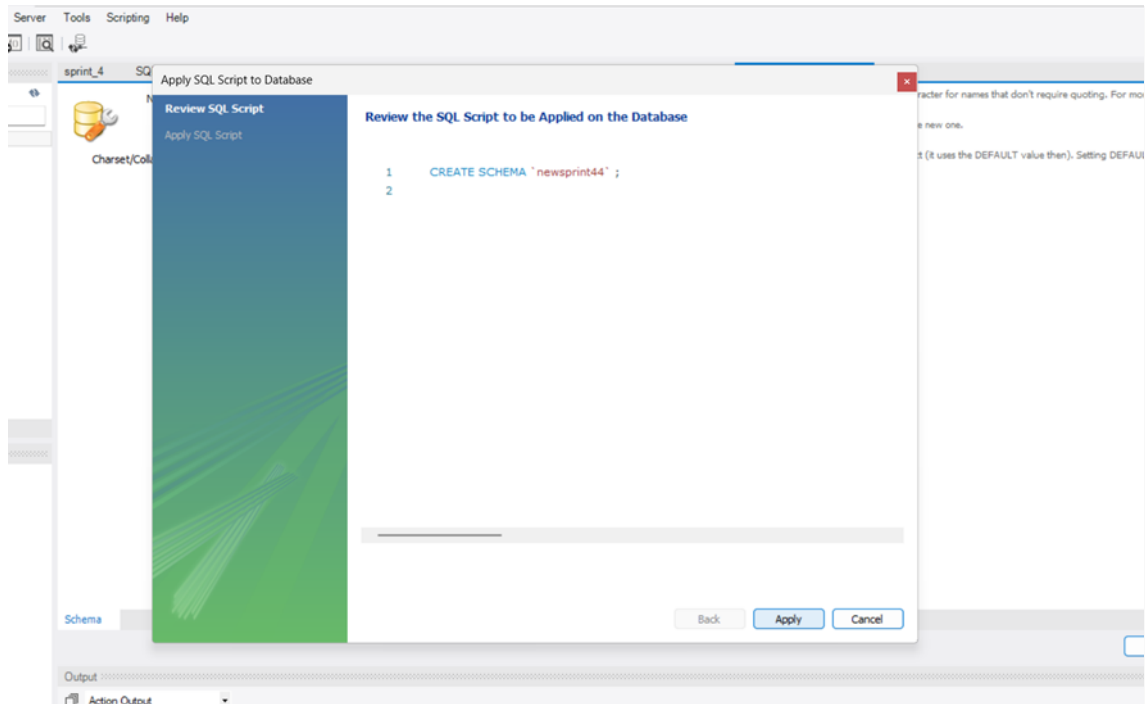


[Escriba aquí]

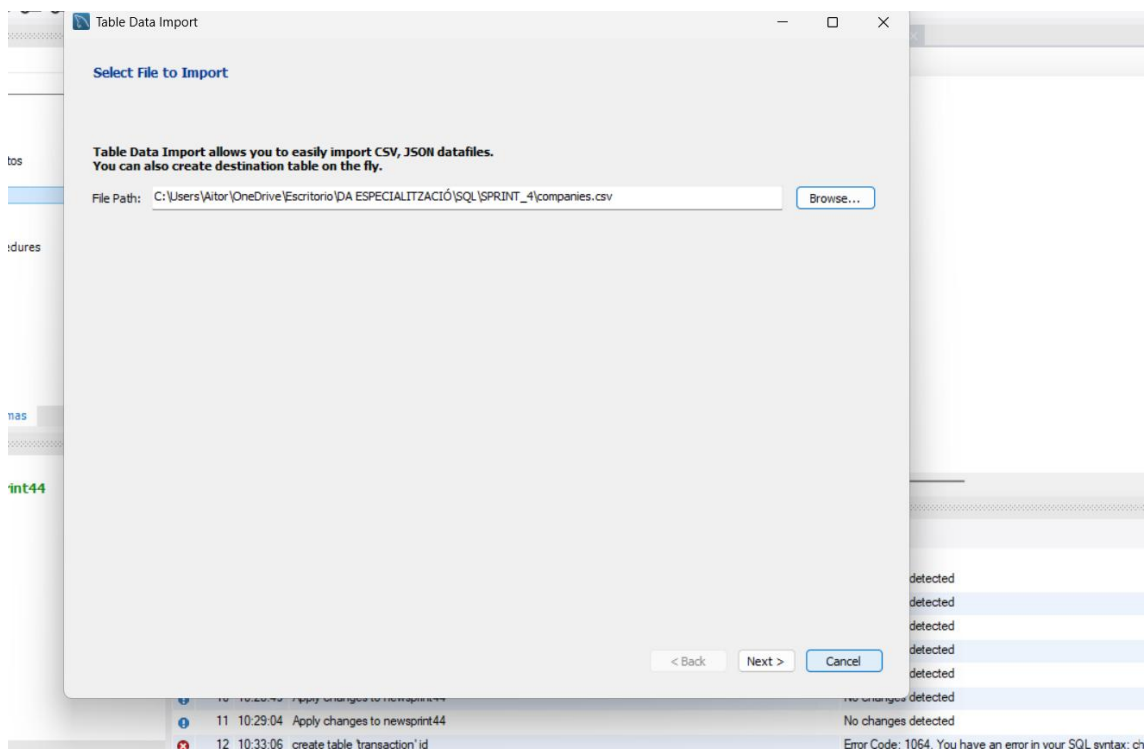
SPRINT 4:

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 puguem realitzar les següents consultes:

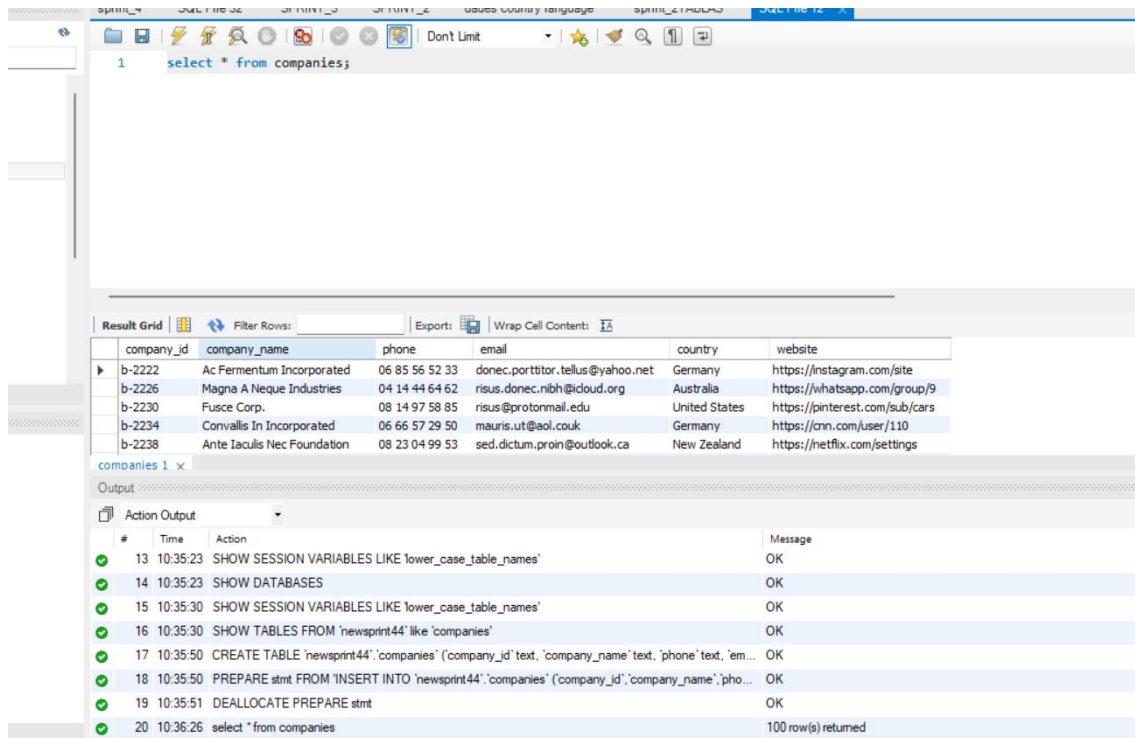
Creació d'un nou schema per a treballar amb una nova base de dades:



Ara importem les dades:



[Escriba aquí]



The screenshot shows a database interface with a SQL query editor at the top containing the query `select * from companies;`. Below the editor is a 'Result Grid' displaying the results of the query. The results are organized into a table with the following columns: `company_id`, `company_name`, `phone`, `email`, `country`, and `website`. The table contains five rows of data. Below the result grid, there is an 'Output' section showing a log of database actions, including session variable checks, database listings, table listings, table creation, data insertion, and deallocation, all of which completed successfully.

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.co.uk	Germany	https://cnn.com/user/110
b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings

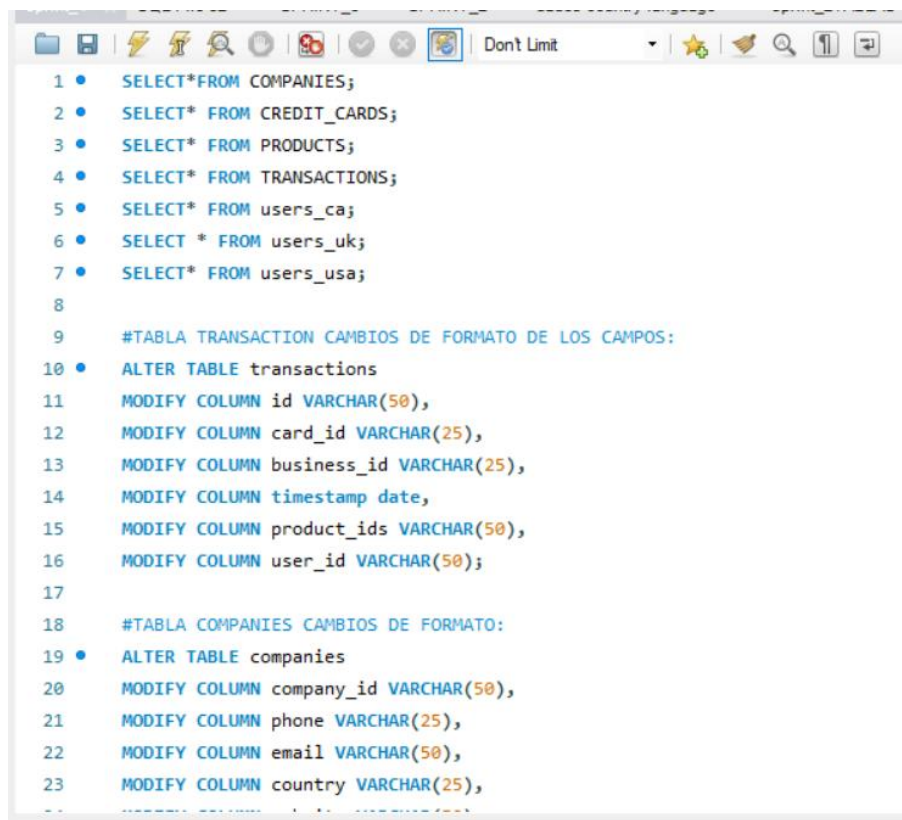
companies 1 x

Output

Action Output

#	Time	Action	Message
13	10:35:23	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
14	10:35:23	SHOW DATABASES	OK
15	10:35:30	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
16	10:35:30	SHOW TABLES FROM 'newsprint44' like 'companies'	OK
17	10:35:50	CREATE TABLE 'newsprint44'.'companies' ('company_id' text, 'company_name' text, 'phone' text, 'em...	OK
18	10:35:50	PREPARE stmt FROM 'INSERT INTO 'newsprint44'.'companies' ('company_id','company_name','pho...	OK
19	10:35:51	DEALLOCATE PREPARE stmt	OK
20	10:36:26	select * from companies	100 row(s) returned

Comprobem que s'han importat bé les dades i això ho fem amb cadascún dels arxius csv.



The screenshot shows a SQL script editor with a series of SQL commands. The first seven lines are `SELECT` statements for different tables: `COMPANIES`, `CREDIT_CARDS`, `PRODUCTS`, `TRANSACTIONS`, `users_ca`, `users_uk`, and `users_usa`. Lines 9 and 18 are comments in Catalan: `#TABLA TRANSACTION CAMBIOS DE FORMATO DE LOS CAMPOS:` and `#TABLA COMPANIES CAMBIOS DE FORMATO:`. Lines 10 through 16 are `ALTER TABLE` commands for the `transactions` table, modifying the data types of `id`, `card_id`, `business_id`, `timestamp`, `product_ids`, and `user_id` to `VARCHAR` with specific lengths. Lines 19 through 23 are `ALTER TABLE` commands for the `companies` table, modifying the data types of `company_id`, `phone`, `email`, and `country` to `VARCHAR` with specific lengths.

```
1 • SELECT*FROM COMPANIES;
2 • SELECT* FROM CREDIT_CARDS;
3 • SELECT* FROM PRODUCTS;
4 • SELECT* FROM TRANSACTIONS;
5 • SELECT* FROM users_ca;
6 • SELECT * FROM users_uk;
7 • SELECT* FROM users_usa;
8
9 #TABLA TRANSACTION CAMBIOS DE FORMATO DE LOS CAMPOS:
10 • ALTER TABLE transactions
11   MODIFY COLUMN id VARCHAR(50),
12   MODIFY COLUMN card_id VARCHAR(25),
13   MODIFY COLUMN business_id VARCHAR(25),
14   MODIFY COLUMN timestamp date,
15   MODIFY COLUMN product_ids VARCHAR(50),
16   MODIFY COLUMN user_id VARCHAR(50);
17
18 #TABLA COMPANIES CAMBIOS DE FORMATO:
19 • ALTER TABLE companies
20   MODIFY COLUMN company_id VARCHAR(50),
21   MODIFY COLUMN phone VARCHAR(25),
22   MODIFY COLUMN email VARCHAR(50),
23   MODIFY COLUMN country VARCHAR(25),
```

[Escriba aquí]

Ara comencem a afegir les primary keys, foreign keys i canviem alguns dels formats dels camps:

```
8  ##AÑADIMOS LAS PK Y FK DE LAS TABLAS:
9  #TABLA TRANSACTIONS:
10 • ALTER TABLE transactions
11   ADD PRIMARY KEY (id(50));
12
13 • ALTER TABLE COMPANIES
14   ADD PRIMARY KEY (company_id(50));
15
16 • ALTER TABLE CREDIT_CARDS
17   ADD PRIMARY KEY (id(50));
18
19 • ALTER TABLE PRODUCTS
20   ADD PRIMARY KEY (id);
21
22 • ALTER TABLE users_ca
23   ADD PRIMARY KEY (id);
24
25 • ALTER TABLE users_uk
26   ADD PRIMARY KEY (id);
27
28 • ALTER TABLE users_usa
29   ADD PRIMARY KEY (id);
30
31
```

```
1  #TABLA TRANSACTION CAMBIOS DE FORMATO DE LOS CAMPOS:
2
3 • ALTER TABLE transactions
4   MODIFY COLUMN id VARCHAR(50),
5   MODIFY COLUMN card_id VARCHAR(25),
6   MODIFY COLUMN business_id VARCHAR(25),
7   MODIFY COLUMN timestamp date,
8   MODIFY COLUMN product_ids VARCHAR(25);
9
10 #TABLA COMPANIES CAMBIOS DE FORMATO:
11 • ALTER TABLE companies
12   MODIFY COLUMN company_id VARCHAR(50),
13   MODIFY COLUMN phone VARCHAR(25),
14   MODIFY COLUMN email VARCHAR(50),
15   MODIFY COLUMN country VARCHAR(25),
16   MODIFY COLUMN website VARCHAR(50);
```

[Escriba aquí]

#TABLA credit_cards CAMBIOS DE FORMATO:

- ```
ALTER TABLE credit_cards
MODIFY COLUMN user_id VARCHAR(50),
MODIFY COLUMN iban VARCHAR(50),
MODIFY COLUMN pan VARCHAR(50),
MODIFY COLUMN pin VARCHAR(10),
MODIFY COLUMN cvv VARCHAR(10),
MODIFY COLUMN track1 VARCHAR(50),
MODIFY COLUMN track2 VARCHAR(50),
MODIFY COLUMN expiring_date VARCHAR(10);
```

#TABLA PRODUCTS DE FORMATO:

- ```
ALTER TABLE products
MODIFY COLUMN id VARCHAR(50),
MODIFY COLUMN product_name VARCHAR(50),
MODIFY COLUMN price VARCHAR(50),
MODIFY COLUMN colour VARCHAR(10),
MODIFY COLUMN weight VARCHAR(10),
MODIFY COLUMN warehouse_id VARCHAR(50);
```

```
MODIFY COLUMN surname VARCHAR(15),
MODIFY COLUMN phone VARCHAR(50),
MODIFY COLUMN email VARCHAR(50),
MODIFY COLUMN birth_date VARCHAR(20),
MODIFY COLUMN country VARCHAR(15),
MODIFY COLUMN city VARCHAR(25),
MODIFY COLUMN postal_code VARCHAR(10),
MODIFY COLUMN address VARCHAR(50);
```

#TABLA users_usa DE FORMATO:

- ```
ALTER TABLE users_usa
MODIFY COLUMN id VARCHAR(50),
MODIFY COLUMN name VARCHAR(15),
MODIFY COLUMN surname VARCHAR(15),
MODIFY COLUMN phone VARCHAR(50),
MODIFY COLUMN email VARCHAR(50),
MODIFY COLUMN birth_date VARCHAR(20),
MODIFY COLUMN country VARCHAR(15),
MODIFY COLUMN city VARCHAR(25),
MODIFY COLUMN postal_code VARCHAR(10),
MODIFY COLUMN address VARCHAR(50);
```

[Escriba aquí]

```
#combinó la tablas de users para tener solo una
• CREATE TABLE users_combined AS
 SELECT *, 'Canada' AS country_origin FROM users_ca
 UNION ALL
 SELECT *, 'United Kingdom' AS country_origin FROM users_uk
 UNION ALL
 SELECT *, 'United States' AS country_origin FROM users_usa;

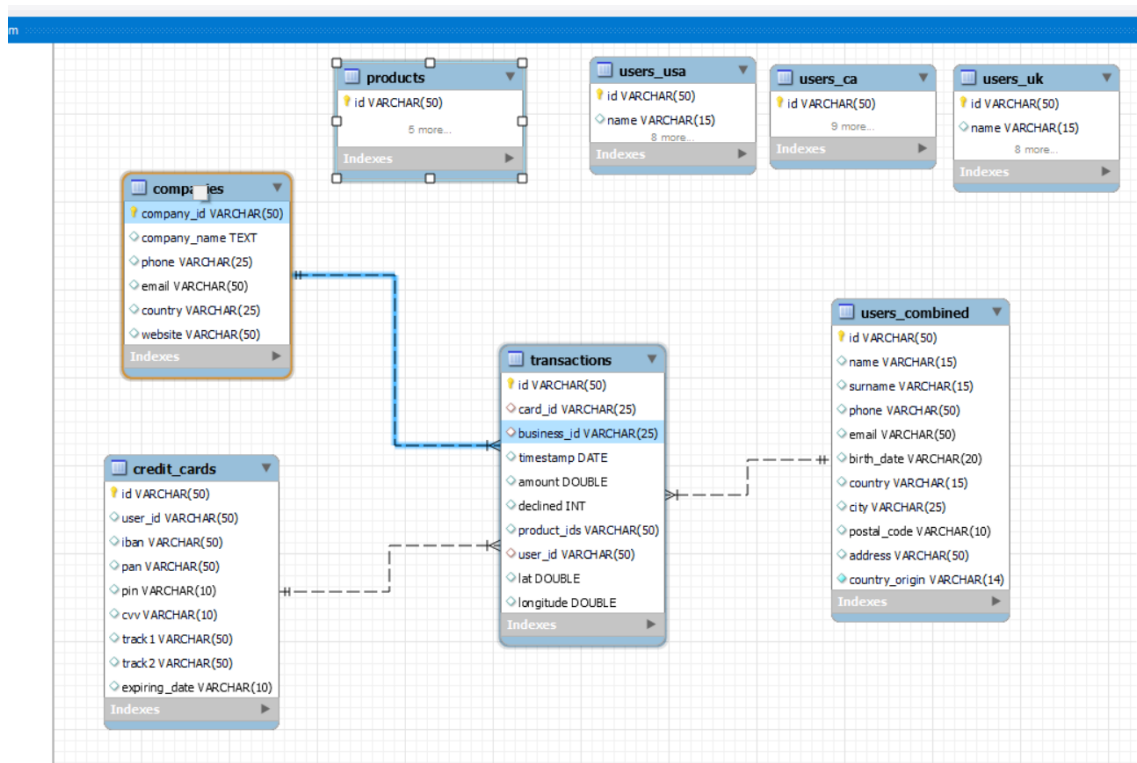
• ALTER TABLE users_combined
 ADD PRIMARY KEY (id);

FK:
```

AQUÍ ADALT HEM CREAT UNA NOVA TAULA DENOMINADA USERS\_COMBINED I EL QUE HEM FET ES COMBINAR TOTES LES TAULES DE USERS “CANADA”, “UNITED KINGDOM” I “UNITED STATES”, per poder relacionarla directament amb la taula de hechos “transaction”, quedan una relació tal que així:

Mirem com han quedat les relacions de les taules:

Ha quedat una relació d'estrella quedan 4 taules relacionades entre si amb una cardinalitat de muchoa a uno des de la taula de hechos que es la de transaction.



[Escriba aquí]

## Exercici 1

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

```
129
130 #Exercici 1
131 #Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.
132
133 • SELECT u.id, u.name, u.email
134 FROM users_combined u
135 WHERE 30 < (
136 SELECT COUNT(*)
137 FROM transactions t
138 WHERE t.user_id = u.id
139);
140
141
...
```

Result Grid

|   | id   | name   | email                           |
|---|------|--------|---------------------------------|
| ▶ | 267  | Ocean  | aenean@yahoo.com                |
|   | 272  | Hedwig | sem.eget@icloud.edu             |
|   | 275  | Kenyon | convallis.ante.lectus@yahoo.com |
|   | 92   | Lynn   | vitae.aliquet@outlook.edu       |
| • | NULL | NULL   | NULL                            |

users\_combined 67 x

Output

Action Output

| #   | Time     | Action                                                                                    | Message           |
|-----|----------|-------------------------------------------------------------------------------------------|-------------------|
| ✓ 1 | 12:43:58 | SELECT u.id, u.name, u.email FROM users_combined u WHERE 30 < ( SELECT COUNT(*) FROM t... | 4 row(s) returned |

Aquesta consulta SQL fa el següent:

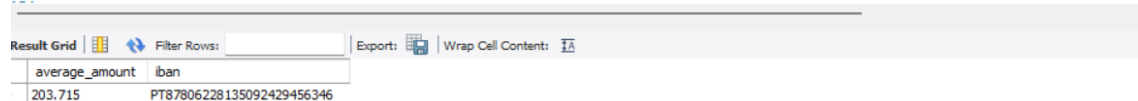
1. Selecciona l'id, nom i email dels usuaris de la taula "users\_combined".
2. Utilitza una subconsulta per comptar les transaccions de cada usuari a la taula "transactions".
3. Compara el nombre de transaccions amb 30 utilitzant la clàusula WHERE.
4. Retorna només els usuaris que tenen més de 30 transaccions.

[Escriba aquí]

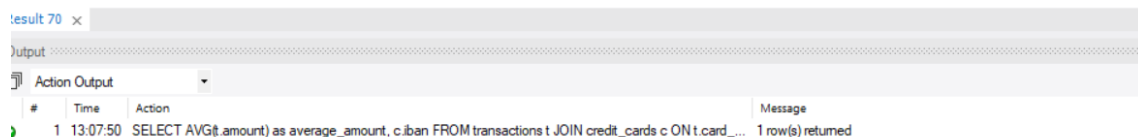
## - Exercici 2

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

```
141
142 #- Exercici 2
143 #Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.
144
145 • SELECT AVG(t.amount) as average_amount, c.iban
146 FROM transactions t
147 JOIN credit_cards c ON t.card_id = c.id
148 join companies co on t.business_id = co.company_id
149 WHERE co.company_name = 'Donec Ltd'
150 GROUP BY c.iban;
```



| average_amount | iban                      |
|----------------|---------------------------|
| 203.715        | PT87806228135092429456346 |



| average_amount | iban                      |
|----------------|---------------------------|
| 203.715        | PT87806228135092429456346 |

Aquesta consulta SQL calcula la mitjana de l'import (amount) de les transaccions per IBAN de targetes de crèdit associades a l'empresa "Donec Ltd".

Utilitza tres taules: transactions, credit\_cards i companies.

La consulta uneix aquestes taules, filtra les transaccions de "Donec Ltd", agrupa els resultats per IBAN i calcula la mitjana de l'import per a cada grup.

El resultat mostra l'IBAN de cada targeta i la seva mitjana d'import de transacció corresponent.

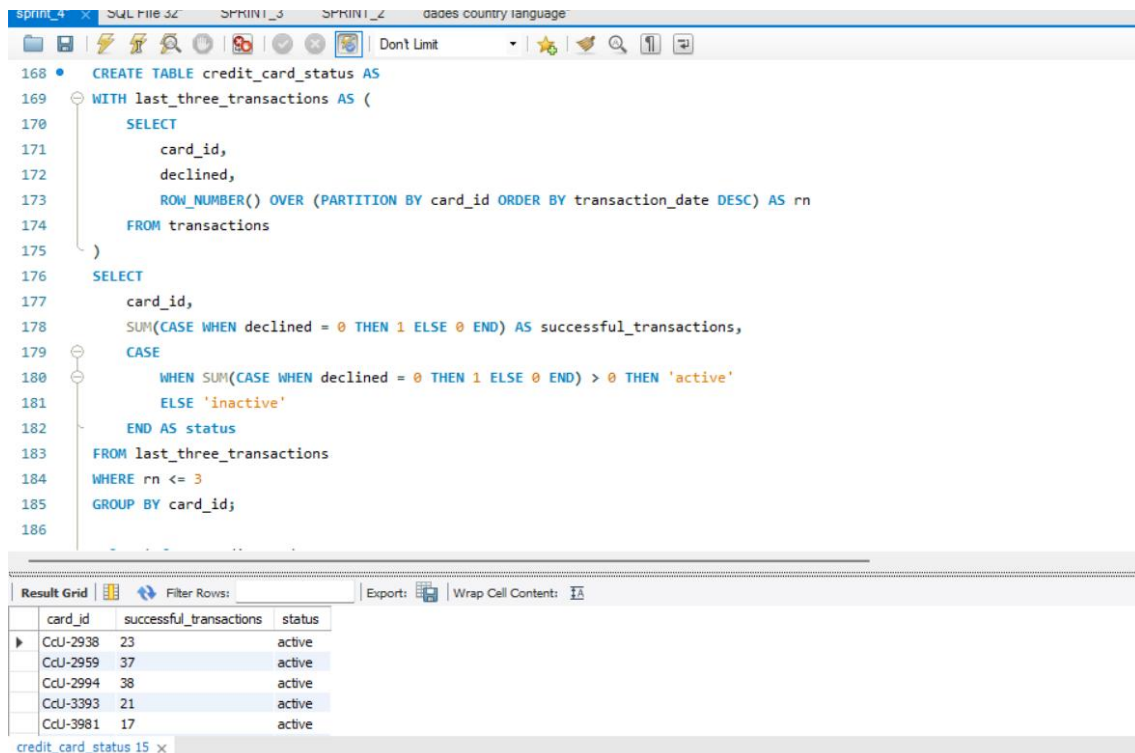
[Escriba aquí]

## 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:

## Exercici 1

Quantes targetes estan actives?



The screenshot shows a SQL IDE with a query editor and a result grid. The query editor contains the following SQL code:

```
168 CREATE TABLE credit_card_status AS
169 WITH last_three_transactions AS (
170 SELECT
171 card_id,
172 declined,
173 ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY transaction_date DESC) AS rn
174 FROM transactions
175)
176 SELECT
177 card_id,
178 SUM(CASE WHEN declined = 0 THEN 1 ELSE 0 END) AS successful_transactions,
179 CASE
180 WHEN SUM(CASE WHEN declined = 0 THEN 1 ELSE 0 END) > 0 THEN 'active'
181 ELSE 'inactive'
182 END AS status
183 FROM last_three_transactions
184 WHERE rn <= 3
185 GROUP BY card_id;
```

The result grid shows the following data:

| card_id  | successful_transactions | status |
|----------|-------------------------|--------|
| CcU-2938 | 23                      | active |
| CcU-2959 | 37                      | active |
| CcU-2994 | 38                      | active |
| CcU-3393 | 21                      | active |
| CcU-3981 | 17                      | active |

## CREO UNA NOVA TABLA: CREDIT\_CARD\_STATUS

He creat la taula credit\_card\_status seleccionant només les tres últimes transaccions de cada targeta utilitzant la funció ROW\_NUMBER() per ordenar-les per data.

Només s'han processat aquestes tres transaccions per comptar quantes van ser exitoses (declined = 0) amb una funció SUM(CASE ...)

.L'estat de la targeta es determina com 'active' si almenys una de les tres transaccions va ser exitosa, o 'inactive' si totes van ser declinades

Els resultats es van agrupar per card\_id per obtenir un registre únic per targeta. Això permet que l'estat reflecteixi exclusivament les tres últimes transaccions.