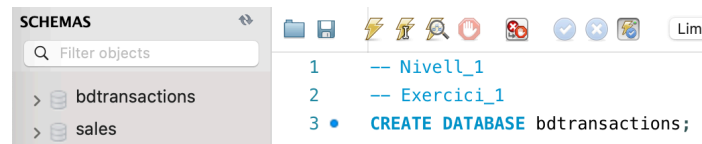


Nivell 1

- Exercici 1

Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguis realitzar les següents consultes:

Primer creem una nova base de dades a través de codi:



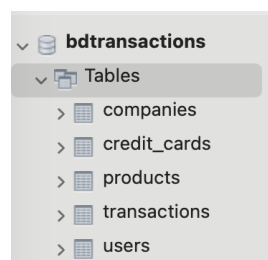
Una vegada observats tots els arxius csv, dissenyarem una base de dades amb un esquema d'estrella. La consulta que utilitzarem per crear les taules és aquesta:

```
6 • CREATE TABLE companies (  
7     company_id VARCHAR(15) PRIMARY KEY,  
8     company_name VARCHAR(255),  
9     phone VARCHAR(15),  
10    email VARCHAR(100),  
11    country VARCHAR(100),  
12    website VARCHAR(255)  
13 );  
14 • CREATE TABLE users (  
15     id INT PRIMARY KEY,  
16     name VARCHAR(255),  
17     surname VARCHAR(255),  
18     phone VARCHAR(20),  
19     email VARCHAR(255),  
20     birth_date VARCHAR(20),  
21     country VARCHAR(255),  
22     city VARCHAR(255),  
23     postal_code VARCHAR(50),  
24     address VARCHAR(255)  
25 );  
26 • CREATE TABLE credit_cards (  
27     id VARCHAR(15) PRIMARY KEY,  
28     user_id INT,  
29     iban VARCHAR(100),  
30     pan VARCHAR(100),  
31     pin VARCHAR(100),  
32     cvv VARCHAR(15),  
33     track1 VARCHAR(255),  
34     track2 VARCHAR(255),  
35     expiring_date VARCHAR(25),  
36     FOREIGN KEY (user_id) REFERENCES users(id)  
37 );  
38 • CREATE TABLE products (  
39     id INT auto_increment PRIMARY KEY,  
40     product_name VARCHAR(100),  
41     price DECIMAL(10,2),  
42     colour VARCHAR(50),  
43     weight DECIMAL(10,2),  
44     warehouse_id VARCHAR(10)  
45 );  
46 • CREATE TABLE transactions (  
47     id VARCHAR(255) PRIMARY KEY,  
48     card_id VARCHAR(15),  
49     business_id VARCHAR(15),  
50     timestamp TIMESTAMP,  
51     amount DECIMAL(10, 2),  
52     declined BOOLEAN,  
53     product_ids INT,  
54     user_id INT,  
55     lat FLOAT,  
56     longitude FLOAT,  
57     FOREIGN KEY (card_id) REFERENCES credit_cards(id),  
58     FOREIGN KEY (business_id) REFERENCES companies(company_id),  
59     FOREIGN KEY (product_ids) REFERENCES products(id),  
60     FOREIGN KEY (user_id) REFERENCES users(id)  
61 );
```

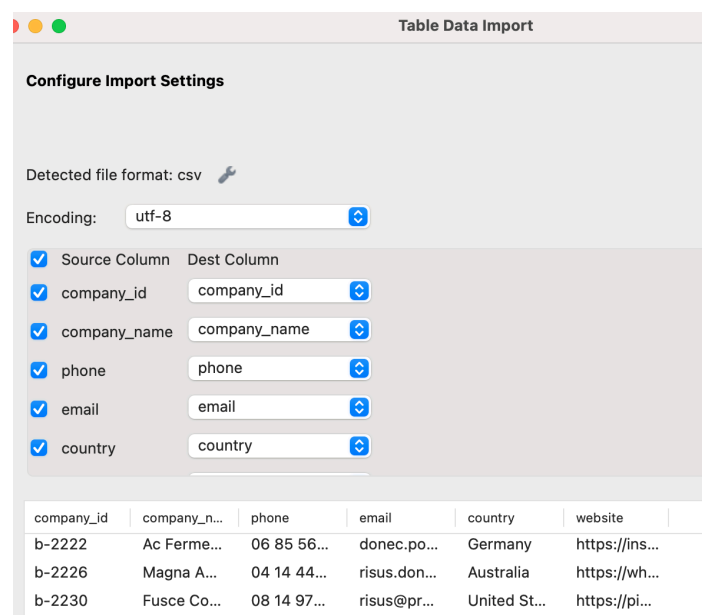
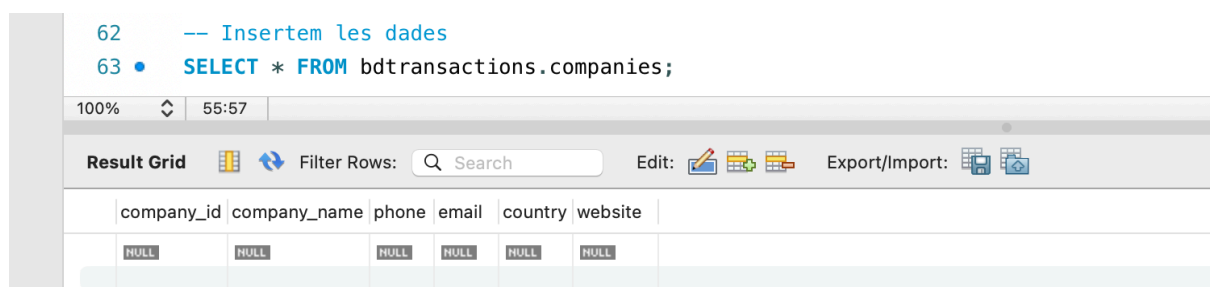
Com es pot observar en la consulta hem creat cinc taules: “transactions” és la taula de fets i la resta són taules de dimensions. La taula “users” agrupa els arxius csv de les tres regions: “users_ca”, “users_uk” i “users_usa”.

Una vegada creades les taules amb les columnes específiques, fem les definicions explícites de les claus forànees. Amb això també ens assegurem que en el moment de relacionar, ambdues columnes, per exemple ‘products.id’ i ‘transactions.products_ids’, tinguin la mateixa estructura de dades, en aquest cas INT.

A la taula “credit_cards”, la clau externa **user_id** estableix una relació amb la taula “users”, perquè donem per fet que cada targeta està associada a un únic usuari.



A continuació fem l'importació des de l'opció “Import records from an external file” per tal d'introduir totes les dades:



Una vegada executats tots els passos, ens apareix que s'han agregat 100 registres i en fer la consulta de nou, observem que les dades s'han introduït correctament:

62 -- Insertem les dades

63 • SELECT * FROM bdtransactions.companies;

100%

40:63

Result Grid

Filter Rows:

Search

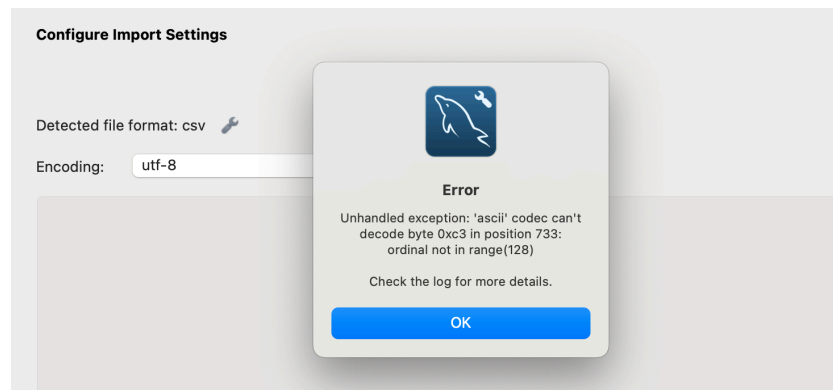
Edit:

Export/Import:

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@icloud.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
b-2238	Ante laculis Nec Foundation	08 23 04 99 53	sed dictum proin@outlook.ca	New Zealand	https://netflix.com/settings

Repetim aquesta acció amb la resta de taules, tenint en compte que la taula “users” agrupa tres csv diferents.

En arribar a l'arxiu “users_ca” ens indica un problema relacionat amb la codificació de caràcters:

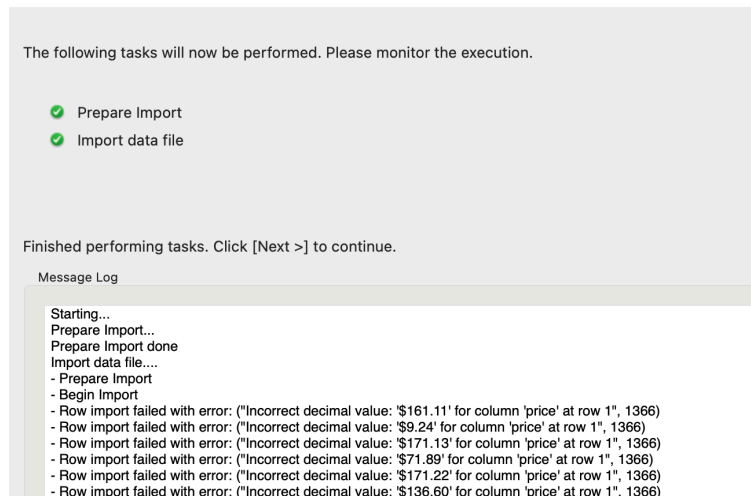


Anem a revisar el csv i ràpidament detectem que el problema ve d'un registre que s'ha escrit amb accent “Québec”, caràcter que no està en el rang ASCII estàndard.

```
edu, Dec 2, 1980", Canada, Murdochville, S7E 6L0, Ap #880-03
.couk, "Jan 6, 1982", Canada, Baddeck, K3X 6Z5, 441-8969 Rhon
tlook.ca, "Oct 15, 1994", Canada, Maple Creek, Y2C 9E6, "517-
olor@google.ca, "May 7, 1983", Canada, Dieppe, E7S 4P8, Ap #9
utlook.net, "Nov 17, 2000", Canada, Québec City, B4K 0J6, 858
m@google.edu, "Sep 22, 1981", Canada, Rae-Edzo, 20Y 8L2, Ap #
eb 23, 1993", Canada, Annapolis Royal, S4Y 8V5, Ap #983-6042
```

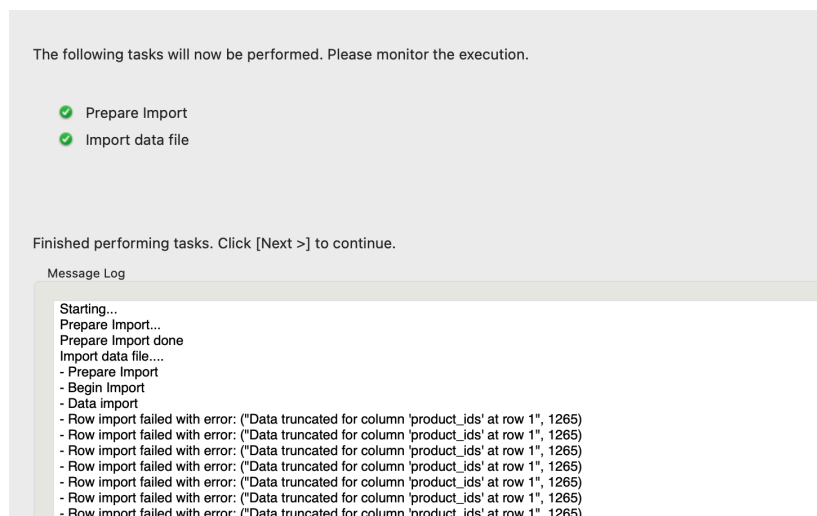
Modifiquem el csv per i desem. Una vegada fet aquest canvi els arxius s'importen correctament.

D'altra banda, a l'hora d'importar els registres de la taula "products" també ens retorna un error en els valors de la columna 'price' ja que aquests contenen el símbol '\$':



Per solucionar-ho, anem a l'arxiu csv "products" i fem un search i replace de tots els símbols '\$' i desem. Tornem a importar i s'executa sense errors.

La última taula que ens queda per importar les dades és "transactions" i de nou ens apareixen errors:



Optem per modificar l'esquema de la taula a format cadena perquè accepti els valors, abans però hem d'eliminar temporalment la restricció de clau externa:

```
67 -- Modifiquem el tipus de dades de la columna 'products_ids'
68 • START TRANSACTION;
69 • ALTER TABLE transactions DROP FOREIGN KEY transactions_ibfk_3;
70 • ALTER TABLE products
71   MODIFY COLUMN id VARCHAR(255);
72 • ALTER TABLE transactions
73   MODIFY COLUMN product_ids VARCHAR(255);
74 • ALTER TABLE transactions ADD CONSTRAINT FOREIGN KEY (product_ids) REFERENCES products(id);
--
```

Un cop modificat tornem a provar d'introduir les dades però tan sols es bolquen 110 registres dels 587, ja que la columna 'product_ids' conté més d'un codi de producte associat separat per comes. Així que optem per crear una taula d'unió que relacioni les transaccions amb els productes. Anomenem la nova taula de relació "transactions_products":

```
78 -- Creem la taula de relació
79 • CREATE TABLE transactions_products (
80     transaction_product_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
81     transaction_id VARCHAR(255),
82     product_id VARCHAR(255),
83     FOREIGN KEY (transaction_id) REFERENCES transactions(id),
84     FOREIGN KEY (product_id) REFERENCES products(id)
85 );
```

Com que ja tenim creada la taula de relació "transactions_products", podem eliminar la columna 'products_ids' de la taula "transactions" i inserir les dades, que s'han creat mitjançant un script de python.

```
86 -- Eliminem la restricció de clau i la columna product_ids de la taula transactions
87 • ALTER TABLE transactions DROP FOREIGN KEY transactions_ibfk_3;
88 • ALTER TABLE transactions DROP COLUMN product_ids;
89 -- Insertem les dades
90 • ALTER TABLE transactions_products DROP FOREIGN KEY transactions_products_ibfk_1;
91 • SELECT * FROM bdtransactions.transactions;
```

Aquest script el que fa és extreure l'id de "transactions" i els id de "products" de cada fila i imprimeix una instrucció SQL INSERT per cada combinació de id de transactions i id de products, per inserir-los a la taula de relació "transactions_products".

```
script.py > ...
1 import csv
2
3 # Open the CSV file and iterate through each row
4 with open('transactions.csv', 'r') as file:
5     reader = csv.DictReader(file, delimiter=',')
6     for row in reader:
7         transaction_id = row['id']
8         product_ids = row['product_ids'].split(',')
9
10        # Insert a record into transactions_products for each product ID
11        for product_id in product_ids:
12            # Perform the insertion
13            print(f"INSERT INTO transactions_products (transaction_id, product_id) VALUES ('{transaction_id}', '{product_id.strip()}');")
14
```

```
1 -- Inserim les dades de transactions_products
2 • INSERT INTO transactions_products (transaction_id, product_id) VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A05DD', '59');
3 • INSERT INTO transactions_products (transaction_id, product_id) VALUES ('7DC26247-20EC-53FE-E555-B6C2E55CA5D5', '71');
4 • INSERT INTO transactions_products (transaction_id, product_id) VALUES ('7DC26247-20EC-53FE-E555-B6C2E55CA5D5', '41');
5 • INSERT INTO transactions_products (transaction_id, product_id) VALUES ('72997E96-DC2C-A4D7-7C24-66C302F8AE5A', '97');
6 • INSERT INTO transactions_products (transaction_id, product_id) VALUES ('72997E96-DC2C-A4D7-7C24-66C302F8AE5A', '41');
7 • INSERT INTO transactions_products (transaction_id, product_id) VALUES ('72997E96-DC2C-A4D7-7C24-66C302F8AE5A', '3');
8 • INSERT INTO transactions_products (transaction_id, product_id) VALUES ('AB069F53-965E-A2A8-CE06-CA8C4FD92501', '11');
9 • INSERT INTO transactions_products (transaction_id, product_id) VALUES ('AB069F53-965E-A2A8-CE06-CA8C4FD92501', '13');
10 • INSERT INTO transactions_products (transaction_id, product_id) VALUES ('AB069F53-965E-A2A8-CE06-CA8C4FD92501', '61');
11 • INSERT INTO transactions_products (transaction_id, product_id) VALUES ('AB069F53-965E-A2A8-CE06-CA8C4FD92501', '29');
```

Fem import les dades de “transactions”:

Import Results


File /Users/maria/Downloads/transactions.csv was imported in 2.190 s






Table bdtransactions.transactions has been used

587 records imported

I la taula de relació queda de la següent manera:

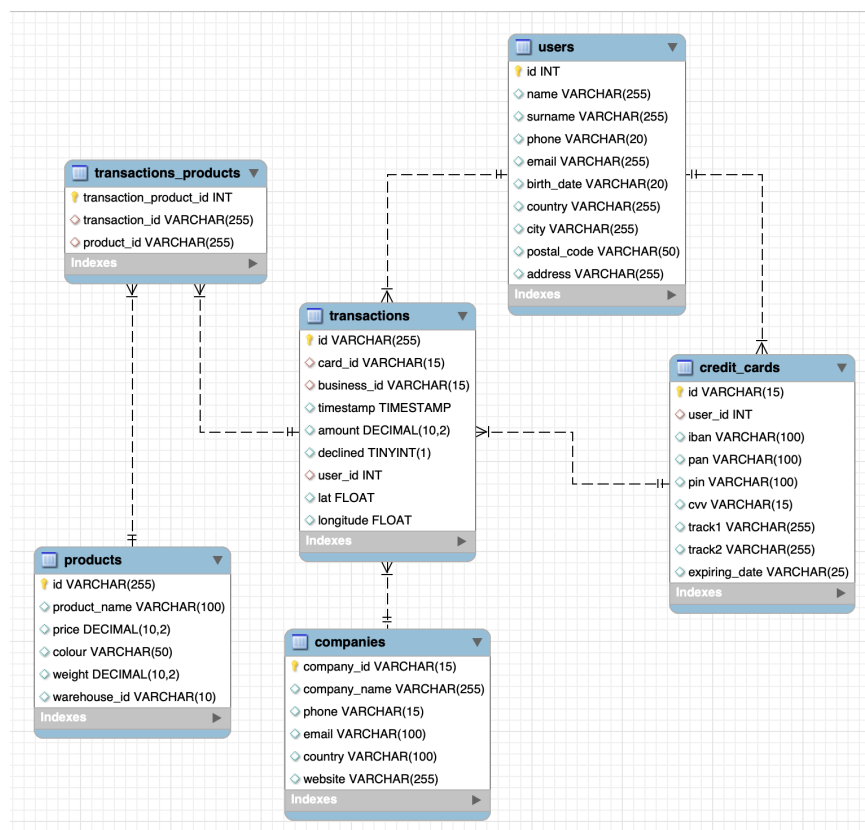
```
92 • SELECT * FROM bdtransactions.transactions_products;
```

100%  85:86

Result Grid   Filter Rows: Edit:   

	transaction_product_id	transaction_id	product_id
1		108B1D1D-5B23-A76C-55FE-C568E49A05DD	59
2		7DC26247-20EC-53FE-E555-B6C2E55CA5D5	71
3		7DC26247-20EC-53FE-E555-B6C2E55CA5D5	41
4		72997E96-DC2C-A4D7-7C24-66C302F8AE5A	97
5		72997E96-DC2C-A4D7-7C24-66C302F8AE5A	41
6		72997E96-DC2C-A4D7-7C24-66C302F8AE5A	3
7		AB069F53-965E-A2A8-CE06-CA8C4FD92501	11
8		AB069F53-965E-A2A8-CE06-CA8C4FD92501	13
9		AB069F53-965E-A2A8-CE06-CA8C4FD92501	61
10		AB069F53-965E-A2A8-CE06-CA8C4FD92501	29

L'esquema final de la base de dades queda de la següent manera:



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

```

95 -- Mostrem tots els usuaris amb més de 30 transaccions
96 • SELECT DISTINCT t.*, counts.user_id AS user_id, counts.num_transactions AS total_transactions
97 FROM transactions t
98 INNER JOIN (
99     SELECT user_id, COUNT(*) AS num_transactions
100 FROM transactions
101 GROUP BY user_id
102 HAVING COUNT(*) > 30
103 ) AS counts ON t.user_id = counts.user_id
104 INNER JOIN users u ON t.user_id = u.id;
105
10% 4:93

```

id	card_id	business_id	timestamp	amount	declined	user_id	lat	longitude	user_id	total_tra...
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	2021-08-28 23:42:24	466.92	0	92	81.9185	-12.5276	92	39
06CD9AA5-9B42-D684-DDDD-A5E394FEB499	CcU-2959	b-2346	2021-10-26 23:00:01	279.93	0	92	33.7381	158.298	92	39
1026DA24-8929-31F1-8250-D7BAB05C13D2	CcU-2959	b-2346	2021-12-07 09:30:38	465.31	0	92	32.0646	-140.147	92	39
156F3F80-7E7D-65CF-727D-6AE03CEB7520	CcU-2959	b-2346	2021-04-29 07:06:10	404.16	0	92	-59.9778	172.731	92	39
17561134-37A4-E316-2C3C-C5541EDAD761	CcU-2938	b-2362	2021-09-15 05:23:32	371.57	0	92	-10.8529	18.0969	92	39
185C9F2C-67A2-8EC0-0BB6-93789CC59F05	CcU-2938	b-2362	2021-07-25 12:34:59	285.24	0	92	9.55928	165.726	92	39
1998A483-B4CD-FAE6-745C-8F2F2EB77A8E	CcU-2959	b-2362	2021-08-29 10:53:03	372.58	0	92	-6.77047	128.679	92	39
1D03001C-97EC-77F6-6E7E-8E2CAC237EA4	CcU-2959	b-2346	2021-06-30 15:59:43	32.36	0	92	-39.7167	-18.0415	92	39
26D8A738-B651-A620-1EEA-D8E2CD353BEB	CcU-2938	b-2362	2021-05-28 15:21:36	238.99	0	92	15.6872	25.2507	92	39
2899837A-45A3-D3B1-0D03-8B33B867DF9B	CcU-2959	b-2346	2021-12-11 21:36:59	324.34	0	92	-1.55702	-95.718	92	39
2A37F5B3-11D8-3BA4-6E94-B7407D91942C	CcU-2959	b-2362	2022-01-10 20:31:41	83.19	0	92	-81.8177	-65.964	92	39
310B8BF9-A441-1470-EDEF-417B7EF2B169	CcU-2959	b-2346	2021-04-04 11:53:52	406.13	0	92	7.53465	-136.923	92	39
36AD59C1-944F-C355-8358-B83A446F3272	CcU-2959	b-2346	2021-06-10 01:43:46	492.19	0	92	-26.9569	-2.8433	92	39

La consulta retorna 215 files, és a dir 215 transaccions. Per tal d'identificar més ràpidament de quins usuaris es tracten fem una nova subconsulta que mostri únicament els usuaris que han fet més de 30 transaccions. Apareixen aquest quatre usuaris:

```

117 • SELECT DISTINCT user_id
118 FROM (
119     SELECT DISTINCT t.*, counts.num_transactions AS total_transactions
120 FROM transactions t
121 INNER JOIN (
122     SELECT user_id, COUNT(*) AS num_transactions
123 FROM transactions
124 GROUP BY user_id
125 HAVING COUNT(*) > 30
126 ) AS counts ON t.user_id = counts.user_id
127 INNER JOIN users u ON t.user_id = u.id
128 ) AS distinct_users;

```

100% 3:116

user_id
92
267
272
275

- Exercici 2

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

```
131 • SELECT cc.iban, AVG(t.amount) AS average_amount
132 FROM transactions t
133 JOIN credit_cards cc ON t.card_id = cc.id
134 WHERE t.business_id IN (SELECT companies.company_id FROM companies WHERE company_name = 'Donec Ltd')
135 GROUP BY cc.iban;
```



iban	average_amount
PT87806228135092429456346	203.715000

Nivell 2

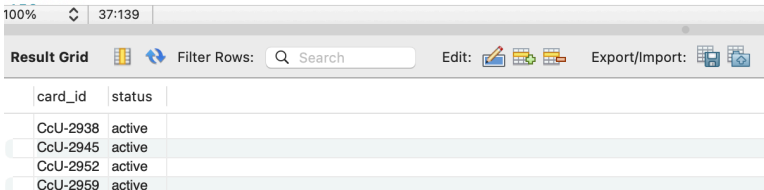
Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions van ser declinades i genera la següent consulta:

En primer lloc creem la nova taula que anomenem "credit_card_status":

```
137 -- Creem la nova taula amb l'estat de les targetes de crèdit
138 • CREATE TABLE credit_card_status (
139     card_id VARCHAR(15) PRIMARY KEY,
140     status VARCHAR(25)
141 );
```

A continuació, definim com estat actiu o inactiu en funció de si les tres últimes transaccions han sigut declinades per cada targeta, i després insertem els resultats a la taula "credit_card_status":

```
142 -- Calculem l'estat de les targetes de crèdit
143 • INSERT INTO credit_card_status (card_id, status)
144     SELECT card_id,
145         CASE
146             WHEN COUNT(*) = 3 AND SUM(declined) = 3 THEN 'inactive'
147             ELSE 'active'
148         END AS status
149     FROM (
150         SELECT card_id, declined
151         FROM transactions
152         ORDER BY card_id, timestamp DESC
153     ) AS sorted_transactions
154     GROUP BY card_id;
155 -- Mostrem la taula credit_card_status
156 • SELECT * FROM bdtransactions.credit_card_status;
```



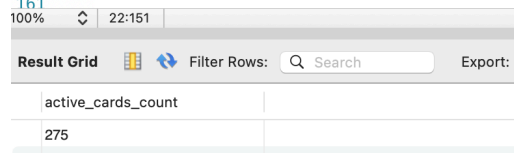
card_id	status
CcU-2938	active
CcU-2945	active
CcU-2952	active
CcU-2959	active

- Exercici 1

Quantes targetes estan actives?

Totes les targetes estan actives. Un total de 275.

```
157 -- Exercici_1
158 • SELECT COUNT(*) AS active_cards_count
159 FROM credit_card_status
160 WHERE status = 'active';
161
```



active_cards_count
275

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:

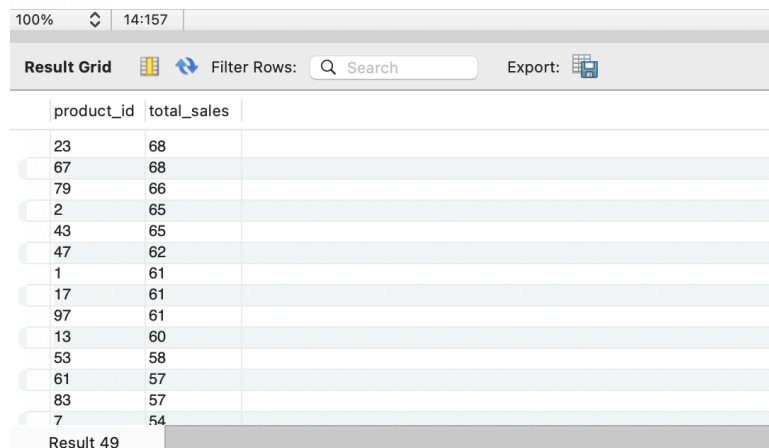
Anteriorment, ja hem creat la taula “transactions_products” que ens permet unir les dades de transaccions amb les de productes.

- Exercici 1

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

Realitzem la consulta que ens mostra quantes vegades s'ha venut cada producte. Recordem que hi ha un total de 1457 transaccions:

```
163 -- Exercici_1
164 • SELECT product_id, COUNT(*) AS total_sales
165 FROM transactions_products
166 GROUP BY product_id
167 ORDER BY total_sales DESC;
```



product_id	total_sales
23	68
67	68
79	66
2	65
43	65
47	62
1	61
17	61
97	61
13	60
53	58
61	57
83	57
7	54