

Descripció

Partint d'alguns arxius CSV dissenyaràs i crearàs la teva base de dades.

Nivell 1

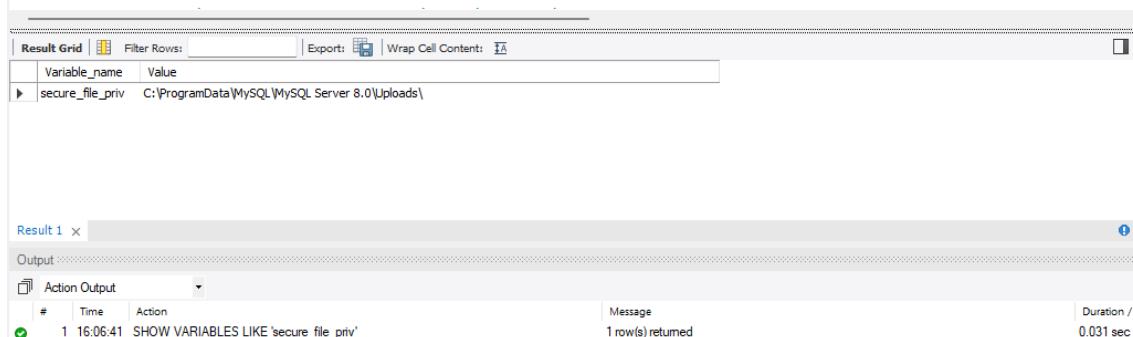
Descàrrega els arxius CSV, estudia'l's i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguis realitzar les següents consultes:

Optem per crear la base de dades, crear totes les taules, carregarles de dades i crear les restriccions per a posteriorment fer els exercicis sol·licitats tots d'una.

Mirem el contingut de les taules

Busquem quina ha de ser la ubicació dels arxius .csv

```
2      -- Mirem on hem de ubicar els arxius .cvs
3
4 •  SHOW VARIABLES LIKE 'secure_file_priv';
5
```



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

Result 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	16:06:41	SHOW VARIABLES LIKE 'secure_file_priv'	1 row(s) returned	0.031 sec /

Donem permisos per a que SQLServer pugui accedir als arxius mitjançant

SET GLOBAL local_infile = 1;

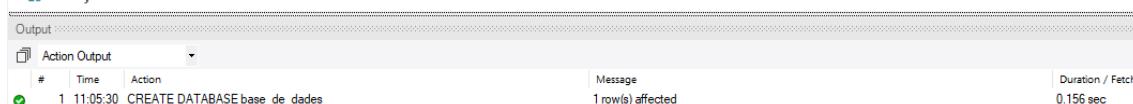
```
9      -- Donem permisos per a que SQLServer pugui accedir als arxius
10 • SET GLOBAL local_infile = 1;
11
```



#	Time	Action	Message	Duration / Fetch
1	10:07:43	SET GLOBAL local_infile = 1	0 row(s) affected	0.000 sec

Creem la base de dades **base_de_dades**

```
11      -- Creem base de dades amb nom base_de_dades
12 • CREATE DATABASE base_de_dades
13 ;
```



#	Time	Action	Message	Duration / Fetch
1	11:05:30	CREATE DATABASE base_de_dades	1 row(s) affected	0.156 sec

Li diem a workbench que utilitzi aquesta base de dades

```
16      -- Marquem que utilitzi aquesta base de dades
17 • USE base_de_dades
18 ;
```



#	Time	Action	Message	Duration / Fetch
1	11:09:34	USE base_de_dades	0 row(s) affected	0.000 sec

Un cop analitzats els arxius .csv decidim que primer crearem i importarem les dades de les taules **dimensionals (users, credit_cards, companies i products)**i seguidament crearem i importarem les

dades de la taula de **fets transactions** on prèviament haurem creat les FOREIGN KEY necessàries per a establir les relacions amb les altres taules.

Com que els 3 arxius de users.csv tenen la mateixa estructura i composició decidim importar els tres arxius **users_usa.csv**, **users_ca.csv** i **users_uk.csv** a una única taula **users** que creem prèviament

Analitzem els 3 arxius users.csv per a veure el tipus de dades i la mida màxima dels camps per a crear la taula adequadament.

TAULA										
users_usa										
column	id	name	surname	phone	email	birth_date	country	city	postal_code	address
no ids	1-150	Zeus	Gamble	1-282-581-05	interdum.enim@protonmail.edu	Nov 17, 1985	United States	Lovell	73544	348-7818 Sagittis St.
users_uk										
column	id	name	surname	phone	email	birth_date	country	city	postal_code	address
no ids	151-200	Meghan	Hayden	0800 745 874	arcu.vel@hotmail.ca	Jul 2, 1980	United Kingdom	Tullibody	A1Y 3TC	Ap #432-4493 Aliquet Rd.
users_ca										
column	id	name	surname	phone	email	birth_date	country	city	postal_code	address
no ids	201-275	Iola	Powers	018-133-4717	ante.blandit@outlook.edu	Mar 20, 2000	Canada	Rigolet	V6T 6M7	154-5415 Auctor St.
uk										
Mides dades	id	name	surname	phone	email	birth_date	country	city	postal_code	address
max	3	9	10	15		35	12	14	18	8
										33
canada										
column	id	name	surname	phone	email	birth_date	country	city	postal_code	address
max	3	8	11	12		35	12	6	24	7
										35
usa										
column	id	name	surname	phone	email	birth_date	country	city	postal_code	address
max	3	9	10	14		40	12	13	14	5
										36
column MAX										
	3	9	11	15		40	12	14	24	8
										36

Amb aquesta informació crearem la taula amb la següent estructura:

Field	Type	Null	Key	Detail
id	int	NO	PRIMARY KEY	Clau primaria
name	varchar(100)	YES		Nom de l'usuari
surname	varchar(100)	YES		Cognom de l'usuari
phone	varchar(50)	YES		Telefon de l'usuari
email	varchar(100)	YES		Correu electrònic de l'usuari
birth_date	date	YES		Data de naixement de l'usuari en fomat MON DD, YYYY
country	varchar(150)	YES		Païs de l'usuari
city	varchar(150)	YES		Ciutat
postal_code	varchar(100)	YES		Codi Postal
address	varchar(255)	YES		Adreça

Creem la taula **users**

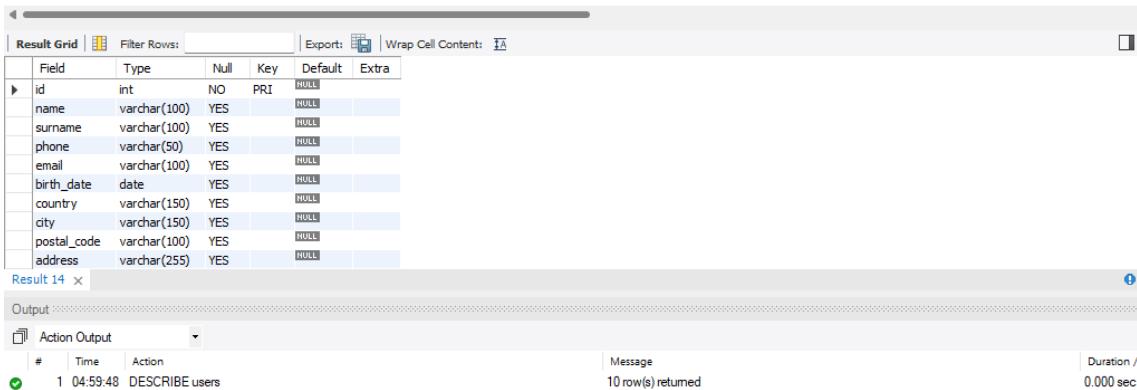
```

28 • CREATE TABLE users (
29     id INT PRIMARY KEY,
30     name VARCHAR(100),
31     surname VARCHAR(100),
32     phone VARCHAR(50),
33     email VARCHAR(100),
34     birth_date DATE,
35     country VARCHAR(150),
36     city VARCHAR(150),
37     postal_code VARCHAR(100),
38     address VARCHAR(255)
39 );
40
Output:
Action Output
# Time Action Message Duration / Fetch
1 04:57:48 CREATE TABLE users ( id INT PRIMARY KEY, name VARCHAR(100), surname... 0 row(s) affected 0.875 sec

```

Fem un **DESCRIBE** de **users** per a veure el resultat.

```
41 -- Visualitzem i verifiquem l'estructura de la taula users
42 • DESCRIBE users;
```



Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
surname	varchar(100)	YES		NULL	
phone	varchar(50)	YES		NULL	
email	varchar(100)	YES		NULL	
birth_date	date	YES		NULL	
country	varchar(150)	YES		NULL	
city	varchar(150)	YES		NULL	
postal_code	varchar(100)	YES		NULL	
address	varchar(255)	YES		NULL	

Result 14 ×

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
1	04:59:48	DESCRIBE users	10 row(s) returned	0.000 sec

Ara carregarem les dades de la taula **users** important els arxius.csv.

Començarem per importar les dades de users_usa.csv amb la següent comanda:

```
LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\users_usa.csv'
INTO TABLE users
```

FIELDS TERMINATED BY ',' → Els valors estan separats per comes

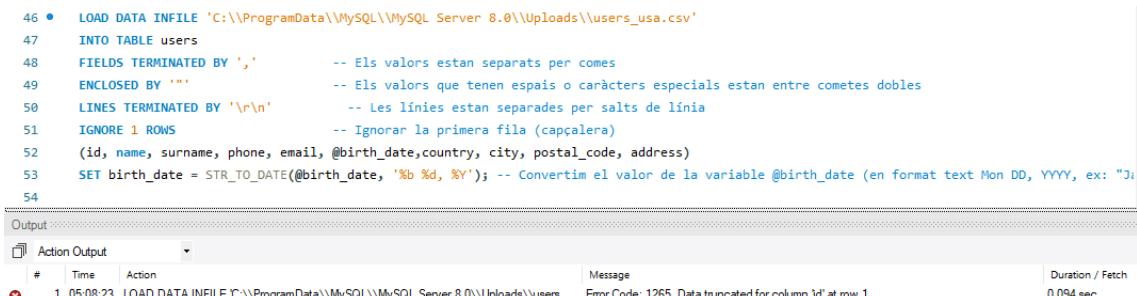
ENCLOSED BY "" → Els valors que tenen espais o caràcters especials estan entre cometes dobles

