

# SPRINT 4

Lucía Trinidad Fernández Gagliano

29/01/2025

Corregido por:

---

## NIVELL 1

**PARA PODER REALIZAR LOS EJERCICIOS DEL NIVEL 1:**

-- > primero creamos la base de datos

```
12 -----
13 -- CREAMOS BASE DE DATOS
14 -----
15 • CREATE database transtrans;
16 • USE transtrans;
```

-- > luego creamos las tablas de dimensiones *users*, *credit\_card*, *company*

```
18 -----
19 -- CREAMOS LAS TABLAS DE 'DIMENSIONES'
20 -----
21 • DROP TABLE IF exists users;
22 • CREATE TABLE IF NOT EXISTS users (
23     id INT NOT NULL PRIMARY KEY,
24     name VARCHAR (50),
25     surname VARCHAR (100),
26     phone VARCHAR (50),
27     email VARCHAR (100),
28     birth_date VARCHAR (30),
29     country VARCHAR (100),
30     city VARCHAR (100),
31     postal_code VARCHAR (20),
32     address VARCHAR (100)
33 );
```

```
35
36 • DROP TABLE IF exists companies;
37 • CREATE TABLE IF NOT EXISTS companies (
38     company_id VARCHAR(10) PRIMARY KEY,
39     company_name VARCHAR (100),
40     phone VARCHAR (30),
41     email VARCHAR (100),
42     country VARCHAR (100),
43     website VARCHAR (100)
44 );
```

```

46 • DROP TABLE IF EXISTS credit_cards;
47 • CREATE TABLE IF NOT EXISTS credit_cards (
48     id VARCHAR (10) PRIMARY KEY,
49     user_id VARCHAR (10),
50     iban VARCHAR (50),
51     pan VARCHAR (50),
52     pin VARCHAR(4),
53     cvv VARCHAR(3),
54     track1 VARCHAR (100),
55     track2 VARCHAR (100),
56     expiring_date VARCHAR (20)
57 );

```

-- > configuramos MySQL Workbench para que permita la carga de archivos CSV desde la carpeta indicada

```

63 • SHOW VARIABLES LIKE 'secure_file_priv'; -- > para ver dónde se alojan los archivos que se cargan
64 -- > 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\'
65 -- > subo las tablas de datos a utilizar a la carpeta Uploads
66
67 #EJECUTO:
68 • SET GLOBAL local_infile=on;
69 • SHOW GLOBAL VARIABLES LIKE 'local_infile'; #ON
70
71 # INTEGRO LA SIGUIENTE ORDEN EN EL Manage Conections - Advanced . Test Conection:
72 # OPT_LOCAL_INFILE=1
73
74 # ME SIGUE DANDO ERROR
75 # Error Code: 2068. LOAD DATA LOCAL INFILE file request rejected due to restrictions on access.
76 -----
77 ### ¡¡¡ PRUEBO QUITANDO EL 'LOCAL' Y AHORA SÍ ME CARGA LOS DATOS !!! (emoción)
78 -----

```

-- > Cargamos lo datos de cada tabla de dimensiones:

```

79 -- > CARGAMOS DATOS DE LA TABLA USERS
80
81 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv"
82 INTO TABLE users
83 FIELDS TERMINATED BY ','
84 ENCLOSED BY '"'
85 LINES TERMINATED BY '\r\n' # agregamos \r porque birthday tiene una coma que divide la celda
86 IGNORE 1 ROWS;
87
88 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv"
89 INTO TABLE users
90 FIELDS TERMINATED BY ','
91 ENCLOSED BY '"'
92 LINES TERMINATED BY '\r\n' # agregamos \r porque birthday tiene una coma que divide la celda
93 IGNORE 1 ROWS;

```

```

95 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv"
96 INTO TABLE users
97 FIELDS TERMINATED BY ','
98 ENCLOSED BY '"'
99 LINES TERMINATED BY '\r\n' # agregamos \r porque birthday tiene una coma que divide la celda
100 IGNORE 1 ROWS;
101
102 -- > CARGAMOS DATOS DE LA TABLA COMPANYY
103
104 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv"
105 INTO TABLE companies
106 FIELDS TERMINATED BY ','
107 ENCLOSED BY '"'
108 LINES TERMINATED BY '\n'
109 IGNORE 1 ROWS;
110
111 -- > CARGAMOS DATOS DE LA TABLA credit_cards
112
113 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv"
114 INTO TABLE credit_cards
115 FIELDS TERMINATED BY ','
116 ENCLOSED BY '"'
117 LINES TERMINATED BY '\n'
118 IGNORE 1 ROWS;

```

-- > y finalmente creamos la tabla de hechos **transactions** y le cargamos sus datos.

```

123 • DROP TABLE IF EXISTS transactions;
124 • CREATE TABLE IF NOT EXISTS transactions (
125     id VARCHAR(255) PRIMARY KEY,
126     card_id VARCHAR(15), #es varchar(10)
127     business_id VARCHAR(10), # ok
128     timestamp TIMESTAMP,
129     amount DECIMAL(10, 2),
130     declined boolean,
131     product_ids varchar (200),
132     user_id INT, #ok
133     lat FLOAT,
134     longitude FLOAT,
135     FOREIGN KEY (card_id) REFERENCES credit_cards(id),
136     FOREIGN KEY (business_id) REFERENCES companies(company_id),
137     FOREIGN KEY (user_id) REFERENCES users(id)
138 );
139
140
141
142 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv"
143 INTO TABLE transactions
144 FIELDS TERMINATED BY ';' # esta tabla está separada por ;
145 ENCLOSED BY '"'
146 LINES TERMINATED BY '\n'
147 IGNORE 1 ROWS;

```

-- > **AHORA SÍ, CON LOS DATOS CARGADOS, PASAMOS A LOS EJERCICIOS**

## NIVELL 1 - EXERCICI 1

-- > Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.

```
156 • SELECT id
157 FROM users
158 WHERE id IN (
159     SELECT user_id
160     FROM (
161         SELECT user_id, COUNT(id) cantidad_transacciones_por_usuarios
162         FROM transactions
163         GROUP BY user_id) trans_por_usuarios
164     WHERE cantidad_transacciones_por_usuarios > 30);
165
```

| Result Grid |      | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |
|-------------|------|--------------|-------|----------------|--------------------|
|             | id   |              |       |                |                    |
| ▶           | 92   |              |       |                |                    |
|             | 267  |              |       |                |                    |
|             | 272  |              |       |                |                    |
|             | 275  |              |       |                |                    |
| •           | NULL |              |       |                |                    |

## NIVELL 1 - EXERCICI 2

-- > Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.

```
171 • SELECT companies.company_name NOM_EMPRESA, credit_cards.iban, ROUND(AVG(transactions.amount),2)
172 FROM companies
173 JOIN transactions
174     ON companies.company_id=transactions.business_id
175 JOIN credit_cards
176     ON credit_cards.id=transactions.card_id
177 WHERE company_name = 'Donec Ltd'
178 GROUP BY 2,1
179 ;
```

| Result Grid |             | Filter Rows:              | Export:                           | Wrap Cell Content: |
|-------------|-------------|---------------------------|-----------------------------------|--------------------|
|             | NOM_EMPRESA | iban                      | ROUND(AVG(transactions.amount),2) |                    |
| ▶           | Donec Ltd   | PT87806228135092429456346 | 203.72                            |                    |

## NIVELL 2 - EXERCICI 1

-- > Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions van ser declinades i genera la següent consulta:

-- > Quantes targetes estan actives?

**ACLARACIÓ: CREÉ DOS TABLAS PORQUE LA PRIMERA NO ME TERMINABA DE CONVENCER. EN LA SEGUNDA VERSIÓN FINAL ADEMÁS DE VER SI LA TARJETA ESTÁ ACTIVA O NO, SE PUEDE VER EL NÚMERO DE DECLINES RECIENTES (EN LAS ÚLTIMAS TRES TRANSACCIONES).**

