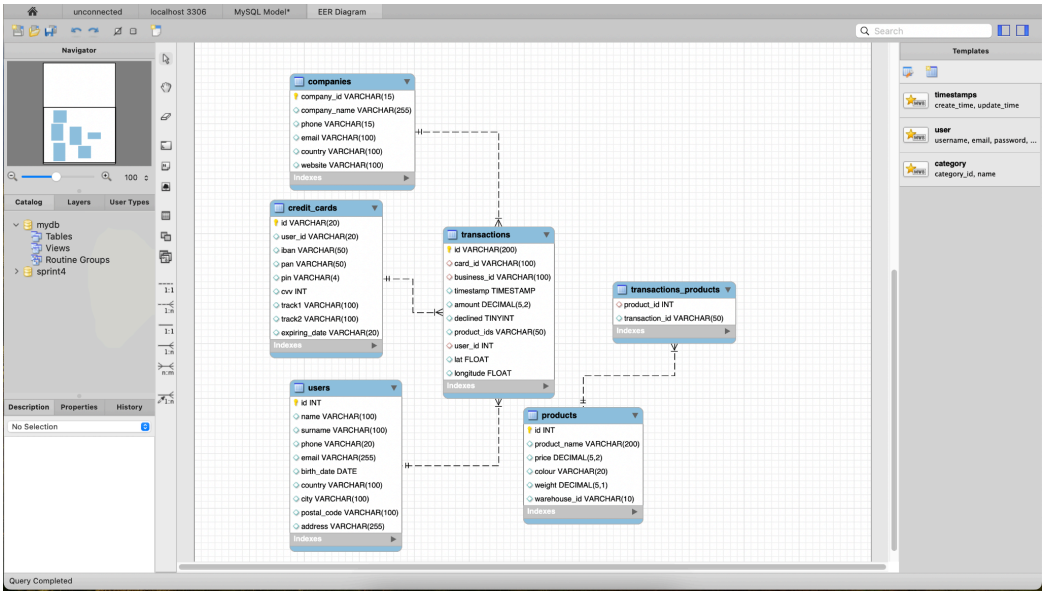
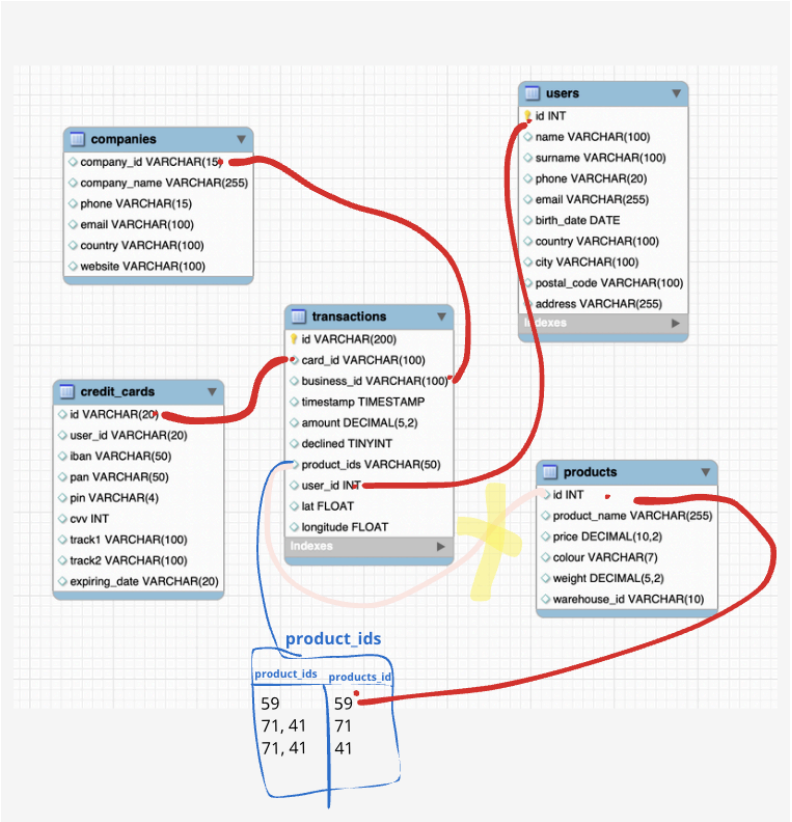
Nivell 1

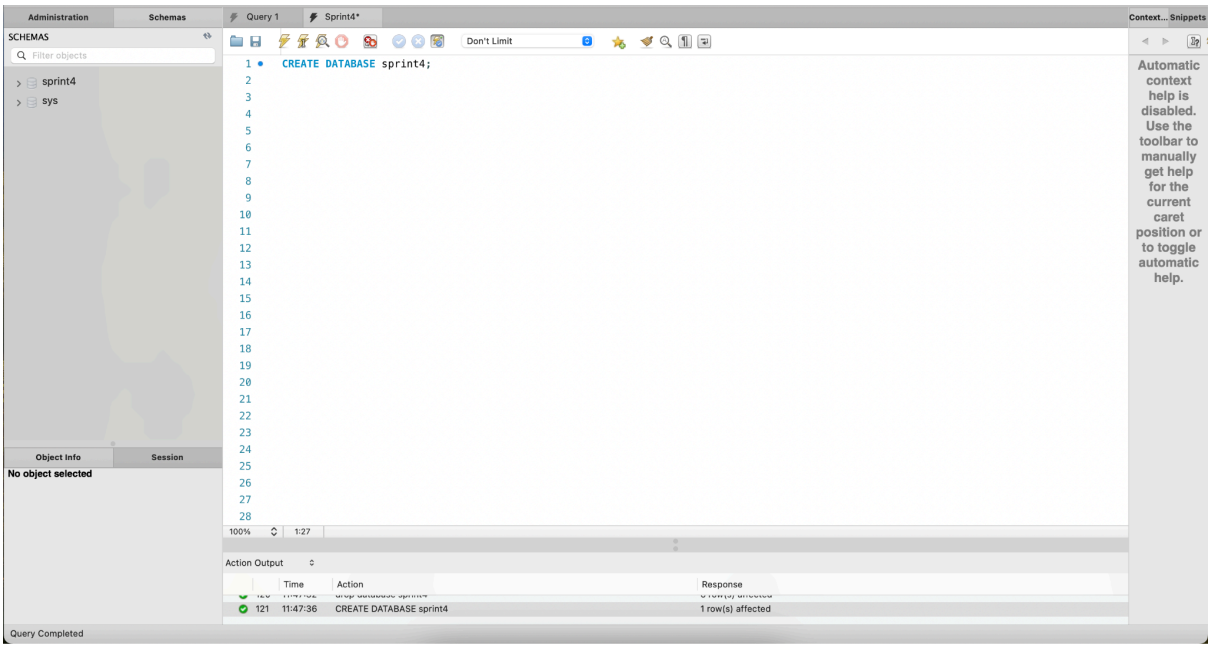
Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguis realitzar les següents consultes:
La base de datos creada va ser aquesta



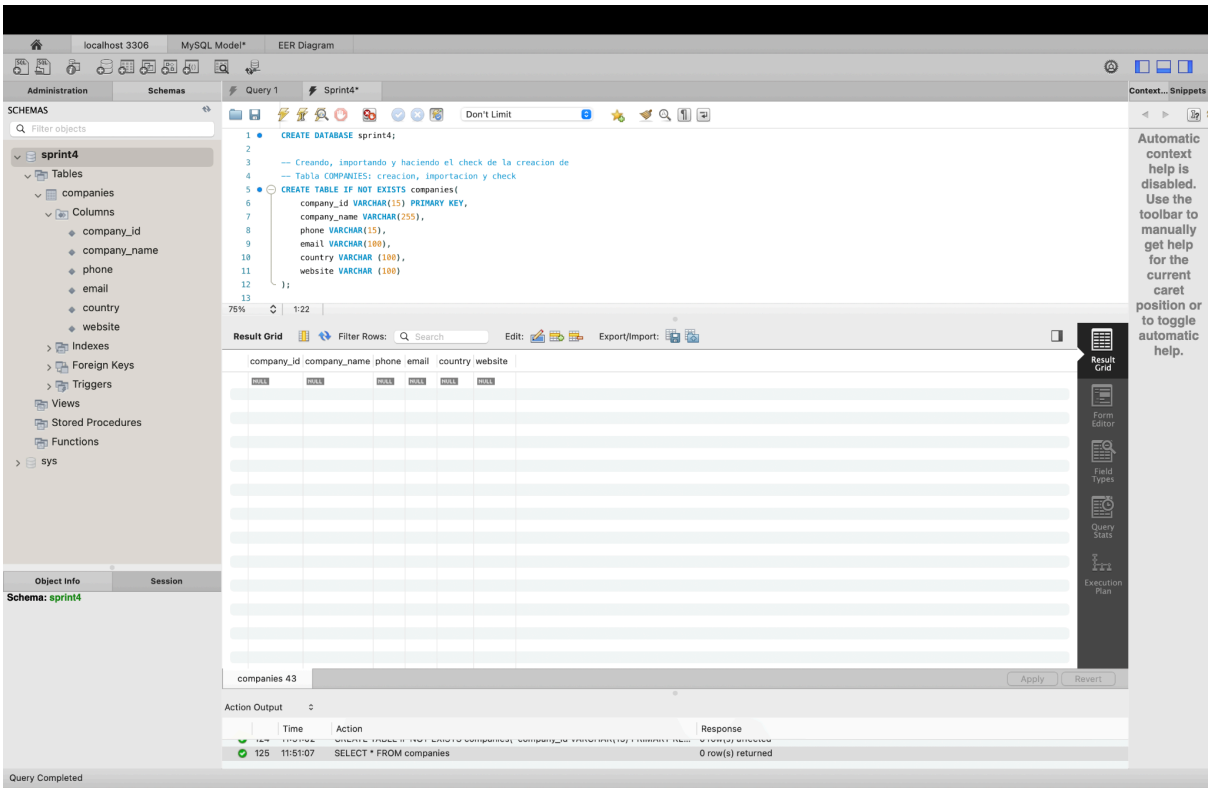
Para crearla he pensado en las relaciones de la tabla de hecho transactions con la de dimensiones, comprendiendo que no era posible crear una relación directa entre la de transactions con products y que hacia falta una transactions_products para relacionar las duas.



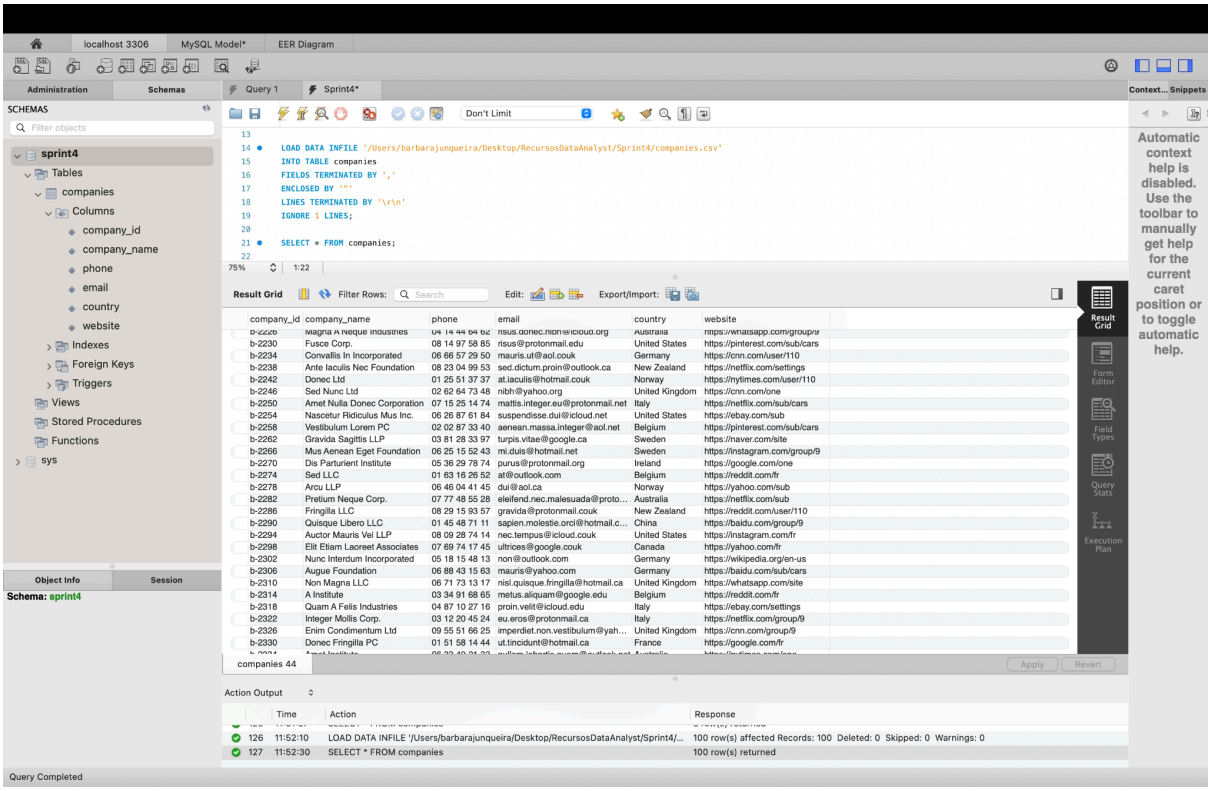
Primero he creado la base de datos:



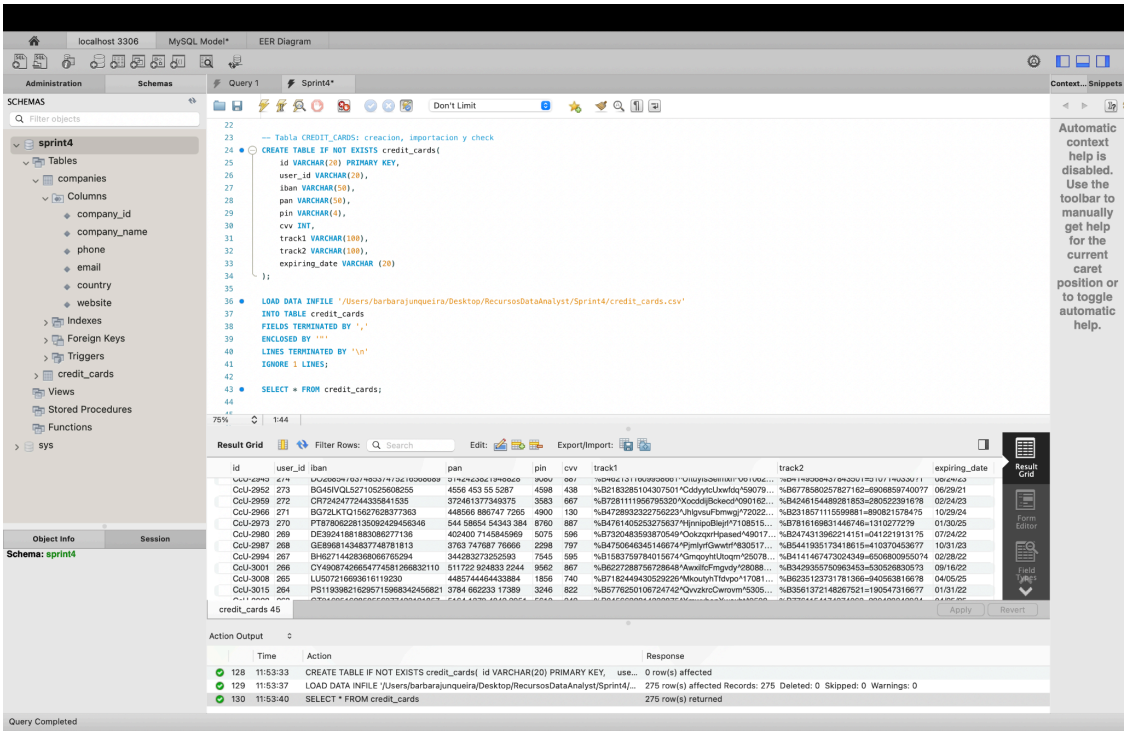
Luego he creado las tablas de dimensiones para luego relacionar con la tabla de hechos. Empeze con Companies:



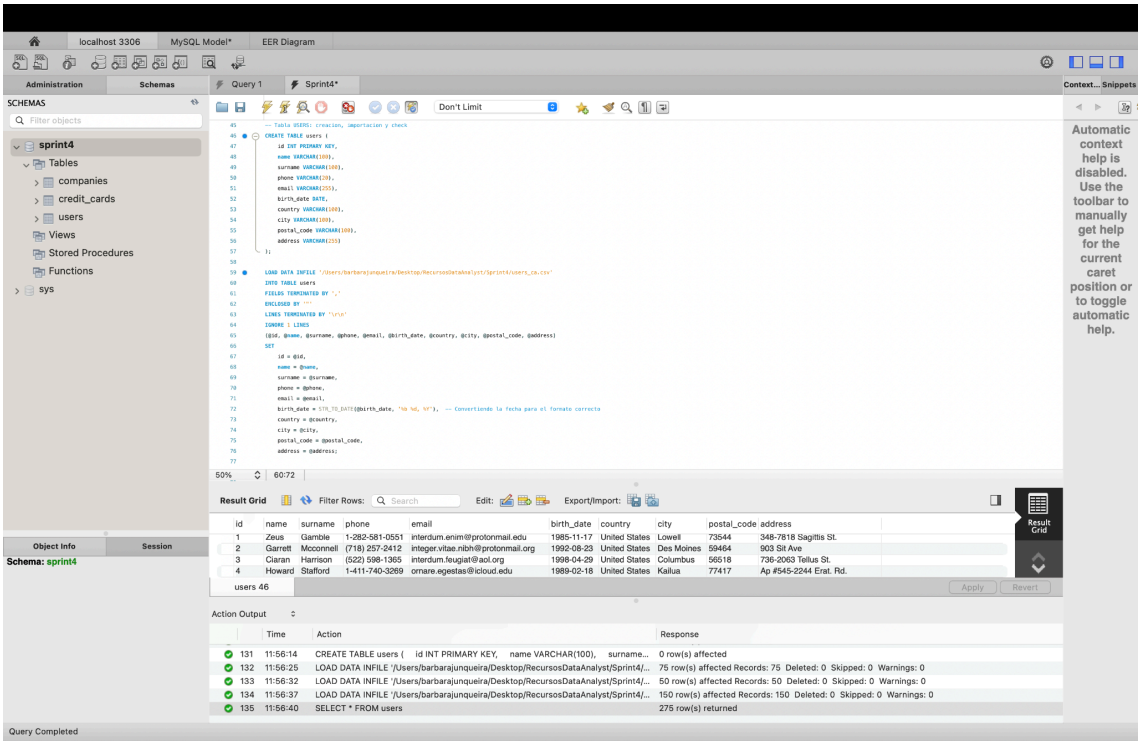
Y importado los datos de companies.csv via codigo en la tabla, haciendo el check visualizando los datos de la tabla:



Exactamente lo mismo con la tabla de Credit_Cards: crear tabla, importar data y conferir importación. Aqui tuve un problema con la terminación de lineas y tuve que sacar o '\r' de la LINES TERMINATED BY. Algunos archivos funcionaban con solo '\n' y otros necesitan ambos con '\r\n'. He pesquisado y entiendo que depende de donde se ha hecho la creación del archivo. Si el archivo ha sido creado en sistemas basados en Unix/Linux, incluidos macOS y otros sistemas derivados, se usa solo \n. Los archivos creados en sistemas Windows típicamente usan la secuencia \r\n para representar el final de una línea.



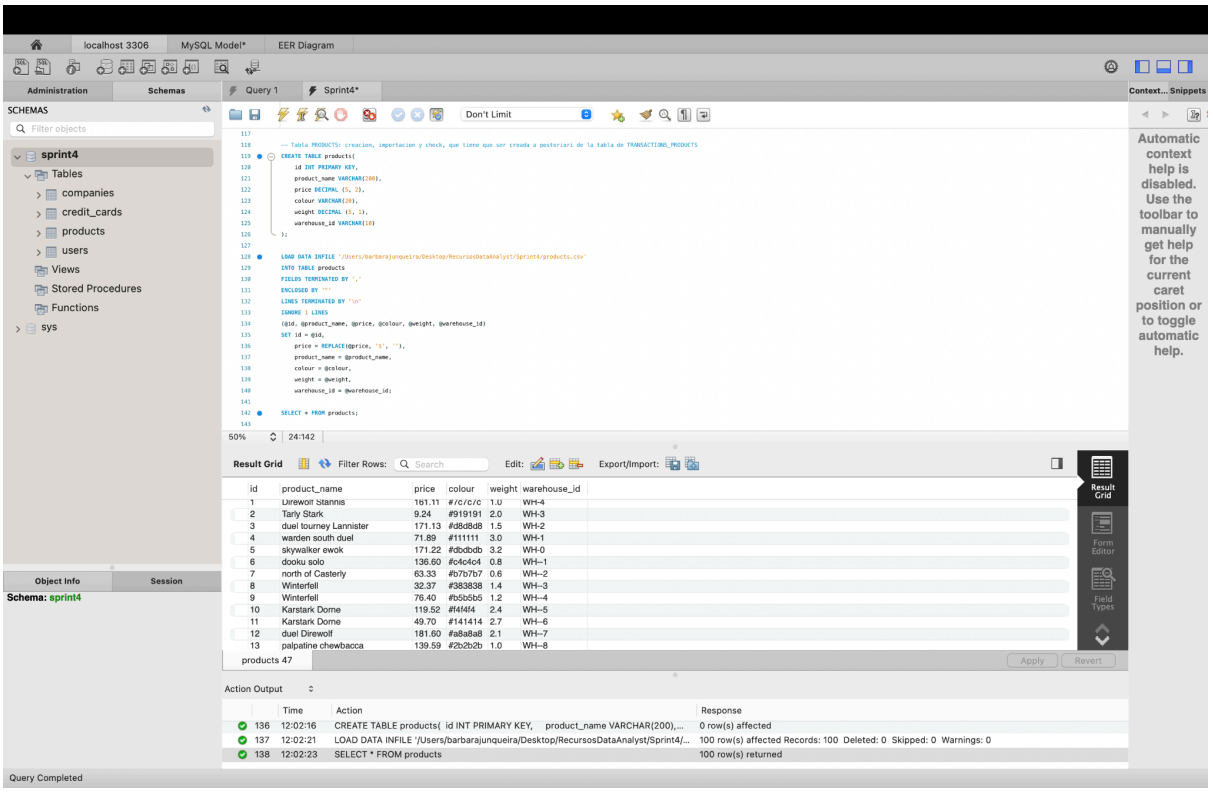
Luego la creacion de la tabla de users, que tuvo un aspecto especial porque quise insertar el dato de birthday como DATE y no como VARCHAR entonces he usado el SET para, al coger los datos, poder alterar la formatación de string to date de esa forma: birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y'), en este orden porque esta asi en el archivo para hacer la transformacion. También he juntado todos los 3 archivos de .csv de users de ca + usa + uk en una sola tabla de users.



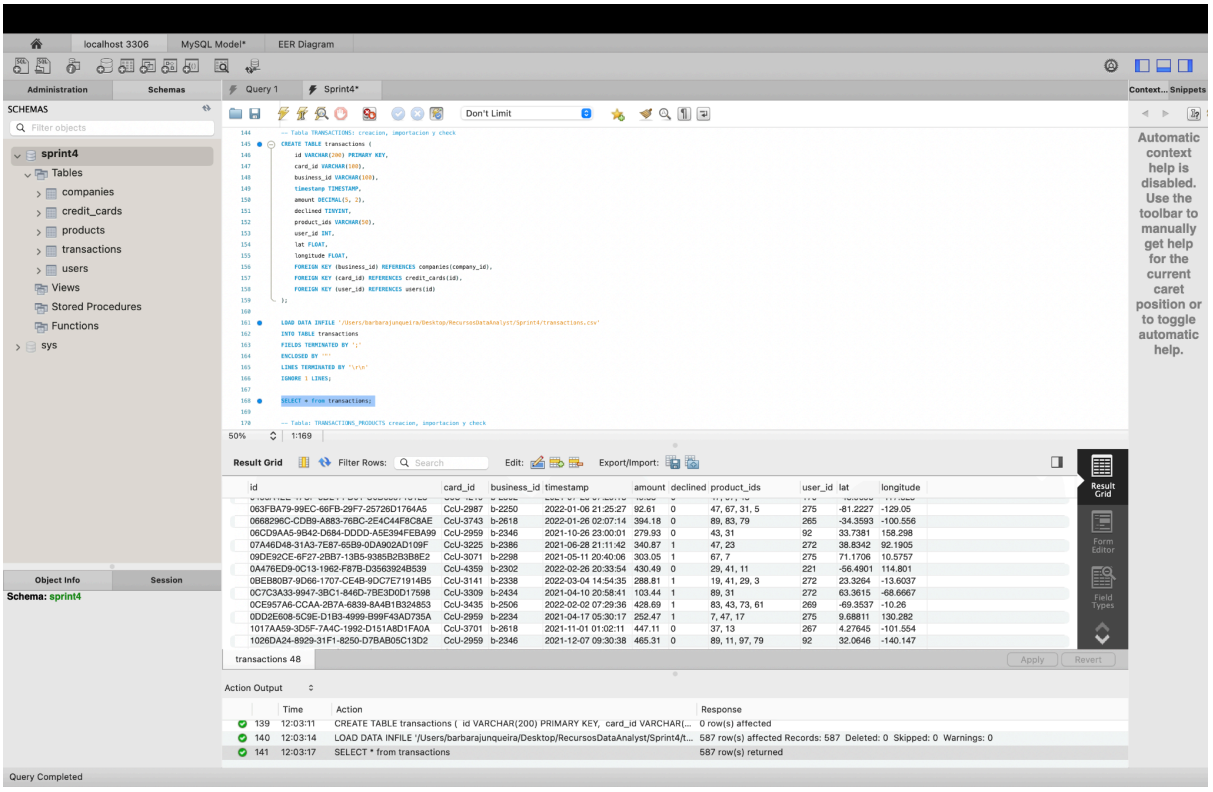
```
LOAD DATA INFILE '/Users/barbarajunqueira/Desktop/RecursosDataAnalyst/Sprint4/users_usa.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(@id, @name, @surname, @phone, @email, @birth_date, @country, @city, @postal_code, @address)
SET
    id = @id,
    name = @name,
    surname = @surname,
    phone = @phone,
    email = @email,
    birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y'),
    country = @country,
    city = @city,
    postal_code = @postal_code,
    address = @address;
```

```
LOAD DATA INFILE '/Users/barbarajunqueira/Desktop/RecursosDataAnalyst/Sprint4/users_uk.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(@id, @name, @surname, @phone, @email, @birth_date, @country, @city, @postal_code, @address)
SET
    id = @id,
    name = @name,
    surname = @surname,
    phone = @phone,
    email = @email,
    birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y'),
    country = @country,
    city = @city,
    postal_code = @postal_code,
    address = @address;
```

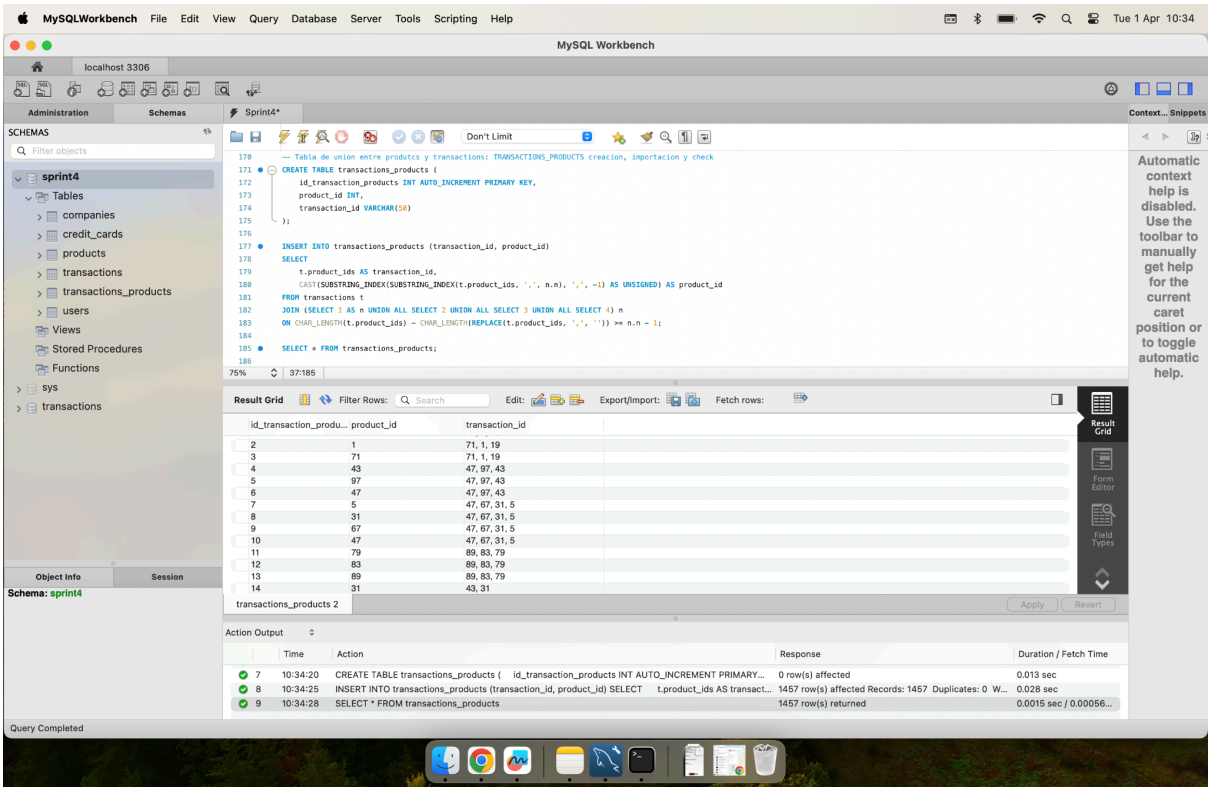
Luego la tabla de productos:



Luego la tabla de de transactions que tiene todas las relaciones con las tablas anteriores con excepción de products, setadas en su creación:

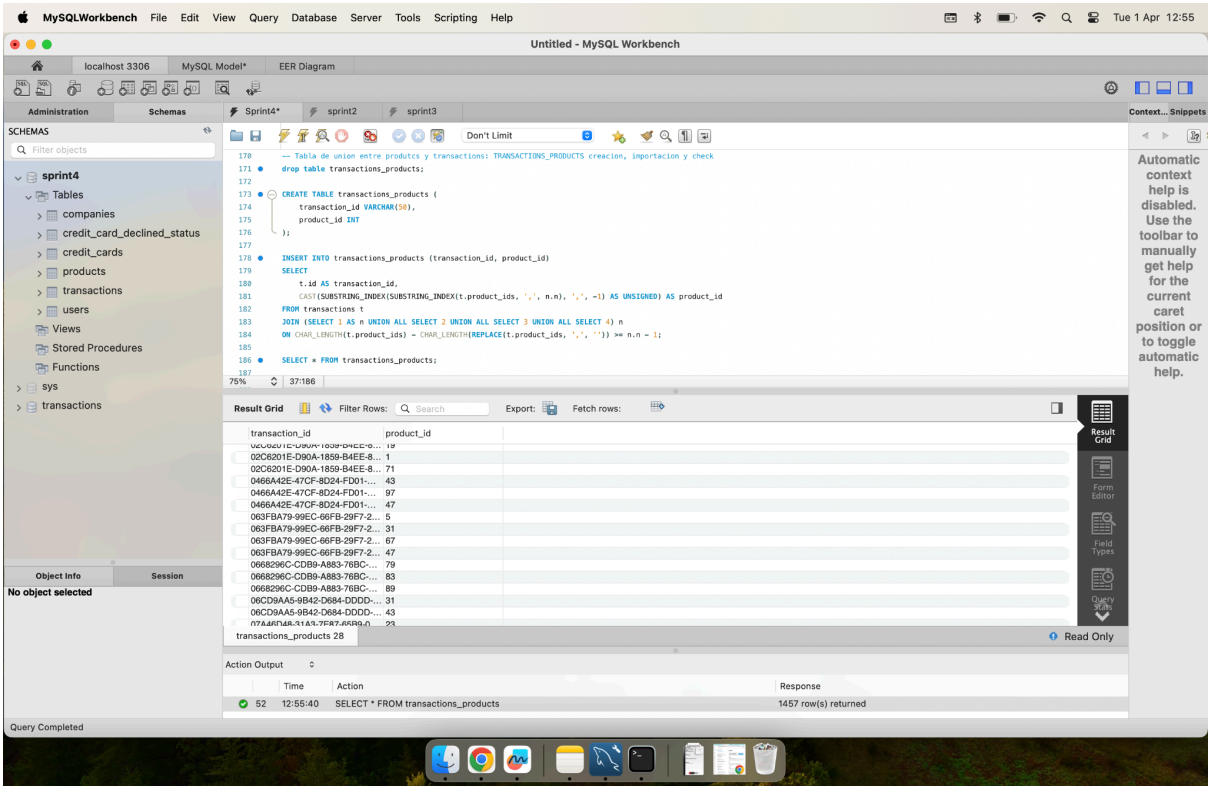


Pensando en establecer las conexiones de la tabla de hechos con la de dimensiones me he dado cuenta de que la tabla de transactions no podria unirse con la de productos porque aunque la de transactions tuviera los productos comprados, estaban estos en un “pack” que era el product_ids. entonces hice la creacion de una tabla llamada tansactions_products con el objetivo de separar los products_ids(transactions_id en la tabla en question) en product_id individual que se podría conectar con la tabla de productos. Como se trata de una tabla de unión entre la tabla de hechos y también la de productos, todas las relaciones en ambos lados se dan de muchos a muchos. por eso he creado un id para identificar de manera unica cada linea.



PD: al principio quise añadir una relación de many-to-many de transactions a transactions_products pero no he podido por el error 1822. como he leído que no tiene sentido crear esa relación en cuestión por esa misma cuestión de ser many-to-many dejé sin la relación como se puede ver en el pantallazo de diagrama.

ROSER M'HA EXPLICAT COM FER-HO MILLOR AMB L'ID DE TRANSACCIÓ I HO HE FET:

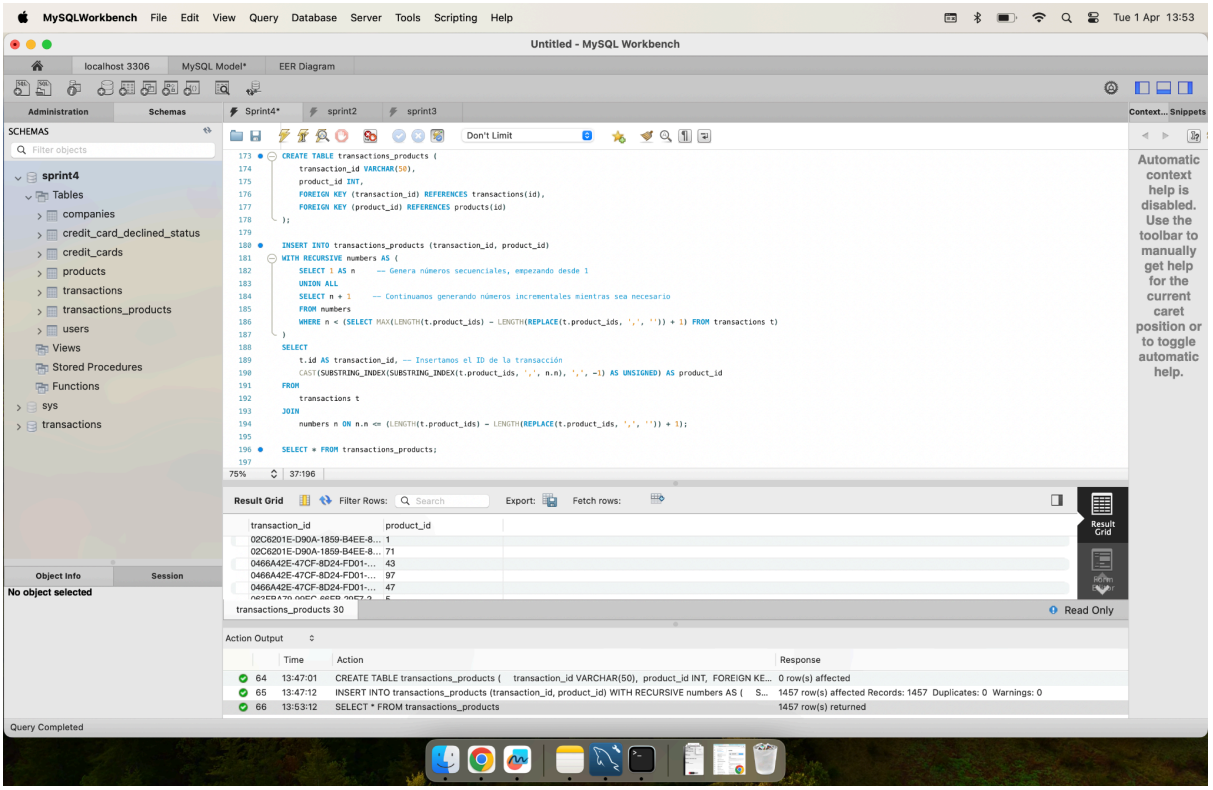


Compreendi que si yo tenia el product_ids estos se podrían repetir porque dos empresas/usuarios diferentes podrían hacer una compra igual. Pero el ID de la transacció de la tabla de transactions siempre seria único y esa es una buena manera de hacerlo mejor!
Luego quise mejorar la selección de elementos para "cuantos haya" y no solo 4 (porque pude hacer el check manual pero entendiendo que habrá archivos en los cuales no podré hacerlo quiero garantizar que todos los elementos de la columna product_ids entre dentro de la nueva tabla, entonces aplique una recursividad que lo que hace es coger la cantidad de elementos que tiene la columna y coletar los datos esa cantidad X de veces.
Explicacion
WHERE n < (SELECT MAX(LENGTH(t.product_ids) - LENGTH(REPLACE(t.product_ids, ',', '')) + 1) FROM transactions t)

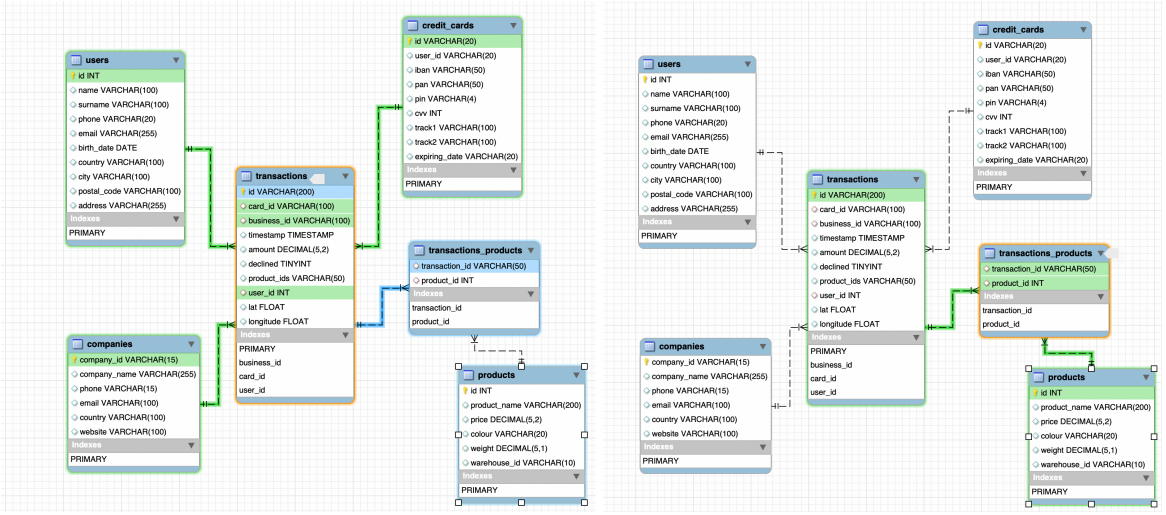
1. LENGTH(t.product_ids): Calcula la longitud total de la cadena product_ids. Por ejemplo, si tienes product_ids = '1,2,3,5,9,12', esto devolvería 11 (porque hay 11 caracteres en total, incluyendo las comas).
2. REPLACE(t.product_ids, ',', ''): Elimina todas las comas en la cadena, dejando solo los números. Usando el mismo ejemplo de product_ids = '1,2,3,5,9,12', después de aplicar REPLACE, obtienes '1235912' (longitud de 7).
3. LENGTH(t.product_ids) - LENGTH(REPLACE(t.product_ids, ',', '')): Esto da como resultado el número de comas en la cadena. En nuestro ejemplo, hay 5 comas, por lo que el resultado de la resta es 5.
4. MAX es usado para calcular el número máximo de productos en una transacción.
5. + 1: Finalmente, sumamos 1, porque el número de elementos es una coma menos que el número de valores. En nuestro caso, como hay 5 comas, sumando 1 nos da 6, que es el número total de elementos separados por comas en la lista.
6. La condición WHERE n < (SELECT LENGTH(...) + 1) asegura que la recursión se detenga cuando el número n llegue a más de los valores que hay en la lista de product_ids. Si la cadena tiene 6 elementos, solo se generarán números hasta el 6, y no más.

CAST(SUBSTRING_INDEX(SUBSTRING_INDEX(t.product_ids, ',', n.n), ',', -1) AS UNSIGNED) AS product_id

La columna de product_id será rellenada por la columna de product_ids pero separando cada una que este separada por virgula hasta el enésimo valor. El -1 es para coger la última entrada de valor (vuelve la casilla anterior). El CAST convierte la subcadena extraída en un número entero sin signo (UNSIGNED). Si fueran números negativos usaría el SIGNED para mantener la señal



Entonces ahora tiene sentido hacer la relación de
1 a muchos de transactions.id a transactions_products.transaction_id y también
muchos a 1 de transactions_products.product_id a products.id 😃😃😃



- Exercici 1

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

ACTUALIZADO

```sql
-- Nivell 1 Exercici 1
-- Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.
SELECT u.name AS userName, u.surname AS userSurname, (SELECT SUM(t.amount) FROM transactions t WHERE t.user_id = u.id) AS totalAmount
FROM users u
WHERE u.id IN ( SELECT t.user_id
                FROM transactions t
                GROUP BY t.user_id
                HAVING COUNT(t.id) > 30
);
```

75% 18:204

Result Grid Filter Rows: Search Export:

userName	userSurname	totalAmou...
Lynn	Riddle	11451.57
Ocean	Nelson	13052.24
Hedwig	Gilbert	18351.30
Kenyon	Hartman	12011.56

Result 2

ction Output

	Time	Action	Response
11	10:25:24	SELECT u.name AS userName, u.s...	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.

```sql
-- Nivell 1 Exercici 2
-- Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.
-- pre consulta para entender si seria un avg alto o bajo, y se puede calcular manualmente. tendria que ser (364,61 + 42,82) /2 = 203,715
SELECT *
FROM credit_cards cc
JOIN transactions t ON t.card_id = cc.id
JOIN companies c ON t.business_id = c.company_id
WHERE c.company_name = "Donec Ltd";

-- querry en question:
SELECT avg(amount) as avgAmount, company_name as companyName, card_id as iban
FROM transactions t
JOIN companies c ON t.business_id = c.company_id
WHERE c.company_name = "Donec Ltd"
GROUP BY iban, companyName;
```

43:228

Result Grid Filter Rows: Search Export:

avgAmount	companyName	iban
203.715000	Donec Ltd	CcU-2973

Result Grid Form Editor

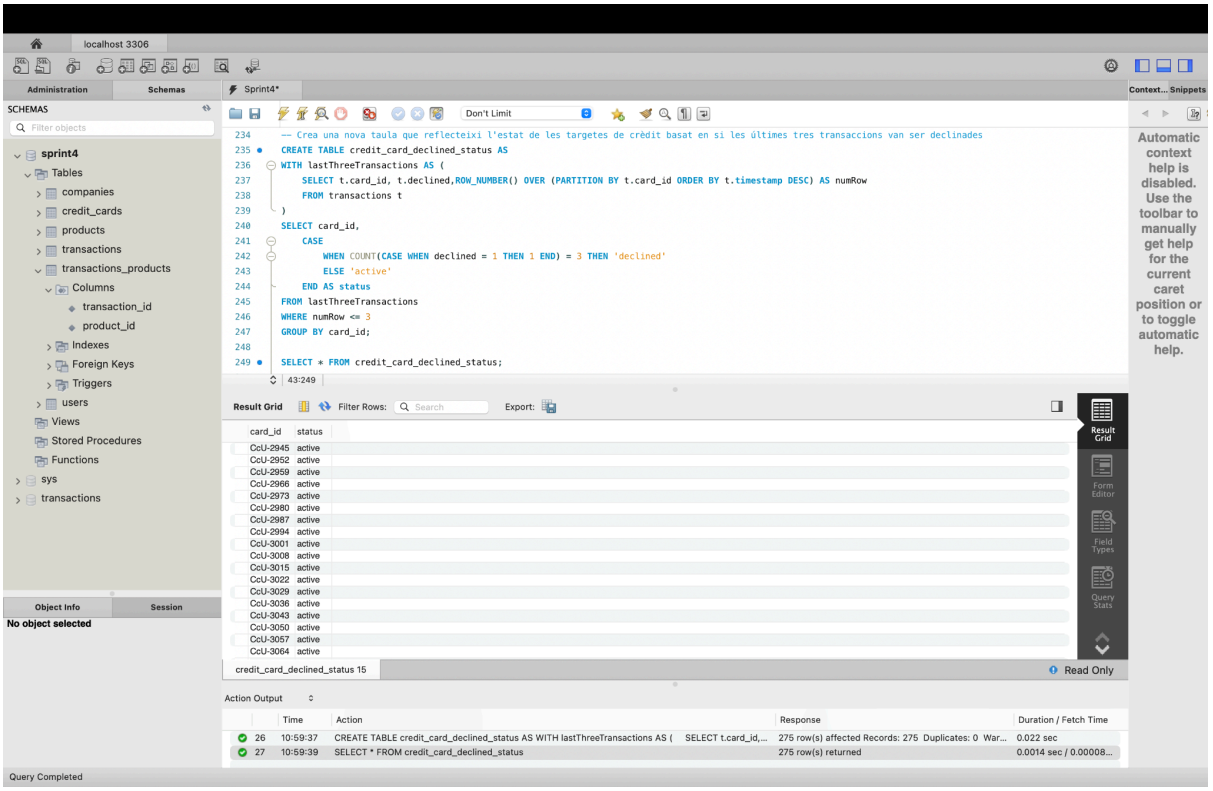
Result 12 Read Only

ction Output

	Time	Action	Response	Duration / Fetch Time
20	10:50:18	SELECT avg(amount) as avgAmount, company_name as companyName, card_id as iban FROM transac...	1 row(s) returned	0.0016 sec / 0.00002...

Nivell 2

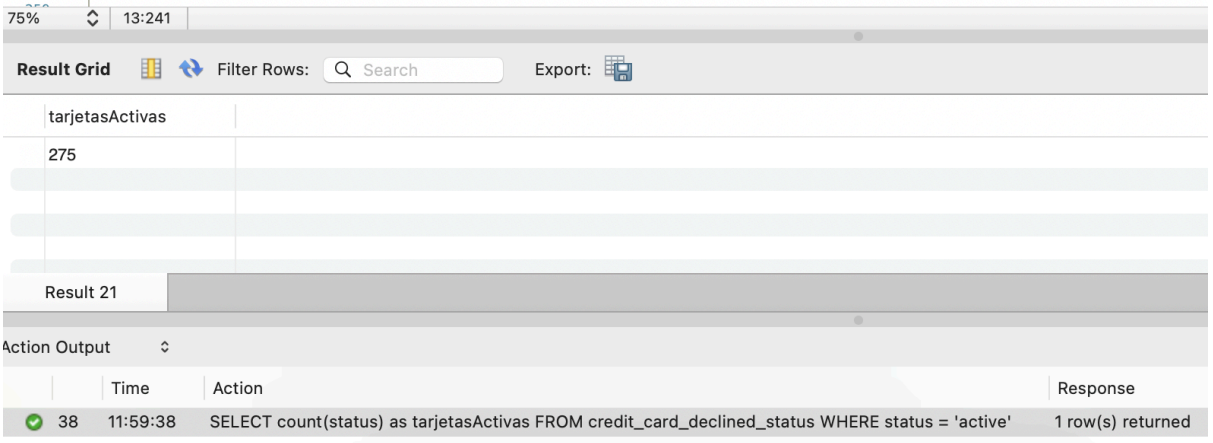
Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions van ser declinades i genera la següent consulta:



Exercici 1

Quantes targetes estan actives?

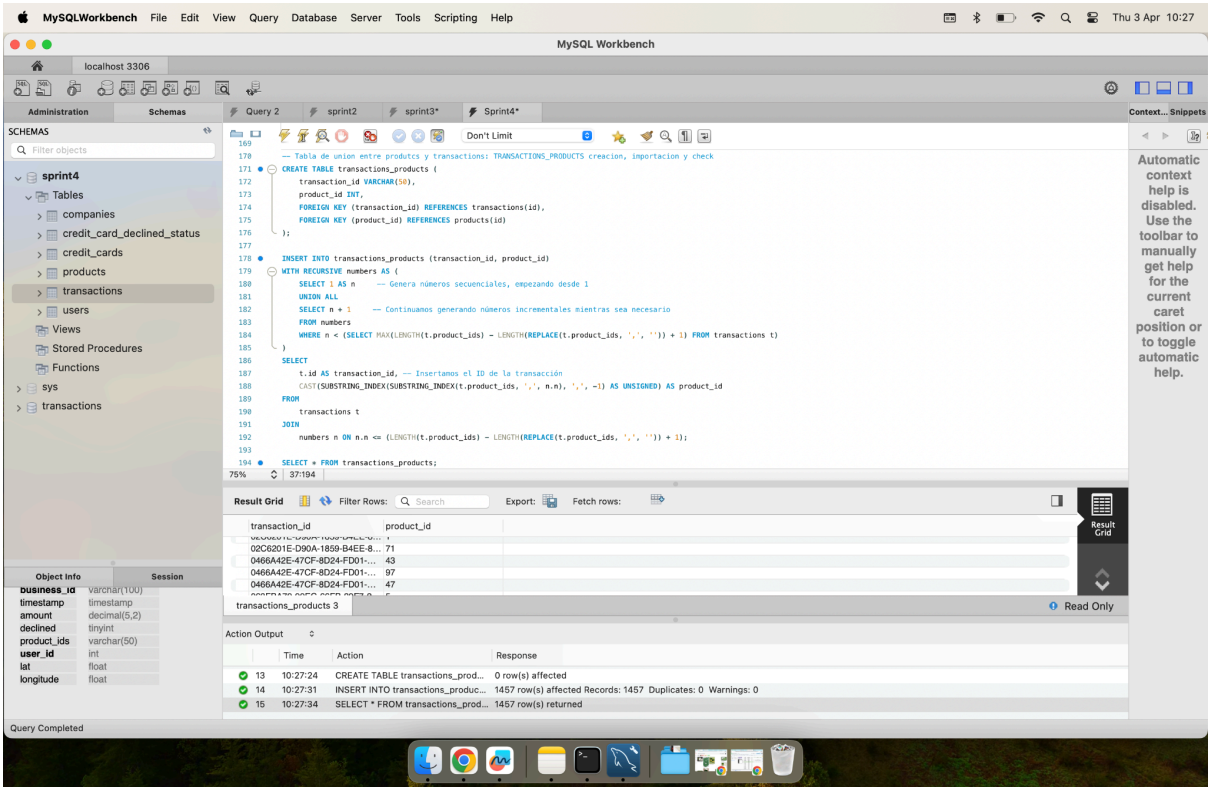
```
256 ● SELECT count(status) as tarjetasActivas
257   FROM credit_card_declined_status
258   WHERE status = 'active';
```



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:

La tabla ya habia sido creada a principio:



Exercici 1

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