

SPRINT 4: Modelat SQL

Índex

Nivell 1.....	2
- Exercici 1.....	10
- Exercici 2.....	12
Nivell 2.....	14
- Exercici 1.....	16

Nivell 1

Descarrega 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:

Primer de tot he examinat els arxius .csv que tenia per determinar quines podien ser les relacions entre ells. Els arxius eren els següents:

- companies.csv
- credit_cards.csv
- products.csv
- transactions.csv
- users_ca.csv
- users_uk.csv
- users_usa.csv

Per fer els primers exercicis, he vist que no necessitava les dades de «products.csv», així que de moment no les he carregat (cal crear aquesta taula al nivell 3). Segons les dades de què es tracta, m'he adonat que el model serà en forma d'estrella, i que la taula de fets serà la de «transactions». Les taules que s'hi relacionaran a partir de les seves primary key seran «companies», «credit_cards» i «users». Aquesta última taula recollirà la informació dels tres arxius «users_ca.csv», «users_uk.csv» i «users_usa.csv».

Primer del tot, he creat la base de dades, «Sprint_4» on volia carregar les taules amb l'ordre següent:

```
# Creo la nova base de dades
CREATE DATABASE IF NOT EXISTS Sprint_4;
USE Sprint_4;
```

Tot seguit, he començat per crear la taula de «credit_cards»:

```
6 # Creo la taula "credit_cards"
7 • CREATE TABLE IF NOT EXISTS credit_cards
8   (card_id VARCHAR(20) PRIMARY KEY,
9    user_id VARCHAR(20) NOT NULL,
10   iban VARCHAR(50) NOT NULL,
11   pan VARCHAR(50) NOT NULL,
12   pin VARCHAR(4) NOT NULL,
13   cvv INT NOT NULL,
14   track1 VARCHAR(255),
15   track2 VARCHAR(255),
16   expiring_date VARCHAR(20) NOT NULL);
17
```

Output

#	Time	Action	Message
✓ 1	18:24:51	CREATE TABLE IF NOT EXISTS credit_cards (card_id VARCHAR(20) PRIMARY KEY, user_id VARCHAR...	0 row(s) affected

Tot seguit, hi he carregat les dades de «credit_cards.csv». Inicialment, volia carregar-les fent servir aquesta ordre:

```
28 • LOAD DATA INFILE 'C:\Users\paula\Documents\Escriptori\IT ACADEMY\Sprint 4'
29 INTO TABLE credit_cards
30 FIELDS TERMINATED BY ','
31 OPTIONALLY ENCLOSED BY '"'
32 LINES TERMINATED BY '\n'
33 IGNORE 1 ROWS;
```

Output

#	Time	Action	Message
✗ 1	18:34:04	LOAD DATA INFILE 'C:\Users\paula\Documents\Escriptori\IT ACADEMY\Sprint 4' INTO TABLE credit_cards...	

Això, però, mostrava el següent missatge d'error:

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

Buscant informació sobre aquest error, he arribat a la conclusió que la carpeta on s'havien de trobar els arxius que volia carregar havien d'estar a la carpeta «Uploads» de MySQL Server 8.0, a la carpeta de MySQL de ProgramData. Això és perquè la configuració de seguretat «secure_file_priv» de MySQL no permet carregar arxius que estiguin a altres carpetes. Podria haver desactivat aquesta funció, però he optat per pujar tots els arxius .csv a la carpeta «Uploads» de MySQL.

Per comprovar que havia de pujar els arxius a aquesta carpeta, he fet servir l'ordre següent:

```
26 #Comprovo la carpeta on he de carregar els arxius perquè funcionin:
27 • SHOW VARIABLES LIKE 'secure_file_priv';
28
29
30
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Variable_name	Value			
secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\			

Així doncs, un cop l'ordre ha quedat així:

```
# Carrego les dades de "credit_cards" a la taula
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv'
INTO TABLE credit_cards
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

Aquesta ordre indica l'origen de les dades (la carpeta de ProgramData), la taula en què s'han de carregar (credit_cards), també indica que cada camp està separat del següent per una coma (com és típic en els documents .csv) o opcionalment separat per cometes, que cada fila està separada de la següent per un salt de línia (\n) i que cal ignorar la primera línia (que és on hi ha els títols de cada columna).

Tot seguit, he comprovat que les dades s'haguessin carregat correctament:

```
31 # Comprovo que s'han carregat bé les dades
32
33 • SELECT *
34 FROM credit_cards;
35
```

Result Grid									Filter Rows:	Edit:	Export/Imports:	Wrap Cell Contents:
card_id	user_id	iban	pan	pin	cvv	track1	track2	expiring_date				
CcU-2938	275	TR301950312213576817638661	5424465566813633	3257	984	%B8383712448554646^WovsxjeDpwiev^8604...	%B7653863056044187=800716333673	10/30/22				
CcU-2945	274	DO26854763748537475216568689	5142423821948828	9080	887	%B4621311609958661^UftuyfsSeimxn^06106...	%B4149568437843501=510714033071	08/24/23				
CcU-2952	273	BG451VQL52710525608255	4556 453 55 5287	4598	438	%B2183285104307501^CddyttUxwfdq^5907...	%B6778580257827162=6906859740077	06/29/21				
CcU-2959	272	CR724247244335841535	372461377349375	3583	667	%B7281111956795320^XocddjBckecd^09016...	%B4246154489281853=280522391678	02/24/23				
CcU-2966	271	BG72LKTQ15627628377363	448566 886747 7265	4900	130	%B4728932322756223^JhlgvsuFbmwgj^7202...	%B2318571115599881=890821578475	10/29/24				
CcU-2973	270	PT87806228135092429456346	544 58654 54343 384	8760	887	%B4761405253275637^HjnpipoBlejrl^7108515...	%B7816169831446746=1310277229	01/30/25				

He seguit el mateix procés per crear la taula «companies», és a dir, primer he creat la taula i després hi he carregat les dades de l'arxiu .csv a partir de l'ordre «LOAD DATA INFILE»:

```
37 # Creo la taula "companies"
38 • CREATE TABLE IF NOT EXISTS companies (
39     company_id VARCHAR(15) PRIMARY KEY,
40     company_name VARCHAR(255),
41     phone VARCHAR(15),
42     email VARCHAR(100),
43     country VARCHAR(100),
44     website VARCHAR(255)
45 );
46
47 # Carrego les dades a la taula "companies"
48 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv'
49 INTO TABLE companies
50 FIELDS TERMINATED BY ','
51 OPTIONALLY ENCLOSED BY '"'
52 LINES TERMINATED BY '\n'
53 IGNORE 1 ROWS ;
```

I tot seguit he comprovat que s'haguessin carregat bé les dades:

```
57 • SELECT *
58 FROM companies;
59
```

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

A continuació he creat la taula «users», on hi carregaré les dades dels arxius .csv següents: «users_ca.csv», «users_uk.csv» i «users_usa.csv».

```
61 # Creo la taula "users"
62 • CREATE TABLE IF NOT EXISTS users (
63     user_id VARCHAR(20) PRIMARY KEY,
64     name VARCHAR(150),
65     surname VARCHAR(150),
66     phone VARCHAR(150),
67     email VARCHAR(150),
68     birth_date VARCHAR(100),
69     country VARCHAR(150),
70     city VARCHAR(150),
71     postal_code VARCHAR(100),
72     address VARCHAR(255)
73 );
```

En el moment de carregar les dades, però, m'apareixia un error:

```
75 # A la nova taula "users", hi passo la info de les taules "users_uk", "users_usa" i "users_ca".
76 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv'
77 INTO TABLE users
78 FIELDS TERMINATED BY ','
79 OPTIONALLY ENCLOSED BY '"'
80 LINES TERMINATED BY '\n'
81 IGNORE 1 ROWS;
82
```

Output

#	Time	Action	Message
1	11:21:11	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv' INTO TABLE users FI...	Error Code: 1262. Row 4 was truncated; it contained more data than there were input columns

