SPRINT 2

Nivell 1

Ejercicio 1

Input 1: Creo la estructura base de la BBDD

```
CREATE DATABASE Sprint2
USE Sprint2

-- Creamos la tabla company
    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)
    );


    -- Creamos la tabla transaction
    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)
    );
```

Output 1:

#	Time	Action	Message
1	11:04:17	CREATE DATABASE Sprint2	1 row(s) affected
2	11:04:40	USE Sprint2	0 row(s) affected
3	11:11:33	CREATE TABLE IF NOT EXISTS company (id VARCHAR(15) PRIMARY KEY, company_name VARCHAR(255), phone VARCHAR(15), email VA...	0 row(s) affected
4	11:11:33	CREATE TABLE IF NOT EXISTS transaction (id VARCHAR(255) PRIMARY KEY, credit_card_id VARCHAR(15) REFERENCES credit_card(id), compa...	0 row(s) affected

Input 2: Ejecuto los datos a introducir en las tablas

```
-- Insertamos datos de company
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2222', 'Ac Fermentum Incorporated', '06 85 56 52 33', 'donec.porttitor.tellus@yahoo.net', 'Germany', 'https://instagram.com
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2226', 'Magna A Neque Industries', '04 14 44 64 62', 'risus.donec.nibh@icloud.org', 'Australia', 'https://whatsapp.com/grou
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2230', 'Fusce Corp.', '08 14 97 58 85', 'risus@protonmail.edu', 'United States', 'https://pinterest.com/sub/cars');
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2234', 'Convallis In Incorporated', '06 66 57 29 50', 'mauris.ut@aol.couk', 'Germany', 'https://cnn.com/user/110');
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2238', 'Ante Iaculis Nec Foundation', '08 23 04 99 53', 'sed.dictum.proin@outlook.ca', 'New Zealand', 'https://netflix.com/
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2242', 'Donec Ltd', '01 25 51 37 37', 'at.iaculis@hotmail.couk', 'Norway', 'https://nytimes.com/user/110');
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2246', 'Sed Nunc Ltd', '02 62 64 73 48', 'nibh@yahoo.org', 'United Kingdom', 'https://cnn.com/one');
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2250', 'Amet Nulla Donec Corporation', '07 15 25 14 74', 'mattis.integer.eu@protonmail.net', 'Italy', 'https://netflix.com/
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2254', 'Nascetur Ridiculus Mus Inc.', '06 26 [illegible] 61 84', 'suspendisse.dui@icloud.net', 'United States', 'https://ebay.com/su
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2258', 'Vestibulum Lorem PC', '02 02 87 33 40', 'aenean.massa.integer@aol.net', 'Belgium', 'https://pinterest.com/sub/cars'
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2262', 'Gravida Sagittis LLP', '03 81 28 33 97', 'turpis.vitae@google.ca', 'Sweden', 'https://naver.com/site');
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2266', 'Mus Aenean Eget Foundation', '06 25 15 52 43', 'mi.duis@hotmail.net', 'Sweden', 'https://instagram.com/group/9');
INSERT INTO company (id, company_name, phone, email, country, website) VALUES (       'b-2270', 'Dis Parturient Institute', '05 36 29 78 74', 'purus@protonmail.org', 'Ireland', 'https://google.com/one');
```

Output 2:

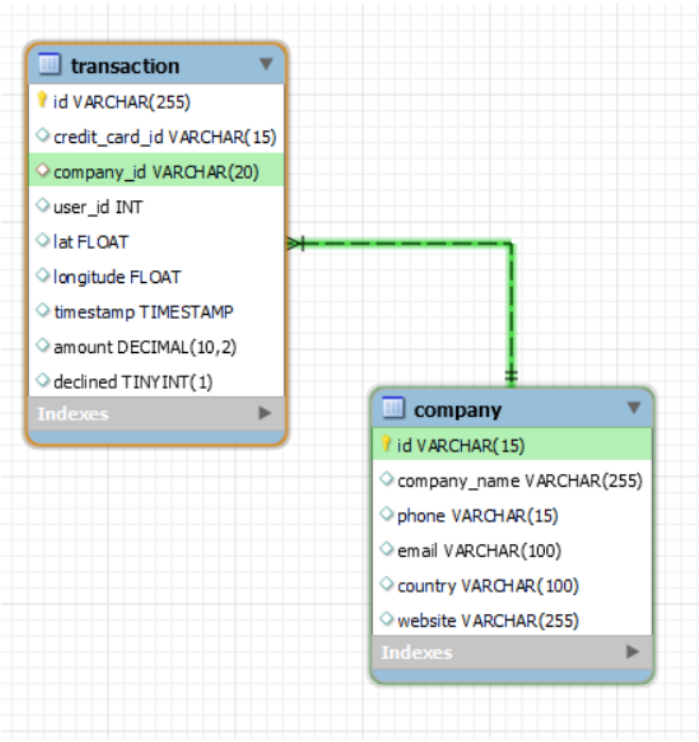
#	Time	Action	Message
718	11:45:49	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('1517E8A8-8B44-A7C9-6691-692C27...	1 row(s) affected
719	11:45:49	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('C1DCC32D-E42A-F2D6-17AD-FF4A6...	1 row(s) affected
720	11:45:49	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('5C3A8A3D-8974-40A6-A8AF-C1F7AA...	1 row(s) affected
721	11:45:49	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('D6EB93C6-5733-84C8-FDD6-57BA41...	1 row(s) affected
722	11:45:49	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('A4D0D84F-4622-BB83-E6B6-51E545...	1 row(s) affected
723	11:45:49	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('9FBB3D61-D3C2-E5BB-4BC3-6CC83...	1 row(s) affected

