SPRINT 4:

Nivell 1:

Primer de tot hem creat la base de dades:

```sql
create database s4_transactions;
```

Tot seguit hem creat les taules transaction, companyies, credit_cards i users.

```sql
create table transactions (
id varchar(255) primary key not null,
card_id varchar(15),
business_id varchar(10),
timestamp timestamp,
amount decimal(10,2),
declined tinyint(1),
product_ids varchar(50),
user_id int(11),
lat float,
longitude float
);
```

```sql
create table credit_cards (
id varchar(15) primary key not null,
user_id int(11),
iban varchar(50),
pan varchar(50),
pin varchar(4),
cvv int(11),
track1 varchar(250),
track2 varchar(250),
expiring_date varchar(20)
);
```

```sql
create table companies (
company_id varchar(10) primary key,
company_name varchar(255),
phone varchar(15),
email varchar(100),
country varchar(100),
website varchar(100)
);
```

```sql
create table users (
id int(11) primary key not null,
name varchar(75),
surname varchar(75),
phone varchar(50),
email varchar(100),
birth_date varchar(20),
country varchar(100),
city varchar(100),
postal_code varchar(10),
address varchar(150)
);
```

A cada taula ja hem determinat la primary key de cada una i li hem afegit el NOT NULL per determinar que la relació ha de ser obligatòria. A la taula companies no li hem afegit el NOT NULL ja que pot ser que una companyia no hagi fet cap transacció.

Per importar les dades des dels csv en primer lloc hem hagut d'activar el local_infile de la següent manera:

```
16  show global variables like "local_infile";
17  set global local_infile=1;
18
```

Variable_name	Value
local_infile	ON

Al continuar sense funcionar hem afegit al fitxer my.ini de mysql el següent:

[mysqld]

secure-file-priv=""

Un cop fets aquests canvis ja hem pogut importar les dades sense cap problema:

```
19  LOAD DATA LOCAL INFILE 'C:/Users/Griselda/Desktop/Bootcamp Data Analytics/Sprint 4/transactions.csv'
20  INTO TABLE transactions
21  FIELDS TERMINATED BY ';'
22  ENCLOSED BY '"'
23  LINES TERMINATED BY '\r\n'
24  IGNORE 1 ROWS;
25
26  select * from transactions;
27
```

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

id	card_id	business_id	timestamp	amount	declined	product_ids	user_id	lat	longitude
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	2021-08-28 23:42:24	466.92	0	71, 1, 19	92	81.9185	-12.5276
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	2021-07-26 07:29:18	49.53	0	47, 97, 43	170	-43.9695	-117.525
063FBA79-99EC-66FB-29F7-25726D1764A5	CcU-2987	b-2250	2022-01-06 21:25:27	92.61	0	47, 67, 31, 5	275	-81.2227	-129.05
0668296C-CDB9-A883-76BC-2E4C44F8C8AE	CcU-3743	b-2618	2022-01-26 02:07:14	394.18	0	89, 83, 79	265	-34.3593	-100.556
06CD9AA5-9B42-D684-DDDD-A5E394FEBA99	CcU-2959	b-2346	2021-10-26 23:00:01	279.93	0	43, 31	92	33.7381	158.298
07A46D48-31A3-7E87-65B9-0DA902AD109F	CcU-3225	b-2386	2021-06-28 21:11:42	340.87	1	47, 23	272	38.8342	92.1905
09DE92CE-6F27-2BB7-13B5-9385B2B3B8E2	CcU-3071	b-2298	2021-05-11 20:40:06	303.05	1	67, 7	275	71.1706	10.5757
0A476ED9-0C13-1962-F87B-D3563924B539	CcU-4359	b-2302	2022-02-26 20:33:54	430.49	0	29, 41, 11	221	-56.4901	114.801

```
40  LOAD DATA LOCAL INFILE 'C:/Users/Griselda/Desktop/Bootcamp Data Analytics/Sprint 4/credit_cards.csv'
41  INTO TABLE credit_cards
42  FIELDS TERMINATED BY ','
43  ENCLOSED BY '"'
44  IGNORE 1 ROWS;
45
46  select * from credit_cards;
```

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

id	user_id	iban	pan	pin	cvv	track1	track2
CcU-2938	275	TR301950312213576817638661	5424465566813633	3257	984	%B8383712448554646^WovsxejDpwiev^8604...	%B7653863056044187=8007163336?3
CcU-2945	274	DO268547637485374752165686 89	5142423821948828	9080	887	%B4621311609958661^UftuyfsSeimxn^06106...	%B4149568437843501=5107140330?1
CcU-2952	273	BG45IVQL52710525608255	4556 453 55 5287	4598	438	%B2183285104307501^CddyytcUxwfdq^5907...	%B6778580257827162=6906859740?7
CcU-2959	272	CR7242477244335841535	372461377349375	3583	667	%B7281111956795320^XocddjBckecd^09016...	%B4246154489281853=2805223916?8
CcU-2966	271	BG72LKTQ15627628377363	448566 886747 7265	4900	130	%B4728932322756223^JhlgvsuFbmwgj^7202...	%B2318571115599881=8908215784?5
CcU-2973	270	PT878062281350924295 46346	544 58654 54343 384	8760	887	%B4761405253275637^HjnnipoBlejrl^7108515...	%B7816169831446746=1310277?9
CcU-2980	269	DE39241881883086277136	402400 7145845969	5075	596	%B7320483593870549^OokzqxrHpased^4901...	%B2474313962214151=04122191315?5

```
57 • LOAD DATA LOCAL INFILE 'C:/Users/Griselda/Desktop/Bootcamp Data Analytics/Sprint 4/companies.csv'
58   INTO TABLE companies
59   FIELDS TERMINATED BY ','
60   ENCLOSED BY '"'
61   LINES TERMINATED BY '\r\n'
62   IGNORE 1 ROWS;
63
64 • select * from companies;
```

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

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
b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.couk	Norway	https://nytimes.com/user/110
b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
b-2250	Amet Nulla Donec Corporation	07 15 25 14 74	mattis.integer.eu@protonmail.net	Italy	https://netflix.com/sub/cars

```
93 • LOAD DATA LOCAL INFILE 'C:/Users/Griselda/Desktop/Bootcamp Data Analytics/Sprint 4/users_ca.csv'
94   INTO TABLE users
95   FIELDS TERMINATED BY ','
96   ENCLOSED BY '"'
97   LINES TERMINATED BY '\r\n'
98   IGNORE 1 ROWS;
99
100 • select * from users;
```

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

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	Lowell	73544	348-7818 Sagittis St.
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Des Moines	59464	903 Sit Ave
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Columbus	56518	736-2063 Tellus St.
4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.edu	Feb 18, 1989	United States	Kailua	77417	Ap #545-2244 Erat. Rd.
5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	Sep 26, 1998	United States	Sandy	31564	341-2821 Ultrices Av.
6	Joel	Tyson	(718) 288-8020	gravida.nunc.sed@yahoo.ca	Oct 15, 1989	United States	Nashville	96838	888-2799 Amet Street
7	Rafael	Jimenez	(817) 689-0478	eget@outlook.ca	Dec 4, 1981	United States	Hillsboro	29874	8627 Malesuada Rd.
8	Nissim	Franks	(692) 157-3469	egestas.aliquam.fringilla@google.ca	Aug 1, 1993	United States	Jackson	61750	Ap #251-7144 Integer St.

