

Level 1

Download the CSV files, study them and design a database with a star schema that contains at least 4 tables from which you can perform the following queries:

###CREATE DATABASE:

```
CREATE DATABASE star_schema_db;
```

```
USE star_schema_db;
```

#CREATE TABLE credit_cards

```
CREATE TABLE if not exists credit_cards (
```

```
    id VARCHAR(20) PRIMARY KEY,
```

```
    user_id VARCHAR(20),
```

```
    iban VARCHAR(100),
```

```
    pan VARCHAR(34),
```

```
    pin CHAR(20),
```

```
    cvv CHAR(4),
```

```
    track1 Varchar(255),
```

```
    track2 varchar(255),
```

```
    expiring_date VARCHAR(50)
```

```
);
```

#Cargar datos en la table credit_cards:

Table Data Import Wizard > Elegir el archivo credit_cards.csv> Next

#CREAR TABLA DE COMPANIES:

```
CREATE TABLE companies (  
  
    company_id VARCHAR(50) PRIMARY KEY,  
  
    company_name VARCHAR(255) NOT NULL,  
  
    phone VARCHAR(20),  
  
    email VARCHAR(150),  
  
    country VARCHAR(100),  
  
    website VARCHAR(255));
```

#Cargar los datos de tabla companies:

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server  
8.0/Uploads/companies.csv'
```

```
INTO TABLE companies
```

```
FIELDS TERMINATED BY ','
```

```
ENCLOSED BY '"'
```

```
LINES TERMINATED BY '\r\n'
```

```
IGNORE 1 ROWS;
```

#CREAR TABLA PRODUCTS:

```
CREATE TABLE IF NOT EXISTS products (  
  
    id INT PRIMARY KEY,
```

```
    product_name VARCHAR(255),
```

```
    price VARCHAR (34),
```

#CARGAR LOS DATOS EN PRODUCTS: (Import Data Table Wizard)

colour CHAR(7),

weight DECIMAL(5, 2),

warehouse_id VARCHAR(10));

#CREAR TABLA USERS:

CREATE TABLE users (

id INT PRIMARY KEY,

name VARCHAR(255),

surname VARCHAR(255),

phone VARCHAR(20),

email VARCHAR(255),

birth_date DATE,

country VARCHAR(100),

city VARCHAR(100),

postal_code VARCHAR(20),

address TEXT);

Alter table users

modify column birth_date varchar(34);

#CARGAR DATOS DE TABLA USERS:

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;

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_usa.csv'

INTO TABLE users

FIELDS TERMINATED BY ','

ENCLOSED BY ''''

LINES TERMINATED BY '\r\n'

IGNORE 1 ROWS;

#CREAR LA TABLA TRANSACTIONS:

CREATE TABLE transactions (

id VARCHAR(255) PRIMARY KEY,

```
card_id VARCHAR(50),  
  
business_id VARCHAR(50),  
  
timestamp DATETIME,  
  
amount DECIMAL(10, 2),  
  
declined BOOLEAN,  
  
product_ids VARCHAR(255),  
  
user_id INT,  
  
lat FLOAT,  
  
longitude FLOAT,  
  
FOREIGN KEY (card_id) REFERENCES credit_cards(id),  
  
FOREIGN KEY (user_id) REFERENCES users(id),  
  
FOREIGN KEY (business_id) REFERENCES companies(company_id));
```

#CARGAR LOS DATOS DE LA TABLA TRANSACTIONS:

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server  
8.0/Uploads/transactions.csv'
```

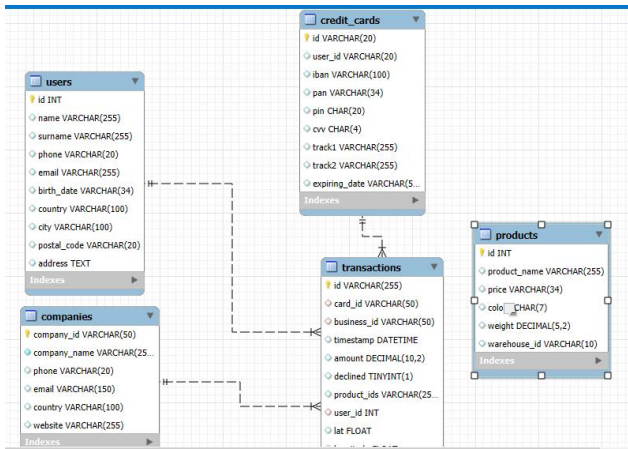
```
INTO TABLE transactions
```

```
FIELDS TERMINATED BY ','
```

```
ENCLOSED BY '"'
```

```
LINES TERMINATED BY '\r\n'
```

```
IGNORE 1 ROWS;
```



- Exercise 1

Perform a subquery that shows all users with more than 30 transactions using at least 2 tables.

```

SELECT u.id AS user_id, u.name, u.surname, transaction_count.count
FROM users u
join (select t.user_id, count(t.id) as count from transactions t
      where declined=0
      group by t.user_id
      having count(t.id)>30) as transaction_count
ON u.id= transaction_count.user_id;
  
```

Don't Limit

```

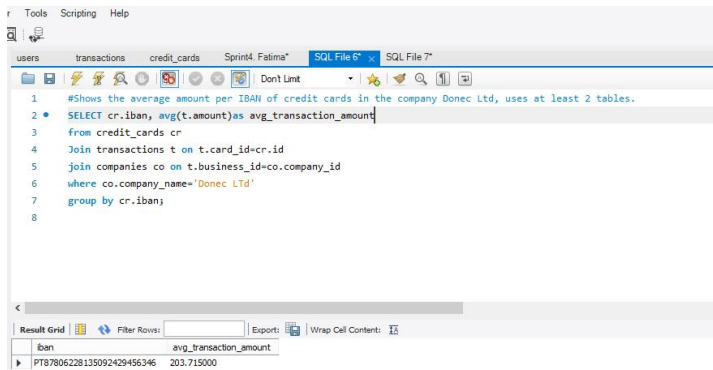
12 GROUP BY
13     u.id, u.name, u.surname
14 HAVING
15     COUNT(t.id) > 30; -- کاربران با تعداد 30+ تراکنش موفق
16
17 select u.id AS user_id, u.name, u.surname, transaction_count.count
18 FROM users u
19 join (select t.user_id, count(t.id) as count from transactions t
20       where declined=0
21       group by t.user_id
22       having count(t.id)>30) as transaction_count on u.id= transaction_count.user_id;
23
  
```

Result Grid | Filter Rows: | Exports: | Wrap Cell Content: IS

	user_id	name	surname	count
▶	92	Lynn	Riddle	39
	267	Ocean	Nelson	39
	272	Hedwig	Gilbert	38

- Exercise 2

Shows the average amount per IBAN of credit cards in the company Donec Ltd, uses at least 2 tables.



The screenshot shows a SQL IDE with a query editor and a result grid. The query editor contains the following SQL code:

```
1 #Shows the average amount per IBAN of credit cards in the company Donec Ltd, uses at least 2 tables.
2 • SELECT cr.iban, avg(t.amount) as avg_transaction_amount
3   from credit_cards cr
4   join transactions t on t.card_id=cr.id
5   join companies co on t.business_id=co.company_id
6   where co.company_name='Donec Ltd'
7   group by cr.iban;
```

The result grid shows the following data:

iban	avg_transaction_amount
PT87806228135092429456346	203.715000

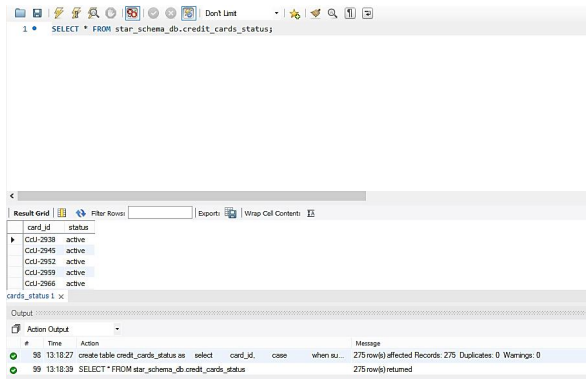
Level 2

Create a new table that reflects the status of credit cards based on whether the last three transactions were declined and generate the following query:

Exercise 1

How many cards are active?

```
#M2-E1: Create a new table that reflects the status of credit cards based on whether the last three transactions
# were declined and generate the following query:How many cards are active?
• create table credit_cards_status as
  select
    card_id,
    case
      when sum(declined) = 3 then 'inactive'
      else 'active'
    end as status
  from
    (select
      card_id,
      timestamp,
      declined,
      row_number() over(partition by card_id order by timestamp desc ) as date # enumera de 1 a N los casos segun el orden
    from star_schema_db.transactions)
  as popular
  where date <=3
```



Level 3

Create a table with which we can join the data from the new products.csv file with the created database, taking into account that from transaction you have product_ids. Generate the following query:

Exercise 1

We need to know the number of times each product has been sold.