1.1.1 Explicación tablas y variables (diagrama)

El modelo está formado por dos tablas, la tabla “transactions” que es la tabla de hechos y la tabla de “company” la tabla de dimensiones. La relación entre Company y transactions por tanto es de uno a varios. Las tablas están conectadas a través de las claves id(Company) y transaction(Company_id).

En el posterior diagrama podemos ver los tipos de datos de las variables.

Parte2-Mostrar diagrama



Vemos que en la tabla “Company” todas las variables son VARCHAR o cadenas. En “transactions” también hay variables de otros tipos como entero, flotantes, decimales, fecha y hora ('YYYY-MM-DD HH:MM’), TINYINT(boolean).

Ejercicio 2

Input 1: Uso el distinct para obtener un listado de los países donde se ubican nuestros clientes, el cual no muestre los duplicados.

```
SELECT distinct country FROM company JOIN transaction ON company.id = transaction.company_id;
```

Output 1:

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

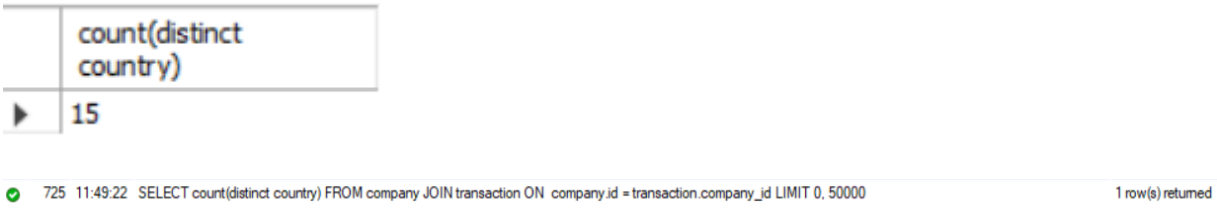
Result 8

724 11:47:01 SELECT distinct country FROM company JOIN transaction ON company.id = transaction.company_id LIMIT 0, 50000 15 row(s) returned

Input 2: Aquí le agrego count como operación de agregación a la query anteriormente realizada. Esta operación me retornara como resultado la cantidad de países distintos a los que pertenecen las empresas con las que trabajamos.

```
SELECT count(distinct country) FROM company JOIN transaction ON  company.id = transaction.company_id;
```

Output 2:



count(distinct country)
15

725 11:49:22 SELECT count(distinct country) FROM company JOIN transaction ON company.id = transaction.company_id LIMIT 0, 50000 1 row(s) returned

Input 3: En la tercera query quiero conocer el promedio de compra agrupado por compañía, en concreto quiero saber cuál de todas las compañías es la que posee el mayor promedio de amount, y para eso ordenamos en DESC y ponemos el LIMIT 1. Además, en el WHERE se eligen solo las transacciones que no han sido declinadas.

```
SELECT company_name, avg(amount)
FROM company
JOIN transaction ON company.id = transaction.company_id
WHERE transaction.declined = 0
group by company_name
order by avg(amount) DESC
LIMIT 1;
```

Output 3:

	company_name	avg(amount)
▶	Eget Ipsum Ltd	481.860000

726 11:50:28 SELECT company_name, avg(amount) FROM company JOIN transaction ON company.id = transaction.company_id WHERE transaction.declined = 0 group by co... 1 row(s) returned

Ejercicio 3

Input 1: A través de subquery le digo que solo me retorne los datos de transacciones de las empresas alemanas.

```
SELECT id FROM transaction WHERE company_id IN (SELECT id FROM company WHERE country = 'Germany');
```

