Tasca S4.01. Manipulació de taules

Referencia: [https://itacademy.barcelonactiva.cat/mod/assign/view.php?id=14388]
Elaborada: 04.12.2025
Author: Levitchi Alexei
P2P: Mariia Zaytseva
Archivos asociados:

- Informe_Tasca_S401.pdf
- Soluciones_Tasca_S401.sql

Contents

Nivell 1

Descripció

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

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:

Comentaris

Utilitzant les taules, puc crear la base de dades 'user_sales' (Il·lustració 1).

```
# Creamos la base de datos
    CREATE DATABASE IF NOT EXISTS user_sales;
    USE user_sales;
```

```
3  10:23:55  CREATE DATABASE IF NOT EXISTS user_sales   1 row(s) affected   0.140 sec
4  10:24:00  USE user_sales                             0 row(s) affected   0.000 sec
```

Il·lustració 1. La consulta per crear la base de dades 'user_sales'

- Creant la taula 'american_users' (Il·lustració 2)

```
16      # Creamos la tabla 'american_users'
17          # field names: id, name, surname, phone, email, birth_date, country, city, postal_code, address
18          CREATE TABLE IF NOT EXISTS american_users (
19              id VARCHAR(15) PRIMARY KEY,
20              name VARCHAR(255),
21              surname VARCHAR(255),
22              phone VARCHAR(15),
23              email VARCHAR(100),
24              birth_date VARCHAR(100),
25              country VARCHAR(100),
26              city VARCHAR(255),
27              postal_code VARCHAR(255),
28              address VARCHAR(255)
29          );
30
```

```
Output
Action Output
#   Time       Action                                                                  Message
5   10:24:13   CREATE TABLE IF NOT EXISTS american_users ( id VARCHAR(15) PRIMARY KEY, name VARCHAR(255),...   0 row(s) affected
```

Il·lustració 2. La consulta per crear la taula 'american_users'

- Creant la taula 'companies' (Il·lustració 3)

```
31      # Creamos la tabla 'companies'
32          # field names: company_id, company_name, phone, email, country, website
33          CREATE TABLE IF NOT EXISTS companies (
34              company_id VARCHAR(15) PRIMARY KEY,
35              company_name VARCHAR(255),
36              phone VARCHAR(15),
37              email VARCHAR(100),
38              country VARCHAR(100),
39              website VARCHAR(255)
40          );
```

Output

Action Output

#	Time	Action	Message
6	10:24:20	CREATE TABLE IF NOT EXISTS companies (company_id VARCHAR(15) PRIMARY KEY, company_name ...	0 row(s) affected

Il·lustració 3. La consulta per crear la taula 'companies'

- Creant la taula 'credit_cards' (Il·lustració 4)

```
42      # Creamos la tabla 'credit_cards'
43          # field names: id, user_id, iban, pan, pin, cvv, track1, track2, expiring_date
44          CREATE TABLE IF NOT EXISTS credit_cards (
45              id VARCHAR(15) PRIMARY KEY,
46              user_id VARCHAR(15),
47              iban VARCHAR(34),
48              pan VARCHAR(19),
49              pin VARCHAR(6),
50              cvv VARCHAR(4),
51              track1 VARCHAR(255),
52              track2 VARCHAR(255),
53              expiring_date VARCHAR(8)
54          );
```

Output

Action Output

#	Time	Action	Message
7	10:24:28	CREATE TABLE IF NOT EXISTS credit_cards (id VARCHAR(15) PRIMARY KEY, user_id VARCHAR(15), ib...	0 row(s) affected

Il·lustració 4. La consulta per crear la taula 'credit_cards'

- Creant la taula 'european_users' (Il·lustració 5)

```
56      # Creamos la tabla 'european_users'
57          # field names: id, name, surname, phone, email, birth_date, country, city, postal_code, address
58          CREATE TABLE IF NOT EXISTS european_users (
59              id VARCHAR(15) PRIMARY KEY,
60              name VARCHAR(255),
61              surname VARCHAR(255),
62              phone VARCHAR(15),
63              email VARCHAR(100),
64              birth_date VARCHAR(100),
65              country VARCHAR(100),
66              city VARCHAR(255),
67              postal_code VARCHAR(255),
68              address VARCHAR(255)
69          );
```

Output

Action Output

#	Time	Action	Message
8	10:24:40	CREATE TABLE IF NOT EXISTS european_users (id VARCHAR(15) PRIMARY KEY, name VARCHAR(255),...	0 row(s) affected

Il·lustració 5. La consulta per crear la taula 'european_users'

- Creant la taula 'products' (Il·lustració 6)

```
72          # field names: id, product_name, price, colour, weight, warehouse_id
73          CREATE TABLE IF NOT EXISTS products (
74              id VARCHAR(15) PRIMARY KEY,
75              product_name VARCHAR(255),
76              price VARCHAR(15),
77              colour VARCHAR(15),
78              weight DECIMAL(2,1),
79              warehouse_id VARCHAR(255)
80          );
```

3	15:00:23	CREATE TABLE IF NOT EXISTS products (id VARCHAR(15) PRIMARY KEY, product_name VARCHAR(255),...	0 row(s) affected

Il·lustració 6. La consulta per crear la taula 'products'

- Creant la taula 'transactions' (Il·lustració 7)

```
82      # Creamos la tabla 'transactions'
83          # field names: id, card_id, business_id, timestamp, amount, declined, product_ids, user_id, lat, longitude
84          CREATE TABLE IF NOT EXISTS transactions (
85              id VARCHAR(255) PRIMARY KEY,
86              card_id VARCHAR(15),
87              business_id VARCHAR(15),
88              timestamp TIMESTAMP,
89              amount DECIMAL(10, 2),
90              declined BOOLEAN,
91              product_ids VARCHAR(255),
92              user_id VARCHAR(15),
93              lat FLOAT,
94              longitude FLOAT
95          );
```

```
11  10:25:21  CREATE TABLE IF NOT EXISTS transactions ( id VARCHAR(255) PRIMARY KEY, card_id VARCHAR(15), b...  0 row(s) affected
```

Il·lustració 7. La consulta per crear la taula 'transactions'

