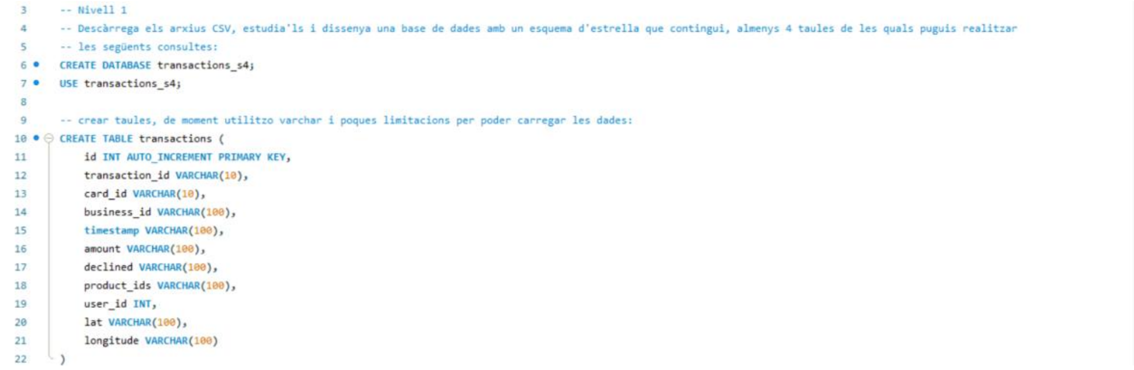
Tasca S4.01. Creació de Base de Dades

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:

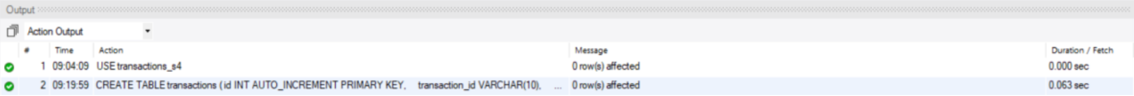
PAS 1: Començo per crear la BBDD i les taules que necessito inicialment per poder crear el model i fer les consultes que es demanen, deixo la creació products pel nivell 3. Al crear les taules, decideixo assignar tipus de variables varchar i poques limitacions per poder carregar les dades sense errors, ajustaré el tipus de variable posteriorment.

Crear taula de fets transactions:



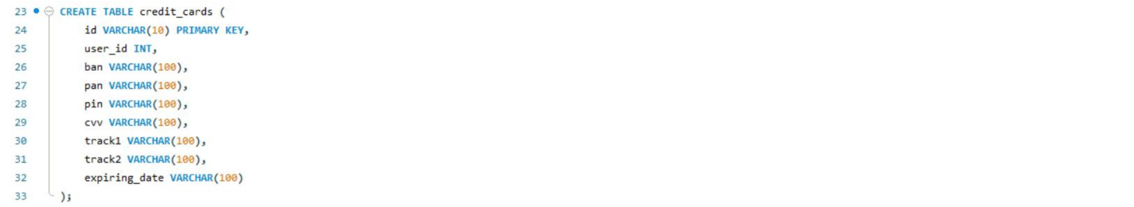
```sql
-- Nivell 1
-- Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguis realitzar
-- les següents consultes:
CREATE DATABASE transactions_s4;
USE transactions_s4;

-- crear taules, de moment utilitzo varchar i poques limitacions per poder carregar les dades:
CREATE TABLE transactions (
    id INT AUTO_INCREMENT PRIMARY KEY,
    transaction_id VARCHAR(10),
    card_id VARCHAR(10),
    business_id VARCHAR(100),
    timestamp VARCHAR(100),
    amount VARCHAR(100),
    declined VARCHAR(100),
    product_ids VARCHAR(100),
    user_id INT,
    lat VARCHAR(100),
    longitude VARCHAR(100)
)
```

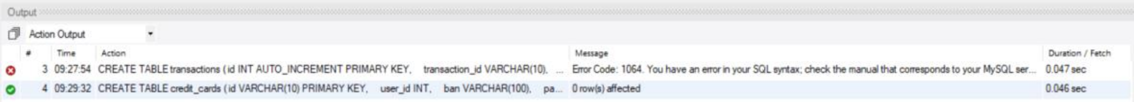


```
Output
Action Output
 #  Time      Action                                                                                    Message              Duration / Fetch
 1  09:04:09  USE transactions_s4                                                                       0 row(s) affected    0.000 sec
 2  09:19:59  CREATE TABLE transactions (id INT AUTO_INCREMENT PRIMARY KEY,   transaction_id VARCHAR(10),   ...   0 row(s) affected    0.063 sec
```

Crear taula dimensions credit_card:

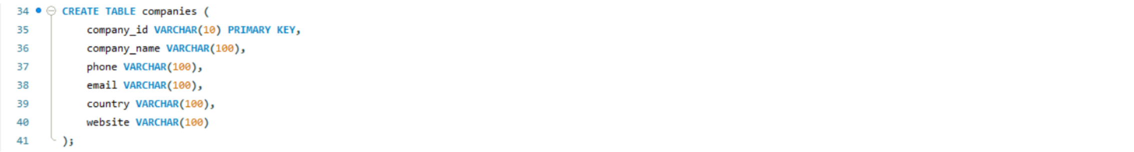


```sql
CREATE TABLE credit_cards (
    id VARCHAR(10) PRIMARY KEY,
    user_id INT,
    ban VARCHAR(100),
    pan VARCHAR(100),
    pin VARCHAR(100),
    cvv VARCHAR(100),
    track1 VARCHAR(100),
    track2 VARCHAR(100),
    expiring_date VARCHAR(100)
);
```

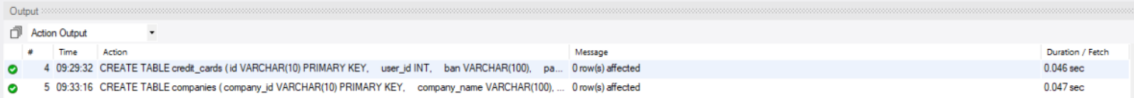


```
Output
Action Output
 #  Time      Action                                                                                          Message                                                                                      Duration / Fetch
 3  09:27:54  CREATE TABLE transactions (id INT AUTO_INCREMENT PRIMARY KEY,   transaction_id VARCHAR(10),   ...   Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL ser...   0.047 sec
 4  09:29:32  CREATE TABLE credit_cards (id VARCHAR(10) PRIMARY KEY,   user_id INT,   ban VARCHAR(100),   pa...   0 row(s) affected                                                                            0.046 sec
```

Crear taula dimensions companies:



```sql
CREATE TABLE companies (
    company_id VARCHAR(10) PRIMARY KEY,
    company_name VARCHAR(100),
    phone VARCHAR(100),
    email VARCHAR(100),
    country VARCHAR(100),
    website VARCHAR(100)
);
```

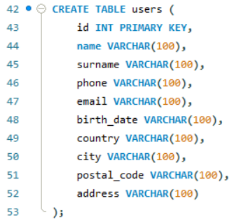


Output

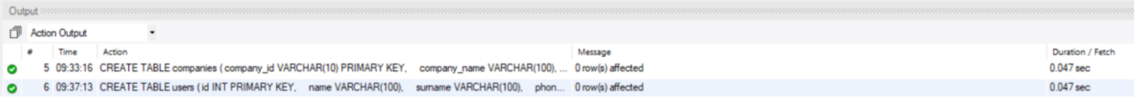
Action Output

#	Time	Action	Message	Duration / Fetch
4	09:29:32	CREATE TABLE credit_cards (id VARCHAR(10) PRIMARY KEY, user_id INT, ban VARCHAR(100), pa...	0 row(s) affected	0.046 sec
5	09:33:16	CREATE TABLE companies (company_id VARCHAR(10) PRIMARY KEY, company_name VARCHAR(100), ...	0 row(s) affected	0.047 sec

Crear taula dimensions users, com que l'estructura de dades és la mateixa per users d'Europa i d'Amèrica, decideixo unificar taules: (Es podria considerar haver posat una columna continent per si es necessites en el futur).



```sql
CREATE TABLE users (
    id INT PRIMARY KEY,
    name VARCHAR(100),
    surname VARCHAR(100),
    phone VARCHAR(100),
    email VARCHAR(100),
    birth_date VARCHAR(100),
    country VARCHAR(100),
    city VARCHAR(100),
    postal_code VARCHAR(100),
    address VARCHAR(100)
);
```



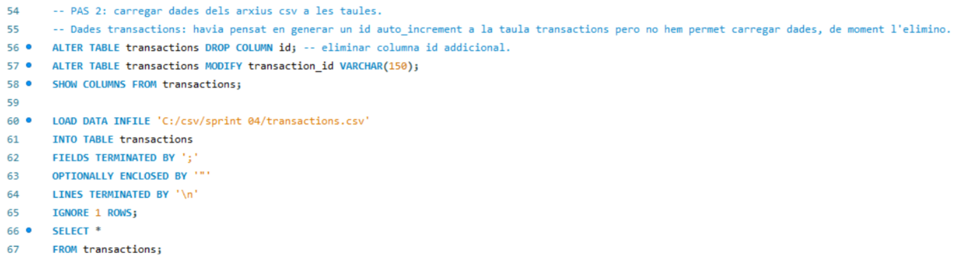
Output

Action Output

#	Time	Action	Message	Duration / Fetch
5	09:33:16	CREATE TABLE companies (company_id VARCHAR(10) PRIMARY KEY, company_name VARCHAR(100), ...	0 row(s) affected	0.047 sec
6	09:37:13	CREATE TABLE users (id INT PRIMARY KEY, name VARCHAR(100), surname VARCHAR(100), phon...	0 row(s) affected	0.047 sec

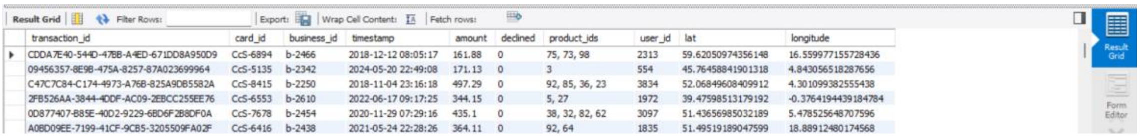
PAS 2: Carregar dades dels arxius csv.

Per la taula transactions havia creat un atribut id auto_increment però de moment l'he hagut de eliminar per carregar dades.



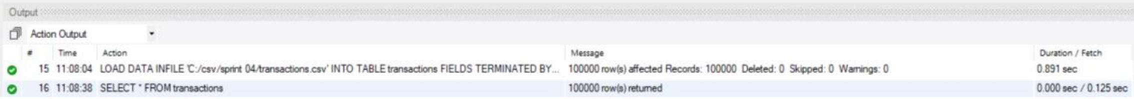
```sql
-- PAS 2: carregar dades dels arxius csv a les taules.
-- Dades transactions: havia pensat en generar un id auto_increment a la taula transactions pero no hem permet carregar dades, de moment l'elimino.
ALTER TABLE transactions DROP COLUMN id; -- eliminar columna id addicional.
ALTER TABLE transactions MODIFY transaction_id VARCHAR(150);
SHOW COLUMNS FROM transactions;

LOAD DATA INFILE 'C:/csv/sprint 04/transactions.csv'
INTO TABLE transactions
FIELDS TERMINATED BY ';'
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
SELECT *
FROM transactions;
```



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

transaction_id	card_id	business_id	timestamp	amount	declined	product_ids	user_id	lat	longitude
CDDA7E40-544D-47BB-A4ED-671DD8A950D9	CcS-6894	b-2466	2018-12-12 08:05:17	161.88	0	75, 73, 98	2313	59.62050974356148	16.559977155728436
09456357-8E9B-475A-8257-87A023699964	CcS-5135	b-2342	2024-05-20 22:49:08	171.13	0	3	554	45.76458841901318	4.843056518287656
C47C7C84-C174-4973-A76B-825A9DB5582A	CcS-8415	b-2250	2018-11-04 23:16:18	497.29	0	92, 85, 36, 23	3834	52.06849608409912	4.301099382555438
2FB526AA-3844-4DDF-AC09-2EBCC255EE76	CcS-6553	b-2610	2022-06-17 09:17:25	344.15	0	5, 27	1972	39.47598513179192	-0.3764194439184784
0D877407-B85E-40D2-9229-6BD6F2B8DF0A	CcS-7678	b-2454	2020-11-29 07:29:16	435.1	0	38, 32, 82, 62	3097	51.43656985032189	5.478525648707596
A0BD09EE-7199-41CF-9CB5-3205509FA02F	CcS-6416	b-2438	2021-05-24 22:28:26	364.11	0	92, 64	1835	51.49519189047599	18.88912480174568

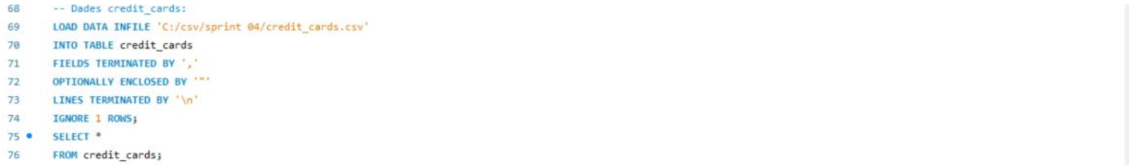


Output

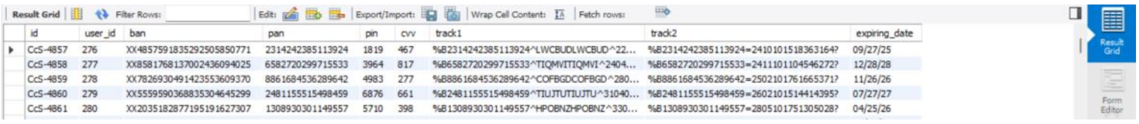
Action Output

#	Time	Action	Message	Duration / Fetch
15	11:08:04	LOAD DATA INFILE 'C:/csv/sprint 04/transactions.csv' INTO TABLE transactions FIELDS TERMINATED BY...	100000 row(s) affected Records: 100000 Deleted: 0 Skipped: 0 Warnings: 0	0.891 sec
16	11:08:38	SELECT * FROM transactions	100000 row(s) returned	0.000 sec / 0.125 sec

Carregar dades de credit_cards:

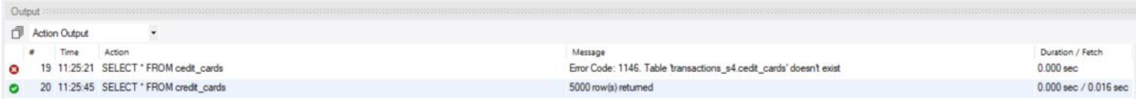


```sql
-- Dades credit_cards:
LOAD DATA INFILE 'C:/csv/sprint 04/credit_cards.csv'
INTO TABLE credit_cards
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
SELECT *
FROM credit_cards;
```



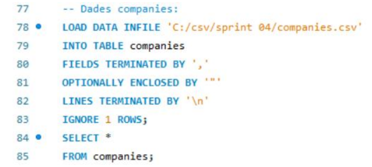
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

id	user_id	ban	pan	pin	cvv	track1	track2	expiring_date
CcS-4857	276	XX4857591835292505850771	2314242385113924	1819	467	%B2314242385113924^LWCBUDLWCBUD^22...	%B2314242385113924=2410101518363164?	09/27/25
CcS-4858	277	XX8581768137002436094025	6582720299715533	3964	817	%B6582720299715533^TIQMVITIQMVI^2404...	%B6582720299715533=2411101104546272?	12/28/28
CcS-4859	278	XX7826930491423553609370	8861684536289642	4983	277	%B8861684536289642^COFBGDCOFBGD^280...	%B8861684536289642=2502101761665371?	11/26/26
CcS-4860	279	XX5559590368835304645299	2481155515498459	6876	661	%B2481155515498459^TIUJTUTIUJTU^31040...	%B2481155515498459=2602101514414395?	07/27/27
CcS-4861	280	XX2035182877195191627307	1308930301149557	5710	398	%B1308930301149557^HPOBNZHPOBNZ^330...	%B1308930301149557=2805101751305028?	04/25/26

