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 puguis realitzar les següents consultes:

1. Creé una base de datos llamada sales

```
2 •     CREATE DATABASE IF NOT EXISTS sales;
3 •     USE sales;
4
```

```
19  10:29:43  CREATE DATABASE IF NOT EXISTS sales    1 row(s) affected    0.063 sec
20  10:29:43  USE sales                              0 row(s) affected    0.016 sec
```

2. Revisé la estructura del file companies.csv. El separador es una coma. Los valores únicos que identifican la tabla están en la columna company_id

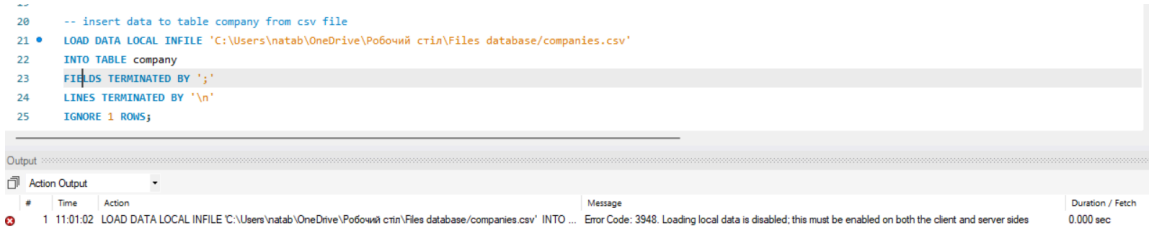
```
company_id,company_name,phone,email,country,website
b-2222,Ac Fermentum Incorporated,06 85 56 52 33,donec.porttitor.tellus@yahoo.net,Germany,https://instagram.com/site
b-2226,Magna A Neque Industries,04 14 44 64 62,risus.donec.nibh@icloud.org,Australia,https://whatsapp.com/group/9
b-2230,Fusce Corp.,08 14 97 58 85,risus@protonmail.edu,United States,https://pinterest.com/sub/cars
b-2234,Convallis In Incorporated,06 66 57 29 50,mauris.ut@aol.couk,Germany,https://cnn.com/user/110
```

3. Creo la tabla company con la PK company_id. Otras columnas tienen el formato VARCHAR. He revisado la longitud de los valores en las columnas en Excel para determinar el tamaño adecuado de VARCHAR.

```
4
5       -- Create table compamy
6 •     CREATE TABLE IF NOT EXISTS company (
7           company_id VARCHAR(15) PRIMARY KEY,
8           company_name VARCHAR(255),
9           phone VARCHAR(15),
10          email VARCHAR(100),
11          country VARCHAR(100),
12          website VARCHAR(255)
13      );
```

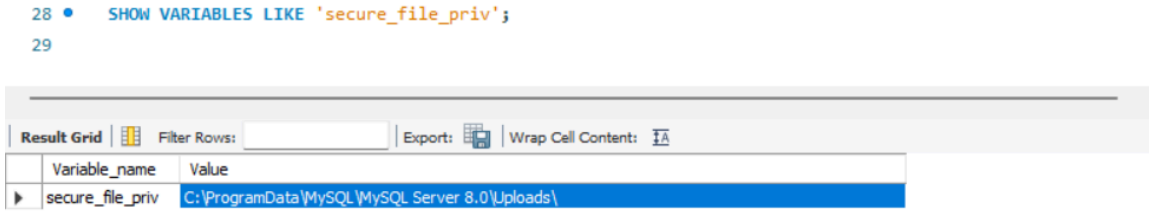
```
Output
Action Output
#   Time       Action                                                                    Message
1   10:37:01   CREATE TABLE IF NOT EXISTS company (  company_id VARCHAR(15) PRIMARY KEY,  company...  0 row(s) affected
```

4. Inserto datos en la tabla company. En la captura de pantalla hay un error: debe ser FIELDS TERMINATED BY ',' en lugar de FIELDS TERMINATED BY ';'



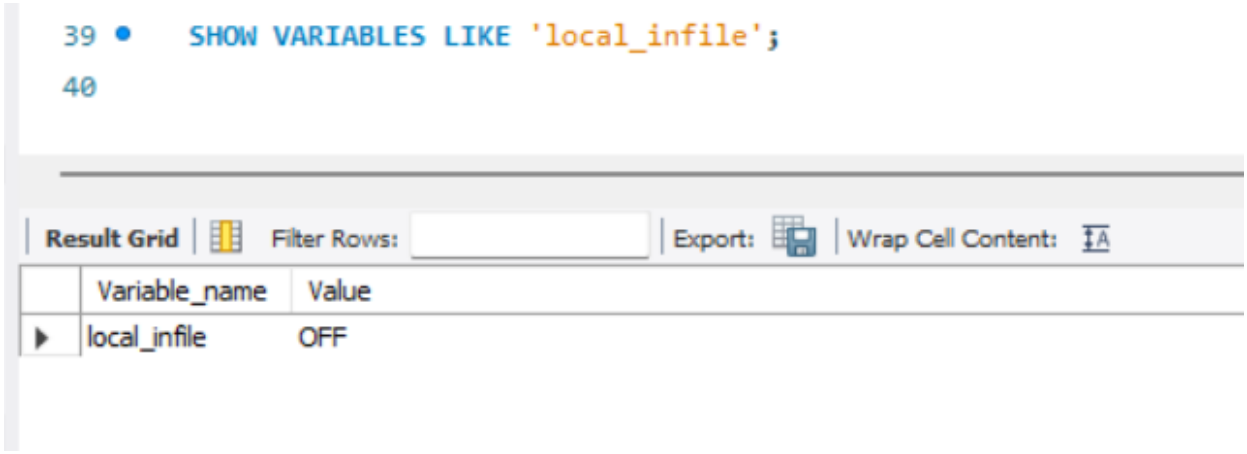
Result error: Error Code: 1290. The MySQL server is running with the --secure-file-priv option so it cannot execute this statement

