

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:

```
8 • CREATE TABLE IF NOT EXISTS companies (  
9     company_id varchar(15) primary key,  
10    company_name varchar(255),  
11    phone varchar(15),  
12    email varchar(100),  
13    country varchar(100),  
14    website varchar(255)  
15 );  
16  
17 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv'  
18 INTO TABLE companies  
19 FIELDS TERMINATED BY ',' -- Especifica que las columnas están separadas por comas  
20 ENCLOSED BY '"' -- Si las cadenas están entre comillas dobles  
21 LINES TERMINATED BY '\n' -- Especifica que cada línea representa un registro  
22 IGNORE 1 ROWS; -- Ignora la primera fila (cabeceras), si las tiene  
  
25 • 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

```

27 • CREATE TABLE IF NOT EXISTS credit_cards (
28     id varchar(20) primary key,
29     user_id int,
30     iban varchar(50),
31     pan varchar(20),
32     pin varchar(4),
33     cvv int,
34     track1 varchar(50),
35     track2 varchar(50),
36     expiring_date varchar(20),
37     FOREIGN KEY(user_id) REFERENCES users(id)
38 );

```

```

40 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv'
41 INTO TABLE credit_cards
42 FIELDS TERMINATED BY ',' -- Especifica que las columnas están separadas por comas
43 ENCLOSED BY '"' -- Si las cadenas están entre comillas dobles
44 LINES TERMINATED BY '\n' -- Especifica que cada línea representa un registro
45 IGNORE 1 ROWS; -- Ignora la primera fila (cabeceras), si las tiene
46
47 • SELECT * FROM credit_cards;

```

Result Grid								
Filter Rows: <input type="text"/> 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...	%B76...
	CcU-2945	274	DO26854763748537475216568689	5142423821948828	9080	887	%B4621311609958661^UftuyfsSeimxn^06106...	%B4...
	CcU-2952	273	BG45IVQL52710525608255	4556 453 55 5287	4598	438	%B2183285104307501^CddytcUxwfdq^5907...	%B6...

```





49 • CREATE TABLE IF NOT EXISTS products (
50     id int primary key,
51     product_name varchar(50),
52     price varchar(15),
53     colour varchar(15),
54     weight float,
55     warehouse_id varchar(15)
56 );
57
58 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv'
59 INTO TABLE products
60 FIELDS TERMINATED BY ',' -- Especifica que las columnas están separadas por comas
61 ENCLOSED BY '"' -- Si las cadenas están entre comillas dobles
62 LINES TERMINATED BY '\n' -- Especifica que cada línea representa un registro
63 IGNORE 1 ROWS; -- Ignora la primera fila (cabeceras), si las tiene

```

```

65 • SELECT * FROM products;

```

Result Grid					
Filter Rows: <input type="text"/>					
Edit:    					
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

```







67 • CREATE TABLE IF NOT EXISTS users (
68     id int primary key,
69     name varchar(100),
70     surname varchar(100),
71     phone varchar(150),
72     email varchar(150),
73     birth_date varchar(100),
74     country varchar(150),
75     city varchar(150),
76     postal_code varchar(100),
77     address varchar(255)
78 );

```

```

80 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv'
81 INTO TABLE users
82 FIELDS TERMINATED BY ',' -- Especifica que las columnas están separadas por comas
83 ENCLOSED BY '"' -- Si las cadenas están entre comillas dobles
84 LINES TERMINATED BY '\r\n' -- Especifica que cada línea representa un registro cuando hay espacios dentro
85 IGNORE 1 ROWS; -- Ignora la primera fila (cabeceras), si las tiene
86
87 • SELECT * FROM users;

```







Result Grid										
Filter Rows: <input type="text"/>										
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	348-70	
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Des Moines	59464	903 Si	
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Columbus	56518	736-20	

```

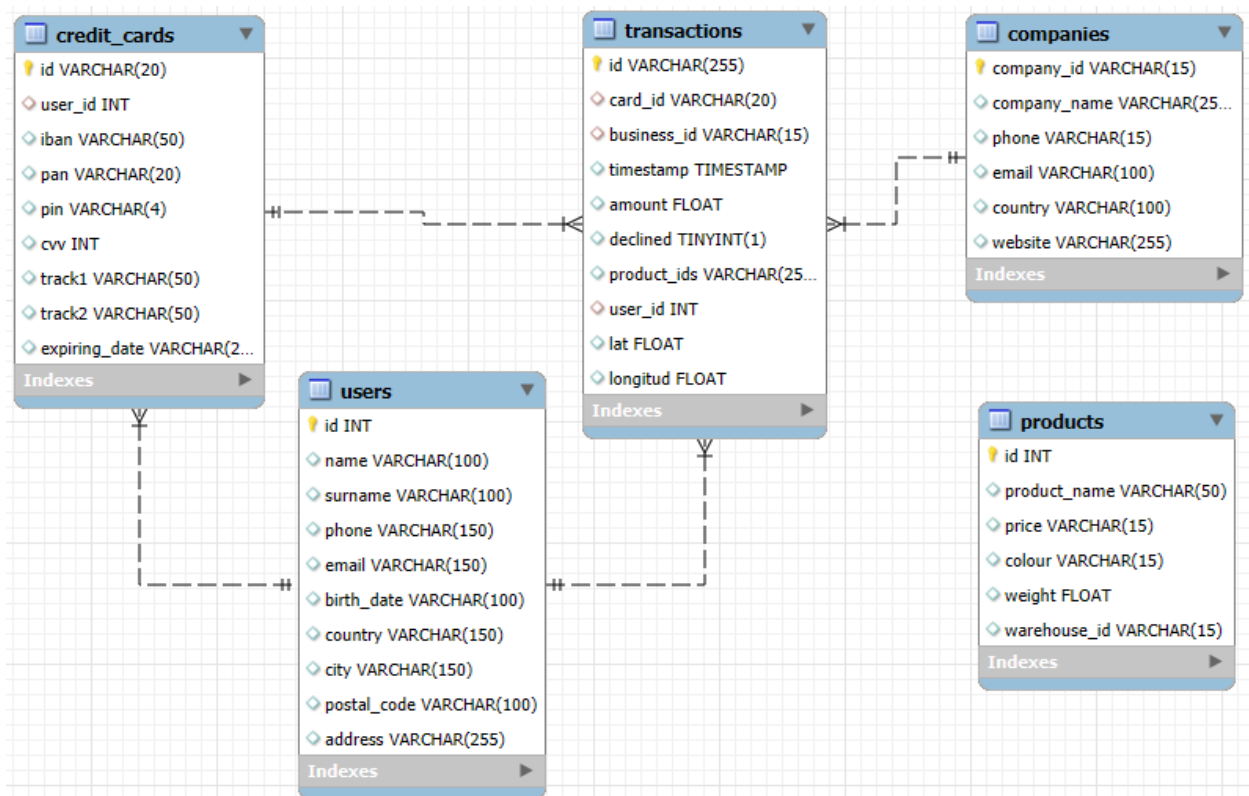
89 • CREATE TABLE IF NOT EXISTS transactions (
90     id varchar(255) primary key,
91     card_id varchar(20),
92     business_id varchar(15),
93     timestamp timestamp,
94     amount float,
95     declined boolean,
96     product_ids varchar(255),
97     user_id int,
98     lat float,
99     longitud float,
100     FOREIGN KEY(card_id) REFERENCES credit_cards(id),
101     FOREIGN KEY(business_id) REFERENCES companies(company_id),
102     FOREIGN KEY(user_id) REFERENCES users(id)
103 );

105 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'
106 INTO TABLE transactions
107 FIELDS TERMINATED BY ';' -- Especifica que las columnas están separadas por punto y coma
108 ENCLOSED BY '"' -- Si las cadenas están entre comillas dobles
109 LINES TERMINATED BY '\r\n' -- Especifica que cada línea representa un registro cuando hay espacios dentro
110 IGNORE 1 ROWS; -- Ignora la primera fila (cabeceras), si las tiene
111
112 -- Si hay comas dentro de un campo, cargar con el wizard pero especificar en "configure import setting" que
113 • SELECT * FROM transactions;

```

Result Grid									
Filter Rows: <input type="text"/>									
Edit:    Export/Import:   Wrap Cell Content: 									
id	card_id	business_id	timestamp	amount	declined	product_ids	user_id	lat	
02C6201E-D90A-1859-84EE-88D2986D3B02	CcU-2938	b-2362	2021-08-28 23:42:24	466.92	0	71, 1, 19	92	81.9185	
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	2021-07-26 07:29:18	49.53	0	47, 97, 43	170	-43.9695	
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	

Con lo que el esquema de la Base de Datos queda de esta manera:



- Exercici 1

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

```
117 • SELECT id, name, surname
118 FROM users
119 WHERE id IN (SELECT user_id
120 FROM transactions t
121 WHERE declined=0
122 GROUP BY user_id
123 HAVING count(id) > 30)
124 ORDER BY id;
```


Result Grid			
	id	name	surname
▶	92	Lynn	Riddle
	267	Ocean	Nelson
	272	Hedwig	Gilbert
*	NULL	NULL	NULL


- Exercici 2


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


```
88 • SELECT co.company_id, co.company_name, cc.iban, round(avg(t.amount),2) as media_importe
89 FROM transactions t
90 LEFT JOIN credit_cards cc
91 ON cc.id = t.card_id
92 LEFT JOIN companies co
93 ON co.company_id = t.business_id
94 WHERE co.company_name = 'Donec Ltd' and t.declined=0
95 GROUP BY co.company_id, co.company_name, cc.iban;
96
```

Result Grid



 Filter Rows:

Export: 

Wrap Cell Content: 

company_id	company_name	iban	media_importe
b-2242	Donec Ltd	PT87806228135092429456346	42.82

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:

```
142 • CREATE TABLE credit_cards_estado AS (  
143     SELECT  
144     a.id,  
145     a.user_id,  
146     a.iban,  
147     CASE WHEN sum(a.declined) >= 3 THEN 'Inactiva' ELSE 'Activa' END AS estado_tarjeta  
148     FROM (SELECT  
149         cc.id,  
150         cc.user_id,  
151         cc.iban,  
152         t.timestamp,  
153         t.declined,  
154         ROW_NUMBER() OVER (PARTITION BY cc.id ORDER BY t.timestamp DESC) AS num_operaciones  
155     FROM credit_cards cc  
156     LEFT JOIN transactions t  
157         ON cc.id = t.card_id  
158     ORDER BY cc.id, t.timestamp DESC  
159     ) a  
160     WHERE a.num_operaciones <= 3  
161     GROUP BY a.id, a.user_id, a.iban  
162 );  
163  
164 • ALTER TABLE credit_cards_estado  
165     ADD CONSTRAINT fk_credit_cards_estado  
166     FOREIGN KEY (id) REFERENCES credit_cards(id);  
167  
168 • SELECT * FROM credit_cards_estado;
```

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

	id	user_id	iban	estado_tarjeta
►	CcU-2938	275	TR301950312213576817638661	Activa
	CcU-2945	274	DO26854763748537475216568689	Activa
	CcU-2952	273	BG45IVQL52710525608255	Activa
	CcU-2959	272	CR7242477244335841535	Activa
	CcU-2966	271	BG72LKTQ15627628377363	Activa

Exercici 1

Quantes targetes estan actives?

```
128 • SELECT count(*) as num_tarjetas_activas
129 FROM credit_cards_estado
130 WHERE estado_tarjeta='Activa';
```

Result Grid		Filter Rows:	Export:
	num_tarjetas_activas		
▶	275		

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:

```
135 • CREATE TABLE transactions_products AS (  
136   WITH RECURSIVE numeros AS (  
137     SELECT 1 AS n  
138     UNION ALL  
139     SELECT n + 1  
140     FROM numeros  
141     WHERE n <= 100 -- Cambiar según el número máximo de elementos en la lista  
142   )  
143   SELECT  
144     t.id,  
145     TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(t.product_ids, ',', numeros.n), ',', -1)) AS product_id  
146   FROM transactions t  
147   LEFT JOIN numeros  
148     ON numeros.n <= 1 + LENGTH(t.product_ids) - LENGTH(REPLACE(t.product_ids, ',', ''))  
149   WHERE TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(t.product_ids, ',', numeros.n), ',', -1)) <> ''  
150   ORDER BY t.id, numeros.n  
151 );  
152  
153 • SELECT * FROM transactions_products;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	id	product_id			
▶	02C6201E-D90A-1859-B4EE-88D2986D3B02	71			
	02C6201E-D90A-1859-B4EE-88D2986D3B02	1			
	02C6201E-D90A-1859-B4EE-88D2986D3B02	19			
	0466A42E-47CF-8D24-FD01-C0B689713128	47			

```
197 • ALTER TABLE transactions_products  
198   ADD CONSTRAINT fk_transactions_products  
199   FOREIGN KEY (id) REFERENCES transactions(id);  
200  
201 • ALTER TABLE transactions_products  
202   MODIFY COLUMN product_id INT;  
203  
204 • ALTER TABLE transactions_products  
205   ADD CONSTRAINT fk_transactions_products2  
206   FOREIGN KEY (product_id) REFERENCES products(id);  
---
```

Exercici 1

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

```
157 • SELECT p.id, p.product_name, count(tp.product_id) as num_ventas_producto
158 FROM products p
159 LEFT JOIN transactions_products tp
160 ON tp.product_id = p.id
161 GROUP BY p.id, p.product_name;
```

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

	id	product_name	num_ventas_producto
▶	1	Direwolf Stannis	61
	2	Tarly Stark	65
	3	duel tourney Lannister	51
	4	warden south duel	0
	5	skywalker ewok	49

Finalmente el esquema de la base de datos queda como sigue:

