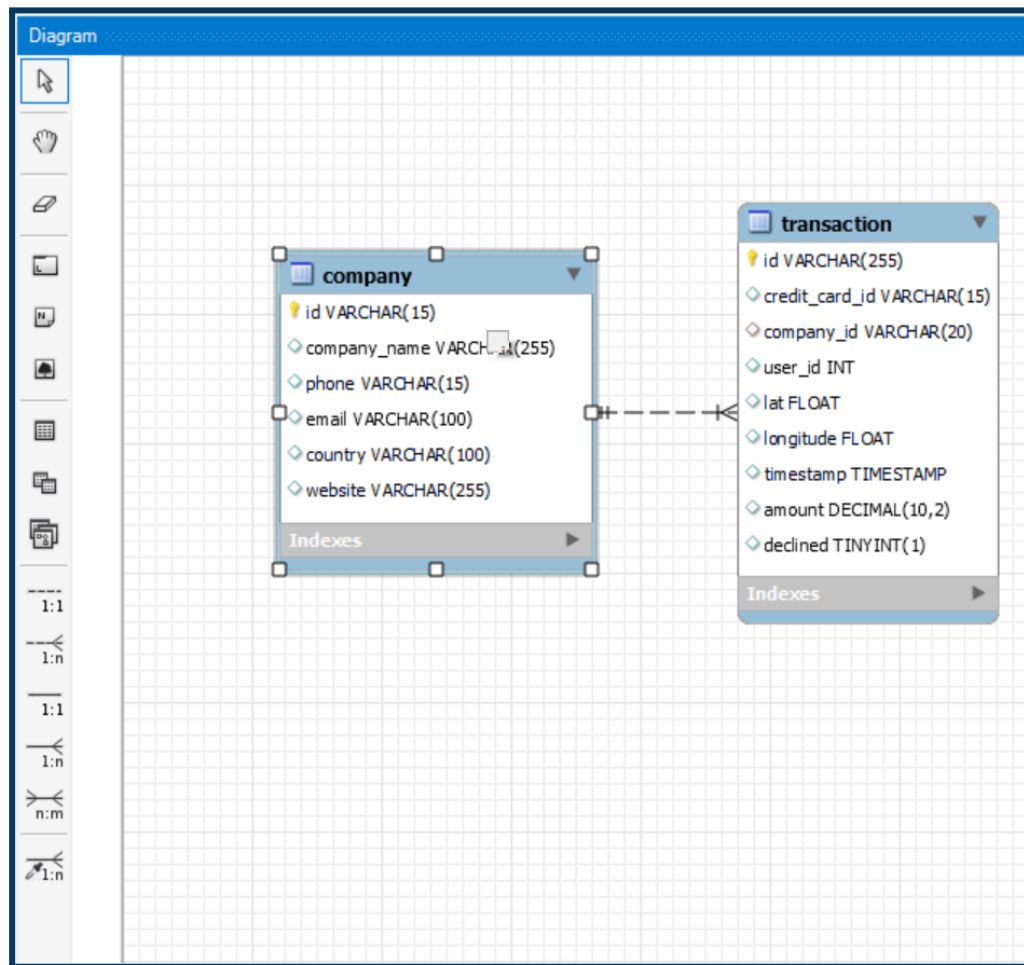


The screenshot shows a window titled 'Information' with a tab 'Object Info' selected. It displays the structure of a table named 'transaction'. The columns and their data types are listed as follows:

Table: transaction	
Columns:	
<b>id</b>	varchar(255) PK
credit_card_id	varchar(15)
<b>company_id</b>	varchar(20)
user_id	int
lat	float
longitude	float
timestamp	timestamp
amount	decimal(10,2)
declined	tinyint(1)

- **Descripción:** Almacena información sobre transacciones financieras.
- **Clave primaria:**
  - **id** (VARCHAR(255))
- **Claves foráneas:**
  - **company\_id** (VARCHAR(20)) → Referencia a **company(id)**.
  - **credit\_card\_id** (VARCHAR(15)) → Referencia a una tabla no incluida en este esquema.
  - **user\_id** (INT) → Referencia a una tabla no incluida en este esquema.
- **Otros campos:**
  - **lat** (FLOAT) → Latitud de la transacción (ubicación).
  - **longitude** (FLOAT) → Longitud de la transacción (ubicación).
  - **timestamp** (TIMESTAMP) → Fecha y hora de la transacción.
  - **amount** (DECIMAL(10,2)) → Monto de la transacción.
  - **declined** (TINYINT(1)) → Indica si la transacción fue rechazada (1 = sí, 0 = no).

## Relación y diagrama ER



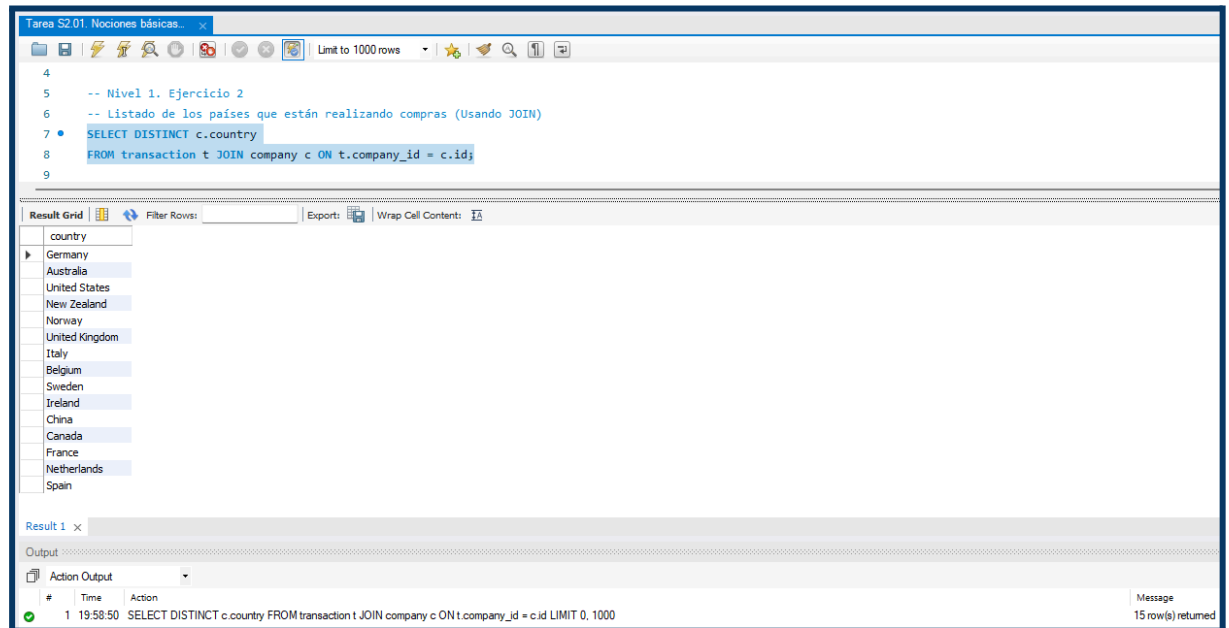
El esquema muestra una relación uno a muchos entre **company** y **transactions**, lo que indica que una empresa puede tener múltiples transacciones asociadas. En cada transacción, además de los datos propios, se incluirá el **id** de la empresa a la que pertenece como clave foránea y forma de relación.

Nota: Las claves foráneas **credit\_card\_id** y **user\_id** hacen referencia a tablas externas que no están definidas en este esquema.

## 1.2. Utilizando JOIN realizarás las siguientes consultas

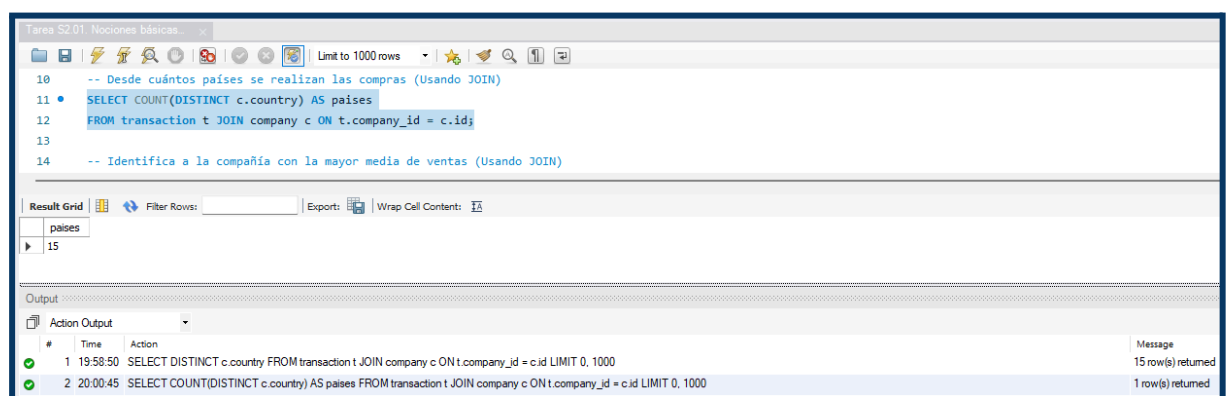
### 1.2.1. Listado de los países que están realizando compras.

```
SELECT DISTINCT c.country  
FROM transaction t JOIN company c ON t.company_id = c.id
```



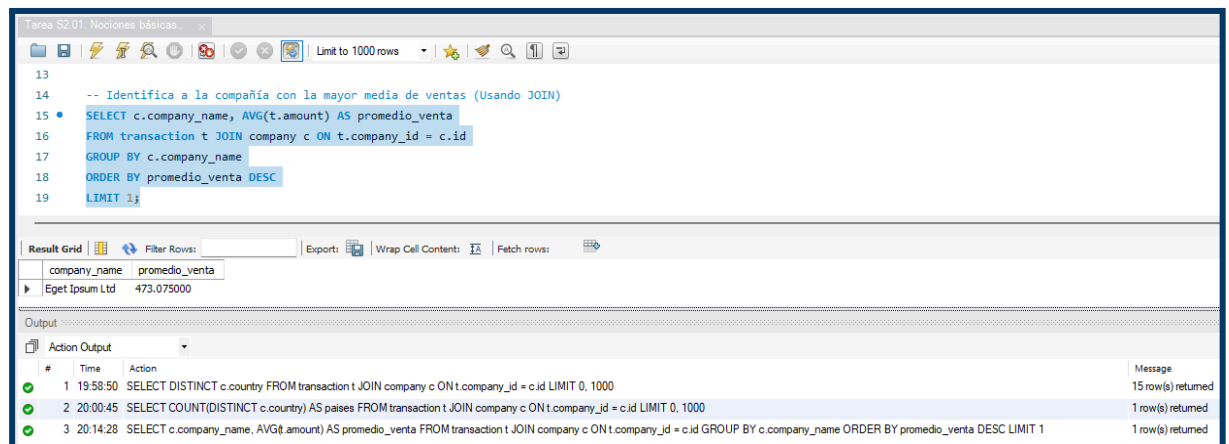
### 1.2.2. Desde cuántos países se realizan las compras.

```
SELECT COUNT(DISTINCT c.country) AS paises  
FROM transaction t JOIN company c ON t.company_id = c.id
```



### 1.2.3. Identifica a la compañía con la mayor media de ventas.

```
SELECT c.company_name, avg(t.amount) AS promedio_venta
FROM transaction t JOIN company c ON t.company_id = c.id
GROUP BY c.company_name
Order By promedio_venta DESC
LIMIT 1
```



### 1.3. Utilizando sólo subconsultas (sin utilizar JOIN)

#### 1.3.1. Muestra todas las transacciones realizadas por empresas de Alemania.

- Si se refiere a todas las transacciones registradas

```
SELECT *
FROM transaction t
WHERE t.company_id IN (SELECT c.id
                       FROM company c
                       WHERE c.country = "Germany");
```

- Si se refiere únicamente a las transacciones que fueron realizadas con éxito, es decir, que no fueron rechazadas.

```
SELECT *
FROM transaction t
WHERE t.declined=0
AND t.company_id IN (SELECT c.id
                     FROM company c
                     WHERE c.country = "Germany");
```

Tarea 52.01. Nociones básicas.

```

22 -- Muestra todas las transacciones realizadas por empresas de Alemania (Sin usar JOIN)
23 SELECT *
24 FROM transaction t
25 WHERE /*t.declined=0 AND*/ t.company_id IN (SELECT c.id
26 FROM company c
27 WHERE c.country = "Germany");
28 /*NOTA: Si "transacciones realizadas" se refiere unicamente a aquellas que fueron realizadas con éxito,
29 es decir, que no fueron rechazadas, basta con descomentar el segmento "t.declined=0 AND"*/
30

```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
1088 ID ID-SB23-A76C-55EF-C568E49A05DD	CdJ-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57	0
EA2C3281-C9C1-A387-44F8-729F84B51C76	CdJ-2938	b-2222	275	20.2004	-116.84	2021-05-09 10:25:08	119.36	1
00D2E608-5C9E-D1B3-4999-899F43AD735A	CdJ-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47	1
A8069F53-965E-A2A8-CE06-CAB8C4FD92501	CdJ-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99	0
0466A42E-47CF-8D24-FD01-C0B89713128	CdJ-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53	0
0A476ED9-0C13-1962-F87B-D3563924B539	CdJ-4359	b-2302	221	-56.4901	114.801	2022-02-26 20:33:54	430.49	0
122D333-E19F-D629-DCD8-9C54CF1EBB9A	CdJ-4366	b-2302	221	29.6372	-166.173	2021-06-09 06:04:14	172.01	0
1352678A-2E7D-957C-C42C-6450A2B3ED54	CdJ-4520	b-2302	210	20.6724	14.9732	2021-12-29 20:38:23	17.97	0
14CAE5B5-8F81-3E4A-C485-0EA416753F4C	CdJ-4849	b-2302	189	-53.6202	93.0533	2021-12-31 00:29:42	388.04	0
158A3ACB-541C-0BCC-65B0-6373C67BF1C	CdJ-4849	b-2302	183	42.5424	-170.347	2022-03-08 05:02:19	240.29	0
162C7E78-2B6B-7971-A1E4-D2124E732451	CdJ-4527	b-2302	210	-49.1381	58.0017	2021-04-11 05:59:18	231.26	0
1717FD68-ADAD-7082-A748-D112BE892CCC	CdJ-4219	b-2302	172	69.4892	-138.411	2021-12-29 16:18:54	249.91	0
18521826-9FC1-52E6-5C39-A1FFB9780D3A	CdJ-4345	b-2302	222	57.9422	-114.729	2021-08-17 05:32:08	497.84	0
186F53DE-DE27-81FE-8B2F-15B61CEB7726	CdJ-4310	b-2302	225	-72.7448	36.6211	2021-12-20 13:13:45	238.16	0
18C4E2DD-1E4C-F35E-2198-C660BD81DC25	CdJ-4849	b-2302	177	68.0133	91.4839	2021-09-24 18:55:25	237.04	0
18CDBA7C-ABC1-813D-FAF3-4B897429368	CdJ-4219	b-2302	173	51.3881	-156.371	2021-06-21 03:21:34	58.16	0
19E1EC3E-2119-1EFD-8AAE-593004A4E63F	CdJ-4219	b-2302	154	-56.0839	116.987	2021-07-06 21:40:15	29.63	0
1B117D49-936C-BA6C-E94B-30C3293AA239	CdJ-4219	b-2302	163	-75.099	109.034	2021-05-09 23:56:04	371.35	0
18521826-5860-5A86-5364-6EB6A5CC21B7	CdJ-4226	b-2302	231	-53.4613	49.1484	2021-08-14 02:39:50	476.33	0
1C3B3CC4-8919-7616-6A57-EEBDD7B7B688	CdJ-4415	b-2302	217	81.1737	62.8835	2021-09-27 16:07:34	462.35	0
207B81D0-694B-7022-57B8-33ED49B72766	CdJ-4247	b-2302	229	-9.03682	88.9005	2022-02-13 04:07:29	87.44	0
241827A6-E87C-4D43-5AC9-8BFCF042B702A	CdJ-4849	b-2302	190	-12.1172	-175.322	2021-08-04 04:44:35	487.64	0
2585891B-C4C7-075E-3586-8575339E5177	CdJ-4450	b-2302	215	32.9171	-40.6059	2021-08-24 05:32:36	253.49	0
263035AA-8758-E413-8CC4-4D08F81B8F899	CdJ-4338	b-2302	223	36.61	83.1953	2021-06-21 11:23:01	499.23	0
2C9A4725-4B19-421C-4EE4-AE9E31CFE5B0	CdJ-4219	b-2302	149	-46.7357	162.102	2021-08-14 09:23:34	28.95	0
33529F74-4864-C19F-7D23-A8D78668B7C4	CdJ-4492	b-2302	212	10.3105	13.9999	2021-12-15 12:22:03	423.71	0
34368DE0-D1B1-1B5C-6698-8DF58C06412D	CdJ-4219	b-2302	153	-8.20452	89.9882	2021-07-05 10:54:54	86.69	0
34A11D64-2989-DE1C-8705-C66C8D0B190D	CdJ-4219	b-2302	166	-17.0772	-169.006	2022-03-02 05:30:41	275.45	0
369213C7-8D65-4715-6644-05E4F9D90914	CdJ-4219	b-2302	147	88.7859	-148.023	2021-12-29 18:54:29	458.66	0
3A3E00C1-C191-02EF-AD8E-42C9DA644E4A	CdJ-4849	b-2302	175	0.340013	174.033	2021-11-10 19:14:50	451.66	0
3E495EC1-E1C5-26F1-0E3E-2D13EAA2593A	CdJ-4282	b-2302	227	-50.5725	-172.508	2022-03-03 14:29:15	417.70	0
41932C77-45B5-C786-6B3F-D8B619B2D535	CdJ-4219	b-2302	142	-8.37247	-67.8776	2022-02-12 09:12:05	358.36	0
442713DA-4BE6-132A-51B8-AC4EA6D5839	CdJ-4436	b-2302	216	-22.3294	-22.0184	2022-01-29 17:12:59	59.67	0
5218E51C-E258-8FA4-8E23-AF815F66FAF8	CdJ-4219	b-2302	157	18.9732	66.3985	2021-09-07 21:38:45	234.52	0
59BC388F-9D74-8B8A-A11D-D604A8C086F2	CdJ-4352	b-2302	222	61.4687	-29.7214	2021-03-25 21:48:18	89.22	0
5AB20418-79A8-DF6C-D446-8CF5A48CDEDB	CdJ-4541	b-2302	210	65.1786	-151.454	2021-08-31 05:44:29	474.70	0
5B60F39F-96C8-8C8D-EA01-E658063A8181	CdJ-4233	b-2302	230	-85.0035	-65.4575	2022-02-28 05:55:03	67.65	0
5B6A81852-D860-169A-805B-E1FA120E9683	CdJ-4471	b-2302	213	7.30204	72.5522	2022-03-13 13:37:34	164.32	0
5CB0C889-D096-0327-ADE7-D11E1E6E2BC	CdJ-4198	b-2302	233	-58.1208	137.506	2021-04-17 12:27:01	304.63	0
5E948202-34BC-F107-29DC-306A2C491A87	CdJ-4478	b-2302	213	78.9252	-137.822	2021-09-11 14:27:14	133.80	0
62A399E3-0FDD-2E65-5A95-21D8C59C33DC	CdJ-4219	b-2302	156	-61.2863	-4.31831	2021-05-14 16:10:09	22.92	0
632B8EE9-4725-E15C-7FC4-5B5D7EC58D29	CdJ-3078	b-2302	275	45.8951	-58.707	2021-08-27 22:13:25	203.23	0
65A7C3AC-FFFC-3495-95B1-52C38507E523	CdJ-4443	b-2302	215	-78.2398	-9.2237	2021-12-08 15:29:30	387.27	0
65F57592-832E-1D2E-3466-57D7F1691D0D	CdJ-4849	b-2302	178	-48.6612	67.8562	2021-06-15 06:26:04	476.75	0
65F73281-0AC6-CE48-6667-1C8E66159728	CdJ-4422	b-2302	217	-12.1197	-5.92885	2021-12-05 04:46:31	329.72	0

transaction 4 x

Output

Action Output

#	Time	Action	Message
1	19:58:50	SELECT DISTINCT c.country FROM transaction t JOIN company c ON t.company_id = c.id LIMIT 0, 1000	15 row(s) returned
2	20:00:45	SELECT COUNT(DISTINCT c.country) AS paises FROM transaction t JOIN company c ON t.company_id = c.id LIMIT 0, 1000	1 row(s) returned
3	20:14:28	SELECT c.company_name, AVG(t.amount) AS promedio_venta FROM transaction t JOIN company c ON t.company_id = c.id GROUP BY c.company_name ORDER BY promedio_venta DESC LIMIT 1	1 row(s) returned
4	20:30:41	SELECT * FROM transaction t WHERE /*t.declined=0 AND*/ t.company_id IN (SELECT c.id FROM company c WHERE c.country = "Germany") LIMIT 0, 1000	118 row(s) returned

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

- Versión 1: Usando operador IN

```
SELECT c.company_name
FROM company c
WHERE c.id IN (SELECT t.company_id
               FROM transaction t
               WHERE t.amount > (SELECT AVG(t1.amount)
                                FROM transaction t1));
```

The screenshot shows a SQL IDE window titled 'Tarea S2.01. Nociones básicas...'. The query editor contains two SQL queries. The first query, labeled 'V1 Con operador IN', is highlighted and matches the query in the previous block. The second query, labeled 'V2 Con operador ANY', is also shown. Below the query editor, the 'Result Grid' shows a list of company names. The 'Output' pane at the bottom shows the execution log with five entries, including the execution of the first query which returned 70 rows.

```
-- Lista las empresas que han realizado transacciones por un amount superior a la media de todas las transacciones. (Sin usar JOIN)
-- V1 Con operador IN
SELECT c.company_name
FROM company c
WHERE c.id IN (SELECT t.company_id
               FROM transaction t
               WHERE t.amount > (SELECT AVG(t1.amount)
                                FROM transaction t1));
-- V2 Con operador ANY
SELECT c.company_name
FROM company c
WHERE c.id = ANY (SELECT t.company_id
                  FROM transaction t
                  WHERE t.amount > (SELECT AVG(t1.amount)
                                    FROM transaction t1));
```

company_name
Ac Fermentum Incorporated
Magna A Neque Industries
Fusce Corp.
Ante Jaculis 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
Arcu LLP
Fringilla LLC
Elit Etiam Laoreet Associates
Nunc Interdum Incorporated
Augue Foundation
Non Magna LLC
A Institute
Quam A Felis Industries
Integer Mollis Corp.
Enim Condimentum Ltd
Donec Fringilla PC
Amet Institute
Magna Incorporated
Fringilla Parturient Incorporated
Ut Semper Foundation
Ac Industries
Lorem Eu Incorporated
Mauris Institute
Erat LLP
Neque Tellus Incorporated
Lacus Quisque Associates
Neque Tellus Imperdiet Corp.
Tortor Nunc Commodo Comp...

#	Time	Action	Message
1	19:58:50	SELECT DISTINCT c.country FROM transaction t JOIN company c ON t.company_id = c.id LIMIT 0, 1000	15 row(s) returned
2	20:00:45	SELECT COUNT(DISTINCT c.country) AS paises FROM transaction t JOIN company c ON t.company_id = c.id LIMIT 0, 1000	1 row(s) returned
3	20:14:28	SELECT c.company_name, AVG(t.amount) AS promedio_venta FROM transaction t JOIN company c ON t.company_id = c.id GROUP BY c.company_name ORDER BY promedio_venta DESC LIMIT 1	1 row(s) returned
4	20:30:41	SELECT * FROM transaction t WHERE /t.declined=0 AND/ t.company_id IN (SELECT c.id FROM company c WHERE c.country = "Germany") LIMIT 0, 1000	118 row(s) returned
5	20:40:36	SELECT c.company_name FROM company c WHERE c.id IN (SELECT t.company_id FROM transaction t WHERE t.amount > (SELECT AVG(t1.amount) FROM transaction t1)) LIMIT 0, 1000	70 row(s) returned



- Versión 2: Usando operador ANY

```
SELECT c.company_name
FROM company c
WHERE c.id = ANY (SELECT t.company_id
                  FROM transaction t
                  WHERE t.amount > (SELECT AVG(t1.amount)
                                   FROM transaction t1));
```

The screenshot shows a SQL IDE with two queries being compared. The first query (V1) uses the IN operator, and the second query (V2) uses the ANY operator. Both queries aim to list companies whose transaction amounts are greater than the average transaction amount of all companies. The results show that both queries return the same set of 70 companies.

**Query 1 (V1):**

```
-- V1 Con operador IN
SELECT c.company_name
FROM company c
WHERE c.id IN (SELECT t.company_id
              FROM transaction t
              WHERE t.amount > (SELECT AVG(t1.amount)
                               FROM transaction t1));
```

**Query 2 (V2):**

```
-- V2 Con operador ANY
SELECT c.company_name
FROM company c
WHERE c.id = ANY (SELECT t.company_id
                 FROM transaction t
                 WHERE t.amount > (SELECT AVG(t1.amount)
                                   FROM transaction t1));
```

**Result Grid:**

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 Aenean Eget Foundation
Dis Parturient Institute
Sed LLC
Arcu LLP
Fringilla LLC
Elit Etiam Laoreet Associates
Nunc Interdum Incorporated
Augue Foundation
Non Magna LLC
A Institute
Quam A Felis Industries
Integer Mollis Corp.
Enim Condimentum Ltd
Donec Fringilla PC
Amet Institute
Magna Incorporated
Fringilla Porttitor Incorporated
Ut Semper Foundation
Ac Industries
Lorem Eu Incorporated
Mauris Institute
Erat LLP
Neque Tellus Incorporated
Lacus Quisque Associates
Neque Tellus Imperdiet Corp.
Tortor Nunc Commodo Comp...

**Output Log:**

#	Time	Action	Message
1	19:58:50	SELECT DISTINCT c.country FROM transaction t JOIN company c ON t.company_id = c.id LIMIT 0, 1000	15 row(s) returned
2	20:00:45	SELECT COUNT(DISTINCT c.country) AS paises FROM transaction t JOIN company c ON t.company_id = c.id LIMIT 0, 1000	1 row(s) returned
3	20:14:28	SELECT c.company_name, AVG(t.amount) AS promedio_venta FROM transaction t JOIN company c ON t.company_id = c.id GROUP BY c.company_name ORDER BY promedio_venta DESC LIMIT 1	1 row(s) returned
4	20:30:41	SELECT * FROM transaction t WHERE (/t.declined=0 AND)/t.company_id IN (SELECT c.id FROM company c WHERE c.country = "Germany") LIMIT 0, 1000	118 row(s) returned
5	20:40:36	SELECT c.company_name FROM company c WHERE c.id IN (SELECT t.company_id FROM transaction t WHERE t.amount > (SELECT AVG(t1.amount) FROM transaction t1)) LIMIT 0, 1000	70 row(s) returned
6	20:44:02	SELECT c.company_name FROM company c WHERE c.id = ANY (SELECT t.company_id FROM transaction t WHERE t.amount > (SELECT AVG(t1.amount) FROM transaction t1)) LIMIT 0, 1000	70 row(s) returned

1.3.3. Eliminarán del sistema las empresas que carecen de transacciones registradas, entrega el listado de estas empresas.

```
SELECT c.id, c.company_name
FROM company c
WHERE NOT EXISTS (SELECT 1
                  FROM transaction t
                  WHERE t.company_id = c.id);
```

The screenshot shows a SQL IDE window titled 'Tarea S2.01. Nociones básicas...'. The query editor contains the following SQL code:

```
46 FROM transaction t1));
47
48 -- Eliminarán del sistema las empresas que carecen de transacciones registradas, entrega el listado de estas empresas. (Sin usar JOIN)
49 -- Con operador EXISTS
50 SELECT c.id, c.company_name
51 FROM company c
52 WHERE NOT EXISTS (SELECT 1
53                  FROM transaction t
54                  WHERE t.company_id = c.id);
55
```

Below the query editor, the 'Result Grid' is visible, showing a table with two columns: 'id' and 'company\_name'. The table is currently empty.

At the bottom, the 'Output' pane shows the execution log. The log contains the following entries:

#	Time	Action	Message
1	19:58:50	SELECT DISTINCT c.country FROM transaction t JOIN company c ON t.company_id = c.id LIMIT 0, 1000	15 row(s) returned
2	20:00:45	SELECT COUNT(DISTINCT c.country) AS paises FROM transaction t JOIN company c ON t.company_id = c.id LIMIT 0, 1000	1 row(s) returned
3	20:14:28	SELECT c.company_name, AVG(t.amount) AS promedio_venta FROM transaction t JOIN company c ON t.company_id = c.id GROUP BY c.company_name ORDER BY promedio_venta DESC LIMIT 1	1 row(s) returned
4	20:30:41	SELECT * FROM transaction t WHERE /t.declined=0 AND/ t.company_id IN (SELECT c.id FROM company c WHERE c.country = 'Germany') LIMIT 0, 1000	118 row(s) returned
5	20:40:36	SELECT c.company_name FROM company c WHERE c.id IN (SELECT t.company_id FROM transaction t WHERE t.amount > (SELECT AVG(t1.amount) FROM transaction t1)) LIMIT 0, 1000	70 row(s) returned
6	20:44:02	SELECT c.company_name FROM company c WHERE c.id = ANY (SELECT t.company_id FROM transaction t WHERE t.amount > (SELECT AVG(t1.amount) FROM transaction t1)) LIMIT 0, 1000	70 row(s) returned
7	20:56:12	SELECT c.id, c.company_name FROM company c WHERE NOT EXISTS (SELECT 1 FROM transaction t WHERE t.company_id = c.id) LIMIT 0, 1000	0 row(s) returned

## 2. Nivel 2

- 2.1. 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.

```
SELECT date(t.timestamp) AS fecha, SUM(t.amount) AS total_ventas
FROM transaction t
GROUP BY fecha
ORDER BY total_ventas DESC
LIMIT 5;
```

The screenshot shows a SQL IDE window titled 'Tarea S2.01. Nociones básicas...'. The query editor contains the following SQL code:

```
-- 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.
SELECT date(t.timestamp) AS fecha, SUM(t.amount) AS total_ventas
FROM transaction t
GROUP BY fecha
ORDER BY total_ventas DESC
LIMIT 5;
```

The 'Result Grid' shows the following data:

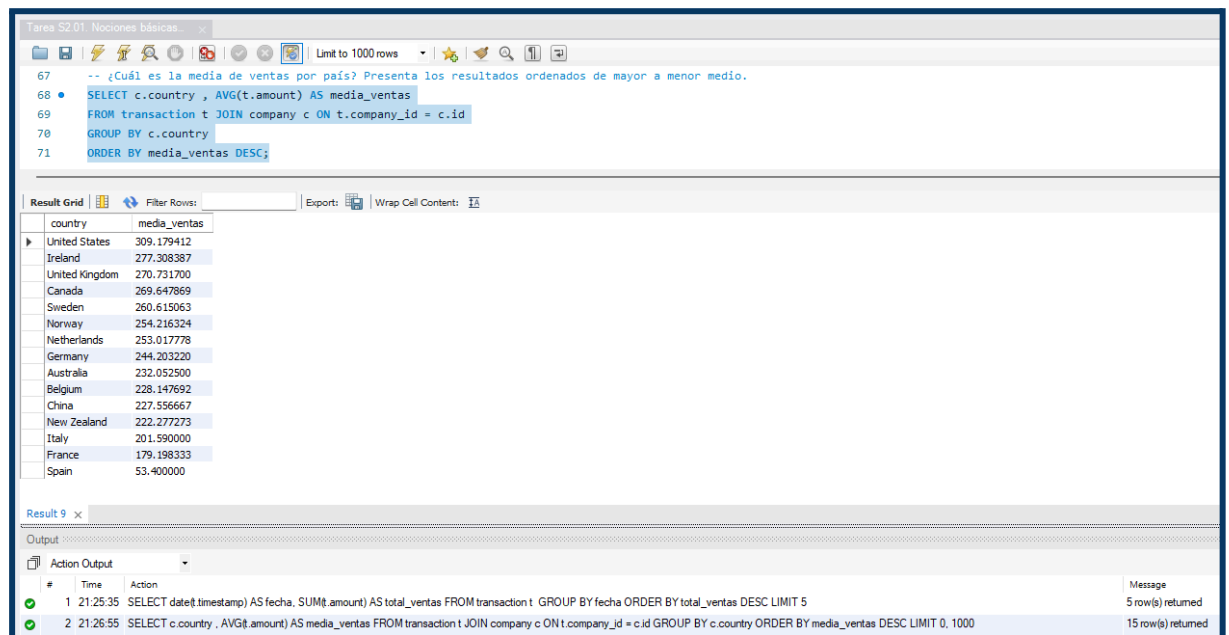
fecha	total_ventas
2021-03-29	1564.87
2021-12-20	1532.36
2021-06-15	1469.90
2021-05-09	1463.73
2021-06-21	1443.11

The 'Output' section shows the execution message:

```
1 21:25:35 SELECT date(t.timestamp) AS fecha, SUM(t.amount) AS total_ventas FROM transaction t GROUP BY fecha ORDER BY total_ventas DESC LIMIT 5
Message
5 row(s) returned
```

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

```
SELECT c.country , AVG(t.amount) AS media_ventas
FROM transaction t JOIN company c ON t.company_id = c.id
GROUP BY c.country
ORDER BY media_ventas DESC;
```



The screenshot shows a database management tool interface. At the top, a SQL query is entered in a text area. Below the query, a 'Result Grid' displays the results of the query. The grid has two columns: 'country' and 'media\_ventas'. The results are ordered by 'media\_ventas' in descending order. Below the grid, an 'Output' section shows the execution log with two entries, each indicating the time taken and the number of rows returned.

country	media_ventas
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

#	Time	Action	Message
1	21:25:35	SELECT date(timestamp) AS fecha, SUM(amount) AS total_ventas FROM transaction t GROUP BY fecha ORDER BY total_ventas DESC LIMIT 5	5 row(s) returned
2	21:26:55	SELECT c.country , AVG(t.amount) AS media_ventas FROM transaction t JOIN company c ON t.company_id = c.id GROUP BY c.country ORDER BY media_ventas DESC LIMIT 0, 1000	15 row(s) returned

- 2.3. 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.

### 2.3.1. Muestra el listado aplicando JOIN y subconsultas.

```
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')
ORDER BY t.company_id;
```

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

```
-- Muestra el listado aplicando JOIN y subconsultas.
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')
ORDER BY t.company_id;
```

The results grid displays a list of transactions for companies located in the United Kingdom. The columns include transaction ID, credit card ID, company ID, user ID, latitude, longitude, timestamp, amount, declined status, company name, phone, email, country, and website. The results are ordered by company ID.

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	id	company_name	phone	email	country	website
28928E1C-EC14-A760-0A75-871477649D6A	CUJ-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	ribh@yahoo.org	United Kingdom	https://crm.com/one
ACD2011A-A2B1-C365-41E1-2A800C55147A	CUJ-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	ribh@yahoo.org	United Kingdom	https://crm.com/one
43D4D9E-CB9-3D68-A4DA-FEB7738A1ACE	CUJ-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	nel.quisque.fringilla@hotmail.ca	United Kingdom	https://whatsapp.com/site
BC289A38-778A-28CD-1FE8-14ED863E773	CUJ-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	nel.quisque.fringilla@hotmail.ca	United Kingdom	https://whatsapp.com/site
147983D2-678A-C78B-4CE3-8D7C26E85A8B	CUJ-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58:07	309.45	0	b-2326	Erin Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yahoo.net	United Kingdom	https://crm.com/group/9
152598C2-029D-D684-4666-91EDF393EBFF	CUJ-2994	b-2326	126	-67.0189	-141.672	2021-07-05 03:10:00	395.43	0	b-2326	Erin Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yahoo.net	United Kingdom	https://crm.com/group/9
18636858-A2E8-7C69-D9C9-C54533DAFD38	CUJ-2994	b-2326	131	70.2543	-13.1336	2021-07-06 08:48:46	195.06	0	b-2326	Erin Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yahoo.net	United Kingdom	https://crm.com/group/9
20418DE5-6804-8E36-8D7A-A95C18FD8F3C	CUJ-2994	b-2326	126	-79.1145	1.51481	2022-01-03 15:59:29	479.52	0	b-2326	Erin Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yahoo.net	United Kingdom	https://crm.com/group/9
23883176-KC06-137A-C2F6-3180A188A2D3	CUJ-2994	b-2326	126	23.6174	137.222	2021-08-26 06:04:05	43.99	0	b-2326	Erin Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yahoo.net	United Kingdom	https://crm.com/group/9
267C4A86-78A7-1C5E-0718-2824983C8700	CUJ-2994	b-2326	126	-17.5259	104.915	2021-10-01 21:08:53	122.63	0	b-2326	Erin Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yahoo.net	United Kingdom	https://crm.com/group/9
3142C93E-6387-49E4-EE2D-29CA834B188D	CUJ-2994	b-2326	126	-67.8476	-119.978	2021-04-06 17:24:44	91.59	0	b-2326	Erin Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yahoo.net	United Kingdom	https://crm.com/group/9
3578688E-781D-8887-38C7-2088673AA31E	CUJ-2994	b-2326	126	87.0665	-22.7339	2021-04-06 17:24:44	303.60	0	b-2326	Erin Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yahoo.net	United Kingdom	https://crm.com/group/9
360C7814-77AF-843A-0946-AB38D2683C86	CUJ-2994	b-2326	116	-7.93005	-79.0733	2021-08-21 10:19:58	494.82	0	b-2326	Erin Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yahoo.net	United Kingdom	https://crm.com/group/9
391E1CFD-D653-E458-A729-F2E893247858	CUJ-2994	b-2326	118	60.5512	103.904	2021-10-09 00:50:38	271.27	0	b-2326	Erin Condimentum Ltd	09 55 51 66 25	imperdiet.non.vestibulum@yahoo.net	United Kingdom	https://crm.com/group/9

The output section shows the execution of the query and the number of rows returned for each step:

#	Time	Action	Message
1	21:25:35	SELECT date(timestamp) AS fecha, SUM(amount) AS total_ventas FROM transaction t GROUP BY fecha ORDER BY total_ventas DESC LIMIT 5	5 row(s) returned
2	21:26:55	SELECT c.country, AVG(amount) AS media_ventas FROM transaction t JOIN company c ON t.company_id = c.id GROUP BY c.country ORDER BY media_ventas DESC LIMIT 0, 1000	15 row(s) returned
3	21:28:35	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') ORDER BY t.company_id LIMIT 100	100 row(s) returned

### 2.3.2. Muestra el listado aplicando solo subconsultas.

```
SELECT *
FROM transaction t
WHERE t.company_id IN (SELECT c.id
                       FROM company c
                       WHERE c.country =
                         (SELECT country FROM company
                          WHERE company_name = 'Non Institute'))
ORDER BY t.company_id;
```

The screenshot shows a database management tool interface. The top pane displays a SQL query: `-- Muestra el listado aplicando solo subconsultas. SELECT * FROM transaction t WHERE t.company_id IN (SELECT c.id FROM company c WHERE c.country = (SELECT country FROM company WHERE company_name = 'Non Institute')) ORDER BY t.company_id;`. The bottom pane shows the 'Result Grid' with columns: id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, amount, and dedined. It contains 20 rows of data. Below the grid, the 'Output' pane shows the execution log with four messages, including the final query execution: `SELECT * FROM transaction t WHERE t.company_id IN (SELECT c.id FROM company c WHERE c.country = (SELECT country FROM company WHERE company_name = 'Non Institute')) ORDER BY ...`.

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	dedined
2B928E1C-EC14-A760-0A75-871477649D6A	CdJ-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14:49	383.73	0
ACD2011A-A2B1-C365-41E1-2A800C65147A	CdJ-2980	b-2246	275	-54.4792	-82.7974	2022-03-05 20:41:20	60.07	1
4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	CdJ-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37:23	458.74	0
BC2B9A38-77B4-28CD-1FEB-14DED863E773	CdJ-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95	1
1479B3D2-87BA-C7B8-4CE3-8D7CDE85A8B	CdJ-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58:07	309.45	0
152598C2-029D-0684-4B66-91EDF393EBFF	CdJ-2994	b-2326	126	-67.0189	-141.672	2021-07-05 03:10:00	395.43	0
1B636B58-42E8-7C69-D9C9-C5453D5AFD3B	CdJ-2994	b-2326	131	70.2543	-13.1336	2021-07-06 08:48:46	195.06	0
20418DE5-8804-8E98-8D7A-A95C1BFD6F5C	CdJ-2994	b-2326	126	-79.1145	1.51481	2022-01-03 15:59:29	479.52	0
239B8576-6C0E-137A-C2F6-3180A188A2D3	CdJ-2994	b-2326	126	23.6174	137.222	2021-08-26 06:04:05	43.90	0
267C4A86-7BA7-1C9E-0718-2824983C87D0	CdJ-2994	b-2326	126	-17.5259	104.915	2021-10-01 21:08:53	122.63	0
3142C93E-83B7-49E4-EE2D-29CA834B1980	CdJ-2994	b-2326	126	-67.8476	-119.978	2021-04-06 17:24:44	91.59	0
357B688E-7B1D-8887-38C7-20B8673AA31E	CdJ-2994	b-2326	126	87.0665	-22.7339	2021-07-26 22:59:24	303.60	0
360C7B14-F7AF-843A-0946-AB38D2583C86	CdJ-2994	b-2326	116	-7.93005	-79.0733	2021-08-21 10:19:58	494.82	0
391E1CFD-0653-E45B-A729-F2EB93247B58	CdJ-2994	b-2326	118	60.5512	103.904	2021-10-09 00:50:38	271.27	0

### 3. Nivel 3

- 3.1. 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 de 2021, 20 de julio de 2021 y 13 de marzo de 2022. Ordena los resultados de mayor a menor cantidad.