-- > Primero buscamos el modo de seleccionar las última tres transacciones por usuario de acuerdo al timestamp utilizando la función ROW\_NUMBER()

```
191 • SELECT id, card_id, timestamp, declined
192 FROM (
193     SELECT id, card_id, timestamp, declined,
194         ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS ultimas3
195     FROM transactions
196 ) AS ultimas_3
197 WHERE ultimas3 <= 3
198 ORDER BY card_id
199 # and declined = '1'
200 ;
```

| Result Grid                                                   |                                      |                                                                                             |                     |          |
|---------------------------------------------------------------|--------------------------------------|---------------------------------------------------------------------------------------------|---------------------|----------|
| Filter Rows: <input type="text"/> Export:  Wrap Cell Content: |                                      |                                                                                             |                     |          |
|                                                               | id                                   | card_id                                                                                     | timestamp           | declined |
| ▶                                                             | AD85A78A-8829-5746-93A0-8B7A792EBC18 | CcU-2938                                                                                    | 2022-03-12 09:23:10 | 0        |
|                                                               | F1A598A2-86C5-50A9-F1CE-FB1D69866C39 | CcU-2938                                                                                    | 2022-03-09 20:53:59 | 0        |
|                                                               | 55166D02-D74C-6A63-6C54-8678467649B4 | CcU-2938                                                                                    | 2022-02-24 11:01:42 | 0        |
|                                                               | 7DC26247-20EC-53FE-E555-B6C2E55CA5D5 | CcU-2945                                                                                    | 2022-02-04 15:52:56 | 0        |
|                                                               | FE96CE47-BD59-381C-4E18-E3CA3D44E8FF | CcU-2945                                                                                    | 2021-06-15 00:26:29 | 1        |
|                                                               | 72997E96-DC2C-A4D7-7C24-66C302F8AE5A | CcU-2952                                                                                    | 2022-01-30 15:16:36 | 0        |
| Result 154 x                                                  |                                      |                                                                                             |                     |          |
| Output                                                        |                                      |                                                                                             |                     |          |
| Action Output                                                 |                                      |                                                                                             |                     |          |
| #                                                             | Time                                 | Action                                                                                      | Message             |          |
| ✓ 689                                                         | 08:21:47                             | SELECT id, card_id, timestamp, declined FROM ( SELECT id, card_id, timestamp, declined, ... | 376 row(s) returned |          |

-- > Luego generamos una columna mediante el IF que me distinga a cada tarjeta como Activa o No Activa si las últimas 3 transacciones fueron declinadas

```
204 • SELECT card_id Card_id,  
205 IF(sum(declined) >=3, 'Not Active', 'Active') as 'Estado_Tarjeta',  
206 sum(declined) Recent_declines  
207 FROM (  
208     SELECT id, card_id, timestamp, declined,  
209         ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS ultimas3  
210     FROM transactions  
211 ) AS ultimas_3  
212 WHERE ultimas3 <= 3  
213 GROUP BY card_id  
214 ORDER BY Estado_Tarjeta DESC
```

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

|   | Card_id  | Estado_Tarjeta | Recent_declines |
|---|----------|----------------|-----------------|
| ▶ | CcU-2938 | Active         | 0               |
|   | CcU-2945 | Active         | 1               |
|   | CcU-2952 | Active         | 1               |
|   | CcU-2959 | Active         | 0               |
|   | CcU-2966 | Active         | 1               |
|   | CcU-2973 | Active         | 1               |

Result 159 x

Output

 Action Output

| #     | Time     | Action                                                                                                  | Message             |
|-------|----------|---------------------------------------------------------------------------------------------------------|---------------------|
| ✓ 694 | 08:34:45 | SELECT card_id Card_id, sum(declined) Recent_declines, IF(sum(declined) >=3, 'Not Active', 'Active')... | 275 row(s) returned |

-- > CREAMOS LA TABLA card\_status con la columna de Recent\_declines

-- > chequeamos cómo queda la tabla

```
235 • CREATE TABLE card_status
236     SELECT card_id Card_id,
237     IF(sum(declined) >=3, 'Not Active', 'Active') as 'Estado_Tarjeta',
238     sum(declined) Recent_declines
239 FROM (
240     SELECT id, card_id, timestamp, declined,
241     ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS ultimas3
242     FROM transactions
243 ) AS ultimas_3
244 WHERE ultimas3 <= 3
245 GROUP BY card_id
246 ORDER BY Estado_Tarjeta DESC
247 ;
248
249 • SELECT * from card_status;
250
```

