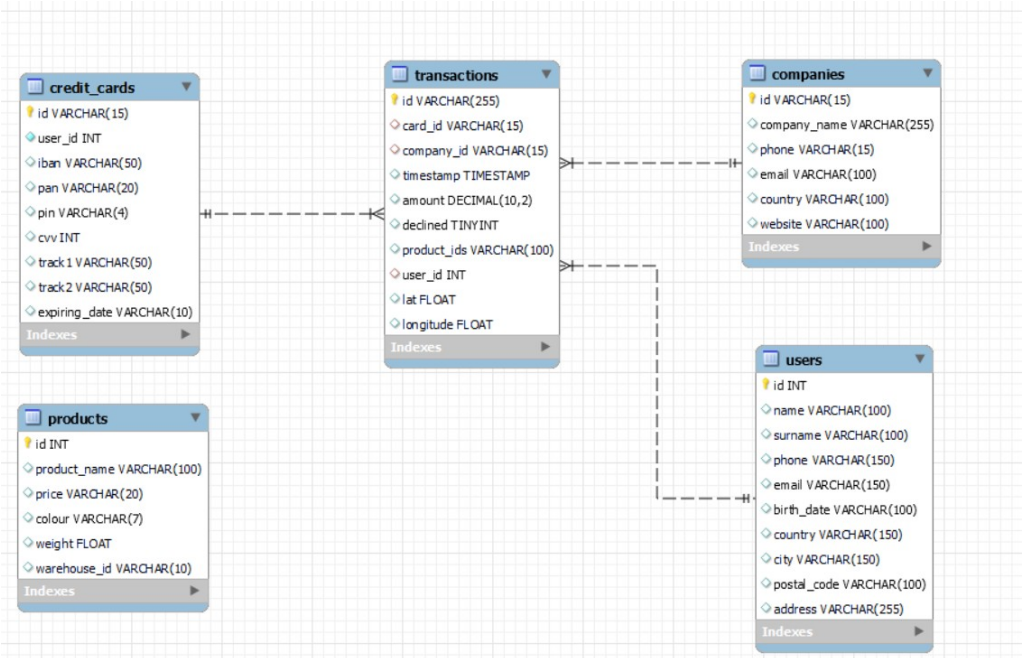
Tasca S4.01. Creació de Base de Dades

Nivell 1

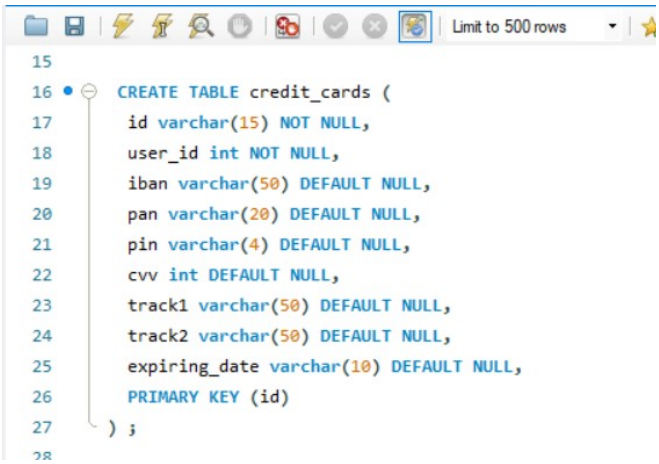
Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguis realitzar les següents consultes:

0.1 Esquema

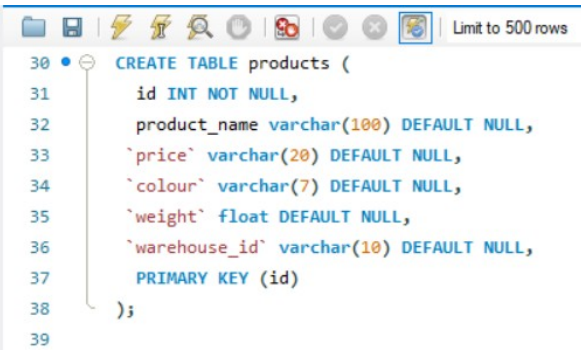


0.2 Creació de la base de dades

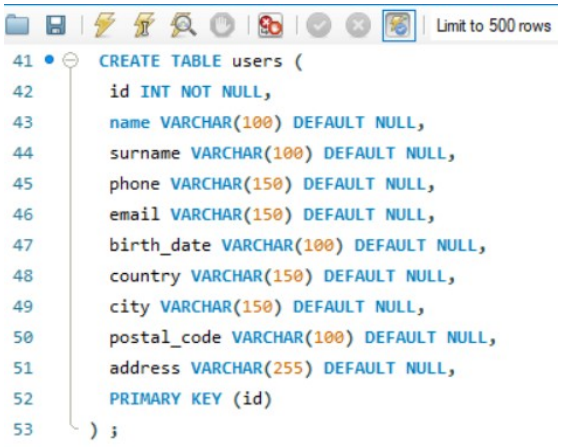
```sql
CREATE database new_transactions;

USE new_transactions;

CREATE TABLE companies (
    id VARCHAR(15) NOT NULL,
    company_name VARCHAR(255) DEFAULT NULL,
    phone VARCHAR(15) DEFAULT NULL,
    email VARCHAR(100) DEFAULT NULL,
    country VARCHAR(100) DEFAULT NULL,
    website VARCHAR(100) DEFAULT NULL,
    PRIMARY KEY (id)
);
```



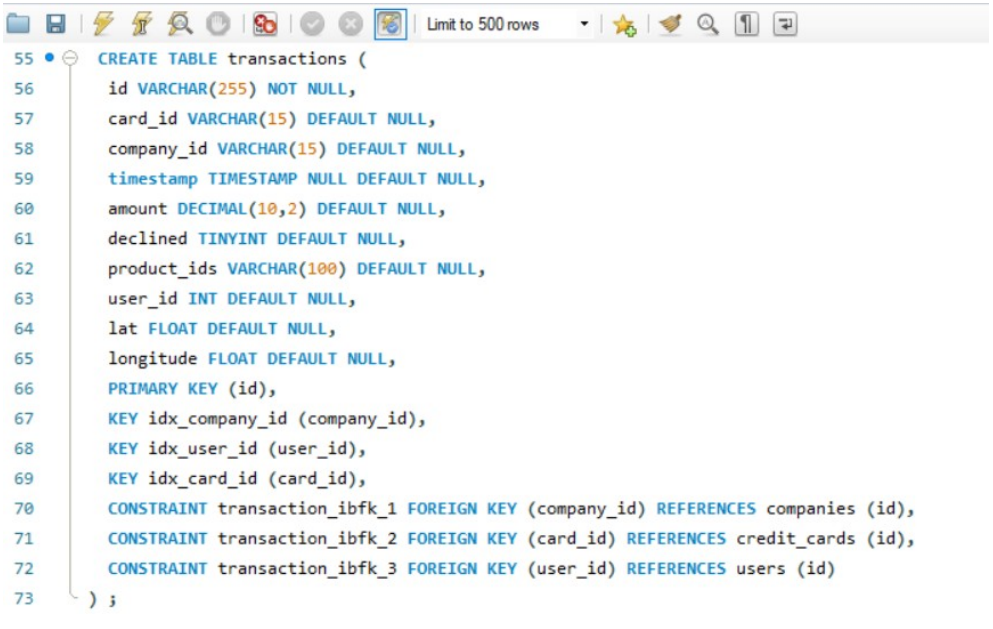
```
15
16  CREATE TABLE credit_cards (
17    id varchar(15) NOT NULL,
18    user_id int NOT NULL,
19    iban varchar(50) DEFAULT NULL,
20    pan varchar(20) DEFAULT NULL,
21    pin varchar(4) DEFAULT NULL,
22    cvv int DEFAULT NULL,
23    track1 varchar(50) DEFAULT NULL,
24    track2 varchar(50) DEFAULT NULL,
25    expiring_date varchar(10) DEFAULT NULL,
26    PRIMARY KEY (id)
27  ) ;
28
```



```
30  CREATE TABLE products (
31    id INT NOT NULL,
32    product_name varchar(100) DEFAULT NULL,
33   `price` varchar(20) DEFAULT NULL,
34   `colour` varchar(7) DEFAULT NULL,
35   `weight` float DEFAULT NULL,
36   `warehouse_id` varchar(10) DEFAULT NULL,
37    PRIMARY KEY (id)
38  );
39
```



