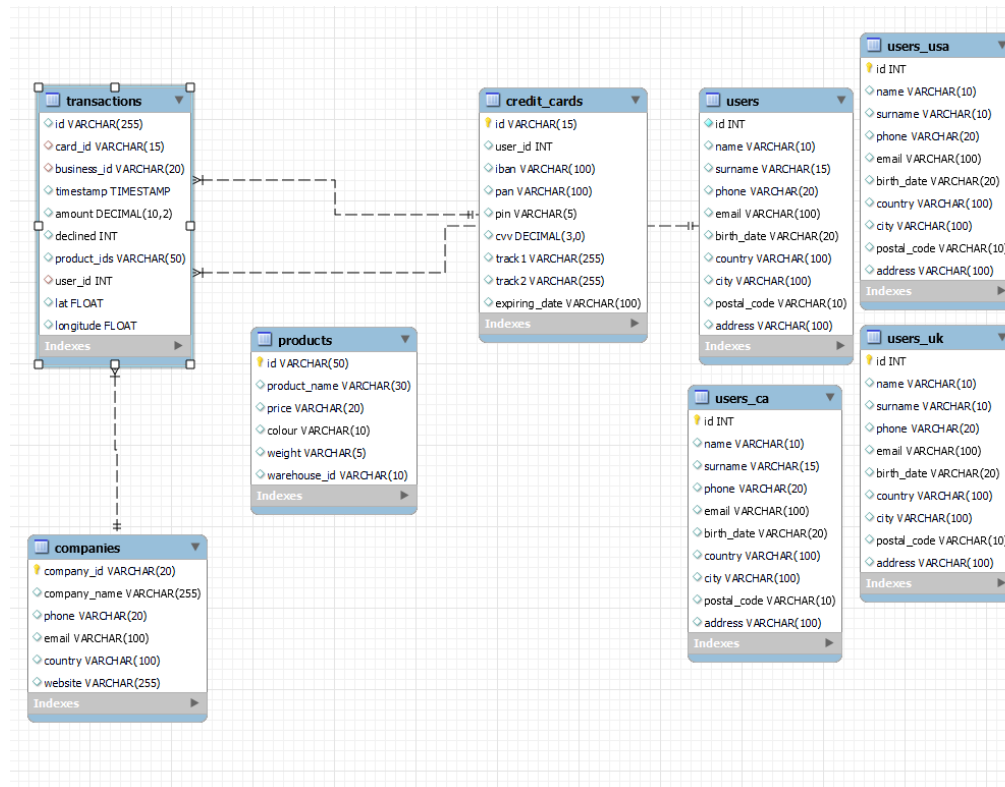


Nivel 1

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:



En la imatge està el diagrama de estrella, per algun motiu no surt la relació entre products i transactions, aunque en el código sí que figura. Para poder facilitar la relación con los users, cree la tabla users uniendo todos los datos de users_ca, users_uk y users ya que sus ids son nominales consecutivas y no repetidas.

```
-- Creamos la tabla transactions
CREATE TABLE IF NOT EXISTS transactions (
  id VARCHAR(255),
  card_id VARCHAR(15) references credit_cards(id),
  business_id varchar (20) references companies(company_id),
  timestamp TIMESTAMP,
  amount decimal (10,2),
  declined int,
  product_ids varchar(50) references products(id),
  user_id INT references users(id),
  lat float,
  longitude float,
  FOREIGN KEY (card_id) REFERENCES credit_cards(id),
  FOREIGN KEY (business_id) REFERENCES companies(company_id),
  FOREIGN KEY (product_ids) REFERENCES products(id),
  foreign key (user_id) references users(id)
);
```