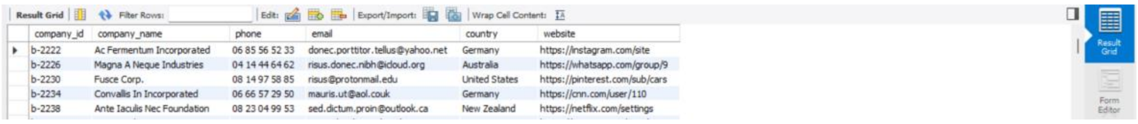


```
Output
Action Output
#   Time      Action                          Message                                                      Duration / Fetch
19  11:25:21  SELECT * FROM cedit_cards       Error Code: 1146. Table 'transactions_s4.cedit_cards' doesn't exist   0.000 sec
20  11:25:45  SELECT * FROM credit_cards      5000 row(s) returned                                         0.000 sec / 0.016 sec
```

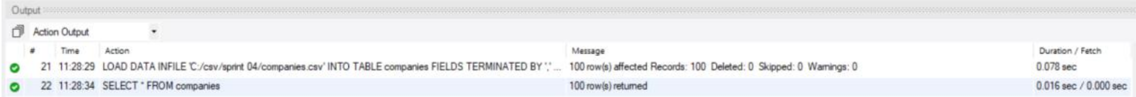
Carregar dades companyies:



```sql
-- Dades companies:
LOAD DATA INFILE 'C:/csv/sprint 04/companies.csv'
INTO TABLE companies
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
SELECT *
FROM companies;
```

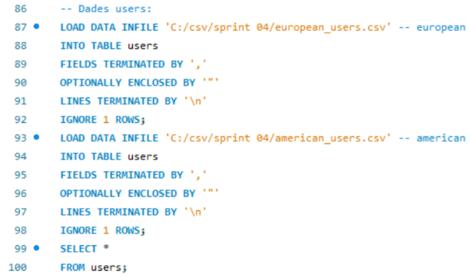


company_id	company_name	phone	email	country	website
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net	Germany	https://instagram.com/site
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia	https://whatsapp.com/group/9
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/sub/cars
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany	https://cnn.com/user/110
b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings

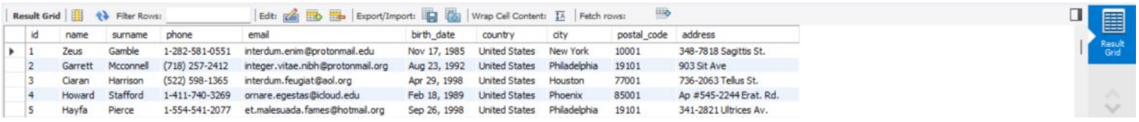


```
Output
Action Output
#   Time      Action                                                                                  Message                                                                  Duration / Fetch
21  11:28:29  LOAD DATA INFILE 'C:/csv/sprint 04/companies.csv' INTO TABLE companies FIELDS TERMINATED BY ',' ...   100 row(s) affected Records: 100  Deleted: 0  Skipped: 0  Warnings: 0   0.078 sec
22  11:28:34  SELECT * FROM companies                                                                 100 row(s) returned                                                      0.016 sec / 0.000 sec
```

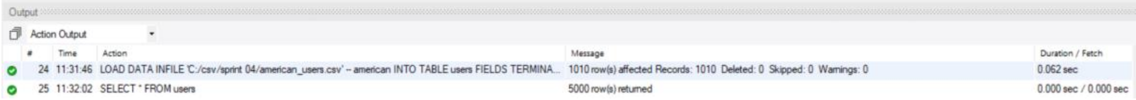
Carregar dades users:



```sql
-- Dades users:
LOAD DATA INFILE 'C:/csv/sprint 04/european_users.csv' -- european
INTO TABLE users
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
LOAD DATA INFILE 'C:/csv/sprint 04/american_users.csv' -- american
INTO TABLE users
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
SELECT *
FROM users;
```



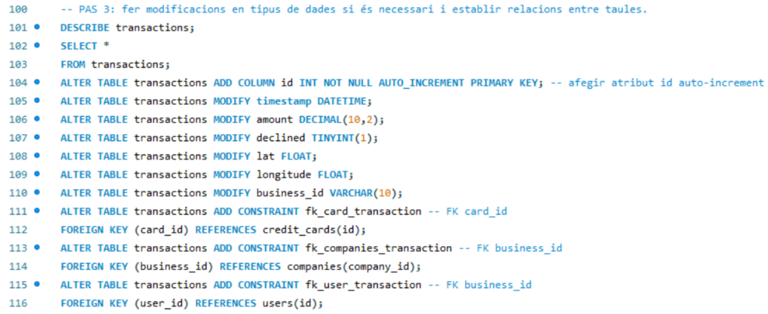
id	name	surname	phone	email	birth_date	country	city	postal_code	address
1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.edu	Nov 17, 1985	United States	New York	10001	348-7818 Sagittis St.
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Philadelphia	19101	903 Sit Ave
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Houston	77001	736-2063 Tellus St.
4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.edu	Feb 18, 1989	United States	Phoenix	85001	Ap #545-2244 Erat. Rd.
5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	Sep 26, 1998	United States	Philadelphia	19101	341-2821 Ultrices Av.



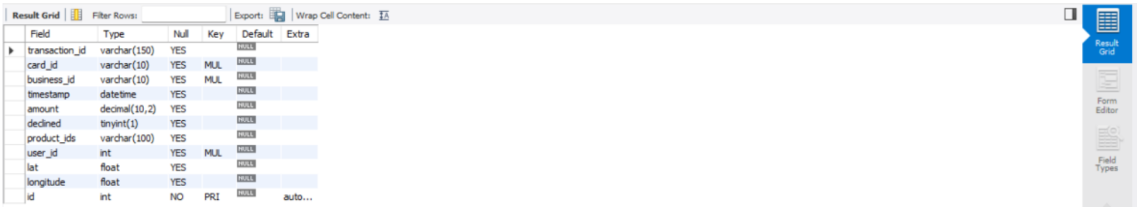
```
Output
Action Output
#   Time      Action                                                                                       Message                                                                     Duration / Fetch
24  11:31:46  LOAD DATA INFILE 'C:/csv/sprint 04/american_users.csv' -- american INTO TABLE users FIELDS TERMINA...   1010 row(s) affected Records: 1010  Deleted: 0  Skipped: 0  Warnings: 0   0.062 sec
25  11:32:02  SELECT * FROM users                                                                          5000 row(s) returned                                                        0.000 sec / 0.000 sec
```

PAS 3: fer modificacions en tipus de dades si és necessari i establir relacions entre taules.

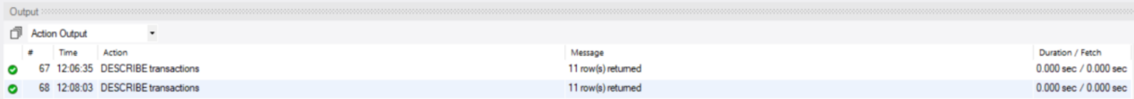
Taula transactions:



```sql
-- PAS 3: fer modificacions en tipus de dades si és necessari i establir relacions entre taules.
DESCRIBE transactions;
SELECT *
FROM transactions;
ALTER TABLE transactions ADD COLUMN id INT NOT NULL AUTO_INCREMENT PRIMARY KEY; -- afegir atribut id auto-increment
ALTER TABLE transactions MODIFY timestamp DATETIME;
ALTER TABLE transactions MODIFY amount DECIMAL(10,2);
ALTER TABLE transactions MODIFY declined TINYINT(1);
ALTER TABLE transactions MODIFY lat FLOAT;
ALTER TABLE transactions MODIFY longitude FLOAT;
ALTER TABLE transactions MODIFY business_id VARCHAR(10);
ALTER TABLE transactions ADD CONSTRAINT fk_card_transaction -- FK card_id
FOREIGN KEY (card_id) REFERENCES credit_cards(id);
ALTER TABLE transactions ADD CONSTRAINT fk_companies_transaction -- FK business_id
FOREIGN KEY (business_id) REFERENCES companies(company_id);
ALTER TABLE transactions ADD CONSTRAINT fk_user_transaction -- FK business_id
FOREIGN KEY (user_id) REFERENCES users(id);
```

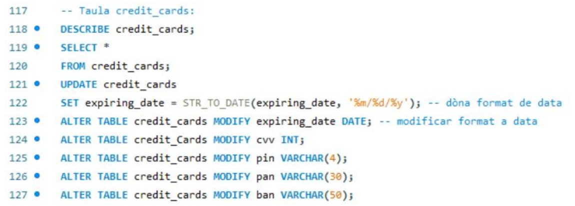


Field	Type	Null	Key	Default	Extra
transaction_id	varchar(150)	YES		NULL	
card_id	varchar(10)	YES	MUL	NULL	
business_id	varchar(10)	YES	MUL	NULL	
timestamp	datetime	YES		NULL	
amount	decimal(10,2)	YES		NULL	
declined	tinyint(1)	YES		NULL	
product_ids	varchar(100)	YES		NULL	
user_id	int	YES	MUL	NULL	
lat	float	YES		NULL	
longitude	float	YES		NULL	
id	int	NO	PRI	NULL	auto...

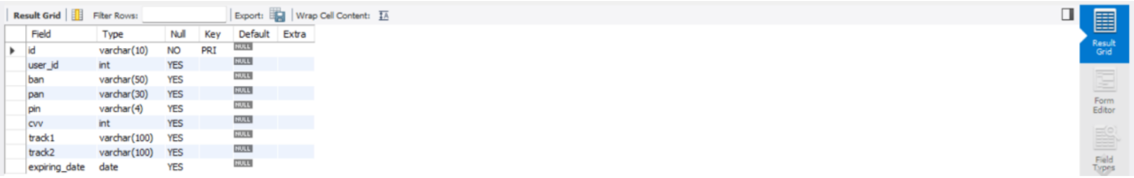


#	Time	Action	Message	Duration / Fetch
67	12:06:35	DESCRIBE transactions	11 row(s) returned	0.000 sec / 0.000 sec
68	12:08:03	DESCRIBE transactions	11 row(s) returned	0.000 sec / 0.000 sec

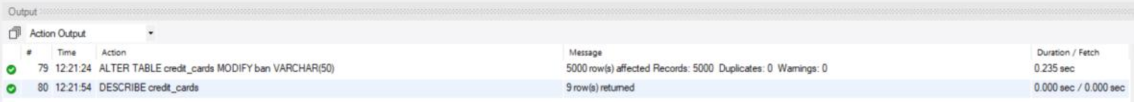
Taula credit_cards:



```
-- Taula credit_cards:
DESCRIBE credit_cards;
SELECT *
FROM credit_cards;
UPDATE credit_cards
SET expiring_date = STR_TO_DATE(expiring_date, '%m/%d/%y'); -- dòna format de data
ALTER TABLE credit_cards MODIFY expiring_date DATE; -- modificar format a data
ALTER TABLE credit_Cards MODIFY cvv INT;
ALTER TABLE credit_cards MODIFY pin VARCHAR(4);
ALTER TABLE credit_cards MODIFY pan VARCHAR(30);
ALTER TABLE credit_cards MODIFY ban VARCHAR(50);
```

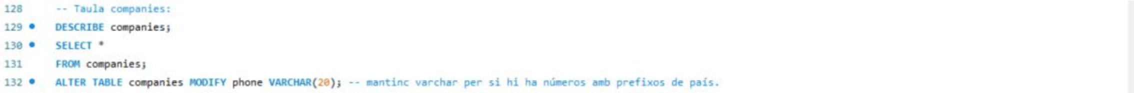


Field	Type	Null	Key	Default	Extra
id	varchar(10)	NO	PRI	NULL	
user_id	int	YES		NULL	
ban	varchar(50)	YES		NULL	
pan	varchar(30)	YES		NULL	
pin	varchar(4)	YES		NULL	
cvv	int	YES		NULL	
track1	varchar(100)	YES		NULL	
track2	varchar(100)	YES		NULL	
expiring_date	date	YES		NULL	

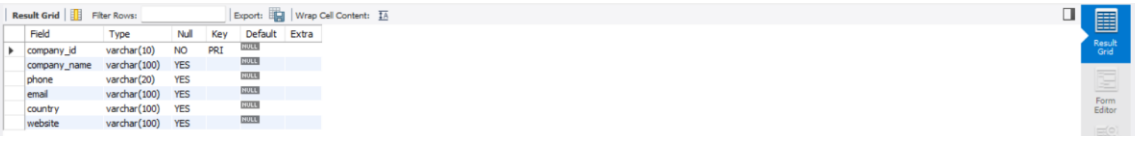


#	Time	Action	Message	Duration / Fetch
79	12:21:24	ALTER TABLE credit_cards MODIFY ban VARCHAR(50)	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0	0.235 sec
80	12:21:54	DESCRIBE credit_cards	9 row(s) returned	0.000 sec / 0.000 sec

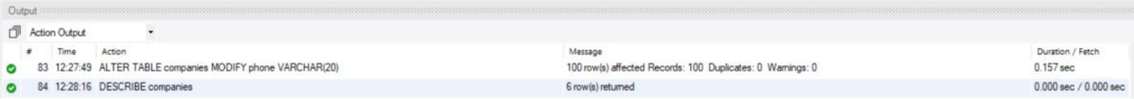
Taula companies:



```
-- Taula companies:
DESCRIBE companies;
SELECT *
FROM companies;
ALTER TABLE companies MODIFY phone VARCHAR(20); -- mantinc varchar per si hi ha números amb prefixos de país.
```

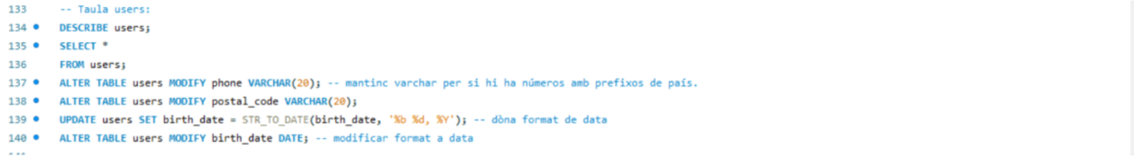


Field	Type	Null	Key	Default	Extra
company_id	varchar(10)	NO	PRI	NULL	
company_name	varchar(100)	YES		NULL	
phone	varchar(20)	YES		NULL	
email	varchar(100)	YES		NULL	
country	varchar(100)	YES		NULL	
website	varchar(100)	YES		NULL	

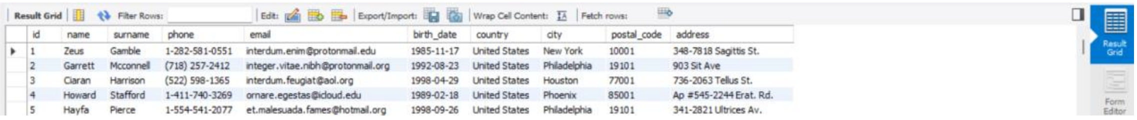


#	Time	Action	Message	Duration / Fetch
83	12:27:49	ALTER TABLE companies MODIFY phone VARCHAR(20)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0	0.157 sec
84	12:28:16	DESCRIBE companies	6 row(s) returned	0.000 sec / 0.000 sec

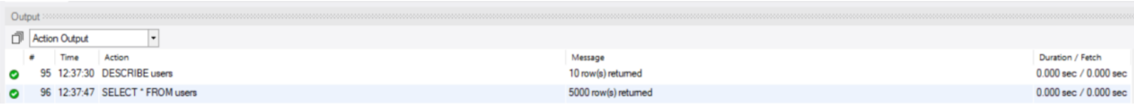
Taula users:



```
-- Taula users:
DESCRIBE users;
SELECT *
FROM users;
ALTER TABLE users MODIFY phone VARCHAR(20); -- mantinc varchar per si hi ha números amb prefixos de país.
ALTER TABLE users MODIFY postal_code VARCHAR(20);
UPDATE users SET birth_date = STR_TO_DATE(birth_date, '%b %d, %Y'); -- dòna format de data
ALTER TABLE users MODIFY birth_date DATE; -- modificar format a data
```

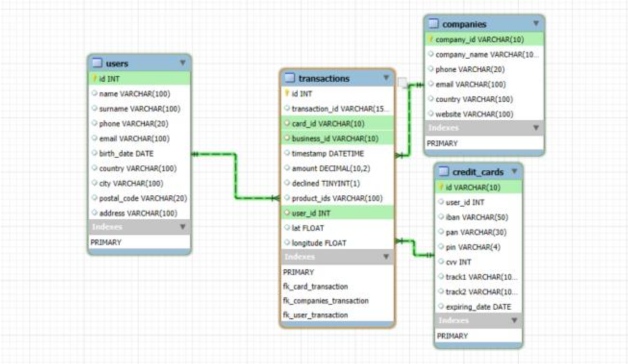


id	name	surname	phone	email	birth_date	country	city	postal_code	address
1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.edu	1985-11-17	United States	New York	10001	348-7818 Sagittis St.
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	1992-08-23	United States	Philadelphia	19101	903 Sit Ave
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	1998-04-29	United States	Houston	77001	736-2063 Tellus St.
4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.edu	1989-02-18	United States	Phoenix	85001	Ap #545-2244 Erat. Rd.
5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	1998-09-26	United States	Philadelphia	19101	341-2821 Ultrices Av.



```
Output
Action Output
   #   Time      Action                  Message              Duration / Fetch
✓  95  12:37:30  DESCRIBE users          10 row(s) returned   0.000 sec / 0.000 sec
✓  96  12:37:47  SELECT * FROM users     5000 row(s) returned 0.000 sec / 0.000 sec
```

Resultat: (es podria haver unit la taula users amb credit_cards, per simplificació del model no ho he fet)



users
- id INT
- name VARCHAR(100)
- surname VARCHAR(100)
- phone VARCHAR(20)
- email VARCHAR(100)
- birth_date DATE
- country VARCHAR(100)
- city VARCHAR(100)
- postal_code VARCHAR(20)
- address VARCHAR(100)
- Indexes
- PRIMARY

transactions
- id INT
- transaction_id VARCHAR(15...
- card_id VARCHAR(10)
- business_id VARCHAR(10)
- timestamp DATETIME
- amount DECIMAL(10,2)
- declined TINYINT(1)
- product_ids VARCHAR(100)
- user_id INT
- lat FLOAT
- longitude FLOAT
- Indexes
- PRIMARY
- fk_card_transaction
- fk_companies_transaction
- fk_user_transaction

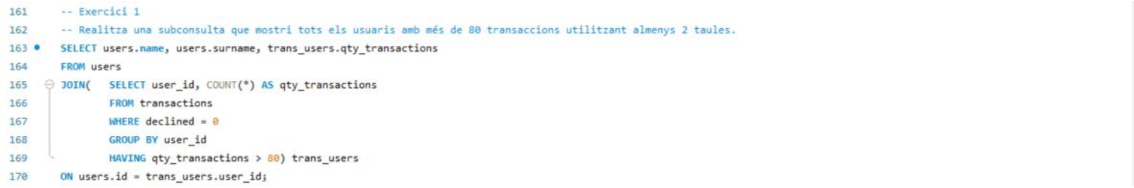
companies
- company_id VARCHAR(10)
- company_name VARCHAR(10...
- phone VARCHAR(20)
- email VARCHAR(100)
- country VARCHAR(100)
- website VARCHAR(100)
- Indexes
- PRIMARY

credit_cards
- id VARCHAR(10)
- user_id INT
- iban VARCHAR(50)
- pan VARCHAR(30)
- pin VARCHAR(4)
- cvv INT
- track1 VARCHAR(10...
- track2 VARCHAR(10...
- expiring_date DATE
- Indexes
- PRIMARY

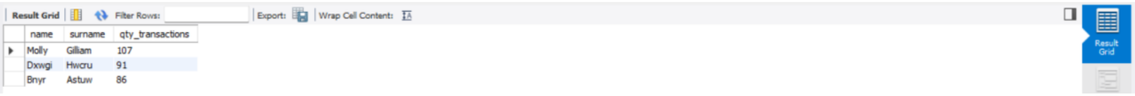
Exercici 1

Realitza una subconsulta que mostri tots els usuaris amb més de 80 transaccions utilitzant almenys 2 taules.

He realitzat una subconsulta per saber la quantitat de transaccions que realitzen els usuaris. Com que només interessa les no declined he utilitzat filtra where perquè no comptabilitzi les declined, i he utilitzat un filtre having per que només consideri usuaris amb més de 80 transaccions. Finalment amb join he unit subconsulta amb taula users per saber el nom dels usuaris que busquem.

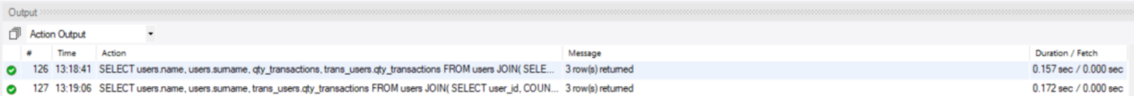


```sql
-- Exercici 1
-- Realitza una subconsulta que mostri tots els usuaris amb més de 80 transaccions utilitzant almenys 2 taules.
SELECT users.name, users.surname, trans_users.qty_transactions
FROM users
JOIN(   SELECT user_id, COUNT(*) AS qty_transactions
        FROM transactions
        WHERE declined = 0
        GROUP BY user_id
        HAVING qty_transactions > 80) trans_users
ON users.id = trans_users.user_id;
```



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

name	surname	qty_transactions
Molly	Gilliam	107
Dxwgi	Hwcru	91
Bnyr	Astuw	86

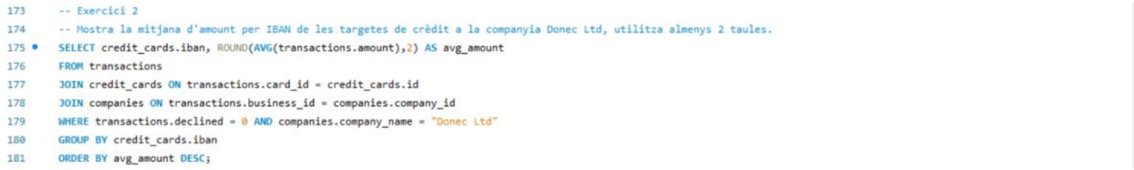


```
Output
Action Output
   #    Time      Action                                                                                                  Message             Duration / Fetch
✓  126  13:18:41  SELECT users.name, users.sumame, qty_transactions, trans_users.qty_transactions FROM users JOIN( SELE...  3 row(s) returned  0.157 sec / 0.000 sec
✓  127  13:19:06  SELECT users.name, users.sumame, trans_users.qty_transactions FROM users JOIN( SELECT user_id, COUN...    3 row(s) returned  0.172 sec / 0.000 sec
```

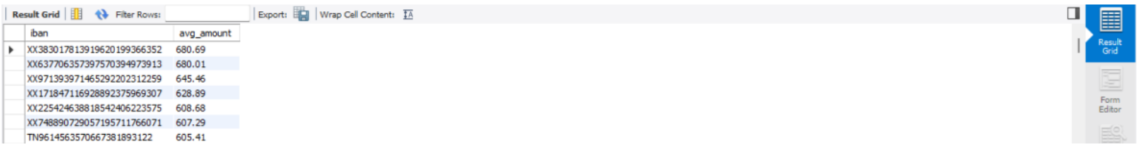
Exercici 2

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

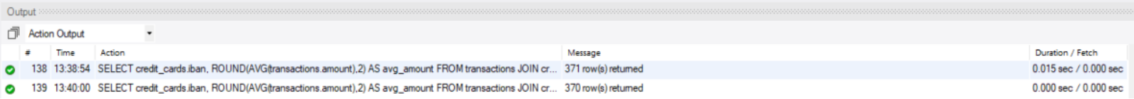
He realitzat una unió de les taules transactions, credit_cards i companyies, per posteriorment filtrar per veure només les transaccions no declined i de la companyia que demanen. Finalment he agrupat per iban per calcular la mitjana d’amount i ordenat el resultat de major a menor.



```
-- Exercici 2
-- Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.
SELECT credit_cards.iban, ROUND(AVG(transactions.amount),2) AS avg_amount
FROM transactions
JOIN credit_cards ON transactions.card_id = credit_cards.id
JOIN companies ON transactions.business_id = companies.company_id
WHERE transactions.declined = 0 AND companies.company_name = "Donec Ltd"
GROUP BY credit_cards.iban
ORDER BY avg_amount DESC;
```



iban	avg_amount
XX383017813919620199366352	680.69
XX637706357397570394973913	680.01
XX971393971465292202312259	645.46
XX171847116928892375969307	628.89
XX225424638818542406223575	608.68
XX748890729057195711766071	607.29
TN9614563570667381893122	605.41



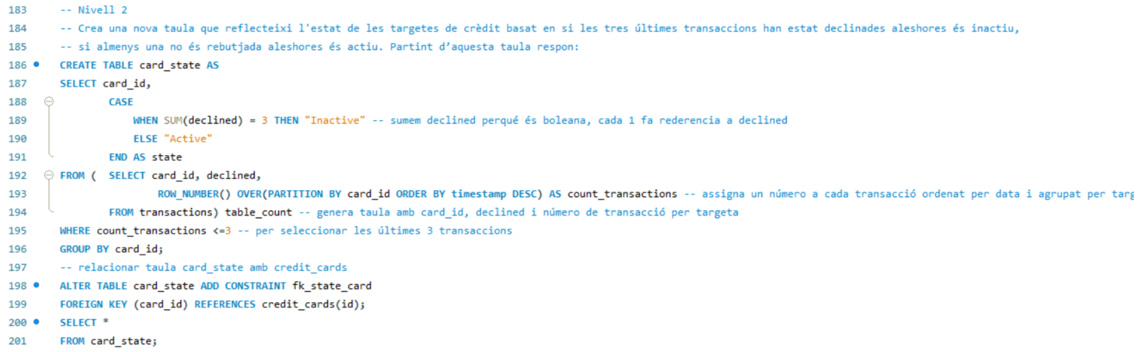
```
#    Time      Action                                                                              Message               Duration / Fetch
138  13:38:54  SELECT credit_cards.iban, ROUND(AVG(transactions.amount),2) AS avg_amount FROM transactions JOIN cr...  371 row(s) returned  0.015 sec / 0.000 sec
139  13:40:00  SELECT credit_cards.iban, ROUND(AVG(transactions.amount),2) AS avg_amount FROM transactions JOIN cr...  370 row(s) returned  0.000 sec / 0.000 sec
```

Nivell 2

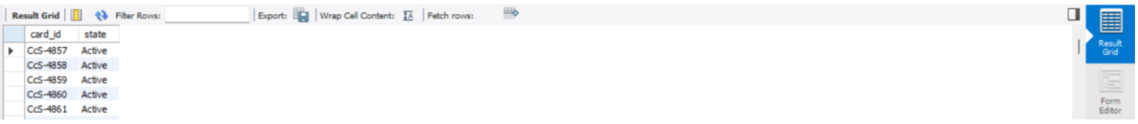
Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en *si les tres últimes transaccions han estat declinades aleshores és inactiu, si almenys una no és rebutjada aleshores és actiu*. Partint d’aquesta taula respon:

PAS INICIAL: Per poder crear aquesta taula primer he creat una subquery per utilitzar al from amb les dades card_id, declined i una window function amb row_number() per comptabilitzar cada transacció per targeta ordenada per data, de més recent a més antiga. Agafant aquestes dades, he creat una taula que mostres les card_id agrupades i l’estat de la targeta, per fer_ho he utilitzat un case amb un sum(declined), ja que declined és una dada boleana, amb un filtre where per sumar només les últimes tres transaccions.

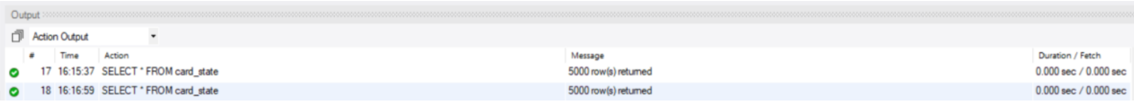
Finalment, he generat una relació entre la taula creada i la taula credit_cards.



```
-- Nivell 2
-- Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les tres últimes transaccions han estat declinades aleshores és inactiu,
-- si almenys una no és rebutjada aleshores és actiu. Partint d’aquesta taula respon:
CREATE TABLE card_state AS
SELECT card_id,
        CASE
            WHEN SUM(declined) = 3 THEN "Inactive" -- sumem declined perqué és boleana, cada 1 fa rederencia a declined
            ELSE "Active"
        END AS state
FROM (  SELECT card_id, declined,
                ROW_NUMBER() OVER(PARTITION BY card_id ORDER BY timestamp DESC) AS count_transactions -- assigna un número a cada transacció ordenat per data i agrupat per targ
        FROM transactions) table_count -- genera taula amb card_id, declined i número de transacció per targeta
WHERE count_transactions <=3 -- per seleccionar les últimes 3 transaccions
GROUP BY card_id;
-- relacionar taula card_state amb credit_cards
ALTER TABLE card_state ADD CONSTRAINT fk_state_card
FOREIGN KEY (card_id) REFERENCES credit_cards(id);
SELECT *
FROM card_state;
```



card_id	state
CcS-4857	Active
CcS-4858	Active
CcS-4859	Active
CcS-4860	Active
CcS-4861	Active

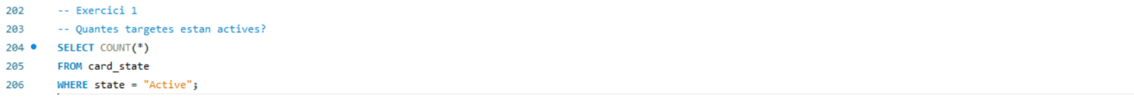


```
#   Time      Action                     Message               Duration / Fetch
17  16:15:37  SELECT * FROM card_state   5000 row(s) returned  0.000 sec / 0.000 sec
18  16:16:59  SELECT * FROM card_state   5000 row(s) returned  0.000 sec / 0.000 sec
```

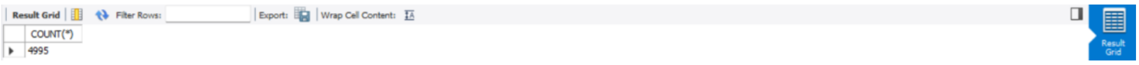
Exercici 1

Quantes targetes estan actives?

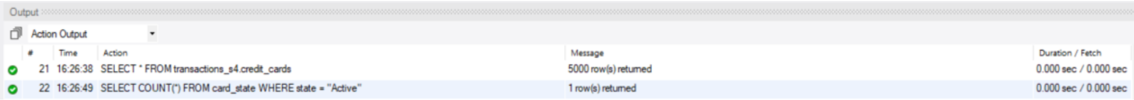
Hi ha 4995 targetes actives de les 5000 registrades a la taula credit_cards.



```sql
-- Exercici 1
-- Quantes targetes estan actives?
SELECT COUNT(*)
FROM card_state
WHERE state = "Active";
```



```
COUNT(*)
4995
```

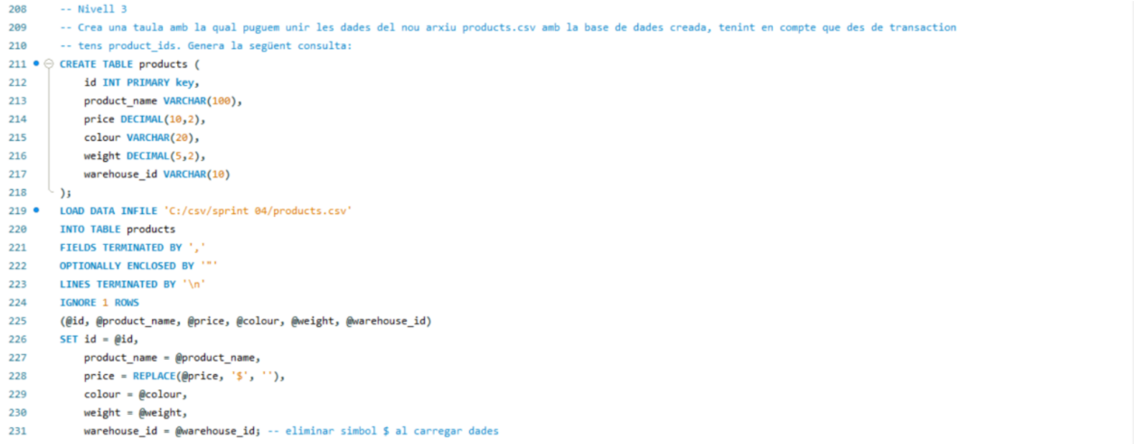


```
#   Time      Action                                                 Message                Duration / Fetch
21  16:26:38  SELECT * FROM transactions_s4.credit_cards             5000 row(s) returned   0.000 sec / 0.000 sec
22  16:26:49  SELECT COUNT(*) FROM card_state WHERE state = "Active" 1 row(s) returned      0.000 sec / 0.000 sec
```

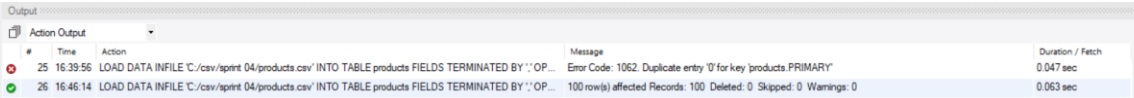
Nivell 3

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:

PAS 1: Crear la taula products i carregar les dades de l'arxiu csv, el símbol $ dona problemes al carregar dades.

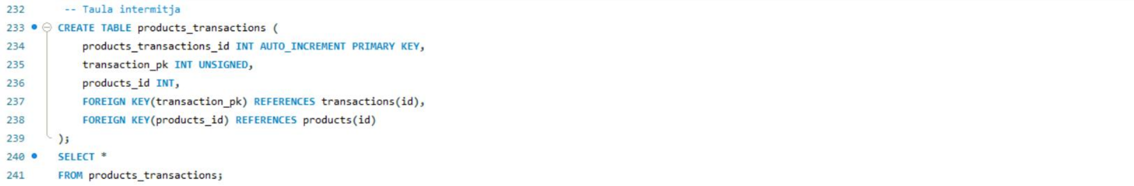


```sql
-- Nivell 3
-- Crea una taula amb la qual puguem unir les dades del nou arxiu products.csv amb la base de dades creada, tenint en compte que des de transaction
-- tens product_ids. Genera la següent consulta:
CREATE TABLE products (
    id INT PRIMARY key,
    product_name VARCHAR(100),
    price DECIMAL(10,2),
    colour VARCHAR(20),
    weight DECIMAL(5,2),
    warehouse_id VARCHAR(10)
);
LOAD DATA INFILE 'C:/csv/sprint 04/products.csv'
INTO TABLE products
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(@id, @product_name, @price, @colour, @weight, @warehouse_id)
SET id = @id,
    product_name = @product_name,
    price = REPLACE(@price, '$', ''),
    colour = @colour,
    weight = @weight,
    warehouse_id = @warehouse_id; -- eliminar simbol $ al carregar dades
```

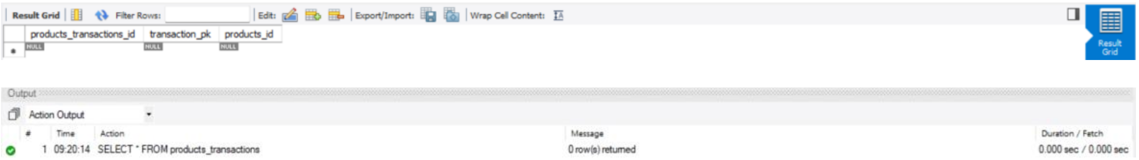


```
#   Time      Action                                                                                  Message                                                              Duration / Fetch
25  16:39:56  LOAD DATA INFILE 'C:/csv/sprint 04/products.csv' INTO TABLE products FIELDS TERMINATED BY ',' OP...  Error Code: 1062. Duplicate entry '0' for key 'products.PRIMARY'  0.047 sec
26  16:46:14  LOAD DATA INFILE 'C:/csv/sprint 04/products.csv' INTO TABLE products FIELDS TERMINATED BY ',' OP...  100 row(s) affected Records: 100  Deleted: 0  Skipped: 0  Warnings: 0  0.063 sec
```

PAS 2: Crear taula entre products i transactions per trencar relació N:M i poder establir relació 1:N entre taules.



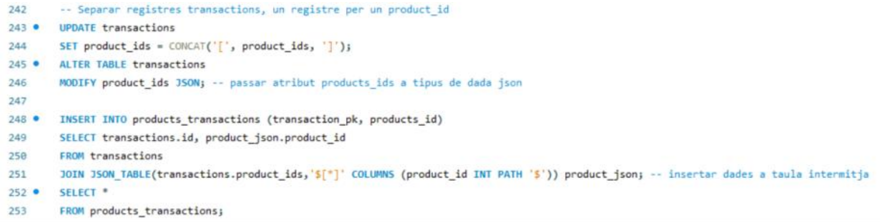
```sql
 -- Taula intermitja
CREATE TABLE products_transactions (
    products_transactions_id INT AUTO_INCREMENT PRIMARY KEY,
    transaction_pk INT UNSIGNED,
    products_id INT,
    FOREIGN KEY(transaction_pk) REFERENCES transactions(id),
    FOREIGN KEY(products_id) REFERENCES products(id)
);
SELECT *
FROM products_transactions;
```



products_transactions_id	transaction_pk	products_id
NULL	NULL	NULL

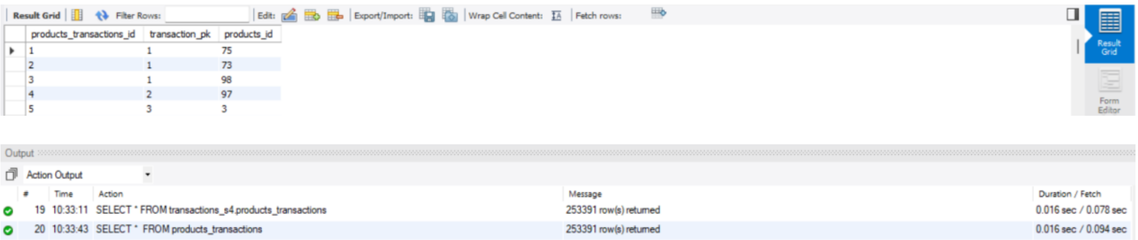
```
Output
Action Output
    #  Time      Action                                   Message            Duration / Fetch
✓   1  09:20:14  SELECT * FROM products_transactions      0 row(s) returned  0.000 sec / 0.000 sec
```

PAS 3: Fer que cada registre de la taula transactions correspongui a un product_id i inserir valors a taula products_transactions. Converteixo product_ids a tipus de dades json, per poder treballar-les com a dades json.



```sql
-- Separar registres transactions, un registre per un product_id
UPDATE transactions
SET product_ids = CONCAT('[', product_ids, ']');
ALTER TABLE transactions
MODIFY product_ids JSON; -- passar atribut products_ids a tipus de dada json

INSERT INTO products_transactions (transaction_pk, products_id)
SELECT transactions.id, product_json.product_id
FROM transactions
JOIN JSON_TABLE(transactions.product_ids,'$[*]' COLUMNS (product_id INT PATH '$')) product_json; -- insertar dades a taula intermitja
SELECT *
FROM products_transactions;
```



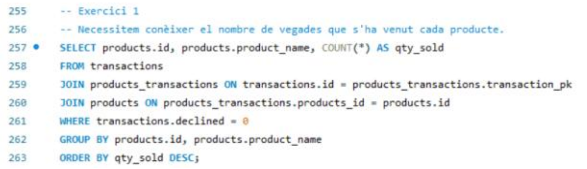
products_transactions_id	transaction_pk	products_id
1	1	75
2	1	73
3	1	98
4	2	97
5	3	3

```
Output
Action Output
    #   Time      Action                                                Message                 Duration / Fetch
✓   19  10:33:11  SELECT * FROM transactions_s4.products_transactions   253391 row(s) returned  0.016 sec / 0.078 sec
✓   20  10:33:43  SELECT *  FROM products_transactions                  253391 row(s) returned  0.016 sec / 0.094 sec
```

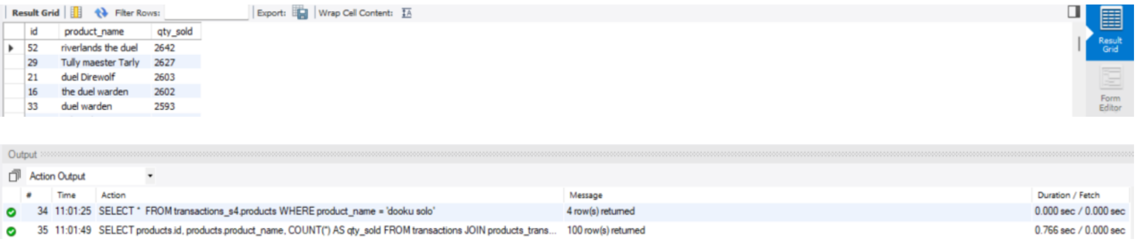
He considerat eliminar l'atribut product_ids de taula transactions per obtenir una taula més neta, de moment he decidit no procedir.

Exercici 1

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



```sql
-- Exercici 1
-- Necessitem conèixer el nombre de vegades que s'ha venut cada producte.
SELECT products.id, products.product_name, COUNT(*) AS qty_sold
FROM transactions
JOIN products_transactions ON transactions.id = products_transactions.transaction_pk
JOIN products ON products_transactions.products_id = products.id
WHERE transactions.declined = 0
GROUP BY products.id, products.product_name
ORDER BY qty_sold DESC;
```

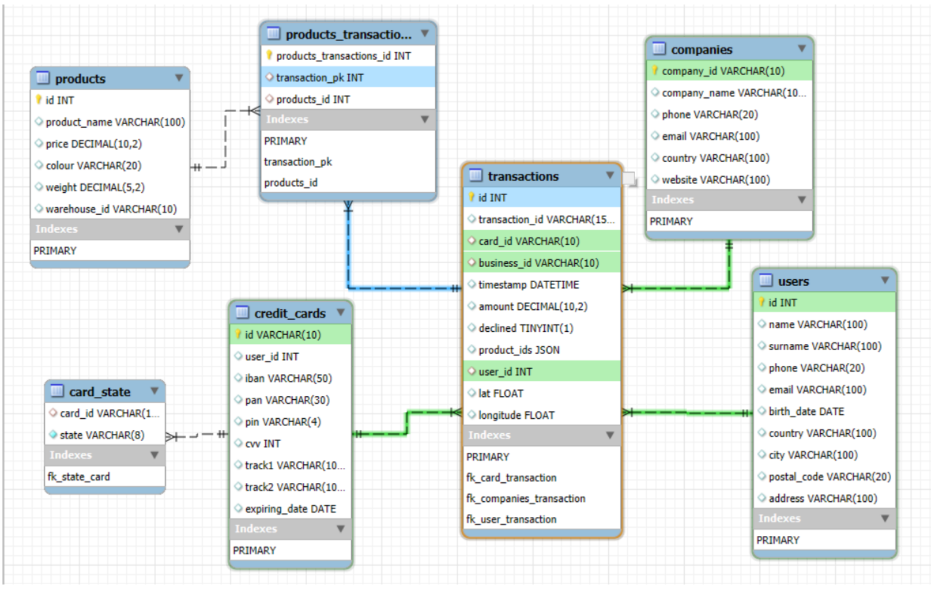


id	product_name	qty_sold
52	riverlands the duel	2642
29	Tully maester Tarly	2627
21	duel Direwolf	2603
16	the duel warden	2602
33	duel warden	2593

```
Output
Action Output
    #   Time      Action                                                                                   Message               Duration / Fetch
✓   34  11:01:25  SELECT *  FROM transactions_s4.products WHERE product_name = 'dooku solo'                 4 row(s) returned     0.000 sec / 0.000 sec
✓   35  11:01:49  SELECT products.id, products.product_name, COUNT(*) AS qty_sold FROM transactions JOIN products_trans...  100 row(s) returned   0.766 sec / 0.000 sec
```

Per saber quantes vegades s'ha venut cada producte he fet join de les taules transactions, products i products_transactions, he filtrar per transaccions no declined i agrupat per id del producte i product_name, és important fer-ho pels dos atributs ja que tenim diferents productes amb el mateix nom.

Resultat diagrama final:


products
id INT
product_name VARCHAR(100)
price DECIMAL(10,2)
colour VARCHAR(20)
weight DECIMAL(5,2)
warehouse_id VARCHAR(10)
Indexes
PRIMARY
products_transactio...
products_transactions_id INT
transaction_pk INT
products_id INT
Indexes
PRIMARY
transaction_pk
products_id
companies
company_id VARCHAR(10)
company_name VARCHAR(10...
phone VARCHAR(20)
email VARCHAR(100)
country VARCHAR(100)
website VARCHAR(100)
Indexes
PRIMARY
transactions
id INT
transaction_id VARCHAR(15...
card_id VARCHAR(10)
business_id VARCHAR(10)
timestamp DATETIME
amount DECIMAL(10,2)
declined TINYINT(1)
product_ids JSON
user_id INT
lat FLOAT
longitude FLOAT
Indexes
PRIMARY
fk_card_transaction
fk_companies_transaction
fk_user_transaction
users
id INT
name VARCHAR(100)
surname VARCHAR(100)
phone VARCHAR(20)
email VARCHAR(100)
birth_date DATE
country VARCHAR(100)
city VARCHAR(100)
postal_code VARCHAR(20)
address VARCHAR(100)
Indexes
PRIMARY
credit_cards
id VARCHAR(10)
user_id INT
iban VARCHAR(50)
pan VARCHAR(30)
pin VARCHAR(4)
cvv INT
track1 VARCHAR(10...
track2 VARCHAR(10...
expiring_date DATE
Indexes
PRIMARY
card_state
card_id VARCHAR(1...
state VARCHAR(8)
Indexes
fk_state_card