5. Check the secure file privileged directory



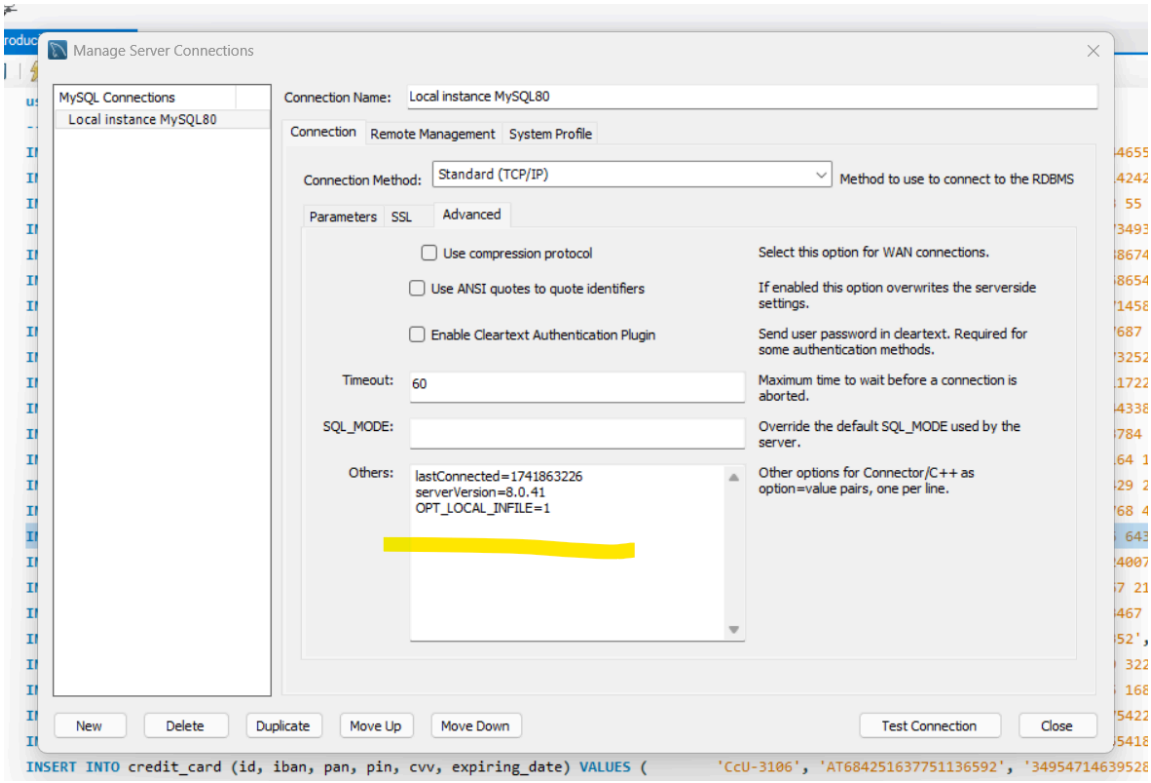
6. Move the csv file to the secure file privileged directory and insert data to the table company again. There is still the same error.

7. Check if local_infile is enabled (on)



8. Enable local_infile

9. Use the solution from [the stackoverflow.](#)
   Edit the connection, on the Connection tab, go to the 'Advanced' sub-tab, and in the 'Others:' box add the line 'OPT_LOCAL_INFILE=1'.



10. Inserto nuevamente los datos en la tabla company y verifica que se hayan agregado correctamente.

```
LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\companies.csv'
INTO TABLE company
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

```
43
44 •  SELECT *
45     FROM company;
46
```

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.porttitor.tellus@yahoo.net	Germany	https://instagram.com/site
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia	https://whatsapp.com/group/9
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/sub/cars
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany	https://cnn.com/user/110
b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings
b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.couk	Norway	https://nytimes.com/user/110
b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
b-2250	Amet Nulla Donec Corporation	07 15 25 14 74	mattis.integer.eu@protonmail.net	Italy	https://netflix.com/sub/cars
b-2254	Nascetur Ridiculus Mus Inc.	06 26 87 61 84	suspendisse.dui@icloud.net	United States	https://ebay.com/sub
b-2258	Vestibulum Lorem PC	02 02 87 33 40	aenean.massa.integer@aol.net	Belgium	https://pinterest.com/sub/cars

company 7 ×

Output

Action Output

#	Time	Action	Message
1	13:07:04	SELECT * FROM company LIMIT 0, 5000	100 row(s) returned

11. Revisé la estructura de los archivos users_ca.csv, users_uk.csv y users_usa.csv.
    El separador de valores es una coma.
    La columna birth_date contiene fechas con coma (ej. " Mar 20, 2000"), por eso están entre comillas dobles para mantenerlas como un solo valor.
    La columna id contiene los identificadores únicos de los usuarios.
    Como tres csv files contienen información sobre la misma dimensión (*user*), los combinaré en una única tabla *user*

```
id,name,surname,phone,email,birth_date,country,city,postal_code,address
201,Iola,Powers,018-139-4717,ante.blandit@outlook.edu,"Mar 20, 2000",Canada,Rigolet,V6T 6M7,154-5415 Auctor St.
202,Maxwell,Holden,045-402-7693,donec@hotmail.edu,"Dec 2, 1986",Canada,Murdochville,S7E 6E0,Ap #880-6372 Ultrices. St.
203,Jarrod,Fields,010-741-8105,sit.amet@google.couk,"Jan 6, 1982",Canada,Baddeck,K3X 6Z5,441-8969 Rhoncus Road
204,Emerson,Sharp,068-138-9383,ante.iaculis@outlook.ca,"Oct 15, 1994",Canada,Maple Creek,Y2C 9E6,"517-6759 Ut, Av."
```

12. Creo la tabla user y verifica que se haya creado correctamente. No estaba segura si debía crear la columna birth_date como DATE o VARCHAR. Entiendo que en Power BI se puede cambiar el formato.

```sql
CREATE TABLE IF NOT EXISTS user (
        id INT PRIMARY KEY,
        name VARCHAR(100),
        surname VARCHAR(100),
        phone VARCHAR(150),
        email VARCHAR(150),
        birth_date VARCHAR(100),
        country VARCHAR(150),
        city VARCHAR(150),
        postal_code VARCHAR(100),
        address VARCHAR(255)
    );
```

Output

Action Output

#	Time	Action	Message
1	13:24:37	CREATE TABLE IF NOT EXISTS user (id INT PRIMARY KEY, name VARCHAR(100), sumam...	0 row(s) affected

```sql
SELECT *
FROM user;
```

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

id	name	surname	phone	email	birth_date	country	city	postal_code	address
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

13. Inserto los datos en la tabla user desde users_usa.csv

```sql
LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\users_usa.csv'
INTO TABLE user
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;
```

14. Inserto los datos en la tabla user desde users_uk.csv

