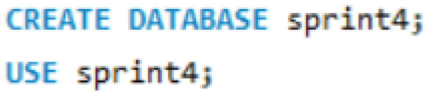
SPRINT 4

Nivell 1

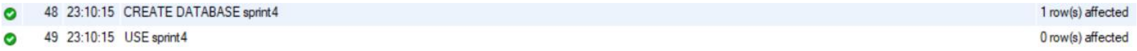
Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules

Input 1 Creación base de datos



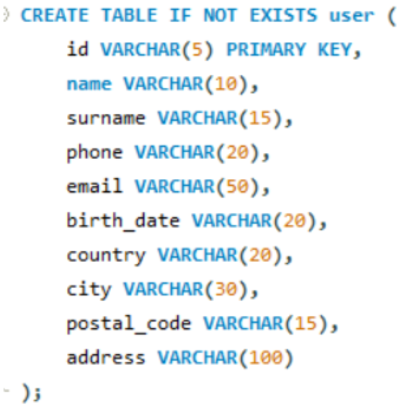
```
CREATE DATABASE sprint4;
USE sprint4;
```

Output 1

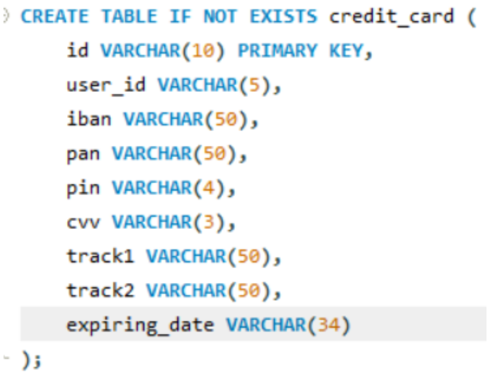


```
48  23:10:15  CREATE DATABASE sprint4        1 row(s) affected
49  23:10:15  USE sprint4                    0 row(s) affected
```

Input 2: Creación de tablas

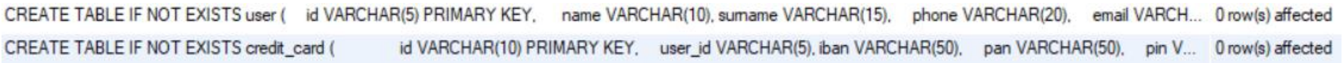


```
CREATE TABLE IF NOT EXISTS user (
    id VARCHAR(5) PRIMARY KEY,
    name VARCHAR(10),
    surname VARCHAR(15),
    phone VARCHAR(20),
    email VARCHAR(50),
    birth_date VARCHAR(20),
    country VARCHAR(20),
    city VARCHAR(30),
    postal_code VARCHAR(15),
    address VARCHAR(100)
);
```



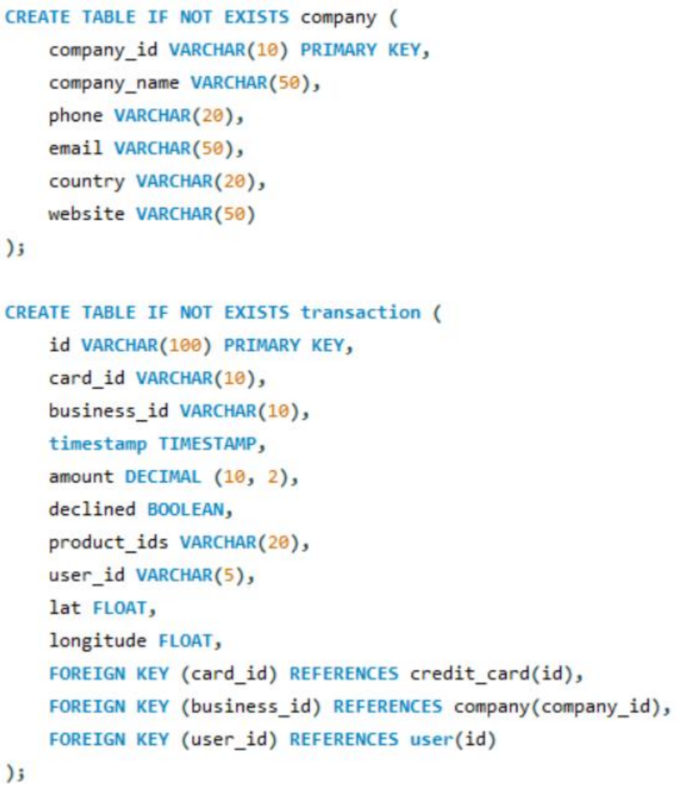
```
CREATE TABLE IF NOT EXISTS credit_card (
    id VARCHAR(10) PRIMARY KEY,
    user_id VARCHAR(5),
    iban VARCHAR(50),
    pan VARCHAR(50),
    pin VARCHAR(4),
    cvv VARCHAR(3),
    track1 VARCHAR(50),
    track2 VARCHAR(50),
    expiring_date VARCHAR(34)
);
```

