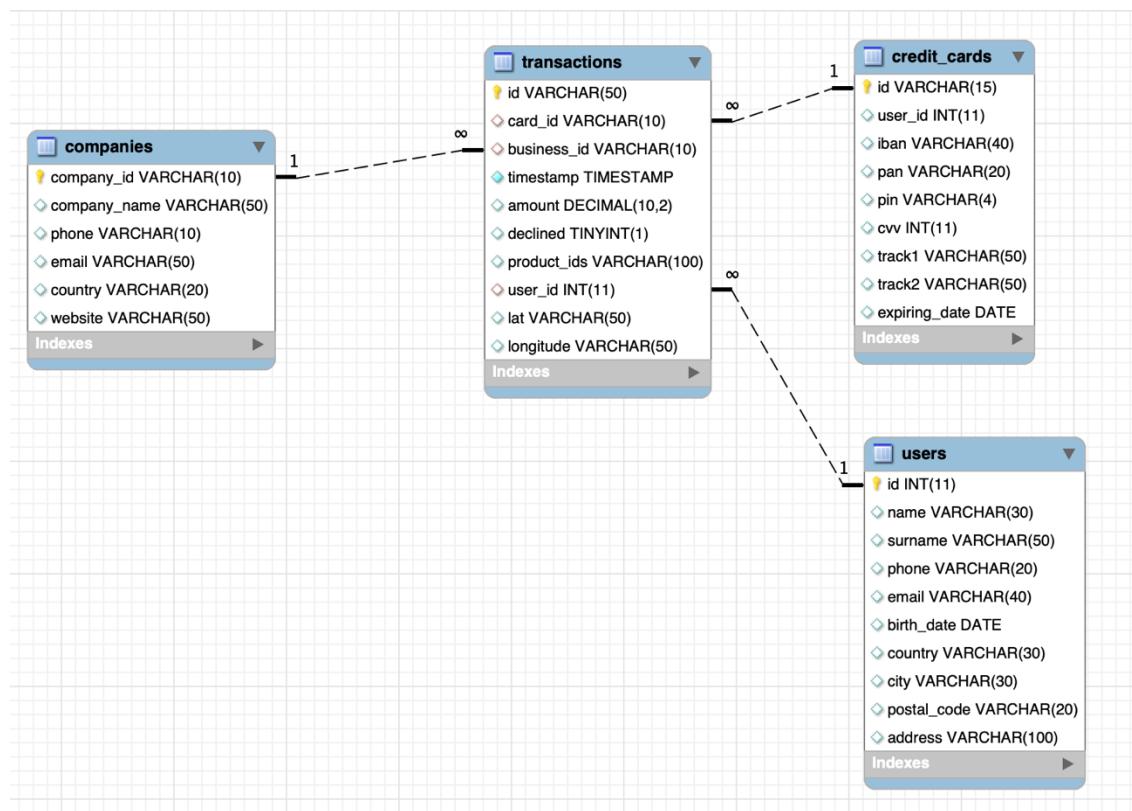


Nivell 1

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:

La base de datos que he diseñado es la siguiente:



La base de datos se llama “sales”. En ella, la tabla “transactions” es tabla de hechos, y de ella salen “companies”, “credit_cards” y “users” como sus tablas de dimensiones.

La Primary Key de “transactions” es “id” y está conectada con “companies” mediante la Foreign Key “business_id”, con “credit_cards” mediante la Foreign Key “id” y con “users” mediante la Foreign Key “user_id”.

La Primary Key de “companies” es “company_id”.

La Primary Key de “credit_cards” es “id”.

La Primary Key de “users” es “id”.

La relación entre “companies” y “transactions” es de uno a muchos, ya que una empresa puede realizar muchas transacciones.

Anabel Martínez Ramírez

La relación entre “credit_cards” y “transactions” también es uno a muchos, ya que con una tarjeta de crédito se pueden realizar muchas transacciones.

La relación entre “users” y “transactions” también es uno a muchos, ya que un usuario puede realizar muchas transacciones.

He creado la base de datos “sales” con el siguiente código:

The screenshot shows the MySQL Workbench interface. In the SQL editor, lines 4 and 5 show the creation of the 'sales' database and switching to it. The status bar at the bottom left shows the time as 11:16. The Action Output panel shows two log entries: one for creating the database and another for switching to it. Both entries have a duration of approximately 0.00017 seconds.

Action	Time	Response	Duration / Fetch Time
CREATE DATABASE IF NOT EXISTS sales;	13:11:43	1 row(s) affected	0.00041 sec
USE sales;	13:11:49	0 row(s) affected	0.00017 sec

Después, he pasado a la creación de la primera tabla, “transactions”. He creado esta tabla con todos los datos de tipo VARCHAR(255) porque no conocemos el tipo de dato y ya lo cambiaré más adelante.

The screenshot shows the MySQL Workbench interface. Line 8 shows the creation of the 'transactions' table with various columns defined using VARCHAR(255). The status bar at the bottom left shows the time as 7:19. The Action Output panel shows the log entry for creating the table, which completed successfully with 0 rows affected in 0.024 seconds.

Action	Time	Response	Duration / Fetch Time
CREATE TABLE IF NOT EXISTS transactions (id VARCHAR(255) PRIMARY KEY, card_id VARCHAR(255) REFERENCES credit_cards(id), business_id VARCHAR(255) REFERENCES companies(company_id), timestamp VARCHAR(255), amount VARCHAR(255), declined VARCHAR(255), product_ids VARCHAR(255) REFERENCES products(id), user_id VARCHAR(255) REFERENCES users(id), lat VARCHAR(255), longitude VARCHAR(255));	13:14:25	0 row(s) affected	0.024 sec

Ahora cargo los datos desde el csv con el comando LOAD DATA INFILE. Marco que el archivo csv está separado con el “;”. En este caso, he limpiado la columna “product_ids” porque había espacios entre los números y las comas y, como he tenido muchos problemas para resolver el Nivel 3, una de las pruebas que hice fue quitando los espacios entre las comas. Este cambio lo he hecho posteriormente a haber creado las tablas, pero la verdad es que he tenido que hacer drop schema y cargar todo de nuevo más de una vez.

Anabel Martínez Ramírez

```
20
21 • LOAD DATA
22   INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/transactions.csv'
23   INTO TABLE transactions
24   FIELDS TERMINATED BY ','
25   IGNORE 1 ROWS
26   (id, card_id, business_id, timestamp, amount, declined, @product_ids, user_id, lat, longitude)
27   SET
28   product_ids = REPLACE(@product_ids, ',', '')
29 ;
30
31 ◇ 14:25 |
```

Action Output	Time	Action	Response	Duration / Fetch Time
2	13:18:31	LOAD DATA INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/transactions.csv' INTO TABLE tr...	587 row(s) affected Records: 587 Deleted: 0 Skippe...	0.013 sec

Ahora hago todos los ALTER TABLE para adecuar el tipo de dato de cada columna:

```
31 • ALTER TABLE transactions MODIFY COLUMN id VARCHAR(50);
32 • ALTER TABLE transactions MODIFY COLUMN card_id VARCHAR(10);
33 • ALTER TABLE transactions MODIFY COLUMN business_id VARCHAR(10);
34 • ALTER TABLE transactions MODIFY COLUMN timestamp TIMESTAMP;
35 • ALTER TABLE transactions MODIFY COLUMN amount DECIMAL(10, 2);
36 • ALTER TABLE transactions MODIFY COLUMN declined BOOLEAN;
37 • ALTER TABLE transactions MODIFY COLUMN product_ids VARCHAR(100);
38 • ALTER TABLE transactions MODIFY COLUMN user_id INT;
39 • ALTER TABLE transactions MODIFY COLUMN lat VARCHAR(50);
40 • ALTER TABLE transactions MODIFY COLUMN longitude VARCHAR(50);
41
42
43 ◇ 55:40 |
```

Action Output	Time	Action	Response	Duration / Fetch T
1	13:25:40	ALTER TABLE transactions MODIFY COLUMN id VARCHAR(50)	587 row(s) affected Records: 587 Duplicates: 0 War...	0.054 sec
2	13:25:42	ALTER TABLE transactions MODIFY COLUMN card_id VARCHAR(10)	587 row(s) affected Records: 587 Duplicates: 0 War...	0.030 sec
3	13:25:42	ALTER TABLE transactions MODIFY COLUMN business_id VARCHAR(10)	587 row(s) affected Records: 587 Duplicates: 0 War...	0.029 sec
4	13:25:43	ALTER TABLE transactions MODIFY COLUMN amount DECIMAL(10, 2)	587 row(s) affected Records: 587 Duplicates: 0 War...	0.050 sec
5	13:25:44	ALTER TABLE transactions MODIFY COLUMN declined BOOLEAN	587 row(s) affected Records: 587 Duplicates: 0 War...	0.023 sec
6	13:25:45	ALTER TABLE transactions MODIFY COLUMN product_ids VARCHAR(100)	587 row(s) affected Records: 587 Duplicates: 0 War...	0.045 sec
7	13:25:45	ALTER TABLE transactions MODIFY COLUMN user_id INT	587 row(s) affected Records: 587 Duplicates: 0 War...	0.031 sec
8	13:25:46	ALTER TABLE transactions MODIFY COLUMN lat VARCHAR(50)	587 row(s) affected Records: 587 Duplicates: 0 War...	0.017 sec
9	13:25:47	ALTER TABLE transactions MODIFY COLUMN longitude VARCHAR(50)	587 row(s) affected Records: 587 Duplicates: 0 War...	0.021 sec

A continuación, creo la siguiente tabla, “companies”.

