

# Nivell 1

Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguem realitzar les següents consultes:

He agafat les quatre taules següents per incloure a l'esquema: users, credit\_cards, company i transaction

```
CREATE DATABASE IF NOT EXISTS vendes;
```

```
USE vendes;
```

```
CREATE TABLE IF NOT EXISTS users (
```

```
  id          VARCHAR(255) PRIMARY KEY,  
  name        VARCHAR(100),  
  surname     VARCHAR(100),  
  phone       VARCHAR(30),  
  email       VARCHAR(50),  
  birth_date  VARCHAR(25),  
  country     VARCHAR(50),  
  city        VARCHAR(50),  
  postal_code VARCHAR(10),  
  address     VARCHAR(255)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS credit_cards (
```

```
  id          VARCHAR(255) PRIMARY KEY,  
  user_id     VARCHAR(255),  
  iban        VARCHAR(255),  
  pan         VARCHAR(255),  
  pin         VARCHAR(4),  
  cvv         VARCHAR(3),  
  track1      VARCHAR(255),  
  track2      VARCHAR(255),  
  expiring_date VARCHAR(25)
```

```
);
```

```
-- Creamos la tabla company
```

```
CREATE TABLE IF NOT EXISTS companies (
```

```
  id          VARCHAR(255) PRIMARY KEY,  
  name        VARCHAR(255),  
  phone       VARCHAR(15),  
  email       VARCHAR(50),  
  country     VARCHAR(50),  
  website     VARCHAR(255)
```

```
);
```

-- Creamos la tabla transactions

CREATE TABLE IF NOT EXISTS transactions (

id VARCHAR(255) PRIMARY KEY,

card\_id VARCHAR(255),

business\_id VARCHAR(255),

date\_tx VARCHAR(25),

amount DECIMAL(10, 2),

declined BOOLEAN,

product\_ids VARCHAR(255),

user\_id VARCHAR(255),

lat VARCHAR(255),

longitude VARCHAR(255),

FOREIGN KEY (card\_id) REFERENCES credit\_cards(id),

FOREIGN KEY (business\_id) REFERENCES companies(id),

FOREIGN KEY (user\_id) REFERENCES users(id)

);

The screenshot shows the MySQL Workbench interface. The main editor displays a SQL script for creating the 'transactions' table and loading data from CSV files. The script includes the following SQL statements:

```
CREATE TABLE IF NOT EXISTS transactions (
  id VARCHAR(255) PRIMARY KEY,
  card_id VARCHAR(255),
  business_id VARCHAR(255),
  date_tx VARCHAR(25),
  amount DECIMAL(10, 2),
  declined BOOLEAN,
  product_ids VARCHAR(255),
  user_id VARCHAR(255),
  lat VARCHAR(255),
  longitude VARCHAR(255),
  FOREIGN KEY (card_id) REFERENCES credit_cards(id),
  FOREIGN KEY (business_id) REFERENCES companies(id),
  FOREIGN KEY (user_id) REFERENCES users(id)
);

SHOW VARIABLES LIKE "secure_file_priv";
SELECT @@GLOBAL.secure_file_priv;
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/european_users.csv" INTO TABLE european_users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(id, name, surname, phone, email, birth_date, country, city, postal_code, address)
-- SET birth_date = DATE_FORMAT(birth_date, '%Y %M %D %H')
;

LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/american_users.csv" INTO TABLE european_users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(id, name, surname, phone, email, birth_date, country, city, postal_code, address)
-- SET birth_date = DATE_FORMAT(birth_date, '%Y %M %D %H')
;
```

The Output tab at the bottom shows the execution results of the script:

#	Time	Action	Message	Duration / Fetch
1	13:36:52	CREATE DATABASE IF NOT EXISTS vendes	1 row(s) affected	0.000 sec
2	13:36:53	USE vendes	0 row(s) affected	0.016 sec
3	13:37:00	CREATE TABLE IF NOT EXISTS users ( id VARCHAR(255) PRIMARY KEY, name VARCHAR(100), phone VARCHAR(100), email VARCHAR(100), birth_date VARCHAR(255), country VARCHAR(255), city VARCHAR(255), postal_code VARCHAR(255), address VARCHAR(255))	0 row(s) affected	0.016 sec
4	13:37:01	CREATE TABLE IF NOT EXISTS credit_cards ( id VARCHAR(255) PRIMARY KEY, user_id VARCHAR(255), iban VARCHAR(255), pan VARCHAR(255), expir VARCHAR(255), cvv VARCHAR(255))	0 row(s) affected	0.016 sec
5	13:37:05	CREATE TABLE IF NOT EXISTS companies ( id VARCHAR(255) PRIMARY KEY, name VARCHAR(255), phone VARCHAR(100), email VARCHAR(100), birth_date VARCHAR(255), country VARCHAR(255), city VARCHAR(255), postal_code VARCHAR(255), address VARCHAR(255))	0 row(s) affected	0.015 sec
6	13:37:07	CREATE TABLE IF NOT EXISTS transactions ( id VARCHAR(255) PRIMARY KEY, card_id VARCHAR(255), business_id VARCHAR(255), date_tx VARCHAR(25), amount DECIMAL(10, 2), declined BOOLEAN, product_ids VARCHAR(255), user_id VARCHAR(255), lat VARCHAR(255), longitude VARCHAR(255))	0 row(s) affected	0.032 sec

```
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/european_users.csv" INTO
TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(id, name, surname, phone, email, birth_date, country, city, postal_code, address)
-- SET birth_date = DATE_FORMAT(@birth_date, '%b %d, %Y');
```

```
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/american_users.csv" INTO
TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(id, name, surname, phone, email, birth_date, country, city, postal_code, address)
-- SET birth_date = DATE_FORMAT(@birth_date, '%b %d, %Y');
```

```
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv" INTO TABLE
credit_cards
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(`id`, `user_id`, `iban`, `pan`, `pin`, `cvv`, `track1`, `track2`, `expiring_date`);
```

```
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv" INTO TABLE
companies
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(`id`, `name`, `phone`, `email`, `country`, `website`);
```

```
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv" INTO TABLE
transactions
FIELDS TERMINATED BY ';'
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(`id`, `card_id`, `business_id`, `date_tx`, `amount`, `declined`, `product_ids`, `user_id`, `lat`, `longitude`);
```

MySQL Workbench

Local instance MySQL84 x MySQL Model x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

- Filter objects
- sakila
- sys
- world

Administration Schemas

Information

No object selected

Query 1

```
-- SET birth_date = DATE_FORMAT(@birth_date, '%Y-%m-%d')
-- SET birth_date = DATE_FORMAT(@birth_date, '%Y-%m-%d')

81 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/american_users.csv' INTO TABLE users
82 FIELDS TERMINATED BY ','
83 ENCLOSED BY '"'
84 LINES TERMINATED BY '\n'
85 IGNORE 1 LINES
86 (id, name, surname, phone, email, birth_date, country, city, postal_code, address)
87 -- SET birth_date = DATE_FORMAT(@birth_date, '%Y-%m-%d')
88
89
90 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv' INTO TABLE credit_cards
91 FIELDS TERMINATED BY ','
92 ENCLOSED BY '"'
93 LINES TERMINATED BY '\n'
94 IGNORE 1 LINES
95 ('id', 'user_id', 'iban', 'pan', 'pin', 'cvv', 'track1', 'track2', 'expiring_date')
96
97
98 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv' INTO TABLE companies
99 FIELDS TERMINATED BY ','
100 ENCLOSED BY '"'
101 LINES TERMINATED BY '\n'
102 IGNORE 1 LINES
103 ('id', 'name', 'phone', 'email', 'country', 'website')
104
105 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv' INTO TABLE transactions
106 FIELDS TERMINATED BY ','
107 ENCLOSED BY '"'
108 LINES TERMINATED BY '\n'
109 IGNORE 1 LINES
110 ('id', 'card_id', 'business_id', 'date_tx', 'amount', 'declined', 'product_id', 'user_id', 'lat', 'longitude')
111
```

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Output

Action	Output	Message	Duration / Fetch
4 13:39:34	CREATE TABLE IF NOT EXISTS credit_cards ( id VARCHAR(255) PRIMARY KEY, user_id VARCHAR(255), iban VARCHAR(255), pan VARCHAR(255), pin VARCHAR(255), cvv VARCHAR(255), track1 VARCHAR(255), track2 VARCHAR(255), expiring_date VARCHAR(255))	0 row(s) affected	0.000 sec
5 13:39:37	CREATE TABLE IF NOT EXISTS companies ( id VARCHAR(255) PRIMARY KEY, name VARCHAR(255), phone VARCHAR(255), email VARCHAR(255), country VARCHAR(255), website VARCHAR(255))	0 row(s) affected	0.016 sec
6 13:39:40	CREATE TABLE IF NOT EXISTS transactions ( id VARCHAR(255) PRIMARY KEY, card_id VARCHAR(255), business_id VARCHAR(255), date_tx VARCHAR(255), amount VARCHAR(255), declined VARCHAR(255), product_id VARCHAR(255), user_id VARCHAR(255), lat VARCHAR(255), longitude VARCHAR(255))	0 row(s) affected	0.031 sec
7 13:39:45	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/american_users.csv' INTO TABLE users FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES	3990 row(s) affected Records: 3990 Deleted: 0 Skipped: 0 Warnings: 0	0.219 sec
8 13:39:48	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/american_users.csv' INTO TABLE users FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES	1010 row(s) affected Records: 1010 Deleted: 0 Skipped: 0 Warnings: 0	0.031 sec
9 13:39:50	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv' INTO TABLE credit_cards FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES	5000 row(s) affected Records: 5000 Deleted: 0 Skipped: 0 Warnings: 0	0.140 sec
10 13:39:53	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv' INTO TABLE companies FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.016 sec
11 13:39:55	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv' INTO TABLE transactions FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES	100000 row(s) affected Records: 100000 Deleted: 0 Skipped: 0 Warnings: 0	5.266 sec

Query Completed

Windows Taskbar: Buscar, Uploads, Discord | Fresh..., MySQL Workbench, Sprint4.pdf - Adobe..., Zoom Reunión, Sprint4.docx - L..., Nuevo2 - Note..., ES, 21°C, 13:41, 16/10/2023

## Exercici 1

Realitza una subconsulta que mostri tots els usuaris amb més de 80 transaccions utilitzant almenys 2 taules.

```
SELECT u.id, count(u.id) num_tx FROM european_users u
JOIN transactions t ON user_id = u.id
GROUP BY u.id
HAVING count(*) > 80;
```

```
SELECT id FROM european_users
WHERE id IN(
```

```
SELECT user_id FROM transactions
GROUP BY user_id
HAVING count(*) > 80
```

```
);
```

The screenshot shows the MySQL Workbench interface. The main editor displays a query for 'Exercici 1' which is a subquery to find users with more than 80 transactions. The query is as follows:

```
-- Exercici 1
-- Realitza una subconsulta que mostri tots els usuaris amb més de 80 transaccions utilitzant almenys 2 taules.

SELECT u.id, count(u.id) num_tx FROM european_users u
JOIN transactions t ON user_id = u.id
GROUP BY u.id
HAVING count(*) > 80;

SELECT id FROM european_users
WHERE id IN(
    SELECT user_id FROM transactions
    GROUP BY user_id
    HAVING count(*) > 80
);

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

The 'Result Grid' shows the results of the query. The first table, 'european\_users', has 4 rows returned. The second table, 'transactions', has 4 rows returned.

id
185
289
318
454

id
185
289
318
454

The 'Output' pane shows the execution of the query. The first query (SELECT u.id, count(u.id) num\_tx FROM european\_users u JOIN transactions t ON user\_id = u.id GROUP BY u.id HAVING count(\*) > 80) returned 4 rows. The second query (SELECT id FROM european\_users WHERE id IN( SELECT user\_id FROM transactions GROUP BY user\_id HAVING count(\*) > 80)) returned 4 rows.

## Exercici 2

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

```
SELECT iban,AVG(amount),count(*) FROM transactions t
JOIN credit_cards cc ON cc.id = card_id
JOIN companies c ON c.id = t.business_id AND name = 'Donec Ltd'
GROUP BY iban;
```

The screenshot shows the MySQL Workbench interface. The main window displays a SQL query for 'Exercici 2' which calculates the average amount and count of transactions for a specific company (Donec Ltd) grouped by IBAN. The query is as follows:

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

SELECT iban,AVG(amount),count(*) FROM transactions t
JOIN credit_cards cc ON cc.id = card_id
JOIN companies c ON c.id = t.business_id AND name = 'Donec Ltd'
GROUP BY iban;
```

The 'Result Grid' shows the output of the query, displaying columns: iban, AVG(amount), and count(\*). The results are as follows:

iban	AVG(amount)	count(*)
XX011406401125586307586805	356.246467	3
9X0946370242474962577306	142.960000	1
XX78782917849592975535640	257.370000	1
XX41382736228973309469990	136.590000	1
XX34787246070769610780308	240.410000	1
XX68768436543090894854602	188.580000	1
MC3836851536680349	439.390000	1
9X7626282566652676345404576	541.560000	3
XX804008331134918341803913	189.210000	1
US6465553777363527873049938	155.500000	2
XX54512784755611381008445	255.830000	1
XX15487686678663690754758	285.110000	1
CR3934463639573721674	240.480000	1
XX8239102392787176786536	88.020000	1
XX8979051542844213813607	271.970000	1
MC9813918248828136126478487	74.310000	1
NC0414757617220	95.365000	2

The 'Output' window shows the execution progress of the query, including messages for table creation, data loading, and the final result set returned.

## Nivell 2

Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en *si les tres últimes transaccions han estat declinades aleshores és inactiu, si almenys una no és rebutjada aleshores és actiu.*

```
CREATE TABLE IF NOT EXISTS estat_credit_cards
WITH ultims AS (
    SELECT card_id, date_tx, declined,
    ROW_NUMBER() OVER(PARTITION BY card_id ORDER
BY date_tx DESC) AS rank_numero
FROM transactions
WHERE declined=1
order by card_id,rank_numero
)
SELECT card_id, 0 AS estat FROM ultims ul WHERE EXISTS (SELECT card_id WHERE rank_numero
=3 AND card_id = ul.card_id)
UNION
SELECT DISTINCT(card_id), 1 AS estat FROM transactions WHERE card_id NOT IN ( SELECT card_id
FROM ultims ul WHERE EXISTS (SELECT card_id WHERE rank_numero =3 AND card_id = ul.card_id )
;
```

The screenshot displays the MySQL Workbench interface. The main window shows a SQL query being executed. The query is as follows:

```
-- Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les tres últimes transaccions han estat declinades aleshores és inactiu,
-- si almenys una no és rebutjada aleshores és actiu.

-- DROP TABLE estat_credit_cards;

CREATE TABLE IF NOT EXISTS estat_credit_cards
WITH ultims AS (
    SELECT card_id, date_tx, declined,
    ROW_NUMBER() OVER(PARTITION BY card_id ORDER BY date_tx DESC) AS rank_numero
FROM transactions
WHERE declined=1
order by card_id,rank_numero
)
SELECT card_id, 0 AS estat FROM ultims ul WHERE EXISTS (SELECT card_id WHERE rank_numero =3 AND card_id = ul.card_id)
UNION
SELECT DISTINCT(card_id), 1 AS estat FROM transactions WHERE card_id NOT IN ( SELECT card_id FROM ultims ul WHERE EXISTS (SELECT card_id WHERE rank_numero =3 AND card_id = ul.card_id )
);
```

The query is executed, and the results are displayed in the 'Result Grid' tab. The results show a list of card IDs and their corresponding 'estat' (status) values. The 'estat' column has two possible values: 0 (indicating inactiu) and 1 (indicating actiu). The results are as follows:

card_id	estat
XX0114064012558630786805	356.246667
909446370242474563277806	142.960000
XX7767291794695207555640	257.370000
XX41382736228971930409990	139.590000
XX347878246070769610780308	240.410000
XX6887683655090994954602	188.580000
MC3636851536688349	439.390000
PL76249283566852676343404576	541.560000
XX004008331134918314802913	896.210000
LB646555777736352787049308	355.500000
XX35451278375821381006445	255.830000
XX154676966783603690754798	285.110000
CS323466023577272474	246.480000
XX823910239278717927986536	85.020000
XX89790515422844213813607	271.970000
MC08191618628138126479467	74.510000
ND4414757761220	95.165000

The 'Output' tab shows the execution progress and messages. The messages indicate that the table was created successfully and that the query executed without errors. The execution time for the query was 0.000 seconds.

Partint d'aquesta taula respon:

## Exercici 1

Quantes targetes estan actives?

`SELECT COUNT(*) FROM estat_credit_cards WHERE estat=1;`

The screenshot shows the MySQL Workbench interface. The main window displays a SQL query in the 'Query Editor' tab. The query is as follows:

```
132 order by card_id,rank_numero
133 )
134 SELECT card_id , 0 AS estat FROM ultims ul WHERE EXISTS (SELECT card_id WHERE rank_numero =3 AND card_id = ul.card_id)
135 UNION
136 SELECT DISTINCT(card_id), 1 AS estat FROM transactions WHERE card_id NOT IN ( SELECT card_id FROM ultims ul WHERE EXISTS (SELECT card_id WHERE rank_numero =3 AND card_id = ul.card_id) )
137 }
138
139 -- Exercici 1
140 -- Quantes targetes estan actives?
141
142 SELECT COUNT(*) FROM estat_credit_cards WHERE estat=1;
143
144 ----- nivell 3 -----
145 -- Crea una taula amb la qual pugem unir les dades del nou arxíu products.csv amb la base de dades creada, tenint en compte que des de transaction tens product_ids.
146
147 -- Creamos la tabla products
148 DROP TABLE products;
```

The 'Result Grid' shows the execution of the query, resulting in a single row with the value 4963.

Result Grid
COUNT(*)
4963

The 'Output' tab shows the execution log, including the following messages:

- 10:40:43 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/loads/companies.csv' INTO TABLE companies FIELDS TERMINATED BY...
- 10:40:45 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/loads/transactions.csv' INTO TABLE transactions FIELDS TERMINATED BY...
- 10:40:46 SELECT u.id, count(u.id) num\_tx FROM european\_users u JOIN transactions t ON user\_id = u.id GROUP BY u.id HAVING count(\*) > 80 LIMIT 0...
- 10:40:47 SELECT ban.AVG(amount) count(\*) FROM transactions t JOIN credit\_cards cc ON cc.id = card\_id JOIN companies c ON c.id = t.business\_id A...
- 10:40:48 SELECT COUNT(\*) FROM estat\_credit\_cards WHERE estat=1 LIMIT 0, 1000
- 10:40:21 CREATE TABLE IF NOT EXISTS estat\_credit\_cards WITH ULTIMA AS ( SELECT card\_id, data\_tx, declined ROW\_NUMBER() OVERPARTIT...
- 10:40:31 SELECT COUNT(\*) FROM estat\_credit\_cards WHERE estat=1 LIMIT 0, 1000

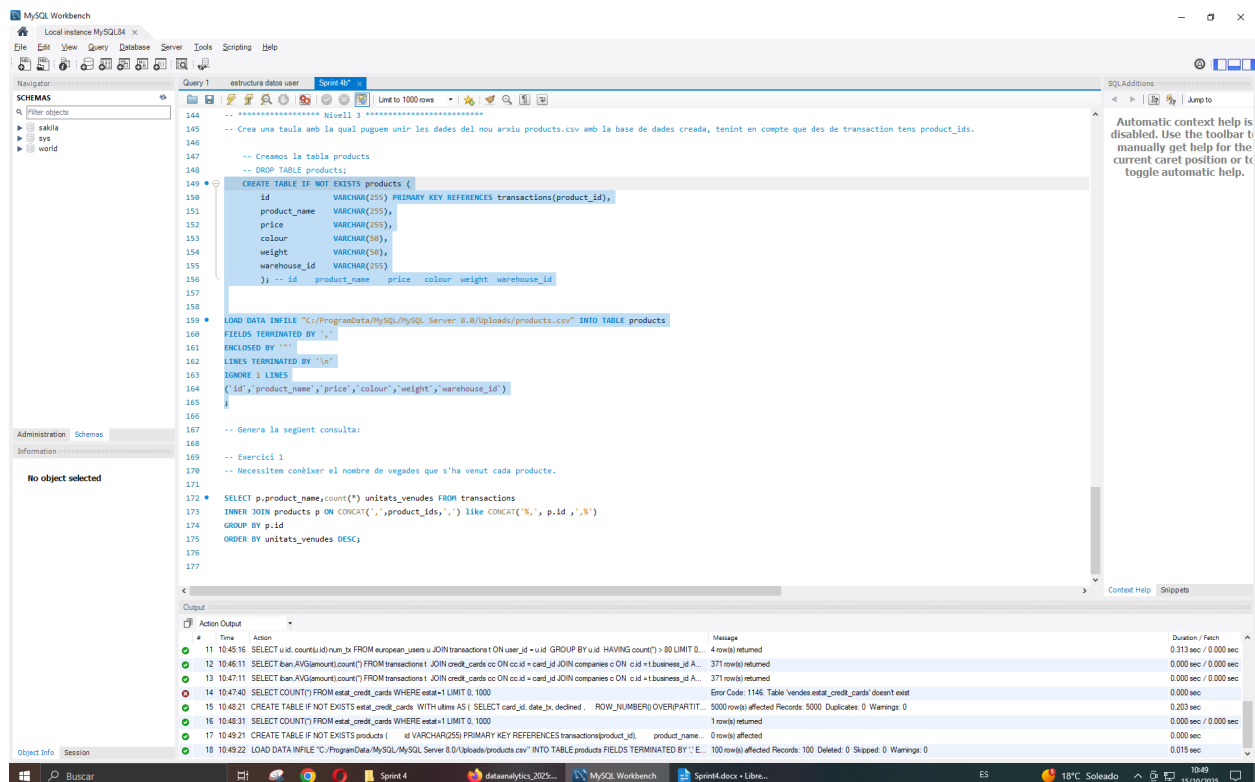


## Nivell 3

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:

```
CREATE TABLE IF NOT EXISTS products (  
    id VARCHAR(255) PRIMARY KEY REFERENCES transactions(product_id),  
    product_name VARCHAR(255),  
    price VARCHAR(255),  
    colour VARCHAR(50),  
    weight VARCHAR(50),  
    warehouse_id VARCHAR(255)  
);
```

```
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv" INTO TABLE  
products  
FIELDS TERMINATED BY ','  
ENCLOSED BY '"'  
LINES TERMINATED BY '\n'  
IGNORE 1 LINES  
(`id`,`product_name`,`price`,`colour`,`weight`,`warehouse_id`)  
;
```



# Exercici 1

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

```
SELECT p.id,count(*) unitats_venudes FROM transactions
INNER JOIN products p ON CONCAT(' ',product_ids,',') like CONCAT('% ', p.id ',,%')
GROUP BY p.id
ORDER BY unitats_venudes DESC;
```

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL script. The script includes a comment in Catalan: 'Necessitem conèixer el nombre de vegades que s'ha venut cada producte.' (We need to know the number of times each product has been sold). The SQL query is as follows:

```
SELECT p.id,count(*) unitats_venudes FROM transactions
INNER JOIN products p ON CONCAT(' ',product_ids,',') like CONCAT('% ', p.id ',,%')
GROUP BY p.id
ORDER BY unitats_venudes DESC;
```

The 'Results' tab shows the output of the query. The table has two columns: 'id' and 'unitats\_venudes'. The results are ordered by 'unitats\_venudes' in descending order.

id	unitats_venudes
52	2654
29	2635
21	2609
16	2608
66	2601
87	2598
33	2597
48	2597

The 'Output' tab shows the execution log, including messages and warnings. The log indicates that the query was executed successfully, with 100 rows affected and 0 warnings.