

N1E1

Primero permitiremos la subida de archivos de datos a nuestra base de datos:

```
40 • SET GLOBAL local_infile = TRUE;
41 • SHOW GLOBAL VARIABLES LIKE 'local_infile';
```

42

Variable_name	Value
local_infile	ON

result 4 x

Output

Action Output

#	Time	Action	Message
1	11:12:26	SET GLOBAL local_infile = TRUE	0 row(s) affected
2	11:12:28	SHOW GLOBAL VARIABLES LIKE 'local_infile'	1 row(s) returned

Una vez hecho, procederemos a la creación de las tablas e inserción de los datos:
(IMPORTANTE: PARA ESTA PARTE DE LA CREACIÓN DE TABLAS HAY QUE LEER EN EL ARCHIVO SQL. AÑADÍ LAS PRIMARY KEY A POSTERIORI YA QUE SE ME OLVIDÓ A LA HORA DE TOMAR LAS CAPTURAS)

```
21 • CREATE TABLE company (
22     company_id VARCHAR(20),
23     company_name VARCHAR(255),
24     phone VARCHAR(15),
25     email VARCHAR(100),
26     country VARCHAR(100),
27     website VARCHAR(255)
28 );
29 • LOAD DATA INFILE
30 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv'
31 INTO TABLE company
32 FIELDS TERMINATED BY ','
33 LINES TERMINATED BY '\n'
34 IGNORE 1 ROWS
35 (company id, company name, phone, email, country, website);
```

Output

Action Output

#	Time	Action	Message
1	11:56:39	CREATE TABLE company (company_id VARCHAR...	0 row(s) affected
2	11:56:43	LOAD DATA INFILE 'C:/ProgramData/MySQL/MyS...	100 row(s) affected Records: 100 Deleted: 0 Skipp...

```
• CREATE TABLE credit_card (
    id VARCHAR(20),
    user_id INT,
    iban VARCHAR(50),
    pan VARCHAR(50),
    pin VARCHAR(4),
    cvv INT,
    track1 VARCHAR(255),
    track2 VARCHAR(255),
    expiring_date VARCHAR(20)
);
• LOAD DATA INFILE
'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv'
INTO TABLE credit_card
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(id, user id, iban, pan, pin, cvv, track1, track2, expiring date);
```

Action Output

Time	Action	Message
1 12:04:22	CREATE TABLE credit_card (id VARCHAR(20), ...	0 row(s) affected
2 12:04:36	LOAD DATA INFILE 'C:/ProgramData/MySQL/MyS...	275 row(s) affected Records: 275 Deleted: 0 Skipp...

```
• ALTER TABLE credit_card
DROP user_id;
```

Action Output

Time	Action	Message
1 13:33:35	ALTER TABLE credit_card DROP user_id	

```

69 • CREATE TABLE data_users (
70     id INT,
71     name VARCHAR(100),
72     surname VARCHAR(100),
73     phone VARCHAR(150),
74     email VARCHAR(150),
75     birth_date VARCHAR(100),
76     country VARCHAR(150),
77     city VARCHAR(150),
78     postal_code VARCHAR(100),
79     address VARCHAR(255)
80 );
81 • LOAD DATA INFILE
82 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv'
83 INTO TABLE data_users
84 FIELDS TERMINATED BY ','
85 ENCLOSED BY '"'
86 LINES TERMINATED BY '\n'
87 IGNORE 1 ROWS
88 (id, name, surname, phone, email, birth_date, country, city, postal_code, address);
89 • LOAD DATA INFILE
90 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv'

```

```

• CREATE TABLE products (
    id INT,
    product_name VARCHAR(100),
    currency VARCHAR(1),
    price VARCHAR(100),
    colour VARCHAR(10),
    weight FLOAT,
    warehouse_id VARCHAR(100)
);
• LOAD DATA INFILE
'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv'
INTO TABLE products
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(id, product_name, price, colour, weight, warehouse_id);
• UPDATE products
SET currency = LEFT(price, 1);
• UPDATE products
SET price = substr(price, 2);
• ALTER TABLE products
MODIFY price DECIMAL(10,2);

```

#	Time	Action	Message
1	12:29:56	CREATE TABLE data_users (id INT, name VA...	0 row(s) affected
2	12:29:58	LOAD DATA INFILE 'C:/ProgramData/MySQL/MyS...	75 row(s) affected Records: 75 Deleted: 0 Skipped...
3	12:30:01	LOAD DATA INFILE 'C:/ProgramData/MySQL/MyS...	50 row(s) affected Records: 50 Deleted: 0 Skipped...
4	12:30:03	LOAD DATA INFILE 'C:/ProgramData/MySQL/MyS...	150 row(s) affected Records: 150 Deleted: 0 Skipp...

#	Time	Action	Message
1	13:19:58	CREATE TABLE products (id INT, product_na...	0 row(s) affected
2	13:20:01	LOAD DATA INFILE 'C:/ProgramData/MySQL/My...	100 row(s) affected Records
3	13:20:04	UPDATE products SET currency = LEFT(price, 1)	100 row(s) affected Rows m
4	13:20:06	UPDATE products SET price = substr(price, 2)	100 row(s) affected Rows m
5	13:20:09	ALTER TABLE products MODIFY price DECIMAL(1...	100 row(s) affected Records

```

127 • LOAD DATA INFILE
128 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'
129 INTO TABLE transaction
130 FIELDS TERMINATED BY ','
131 ENCLOSED BY '"'
132 LINES TERMINATED BY '\n'
133 IGNORE 1 ROWS
134 (id, card_id, business_id, timestamp, amount, declined, product_ids, user_id, lat, longitude);
135 • CREATE TABLE transaction_product (
136     transaction_id VARCHAR(255),
137     product_id INT
138 );
139 • INSERT INTO transaction_product (transaction_id, product_id)
140 WITH RECURSIVE SplitValues AS (
141     SELECT id, SUBSTRING_INDEX(product_ids, ',', 1) AS split_value, IF(LOCATE(',', product
142     FROM transaction
143     UNION ALL
144     SELECT id, SUBSTRING_INDEX(remaining_values, ',', 1) AS split_value, IF(LOCATE(',', re
145     FROM SplitValues
146     WHERE remaining_values IS NOT NULL)
147     SELECT id, split_value
148     FROM SplitValues;
149 • ALTER TABLE transaction
150 DROP product_ids;

```

#	Time	Action	Message	Duration / Fetch
1	11:51:01	CREATE TABLE transaction (id VARCHAR(255)...	0 row(s) affected	0.031 sec
2	11:51:04	LOAD DATA INFILE 'C:/ProgramData/MySQL/My...	587 row(s) affected Records: 587 Deleted: 0 Skipp...	0.047 sec
3	11:51:07	CREATE TABLE transaction_product (transaction...	0 row(s) affected	0.047 sec
4	11:51:10	INSERT INTO transaction_product (transaction_id, p...	1457 row(s) affected Records: 1457 Duplicates: 0 ...	0.031 sec
5	11:51:13	ALTER TABLE transaction DROP product_ids	0 row(s) affected Records: 0 Duplicates: 0 Warning...	0.031 sec

Con todas las tablas creadas, ahora crearemos las relaciones entre ellas.

```

• ALTER TABLE transaction
ADD CONSTRAINT fk_card_id
FOREIGN KEY (card_id) REFERENCES credit_card(id),
ADD CONSTRAINT fk_business_id
FOREIGN KEY (business_id) REFERENCES company(company_id),
ADD CONSTRAINT fk_user_id
FOREIGN KEY (user_id) REFERENCES data_users(id);

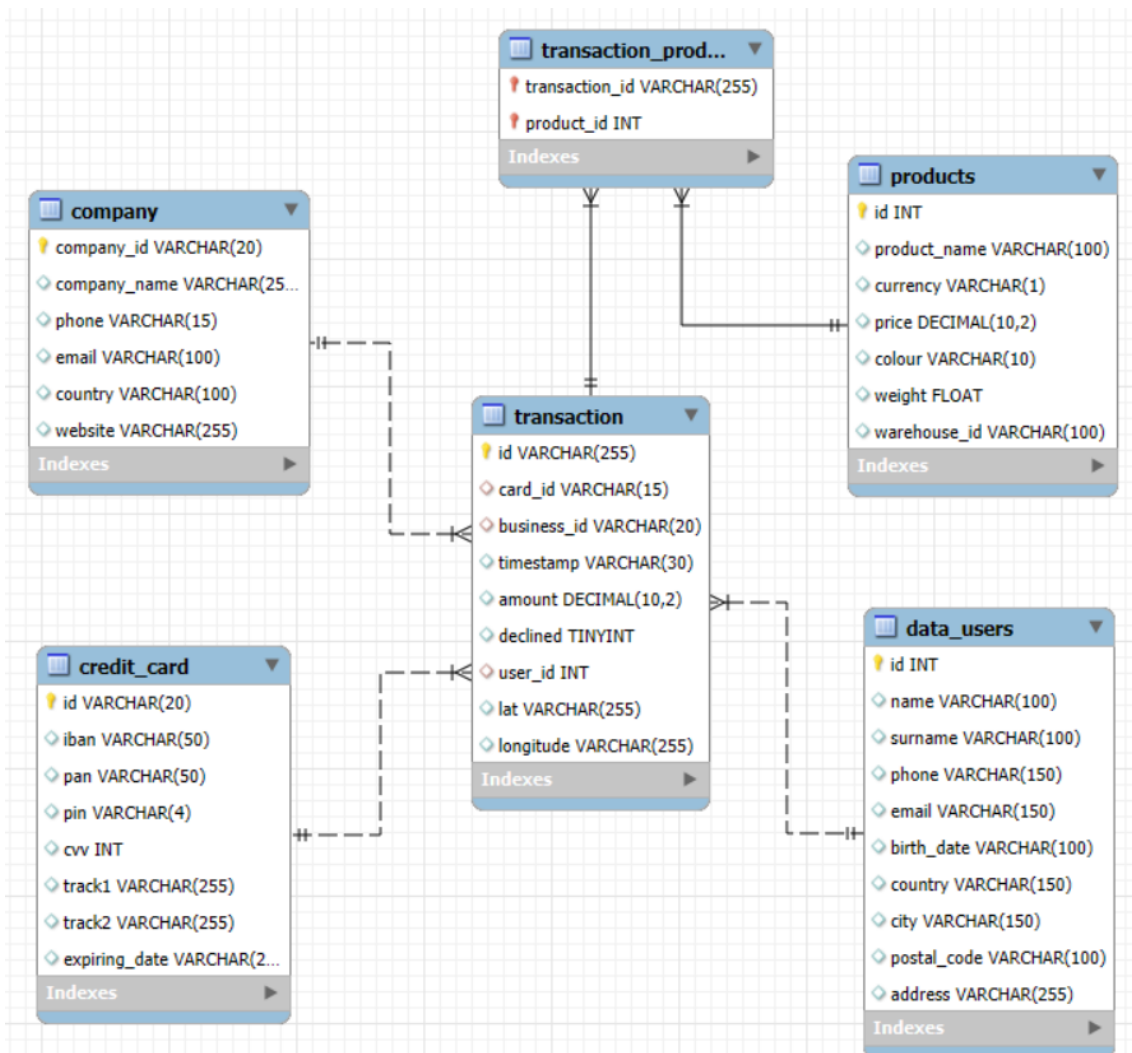
```

#	Time	Action	Message
1	13:16:40	ALTER TABLE transaction ADD CONSTRAINT fk_c...	587 row(s) affected Records: 587 Duplicates: 0 Wa...

- ```
ALTER TABLE transaction_product
ADD CONSTRAINT fk_transaction_id
FOREIGN KEY (transaction_id) REFERENCES transaction(id),
ADD CONSTRAINT fk_product_id
FOREIGN KEY (product_id) REFERENCES products(id);
```

| Action Output |                                                |  | Message              |
|---------------|------------------------------------------------|--|----------------------|
| Time          | Action                                         |  |                      |
| 1 13:22:41    | ALTER TABLE transaction_product ADD CONSTRA... |  | 1457 row(s) affected |

Al final el esquema quedará parecido a esto:



N1E1

```
175 • SELECT data_users.id as User, count(transaction.id) as NumTrans
176 FROM data_users
177 JOIN transaction
178 ON (data_users.id = user_id)
179 GROUP BY data_users.id
180 HAVING count(transaction.id) > 30
181 ORDER BY NumTrans desc;
```

Result Grid

| User | NumTrans |
|------|----------|
| 272  | 76       |
| 267  | 52       |
| 275  | 48       |
| 92   | 39       |

Result 13

Output

Action Output

| # | Time     | Action                                                                               |
|---|----------|--------------------------------------------------------------------------------------|
| 1 | 13:46:41 | SELECT data_users.id as User, count(transaction.id) as NumTrans FROM data_users JOIN |

N1E2

```
-- Mostra la mitjana d'amount per IBAN de les t
• SELECT iban, round(avg(amount), 2) as Average
FROM credit_card
JOIN transaction
ON (credit_card.id = card_id)
JOIN company
ON (company_id = business_id)
WHERE company_name = "Donec Ltd"
GROUP BY iban;
```

