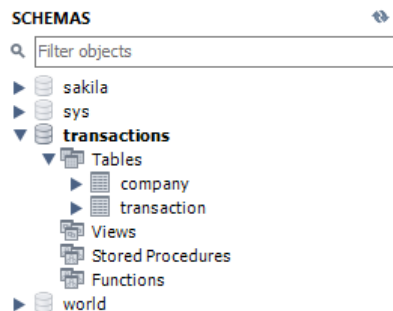


>>>>>>>>>>>>>>> SPRINT 1 <<<<<<<<<<<<<<<<<<<

===== 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.



Se creó un esquema en base al código que contenía los archivos facilitados, este esquema consta de 2 tablas:

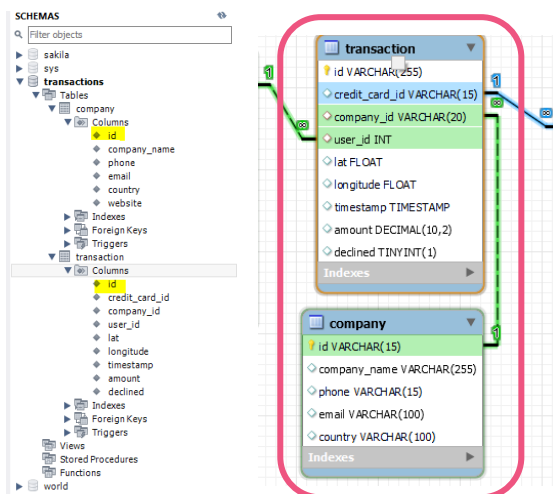
- company
- transaction

La tabla “company” contiene columnas:

id, company_name, phone, email, country, website

La tabla "transaction" contiene columnas:

Id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined



Relación: Las tablas transaction y company se relacionan mediante la columna company_id en la tabla transaction. Esta columna es una clave externa que hace referencia a la columna id en la tabla company. Esto significa que cada transacción en la tabla transaction está asociada con una empresa en la tabla company.

Cardinalidad: La cardinalidad de la relación entre transaction y company es de uno a muchos. Esto significa que una empresa puede tener muchas transacciones, pero una transacción solo puede estar asociada con una empresa, por esta razón vemos en el grafico el número 1 y el signo de infinito.

- Ejercicio 2

Realiza la siguiente consulta: Debes obtener el nombre, email y país de cada compañía, ordena los datos en función del nombre de las compañías.

```
SELECT company_name, email, country
FROM transactions.company
ORDER BY company_name;
```

ORDER BY: nos ayuda para ordenar por el nombre de la compañía, ya que los contenidos de búsqueda son textos, si fuera numérico se usaría ASC (ascendente) / DESC (descendente)

```

2  -- Ejercicio 2
3  -- Realiza la siguiente consulta: Debes obtener el nombre, email y país de cada compañía,
4  -- ordena los datos en función del nombre de las compañías.
5  • SELECT company_name, email, country
6  FROM transactions.company
7  ORDER BY company_name;
8
9

```

company_name	email	country
A Institute	metus.aliquam@google.edu	Belgium
Ac Fermentum Incorporated	donec.porttitor.tellus@yahoo.net	Germany
Ac Industries	ipsum@yahoo.com	Germany
Ac Libero Inc.	mollis.lectus@protonmail.ca	United Kingdom
Aliquam Erat Vulputat LLP	pede.nunc@icloud.net	Italy
Aliquam Iaculis Lacus Corp.	dictum@aol.org	Belgium
Aliquam PC	scelerisque.mollis@icloud.org	Germany
Aliquet Diam Limited	eu.eros.nam@icloud.org	United States
Aliquet Sem Limited	sem.magna@yahoo.edu	Netherlands
Aliquet Vel Vulputate Incorporated	suspendisse.commodo@outlook.co.uk	Netherlands
Amet Faucibus Ut Foundation	malesuada@icloud.net	United Kingdom
Amet Institute	nullam.lobortis.quam@outlook.net	Australia
Amet Lorem LLP	ultrices.adipiscing@icloud.edu	Spain
Amet Luctus Vulputate Foundation	nulla@outlook.com	Canada
Amet Nulla Donec Corporation	mattis.integer.eu@protonmail.net	Italy
Ante Iaculis Nec Foundation	sed.dictum.proin@outlook.ca	New Zealand
Arcu LLP	dui@aol.ca	Norway
At Associates	tristique.neque@yahoo.co.uk	New Zealand
At Pedes Corp.	ac.mi.eleifend@hotmail.edu	Italy
Auctor Mauris Corp.	eget.metus@protonmail.ca	Germany
Auctor Mauris Vel LLP	nec.tempus@icloud.co.uk	United States
Augue Foundation	mauris@yahoo.com	Germany
Convallis In Incorporated	mauris.ut@aol.co.uk	Germany
Cras Consulting	sed.consequat@google.ca	Belgium
Cras Vehicula Aliquet Industries	arcu@hotmail.org	Netherlands
Dictum Eu Corp.	donec.vitae@icloud.ca	Canada
Dis Parturient Institute	purus@protonmail.org	Ireland
Dolor Vitae Limited	purus.maecenas@yahoo.edu	France
Donec Fringilla PC	ut.tincidunt@hotmail.ca	France
Donec Ltd	at.iaculis@hotmail.co.uk	Norway
Dui Cras Associates	tristique@yahoo.com	Italy
Dui Quis Institute	luctus.sit.amet@yahoo.co.uk	New Zealand

-- Ejercicio 3

Desde la sección de marketing te solicitan que les pases un listado de los países que están realizando compras

```

SELECT DISTINCT company.country
FROM transactions.company
JOIN transactions.transaction ON transaction.company_id = company.id;
10  -- Ejercicio 3
11  -- Desde la sección de marketing te solicitan que les pases un listado de los países que están realizando compras
12  • SELECT DISTINCT company.country
13  FROM transactions.company
14  JOIN transactions.transaction ON transaction.company_id = company.id;
15
16

```

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

SUGERENCIA:

En este caso el pedido hubiera sido interesante agregar una columna con los valores de compra por país

```

SELECT DISTINCT company.country, SUM(amount) AS total_venta
FROM transactions.company
JOIN transactions.transaction ON transaction.company_id = company.id
GROUP BY company.country
order by total_venta desc;

```

country	total_venta
Germany	28815.98
United Kingdom	27073.17
Sweden	20588.59
Norway	17286.71
Ireland	17193.12
Canada	16448.52
United States	5256.05
Netherlands	4554.32
Italy	3427.03
Belgium	2965.92
Australia	2784.63
New Zealand	2445.05
France	1075.19
China	682.67
Spain	106.80

-- Ejercicio 4

Desde marketing también quieren saber desde cuántos países se realizan las compras

```
SELECT COUNT(DISTINCT company.country) AS total_countries
FROM transactions.company
INNER JOIN transactions.transaction ON transaction.company_id = company.id
WHERE transaction.declined = 0;
```

```
22 -- Ejercicio 4
23 -- Desde marketing también quieren saber desde cuántos países se realizan las compras
24 • SELECT COUNT(DISTINCT company.country) AS total_countries
25 FROM transactions.company
26 INNER JOIN transactions.transaction ON transaction.company_id = company.id
27 WHERE transaction.declined = 0;
28
29
```

total_countries
15

-- Ejercicio 5

Tu jefe identifica un error con la compañía que tiene vaya 'b-2354'.

Por tanto, te solicita que le indiques el país y nombre de compañía de este ve.

```
SELECT c.country, c.company_name
FROM transactions.company c
WHERE c.id = 'b-2354';
```

```
33 • SELECT c.country, c.company_name
34 FROM transactions.company c
35 WHERE c.id = 'b-2354';
36
```

country	company_name
United Kingdom	Ac Libero Inc.

-- Ejercicio 6

Además, ¿tu jefe te solicita que indiques cuál es la compañía con mayor gasto medio?

```
SELECT company_name, AVG(transaction.amount) AS average_spend
FROM transactions.company
INNER JOIN transactions.transaction ON transaction.company_id = company.id
GROUP BY company.id
ORDER BY average_spend DESC
LIMIT 1;
```

```
42 • SELECT company_name, AVG(transaction.amount) AS average_spend
43 FROM transactions.company
44 INNER JOIN transactions.transaction ON transaction.company_id = company.id
45 GROUP BY company.id
46 ORDER BY average_spend DESC
47 LIMIT 1;
```

company_name	average_spend
Eget Ipsum Ltd	473.075000

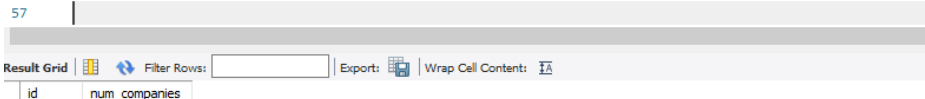
===== NIVEL 2 =====

-- Ejercicio 1

Tu jefe está redactando un informe de cierre del año y te solicita que le envíes información relevante para el documento.

Para ello te solicita verificar si en la base de datos existen compañías con identificadores (id) duplicados.

```
SELECT company.id, COUNT(*) AS num_companies
FROM transactions.company
GROUP BY company.id
HAVING COUNT(*) > 1;
51 • SELECT company.id, COUNT(*) AS num_companies -- COUNT() para el conteo de las id y luego la asignación
52 FROM transactions.company
53 GROUP BY company.id
54 HAVING COUNT(*) > 1; -- filtro para indicar las compañías que se repiten mas de 1 vez
55 -- CONCLUSION: aparentemente no hay duplicados en la base de datos
56
57
```



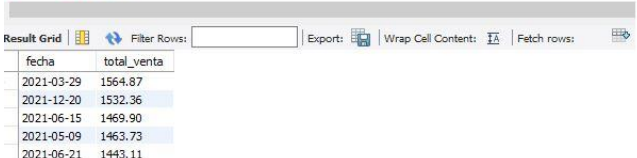
id	num_companies
----	---------------

Aparentemente no tenemos resultados duplicados conforme la petición y búsqueda conforme el filtrado de `HAVING COUNT(*) > 1;`

-- Ejercicio 2

¿En qué día se realizaron las cinco ventas más costosas? Muestra la fecha de la transacción y la sumatoria de la cantidad de dinero.

```
SELECT DATE(timestamp) as fecha, SUM(amount) AS total_venta
FROM transactions.transaction
GROUP BY fecha
ORDER BY total_venta DESC -- ordenado por el total de venta en descenso
LIMIT 5;
63 • SELECT DATE(timestamp) as fecha, SUM(amount) AS total_venta
64 FROM transactions.transaction
65 GROUP BY fecha
66 ORDER BY total_venta DESC
67 LIMIT 5;
```

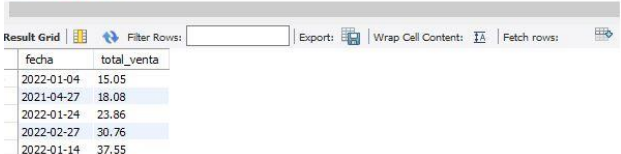


fecha	total_venta
2021-03-29	1564.87
2021-12-20	1532.36
2021-06-15	1469.90
2021-05-09	1463.73
2021-06-21	1443.11

-- Ejercicio 3

¿En qué día se realizaron las cinco ventas de menor valor? Muestra la fecha de la transacción y la sumatoria de la cantidad de dinero.

```
SELECT DATE(timestamp) as fecha, SUM(amount) AS total_venta
FROM transactions.transaction
GROUP BY fecha
ORDER BY total_venta ASC -- ordenado por el total de venta en descenso
LIMIT 5;
72 • SELECT DATE(timestamp) as fecha, SUM(amount) AS total_venta
73 FROM transactions.transaction
74 GROUP BY fecha
75 ORDER BY total_venta ASC
76 LIMIT 5;
```



fecha	total_venta
2022-01-04	15.05
2021-04-27	18.08
2022-01-24	23.86
2022-02-27	30.76
2022-01-14	37.55

-- Ejercicio 4

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

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

```
76 -- Ejercicio 4
77 -- ¿Cuál es la media de gasto por país? Presenta los resultados
78 • SELECT country, AVG(transaction.amount) AS average_amount
79 FROM transaction
80 INNER JOIN company ON transaction.company_id = company.id
81 GROUP BY company.country
82 ORDER BY average_amount DESC;
```

country	average_amount
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

-- ===== NIVEL 3 =====

-- Ejercicio 1

Presenta el nombre, teléfono y país de las compañías, junto con la cantidad total gastada, de aquellas que realizaron transacciones con un gasto comprendido entre 100 y 200 euros. ordena los resultados de mayor a menor cantidad gastada.

```
SELECT company_name, phone, country, t.amount AS cantidad_total
FROM transaction t
INNER JOIN company c ON t.company_id = c.id
WHERE t.amount BETWEEN 100 AND 200
ORDER BY cantidad_total DESC;
```

```
100 • SELECT company_name, phone, country, t.amount AS cantidad_total
101 FROM transaction t
102 INNER JOIN company c ON t.company_id = c.id
103 WHERE t.amount BETWEEN 100 AND 200
104 ORDER BY cantidad_total DESC;
```

company_name	phone	country	cantidad_total
Loren Eu Incorporated	01 83 66 62 07	Canada	199.83
Ut Sempser Foundation	01 60 36 33 06	Sweden	199.81
At Associates	09 56 61 10 65	New Zealand	199.71
Amet Nulla Donec Corporation	07 15 25 14 74	Italy	197.65
Non Institute	06 77 15 31 14	United Kingdom	196.71

-- Ejercicio 2

Indica el nombre de las compañías que realizaron compras el 16 de marzo de 2022, 28 de febrero de 2022 y 13 de febrero de 2022.

```
SELECT DISTINCT company.company_name, transaction.timestamp
FROM transaction
INNER JOIN company ON transaction.company_id = company.id
WHERE date(timestamp) IN ('2022-03-16', '2022-02-28', '2022-02-13')
GROUP BY company_name, transaction.timestamp;
```

```
98 -- Ejercicio 2
99 -- Indica el nombre de las compañías que realizaron compras el 16 de marzo de 2022,
100 -- 28 de febrero de 2022 y 13 de febrero de 2022.
101 • SELECT DISTINCT company.company_name, transaction.timestamp
102 FROM transaction
103 INNER JOIN company ON transaction.company_id = company.id
104 WHERE date(timestamp) IN ('2022-03-16', '2022-02-28', '2022-02-13')
105 GROUP BY company_name, transaction.timestamp;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
company_name	timestamp			
Sed LLC	2022-02-13 16:33:50			
Arcu LLP	2022-02-13 12:02:43			
Nunc Interdum I...	2022-02-13 04:07:29			
Nunc Interdum I...	2022-02-28 05:55:03			
Ut Semper Foun...	2022-02-28 00:10:50			
Lorem Eu Incorp...	2022-03-16 14:01:36			
Malesuada PC	2022-02-13 15:13:38			
Malesuada PC	2022-02-28 08:09:21			
Malesuada PC	2022-03-16 02:35:05			