

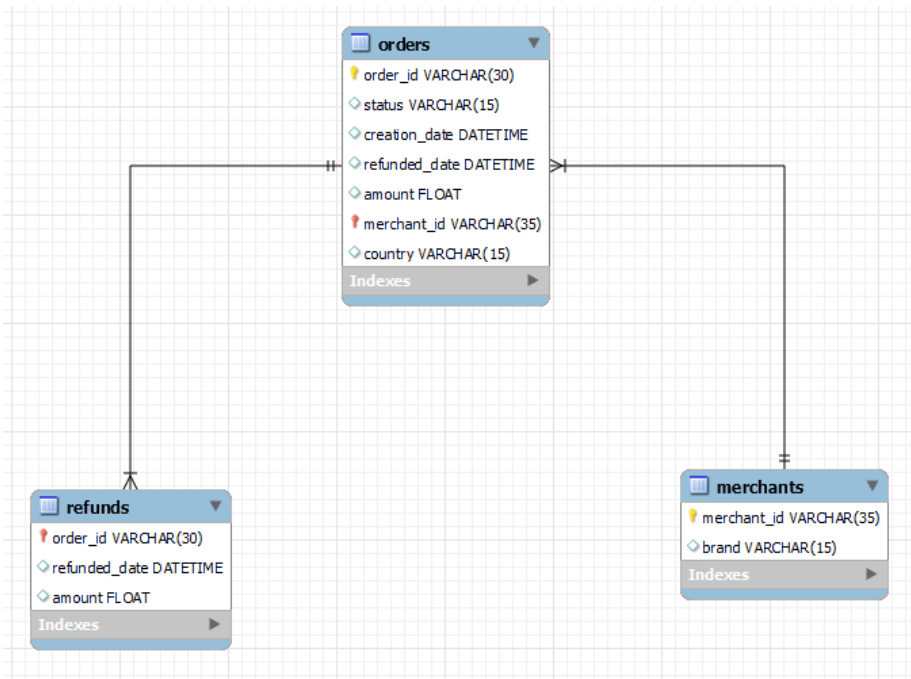
BASE DE DATOS: TAREA FINAL

ÍNDICE

- Ejercicio 1.....	2
○ Modelo Entidad-Relación.....	2
○ Creación base de datos.....	3
○ Creación esquema.....	3
○ Creación de las tres tablas.....	3
- Ejercicio 2.....	4
○ Consulta 1.....	4
○ Consulta 2.....	5
- Ejercicio 3.....	6
○ Consulta 3.....	6
○ Consulta 4.....	7
- Ejercicio 4.....	8
○ Explicación de la query.....	8
○ Desarrollo de la query.....	8

EJERCICIO 1

Desarrollar un Modelo de Entidad-Relación indicando las entidades, atributos, relaciones y cardinalidad entre las tablas.



Dentro del Modelo Entidad-Relación encontramos:

- Las entidades del modelo: orders, refunds y merchants.
- Cada una de las entidades tiene una serie de atributos, que son las características que definen las mismas, podemos observar los diferentes atributos en la imagen del Modelo Entidad-Relación.
- La relación entre las tablas se produce a través de estos atributos, que suponen una característica común entre dos o más entidades, para una de estas entidades este atributo tendrá la característica de clave primaria y para la otra entidad relacionada con este atributo será una clave foránea.
- La cardinalidad es el tipo de relación que tienen las entidades entre sí, la cardinalidad será:
 - Cardinalidad “refunds” - “orders” -> n:1. Cada préstamo concedido puede tener varias devoluciones (porque se devuelva la cantidad prestada en varias operaciones), pero una devolución está relacionada solo con una concesión de préstamo.
 - Cardinalidad “orders” - “merchants” -> n:1. Cada préstamo está relacionado con una marca, pero cada marca puede estar vinculada a diferentes préstamos.

A continuación, vamos a realizar las siguientes consultas y para ello vamos a necesitar los archivos incluidos en el comprimido operaciones ucm.zip disponibles en la plataforma del máster:

- Crear una base de datos con el nombre tarea_ucm

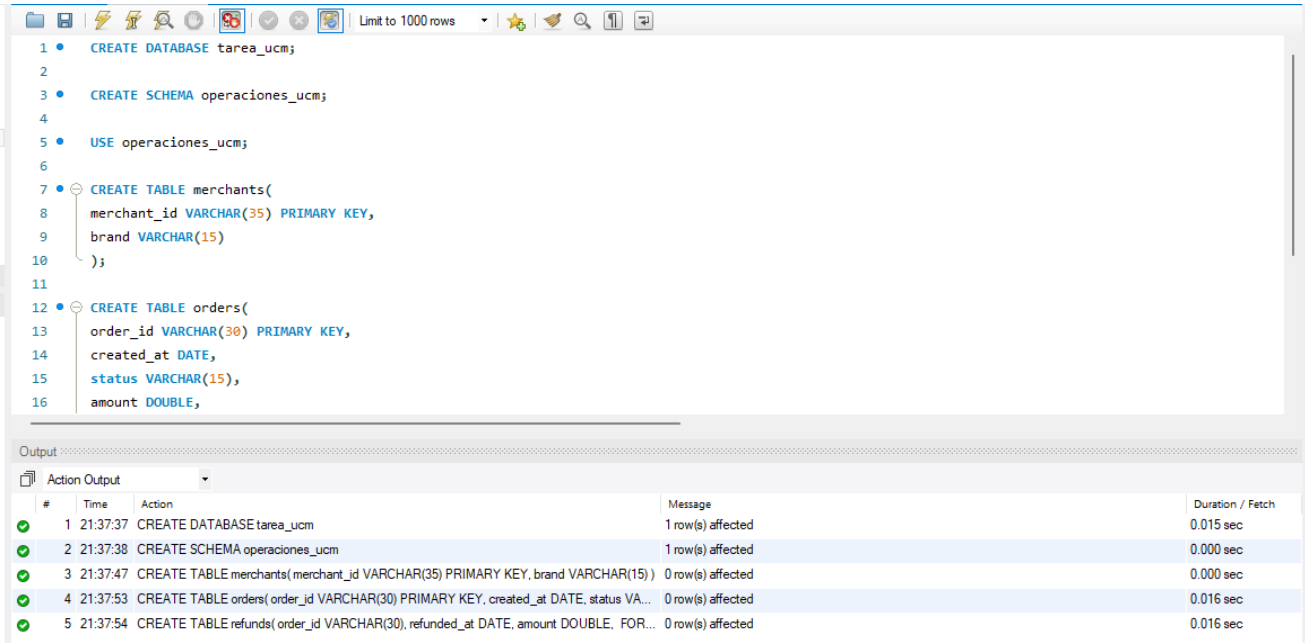
- Crear un esquema de base de datos con el nombre operaciones_ucm
- Creamos las tres tablas correspondientes a los 3 archivos: orders, refunds y merchants. Recuerda seleccionar el tipo de dato más adecuado para cada uno de los campos de las tres tablas.
- Cargamos los datos desde el.csv en nuestro MySQL Workbench.

```
CREATE DATABASE tarea_ucm;
CREATE SCHEMA operaciones_ucm;
```

```
USE operaciones_ucm;
CREATE TABLE merchants(
merchant_id VARCHAR(35) PRIMARY KEY,
brand VARCHAR(15)
);
```

```
CREATE TABLE orders(
order_id VARCHAR(30) PRIMARY KEY,
created_at DATE,
status VARCHAR(15),
amount DOUBLE,
merchant_id VARCHAR(35),
FOREIGN KEY (merchant_id) REFERENCES merchants(merchant_id)
);
```

```
CREATE TABLE refunds(
order_id VARCHAR(30),
refunded_at DATE,
amount DOUBLE,
FOREIGN KEY (order_id) REFERENCES orders(order_id)
);
```



The screenshot shows the MySQL Workbench interface. The top pane displays the SQL script being executed, and the bottom pane shows the 'Output' window with a table of execution results.

#	Time	Action	Message	Duration / Fetch
1	21:37:37	CREATE DATABASE tarea_ucm	1 row(s) affected	0.015 sec
2	21:37:38	CREATE SCHEMA operaciones_ucm	1 row(s) affected	0.000 sec
3	21:37:47	CREATE TABLE merchants(merchant_id VARCHAR(35) PRIMARY KEY, brand VARCHAR(15))	0 row(s) affected	0.000 sec
4	21:37:53	CREATE TABLE orders(order_id VARCHAR(30) PRIMARY KEY, created_at DATE, status VA...	0 row(s) affected	0.016 sec
5	21:37:54	CREATE TABLE refunds(order_id VARCHAR(30), refunded_at DATE, amount DOUBLE, FOR...	0 row(s) affected	0.016 sec

EJERCICIO 2

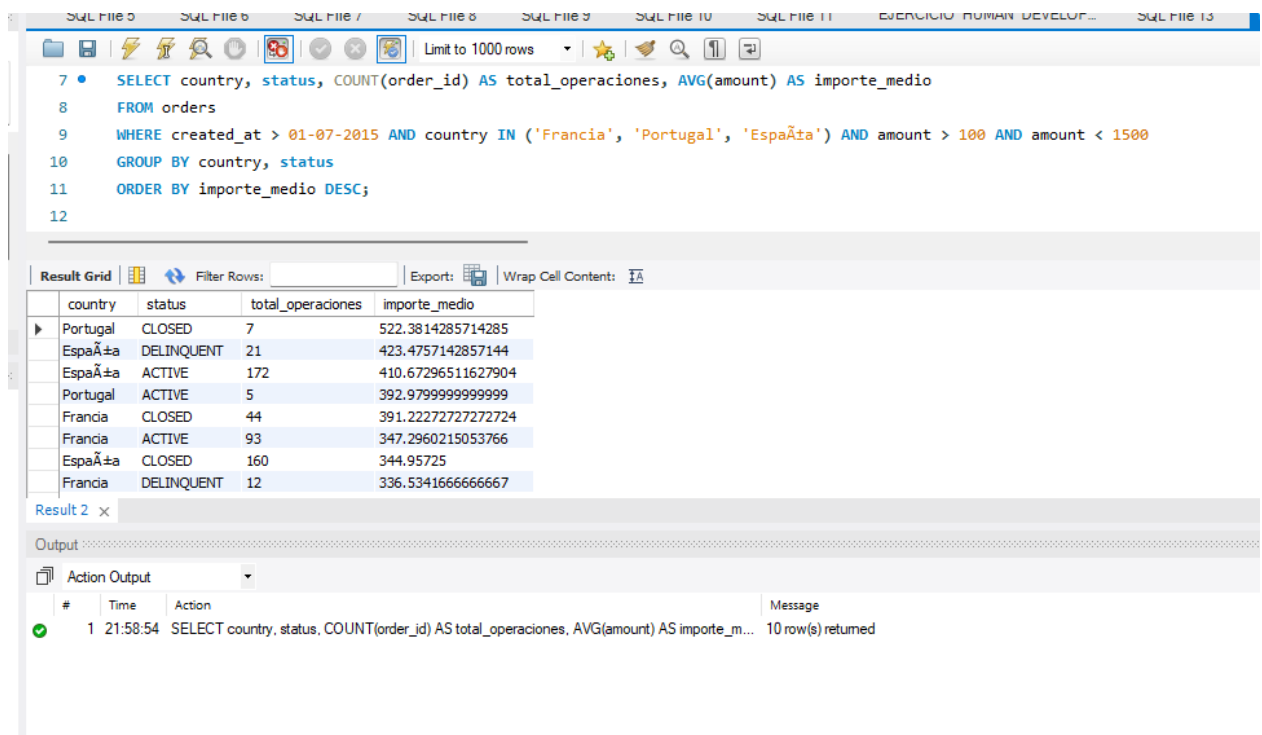
Consulta 1

Realizamos una consulta donde obtenemos por país y estado de operación, el total de operaciones y su importe promedio. La consulta debe cumplir las siguientes condiciones:

- Operaciones posteriores al 01-07-2015
- Operaciones realizadas en Francia, Portugal y España.
- Operaciones con un valor mayor de 100€ y menor de 1.500€

Ordenamos los resultados por el promedio del importe de manera descendente.

```
SELECT country, status, COUNT(order_id) AS total_operaciones, AVG(amount) AS
importe_medio
FROM orders
WHERE created_at > 01-07-2015 AND country IN ('Francia', 'Portugal', 'España') AND amount
> 100 AND amount < 1500
GROUP BY country, status
ORDER BY importe_medio DESC;
```



The screenshot shows a SQL IDE interface with a query editor and a results grid. The query is the same as the one provided in the previous block. The results grid displays the following data:

country	status	total_operaciones	importe_medio
Portugal	CLOSED	7	522.3814285714285
España	DELINQUENT	21	423.4757142857144
España	ACTIVE	172	410.67296511627904
Portugal	ACTIVE	5	392.97999999999999
Francia	CLOSED	44	391.22272727272724
Francia	ACTIVE	93	347.2960215053766
España	CLOSED	160	344.95725
Francia	DELINQUENT	12	336.5341666666667

Below the results grid, the output pane shows the execution message: "1 21:58:54 SELECT country, status, COUNT(order_id) AS total_operaciones, AVG(amount) AS importe_m... 10 row(s) returned".

Consulta 2.

Realizamos una consulta donde obtengamos los 3 países con el mayor número de operaciones, el total de operaciones, la operación con un valor máximo y la operación con el valor mínimo para cada país. La consulta debe cumplir las siguientes condiciones:

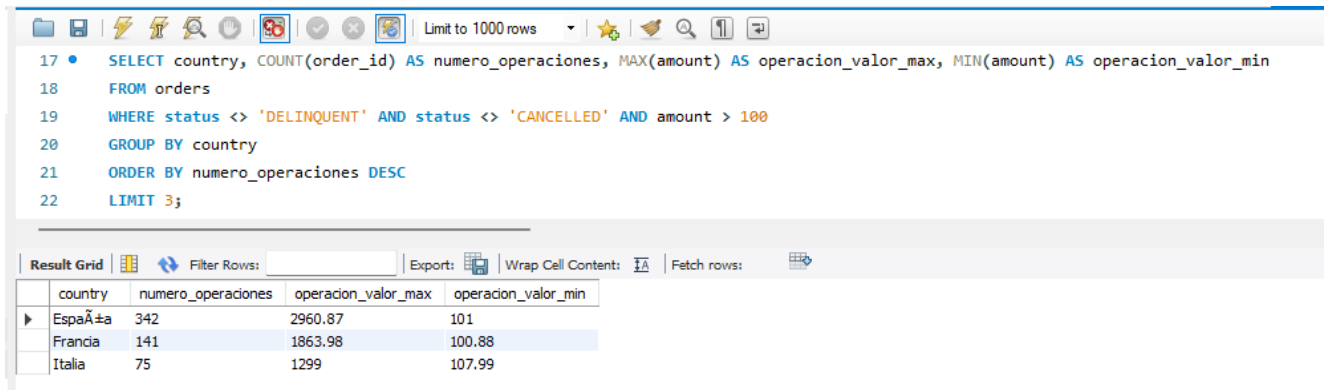
- Excluimos aquellas operaciones con el estado “Delinquent” y “Cancelled”.
- Operaciones con un valor mayor de 100€.

```
SELECT country, COUNT(order_id) AS numero_operaciones, MAX(amount) AS
operacion_valor_max, MIN(amount) AS operacion_valor_min
```

```

FROM orders
WHERE status <> 'DELINQUENT' AND status <> 'CANCELLED' AND amount > 100
GROUP BY country
ORDER BY numero_operaciones DESC
LIMIT 3;

```



The screenshot shows a SQL query editor with the following query:

```

17 SELECT country, COUNT(order_id) AS numero_operaciones, MAX(amount) AS operacion_valor_max, MIN(amount) AS operacion_valor_min
18 FROM orders
19 WHERE status <> 'DELINQUENT' AND status <> 'CANCELLED' AND amount > 100
20 GROUP BY country
21 ORDER BY numero_operaciones DESC
22 LIMIT 3;

```

Below the query, the results are displayed in a table:

country	numero_operaciones	operacion_valor_max	operacion_valor_min
España	342	2960.87	101
Francia	141	1863.98	100.88
Italia	75	1299	107.99

EJERCICIO 3

Consulta 3

Realizamos una consulta donde obtengamos, por país y comercio, el total de operaciones, su valor promedio y el total de devoluciones. La consulta debe cumplir las siguientes condiciones:

- Se debe mostrar el nombre y el id del comercio.
- Comercios con más de 10 ventas.
- Comercios de Marruecos, Italia, España y Portugal.
- Creamos un campo que identifique si el comercio acepta o no devoluciones. Si no acepta (total de devoluciones es igual a cero) el campo debe contener el valor “No” y si sí lo aceptan (total de devoluciones es mayor que cero) el campo debe contener el valor “Sí”. Llamaremos al campo “acepta devoluciones”.

Ordenamos los resultados por el total de operaciones de manera ascendente.

```

SELECT country, brand, merchant_id, COUNT(order_id) AS total_operaciones, AVG(amount) AS
valor_promedio, COUNT(refunded_at) AS total_devoluciones,
CASE
WHEN COUNT(refunded_at) = 0 THEN 'NO'
ELSE 'SI'
END AS acepta_devoluciones
FROM orders AS o
LEFT JOIN merchants AS m
ON o.merchant_id = m.`merchant ID`
LEFT JOIN refunds AS r
ON o.order_id = r.`order ID`
WHERE country IN ('Italia', 'España', 'Marruecos', 'Portugal')
GROUP BY country, brand, merchant_id
HAVING COUNT(order_id) > 10
ORDER BY total_operaciones ASC;

```

Limit to 1000 rows

```

46 • SELECT country, brand, merchant_id, COUNT(order_id) AS total_operaciones, ROUND(AVG(amount),2) AS valor_promedio, COUNT(refunded_at) AS total_devoluciones
47 CASE
48 WHEN COUNT(refunded_at) = 0 THEN 'NO'
49 ELSE 'SI'
50 END AS acepta_devoluciones
51 FROM orders AS o
52 LEFT JOIN merchants AS m
53 ON o.merchant_id = m.`merchant ID`
54 LEFT JOIN refunds AS r
55 ON o.order_id = r.`order ID`
56 WHERE country IN ('Italia', 'España', 'Marruecos', 'Portugal')
57 GROUP BY country, brand, merchant_id

```

Result Grid

	country	brand	merchant_id	total_operaciones	valor_promedio	total_devoluciones	acepta_devoluciones
▶	España	Pepe Jeans	pk_743f2fdecb876b75e975c005	11	171.99	0	NO
	Marruecos	Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	13	365.36	3	SI
	España	Massimo Dutti	pk_736c7094ea96eda38b098f56	13	169.88	0	NO
	España	Havaianas	pk_c15afcbd3a31b732f097ba7b	16	323.02	0	NO
	España	Netflix	pk_07225590b8fea17e739aa451	21	363.57	0	NO

Result 14 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:37:42	SELECT country, brand, merchant_id, COUNT(order_id) AS total_operaciones, ROUND(AVG(...	11 row(s) returned	0.000 sec / 0.000 sec

Consulta 4

Realizamos una consulta donde vamos a traer todos los campos de las tablas operaciones y comercios. De la tabla devoluciones vamos a traer el conteo de devoluciones por operación y la suma del valor de las devoluciones. Una vez tengamos la consulta anterior, creamos una vista con el nombre orders_view dentro del esquema tarea_ucm con esta consulta.

Nota: La tabla refunds contiene más de una devolución por operación por lo que, para hacer el cruce, es muy importante que agrupemos las devoluciones.

```

SELECT `merchant ID`, brand, order_id, created_at, status, amount, merchant_id, country,
COUNT(refunded_at) AS total_devoluciones, SUM(Cantidad) AS valor_devolucion
FROM merchants AS m
INNER JOIN orders AS o
ON m.`merchant ID` = o.merchant_id
LEFT JOIN refunds AS r
ON o.order_id = r.`order ID`
GROUP BY `merchant ID`, brand, order_id, created_at, status, amount, merchant_id, country;

```

56 • SELECT `merchant ID`, brand, order_id, created_at, status, amount, merchant_id, country, COUNT(refunded_at) AS total_devoluciones, SUM(Cantidad) AS val
57 FROM merchants AS m
58 INNER JOIN orders AS o
59 ON m.`merchant ID` = o.merchant_id
60 LEFT JOIN refunds AS r
61 ON o.order_id = r.`order ID`
62 GROUP BY `merchant ID`, brand, order_id, created_at, status, amount, merchant_id, country;

merchant_id	brand	order_id	created_at	status	amount	merchant_id	country	total_devoluciones	total_devolucion
pk_317b4fc6fd80a5f8fb2ff216	Calcedonia	5c3ef8170aee697c1ba8432a	2015-07-17 16:55:20	CLOSED	163.08	pk_317b4fc6fd80a5f8fb2ff216	Francia	2	
pk_317b4fc6fd80a5f8fb2ff216	Calcedonia	5c3ef8170aee697c1ba8432b	2015-07-21 11:25:26	CLOSED	773.14	pk_317b4fc6fd80a5f8fb2ff216	Portugal	1	
pk_317b4fc6fd80a5f8fb2ff216	Calcedonia	5c3ef8170aee697c1ba8432c	2015-07-23 16:52:13	CLOSED	191.05	pk_317b4fc6fd80a5f8fb2ff216	Alemania	2	
pk_317b4fc6fd80a5f8fb2ff216	Calcedonia	5c3ef8170aee697c1ba8432d	2015-07-23 17:30:59	CLOSED	235.53	pk_317b4fc6fd80a5f8fb2ff216	Alemania	1	
pk_317b4fc6fd80a5f8fb2ff216	Calcedonia	5c3ef8170aee697c1ba8432e	2015-07-24 14:07:56	CLOSED	302.06	pk_317b4fc6fd80a5f8fb2ff216	Alemania	1	
pk_317b4fc6fd80a5f8fb2ff216	Calcedonia	5c3ef8170aee697c1ba8432f	2015-07-25 08:46:23	CLOSED	282.72	pk_317b4fc6fd80a5f8fb2ff216	Marruecos	2	

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Read On

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	23:17:46	SELECT `merchant ID`, brand, order_id, created_at, status, amount, merchant_id, country, C...	873 row(s) returned	0.000 sec / 0.000 sec

Creación de la vista.

```
CREATE VIEW orders_view AS (SELECT `merchant ID`, brand, order_id, created_at,
status, amount, merchant_id, country, COUNT(refunded_at) AS total_devoluciones,
SUM(Cantidad) AS valor_devolucion
FROM merchants AS m
INNER JOIN orders AS o
ON m.`merchant ID` = o.merchant_id
LEFT JOIN refunds AS r ON o.order_id = r.`order ID`
GROUP BY `merchant ID`, brand, order_id, created_at, status, amount, merchant_id,
country);
```

70
71 • CREATE VIEW orders_view AS (SELECT `merchant ID`, brand, order_id, created_at, status, amount, merchant_id, country, COUNT(refunded_at) AS total_devolu
72 FROM merchants AS m
73 INNER JOIN orders AS o
74 ON m.`merchant ID` = o.merchant_id
75 LEFT JOIN refunds AS r
76 ON o.order_id = r.`order ID`
77 GROUP BY `merchant ID`, brand, order_id, created_at, status, amount, merchant_id, country);
78

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:28:04	CREATE VIEW orders_view AS (SELECT `merchant ID`, brand, order_id, created_at, status, a...	0 row(s) affected	0.015 sec

EJERCICIO 4

A partir de los datos disponibles diseñar una funcionalidad a tu elección que permita obtener un insight de interés sobre el caso de uso estudiado.

Para ello debes plantear primeramente en un breve texto el objetivo de tu funcionalidad, la queries desarrollada y una reflexión sobre el insight obtenido. Para ello puedes usar cualquier recurso estudiado en clase.

Algunos ejemplos de funcionalidad podría ser: segmentación de clientes en función del valor de las operaciones, sistema de alertas para operaciones delictivas,

identificación de estacionalidad, etc.. Tienes libertad total para desarrollar tu funcionalidad, lo importante es que tenga tu sello personal.

La query que voy a realizar trata de estudiar la propensión de operaciones “DELINQUENT” que hay por tienda y país, de esta forma podríamos tratar de detectar, si solo realizamos la agrupación por marca, que tiendas están más expuestas a estas operaciones, si realizamos la agrupación además de por marca por país, entonces se podrían tomar decisiones en materia de seguridad más precisas, por ser la opción más específica he realizado esta segunda query.

No podemos contabilizar las operaciones “CANCELLED” ya que no son ni un ingreso para la empresa.

```
SELECT brand, country,
SUM(CASE
WHEN status='DELINQUENT' THEN 1
ELSE 0
END)/(SUM(CASE
WHEN status IN('ACTIVE','CLOSED') THEN 1
ELSE 0
END)+SUM(CASE
WHEN status='DELINQUENT' THEN 1
ELSE 0
END)) AS porcentaje
FROM orders AS o
LEFT JOIN merchants AS m
ON o.merchant_id = m.`merchant ID`
GROUP BY brand, country
ORDER BY porcentaje DESC;
```

73 • SELECT brand, country,
 74 SUM(CASE
 75 WHEN status='DELINQUENT' THEN 1
 76 ELSE 0
 77 END)/(SUM(CASE
 78 WHEN status IN('ACTIVE','CLOSED') THEN 1
 79 ELSE 0
 80 END)+SUM(CASE
 81 WHEN status='DELINQUENT' THEN 1

Result Grid

brand	country	porcentaje
Netflix	Alemania	1.0000
Lefties	España	1.0000
Loreal	Francia	0.5000
Carhart	Francia	0.5000
Speedo	Francia	0.4286

Result 25 x

Output

Action Output

#	Time	Action	Message
1	23:34:13	SELECT brand, country, SUM(CASE WHEN status='DELINQUENT' THEN 1 ELSE 0 E...	110 row(s) returned

Según los resultados obtenidos, en algunas tiendas de algunos países existiría un 100% de estas operaciones estudiadas, este resultado no es real y esta funcionalidad, para esta base de datos no es interesante, el motivo por el cual se produce esto es que, en realidad la base de datos es pequeña, tenemos 800 datos distribuidos entre 40 marcas y 7 países, esto arroja un promedio de operaciones por marca y país de dos.

Dos operaciones por marca y país de media es una cantidad demasiado pequeña, por tanto, ante una base de datos más amplia, esta query sería interesante para valorar la seguridad por tienda y ubicación.