```sql
LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\users_uk.csv'
INTO TABLE user
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS
```

Output

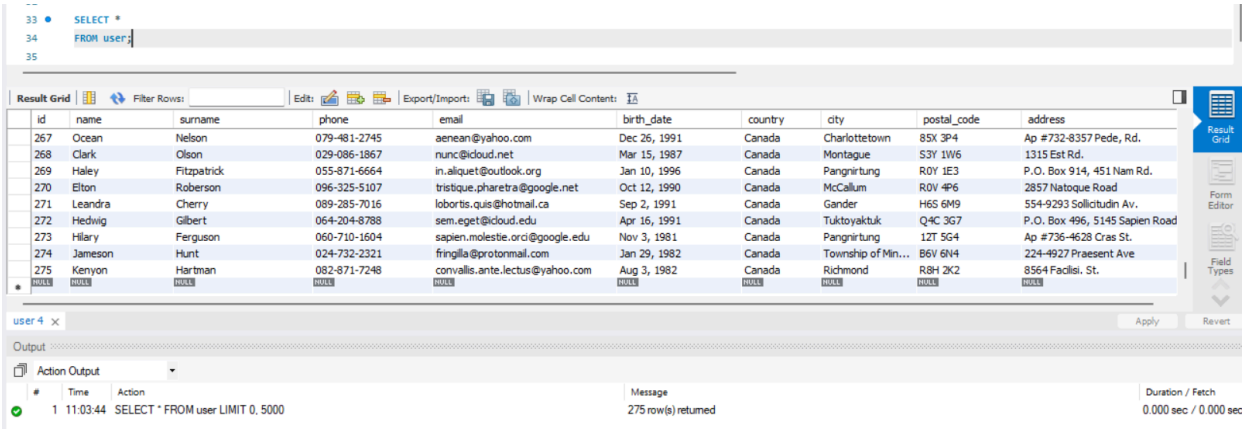
Action Output

#	Time	Action	Message
1	10:57:00	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\users_uk.csv' INTO TABLE ...	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0

15. Inserto los datos en la tabla user desde users_ca.csv

```
54 •  LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\users_ca.csv'
55    INTO TABLE user
56    FIELDS TERMINATED BY ','
57    ENCLOSED BY '"'
58    LINES TERMINATED BY '\r\n'
59    IGNORE 1 ROWS;
```

16. Verifico que todos los datos se hayan añadido en la tabla user.



17. Cambié la columna id para que sea autoincremental. Así, cada nuevo usuario recibe automáticamente un número único y no necesito asignarlo manualmente

```
58 •  ALTER TABLE user MODIFY id INT NOT NULL AUTO_INCREMENT;
59
```

18. Revisé la estructura de credit_cards.csv.
    El separador de valores es una coma.
    La columna id contiene los identificadores únicos de credit cards.

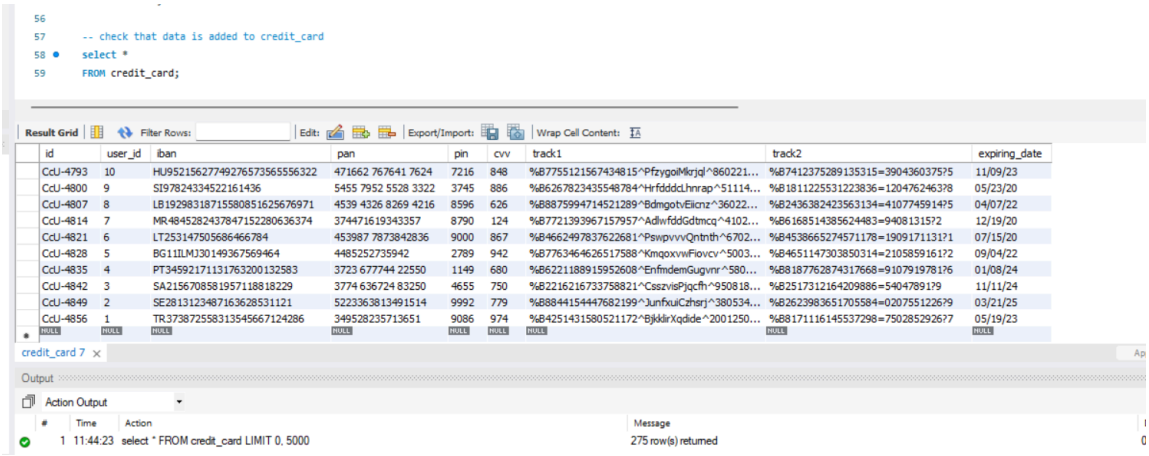
```
id,user_id,iban,pan,pin,cvv,track1,track2,expiring_date
CcU-2938,275,TR301950312213576817638661,5424465566813633,3257,984,%B838371244855464 6^WovsxejDpwiev^86041142?7,%B7653863056044187=
8007163336?3,10/30/22
CcU-2945,274,DO26854763748537475216568689,5142423821948828,9080,887,%B4621311609958661^UftuyfsSeimxn^0610628241?7,%B4149568437843501=
5107140330?1,08/24/23
CcU-2952,273,BG45IVQL52710525608255,4556 453 55 5287,4598,438,%B2183285104307501^CddyytcUxwfdq^5907955430?9,%B6778580257827162=
```

19. Creo la tabla credit_card

```sql
CREATE TABLE IF NOT EXISTS credit_card (
id VARCHAR(15) PRIMARY KEY,
user_id INT,
iban  VARCHAR(35),
pan VARCHAR(20),
pin INT,
cvv INT,
track1 VARCHAR(255),
track2 VARCHAR(255),
expiring_date VARCHAR(8)
);
```

20. Inserto los datos en la tabla credit_card desde credit_cards.csv

21. Verifico que todos los datos se hayan añadido en la tabla credit_card



22. Revisé la estructura de products.csv
    El separador de valores es una coma.
    La columna id contiene los identificadores únicos de los productos.
    Los precios tienen un símbolo de dólar. Además, como representan valores monetarios, pueden tener un máximo de dos decimales después de la coma.

```
x       x       id,product_name,price,colour,weight,warehouse_id
1,Direwolf Stannis,$161.11,#7c7c7c,1,WH-4
2,Tarly Stark,$9.24,#919191,2,WH-3
3,duel tourney Lannister,$171.13,#d8d8d8,1.5,WH-2
4,warden south duel,$71.89,#111111,3,WH-1
5,skywalker ewok,$171.22,#dbdbdb,3.2,WH-0
6,dooku solo,$136.60,#c4c4c4,0.8,WH--1
7,north of Casterly,$63.33,#b7b7b7,0.6,WH--2
8,Winterfell,$32.37,#383838,1.4,WH--3
```

23. Creo la tabla product.

```
61      -- create table product
62  •   CREATE TABLE IF NOT EXISTS product (
63      id INT AUTO_INCREMENT PRIMARY KEY,
64      product_name  VARCHAR(300),
65      price DECIMAL(10,2),
66      colour VARCHAR(10),
67      weight DECIMAL(6,2),
68      warehouse_id VARCHAR(10)
69      );
70
```

```
Output
Action Output
#   Time      Action                                                                                  Message
1   12:11:35  CREATE TABLE IF NOT EXISTS product ( id INT AUTO_INCREMENT PRIMARY KEY, product_name  VARCH...  0 row(s) affected
```

24. Inserta datos en la tabla product.
    Uso SET price = REPLACE(price, '$', '') + 0 para reemplazar el símbolo de dólar ($) con una cadena vacía ('') y sumo 0 para convertirlo en un valor decimal.

    Hay un error:: Error Code: 1366. Incorrect decimal value: '$161.11' for column 'price' at row 1.
    Este error ocurre porque REPLACE() solo funciona con datos de tipo char, varchar.

```
70
71      -- insert data to product
72  •   LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv'
73      INTO TABLE product
74      FIELDS TERMINATED BY ','
75      LINES TERMINATED BY '\n'
76      IGNORE 1 ROWS
77      (id, product_name, price, colour, weight, warehouse_id)
78      SET price = REPLACE(price, '$', '') + 0;
79
```

```
Output
Action Output
#   Time      Action                                                                                  Message                                                                                  Duration / Fetch
1   12:18:11  LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv' INTO TABLE ...  Error Code: 1366. Incorrect decimal value: '$161.11' for column 'price' at row 1  0.000 sec
```

25. Para poder reemplazar el símbolo de dólar cambio tipo de datos de price a VARCHAR(50)

```
80 •    ALTER TABLE product MODIFY COLUMN price VARCHAR(50);
81
```

Output

Action Output

#	Time	Action	Message
1	12:18:11	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv' INTO TABLE ...	Error Code: 1366. Incorrect decimal value: '$161.11' for column 'price' at row 1
2	12:20:13	ALTER TABLE product MODIFY COLUMN price VARCHAR(50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

26. Inserto otra vez datos a product. Esta vez sin errores.

```
84 •    LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv'
85      INTO TABLE product
86      FIELDS TERMINATED BY ','
87      LINES TERMINATED BY '\n'
88      IGNORE 1 ROWS
89      (id, product_name, price, colour, weight, warehouse_id)
90      SET price = REPLACE(price, '$', '') + 0;
```

Output

Action Output

#	Time	Action	Message
1	12:44:52	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv' INTO TABLE pro...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

27. Cambio el tipo de dato de la columna price a decimal

```
93 •    ALTER TABLE product MODIFY COLUMN price DECIMAL(10,2);
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	12:49:50	ALTER TABLE product MODIFY COLUMN price DECIMAL(10,2)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0	0.094 sec

28. En lugar de usar el símbolo $ dentro del precio, agregó una nueva columna llamada currency. Esta columna guarda la información de la moneda como texto (USD), para saber que todos los precios están en dólares.

```sql
ALTER TABLE product ADD COLUMN currency VARCHAR(3) DEFAULT 'USD';

select *
FROM product;
```

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

id	product_name	price	colour	weight	warehouse_id	currency
1	Direwolf Stannis	161.11	#7c7c7c	1.00	WH-4	USD
2	Tarly Stark	9.24	#919191	2.00	WH-3	USD
3	duel tourney Lannister	171.13	#d8d8d8	1.50	WH-2	USD
4	warden south duel	71.89	#111111	3.00	WH-1	USD
5	skywalker ewok	171.22	#dbdbdb	3.20	WH-0	USD
6	dooku solo	136.60	#c4c4c4	0.80	WH--1	USD
7	north of Casterly	63.33	#b7b7b7	0.60	WH--2	USD
8	Winterfell	32.37	#383838	1.40	WH--3	USD
9	Winterfell	76.40	#b5b5b5	1.20	WH--4	USD
10	Karstark Dorne	119.52	#f4f4f4	2.40	WH--5	USD
11	Karstark Dorne	49.70	#141414	2.70	WH--6	USD

product 2

Output

Action Output

#	Time	Action	Message
1	12:49:50	ALTER TABLE product MODIFY COLUMN price DECIMAL(10,2)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0
2	12:52:21	ALTER TABLE products ADD COLUMN currency VARCHAR(3) DEFAULT 'USD'	Error Code: 1146. Table 'sales.products' doesn't exist
3	12:52:32	ALTER TABLE product ADD COLUMN currency VARCHAR(3) DEFAULT 'USD'	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
4	12:53:02	select * FROM product LIMIT 0, 5000	100 row(s) returned

29. Revisé la estructura de transactions.csv
    El separador de valores es ;.
    La columna id contiene los identificadores únicos de los productos.

```
id;card_id;business_id;timestamp;amount;declined;product_ids;user_id;lat;longitude
108B1D1D-5B23-A76C-55EF-C568E49A05DD;CcU-2938;b-2222;2021-07-07 17:43:16;293.57;0;59;275;83.7839152128;-178.860353536
7DC26247-20EC-53FE-E555-B6C2E55CA5D5;CcU-2945;b-2226;2022-02-04 15:52:56;312.5;0;71, 41;275;58.9367181312;-76.8171099136
72997E96-DC2C-A4D7-7C24-66C302F8AE5A;CcU-2952;b-2230;2022-01-30 15:16:36;239.87;0;97, 41, 3;275;43.3584055296;-17.6579677184
AB069F53-965E-A2A8-CE06-CA8C4FD92501;CcU-2959;b-2234;2021-04-15 13:37:18;60.99;0;11, 13, 61, 29;275;1.6481916928;-158.0065729536
```

30. Creo la tabla transaction. La tabla referencia las tablas credit_card, company, user (FKs)

```sql
-- Creamos la tabla transaction
CREATE TABLE IF NOT EXISTS transaction (
    id VARCHAR(255) PRIMARY KEY,
    credit_card_id VARCHAR(15),
    company_id VARCHAR(15),
    user_id INT,
    lat DECIMAL (12,10),
    longitude DECIMAL(13,10),
    timestamp TIMESTAMP,
    amount DECIMAL(10, 2),
    declined BOOLEAN,
    product_ids VARCHAR(100),
    FOREIGN KEY (credit_card_id) REFERENCES credit_card(id),
    FOREIGN KEY (company_id) REFERENCES company(company_id),
    FOREIGN KEY (user_id) REFERENCES user(id)
);
```

Output

Action Output

#	Time	Action	Message
1	10:56:51	CREATE TABLE IF NOT EXISTS transaction (id VARCHAR(255) PRIMARY KEY, credit_card_id ...	0 row(s) affected

31. Inserto los datos en la tabla transaction.

El orden de las columnas en el CSV es diferente al de la tabla transaction, por eso en LOAD DATA INFILE especifiqué el orden explícitamente entre paréntesis

```
130 •  LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\transactions.csv'
131    INTO TABLE transaction
132    FIELDS TERMINATED BY ';'
133    LINES TERMINATED BY '\n'
134    IGNORE 1 ROWS
135    (id, credit_card_id, company_id, timestamp, amount, declined, product_ids, user_id, lat, longitude);
136
137 •  select *
138    from transaction;
139
```

```
Output
Action Output
#  Time      Action                                                                                  Message
1  11:02:27  LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\transactions.csv' INTO TA...  587 row(s) affected Records: 587  Deleted: 0  Skipped: 0  Warnings: 0
```

La columna product_ids tiene varios ids de productos en una misma transacción, pero para organizar mejor los datos, hay que normalizarla. Por eso, creo una tabla intermedia (transaction_product) donde cada fila tendrá solo un transaction_id y un product_id. Así, cada producto se almacena por separado y es más fácil hacer consultas en SQL.

