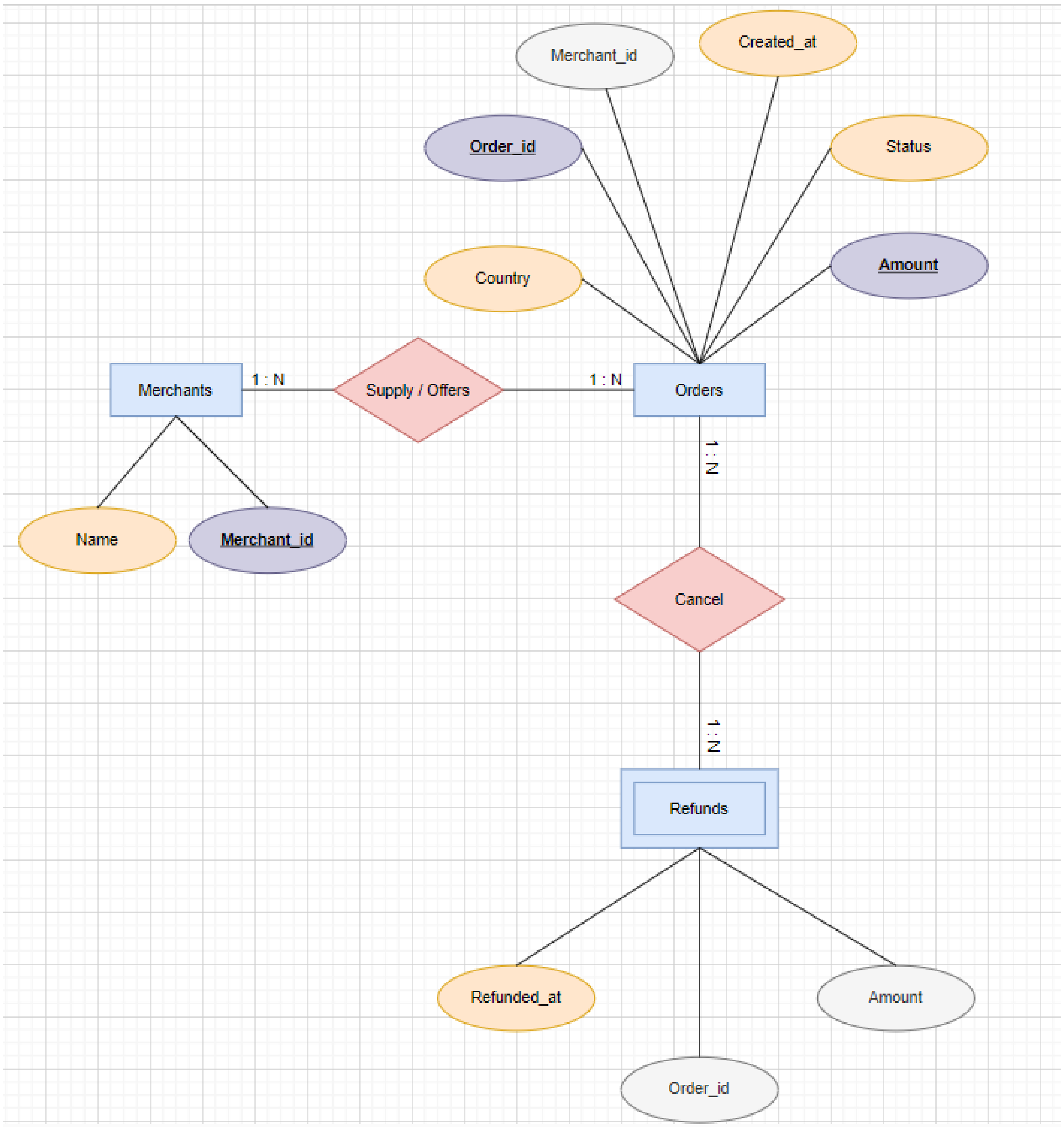
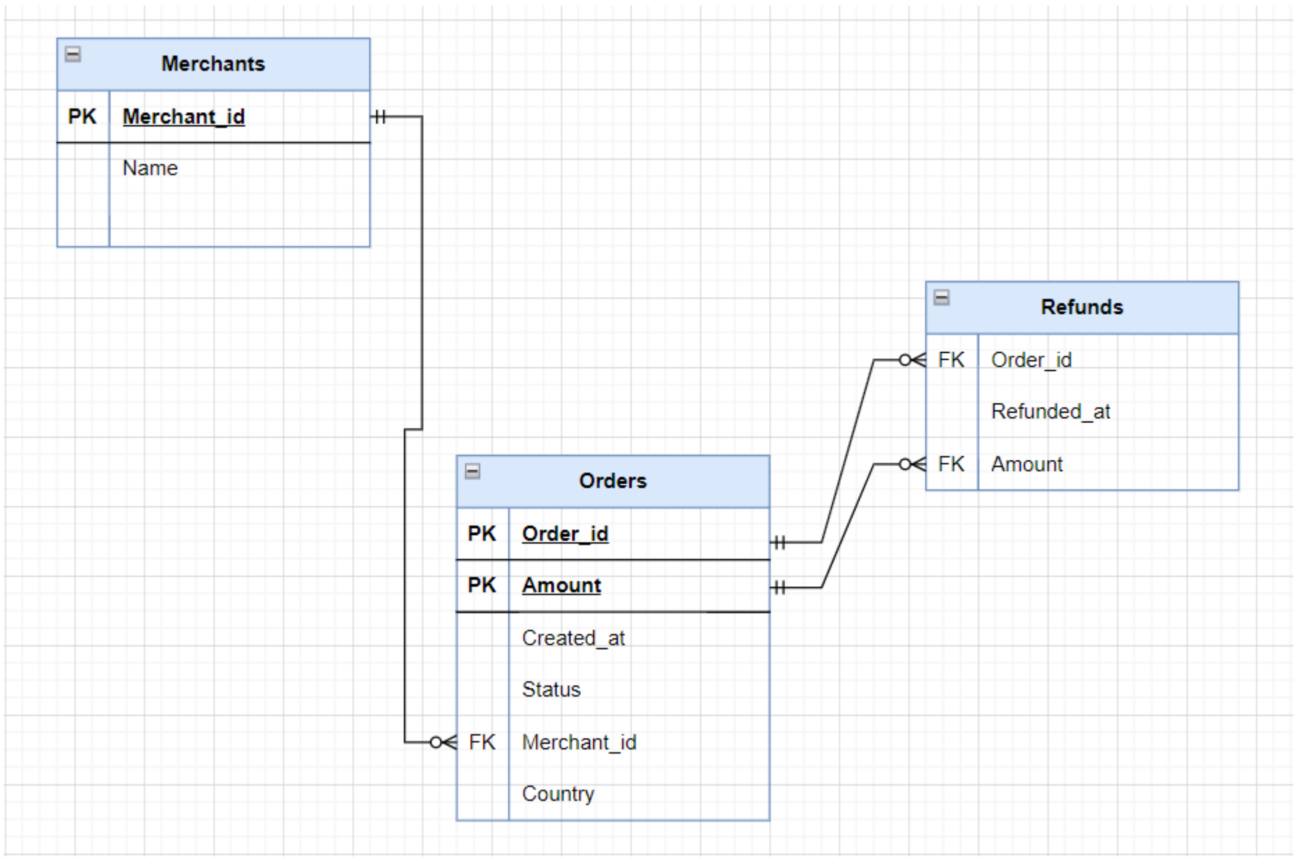
SQL Activity

The first part of the exercise consists of developing an Entity-Relationship Model indicating the entities, attributes, relationships, and cardinality between the tables, the Logical Model, and the Physical Model using DLL.

1.1 – Entity-Relationship Model



1.2 – Logical Model



1.3 – Physical Model

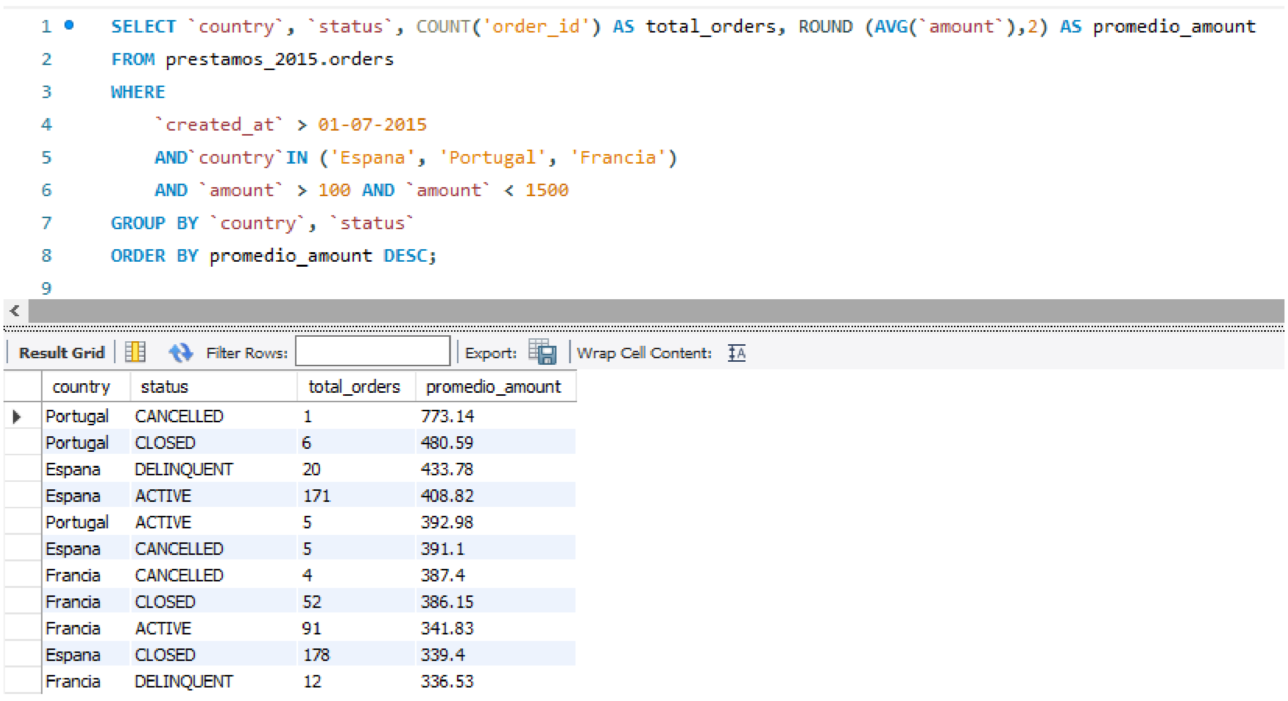
```
USE prestamos_2015;
CREATE TABLE merchants (
  merchant_id VARCHAR(50) NOT NULL,
  name VARCHAR(50) NULL,
  PRIMARY KEY (`merchant_id`));

CREATE TABLE orders (
  order_id VARCHAR(50) NOT NULL,
  created_at DATETIME  NULL,
   status VARCHAR(50)  NULL,
  amount DOUBLE NOT NULL,
  merchant_id VARCHAR(50) NULL,
  country VARCHAR(50) NULL,
  PRIMARY KEY (`order_id`,`amount`));

CREATE TABLE refunds (
  order_id VARCHAR(50) NULL,
  refunded_at DATETIME NULL,
  amount DOUBLE NULL );
```

In the following exercise, based on the tables included in the loans database, the task is to obtain, by country and operation status, the total number of operations and their average amount. However, the operations must be subsequent to 01/07/2015, carried out in France, Portugal, and Spain, and with a value between 100€ and 1500€.

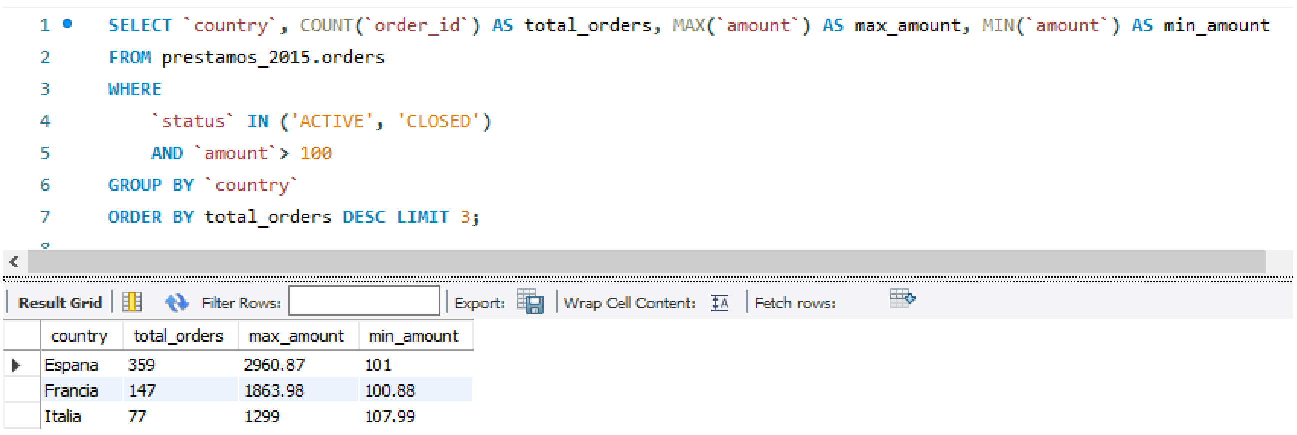
Additionally, the results must be sorted in descending order by the average amount.



```
SELECT `country`, `status`, COUNT('order_id') AS total_orders, ROUND (AVG(`amount`),2) AS promedio_amount
FROM prestamos_2015.orders
WHERE
    `created_at` > 01-07-2015
    AND`country`IN ('Espana', 'Portugal', 'Francia')
    AND `amount` > 100 AND `amount` < 1500
GROUP BY `country`, `status`
ORDER BY promedio_amount DESC;
```

country	status	total_orders	promedio_amount
Portugal	CANCELLED	1	773.14
Portugal	CLOSED	6	480.59
Espana	DELINQUENT	20	433.78
Espana	ACTIVE	171	408.82
Portugal	ACTIVE	5	392.98
Espana	CANCELLED	5	391.1
Francia	CANCELLED	4	387.4
Francia	CLOSED	52	386.15
Francia	ACTIVE	91	341.83
Espana	CLOSED	178	339.4
Francia	DELINQUENT	12	336.53

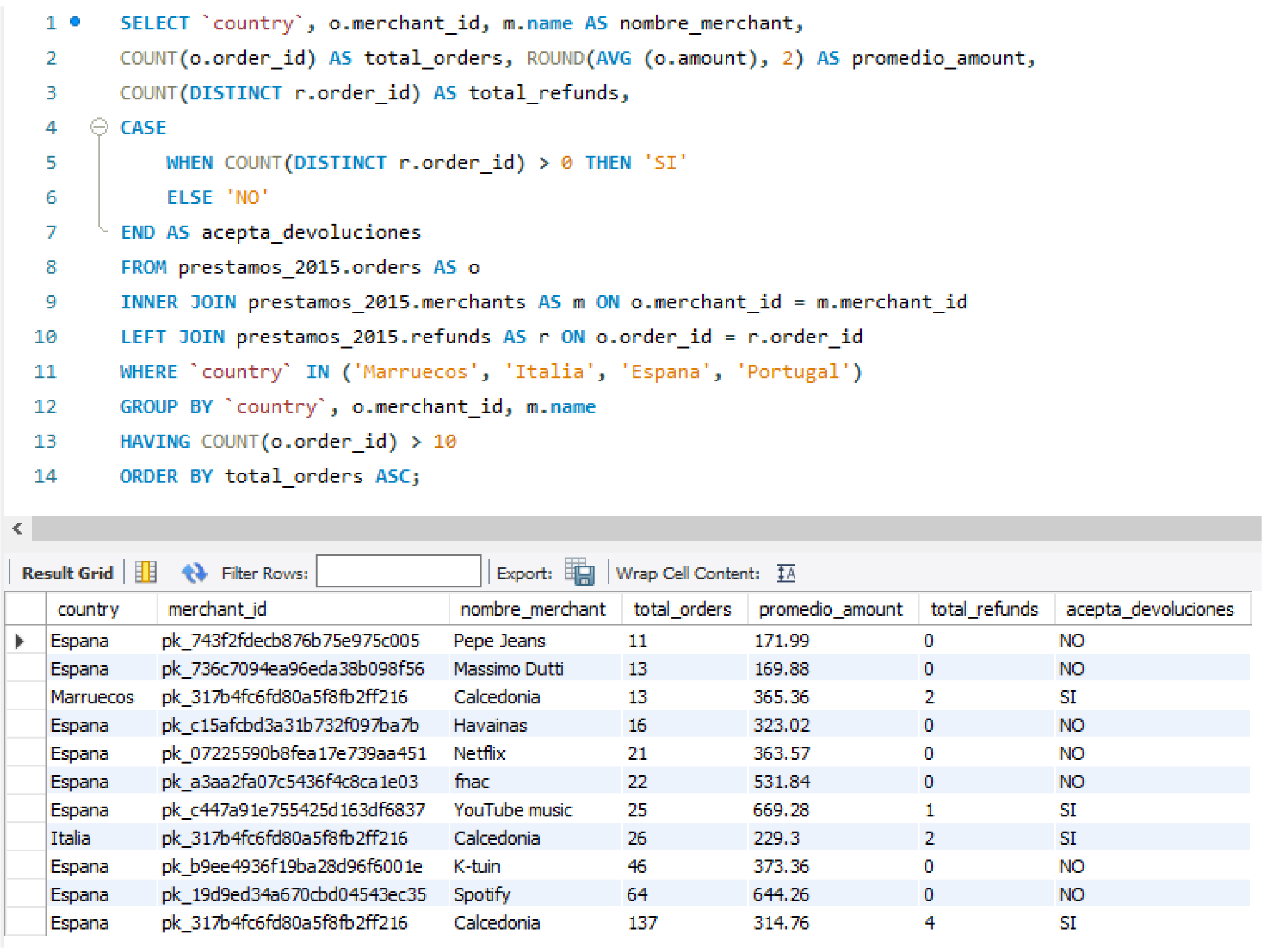
Next, a query is requested to obtain the three countries with the highest number of operations, total number of operations, the operation with the maximum value, and the operation with the minimum value for each country. Additionally, the operations labeled 'Delinquent' and 'Cancelled' must be excluded, and the operations must have a value greater than 100€.



```
SELECT `country`, COUNT(`order_id`) AS total_orders, MAX(`amount`) AS max_amount, MIN(`amount`) AS min_amount
FROM prestamos_2015.orders
WHERE
    `status` IN ('ACTIVE', 'CLOSED')
    AND `amount`> 100
GROUP BY `country`
ORDER BY total_orders DESC LIMIT 3;
```

country	total_orders	max_amount	min_amount
Espana	359	2960.87	101
Francia	147	1863.98	100.88
Italia	77	1299	107.99

In the following query, we must obtain, by country and merchant, the total number of operations, their average value, and the total number of returns. The merchant's name and ID must be displayed. Merchants with more than 10 sales, and merchants from Morocco, Italy, Spain, and Portugal, should be included. Additionally, a field is required to identify whether the merchant accepts returns or not, and the results must be sorted by the total number of operations in ascending order.



```sql
SELECT `country`, o.merchant_id, m.name AS nombre_merchant,
COUNT(o.order_id) AS total_orders, ROUND(AVG (o.amount), 2) AS promedio_amount,
COUNT(DISTINCT r.order_id) AS total_refunds,
CASE
    WHEN COUNT(DISTINCT r.order_id) > 0 THEN 'SI'
    ELSE 'NO'
END AS acepta_devoluciones
FROM prestamos_2015.orders AS o
INNER JOIN prestamos_2015.merchants AS m ON o.merchant_id = m.merchant_id
LEFT JOIN prestamos_2015.refunds AS r ON o.order_id = r.order_id
WHERE `country` IN ('Marruecos', 'Italia', 'Espana', 'Portugal')
GROUP BY `country`, o.merchant_id, m.name
HAVING COUNT(o.order_id) > 10
ORDER BY total_orders ASC;
```

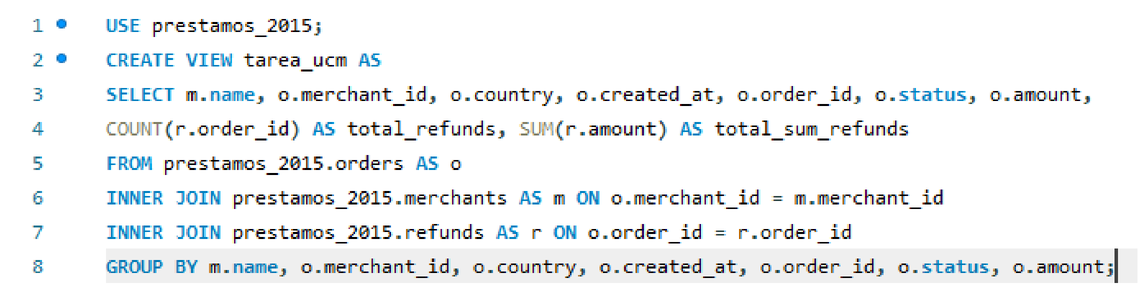
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

country	merchant_id	nombre_merchant	total_orders	promedio_amount	total_refunds	acepta_devoluciones
Espana	pk_743f2fdecb876b75e975c005	Pepe Jeans	11	171.99	0	NO
Espana	pk_736c7094ea96eda38b098f56	Massimo Dutti	13	169.88	0	NO
Marruecos	pk_317b4fc6fd80a5f8fb2ff216	Calcedonia	13	365.36	2	SI
Espana	pk_c15afcbd3a31b732f097ba7b	Havainas	16	323.02	0	NO
Espana	pk_07225590b8fea17e739aa451	Netflix	21	363.57	0	NO
Espana	pk_a3aa2fa07c5436f4c8ca1e03	fnac	22	531.84	0	NO
Espana	pk_c447a91e755425d163df6837	YouTube music	25	669.28	1	SI
Italia	pk_317b4fc6fd80a5f8fb2ff216	Calcedonia	26	229.3	2	SI
Espana	pk_b9ee4936f19ba28d96f6001e	K-tuin	46	373.36	0	NO
Espana	pk_19d9ed34a670cbd04543ec35	Spotify	64	644.26	0	NO
Espana	pk_317b4fc6fd80a5f8fb2ff216	Calcedonia	137	314.76	4	SI

