

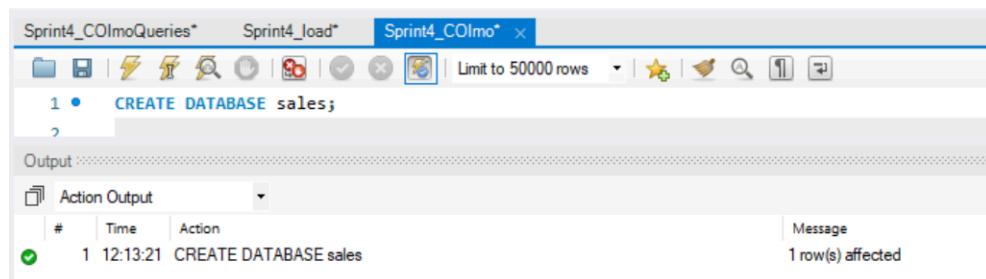
# SPRINT 4. CREACIÓ DE BASE DE DADES

## Nivell 1

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

Per a la realització d'aquest exercici he seguit una sèrie de pasos:

1. He anomenat la base de dades 'sales'. He creat la base de dades al MYSQL.



The screenshot shows the MySQL Workbench interface. The top tab bar has three tabs: 'Sprint4\_COLmoQueries\*', 'Sprint4\_load\*', and 'Sprint4\_COLmo\*' (which is active). Below the tabs is a toolbar with various icons. The main area contains a SQL editor window with the following content:

```
1 • CREATE DATABASE sales;
```

Below the SQL editor is an 'Output' pane. It has a header 'Action Output' with columns '#', 'Time', 'Action', and 'Message'. There is one entry:

#	Time	Action	Message
1	12:13:21	CREATE DATABASE sales	1 row(s) affected

2. He obert i analitzat els diferents .csv per tal de veure quin tipus de dades emmagatzemen i per crear les diferents taules.
3. Abans de començar la creació mitjançant el comandament USE m'asseguro que tots els canvis es faran sobre la nova DATABASE sales
4. En crear les taules he seguit el següent ordre:
  - a. Taula 'Companies'
    - i. company\_id com a alfanumèric de 20 caràcters i PK de la taula amb el CONSTRAINT NOT NULL
    - ii. company\_name com a alfanumèric de 100 caràcters
    - iii. phone com a alfanumèric de 20 caràcters (encara que és un camp numèric és merament informatiu i no hem d'operar amb aquests números)
    - iv. email com a alfanumèric de 100 caràcters
    - v. country com a alfanumèric de 100 caràcters
    - vi. website com a alfanumèric de 255 caràcters

```

3 • USE sales;
4 • CREATE TABLE companies (
5     company_id VARCHAR(20) PRIMARY KEY NOT NULL,
6     company_name VARCHAR (100),
7     phone VARCHAR (50),
8     email VARCHAR (100),
9     country VARCHAR (100),
10    website VARCHAR (255)
11 );
--
```

#### Output

Action Output			Message
#	Time	Action	
336	13:39:18	USE sales	0 row(s) affected
337	13:39:18	CREATE TABLE companies ( company_id VARCHAR(20) PRIMARY KEY NOT NU...	0 row(s) affected

b. Taula 'Products'

- i. id com a alfanumèric de 20 caràcters, PK de la taula products i amb constraint NOT NULL
- ii. product\_name com a alfanumèric de 100 caràcters
- iii. price com a alfanumèric de 20 caràcters. Aquest camp seria més útil com a INT per a poder operar i realitzar mitjanes de preus, etc. però al .csv apareix acompañat del simbol de currency, per tant, l'he creat com a varchar i posteriorment, es podrà transformar a numèric.
- iv. colour com a alfanumèric de 50 caràcters
- v. weight com a numèric decimal
- vi. warehouse\_id com a alfanumèric de 20 caràcters.

```

13 • CREATE TABLE products(
14     id varchar(20) PRIMARY KEY NOT NULL,
15     product_name VARCHAR(100),
16     price VARCHAR(20),
17     colour VARCHAR(50),
18     weight FLOAT,
19     warehouse_id VARCHAR(20)
20 );
```

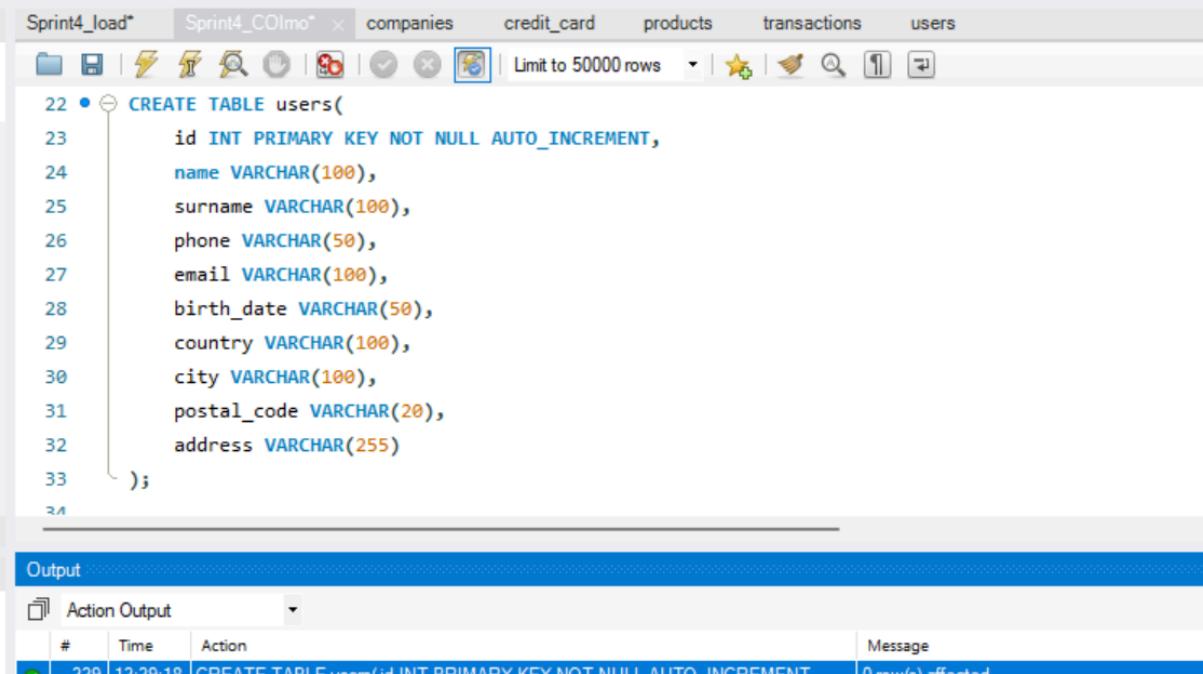
#### Output

Action Output			Message
#	Time	Action	
338	13:39:18	CREATE TABLE products(id varchar(20) PRIMARY KEY NOT NULL, product_n...	0 row(s) affected

c. Taula 'users'.

Per a l'elaboració d'aquesta taula he decidit crear una estructura de dades base que pugui fer servir per les tres taules de les quals disposem que es refereixen als usuaris (users\_usa.csv, users\_uk.csv, users\_ca.csv). En evaluar les diferents taules, observem que presenten els mateixos atributs i amb el mateix format, per tant, amb la creació d'una taula comuna podem optimitzar recursos en relació al codi necessitat per a crear la taula, i el codi de càrrega de les dades. A més la nostra database compta amb una taula menys, per tant, és més eficient ja que no precisem fer UNION sobre les dades per obtenir els totals d'usuaris. I, en cas de necessitar una segmentació per país podem fer-la fàcilment amb un GROUP BY per country.

- i. id com a INT que s'autoincrementa amb la introducció de nous usuaris, funciona com a PK de la taula users i per tant, compta amb el CONSTRAINT NOT NULL.
- ii. name com a camp alfanumèric de 100 caràcters
- iii. surname com a alfanumèric de 100 caràcters
- iv. phone com a alfanumèric de 50 caràcters
- v. email com alfanumèric de 100 caràcters
- vi. birth\_date com a alfanumèric de 50 caràcters (ja que el camp apareix en aquest format al .csv; en cas de necessitar modificar-lo posteriorment sempre ho podem convertir a DATE)
- vii. country com a alfanumèric de 100 caràcters
- viii. city com a alphanumeric de 100 caràcters
- ix. postal\_code com a alfanumèric de 20 caràcters
- x. address com a alfanumèric de 255 (ho creem amb el màxim de caracters ja que és un camp molt variable en funció de les especificacions emprades per registrar una adreça, així ens assegurem que la llargada de l'adreça no serà problema a futur).



The screenshot shows the MySQL Workbench interface with the 'Sprint4\_Load\*' connection selected. In the central pane, the SQL editor contains the following code:

```

22 • CREATE TABLE users(
23     id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
24     name VARCHAR(100),
25     surname VARCHAR(100),
26     phone VARCHAR(50),
27     email VARCHAR(100),
28     birth_date VARCHAR(50),
29     country VARCHAR(100),
30     city VARCHAR(100),
31     postal_code VARCHAR(20),
32     address VARCHAR(255)
33 );

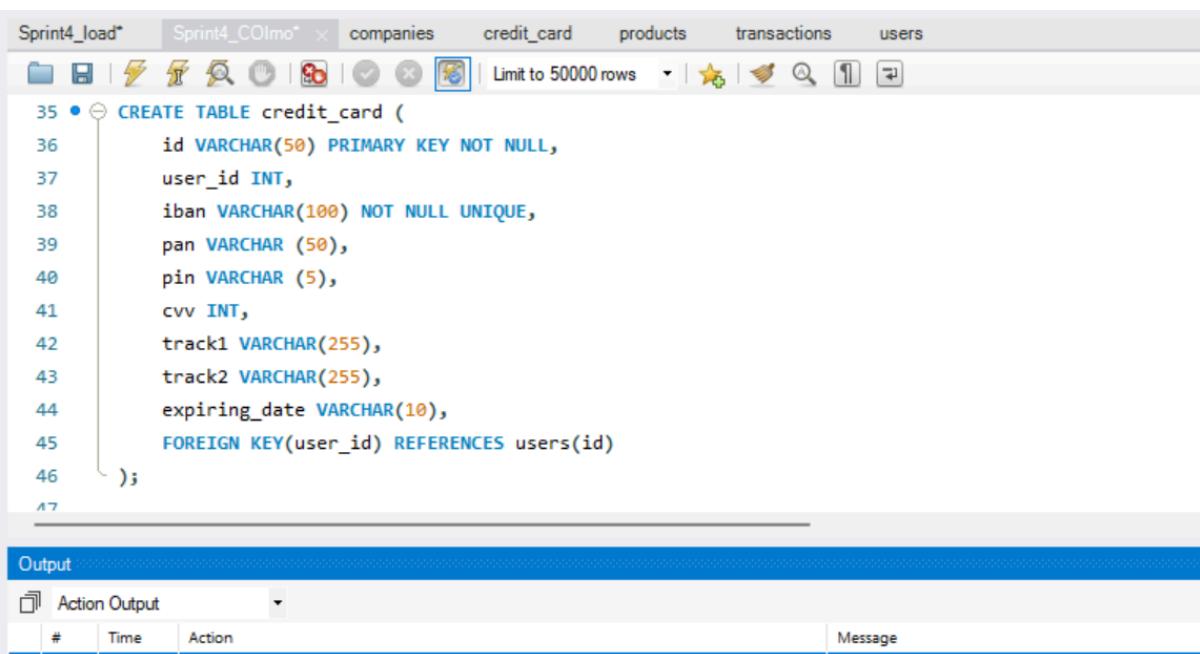
```

The 'Output' tab at the bottom shows the execution log:

#	Time	Action	Message
339	13:39:18	CREATE TABLE users(id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, ...)	0 row(s) affected

- d. Taula 'credit\_card'
- i. id com a alfanumèric de 50 caràcters PK de la taula credit\_card i per tant, camp amb el CONSTRAINT NOT NULL
  - ii. user\_id com a numeral enter, mantenint la integritat de la dada que havíem creat a la taula users
  - iii. iban com a camp alfanumèric de 50 caràcters, ja que tot i estar principalment format per números en ocasions s'introduceix respectant els espais que separen els blocs numèrics de 4 dígits dels bancs, tal com hem observat al .csv
  - iv. pan com a alfanumèric de 50 caràcters
  - v. pin com a alfanumèric de 5 caràcters ja que tot i estar format per números no volem que no detecti el 0 en primera posició

- vi. cvv com a numeral enter
- vii. track1 com a alfanumèric amb el màxim de caràcters
- viii. track2 com a alfanumèric amb el màxim de caràcters
- ix. expiring\_date com a alfanumèric de 10 caràcters (ja que així apareix al .csv; en cas de necessitar posteriorment fer qualsevol transformació podriem convertir to DATE per operar amb funcions de temps. A més observem que al .csv apareix amb una estructura poc usual mm/dd/yy que no correspon amb el sistema europeu ni amb l'anglosaxó sinó que sembla un format personalitzat)
- x. per últim, la taula credit\_card es troba relacionada amb la taula user per mitjà del user\_id que havíem creat anteriorment. Així que establím la relació de FK entre la taula credit\_card i la PK(l'id) de la taula users.



The screenshot shows the MySQL Workbench interface with the 'Sprint4\_Load\*' connection selected. In the top tab bar, there are tabs for 'Sprint4\_COImo\*', 'companies', 'credit\_card', 'products', 'transactions', and 'users'. Below the tabs is a toolbar with various icons for database management. The main area displays the SQL code for creating the 'credit\_card' table:

```

35 • CREATE TABLE credit_card (
36     id VARCHAR(50) PRIMARY KEY NOT NULL,
37     user_id INT,
38     iban VARCHAR(100) NOT NULL UNIQUE,
39     pan VARCHAR(50),
40     pin VARCHAR(5),
41     cvv INT,
42     track1 VARCHAR(255),
43     track2 VARCHAR(255),
44     expiring_date VARCHAR(10),
45     FOREIGN KEY(user_id) REFERENCES users(id)
46 );

```

Below the code, the 'Output' tab is active, showing the execution log:

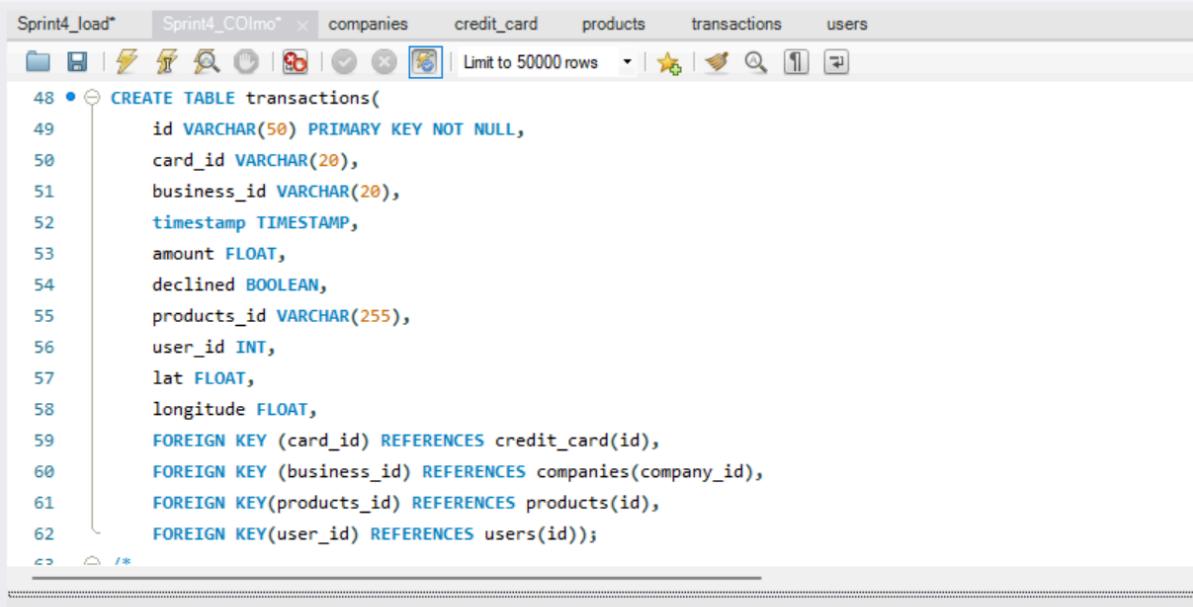
Action Output			
#	Time	Action	Message
340	13:39:18	CREATE TABLE credit_card ( id VARCHAR(50) PRIMARY KEY NOT NULL, user_id INT, iban VARCHAR(100) NOT NULL UNIQUE, pan VARCHAR(50), pin VARCHAR(5), cvv INT, track1 VARCHAR(255), track2 VARCHAR(255), expiring_date VARCHAR(10), FOREIGN KEY(user_id) REFERENCES users(id) );	0 row(s) affected

#### e. Taula 'transactions'

La taula transactions es crea en últim lloc ja que és la nostra taula de fets. És recomenable primer concretar l'estructura de les taules de dimensions, per determinar tots els identificadors -PK- de cada taula, abans de crear la taula de fets que compta amb les FK relatives a les diferents taules de dimensions.

- i. id com a camp alfanumèric de 50 caràcters que consta del CONSTRAINT NOT NULL ja que és la PK de la taula transactions
- ii. card\_id com a alfanumèric de 20 caràcters -mantenint la integritat amb la PK de la taula credit\_card
- iii. business\_id com a alfanumèric de 20 caràcters -mantenint la integritat amb la PK de la taula companies
- iv. timestamp amb format directe timestamp, ja que recull tant la data com l'hora
- v. amount com a numèric decimal
- vi. declined com a boolean ja que només admet els valors Y/N

- vii. products\_id com a camp alfanumèric de 255 caràcters. Aquest camp el tractarem com a cadena de text ja que agrupa els diferents ids dels productes comprats en una mateix transacció separats per comes. Si posteriorment necessitem operar amb aquests camps podem extreure els valors tenint en compte el separador com a valors de text mitjançant la funció SUBSTRING.
- viii. user\_id com a numèric enter -mantenint la integritat amb la PK de la taula user
- ix. lat com a camp numèric decimal
- x. longitude com a camp numèric decimal
- xi. per últim, les diferents FK:
  1. credit\_card amb una relació d'1 a molts (1 credit card, moltes transactions)
  2. companies amb una relació d'1 a molts (1 company, moltes transactions)
  3. products amb una relació d'1 a molts (1 product, moltes transactions, a més de la versió multivalue que trobem per transactions)
  4. users amb una relació d'1 a molts (1 user, moltes transactions)



```

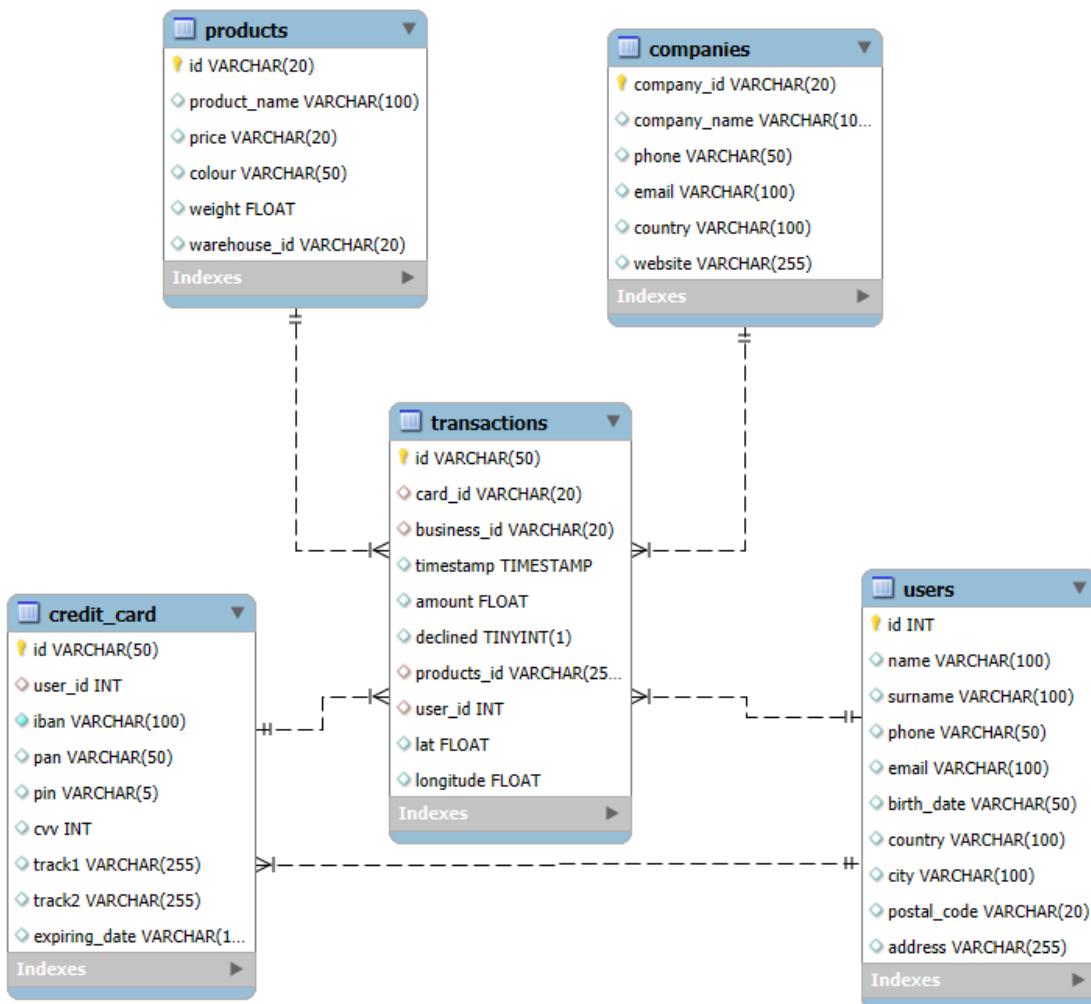
48 • CREATE TABLE transactions(
49     id VARCHAR(50) PRIMARY KEY NOT NULL,
50     card_id VARCHAR(20),
51     business_id VARCHAR(20),
52     timestamp TIMESTAMP,
53     amount FLOAT,
54     declined BOOLEAN,
55     products_id VARCHAR(255),
56     user_id INT,
57     lat FLOAT,
58     longitude FLOAT,
59     FOREIGN KEY (card_id) REFERENCES credit_card(id),
60     FOREIGN KEY (business_id) REFERENCES companies(company_id),
61     FOREIGN KEY(products_id) REFERENCES products(id),
62     FOREIGN KEY(user_id) REFERENCES users(id));
  
```

Output:

#	Time	Action	Message
341	13:39:18	CREATE TABLE transactions(id VARCHAR(50) PRIMARY KEY NOT NULL, ca...	0 row(s) affected

El procés de creació de les diferents taules per a la base de dades SALES, ens dóna com a resultat un model d'estrella, on la taula de fets és la taula transaccions i la resta de taules són taules de dimensions. Aquestes taules es relacionen amb la taula transaccions amb una relació d'1 (users, credit\_cards, products i companies) a molts (registres múltiples en transactions).

Tanmateix existeix una relació directa entre la taula users i credit\_card, mitjançant una relació d'1 (user) a molts (crèdit cards).



Un cop creades les diferents taules i per evitar els CONSTRAINTS de FK que ens creen les relacions entre taules en el moment d'introduir les dades fem servir el SET FOREIGN CHECKS = 0. Per defecte, MYSQL estableix el detector de relacions com a 1, per tant, ens dóna un error quan intentem introduir dades si els camps estan vinculats amb una relació de pares a fills entre taules. Amb la modificació del SET FOREIGN CHECKS eliminem les restriccions temporalment per a poder introduir les dades.

```

    Sprint4_load* | Sprint4_COLmo* | companies | credit_card | products | transactions | users
    1  -- Introducir datos
    2  •  use sales;
    3
    4  •  SET FOREIGN_KEY_CHECKS = 0;
    5
    Output
    Action Output
    # Time Action Message
    342 13:39:22 use sales 0 row(s) affected
    343 13:39:22 SET FOREIGN_KEY_CHECKS = 0 0 row(s) affected
  
```

Un cop fem aquesta modificació podem introduir les dades directament des dels nostres .csv per mitjà del comandament LOAD DATA.

(En aquest punt vaig sofrir el que coneixerem com 'La Odisea del ERROR LOAD DATA', que figura completament detallat a l'Annex i que va consumir una setmana de la meva vida en que tenia malsons amb la carpeta ProgramData, ProgramFiles i l'error 1290. Amb un toc d'humor. ).

## LOAD DATA DETALLAT

Per a introduir les dades dels .csv directament a MYSQL sense fer servir els comandaments màgics apodats Wizard i fer totes les gestions per mitjà de codi, cal emprar el comandament LOAD DATA.

Aquest tipus de comandament consta d'una estructura tipus amb línies de codi estandarditzat:

```
[ LOAD DATA INFILE (on s'especifica d'on obtenim l'arxiu .csv)
  INTO TABLE (on especificuem a quina de les taules creades anteriorment volem volcar les
dades del .csv)
  FIELDS TERMINATED BY (especificuem en format string el tipus de separador que apareix
entre els diferents camps de cada registre. Els mes comuns son ',' o ''
  ENCLOSED BY (el simbol de puntuació que separa cada camp dins d'un mateix registre)
  LINES TERMINATED BY (el tipus de salt de línia que presenta el registre. MySQL treballa amb
els comandaments \n i \r per representar els finals de registre, per tant segons l'últim camp del registre
haurem d'especificar si el salt de línia és només '\n' o '\r\n' - generalment els arxius generats amb
software Microsoft, en aquest cas, un .csv provinent d'Excel, utilitzen la segona opció-)
  IGNORE 1 ROWS; (especificuem si cal ignorar els registres del .csv que apareixin com a
duplicats)
```

```
6 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv'
7   INTO TABLE companies
8   FIELDS TERMINATED BY ','
9   ENCLOSED BY ""
10  LINES TERMINATED BY '\n'
11  IGNORE 1 ROWS;
12
```

Output			
#	Time	Action	Message
344	13:39:22	LOAD DATA INFILE C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/compa...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

*LOAD DATA per a taula Company*

```
13 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv'
14   INTO TABLE products
15   FIELDS TERMINATED BY ','
16   ENCLOSED BY ""
17   LINES TERMINATED BY '\n'
18   IGNORE 1 ROWS;
19
```

Output			
#	Time	Action	Message
345	13:39:22	LOAD DATA INFILE C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/produc...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

*LOAD DATA per a taula Products*

Sprint4\_load\* × Sprint4\_COImo\* companies credit\_card products transactions users

20 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users\_ca.csv'  
 21 INTO TABLE users  
 22 FIELDS TERMINATED BY ','  
 23 ENCLOSED BY '\"'  
 24 LINES TERMINATED BY '\r\n'  
 25 IGNORE 1 ROWS;  
 26  
 27 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users\_uk.csv'  
 28 INTO TABLE users  
 29 FIELDS TERMINATED BY ','  
 30 ENCLOSED BY '\"'  
 31 LINES TERMINATED BY '\r\n'  
 32 IGNORE 1 ROWS;  
 33  
 34 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users\_usa.csv'  
 35 INTO TABLE users  
 36 FIELDS TERMINATED BY ','  
 37 ENCLOSED BY '\"'  
 38 LINES TERMINATED BY '\r\n'  
 39 IGNORE 1 ROWS;

*LOAD DATA per a Taula Users. En aquest cas podem observar que hem hagut de modificar el LINES TERMINATED BY per afegir \r\n. Els arxius relacionats amb aquesta taula havien estat creats a Excel.*

LJL Action Output

#	Time	Action	Message
346	13:39:22	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv'	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0
347	13:39:22	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv'	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0
348	13:39:22	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv'	150 row(s) affected Records: 150 Deleted: 0 Skipped: 0 Warnings: 0

41 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit\_cards.csv'  
 42 INTO TABLE credit\_card  
 43 FIELDS TERMINATED BY ','  
 44 ENCLOSED BY '\"'  
 45 LINES TERMINATED BY '\n'  
 46 IGNORE 1 ROWS;  
 47

Output

LJL Action Output

#	Time	Action	Message
349	13:39:22	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv'	275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings: 0

*LOAD DATA per a la taula credit\_card*

48 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'  
 49 INTO TABLE transactions  
 50 FIELDS TERMINATED BY ';'  
 51 ENCLOSED BY '\"'  
 52 LINES TERMINATED BY '\r\n'  
 53 IGNORE 1 ROWS;  
 54

Output

LJL Action Output

#	Time	Action	Message
350	13:39:22	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0

*LOAD DATA per a la taula transactions. En aquest cas també hem hagut de modificar tant l'element separador de camps, com el salt de línia.*

Un cop finalitzada la càrrega de dades, tornem a configurar el SET FOREIGN CHECKS a la seva configuració per defecte per tal de que mantingui els CONSTRAINTS en afegir noves dades a les taules preexistents.

```
55 • SET FOREIGN_KEY_CHECKS = 1;
```

Output			
Action Output			Message
#	Time	Action	Message
351	13:39:22	SET FOREIGN_KEY_CHECKS = 1	0 row(s) affected

Després d'introduir les dades a les taules que hem creat anteriorment per a la database SALES, procedim a verificar ràpidament que totes les dades han estat introduïdes sense cap incident mitjançant un SELECT \* .

### 1. Companies

Sprint4_load* Sprint4_COImo* companies credit_card products transactions users						
<input type="button" value=""/>	<input type="button" value=""/>	<input type="button" value=""/>	<input type="button" value=""/>	<input type="button" value=""/>	<input type="button" value=""/>	<input type="button" value=""/>
				Limit to 50000 rows		
1 •	SELECT * FROM sales.companies;					

Result Grid   Filter Rows:   Edit:   Export/Import:   Wrap Cell Content:						
company_id	company_name	phone	email	country	website	
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.prttitor.tellus@yahoo.net	Germany	<a href="https://instagram.com/site">https://instagram.com/site</a>	
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia	<a href="https://whatsapp.com/group/9">https://whatsapp.com/group/9</a>	
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	<a href="https://pinterest.com/sub/cars">https://pinterest.com/sub/cars</a>	
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.co.uk	Germany	<a href="https://cnn.com/user/110">https://cnn.com/user/110</a>	
b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	<a href="https://netflix.com/settings">https://netflix.com/settings</a>	

companies 1 x			
Output			
Action Output			Message
352	13:40:18	SELECT * FROM sales.companies LIMIT 0, 50000	100 row(s) returned

## 2. Products

The screenshot shows the MySQL Workbench interface with the 'products' tab selected. A query window displays the command: `SELECT * FROM sales.products;`. Below it, the 'Result Grid' shows the following data:

	id	product_name	price	colour	weight	warehouse_id
▶	1	Direwolf Stannis	\$161.11	#7c7c7c	1	WH-4
	10	Karstark Dorne	\$119.52	#f4f4f4	2.4	WH--5
	100	south duel	\$40.43	#6d6d6d	3	WH--95
	11	Karstark Dorne	\$49.70	#141414	2.7	WH-6
	12	duel Direwolf	\$181.60	#a8a8a8	2.1	WH-7
	13	duel Direwolf	\$181.60	#a8a8a8	2.1	WH-7

The 'Output' pane shows the execution log:

#	Time	Action	Message
354	13:40:23	SELECT * FROM sales.products LIMIT 0, 50000	100 row(s) returned

## 3. Users

The screenshot shows the MySQL Workbench interface with the 'users' tab selected. A query window displays the command: `SELECT * FROM sales.users;`. Below it, the 'Result Grid' shows the following data:

	id	name	surname	phone	email	birth_date	country	city	postal_code	address
▶	1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.edu	Nov 17, 1985	United States	Lowell	73544	348-7818 Sagittis St.
	2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Des Moines	59464	903 Sit Ave
	3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Columbus	56518	736-2063 Tellus St.
	4	Howard	Stafford	1-411-740-3269	ornare.egestas@cloud.edu	Feb 18, 1989	United States	Kailua	77417	Ap #545-2244 Erat. Rd.
	5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	Sep 26, 1998	United States	Sandy	31564	341-2821 Ultrices Av.
	6	Tanya	Thorne	710-200-0000	consectetur.auctor@commodo.com	Oct 15, 1990	United States	Milwaukee	00000	000-0000 Amet Street

The 'Output' pane shows the execution log:

#	Time	Action	Message
355	13:40:25	SELECT * FROM sales.transactions LIMIT 0, 50000	587 row(s) returned
356	13:40:27	SELECT * FROM sales.users LIMIT 0, 50000	275 row(s) returned

#### 4. Credit\_card

## 5. Transactions

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons like file, edit, search, and refresh. A dropdown menu says "Limit to 50000 rows". Below the toolbar, a query window displays the command: "SELECT \* FROM sales.transactions;". The main area is a "Result Grid" showing transaction data with columns: id, card\_id, business\_id, timestamp, amount, declined, products\_id, user\_id, lat, and longitude. The first few rows of data are shown. At the bottom, there's an "Output" section with an "Action Output" table showing two log entries: one for selecting users and one for selecting transactions, both returning the expected number of rows.

Un cop verificat que les dades dels .csv han estat correctament introduïdes, podem començar a realitzar els exercicis.

## - Exercici 1

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

```

4
5 •  SELECT t.user_id, COUNT(t.id) AS total_transactions
6   FROM transactions AS t
7   JOIN users AS u
8     ON t.user_id = u.id
9   GROUP BY t.user_id
10  HAVING total_transactions > 30;
11

```

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

user_id	total_transactions
92	39
267	52
272	76
275	48

Result 2 x

Output:

Action Output

#	Time	Action	Message
1	13:57:33	SELECT t.user_id, COUNT(t.id) AS total_transactions FROM transactions AS t JOIN ...	4 row(s) returned

## - Exercici 2

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

```

Query 1 Sprint4_COLmoQueries* x
File Edit View Insert Tools Options Help
15 •  SELECT ROUND(avg(t.amount),2), cd.iban, c.company_name
16   FROM transactions AS t
17   JOIN credit_card AS cd
18     ON t.card_id = cd.id
19     LEFT JOIN companies AS c
20       ON c.company_id = t.business_id
21   WHERE c.company_name LIKE 'Donec Ltd'
22   GROUP BY 2,3;
23

```

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

ROUND(avg(t.amount),2)	iban	company_name
203.71	PT87806228135092429456346	Donec Ltd

Output:

Action Output

#	Time	Action	Message
1	09:54:53	SELECT ROUND(avg(t.amount),2), cd.iban, c.company_name FROM transactions A...	1 row(s) returned

En aquesta primera query no hem tingut en compte si les transaccions estan o no declinades ja que s'especifica a l'exercici que necessitem el amount. Tanmateix, hem realitzat una segona consulta especificant que les transaccions hagin estat cobrades per a oferir una visió més realista dels pagaments d'aquesta targeta.

Query 1 Sprint4\_COImoQueries\* x

15 • SELECT ROUND(avg(t.amount),2), cd.iban, c.company\_name  
 16 FROM transactions AS t  
 17 JOIN credit\_card AS cd  
 18 ON t.card\_id = cd.id  
 19 LEFT JOIN companies AS c  
 20 ON c.company\_id = t.business\_id  
 21 WHERE c.company\_name LIKE 'Donec Ltd'  
 22 AND t.declined = 'N'  
 23 GROUP BY 2,3;

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

	ROUND(avg(t.amount),2)	iban	company_name
▶	203.71	PT87806228135092429456346	Donec Ltd

Result 3 x

Output:

Action Output

#	Time	Action	Message
1	09:54:53	SELECT ROUND(avg(t.amount),2), cd.iban, c.company_name FROM transactions A...	1 row(s) returned

## Nivell 2

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:

Per a fer aquest exercici hem tingut en compte que les targetes analitzades han de tenir més de 3 transaccions i aquestes transaccions no han estat declinades. Tanmateix, per comprovar si les targetes estan o no actives és necessari consultar la expiring\_date i comparar-la amb la data d'avui. La dada expiring\_date havia estat enregistrada com a alfanumèric amb la peculiaritat que l'ordre dels elements de la data no seguien els formats específics sino que els trobàvem com 'mm/dd/yy', per tant, per poder comparar amb el dia actual fent servir la funció CURDATE (currentDate, més optimitzat que comparar amb la data introduint-la nosaltres ja que hauríem de modificar-la cada cop que fem la consulta) cal transformar el format d'expiring\_date alfanumèric a DATE especificant el format inusual com s'han enregistrat els mesos i els dies.

The screenshot shows the MySQL Workbench interface. In the top panel, there is a query editor titled "Query 1" with the database name "Sprint4\_COlmoQueries" and a table name "credit\_card". The query itself is:

```

31 •  SELECT t.card_id, t.declined, count(t.id) AS transactions_number, cd.expiring_date
32   FROM transactions AS t
33   LEFT JOIN credit_card AS cd
34     ON cd.id = t.card_id
35   WHERE STR_TO_DATE(cd.expiring_date, '%m/%d/%y') > CURDATE()
36   GROUP BY 1,2
37   HAVING transactions_number > 3
38     AND t.declined = 'N'
39   ORDER BY 1;

```

Below the query editor is a "Result Grid" table with four columns: "card\_id", "declined", "transactions\_number", and "expiring\_date". The data is as follows:

	card_id	declined	transactions_number	expiring_date
▶	CcU-4093	0	5	02/13/26
	CcU-4219	0	39	09/15/25
	CcU-4849	0	29	03/21/25

At the bottom of the interface, there is an "Output" section showing the execution log:

- Action Output
 

#	Time	Action	Message
1	10:55:52	SELECT t.card_id, t.declined, count(t.id) AS transactions_number, cd.expiring_date F...	3 row(s) returned

## Exercici 1

Quantes targetes estan actives?

Finalment, la consulta ens torna les 3 úniques targetes actives, amb més de 3 transaccions no declinades, que ens consten al sistema actualment.

## 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.

Aquesta taula va ser creada originàriament durant els procés de creació de la database sales.

Genera la següent consulta:

## Exercici 1

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

Per a resoldre aquest exercici hem de fer una sèrie de pases:

- Creem una taula nova on enmagatzemarem els ids dels productes que s'han enregistrat segons els registres de la taula transaction. Determinem que els articles es registraran com a n (número) i seran un INT.

54 • CREATE TABLE products\_numbers (n INT);

Output		
Action Output	#	Time
CREATE TABLE products_numbers (n INT)	1	12:01:16

Message  
0 row(s) affected

- Com desconeixem quines dades introduir dins d'aquesta taula, establim un sistema de introducció de funcionament similar als bucles FOR de Python (per exemple). Així que creem els valors com a 10 índexes que apareixen com a registres, que posteriorment s'ompliran amb els números dels productes. Hem fet servir aquesta opció perquè desconeixem la quantitat exacta de valors que conté cada registre transaction al camp products\_id.

57 • INSERT INTO products\_numbers (n) VALUES (1), (2), (3), (4), (5), (6), (7), (8), (9), (10);

Output		
Action Output	#	Time
INSERT INTO products_numbers (n) VALUES (1), (2), (3), (4), (5), (6), (7), (8), (9), (10)	1	12:01:49

Message  
10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0

- Elaborem la query utilitzant la funció SUBSTRING\_INDEX que ens permet extreure els valors separats per comes dins d'un valor alfanumèric especificant 1) el camp d'on s'extreuen els valors; 2) el tipus de separador que delimita els valors; 3) on s'emmagatzemen els valors extrets (en el nostre cas com a números o n)
- Per tal d'extreure els valors dins de la cadena de valor, hem d'emprar una segona funció SUBSTRING que extregui el següent valor reconeixent-lo com a -1. D'aquesta forma actua com un comptador i recupera quan hi ha més de 2 valors.
- A més, dins del comandament SELECT afegim el COUNT dels valors extrets, per a que ens digui quants cops s'ha comprat cada producte. Per obtenir aquest recompte fem el COUNT sobre la doble funció SUBSTRING\_INDEX anteriorment descrita.
- Fem una JOIN entre la taula transactions i la taula products\_numbers, que ha enmagatzemat els valors dels productes seguint la substracció per índex i, a més, ens ha fet el recompte. Aquesta nova taula, com desconeix la clau amb la que es vincula a transactions, fa servir el número dels productes sempre que la llargària del número sigui igual o inferior al product\_id, menys la llargària del camp multivalor sense l'enumeració de productes (remplàçant la llargària pels valors extrets un cop desapareixen els separadors).

Query 1 Sprint4\_COLmoQueries\* credit\_card

```

58
59 •  SELECT SUBSTRING_INDEX(SUBSTRING_INDEX(products_id, ',', n), ',', -1) AS product_number,
60      COUNT(SUBSTRING_INDEX(SUBSTRING_INDEX(products_id, ',', n), ',', -1)) AS total_product
61  FROM transactions
62  JOIN products_numbers
63    ON n <= LENGTH(products_id) - LENGTH(REPLACE(products_id, ',', '')) + 1
64  GROUP BY 1
65  ORDER BY 1;
66

```

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

	product_number	total_product
▶	1	39
	11	30
	13	38
	17	35
	19	29

Result 53 ×

Output:

Action Output

#	Time	Action	Message
1	12:02:26	SELECT SUBSTRING_INDEX(SUBSTRING_INDEX(products_id, ',', n), ',', -1) AS pro...	52 row(s) returned

## ANNEX: LA ODISEA DEL ERROR LOAD DATA.

En el moment de començar a fer la importació dels arxius .csv mitjançant la funció LOAD DATA em vaig topar amb l'error 1290. Per motius de seguretat, el problema 1290 ens limita la càrrega de dades desde fora del servidor.

Sprint4\_COLmoQueries\* Sprint4\_load\* Sprint4\_COLmo\*

```

1  -- Introducir datos
2
3 •  LOAD DATA INFILE 'companies.csv'
4  INTO TABLE companies
5  FIELDS TERMINATED BY ','
6  ENCLOSED BY ""
7  LINES TERMINATED BY '\n'
8  IGNORE 1 ROWS;
9
10 • LOAD DATA INFILE 'credit_cards.csv'
11  INTO TABLE credit_card
12  FIELDS TERMINATED BY ','
13  ENCLOSED BY ""
14  LINES TERMINATED BY '\n'
15  IGNORE 1 ROWS;
16

```

Output:

Action Output

#	Time	Action	Message
1	12:28:12	LOAD DATA INFILE 'companies.csv' INTO TABLE companies FIELDS TERMINATED B...	Error Code: 1290. The MySQL server is running with the --secure-file-priv option so it cann.

Comprovem a quina és la carpeta des d'on s'accedeixen als documents per part del nostre server de MySQL amb el comandament SHOW VARIABLES LIKE .

La carpeta que ens apareix és

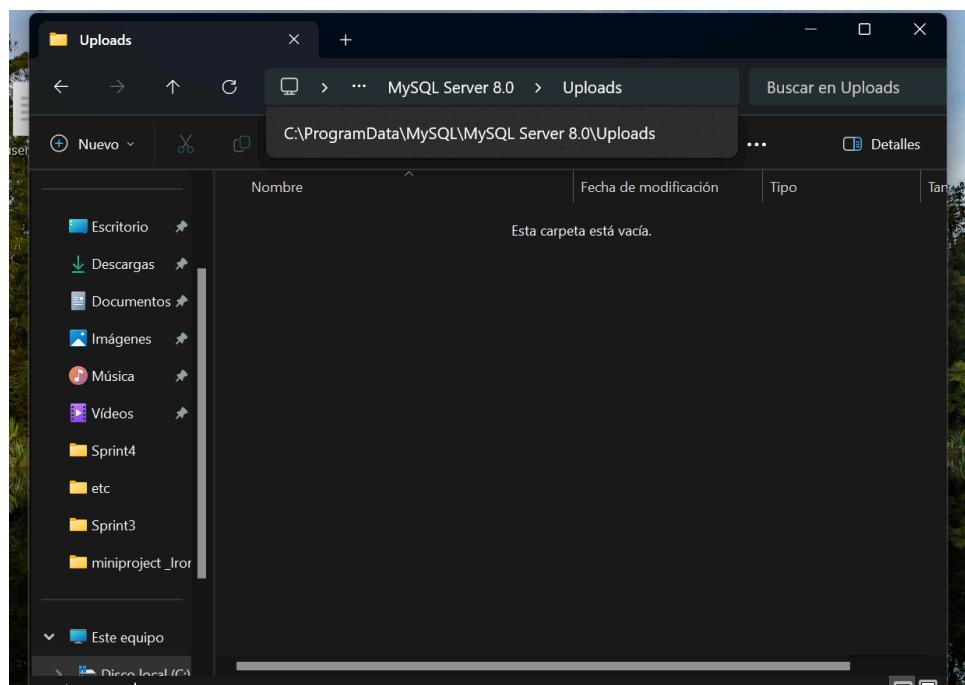
The screenshot shows the MySQL Workbench interface with two tabs open: 'Result Grid' and 'Result 2'. The 'Result Grid' tab displays the output of the query:

Variable_name	Value
secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\

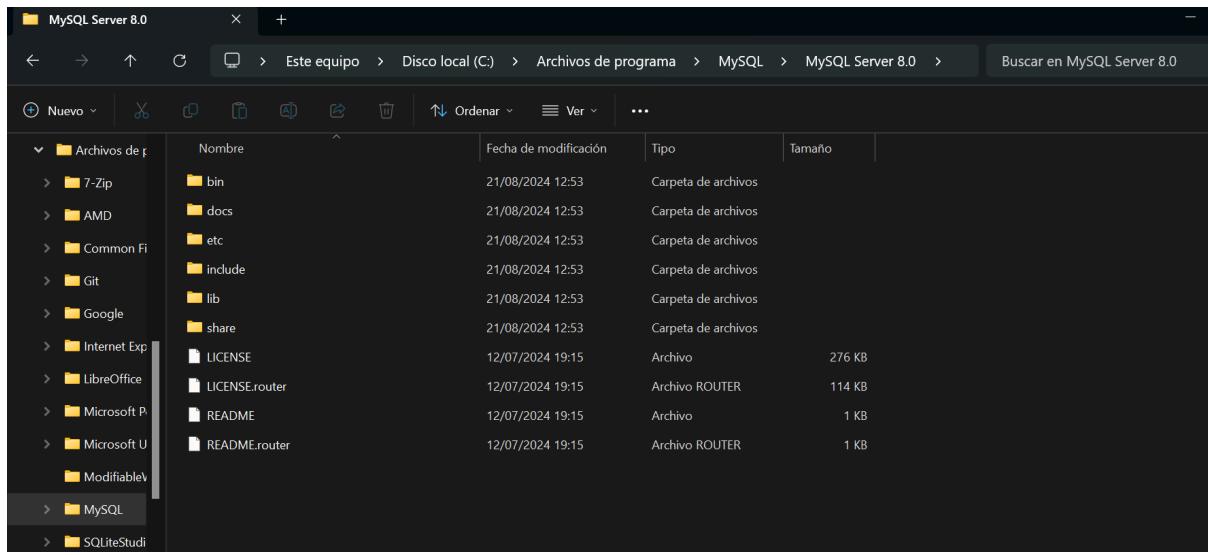
The 'Result 2' tab shows the history of the query execution:

#	Time	Action	Message
1	12:29:40	SHOW VARIABLES LIKE "secure_file_priv"	1 row(s) returned

En primer lloc, vaig seleccionar el path de la carpeta Uploads que m'apareixia a MYSQL, i se m'obria una carpeta en blanc, però a l'hora d'anar enrere dins de la seva ruta de carpeta. Em tornava a l'escriptori. I quan intentava tornar a obrir-la, la carpeta no existia. No aconseguia trobar-la de nou.

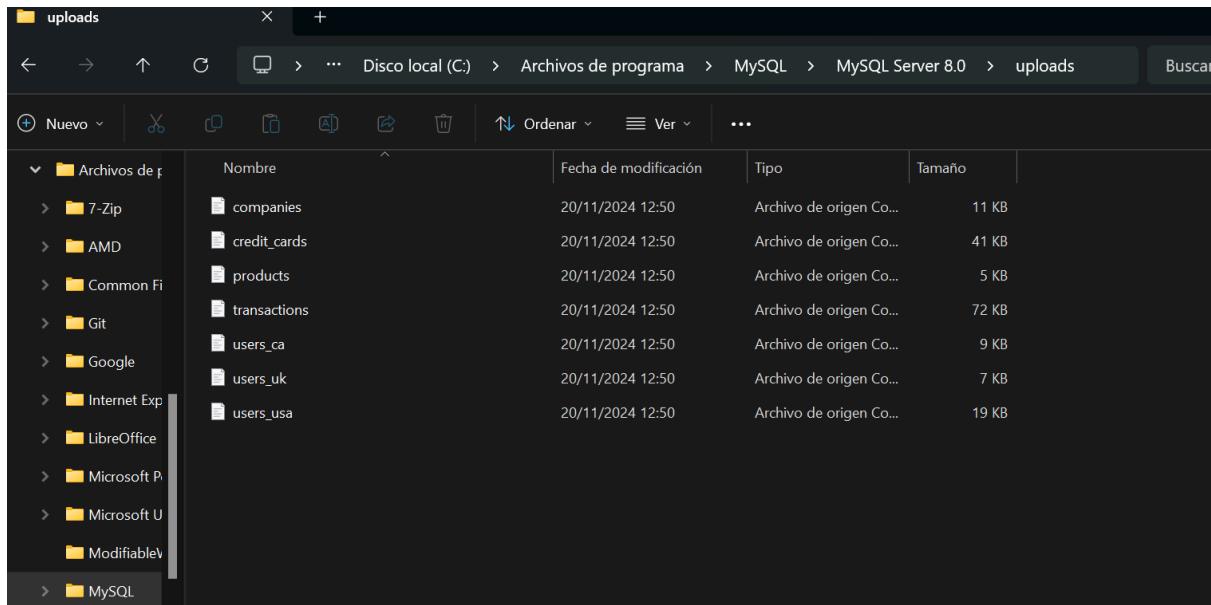


En aquest punt, amb la carpeta fantasma localitzada, vaig explorar els arxius de MySQL, dins de la ruta MySQL Server 8.0 > Uploads, no existeix com a carpeta de ProgramData, ni dins de Program Files.



Arribats a aquest moment de la crisi, he probat varíes formes d'intentar solucionar-lo.

He intentat crear la carpeta uploads amb els documents .csv necessaris per fer l'exercici dins del MYSQL Server 8.0 a la carpeta Program Files però el Workbench no accepta aquests .csv al executar LOAD DATA.



Com el meu 'apaño' no ha funcionat, he revisat a diverses fonts (<https://sincrack.com/solucionando-el-error-mysql-server-is-running-with-the-secure-file-priv/> <https://www.basedash.com/blog/how-to-resolve-the-mysql-server-is-running-with-the-secure-file-priv-option-so-it-cannot-execute-this-statement> [https://www.youtube.com/watch?v=IOYNsyVUdew&ab\\_channel=VictorHugoDelaOMart%C3%ADnez](https://www.youtube.com/watch?v=IOYNsyVUdew&ab_channel=VictorHugoDelaOMart%C3%ADnez)) que m'indiquen que he de fer una modificació en l'arxiu de configuració de MYSQL per canviar la ruta del server.

## How to adjust the --secure-file-priv setting

### VIA CONFIGURATION FILE

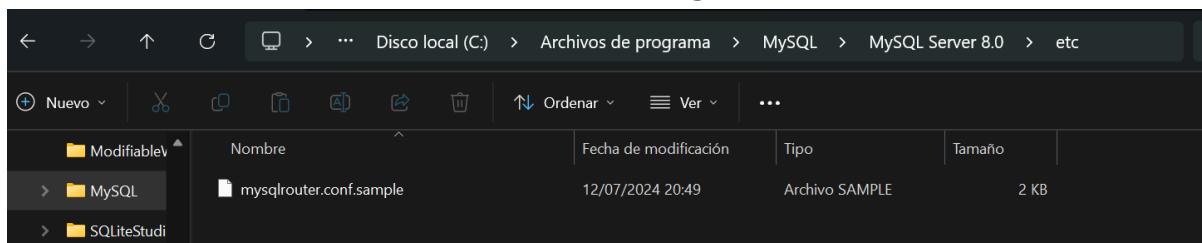
1. Locate your MySQL configuration file (commonly `my.cnf` or `my.ini`).
2. Under the `[mysqld]` section, add or modify the line: Replace `path/to/your/directory` with your desired path or leave it empty to disable the restriction.

```
secure-file-priv="path/to/your/directory"
```
3. Restart the MySQL server to apply changes.

He intentat trobar l'arxiu especificat a les opcions de solució que m'apareixen en diversos blogs i en StackOverflow per a fer la modificació com my.cnf.

- Abre el archivo `my.cnf` (ubicado comúnmente en `/etc/mysql/my.cnf` o dependiendo del sistema operativo).

Però no existeix. No el trobo enllot dins dels arxius de ProgramFiles.



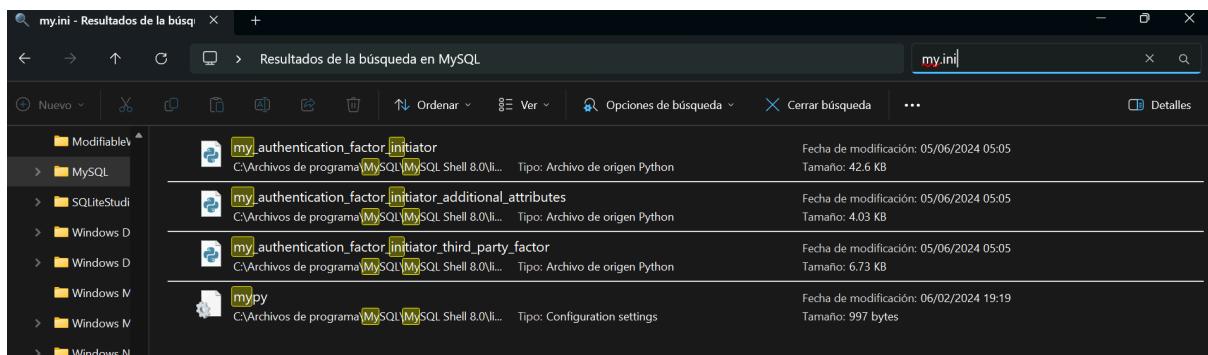
He probat de trobar-lo seguint les indicacions de les possibles ubicacions de l'arxiu que s'especifiquen a la següent captura però no l'he trobat enllot.

Dependiendo de su sistema operativo y el tipo de instalación, el archivo my.cnf generalmente se coloca en una de estas cinco ubicaciones.

```
* /etc/mi.cnf
* /etc/mysql/my.cnf
* $MYSQL_HOME/mi.cnf
* [directorio de datos]/my.cnf
* ~/.mi.cnf
```

L'he cercat per mitjà de consola, consultant les carpetes. Però, res... no apareix.

També he probat amb l'arxiu my.ini però tampoc es troba dins dels arxius de configuració de MySQL.



<https://bugs.mysql.com/bug.php?id=91891>

Desesperada, després de 3 dies amb malsons per la carpeta fantasma, vaig consultar a la mentora Lucía que em va enviar un arxiu d'un fòrum on parlen del mateix problema i fan servir les solucions abans emprades. Tanmateix, en aquest fòrum especificuen unes instruccions que no havia emprat.

bugs.mysql.com/bug.php?id=91891

5. Put the files you want to upload in the /tmp folder

6. LOAD DATA without the LOCAL keyword (i.e., LOAD DATA INFILE 'your\_file' INTO TABLE table\_name ...)

On Windows (a little bit easier):

1. Run the following statement in MySQL Workbench:  
SET GLOBAL local\_infile = 'ON';
2. Run the following statement and make sure it shows ON:  
SHOW VARIABLES LIKE "local\_infile";
3. SHOW VARIABLES LIKE "secure\_file\_priv"; //Then you'll get the default path, usually C:\ProgramData\MySQL\MySQL Server 8.0\Uploads
4. Put your files into the folder shown in step 3
5. LOAD DATA without the LOCAL keyword (i.e., LOAD DATA INFILE 'your\_file' INTO TABLE table\_name ...)

Per tant, verifico els GLOBAL VARIABLES (a veure si és això...)

Sprint4\_COlmoQueries\* Sprint4\_load\* Sprint4\_COlmo\* SQL File 8\* credit\_card

SET GLOBAL local\_infile = 'ON';

Output

Action Output

#	Time	Action	Message
1	14:32:41	SET GLOBAL local_infile = 'ON'	0 row(s) affected

Sprint4\_COImoQueries\* Sprint4\_load\* Sprint4\_COImo\* SQL File 8\* credit\_card

```
1 #SET GLOBAL local_infile = 'ON';
2
3 • SHOW VARIABLES LIKE "local_infile";
```

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

Variable_name	Value
local_infile	ON

Result 1 x

Output

Action Output

#	Time	Action	Message
1	14:32:41	SET GLOBAL local_infile = 'ON'	0 row(s) affected
2	14:34:21	SHOW VARIABLES LIKE "local_infile"	1 row(s) returned

Sprint4\_COImoQueries\* Sprint4\_load\* Sprint4\_COImo\* SQL File 8\* credit\_card

```
1 #SET GLOBAL local_infile = 'ON';
2
3 #SHOW VARIABLES LIKE "local_infile";
4
5 • SHOW VARIABLES LIKE "secure_file_priv";
```

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

Variable_name	Value
secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\

Result 2 x

Output

Action Output

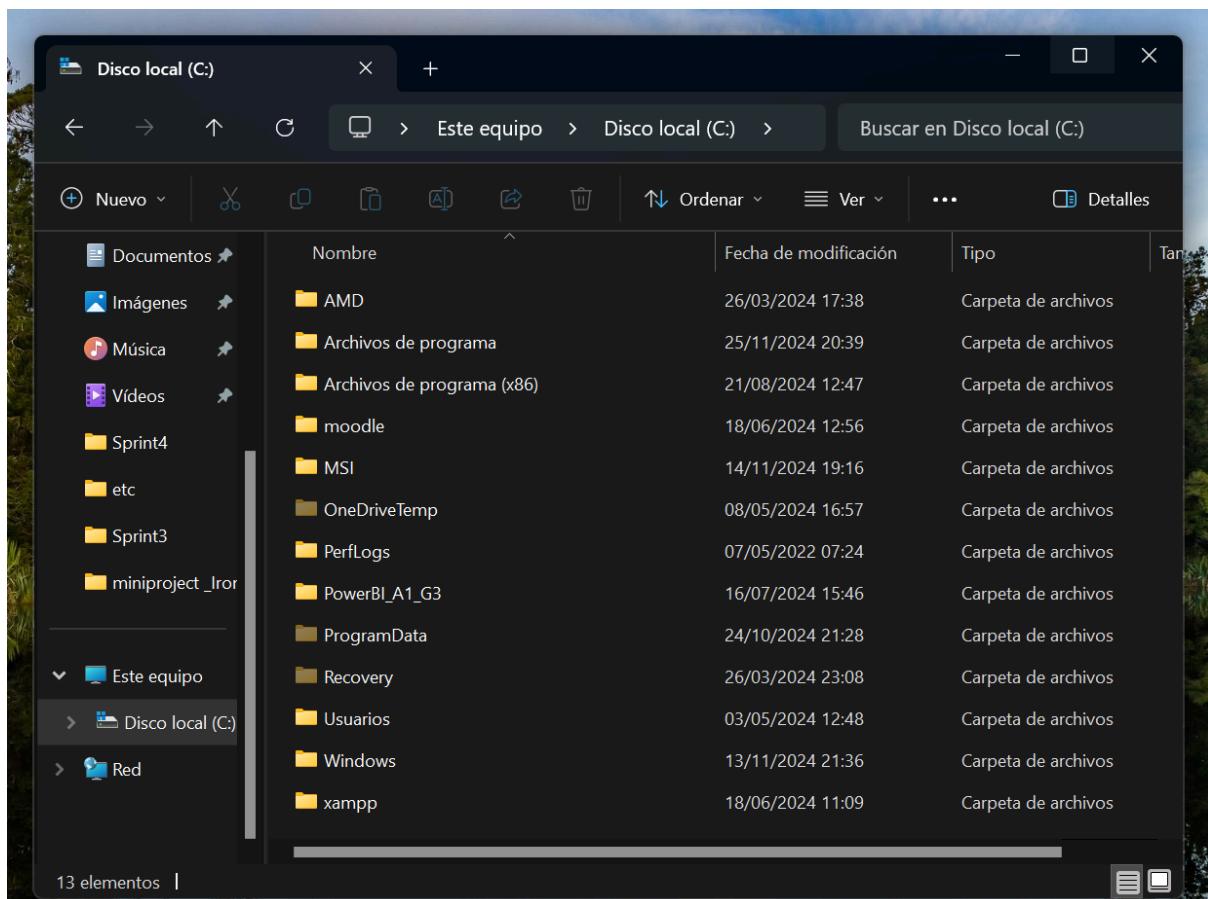
#	Time	Action	Message
1	14:32:41	SET GLOBAL local_infile = 'ON'	0 row(s) affected
2	14:34:21	SHOW VARIABLES LIKE "local_infile"	1 row(s) returned
3	14:35:02	SHOW VARIABLES LIKE "secure_file_priv"	1 row(s) returned

Res, sembla que el MYSQL té tot en regla i sóc jo incapaç de trobar les carpetes fantasmares.

Per últim, després de consultar a diferents companys -alguns em recomenaven que desinstalés el MYSQL per complert i vaig entrar en shock, desconsol i ganes de plorar-, el company Alex m'ha recomenat una solució simple que no se m'havia passat pel cap... deixar totes les carpetes visibles.

I VOILA!!!!

Ha aparegut M.Rajoy, perdò, ProgramData... amb la seva subcarpeta MYSQL SERVER i amb la carpeta fantasma Uploads, junt amb l'arxiu my.ini. Tots juntets!!!



I amb aquesta pausa de ximpresa, sentiment d'inutilitat inclós, després de gairebé una setmana llegint fòrums, StackOverflow, discutint amb CHATGPT i revisant tutorials on NINGÚ MENCIONA QUE AQUESTA CARPETA APAREIX OCULTA PER DEFECTE, he pogut continuar amb l'Sprint4 i retomar la meva vida...

Per cert, la carpeta va fer la seva aparició estelar en el moment que ja desesperada consultava a la mentora, feia 5 minuts que havia tret l'ocult i encara no havia pogut verificar si apareixia o no. Millor moment per saludar i que encara em quedés amb més cara d'al-lucinada.

Però errors així passen perquè encara som humans i no ens ha suplantat la intel·ligència artificial...