D'aquesta manera hem creat les taules presentades en l'exercici. Podem utilitzar els metadades per obtenir els detalls de cada taula i columna, com ara: TABLE_NAME, COLUMN_NAME, DATA_TYPE, COLUMN_TYPE, IS_NULLABLE, COLUMN_KEY, COLUMN_DEFAULT (Il·lustració 8).

```
98          SELECT
99            TABLE_NAME, COLUMN_NAME, DATA_TYPE,
100           COLUMN_TYPE, IS_NULLABLE, COLUMN_KEY,
101           COLUMN_DEFAULT
102         FROM
103           information_schema.columns
104         WHERE
105           table_schema = 'user_sales'
106         ORDER BY
107           TABLE_NAME, ORDINAL_POSITION;
108
```

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

TABLE_NAME	COLUMN_NAME	DATA_TYPE	COLUMN_TYPE	IS_NULLABLE	COLUMN_KEY	COLUMN_DEFAULT
american_users	id	varchar	varchar(15)	NO	PRI	NULL
american_users	name	varchar	varchar(255)	YES		NULL
american_users	surname	varchar	varchar(255)	YES		NULL
american_users	phone	varchar	varchar(15)	YES		NULL
american_users	email	varchar	varchar(100)	YES		NULL
american_users	birth_date	varchar	varchar(100)	YES		NULL
american_users	country	varchar	varchar(100)	YES		NULL
transactions	timestamp	timestamp	timestamp	YES		NULL
transactions	amount	decimal	decimal(10,2)	YES		NULL
transactions	declined	tinyint	tinyint(1)	YES		NULL
transactions	product_ids	varchar	varchar(255)	YES		NULL
transactions	user_id	varchar	varchar(15)	YES		NULL
transactions	lat	float	float	YES		NULL
transactions	longitude	float	float	YES		NULL

columns 2

Output

Action Output

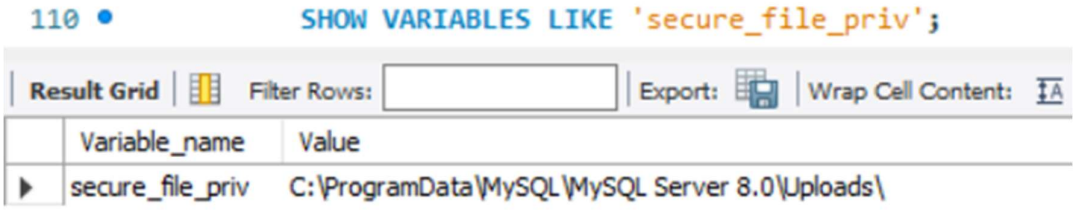
#	Time	Action	Message
14	10:45:04	SELECT TABLE_NAME, COLUMN_NAME, DATA_TYPE, COLUMN_TYPE, IS_NULLABLE, CO...	51 row(s) returned
15	10:53:45	SELECT TABLE_NAME, COLUMN_NAME, DATA_TYPE, COLUMN_TYPE, IS_NULLABLE, CO...	51 row(s) returned

Il·lustració 8. L'estructura de la base de dades 'user_sales' basada en una consulta amb metadades

Intentem carregar els fitxers CSV. Però rebem un error amb el codi 1290, que indica que MySQL Workbench té restriccions sobre les operacions d'importació/exportació d'alguna carpeta no autoritzada, tant pel costat del client com pel servidor.

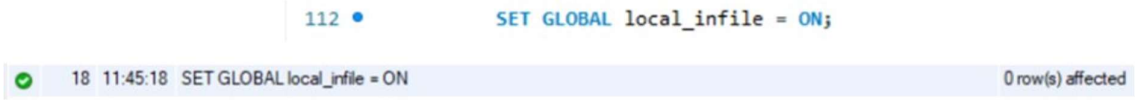
L'opció '--secure-file-priv' indica la carpeta on els fitxers es poden enganxar i després inserir a la base de dades en modalitat segura.

Per comprovar quina és la carpeta, s'utilitza SHOW VARIABLES (Il·lustració 9).



Il·lustració 9. Ubicació de la carpeta Upload segura per als fitxers que es carreguen a MyWorkbench

Aleshores tenim dues opcions: copiar els fitxers en aquesta carpeta o canviar l'opció per permetre l'accés als fitxers de qualsevol carpeta. Aplico l'última amb (Il·lustració 10)

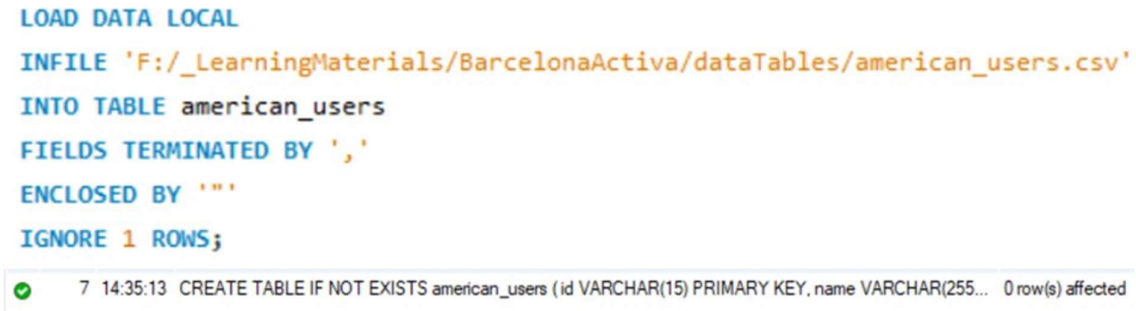


Il·lustració 10. Establint el valor global de local_infile a ON

Llavors vaig accedir a la configuració de connexió i vaig establir OPT_LOCAL_INFILE=1 [https://stackoverflow.com/questions/31450389/connect-with-local-infile-option-in-mysql-workbench].

Un cop fet, podem carregar els valors dels fitxers CSV. Com que hi ha dues opcions de delimitador, ',' o ';', és important comprovar quin delimitador s'utilitza en cada fitxer. Jo faig servir LibreOffice Calc o Notepad (o una eina similar). Anotem el delimitador de cada taula i l'assignem a FIELDS TERMINATED BY.