Result Grid

| iban                      | Average |
|---------------------------|---------|
| PT87806228135092429456346 | 203.72  |

Result 19

Output

Action Output

| # | Time     | Action                                                 |
|---|----------|--------------------------------------------------------|
| 1 | 13:53:41 | SELECT iban, round(avg(amount), 2) as Average FROM cre |

## N2

Primero cambiamos los datos de timestamp para que se puedan ordenar como fecha, i después creamos la tabla nueva active\_cards, que se relaciona 1:1 con credit\_card.

```
• UPDATE transaction
 SET timestamp = str_to_date(timestamp, '%d/%m/%Y %H:%i');

ut :-----
Action Output
Time Action Message
1 16:43:26 UPDATE transaction SET timestamp = str_to_date(timestamp, '%d/%m/%Y %H:%i') 587 row(s) affected

2 • UPDATE transaction
3 SET timestamp = str_to_date(timestamp, '%d/%m/%Y %H:%i');
4
5 • ALTER TABLE transaction
6 MODIFY timestamp TIMESTAMP;

input :-----
Action Output
Time Action Message
1 16:43:26 UPDATE transaction SET timestamp = str_to_date(timestamp, '%d/%m/%Y %H:%i') 587 row(s) affected
2 16:44:57 ALTER TABLE transaction MODIFY timestamp TIMESTAMP 587 row(s) affected

9 • CREATE TABLE active_cards (
10 card_id VARCHAR(15) PRIMARY KEY,
11 active TINYINT
12);
13
14 • INSERT INTO active_cards (card_id, active)
15 SELECT card_id, IF(sum(declined) > 0, 0, 1) as Active
16 FROM (SELECT card_id, timestamp, declined
17 FROM (SELECT *, row_number() over (partition by card_id order by card_id, timestamp desc) as seqnum
18 from transaction) as a
19 WHERE seqnum <= 3
20 order by card_id, timestamp desc, seqnum) as b
21 GROUP BY card_id;
22
23 • ALTER TABLE active_cards
24 ADD CONSTRAINT fk_active_card
25 FOREIGN KEY (card_id) REFERENCES credit_card(id);
26

input :-----
Action Output
Time Action Message
1 17:56:48 CREATE TABLE active_cards (card_id VARCHAR(15) PRIMARY KEY, active TINYINT) 0 row(s) affected
2 17:56:52 INSERT INTO active_cards (card_id, active) SELECT card_id, IF(sum(declined) > 0, 0, 1) as Active FROM (SELE... 275 row(s) affected Record
3 18:02:34 ALTER TABLE active_cards ADD CONSTRAINT fk_active_card FOREIGN KEY (card_id) REFERENCES credit... 275 row(s) affected Record
```

## N2E1

En este hay dos formas de hacerlo según tengo la tabla creada, siendo o bien sumando o bien contando.

```
55 • SELECT count(card_id) as ActiveCards
56 FROM active_cards
57 WHERE active = 1;
58
59 • SELECT sum(active) as ActiveCards
60 FROM active_cards;
```


Result Grid  Filter Rows:  Export:  Wrap Cell Content:

| ActiveCards |
|-------------|
|-------------|

|     |
|-----|
| 191 |
|-----|

Result 34 x

Output

 Action Output

| # | Time     | Action                                                             |
|---|----------|--------------------------------------------------------------------|
| 1 | 18:23:44 | SELECT sum(active) as ActiveCards FROM active_cards LIMIT 0, 50000 |

N3E1

(La tabla ya ha sido creada en el N1)

3

•

```
SELECT products.id, count(transaction.id) as NumSales
FROM products
JOIN transaction_product AS tp
ON (products.id = tp.product_id)
JOIN transaction
ON (tp.transaction_id = transaction.id)
GROUP BY products.id
ORDER BY products.id;
```

result Grid

Filter Rows:

Export:

Wrap Cell Content:

| id | NumSales |
|----|----------|
| 1  | 61       |
| 2  | 65       |
| 3  | 51       |
| 5  | 49       |
| 7  | 54       |
| 11 | 48       |
| 13 | 60       |
| 17 | 61       |

result 4

×

output

Action Output

▼

| # | Time     | Action                                                                  |
|---|----------|-------------------------------------------------------------------------|
| 1 | 19:02:41 | SELECT products.id, count(transaction.id) as NumSales FROM products JOI |