Output 2:



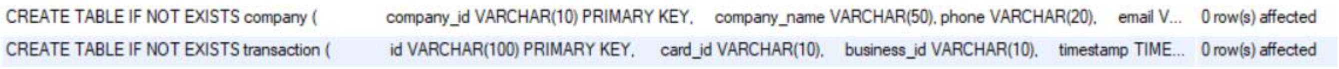
```
CREATE TABLE IF NOT EXISTS user (    id VARCHAR(5) PRIMARY KEY,    name VARCHAR(10), surname VARCHAR(15),    phone VARCHAR(20),    email VARCH...   0 row(s) affected
CREATE TABLE IF NOT EXISTS credit_card (         id VARCHAR(10) PRIMARY KEY,    user_id VARCHAR(5), iban VARCHAR(50),    pan VARCHAR(50),    pin V...   0 row(s) affected
```

Input 3: Más tablas



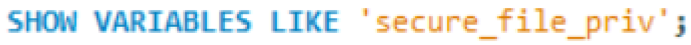
```
CREATE TABLE IF NOT EXISTS company (
    company_id VARCHAR(10) PRIMARY KEY,
    company_name VARCHAR(50),
    phone VARCHAR(20),
    email VARCHAR(50),
    country VARCHAR(20),
    website VARCHAR(50)
);

CREATE TABLE IF NOT EXISTS transaction (
    id VARCHAR(100) PRIMARY KEY,
    card_id VARCHAR(10),
    business_id VARCHAR(10),
    timestamp TIMESTAMP,
    amount DECIMAL (10, 2),
    declined BOOLEAN,
    product_ids VARCHAR(20),
    user_id VARCHAR(5),
    lat FLOAT,
    longitude FLOAT,
    FOREIGN KEY (card_id) REFERENCES credit_card(id),
    FOREIGN KEY (business_id) REFERENCES company(company_id),
    FOREIGN KEY (user_id) REFERENCES user(id)
);
```

Output 3:



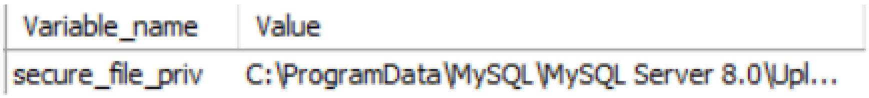
```
CREATE TABLE IF NOT EXISTS company (        company_id VARCHAR(10) PRIMARY KEY,    company_name VARCHAR(50), phone VARCHAR(20),    email V...    0 row(s) affected
CREATE TABLE IF NOT EXISTS transaction (     id VARCHAR(100) PRIMARY KEY,    card_id VARCHAR(10),    business_id VARCHAR(10),    timestamp TIME...    0 row(s) affected
```

Input 4: Localizar carpeta donde colocar CSV’s para cargarlos en las tablas.

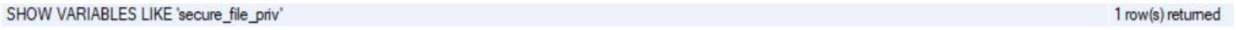


```
SHOW VARIABLES LIKE 'secure_file_priv';
```

Output 4:

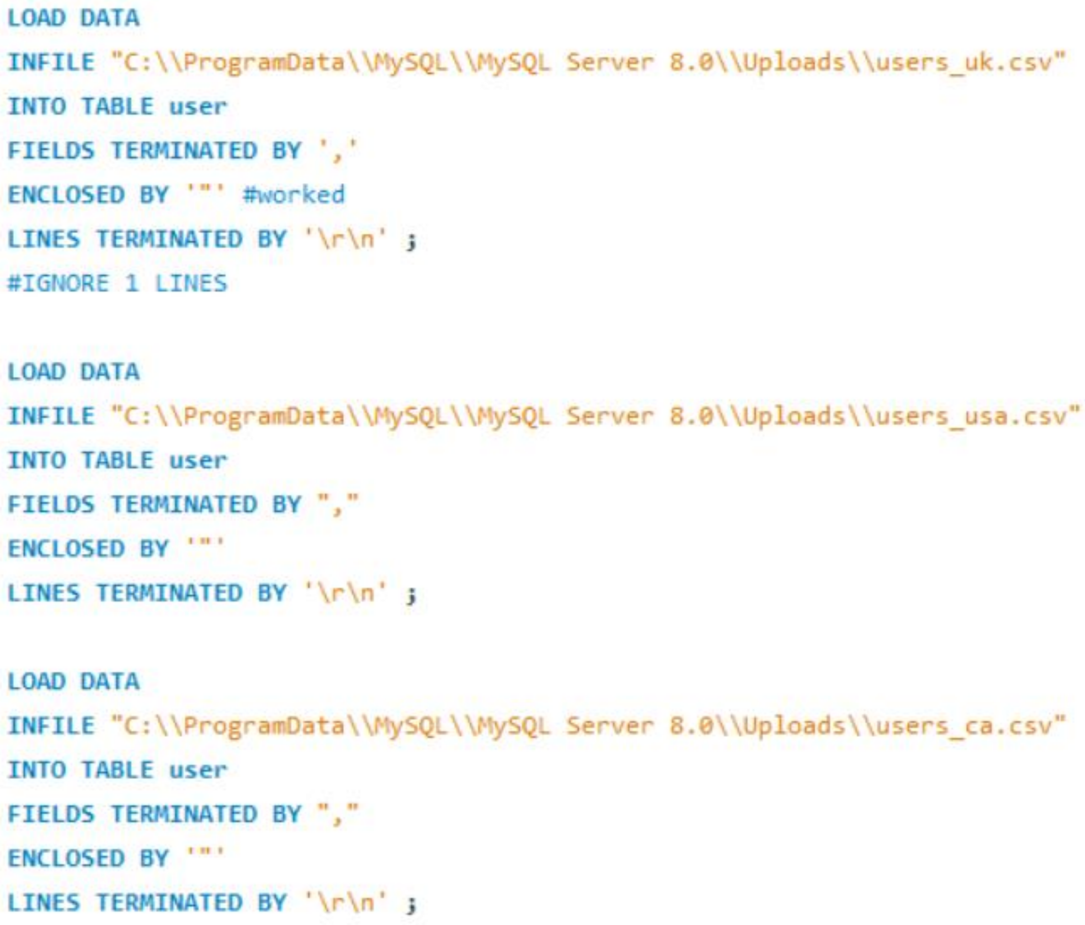


Variable_name	Value
secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.0\Upl...



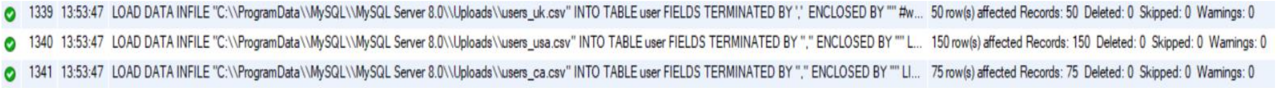
```
SHOW VARIABLES LIKE 'secure_file_priv'                                                1 row(s) returned
```

