

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:

Se crea un nuevo esquema donde ubicar las tablas que vamos a crear:

```
1 • CREATE SCHEMA data;
2 • USE data;
3
```

Output			
Action Output			
#	Time	Action	Message
✓ 1	10:29:41	CREATE SCHEMA data	1 row(s) affected
✓ 2	10:29:41	USE data	0 row(s) affected

Se crean las tablas donde se cargará posteriormente la información de los archivos CSV:

```
7 • CREATE TABLE IF NOT EXISTS companies (
8     company_id VARCHAR(15) PRIMARY KEY,
9     company_name VARCHAR(255),
10    phone VARCHAR(15),
11    email VARCHAR(100),
12    country VARCHAR(100),
13    website VARCHAR(255)
14 );
15
16 • CREATE TABLE IF NOT EXISTS credit_card (
17     id VARCHAR(20) PRIMARY KEY,
18     user_id INT,
19     iban VARCHAR(50),
20     pan VARCHAR(20),
21     pin VARCHAR(20),
22     cvv INT,
23     track1 VARCHAR(255),
24     track2 VARCHAR(255),
25     expiring_date VARCHAR(20)
26 );
```

Se crea una única tabla de usuarios llamada “users” para albergar la totalidad de los datos de los usuarios de Canadá, United Kingdom y USA.

```
28 • CREATE TABLE IF NOT EXISTS users (
29     id INT PRIMARY KEY,
30     name VARCHAR(100),
31     surname VARCHAR(100),
32     phone VARCHAR(150),
33     email VARCHAR(150),
34     birth_date VARCHAR(100),
35     country VARCHAR(150),
36     city VARCHAR(150),
37     postal_code VARCHAR(100),
38     address VARCHAR(255)
39 );
40
```

```

40
41 • CREATE TABLE IF NOT EXISTS transactions (
42     id VARCHAR(255) PRIMARY KEY,
43     card_id VARCHAR(20),
44     business_id VARCHAR(20),
45     timestamp VARCHAR(30),
46     amount DECIMAL(10, 2),
47     declined BOOLEAN,
48     products_ids VARCHAR(30),
49     user_id INT,
50     lat FLOAT,
51     longitude FLOAT,
52     FOREIGN KEY (card_id) REFERENCES credit_card(id),
53     FOREIGN KEY (business_id) REFERENCES companies(company_id),
54     FOREIGN KEY (user_id) REFERENCES users(id)
55 );
56

```

Output

Action Output

#	Time	Action	Message
✓ 1	10:33:46	CREATE TABLE IF NOT EXISTS companies (company_id VARCHAR(15) PRIMARY KEY, company_name VARCHAR(255), phone VARCHAR(15), e...	0 row(s) affected
✓ 2	10:33:46	CREATE TABLE IF NOT EXISTS credit_card (id VARCHAR(20) PRIMARY KEY, user_id INT, iban VARCHAR(50), pan VARCHAR(20), pin VARCHA...	0 row(s) affected
✓ 3	10:33:46	CREATE TABLE IF NOT EXISTS users (id INT PRIMARY KEY, name VARCHAR(100), surname VARCHAR(100), phone VARCHA...	0 row(s) affected
✓ 4	10:33:46	CREATE TABLE IF NOT EXISTS transactions (id VARCHAR(255) PRIMARY KEY, card_id VARCHAR(20), business_id VARCHAR(20), ti...	0 row(s) affected

Buscar donde está “secure_file_priv” pues es allí desde donde el sistema va a permitir que se carguen los archivos CSV:

```

57 • SHOW VARIABLES LIKE "secure_file_priv";
58

```

Result Grid

Variable_name	Value
secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.4\Upl...

Result 13 x

Output

Action Output

#	Time	Action	Message
✓ 1	10:44:59	SHOW VARIABLES LIKE "secure_file_priv"	1 row(s) returned

Se cargan los datos en las tablas ya creadas. Los datos de las transacciones se cargarán en último lugar porque antes han de introducirse los valores de las tablas relacionadas con sus foreigns key:

```

59 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\companies.csv'
60 INTO TABLE companies
61 FIELDS TERMINATED BY ','      # indica que los campos están separados por una coma
62 ENCLOSED BY ''               # indica que los valores contienen un separador de campos o la terminación de línea, garantiza la integridad de los datos y evita er
63 LINES TERMINATED BY '\\n'    # indica que las líneas están separadas por un carácter de nueva línea \\n
64 IGNORE 1 ROWS;              # ignora la primera línea porque es un encabezado
65
66 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\credit_cards.csv'
67 INTO TABLE credit_card
68 FIELDS TERMINATED BY ','
69 ENCLOSED BY ''
70 LINES TERMINATED BY '\\n'
71 IGNORE 1 ROWS;
72