Ampliat, l'error era el següent:

Error Code: 1262. Row 4 was truncated; it contained more data than there were input columns

La fila 4, en què, s'havia detectat aquest error, contenia l'adreça següent, separada per una coma: «517-6759 Ut, Av.», crec que l'error venia d'aquí, perquè s'interpretava que aquesta coma indicava el final d'un camp. Finalment, he vist que calia indicar els salts de línia amb la fórmula LINES TERMINATED BY '\r\n', i que aleshores es detectaven correctament tots els salts de línia:

```
75 # A la nova taula "users", hi passo la info de les taules "users_uk", "users_usa" i "users_ca".
76 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv'
77 INTO TABLE users
78 FIELDS TERMINATED BY ','
79 OPTIONALLY ENCLOSED BY '"'
80 LINES TERMINATED BY '\r\n'
81 IGNORE 1 ROWS;
82
83
```

Output

#	Time	Action	Message
1	12:00:06	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv' INTO TABLE users FI...	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0

He seguit el mateix procés per afegir les taules de «users_uk» i «users_usa», i d'aquesta manera les he pogut afegir sense problemes:

```
84 # "users_uk"
85 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv'
86 INTO TABLE users
87 FIELDS TERMINATED BY ','
88 OPTIONALLY ENCLOSED BY '"'
89 LINES TERMINATED BY '\r\n'
90 IGNORE 1 ROWS;
91
92 # "users_usa"
93 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv'
94 INTO TABLE users
95 FIELDS TERMINATED BY ','
96 OPTIONALLY ENCLOSED BY '"'
97 LINES TERMINATED BY '\r\n'
98 IGNORE 1 ROWS;
```

Tot seguit, he comprovat que funcionés correctament:

```
101 # Comprovo els registres de la taula "users".
102 • SELECT *
103 FROM users;
104
```

Result Grid										
Filter Rows:										
Edit: Export/Import: Wrap Cell Content:										
user_id	name	surname	phone	email	birth_date	country	city	postal_code	address	
1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.edu	Nov 17, 1985	United States	Lowell	73544	348-7818 Sagittis St.	
10	Robert	Mccarthy	(324) 746-6771	fermentum@protonmail.com	Apr 30, 1984	United States	Eugene	85526	P.O. Box 773, 3594 Ornare St.	
100	Melodie	Mclean	1-677-221-7152	risus.varius@google.ca	Sep 15, 1989	United States	College	11838	Ap #644-8492 Sagittis St.	
101	Sarah	Beck	(358) 691-4345	vitae.risus@aol.couk	Apr 9, 1983	United States	Great Falls	67129	665-9047 In, Rd.	
102	Jasper	Landry	1-397-765-1118	consectetur.euismod@aol.org	Apr 16, 1982	United States	Columbus	11595	Ap #374-7325 Sodales Rd.	
103	Upton	Chavez	(227) 785-6484	euismod.est@aol.ca	Mar 15, 1986	United States	Essex	95631	1990 Vel, Av.	
104	Martha	Barlow	(732) 326-5448	vulputate@hotmail.net	Oct 29, 1988	United States	Chicago	41512	Ap #311-7103 In Avenue	

users 7 x

Output

Action Output

#	Time	Action	Message
1	12:20:27	SELECT * FROM users	275 row(s) returned

A continuació, hi he afegit la taula «transactions», que farà de taula de fets (és a dir, serà el centre de l'estrella) i per tant hauré d'afegir-hi com a *foreign keys* les *primary keys* que he indicat a les taules anteriors.

```
107 • CREATE TABLE IF NOT EXISTS transactions (  
108     transaction_id VARCHAR(255) PRIMARY KEY,  
109     card_id VARCHAR(15) ,  
110     company_id VARCHAR(15),  
111     timestamp TIMESTAMP,  
112     amount DECIMAL(10,2),  
113     declined BOOLEAN,  
114     product_ids VARCHAR(25),  
115     user_id VARCHAR(5),  
116     lat FLOAT,  
117     longitude FLOAT,  
118     FOREIGN KEY (company_id) REFERENCES companies(company_id),  
119     FOREIGN KEY (card_id) REFERENCES credit_cards(card_id),  
120     FOREIGN KEY (user_id) REFERENCES users(user_id)  
121 );
```