Input 5: Importo los datos de users a la tabla user



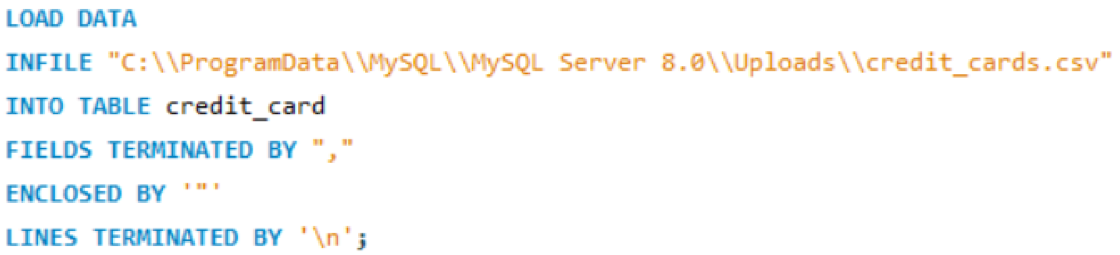
```
LOAD DATA
INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_uk.csv"
INTO TABLE user
FIELDS TERMINATED BY ','
ENCLOSED BY '"' #worked
LINES TERMINATED BY '\r\n' ;
#IGNORE 1 LINES

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

LOAD DATA
INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_ca.csv"
INTO TABLE user
FIELDS TERMINATED BY ","
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n' ;
```

Output 5:



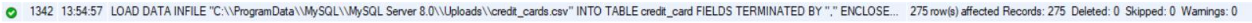
```
1339  13:53:47  LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_uk.csv" INTO TABLE user FIELDS TERMINATED BY ','  ENCLOSED BY '"' #w...   50 row(s) affected Records: 50  Deleted: 0  Skipped: 0  Warnings: 0
1340  13:53:47  LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_usa.csv" INTO TABLE user FIELDS TERMINATED BY "," ENCLOSED BY '"' L...   150 row(s) affected Records: 150  Deleted: 0  Skipped: 0  Warnings: 0
1341  13:53:47  LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_ca.csv" INTO TABLE user FIELDS TERMINATED BY "," ENCLOSED BY '"' LI...   75 row(s) affected Records: 75  Deleted: 0  Skipped: 0  Warnings: 0
```

Input 6: Credit_cards se importa con '\n' porque se basa en la arquitectura de archivo de Linux, a diferencia del resto de archivos que hemos importado que se basan en Windows. Esto se detecta al abrir los archivos en notepad++.



```
LOAD DATA
INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\credit_cards.csv"
INTO TABLE credit_card
FIELDS TERMINATED BY ","
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Output 6:

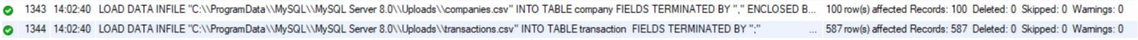


```
1342  13:54:57  LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\credit_cards.csv" INTO TABLE credit_card FIELDS TERMINATED BY "," ENCLOSE...   275 row(s) affected Records: 275  Deleted: 0  Skipped: 0  Warnings: 0
```

Input 7:

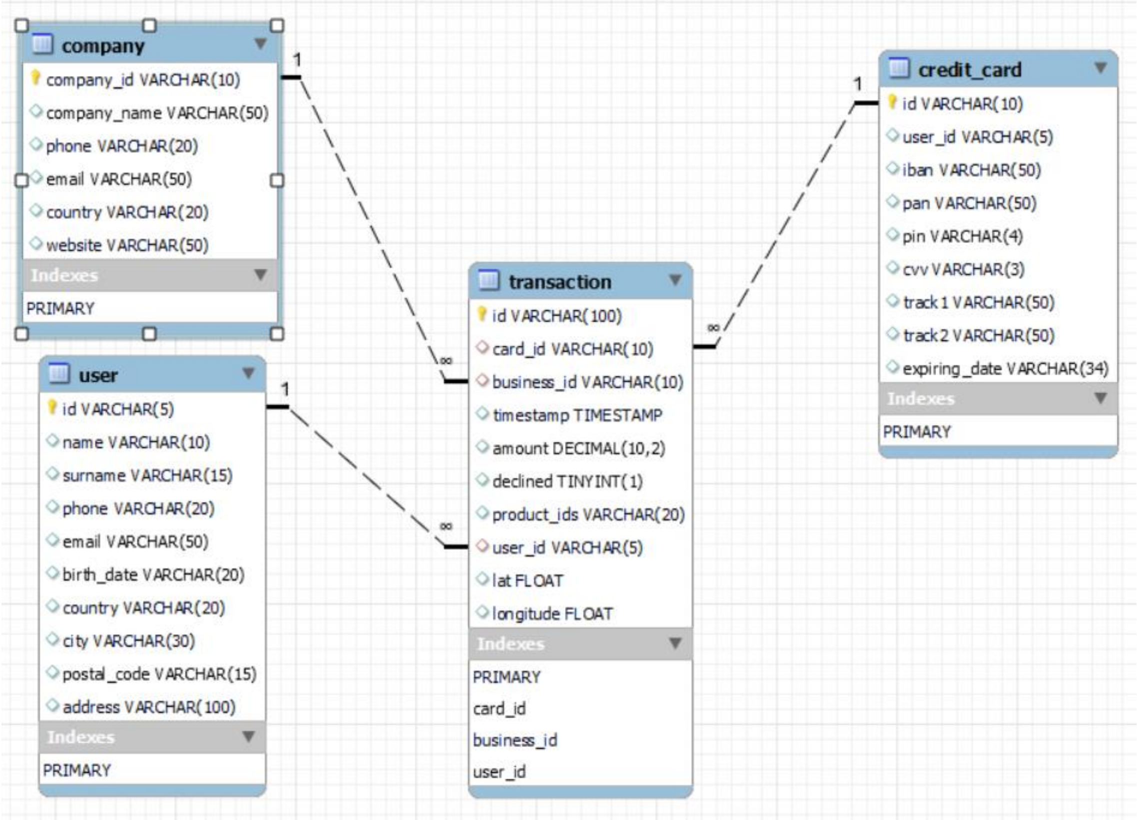
```
LOAD DATA
INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\companies.csv"
INTO TABLE company
FIELDS TERMINATED BY ","
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n' ; #worked

#Tabla 4
LOAD DATA
INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\transactions.csv"
INTO TABLE transaction
FIELDS TERMINATED BY ";"
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n' ;
```

Output 7:



```
1343  14:02:40  LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\companies.csv" INTO TABLE company FIELDS TERMINATED BY "," ENCLOSED B...   100 row(s) affected Records: 100  Deleted: 0  Skipped: 0  Warnings: 0
1344  14:02:40  LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\transactions.csv" INTO TABLE transaction  FIELDS TERMINATED BY ";"      ...   587 row(s) affected Records: 587  Deleted: 0  Skipped: 0  Warnings: 0
```

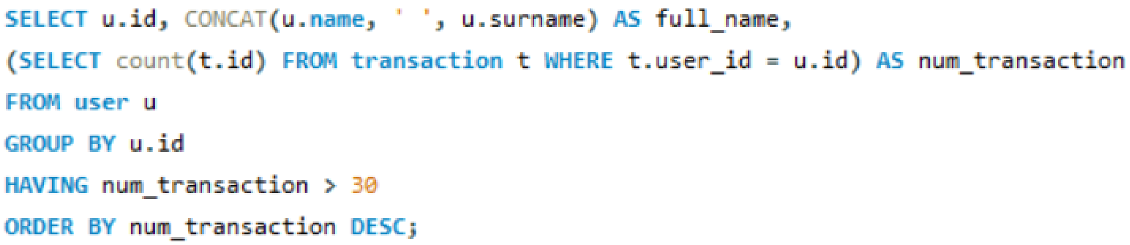
Tablas resultantes:



Ejercicio 1

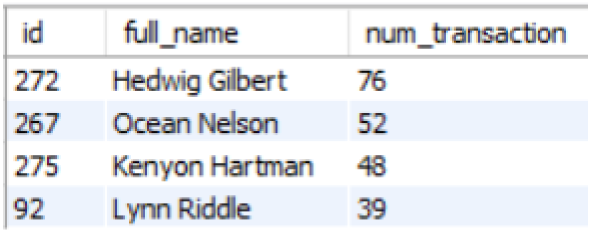
Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.

Input 1: Concateno el nombre y el apellido. Uso una subquery con SELECT para conectar las tablas user y transaction, lo que me permite usar la columna t.id que identifica cada transacción. A continuación, agrupo por id de cliente, y al final realizo HAVING para limitar las agrupaciones a aquellas que tienen más de 30 transacciones.

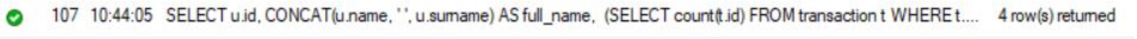