Result Grid

|   | Card_id  | Estado_Tarjeta | Recent_declines |
|---|----------|----------------|-----------------|
| ▶ | CcU-2938 | Active         | 0               |
|   | CcU-2945 | Active         | 1               |
|   | CcU-2952 | Active         | 1               |
|   | CcU-2959 | Active         | 0               |
|   | CcU-2966 | Active         | 1               |

card\_status 197 x

Output

Action Output

| #     | Time     | Action                                                                                      | Message             |
|-------|----------|---------------------------------------------------------------------------------------------|---------------------|
| ✓ 746 | 10:59:44 | SELECT id, card_id, timestamp, declined FROM ( SELECT id, card_id, timestamp, declined, ... | 376 row(s) returned |
| ✓ 747 | 11:02:13 | SELECT * from card_status                                                                   | 275 row(s) returned |

-- > AGREGAMOS LA PRIMERY KEY A LA NUEVA TABLA Y LA FK QUE LA CONECTE A LA TABLA credit\_cards

```
252
253 • ALTER TABLE card_status ADD PRIMARY KEY(card_id);
254
255 • ALTER TABLE credit_cards
256     ADD FOREIGN KEY (id) REFERENCES card_status(card_id);
```

-- > A PARTIR DE AHORA, NUESTRO MODELO SE CONVIERTE EN UN MODELO 'COPO DE NIEVE' < --



## -- > VEMOS CUÁNTAS DE LAS TARJETAS ESTÁN ACTIVAS

```
266 • SELECT count(card_id) Total_active_cards
267 FROM card_status
268 where Estado_Tarjeta ='Active'
269 GROUP BY Estado_Tarjeta
270 ;
271 #275 Total_active_cards, es decir no hay ninguna de las tarjetas que haya sido declinada 3 veces en las últimas 3 transacciones
272
```

| Result Grid        | Filter Rows: | Export: | Wrap Cell Content: |
|--------------------|--------------|---------|--------------------|
| Total_active_cards |              |         |                    |
| 275                |              |         |                    |

## -- > esta Opción me permite ver las tarjetas que tuvieron al menos un decline en las últimas 3 transacciones y ver cuántas veces fueron declinadas

```
290 -- > esta Opción me permitiría ver las tarjetas que tuvieron al menos un decline en las últimas 3 transacciones
291 -- y ver cuántas veces fueron declinadas
292 • SELECT * from card_status
293 where Recent_declines >= 1;
294
```

| Result Grid | Filter Rows:   | Edit:           | Export/Import: | Wrap Cell Content: |
|-------------|----------------|-----------------|----------------|--------------------|
| Card_id     | Estado_Tarjeta | Recent_declines |                |                    |
| CcU-2945    | Active         | 1               |                |                    |
| CcU-2952    | Active         | 1               |                |                    |
| CcU-2966    | Active         | 1               |                |                    |
| CcU-2973    | Active         | 1               |                |                    |
| CcU-2980    | Active         | 1               |                |                    |

card\_status 201 x

Output

Action Output

| #   | Time     | Action                                                                                                   | Message            |
|-----|----------|----------------------------------------------------------------------------------------------------------|--------------------|
| 750 | 11:07:28 | select transactions.declined, transactions.card_id transcard, card_status.Estado_Tarjeta, timestamp f... | 0 row(s) returned  |
| 751 | 11:08:10 | SELECT * from card_status where Recent_declines >= 1                                                     | 84 row(s) returned |

## -- > HACEMOS CORROBORACIONES:

```
276 #es correcta la relación entre la tabla credit_cards y card_status
277 • select credit_cards.id, card_status.Card_id
278 from credit_cards left join card_status
279 on credit_cards.id=card_status.Card_id; # resultado 275 rows, correcto
280
```

| Result Grid | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|--------------|---------|--------------------|
| id          | Card_id      |         |                    |
| CcU-4856    | CcU-4856     |         |                    |
| CcU-4849    | CcU-4849     |         |                    |
| CcU-4842    | CcU-4842     |         |                    |
| CcU-4835    | CcU-4835     |         |                    |
| CcU-4828    | CcU-4828     |         |                    |