Después, cargo los datos desde el csv. En este caso, el separador de este csv es la “,”.

He decidido eliminar los espacios en los números de teléfono para que haya una consistencia en los datos.

```
43 • CREATE TABLE IF NOT EXISTS companies (
44   company_id VARCHAR(255) PRIMARY KEY,
45   company_name VARCHAR(255),
46   phone VARCHAR(255),
47   email VARCHAR(255),
48   country VARCHAR(255),
49   website VARCHAR(255)
50 );
51
52 • LOAD DATA
53   INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/companies.csv'
54   INTO TABLE companies
55   FIELDS TERMINATED BY ','
56   IGNORE 1 ROWS
57   (company_id, company_name, @phone, email, country, website)
58   SET
59   phone = REPLACE(@phone, ' ', '')|
```

Action Output	Time	Action	Response	Duration / Fetch Time
1	13:26:59	CREATE TABLE IF NOT EXISTS companies (company_id VARCHAR(255) PRIMARY KEY, company... 0 row(s) affected		0.027 sec
2	13:27:02	LOAD DATA INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/companies.csv' INTO TABLE co...	100 row(s) affected Records: 100 Deleted: 0 Skippe...	0.0036 sec

Anabel Martínez Ramírez

Ahora hago todos los ALTER TABLE para adecuar el tipo de dato de cada columna:

```
61 • ALTER TABLE companies MODIFY COLUMN company_id VARCHAR(10);
62 • ALTER TABLE companies MODIFY COLUMN company_name VARCHAR(50);
63 • ALTER TABLE companies MODIFY COLUMN phone VARCHAR(10);
64 • ALTER TABLE companies MODIFY COLUMN email VARCHAR(50);
65 • ALTER TABLE companies MODIFY COLUMN country VARCHAR(20);
66 • ALTER TABLE companies MODIFY COLUMN website VARCHAR(50);
67
68 ◇ 60:57 |
```

Action Output	Time	Action	Response	Duration / Fetch Time
✓ 1	13:42:45	ALTER TABLE companies MODIFY COLUMN company_id VARCHAR(10)	100 row(s) affected Records: 100 Duplicates: 0 War...	0.021 sec
✓ 2	13:42:46	ALTER TABLE companies MODIFY COLUMN company_name VARCHAR(50)	100 row(s) affected Records: 100 Duplicates: 0 War...	0.033 sec
✓ 3	13:42:47	ALTER TABLE companies MODIFY COLUMN phone VARCHAR(10)	100 row(s) affected Records: 100 Duplicates: 0 War...	0.030 sec
✓ 4	13:42:48	ALTER TABLE companies MODIFY COLUMN email VARCHAR(50)	100 row(s) affected Records: 100 Duplicates: 0 War...	0.015 sec
✓ 5	13:42:49	ALTER TABLE companies MODIFY COLUMN country VARCHAR(20)	100 row(s) affected Records: 100 Duplicates: 0 War...	0.028 sec
✓ 6	13:42:50	ALTER TABLE companies MODIFY COLUMN website VARCHAR(50)	100 row(s) affected Records: 100 Duplicates: 0 War...	0.026 sec

Y creo la Foreign Key entre transactions y companies que relaciona “business_id” con “company_id”:

```
68 • ALTER TABLE transactions
69 ADD CONSTRAINT FK_companiestrans
70 FOREIGN KEY (business_id) REFERENCES companies(company_id);
71
72 ◇ 55:63 |
```

Action Output	Time	Action	Response	Duration / Fetch Time
✓ 7	13:47:25	ALTER TABLE transactions ADD CONSTRAINT FK_companiestrans FOREIGN KEY (business_id) RE...	587 row(s) affected Records: 587 Duplicates: 0 War...	0.041 sec

Ahora creo la tabla “credit_cards”.

Después, cargo los datos desde el csv. En este caso, el separador de este csv es la “,”.

He decidido eliminar los espacios en la columna “pan” para que haya una consistencia en los datos.

También he cambiado el formato de la columna “expiring_date” para que tenga un formato que acepte el cambio al tipo de dato DATE.

```
72 • CREATE TABLE IF NOT EXISTS credit_cards(
73     id VARCHAR(255) PRIMARY KEY,
74     user_id VARCHAR(255),
75     iban VARCHAR(255),
76     pan VARCHAR(255),
77     pin VARCHAR(255),
78     cvv VARCHAR(255),
79     track1 VARCHAR(255),
80     track2 VARCHAR(255),
81     expiring_date VARCHAR(255)
82 );
83
84 • LOAD DATA
85    INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/credit_cards.csv'
86    INTO TABLE credit_cards
87    FIELDS TERMINATED BY ','
88    IGNORE 1 ROWS
89    (id, user_id, iban, @pan, pin, cvv, track1, track2, @expiring_date)
90    SET
91    pan = REPLACE(@pan, ' ', ''),
92    expiring_date = STR_TO_DATE(@expiring_date, '%m/%d/%y');
93
94 ◇ 60:70 |
```

Action Output	Time	Action	Response	Duration / Fetch Time
✓ 1	14:32:09	CREATE TABLE IF NOT EXISTS credit_cards(id VARCHAR(255) PRIMARY KEY, user_id VARC... 0 row(s) affected	0 row(s) affected	0.012 sec
✓ 2	14:32:12	LOAD DATA INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/credit_cards.csv' INTO TABLE cr...	275 row(s) affected Records: 275 Deleted: 0 Skippe...	0.0090 sec

Anabel Martínez Ramírez

Ahora hago todos los ALTER TABLE para adecuar el tipo de dato de cada columna:

```
94 • ALTER TABLE credit_cards MODIFY COLUMN id VARCHAR(15);
95 • ALTER TABLE credit_cards MODIFY COLUMN user_id INT;
96 • ALTER TABLE credit_cards MODIFY COLUMN iban VARCHAR(40);
97 • ALTER TABLE credit_cards MODIFY COLUMN pan VARCHAR(20);
98 • ALTER TABLE credit_cards MODIFY COLUMN pin VARCHAR(4);
99 • ALTER TABLE credit_cards MODIFY COLUMN cvv INT;
100 • ALTER TABLE credit_cards MODIFY COLUMN track1 VARCHAR(50);
101 • ALTER TABLE credit_cards MODIFY COLUMN track2 VARCHAR(50);
102 • ALTER TABLE credit_cards MODIFY COLUMN expiring_date DATE;
103
```

Action Output

Time	Action	Response	Duration / Fetch Time
1 14:40:25	ALTER TABLE credit_cards MODIFY COLUMN id VARCHAR(15)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.048 sec
2 14:40:27	ALTER TABLE credit_cards MODIFY COLUMN user_id INT	275 row(s) affected Records: 275 Duplicates: 0 War...	0.017 sec
3 14:40:29	ALTER TABLE credit_cards MODIFY COLUMN iban VARCHAR(40)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.027 sec
4 14:40:30	ALTER TABLE credit_cards MODIFY COLUMN pan VARCHAR(20)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.021 sec
5 14:40:31	ALTER TABLE credit_cards MODIFY COLUMN pin VARCHAR(4)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.027 sec
6 14:40:31	ALTER TABLE credit_cards MODIFY COLUMN cvv INT	275 row(s) affected Records: 275 Duplicates: 0 War...	0.025 sec
7 14:40:32	ALTER TABLE credit_cards MODIFY COLUMN track1 VARCHAR(50)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.025 sec
8 14:40:33	ALTER TABLE credit_cards MODIFY COLUMN track2 VARCHAR(50)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.027 sec
9 14:40:34	ALTER TABLE credit_cards MODIFY COLUMN expiring_date DATE	275 row(s) affected Records: 275 Duplicates: 0 War...	0.018 sec

Y creo la Foreign Key entre “transactions” y “credit_cards” que relaciona “card_id” con “id”:

```
105
106 • ALTER TABLE transactions
107   ADD CONSTRAINT FK_creditcardstrans
108     FOREIGN KEY (card_id) REFERENCES credit_cards(id);
```

Action Output

Time	Action	Response	Duration / Fetch Time
1 14:41:19	ALTER TABLE transactions ADD CONSTRAINT FK_creditcardstrans FOREIGN KEY (card_id) REFER...	587 row(s) affected Records: 587 Duplicates: 0 War...	0.046 sec

A continuación, creo la tabla “users”.

```
110 • CREATE TABLE IF NOT EXISTS users(
111   id VARCHAR(255) PRIMARY KEY,
112   name VARCHAR(255),
113   surname VARCHAR(255),
114   phone VARCHAR(255),
115   email VARCHAR(255),
116   birth_date VARCHAR(255),
117   country VARCHAR(255),
118   city VARCHAR(255),
119   postal_code VARCHAR(255),
120   address VARCHAR(255)
121 );
122
```

Action Output

Time	Action	Response	Duration / Fetch Time
1 14:42:30	CREATE TABLE IF NOT EXISTS users(id VARCHAR(255) PRIMARY KEY, name VARCHAR(25... 0 row(s) affected		0.043 sec

He decidido que voy a cargar los tres csv “users_ca.csv”, “users_uk.csv” y “users_usa.csv”, en esta tabla.

Tuve bastantes problemas para dar con la tecla y conseguir cargar correctamente estos csv, estuve haciendo varias pruebas hasta que di con la solución: que es necesario poner que hay un salto de línea y que se expresa con \r\n. También tuve que expresar el ENCLOSED BY "" porque hay datos expresados entre comillas, así que se deben tener en cuenta al importar el csv. En este caso, el separador de este csv es la “,”.

Anabel Martínez Ramírez

He decidido eliminar los caracteres especiales y los espacios de la columna “phone” para estandarizar todos los teléfonos.
También he cambiado el formato de la columna “birth_date” para que tenga un formato que acepte el cambio al tipo de dato DATE.
Por último, también he decidido eliminar el espacio de la columna “postal_code” para estandarizarlo.

```
122
123 • LOAD DATA
124   INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/users_usa.csv'
125   INTO TABLE users
126   FIELDS TERMINATED BY ','
127   ENCLOSED BY ""
128   LINES TERMINATED BY '\r\n' #esta es la forma de saltar linea de windows (CRLF)
129   IGNORE 1 ROWS
130   (@id,@name,@surname,@phone,@email,@birth_date,@country,@city,@postal_code,@address)
131   SET
132   phone = REPLACE(REPLACE(REPLACE(@phone, '(', ''), ')', ''), '+', ''), ' ', ''),
133   birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y'),
134   postal_code = REPLACE(@postal_code, ' ', '');
135
⌚ 11:12 | ⏴
Action Output
| Time | Action | Response | Duration / Fetch Time
✓ 1 14:45:34 LOAD DATA INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/users_usa.csv' INTO TABLE user... 150 row(s) affected Records: 150 Deleted: 0 Skippe... 0.0067 sec
```

```
136 • LOAD DATA
137   INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/users_uk.csv'
138   INTO TABLE users
139   FIELDS TERMINATED BY ','
140   ENCLOSED BY ""
141   LINES TERMINATED BY '\r\n'
142   IGNORE 1 ROWS
143   (@id,@name,@surname,@phone,@email,@birth_date,@country,@city,@postal_code,@address)
144   SET
145   phone = REPLACE(REPLACE(REPLACE(@phone, '(', ''), ')', ''), '+', ''), ' ', ''),
146   birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y'),
147   postal_code = REPLACE(@postal_code, ' ', '');
148
⌚ 46:13 | ⏴
Action Output
| Time | Action | Response | Duration / Fetch Time
✓ 1 14:45:58 LOAD DATA INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/users_uk.csv' INTO TABLE user... 50 row(s) affected Records: 50 Deleted: 0 Skippe... 0.0040 sec
```

```
149 • LOAD DATA
150   INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/users_ca.csv'
151   INTO TABLE users
152   FIELDS TERMINATED BY ','
153   ENCLOSED BY ""
154   LINES TERMINATED BY '\r\n'
155   IGNORE 1 ROWS
156   (@id,@name,@surname,@phone,@email,@birth_date,@country,@city,@postal_code,@address)
157   SET
158   phone = REPLACE(REPLACE(REPLACE(@phone, '(', ''), ')', ''), '+', ''), ' ', ''),
159   birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y'),
160   postal_code = REPLACE(@postal_code, ' ', '');
161
⌚ 22:14 | ⏴
Action Output
| Time | Action | Response | Duration / Fetch Time
✓ 1 14:46:26 LOAD DATA INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/users_ca.csv' INTO TABLE user... 75 row(s) affected Records: 75 Deleted: 0 Skippe... 0.0034 sec
```

Anabel Martínez Ramírez

Ahora hago todos los ALTER TABLE para adecuar el tipo de dato de cada columna:

```
162 • ALTER TABLE users MODIFY COLUMN id INT;
163 • ALTER TABLE users MODIFY COLUMN name VARCHAR(30);
164 • ALTER TABLE users MODIFY COLUMN surname VARCHAR(50);
165 • ALTER TABLE users MODIFY COLUMN phone VARCHAR(20);
166 • ALTER TABLE users MODIFY COLUMN email VARCHAR(40);
167 • ALTER TABLE users MODIFY COLUMN birth_date DATE;
168 • ALTER TABLE users MODIFY COLUMN country VARCHAR(30);
169 • ALTER TABLE users MODIFY COLUMN city VARCHAR(30);
170 • ALTER TABLE users MODIFY COLUMN postal_code VARCHAR(20);
171 • ALTER TABLE users MODIFY COLUMN address VARCHAR(100);
172 |
```

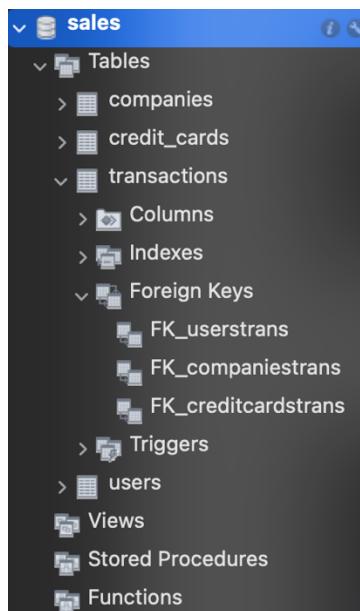
Action Output					
	Time	Action	Response	Duration / Fetch Time	
✓ 1	14:55:37	ALTER TABLE users MODIFY COLUMN id INT	275 row(s) affected Records: 275 Duplicates: 0 War...	0.038 sec	
✓ 2	14:55:38	ALTER TABLE users MODIFY COLUMN name VARCHAR(30)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.021 sec	
✓ 3	14:55:38	ALTER TABLE users MODIFY COLUMN surname VARCHAR(50)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.014 sec	
✓ 4	14:55:39	ALTER TABLE users MODIFY COLUMN phone VARCHAR(20)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.020 sec	
✓ 5	14:55:40	ALTER TABLE users MODIFY COLUMN email VARCHAR(40)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.028 sec	
✓ 6	14:55:40	ALTER TABLE users MODIFY COLUMN birth_date DATE	275 rows affected Records: 275 Duplicates: 0 War...	0.029 sec	
✓ 7	14:55:41	ALTER TABLE users MODIFY COLUMN country VARCHAR(30)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.034 sec	
✓ 8	14:55:42	ALTER TABLE users MODIFY COLUMN city VARCHAR(30)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.017 sec	
✓ 9	14:55:43	ALTER TABLE users MODIFY COLUMN postal_code VARCHAR(20)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.036 sec	
✓ 10	14:55:44	ALTER TABLE users MODIFY COLUMN address VARCHAR(100)	275 row(s) affected Records: 275 Duplicates: 0 War...	0.016 sec	

Y creo la Foreign Key entre “transactions” y “users” que relaciona “user_id” con “id”:

```
173 • ALTER TABLE transactions
174 ADD CONSTRAINT FK_userstrans
175 FOREIGN KEY (user_id) REFERENCES users(id);
176 |
```

Action Output					
	Time	Action	Response	Duration / Fetch Time	
✓ 1	14:56:04	ALTER TABLE transactions ADD CONSTRAINT FK_userstrans FOREIGN KEY (user_id) REFERENCES users(id)	587 row(s) affected Records: 587 Duplicates: 0 War...	0.038 sec	

Con esto, ya tenemos la base de datos “sales” creada con las cuatro tablas:



Exercici 1:

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

```

177
178     # Exercici 1: Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.
179
180 •   SELECT *
181     FROM users
182     WHERE id IN(SELECT COUNT(DISTINCT id) AS conteo
183                  FROM transactions
184                 GROUP BY user_id
185                HAVING conteo > 30);
186
100%  21:185 |
```

Result Grid

id	name	surname	phone	email	birth_date	country	city	postal_code	address
39	Nyssa	Shaffer	1899494941	malesuada.fames@google.ca	1991-07-02	United States	Columbus	27828	4215 Libero Road
48	Patrick	Reyes	16077295993	conubia.nostra@icloud.com	1986-07-29	United States	San Jose	96740	1150 Etiam Av.
52	Sheila	Sullivan	4266081653	arcu.vel@google.edu	1981-05-17	United States	Sacramento	48147	Ap #644-7695 Nec, Rd.
76	Glenna	Gutierrez	5674285368	amet@outlook.co.uk	1990-09-25	United States	Newark	58673	9288 Posuere St.
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

users 18

Action Output

Time	Action	Response	Duration / Fetch Time
93 12:40:01	SELECT * FROM users WHERE id IN(SELECT COUNT(DISTINCT id) AS conteo FROM transactions...	4 row(s) returned	0.065 sec / 0.000014...

En este caso, he decidido poner todos los datos (hacer un *) de los usuarios porque no se especifica que se deba seleccionar un dato concreto.

Exercici 2:

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

```

187     # Exercici 2: Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd,
188     #utilitza almenys 2 taules.
189
190 •   SELECT c.company_name, cc.iban, ROUND(AVG(t.amount), 2) AS average
191     FROM companies c
192     INNER JOIN transactions t
193     ON c.company_id = t.business_id
194     INNER JOIN credit_cards cc
195     ON t.card_id = cc.id
196     WHERE c.company_name = 'Donec Ltd' AND declined = 0
197     GROUP BY cc.iban;
198
100%  1:189 |
```

Result Grid

company_name	iban	average
Donec Ltd	PT87806228135092429456346	42.82

Result 31

Action Output

Time	Action	Response	Duration / Fetch Time
1 15:08:22	SELECT c.company_name, cc.iban, ROUND(AVG(t.amount), 2) AS average FROM...	1 row(s) returned	0.0074 sec / 0.00001...

En este caso, he decidido poner un “declined = 0” para mostrar la media de transacciones reales y aprobadas de la empresa Donec Ltd.

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:

Primero, creo la tabla “cc_status” con las columnas “card_id” como PRIMARY KEY, la columna “expiring_date” y la columna “status”.

```
203 • CREATE TABLE IF NOT EXISTS cc_status(
204     card_id VARCHAR(15) PRIMARY KEY,
205     expiring_date DATE,
206     status VARCHAR(15)
207 );
208
```

Action Output

Time	Action	Response	Duration / Fetch Time
16:11:05	CREATE TABLE IF NOT EXISTS cc_status(card_id VARCHAR(15) PRIMARY KEY, ex...	0 row(s) affected	0.047 sec

Luego, voy a insertar los datos desde las tablas transactions y credit_cards desde un SELECT:

```
209 • INSERT INTO cc_status (card_id, expiring_date, status)
210     SELECT c.id AS card_id, c.expiring_date,
211             CASE WHEN SUM(CASE WHEN t.declined THEN 1 ELSE 0 END) = 3 THEN 'Declined' ELSE 'Active' END AS status
212         FROM (SELECT t1.*
213             FROM transactions t1
214             WHERE (SELECT COUNT(*)
215                 FROM transactions t2
216                 WHERE t2.card_id = t1.card_id AND t2.timestamp > t1.timestamp) < 3
217         ) t
218     JOIN credit_cards c ON t.card_id = c.id
219     GROUP BY c.id, c.expiring_date;
220
```

Action Output

Time	Action	Response	Duration / Fetch Time
15:24:15	INSERT INTO cc_status (card_id, expiring_date, status) SELECT c.id AS card_id, c... 275 row(s) affected Records: 275 Duplicates: 0 War...	0.019 sec	0.019 sec

Y creo la Foreign Key entre “transactions” y “cc_status” que relaciona “card_id” con “card_id”:

```
222 • ALTER TABLE transactions
223     ADD CONSTRAINT FK_transcardstatus
224     FOREIGN KEY (card_id) REFERENCES cc_status(card_id);
225
```

Action Output

Time	Action	Response	Duration / Fetch Time
15:25:53	ALTER TABLE transactions ADD CONSTRAINT FK_transcardstatus FOREIGN KEY (... 587 row(s) affected Records: 587 Duplicates: 0 War...	0.045 sec	0.045 sec

Exercici 1:

Quantes targetes estan actives?

The screenshot shows a MySQL Workbench interface. In the SQL editor, line 228 contains the query: `SELECT COUNT(card_id) as active_cards FROM cc_status WHERE status = 'active';`. The result grid shows one row with `active_cards` set to 275. The action output table shows a single log entry for the query execution.

Action	Time	Response	Duration / Fetch Time
SELECT COUNT(card_id) as active_cards FROM cc_status WHERE status = 'active';	15:26:53	1 row(s) returned	0.00036 sec / 0.000...

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:

Primero, creo la tabla “products” con las columnas “id” como PRIMARY KEY. He creado esta tabla con todos los datos de tipo VARCHAR(255) porque no conocemos el tipo de dato y ya lo cambiaré más adelante.

A continuación, cargo los datos desde el csv. En este caso, el separador de este csv es la “,”.

He decidido eliminar el símbolo “\$” para que sea más fácil manipular los datos en caso de que sea necesario.

También he observado que los datos de la columna “warehouse_id” a veces tienen dos guiones y a veces uno, por lo que he decidido hacer que todos sean iguales, con un solo guion.

The screenshot shows the creation of the `products` table and its population from a CSV file. Line 236 creates the table with columns `id`, `product_name`, `price`, `colour`, `weight`, and `warehouse_id`. Line 245 loads data from `products.csv` into the table, handling commas as the field separator and ignoring the first row. The action output table shows the creation of the table and the successful import of 100 rows.

Action	Time	Response	Duration / Fetch Time
CREATE TABLE IF NOT EXISTS products(id VARCHAR(255) PRIMARY KEY, product_name VARCHAR(255), price VARCHAR(255), colour VARCHAR(255), weight VARCHAR(255), warehouse_id VARCHAR(255));	16:30:17	0 row(s) affected	0.034 sec
LOAD DATA INFILE '/Volumes/NO NAME/2. DATOS/1. SQL/Sprint 4/products.csv' INTO TABLE products FIELDS TERMINATED BY ',' IGNORE 1 ROWS (id, product_name, @price, colour, weight, @warehouse_id);	16:30:48	100 row(s) affected Records: 100 Deleted: 0 Skipped... 0.0081 sec	

Anabel Martínez Ramírez

Ahora hago todos los ALTER TABLE para adecuar el tipo de dato de cada columna:

```
258 • ALTER TABLE products MODIFY COLUMN id INT;
259 • ALTER TABLE products MODIFY COLUMN product_name VARCHAR(100);
260 • ALTER TABLE products MODIFY COLUMN price FLOAT;
261 • ALTER TABLE products MODIFY COLUMN colour VARCHAR(20);
262 • ALTER TABLE products MODIFY COLUMN weight FLOAT;
263 • ALTER TABLE products MODIFY COLUMN warehouse_id VARCHAR(10);
264
```

Action Output

Time	Action	Response	Duration / Fetch Time
1 16:36:34	ALTER TABLE products MODIFY COLUMN id INT	100 row(s) affected Records: 100 Duplicates: 0 Warn... 0.047 sec	
2 16:36:35	ALTER TABLE products MODIFY COLUMN product_name VARCHAR(100)	100 row(s) affected Records: 100 Duplicates: 0 Warn... 0.0090 sec	
3 16:36:36	ALTER TABLE products MODIFY COLUMN price FLOAT	100 row(s) affected Records: 100 Duplicates: 0 Warn... 0.015 sec	
4 16:36:36	ALTER TABLE products MODIFY COLUMN colour VARCHAR(20)	100 row(s) affected Records: 100 Duplicates: 0 Warn... 0.030 sec	
5 16:36:37	ALTER TABLE products MODIFY COLUMN weight FLOAT	100 row(s) affected Records: 100 Duplicates: 0 Warn... 0.016 sec	
6 16:36:38	ALTER TABLE products MODIFY COLUMN warehouse_id VARCHAR(10)	100 row(s) affected Records: 100 Duplicates: 0 Warn... 0.017 sec	

Ahora creo la tabla “transaction_products” para que haga de unión entre la tabla “products” y la tabla “transactions”. En esta tabla, solo existen las columnas “transaction_id” y “product_id”, y la combinación de ambas es lo que funciona como PRIMARY KEY.

```
256 • CREATE TABLE IF NOT EXISTS transaction_products (
257     transaction_id VARCHAR(50),
258     product_id INT,
259     PRIMARY KEY (transaction_id, product_id)
260 );
261
```

Action Output

Time	Action	Response	Duration / Fetch Time
11 15:28:43	CREATE TABLE IF NOT EXISTS transaction_products (transaction_id VARCHAR... 0 row(s) affected		0.0087 sec

Ahora inserto los datos. Para hacerlo, he usado un SELECT que utiliza el comando FIND IN SET.

```
263 • INSERT INTO transaction_products (transaction_id, product_id)
264     SELECT
265         t.id as transaction_id,
266         p.id as product_id
267     FROM transactions as t
268     JOIN products p
269     ON FIND_IN_SET(p.id, t.product_ids) > 0;
270
271
```

Action Output

Time	Action	Response	Duration / Fetch Time
12 15:29:22	INSERT INTO transaction_products (transaction_id, product_id) SELECT t.id as... 1457 row(s) affected Records: 1457 Duplicates: 0 W... 0.026 sec		

Finalmente, creo la Foreign Key entre “transactions” y “transaction_products”, que relaciona “transaction_id” con “id”. También creo la Foreign Key entre “products” y “transaction_products”, que relaciona “product_id” con “id”.

```
273
274 • ALTER TABLE transaction_products ADD FOREIGN KEY (transaction_id) REFERENCES transactions(id);
275
276 • ALTER TABLE transaction_products ADD FOREIGN KEY (product_id) REFERENCES products(id);
277
17:270

Action Output
Time Action Response Duration / Fetch Time
2 15:19:26 ALTER TABLE transaction_products ADD FOREIGN KEY (transaction_id) REFEREN... 1457 row(s) affected Records: 1457 Duplicates: 0 W... 0.063 sec
3 15:19:30 ALTER TABLE transaction_products ADD FOREIGN KEY (product_id) REFERENCES... 1457 row(s) affected Records: 1457 Duplicates: 0 W... 0.043 sec
```

Exercici 1:

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

```
282 #Exercici 1:
283 #Necessitem conèixer el nombre de vegades que s'ha venut cada producte.
284
285 • SELECT tp.product_id, p.product_name, COUNT(DISTINCT tp.transaction_id) AS items_sold
286   FROM transaction_products tp
287   JOIN products p
288  ON tp.product_id = p.id
289   JOIN transactions t
290  ON t.id = tp.transaction_id
291 WHERE t.declined = 0
292 GROUP BY tp.product_id;
100% 1:284

Result Grid Filter Rows: Search Export: Result Grid
Form Editor
Read Only

product_id product_name items_sold
1 Direwolf Stannis 51
2 Tarly Stark 56
3 duel tourney Lannister 43
5 skywalker ewok 42
7 north of Casterly 44
11 Karstark Dorne 40

Result 4

Action Output
Time Action Response Duration / Fetch Time
1 07:22:09 SELECT tp.product_id, p.product_name, COUNT(DISTINCT tp.transaction_id) AS i... 26 row(s) returned 0.0042 sec / 0.00001...
```

En este caso, he decidido incluir tanto el id del producto como el nombre del producto para que haya una mayor claridad en el momento de la identificación del mismo.

Como se menciona que buscamos el número de veces que se ha vendido cada producto, he incluido el “declined = 0” para asegurarme de que las transacciones han sido exitosas.

Anabel Martínez Ramírez

Por último, este es el esquema final que resulta de añadir todas las tablas a la base de datos. El diagrama final es de copo de nieve y hemos conseguido una relación de muchos a muchos entre “transactions” y “products” gracias a la tabla “transaction_products”.