Output 1:

id
108B1D1D-5B23-A76C-55EF-C568E49A05DD
EA2C3281-C9C1-A387-44F8-729FB4B51C76
0DD2E608-5C9E-D1B3-4999-B99F43AD735A
AB069F53-965E-A2A8-CE06-CA8C4FD92501
0466A42E-47CF-8D24-FD01-C0B689713128
0A476ED9-0C13-1962-F87B-D3563924B539
122DC333-E19F-D629-DCD8-9C54CF1EBB9A
135267BA-2E7D-957C-C42C-6450A2B3ED54
14CAE5B5-8FB1-3E4A-4C85-0EA4167534F4

727 11:53:19 SELECT id FROM transaction WHERE company_id IN (SELECT id FROM company WHERE country = 'Germany') LIMIT 0, 50000 118 row(s) returned

Input 2: Realizo query pidiendo el nombre de las compañías que están por encima del promedio en amount, usando el método de subquery de WHERE.

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

Output 2:

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

728 11:53:19 SELECT company_name FROM company WHERE id IN (SELECT company_id FROM transaction WHERE amount > (SELECT AVG(amount) F... 70 row(s) returned

Input 3: Busco conocer las empresas que no tienen transacciones, para ello uso la cláusula NOT IN.

```
SELECT c.company_name
FROM company c
WHERE c.id NOT IN (
    SELECT t.company_id
    FROM transaction t);
```

Output 3: No hay empresas sin transacciones

company_name

```
64  22:56:45  SELECT c.company_name  FROM company c  WHERE c.id NOT IN ( SELECT t.company_id  FROM transaction t) LIMIT 0, 50000  0 row(s) returned
```

Input 4 Testeo para confirmar que no haya ninguna empresa sin transacciones.

```
SELECT c.company_name, count(t.id)
FROM company c
JOIN transaction t ON c.id = t.company_id
GROUP BY c.company_name
ORDER BY count(t.id);
```

Output 4 El resultado muestra como las empresas con menos transacciones tienen una transacción como mínimo.

company_name	count(t.id)
Dui Quis Institute	1
Nibh Phasellus Corporation	1
Orci Adipiscing Limited	1
Aliquet Diam Limited	1
Tempor Diam Institute	1
Euismod Mauris Institute	1

```
65  23:01:41  SELECT c.company_name, count(t.id) FROM company c  JOIN transaction t ON c.id = t.company_id GROUP BY c.company_name ORDER BY count(t.id) LIMIT 0, 50000  100 row(s) returned
```

Nivell 2

Ejercicio 1

Input 1: Esta query busca reconocer los cinco días con más ventas. Al no disponer de columna fecha he aplicado la función DATE sobre timestamp(fecha-hora), lo que lo convierte en fecha dentro de la query. Posteriormente agrupo por esta misma columna, ordeno por la suma total de amount en orden DESC y limito a cinco outputs con LIMIT 5.

```sql
SELECT DATE(timestamp), SUM(amount)
FROM transaction
WHERE declined = 0
GROUP BY DATE(timestamp)
ORDER BY sum(amount) DESC
LIMIT 5;
```

Output 1:

	DATE(timestamp)	SUM(amount)
▶	2021-12-20	1532.36
	2021-04-22	1397.96
	2021-05-09	1344.37
	2022-02-26	1337.62
	2021-03-29	1325.12

733 12:50:21 SELECT DATE(timestamp), SUM(amount) FROM transaction WHERE declined = 0 GROUP BY DATE(timestamp) ORDER BY sum(amount) DESC LIMIT 5 5 row(s) returned

Ejercicio 2

Input 1: En esta query agrupo país con su promedio de amount y ordenado por amount en orden descendiente. El objetivo es destacar los países con importes superiores.

```sql
SELECT company.country, avg(amount) FROM company JOIN transaction ON company.id = transaction.company_id
WHERE declined = 0
GROUP BY company.country
ORDER BY avg(amount) DESC;
```

Output 1:

	country	avg(amount)
▶	United States	287.531111
	Ireland	285.825357
	Sweden	276.668382
	United Kingdom	271.767527
	Canada	261.941930
	Belgium	255.217500
	Norway	251.114918
	Italy	243.342222

734 12:57:32 SELECT company.country, avg(amount) FROM company JOIN transaction ON company.id = transaction.company_id WHERE declined = 0 GROUP BY company.c... 15 row(s) returned

