

Tasca S2.01. Nocions bàsiques SQL

Nivel 1

Ejercicio 1

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

**No ejecuté directamente los documentos adjuntos en MySQL, sino que lo hice poco a poco (para practicar).*

Creación de la base de datos ‘transactions’ vacía a partir del documento estructura_dades.

```
Query 1 | dades_introduir_sprint2 | estructura dades
CREATE DATABASE IF NOT EXISTS transactions;
```

#	Time	Action	Message	Duration / Fetch
1	09:43:11	CREATE DATABASE IF NOT EXISTS transactions	1 row(s) affected	0.125 sec

Actualización desde SCHEMAS para corroborar que aparezca la base de datos recién creada.

Navigator: SCHEMAS

- sakila
- sys
- transactions
- world

```
Query 1 | dades_introduir_sprint2 | estructura dades
CREATE DATABASE IF NOT EXISTS transactions;
```

Se trabajará con una sola base de datos. Por eso, se define como *default* a través del USE al inicio del script.

```

2
3 • USE transactions;

```

The screenshot shows the MySQL Workbench interface. On the left, there's a tree view of databases: sakila, sys, transactions (which is expanded to show Tables, Views, Stored Procedures, Functions), and world. On the right, a query editor window displays the following code:

```

Action Output
# Time Action Message
1 09:43:11 CREATE DATABASE IF NOT EXISTS transactions 1 row(s) affected
2 10:02:26 USE transactions 0 row(s) affected

```

Creación de la tabla 'company' ejecutando el código del documento estructura_dades.

```

• CREATE TABLE IF NOT EXISTS company (
    id VARCHAR(15) PRIMARY KEY,
    company_name VARCHAR(255),
    phone VARCHAR(15),
    email VARCHAR(100),
    country VARCHAR(100),
    website VARCHAR(255)
);

```

3 10:20:57 CREATE TABLE IF NOT EXISTS company (id VARCHAR(15) PRIMARY K... 0 row(s) affected 0.312 sec

Crear la tabla 'transaction' ejecutando el código del documento estructura_dades.

```

14 • CREATE TABLE IF NOT EXISTS transaction (
    id VARCHAR(255) PRIMARY KEY,
    credit_card_id VARCHAR(15) REFERENCES credit_card(id),
    company_id VARCHAR(20),
    user_id INT REFERENCES user(id),
    lat FLOAT,
    longitude FLOAT,
    timestamp TIMESTAMP,
    amount DECIMAL(10, 2),
    declined BOOLEAN,
    FOREIGN KEY (company_id) REFERENCES company(id)
);

```

4 10:23:48 CREATE TABLE IF NOT EXISTS transaction (id VARCHAR(255) PRIMARY KEY, credit_card_id VARCHAR(15) ... 0 row(s) affected 0.109 sec