Result 199 x

Output

Action Output

| #   | Time     | Action                                                                                                     | Message             |
|-----|----------|------------------------------------------------------------------------------------------------------------|---------------------|
| 748 | 11:05:31 | SELECT count(card_id) Total_active_cards FROM card_status where Estado_Tarjeta ='Active' GRO...            | 1 row(s) returned   |
| 749 | 11:06:45 | select credit_cards.id, card_status.Card_id from credit_cards left join card_status on credit_cards.id=... | 275 row(s) returned |



```

281 -- > PRUEBO UN JOIN CON LA TABLA TRANSACCIONES (armo búsqueda para que me muestra si hubiera alguna 'Not Active'
282 • select transactions.declined, transactions.card_id transcard, card_status.Estado_Tarjeta, timestamp
283 from transactions join credit_cards
284 on transactions.card_id=credit_cards.id
285 join card_status
286 on credit_cards.id=card_status.Card_id
287 where Estado_Tarjeta = 'Not Active'
288 order by transcard, timestamp desc;
289

```

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

| declined | transcard | Estado_Tarjeta | timestamp |
|----------|-----------|----------------|-----------|
|----------|-----------|----------------|-----------|

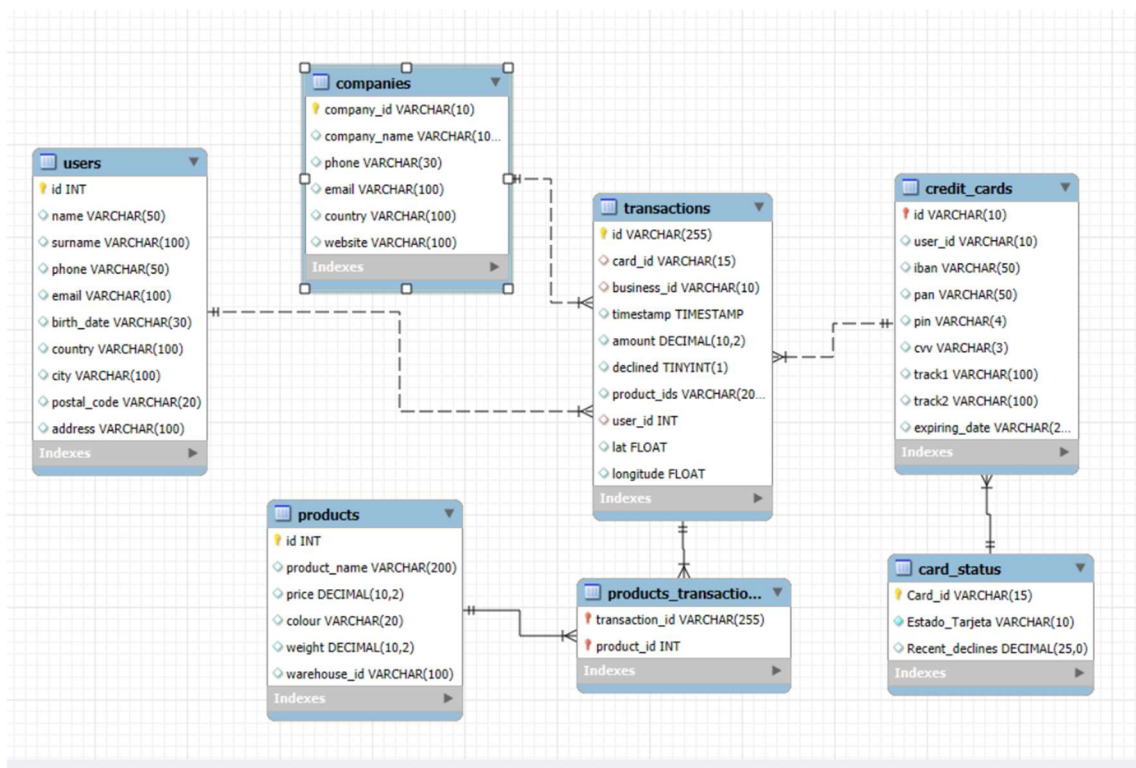
Result 200 x

Output

Action Output

| #     | Time     | Action                                                                                                     | Message             |
|-------|----------|------------------------------------------------------------------------------------------------------------|---------------------|
| ✓ 749 | 11:06:45 | select credit_cards.id, card_status.Card_id from credit_cards left join card_status on credit_cards.id=... | 275 row(s) returned |
| ✓ 750 | 11:07:28 | select transactions.declined, transactions.card_id transcard, card_status.Estado_Tarjeta, timestamp f...   | 0 row(s) returned   |

## DIAGRAMA FINAL DEL MODELO CREADO:





## NIVELL 3 - EXERCICI 1

-- > Crea una taula amb la qual puguem unir les dades del nou arxiu products.csv amb la base de dades creada, tenint en compte que des de transaction tens product\_ids. Genera la següent consulta:

-- > Necessitem conèixer el nombre de vegades que s'ha venut cada producte.

### -- CREAMOS LA TABLA products

```
346 -- CREAMOS LA TABLA products
347 • DROP TABLE IF EXISTS products;
348 • CREATE TABLE IF NOT EXISTS products (
349     id INT PRIMARY KEY NOT NULL,
350     product_name VARCHAR(200),
351     price VARCHAR (200),
352     colour VARCHAR(20),
353     weight DECIMAL (10,2),
354     warehouse_id VARCHAR(100)
355 );
```

### -- CARGAMOS DATOS

# Me genera conflicto con el símbolo \$ en la columna Price. Resuelvo mediante:

SET price = REPLACE(@price\_raw, '\$', '');

```
369 -- CON REEMPLAZO DEL $ EN PRICE
370 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv"
371 INTO TABLE products
372 FIELDS TERMINATED BY ','
373 ENCLOSED BY '"'
374 LINES TERMINATED BY '\n'
375 IGNORE 1 ROWS
376 (id, product_name, @price_raw, colour, weight, warehouse_id)
377 SET price = REPLACE(@price_raw, '$', ''); #QUITAMOS EL $ DE PRICE
```

-- CAMBIAMOS EL CAMPO 'PRICE' DE VARCHAR A DECIMAL y  
CORROBORAMOS LA CARGA DE DATOS, OK ;)

```

380      -- CAMBIAMOS EL CAMPO 'PRICE' DE VARCHAR A DECIMAL
381 •    ALTER TABLE products MODIFY price DECIMAL (10,2);
382
383      -- CORROBORAMOS LA CARGA DE DATOS, OK ;)
384 •    SELECT * from products;
385

```

| Result Grid |    |                        |        |         |                |              |
|-------------|----|------------------------|--------|---------|----------------|--------------|
|             |    | Filter Rows:           | Edit:  |         | Export/Import: |              |
|             | id | product_name           | price  | colour  | weight         | warehouse_id |
| ▶           | 1  | Direwolf Stannis       | 161.11 | #7c7c7c | 1.00           | WH-4         |
|             | 2  | Tarly Stark            | 9.24   | #919191 | 2.00           | WH-3         |
|             | 3  | duel tourney Lannister | 171.13 | #d8d8d8 | 1.50           | WH-2         |
|             | 4  | warden south duel      | 71.89  | #111111 | 3.00           | WH-1         |
|             | 5  | skywalker ewok         | 171.22 | #dbdbdb | 3.20           | WH-0         |
|             | 6  | dooku solo             | 136.60 | #c4c4c4 | 0.80           | WH--1        |

products 166 x

Output :

Action Output

| #     | Time     | Action                                                |
|-------|----------|-------------------------------------------------------|
| ✓ 701 | 08:51:36 | SELECT * from card_status2 where Recent_declines >= 1 |

## -----

### -- PROBLEMA:

-- la columna 'produc\_ids' de la tabla transactions, al tener varios product ids en cada registro no me matchea con los id de la tabla products

-- CREAMOS UNA TABLA INTERMEDIA PARA EVITAR LA RELACIÓN MUCHOS A MUCHOS ENTRE LA TABLA PRODUCTS Y LA TABLA TRANSACTIONS

-- utilizamos estrategia mediante tabla JSON para separar los valores de la columna produc\_ids de la tabla transactions

```
408 • DROP TABLE IF EXISTS products_transactions;
409 • CREATE TABLE IF NOT EXISTS products_transactions
410     SELECT DISTINCT
411         transactions.id AS transaction_id,
412         product_id
413     FROM
414         transactions
415     JOIN
416         JSON_TABLE(
417             CONCAT('["', REPLACE(transactions.product_ids, ',', '","'), "']'),
418             '$[*]' COLUMNS (product_id INT PATH '$')
419         ) jt
420     ON jt.product_id = product_id
421     LEFT JOIN
422         products
423     ON jt.product_id = product_id
424     ORDER BY
425         transactions.id;
```

```
427 • SELECT * FROM products_transactions;
```

428

429

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

|   | transaction_id                       | product_id |
|---|--------------------------------------|------------|
| ▶ | 02C6201E-D90A-1859-84EE-88D2986D3B02 | 1          |
|   | 122DC333-E19F-D629-DCD8-9C54CF1EBB9A | 1          |
|   | 1753A288-9FC1-52E6-5C39-A1FFB97B0D3A | 1          |
|   | 1A6CECFB-2E3A-65A3-72D9-2FDB58A1E4BA | 1          |
|   | 1EA2B262-D507-AD14-4374-4D532967113F | 1          |

products\_transactions 168 x

Output

Action Output

| #     | Time     | Action                              | Message              |
|-------|----------|-------------------------------------|----------------------|
| ✓ 703 | 08:59:24 | SELECT * FROM products_transactions | 1457 row(s) returned |

```
429     -----
430     -- > GENERAMOS LA PRIMARY KEY COMPUESTA DE NUESTRA TABLA INTERMEDIA products_transactions
431     -----
432 • ALTER TABLE products_transactions
433     ADD PRIMARY KEY (transaction_id, product_id)
434     ;
```

```

436 -- > GENERAMOS LAS FOREIGN KEYS EN LAS TABLA products_transactions CON LAS TABLAS transactions y products
437
438 # LA FK CON transactions ME FUNCIONA
439 • ALTER TABLE products_transactions
440   ADD CONSTRAINT fk_transactions_id
441   FOREIGN KEY (transaction_id)
442   REFERENCES transactions(id)
443   ;
444 • ALTER TABLE products_transactions
445   ADD CONSTRAINT fk_product_id
446   FOREIGN KEY (product_id)
447   REFERENCES products(id)
448   ;

```

-----

-- > Necesitem conèixer el nombre de vegades que s'ha venut cada producte.

-----

```

451 -----
452 -- > Necesitem conèixer el nombre de vegades que s'ha venut cada producte.
453 -----
454 • SELECT product_id, COUNT(transaction_id)cantidad_vendida
455   FROM products_transactions
456   GROUP BY product_id
457   ORDER BY product_id;
458

```

| Result Grid |                  | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|------------------|--------------|---------|--------------------|
| product_id  | cantidad_vendida |              |         |                    |
| 1           | 61               |              |         |                    |
| 2           | 65               |              |         |                    |
| 3           | 51               |              |         |                    |
| 5           | 49               |              |         |                    |
| 7           | 54               |              |         |                    |

Result 170 x

Output

Action Output

| #     | Time     | Action                                                                                      | Message            |
|-------|----------|---------------------------------------------------------------------------------------------|--------------------|
| ✓ 705 | 09:02:29 | SELECT product_id, COUNT(transaction_id)cantidad_vendida FROM products_transactions GROU... | 26 row(s) returned |

-- > EXTRA: así podemos corroborar la cantidad de ventas de todos los productos presentes en la tabla 'products' viendo también cuál no ha tenido ninguna venta hasta ahora




```

462 • SELECT products.id, cantidad_vendida from products
463 left join (
464     SELECT product_id, COUNT(transaction_id)cantidad_vendida
465     FROM products_transactions
466     GROUP BY product_id
467     ORDER BY product_id) cantidades
468 ON products.id=cantidades.product_id;
469

```

Result Grid  Filter Rows:  Export:  Wrap Cell Content: 

|   | id | cantidad_vendida |
|---|----|------------------|
| ▶ | 1  | 61               |
|   | 2  | 65               |
|   | 3  | 51               |
|   | 4  | NULL             |
|   | 5  | 49               |

Result 176 

Output

 Action Output

| #     | Time     | Action                                                                                             | Message             |
|-------|----------|----------------------------------------------------------------------------------------------------|---------------------|
| ✓ 711 | 09:05:12 | SELECT product_id, COUNT(transaction_id)cantidad_vendida FROM products_transactions GROU...        | 26 row(s) returned  |
| ✓ 712 | 09:05:26 | SELECT products.id, cantidad_vendida from products left join ( SELECT product_id, COUNT(transac... | 100 row(s) returned |