Ejercició 3

Input 1: La query retorna las transacciones de las empresas que comparten país con 'Non Institute'. Se utiliza una subquery basada en WHERE para conectar ambas tablas y abajo se realiza un Self Join (c1 y c2) para comprobar que el país coincide con el país de la empresa.

```
SELECT t.id
FROM transaction t
WHERE t.company_id IN
(SELECT c1.id
FROM company c1
JOIN company c2
WHERE c1.country = c2.country
AND c2.company_name = 'Non Institute');
```

Output 1:

id
2B928E1C-EC14-A760-0A75-871477649D6A
ACD2011A-A2B1-C365-41E1-2AB00C65147A
4334349E-CEB0-3D68-A4D4-FEB7718A1ACE
BC2B9A38-77B4-28CD-1FE8-14DED863E773

Input 2: Repito la operación anteriormente realizada, pero ahora sin utilizar JOIN.

Empezando por abajo, la primera query retorna el país al que pertenece 'Non Institute'. Y a continuación lo conectamos con la clave primaria 'id' de la tabla Company, para finalmente poderlo conectar con la clave foránea de la tabla transaction 'company_id', y seleccionar el id de transaction que ha pasado por los filtros anteriormente realizados.

```sql
SELECT id
FROM transaction
WHERE company_id IN(
    SELECT id
    FROM company
    WHERE country = (
            (SELECT country
            FROM company
            WHERE company_name = "Non Institute")
            )
        );
```

Output 2:

	id
▶	2B928E1C-EC14-A760-0A75-871477649D6A
	ACD2011A-A2B1-C365-41E1-2AB00C65147A
	4334349E-CEB0-3D68-A4D4-FEB7718A1ACE
	BC2B9A38-77B4-28CD-1FE8-14DED863E773
	1479B3D2-B7BA-C7BB-4CE3-8D7C2DE85ABB
	152598C2-029D-D684-4B66-91EDF393EBFF
	1B636B58-A2E8-7C69-D9C9-C54535DAFD3B

741 13:02:11 SELECT id FROM transaction WHERE company_id IN(SELECT id FROM company WHERE country IN(SELECT country FROM company WHERE country IN (S... 100 row(s) returned

Nivell 3

Ejercicio 1

Input 1: En esta query quiero obtener detalles de venta de las empresas. filtrando por aquellas que su amount se encuentra entre 100 y 200 usando 'BETWEEN" y que además tengan transacciones en alguna de las 3 fechas escritas en el where.

```sql
SELECT c.company_name, c.phone,c.country,DATE(t.timestamp),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;
```

Output 1:

	company_name	phone	country	timestamp	amount
▶	Interdum Feugiat Sed Associates	04 88 40 32 52	United Kingdom	2021-07-20 10:12:15	164.86
	Nunc Interdum Incorporated	05 18 15 48 13	Germany	2022-03-13 13:37:34	164.32
	Enim Condimentum Ltd	09 55 51 66 25	United Kingdom	2021-04-29 06:17:02	149.89
	Lorem Eu Incorporated	01 83 66 62 07	Canada	2021-07-20 08:47:44	133.39
	Nunc Interdum Incorporated	05 18 15 48 13	Germany	2021-04-29 02:25:49	111.51

714 20:52:21 SELECT c.company_name, c.phone,c.country,DATE(t.timestamp),t.amount FROM company c JOIN transaction t ON c.id = t.company_id WHERE t.amount BETWEEN ... 5 row(s) returned

Ejercicio 2

Input 1: Uso de case para crear una columna con varias categorías, dependiendo en este caso de cuantas transacciones(t.id) tiene una misma empresa. Así mismo por eso al final es agrupado por Company_name.

```
SELECT c.company_name, count(t.id) AS transaction_count,
CASE WHEN count(t.id) > 4  THEN 'more than 4'
ELSE 'is 4 or lower'
END AS clasification_count
FROM company c
JOIN transaction t ON c.id = t.company_id
GROUP BY  c.company_name;
```

Output 1:

company_name	transaction_count	clasification_count
Ac Fermentum Incorporated	2	is 4 or lower
Magna A Neque Industries	2	is 4 or lower
Fusce Corp.	2	is 4 or lower
Convallis In Incorporated	2	is 4 or lower
Ante Iaculis Nec Foundation	2	is 4 or lower
Donec Ltd	2	is 4 or lower
Sed Nunc Ltd	2	is 4 or lower

749 14:44:19 SELECT c.company_name, count(t.id) AS transaction_count, CASE WHEN count(t.id) > 4 THEN 'more than 4' ELSE 'is 4 or lower' END AS clasification_count FR... 100 row(s) returned