id	product_ids
02C6201E-D90A-1859-B4EE-88D2986D3B02	71, 1, 19
0466A42E-47CF-8D24-FD01-C0B689713128	47, 97, 43
063FBA79-99EC-66FB-29F7-25726D1764A5	47, 67, 31, 5
0668296C-CDB9-A883-76BC-2E4C44F8C8AE	89, 83, 79
06CD9AA5-9B42-D684-DDDD-A5E394FEBA99	43, 31

32. Create transaction_product table.

    Esta tabla intermedia transaction_product normaliza la relación muchos a muchos entre transacciones y productos. La clave primaria compuesta (PRIMARY KEY (transaction_id, product_id)) garantiza que cada combinación de transacción y producto sea única, evitando duplicados. Además, las claves foráneas aseguran que transaction_id exista en la tabla transaction y product_id en la tabla product, manteniendo la integridad de los datos.

```
142 •  CREATE TABLE IF NOT EXISTS transaction_product(
143    transaction_id VARCHAR(255),
144    product_id  INT,
145    PRIMARY KEY (transaction_id, product_id), -- composite PK
146    FOREIGN KEY (transaction_id) REFERENCES transaction(id),
147    FOREIGN KEY (product_id) REFERENCES product(id)
148    );
149
150
151
```

```
Output
Action Output
#  Time      Action                                                                                  Message
1  11:05:43  CREATE TABLE IF NOT EXISTS transaction_product( transaction_id VARCHAR(255), product_id  INT, PRI...  0 row(s) affected
```

33. Inserto los datos en la tabla transaction_product table

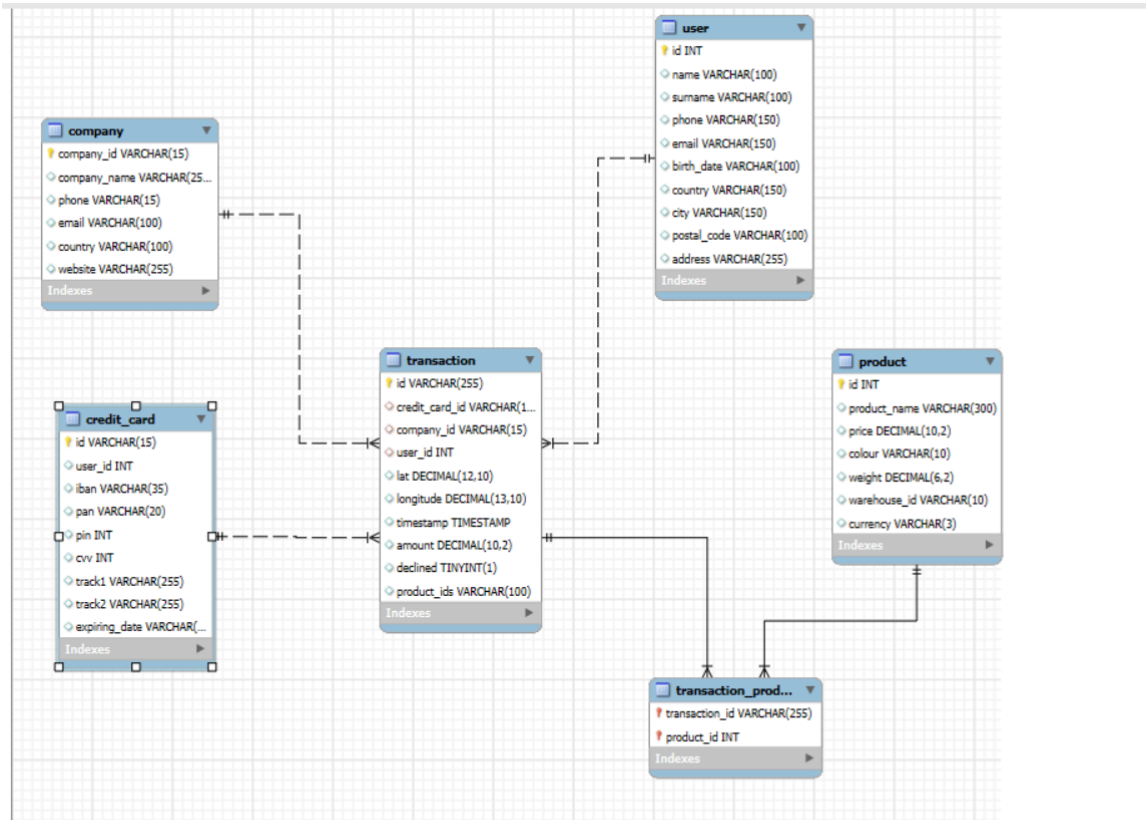
```
151 •   INSERT INTO transaction_product (transaction_id, product_id)
152     SELECT t.id, p.id
153     FROM transaction t
154     JOIN product p ON FIND_IN_SET(p.id, REPLACE (t.product_ids,' ','' )) > 0;
155
156
157
```

Output

Action Output

#	Time	Action	Message
1	11:05:43	CREATE TABLE IF NOT EXISTS transaction_product(transaction_id VARCHAR(255), product_id INT, PRI...	0 row(s) affected
2	11:15:17	INSERT INTO transaction_product (transaction_id, product_id) SELECT t.id, p.id FROM transaction t JOIN ...	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0

Final diagram:



Estructura del Star Schema en esta base de datos:

- Fact table (transaction)
  - Es el centro del esquema y almacena eventos clave - transacciones.
  - Se conecta con tablas de dimensión company, credit_card, user a través de claves foráneas.
- Dimension tables
  - User
  - Company
  - Credit_card
  - product

- Tabla intermedia (transaction_product) qu normaliza la relación entre transaction y product.

- Exercici 1

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

Primero miro la tabla user y, para cada usuario, reviso en la tabla transacción si hay transacciones con el mismo user_id. Agrupo las transacciones por usuario y cuento cuántas tiene cada uno. Si tiene más de 30, ese usuario se incluye en el resultado, porque WHERE EXISTS se cumple. Al final, ordeno el resultado por el id del usuario.

```
152 •  SELECT u.id, u.name, u.surname
153    FROM user u
154    WHERE EXISTS (
155        SELECT 1
156        FROM transaction t
157        WHERE t.user_id = u.id
158        GROUP BY t.user_id
159        HAVING COUNT(*) > 30
160    )
161    ORDER BY u.id;
162
```

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

id	name	surname
92	Lynn	Riddle
267	Ocean	Nelson
272	Hedwig	Gilbert
275	Kenyon	Hartman
NULL	NULL	NULL

user 24

Output

Action Output

#	Time	Action	Message
1	12:02:33	SELECT u.id, u.name, u.surname FROM user u WHERE EXISTS (SELECT 1 FROM transaction t ...	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.