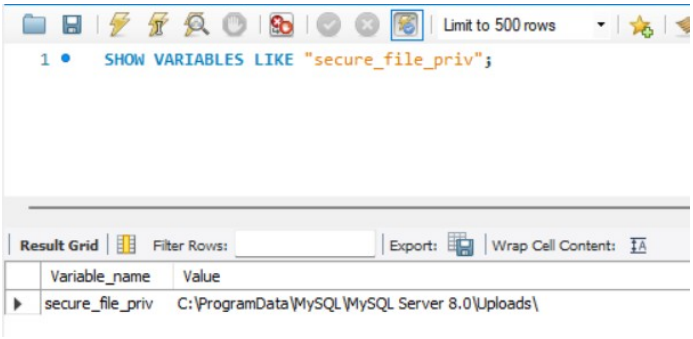
```
41  CREATE TABLE users (
42    id INT NOT NULL,
43    name VARCHAR(100) DEFAULT NULL,
44    surname VARCHAR(100) DEFAULT NULL,
45    phone VARCHAR(150) DEFAULT NULL,
46    email VARCHAR(150) DEFAULT NULL,
47    birth_date VARCHAR(100) DEFAULT NULL,
48    country VARCHAR(150) DEFAULT NULL,
49    city VARCHAR(150) DEFAULT NULL,
50    postal_code VARCHAR(100) DEFAULT NULL,
51    address VARCHAR(255) DEFAULT NULL,
52    PRIMARY KEY (id)
53  ) ;
```



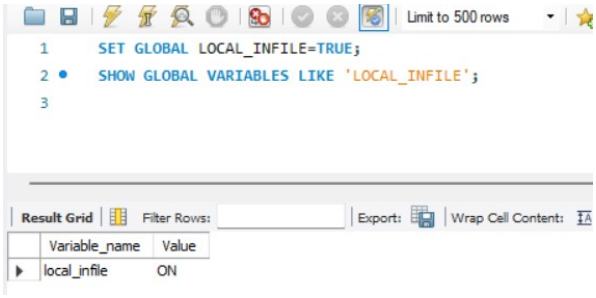
```
55  CREATE TABLE transactions (
56    id VARCHAR(255) NOT NULL,
57    card_id VARCHAR(15) DEFAULT NULL,
58    company_id VARCHAR(15) DEFAULT NULL,
59    timestamp TIMESTAMP NULL DEFAULT NULL,
60    amount DECIMAL(10,2) DEFAULT NULL,
61    declined TINYINT DEFAULT NULL,
62    product_ids VARCHAR(100) DEFAULT NULL,
63    user_id INT DEFAULT NULL,
64    lat FLOAT DEFAULT NULL,
65    longitude FLOAT DEFAULT NULL,
66    PRIMARY KEY (id),
67    KEY idx_company_id (company_id),
68    KEY idx_user_id (user_id),
69    KEY idx_card_id (card_id),
70    CONSTRAINT transaction_ibfk_1 FOREIGN KEY (company_id) REFERENCES companies (id),
71    CONSTRAINT transaction_ibfk_2 FOREIGN KEY (card_id) REFERENCES credit_cards (id),
72    CONSTRAINT transaction_ibfk_3 FOREIGN KEY (user_id) REFERENCES users (id)
73  ) ;
```

0.3 Seguretat de la base de dades per poder pujar els fitxers

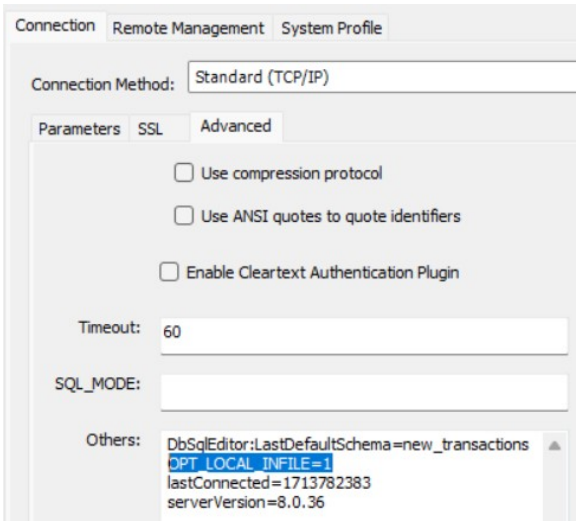
0.3.1 On deixar els fitxers



0.3.2 Canviar variable global

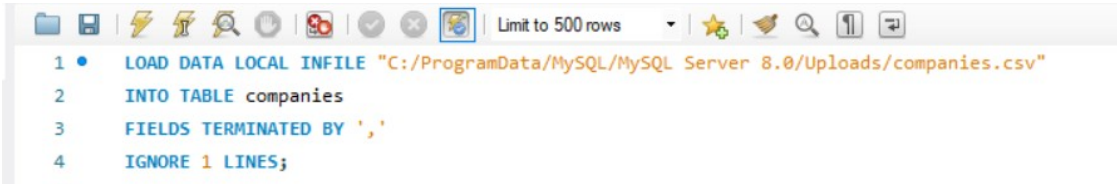


0.3.3 Afegir OPT_LOCAL_INFILE a la connexió



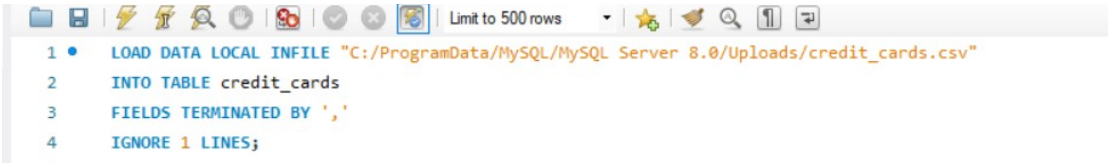
0.4 Carregar els fitxers CSV

Companies



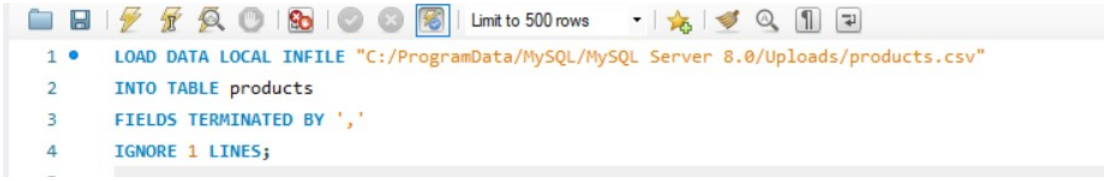
```
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv"
INTO TABLE companies
FIELDS TERMINATED BY ','
IGNORE 1 LINES;
```

