

Nivel 1

- Ejercicio 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.

La base de datos “Transactions” contiene dos tablas:

- “transaction” (hechos)

Clave primaria: “id”

Clave foránea: “company_id”

La “transaction” es la tabla principal (hechos) que contiene la información sobre las transacciones realizadas: id de la tarjeta de crédito, id de la empresa, id del usuario, latitud, longitud, la fecha, el monto y si fue declinada o no.

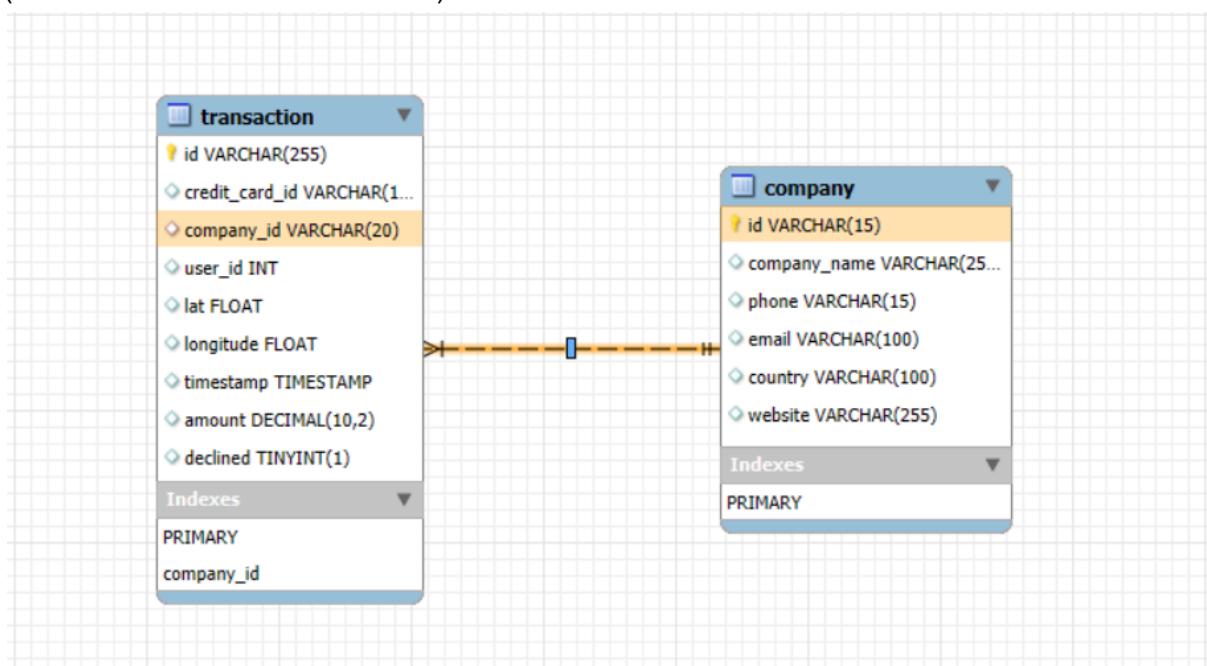
- “company” (dimensión)

Clave primaria: “id”

La “company” tiene la información de las empresas: id, nombre, teléfono, email, país y la página web.

La tabla “company” se relaciona con la tabla de hechos “transaction” a través de la clave foránea “company_id”. Es una relación de 1 uno a muchos ($1 \rightarrow \infty$): una empresa puede tener muchas transacciones, mientras que una transacción pertenece a una sola empresa.

La estructura se asemeja a un modelo estrella simple, donde “transaction” funciona como la tabla de hechos y “company” como una tabla de dimensión, aunque es una versión mínima (1 tabla de hechos + 1 dimensión).



- Ejercicio 2

Nota general para el resto de los ejercicios: Transacciones vs Ventas.

Las transacciones son todas las operaciones registradas en el sistema, sin importar el resultado - rechazada/aprobada (declined=1/ declined=0). Cuando un ejercicio pide “transacciones” uso todas las filas de la tabla de transacciones, porque se refiere a la actividad general, no al dinero generado.

Las ventas representan sólo las transacciones aprobadas, es decir, las que generaron dinero real, aunque no lo diga explícitamente. La razón es que una transacción rechazada no genera ingresos, así que incluirla daría resultados incorrectos. Cuando se habla de ventas filtro por “WHERE declined=0”.

Utilizando JOIN realizarás las siguientes consultas:

- Listado de los países que están generando ventas.

The screenshot shows a MySQL Workbench window titled "Sprint 2_SQL_noções_básicas". The SQL editor contains the following code:

```
1 # SPRINT 2.SQL
2 • USE transactions;
3 # NIVEL 1
4
5 # - Ejercicio 2
6 # Utilizando JOIN realizarás las siguientes consultas:
7
8 # Listado de los países que están generando ventas.
9 • SELECT DISTINCT c.country
10 FROM company c
11 JOIN transaction t ON c.id=t.company_id
12 WHERE t.declined=0;
13
```

The Result Grid shows the following data:

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

The Output pane shows the executed query and its results:

```
1 11:16:28 SELECT DISTINCT c.country FROM company c JOIN transaction t ON c.id=t.company_id WHERE t.declined=0 15 row(s) returned
```

Message: Duration / Fetch 0.000 sec / 0.000 sec

Nota: Necesitamos la información de ambas tablas “company” para extraer los países y “transaction” para saber cuales realizan transacciones no rechazadas. Utilizamos DISTINCT para obtener la lista de los países únicos y evitar las repeticiones.

- Desde cuántos países se generan las ventas.

Sprint 2_SQL_noções_básicas* | Don't Limit

```

1 # SPRINT 2. SQL
2 • USE transactions;
3 # NIVEL 1
4
5 # - Ejercicio 2
6 # Utilizando JOIN realizarás las siguientes consultas:
7
8 # Desde cuántos países se generan las ventas.
9 • SELECT COUNT(DISTINCT country) AS Países_que_compran
10 FROM company c
11 JOIN transaction t ON c.id=t.company_id
12 WHERE t.declined=0;

```

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

Países_que_compran
15

Result 14 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	11:37:42	SELECT COUNT(DISTINCT country) AS Países_que_compran FROM company c JOIN transaction t ON c.id=t.company_id WHERE t.declined=0;	1 row(s) returned	0.000 sec / 0.000 sec

Nota: Utilizamos COUNT DISTINCT para calcular la cantidad de los países únicos que generan ventas y evitar las repeticiones, tomamos en cuenta solo los que tienen al menos una transacción aprobada.

- Identifica a la compañía con la mayor media de ventas.

Sprint 2_SQL_noções_básicas* | Don't Limit

```

5 # - Ejercicio 2
6 # Utilizando JOIN realizarás las siguientes consultas:
7
8 # Identifica a la compañía con la mayor media de ventas.
9 • SELECT c.id, c.company_name, ROUND(AVG(t.amount),2) AS media_de_ventas
10 FROM company c
11 JOIN transaction t ON c.id=t.company_id
12 WHERE t.declined=0
13 GROUP BY c.id, c.company_name
14 ORDER BY media_de_ventas DESC
15 LIMIT 1;
16

```

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

	id	company_name	media_de_ventas
1	b-2398	Eget Ipsum Ltd	481.86

Result 15 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	11:40:06	SELECT c.id, c.company_name, ROUND(AVG(t.amount),2) AS media_de_ventas FROM company c JOIN transaction t ON c.id=t.company_id WHERE t.declined=0 GROUP BY c.id, c.company_name ORDER BY media_de_ventas DESC LIMIT 1;	1 row(s) returned	0.000 sec / 0.000 sec

Nota: Necesitamos la información de ambas tablas de “company” el nombre de la compañía y de “transaction” el monto medio (AVG) de las ventas solo de las transacciones no rechazadas. Buscamos la media de ventas de cada una de las empresas, las ordenamos de mayor a menor (DESC) y LIMITamos el resultado a un registro - la compañía con la mayor media.

Siguiendo buenas prácticas utilizo JOIN frente a la subconsulta y aplico ROUND para limitar la cantidad de decimales a 2.

- Ejercicio 3

Nota general sobre IN y EXISTS: por buenas prácticas he utilizado EXISTS en vez de IN ya que: IN busca un valor en una lista, pero puede dar problemas si hay NULL o la lista es muy grande, mientras que EXISTS solo comprueba si existe al menos un registro que cumpla la condición; es más seguro y eficiente.

Utilizando sólo subconsultas (sin utilizar JOIN):

- Muestra todas las transacciones realizadas por empresas de Alemania.

The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following code:

```
3  # Nivel 1
4  # - Ejercicio 3
5  # Utilizando sólo subconsultas (sin utilizar JOIN):
6
7  # Muestra todas las transacciones realizadas por empresas de Alemania.
8 • SELECT t.id
9  FROM transaction t
10 WHERE EXISTS
11   (SELECT c.id
12    FROM company c
13    WHERE c.id = t.company_id
14    AND c.country = 'Germany');
```

The results grid shows a list of transaction IDs (id) from the transaction table, filtered by the EXISTS clause. The output pane at the bottom shows the execution log with two entries: the first is a USE statement for the transactions database, and the second is the execution of the SELECT query, which returned 118 rows.

Nota: Necesitamos únicamente las transacciones de la tabla “transaction” que cumplen el criterio de pertenecer a una empresa de Alemania (esta información se encuentra en la tabla “company”). Utilizamos la subconsulta con WHERE para filtrar las transacciones de las compañías alemanas. Se puede utilizar WHERE ... IN, pero para seguir las buenas prácticas opto por el uso de EXISTS.

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

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query is:

```

1  # SPRINT 2. SQL
2 • USE transactions;
3  # Nivel 1
4  # - Ejercicio 3
5  # Utilizando sólo subconsultas (sin utilizar JOIN):
6
7  # Lista las empresas que han realizado transacciones por un amount superior a la media de todas las transacciones.
8 • SELECT c.id, c.company_name
9  FROM company c
10 WHERE EXISTS
11   (SELECT t.company_id
12    FROM transaction t
13    WHERE t.company_id = c.id
14      AND t.amount > (SELECT AVG(amount) FROM transaction));

```

The results grid shows the following data:

id	company_name
b-2222	Ac Fermentum Incorporated
b-2226	Magna A Neque Industries
b-2230	Fusce Corp.
b-2238	Ante Iaculis Nec Foundation
b-2242	Donec Ltd
b-2246	Sed Nunc Ltd
b-2254	Nascetur Ridiculus Mus Inc.
b-2258	Vestibulum Lorem PC
b-2262	Gravida Sagittis LLP

The output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
1	11:56:35	USE transactions	0 row(s) affected	0.000 sec
2	11:56:35	SELECT c.id, c.company_name FROM company c WHERE EXISTS (SELECT t.company_id ...	70 row(s) returned	0.016 sec / 0.000 sec

Nota: Utilizamos la subconsulta para encontrar la media de todas las transacciones y después con la consulta principal obtenemos la lista de los países cuyas transacciones superan la media.

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

The screenshot shows the MySQL Workbench interface. The top window is titled "Sprint 2 SQL_nociones_basicas" and contains the following SQL code:

```

1  # SPRINT 2. SQL
2 • USE transactions;
3  # Nivel 1
4  # - Ejercicio 3
5  # Utilizando sólo subconsultas (sin utilizar JOIN):
6
7  # Eliminarán del sistema las empresas que carecen de transacciones registradas, entrega el listado de estas empresas.
8 • SELECT c.id, c.company_name
9  FROM company c
10 WHERE NOT EXISTS
11   (
12     SELECT t.company_id
13     FROM transaction t
14     WHERE t.company_id=c.id);

```

Below the editor is a "Result Grid" showing the output of the query:

	id	company_name
*	NULL	NULL

At the bottom, the "Output" pane displays the execution log:

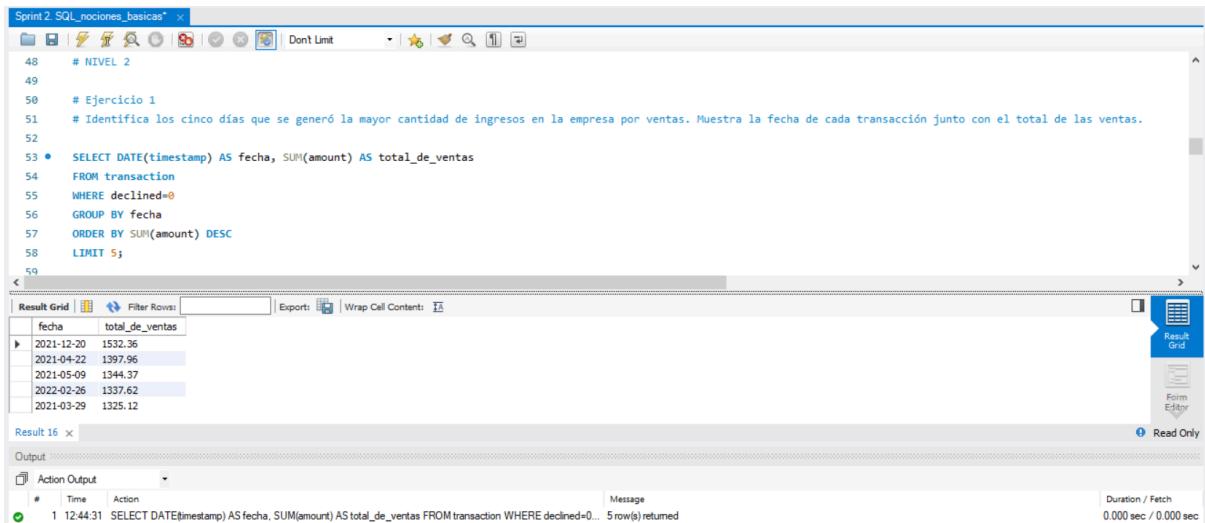
Action	Time	Action	Message	Duration / Fetch
1	12:02:56	USE transactions	0 row(s) affected	0.000 sec
2	12:02:56	SELECT c.id, c.company_name FROM company c WHERE NOT EXISTS (SELECT t.company_id FROM trans...	0 row(s) returned	0.000 sec / 0.000 sec

Nota: Aplicamos la subconsulta correlacionada y seleccionamos las transacciones de la tabla “transaction”, después con la consulta principal filtramos las compañías que no tiene las transacciones. El resultado “NULL” nos demuestra que todas las compañías han realizado al menos una transacción.

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.



The screenshot shows a SQL editor window titled "Sprint 2. SQL_nociones basicas". The code in the editor is:

```
48 # NIVEL 2
49
50 # Ejercicio 1
51 # 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.
52
53 • SELECT DATE(timestamp) AS fecha, SUM(amount) AS total_de_ventas
54   FROM transaction
55   WHERE declined=0
56   GROUP BY fecha
57   ORDER BY SUM(amount) DESC
58   LIMIT 5;
59
```

The result grid displays the following data:

fecha	total_de_ventas
2021-12-20	1532.36
2021-04-22	1397.96
2021-05-09	1344.57
2022-02-26	1337.62
2021-03-29	1325.12

The output pane shows the executed query and its duration:

#	Time	Action	Message	Duration / Fetch
1	12:44:31	SELECT DATE(timestamp) AS fecha, SUM(amount) AS total_de_ventas FROM transaction WHERE declined=0...	5 row(s) returned	0.000 sec / 0.000 sec

Nota: Para obtener la fecha a partir de timestamp utilice DATE, agrupo todas las transacciones aprobadas por la fecha y calculo el total de transacciones realizadas en el mismo día. Como necesitamos el TOP-5, ordenamos de mayor a menor (DESC) y LIMITamos el output a 5 registros.

- Ejercicio 2

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

The screenshot shows a SQL editor window titled "Sprint 2. SQL_nociones_basicas*". The code in the editor is:

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

The result grid displays the following data:

country	media_de_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

The output pane shows the query executed at 12:51:11, returning 15 rows in 0.000 sec / 0.000 sec.

- 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.

- Muestra el listado aplicando JOIN y subconsultas.

The screenshot shows a MySQL Workbench window titled "Sprint 2_SQL_noções_básicas". The code editor contains the following SQL query:

```
1 # SPRINT 2. SQL
2 # Nivel 2
3 # Ejercicio 3
4 # En tu empresa, se plantea un nuevo proyecto para lanzar algunas campañas publicitarias para hacer competencia a la compañía "Non Institute".
5 # Para ello, te piden la lista de todas las transacciones realizadas por empresas que están ubicadas en el mismo país que esta compañía.
6
7 # Muestra el listado aplicando JOIN y subconsultas.
8 * SELECT t.id, c.id, c.company_name, c.country
9   FROM transaction t
10  JOIN company c ON t.company_id=c.id
11 WHERE c.country=
12   (SELECT country
13    FROM company
14   WHERE company_name = 'Non Institute');
```

The results grid displays the following data:

id	id	company_name	country
b2928E1C-EC14-A760-0475-87147764906A	b-2246	Sed Nunc Ltd	United Kingdom
ACD2011A-A2B1-C365-41E1-2AB00C65147A	b-2246	Sed Nunc Ltd	United Kingdom
4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	b-2310	Non Magna LLC	United Kingdom
BC2B9A38-7784-28CD-1FE8-14ED863E773	b-2310	Non Magna LLC	United Kingdom
1479B3D2-B7BA-C7B8-4CE3-8D7C2DE85AB8	b-2326	Enim Condimentum Ltd	United Kingdom
152598C2-0290-D684-4B66-91EDF393EBFF	b-2326	Enim Condimentum Ltd	United Kingdom
1B636B5B-A2E8-7C69-D9C9-C54535DAFD3B	b-2326	Enim Condimentum Ltd	United Kingdom
204180E5-BB04-BE98-8D7A-A95C1BFDBF5C	b-2326	Enim Condimentum Ltd	United Kingdom

The output pane shows the following message:

```
1 12:22:58 SELECT t.id, c.id, c.company_name, c.country FROM transaction t JOIN company c ON t.company_id=c.id WHERE c.country = 'Non Institute' 100 row(s) returned
```

Nota: Necesitamos JOIN para obtener la información de ambas tablas id, nombre y país (solo para ver cuál es) de “company” y id de “transactions”. Utilizamos la subconsulta para obtener la información sobre de qué país es la empresa ‘Non-Institute’ y la colocamos en la condición WHERE porque nos interesan solo las compañías del mismo país.

- Muestra el listado aplicando solo subconsultas.

The screenshot shows the MySQL Workbench interface. The query editor window contains the following SQL code:

```

2  # Nivel 2
3  # Ejercicio 3
4  # En tu empresa, se plantea un nuevo proyecto para lanzar algunas campañas publicitarias para hacer competencia a la compañía "Non Institute".
5  # Para ello, te piden la lista de todas las transacciones realizadas por empresas que están ubicadas en el mismo país que esta compañía.
6
7 • SELECT t.id
8   FROM transaction t
9   WHERE EXISTS
10    (
11      SELECT c.id
12        FROM company c
13       WHERE t.company_id=c.id
14       AND c.country=
15          (
16              SELECT country
17                FROM company
18               WHERE company_name ='Non Institute'));
```

The results grid displays a list of transaction IDs (id) from the transaction table, filtered by the query above. The results are as follows:

id
28928E1C-EC14-A760-0475-871477649D6A
ACD2011A-A2B1-C365-41E1-2AB00C65147A
4334349E-CEB9-3D68-A4D4-FEB7718A1ACE
BC289A38-7B44-28CD-1FEB-14E0E0863E773
147983C2-87BA-C788-4CE3-8D7C2DE85BAB
152598C8-029D-D684-4B66-91EDF393EBFF
1B63655B-A2E8-7C69-D5C9-C5453DAD030
204180E5-BB04-BE9B-BD7A-A95C1BFDBF5C
322000C7-6605-127A-C7E6-2192A102A1202

The status bar at the bottom shows the following information:

- Action Output
- # 1 Time 12:28:34 Action SELECT t.id FROM transaction t WHERE EXISTS (SELECT c.id FROM company c WHERE t.company_id=c.i...
- Message 100 row(s) returned
- Duration / Fetch 0.000 sec / 0.000 sec

Nota: Como no está permitido utilizar JOIN, hacen falta dos subconsultas (anidadas). Primero ver de qué país es la compañía 'Non-Institute', luego elegir solo las compañías que son de aquel país para poder con la consulta principal visualizar las transacciones de las empresas que son del mismo país que 'Non-Institute'.

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.

The screenshot shows a database interface with the following details:

SQL Editor:

```
estructura_dades (1)      dades_introduir (1)      Sprint 2: SQL_nociones_basicas* 
1  # SPRINT 2. SQL
2  # Nivel 3
3  # Ejercicio 1
4  # Presenta el nombre, teléfono, país, fecha y amount,
5  # de aquellas empresas que realizaron transacciones con un valor comprendido entre 100 y 200 euros
6  # y en alguna de estas fechas: 29 de abril de 2021, 20 de julio de 2021 y 13 de marzo de 2022.
7  # Ordena los resultados de mayor a menor cantidad.
8
9 •  SELECT c.company_name, c.phone, c.country, DATE(t.timestamp) AS fecha, t.amount
10 FROM company c
11 JOIN transaction t ON c.id=t.company_id
12 WHERE (t.amount BETWEEN 100 AND 200) AND DATE(t.timestamp) IN('2021-04-29', '2021-07-20', '2022-03-13')
13 ORDER BY t.amount DESC;
```

Result Grid:

company_name	phone	country	fecha	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 Condimentum 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

Output:

#	Time	Action	Message	Duration / Fetch
1	12:40:15	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;	5 row(s) returned	0.000 sec / 0.000 sec

Nota: Tenemos dos condiciones que tienen que cumplirse al mismo tiempo: el valor y la fecha, las colocamos en WHERE y las unimos con AND.

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

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

```
estructura_dades (1) dades_introduir (1) Sprint 2_SQL_nociones_basicas* x
Dont Limit
1 # SPRINT 2. SQL
2 # Nivel 3
3 # Ejercicio 2
4 # Necesitamos optimizar la asignación de los recursos y dependerá de la capacidad operativa que se requiera, por lo que te piden la información sobre
5 # la cantidad de transacciones que realizan las empresas, pero el departamento de recursos humanos es exigente y
6 # quiere un listado de las empresas en las que especifiques si tienen más de 4 transacciones o menos.
7 • SELECT t.company_id, c.company_name, COUNT(t.id) AS cantidad_de_trasacciones,
8     CASE
9         WHEN COUNT(t.id) > 4 THEN 'Sí'
10        ELSE 'No'
11    END AS mas_de_4_transacciones
12 FROM transaction t
13 JOIN company c ON t.company_id=c.id
14 GROUP BY t.company_id;
```

The results grid displays the following data:

company_id	company_name	cantidad_de_trasacciones	mas_de_4_transacciones
b-2222	Ac Fermentum Incorporated	2	No
b-2226	Magna A Neque Industries	2	No
b-2230	Fusce Corp.	2	No
b-2234	Convallis In Incorporated	2	No
b-2238	Ante Iaculis Nec Foundation	2	No
b-2242	Donec Ltd	2	No
b-2246	Sed Nunc Ltd	2	No
b-2250	Amet Nulla Donec Corporation	2	No
b-2254	Nascetur Ridiculus Mus Inc.	2	No

The output pane shows the following message:

Action Output
Time Action Message Duration / Fetch
1 12:43:46 SELECT t.company_id, c.company_name, COUNT(t.id) AS cantidad_de_trasacciones, CASE WHEN COUNT(t... 100 row(s) returned 0.016 sec / 0.000 sec

Nota: Necesitamos agrupar las transacciones por la empresa (GROUP BY) y calcular su cantidad respectiva (COUNT). Se nos solicita indicar explícitamente si hay más de 4 transacciones para lograrlo creamos nueva columna utilizando CASE y detallando las condiciones.