LINES TERMINATED BY '\r\n' → Les línies estan separades per salts de línia

IGNORE 1 ROWS → Ignorar la primera fila (capçalera)

(id, name, surname, phone, email, @birth_date,country, city, postal_code, address)

SET birth_date = STR_TO_DATE(@birth_date, "%b %d, %Y"); → Convertim el valor de la variable @birth_date (en format text Mon DD, YYYY, ex: "Jan 15, 1990") a un format de data (DATE) utilitzant STR_TO_DATE().



```
46 • LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\users_usa.csv'
47 INTO TABLE users
48 FIELDS TERMINATED BY ','          -- Els valors estan separats per comes
49 ENCLOSED BY ""                  -- Els valors que tenen espais o caràcters especials estan entre cometes dobles
50 LINES TERMINATED BY '\r\n'       -- Les línies estan separades per salts de línia
51 IGNORE 1 ROWS                  -- Ignorar la primera fila (capçalera)
52 (id, name, surname, phone, email, @birth_date,country, city, postal_code, address)
53 SET birth_date = STR_TO_DATE(@birth_date, "%b %d, %Y"); -- Convertim el valor de la variable @birth_date (en format text Mon DD, YYYY, ex: "Jan 15, 1990")
54
```

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
1	05:08:23	LOAD DATA INFILE C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\users_usa.csv	Error Code: 1265. Data truncated for column 'id' at row 1	0.094 sec

Tot i haver verificat prèviament el contingut de l'arxiu users.usa.csv i verificar-ne la composició i estructura i comparar-la amb les altres taules users.csv Workbench no ens permet fer la importació

users_usa.csv
1 id,name,surname,phone,email,birth_date,country,city,postal_code,address
2 1,Zeus,Gamble,1-282-581-0551,interdum.enim@protonmail.edu,"Nov 17, 1985",United States,Lowell,73544,348-7818 Sagittis St.
3 2,Garrett,Mcconnell,(718) 257-2412,integer.vitae.nibh@protonmail.org,"Aug 23, 1992",United States,Des Moines,59464,903 Sit Ave
4 3,Ciaran,Harrison,(522) 598-1365,interdum.feugiat@aol.org,"Apr 29, 1998",United States,Columbus,56518,736-2063 Tellus St.
5 4,Howard,Stafford,1-411-740-3269,ornare.egestas@icloud.edu,"Feb 18, 1989",United States,Kailua,77417,Ap #545-2244 Erat. Rd.
6 5,Hayfa,Pierce,1-554-541-2077,et.malesuada.fames@hotmail.org,"Sep 26, 1998",United States,Sandy,31564,341-2821 Ultrices Av.
7 6,Joel,Tyson,(718) 288-8020,gravidia.nunc.sed@yahoo.ca,"Oct 15, 1989",United States,Nashville,96838,888-2799 Amet Street
8 7,Rafael,Jimenez,(817) 689-0478,eget@outlook.ca,"Dec 4, 1981",United States,Hillsboro,29874,8627 Malesuada Rd.

Provem aleshores de fer-ho amb el Table Data Import Wizard

Import Results

File G:\La meva unitat\IT ACADEMY\Reskilling Data Analytics\SPRINT 4\users_usa.csv was imported in 1.725 s

Table base_de_dades.users has been used

0 records imported

	#	Time	Action	Message
✓	1	05:23:31	SELECT * FROM base_de_dades.users LIMIT 0, 2000	0 row(s) returned
✓	2	05:23:51	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
✓	3	05:23:51	SHOW DATABASES	OK
✓	4	05:23:55	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK
✓	5	05:23:55	SHOW COLUMNS FROM 'base_de_dades'.'users'	OK
✓	6	05:24:01	PREPARE stmt FROM 'INSERT INTO `base_de_dades`.`users` (`id`, `name`, `surname`, `email`, `password`, `role`, `status`, `created_at`, `updated_at`) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)'	OK
✓	7	05:24:02	DEALLOCATE PREPARE stmt	OK

Decidim aleshores crear una comanda **INSERT TO users** això si amb les dades de birth_date normalitzades amb el format DATE i provem de carregar les dades

```

55 -- Insertem les dades de users_usa.csv mitjançant un INSERT TO amb les dades de birth_date previament adaptades a DATE.
56
57 • INSERT INTO users (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES
58   (1, 'Zeus', 'Gamble', '1-282-581-0551', 'interdum.enim@protonmail.edu', '1985-11-17', 'United States', 'Lowell', 73544, '348-7818 Sagittis St.'),
59   (2, 'Garrett', 'Mcconnell', '(718) 257-2412', 'integer.vitae.nibh@protonmail.org', '1992-08-23', 'United States', 'Des Moines', 59464, '903 Sit Ave'),
60   (3, 'Ciaran', 'Harrison', '(522) 598-1365', 'interdum.feugiat@aol.org', '1998-04-29', 'United States', 'Columbus', 56518, '736-2063 Tellus St.'),
61   (4, 'Howard', 'Stafford', '1-411-740-3269', 'ornare.egesta@icloud.edu', '1989-02-10', 'United States', 'Kailua', 77417, 'Ap #545-2244 Erat Rd.'),
62   (5, 'Hayfa', 'Pierce', '1-554-541-2077', 'et.malesuada.fames@hotmail.org', '1998-09-26', 'United States', 'Sandy', 31564, '341-2821 Ultrices Av.'),
63   (6, 'Joel', 'Tyson', '(718) 288-8020', 'gravida.nunc.sed@yahoo.ca', '1989-10-15', 'United States', 'Nashville', 96838, '888-2799 Amet Street'),
64   (7, 'Rafael', 'Jimenez', '(817) 689-0478', 'eget@outlook.ca', '1981-12-04', 'United States', 'Hillsboro', 29874, '8627 Malesuada Rd.'),
65   (8, 'Nissim', 'Franks', '(692) 157-3469', 'egesta.aliquam.fringilla@google.ca', '1993-08-01', 'United States', 'Jackson', 61750, 'Ap #251-7144 Integer St.'),

```

Output:

Action Output	#	Time	Action	Message	Duration / Fetch
1 05:31:07	INSERT INTO users (id, name, surname, phone, email, birth_date, country, city, postal...	150	row(s) affected	Records: 150 Duplicates: 0 Warnings: 0	0.297 sec

Ara sí hem aconseguit carregar a la taula **users** els 150 registres de **users_usa.csv**

Podem comprovar que l'estructura i format de **users.uk.csv** es igual al de **users_usa.csv**

users_uk.csv
1 id,name,surname,phone,email,birth_date,country,city,postal_code,address
2 151,Meghan,Hayden,0800 746 6747,arcu.vel@hotmail.ca,"Jul 2, 1980",United Kingdom,Tullibody,A1Y 3TC,Ap #432-4493 Aliquet Rd.
3 152,Hakeem,Alford,(0111) 367 0184,adipiscing.ligula@google.edu,"Sep 30, 1979",United Kingdom,Kettering,O21 7JV,551-8930 Lobortis Street
4 153,Kegan,Pugh,(016977) 3851,sodales.nisi@aol.org,"Jul 27, 1994",United Kingdom,Whitehaven,HQ8V 7YP,Ap #312-5898 Consectetuer St.
5 154,Cooper,Bullock,(021) 2521 6627,et@outlook.net,"Nov 2, 1986",United Kingdom,Presteigne,U18 ODN,872-1866 Pede Rd.
6 155,Joshua,Russell,055 4409 5286,justo.nec.anne@outlook.edu,"Jan 23, 1984",United Kingdom,Hatfield,B5H 5CS,Ap #285-4727 Auctor, Av.
7 156,Remedios,Case,055 3114 1566,mollis.phasellus.libero@aol.com,"Oct 9, 1994",United Kingdom,North Berwick,QRO 8CW,479-3690 Turpis Road

Seguim carregant dades a **users** amb les dades de **users.uk.csv** amb la següent comanda:

```
LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\users_uk.csv'
```

INTO TABLE **users**

FIELDS TERMINATED BY ',' → Els valors estan separats per comes

ENCLOSED BY "" → Els valors que tenen espais o caràcters especials estan entre cometes dobles

LINES TERMINATED BY '\r\n' → Les línies estan separades per salts de línia. Creat amb Windows

IGNORE 1 ROWS → Ignorar la primera fila (capçalera)

(**id, name, surname, phone, email, @birth_date, country, city, postal_code, address**)

SET **birth_date = STR_TO_DATE(@birth_date, "%b %d, %Y")**; → Convertir el valor de la variable **@birth_date** (en format text Mon DD, YYYY, ex: "Jan 15, 1990") a un format de data (DATE) utilitzant **STR_TO_DATE()**.

```

213 -- Importem les dades de l'arxiu users_uk.csv a la taula users
214
215 • LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\users_uk.csv'
216   INTO TABLE users
217   FIELDS TERMINATED BY ','          -- Els valors estan separats per comes
218   ENCLOSED BY ""
219   LINES TERMINATED BY '\r\n'        -- Les línies estan separades per salts de línia. Creat amb Windows
220   IGNORE 1 ROWS                  -- Ignorar la primera fila (capçalera)
221   (id, name, surname, phone, email, @birth_date, country, city, postal_code, address)
222   SET birth_date = STR_TO_DATE(@birth_date, "%b %d, %Y"); -- Convertir el valor de la variable @birth_date (en format text Mon DD, YYYY, ex: "Jan 15, 1990") a un format de data (DATE)
223

```

Output:

Action Output	#	Time	Action	Message	Duration / Fetch
1 05:59:34	LOAD DATA INFILE C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\users_uk.csv	50	row(s) affected	Records: 50 Deleted: 0 Skipped: 0 Warnings: 0	0.266 sec