Credit_cards



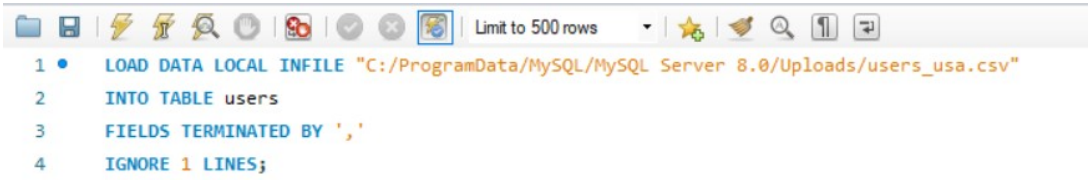
```
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv"
INTO TABLE credit_cards
FIELDS TERMINATED BY ','
IGNORE 1 LINES;
```

Products



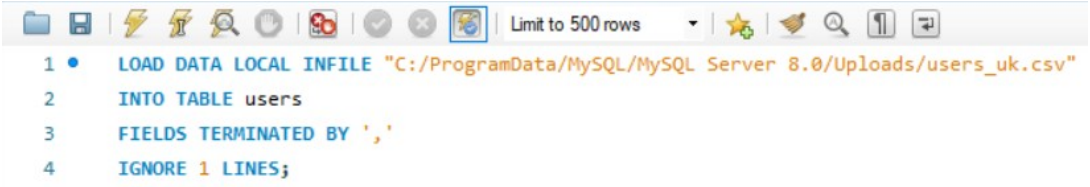
```
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv"
INTO TABLE products
FIELDS TERMINATED BY ','
IGNORE 1 LINES;
```

Users



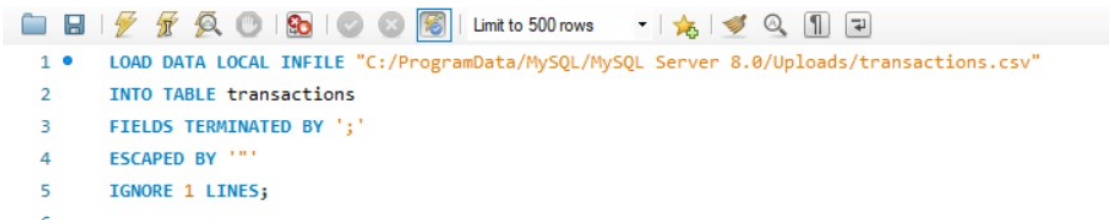
```
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv"
INTO TABLE users
FIELDS TERMINATED BY ','
IGNORE 1 LINES;
```

```
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv"
INTO TABLE users
FIELDS TERMINATED BY ','
IGNORE 1 LINES;
```



```
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv"
INTO TABLE users
FIELDS TERMINATED BY ','
IGNORE 1 LINES;
```

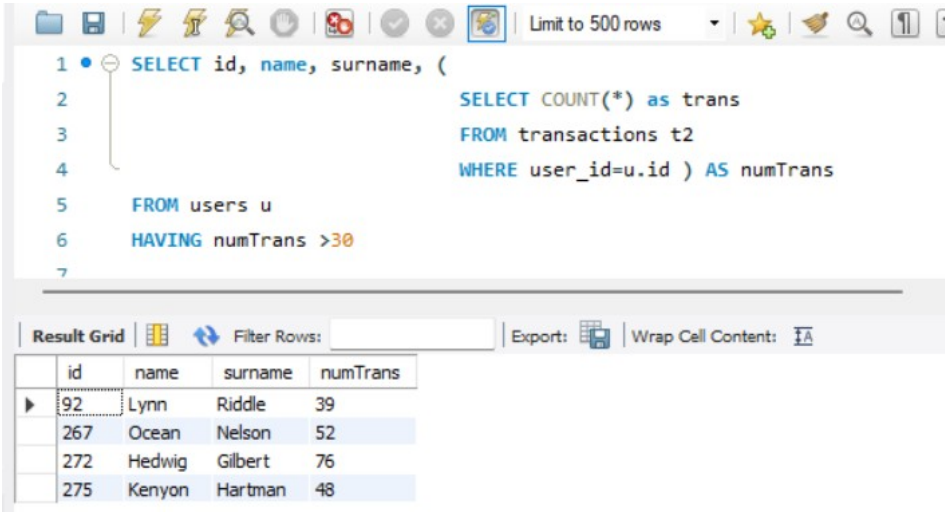
Transacctions



```
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv"
INTO TABLE transactions
FIELDS TERMINATED BY ';'
ESCAPED BY '"'
IGNORE 1 LINES;
```

Exercici 1

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



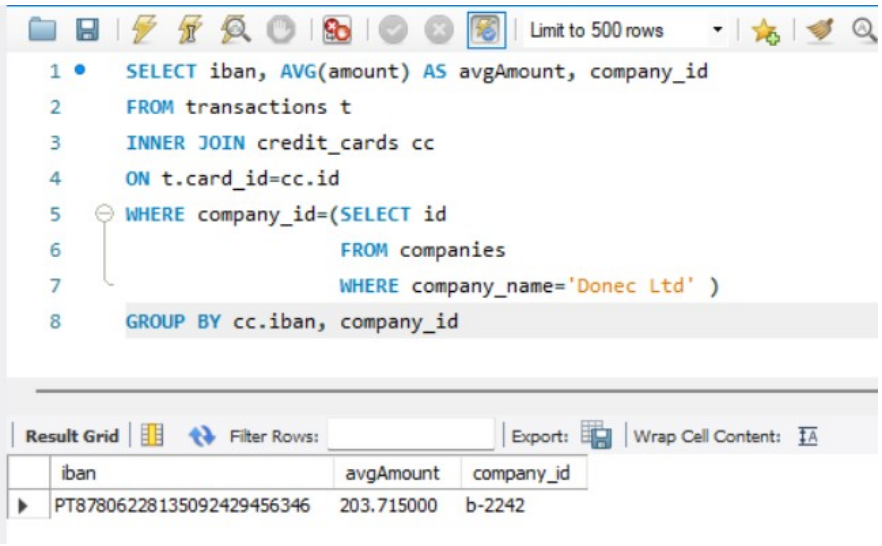
```
SELECT id, name, surname, (
                           SELECT COUNT(*) as trans
                           FROM transactions t2
                           WHERE user_id=u.id ) AS numTrans
FROM users u
HAVING numTrans >30
```

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

id	name	surname	numTrans
92	Lynn	Riddle	39
267	Ocean	Nelson	52
272	Hedwig	Gilbert	76
275	Kenyon	Hartman	48

Exercici 2

Mostra la mitjana de la suma de transaccions per IBAN de les targetes de crèdit en la companyia Donec Ltd. utilitzant almenys 2 taules.



```
SELECT iban, AVG(amount) AS avgAmount, company_id
FROM transactions t
INNER JOIN credit_cards cc
ON t.card_id=cc.id
WHERE company_id=(SELECT id
                  FROM companies
                  WHERE company_name='Donec Ltd' )
GROUP BY cc.iban, company_id
```

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

iban	avgAmount	company_id
PT87806228135092429456346	203.715000	b-2242

Nivell 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:

0.1) Creació de la taula

```sql
CREATE TABLE `active_cards` (
    `card_id` VARCHAR(15) NOT NULL,
    `declined` TINYINT DEFAULT NULL,
    PRIMARY KEY (`card_id`),
    CONSTRAINT `active_cards_ibfk_1` FOREIGN KEY (`card_id`) REFERENCES `credit_cards` (`id`)
  ) ;
```