Un cop creada la taula, hi he carregat les dades de l'arxiu .csv:

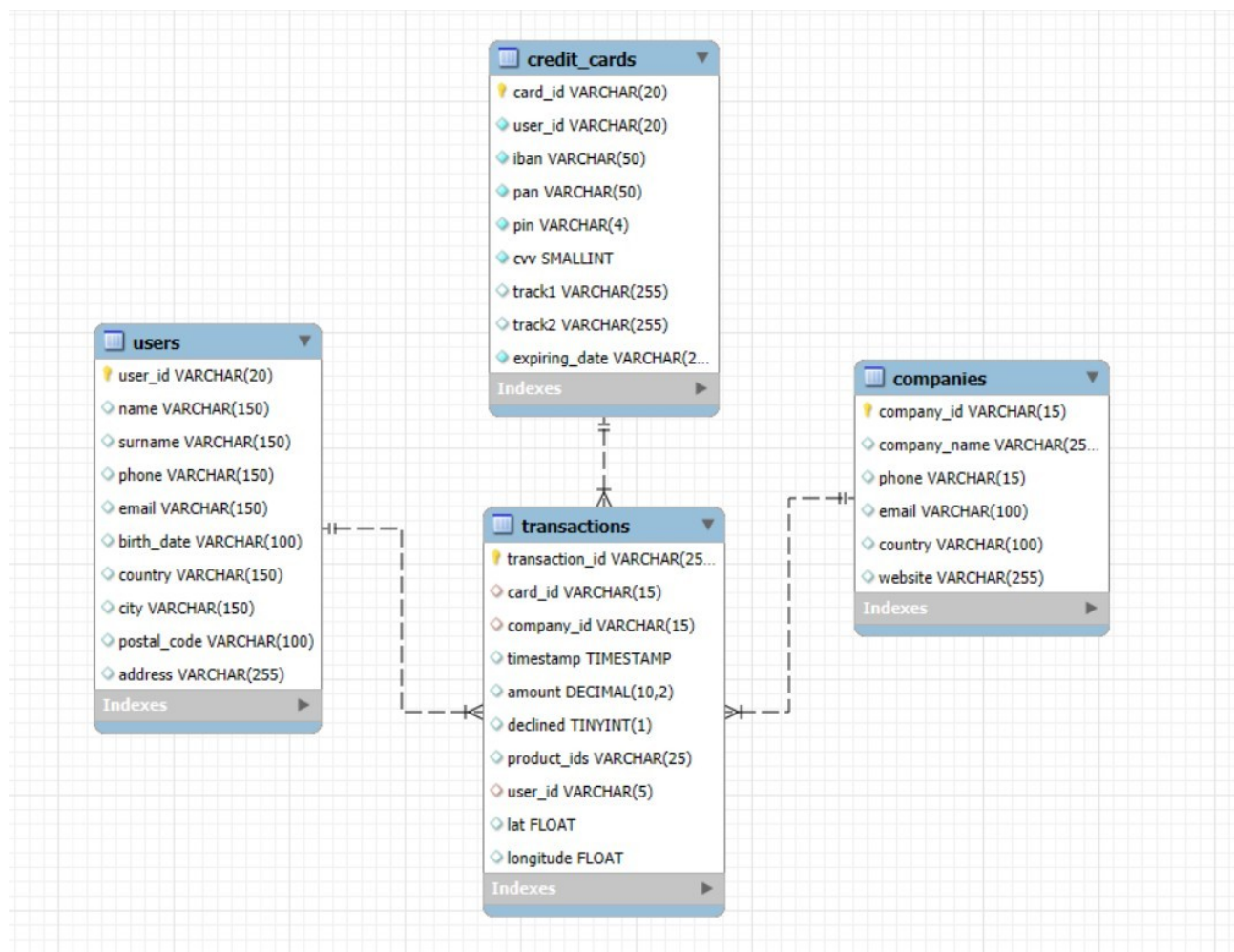
```
123     # Carrego les dades a "transactions"  
124 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'  
125 INTO TABLE transactions  
126 FIELDS TERMINATED BY ';'   
127 OPTIONALLY ENCLOSED BY '''  
128 LINES TERMINATED BY '\n'  
129 IGNORE 1 ROWS;
```


I tot seguit he comprovat que s'haguessin carregat correctament les dades:

```
130 #Comprovo que la taula s'ha carregat correctament
131
132 • SELECT *
133 FROM transactions;
```

	transaction_id	card_id	company_id	timestamp	amount	declined	product_ids	user_id	lat	longitude
▶	02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	2021-08-28 23:42:24	466.92	0	71, 1, 19	92	81.9185	-12.5276
	0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	2021-07-26 07:29:18	49.53	0	47, 97, 43	170	-43.9695	-117.525
	063FBA79-99EC-66FB-29F7-25726D1764A5	CcU-2987	b-2250	2022-01-06 21:25:27	92.61	0	47, 67, 31, 5	275	-81.2227	-129.05
	0668296C-CDB9-A883-76BC-2E4C44F8C8AE	CcU-3743	b-2618	2022-01-26 02:07:14	394.18	0	89, 83, 79	265	-34.3593	-100.556
	06CD9AA5-9B42-D684-DDDD-A5E394FEBA99	CcU-2959	b-2346	2021-10-26 23:00:01	279.93	0	43, 31	92	33.7381	158.298

Així doncs, després de carregar aquestes 4 taules (la de «products» la deixo per al nivell 3, perquè no la necessito per resoldre les consultes del Nivell 1), l'esquema que he obtingut és el següent:



És a dir, un model en estrella amb la taula de fets («transactions») al centre, que es relaciona amb «users», «credit_cards» i «companies» a partir de les seves primary keys corresponents: («user_id», «card_id» i «company_id»)

- Exercici 1

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

Per fer aquesta consulta, he seleccionat dades de la taula «users» (id i nom i cognoms dels usuaris) i dades de la transactions, l'id, al qual he aplicat un COUNT, perquè volia una suma de transaccions per veure quins usuaris n'havien fet més de 30. He fet un JOIN de les taules «users» i «transactions», he agrupat els resultats amb un GROUP BY dels id, nom i cognom dels usuaris, i he aplicat la condició que la suma de les transaccions fos de 30 o més amb un HAVING. Aquesta consulta ha retornar 4 files. Tot això es pot apreciar a la captura de pantalla següent:

```
143 # Amb JOIN
144 • SELECT u.user_id, u.name, u.surname, COUNT(t.transaction_id) as Total_transaccions
145 FROM users u
146 JOIN transactions t ON t.user_id = u.user_id
147 GROUP BY u.user_id, u.name, u.surname
148 HAVING Total_transaccions >= 30
149 ORDER BY Total_transaccions;
```

Result Grid				
Filter Rows: <input type="text"/>				
Export:				
Wrap Cell Content:				
	user_id	name	surname	Total_transaccions
▶	92	Lynn	Riddle	39
	275	Kenyon	Hartman	48
	267	Ocean	Nelson	52
	272	Hedwig	Gilbert	76

Result 10			
Output			
Action Output			
#	Time	Action	Message
✓ 1	12:50:41	SELECT u.user_id, u.name, u.surname, COUNT(t.transaction_id) as Total_transaccions FROM users u JOIN tran...	4 row(s) returned

Aquesta consulta també es pot resoldre sense fer cap JOIN, només a través d'una subconsulta, però en aquest cas només es podran mostrar dades de la taula «users». Faig una subconsulta on selecciono els «user_id» de la taula «transactions» amb la condició HAVING de 30 o més transaccions, i després a la consulta principal selecciono l'«user_id», «name» i «surname» de la taula «users» amb la condició que els «user_id» constin al resultat de la subconsulta. Això es pot apreciar a la captura de pantalla següent:

```
154 # Amb subconsulta
155 • SELECT u.user_id, u.name, u.surname
156 FROM users u
157 WHERE u.user_id IN (SELECT t.user_id
158                     FROM transactions t
159                     GROUP BY user_id
160                     HAVING COUNT(t.transaction_id) >= 30);
161
162
163
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	user_id	name	surname
▶	267	Ocean	Nelson
	272	Hedwig	Gilbert
	275	Kenyon	Hartman
	92	Lynn	Riddle
✱	NULL	NULL	NULL

users 13

✕

Output

Action Output

▼

#	Time	Action	Message
✔ 1	12:52:05	SELECT u.user_id, u.name, u.surname FROM users u WHERE u.user_id IN (SELECT t.user_id FROM transactio...	4 row(s) returned

- Exercici 2

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

Per resoldre aquest exercici, he fet un JOIN i una subconsulta. He seleccionat les dades que m'interessaven (l'iban de «credit_cards», un AVG arrodonit a 2 decimals de l'amount de la taula «transactions», que he rebatejat com a «mitjana_import»). He unit les dues taules amb un JOIN, i a continuació hi he fet la subconsulta. El «company_id» de la taula «transaction» ha de ser igual que el de la subconsulta, que selecciona el «company_id» de la taula «companies» en què el nom de la companyia és igual al nom que ens interessa, «Donec Ltd». No he afegit el filtre que la transacció havia de ser exitosa (WHERE declined = 0) perquè he interpretat que volíem totes les transaccions, les exitoses i les que no. Finalment, he agrupat els resultats segons l'iban, i el resultat de la mitjana ha estat de 203.72:

```
167      # Join i subconsulta
168 •    SELECT cc.iban, ROUND(AVG(t.amount), 2) as mitjana_import
169      FROM credit_cards cc
170      JOIN transactions t
171      ON t.card_id = cc.card_id
172      WHERE t.company_id = (SELECT company_id
173                          FROM companies
174                          WHERE company_name = 'Donec Ltd')
175      GROUP BY cc.iban;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	iban	mitjana_import			
▶	PT87806228135092429456346	203.72			

Aquesta mateixa consulta també s'hauria pogut resoldre amb 3 joins, però he considerat més simple l'opció anterior:

```
177      # 3 joins
178 •    SELECT cc.iban, ROUND(AVG(t.amount), 2) as mitjana_import
179      FROM credit_cards cc
180      JOIN transactions t
181      ON t.card_id = cc.card_id
182      JOIN companies c
183      on c.company_id = t.company_id
184      WHERE company_name = 'Donec Ltd'
185      GROUP BY cc.iban;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
iban	mitjana_import			
PT87806228135092429456346	203.72			

Nivell 2

Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions van ser declinades i genera la següent consulta:

Primer de tot he creat la nova taula, «estat_targetes», amb dues columnes: «id» (el mateix que a la taula «credit_cards») i l'estat de la targeta, que és de tipus ENUM (és a dir enumeració). Aquest tipus de dada inclou un conjunt de constants predefinides, que en aquest cas són «activa» i «no activa». També vaig considerar que el tipus de dada de la columna «estat» fos TINYINT per incloure els valors booleans 0 i 1 (0 targeta activa, 1 no activa), però finalment vaig pensar que quedaria tot més clar d'aquesta manera.

```
# Creo la taula "estat_targetes"
```

```
CREATE TABLE estat_targetes
  (id VARCHAR(15) PRIMARY KEY,
   estat ENUM('activa', 'no activa') NOT NULL);
```

Volia omplir la taula d'estat targetes només amb les targetes actives, és a dir, aquelles en que com a mínim una de les últimes tres transaccions no hagués estat declinada. Per fer-ho, he fet servir aquest codi:

```
# Creo les instruccions per afegir les targetes que tenen més de tres transaccions declinades a la nova taula.
```

```
INSERT INTO estat_targetes (card_id, estat)
SELECT card_id,
  CASE
    WHEN COUNT(CASE WHEN declined = 1 THEN 1 END) >= 3
    THEN 'no activa'
    ELSE 'activa'
  END AS estat
FROM (SELECT card_id, declined
      FROM (SELECT card_id, declined, ROW_NUMBER()
              OVER (PARTITION BY card_id ORDER BY data_timestamp DESC) AS Num_tr
            FROM transactions) subconsulta1
      WHERE Num_tr <= 3) subconsulta_transaccions
GROUP BY card_id;
```


La consulta més interna, «subconsulta1» assigna un número a cada transacció segons la data (elfeta amb targeta (el número 1 la més recent):

```
FROM (SELECT card_id, declined, ROW_NUMBER()  
OVER (PARTITION BY card_id ORDER BY data_timestamp DESC) AS Num_tr  
FROM transactions) subconsulta1
```

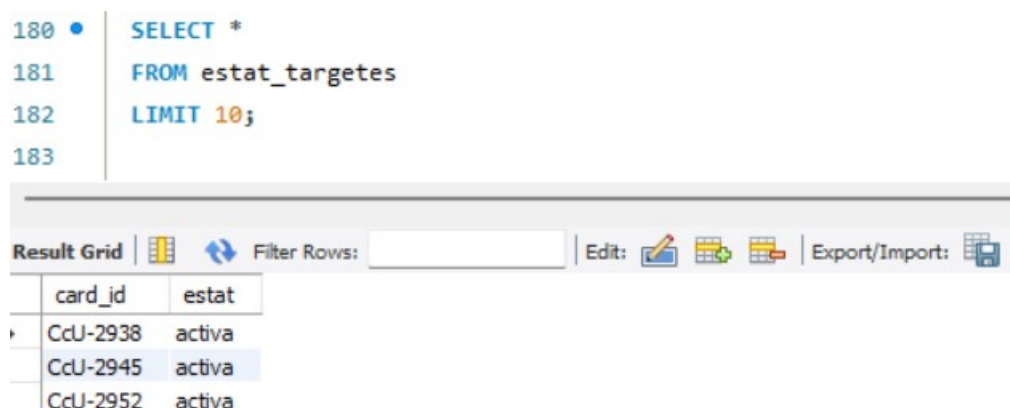
La següent subconsulta, més externa, selecciona només les últimes tres transaccions de casa targeta:

```
SELECT card_id, declined  
FROM (SELECT card_id, declined, ROW_NUMBER()  
OVER (PARTITION BY card_id ORDER BY data_timestamp DESC) AS Num_tr  
FROM transactions) subconsulta1  
WHERE Num_tr <= 3
```

Tot seguit, amb un CASE, classifico les targetes en «activa» o «no activa»: Si com a mínim una de les tres últimes transaccions no ha estat declinada, la targeta passarà a considerar-se «activa», però si totes les últimes tres transaccions han estat declinades, serà «no activa»:

```
CASE  
WHEN COUNT(CASE WHEN declined = 1 THEN 1 END) >= 3  
THEN 'no activa'  
ELSE 'activa'  
END AS estat
```

I finalment introdueixo a la taula d'«estat_targetes» només les targetes que es consideren actives, i confirmo que les dades s'hi han afegit correctament (amb un límit de 10 perquè no necessito veure tots els registres):



```
180 • SELECT *  
181 FROM estat_targetes  
182 LIMIT 10;  
183
```





card_id	estat
CcU-2938	activa
CcU-2945	activa
CcU-2952	activa

- Exercici 1

Quantes targetes estan actives?

Per saber quantes targetes estan actives, necessitava saber el número de registres de targetes actives a la taula d'«estat_targetes». Per fer-ho he fet un COUNT de tots els registres d'aquesta taula, i el resultat ha estat 275. Així doncs, hi ha 275 targetes actives.

185	# Quantes targetes estan actives?
186	
187 •	SELECT COUNT(*) as Núm_targetes_actives
188	FROM estat_targetes;
189	

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	Núm_targetes_actives				
▶	275				

Un cop creada aquesta nova taula «estat_targetes», que es relaciona amb la de «credit_cards», l'esquema que ens queda és el següent:

