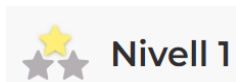


SPRINT 4

Partint d'alguns arxius CSV dissenyaràs i crearàs la teva base de dades.

Important

Totes les transformacions i importacions que se't demanen en aquesta tasca s'han de realitzar utilitzant codi SQL. NO ES PERMET fer els canvis fent servir el Wizard.



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  -- Crear la base de datos nueva
2
3  CREATE DATABASE IF NOT EXISTS new_db_transactions;
4
5  USE new_db_transactions;
6
```

#	Time	Action	Message	Duration / Fetch
1	13:21:51	CREATE DATABASE IF NOT EXISTS new_db_transactions	1 row(s) affected	0.000 sec
2	13:21:51	USE new_db_transactions	0 row(s) affected	0.000 sec

Después de revisar los archivos proporcionados y las preguntas de los ejercicios, crearé, de momento, la base de datos con 4 tablas (companies, transactions, users y credit_cards) para poder tener un esquema en estrella. La tabla products se creará en el nivel 3.

Crearé primero cada una de las 4 tablas y utilizaré los archivos CSV para cargar los datos en ellas.

```
7  -- crear la tabla companies
8
9  CREATE TABLE IF NOT EXISTS companies (
10     company_id VARCHAR(20) PRIMARY KEY,
11     company_name VARCHAR(255),
12     phone VARCHAR(15),
13     email VARCHAR(100),
14     country VARCHAR(100),
15     website VARCHAR(255)
16 );
17
```

#	Time	Action	Message	Duration / Fetch
1	13:21:51	CREATE DATABASE IF NOT EXISTS new_db_transactions	1 row(s) affected	0.000 sec
2	13:21:51	USE new_db_transactions	0 row(s) affected	0.000 sec
3	13:36:50	CREATE TABLE IF NOT EXISTS companies (company_id VARCHAR(20) PRIMARY KEY, company_nam...	0 row(s) affected	0.031 sec

Al intentar cargar los datos en la tabla companies tuve un error porque MySQL tiene activa la opción secure-file-priv

```
18 -- cargar los datos del archivo companies.csv en la tabla
19
20 • LOAD DATA INFILE 'C:\Users\tinke\Desktop\SQL\aprint 4\companies.csv'
21 INTO TABLE companies
22 FIELDS TERMINATED BY ','
23 LINES TERMINATED BY '\r\n'
24 IGNORE 1 LINES;
25
```

Output

#	Time	Action	Message	Duration / Fetch
1	11:23:35	LOAD DATA INFILE 'C:\Users\tinke\Desktop\SQL\aprint 4\companies.csv' INTO TABLE companies FIELDS T...	Error Code: 1290. The MySQL server is running with the --secure-file-priv option so it cannot execute this statement	0.015 sec

Para poder ingresar los datos tuve que buscar en qué carpeta se tienen que guardar los archivos y copiarlos en la correcta que acepta el modo de configuración de MySQL y finalmente los pude ingresar correctamente

```
26 • SHOW VARIABLES LIKE "secure_file_priv"; -- revisar en que carpeta se tienen que guardar los archivos de datos para poder luego cargarlos
27
28 -- cargar los datos del archivo companies.csv en la tabla
29
30 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv'
31 INTO TABLE companies
32 FIELDS TERMINATED BY ',' -- cada campo se separa por coma
33 LINES TERMINATED BY '\r\n' -- el salto de línea en windows(CRLF)
34 IGNORE 1 LINES;
```

Result Grid

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

Result 1

Output

#	Time	Action	Message	Duration / Fetch
1	11:23:35	LOAD DATA INFILE 'C:\Users\tinke\Desktop\SQL\aprint 4\companies.csv' INTO TABLE companies FIELDS T...	Error Code: 1290. The MySQL server is running with the --secure-file-priv option so it cannot execute this statement	0.015 sec
2	11:34:47	SHOW VARIABLES LIKE "secure_file_priv"	1 row(s) returned	0.000 sec / 0.000 sec
3	11:34:47	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv' INTO TABLE comp...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.000 sec

Para los usuarios, crearé una sola tabla e ingresaré en ella los tres diferentes archivos con los datos de los usuarios de los diferentes países.