0.2) Funció per calcular si ha estat declinada en les últimes tres transaccions

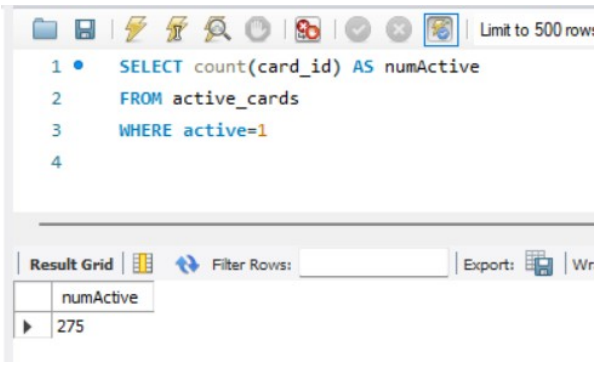
```sql
DELIMITER //

CREATE FUNCTION LastDeclined (idCard VARCHAR(15) )
RETURNS INT
READS SQL DATA

BEGIN

   DECLARE SumDeclined INT;

    SELECT SUM(declined) INTO SumDeclined FROM (
        SELECT t.card_id,timestamp,declined from transactions  t
        WHERE t.card_id=idCard
        ORDER BY t.card_id,timestamp DESC
        LIMIT 3) s;

   RETURN SumDeclined;

END; //

DELIMITER ;
```

0.3) Inserció de les dades a la nova taula

```sql

INSERT INTO active_cards
(
    SELECT DISTINCT card_id, IF (lastDeclined(card_id)=3,0,1) AS active
    FROM transactions
)
```

Exercici 1

Quantes targetes estan actives?



Hi ha 275 targetes actives

Nivell 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:

0.1) Creació de la taula

```sql
CREATE TABLE `product_transac` (
    `id` int NOT NULL AUTO_INCREMENT,
  `product_id` int NOT NULL,
  `transaction_id` VARCHAR(255) DEFAULT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `product_transac_ibfk_1` FOREIGN KEY (`product_id`) REFERENCES `products` (`id`),
  CONSTRAINT `product_transac_ibfk_2` FOREIGN KEY (`transaction_id`) REFERENCES `transactions` (`id`)
 ) ;
```

0.2) Creació d' un stored procedure per insertar els registres

```sql
DELIMITER //
CREATE PROCEDURE sp_Insert_In_Product_Trans()
BEGIN
    DECLARE sId varchar(150);
    DECLARE sProduct_ids varchar(100);
    DECLARE inumComas INT;
    DECLARE i INT;
    DECLARE var_final INT DEFAULT 0;
    DECLARE cur1 CURSOR FOR SELECT id,product_ids, ((LENGTH(product_ids)) - LENGTH((REPLACE(product_ids,",","")))  ) AS numComas
                            FROM transactions
                            WHERE id NOT IN (SELECT transaction_id
                                             FROM product_transac);
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET var_final = 1;

    OPEN cur1;
    read_loop: LOOP
    FETCH cur1 INTO sId, sProduct_ids,inumComas;
        IF var_final = 1 THEN
                LEAVE read_loop;
            END IF;
        SET i=0;
        getIdProd: LOOP
            INSERT INTO product_transac (transaction_id,product_id) VALUES (sId, SUBSTRING_INDEX(SUBSTRING_INDEX(sProduct_ids, ',', i+1),
            SET i = i +1;
            IF i > inumComas THEN
                LEAVE getIdProd;
            END IF;
        END LOOP getIdProd;

    END LOOP read_loop;
    CLOSE cur1;

END //

DELIMITER ;
```

