

# SPRINT 4

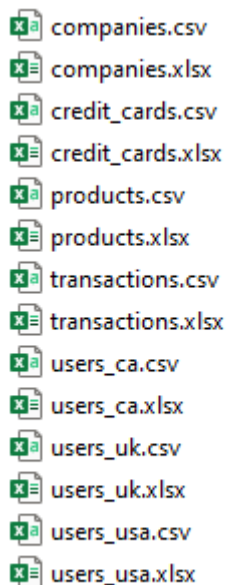
## Nivel 1

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:

```

6
7  -- Creamos la base de datos
8  • CREATE DATABASE BBDD_Sprint4;
9
10 • USE BBDD_Sprint4;
11
12  -- Creamos la tabla company
13 • CREATE TABLE IF NOT EXISTS companies (
14     id VARCHAR(15) PRIMARY KEY,
15     company_name VARCHAR(255),
16     phone VARCHAR(15),
17     email VARCHAR(100),
18     country VARCHAR(100),
19     website VARCHAR(255)

```

#	Time	Action	Message
1	14:14:58	CREATE DATABASE BBDD_Sprint4	1 row(s) affected
2	14:14:58	USE BBDD_Sprint4	0 row(s) affected
3	14:14:58	CREATE TABLE IF NOT EXISTS companies ( id VARCHAR(15) PRIMARY KEY, compan...	0 row(s) affected
4	14:14:58	CREATE TABLE IF NOT EXISTS credit_cards ( id VARCHAR(20) PRIMARY KEY, user_id ...	0 row(s) affected
5	14:14:58	CREATE TABLE IF NOT EXISTS products ( id INT PRIMARY KEY, product_name VARCHA...	0 row(s) affected
6	14:14:58	CREATE TABLE IF NOT EXISTS transactions ( id VARCHAR(255) PRIMARY KEY, card_id ...	0 row(s) affected
7	14:14:58	CREATE TABLE IF NOT EXISTS users ( id INT PRIMARY KEY, name VARCHAR(100), sum...	0 row(s) affected

-- 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.

```

73
74 -- Cargar los archivos csv en las tablas
75 • LOAD DATA LOCAL INFILE
76 'C:\Users\Nathalia\OneDrive\Escritorio\BootCamp\Proporcionados\Sprint_4_csv\companies.csv'
77 INTO TABLE companies
78 FIELDS TERMINATED BY ','
79 ENCLOSED BY '"'
80 LINES TERMINATED BY '\r\n'
81 IGNORE 1 ROWS;
82
83 • LOAD DATA LOCAL INFILE
84 'C:\Users\Nathalia\OneDrive\Escritorio\BootCamp\Proporcionados\Sprint_4_csv\credit_cards.csv'
85 INTO TABLE companies
86 FIELDS TERMINATED BY ','
87 ENCLOSED BY '"'
88 LINES TERMINATED BY '\r\n'
89 IGNORE 1 ROWS;
90
Output
Action Output
# Time Action Message Duration / Fetch
1 17:50:48 LOAD DATA LOCAL INFILE 'C:\Users\Nathalia\OneDrive\Escritorio\BootCamp\... Error Code: 2068. LOAD DATA LOCAL INFILE file request rejected due to restrictions... 0.000 sec

```

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.

```

*my.ini: Bloc de notas
Archivo Edición Formato Ver Ayuda

slow_query_log_file="LAPTOP-DNFE80P3-slow.log"

long_query_time=10

# Error Logging.
log-error="LAPTOP-DNFE80P3.err"

local_infile=1

# ***** Group Replication Related *****
# Specifies the base name to use for binary log files.
# enabled, the server logs all statements that change (
# log, which is used for backup and replication.
log_bin="LAPTOP-DNFE80P3-bin"

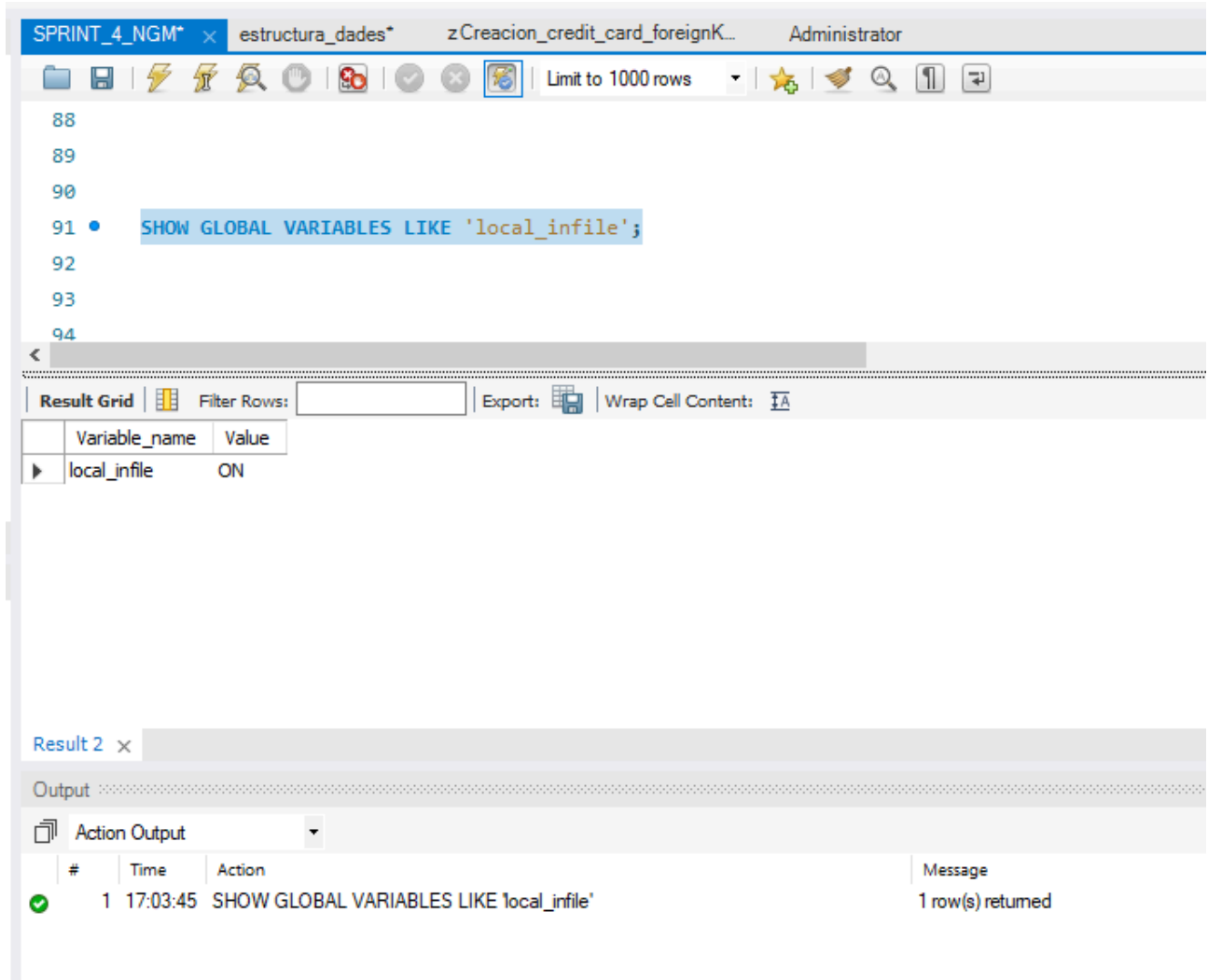
```

Usar los siguientes comandos para parar e iniciar MySQL Server:

```
net stop MySQL80
```

```
net start MySQL80
```

Verifica que local\_infile esté habilitado:



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the command `SHOW GLOBAL VARIABLES LIKE 'local_infile';` at line 91. The bottom pane shows the execution results in a grid format.

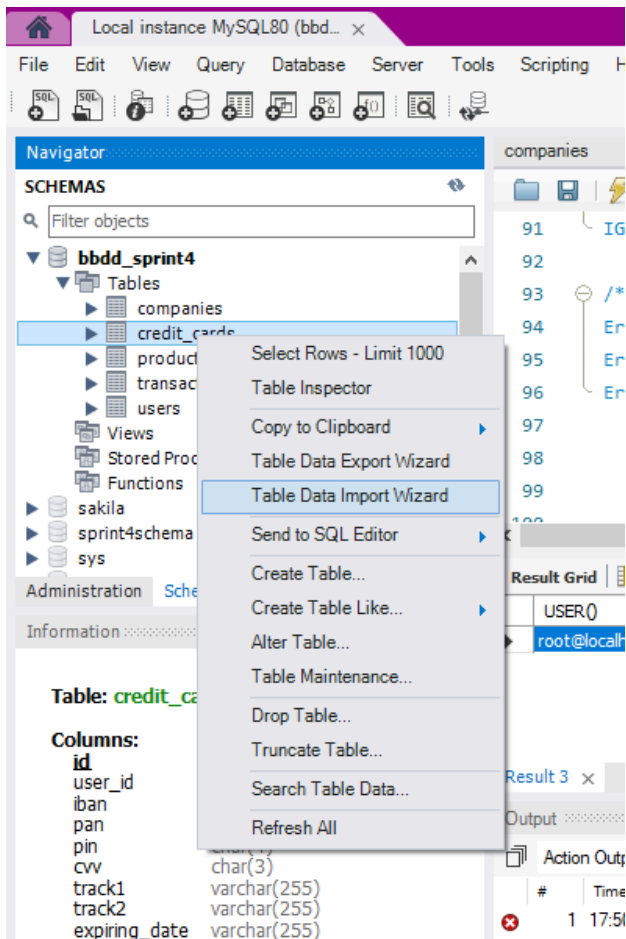
Variable_name	Value
local_infile	ON

Below the grid, the 'Output' section shows the 'Action Output' for the command. It indicates that the command was executed successfully at 17:03:45 and returned 1 row(s).

#	Time	Action	Message
1	17:03:45	SHOW GLOBAL VARIABLES LIKE 'local_infile'	1 row(s) returned

Igualmente no me deja importar:

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




### Select Destination

#### Select destination table and additional options.

- ☒ Use existing table:
- ☐ Create new table:  .
- ☐ Truncate table before import

**Table Data Import**

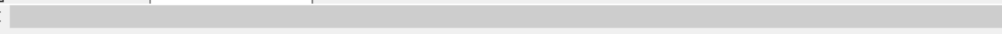
**Configure Import Settings**

Detected file format: csv 

Encoding:

Columns:

<input checked="" type="checkbox"/>	Source Column	Dest Column
<input checked="" type="checkbox"/>	id	<input type="text" value="id"/>
<input checked="" type="checkbox"/>	user_id	<input type="text" value="user_id"/>
<input checked="" type="checkbox"/>	iban	<input type="text" value="iban"/>
<input checked="" type="checkbox"/>	pan	<input type="text" value="pan"/>
<input checked="" type="checkbox"/>	pin	<input type="text" value="pin"/>
<input checked="" type="checkbox"/>	cvv	<input type="text" value="cvv"/>

< 

id	user_id	iban	pan	pin	cvv	track1	track2	expiring_d...
CcU-2938	275	TR3019503...	542446556...	3257	984	%B838371...	%B765386...	10/30/22
CcU-2945	274	DO2685476...	514242382...	9080	887	%B462131...	%B414956...	08/24/23
CcU-2952	273	BG45IVQL5...	4556 453 5...	4598	438	%B218328...	%B677858...	06/29/21
CcU-2959	272	CR7242477...	372461377...	3583	667	%B728111...	%B424615...	02/24/23
CcU-2966	271	BG72LKTQ...	448566 886...	4900	130	%B472893...	%B231857...	10/29/24

### Import Data

The following tasks will now be performed. Please monitor the execution.

- ☒ Prepare Import
- ☒ Import data file

Finished performing tasks. Click [Next >] to continue.

### Import Results

File C:\Users\Nathalia\OneDrive\Escritorio\BootCamp\Proporcionados\Sprint\_4\_csv\credit\_cards.csv was imported in 0.867 s

Table bdd\_sprint4.credit\_cards has been used

275 records imported

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', 'sumame...	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', 'sumame...	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', 'sumame...	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:

```

102 -- alter table transactions change bussiness_id business_id varchar(20);
103
104 • CREATE INDEX idx_companies ON transactions(business_id);
105 • CREATE INDEX idx_credit_cards ON transactions(card_id);
106 • CREATE INDEX idx_users ON transactions(user_id);
107
108 -- Se crean las relaciones entre las tablas con sus respectivos foreign key
109

```

Output

#	Time	Action	Message
1	20:24:17	CREATE INDEX idx_companies ON transactions(business_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
2	20:24:17	CREATE INDEX idx_credit_cards ON transactions(card_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3	20:24:17	CREATE INDEX idx_users ON transactions(user_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Y luego las Foreign Keys:

```

110 -- Relación entre transactions y companies
111 • ALTER TABLE transactions
112     ADD CONSTRAINT fk_company
113     FOREIGN KEY (business_id) REFERENCES companies(id);
114
115 -- Relación entre transactions y credit_cards
116 • ALTER TABLE transactions
117     ADD CONSTRAINT fk_credit_card
118     FOREIGN KEY (card_id) REFERENCES credit_cards(id);
119
120 -- Relación entre transactions y users
121 • ALTER TABLE transactions
122     ADD CONSTRAINT fk_user
123     FOREIGN KEY (user_id) REFERENCES users(id);
124

```

Output

#	Time	Action	Message
2	20:24:17	CREATE INDEX idx_credit_cards ON transactions(card_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3	20:24:17	CREATE INDEX idx_users ON transactions(user_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
4	20:25:45	ALTER TABLE transactions ADD CONSTRAINT fk_company FOREIGN KEY (business...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
5	20:25:45	ALTER TABLE transactions ADD CONSTRAINT fk_credit_card FOREIGN KEY (card_id...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
6	20:25:45	ALTER TABLE transactions ADD CONSTRAINT fk_user FOREIGN KEY (user_id) REFE...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

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

```

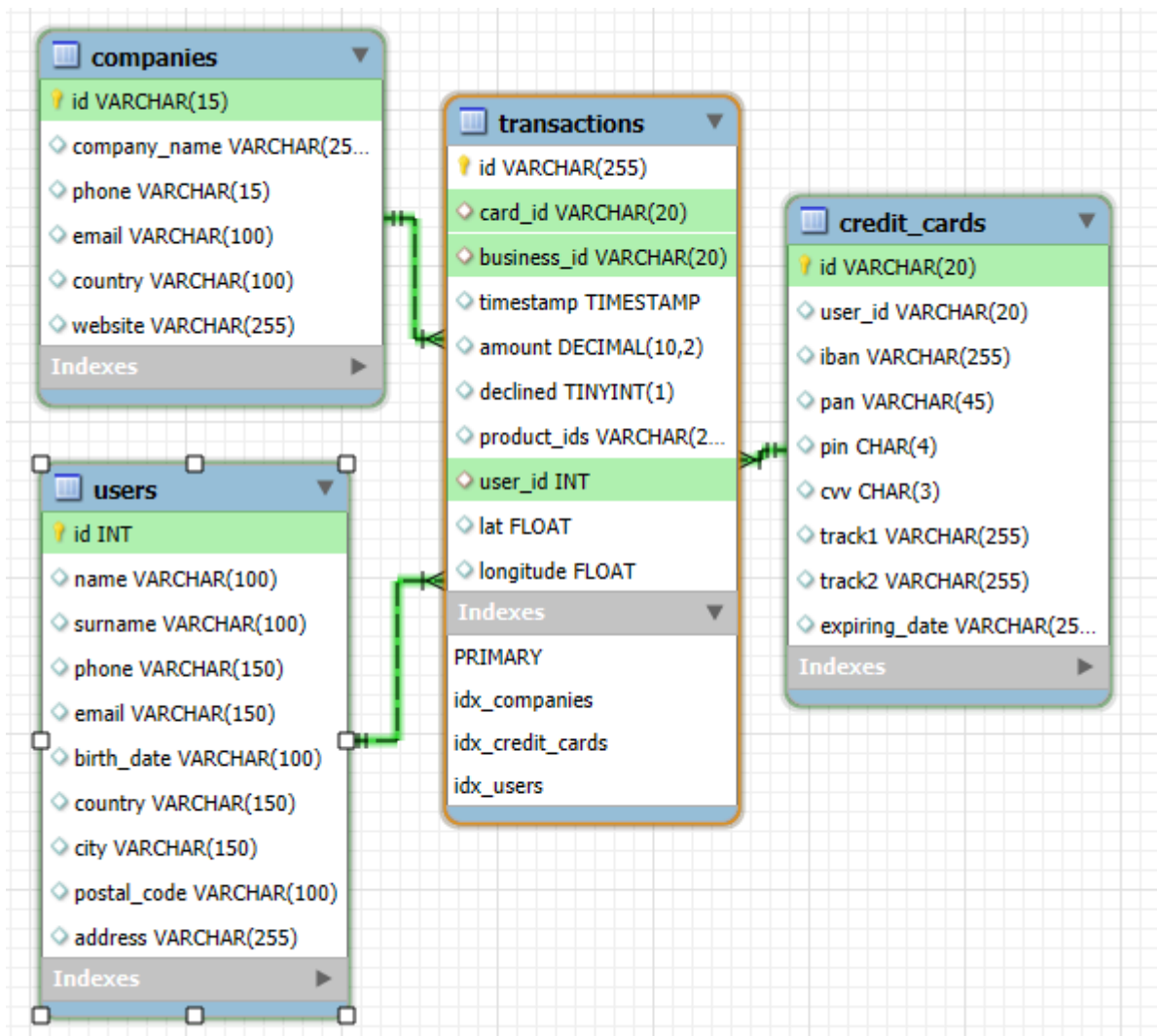
101
102 • alter table transactions change bussiness_id business_id varchar(20);
103

```

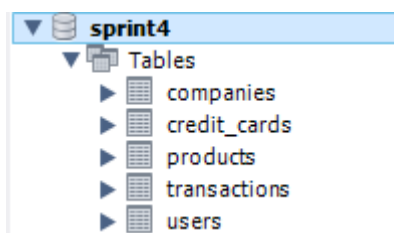
Comprobación de la modificación:

id	card_id	business_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-2E4C4F8C8AE	CcU-3743	b-2618	2022-01-26 02:07:14	394.18	0	89, 83, 79	265	-34.3593	-100.556

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.

```

133  /* Ejercicio 1 *****
134  Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.*/
135
136  • SELECT CONCAT(u.name, " ", u.surname) AS 'Nombre Completo'
137      FROM users u
138      WHERE (SELECT COUNT(t.id) FROM transactions t WHERE t.user_id = u.id) > 30;
139

```

Result Grid

	Nombre Completo
▶	Lynn Riddle
	Ocean Nelson
	Hedwig Gilbert
	Kenyon Hartman

Result 3 ×

Output

Action Output

#	Time	Action	Message	Duration
7	13:03:35	SELECT CONCAT(u.name, " ", u.surname) AS 'Nombre Completo', COUNT(transaccio...	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
10	13:23:46	SELECT CONCAT(u.name, " ", u.surname) AS 'Nombre Completo' FROM users u W...		4 row(s) returned

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

```

140  -- Con esta consulta me faltaba información y quería añadirle más.
141
142  • SELECT CONCAT(u.name, " ", u.surname) AS 'Nombre Completo', country, city,
143      (SELECT COUNT(t.id)
144      FROM transactions t
145      WHERE u.id = t.user_id) AS Numero_transacciones
146      FROM users u
147      GROUP BY u.id
148      HAVING Numero_transacciones > 30
149      ORDER BY Numero_transacciones DESC;
150

```

Result Grid

	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

company_name	Tarjeta	promedio_importe
Donec Ltd	PT87806228135092429456346	203.72

Result 12 x

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
173  Se introducen los datos con filtros según la petición

```

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 x

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?

```
190
191 • SELECT COUNT(*) AS 'tarjetas activas'
192 FROM card_status
193 WHERE status ='tarjeta activa';
```

Result Grid

tarjetas activas
275

Result 5 x

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

## 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 correctament la foreign key de transaction: