

SPRINT 4

ÚLTIMO REPASO DE SQL EN EL QUE TRABAJAREMOS LA
CREACIÓN DE BASES DE DATOS, LAS RELACIONES ENTRE
TABLAS, TÉCNICAS DE MODELADO Y GRANULARIDAD DE LOS
DATOS.

Raquel Limpo
Martínez

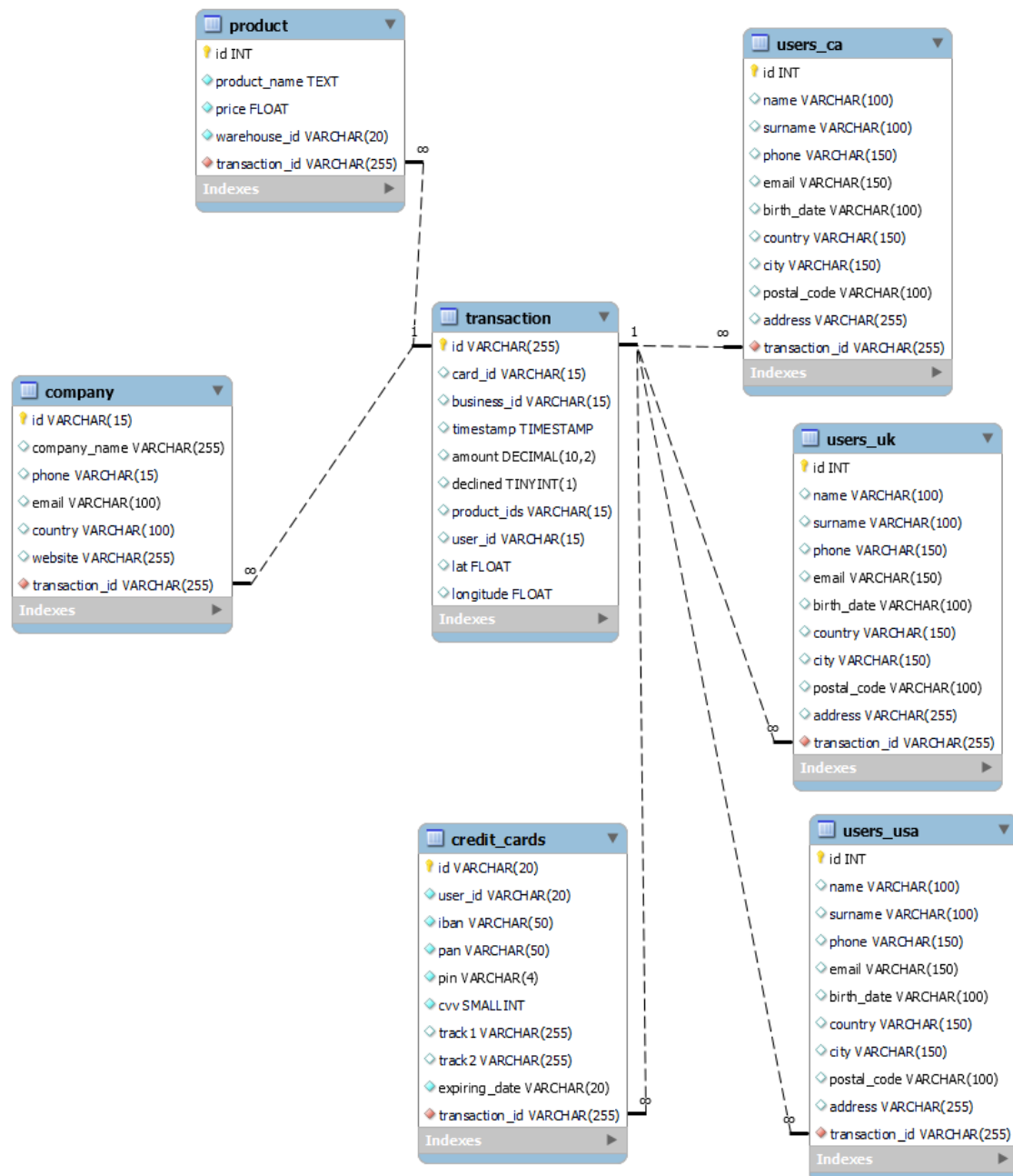
TABLA DE CONTENIDO

Nivel 1	2
Esquema en estrella	2
Creación de la base de datos e importación de los datos de los CSV	5
Ejercicio 1: Realiza una subconsulta que muestre a todos los usuarios con más de 30 transacciones utilizando al menos 2 tablas.....	11
Ejercicio 2 Muestra la media de amount por IBAN de las tarjetas de crédito a la compañía Donec Ltd., utiliza por lo menos 2 tablas.	12
Nivel 2:	13
Ejercicio 1: Crea una nueva tabla que refleje el estado de las tarjetas de crédito basado en si las últimas tres transacciones fueron declinadas y genera la siguiente consulta: Cuántas tarjetas están activas?.....	13
Nivel 3	14
Ejercicio 1: Necesitamos conocer el número de veces que se ha vendido cada producto.	14

Nivel 1

Descarga los archivos CSV, estúdialos y diseña una base de datos con un esquema en estrella que contengas al menos 4 tablas de las cuales puedas realizar las consultas de los ejercicios 1 y 2:

Esquema en estrella



En este esquema de estrella, la tabla factual o de hechos sería **Transaction**, no solo porque está conectada con todas las demás, sino que contiene los indicadores del negocio, cuanto se ha vendido, cuando a quien de parte de quien, cuanto dinero se ha ganado...

Esta unida a las tablas de dimensiones: **Product**, **Company**, **Credit_cards**, **Users_uk**, **Users_ca** y **Users_usa**. Podría haber unido todas las tablas *users* en una sola, cómo estaba en el Sprint 3,

pero he pensado que si se entregaban así era porque se necesitaban por separado:

Tabla Transaction:

- id : El identificador de la transacción realizada. Es la clave primaria de esta tabla Como he comentado previamente, sería ideal que este campo fuera distintivo como *transaction_id*. Lo mismo aplica para el resto de tablas.
- card_id : El identificador de la tarjeta de crédito con la que se ha realizado la transacción, nos une a la tabla *Credit_card*
- business_id : El identificador de la compañía desde la que ha realizado la transacción, nos une a la tabla *Company*.
- timestamp : La hora a la que se ha realizado la transacción
- amount : La cantidad de dinero que se ha usado en la transacción
- declined : Campo booleano, nos indica si la transacción se ha realizado (0) o se ha cancelado (1)
- product_ids : los identificadores de los diferentes productos que se han intercambiado por dinero en la transacción. Cuando hay más de uno está separado por comas. Nos une a la tabla *Product*.
- user_id : El identificador del usuario que ha realizado la transacción. Nos une por clave foránea a las tablas *Users*
- lat y longitude: Las coordenadas desde donde se ha hecho la transacción.

Tabla Company:

- id: El identificador para cada compañía. ES la clave primaria de esta tabla.
- company_name. El nombre de la compañía.
- phone : El número de contacto de cada compañía.
- email: el email de contacto de cada compañía
- country: el país donde esta la sede central de cada compañía. Este sería el único campo que no tendría por qué ser único de esta tabla.
- website: La página oficial de la compañía.

Tabla Credit_cards:

- id: Es el identificador de la tarjeta de crédito. Este campo es la clave primaria,
- iban: Es el código internacional de cuenta bancaria. Es un código alfanumérico que identifica una cuenta bancaria en cualquier lugar del mundo.
- pan: es el acrónimo de personal account number y es el número que aparece en la tarjeta de crédito.
- pin: personal identification number es tu clave para entrar en tu cuenta.
- cvv: código valor de verificación. Es un número que está detrás de la tarjeta y que se suele pedir por seguridad en transacciones online.
- track1 y track2 : Son las pistas de banda magnética. Los lectores de losTPV siempre leen las pistas 1 o 2: la información mínima necesaria para realizar una transacción se encuentra en ambas pistas. La pista 1 tiene una mayor densidad de bits, es la única que puede contener caracteres alfanuméricos, y por tanto es la única que puede contener el nombre del portador de la tarjeta.
- Expiring_date La fecha de caducidad de la tarjeta. Al igual que el cvv se pide a menudo en transacciones online.

Tabla Product:

- id : El identificador del producto, es la primary key de la tabla.
- product_name : El nombre del producto.
- price El precio del producto, es un float y lleva símbolo de dólar, con lo cual puede dar problema al cargar la tabla.
- Colour: color en código hexadecimal del producto
- weight Peso del producto, sirve a la hora de enviar el producto y para tenerlo en cuenta al recibirlo.
- warehouse_id : Identificador del almacén, sirve para saber dónde está cada producto (qué almacén, qué zona del almacén, qué estante...).

Tablas Users de Canada, Reino Unido y Estados Unidos: todas tienen los mismos campos

- id: Identificador del usuario, clave primaria.
- Name: nombre del usuario
- Surname: apellido del usuario
- Phone: número de contacto del usuario
- Email: Correo electrónico de contacto del usuario.
- birth_date : Fecha de nacimiento del usuario
- country: país de residencia del usuario
- city: Ciudad de residencia del usuario
- postal_code: Código postal del lugar de residencia del usuario
- address : dirección del lugar de residencia del usuario

Creación de la base de datos e importación de los datos de los CSV

```
-- Creamos la base de datos
CREATE DATABASE IF NOT EXISTS transactions;
USE transactions;

-- Creamos las tablas

#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)
);

# transactions
CREATE TABLE IF NOT EXISTS transactions (
  id VARCHAR(255) PRIMARY KEY,
  card_id VARCHAR(15) REFERENCES credit_card(id),
  business_id VARCHAR(15),
  timestamp TIMESTAMP,
  amount DECIMAL(10, 2),
  declined BOOLEAN,
  product_ids VARCHAR(15),
  user_id VARCHAR(15),
  lat FLOAT,
  longitude FLOAT
);

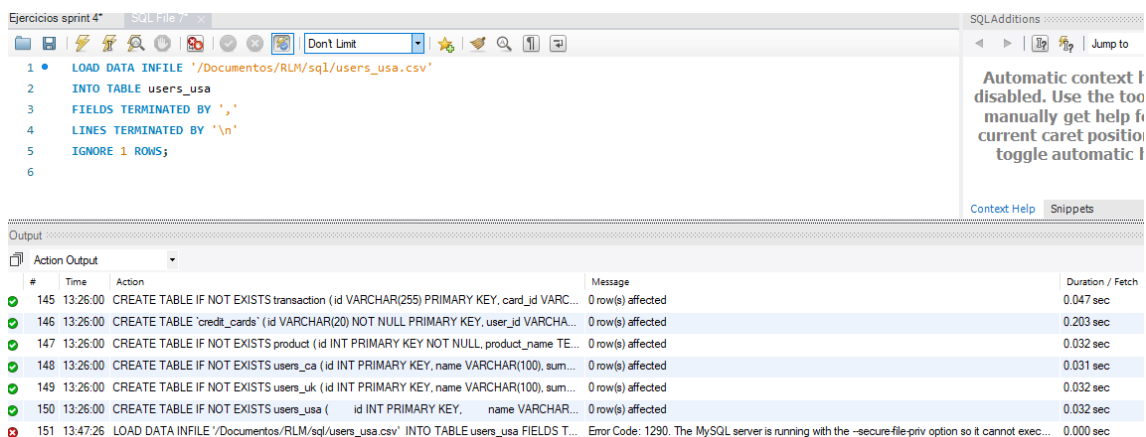
CREATE TABLE credit_cards (
  id VARCHAR(20) PRIMARY KEY,
  user_id VARCHAR(20) NOT NULL,
  iban VARCHAR(50) NOT NULL,
  pan VARCHAR(50) NOT NULL,
  pin VARCHAR(4) NOT NULL,
  cvv SMALLINT NOT NULL,
  track1 VARCHAR(255),
  track2 VARCHAR(255),
  expiring_date VARCHAR(20) NOT NULL
);

#product
CREATE TABLE IF NOT EXISTS product (
  id INT PRIMARY KEY,
  product_name TEXT NOT NULL ,
  price FLOAT NOT NULL ,
  warehouse_id VARCHAR(20) NOT NULL
);

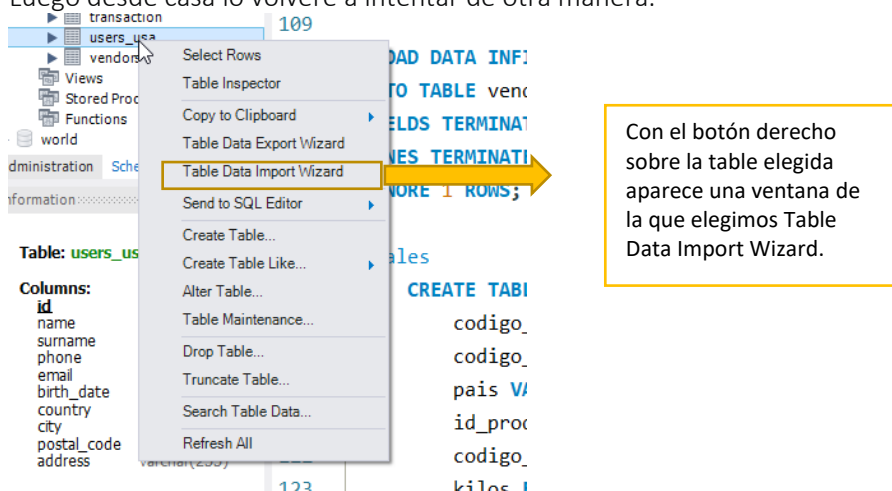
#user_ca
CREATE TABLE IF NOT EXISTS users_ca (
  id INT PRIMARY KEY,
  name VARCHAR(100),
  surname VARCHAR(100),
  phone VARCHAR(150),
  email VARCHAR(150),
  birth_date VARCHAR(100),
  country VARCHAR(150),
  city VARCHAR(150),
  postal_code VARCHAR(100),
  address VARCHAR(255)
);

#user_uk
CREATE TABLE IF NOT EXISTS users_uk (
  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)
);

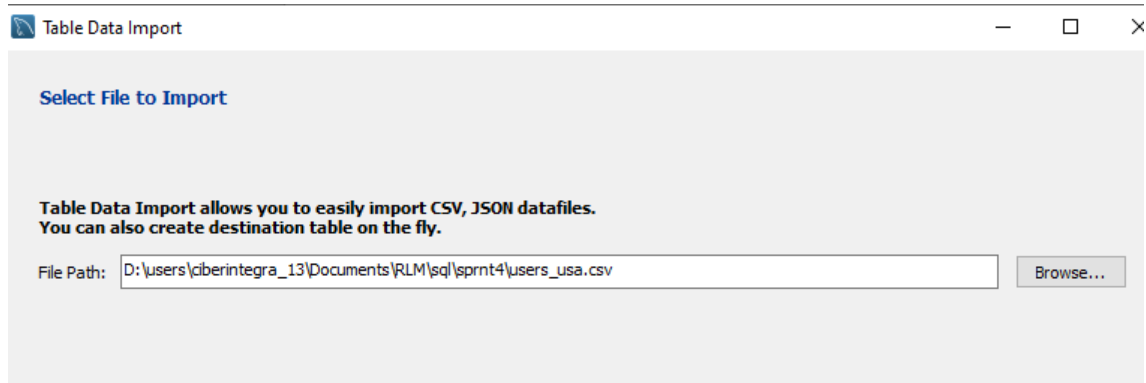
#user_usa
CREATE TABLE IF NOT EXISTS users_usa (
  id INT PRIMARY KEY,
  name VARCHAR(100),
  surname VARCHAR(100),
  phone VARCHAR(150),
  email VARCHAR(150),
  birth_date VARCHAR(100),
  country VARCHAR(150),
  city VARCHAR(150),
  postal_code VARCHAR(100),
  address VARCHAR(255)
);
```



Originalmente iba a cargarlo así, però me aparecía el error *"Error Code: 1290 - The MySQL server is running with the --secure-file-priv option so it cannot execute this statement."* Y las maneras de corregirlo son difíciles en el ordenador del cibernàrium, ya que tengo que entrar en zonas seguras y no tengo los permisos, así que voy a ir avanzando, usando el import wizard. Luego desde casa lo volveré a intentar de otra manera.



Nos abre el Wizard y le decimos de donde sacamos el archivo.



Le damos a Next y elegimos destino

Select Destination

Select destination table and additional options.

☒ Use existing table: transactions.users_usa

☐ Create new table: transactions . users_usa

☐ Truncate table before import

Le volvemos a dar a Next

Table Data Import

Configure Import Settings

Detected file format: csv

Encoding: utf-8

Columns:

Source Column	Dest Column
<input checked="" type="checkbox"/> id	id
<input checked="" type="checkbox"/> name	name
<input checked="" type="checkbox"/> surname	surname
<input checked="" type="checkbox"/> phone	phone
<input checked="" type="checkbox"/> email	email
<input checked="" type="checkbox"/> birth_date	birth_date

id	name	surname	phone	email	birth_date	country	city	postal_code	address
1	Zeus	Gamble	1-282-581-...	interdum.e...	Nov 17, 1985	United States	Lowell	73544	348-78:
2	Garrett	Mcconnell	(718) 257-...	integer.vita...	Aug 23, 1992	United States	Des Moines	59464	903 Sit
3	Ciaran	Harrison	(522) 598-...	interdum.fe...	Apr 29, 1998	United States	Columbus	56518	736-206
4	Howard	Stafford	1-411-740-...	ornare.eques...	Feb 18, 1989	United States	Kailua	77417	Ap #54

< Back Next > Cancel

Table Data Import

Import Data

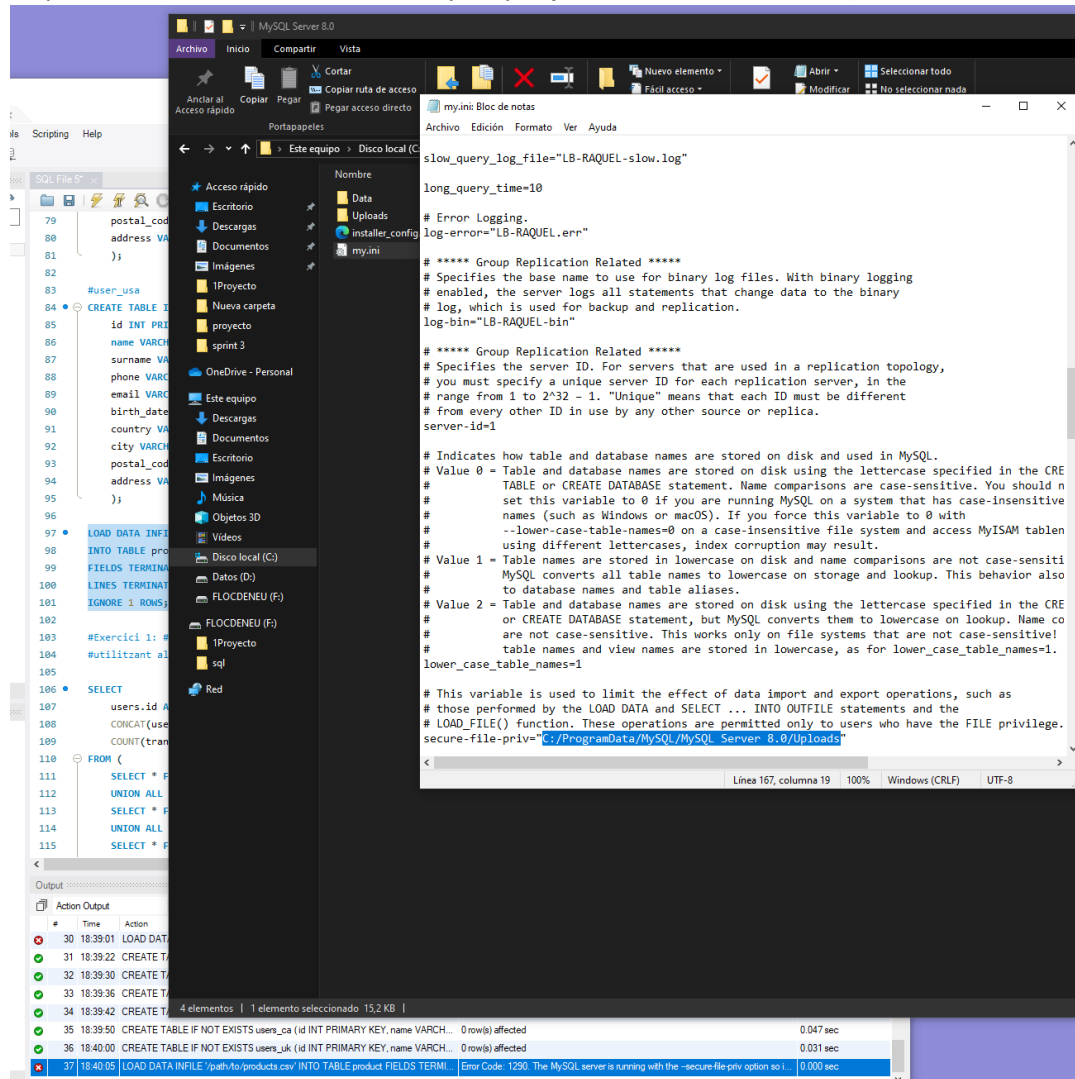
The following tasks will now be performed. Please monitor the execution.

☐ Prepare Import
☐ Import data file

Click [Next >] to execute.

Como corregirlo desde casa:

Hay dos maneras, una abriendo el my.ini y dejando las comillas vacías (dónde esta resaltado).



La segunda, llevar los archivos csv a este directorio C:/ProgramData/MySQL/MySQL Server 8.0/Uploads , que es la que voy a hacer ahora.

Nombre	Fecha de modificación	Tipo	Tamaño
companies.csv	20/11/2024 10:37	Archivo de valores...	11 KB
credit_cards.csv	20/11/2024 10:37	Archivo de valores...	41 KB
products (1).csv	25/11/2024 11:14	Archivo de valores...	5 KB
transactions.csv	20/11/2024 10:37	Archivo de valores...	72 KB
users_ca.csv	20/11/2024 10:38	Archivo de valores...	9 KB
users_uk.csv	20/11/2024 10:38	Archivo de valores...	7 KB
users_usa.csv	20/11/2024 10:38	Archivo de valores...	19 KB

Para cualquiera de ellas hay que volver a iniciar el mysql.

Bueno, a pesar de que ahora funciona, cada tabla da un error y hay que corregir cada una. Products, da problemas a causa del símbolo del dólar y hay que añadir SET price = REPLACE(@price_raw, '\$', '');

Companies , transactions y credit_cards, estan separadas por ; en vez de , y tienen problemas con la falta de comillas en algunos registros, mientras que users, tenía un problema de salto de línea. Estuve varias horas preguntandome qué no funcionaba en las tablas de usuarios y revise el manual <http://download.nust.na/pub6/mysql/doc/refman/5.0/es/load-data.html#:~:text=Los%20valores%20FIELDS%20TERMINATED%20BY,BY%20'%5Cr%5Cn'%20.>

Vi que depende de qué documentos y cómo hubieran sido creados hacia falta poner el /r
“**Nota:** Si ha generado el fichero de texto en un sistema Windows , puede tener que usar **LINES TERMINATED BY '\r\n'** para leer correctamente el fichero, ya que los programas de Windows típicamente usan dos caracteres como terminadores de línea . Algunos programas como *WordPad*, pueden usar **\r** como terminador de línea al escribir ficheros. Para leer tales ficheros, use **LINES TERMINATED BY '\r'**.”

Cuando ya he conseguido terminar los ejercicios, Mailin me ha comentado que había otra manera de hacerlo que es buscando la ruta a través del código sql “SHOW VARIABLES LIKE 'secure_file_priv';”, este te da la ruta exacta y te permite entrar en carpetas, aunque estén ocultas, pudiendo copiar ahí los archivos.

```

#cargar archivos csv.
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv'
INTO TABLE product
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(id, product_name, @price_raw, colour, weight, warehouse_id)
SET price = REPLACE(@price_raw, '$', '');

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv'
INTO TABLE companies
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS ;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv'
INTO TABLE credit_cards
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'
INTO TABLE transactions
FIELDS TERMINATED BY ';'
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv'
INTO TABLE users_ca
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv'
INTO TABLE users_uk
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv'
INTO TABLE users_usa
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;

```

```

CREATE INDEX idx_companies
  ON transactions(business_id);

CREATE INDEX idx_credit_cards
  ON transactions(card_id);

CREATE INDEX idx_users
  ON transactions(user_id);

ALTER TABLE companies
  ADD FOREIGN KEY (id) REFERENCES transactions(business_id);

ALTER TABLE credit_cards
  ADD FOREIGN KEY (id) REFERENCES transactions(card_id);

```

Ejercicio 1: Realiza una subconsulta que muestre a todos los usuarios con más de 30 transacciones utilizando al menos 2 tablas.

```

153
154 • SELECT
155     users.id,
156     CONCAT(users.name, ' ', users.surname) AS full_name,
157     COUNT(transactions.id) AS transactions_count
158 FROM (
159     SELECT * FROM users_ca
160     UNION ALL
161     SELECT * FROM users_uk
162     UNION ALL
163     SELECT * FROM users_usa
164 ) users
165 JOIN transactions
166     ON users.id = transactions.user_id
167 GROUP BY users.id, users.name, users.surname
168 HAVING COUNT(transactions.id) > 30;

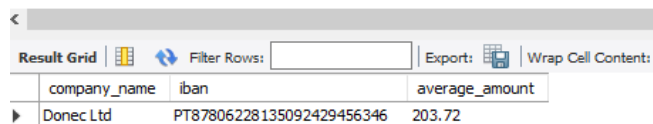
```

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

id	full_name	transactions_count
92	Lynn Riddle	39
275	Kenyon Hartman	48
272	Hedwig Gilbert	76
267	Ocean Nelson	52

Ejercicio 2 Muestra la media de amount por IBAN de las tarjetas de crédito a la compañía Donec Ltd., utiliza por lo menos 2 tablas.

```
1 • SELECT credit_cards.iban, companies.company_name,  
2        ROUND(AVG(amount),2) AS average_amount  
3        FROM credit_cards  
4       JOIN transactions  
5         ON credit_cards.id = transactions.card_id  
6       JOIN companies  
7         ON transactions.business_id = companies.company_id  
8      WHERE companies.company_name = 'Donec Ltd'  
9      GROUP BY credit_cards.iban;  
10
```



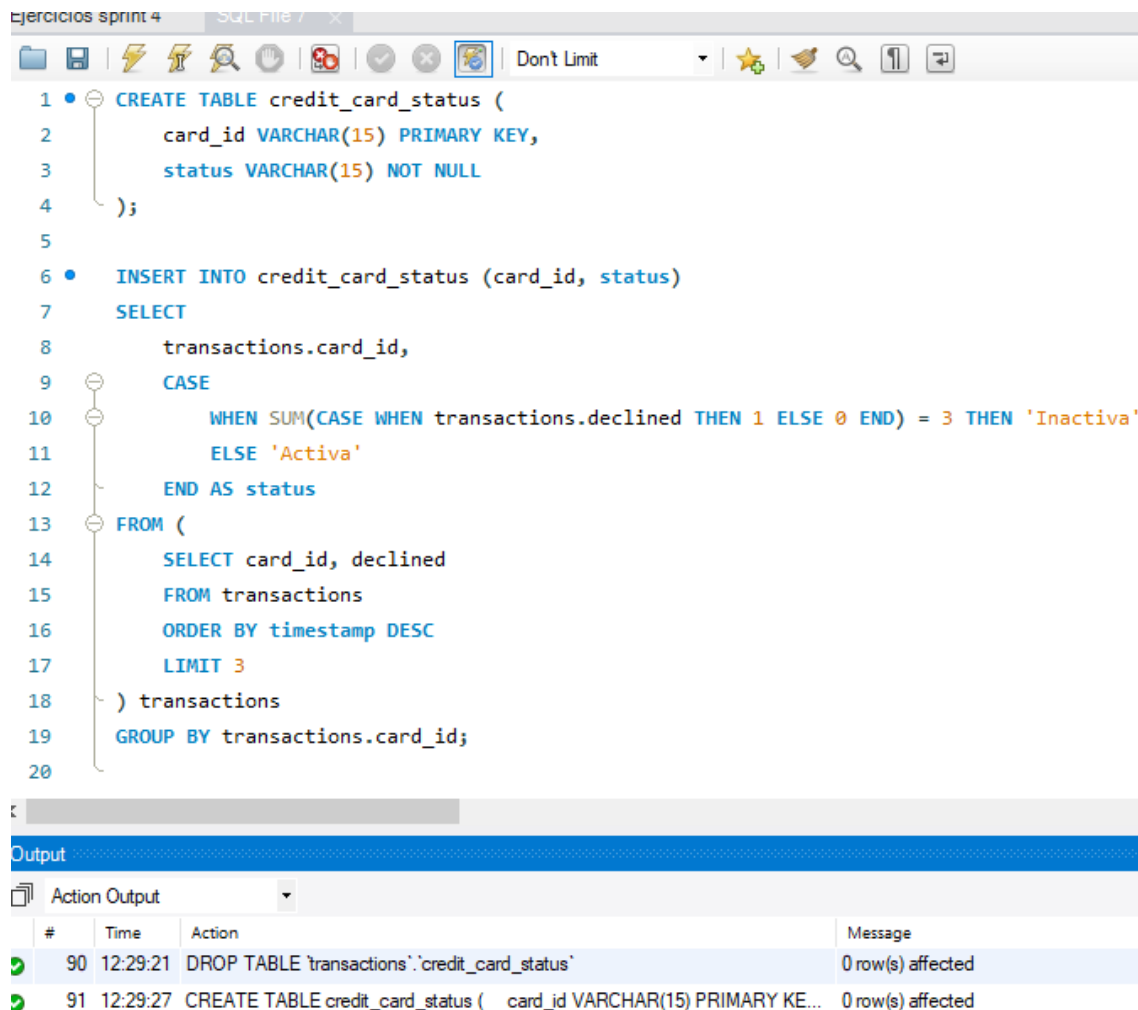
The screenshot shows the SQL Workbench interface with the 'Result Grid' tab selected. The grid displays the results of the SQL query, showing the company name, IBAN, and the average amount for Donec Ltd.

	company_name	iban	average_amount
▶	Donec Ltd	PT87806228135092429456346	203.72

Nivel 2

Ejercicio 1: Crea una nueva tabla que refleje el estado de las tarjetas de crédito basado en si las últimas tres transacciones fueron declinadas y genera la siguiente consulta: ¿Cuántas tarjetas están activas?

Crea una nueva tabla para el estado de las tarjetas.



The screenshot shows the SQL Workbench interface. The top pane displays the following SQL code:

```
1 • CREATE TABLE credit_card_status (  
2     card_id VARCHAR(15) PRIMARY KEY,  
3     status VARCHAR(15) NOT NULL  
4 );  
5  
6 • INSERT INTO credit_card_status (card_id, status)  
7     SELECT  
8         transactions.card_id,  
9         CASE  
10            WHEN SUM(CASE WHEN transactions.declined THEN 1 ELSE 0 END) = 3 THEN 'Inactiva'  
11            ELSE 'Activa'  
12        END AS status  
13     FROM (  
14         SELECT card_id, declined  
15         FROM transactions  
16         ORDER BY timestamp DESC  
17         LIMIT 3  
18     ) transactions  
19     GROUP BY transactions.card_id;  
20
```

The bottom pane shows the 'Output' tab with the 'Action Output' view. It contains the following table:

#	Time	Action	Message
90	12:29:21	DROP TABLE `transactions`.`credit_card_status`	0 row(s) affected
91	12:29:27	CREATE TABLE credit_card_status (card_id VARCHAR(15) PRIMARY KE...	0 row(s) affected

Nivel 3

Crea una tabla con la que podamos unir los datos del nuevo archivo products.csv con la base de datos creada, teniendo en cuenta que desde transaction tienes product_ids. Genera la siguiente consulta:

Ejercicio 1: Necesitamos conocer el número de veces que se ha vendido cada producto.

```
1  #Exercici 1: Necessitem conèixer el nombre de vegades que s'ha venut cada producte.
2  • SELECT product.product_name,
3     COUNT(transactions.product_ids) AS sales_count
4  FROM product
5  JOIN transactions ON product.id = transactions.product_ids
6  GROUP BY product.product_name
7  ORDER BY sales_count DESC
8  ;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
product_name	sales_count			
skywalker ewok	46			
Direwolf Stannis	42			
riverlands north	32			
Direwolf riverlands the	31			
north of Casterly	30			
duel	29			
Winterfell Lannister	28			
Tully	26			
skywalker ewok sith	26			
kingsblood Littlefinger the	26			
Winterfell	24			
Direwolf Littlefinger	23			
duel tourney	22			
palpatine chewbacca	22			
Lannister Barratheon Dir...	22			
jinn Winterfell	21			
Tarly Stark	21			
dooku solo	20			
Tully Dorne	18			
Karstark Dorne	18			
duel tourney Lannister	17			
Dorne bastard	16			
Tully maester Tarly	14			
Lannister	13			