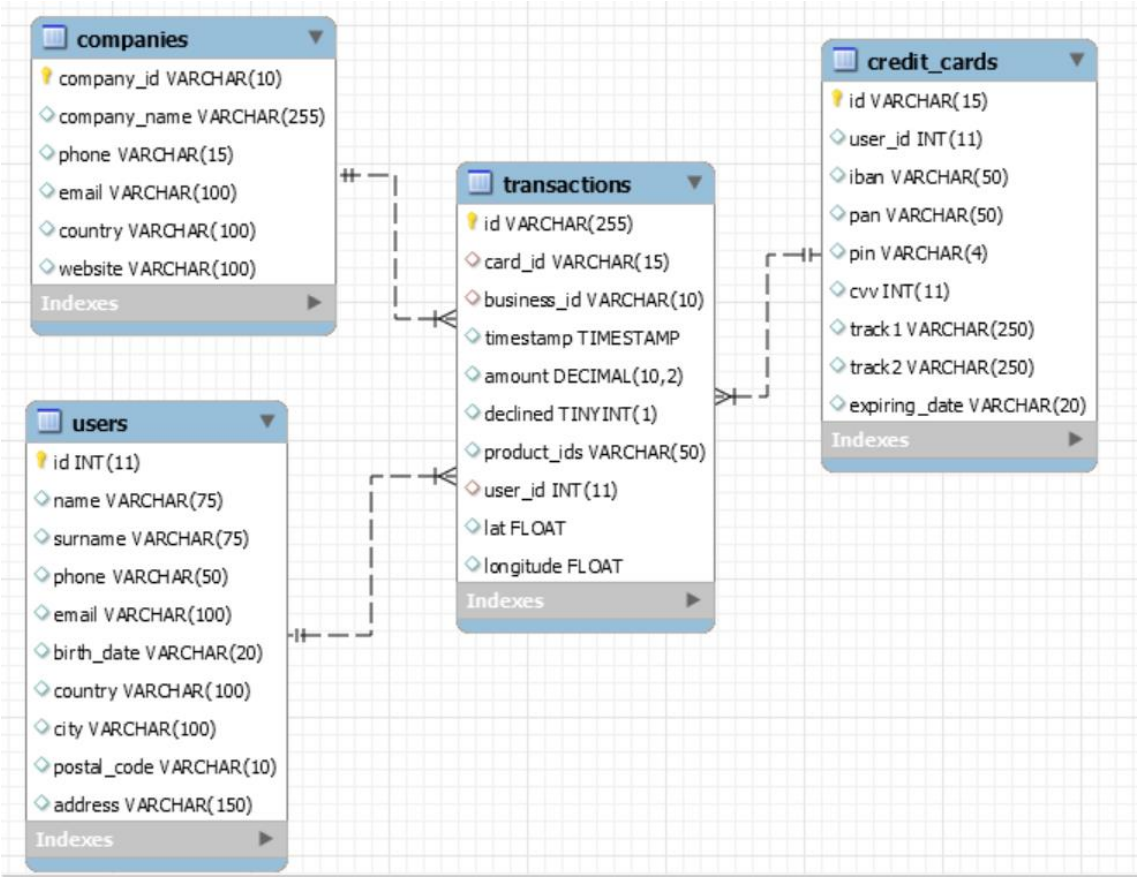
Amb users hem repetit el procés 3 cops per importar els tres fitxers de csv (users_usa, users_uk i users_ca). Només canvia el nom del fitxer en qüestió.

Un cop importats tots els fitxer, hem procedit primer a crear un índex per cada taula per optimitzar el rendiment i després a fer les unions:

```
102 • create index idx_companies on companies(company_id);
103 • create index idx_credit_cards on credit_cards(id);
104 • create index idx_transactions on transactions(id);
105 • create index idx_users on users(id);
```

```
107 • alter table transactions
108   add constraint fk_user_id foreign key (user_id) REFERENCES users(id);
109
110 • alter table transactions
111   add constraint fk_card_id foreign key (card_id) REFERENCES credit_cards(id);
112
113 • alter table transactions
114   add constraint fk_business_id foreign key (business_id) REFERENCES companies(company_id);
```

Amb els canvis ens queda el següent diagrama:



- Exercici 1: Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.

```
116  -- N1.ex1
117  select u.id, u.name, u.surname, count(user_id) transaccions from transactions t
118  left join users u on t.user_id=u.id
119  group by u.id, u.name, u.surname
120  having transaccions >30
121  order by transaccions desc;
```

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

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

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

```
122  -- N1.ex2
123  select c.company_id, c.company_name, cc.iban, round(avg(amount), 2) mitjana from transactions t
124  left join companies c on t.business_id=c.company_id
125  left join credit_cards cc on t.card_id=cc.id
126  where c.company_name = "Donec Ltd"
127  group by c.company_id, c.company_name, cc.iban;
```

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

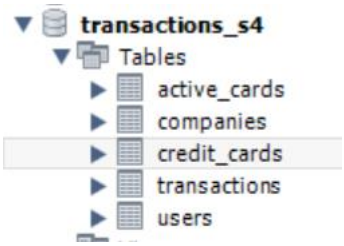
company_id	company_name	iban	mitjana
b-2242	Donec Ltd	PT87806228135092429456346	203.72

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:

Per crear la taula independent utilitzem el codi:

```
64    -- N2.ex1
65    create table active_cards as select card_id, timestamp, declined from transactions;
```




```
133   select * from active_cards;
```

Result Grid | Filter Rows: | Export:

card_id	timestamp	declined
CcU-2938	2021-08-28 23:42:24	0
CcU-4219	2021-07-26 07:29:18	0
CcU-2987	2022-01-06 21:25:27	0
CcU-3743	2022-01-26 02:07:14	0
CcU-2959	2021-10-26 23:00:01	0
CcU-3225	2021-06-28 21:11:42	1
CcU-3071	2021-05-11 20:40:06	1

- Exercici 1: Quantes targetes estan actives?

Per saber quantes targetes estan actives amb les condicions de l'enunciat utilitzem el codi:

```
135   with tarja_activa as (
136   select card_id, declined,
137       row_number () over (partition by card_id order by timestamp desc) as partition_time
138    from active_cards
139    )
140    select card_id,
141    case
142       when sum(declined) <3 then "Activa"
143       else "interval"
144       end as estat
145   from tarja_activa
146   where partition_time <=3
147   group by card_id
148   having estat = "activa";
```

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

card_id	estat
CcU-2938	Activa
CcU-2945	Activa
CcU-2952	Activa
CcU-2959	Activa
CcU-2966	Activa
CcU-2973	Activa

Result 46

En primer lloc creem una taula provisional amb el nom de tarja_activa. Aquesta taula ens mostra les columnes card_id i declined i amb la funció row_number creem una nova columna que ens numera cada fila a partir de card_id, és a dir cada card_id va de l'1 a n, ordenat pel timestamp desc, perquè les 3 últimes transaccions ens surtin les primeres de cada card_id.

Un cop tenim aquests filtres a través de la funció de case li diem que si la suma de les 3 últimes transaccions de cada card_id és inferior a 3, ens retorni "activa", ja que voldrà dir que hi ha com a mínim un moviment acceptat dins les tres últims transaccions. En cas contrari (si la suma és 3 o més) ens retornarà "inactiva".

Finalment filtrem amb un having by perquè ens mostri només les files de les targetes actives, que son un total de 275, és a dir, totes.

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:

Creem una taula amb les columnes id i product_ids de transaction i creem la taula sold_products:

```
72  -- N3.ex1
73  create table sold_products as select id, product_ids from transactions;
74
75  select * from sold_products;
76
```

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

id	product_ids
02C6201E-D90A-1859-B4EE-88D2986D3B02	71, 1, 19
0466A42E-47CF-8D24-FD01-C0B689713128	47, 97, 43
063FBA79-99EC-66FB-29F7-25726D1764A5	47, 67, 31, 5
0668296C-CDB9-A883-76BC-2E4C44F8C8AE	89, 83, 79
06CD9AA5-9B42-D684-DDDD-A5E394FEBA99	43, 31
07A46D48-31A3-7E87-65B9-0DA902AD109F	47, 23
09DE92CE-6F27-2BB7-13B5-9385B2B3B8E2	67, 7
0A476ED9-0C13-1962-F87B-D3563924B539	29, 41, 11

Creem la taula products i pugem les dades:

```
154 create table products (
155 id int(11) primary key not null,
156 product_name varchar(75),
157 price varchar(10),
158 colour varchar(50),
159 weight varchar(100),
160 warehouse_id varchar(20)
161 );
```

```
163 • LOAD DATA LOCAL INFILE 'C:/Users/Griselda/Desktop/Bootcamp Data Analytics/Sprint 4/products.csv'
164   INTO TABLE products
165   FIELDS TERMINATED BY ','
166   ENCLOSED BY '"'
167   IGNORE 1 ROWS;
168
169 • select * from products;
```

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

id	product_name	price	colour	weight	warehouse_id
1	Direwolf Stannis	$161.11	#7c7c7c	1	WH-4
2	Tarly Stark	$9.24	#919191	2	WH-3
3	duel tourney Lannister	$171.13	#d8d8d8	1.5	WH-2
4	warden south duel	$71.89	#111111	3	WH-1

- Exercici 1: Necessitem conèixer el nombre de vegades que s'ha venut cada producte.

```
171 • with prods_indiv as (
172   select id, productes.product_id
173   from sold_products
174   join json_table (concat("[", product_ids, "]" ),
175   "$[*]" columns (product_id varchar(100) path "$")
176    )  as productes
177   )
178   select product_id, count(product_id) recompte from prods_indiv
179   group by product_id
180   order by recompte desc;
```

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

product_id	recompte
67	68
23	68
79	66
43	65

Per poder fer el recompte, primer hem de separar els productes en files diferents. Per fer-ho utilitzem json_table per convertir les cel·les de la columna product_ids en format json. Perquè faci la separació correcta, hem d'afegir el $[*] que fa que separi cada número a través de la funció columns. Un cop ho hem creat, amb la taula auxiliar prods_indiv ja podem fer el recompte de cada producte.