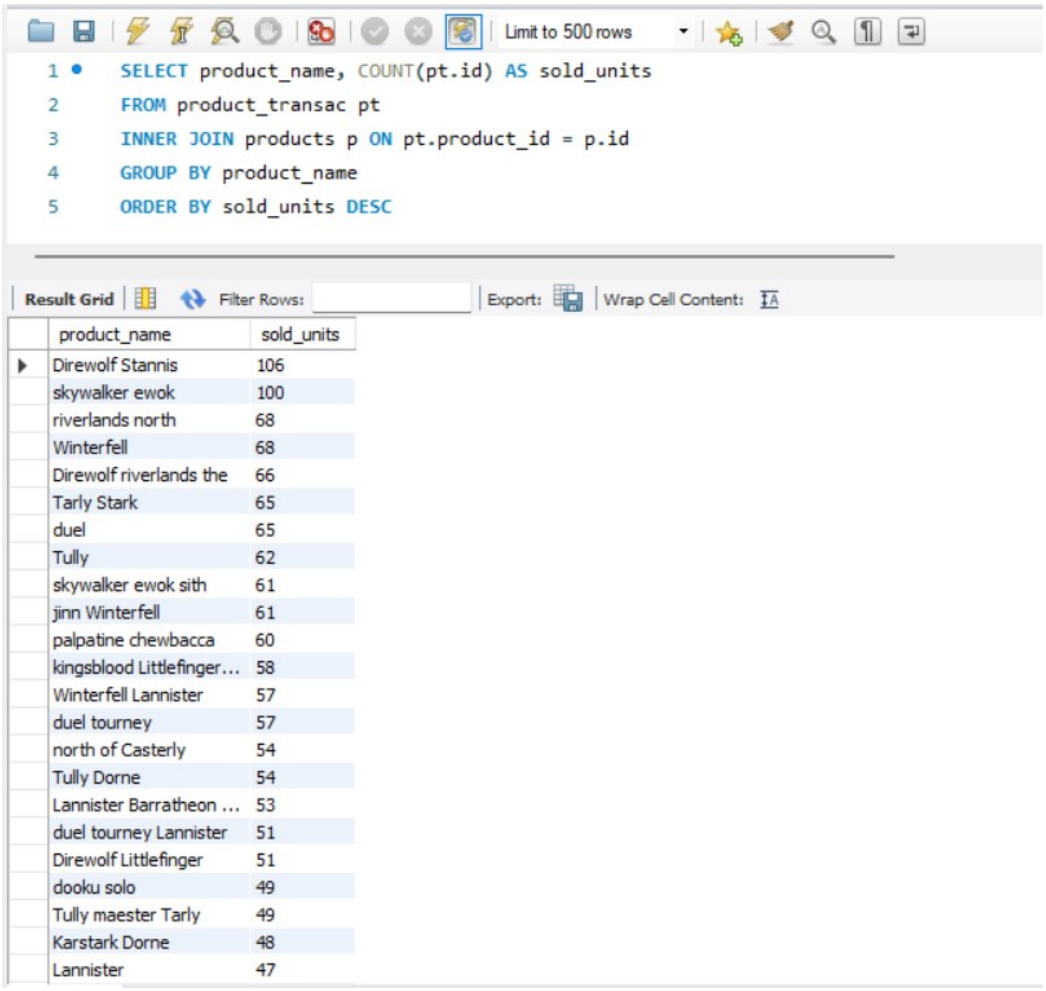
0.3) Execució del stored procedure

SQL File 19* SQL File 20* SQL File 21* SQL File 8* SQL File

```sql
CALL `sp_Insert_In_Product_Trans`();
```

Exercici 1

Necessitem conèixer el nombre de vegades que s'ha venut cada producte.



```
SELECT product_name, COUNT(pt.id) AS sold_units
FROM product_transac pt
INNER JOIN products p ON pt.product_id = p.id
GROUP BY product_name
ORDER BY sold_units DESC
```

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

product_name	sold_units
Direwolf Stannis	106
skywalker ewok	100
riverlands north	68
Winterfell	68
Direwolf riverlands the	66
Tarly Stark	65
duel	65
Tully	62
skywalker ewok sith	61
jinn Winterfell	61
palpatine chewbacca	60
kingsblood Littlefinger...	58
Winterfell Lannister	57
duel tourney	57
north of Casterly	54
Tully Dorne	54
Lannister Barratheon ...	53
duel tourney Lannister	51
Direwolf Littlefinger	51
dooku solo	49
Tully maester Tarly	49
Karstark Dorne	48
Lannister	47

Annexe

Diagrama final de la base de dades