Descripción

Partiendo de algunos archivos CSV diseñarás y crearás tu base de datos.

Nivel 1

Descarga los archivos CSV, estudiales y diseña una base de datos con un esquema de estrella que contenga, al menos 4 tablas de las que puedas realizar las siguientes consultas:

Creo una base de datos nueva y le llamo sales_track. Indico que es la base que vamos a usar

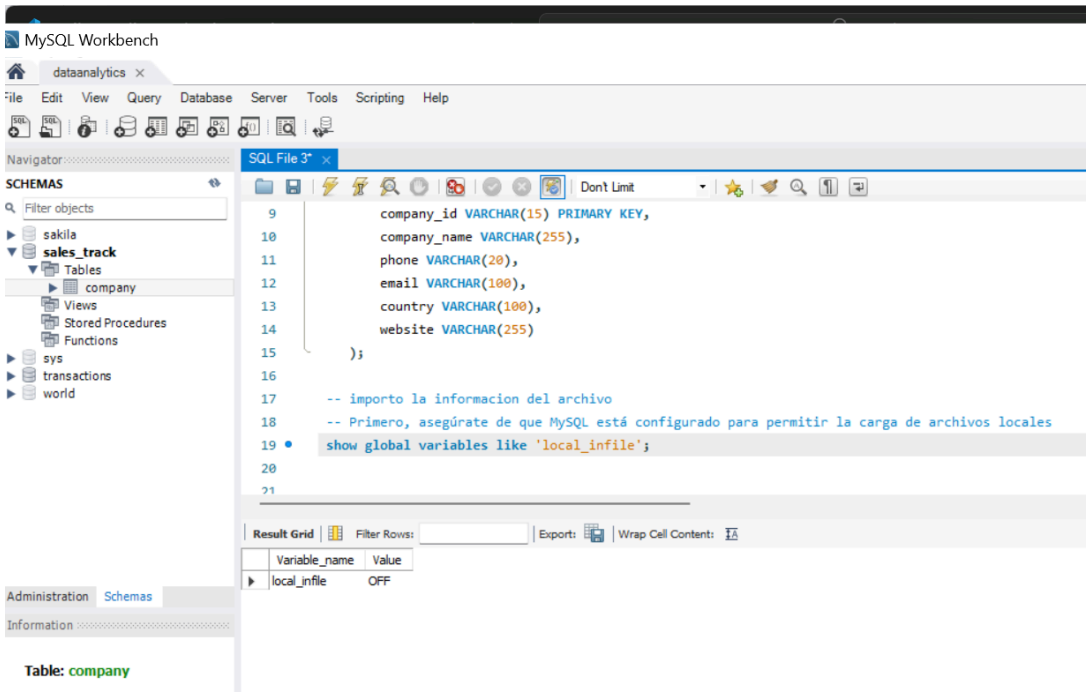
```
CREATE DATABASE IF NOT EXISTS sales_track;
USE sales_track;
```

Creo estructura de tabla company

```
CREATE TABLE IF NOT EXISTS company (
company_id VARCHAR(15) PRIMARY KEY,
company_name VARCHAR(255),
phone VARCHAR(20),
email VARCHAR(100),
country VARCHAR(100),
website VARCHAR(255)
);
```

Voy a importar la información del archivo
- Primero, reviso que MySQL está configurado para permitir la carga de archivos locales

```
show global variables like 'local_infile';
```



-- Si 'local_infile' está OFF, lo activo con esta línea de comandos:

```
SET GLOBAL local_infile = 1;
```

– Renombro la tabla company como companies

```
alter table company rename to companies;
```

– Importación de los datos desde el archivo companies.csv:
No me permite importar los datos desde el directorio donde los tengo.

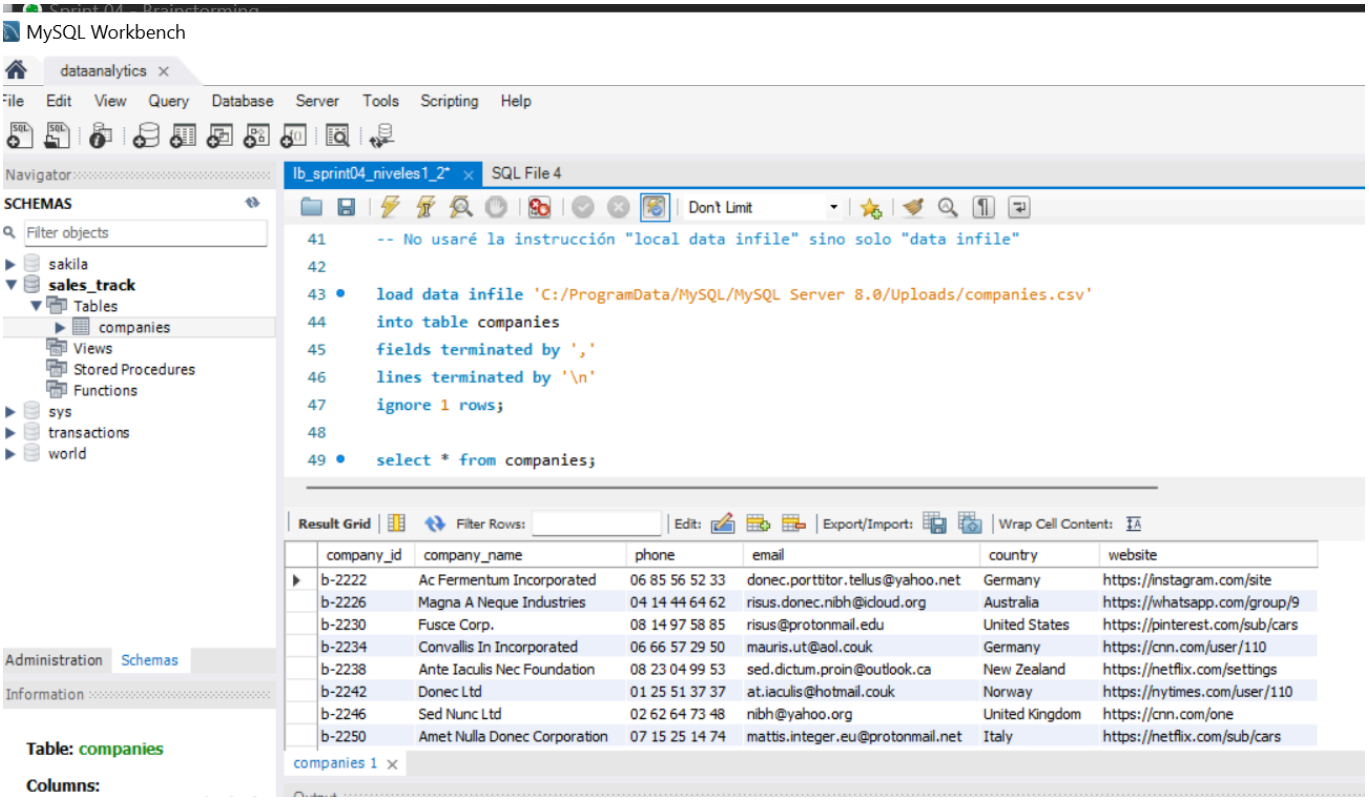
Ejecuto el script: SHOW VARIABLES LIKE 'secure_file_priv';

Me aparece el directorio desde donde me está permitido importar archivos:
"C:\ProgramData\MySQL\MySQL Server 8.0\Uploads”

Coloco el archivo companies.csv en el directorio que me indica secure_file_priv:
“C:\ProgramData\MySQL\MySQL Server 8.0\Uploads"

-- No usaré la instrucción "local data infile" sino solo "data infile":

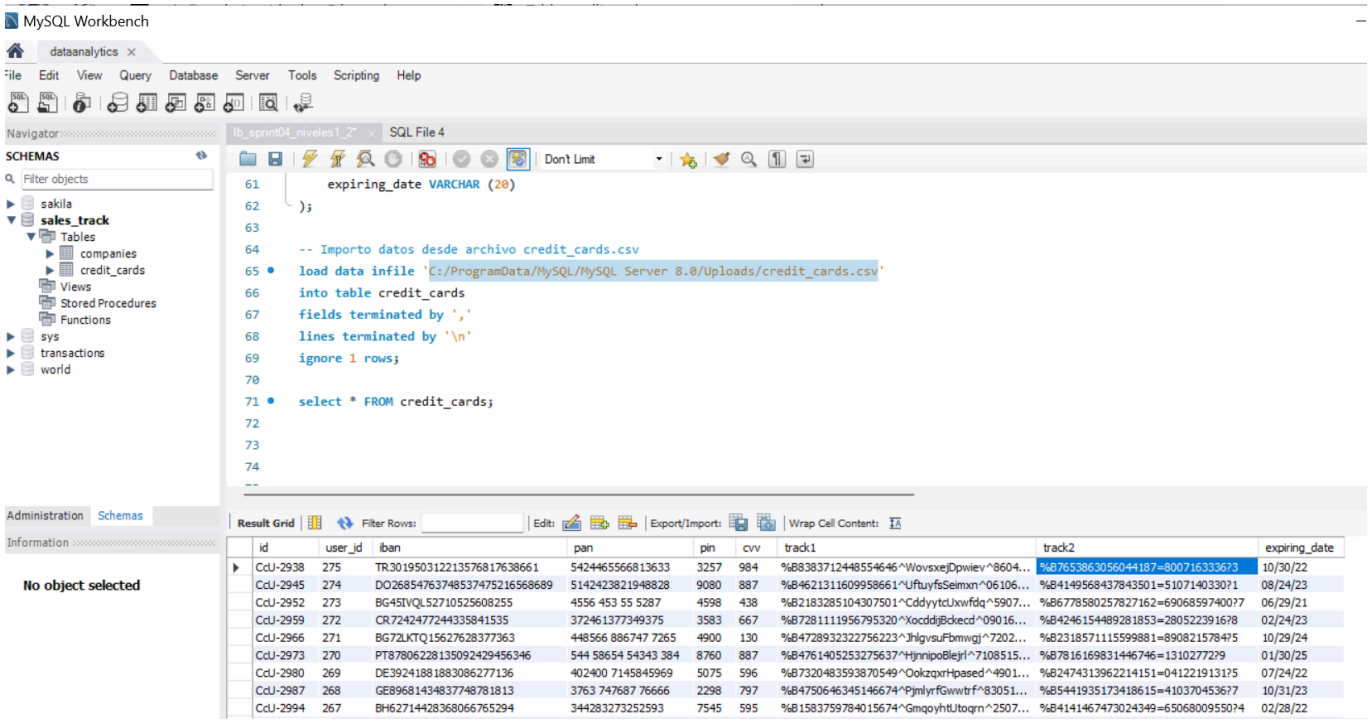
```
load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv'
into table companies
fields terminated by ','
lines terminated by '\n'
ignore 1 rows;
```



Creo tabla credit_card e importo datos del archivo credit_cards.csv

```
CREATE TABLE IF NOT EXISTS credit_cards (
id VARCHAR(15) PRIMARY KEY,
user_id SMALLINT UNSIGNED,
iban VARCHAR(40),
pan VARCHAR(20) UNIQUE,
pin SMALLINT UNSIGNED,
cvv  SMALLINT UNSIGNED,
track1 VARCHAR(50),
track2 VARCHAR(50),
expiring_date VARCHAR (20)
);

load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv'
into table credit_cards
fields terminated by ','
lines terminated by '\n'
ignore 1 rows;
```



Creo la tabla users e importo la data de todos los archivos: users_usa, users_ca, users_uk

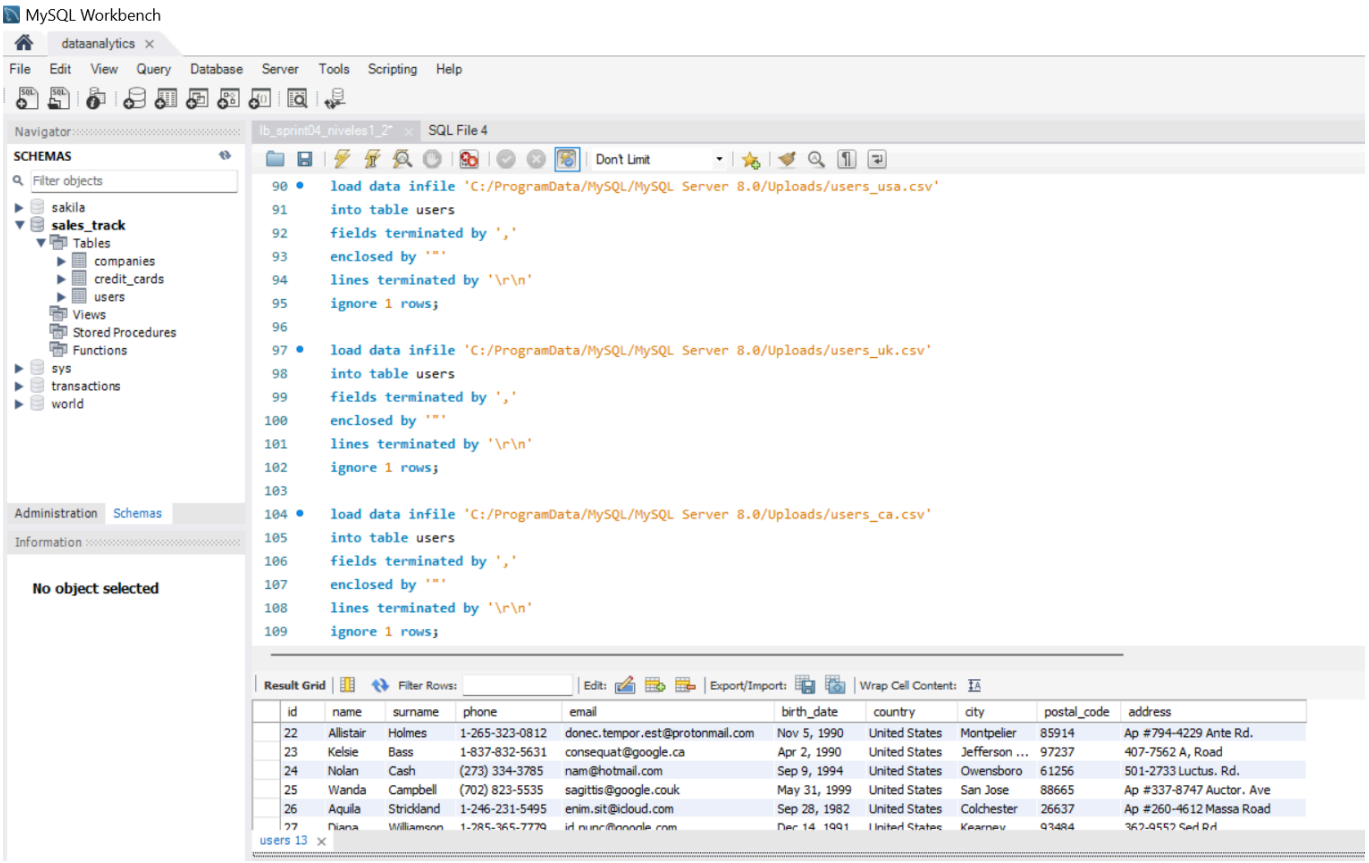
```
CREATE TABLE IF NOT EXISTS users (

    id SMALLINT UNSIGNED PRIMARY KEY,
    name VARCHAR (100 ),
    surname VARCHAR(100),
    phone VARCHAR(50),
    email VARCHAR (100),
    birth_date  VARCHAR (50),
    country VARCHAR(50),
    city VARCHAR(50),
    postal_code VARCHAR (15),
    address VARCHAR (255)
);

load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv'
into table users
fields terminated by ','
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
fields terminated by ','
enclosed by '"'
lines terminated by '\r\n'
ignore 1 rows;

load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv'
into table users
fields terminated by ','
enclosed by '"'
lines terminated by '\r\n'
ignore 1 rows;
```



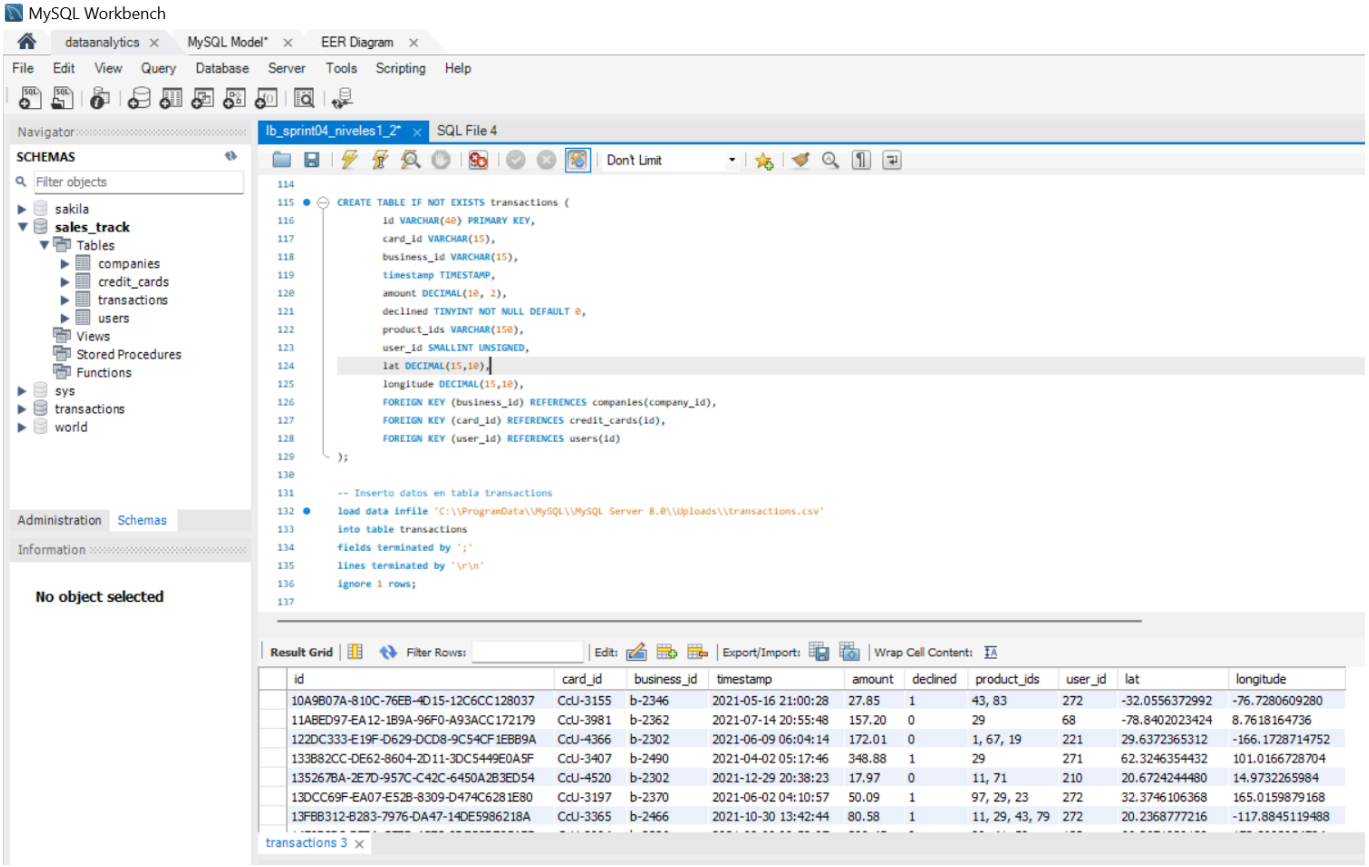
Creo tabla transactions con este script:

```
CREATE TABLE IF NOT EXISTS transactions (

 id VARCHAR(40) PRIMARY KEY,
	card_id VARCHAR(15),
	business_id VARCHAR(15),
	timestamp TIMESTAMP,
	amount DECIMAL(10, 2),
	declined TINYINT NOT NULL DEFAULT 0,
	product_ids VARCHAR(150),
	user_id SMALLINT UNSIGNED,
	lat DECIMAL(15,10),
	longitude DECIMAL(15,10),
	FOREIGN KEY (business_id) REFERENCES companies(company_id),
	FOREIGN KEY (card_id) REFERENCES credit_cards(id),
	FOREIGN KEY (user_id) REFERENCES users(id)
);
```

Inserto datos desde el archivo transactions.csv en tabla transactions

```
load data infile 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\transactions.csv'
into table transactions
fields terminated by ';'
lines terminated by '\r\n'
ignore 1 rows;
```

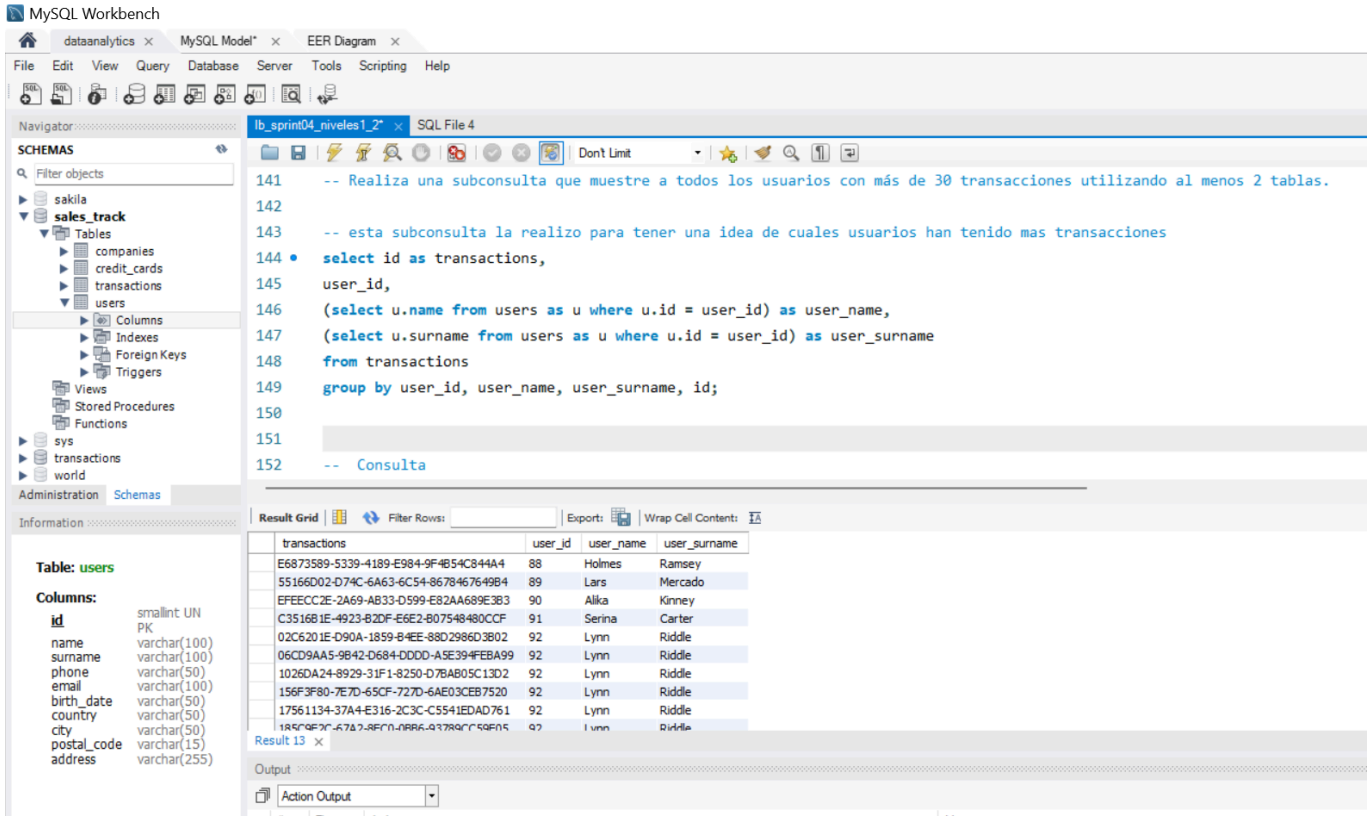


- Ejercicio 1

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

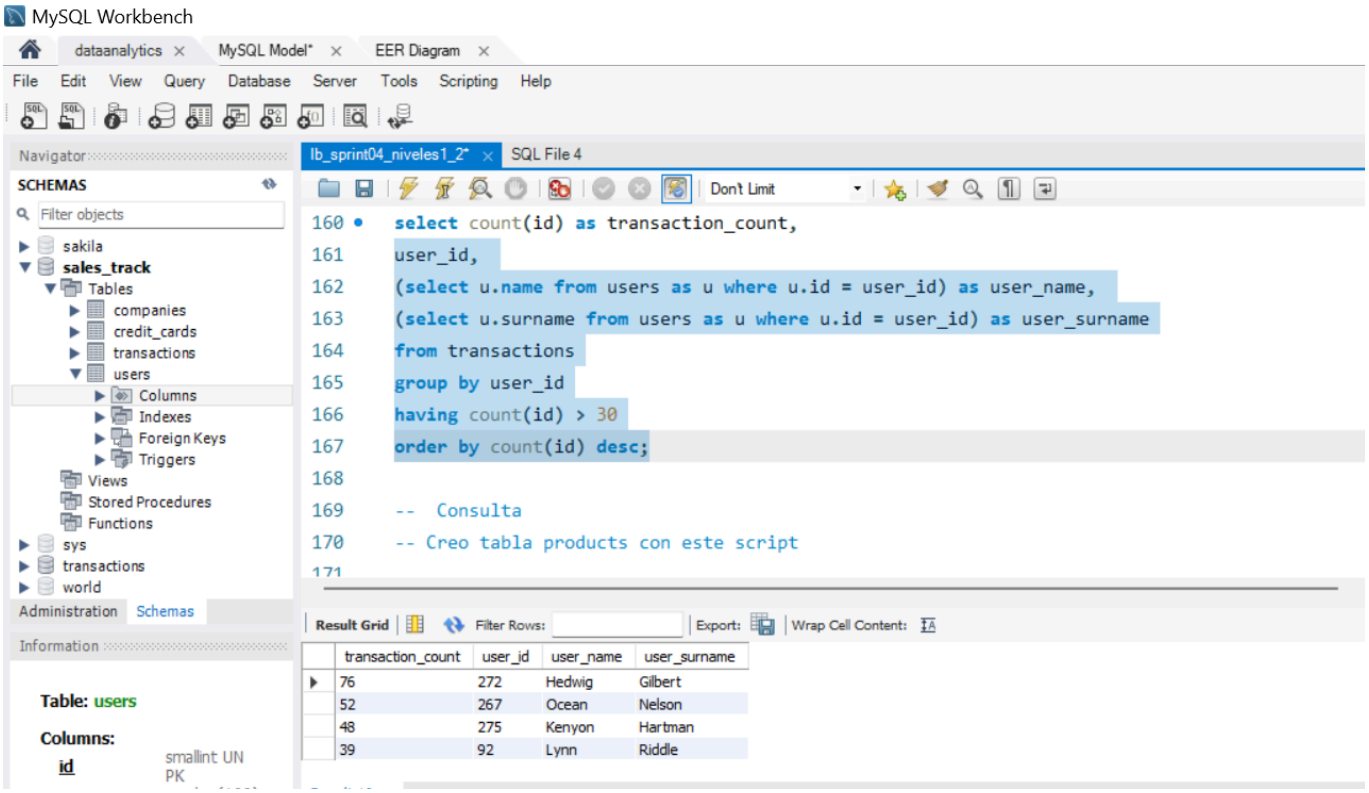
Esta subconsulta la realizo para tener una idea de cuales usuarios han tenido más transacciones

```
select id as transactions, user_id,
(select u.name from users as u where u.id = user_id) as user_name,
(select u.surname from users as u where u.id = user_id) as user_surname
from transactions
group by user_id, user_name, user_surname, id;
```



Ahora ajusto la consulta para que muestre a todos los usuarios con más de 30 transacciones de mayor a menor

```
select count(id) as transaction_count,
user_id,
(select u.name from users as u where u.id = user_id) as user_name,
(select u.surname from users as u where u.id = user_id) as user_surname
from transactions
group by user_id
having count(id) > 30
order by count(id) desc;
```

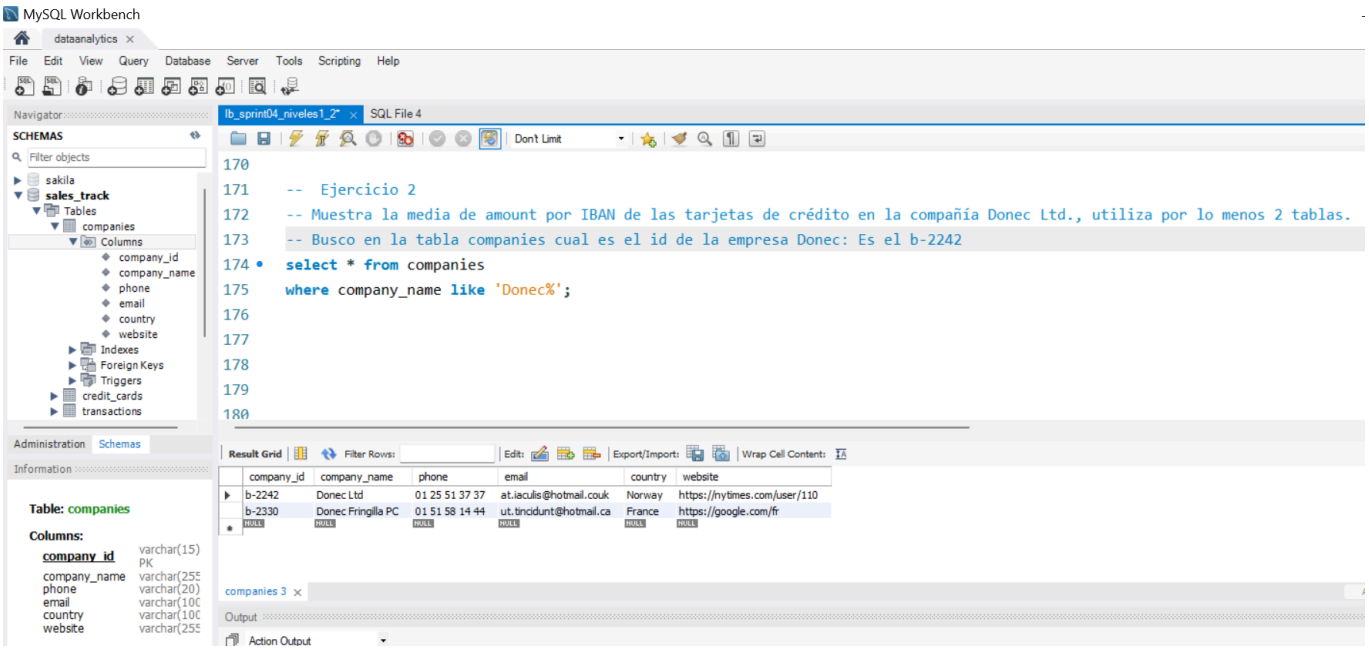


- Ejercicio 2

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

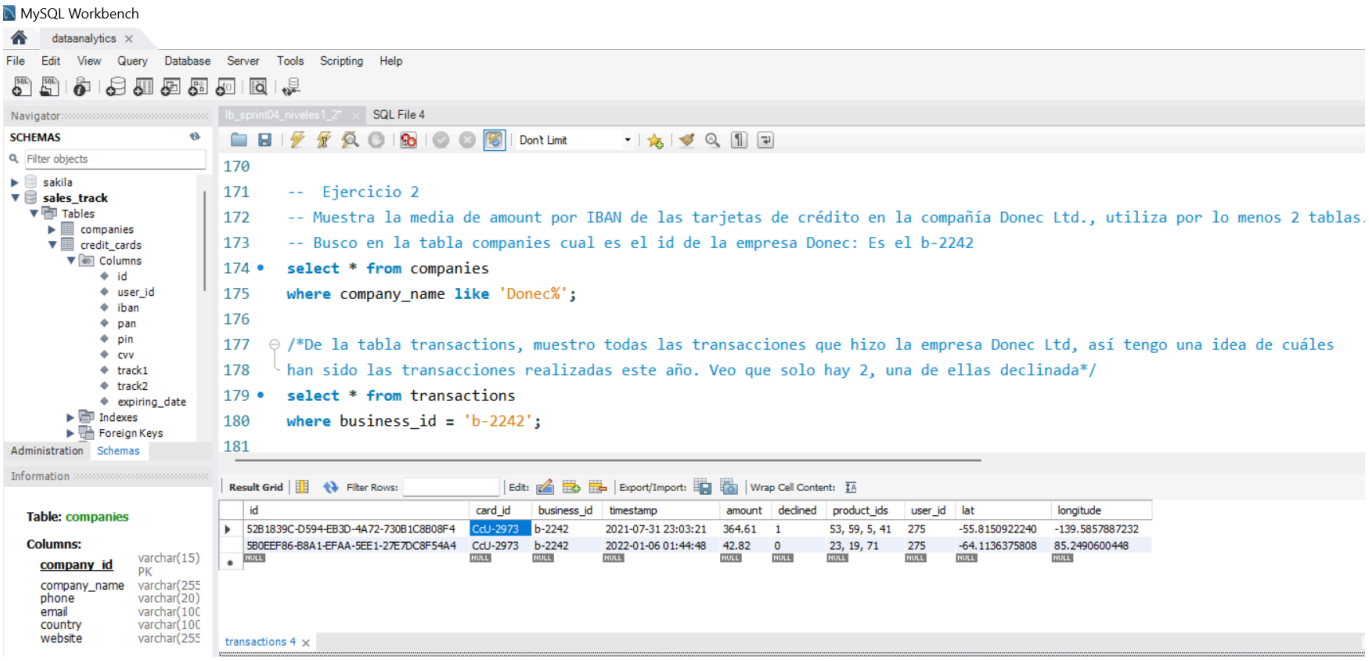
Busco en tabla companies el company_id de la empresa Donec Ltd: es el b-2242

```
select * from companies
where company_name like 'Donec%';
```



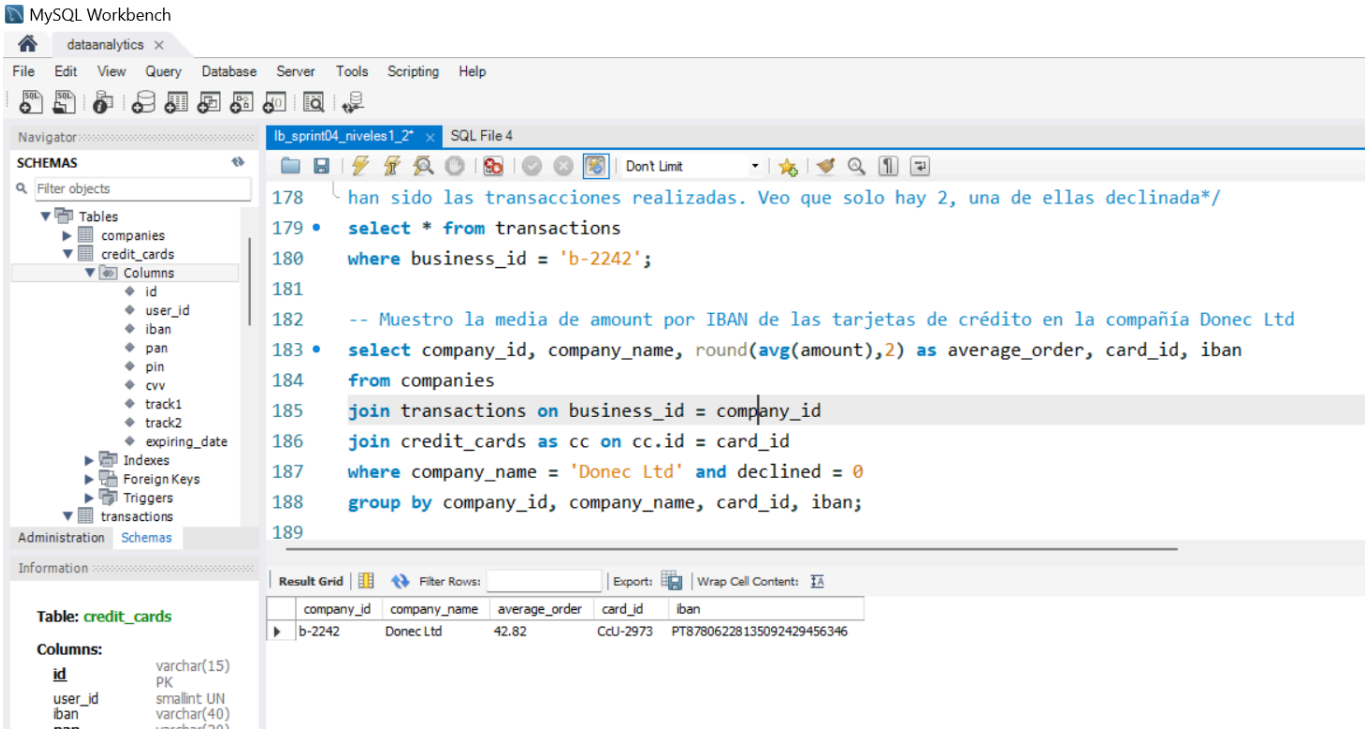
De la tabla transactions, muestro todas las transacciones que hizo la empresa Donec Ltd, así tengo una idea de cuáles han sido las transacciones realizadas este año. Veo que solo hay 2, una de ellas declinada

```
select * from transactions
where business_id = 'b-2242';
```



Muestro la media de amount por IBAN de las tarjetas de crédito en la compañía Donec Ltd

```
select company_id, company_name, round(avg(amount),2) as average_order, card_id, iban
from companies
join transactions on business_id = company_id
join credit_cards as cc on cc.id = card_id
where company_name = 'Donec Ltd' and declined = 0
group by company_id, company_name, card_id, iban;
```



Nivel 2

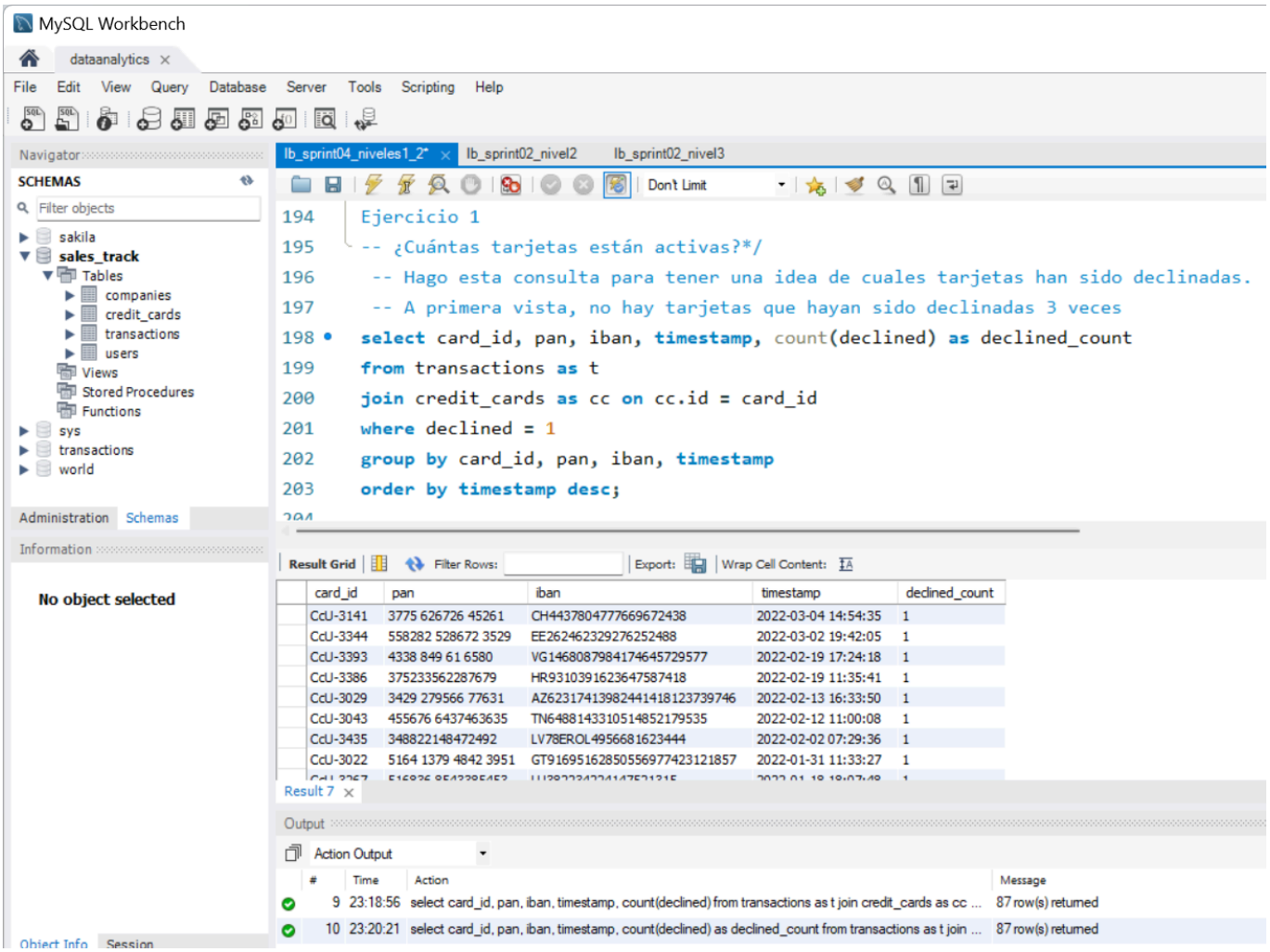
Crea una nueva tabla que refleje el estado de las tarjetas de crédito basado en si las últimas tres transacciones fueron declinadas y genera la siguiente consulta:

Ejercicio 1

¿Cuántas tarjetas están activas?

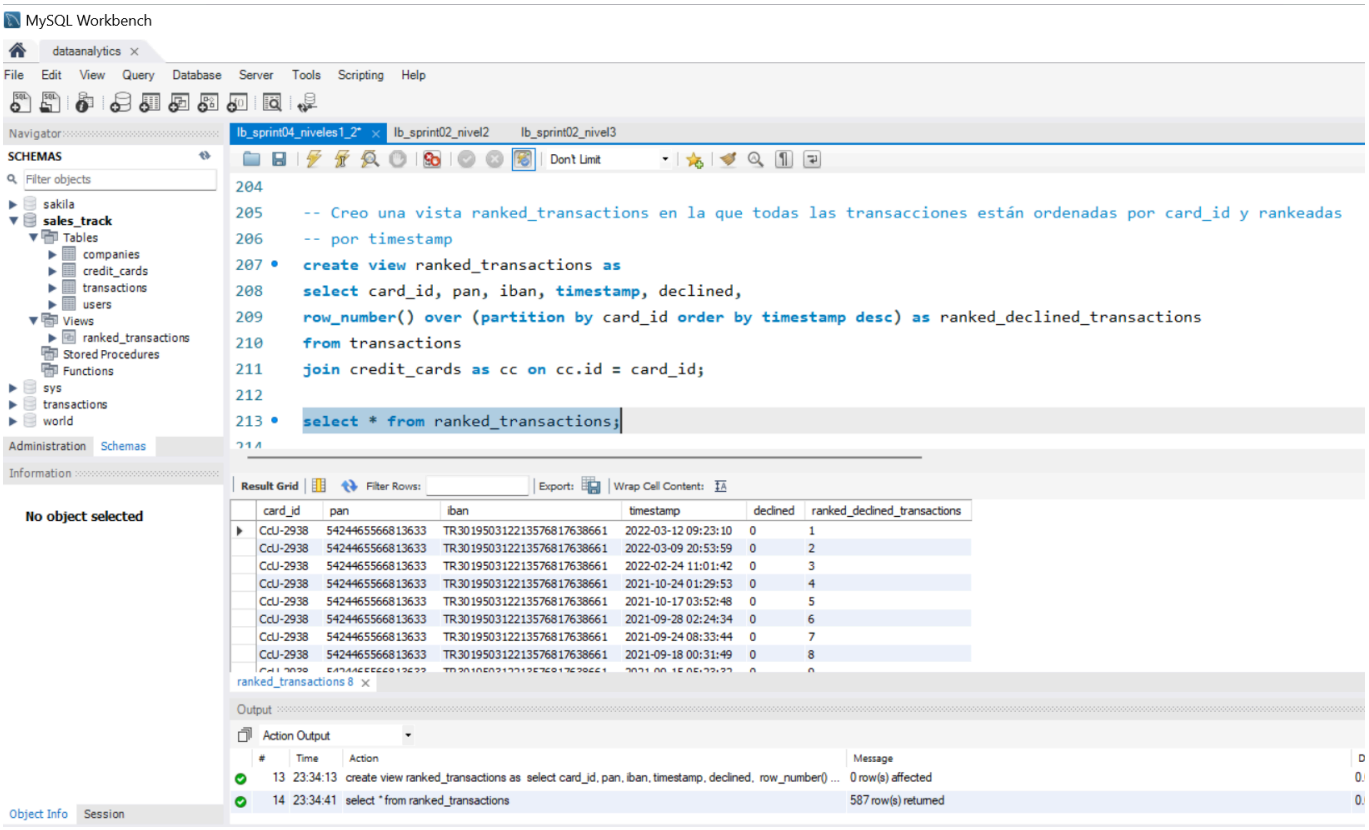
Hago esta consulta para tener una idea de cuáles tarjetas han sido declinadas. Hay 87 transacciones declinadas. A primera vista, no hay tarjetas que hayan sido declinadas 3 veces

```
select card_id, pan, iban, t.id as transaction_id, timestamp as transaction_time, count(declined) as declined_count
from transactions as t
join credit_cards as cc on cc.id = card_id
group by card_id, t.id, timestamp
having declined = 1
order by card_id, timestamp desc;
```



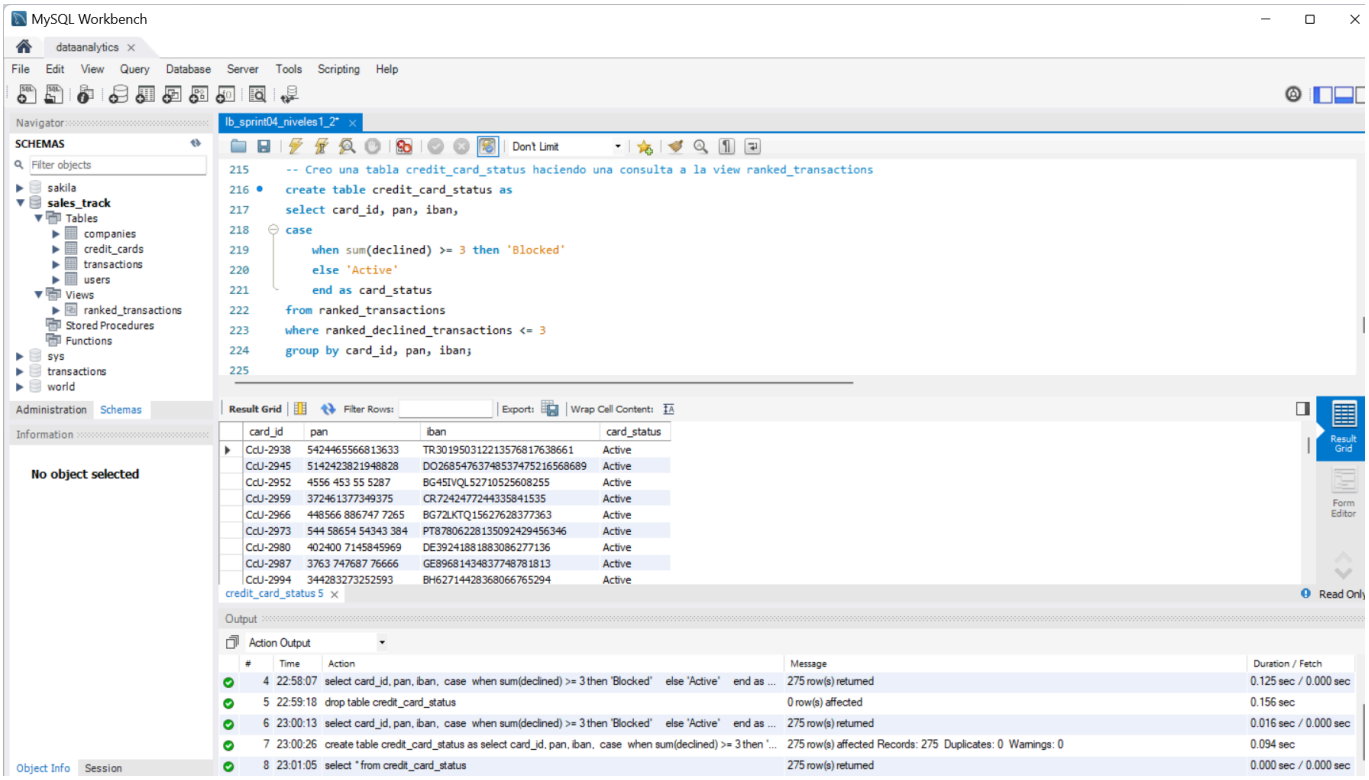
Creo una vista ranked_transactions en la que todas las transacciones están ordenadas por card_id y rankeadas por timestamp

```
create view ranked_transactions as
select card_id, pan, iban, timestamp, declined,
row_number() over (partition by card_id order by timestamp desc) as ranked_declined_transactions
from transactions
join credit_cards as cc on cc.id = card_id;
```



Creo una tabla credit_card_status haciendo una consulta desde la view ranked_transactions, en la que puedo ver todas las tarjetas de crédito y su estado: bloqueada o activa

```
create table credit_card_status as
select card_id, pan, iban,
case
        when sum(declined) >= 3 then 'Blocked'
        else 'Active'
end as card_status
from ranked_transactions
where ranked_declined_transactions <= 3
group by card_id, pan, iban;
```



-- Hago consulta de cuantas tarjetas están activas

```
select count(*) from credit_card_status
where card_status = 'Active';
```