```
36 -- crear la siguiente tabla, crearé una sola tabla para agregar en ella los tres archivos que contienen la información de los usuarios
37 • CREATE TABLE IF NOT EXISTS users (
38     id INT PRIMARY KEY,
39     name VARCHAR(100),
40     surname VARCHAR(100),
41     phone VARCHAR(150),
42     email VARCHAR(150),
43     birth_date VARCHAR(100),
44     country VARCHAR(150),
45     city VARCHAR(150),
46     postal_code VARCHAR(100),
47     address VARCHAR(255)
48 );
```

Output

#	Time	Action	Message	Duration / Fetch
2	11:34:47	SHOW VARIABLES LIKE "secure_file_priv"	1 row(s) returned	0.000 sec / 0.000 sec
3	11:34:47	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv' INTO TABLE com...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.000 sec
4	11:39:33	CREATE TABLE IF NOT EXISTS users (id INT PRIMARY KEY, name VARCHAR(100), sur...	0 row(s) affected	0.016 sec

```

50 -- primero insertaré los datos del archivo users_usa.csv
51 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv'
52 INTO TABLE users
53 FIELDS TERMINATED BY ','          -- cada campo se separa por coma
54 ENCLOSED BY '"'                  -- los valores del campo 'birth_date' están entre comillas
55 LINES TERMINATED BY '\r\n'       -- el salto de línea en windows(CRLF)
56 IGNORE 1 LINES;
57
58 -- luego el segundo archivo users_uk.csv
59 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv'
60 INTO TABLE users
61 FIELDS TERMINATED BY ','
62 ENCLOSED BY '"'
63 LINES TERMINATED BY '\r\n'
64 IGNORE 1 LINES;
65
66 -- por último el tercer archivo users_ca.csv
67 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv'
68 INTO TABLE users
69 FIELDS TERMINATED BY ','
70 ENCLOSED BY '"'
71 LINES TERMINATED BY '\r\n'
72 IGNORE 1 LINES;
73
74

```

Output

#	Time	Action	Message	Duration / Fetch
5	11:46:19	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv' INTO TABLE users	150 row(s) affected Records: 150 Deleted: 0 Skipped: 0 Warnings: 0	0.000 sec
6	11:46:19	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv' INTO TABLE users	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0	0.016 sec
7	11:46:19	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv' INTO TABLE users	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0	0.000 sec

```

74 -- crear la tabla credit_cards
75 • CREATE TABLE IF NOT EXISTS credit_cards (
76     id VARCHAR(15) PRIMARY KEY,
77     user_id INT,
78     iban VARCHAR(34),
79     pan VARCHAR(23),
80     pin VARCHAR(4),
81     cvv VARCHAR(4),
82     track1 VARCHAR(255),
83     track2 VARCHAR(255),
84     expiring_date VARCHAR(10)
85 );
86 -- insertar los datos del archivo credit_cards.csv
87 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv'
88 INTO TABLE credit_cards
89 FIELDS TERMINATED BY ','          -- cada campo se separa por coma
90 LINES TERMINATED BY '\n'         -- el salto de línea de Unix(LF)
91 IGNORE 1 LINES;
92
93

```

Output

#	Time	Action	Message	Duration / Fetch
7	11:46:19	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv' INTO TABLE users	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0	0.000 sec
8	11:50:55	CREATE TABLE IF NOT EXISTS credit_cards (id VARCHAR(15) PRIMARY KEY, user_id INT, ...	0 row(s) affected	0.031 sec
9	11:50:55	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv' INTO TABLE cr...	275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings: 0	0.016 sec

```

93 -- crear transaction que es la última tabla
94 CREATE TABLE IF NOT EXISTS transactions (
95     id VARCHAR(255) PRIMARY KEY,
96     card_id VARCHAR(15),
97     business_id VARCHAR(20),
98     timestamp TIMESTAMP,
99     amount DECIMAL(10, 2),
100     declined BOOLEAN,
101     product_ids VARCHAR(100),
102     user_id INT,
103     lat FLOAT,
104     longitude FLOAT
105 );
106
107 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'
108 INTO TABLE transactions
109 FIELDS TERMINATED BY ';' -- el separador de los campos en este archivo es punto y coma (;)
110 LINES TERMINATED BY '\r\n' -- el salto de línea en windows(CRLF)
111 IGNORE 1 LINES;
112

```

#	Time	Action	Message	Duration / Fetch
9	11:50:55	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv' INTO TABLE cr...	275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings: 0	0.016 sec
10	11:54:51	CREATE TABLE IF NOT EXISTS transactions (id VARCHAR(255) PRIMARY KEY, card_id VARCH...	0 row(s) affected	0.031 sec
11	11:54:51	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv' INTO TABLE tra...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0	0.016 sec

El siguiente paso después de la creación de las tablas y la carga de los archivos csv es crear las relaciones entre ellas, definiendo las Foreign Key de la tabla de hechos ‘transactions’. Las Primary Key se indicaron al momento de la creación de las tablas.

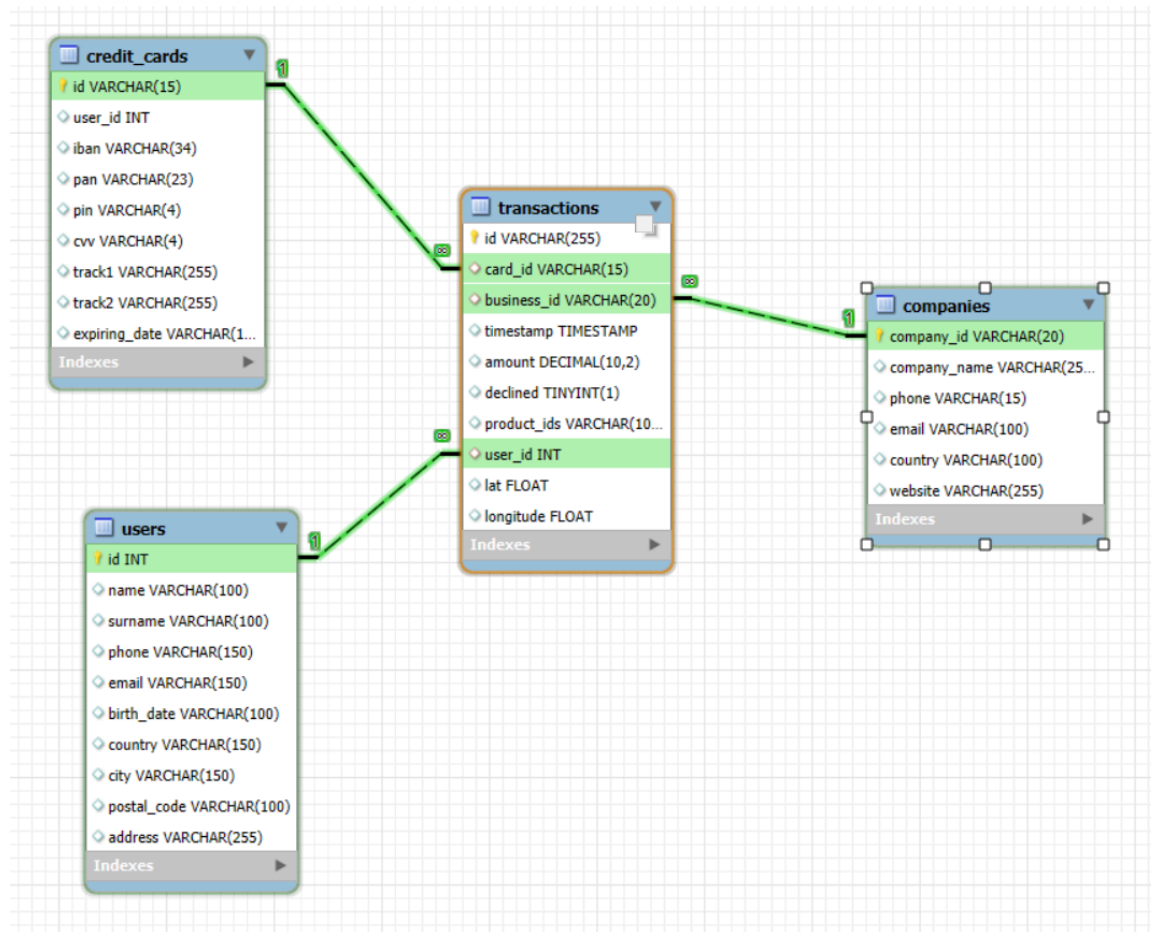
```

113 -- proximo paso es definir las FK en la tabla de hechos transactions para poder conectar las tablas de dimensiones con ella
114
115 ALTER TABLE transactions -- crear FK para conectar la tabla transactions con la tabla companies
116 ADD CONSTRAINT fk_company_transaction
117 FOREIGN KEY (business_id) REFERENCES companies(company_id);
118
119 ALTER TABLE transactions -- crear FK para conectar la tabla transactions con la tabla users
120 ADD CONSTRAINT fk_user_transaction
121 FOREIGN KEY (user_id) REFERENCES users(id);
122
123 ALTER TABLE transactions -- crear FK para conectar la tabla transactions con la tabla credit_cards
124 ADD CONSTRAINT fk_creditcard_transaction
125 FOREIGN KEY (card_id) REFERENCES credit_cards(id);
126

```

#	Time	Action	Message
12	11:56:33	ALTER TABLE transactions-- crear FK para conectar la tabla transactions con la tabla companies ADD CONS...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
13	11:56:33	ALTER TABLE transactions-- crear FK para conectar la tabla transactions con la tabla users ADD CONSTRAI...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
14	11:56:33	ALTER TABLE transactions-- crear FK para conectar la tabla transactions con la tabla credit_cards ADD CON...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

El diagrama de la base de datos creada en esquema de estrella es el siguiente:



La tabla **transactions**, que contine la informaci3n de las transacciones realizadas, es la **tabla de hechos** y las tablas **companies**, **users** y **credit_cards** son las **tablas de dimensiones**.

La tabla **companies** se conecta con **transactions** con una relaci3n de **1:N** mediante su PK `company_id` y la FK `business_id` de `transactions`.

La tabla **users** se conecta con **transactions** con una relaci3n de **1:N** mediante su PK `id` y la FK `user_id` de `transactions`.

La tabla **credit_cards** se conecta con **transactions** con una relaci3n de **1:N** mediante su PK `id` y la FK `card_id` de `transactions`.

- Exercici 1

Realitza una subconsulta que mostri tots els usuaris amb m3s de 30 transaccions utilitzant almenys 2 taules.

```

129 • SELECT name, surname
130 FROM users
131 WHERE id IN (
132     SELECT (user_id)
133     FROM transactions
134     GROUP BY user_id
135     HAVING COUNT(id) > 30
136 );

```

name	surname
Lynn	Riddle
Ocean	Nelson
Hedwig	Gilbert
Kenyon	Hartman

users 2 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 13	11:56:33	ALTER TABLE transactions-- crear FK para conectar la tabla transactions con la tabla users ADD CONSTRAI...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0	0.032 sec
✓ 14	11:56:33	ALTER TABLE transactions-- crear FK para conectar la tabla transactions con la tabla credit_cards ADD CON...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0	0.047 sec
✓ 15	12:20:46	SELECT name, surname FROM users WHERE id IN (SELECT (user_id) FROM transactions GROU...	4 row(s) returned	0.000 sec / 0.000 sec

- Exercici 2

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

```

138 -- Exercici 2 : Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.
139 • SELECT cc.iban AS card_iban, ROUND(AVG(tr.amount), 2) AS average_amount
140 FROM credit_cards cc
141 JOIN transactions tr
142 ON cc.id = tr.card_id
143 JOIN companies c
144 ON tr.business_id = c.company_id
145 WHERE c.company_name = "Donec Ltd" AND tr.declined = 0
146 GROUP BY cc.iban;

```

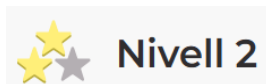
card_iban	average_amount
PT87806228135092429456346	42.82

Result 3 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 14	11:56:33	ALTER TABLE transactions-- crear FK para conectar la tabla transactions con la tabla credit_cards ADD CON...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0	0.047 sec
✓ 15	12:20:46	SELECT name, surname FROM users WHERE id IN (SELECT (user_id) FROM transactions GROU...	4 row(s) returned	0.000 sec / 0.000 sec
✓ 16	12:22:24	SELECT cc.iban AS card_iban, ROUND(AVG(tr.amount), 2) AS average_amount FROM credit_cards cc JOIN ...	1 row(s) returned	0.000 sec / 0.000 sec



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:

Para poder crear la nueva tabla que muestre el estado de las tarjetas de crédito se tiene que utilizar la función ventana ROW_NUMBER() OVER(...) para enumerar las transacciones hechas. Necesitaremos también una agrupación

por la id de cada tarjeta y ordenar las transacciones por fecha descendente, utilizando PARTITION BY. Cuando se crea la partición, la función ventana añade el número de secuencia de la transacción ordenado, que como paso final, filtraremos con la condición necesaria.

```

148 -- Nivell 2
149 -- Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions van ser declinades
150
151 • CREATE TABLE IF NOT EXISTS credit_card_status AS
152     SELECT card_id,
153     CASE
154         WHEN SUM(declined) >= 3 THEN 'INACTIVE' -- declined: 1 = TRUE, 0 = FALSE
155         ELSE 'ACTIVE'
156     END AS card_status
157     FROM (
158         SELECT card_id, declined, ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS line_number
159         FROM transactions
160         ) AS ordered_transactions
161     WHERE line_number <= 3
162     GROUP BY card_id;
163
Output:
Action Output
# Time Action Message Duration / Fetch
1 13:54:58 CREATE TABLE IF NOT EXISTS credit_card_status AS SELECT card_id, CASE WHEN SUM(declined) >= 3 THEN 'INACTIVE' ELSE 'ACTIVE' END AS card_status FROM (SELECT card_id, declined, ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS line_number FROM transactions) AS ordered_transactions WHERE line_number <= 3 GROUP BY card_id; 275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0 0.032 sec

164 -- revisar la tabla credit_card_status
165
166 • SELECT * FROM credit_card_status;
167
Result Grid
card_id card_status
CdU-2938 ACTIVE
CdU-2945 ACTIVE
CdU-2952 ACTIVE
CdU-2959 ACTIVE
CdU-2966 ACTIVE
CdU-2973 ACTIVE
CdU-2980 ACTIVE
credit_card_status 5 x
Read Only

Output:
Action Output
# Time Action Message Duration / Fetch
1 13:54:58 CREATE TABLE IF NOT EXISTS credit_card_status AS SELECT card_id, CASE WHEN SUM(declined) >= 3 THEN 'INACTIVE' ELSE 'ACTIVE' END AS card_status FROM (SELECT card_id, declined, ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS line_number FROM transactions) AS ordered_transactions WHERE line_number <= 3 GROUP BY card_id; 275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0 0.032 sec
2 13:56:30 SELECT * FROM credit_card_status LIMIT 0, 1000 275 row(s) returned 0.000 sec / 0.000 sec

```

Añadir las PK y FK (card_id) para conectar credit_card_status con la tabla credit_cards por la id de la última.

```

168 -- crear la PK y FK para poder relacionar la tabla correctamente con la credit_cards
169 • ALTER TABLE credit_card_status
170     ADD PRIMARY KEY (card_id);
171
172 • ALTER TABLE credit_card_status
173     ADD FOREIGN KEY (card_id) REFERENCES credit_cards(id);
174
175
Output:
Action Output
# Time Action Message Duration / Fetch
2 13:56:30 SELECT * FROM credit_card_status LIMIT 0, 1000 275 row(s) returned 0.000 sec / 0.000 sec
3 13:58:03 ALTER TABLE credit_card_status ADD PRIMARY KEY (card_id) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.047 sec
4 13:58:03 ALTER TABLE credit_card_status ADD FOREIGN KEY (card_id) REFERENCES credit_cards(id) 275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0 0.031 sec

```

- Exercici 1

Quantes targetes estan actives?

```

176 -- Quantes targetes estan actives?
177
178 • SELECT COUNT(card_id) AS total_active_cards
179 FROM credit_card_status
180 WHERE card_status = 'ACTIVE';
181

```

Result Grid

total_active_cards
275

Result 6 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
5	14:01:56	ALTER TABLE credit_card_status ADD PRIMARY KEY (card_id)	Error Code: 1068. Multiple primary key defined	0.000 sec
6	14:07:09	SELECT COUNT(card_id) AS total_active_cards FROM credit_card_status WHERE card_status = 'ACTIVE' LI...	1 row(s) returned	0.000 sec / 0.000 sec

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 hay que crear la tabla products, que no se creó al principio y cargar los datos del archivo products.csv en ella.

```

182 -- Nivell 3
183 -- 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.
184
185 • CREATE TABLE IF NOT EXISTS products (
186     id VARCHAR(100) PRIMARY KEY,
187     product_name VARCHAR(255),
188     price VARCHAR(30),
189     colour VARCHAR(30),
190     weight FLOAT,
191     warehouse_id VARCHAR(50)
192 );
193
194 -- ingresar los datos del archivo products.csv a la tabla
195
196 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv'
197 INTO TABLE products
198 FIELDS TERMINATED BY ','          -- cada campo se separa por coma
199 LINES TERMINATED BY '\n'         -- el salto de línea Unix(LF)
200 IGNORE 1 LINES;
201

```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
8	14:28:39	CREATE TABLE IF NOT EXISTS products (id VARCHAR(100) PRIMARY KEY, product_name V...	0 row(s) affected	0.047 sec
9	14:28:39	LOAD DATA INFILE C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv INTO TABLE produ...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.016 sec


```

202 -- mostrar la tabla products
203 • SELECT * FROM products;
204

```

id	product_name	price	colour	weight	warehouse_id
1	Direwolf Stannis	\$161.11	#7c7c7c	1	WH-4
10	Karstark Dome	\$119.52	#f4f4f4	2.4	WH-5
100	south duel	\$40.43	#6d6d6d	3	WH-95
11	Karstark Dome	\$49.70	#141414	2.7	WH-6
12	duel Direwolf	\$181.60	#a8a8a8	2.1	WH-7
13	palpatine chewbacca	\$139.59	#2b2b2b	1	WH-8
14	Direwolf	\$147.53	#c4c4c4	2	WH-9
15	Stannis warden	\$194.29	#bdbdbd	1.5	WH-10
16	the duel warden	\$180.91	#666666	3	WH-11
17	skywalker ewok sith	\$91.89	#7c7c7c	3.2	WH-12
18	Karstark warden	\$148.91	#c4c4c4	0.8	WH-13

products 7 x

Output:

#	Time	Action	Message	Duration / Fetch
1	14:34:50	SELECT * FROM products LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec

Los valores de la columna 'precio' tienen el símbolo \$ antes de los números, lo eliminaré y cambiaré el tipo de datos a decimales.

```

191 -- tendré que actualizar la columna price eliminando el símbolo $ y el tipo de los datos a decimal para poder realizar consultas
192
193 • UPDATE products
194 SET price = REPLACE(price, "$", ""); -- -- eliminar el símbolo $ que aparece antes de los dígitos del precio

```

Output:

#	Time	Action	Message	Duration / Fetch
1	21:51:45	UPDATE products	SET price = REPLACE(price, "\$", "");	Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a

```

205 -- actualizar la columna price eliminando el símbolo $ y el tipo de los datos a decimal para poder realizar consultas
206 -- me da error por el safe update mode, para poder actualizar se tiene que desactivar y luego volver a activarse
207
208 • SET SQL_SAFE_UPDATES=0; -- desactivar safe mode
209
210 • UPDATE products
211 SET price = REPLACE(price, '$', ''); -- eliminar el símbolo $ que aparece antes de los dígitos del precio
212
213 • SET SQL_SAFE_UPDATES=1; -- volver a activar el safe mode
214
215 -- actualizar el tipo de la columna a decimal, todos ahora son valores numéricos
216
217 • ALTER TABLE products
218 MODIFY COLUMN price DECIMAL(10, 2);
219

```

Output:

#	Time	Action	Message	Duration / Fetch
2	14:37:55	SET SQL_SAFE_UPDATES=0	0 row(s) affected	0.000 sec
3	14:37:55	UPDATE products	SET price = REPLACE(price, '\$', '');	100 row(s) affected Rows matched: 100 Changed: 100 Warnings: 0
4	14:37:55	SET SQL_SAFE_UPDATES=1	0 row(s) affected	0.000 sec
5	14:37:55	ALTER TABLE products MODIFY COLUMN price DECIMAL(10, 2)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0	0.031 sec

Para conectar **products** que es una **tabla de dimensiones** con la tabla de hechos transactions se tiene que crear una **tabla intermedia** porque la relación entre ellas es de **N:M**. Por eso se crea la tabla **transaction_products** con los campos transaction_id y product_id que sirven como PK y FK para conectar la tabla con transaction por id y con products por id.

```

220 -- para realizar la relación de la tabla productos con la tabla transactions hay que crear una tabla intermedia de union porque la relación entre ellas es de N:M
221
222 CREATE TABLE IF NOT EXISTS transaction_products (
223     transaction_id VARCHAR(255) NOT NULL,
224     product_id VARCHAR(100) NOT NULL,
225     PRIMARY KEY (transaction_id, product_id),
226     FOREIGN KEY (transaction_id) REFERENCES transactions(id),
227     FOREIGN KEY (product_id) REFERENCES products(id)
228 );
229

```

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
3	14:37:55	UPDATE products SET price = REPLACE(price, '\$, ')	100 row(s) affected Rows matched: 100 Changed: 100 Warnings: 0	0.015 sec
4	14:37:55	SET SQL_SAFE_UPDATES=1	0 row(s) affected	0.000 sec
5	14:37:55	ALTER TABLE products MODIFY COLUMN price DECIMAL(10, 2)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0	0.031 sec
6	14:40:52	CREATE TABLE IF NOT EXISTS transaction_products (transaction_id VARCHAR(255) NO...	0 row(s) affected	0.031 sec

Para cargar datos en esta tabla se tienen que copiar los datos de las tablas transactions y products. En la tabla transactions, la columna product_ids contiene varios valores por línea así que necesitamos la función FIND_IN_SET para ingresar en la nueva tabla sólo los valores de product_id que se encuentran en cada transacción de la tabla transacciones.

```

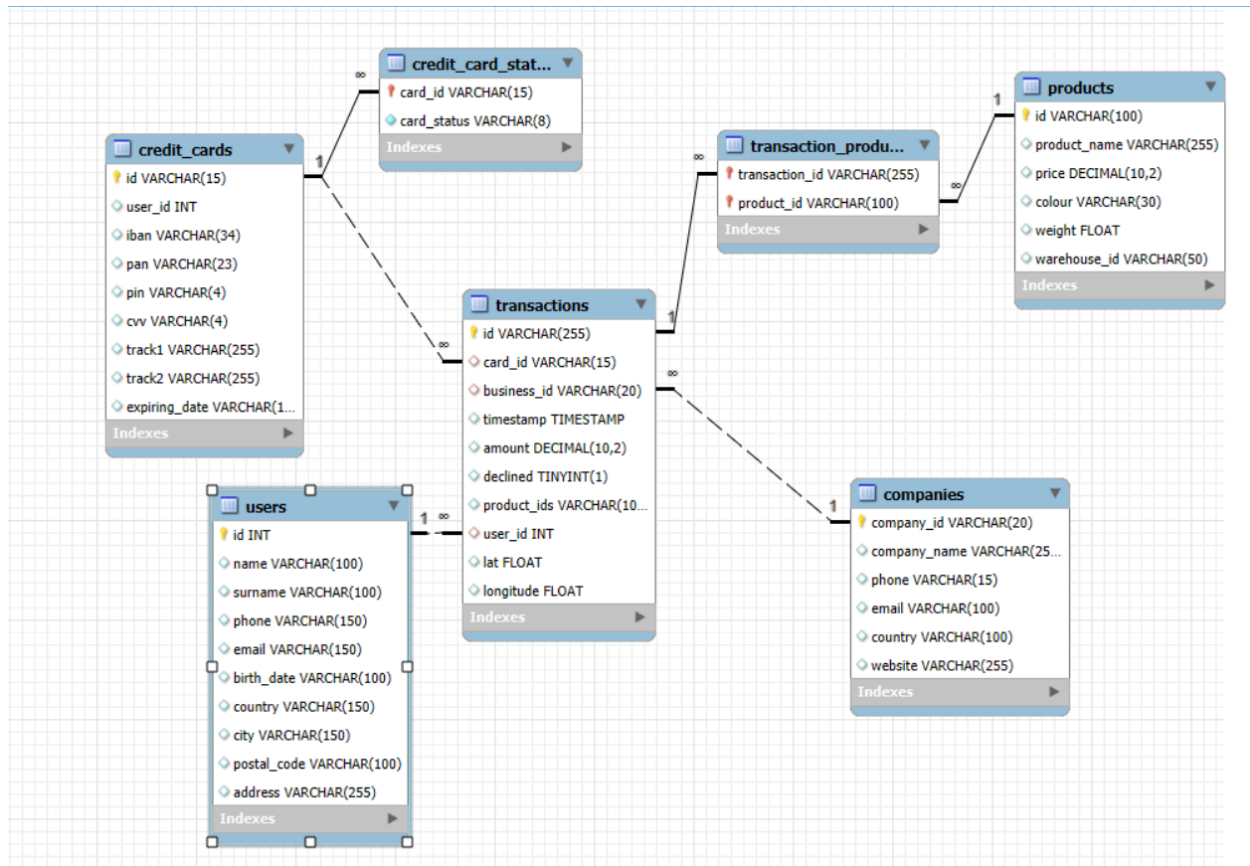
232 INSERT INTO transaction_products (transaction_id, product_id)
233 SELECT tr.id, pr.id
234 FROM transactions tr
235 JOIN products pr
236     ON FIND_IN_SET(pr.id, REPLACE(tr.product_ids, ' ', '')) > 0; -- cambiar el espacio después del separador de valores a sin espacio
237
238 -- revisar la tabla con valores
239
240 SELECT * FROM transaction_products;
241

```

Result Grid				
transaction_id	product_id			
02C6201E-D90A-1859-B4EE-88D2986D3802	1			
122DC333-E19F-D629-DCD8-9C54CF1EBB9A	1			
1753A288-9FC1-52E6-5C39-A1FFB97B0D3A	1			
1A6CECFB-2E3A-65A3-72D9-2FDB88A1E4BA	1			
1EA2B262-D507-AD14-4374-4D532967113F	1			
23CF8ED3-402C-7C54-59CD-DB505C5CCCCC	1			
2A5A3001-104F-ID1F-7852-5BA801869B6F	1			
2F386AB6-147D-EB08-FE8D-9A4E2EA90B05	1			
2F499B4D-4DC7-8337-010D-8B7471812A80	1			

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
1	21:50:42	INSERT INTO transaction_products (transaction_id, product_id) SELECT tr.id, pr.id FROM transactions tr JOIN p...	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0	0.047 sec
2	21:50:42	SELECT * FROM transaction_products LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.000 sec

Después de crear todas las tablas necesarias en la base de datos el diagrama cambia a un esquema de copo de nieve:



- Exercici 1

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

```

242 -- Genera la següent consulta: Necessitem conèixer el nombre de vegades que s'ha venut cada producte.
243
244 • SELECT pr.product_name, count(transaction_id) AS total_sold_items
245 FROM products pr
246 JOIN transaction_products tp
247 ON pr.id = tp.product_id
248 GROUP BY pr.id;
  
```

-- agrupar por id que es valor único por si existen productos con el mismo nombre

product_name	total_sold_items
Direwolf Stannis	61
Karstark Dorne	48
palpatine chevbacca	60
skywalker evok sith	61
dooku solo	49
Tarly Stark	65
riverlands north	68
Tully maester Tarly	49
duel tourney Lannister	51

Result 2 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	21:50:42	SELECT * FROM transaction_products LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.000 sec
3	21:54:46	SELECT pr.product_name, count(transaction_id) AS total_sold_items FROM products pr JOIN transaction_products tp ON pr.id = tp.product_id GROUP BY pr.id	26 row(s) returned	0.000 sec / 0.000 sec

Como último paso, aunque no se solicita, se corregirán los tipos de datos en las columnas que se entienden como fechas por el nombre de columna pero tienen un tipo de datos diferente. El cambio será de VARCHAR a DATE y esto

afecta las users.birth_date y credit_cards.expiring_date. Para ello se usa la función STR_TO_DATE para dar al valor de cadena de texto el formato de fecha.

```
254 • SET SQL_SAFE_UPDATES=0;
255
256 • UPDATE users
257 SET birth_date = STR_TO_DATE(REPLACE(birth_date,',', ''), '%b %e %Y'); -- replace para el coma en el valor, %b nombre de mes abreviado, %e día mes 0-31, %Y año 4 dígitos
258 • ALTER TABLE users
259 MODIFY COLUMN birth_date DATE; -- y modificar el tipo de datos a DATE
260
261 • UPDATE credit_cards
262 SET expiring_date = STR_TO_DATE(expiring_date,'%m/%d/%y');
263 • ALTER TABLE credit_cards
264 MODIFY COLUMN expiring_date DATE; -- %m mes con número 0-12, %d día mes 01-31, %y año 2 dígitos
265
266 • SET SQL_SAFE_UPDATES=1;
267
```

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
1	19:38:48	UPDATE users SET birth_date = STR_TO_DATE(REPLACE(birth_date,',', ''), '%b %e %Y')	275 row(s) affected Rows matched: 275 Changed: 275 Warnings: 0	0.047 sec
2	19:38:48	ALTER TABLE users MODIFY COLUMN birth_date DATE	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0	0.078 sec
3	19:38:48	UPDATE credit_cards SET expiring_date = STR_TO_DATE(expiring_date,'%m/%d/%y')	275 row(s) affected Rows matched: 275 Changed: 275 Warnings: 0	0.016 sec
4	19:38:48	ALTER TABLE credit_cards MODIFY COLUMN expiring_date DATE	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0	0.047 sec
5	19:39:30	SELECT * FROM new_db_transactions.users LIMIT 0, 1000	275 row(s) returned	0.000 sec / 0.000 sec