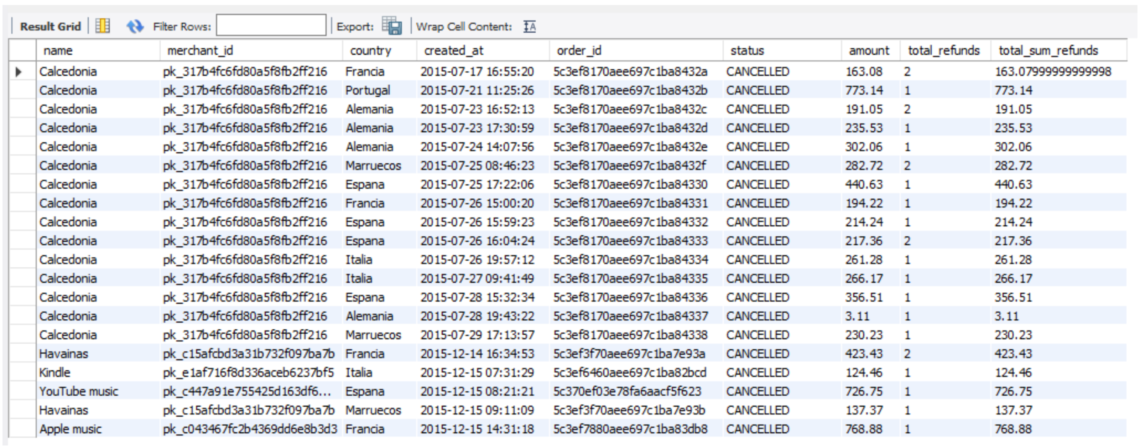
We perform a query where all fields from the operations and merchants tables will be included. From the returns table, we request the count of returns per operation and the sum of the return values. We create a view named 'orders_view' within the 'tarea_ucm' schema with this query.

If we understand that the query only asks for all fields, but only for the status of CANCELED, that is, those with returns:

<u>Case 1:</u>



```
USE prestamos_2015;
CREATE VIEW tarea_ucm AS
SELECT m.name, o.merchant_id, o.country, o.created_at, o.order_id, o.status, o.amount,
COUNT(r.order_id) AS total_refunds, SUM(r.amount) AS total_sum_refunds
FROM prestamos_2015.orders AS o
INNER JOIN prestamos_2015.merchants AS m ON o.merchant_id = m.merchant_id
INNER JOIN prestamos_2015.refunds AS r ON o.order_id = r.order_id
GROUP BY m.name, o.merchant_id, o.country, o.created_at, o.order_id, o.status, o.amount;
```

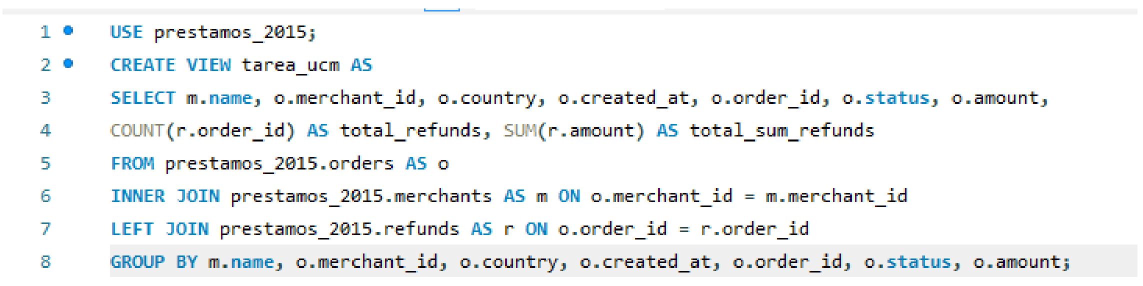


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

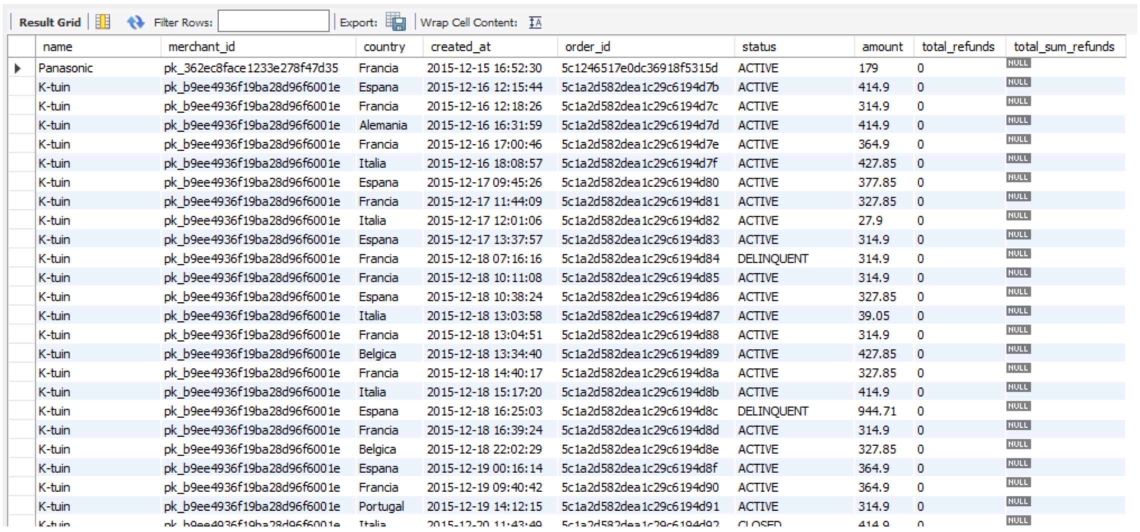
name	merchant_id	country	created_at	order_id	status	amount	total_refunds	total_sum_refunds
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Francia	2015-07-17 16:55:20	5c3ef8170aee697c1ba8432a	CANCELLED	163.08	2	163.07999999999998
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Portugal	2015-07-21 11:25:26	5c3ef8170aee697c1ba8432b	CANCELLED	773.14	1	773.14
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Alemania	2015-07-23 16:52:13	5c3ef8170aee697c1ba8432c	CANCELLED	191.05	2	191.05
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Alemania	2015-07-23 17:30:59	5c3ef8170aee697c1ba8432d	CANCELLED	235.53	1	235.53
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Alemania	2015-07-24 14:07:56	5c3ef8170aee697c1ba8432e	CANCELLED	302.06	1	302.06
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Marruecos	2015-07-25 08:46:23	5c3ef8170aee697c1ba8432f	CANCELLED	282.72	2	282.72
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Espana	2015-07-25 17:22:06	5c3ef8170aee697c1ba84330	CANCELLED	440.63	1	440.63
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Francia	2015-07-26 15:00:20	5c3ef8170aee697c1ba84331	CANCELLED	194.22	1	194.22
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Espana	2015-07-26 15:59:23	5c3ef8170aee697c1ba84332	CANCELLED	214.24	1	214.24
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Espana	2015-07-26 16:04:24	5c3ef8170aee697c1ba84333	CANCELLED	217.36	2	217.36
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Italia	2015-07-26 19:57:12	5c3ef8170aee697c1ba84334	CANCELLED	261.28	1	261.28
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Italia	2015-07-27 09:41:49	5c3ef8170aee697c1ba84335	CANCELLED	266.17	1	266.17
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Espana	2015-07-28 15:32:34	5c3ef8170aee697c1ba84336	CANCELLED	356.51	1	356.51
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Alemania	2015-07-28 19:43:22	5c3ef8170aee697c1ba84337	CANCELLED	3.11	1	3.11
Calcedonia	pk_317b4fc6fd80a5f8fb2ff216	Marruecos	2015-07-29 17:13:57	5c3ef8170aee697c1ba84338	CANCELLED	230.23	1	230.23
Havainas	pk_c15afcbd3a31b732f097ba7b	Francia	2015-12-14 16:34:53	5c3ef3f70aee697c1ba7e93a	CANCELLED	423.43	2	423.43
Kindle	pk_e1af716f8d336aceb6237bf5	Italia	2015-12-15 07:31:29	5c3ef6460aee697c1ba82bcd	CANCELLED	124.46	1	124.46
YouTube music	pk_c447a91e755425d163df6...	Espana	2015-12-15 08:21:21	5c370ef03e78fa6aacf5f623	CANCELLED	726.75	1	726.75
Havainas	pk_c15afcbd3a31b732f097ba7b	Marruecos	2015-12-15 09:11:09	5c3ef3f70aee697c1ba7e93b	CANCELLED	137.37	1	137.37
Apple music	pk_c043467fc2b4369dd6e8b3d3	Francia	2015-12-15 14:31:18	5c3ef7880aee697c1ba83db8	CANCELLED	768.88	1	768.88

But if we understand that the query asks for all fields, including all statuses, but when there is a return, it should provide the total number and the sum of refunds, then we write the following code. It should be noted that those without returns will appear as "null":

<u>Case 2:</u>



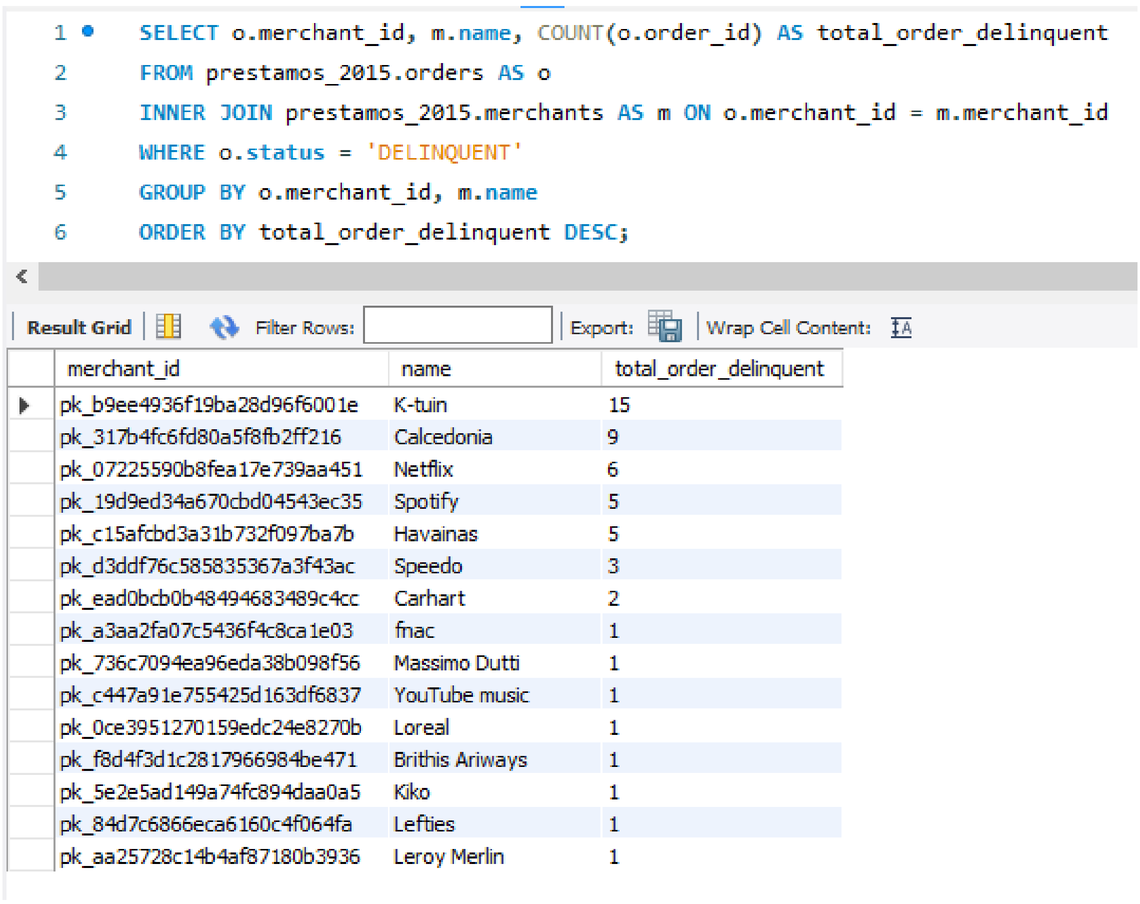
```
USE prestamos_2015;
CREATE VIEW tarea_ucm AS
SELECT m.name, o.merchant_id, o.country, o.created_at, o.order_id, o.status, o.amount,
COUNT(r.order_id) AS total_refunds, SUM(r.amount) AS total_sum_refunds
FROM prestamos_2015.orders AS o
INNER JOIN prestamos_2015.merchants AS m ON o.merchant_id = m.merchant_id
LEFT JOIN prestamos_2015.refunds AS r ON o.order_id = r.order_id
GROUP BY m.name, o.merchant_id, o.country, o.created_at, o.order_id, o.status, o.amount;
```



name	merchant_id	country	created_at	order_id	status	amount	total_refunds	total_sum_refunds
Panasonic	pk_362ec8face1233e278f47d35	Francia	2015-12-15 16:52:30	5c1246517e0dc36918f5315d	ACTIVE	179	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Espana	2015-12-16 12:15:44	5c1a2d582dea1c29c6194d7b	ACTIVE	414.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Francia	2015-12-16 12:18:26	5c1a2d582dea1c29c6194d7c	ACTIVE	314.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Alemania	2015-12-16 16:31:59	5c1a2d582dea1c29c6194d7d	ACTIVE	414.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Francia	2015-12-16 17:00:46	5c1a2d582dea1c29c6194d7e	ACTIVE	364.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Italia	2015-12-16 18:08:57	5c1a2d582dea1c29c6194d7f	ACTIVE	427.85	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Espana	2015-12-17 09:45:26	5c1a2d582dea1c29c6194d80	ACTIVE	377.85	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Francia	2015-12-17 11:44:09	5c1a2d582dea1c29c6194d81	ACTIVE	327.85	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Italia	2015-12-17 12:01:06	5c1a2d582dea1c29c6194d82	ACTIVE	27.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Espana	2015-12-17 13:37:57	5c1a2d582dea1c29c6194d83	ACTIVE	314.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Francia	2015-12-18 07:16:16	5c1a2d582dea1c29c6194d84	DELINQUENT	314.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Francia	2015-12-18 10:11:08	5c1a2d582dea1c29c6194d85	ACTIVE	314.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Espana	2015-12-18 10:38:24	5c1a2d582dea1c29c6194d86	ACTIVE	327.85	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Italia	2015-12-18 13:03:58	5c1a2d582dea1c29c6194d87	ACTIVE	39.05	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Francia	2015-12-18 13:04:51	5c1a2d582dea1c29c6194d88	ACTIVE	314.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Belgica	2015-12-18 13:34:40	5c1a2d582dea1c29c6194d89	ACTIVE	427.85	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Francia	2015-12-18 14:40:17	5c1a2d582dea1c29c6194d8a	ACTIVE	327.85	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Italia	2015-12-18 15:17:20	5c1a2d582dea1c29c6194d8b	ACTIVE	414.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Espana	2015-12-18 16:25:03	5c1a2d582dea1c29c6194d8c	DELINQUENT	944.71	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Francia	2015-12-18 16:39:24	5c1a2d582dea1c29c6194d8d	ACTIVE	314.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Belgica	2015-12-18 22:02:29	5c1a2d582dea1c29c6194d8e	ACTIVE	327.85	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Espana	2015-12-19 00:16:14	5c1a2d582dea1c29c6194d8f	ACTIVE	364.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Francia	2015-12-19 09:40:42	5c1a2d582dea1c29c6194d90	ACTIVE	364.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Portugal	2015-12-19 14:12:15	5c1a2d582dea1c29c6194d91	ACTIVE	314.9	0	NULL
K-tuin	pk_b9ee4936f19ba28d96f6001e	Italia	2015-12-20 11:43:49	5c1a2d582dea1c29c6194d92	CLOSED	414.9	0	NULL

From the data, a functionality needs to be created to obtain an insight. A functionality is proposed to identify any existing cause or pattern leading to non-payments, that is, 'Delinquent' customers.

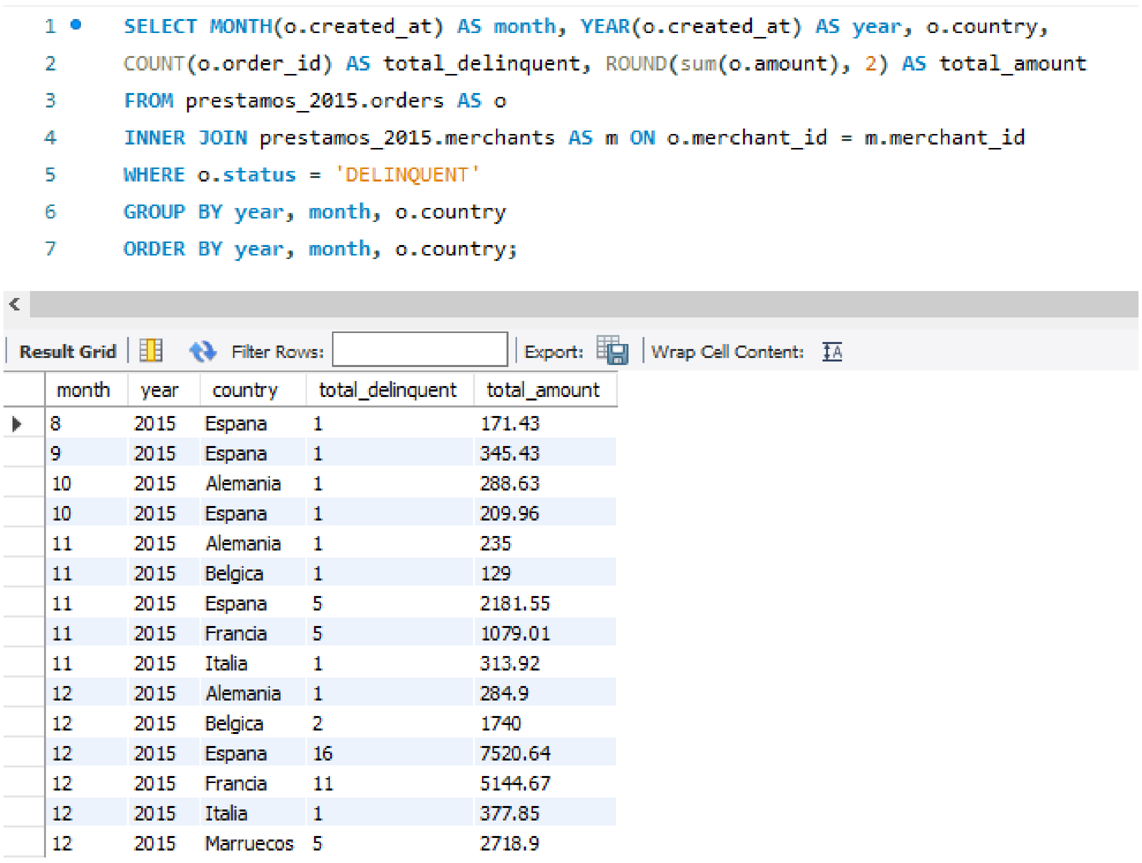
An exploratory analysis is initiated on merchants with the highest number of unpaid orders/purchases.



```sql
SELECT o.merchant_id, m.name, COUNT(o.order_id) AS total_order_delinquent
FROM prestamos_2015.orders AS o
INNER JOIN prestamos_2015.merchants AS m ON o.merchant_id = m.merchant_id
WHERE o.status = 'DELINQUENT'
GROUP BY o.merchant_id, m.name
ORDER BY total_order_delinquent DESC;
```

merchant_id	name	total_order_delinquent
pk_b9ee4936f19ba28d96f6001e	K-tuin	15
pk_317b4fc6fd80a5f8fb2ff216	Calcedonia	9
pk_07225590b8fea17e739aa451	Netflix	6
pk_19d9ed34a670cbd04543ec35	Spotify	5
pk_c15afcbd3a31b732f097ba7b	Havainas	5
pk_d3ddf76c585835367a3f43ac	Speedo	3
pk_ead0bcb0b48494683489c4cc	Carhart	2
pk_a3aa2fa07c5436f4c8ca1e03	fnac	1
pk_736c7094ea96eda38b098f56	Massimo Dutti	1
pk_c447a91e755425d163df6837	YouTube music	1
pk_0ce3951270159edc24e8270b	Loreal	1
pk_f8d4f3d1c2817966984be471	Brithis Ariways	1
pk_5e2e5ad149a74fc894daa0a5	Kiko	1
pk_84d7c6866eca6160c4f064fa	Lefties	1
pk_aa25728c14b4af87180b3936	Leroy Merlin	1

From the extracted data, we generate a functionality that informs us about the time of the year with the highest number of non-payments and the countries involved.



```sql
SELECT MONTH(o.created_at) AS month, YEAR(o.created_at) AS year, o.country,
COUNT(o.order_id) AS total_delinquent, ROUND(sum(o.amount), 2) AS total_amount
FROM prestamos_2015.orders AS o
INNER JOIN prestamos_2015.merchants AS m ON o.merchant_id = m.merchant_id
WHERE o.status = 'DELINQUENT'
GROUP BY year, month, o.country
ORDER BY year, month, o.country;
```

month	year	country	total_delinquent	total_amount
8	2015	Espana	1	171.43
9	2015	Espana	1	345.43
10	2015	Alemania	1	288.63
10	2015	Espana	1	209.96
11	2015	Alemania	1	235
11	2015	Belgica	1	129
11	2015	Espana	5	2181.55
11	2015	Francia	5	1079.01
11	2015	Italia	1	313.92
12	2015	Alemania	1	284.9
12	2015	Belgica	2	1740
12	2015	Espana	16	7520.64
12	2015	Francia	11	5144.67
12	2015	Italia	1	377.85
12	2015	Marruecos	5	2718.9