Shows the average amount per IBAN of credit cards in the company Donec Ltd, use at least 2 tables.

Uno las tablas company, transaction y credit_card y filtro solo las transacciones de la empresa Donec Ltd. Después, GROUP BY company_id, credit_card_id, iban y cálculo avg amount de las transacciones para cada IBAN. Como el formato de amount en la tabla transaction es decimal(10,2), uso RUND para devolver AVG(amount).

```sql
SELECT
    t.company_id,
    t.credit_card_id,
    c.company_name,
    cc.iban,
    ROUND(AVG(t.amount),2) AS avg_amount
FROM company c
JOIN transaction t ON c.company_id = t.company_id
JOIN credit_card cc ON cc.id = t.credit_card_id
WHERE c.company_name = "Donec Ltd"
GROUP  BY t.company_id, t.credit_card_id, cc.iban;

CREATE TABLE credit_card_status_history (
```

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

company_id	credit_card_id	company_name	iban	avg_amount
b-2242	CcU-2973	Donec Ltd	PT87806228135092429456346	203.72

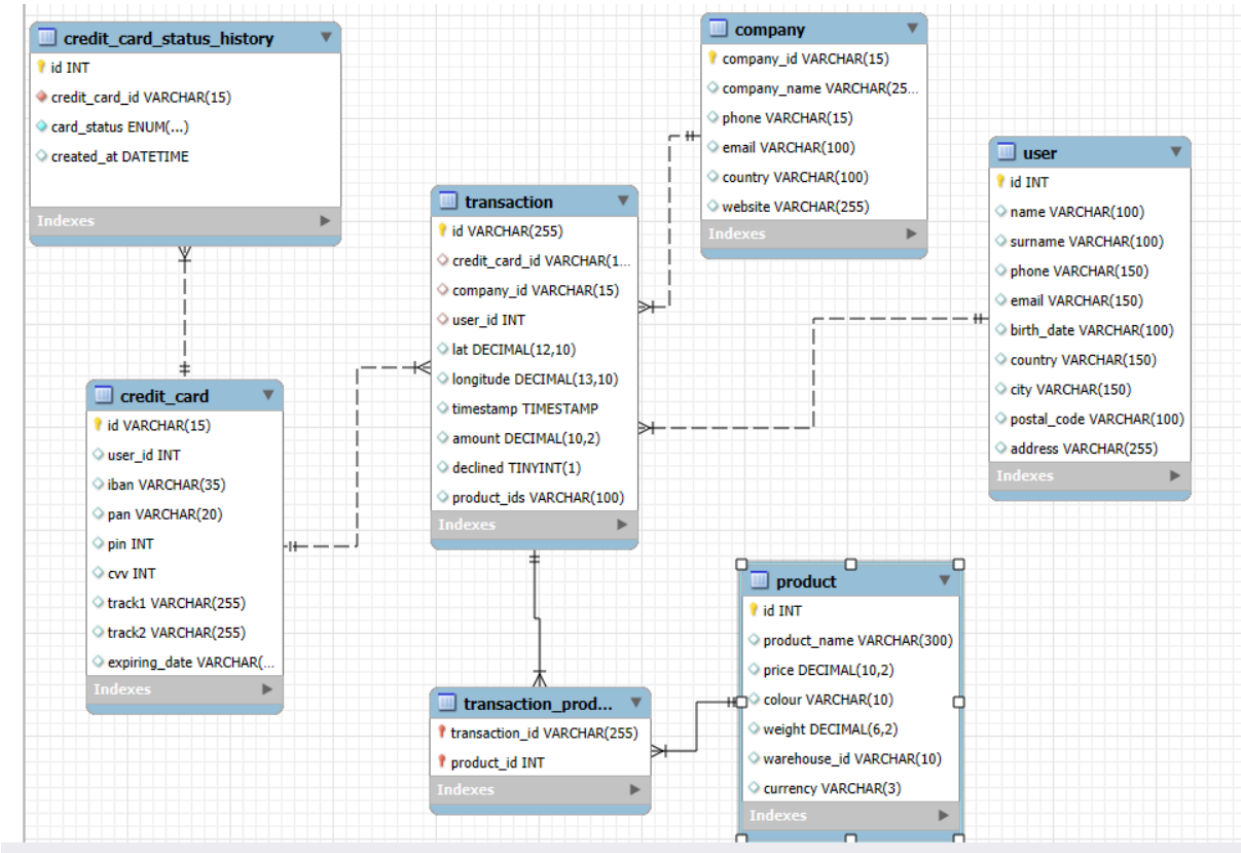
Nivell 2

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:

Credit_card_status_history guarda el historial de cambios de estado para cada tarjeta de crédito.
Se añade una nueva fila cada vez que se crea una tarjeta o cuando cambia su estado (Activo/Inactivo).
Una tarjeta puede aparecer múltiples veces en esta tabla. El campo created_at registra cuándo ocurrió cada cambio. Entonces para obtener el estado actual de una tarjeta, se puede filtrar por craeted_at más reciente.
Ahora cada tarjeta de crédito tendrá un solo estado, ya que es la primera vez que se registra el estado y aún no se han producido cambios.

```sql
CREATE TABLE credit_card_status_history (
    id INT AUTO_INCREMENT PRIMARY KEY,
    credit_card_id VARCHAR(15) NOT NULL,
    card_status ENUM('Active', 'Inactive') NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (credit_card_id) REFERENCES credit_card(id) ON DELETE CASCADE
);
```

Diagrama después de crear la tabla credit_card_status_history.



Inserto los datos en la tabla desde select.

- CTE ranked_transactions: asigna un número de orden a cada transacción por tarjeta de crédito, ordenándolas de más reciente a más antigua. Rank () Over asigna el mismo número a operaciones que se realizan al mismo tiempo.
- SELECT: verifica si las transacciones más recientes (con rank <=3) de una tarjeta fueron rechazadas (declined >= 3). Si esto ocurre, la tarjeta se marca como "Inactive", de lo contrario, se mantiene como "Active".

```
184 •  INSERT INTO credit_card_status_history (credit_card_id, card_status)
185    WITH ranked_transactions AS (
186    SELECT credit_card_id, timestamp, declined, RANK() OVER(PARTITION BY credit_card_id ORDER BY timestamp DESC) AS transaction_rank
187    FROM transaction
188    )
189    SELECT
190    credit_card_id,
191    CASE WHEN SUM(declined) >= 3 THEN "Inactive" ELSE "Active" END AS status
192    FROM ranked_transactions
193    WHERE transaction_rank <= 3
194    GROUP BY credit_card_id;
195
```

Output

Action Output

#	Time	Action	Message
1	12:48:50	INSERT INTO credit_card_status_history (credit_card_id, card_status) WITH ranked_transactions AS (SELE...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

```
155
156 •  SELECT *
157    FROM credit_card_status_history
158    LIMIT 10;
159
160 •  SELECT COUNT(*) as active_cards_count
```

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

id	credit_card_id	card_status	created_at
1	CcU-2938	Active	2025-03-26 12:48:50
2	CcU-2945	Active	2025-03-26 12:48:50
3	CcU-2952	Active	2025-03-26 12:48:50
4	CcU-2959	Active	2025-03-26 12:48:50
5	CcU-2966	Active	2025-03-26 12:48:50

credit_card_status_history 2

Output

Action Output

#	Time	Action	Message
1	20:28:38	SELECT * FROM credit_card_status_history LIMIT 10	10 row(s) returned

Exercici 1

Quantes targetes estan actives?

```
159
160 •  SELECT COUNT(*) as active_cards_count
161    FROM credit_card_status_history
162    WHERE card_status = "Active";
163
```

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

active_cards_count
275

Result 3

Output

Action Output

#	Time	Action	Message
1	20:29:11	SELECT COUNT(*) as active_cards_count FROM credit_card_status_history WHERE card_status = "Active...	1 row(s) returned

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:

- Esta tabla ya fue creada en el ejercicio anterior. La tabla transaction_product ya conecta las transacciones con los productos mediante product_id.

1. Revisé la estructura de products.csv
   El separador de valores es una coma.
   La columna id contiene los identificadores únicos de los productos.
   Los precios tienen un símbolo de dólar. Además, como representan valores monetarios, pueden tener un máximo de dos decimales después de la coma.

```
x       x       id,product_name,price,colour,weight,warehouse_id
1,Direwolf Stannis,$161.11,#7c7c7c,1,WH-4
2,Tarly Stark,$9.24,#919191,2,WH-3
3,duel tourney Lannister,$171.13,#d8d8d8,1.5,WH-2
4,warden south duel,$71.89,#111111,3,WH-1
5,skywalker ewok,$171.22,#dbdbdb,3.2,WH-0
6,dooku solo,$136.60,#c4c4c4,0.8,WH--1
7,north of Casterly,$63.33,#b7b7b7,0.6,WH--2
8,Winterfell,$32.37,#383838,1.4,WH--3
```

2. Creo la tabla product.

```
61      -- create table product
62      CREATE TABLE IF NOT EXISTS product (
63      id INT AUTO_INCREMENT PRIMARY KEY,
64      product_name  VARCHAR(300),
65      price DECIMAL(10,2),
66      colour VARCHAR(10),
67      weight DECIMAL(6,2),
68      warehouse_id VARCHAR(10)
69      );
70
```

```
Output
Action Output
#   Time      Action                                                                                  Message
1   12:11:35  CREATE TABLE IF NOT EXISTS product ( id INT AUTO_INCREMENT PRIMARY KEY, product_name  VARCH...  0 row(s) affected
```

3. Inserta datos en la tabla product.
   Uso SET price = REPLACE(price, '$', '') + 0 para reemplazar el símbolo de dólar ($) con una cadena vacía ('') y sumo 0 para convertirlo en un valor decimal.

   Hay un error:: Error Code: 1366. Incorrect decimal value: '$161.11' for column 'price' at row 1.
   Este error ocurre porque REPLACE() solo funciona con datos de tipo char, varchar.

```
70
71      -- insert data to product
72 •    LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv'
73      INTO TABLE product
74      FIELDS TERMINATED BY ','
75      LINES TERMINATED BY '\n'
76      IGNORE 1 ROWS
77      (id, product_name, price, colour, weight, warehouse_id)
78      SET price = REPLACE(price, '$', '') + 0;
79
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	12:18:11	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv' INTO TABLE ...	Error Code: 1366. Incorrect decimal value: '$161.11' for column 'price' at row 1	0.000 sec

4. Para poder reemplazar el símbolo de dólar cambio tipo de datos de price a VARCHAR(50)

```
80 •    ALTER TABLE product MODIFY COLUMN price VARCHAR(50);
81
```

Output

Action Output

#	Time	Action	Message
1	12:18:11	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv' INTO TABLE ...	Error Code: 1366. Incorrect decimal value: '$161.11' for column 'price' at row 1
2	12:20:13	ALTER TABLE product MODIFY COLUMN price VARCHAR(50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

5. Inserto otra vez datos a product. Esta vez sin errores.

```
84 •    LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv'
85      INTO TABLE product
86      FIELDS TERMINATED BY ','
87      LINES TERMINATED BY '\n'
88      IGNORE 1 ROWS
89      (id, product_name, price, colour, weight, warehouse_id)
90      SET price = REPLACE(price, '$', '') + 0;
```

Output

Action Output

#	Time	Action	Message
1	12:44:52	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv' INTO TABLE pro...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

6. Cambio el tipo de dato de la columna price a decimal

```
93 •    ALTER TABLE product MODIFY COLUMN price DECIMAL(10,2);
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	12:49:50	ALTER TABLE product MODIFY COLUMN price DECIMAL(10,2)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0	0.094 sec

7. En lugar de usar el símbolo $ dentro del precio, agregó una nueva columna llamada currency. Esta columna guarda la información de la moneda como texto (USD), para saber que todos los precios están en dólares.

```
96 •  ALTER TABLE product ADD COLUMN currency VARCHAR(3) DEFAULT 'USD';
97
98 •  select *
99     FROM product;
100
```

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

id	product_name	price	colour	weight	warehouse_id	currency
1	Direwolf Stannis	161.11	#7c7c7c	1.00	WH-4	USD
2	Tarly Stark	9.24	#919191	2.00	WH-3	USD
3	duel tourney Lannister	171.13	#d8d8d8	1.50	WH-2	USD
4	warden south duel	71.89	#111111	3.00	WH-1	USD
5	skywalker ewok	171.22	#dbdbdb	3.20	WH-0	USD
6	dooku solo	136.60	#c4c4c4	0.80	WH--1	USD
7	north of Casterly	63.33	#b7b7b7	0.60	WH--2	USD
8	Winterfell	32.37	#383838	1.40	WH--3	USD
9	Winterfell	76.40	#b5b5b5	1.20	WH--4	USD
10	Karstark Dorne	119.52	#f4f4f4	2.40	WH--5	USD
11	Karstark Dorne	49.70	#141414	2.70	WH--6	USD

product 2

Output

Action Output

#	Time	Action	Message
1	12:49:50	ALTER TABLE product MODIFY COLUMN price DECIMAL(10,2)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0
2	12:52:21	ALTER TABLE products ADD COLUMN currency VARCHAR(3) DEFAULT 'USD'	Error Code: 1146. Table 'sales.products' doesn't exist
3	12:52:32	ALTER TABLE product ADD COLUMN currency VARCHAR(3) DEFAULT 'USD'	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
4	12:53:02	select * FROM product LIMIT 0, 5000	100 row(s) returned

8. La columna product_ids tiene varios ids de productos en una misma transacción, pero para organizar mejor los datos, hay que normalizarla. Por eso, creo una tabla intermedia (transaction_product) donde cada fila tendrá solo un transaction_id y un product_id. Así, cada producto se almacena por separado y es más fácil hacer consultas en SQL.

id	product_ids
02C6201E-D90A-1859-B4EE-88D2986D3B02	71, 1, 19
0466A42E-47CF-8D24-FD01-C0B689713128	47, 97, 43
063FBA79-99EC-66FB-29F7-25726D1764A5	47, 67, 31, 5
0668296C-CDB9-A883-76BC-2E4C44F8C8AE	89, 83, 79
06CD9AA5-9B42-D684-DDDD-A5E394FEBA99	43, 31

9. Create transaction_product table.

   Esta tabla intermedia transaction_product normaliza la relación muchos a muchos entre transacciones y productos. La clave primaria compuesta (PRIMARY KEY (transaction_id, product_id)) garantiza que cada combinación de transacción y producto sea única, evitando duplicados. Además, las claves foráneas aseguran que transaction_id exista en la tabla transaction y product_id en la tabla product, manteniendo la integridad de los datos.

```sql
CREATE TABLE IF NOT EXISTS transaction_product(
transaction_id VARCHAR(255),
product_id  INT,
PRIMARY KEY (transaction_id, product_id), -- composite PK
FOREIGN KEY (transaction_id) REFERENCES transaction(id),
FOREIGN KEY (product_id) REFERENCES product(id)
);
```

Output

Action Output

#	Time	Action	Message
1	11:05:43	CREATE TABLE IF NOT EXISTS transaction_product(transaction_id VARCHAR(255), product_id INT, PRI...	0 row(s) affected

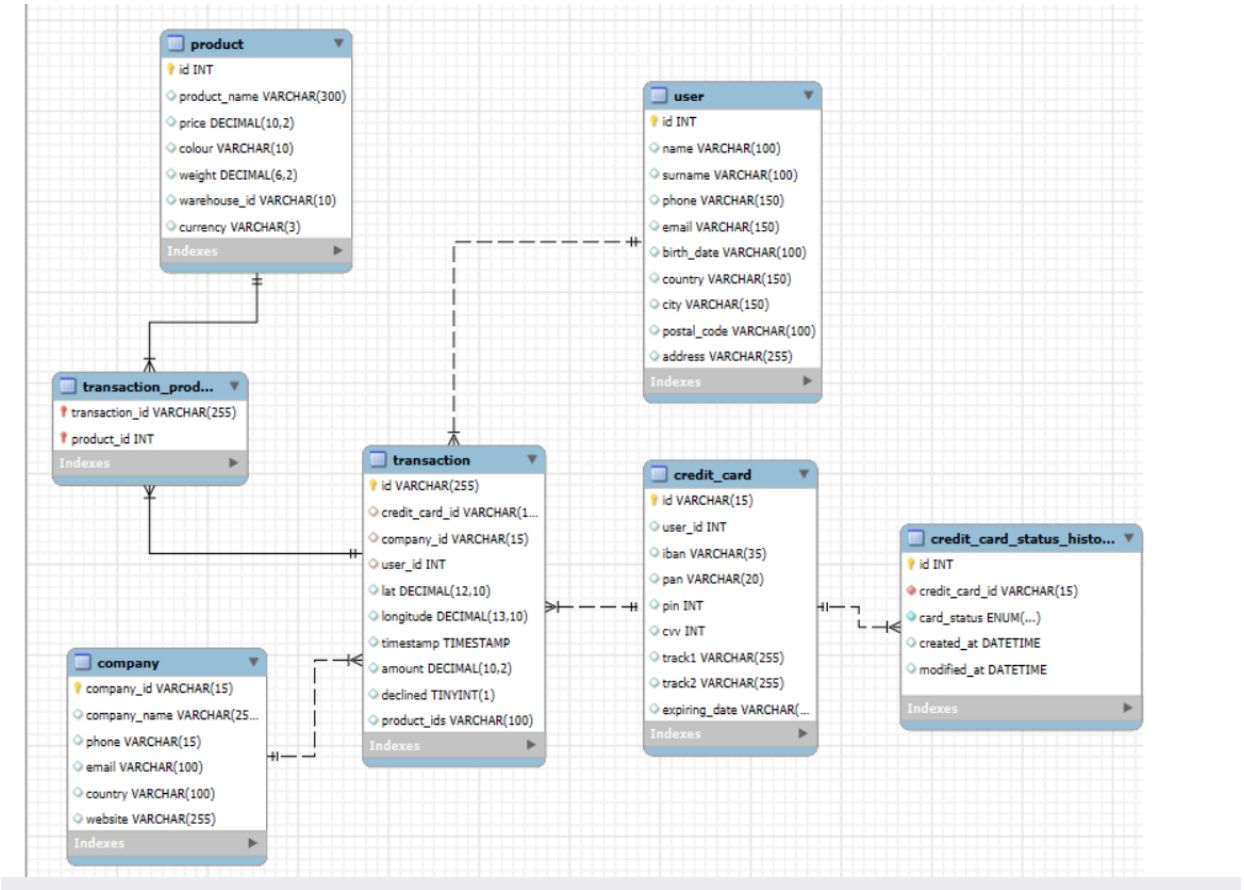
10. Inserto los datos en la tabla transaction_product table

```sql
INSERT INTO transaction_product (transaction_id, product_id)
SELECT t.id, p.id
FROM transaction t
JOIN product p ON FIND_IN_SET(p.id, REPLACE (t.product_ids,' ','' )) > 0;
```

Output

Action Output

#	Time	Action	Message
1	11:05:43	CREATE TABLE IF NOT EXISTS transaction_product(transaction_id VARCHAR(255), product_id INT, PRI...	0 row(s) affected
2	11:15:17	INSERT INTO transaction_product (transaction_id, product_id) SELECT t.id, p.id FROM transaction t JOIN ...	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0



Exercici 1

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

- La tabla transaction_product registra cada producto por transacción, por lo que solo es necesario agrupar por product_id y usar COUNT(transaction_id) para contar cuántas veces se ha vendido cada producto.

```
203
204 •  SELECT
205        tp.product_id,
206       (SELECT p.product_name FROM product p WHERE tp.product_id = p.id) AS product_name,
207        COUNT(tp.product_id) AS total_sales
208    FROM transaction_product tp
209    GROUP BY tp.product_id;
210
211
```

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

product_id	product_name	total_sales
1	Direwolf Stannis	61
2	Tarly Stark	65
3	duel tourney Lannister	51
5	skywalker ewok	49
7	north of Casterly	54

Result 45

Output

Action Output

#	Time	Action	Message
1	13:26:29	SELECT tp.product_id, (SELECT p.product_name FROM product p WHERE tp.product_id = p.id) AS pro...	26 row(s) returned