Podem veure que s'han carregat correctament els 50 registres de **users_uk.csv**

Veiem també que l'estructura i format de **users_ca.csv** coincideix amb **users_usa.csv** i **users_uk.csv**

users_ca.csv
1 id,name,surname,phone,email,birth_date,country,city,postal_code,address
2 201,Iola,Powers,018-139-4717,ante.blandit@outlook.edu,"Mar 20, 2000",Canada,Rigolet,V6T 6M7,154-5415 Auctor St.
3 202,Maxwell,Holden,045-402-7693,donec@hotmail.edu,"Dec 2, 1986",Canada,Murdochville,S7E 6Z0,Ap #880-6372 Ultrices. St.
4 203,Jarrod,Fields,010-741-8105,sit.amet@google.co.uk,"Jan 6, 1982",Canada,Baddeck,K3X 6Z5,441-8969 Rhoncus Road
5 204,Emerson,Sharp,068-138-9383,ante.iaculis@outlook.ca,"Oct 15, 1994",Canada,Maple Creek,Y2C 9E6,"517-6759 Ut, Av."
6 205,Sonya,Mckee,041-151-9737,magna.phasellus.dolor@google.ca,"May 7, 1983",Canada,Dieppe,E7S 4P8,Ap #916-8051 A St.
7 206,Harper,Hart,030-656-1670,fringilla.donec@outlook.net,"Nov 17, 2000",Canada,Québec City,B4K 0J6,8588 Massa. Ave

Acabem carregant les dades a **users** amb les dades de **users_ca.csv** amb la següent comanda:

```
LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\users_ca.csv'
```

INTO TABLE users

FIELDS TERMINATED BY ',' → Els valors estan separats per comes

ENCLOSED BY "" → Els valors que tenen espais o caràcters especials estan entre cometes dobles

LINES TERMINATED BY '\r\n' → Les línies estan separades per salts de línia. **Creat amb Windows**

IGNORE 1 ROWS → Ignorar la primera fila (capçalera)

(**id, name, surname, phone, email, @birth_date,country, city, postal_code, address**)

SET birth_date = STR_TO_DATE(@birth_date, "%b %d, %Y"); → Convertim el valor de la variable @birth_date (en format text Mon DD, YYYY, ex: "Jan 15, 1990") a un format de data (DATE) utilitzant STR_TO_DATE().

```

225    -- Importem les dades de l'arxiu users_ca.csv a la taula users
226
227 • LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\users_ca.csv'
228   INTO TABLE users
229   FIELDS TERMINATED BY ','          -- Els valors estan separats per comes
230   ENCLOSED BY ""                  -- Els valors que tenen espais o caràcters especials estan entre cometes dobles
231   LINES TERMINATED BY '\r\n'        -- Les línies estan separades per salts de línia. Creat amb Windows
232   IGNORE 1 ROWS                  -- Ignorar la primera fila (capçalera)
233   (id, name, surname, phone, email, @birth_date,country, city, postal_code, address)
234   SET birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y'); -- Convertim el valor de la variable @birth_date (en format text Mon DD, YYYY, ex: "Jan 15, 1990") a un forma
235

```

Output:

#	Time	Action	Message	Duration / Fetch
1	06:05:56	LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\users_ca.csv'	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0	0.203 sec

Podem comprovar que s'han importat correctament els 75 registres de users_ca.csv

Visualitzem les dades de la taula **users** completa amb:

SELECT * FROM users;

```

237    -- Visualitzem les dades de la taula users completa
238 • SELECT * FROM users;

```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:	Result Grid																																																											
<table border="1"> <thead> <tr> <th>id</th> <th>name</th> <th>surname</th> <th>phone</th> <th>email</th> <th>birth_date</th> <th>country</th> <th>city</th> <th>postal_code</th> <th>address</th> </tr> </thead> <tbody> <tr><td>1</td><td>Zeus</td><td>Gamble</td><td>1-282-581-0551</td><td>interdum.enim@protonmail.edu</td><td>1985-11-17</td><td>United States</td><td>Lowell</td><td>73544</td><td>348-7818 Sagittis St.</td></tr> <tr><td>2</td><td>Garrett</td><td>Mcconnell</td><td>(718) 257-2412</td><td>integer.vitae.nibh@protonmail.org</td><td>1992-08-23</td><td>United States</td><td>Des Moines</td><td>59464</td><td>903 Sit Ave</td></tr> <tr><td>3</td><td>Ciaran</td><td>Harrison</td><td>(522) 598-1365</td><td>interdum.feugiat@aol.org</td><td>1998-04-29</td><td>United States</td><td>Columbus</td><td>56518</td><td>736-2063 Tellus St.</td></tr> <tr><td>4</td><td>Howard</td><td>Stafford</td><td>1-411-740-3269</td><td>ornare.egestas@cloud.edu</td><td>1989-02-18</td><td>United States</td><td>Kailua</td><td>77417</td><td>Ap #545-2244 Erat. Rd.</td></tr> <tr><td>5</td><td>Hayfa</td><td>Pierce</td><td>1-554-541-2077</td><td>et.malesuada.fames@hotmail.org</td><td>1998-09-26</td><td>United States</td><td>Sandy</td><td>31564</td><td>341-2821 Ultrices Av.</td></tr> </tbody> </table>	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	Lowell	73544	348-7818 Sagittis St.	2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	1992-08-23	United States	Des Moines	59464	903 Sit Ave	3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	1998-04-29	United States	Columbus	56518	736-2063 Tellus St.	4	Howard	Stafford	1-411-740-3269	ornare.egestas@cloud.edu	1989-02-18	United States	Kailua	77417	Ap #545-2244 Erat. Rd.	5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	1998-09-26	United States	Sandy	31564	341-2821 Ultrices Av.	users 1	Output:	Applv	Revert
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	Lowell	73544	348-7818 Sagittis St.																																																							
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	1992-08-23	United States	Des Moines	59464	903 Sit Ave																																																							
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	1998-04-29	United States	Columbus	56518	736-2063 Tellus St.																																																							
4	Howard	Stafford	1-411-740-3269	ornare.egestas@cloud.edu	1989-02-18	United States	Kailua	77417	Ap #545-2244 Erat. Rd.																																																							
5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	1998-09-26	United States	Sandy	31564	341-2821 Ultrices Av.																																																							

Output:

#	Time	Action	Message	Duration / Fetch
1	06:09:34	SELECT * FROM users LIMIT 0, 2000	275 row(s) returned	0.000 sec / 0.016 sec

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:	Result Grid																																																											
<table border="1"> <thead> <tr> <th>id</th> <th>name</th> <th>surname</th> <th>phone</th> <th>email</th> <th>birth_date</th> <th>country</th> <th>city</th> <th>postal_code</th> <th>address</th> </tr> </thead> <tbody> <tr><td>151</td><td>Meghan</td><td>Hayden</td><td>0800 746 6747</td><td>arcu.vel@hotmail.ca</td><td>1980-07-02</td><td>United Kingdom</td><td>Tullibody</td><td>A1Y 3TC</td><td>Ap #432-4493 Aliquet Rd.</td></tr> <tr><td>152</td><td>Hakeem</td><td>Alford</td><td>(0111) 367 0184</td><td>adipiscing.ligula@google.edu</td><td>1979-09-30</td><td>United Kingdom</td><td>Kettering</td><td>O21 73V</td><td>551-8930 Lobortis Street</td></tr> <tr><td>153</td><td>Keegan</td><td>Pugh</td><td>(016977) 3851</td><td>sodales.nisi@aol.org</td><td>1994-07-27</td><td>United Kingdom</td><td>Whitehaven</td><td>HQ8V 7YP</td><td>Ap #312-5898 Consectetuer St.</td></tr> <tr><td>154</td><td>Cooper</td><td>Bullock</td><td>(021) 2521 6627</td><td>et@outlook.net</td><td>1986-11-02</td><td>United Kingdom</td><td>Presteigne</td><td>U18 0DN</td><td>872-1866 Pede Rd.</td></tr> <tr><td>155</td><td>Joshua</td><td>Russell</td><td>055 4409 5286</td><td>justo.nec.antra@outlook.edu</td><td>1984-01-23</td><td>United Kingdom</td><td>Hatfield</td><td>B5H 5CS</td><td>Ap #285-4727 Auctor. Av.</td></tr> </tbody> </table>	id	name	surname	phone	email	birth_date	country	city	postal_code	address	151	Meghan	Hayden	0800 746 6747	arcu.vel@hotmail.ca	1980-07-02	United Kingdom	Tullibody	A1Y 3TC	Ap #432-4493 Aliquet Rd.	152	Hakeem	Alford	(0111) 367 0184	adipiscing.ligula@google.edu	1979-09-30	United Kingdom	Kettering	O21 73V	551-8930 Lobortis Street	153	Keegan	Pugh	(016977) 3851	sodales.nisi@aol.org	1994-07-27	United Kingdom	Whitehaven	HQ8V 7YP	Ap #312-5898 Consectetuer St.	154	Cooper	Bullock	(021) 2521 6627	et@outlook.net	1986-11-02	United Kingdom	Presteigne	U18 0DN	872-1866 Pede Rd.	155	Joshua	Russell	055 4409 5286	justo.nec.antra@outlook.edu	1984-01-23	United Kingdom	Hatfield	B5H 5CS	Ap #285-4727 Auctor. Av.	users 1	Output:	Applv	Revert
id	name	surname	phone	email	birth_date	country	city	postal_code	address																																																							
151	Meghan	Hayden	0800 746 6747	arcu.vel@hotmail.ca	1980-07-02	United Kingdom	Tullibody	A1Y 3TC	Ap #432-4493 Aliquet Rd.																																																							
152	Hakeem	Alford	(0111) 367 0184	adipiscing.ligula@google.edu	1979-09-30	United Kingdom	Kettering	O21 73V	551-8930 Lobortis Street																																																							
153	Keegan	Pugh	(016977) 3851	sodales.nisi@aol.org	1994-07-27	United Kingdom	Whitehaven	HQ8V 7YP	Ap #312-5898 Consectetuer St.																																																							
154	Cooper	Bullock	(021) 2521 6627	et@outlook.net	1986-11-02	United Kingdom	Presteigne	U18 0DN	872-1866 Pede Rd.																																																							
155	Joshua	Russell	055 4409 5286	justo.nec.antra@outlook.edu	1984-01-23	United Kingdom	Hatfield	B5H 5CS	Ap #285-4727 Auctor. Av.																																																							