Álvaro Míguez

Sprint 04

```
87 -- Creamos la tabla transactions
88 CREATE TABLE IF NOT EXISTS transactions (
89     id VARCHAR(255),
90     card_id VARCHAR(15) REFERENCES credit_cards(id),
91     business_id VARCHAR (20) REFERENCES companies(company_id),
92     timestamp TIMESTAMP,
93     amount decimal (10,2),
94     declined int,
95     product_ids VARCHAR(50) REFERENCES products(id),
96     user_id INT REFERENCES users(id),
97     lat float,
98     longitude float,
99     FOREIGN KEY (card_id) REFERENCES credit_cards(id),
100    FOREIGN KEY (business_id) REFERENCES companies(company_id),
101    FOREIGN KEY (product_ids) REFERENCES products(id),
102    foreign key (user_id) REFERENCES users(id)
103 );
104
105 alter table transactions drop constraint transactions_ibfk_4; #drop constraint para poder hacer la relacion entre users y la tabla de transactions
106 Alter table transactions add foreign key (user_id) REFERENCES users(id);
107 CREATE INDEX transactions_ibfk_4 ON users(id); #create index para la tabla de transactions y añadir los datos de "datos_insertar_sprint_04"
108 alter table transactions drop constraint transactions_ibfk_3; #drop constraint en caso de que haya problemas entre products y transactions
109 alter table transactions modify product_ids VARCHAR(50); #modificación hecha posteriori en transactions(product_ids y products(id), valor original INT.
110 alter table products modify id VARCHAR(50);
```

He tenido que hacer varias alteraciones de tablas ya que me saltaban errores de constraint, de primary y foreign keys y dejo constancia de la modificación de products(id) y transactions(product_ids).

Recomiendo cargar las tablas de companies, credit_cards, products y users antes de cargar la tabla de transactions.

Nota añadida: es posible que la relación entre transactions y products no funcionase ya que tuve que hacer un drop de la tabla de products para poder gestionar las keys y el cambio de tipo de valor (originalmente INT después VARCHAR(50)) ya que al introducir los datos me daba error "Error Code: 1265. Data truncated for column 'product_ids' at row 2". Seguramente cargadas las tablas desde cero no surja este problema.

Álvaro Míguez
Sprint 04

Para insertar los datos tuve que utilizar dos conversores diferentes.

<https://konbert.com/convert/csv/to/tsv> que me daba la conversión más precisa directamente de archivo a insert into de mysql y <https://www.convertcsv.com/> que me ayudó a limpiar primero el archivo de transactions antes de pasarlo por www.konbert.com

Como uso el ordenador del aula no podía añadir los archivos mediante query.

```
LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.3\Uploads\copia excels/transactions.csv'
INTO TABLE transactions
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

```
"11:37:07      LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.3\Uploads\copia
excels/transactions.csv' INTO TABLE transactions FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\n' IGNORE 1 ROWS
```

Error Code: 1290. The MySQL server is running with the --secure-file-priv option so it cannot execute this statement 0.000 sec"

En el código subido a github saldrán con # para que no sean interpretados cuando se utilice el script.

Un problema añadido que tuve fue que en mi ordenador privado tampoco pude subir los archivos. Me decía que tenía que utilizar la carpeta designada por mysqlbench. Utilice la siguiente función para averiguar que carpeta era.

```
# SHOW VARIABLES LIKE "secure_file_priv";
```

Aunque me dio el resultado y pude acceder a la carpeta, no me dejó subir los archivos. Por eso utilice los conversores.

Álvaro Míguez

Sprint 04

- Exercici 1

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

Usando las tablas transactions y users podemos sacar nombre y apellidos y hacer en el user_id de transactions para ver cuántas veces figuran en la tabla de transactions y así saber el número de pedidos. He añadido, además, la suma total de sus pedidos y el id de usuario.

```
8      #ex 1
9
10
11 •   SELECT users.name, users.surname, users.id, count(transactions.user_id) as ct, sum(amount)
12     FROM transactions
13     INNER JOIN users on users.id = user_id
14     GROUP BY users.name, users.surname, users.id, transactions.user_id
15     HAVING ct>=30
16     order by ct desc;
17
18 •   select count(user_id) as total_orders, sum(amount), users.name, users.surname, users.id
19     from transactions
20     INNER JOIN users on users.id = user_id
21     GROUP BY users.name, users.surname, users.id, transactions.user_id
22     having count(transactions.user_id) >= 30;
23
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	name	surname	id	ct	sum(amount)
▶	Hedwig	Gilbert	272	76	18351.30
	Ocean	Nelson	267	52	13052.24
	Kenyon	Hartman	275	48	12011.56
	Lynn	Riddle	92	39	11451.57

Para ver la 2ª formula en acción y ver que dan los mismos resultados

```
8      #ex 1
9
10
11 •   SELECT users.name, users.surname, users.id, count(transactions.user_id) as ct, sum(amount)
12     FROM transactions
13     INNER JOIN users on users.id = user_id
14     GROUP BY users.name, users.surname, users.id, transactions.user_id
15     HAVING ct>=30
16     order by ct desc;
17
18 •   select count(user_id) as total_orders, sum(amount), users.name, users.surname, users.id
19     from transactions
20     INNER JOIN users on users.id = user_id
21     GROUP BY users.name, users.surname, users.id, transactions.user_id
22     having count(transactions.user_id) >= 30;
23
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

total_orders	sum(amount)	name	surname	id
48	12011.56	Kenyon	Hartman	275
76	18351.30	Hedwig	Gilbert	272
52	13052.24	Ocean	Nelson	267
39	11451.57	Lynn	Riddle	92

Álvaro Míguez

Sprint 04

- 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.

Usando las tablas, transactions, users, credit_cards y companies podemos encontrar la compañía de Donec Ltd, en credit_cards la información del iban y de users el id de la tarjeta y juntándolo con transacciones para ver los pedidos y su cantidad para hacer la media (avg).

```
24
25 #ex 2
26 • select credit_cards.iban as iban, transactions.card_id, avg(amount), transactions.user_id, companies.company_name from credit_cards
27 JOIN transactions ON transactions.card_id = credit_cards.id
28 join users on credit_cards.user_id = users.id
29 JOIN companies ON transactions.business_id = company_id
30 where iban = 'PT87806228135092429456346'
31 group by iban, transactions.card_id, transactions.user_id, company_name
32 ;
33
```

iban	card_id	avg(amount)	user_id	company_name
PT87806228135092429456346	CcU-2973	203.715000	275	Donec Ltd

Nivel 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:

Exercici 1

Quantes targetes estan actives?

Juntando la información de credit_cards, users y transaction, podemos contar las veces que se les ha declinado una compra. En este caso solo hay 3 usuarios que tendrían las tarjetas desactivadas. Los usuarios 272, 275 y 267.

```
34 #ex 2.1
35 • select sum(declined) as declined_total, transactions.user_id, if(sum(declined) >= 3, "inactiva", "activa") as activacion from transactions
36 JOIN credit_cards ON transactions.card_id = credit_cards.id
37 join users on transactions.user_id = users.id
38 group by user_id
39 order by declined_total desc;
```

declined_total	user_id	activacion
38	272	inactiva
24	275	inactiva
13	267	inactiva
2	274	activa
2	273	activa
2	271	activa
2	270	activa
2	269	activa
2	268	activa
0	92	activa
0	91	activa
0	90	activa
0	89	activa
0	88	activa
0	87	activa
0	86	activa
0	85	activa
0	84	activa
0	83	activa
0	82	activa
0	81	activa
0	80	activa
0	96	activa
0	95	activa
0	94	activa
0	93	activa
0	136	activa
0	135	activa
0	134	activa

Nivel 3

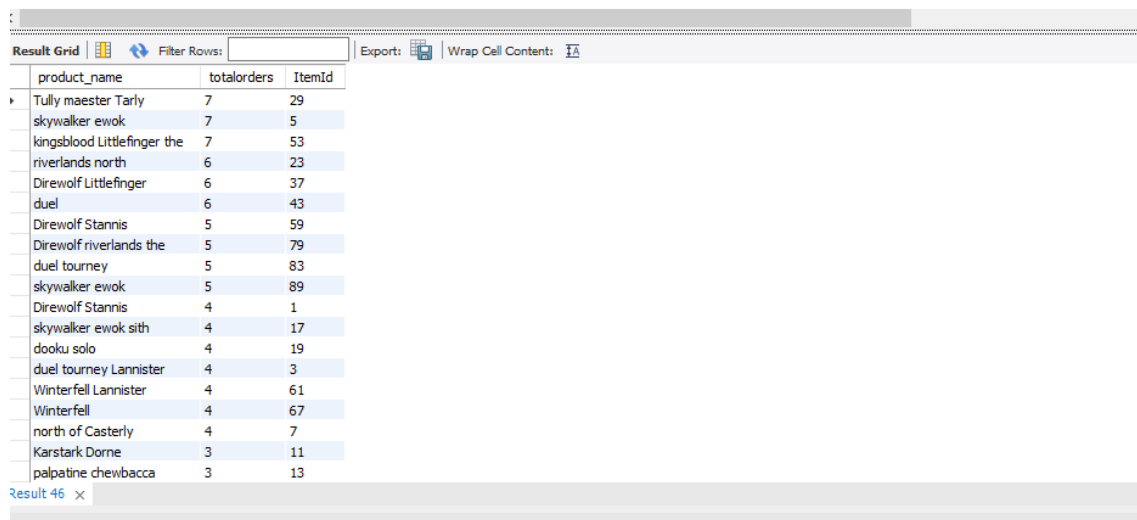
Crea una taula amb la qual puguem unir les dades del nou arxiu products.csv amb la base de dades creada, tenint en compte que des de transaction tens product_ids. Genera la següent consulta:

Exercici 1

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

Haciendo el count en product_ids de transactions podemos contar el número de veces que se han pedido. Juntándolo con la tabla de products, podemos ver el Id de producto y su nombre.

```
41 #ex 3.1
42 • select product_name, count(product_ids) as totalorders, products.id as ItemId from transactions
43 join products on product_ids = products.id
44 group by product_name, itemid
45 order by totalorders desc;
```



The screenshot shows a database interface with a query result grid. The grid has three columns: product_name, totalorders, and ItemId. The results are sorted by totalorders in descending order. The first row is 'Tully maester Tarly' with 7 totalorders and ItemId 29. The last row is 'palpatine chewbacca' with 3 totalorders and ItemId 13.

product_name	totalorders	ItemId
Tully maester Tarly	7	29
skywalker ewok	7	5
kingsblood Littlefinger the	7	53
riverlands north	6	23
Direwolf Littlefinger	6	37
duel	6	43
Direwolf Stannis	5	59
Direwolf riverlands the	5	79
duel tourney	5	83
skywalker ewok	5	89
Direwolf Stannis	4	1
skywalker ewok sith	4	17
dooku solo	4	19
duel tourney Lannister	4	3
Winterfell Lannister	4	61
Winterfell	4	67
north of Casterly	4	7
Karstark Dorne	3	11
palpatine chewbacca	3	13

```
41 #ex 3.1
42 • select product_name, count(product_ids) as totalorders, products.id as ItemId from transactions
43 join products on product_ids = products.id
44 group by product_name, itemid
45 order by totalorders desc;
46
47 • select * from transactions
48 join products on product_ids = products.id
49 where product_ids = "47";
50
51 • select * from products
52 where product_name = 'direwolf stannis';
53
54
55
--
```

Para poder hacer otras pruebas de conteo, buscaba un id de producto concreto como se puede ver en la 2ª formula. O buscaba un nombre para ver si se correspondía.

Álvaro Míguez
Sprint 04