- Si se refiere a todas las transacciones registradas

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

- Si se refiere únicamente a las transacciones que fueron realizadas con éxito, es decir, que no fueron rechazadas.

```
SELECT c.company_name, c.phone, c.country, date(t.timestamp) AS fecha, t.amount
FROM company c JOIN transaction t ON c.id = t.company_id
WHERE t.amount BETWEEN 100 AND 200
AND date(t.timestamp) IN ('2021-04-29', '2021-07-20', '2022-03-13')
AND t.declined = 0
ORDER BY t.amount DESC;
```

The screenshot shows a SQL IDE window titled 'Tarea S2.01. Nociones básicas...'. The query editor contains the following SQL code:

```
-- 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 de 2021, 20 de julio de 2021 y 13 de marzo de 2022.
Ordena los resultados de mayor a menor cantidad.*/
SELECT c.company_name, c.phone, c.country, date(t.timestamp) AS fecha, t.amount
FROM company c JOIN transaction t ON c.id = t.company_id
WHERE t.amount BETWEEN 100 AND 200
AND date(t.timestamp) IN ('2021-04-29', '2021-07-20', '2022-03-13')
-- AND t.declined = 0
ORDER BY t.amount DESC;
```

The results are displayed in a table with the following columns: company\_name, phone, country, fecha, and amount. The results are ordered by amount in descending order.

company_name	phone	country	fecha	amount
Interdum Feugiat Sed Associates	04 88 40 32 52	United Kingdom	2021-07-20	164.8
Nunc Interdum Incorporated	05 18 15 48 13	Germany	2022-03-13	164.32
Enim Conditum 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 'Output' section with the following message: '1 21:32:14 SELECT c.company\_name, c.phone, c.country, date(t.timestamp) AS fecha, t.amount FROM company c JOIN transaction t ON c.id = t.company\_id WHERE t.amount BETWEEN 100 AND 200 AND date(t.timestamp) IN ('2021-04-29', '2021-07-20', '2022-03-13') 5 row(s) returned'.

- 3.2. 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 en las que especifiques si tienen más de 4 transacciones o menos.

```
SELECT c.id, c.company_name, COUNT(t.id) AS cantidad_transacciones,
       CASE
         WHEN COUNT(t.id) > 4 THEN 'Más de 4'
         ELSE '4 o menos'
       END AS grupo
FROM company c JOIN transaction t ON c.id = t.company_id
GROUP BY c.id
ORDER BY cantidad_transacciones DESC;
```

The screenshot shows a SQL IDE window titled 'Tarea S2.01. Nociones básicas...'. The query editor contains the following SQL code:

```
106 ORDER BY t.amount DESC;
107
108 -- Nivel 3. Ejercicio 2
109 /*Necesitamos optimizar la asignación de los recursos y dependerá de la capacidad operativa que se requiera,
110 por lo que te piden la información sobre la cantidad de transacciones que realizan las empresas, pero el departamento de
111 recursos humanos es exigente y quiere un listado de las empresas en las que especifiques si tienen más de 4 transacciones o menos.*/
112 SELECT c.id, c.company_name, COUNT(t.id) AS cantidad_transacciones,
113        CASE
114          WHEN COUNT(t.id) > 4 THEN 'Más de 4'
115          ELSE '4 o menos'
116        END AS grupo
117 FROM company c
118 JOIN transaction t ON c.id = t.company_id
119 GROUP BY c.id
120 ORDER BY cantidad_transacciones DESC;
121
```

The 'Result Grid' shows the following data:

id	company_name	cantidad_transacciones	grupo
b-2302	Nunc Interdum Incorporated	105	Más de 4
b-2346	Ut Semper Foundation	59	Más de 4
b-2326	Enim Condimentum Ltd	57	Más de 4
b-2278	Arcu LLP	56	Más de 4
b-2362	Lorem Eu Incorporated	54	Más de 4
b-2494	Malesuada PC	52	Más de 4
b-2618	Non Institute	30	Más de 4
b-2230	Fusce Corp.	2	4 o menos
b-2234	Convallis In Incorporated	2	4 o menos
b-2238	Ante Iaculis Nec Foundation	2	4 o menos
b-2242	Donec Ltd	2	4 o menos
b-2246	Sed Nunc Ltd	2	4 o menos
b-2250	Amet Nulla Donec Corporation	2	4 o menos
b-2254	Nascetur Ridiculus Mus Inc.	2	4 o menos
b-2258	Vestibulum Lorem PC	2	4 o menos
b-2262	Gravida Sagittis LLP	2	4 o menos
b-2266	Mus Aenean Eget Foundation	2	4 o menos
b-2270	Dis Parturient Institute	2	4 o menos
b-2274	Sed LLC	2	4 o menos
b-2282	Pretium Neque Corp.	2	4 o menos
b-2286	Fringilla LLC	2	4 o menos
b-2290	Quisque Libero LLC	2	4 o menos

The 'Output' section shows the following messages:

```
1 21:32:14 SELECT c.company_name, c.phone, c.country, date(timestamp) AS fecha, t.amount FROM company c JOIN transaction t ON c.id = t.company_id WHERE t.amount BETWEEN 100 AND 200 AND date(timestamp) BETWEEN 2019-01-01 AND 2019-01-01 5 row(s) returned
2 21:34:29 SELECT c.id, c.company_name, COUNT(t.id) AS cantidad_transacciones, CASE WHEN COUNT(t.id) > 4 THEN 'Más de 4' ELSE '4 o menos' END AS grupo FROM company c JOIN transaction t ON c.id = t.company_id 100 row(s) returned
```