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:	Result Grid																																																											
<table border="1"> <thead> <tr> <th>id</th> <th>name</th> <th>surname</th> <th>phone</th> <th>email</th> <th>birth_date</th> <th>country</th> <th>city</th> <th>postal_code</th> <th>address</th> </tr> </thead> <tbody> <tr><td>201</td><td>Iola</td><td>Powers</td><td>018-139-4717</td><td>ante.blandit@outlook.edu</td><td>2000-03-20</td><td>Canada</td><td>Rigolet</td><td>V6T 6M7</td><td>154-5415 Auctor St.</td></tr> <tr><td>202</td><td>Maxwell</td><td>Holden</td><td>045-402-7693</td><td>donec@hotmail.edu</td><td>1986-12-20</td><td>Canada</td><td>Murdochville</td><td>S7E 6E0</td><td>Ap #880-6372 Ultrices. St.</td></tr> <tr><td>203</td><td>Jarrad</td><td>Fields</td><td>010-741-8105</td><td>sit.amet@google.co.uk</td><td>1982-01-06</td><td>Canada</td><td>Baddeck</td><td>K3X 6Z5</td><td>441-8969 Rhoncus Road</td></tr> <tr><td>204</td><td>Emerson</td><td>Sharp</td><td>068-138-9383</td><td>ante.iaculis@outlook.ca</td><td>1994-10-15</td><td>Canada</td><td>Maple Creek</td><td>Y2C 9E6</td><td>517-6759 Ut, Av.</td></tr> <tr><td>205</td><td>Sonya</td><td>McKee</td><td>041-151-9737</td><td>magna.phasellus.dolor@google.ca</td><td>1983-05-07</td><td>Canada</td><td>Dieppe</td><td>E7S 4P8</td><td>Ap #916-8051 A St.</td></tr> </tbody> </table>	id	name	surname	phone	email	birth_date	country	city	postal_code	address	201	Iola	Powers	018-139-4717	ante.blandit@outlook.edu	2000-03-20	Canada	Rigolet	V6T 6M7	154-5415 Auctor St.	202	Maxwell	Holden	045-402-7693	donec@hotmail.edu	1986-12-20	Canada	Murdochville	S7E 6E0	Ap #880-6372 Ultrices. St.	203	Jarrad	Fields	010-741-8105	sit.amet@google.co.uk	1982-01-06	Canada	Baddeck	K3X 6Z5	441-8969 Rhoncus Road	204	Emerson	Sharp	068-138-9383	ante.iaculis@outlook.ca	1994-10-15	Canada	Maple Creek	Y2C 9E6	517-6759 Ut, Av.	205	Sonya	McKee	041-151-9737	magna.phasellus.dolor@google.ca	1983-05-07	Canada	Dieppe	E7S 4P8	Ap #916-8051 A St.	users 1	Output:	Applv	Revert
id	name	surname	phone	email	birth_date	country	city	postal_code	address																																																							
201	Iola	Powers	018-139-4717	ante.blandit@outlook.edu	2000-03-20	Canada	Rigolet	V6T 6M7	154-5415 Auctor St.																																																							
202	Maxwell	Holden	045-402-7693	donec@hotmail.edu	1986-12-20	Canada	Murdochville	S7E 6E0	Ap #880-6372 Ultrices. St.																																																							
203	Jarrad	Fields	010-741-8105	sit.amet@google.co.uk	1982-01-06	Canada	Baddeck	K3X 6Z5	441-8969 Rhoncus Road																																																							
204	Emerson	Sharp	068-138-9383	ante.iaculis@outlook.ca	1994-10-15	Canada	Maple Creek	Y2C 9E6	517-6759 Ut, Av.																																																							
205	Sonya	McKee	041-151-9737	magna.phasellus.dolor@google.ca	1983-05-07	Canada	Dieppe	E7S 4P8	Ap #916-8051 A St.																																																							

Podem veure que està correctament i completament carregada amb els 275 registres.

Seguim visualitzant les dades de credit_cards.csv

credit_cards.csv	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:	Result Grid
<pre> 1 id,user_id,iban,pan,pin,cvv,track1,track2,expiring_date 2 CcU-2938, 275,TR30195031221357681763861,5424465566813633,3257,984,%B8383712448554646%WovaxejDpwiev%8604114277,%B7653863056044187=800716333673,10/30/22 3 CcU-2945, 274,D026854763748537475216568689,514242382194828,9080,887,%B4621311609958661%Ur tuyfSeimxn%061062824177,%B4149568437843501=%10714033071,08/24/23 4 CcU-2952, 273,BG45IVQL52710525608255,4556 453 55 5287,4598,438,%B2183285104307501%CddyytcUxwfdg%590795543079,%B677850257827162=6906859740077,06/29/21 5 CcU-2959, 272,CR724227744338841535,3724613773493975,3583,667,%B7281111956795320%XccddiBckecd%0901625373,%B424615489281853=280522391678,02/24/23 6 CcU-2966, 271,BG72LKTQ15627628377563,448566 886747 7265,4900,130,%B472893232756223%JhlyvasEkmwgs%7202289494377,%B231857115599881=%890821578475,10/29/24 7 CcU-2973, 270,PT87806228135092429456346,544 58654 54343 384,8760,887,%B4761405253275637%HjnnipoBlejrl%710851563271,%B7816169831446746=%1310277279,01/30/25 </pre>	credit_cards 1	Output:	Applv	Revert	