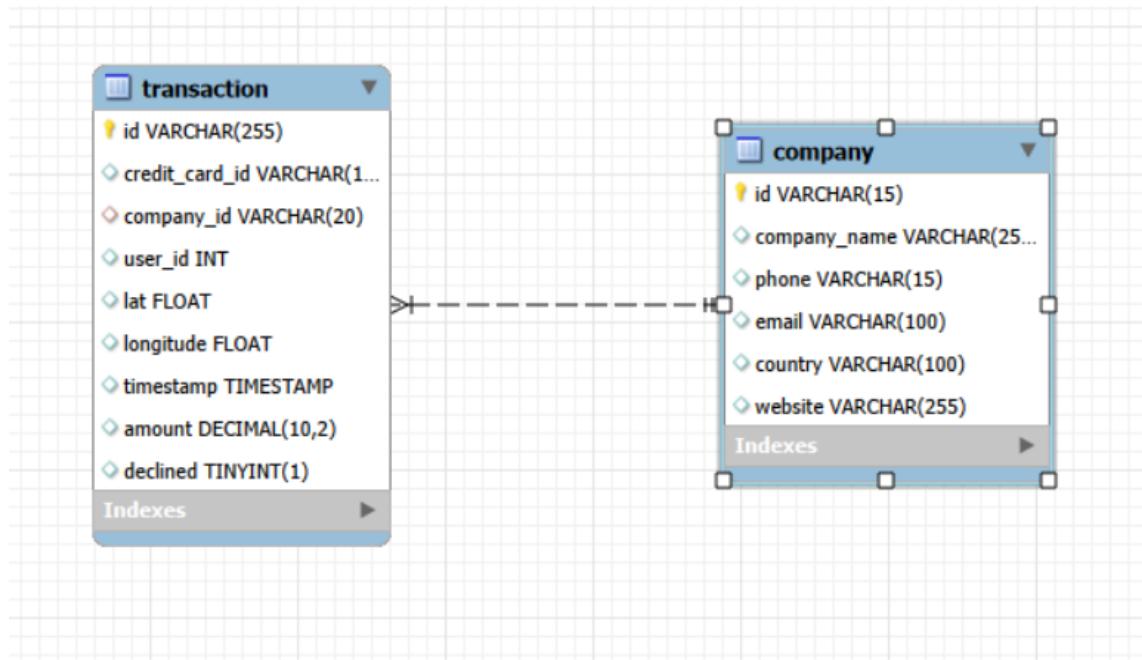
Insertar datos de ‘company’ y de ‘transaction’ a partir del documento dades_introduir.

```
90100 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('3D39E00C-1F9D-42AE-B0F1')
90101 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('E183116A-B83A-412C-87C1')
90102 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('65FF0A2A-3141-4378-AEF1')
90103 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('D7AE2C20-8119-4DCE-BB21')
90104 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('C312BA55-0FCC-4623-A821')
90105 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('3EFB1984-E1FC-4F43-8A90')
```

100113 11:52:24 INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES (D7AE2C20-8119-4DCE-BB21, 'CcS-8141', b-2322, '3...', 1 row(s) affected)	0.016 sec
100114 11:52:24 INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES (C312BA55-0FCC-4623-A822-22E26504255B, 'CcS-8255', b-2406, '36...', 1 row(s) affected)	0.000 sec
100115 11:52:24 INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES (3EFB1984-E1FC-4F43-8A9C-4E89ACD86605, 'CcS-5277', b-2262, '6...', 1 row(s) affected)	0.000 sec

Esquema de la base de datos ‘transactions’, a partir de la recomendación de buenas prácticas:

- En Workbench: *Database* → *Reverse Engineer o Models* → *Create EER Model from Database*: para mostrar el diagrama o modelo de la base de datos.



Se trata de un **modelo relacional**, que en un futuro podría configurar un **modelo en estrella** (si se le añadieran más datos, por ejemplo, de usuarios o de las tarjetas de crédito. La tabla ‘transaction’ es la de hechos, mientras que la tabla ‘company’ es la dimensión.

La **tabla de hechos** (‘transaction’) muestra operaciones realizadas por diferentes empresas, en donde cada una de las líneas o registros representa una transacción diferente.

Visualización de la tabla ‘transaction’:

27 • `SELECT * FROM transaction;`

Result Grid									
	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶	00043A49-2949-494B-A5DD-ASBAE3BB19DD	CcS-9294	b-2458	4713	46.1999	1.43554	2024-08-28 07:16:46	395.43	0
	000447FE-B650-4DCF-85DE-C7ED0EE1CAAD	CcS-5019	b-2370	438	41.5972	12.2218	2016-12-21 20:07:18	155.63	0
	00045D6B-ED2E-4F2F-8186-CEE074D875D0	CcS-6699	b-2390	2118	29.7573	-95.3796	2020-07-14 15:37:45	326.01	0
	000481C3-1C26-4FEB-83A0-4CD0EB004BBD	CcS-6696	b-2230	2115	53.5489	-113.503	2017-09-04 19:44:53	161.60	0

98 19:18:15 SELECT * FROM transaction 100000 row(s) returned 0.078 sec / 0.297 sec

Descripción de la tabla ‘transaction’:

31 • `DESCRIBE transaction;`

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

9 row(s) returned 0.156 sec / 0.000 sec

Respecto a las **variables**, se observa que el id está definido como Primary Key y es de tipo VARCHAR, es decir, una cadena de caracteres de longitud variable. En este caso, se permite introducir hasta 255 caracteres.

También son de este tipo, las variables credit_card_id y company_id, aunque de longitud máxima 15 y 20 respectivamente, para introducir los identificadores de las tarjetas de crédito y de la empresa.

Siguiendo el esquema, la variable company_id es la Foreign Key de la tabla de dimensión, ‘company’.

La variable user_id es de tipo INT (integer) y representa un número entero. En esta columna se almacenarán los identificadores de cada usuario.

La tabla incluye la ubicación geográfica (coordenadas) desde donde se realizó cada una de las transacciones registradas, a través de la latitud (variable ‘lat’) y longitud (variable ‘longitude’). Ambas son de tipo FLOAT, que permite introducir un número decimal aproximado.

La variable timestamp es de tipo TIMESTAMP permite almacenar la fecha y hora de cada una de las transacciones.

La variable amount es de tipo DECIMAL y, a diferencia del tipo FLOAT, permite introducir un número decimal exacto, en este caso, para registrar la cantidad de dinero de cada una de las transacciones. Entre paréntesis se especifica la cantidad de dígitos y la cantidad de decimales (10,2), respectivamente.

Finalmente, la variable declined es de tipo TINYINT(1) que se utiliza para la información de tipo boolean, por eso se visualiza que en los registros aparece un 0, si es FALSE (transacción no rechazada) y un 1, si es TRUE (transacción rechazada).

Por otro lado, cada una de las líneas de la **tabla de dimensión** ('company') representa a una empresa diferente, con un identificador (id) para cada una de ellas. También registra información como el nombre (company_name), información de contacto como el teléfono (phone) y el correo electrónico (email), el país de la empresa (country) y finalmente su sitio web (website).

Visualización de la tabla 'company':

Result Grid						
	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

🕒 100 19:19:00 SELECT * FROM company 100 row(s) returned 0.000 sec / 0.000 sec

Descripción de la tabla 'company':

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

🕒 12 11:21:17 DESCRIBE company 6 row(s) returned 0.016 sec / 0.000 sec

La Primary Key de la tabla 'company' es la variable id y es de tipo VARCHAR(15), por ello se ve que no puede ser un campo nulo. Al igual que en la tabla 'transaction', se refiere a una cadena de caracteres de longitud variable. Todas las variables de esta tabla son del mismo tipo, en donde se define la longitud específica de cada una de ellas.

La PK de 'company' (id) se relaciona con la FK de 'transaction' (company_id) a través de una relación 1-N. Una empresa concreta aparece en la tabla de hechos más de una vez, es decir, ha hecho más de una transacción. Este tipo de relación 1-N entre la tabla de dimensión y la de hechos justifica el modelo relacional (posiblemente de estrella), porque el grueso de las filas está en la tabla 'transaction', que es la de hechos, y en la tabla de dimensión 'company' se desglosa la información específica de la empresa que realiza cada una de las transacciones. De esta manera, es más fácil actualizar las tablas y no se repite información en la tabla de hechos.

Nivel 1

Ejercicio 2:

Utilizando JOIN, realizarás las siguientes consultas:

- Lista de países que están generando ventas

```

41 •   SELECT comp.country
42     FROM transaction AS trans
43     JOIN company AS comp
44       ON trans.company_id = comp.id
45     WHERE trans.declined = 0      -- representa que la venta no fue rechazada
46     GROUP BY comp.country;
47

```

* Considero que la venta se genera cuando no es rechazada (pero puede hacerse sin el WHERE).

Result Grid		Filter Row
	country	
▶	Germany	
	Australia	
	United States	
	New Zealand	
	Norway	
	United Kingdom	
	Italy	
	Belgium	
	Sweden	
	Ireland	
	China	
	Canada	
Result 11		x

```
0 100121 12:21:59 SELECT comp.country FROM transaction AS trans JOIN company AS comp ON trans.company_id = comp.id WHERE trans.declined = 0-- representa que la venta no fue rechazada GROUP ... 15 row(s) returned 0.609 sec / 0.000 sec
```

- Desde cuántos países se generan las ventas.

```

50 •   SELECT COUNT(DISTINCT comp.country) AS paises_con_ventas
51     FROM transaction AS trans
52     JOIN company AS comp
53       ON trans.company_id = comp.id
54     WHERE trans.declined = 0;

```

Result Grid	
	países_con_ventas
▶	15

100130 12:38:46 | SELECT COUNT(DISTINCT comp.country) AS paises_con_ventas FROM transaction AS trans JOIN company AS comp ON trans.company_id = comp.id WHERE trans.declined = 0 | 1 row(s) returned | 0.375 sec / 0.000 sec

- Identifica la empresa con la media más alta de ventas

```

•   SELECT comp.company_name, ROUND(AVG(trans.amount),2) AS media_ventas
      FROM transaction AS trans
      JOIN company AS comp
        ON trans.company_id = comp.id
      WHERE trans.declined = 0
      GROUP BY comp.company_name
      ORDER BY media_ventas DESC
      LIMIT 1;

```

Result Grid		
	company_name	media_ventas
▶	Ac Fermentum Incorporated	284.91

100138 13:17:41 | SELECT comp.company_name, ROUND(AVG(trans.amount),2) AS media_ventas FROM transaction AS trans JOIN company AS comp ON trans.company_id = comp.id WHERE trans.decline... | 1 row(s) returned | 0.578 sec / 0.000 sec

Nivel 1**Ejercicio 3:**

Utilizando solo subconsultas (sin utilizar JOIN):

- Muestra todas las transacciones realizadas para empresas de Alemania

```

76 •   SELECT trans.company_id AS id_transaccion
77     FROM transaction AS trans
78     WHERE EXISTS  (  SELECT *
79                   FROM company AS comp
80                   WHERE comp.country = 'Germany'
81                   AND comp.id = trans.company_id
82               )
83 ;

```

**Uso de EXISTS para corroborar si hay coincidencias, es decir, si existe alguna empresa de Alemania cuyo id coincida con el company_id de la transacción. Si es así, la incluye en la lista. El uso de EXISTS es más eficiente para este caso que el uso de IN.*

Result Grid	
	id_transaccion
▶	00138D3B-206D-4C03-94B7-63A2676EB9B4
	0013C1B6-3B84-4D6C-8154-E2B3FEBCA8E9
	00201A11-2E62-44C4-941D-198FC8DB77F0
	00235618-0A5C-4D49-9DCB-B3A9405D8923
	005A5A7B-1F1A-4B6C-9B15-1625A78C9C38
	00687139-48B2-4FFA-8E73-B20376F04AB4
	0074F4DD-32F1-4827-8758-55896314623A
	00AAB9CD-39D6-4DCB-8A1D-13BE73DC90A9
	00BE09D4-6920-47D8-ABE8-325E2269829D
	00DA0383-E048-4577-8ED1-3C56C258FF2F
	00DD11DE-ED01-4BBD-93A0-174D183A59DF
	01449CE0-98E9-4DE5-9810-728C6BA00E6F
	0175E8C7-241E-42DA-A8B9-9F246DBF4D2F
	01ABDAB8-06E2-4CA0-A131-AEE6FF11B749

2 12:17:15 SELECT trans.id AS id_transaccion FROM transaction AS trans WHERE EXISTS (SELECT * FROM company AS comp WHERE comp.country = 'Germany' AND comp.id = trans.company_id) 13291 row(s) returned 0.000 sec / 0.016 sec

- Listar las empresas que han realizado transacciones por una cantidad superior a la media de todas las transacciones.

```

94 •   SELECT comp.company_name
95     FROM company AS comp
96     WHERE EXISTS  (  SELECT *
97                   FROM transaction AS trans
98                   WHERE amount > (  SELECT ROUND(AVG(transaction.amount), 2) AS Media
99                           FROM transaction
100                          )
101                         AND trans.company_id = comp.id);

```

Result Grid Filter Rows:	
	company_name
▶	Ac Fermentum Incorporated
	Magna A Neque Industries
	Fusce Corp.
	Convallis In Incorporated
	Ante Iaculis Nec Foundation
	Donec Ltd
	Sed Nunc Ltd
	Amet Nulla Donec Corporation
	Nascetur Ridiculus Mus Inc.
	Vestibulum Lorem PC
	Gravida Sagittis LLP
	Mus Aenean Eget Foundation
	Dis Parturient Institute

7 12:25:03 SELECT comp.company_name FROM company AS comp WHERE EXISTS(SELECT * FROM transaction AS t... 100 row(s) returned 0.078 sec / 0.000 sec

- Eliminarán del sistema las empresas que no tienen transacciones registradas, entrega el listado de estas empresas.

```

112 •   SELECT comp.company_name AS empresas_sin_transacciones
113     FROM company AS comp
114     WHERE NOT EXISTS (  SELECT trans.company_id
115                           FROM transaction AS trans
116                           WHERE trans.company_id = comp.id);
117

```

Result Grid		Filter Rows:
	empresas_sin_transacciones	
<pre>11 09:26:27 SELECT comp.company_name AS empresas_sin_transacciones FROM company AS comp WHERE NOT EXI... 0 row(s) returned</pre>		
0.016 sec / 0.000 sec		

Nivel 2

Ejercicio 1

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

```
132 •   SELECT DATE(trans.timestamp) AS fecha, SUM(amount) AS total_ventas
133     FROM transaction AS trans
134     WHERE declined = 0           -- para filtrar las ventas reales (no rechazadas)
135     GROUP BY fecha
136     ORDER BY total_ventas DESC
137     LIMIT 5;
```

* Al igual que antes, considero que el ingreso se genera cuando la venta no es rechazada (pero puede hacerse sin el WHERE).

Result Grid		Filter Rows:
	fecha	total_ventas
▶	2022-12-13	14337.44
	2019-11-18	13591.32
	2023-02-20	13332.59
	2017-12-20	13318.43
	2019-03-18	12680.95

```
29 10:19:02 SELECT DATE(trans.timestamp) AS fecha, SUM(amount) AS total_ventas FROM transaction AS trans WHER... 5 row(s) returned
0.218 sec / 0.000 sec
```

Nivel 2**Ejercicio 2**

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

```

146 •   SELECT comp.country AS pais, ROUND(AVG(trans.amount),2) AS media
147     FROM transaction AS trans
148   JOIN company AS comp
149   ON trans.company_id = comp.id
150 WHERE trans.declined = 0
151 GROUP BY pais
152 ORDER BY media DESC;

```

Result Grid		Filter Rows:
	pais	media
▶	Australia	265.54
	United States	264.42
	Belgium	260.97
	Germany	260.83
	Ireland	260.39
	Spain	260.28
	France	259.91
	New Zealand	259.59
	Norway	259.14
	Netherlands	258.34
	Italy	258.12
	Canada	257.41
	Sweden	257.39
	United Kingdom	256.68

8 12:32:55 SELECT comp.country AS pais, ROUND(AVG(trans.amount),2) AS media FROM transaction AS trans JOIN co... 15 row(s) returned 0.313 sec / 0.000 sec

Nivel 2**Ejercicio 3**

En tu empresa, se plantea un nuevo proyecto para lanzar algunas campañas publicitarias para hacerle competencia a la empresa “Non Institute”. Para ello, te pedimos la lista de todas las transacciones realizadas para empresas que están situadas en el mismo país que dicha empresa.

- a) Muestra la lista aplicando JOIN y subconsultas

11

```

161 •   SELECT trans.id AS id_transaccion, comp.company_name
162     FROM transaction AS trans
163       JOIN company AS comp
164         ON trans.company_id = comp.id
165   WHERE comp.country =  (    SELECT country
166     FROM company
167       WHERE company_name = 'Non Institute'
168   )
169   AND comp.company_name != 'Non Institute'
170 ;

```

* Aunque no se pida, creo que es interesante que se visualice, además del id de la transacción, la empresa relacionada. Por eso, he incluido el company_name. También filtré del listado las transacciones de la empresa 'Non Institute', ya que me piden las transacciones de las empresas del mismo país que dicha empresa, justamente para lanzar campañas publicitarias y competir con ella. No interesa tener las transacciones de la empresa con la que quiero competir, ya que no le enviaremos campañas publicitarias.

	Result Grid	Filter Rows:	Export:
	id_transaccion	company_name	
▶	008629B4-C9A9-406C-A3D2-71FDA47BC546	Sed Nunc Ltd	
	00B72BA4-54A3-4B8E-B13F-2D57535AA17A	Sed Nunc Ltd	
	01F075B1-D7AE-4D02-AA09-5FFD72A43F3C	Sed Nunc Ltd	
	023FFCE8-E618-4938-BF56-C8DF80540ADD	Sed Nunc Ltd	
	02683BE8-EF91-4564-957B-D6F1662AB7C5	Sed Nunc Ltd	
	02C2F29E-CEF2-4C1E-A594F476EB8F279C0	Sed Nunc Ltd	
	02F468DC-426C-47C2-8B0A-D8B25B7A81AF	Sed Nunc Ltd	
	0306BE3B-817B-4A49-934E-0E439291A104	Sed Nunc Ltd	
	0347BFE6-8EB5-4387-B187-0E78E8F2B8FB	Sed Nunc Ltd	
	03AEBD0E-DC97-4BD3-9C57-6A6DB78026FD	Sed Nunc Ltd	
	03CA36D3-88FF-4DBF-BFD4-4CC7DA4ED2B	Sed Nunc Ltd	
	04494182-96DD-42EB-82FE-5F92C5210537	Sed Nunc Ltd	
	045AACF6-FF85-49FB-9DE4-E6730655366A	Sed Nunc Ltd	

9 12:36:14 SELECT trans.id AS id_transaccion, comp.company_name FROM transaction AS trans JOIN company AS com... 12233 row(s) returned 0.016 sec / 0.078 sec

- b) Muestra la lista aplicando solo subconsultas

```

174 • Ⓜ SELECT trans.id AS id_transaccion, (   SELECT comp.company_name
175
176
177     FROM transaction AS trans
178     WHERE trans.company_id IN (    SELECT comp.id
179         FROM company AS comp
180         WHERE country = (    SELECT country
181             FROM company AS c
182             WHERE c.company_name = 'Non Institute'
183                 )
184             AND comp.company_name != 'Non Institute'
185         );

```

	id_transaccion	company_name
▶	008629B4-C9A9-406C-A3D2-71FDA47BC546	Sed Nunc Ltd
	00B72BA4-54A3-4B8E-B13F-2D57535AA17A	Sed Nunc Ltd
	01F075B1-D7AE-4D02-AAD9-5FFD72A43F3C	Sed Nunc Ltd
	023FFCE8-E618-4938-BF56-C8DF80540ADD	Sed Nunc Ltd
	02683BEB-EF91-4564-957B-D6F1662AB7C5	Sed Nunc Ltd
	02C2F29E-CEF2-4C1E-A594-F476E8F279C0	Sed Nunc Ltd
	02F468DC-426C-47C2-8B0A-D8B25B7A81AF	Sed Nunc Ltd
	0306BE3B-817B-4A49-934E-0E439291A104	Sed Nunc Ltd
	0347BFE6-8EB5-4387-B187-0E78E8F2B8FB	Sed Nunc Ltd
	03AEBD0E-DC97-4BD3-9C57-6A6DB78026FD	Sed Nunc Ltd
	03CA36D3-88FF-4DBF-BFD4-4CC7DA4EED2B	Sed Nunc Ltd
	04494182-96DD-42EB-82FE-5F92C5210537	Sed Nunc Ltd
	045AACF6-FF85-49FB-9DE4-E6730655366A	Sed Nunc Ltd
	0489FDAC-86A4-4929-954E-9A2272189CAC	Sed Nunc Ltd

14 12:46:20 SELECT trans.id AS id_transaccion, (SELECT comp.company_name FROM company AS comp ... 12233 row(s) returned 0.000 sec / 0.062 sec

Nivel 3

Ejercicio 1

Presenta el nombre, teléfono, país, fecha i cantidad de aquellas empresas que realizaron transacciones con un valor entre 350 y 400 euros y en alguna de las siguientes fechas: 29 de abril del 2015, 20 de julio de 2018 y 13 de marzo de 2024. Ordena los resultados de mayor a menor cantidad.

```
194 •   SELECT comp.company_name AS nombre_empresa, comp.phone AS telefono, comp.country AS pais, date AS fecha, t.cantidad AS total
195   FROM company AS comp
196   JOIN  (   SELECT trans.company_id, DATE(timestamp) AS date, ROUND(SUM(trans.amount),2) AS cantidad
197     FROM transaction AS trans
198    WHERE amount BETWEEN 350 AND 400
199    AND DATE(timestamp) IN ('2015-04-29', '2018-07-20', '2024-03-13')
200    GROUP BY trans.company_id, date
201  ) t
202 ON comp.id = t.company_id
203 ORDER BY t.cantidad DESC;
```

**He añadido los alias para que sea más clara la visualización de los datos, siguiendo las buenas prácticas.*

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	nombre_empresa	telefono	pais	fecha	total
▶	Aliquam PC	01 45 73 52 16	Germany	2024-03-13	788.13
	At Pede Corp.	06 14 48 33 15	Italy	2015-04-29	390.69
	Fringilla LLC	08 29 15 93 57	New Zealand	2015-04-29	367.62
	Pede Cum Ltd	07 62 26 48 38	Norway	2018-07-20	356.87
	Orci Adipiscing Limited	03 18 00 77 81	United Kingdom	2018-07-20	373.71
	Auctor Mauris Vel LLP	08 09 28 74 14	United States	2018-07-20	399.51
	Auctor Mauris Vel LLP	08 09 28 74 14	United States	2024-03-13	353.75

✓ 24 12:56:40 SELECT comp.company_name AS nombre_empresa, comp.phone AS telefono, comp.country AS pais, date A... 7 row(s) returned 0.140 sec / 0.000 sec

Nivel 3

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 donde especifiques si tienen más de 400 transacciones o menos.

```
211 • SELECT trans.company_id AS id_empresa, comp.company_name AS nombre_empresa, COUNT(trans.id) AS total_transacciones,  
212  
213  
214  
215     FROM transaction AS trans  
216     JOIN company AS comp  
217     ON trans.company_id = comp.id  
218     GROUP BY trans.company_id;  
  
CASE  
    WHEN COUNT(trans.id) > 400 THEN 'Más de 400'  
    ELSE 'Igual o menos de 400'  
END AS mas_menos_400
```

* Utilicé una JOIN para poder mostrar también el nombre de la empresa y además añadí alias en la misma línea que en el ejercicio anterior.

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	id_empresa	nombre_empresa	total_transacciones	mas_menos_400
▶	b-2286	Fringilla LLC	397	Igual o menos de 400
	b-2362	Lorem Eu Incorporated	380	Igual o menos de 400
	b-2470	Nec Luctus LLC	399	Igual o menos de 400
	b-2222	Ac Fermentum Incorporated	2401	Más de 400
	b-2226	Magna A Neque Industries	410	Más de 400
	b-2230	Fusce Corp.	447	Más de 400
	b-2234	Convallis In Incorporated	1514	Más de 400
	b-2238	Ante Iaculis Nec Foundation	472	Más de 400
	b-2242	Donec Ltd	449	Más de 400
	b-2246	Sed Nunc Ltd	1541	Más de 400

29 13:07:39 SELECT trans.company_id AS id_empresa, comp.company_name AS nombre_empresa, COUNT(trans.id) AS total_transacciones, C... 100 row(s) returned 0.156 sec / 0.000 sec