- carregem els valors a la taula 'american_users' (Il·lustració 11)



Il·lustració 11. Cargant dades a la taula 'american_users'

- carregem els valors a la taula ‘companies’ (Il·lustració 12)

```
123 •       LOAD DATA LOCAL
124         INFILE 'F:/_LearningMaterials/BarcelonaActiva/dataTables/companies.csv
125         INTO TABLE companies
126         FIELDS TERMINATED BY ','
127         ENCLOSED BY '"'
128         IGNORE 1 ROWS;
```

```
24 14:53:52 LOAD DATA LOCAL INFILE 'F:/_LearningMaterials/BarcelonaActiva/dataTables/companies.csv' INTO TA... 100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
```

Il·lustració 12. Cargant dades a la taula 'companies'

- carregem els valors a la taula ‘credit_cards’ (Il·lustració 13)

```
130 •       LOAD DATA LOCAL
131         INFILE 'F:/_LearningMaterials/BarcelonaActiva/dataTables/credit_cards.csv'
132         INTO TABLE credit_cards
133         FIELDS TERMINATED BY ','
134         ENCLOSED BY '"'
135         IGNORE 1 ROWS;
```

```
26 14:56:59 LOAD DATA LOCAL INFILE 'F:/_LearningMaterials/BarcelonaActiva/dataTables/credit_cards.csv' INTO T... 5000 row(s) affected Records: 5000 Deleted: 0 Skipped: 0 Warnings: 0
```

Il·lustració 13. Cargant dades a la taula 'credit_cards'

- carregem els valors a la taula ‘european_users’ (Il·lustració 14)

```
137 •   LOAD DATA LOCAL
138     INFILE 'F:/_LearningMaterials/BarcelonaActiva/dataTables/european_users.csv'
139     INTO TABLE european_users
140     FIELDS TERMINATED BY ','
141     ENCLOSED BY '"'
142     IGNORE 1 ROWS;
143
Output
Action Output
#  Time      Action                                                                                                                                  Message
1  14:58:17  LOAD DATA LOCAL INFILE 'F:/_LearningMaterials/BarcelonaActiva/dataTables/european_users.csv' INTO T...  3990 row(s) affected Records: 3990  Deleted: 0  Skipped: 0  Warnings: 0
```

Il·lustració 14. Cargant dades a la taula 'european_users'

- carregem els valors a la taula ‘products’ (Il·lustració 15)

```
144 •       LOAD DATA LOCAL
145         INFILE 'F:/_LearningMaterials/BarcelonaActiva/dataTables/products.csv'
146         INTO TABLE products
147         FIELDS TERMINATED BY ','
148         ENCLOSED BY '"'
149         IGNORE 1 ROWS;
```

```
4 15:01:18 LOAD DATA LOCAL INFILE 'F:/_LearningMaterials/BarcelonaActiva/dataTables/products.csv' INTO TABLE p... 100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
```

Il·lustració 15. Cargant dades a la taula 'products'

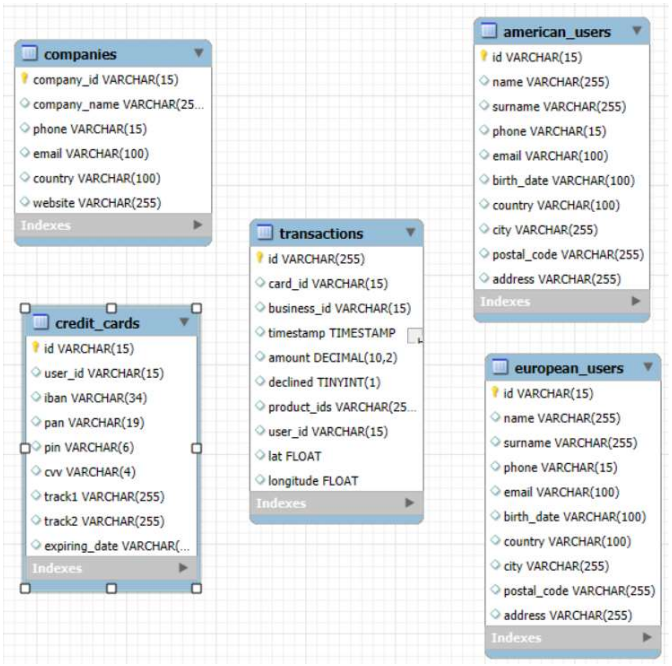
- carregem els valors a la taula ‘transactions’ (Il·lustració 16)

```
151 •       LOAD DATA LOCAL
152         INFILE 'F:/_LearningMaterials/BarcelonaActiva/dataTables/transactions.csv'
153         INTO TABLE transactions
154         FIELDS TERMINATED BY ';'
155         ENCLOSED BY '"'
156         IGNORE 1 ROWS;
157
```

```
Output
Action Output
#  Time      Action                                                                                  Message
1  15:02:09  LOAD DATA LOCAL INFILE 'F:/_LearningMaterials/BarcelonaActiva/dataTables/transactions.csv' INTO TABL...  100000 row(s) affected Records: 100000  Deleted: 0  Skipped: 0  Warnings: 0
```

Il·lustració 16. Cargant dades a la taula 'transactions'

Al final, vam rebre la base de dades amb l'esquema sense relacions entre les taules (Il·lustració 17).



Il·lustració 17. Estructura actual de la base de dades

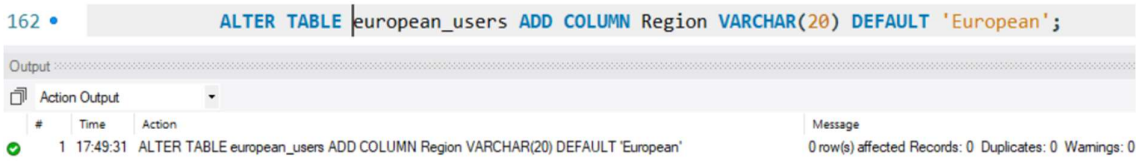
Abans d'enllaçar les taules, estudiem les estructures:

- La taula 'transaccions' és similar a una taula de fets, i té camps que es poden utilitzar com a claus foranes:
- *card_id* permet la relació amb la taula 'credit_cards';
- *business_id* permet la relació amb la taula 'companies',
- *user_id* permet la relació amb les taules 'american_users' i 'european_users',
- *product_ids* permite relacio con una tabla de los productos.

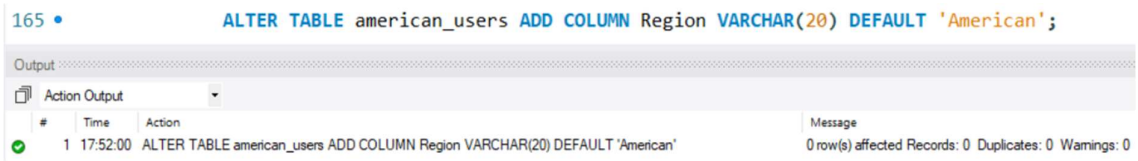
- La taula 'credit_card' inclou el camp 'user_id', que permet enllaçar-la amb les taules ‘american_users’ i ‘european_users’

Cal tenir en compte que si establim relacions entre 'transaction' i 'credit_cards', 'transaction' i taules d'usuaris, i entre 'credit_cards' i usuaris, correm el risc de crear un bucle de relacions. Es recomana evitar-ho, de manera que, en dissenyar el model de dades, no considerem crear una relació entre 'credit_cards' i les taules d'usuaris.

A més, per millorar les consultes amb aquesta base de dades i mantenir la coherència de les dades, creem una taula 'users', que inclou 'american_users' i 'european_users'. Els camps de totes dues són els mateixos. No obstant això, per mantenir l'aspecte regional dels usuaris, afegim un nou camp 'Region' amb els valors 'European' (Il·lustració 18) i 'American' (Il·lustració 19) corresponents a aquestes taules.



Il·lustració 18. Afegir la columna 'Region' a la taula 'european_users' amb el valor "European" per a tots.

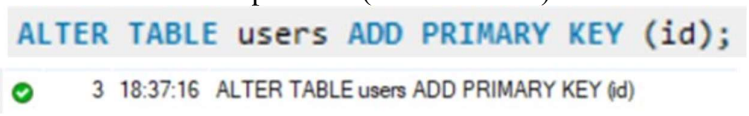


Il·lustració 19. Afegir la columna 'Region' a la taula 'american_users' amb el valor "American" per a tots.

Ara podem crear la taula 'users' mitjançant UNION (Il·lustració 20)

Il·lustració 20. Ara podem crear la taula 'users' mitjançant UNION.

Hem de configurar l' 'id' com a clau primària (Il·lustració 21).



Il·lustració 21. Estableix el 'id' com a clau primària.

Fem la consulta per comprovar el resultat, ordenant la taula per data de naixement. Això permet identificar les persones de les regions europea i americana (Il·lustració 22).

```
173 •  SELECT * FROM users ORDER BY birth_date;
```

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

id	name	surname	phone	email	birth_date	country	city	postal_code	address	Region
1592	Eurifr	Lwqpibnq	+67-455-2564	eurifr.lwqpibnq@example.com	Apr 1, 1950	Spain	Madrid	28001	643 Lwqpibnq St	European
3443	Zeerpq	Mqubzlhs	+48-606-7793	zeerpq.mqubzlhs@example.com	Apr 1, 1953	United States	Chicago	60601	485 Mqubzlhs St	American
3719	Otesug	Ompbazjy	+69-910-9177	otesug.ompbazjy@example.com	Apr 1, 1957	Canada	Calgary	T1Y 0A1	289 Ompbazjy St	American
2912	Tjdaad	Gtaedqfz	+49-260-5338	tjdaad.gtaedqfz@example.com	Apr 1, 1961	France	Paris	75001	996 Gtaedqfz St	European
1391	Hweorx	Qqljbyae	+78-271-9761	hweorx.qqljbyae@example.com	Apr 1, 1970	United Kingdom	London	EC1A 1BB	390 Qqljbyae St	European

users 5 ×

Output

Action Output

#	Time	Action	Message
1	17:56:22	SELECT * FROM users ORDER BY birth_date	5000 row(s) returned

Il·lustració 22. El contingut de la taula 'users'

Assignem claus estranyes per a la taula 'transaccions' (Il·lustració 23).

```
181 •  ALTER TABLE transactions
182        ADD CONSTRAINT fk_transactions_users
183        FOREIGN KEY (user_id) REFERENCES users(id);
184 •  ALTER TABLE transactions
```

Output

Action Output

#	Time	Action	Message
1	18:38:51	ALTER TABLE transactions ADD CONSTRAINT fk_transactions_users FOREIGN KEY (user_id) REFERENCE...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0

```
184 •  ALTER TABLE transactions
185        ADD CONSTRAINT fk_transactions_companies
186        FOREIGN KEY (business_id) REFERENCES companies(company_id);
187 •  ALTER TABLE transactions
```

Output

Action Output

#	Time	Action	Message
1	18:39:44	ALTER TABLE transactions ADD CONSTRAINT fk_transactions_companies FOREIGN KEY (business_id) REF...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0

```
187 •  ALTER TABLE transactions
188        ADD CONSTRAINT fk_transactions_credit_cards
189        FOREIGN KEY (card_id) REFERENCES credit_cards(id);
190
```

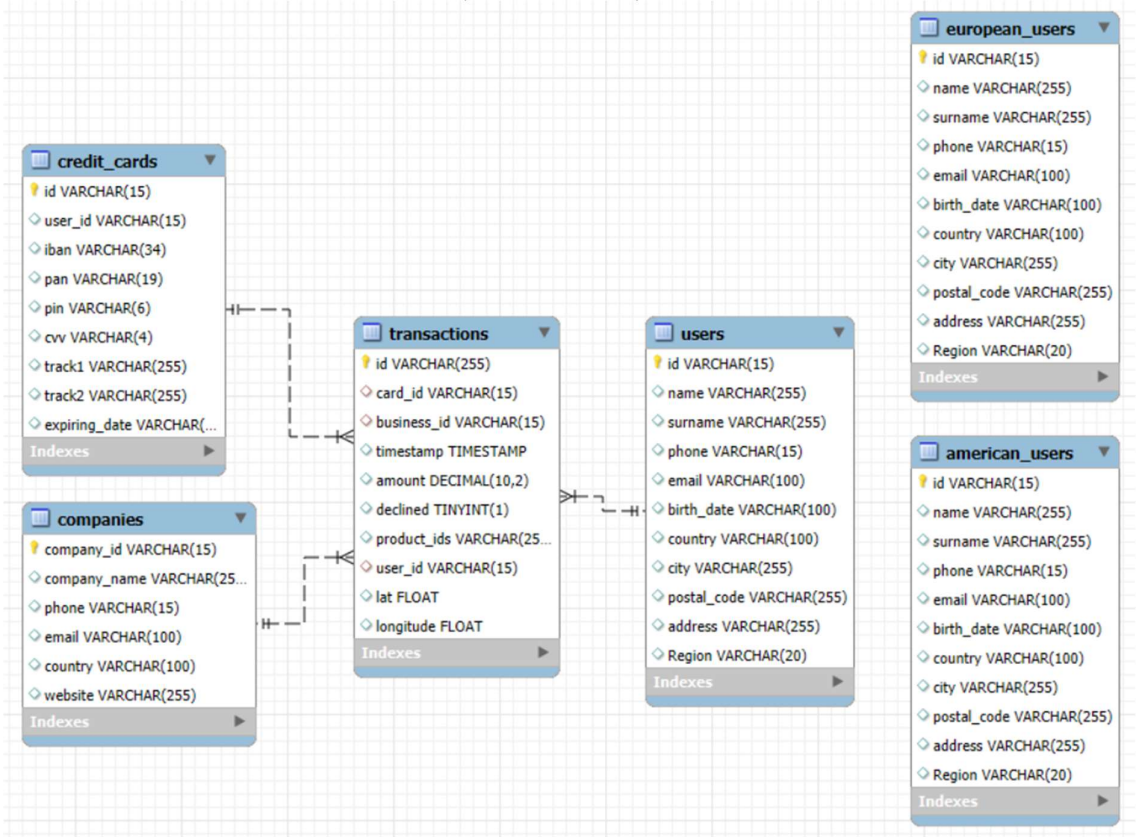
Output

Action Output

#	Time	Action	Message
1	18:40:19	ALTER TABLE transactions ADD CONSTRAINT fk_transactions_credit_cards FOREIGN KEY (card_id) REFE...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0

Il·lustració 23. Assignació de claus foranes per a la taula 'transaccions'

En aquesta fase, rebem el model d'esquema d'estrella, amb 'transaccions' com a taula de fets i les altres com a taules de dimensions (Il·lustració 24).



Il·lustració 24. Model de dades d'estel·la de la base de dades 'user_sales'

Les taules 'european_users' i 'american_users' no tenen cap connexió amb 'users', perquè aquesta última conté tots els valors i garanteix les consultes necessàries. Si ens imaginem que les taules d'usuaris provenen d'altres recursos (i n'hi pot haver més), és una bona idea crear una funció amb un trigger per recrear la taula 'users'.

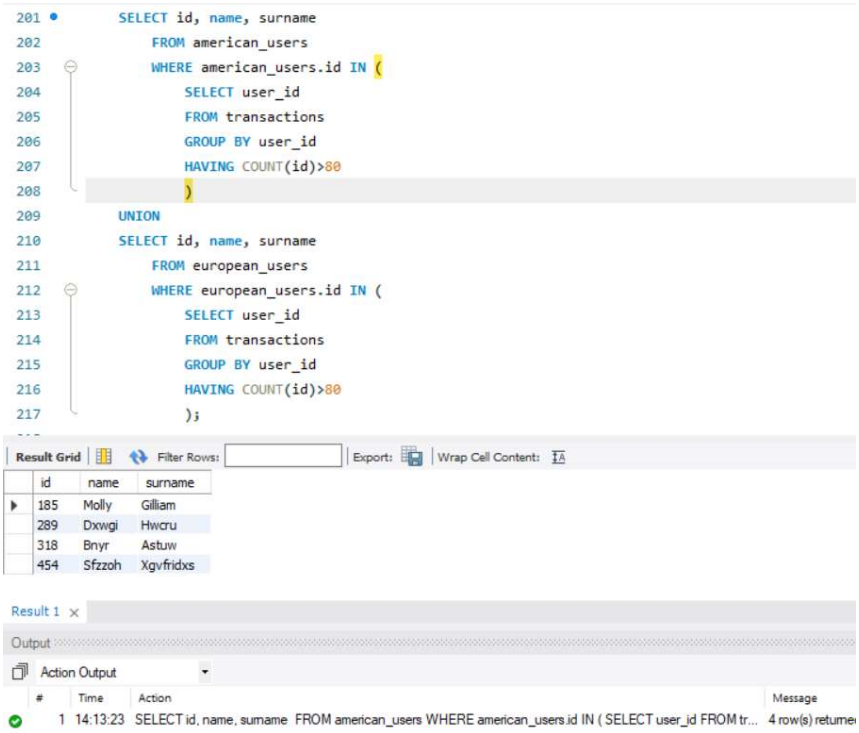
Exercici 1

Tasca:

Realitza una subconsulta que mostri tots els usuaris amb més de 80 transaccions utilitzant almenys 2 taules.

Comentaris

Es poden implementar dues opcions de consulta: amb les taules 'american_users' i 'european_users' o directament amb la taula 'user'. En la primera opció, cerquem quins usuaris de cada taula estan registrats a la taula 'transactions' comparant els valors de 'id'. Combinem el resultat de la subconsultas amb UNION (Il·lustració 25).



```
SELECT id, name, surname
    FROM american_users
    WHERE american_users.id IN (
        SELECT user_id
        FROM transactions
        GROUP BY user_id
        HAVING COUNT(id)>80
        )
UNION
SELECT id, name, surname
    FROM european_users
    WHERE european_users.id IN (
        SELECT user_id
        FROM transactions
        GROUP BY user_id
        HAVING COUNT(id)>80
        );
```

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

id	name	surname
185	Molly	Gilliam
289	Dxwgi	Hwcru
318	Bnyr	Astuw
454	Sfzzoh	Xgvfridxs

Result 1

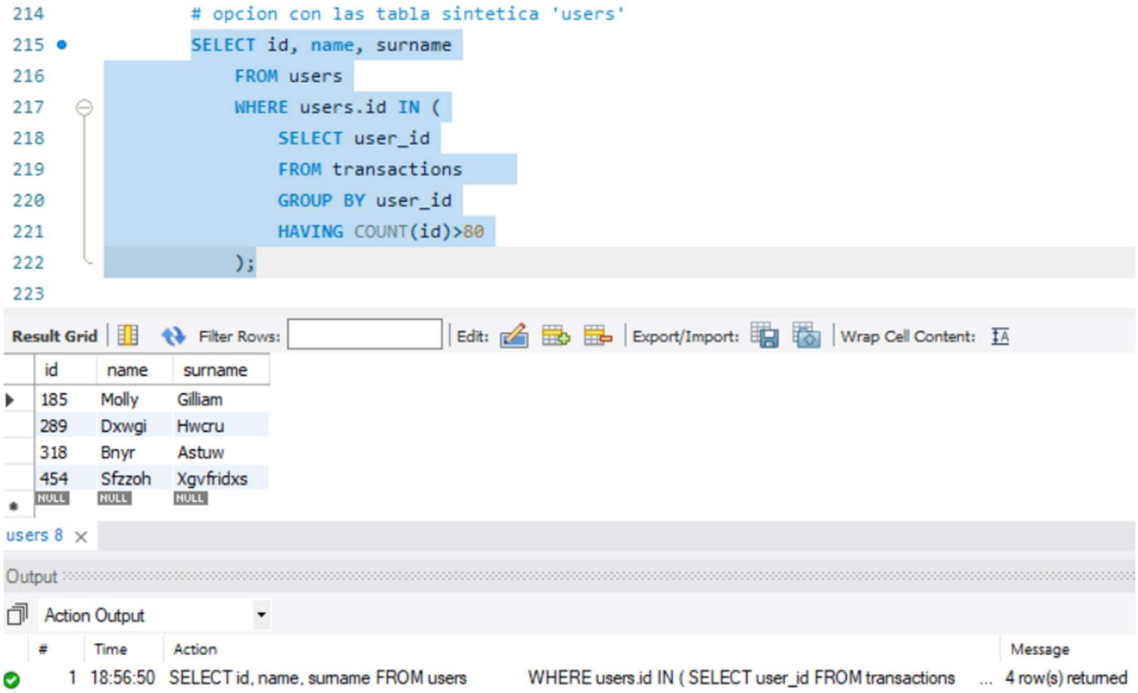
Output

Action Output

#	Time	Action	Message
1	14:13:23	SELECT id, name, surname FROM american_users WHERE american_users.id IN (SELECT user_id FROM tr...	4 row(s) returne

Il·lustració 25. Els usuaris amb més de 80 transaccions utilitzant 3 taules

L'altra opció es basa en seleccionar usuaris de la taula ‘users’ (Il·lustració 26).



Il·lustració 26. Els usuaris amb més de 80 transaccions utilitzant 2 taules

El resultat són quatre usuaris que han realitzat més de 80 transaccions. La diferència més important és la complexitat de la consulta, que és més senzilla en el segon cas.

Exercici 2

Tasca

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

Comentaris

Estimar una mitjana pressuposa que hi ha més valors per a cada entitat (l'IBAN). Però com que això no és explícit, proposo dues etapes:

- unim les taules 'transactions', 'credit_cards' i 'companies', apliquem el filtre per extreure els valors de l'empresa 'Donec Ltd', només amb transaccions vàlides (declined = 0) i calculem la mitjana de cada IBAN. (Il·lustració 27)

```
222 •   SELECT credit_cards.iban, ROUND(AVG(transactions.amount), 2) AS Mitjana
223         -- COUNT(transactions.id) AS NrTrans
224     FROM transactions
225     JOIN credit_cards
226         ON transactions.card_id = credit_cards.id
227     JOIN companies
228         ON transactions.business_id = companies.company_id
229     WHERE companies.company_name = 'Donec Ltd'
230         AND transactions.declined = 0
231     GROUP BY credit_cards.iban
232     -- HAVING NrTrans > 1
```

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

iban	Mitjana
XX9114064011255863075868O5	356.25
SK9446370242474562577506	142.96
XX7767529178459529755556 40	257.37
XX4138273622897193049089 90	139.59
XX3477872460707696107803 08	240.41

Result 6

Output

Action Output

#	Time	Action	Message
1	14:36:51	SELECT credit_cards.iban, ROUND(AVG(transactions.amount), 2) AS Mitjana -- COUNT(transactions.id) AS N...	370 row(s) returned

Il·lustració 27. Càlcul de l'import vàlid mitjà per a cada IBAN per l'empresa 'Donec Ltd'

Hem rebut 370 IBAN associats a transaccions amb aquesta empresa.

- Però un d'ells només tenia una transacció. Per tant, podem plantejar-nos cercar IBAN que tinguin més d'una transacció, aplicant un filtre amb HAVING (Il·lustració 28).

```
222 •   SELECT credit_cards.iban, ROUND(AVG(transactions.amount), 2) AS Mitjana,
223         COUNT(transactions.id) AS NrTrans
224     FROM transactions
225     JOIN credit_cards
226         ON transactions.card_id = credit_cards.id
227     JOIN companies
228         ON transactions.business_id = companies.company_id
229     WHERE companies.company_name = 'Donec Ltd'
230         AND transactions.declined = 0
231     GROUP BY credit_cards.iban
232     HAVING NrTrans > 1
233     ;
```

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

iban	Mitjana	NrTrans
XX9114064011255863075868O5	356.25	3
PL76249283566852676343404576	541.56	3
LB64655537777363327873049938	155.50	2
NO4414757761220	95.17	2
CH8995351081824762557	199.36	3

Result 7

Output

Action Output

#	Time	Action	Message
1	14:36:51	SELECT credit_cards.iban, ROUND(AVG(transactions.amount), 2) AS Mitjana -- COUNT(transactions.id) AS N...	370 row(s) returned
2	14:40:25	SELECT credit_cards.iban, ROUND(AVG(transactions.amount), 2) AS Mitjana, COUNT(transactions.id) AS Nr...	66 row(s) returned

Il·lustració 28. Càlcul de l'import vàlid mitjà per a cada IBAN per l'empresa 'Donec Ltd' amb més de 2 transaccions

En aquest cas, només podem validar 66 IBANs amb un nombre de transaccions ≥ 2.

Nivell 2

Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les tres últimes transaccions han estat declinades aleshores és inactiu, si almenys una no és rebutjada aleshores és actiu. Partint d'aquesta taula respon:

Comentaris

Creiem la taula 'card_status' (Il·lustració 29). Hem de fer més subconsultes anidades. Primer, seleccionem les dates consecutives ('timestamps') i comptem quantes n'hi ha per a cada targeta. Ho faig amb una subconsulta correlacionada de la taula 'transactions' amb si mateixa. A continuació, selecciono només les transaccions amb les tres últimes dates. Ara, he de comprovar si la transacció ha estat rebutjada ('declined') i quantes vegades. Finalment, assigno l'estat "inactiu" a cada targeta (pel seu 'card_id') que hagi tingut tres rebutjos, i "actiu" en qualsevol altre cas.

```
246 •   CREATE TABLE card_status
247         SELECT fechas.card_id AS card_id,
248             CASE WHEN SUM(fechas.declined) = 3 THEN 'inactiva'
249                 ELSE 'activa'
250                 END AS 'Status'
251             FROM (
252                 SELECT t1.card_id, t1.timestamp, t1.declined
253                     FROM transactions t1
254                     WHERE (
255                       SELECT COUNT(*)
256                       FROM transactions t2
257                       WHERE t2.card_id = t1.card_id
258                         AND t2.timestamp > t1.timestamp
259                     ) < 3
260                 ) AS fechas
261             GROUP BY fechas.card_id;
```

```
5  14:44:39  CREATE TABLE card_status  SELECT fechas.card_id AS card_id, CASE WHEN SUM(fechas.declined) = 3...  5000 row(s) affected
```

Il·lustració 29. Creació de la taula 'card_status' en funció del seu ús

Per comprovar el contingut de 'card_status', fem una consulta (Il·lustració 30).

```
262 •   SELECT * FROM card_status;
```

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

card_id	Status
CcS-4857	activa
CcS-4858	activa
CcS-4859	activa
CcS-4860	activa
CcS-4861	activa

card_status 23 ×

Output

Action Output

#	Time	Action	Message
1	14:51:21	SELECT * FROM card_status	5000 row(s) returned

Il·lustració 30. El contingut de la taula 'card_status'

Exercici 1

Tasca

Quantes targetes estan actives?

Comentaris

Executem la consulta per comptar el nombre de valors 'actius' al camp ‘status’ (Il·lustració 31).

```
266 •   SELECT COUNT(card_id)
267         FROM card_status
268         WHERE status = 'activa';
```

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

COUNT(card_id)
4995

Result 24

Output

Action Output

#	Time	Action	Message
1	14:51:21	SELECT * FROM card_status	5000 row(s) returned
2	14:52:09	SELECT COUNT(card_id) FROM card_status WHERE status = 'activa'	1 row(s) returned

Il·lustració 31. Subconsulta per comptar el nombre de targetes actives

Tanmateix, per a una visualització més senzilla, podeu mostrar el nombre de targetes amb estat actiu i inactiu com a noms de columnas (Il·lustració 32).

```
271 •   SELECT
272         SUM(CASE WHEN status = 'activa' THEN 1 ELSE 0 END) 'Activa',
273         SUM(CASE WHEN status = 'inactiva' THEN 1 ELSE 0 END) 'Inactiva'
274     FROM card_status;
```

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

Activa	Inactiva
4995	5

Result 29

Output

Action Output

#	Time	Action	Message
1	15:07:41	SELECT SUM(CASE WHEN status = 'activa' THEN 1 ELSE 0 END) 'Activa', SUM(CASE WHEN status ...	1 row(s) returned

Il·lustració 32. Tanmateix, per a una visualització més senzilla, el nombre de targetes amb estat actiu i inactiu es pot mostrar com a noms de columna.

La base de dades inclou 4995 targetes actives.

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:

Comentaris

Actualment, no és possible comptar els productes venuts en cada transacció, perquè cada transacció inclou una llista de productes descrits al camp 'product_ids'. Això significa que hem de desagregar les transaccions per cada producte venut. A més, hem de mantenir la relació entre transaccions i productes com a 'molts-a-molts'. Per tant, hem de crear una nova taula anomenada 'transactions_products' que ens permeti fer consultes per a cada producte individual.

Segons diferents recursos, podem considerar els 'product_ids' com una llista de valors que es pot extreure com un registre en format JSON. Per a aquest propòsit, hem d'ajustar cada registre eliminant els espais en blanc i separant-lo per ','. Al mateix temps, assignem a cada id de producte obtingut l'id de transacció, la qual cosa crea una combinació única (Il·lustració 33).

```
286 •   CREATE TABLE transactions_products
287         SELECT
288           t.id AS transaction_id,
289           jsontab.product_id AS product_id
290         FROM transactions AS t,
291           JSON_TABLE(
292             CONCAT('[', REPLACE(REPLACE(t.product_ids, ' ', ''), ',', ','), ']'),
293             "$[*]" COLUMNS (
294               product_id INT PATH "$"
295             )
296           ) AS jsontab;
```

```
Output
Action Output
#  Time      Action                                                                                  Message
1  19:19:36  CREATE TABLE transactions_products SELECT  t.id AS transaction_id,  /* t.card_id, t.business_id, t.timestam...  253391 row(s) affected Records: 253391  Duplicates: 0  Warnings: 0
```

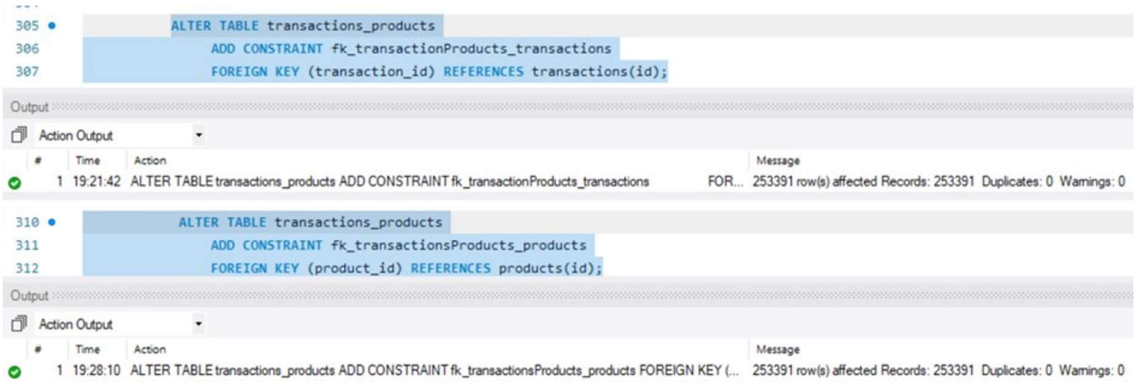
Il·lustració 33. La consulta per crear la taula 'transactions_products'

Així, la taula només contindrà parells de 'transaction_id' i 'product_id'. Hem de modificar el tipus de les columnes 'transaction_id' i 'product_id' (a VARCHAR(255)), perquè es va canviar durant la creació de la taula. Això ens permet utilitzar-los com a clau primària combinada (Il·lustració 34), així com clau estranya en relació amb les taules 'transactions' i 'products', respectivament (Il·lustració 35).

```
303 •   ALTER TABLE transactions_products ADD PRIMARY KEY (transaction_id, product_id);
304 •   ALTER TABLE transactions_products MODIFY COLUMN transaction_id VARCHAR(255);
```

```
Output
Action Output
#  Time      Action                                                                  Message
2  19:27:26  ALTER TABLE transactions_products MODIFY COLUMN product_id VARCHAR(255)  253391 row(s) affected Records: 253391  Duplicates: 0  Warnings: 0
3  19:27:33  ALTER TABLE transactions_products MODIFY COLUMN product_id VARCHAR(255)  0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

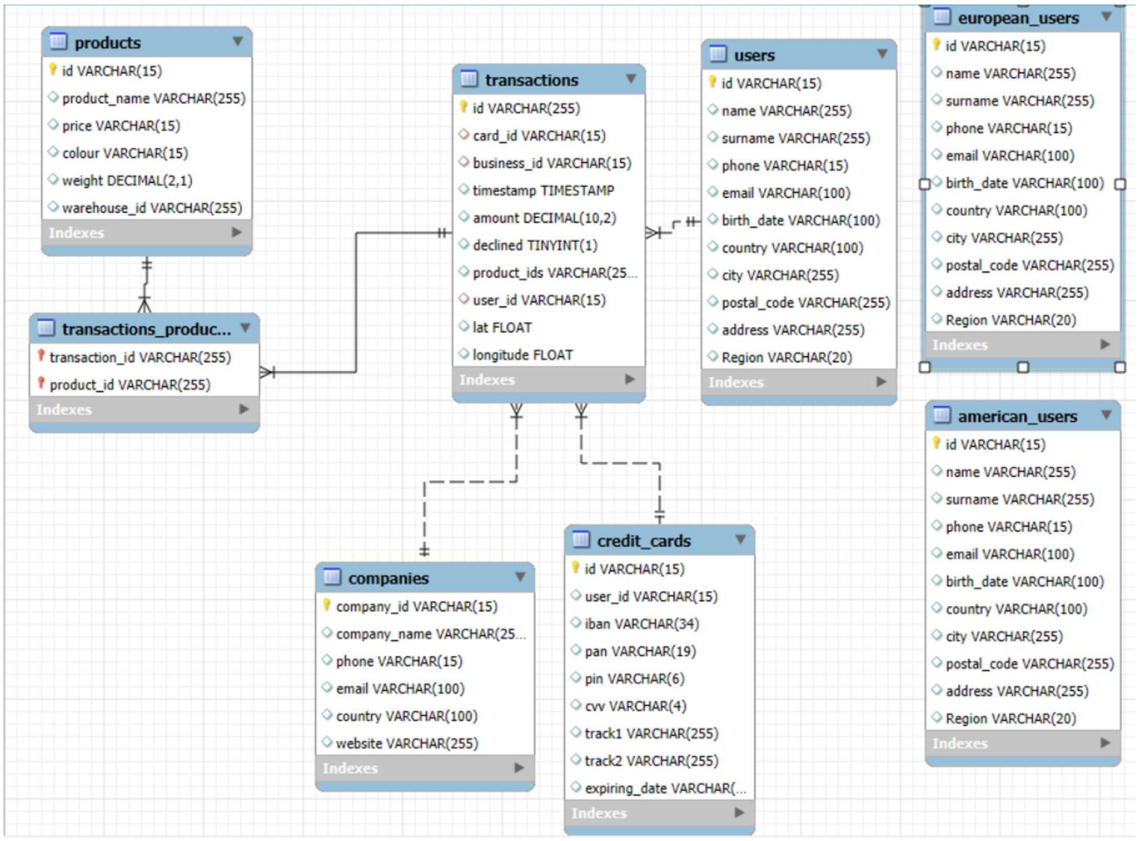
Il·lustració 34. Establir la clau primària combinada per a la taula 'transactions_products'



Il·lustració 35. Establint les claus foranes de la taula 'transactions_products' amb les taules 'transactions' i 'products'.

Com a resultat obtenim una base de dades de tipus Snowflake (Il·lustració 36), amb dues taules de fets: 'transactions_products' en relació amb 'transactions' i 'products' (N:N), i la taula 'transactions' en relació amb 'credit_cards', 'companies', 'users' (N:1).

'card_status' és una taula derivada, que s'adapta més com una taula de informes que s'hauria de crear mitjançant una vista i un disparador, basada en el contingut actual de 'credit_cards'.



Il·lustració 36. Forma de Snowflake del model de base de dades 'user_sales'

Exercici 1

Tasca

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

Comentaris

Per obtenir el nom d'un producte, el seu ID i el nombre de vegades que s'ha venut, cal unir tres taules: 'transactions_products', 'transactions' i 'products'. A partir d'aquestes, només considerem les vendes realitzades (declined=0). Després comptem quantes vegades cada producte s'ha associat amb una transacció ('nrSales'). Per ordenar els resultats per product_id, que actualment és un caràcter, vaig haver d'aplicar CAST i AS USINGNED per convertir-lo en un nombre enter (Il·lustració 37).

```
316 •   SELECT CAST(t_p.product_id AS UNSIGNED) AS product_id, products.product_name AS product,
317         COUNT(t_p.product_id) AS nrSales
318     FROM transactions_products AS t_p
319     JOIN transactions ON t_p.transaction_id = transactions.id
320     JOIN products ON t_p.product_id = products.id
321         WHERE transactions.declined = 0
322         GROUP BY product_id
323         ORDER BY product_id;
324
```

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

product_id	product	nrSales
1	Direwolf Stannis	2467
2	Tarly Stark	2562
3	duel tourney Lannister	2520
4	warden south duel	2573
5	skywalker ewok	2543
6	dooku solo	2487

Result 26 ×

Output

Action Output

#	Time	Action	Message
1	18:28:47	SELECT CAST(t_p.product_id AS UNSIGNED) AS product_id, products.product_name AS product, COUNT(t...	100 row(s) returned

Il·lustració 37. Càlcul de nombre de vegades que s'ha venut cada producte

Així, podem veure que hi ha 100 productes, que es van comprar entre 2405 i 2642 vegades.