```
SELECT u.id, CONCAT(u.name, ' ', u.surname) AS full_name,
(SELECT count(t.id) FROM transaction t WHERE t.user_id = u.id) AS num_transaction
FROM user u
GROUP BY u.id
HAVING num_transaction > 30
ORDER BY num_transaction DESC;
```

Output 1:



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

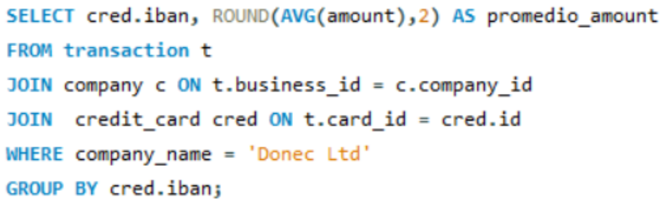


```
✓ 107 10:44:05 SELECT u.id, CONCAT(u.name, ' ', u.surname) AS full_name,  (SELECT count(t.id) FROM transaction t WHERE t....   4 row(s) returned
```

Ejercicio 2

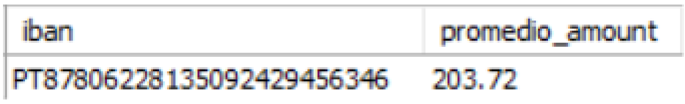
Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.

Input 1: Uso JOIN's para unir 3 tablas (transaction, company y credit_card), filtro por company_name con el WHERE y agrupo el promedio de amount por iban.

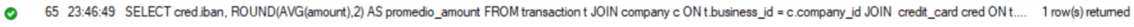


```
SELECT cred.iban, ROUND(AVG(amount),2) AS promedio_amount
FROM transaction t
JOIN company c ON t.business_id = c.company_id
JOIN  credit_card cred ON t.card_id = cred.id
WHERE company_name = 'Donec Ltd'
GROUP BY cred.iban;
```

Output 1:



iban	promedio_amount
PT87806228135092429456346	203.72



```
✓ 65 23:46:49 SELECT cred.iban, ROUND(AVG(amount),2) AS promedio_amount FROM transaction t JOIN company c ON t.business_id = c.company_id JOIN  credit_card cred ON t....   1 row(s) returned
```

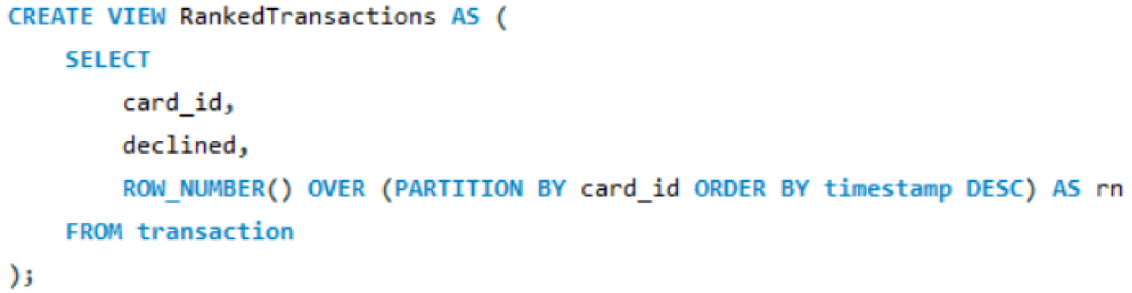
Nivell 2

Ejercicio 1

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:

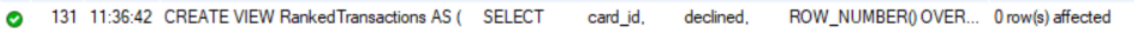
Quantes targetes estan actives?

Input 1: Primera vista en la que creo la partición por card_id ordenada por la fecha-hora



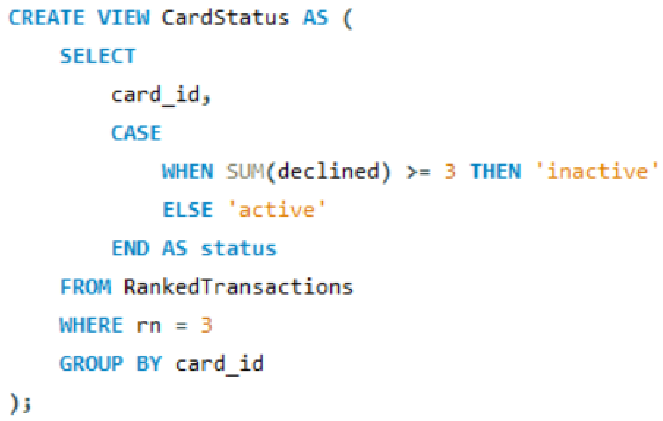
```
CREATE VIEW RankedTransactions AS (
    SELECT
        card_id,
        declined,
        ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS rn
    FROM transaction
);
```

Output 1



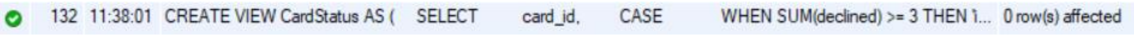
```
131  11:36:42  CREATE VIEW RankedTransactions AS (   SELECT      card_id,      declined,      ROW_NUMBER() OVER...  0 row(s) affected
```

Input 2 Esta vista hace un CASE donde si la suma de DECLINED > =3 se convierte en inactive. Más adelante filtro con el WHERE por las rn = 3 , que representan las ultimas tres transacciones de cada card_id.



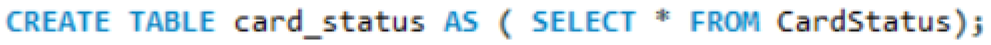
```
CREATE VIEW CardStatus AS (
    SELECT
        card_id,
        CASE
            WHEN SUM(declined) >= 3 THEN 'inactive'
            ELSE 'active'
        END AS status
    FROM RankedTransactions
    WHERE rn = 3
    GROUP BY card_id
);
```

Output 2



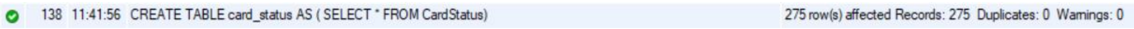
```
132  11:38:01  CREATE VIEW CardStatus AS (   SELECT      card_id,      CASE      WHEN SUM(declined) >= 3 THEN '...  0 row(s) affected
```

Input 3: Creación de la tabla basada en la última vista



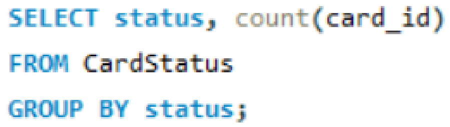
```
CREATE TABLE card_status AS ( SELECT * FROM CardStatus);
```

Output 3:



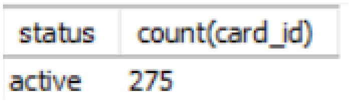
```
138  11:41:56  CREATE TABLE card_status AS ( SELECT * FROM CardStatus)     275 row(s) affected Records: 275  Duplicates: 0  Warnings: 0
```

Input 4: Comprobación de cuantas tarjetas están en cada estado

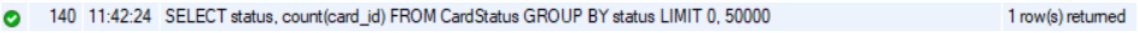


```
SELECT status, count(card_id)
FROM CardStatus
GROUP BY status;
```

Output 4:



status	count(card_id)
active	275



```
140  11:42:24  SELECT status, count(card_id) FROM CardStatus GROUP BY status LIMIT 0, 50000    1 row(s) returned
```