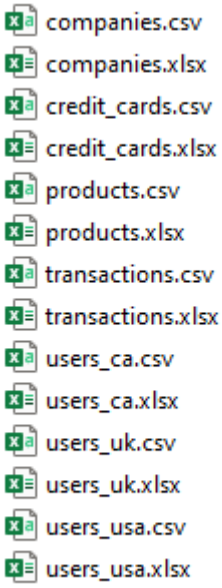
SPRINT 4

Nivel 1

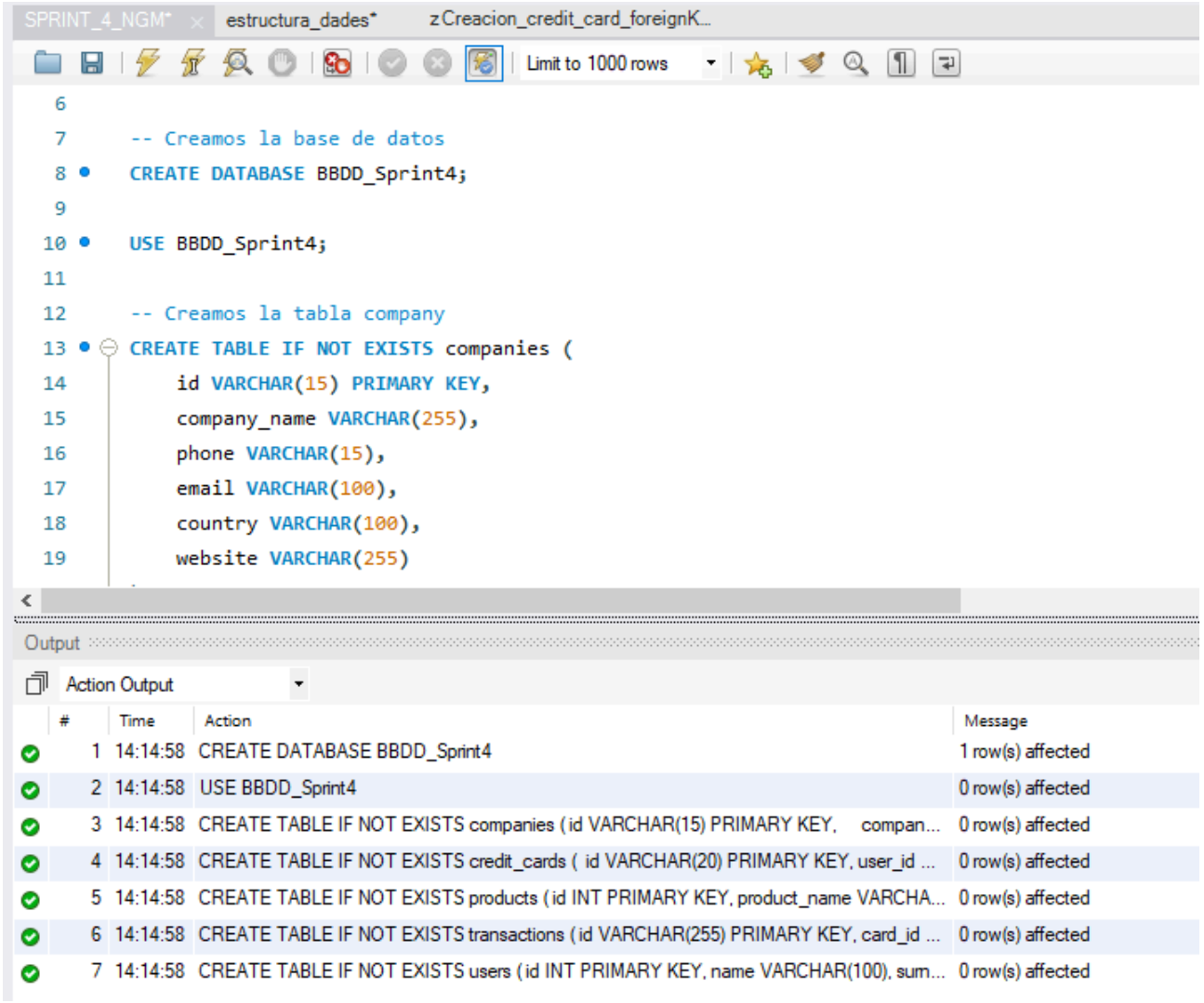
Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguis realitzar les següents consultes:

Los archivos proporcionados:

Procedí a abrirlos en excel para poder analizarlos y poder definir como organizar la base de datos y tablas dentro de esta. Algunos estaban separados con comas y otros por punto y coma, punto especial a tener en cuenta para cuando se carguen los datos del .csv.



Creamos la base de datos y las tablas:




```
-- Creamos la base de datos
CREATE DATABASE BBDD_Sprint4;

USE BBDD_Sprint4;

-- Creamos la tabla company
CREATE TABLE IF NOT EXISTS companies (
        id VARCHAR(15) PRIMARY KEY,
    company_name VARCHAR(255),
    phone VARCHAR(15),
    email VARCHAR(100),
    country VARCHAR(100),
    website VARCHAR(255)
);
-- Creamos la tabla credit_cards
CREATE TABLE IF NOT EXISTS credit_cards (
        id VARCHAR(20) PRIMARY KEY,
        user_id VARCHAR(20),
        iban VARCHAR(255),
```

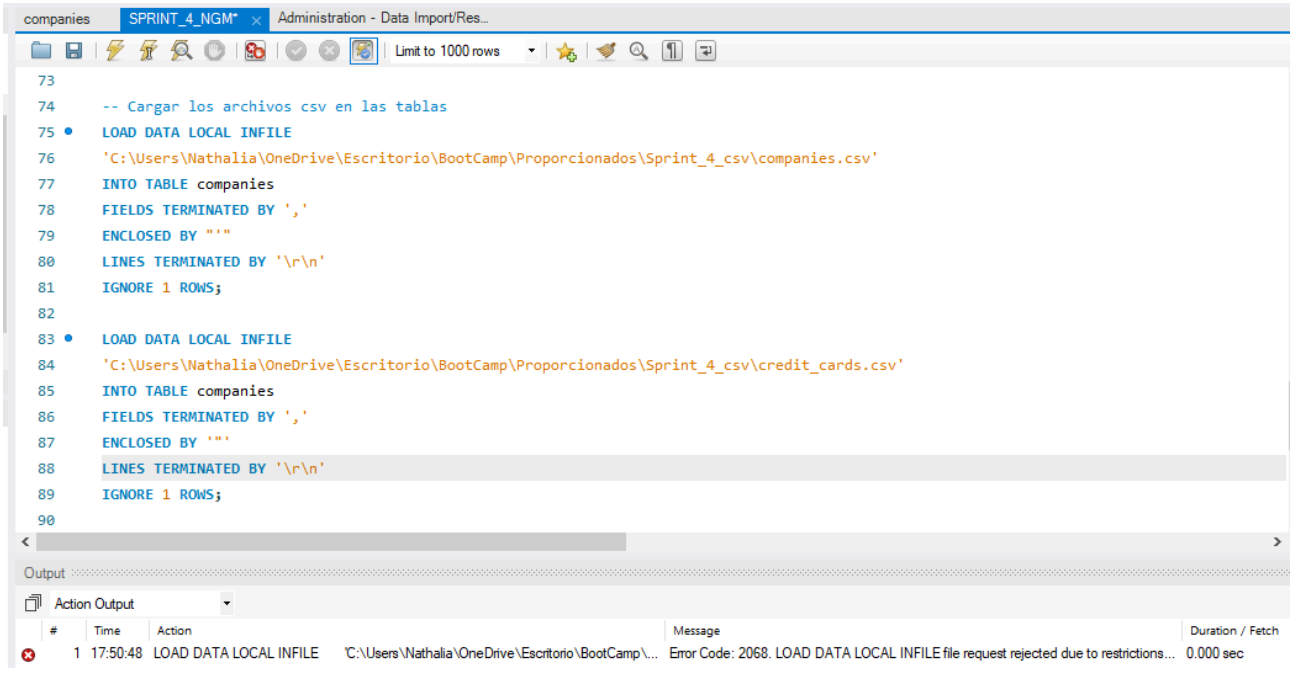
```
        pan VARCHAR(45),
        pin CHAR(4),
        cvv CHAR(3),
        track1 VARCHAR(255),
        track2 VARCHAR(255),
        expiring_date varchar(255)
);

-- Creamos la tabla products
CREATE TABLE IF NOT EXISTS products (
        id INT PRIMARY KEY,
        product_name VARCHAR(100),
        price VARCHAR(10),
        colour VARCHAR(100),
        weight VARCHAR(100),
        warehouse_id VARCHAR(100)
);

-- Creamos tabla transactions
CREATE TABLE IF NOT EXISTS transactions (
        id VARCHAR(255) PRIMARY KEY,
        card_id VARCHAR(20),
        bussiness_id VARCHAR(20),
        timestamp TIMESTAMP,
        amount DECIMAL(10,2),
        declined BOOLEAN,
        product_ids VARCHAR(20),
        user_id INT,
        lat FLOAT,
        longitude FLOAT
);

-- Creamos la tabla users
CREATE TABLE IF NOT EXISTS users (
        id INT PRIMARY KEY,
        name VARCHAR(100),
        surname VARCHAR(100),
        phone VARCHAR(150),
        email VARCHAR(150),
        birth_date VARCHAR(100),
        country VARCHAR(150),
        city VARCHAR (150),
        postal_code VARCHAR(100),
        address VARCHAR(255)
);
```

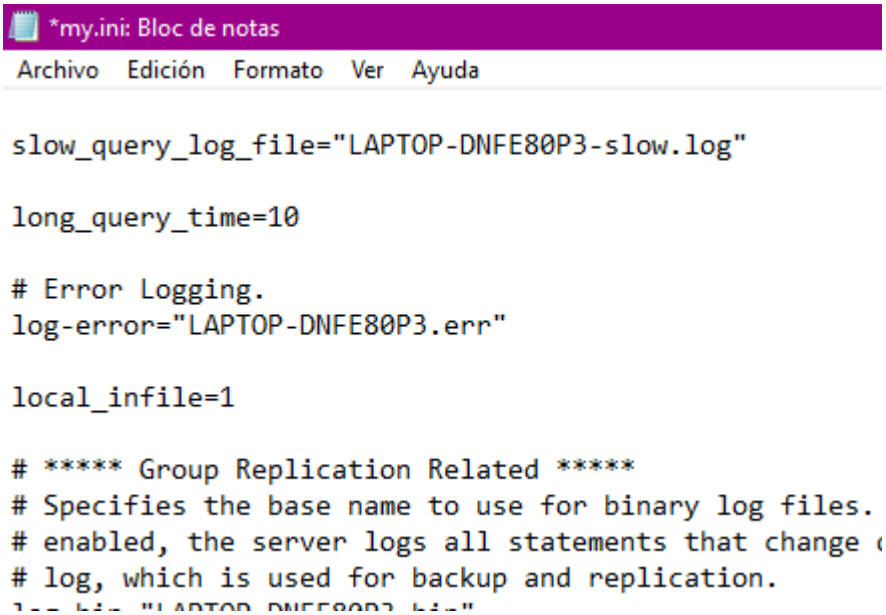
Cargamos los datos desde los archivos csv proporcionados. Usamos LOAD DATA LOCAL INFILE.



Al hacer una prueba con los datos de companies y credit_cards ha salido el siguiente error:

Error Code: 3948. Loading local data is disabled; this must be enabled on both the client and server sides.

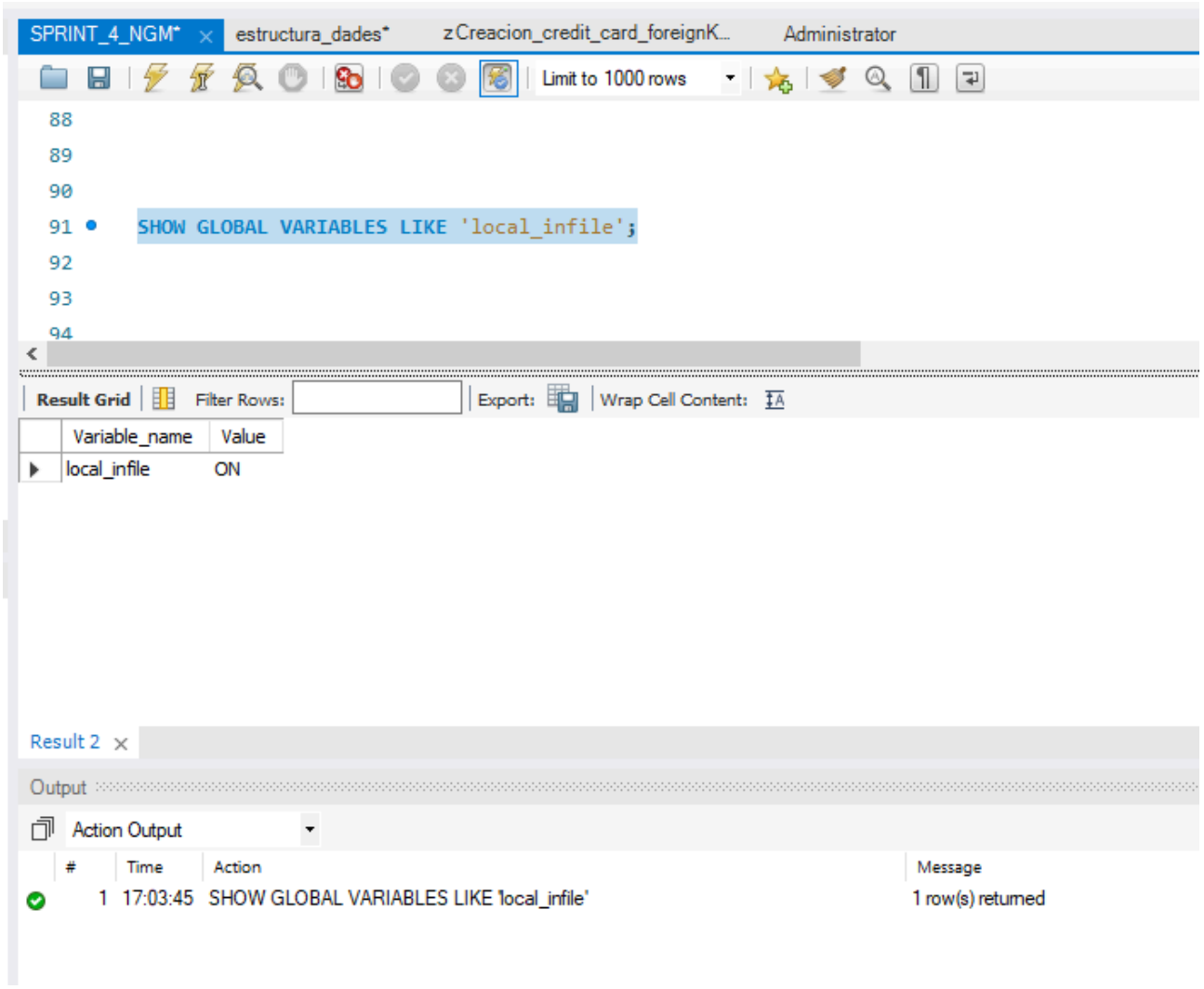
Procedo a habilitar la carga de archivos locales en el servidor.



Usar los siguientes comandos para parar e iniciar MYSQL Server:

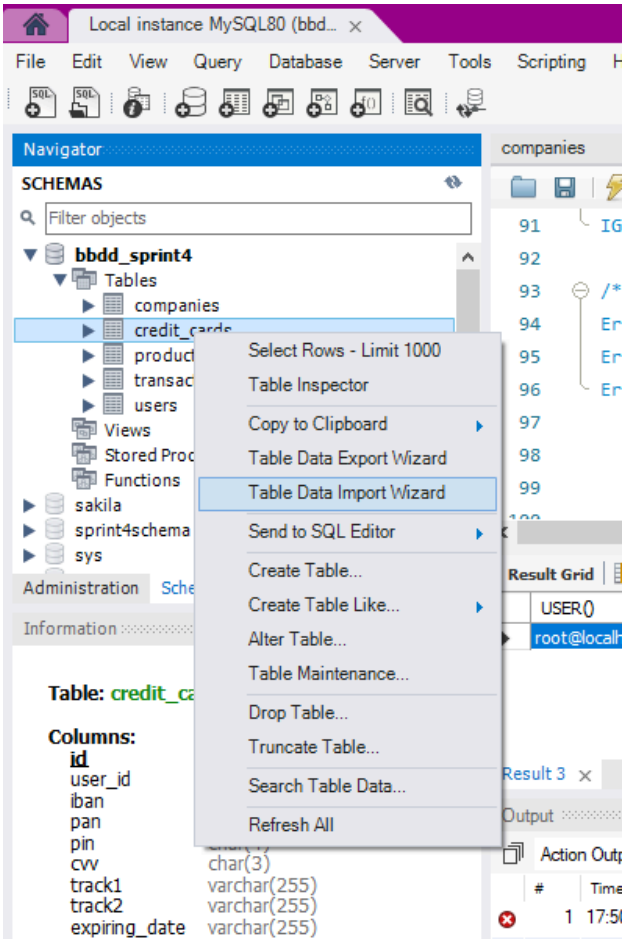
net stop MySQL80
net start MySQL80

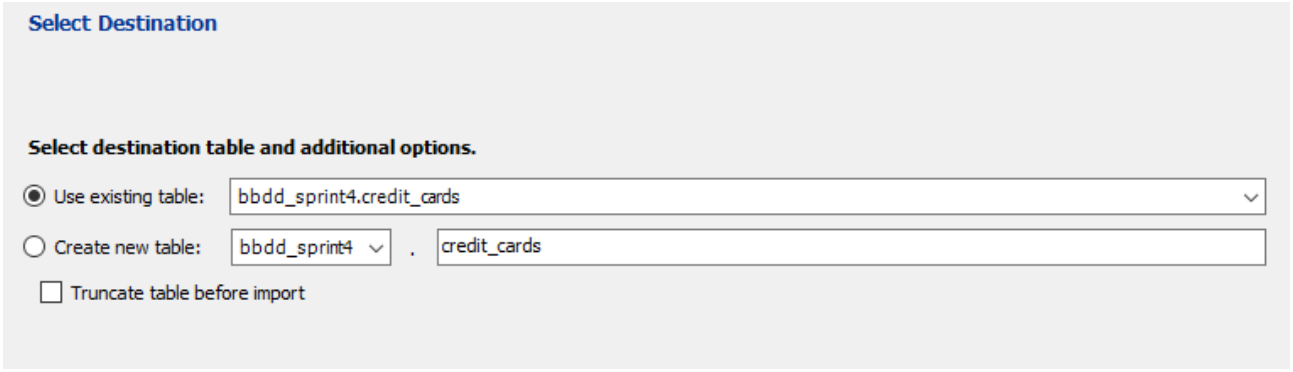
Verifica que local_infile esté habilitado:

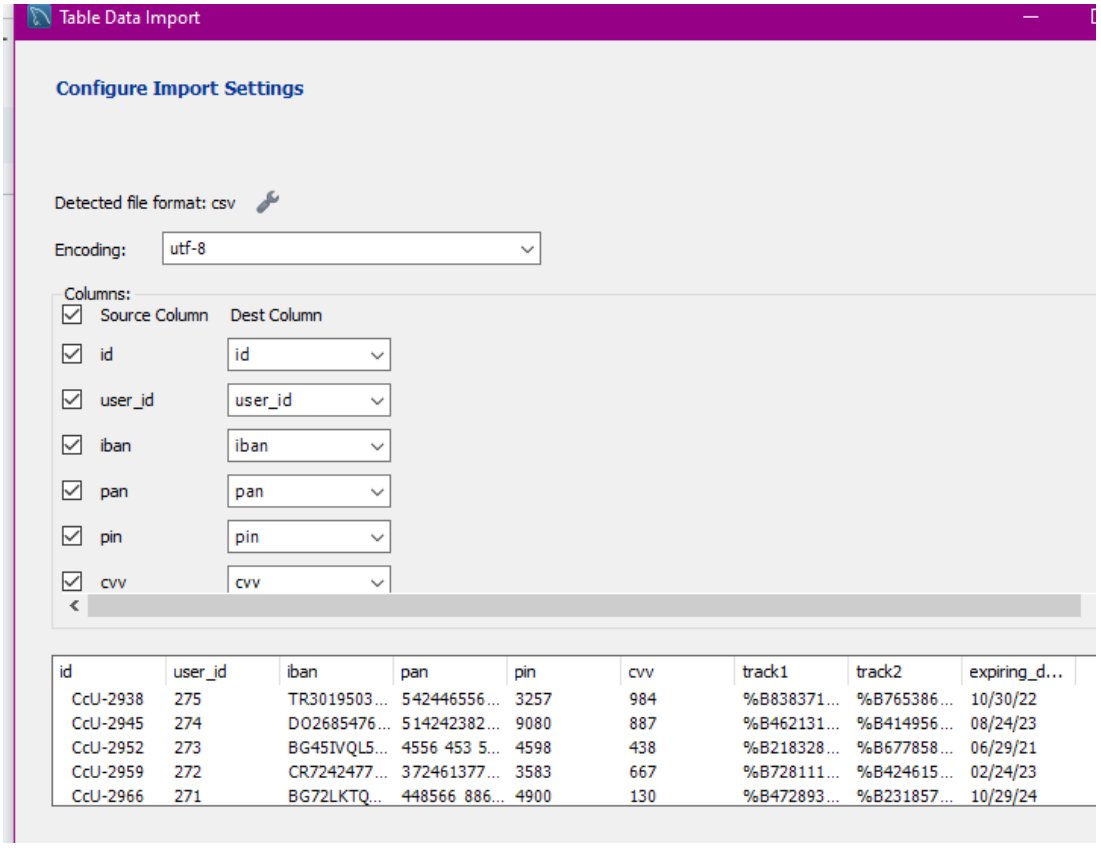


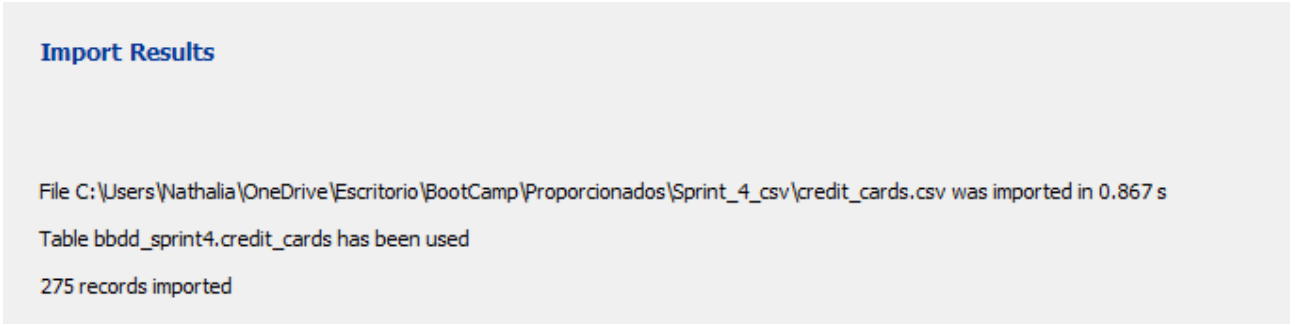
Igualmente no me deja importar:

Utilizo el “**Table Data Import Wizard**”

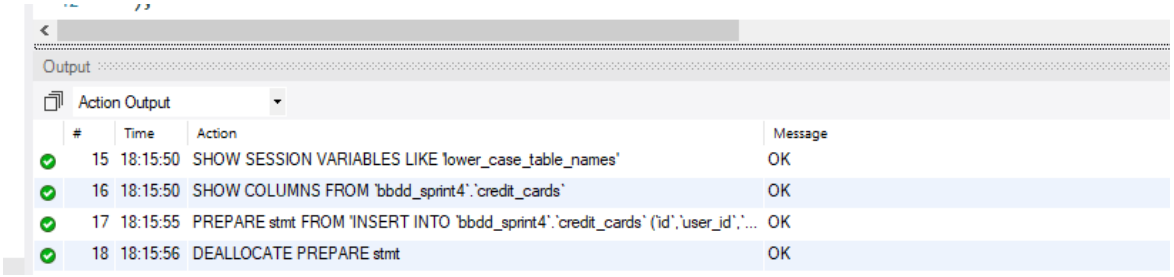

Local instance MySQL80 (bbd...
File Edit View Query Database Server Tools Scripting
Navigator
SCHEMAS
Filter objects
bbdd_sprint4
Tables
companies
credit_cards
users
Views
Functions
sakila
sprint4schema
sys
Administration
Information
Table: credit_ca
Columns:
id
user_id
iban
pan
pin
cvv char(3)
track1 varchar(255)
track2 varchar(255)
expiring_date varchar(255)
Select Rows - Limit 1000
Table Inspector
Copy to Clipboard
Table Data Export Wizard
Table Data Import Wizard
Send to SQL Editor
Create Table...
Create Table Like...
Alter Table...
Table Maintenance...
Drop Table...
Truncate Table...
Search Table Data...
Refresh All
companies
Result Grid
USER()
Result 3
Output
Action Outp


Select Destination
Select destination table and additional options.
Use existing table:
bbdd_sprint4.credit_cards
Create new table:
bbdd_sprint4
credit_cards
Truncate table before import





Aquí vemos el output de credit_cards:



Output

Action Output

#	Time	Action	Message
15	18:15:50	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
16	18:15:50	SHOW COLUMNS FROM `bbdd_sprint4`.`credit_cards`	OK
17	18:15:55	PREPARE stmt FROM 'INSERT INTO `bbdd_sprint4`.`credit_cards` (`id`,`user_id`,`...	OK
18	18:15:56	DEALLOCATE PREPARE stmt	OK

Cargamos la tabla products

24	18:27:45	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
25	18:27:45	SHOW DATABASES	OK
26	18:27:47	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
27	18:27:47	SHOW COLUMNS FROM `bbdd_sprint4`.`products`	OK
28	18:27:52	PREPARE stmt FROM 'INSERT INTO `bbdd_sprint4`.`products` (`id`,`product_na...	OK
29	18:27:52	DEALLOCATE PREPARE stmt	OK
30	18:28:01	SELECT * FROM bbdd_sprint4.products LIMIT 0, 1000	100 row(s) returned

Cargamos la tabla transactions

30	18:28:01	SELECT * FROM bbdd_sprint4.products LIMIT 0, 1000	100 row(s) returned
31	18:28:33	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
32	18:28:33	SHOW DATABASES	OK
33	18:28:34	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
34	18:28:34	SHOW COLUMNS FROM `bbdd_sprint4`.`transactions`	OK
35	18:28:39	PREPARE stmt FROM 'INSERT INTO `bbdd_sprint4`.`transactions` (`id`,`card_id`,`...	OK
36	18:28:41	DEALLOCATE PREPARE stmt	OK
37	18:28:48	SELECT * FROM bbdd_sprint4.transactions LIMIT 0, 1000	587 row(s) returned

Cargamos la tabla Users _ca

44	18:31:38	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
45	18:31:38	SHOW DATABASES	OK
46	18:31:41	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
47	18:31:41	SHOW COLUMNS FROM `bbdd_sprint4`.`users`	OK
48	18:32:04	PREPARE stmt FROM 'INSERT INTO `bbdd_sprint4`.`users` (`id`,`name`,`surname...	OK
49	18:32:04	DEALLOCATE PREPARE stmt	OK
50	18:32:19	SELECT * FROM bbdd_sprint4.users LIMIT 0, 1000	75 row(s) returned

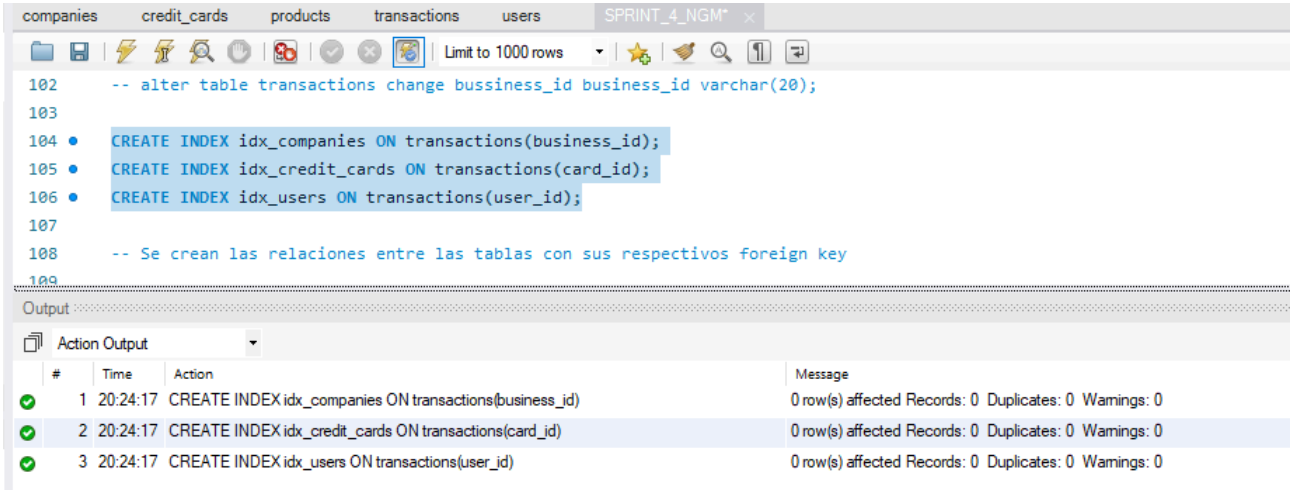
Aquí se ven 50 registros más de Users _uk

55	18:33:03	PREPARE stmt FROM 'INSERT INTO `bbdd_sprint4`.`users` (`id`,`name`,`surname...	OK
56	18:33:03	DEALLOCATE PREPARE stmt	OK
57	18:33:16	SELECT * FROM bbdd_sprint4.users LIMIT 0, 1000	125 row(s) returned

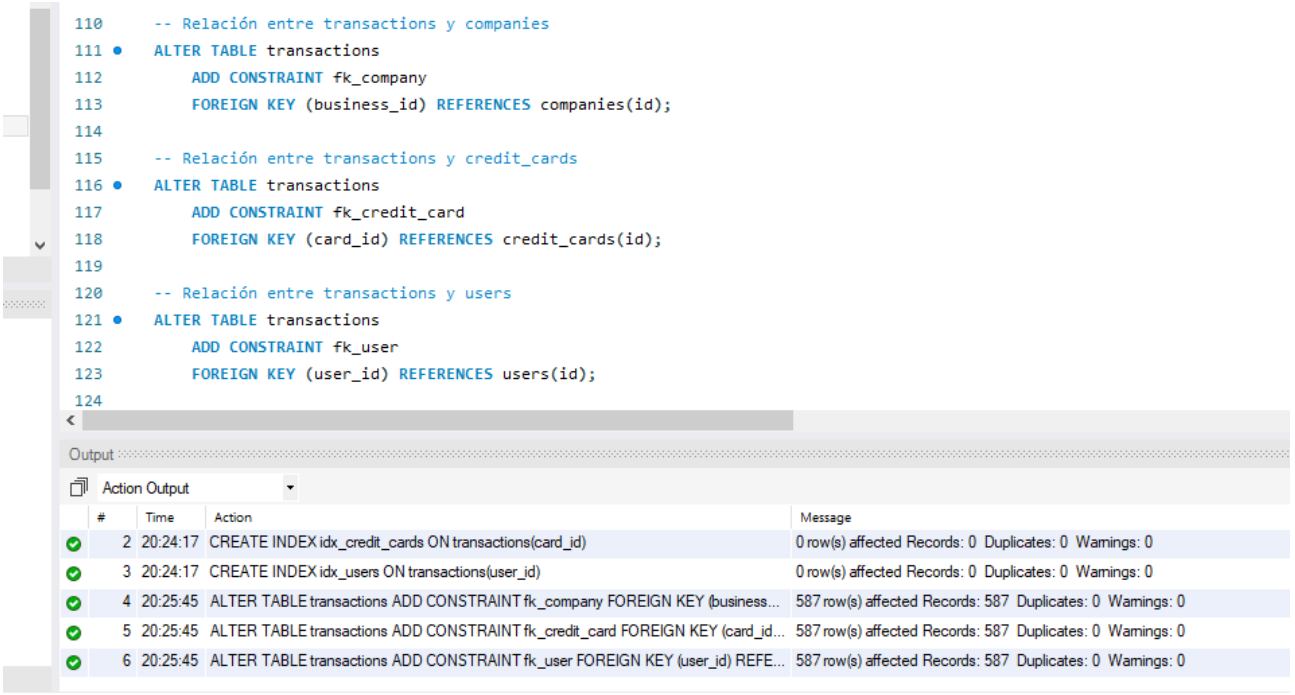
Aquí se ven 150 registros más de Users _usa

62	18:35:40	PREPARE stmt FROM 'INSERT INTO `bbdd_sprint4`.`users` (`id`,`name`,`surname...	OK
63	18:35:41	DEALLOCATE PREPARE stmt	OK
64	18:35:58	SELECT * FROM bbdd_sprint4.users LIMIT 0, 1000	275 row(s) returned

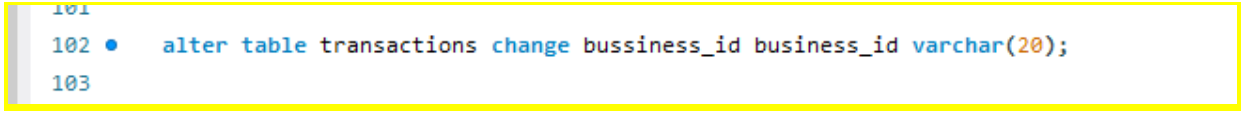
Se crean los índices para mejorar el rendimiento:



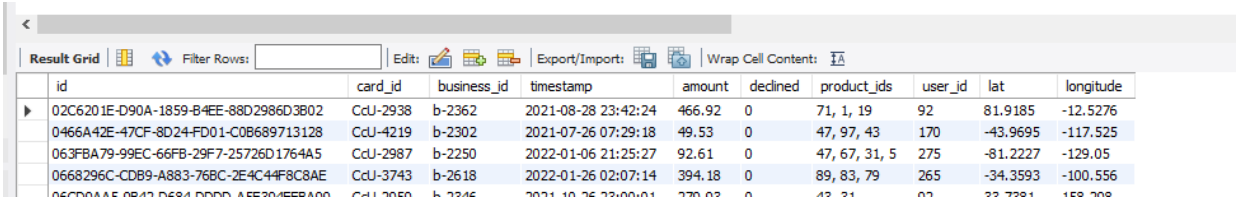
Y luego las Foreign Keys:



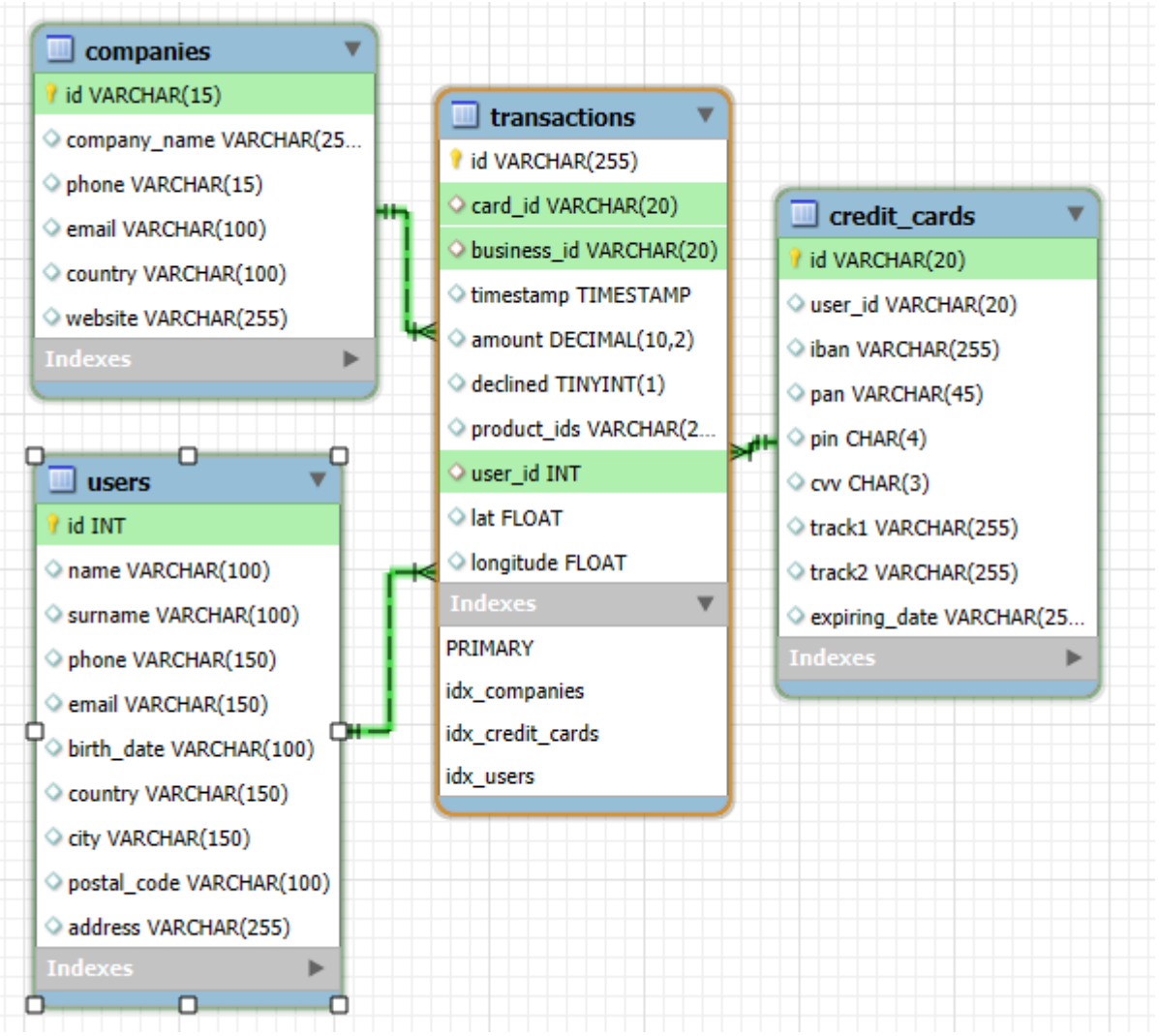
Anteriormente salió un error al cargar la relación con la FK en transactions “business_id” que tenía una “s” de más. Lo he modificado con



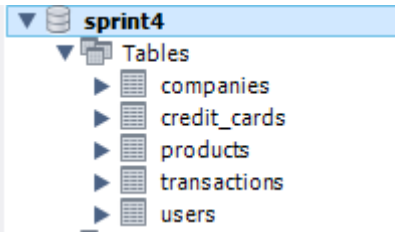
Comprobación de la modificación:



Por ahora tenemos este diagrama:



He cambiado el nombre de la base de datos ya que me parecía redundante.
He vuelto a ejecutar el script y he eliminado la base de datos anterior.



Ejercicio 1

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

```
/* Ejercicio 1 **********************
Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.*/

SELECT CONCAT(u.name, " " , u.surname) AS 'Nombre Completo'
    FROM users u
    WHERE (SELECT COUNT(t.id) FROM transactions t WHERE t.user_id = u.id) > 30;
```

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

Nombre Completo
Lynn Riddle
Ocean Nelson
Hedwig Gilbert
Kenyon Hartman

Result 3

Output

Action Output

#	Time	Action	Message	Durati
7	13:03:35	SELECT CONCAT(u.name, " " , u.surname) AS 'Nombre Completo', COUNT(transactio...	Error Code: 1054. Unknown column 'transactions.id' in 'field list'	0.000
8	13:03:49	SELECT CONCAT(u.name, " " , u.surname) AS 'Nombre Completo', COUNT(t.id) AS Nu...	Error Code: 1054. Unknown column 'Numero_transacciones' in 'where clause'	0.000
9	13:11:01	SELECT u.*, -- Consulta definitiva como respuesta de la tarea (SELECT...	4 row(s) returned	0.000
10	13:23:46	SELECT CONCAT(u.name, " " , u.surname) AS 'Nombre Completo' FROM users u W...	4 row(s) returned	0.000

Con esta consulta me faltaba información y faltaba la subconsulta. Le he añadido más datos:

```
-- Con esta consulta me faltaba información y quería añadirle más.

SELECT CONCAT(u.name, " " , u.surname) AS 'Nombre Completo', country, city,
        (SELECT COUNT(t.id)
        FROM transactions t
        WHERE u.id = t.user_id) AS Numero_transacciones
    FROM users u
    GROUP BY u.id
    HAVING Numero_transacciones > 30
    ORDER BY Numero_transacciones DESC;
```

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

Nombre Completo	country	city	Numero_transacciones
Hedwig Gilbert	Canada	Tuktoyaktuk	76
Ocean Nelson	Canada	Charlottetown	52
Kenyon Hartman	Canada	Richmond	48
Lynn Riddle	United States	Bozeman	39

Result 10

Output

Action Output

#	Time	Action	Message
16	13:43:06	SELECT CONCAT(u.name, " " , u.surname) AS 'Nombre Completo', COUNT(t.id) AS Nu...	4 row(s) returned
17	13:49:46	SELECT CONCAT(u.name, " " , u.surname) AS 'Nombre Completo', country, city, ...	4 row(s) returned

Ejercicio 2

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

```
151  /* Ejercicio 2 **********************
152  Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.*/
153
154      -- Definir alias diferente a credit_cards
155      SELECT co.company_name, cc.iban AS Tarjeta, ROUND(AVG(t.amount),2) AS promedio_importe
156          FROM companies co                                          -- alias companies = co
157      JOIN transactions t ON co.id = t.business_id
158      JOIN credit_cards cc ON cc.id = t.card_id                      -- alias credit_cards = cc
159          WHERE co.company_name = 'Donec Ltd'
160          GROUP BY cc.iban;
161
```

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

company_name	Tarjeta	promedio_importe
Donec Ltd	PT87806228135092429456346	203.72

Result 12

Output

Action Output

#	Time	Action	Message
19	14:12:26	SELECT co.company_name, cc.iban AS Tarjeta, ROUND(AVG(t.amount),2) AS prome...	1 row(s) returned
20	14:14:06	SELECT co.company_name, cc.iban AS Tarjeta, ROUND(AVG(t.amount),2) AS prome...	1 row(s) returned

Nivel 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:

```
161
162      -- ************** Nivel 2  **************
163
164  /*Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions van ser declinades
165   i genera la següent consulta:  */
166
167      -- Creamos la tabla Estado_Tarjeta
168  CREATE TABLE IF NOT EXISTS Card_Status (
169      card_id VARCHAR(20) PRIMARY KEY,
170      Status VARCHAR(30)
171  );
172
```

Output

Action Output

#	Time	Action	Message	Duration / Fe
15	16:38:50	CREATE TABLE IF NOT EXISTS Card_Status (card_id VARCHAR(20) PRIMARY ...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corr...	0.000 sec
16	16:38:55	CREATE TABLE IF NOT EXISTS Card_Status (card_id VARCHAR(20) PRIMARY ...	0 row(s) affected	0.032 sec

Para generar los registros dentro de la tabla creada “card_status”, utilizamos WITH, Common Table Expression (CTE), que es una construcción que permite definir una subconsulta a la que puedes referirte varias veces dentro de una consulta más grande. Es una tabla temporal que se crea y no se almacena de manera persistente en la base de datos. Es una tabla intermedia.

Utilizo también la función ROW_NUMBER() que asigna un número secuencial a cada fila, empezando desde 1. Este numerado se reinicia cada vez, usando PARTITION BY, lo que nos permite organizar los datos y numerar las filas dentro de cada partición de forma independiente.

```
175  INSERT INTO card_status (card_id, status)
176  WITH transacciones_tarjeta AS (
177      SELECT card_id,
178             declined,
179             ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS row_transaction
180      FROM transactions
181  )
182  SELECT card_id,
183         CASE
184             WHEN SUM(declined) = 3 THEN 'tarjeta inactiva'
185             ELSE 'tarjeta activa'
186         END AS estado_tarjeta
187  FROM transacciones_tarjeta
188  WHERE row_transaction <= 3
189  GROUP BY card_id;
190
191
192
```

Output

Action Output

#	Time	Action	Message
20	16:58:32	CREATE TABLE IF NOT EXISTS Card_Status (card_id VARCHAR(20) PRIMARY ...	0 row(s) affected
21	16:58:42	INSERT INTO card_status (card_id, status) WITH transacciones_tarjeta AS (SE...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

Chequeamos que hay dentro de la tabla recién creada:

```
187      FROM transacciones_tarjeta
188      WHERE row_transaction <= 3
189      GROUP BY card_id;
190
191      SELECT * FROM card_status;
192
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

card_id	Status
CcU-2938	tarjeta activa
CcU-2945	tarjeta activa
CcU-2952	tarjeta activa
CcU-2959	tarjeta activa
CcU-2966	tarjeta activa

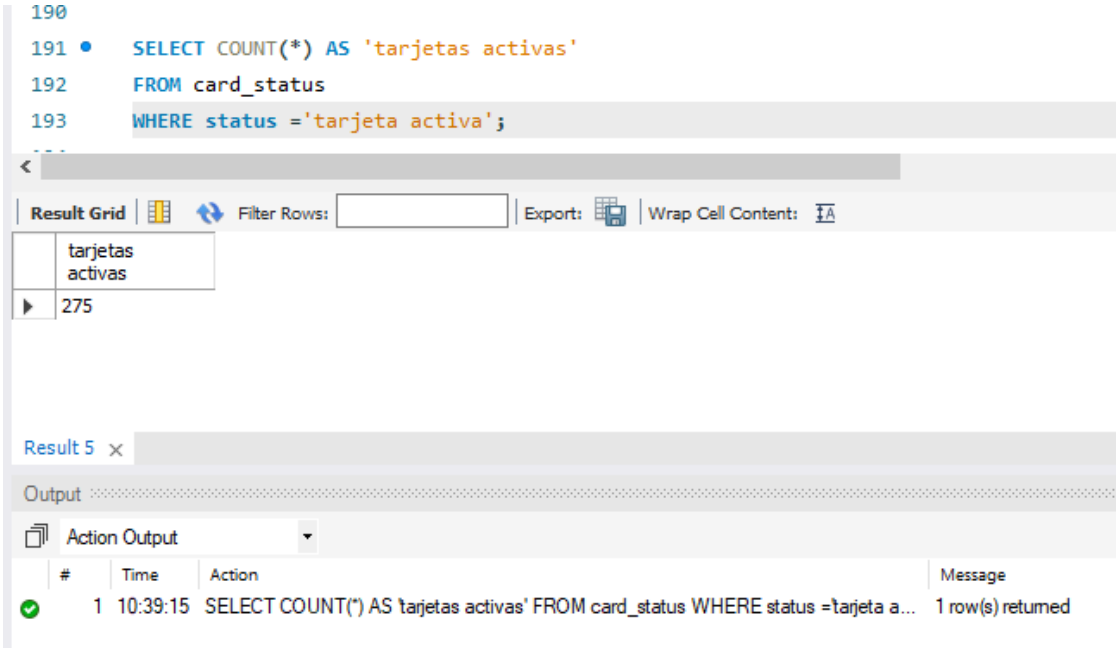
card_status 6

Output

Action Output

#	Time	Action	Message
1	10:39:15	SELECT COUNT(*) AS 'tarjetas activas' FROM card_status WHERE status ='tarjeta a...	1 row(s) returned
2	10:40:30	SELECT * FROM card_status LIMIT 0, 1000	275 row(s) returned

Ejercicio 1
Quantes targetes estan actives?



Nivel 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:

Ejercicio 1
Necessitem conèixer el nombre de vegades que s'ha venut cada producte.
Crear correctamente la foreign key de transaction: