

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:

- 1) En primer lloc he carregat els Csv en Xcell per mirar una mica la informació que contenen.

Consultes i connexions

Queries | Connections

6 queries

- products
100 rows loaded.
- european_users
3.990 rows loaded.
- transactions
100.000 rows loaded.
- american_users
1.010 rows loaded.
- companies
101 rows loaded.
- credit_cards
5.000 rows loaded.

- 2) Els productes sembla que conté una llista de noms generats combinant elements de l'univers de Joc de Trons i Star Wars (com Direwolf Stannis, Skywalker Ewok, Dooku Solo, etc.) per la qual cosa he decidit crear una base de dades amb nom Skycaster.

```

3 • CREATE DATABASE Skycaster
4 CHARACTER SET utf8mb4
5 COLLATE utf8mb4_general_ci;
6
7 • USE Skycaster;
8

```

Output

Action Output			Message
#	Time	Action	
1	11:05:27	CREATE DATABASE Skycaster CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci	1 row(s) affected
2	11:05:29	USE Skycaster	0 row(s) affected

CHARACTER SET utf8 →

CHARACTER SET utf8mb4 → Defineix el conjunt de caràcters per defecte per a aquesta base. indica que la base farà servir UTF-8 i utf8mb4 és la versió completa d'UTF-8 a MySQL. Suporta tots els caràcters Unicode, inclosos emojis 😊, símbols asiàtics, etc. (es recomana usar utf8mb4 en lloc d'utf8, perquè utf8 a MySQL només admet fins a 3 bytes i no pot manejar certs caràcters).

COLLATE utf8_general_ci → defineix la col·locació (regles d'ordenament i comparació). ci significa case-insensitive (no distingeix majúscules/minúscules). (Si vols que distingeixi majúscules/minúscules, faràs servir tf8_general_cs.)

3) Ara hem de crear les diferents taules: com les creo des del principi estableixo que utilitzaré id_XXX per les claus primàries i XXX_id per les claus foranies en les altres taules. En les diferents taules he considerat NOT NULL la informació que considero rellevant per crear un registre (tot i que de vegades després m'ha donat errors i de vegades he hagut d'eliminar alguns NOT NULL).

En la taula productes el pes està en números enters i així ho creo (tot i que pensant en productes futurs podríem haver pensat en fer-ho en DECIMAL). He creat una única taula d'usuaris on pujaré les dades dels dos fitxers i crearé una nova columna amb el nom del continent. En general utilitzo VARCHAR(255) i VARCHAR(100) per crear tots els camps.

Taules principals:

1. **company**: Empreses (comerços o negocis). Hi ha 100 companies.
2. **credit_card**: Targetes de crèdit associades als usuaris. Hi ha 5000 targetes.
3. **product**: Productes disponibles. Hi ha 100 productes. Aquesta taula no es pot lligar directament amb la taula de transaccions perquè en la taula de transaccions hi ha més d'un id_producte per transacció (necessitarem una taula intermèdia que la connecti amb transaccions i trenqui la relació molts a molts).
4. **users**: Usuaris/clients. Hi ha 5000 usuaris (3990 europeus i 1010 americans)
5. **transactions**: Operacions fetes amb targetes (relacions entre usuaris, targetes, empreses i productes). Es la taula de fets. les altres quatre són taules de dimensions. Hi ha 100.000 transaccions.

Creem les 5 taules en aquest ordre:

```

8
9 • CREATE TABLE IF NOT EXISTS company (
10     id_company VARCHAR(100) PRIMARY KEY,
11     company_name VARCHAR(255) NOT NULL,
12     phone VARCHAR(15),
13     email VARCHAR(100),
14     country VARCHAR(100),
15     website VARCHAR(255)
16 );
17
18 • CREATE TABLE IF NOT EXISTS credit_card (
19     id_card VARCHAR(100) PRIMARY KEY, -- Identificador únic de la targeta
20     user_id VARCHAR(100) NOT NULL, -- el identificador d'usuari
21     iban VARCHAR(34) NOT NULL, -- El IBAN pot tenir fins 34 caracters segons standard
22     pan VARCHAR(20) NOT NULL, -- PAN (Primary Account Number, 16 digits)
23     pin CHAR(4) NOT NULL, -- PIN (4 dígitos)
24     cvv CHAR(3) NOT NULL, -- CVV (3 dígitos)
25     track1 VARCHAR(255), -- texte
26     track2 VARCHAR(255), --
27     expiring_date VARCHAR(20) -- Data de caducitat
28 );

```

Output:

Action Output

#	Time	Action	Message
2	11:48:04	USE Skycaster	0 row(s) affected
3	11:48:15	CREATE TABLE IF NOT EXISTS company (id_company VARCHAR(100) PRIMARY KEY, company_name ...	0 row(s) affected
4	11:48:47	CREATE TABLE IF NOT EXISTS credit_card (id_card VARCHAR(100) PRIMARY KEY, - Identificador únic...	0 row(s) affected

```

30 • CREATE TABLE IF NOT EXISTS product (
31     id_product VARCHAR(100) PRIMARY KEY,
32     product_name VARCHAR(255) NOT NULL,
33     price VARCHAR(50) NOT NULL,
34     colour VARCHAR(100),
35     weight INT,
36     warehouse_id VARCHAR(15) NOT NULL
37 );
38
39 • CREATE TABLE IF NOT EXISTS users (
40     id_user VARCHAR(100) PRIMARY KEY,
41     name VARCHAR(100) NOT NULL,
42     surname VARCHAR(100) NOT NULL,
43     phone VARCHAR(20),
44     email VARCHAR(150),
45     birth_date VARCHAR(100),
46     country VARCHAR(150),
47     city VARCHAR(150),
48     postal_code VARCHAR(20),
49     address VARCHAR(255)
50 );

```

Output:

Action Output

#	Time	Action	Message
3	11:48:15	CREATE TABLE IF NOT EXISTS company (id_company VARCHAR(100) PRIMARY KEY, company_name ...	0 row(s) affected
4	11:48:47	CREATE TABLE IF NOT EXISTS credit_card (id_card VARCHAR(100) PRIMARY KEY, - Identificador únic...	0 row(s) affected
5	11:49:06	CREATE TABLE IF NOT EXISTS product (id_product VARCHAR(100) PRIMARY KEY, product_name VAR...	0 row(s) affected
6	11:52:08	CREATE TABLE IF NOT EXISTS users (id_user VARCHAR(100) PRIMARY KEY, name VARCHAR(100) N...	0 row(s) affected

```

52 • CREATE TABLE IF NOT EXISTS transactions (
53     id_transactions VARCHAR(255) PRIMARY KEY,
54     card_id VARCHAR(100) NOT NULL,
55     business_id VARCHAR(100) NOT NULL,
56     timestamp DATETIME NOT NULL,
57     amount DECIMAL(10, 2) NOT NULL,
58     declined boolean NOT NULL DEFAULT 0, -- 0 = acceptada, 1 = denegada
59     product_id VARCHAR(100) NOT NULL,
60     user_id VARCHAR(100) NOT NULL,
61     lat DECIMAL(9,6),
62     longitude DECIMAL(9,6),
63     FOREIGN KEY (card_id) REFERENCES credit_card(id_card),
64     FOREIGN KEY (business_id) REFERENCES company(id_company),
65     FOREIGN KEY (user_id) REFERENCES users(id_user)
66 );

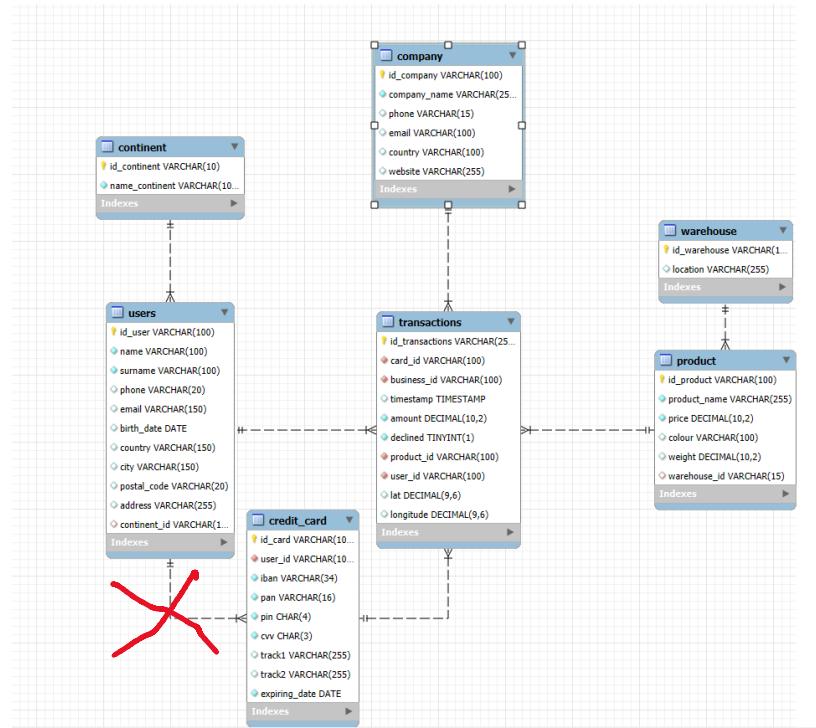
```

Output

Action Output			Message
#	Time	Action	
✓ 4	11:48:47	CREATE TABLE IF NOT EXISTS credit_card (id_card VARCHAR(100) PRIMARY KEY, -- Identificador únic...	0 row(s) affected
✓ 5	11:49:06	CREATE TABLE IF NOT EXISTS product (id_product VARCHAR(100) PRIMARY KEY, product_name VAR...	0 row(s) affected
✓ 6	11:52:08	CREATE TABLE IF NOT EXISTS users (id_user VARCHAR(100) PRIMARY KEY, name VARCHAR(100) N...	0 row(s) affected
✓ 7	11:52:54	CREATE TABLE IF NOT EXISTS transactions (id_transactions VARCHAR(255) PRIMARY KEY, card_id VA...	0 row(s) affected

Lògica i relacions:

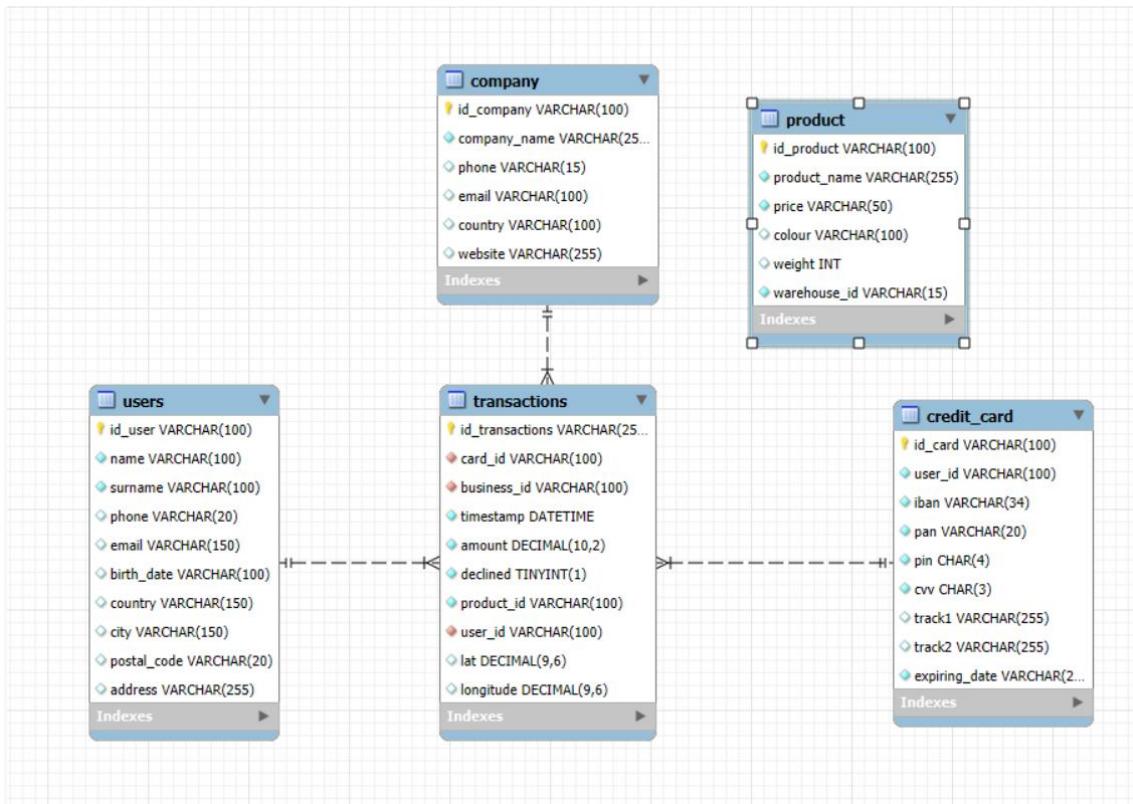
1. **company**: desa informació de l'empresa (nom, telèfon, email, país, web).
 - o Es relaciona amb les transaccions com a business_id.
2. **credit_card**: desa dades sensibles de la targeta (IBAN, PAN, PIN, CVV, data caducitat, pistes magnètiques).
 - o Cada targeta està vinculada a un user_id (usuari), però no creo la clau forànea perquè sinó es crea una referència circular (com es mostra a la figura següent; que va ser la meva primera opció i que descarto en la versió final).



- `expiring_date` està com a `VARCHAR`, ho passaré a `DATE` quan faci la càrrega per fer comparacions fàcils.

3. **product**: desa productes (nom, preu, color, pes, magatzem).
 - `price` està definit com a `VARCHAR`, però normalment hauria de ser `DECIMAL` per operacions matemàtiques. Ho transformaré durant la càrrega de les dades.
4. **users**: desa dades personals dels clients (nom, cognoms, telèfon, email, adreça, data de naixement, etc.).
 - Clau primària: `id_user`.
4. **transactions**: desa les operacions realitzades amb una targeta.
 - Té vincles foranis amb: `credit_card(id_card)`, `company(id_company)` i `users(id_user)`
 - Conté:
 - `amount` (import de la transacció).
 - `declined` (booleà per si ha estat rebutjada o acceptada).
 - `timestamp` (moment exacte de la transacció).
 - `lat, longitude` (posició de la compra).
 - `product_id` (enllaç a un producte, però aquí no hi ha FOREIGN KEY definida → podries afegir-la a `product(id_product)`).

- 4) A la pàgina següent es mostra la estructura del nostre model on com ja he comentat la taula `users` queda deslligada de transaccions fins que no crei la taula intermèdia:



5) A continuació carregarem les taules amb la informació del csv

Sovint MySQL té la **opció LOCAL restringida** per motius de seguretat. Això significa que el servidor no permet llegir fitxers locals des del client de manera predeterminada. No és un problema amb la ruta ni amb el CSV en si, sinó amb la configuració del MySQL o del client.

```

68      -- Ara carreguem les taules:
69  •  SHOW GLOBAL VARIABLES LIKE 'local_infile';
  
```

Result Grid

Variable_name	Value
local_infile	OFF

Output

#	Time	Action	Message
1	12:05:13	SHOW GLOBAL VARIABLES LIKE 'local_infile'	1 row(s) returned

Haurem de carregar els fitxers des de la carpeta de Uploads de MySQL. Per tant, busquem la carpeta des d'on haurem de carregar els csv a la nostra Base de Dades (BBDD).

```
71 • SHOW VARIABLES LIKE 'secure_file_priv';
72
```

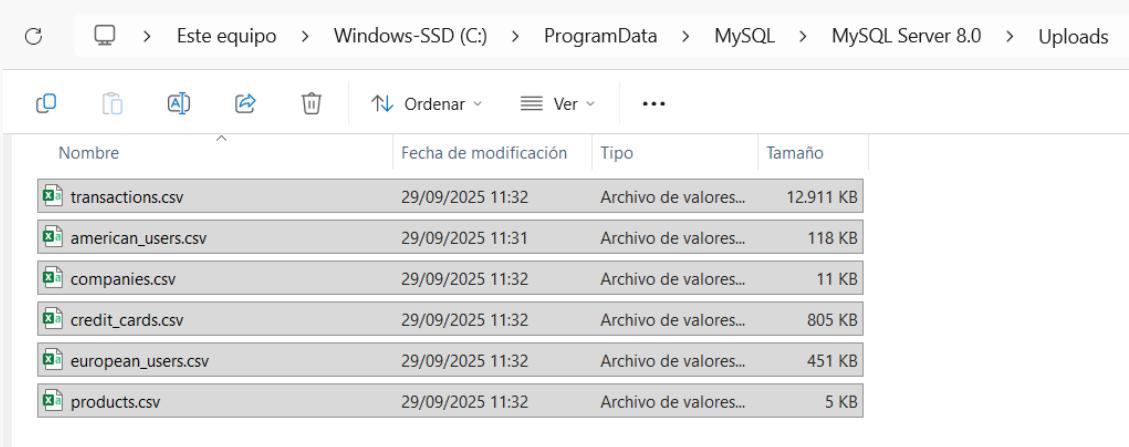
Result Grid | Filter Rows: Export: Wrap Cell Content:

Variable_name	Value
secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\

Output

Action Output	#	Time	Action	Message
1	12:05:13	SHOW GLOBAL VARIABLES LIKE 'local_infile'	1 row(s) returned	
2	12:06:10	SHOW VARIABLES LIKE 'secure_file_priv'	1 row(s) returned	

Carreguem els documents en aquesta carpeta:



I a continuació creem els scripts per carregar les dades dels fitxers:

ATENCIÓ!!! posar doble barra en la mateixa direcció de la ruta o canviar la direcció de la barra per poder carregar els documents!

Nota: comprovar que es carregen els registres i no passa com en el cas de sota que sembla que s'han carregat i no ha carregat cap registre (0 row(s)):

```
72 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\products.csv'
INTO TABLE product
FIELDS TERMINATED BY ','
ENCLOSED BY ""
LINES TERMINATED BY '\\r\\n'
IGNORE 1 ROWS;
```

Output

Action Output	#	Time	Action	Message
1	13:13:34	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\products.csv' INTO T...	0 row(s) affected Records: 0 Deleted: 0 Skipped: 0 Warnings: 0	

Començo per company i segueixo en el mateix ordre en que les havia creat (crèdit_card, product, users i transaccions) , per evitar problemes, ja que he creat les claus foranies durant la creació de taules.

```

76
77 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\companies.csv'
78   INTO TABLE company
79   FIELDS TERMINATED BY ','
80   ENCLOSED BY ""
81   LINES TERMINATED BY '\\r\\n'
82   IGNORE 1 ROWS;
83

```

Output

#	Time	Action	Message
1	12:09:56	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\companies.csv' I...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

Comprovo que s'ha carregat tots els registres (100 empreses):

Result Grid						
Filter Rows: <input type="text"/> Edit: Wrap Cell Content: <input type="checkbox"/>						
	id_company	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.co.uk	Germany	https://cnn.com/user/110
	b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.orion@outlook.ca	New Zealand	https://netflix.com/settin

Output

#	Time	Action	Message
1	12:09:56	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\companies.csv' I...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
2	12:10:43	SELECT * FROM skycaster.company	100 row(s) returned

Això ho repeteixo amb la resta de les taules, tenint en compte que s'han de registrar els camps i en el cas de que modifiquem alguna dada (com les @dates) ho hem de indicar amb el símbol @.

Comprovem que es carreguen 5000 targetes de crèdit i 100 productes

```

83
84 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\credit_cards.csv'
85   INTO TABLE credit_card
86   FIELDS TERMINATED BY ','
87   ENCLOSED BY ""
88   LINES TERMINATED BY '\\n'
89   IGNORE 1 ROWS
90   (@id_card, user_id, iban, pan, pin, cvv, track1, track2, @expiring_date)
91   SET expiring_date = STR_TO_DATE(@expiring_date, '%m/%d/%y'); -- y25 Y2025
92
93 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\products.csv'
94   INTO TABLE product
95   FIELDS TERMINATED BY ','
96   ENCLOSED BY ""
97   LINES TERMINATED BY '\\n'
98   IGNORE 1 ROWS;

```

#	Time	Action	Message
1	12:26:21	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\credit_cards.cs...	5000 row(s) affected Records: 5000 Deleted: 0 Skipped: 0 Warnings: 0
2	12:26:34	SELECT * FROM skycaster.credit_card	5000 row(s) returned
3	12:26:43	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\products.csv' I...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
4	12:26:47	SELECT * FROM skycaster.product	100 row(s) returned

En el cas de la taula d'usuaris, carrego els dos arxius (european & american) i comprovo la càrrega (3990 i 1010 usuaris) i després afegiré una columna pel continent i la modificaré amb el nom 'america' o 'europe'

```

99
100 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\american_users.csv'
101 INTO TABLE users
102 FIELDS TERMINATED BY ','
103 ENCLOSED BY '\"'
104 LINES TERMINATED BY '\\n'
105 IGNORE 1 ROWS
106 (id_user, name, surname, phone, email, @birth_date, country, city, postal_code, address)
107 SET birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y');

108
109 -- SET birth_date = STR_TO_DATE(@birth_date, '%m/%d/%Y'); -- y25 Y2025-- no funciona
110
111 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\european_users.csv'
112 INTO TABLE users
113 FIELDS TERMINATED BY ','
114 ENCLOSED BY '\"'
115 LINES TERMINATED BY '\\n'
116 IGNORE 1 ROWS
117 (id_user, name, surname, phone, email, @birth_date, country, city, postal_code, address)
118 SET birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y');

Output
Action Output
# Time Action
1 13:01:12 LOAD DATA INFILE C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\american_users... 1010 row(s) affected Records: 1010 Deleted: 0 Skipped: 0 Warnings: 0
2 13:01:47 LOAD DATA INFILE C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\european_user... 3990 row(s) affected Records: 3990 Deleted: 0 Skipped: 0 Warnings: 0
3 13:02:05 SELECT * FROM skycaster.users 5000 row(s) returned

```

Comprovo tots els registres (5000 usuaris):

id_user	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.
10	Robert	Mccarthy	(324) 746-6771	fermentum@protonmail.com	1984-04-30	United States	San Jose	95101	P.O. Box 773
100	Melodie	Mclean	1-677-221-7152	risus.varius@google.ca	1989-09-15	United States	San Jose	95101	Ap #644-8492 Sagittis St.
1000	Amkriv	Qbulrxbp	+48-258-9936	amkriv.abulrxbp@example.com	1970-05-17	Germany	Stuttgart	70173	215 Qbulrxbp St
1001	Nfvrlb	Oydawibg	+94-121-2522	nfvrlb.oydawibg@example.com	1994-03-04	Germany	Cologne	50667	121 Oydawibg St
1002	Ijbfmid	Jbddzhvp	+70-120-3668	ijbfmid.jbddzhvp@example.com	2001-09-27	Germany	Munich	80331	412 Jbddzhvp St
1003	Uydig	Sfbdynmz	+58-123-6968	uydig.sfbdynmz@example.com	1981-01-20	Germany	Stuttgart	70173	735 Sfbdynmz St
1004	Yjqrq	Ojizvgqi	+77-944-2340	yjqrq.ojizvgqi@example.com	1954-07-27	Germany	Munich	80331	685 Ojizvgqi St
1005	Mnlqtu	Glofegwjk	+54-801-2627	mnlqtu.glofegwjk@example.com	1962-11-15	Portugal	Funchal	9000-001	8 Glofegwjk St
1006	Rhknrr	Suklmunri	+38-186-8580	rhknrr.suklmunri@example.com	1989-12-07	United States	New York	10001	567 Suklmunri St

Output

Action Output

Time Action

1 13:01:12 LOAD DATA INFILE C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\american_users... 1010 row(s) affected Records: 1010 Deleted: 0 Skipped: 0 Warnings: 0

2 13:01:47 LOAD DATA INFILE C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\european_user... 3990 row(s) affected Records: 3990 Deleted: 0 Skipped: 0 Warnings: 0

3 13:02:05 SELECT * FROM skycaster.users 5000 row(s) returned

4 13:05:02 SELECT * FROM skycaster.users 5000 row(s) returned

Afegeixo una nova columna continent_id i actualitzo les dades en funció del continent segons siguin usuaris europeus o americans. La modificació es fa amb un WHEN en comptes d'un WHERE i ho he de fer modificant els termes de seguretat i tornant a entorn segur amb SET SQL_SAFE_UPDATE.

```

120 • ALTER TABLE users
121   ADD COLUMN continent_id VARCHAR(50);
122
123 • SELECT @@sql_mode;
124  /* per evitar l'errorCode 1175: You are using safe update mode and you tried to update a table whiout a WHERE

```

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

@@sql_mode
ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION

Result 1 x

Output

Action	Output		
#	Time	Action	Message
1	13:01:12	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\american_users.csv' INTO TABLE users	1010 row(s) affected Records: 1010 Deleted: 0 Skipped: 0 Warnings: 0
2	13:01:47	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\european_users.csv' INTO TABLE users	3990 row(s) affected Records: 3990 Deleted: 0 Skipped: 0 Warnings: 0
3	13:02:05	SELECT * FROM skycaster.users	5000 row(s) returned
4	13:05:02	SELECT * FROM skycaster.users	5000 row(s) returned
5	13:05:33	ALTER TABLE users ADD COLUMN continent_id VARCHAR(50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
6	13:05:45	SELECT @@sql_mode	1 row(s) returned

```

126 • SET SQL_SAFE_UPDATES=0;
127
128 • UPDATE users
129  SET continent_id = CASE
130    WHEN country IN ('United States', 'Canada') THEN 'america'
131    ELSE 'europe'
132  END;
133 • SET SQL_SAFE_UPDATES=1;-- tornar a la situació anterior
134
135 -- comprovo que no queda cap registre orfe ambas de crear la clau foranea
136 • SELECT *
137   FROM transactions t
138   LEFT JOIN credit_card c
139   ON t.card_id = c.id_card

```

Output

Action	Output		
#	Time	Action	Message
1	13:01:12	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\american_users.csv' INTO TABLE users	1010 row(s) affected Records: 1010 Deleted: 0 Skipped: 0 Warnings: 0
2	13:01:47	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\european_users.csv' INTO TABLE users	3990 row(s) affected Records: 3990 Deleted: 0 Skipped: 0 Warnings: 0
3	13:02:05	SELECT * FROM skycaster.users	5000 row(s) returned
4	13:05:02	SELECT * FROM skycaster.users	5000 row(s) returned
5	13:05:33	ALTER TABLE users ADD COLUMN continent_id VARCHAR(50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
6	13:05:45	SELECT @@sql_mode	1 row(s) returned
7	13:06:32	SET SQL_SAFE_UPDATES=0	0 row(s) affected
8	13:06:51	UPDATE users SET continent_id = CASE WHEN country IN ('United States', 'Canada') THEN 'america' ELSE 'europe' END	5000 row(s) affected Rows matched: 5000 Changed: 5000 Warnings: 0
9	13:06:57	SET SQL_SAFE_UPDATES=1	0 row(s) affected

Finalment carrego la taula de fets (transactions) i també comprovem que s'han carregat tots els registres (100.000 transaccions).

Nota: la carrega amb un warning, elimino i la carrega de nou i continua el warning però carrega totes les dades.

```

141
142 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\transactions.csv'
143 INTO TABLE transactions
144 FIELDS TERMINATED BY ';'
145 ENCLOSED BY ''
146 LINES TERMINATED BY '\\n'
147 IGNORE 1 ROWS
148 (id_transactions, card_id, business_id, timestamp, amount, declined, product_id, user_id, lat, longitude);
149

```

Output :

#	Time	Action	Message
1	13:10:44	SELECT * FROM skycaster.transactions	100000 row(s) returned
2	13:12:58	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\transactions.csv'	Error Code: 1062. Duplicate entry 'CDDA7E40-544D-47BB-A4ED-671DD8A950D9' for key 'transactions.PRIMARY'
3	13:14:04	DROP TABLE 'skycaster'.transactions'	0 row(s) affected
4	13:14:21	USE Skycaster	0 row(s) affected
5	13:14:27	CREATE TABLE IF NOT EXISTS transactions (id_transactions VARCHAR(255) PRIMARY KEY, card_id VA...	0 row(s) affected
6	13:15:00	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\transactions.cs...	

Comprovació de la càrrega(100.000 transaccions):

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

id_transactions	card_id	business_id	timestamp	amount	declined	product_id	user_id	lat	longitude
00043A49-2949-494B-A5DD-A5BAE38B19DD	CcS-9294	b-2458	2024-08-28 07:16:46	395.43	0	16, 26, 97, 87	4713	46.199929	1.435540
000447FE-B650-4DCF-85DE-C7ED0EE1CAAD	CcS-5019	b-2370	2016-12-21 20:07:18	155.63	0	66, 69, 87	438	41.597206	12.221760
00045D68-ED2E-4F2F-8186-CEE074D875D0	CcS-6699	b-2390	2020-07-14 15:37:45	326.01	0	30, 11, 16, 81	2118	29.757296	-95.379637
000481C3-1C26-4FEB-83A0-4CD0EB004BBD	CcS-6696	b-2230	2017-09-04 19:44:53	161.60	0	72	2115	53.548884	-113.503053
00051AA4-9CBE-4268-B070-C38062A1B3E2	CcS-7606	b-2266	2017-01-05 18:19:25	148.91	0	18	3025	52.208370	5.690806
0008A312-EDF0E-4A4F-BC99-E9C92EC3CA4D	CcU-3358	b-2598	2023-09-23 04:51:43	294.59	0	35, 33, 19	215	53.553486	-113.499134

transactions 1 x

Output :

#	Time	Action	Message
1	13:10:44	SELECT * FROM skycaster.transactions	100000 row(s) returned
2	13:12:58	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\transactions.csv'	Error Code: 1062. Duplicate entry 'CDDA7E40-544D-47BB-A4ED-671DD8A950D9' for key 'transactions.PRIMARY'
3	13:14:04	DROP TABLE 'skycaster'.transactions'	0 row(s) affected
4	13:14:21	USE Skycaster	0 row(s) affected
5	13:14:27	CREATE TABLE IF NOT EXISTS transactions (id_transactions VARCHAR(255) PRIMARY KEY, card_id VA...	0 row(s) affected
6	13:15:00	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Skycaster\\transactions.cs...	
7	13:16:33	SELECT * FROM skycaster.transactions	100000 row(s) returned

Exercici 1

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

```

153 •   SELECT u.id_user,
154   (
155     (
156       SELECT COUNT(t.id_transactions)
157       FROM transactions t
158       WHERE t.user_id = u.id_user
159     ) AS num_transactions
160   FROM users u
161   WHERE (
162     (
163       SELECT COUNT(t.id_transactions)
164       FROM transactions t
165       WHERE t.user_id = u.id_user
166     ) > 80;
167   -- comprobació fàcil

```

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

	id_user	num_transactions
▶	185	110
	289	94
	318	91
	454	81

Result 2 | x

Output

Action Output

#	Time	Action	Message
✓	1 13:18:02	SELECT u.id_user, ((SELECT COUNT(t.id_transactions) FROM transactions t WHERE t.user_id = u.id_user) > 80;	4 row(s) returned

Exercici 2

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

```

176  /* Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.*/
177 •  SELECT c.id_company, cc.iban, ROUND(AVG(t.amount),2)
178  FROM transactions t
179  LEFT JOIN credit_card cc -- probablement no sigui consistent amb altres queries
180  ON t.card_id = cc.id_card
181  LEFT JOIN company c
182  ON t.business_id = c.id_company
183  WHERE c.company_name = 'Donec Ltd'
184  GROUP BY c.id_company, cc.iban;
185

```

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

id_company	iban	ROUND(AVG(t.amount),2)
b-2242	XX911406401125586307586805	356.25
b-2242	SK9446370242474562577506	142.96
b-2242	XX776752917845952975555640	257.37
b-2242	XX413827362289719304908990	139.59
b-2242	XX347787246070769610780308	240.41

Result 8 x

Output

Action Output

#	Time	Action	Message
1	13:02:07	SELECT c.id_company, cc.iban, ROUND(AVG(t.amount),2) FROM tran...	371 row(s) returned

Si afegim la targeta de crèdit, tenim el mateix resultat i assumeixo que cada targeta de crèdit és única i que està associada a un únic IBAN (que també és únic) perquè dona el mateix resultat

```

2 •  SELECT c.id_company, cc.iban, cc.id_card, ROUND(AVG(t.amount),2)
3  FROM transactions t
4  LEFT JOIN credit_card cc
5  ON t.card_id = cc.id_card
6  LEFT JOIN company c
7  ON t.business_id = c.id_company
8  WHERE c.company_name = 'Donec Ltd'
9  GROUP BY c.id_company, cc.iban, cc.id_card;

```

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

id_company	iban	id_card	ROUND(AVG(t.amount),2)
b-2242	XX911406401125586307586805	CcS-5928	356.25
b-2242	SK9446370242474562577506	CcU-4009	142.96
b-2242	XX776752917845952975555640	CcS-6066	257.37
b-2242	XX413827362289719304908990	CcS-8452	139.59
b-2242	XX347787246070769610780308	CcS-7810	240.41
b-2242	XX688768436543090894854602	CcS-6547	188.58
b-2242	MK28368851538688349	CcU-4135	439.39
b-2242	PL76249283566852676343404576	CcU-4541	541.56
b-2242	XX804008331134918341803913	CcS-9477	189.21
b-2242	LB6465553777363327873049938	CcU-4205	155.50
b-2242	XX354512783755821381008445	CcS-6489	255.83

Result 9 x

Output

Action Output

#	Time	Action	Message
1	13:24:26	SELECT c.id_company, cc.iban, cc.id_card, ROUND(AVG(t.amount),2)...	371 row(s) returned

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:

```

185
186 • CREATE TABLE estat_targetes AS
187   SELECT ultimes_3transacciones.card_id,
188     CASE
189       WHEN SUM(CASE WHEN ultimes_3transacciones.declined = 1 THEN 1 ELSE 0 END) = 3 THEN 'Inactiva'
190       ELSE 'Activa'
191     END AS estat_targeta
192   FROM (
193     SELECT
194       t.card_id,
195       t.declined,
196       t.timestamp,
197       ROW_NUMBER() OVER (PARTITION BY t.card_id ORDER BY timestamp DESC) AS rn
198     FROM transactions t
199   ) ultimes_3transacciones
200   WHERE rn <= 3
201   GROUP BY ultimes_3transacciones.card_id;

```

Output:

#	Time	Action	Message
1	12:06:20	CREATE TABLE estat_targetes AS SELECT ultimes_3transacciones.card_id, CASE WHEN SUM(CAS... 5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0	

He creat la taula ultimes 3 transaccions que anomeno ultimes_3transacciones i creo la taula a partir del card_id d'aquesta taula i amb un CASE classifico en 'activa' e 'inactiva' en funció del resultat de la suma de transaccions declinades. Per saber quines son les tres ultimes faig un ROW NUMBER() OVER (PARTITION BY) per cada targeta ordenades per la data de la transacció.

RESUM:

1. ROW_NUMBER() OVER (PARTITION BY t.card_id ORDER BY timestamp DESC): enumera cada transacció de la més recent a la més antiga.
2. WHERE rn <= 3 → pren les 3 últimes transaccions (si posem rn=3 només agafaria la tercera)
3. SUM(CASE WHEN declined = 1 THEN 1 ELSE 0 END) = 3: comprova si les 3 últimes transaccions van ser declinades
4. CASE: marca com 'Inactiva' o 'Activa' segons correspongui.

Comprovo que tinc 5000 registres en la taula estat de les targetes.

```
1 •  SELECT * FROM skycaster.estat_targetes;
```

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar is a 'Result Grid' section. The results of the query 'SELECT * FROM skycaster.estat_targetes;' are displayed in a table with two columns: 'card_id' and 'estat_targeta'. The data shows 10 rows, all of which have 'estat_targeta' set to 'Activa'. Below the result grid is a 'Message' section showing the creation of a temporary table and a SELECT query. The message for the SELECT query indicates 5000 rows affected, 5000 records returned, and 0 warnings.

card_id	estat_targeta
CcS-6285	Activa
CcS-6286	Activa
CcS-6287	Activa
CcS-6288	Activa
CcS-6289	Activa
CcS-6290	Activa
CcS-6291	Activa
CcS-6292	Activa
CcS-6293	Activa

Message

#	Time	Action	Message
1	12:06:20	CREATE TABLE estat_targetes AS SELECT ultimes_3transaccions.card_id, CASE WHEN SUM(CA...	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0
2	12:07:29	SELECT * FROM skycaster.estat_targetes	5000 row(s) returned

i ara comptaré de les 500 quantes estàn actives

Exercici 1

Quantes targetes estan actives?

```

207
208 •   SELECT COUNT(*)
209   FROM skycaster.estat_targetes
210  WHERE estat_targeta='Activa';
211

```

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

COUNT(*)
4995

Output

Action	Output		
#	Time	Action	Message
1	12:06:20	CREATE TABLE estat_targetes AS SELECT ultimes_3transaccions.card_id, CASE WHEN SUM(CA...	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0
2	12:07:29	SELECT * FROM skycaster.estat_targetes	5000 row(s) returned
3	12:09:47	SELECT COUNT(*) FROM skycaster.estat_targetes WHERE estat_targeta='Inactiva'	1 row(s) returned
4	12:10:02	SELECT COUNT(*) FROM skycaster.estat_targetes WHERE estat_targeta='Activa'	1 row(s) returned

N'hi ha 4.995 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:

Es un escenari clàssic de BBDD amb relació many-to-many implícita. La taula transactions té un camp product_ids que conté més d'un id de producte per transacció; per exemple: 00043A49-2949-494B-A5DD-A5BAE3BB19DD te com product_id: "16, 26, 97, 87". Els id_producte, estan separats per comes, i no es repeteixen (no hi ha cap producte repetit dues vegades). El nostre objectiu es unir al model la taula products però hem d'obtenir informació de cada producte. Aquest tipus de situació no permet un JOIN directe, perquè product_ids no és un valor normalitzat, sinó una cadena separada per comes. Haurem de crear una taula temporal de relació normalitzada (transaction_products).

Creem la taula transaction_products amb dos primary keys per garantir que la combinació transacció i producte és única. La taula intermèdia representa una relació **de molts a molts (N:M)**:

- Una **transacció** pot contenir **múltiples productes**
- Un **producte** pot aparèixer en **múltiples transaccions**

Per tant, transaction_products és una **taula d'unió** o *junction table*, i la seva clau primària ha de garantir que una combinació concreta de transacció i producte és única.

NOTA: creem la taula d'unió assegurant que les dades siguin compatibles amb les taules transactions i product. Si es repetís algun id_product hauríem

d'incloure untercer camp amb la quantitat de productes per transacció (en aquest cas no cal).

```

220 • CREATE TABLE transaction_products (
221     id_transactions VARCHAR(255),
222     id_product VARCHAR(100),
223     PRIMARY KEY (id_transactions, id_product),
224     FOREIGN KEY (id_transactions) REFERENCES transactions(id_transactions),
225     FOREIGN KEY (id_product) REFERENCES product(id_product)
226 );

```

Output:

#	Time	Action	Message
1	11:25:37	CREATE TABLE transaction_products (id_transactions VARCHAR(2...)	0 row(s) affected

Introduïm les dades en la taula transaction_products:

```

227
228 • INSERT INTO transaction_products (id_transactions, id_product)
229     SELECT
230         t.id_transactions,
231         j.product_id
232     FROM transactions t
233     JOIN JSON_TABLE(
234         CONCAT('[', t.product_id, ']'),
235         "$[*]" COLUMNS (product_id VARCHAR(100) PATH "$")
236     ) AS j3
237
238

```

Output:

#	Time	Action	Message
1	11:30:07	INSERT INTO transaction_products (id_transactions, id_product) SELE...	253391 row(s) affected Records: 253391 Duplicates: 0 Warnings: 0

NOTAS:

- a) CONCAT('[', t.product_ids, ']') → converteix la cadena en un array JSON.
- b) JSON_TABLE → crea una fila per cada product_id.

Comprovem que les dades s'han carregat correctament en la taula transaction_products (253391 registres):

```
1 •  SELECT * FROM skycaster.transaction_products;
```

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

id_transactions	id_product
001A60EA-DC9C-4E5A-9460-6628B100E7E1	1
0032F0BB-BBE6-4AA5-B5EE-EEAD533C0C48	1
00342381-503D-422D-85AB-F2D4FFAADC7	1
004C0A80-E537-46D8-BE44-343D2176DF15	1
004D1D85-B2CB-4460-9886-31C42CA96E5F	1
0062599C-0A1F-4405-AD55-45E20043B551	1
007B1297-C3ED-4966-802C-68FE106AD25C	1
007EA15D-AA1F-4FB3-9483-7D055E99CEC0	1
009679A1-1A49-47C1-98F8-8F6C0FA59F32	1

ion_products1 x

Output :

Action Output

#	Time	Action	Message
1	11:30:07	INSERT INTO transaction_products (id_transactions, id_product) SEL...	253391 row(s) affected Records: 253391 Duplicates: 0 Warnings: 0
2	11:31:18	SHOW COLUMNS FROM transactions	10 row(s) returned
3	11:32:37	SHOW COLUMNS FROM product	6 row(s) returned
4	11:37:56	SELECT * FROM skycaster.transaction_products	253391 row(s) returned

Exercici 1

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

```
249 •  SELECT tp.id_product, p.product_name, count(tp.id_transactions) AS nombre_productes_venuts
250   FROM transaction_products tp
251   JOIN product p
252   ON tp.id_product = p.id_product
253   GROUP BY tp.id_product, p.product_name
254   ORDER BY nombre_productes_venuts DESC;
255
```

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

id_product	product_name	nombre_productes_venuts
52	riverlands the duel	2654
29	Tully maester Tarly	2635
21	duel Direwolf	2609
16	the duel warden	2608
66	mustafar jinn	2601

Result 10 x

Output :

Action Output

#	Time	Action	Message
1	13:05:37	SELECT tp.id_product, p.product_name, count(tp.id_transactions) AS n...	100 row(s) returned

Si no ens cal conèixer el nom podrem fer servir només la taula transitions_product:

```
257 •   SELECT tp.id_product, count(tp.id_transactions) AS nombre_productes_venuts
258   FROM transaction_products tp
259   GROUP BY tp.id_product
260   ORDER BY nombre_productes_venuts DESC;
```

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

id_product	nombre_productes_venuts
52	2654
29	2635
21	2609
16	2608
66	2601

Result 11 x

Output

Action Output

#	Time	Action	Message
1	13:05:37	SELECT tp.id_product, p.product_name, count(tp.id_transactions) AS n...	100 row(s) returned
2	13:07:41	SELECT tp.id_product, count(tp.id_transactions) AS nombre_productes...	100 row(s) returned