```

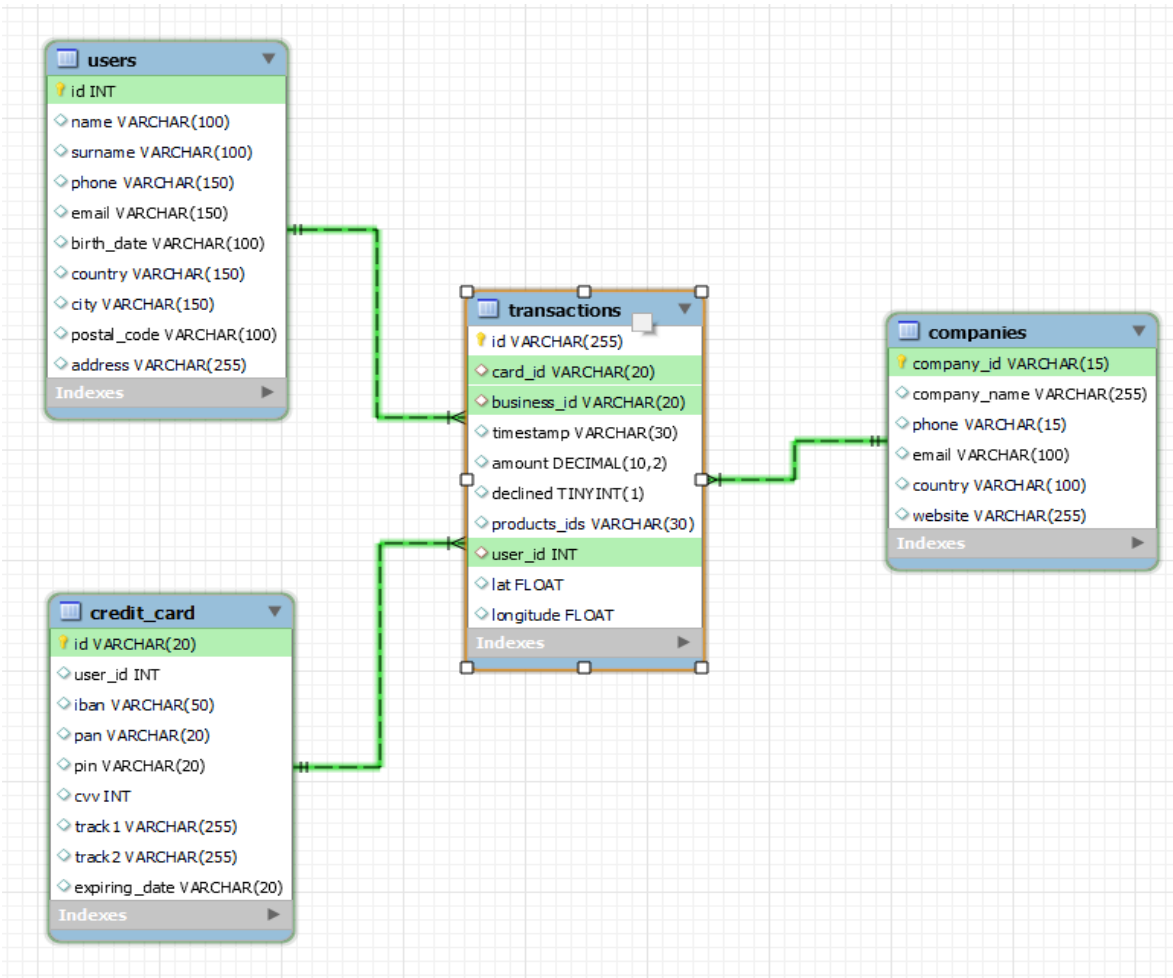
```

73 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\users_ca.csv'
74 INTO TABLE users
75 FIELDS TERMINATED BY ','
76 ENCLOSED BY ''
77 LINES TERMINATED BY '\\r\\n' # indica que el archivo de texto ha sido creado en Windows CRLF y utiliza \\r\\n como terminador de línea.
78 IGNORE 1 ROWS;
79
80 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\users_uk.csv'
81 INTO TABLE users
82 FIELDS TERMINATED BY ','
83 ENCLOSED BY ''
84 LINES TERMINATED BY '\\r\\n'
85 IGNORE 1 ROWS;
86
87 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\users_usa.csv'
88 INTO TABLE users
89 FIELDS TERMINATED BY ','
90 ENCLOSED BY ''
91 LINES TERMINATED BY '\\r\\n'
92 IGNORE 1 ROWS;
93
96 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\transactions.csv' # se carga en último lugar por las FK's
97 INTO TABLE transactions
98 FIELDS TERMINATED BY ';' # indica que los campos están separados por punto y coma
99 ENCLOSED BY ''
100 LINES TERMINATED BY '\\n'
101 IGNORE 1 ROWS;

```

#	Time	Action	Message
2	12:51:16	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\credit_cards.csv' INTO TABLE credit_card FIELDS TERMINATED ...	275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings: 0
3	12:51:16	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\users_ca.csv' INTO TABLE users FIELDS TERMINATED BY ';' E...	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0
4	12:51:16	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\users_uk.csv' INTO TABLE users FIELDS TERMINATED BY ';' E...	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0
5	12:51:16	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\users_usa.csv' INTO TABLE users FIELDS TERMINATED BY ';' ...	150 row(s) affected Records: 150 Deleted: 0 Skipped: 0 Warnings: 0
6	12:51:16	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\transactions.csv' # se carga en último lugar por las FK's INTO TA...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0

El resultado es un diagrama en estrella que indica que todas las tablas de dimensiones están relacionadas con la tabla de hechos:



La tabla de hechos es “transactions” y las tablas de dimensiones son “companies”, “users” y “credit_card”. En este caso, las tablas de dimensiones ampliarían la información de algunas de sus variables. La relación de las tres tablas de dimensiones es de 1 a muchos, es decir, cada compañía puede realizar múltiples transacciones, un mismo usuario puede realizar múltiples transacciones y con una misma tarjeta de crédito pueden realizarse diversas transacciones.

- Exercici 1

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

Sólo se tienen en cuenta las transacciones que efectivamente se han realizado, es decir, las transacciones que no han sido rechazadas.

```

110 • SELECT t.user_id AS 'Id. usuario', u.name AS Nombre, u.surname AS Apellido, COUNT(*) AS Num_Transacciones
111 FROM transactions t
112 JOIN users u
113 ON t.user_id = u.id
114 WHERE t.declined = 0
115 GROUP BY t.user_id, u.name, u.surname
116 HAVING Num_Transacciones > 30;
117
118

```

Id. usuario	Nombre	Apellido	Num_Transacciones
92	Lynn	Riddle	39
267	Ocean	Nelson	39
272	Hedwig	Gilbert	38

Result 14 x

Output

Action Output

#	Time	Action	Message
1	10:54:35	SELECT t.user_id AS 'Id. usuario', u.name AS Nombre, u.surname AS Apellido, COUNT(*) AS Num_Transacciones FROM transactions t JOIN users ...	3 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.

Sólo se tienen en cuenta las transacciones que efectivamente se han realizado.

```

131 • SELECT cc.iban AS IBAN, AVG(tr.amount) AS Media_Amount
132 FROM transactions tr
133 JOIN credit_card cc
134 ON tr.card_id = cc.id
135 JOIN companies co
136 ON tr.business_id = co.company_id
137 WHERE co.company_name = 'Donec Ltd' AND tr.declined = 0
138 GROUP BY cc.iban;
139

```

IBAN	Media_Amount
PT87806228135092429456346	42.820000

Result 19 x			
Output			
Action Output			
#	Time	Action	Message
1	10:59:35	SELECT cc.iban AS IBAN, AVG(tr.amount) AS Media_Amount FROM transactions tr JOIN credit_card cc ON tr.card_id = cc.id JOIN companies co O...	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:

146	•	CREATE TABLE estado_tarjetas AS	
147		SELECT	
148		card_id,	
149		CASE	
150		WHEN SUM(# la variable declined es booleana y, aunque podría sumar el valor 1, podría dar error
151		CASE	
152		WHEN declined = 1 THEN 1	
153		ELSE 0	
154		END) = 3 THEN 0	
155		ELSE 1	
156		END AS tarjeta_activa	
157		FROM	
158		(SELECT card_id, declined,	
159		ROW_NUMBER() OVER(PARTITION BY card_id ORDER BY timestamp DESC) AS cuenta_transacciones	
160		FROM transactions	
161) AS ordena_transacciones	
162		WHERE cuenta_transacciones <= 3	
163		GROUP BY card_id;	
164			
Output			
Action Output			
#	Time	Action	Message
1	11:05:40	CREATE TABLE estado_tarjetas AS SELECT card_id, CASE WHEN SUM(# la variable declined es booleana y, aunque podría suma...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

- Exercici 1

Quantes targetes estan actives?

169	•	SELECT COUNT(*) AS 'Número de tarjetas activas'	
170		FROM estado_tarjetas	
171		WHERE tarjeta_activa = 1;	
172			
173			
Result Grid			
		Número de tarjetas activas	
		275	

Result 20 x			
Output			
Action Output			
#	Time	Action	Message
1	11:06:52	SELECT COUNT(*) AS 'Número de tarjetas activas' FROM estado_tarjetas WHERE tarjeta_activa = 1	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:

Se crea una tabla llamada “products” que contendrá los datos de los productos y luego se carga el archivo CSV que los incluye.

```
178 CREATE TABLE IF NOT EXISTS products (  
179     id VARCHAR(30) PRIMARY KEY,  
180     product_name VARCHAR(255),  
181     price VARCHAR(10),  
182     colour VARCHAR(15),  
183     weight FLOAT,  
184     warehouse_id VARCHAR(10)  
185 );  
186  
187 LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\products.csv'  
188 INTO TABLE products  
189 FIELDS TERMINATED BY ','  
190 ENCLOSED BY ''''  
191 LINES TERMINATED BY '\\n'  
192 IGNORE 1 ROWS;
```

Output

#	Time	Action	Message
1	11:09:00	CREATE TABLE IF NOT EXISTS products (id VARCHAR(30) PRIMARY KEY, product_name VARCHAR(255), price VARCHAR(10), colour VARCHAR(15), weight FLOAT, warehouse_id VARCHAR(10));	0 row(s) affected
2	11:09:00	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.4\\Uploads\\products.csv' INTO TABLE products FIELDS TERMINATED BY ',' ENCLOSED BY '''' LINES TERMINATED BY '\\n' IGNORE 1 ROWS;	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

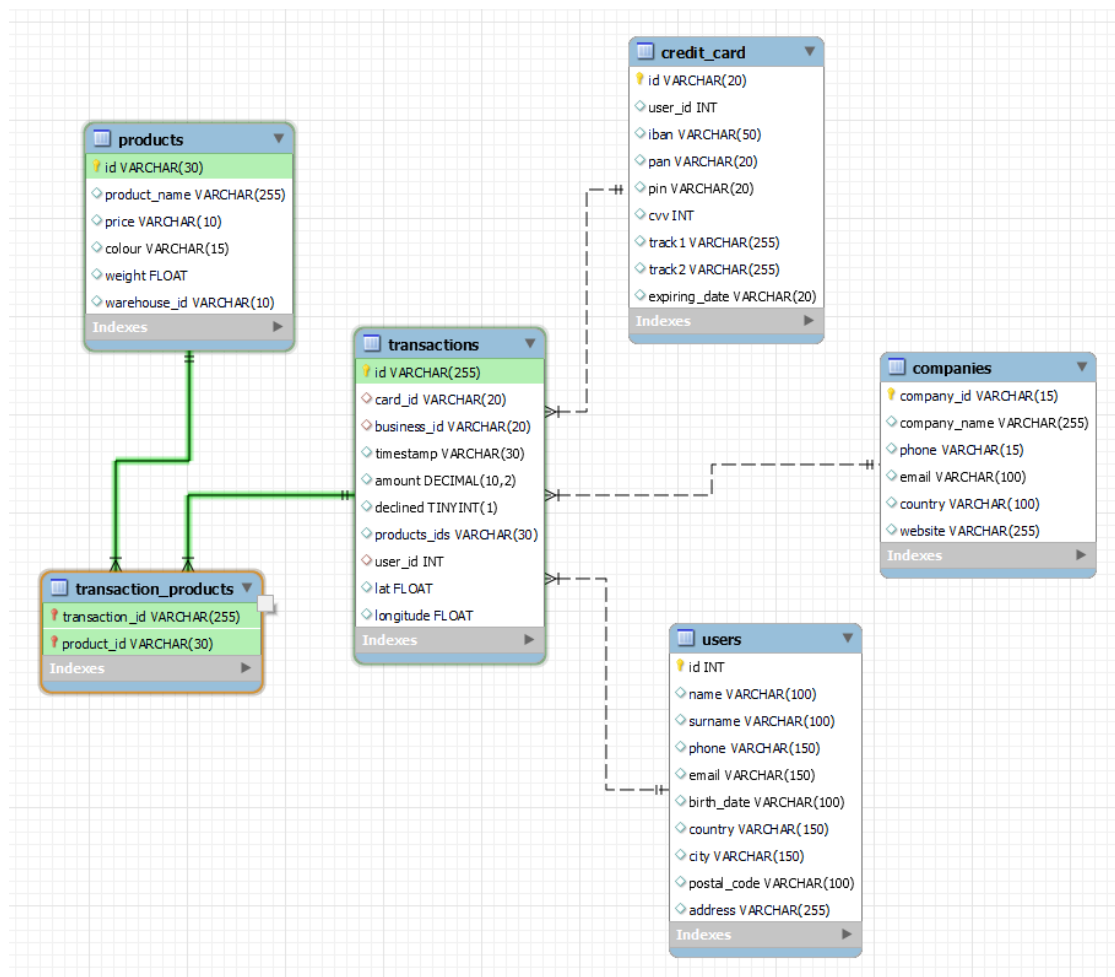
Se crea una tabla intermedia que va a unir la tabla “products” con la tabla “transactions”, ya que la relación entre estas dos tablas es de muchos a muchos, es decir, un producto puede estar en diversas transacciones o en ninguna, y una transacción puede contener uno o varios productos. La nueva tabla, llamada “transaction_products”, estará relacionada de muchos a uno con las tablas ya mencionadas. Va a contener dos variables, transaction_id y product_id, y ambas serán primary key, a la vez que son foreign key de las respectivas tablas con las que se relacionan. Así pues, esta tabla contendrá todos los productos que se incluyeron en cada una de las transacciones.

```
194 CREATE TABLE transaction_products (  
195     transaction_id VARCHAR(255),  
196     product_id VARCHAR(30),  
197     PRIMARY KEY (transaction_id, product_id),  
198     FOREIGN KEY (transaction_id) REFERENCES transactions(id),  
199     FOREIGN KEY (product_id) REFERENCES products(id)  
200 );  
201  
202 INSERT INTO transaction_products (transaction_id, product_id)  
203 SELECT tr.id, pr.id  
204 FROM transactions tr  
205 JOIN products pr  
206 ON FIND_IN_SET(pr.id, REPLACE(tr.products_ids, ', ', ''));  
207  
208
```

Output

#	Time	Action	Message
1	11:10:26	CREATE TABLE transaction_products (transaction_id VARCHAR(255), product_id VARCHAR(30), PRIMARY KEY (transaction_id, product_id), FOREIGN KEY (transaction_id) REFERENCES transactions(id), FOREIGN KEY (product_id) REFERENCES products(id));	0 row(s) affected
2	11:10:26	INSERT INTO transaction_products (transaction_id, product_id) SELECT tr.id, pr.id FROM transactions tr JOIN products pr ON FIND_IN_SET(pr.id, REPLACE(tr.products_ids, ', ', ''));	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0

El nuevo diagrama queda del siguiente modo, pudiéndose distinguir las líneas continuas que unen la tabla “transaction_products” con las tablas “products” y “transactions” que indican que hay una relación dependiente entre las variables que unen las tablas, de modo que no hay valores nulos. En cambio, las líneas discontinuas indican que, aunque hay relación, la foreign key puede tener valores nulos.



- Exercici 1

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

Sólo se tienen en cuenta las transacciones que efectivamente se han realizado.

```

213 • SELECT product_id AS 'Identificador del producto', product_name AS 'Nombre del producto', COUNT(*) AS 'Número de Ventas'
214 FROM transaction_products tp
215 JOIN products p
216 ON tp.product_id = p.id
217 JOIN transactions t
218 ON tp.transaction_id = t.id
219 WHERE declined = 0
220 GROUP BY tp.product_id, p.product_name
221 ORDER BY p.product_name;
222
  
```

Result Grid			
Filter Rows: <input type="text"/>			
Export: <input type="button" value=""/>			
Wrap Cell Content: <input type="button" value=""/>			
	Identificador del producto	Nombre del producto	Número de Ventas
37		Direwolf Littlefinger	45
79		Direwolf riverlands the	52
1		Direwolf Stannis	51
59		Direwolf Stannis	35
19		dooku solo	44
73		Dorne bastard	39
43		duel	54
83		duel tourney	46
3		duel tourney Lannister	43
67		duel Winterfell	50

Result 25 x

Output:

Action Output

#	Time	Action	Message
1	11:29:26	SELECT product_id AS 'Identificador del producto', product_name AS 'Nombre del producto', COUNT(*) AS 'Número de Ventas' FROM transaction_pr...	26 row(s) returned