Analitzem els tipus de dades i les mides de les dades de credit_cards.csv

credit_cards	Column	id	user_id	iban	pan	pin	cvv	track1	track2	expiring_date
Exemple		CcU-4842	3	SA2156708581957118818229	3774 636724 832	4655	750	%B22162167337%	%B2517312	11/11/2024
		CcU-4849	2	SE2813123487163628531121	5,22336E+15	9992	779	%B88441544476%	%B262398	03/21/25
MAX			8	3	31	19	4	3	46	32

Crearem la taula credit_cards tenint en compte els següents paràmetres:

Field	Type	Null	Key	Detail
id	varchar(20)	NO	PRIMARY KEY	Id de la credit_card
user_id	int	YES		id de l'usuari
iban	varchar(50)	YES		IBAN (31 caràcters màxims)
pan	varchar(50)	YES		Número de la targeta (19 caràcters)
pin	varchar(4)	YES		PIN (4 díigits)
cvv	int	YES		CVV (3 díigits)
track1	varchar(50)	YES		Dades de la banda magnètica (46 caràcters)
track2	varchar(50)	YES		Dades de la banda magnètica (32 caràcters)
expiring_date	date	YES		Data d'expiració (8 caràcters, format MM/YY/YYYY o MM/YY) Convertirem les dades a DATE

```

252 • CREATE TABLE credit_cards (
253     id VARCHAR(20) PRIMARY KEY,
254     user_id INT,
255     iban VARCHAR(50),
256     pan VARCHAR(50),
257     pin VARCHAR(4),
258     cvv INT,
259     track1 VARCHAR(50),
260     track2 VARCHAR(50),
261     expiring_date DATE
262 );
263
Output
Action Output
# Time Action
1 06:40:00 CREATE TABLE credit_cards ( id VARCHAR(20) PRIMARY KEY, - ID de la targ... 0 row(s) affected
Duration / Fetch
1.156 sec

```

Visualitzem com ha quedat la taula amb un **DESCRIBE**

```

266 • DESCRIBE credit_cards;
---

Result Grid | Filter Rows: Export: Wrap Cell Content: □
| Field | Type | Null | Key | Default | Extra |
|▶ id | varchar(20) | NO | PRI | NULL | |
| user_id | int | YES | | NULL | |
| iban | varchar(50) | YES | | NULL | |
| pan | varchar(50) | YES | | NULL | |
| pin | varchar(4) | YES | | NULL | |
| cvv | int | YES | | NULL | |
| track1 | varchar(50) | YES | | NULL | |
| track2 | varchar(50) | YES | | NULL | |
| expiring_date | date | YES | | NULL | |

Result 2 ×
Output
Action Output
# Time Action
1 06:43:26 DESCRIBE credit_cards
Message
9 row(s) returned
Duration /
0.000 sec.

```

Importem les dades de credit_cards.csv a la taula credit_cards amb la següent comanda

LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\credit_cards.csv'

INTO TABLE credit_cards

FIELDS TERMINATED BY ',' → Els valors estan separats per comes

ENCLOSED BY "" → Els valors que tenen espais o caràcters especials estan entre cometes dobles

LINES TERMINATED BY '\n' → Les línies estan separades per salts de línia. Creat amb LINUX o MAC

IGNORE 1 ROWS → Ignorar la primera fila (capçalera)

(id, user_id, iban, pan, pin, cvv, track1, track2, @expiring_date)

SET `expiring_date` = STR_TO_DATE(@`expiring_date`, '%m/%d/%y'); → converteix el valor de la variable @`expiring_date` (en format text MM/DD/YY) a un format de data (DATE) utilitzant STR_TO_DATE()

```

278 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\credit_cards.csv'
271 INTO TABLE credit_cards
272 FIELDS TERMINATED BY ','          -- Els valors estan separats per comas
273 ENCLOSED BY '\"'                 -- Els valors que tenen espais o caràcters especials estan entre cometes dobles
274 LINES TERMINATED BY '\\n'         -- Les línies estan separades per salts de línia. Creat amb LINUX o MAC
275 IGNORE 1 ROWS                  -- Ignorar la primera fila (capçalera)
276 (id, user_id, iban, pan, pin, cvv, track1, track2, @expiring_date)
277 SET expiring_date = STR_TO_DATE(@expiring_date, '%m/%d/%y'); -- converteix el valor de la variable @expiring_date (en format text MM/DD/YY) a un format de data (DATE)

Output:
Action Output
# Time Action
1 06:51:12 LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\credit_cards.csv' 275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings: 0 Duration / Fetch 0.359 sec

```

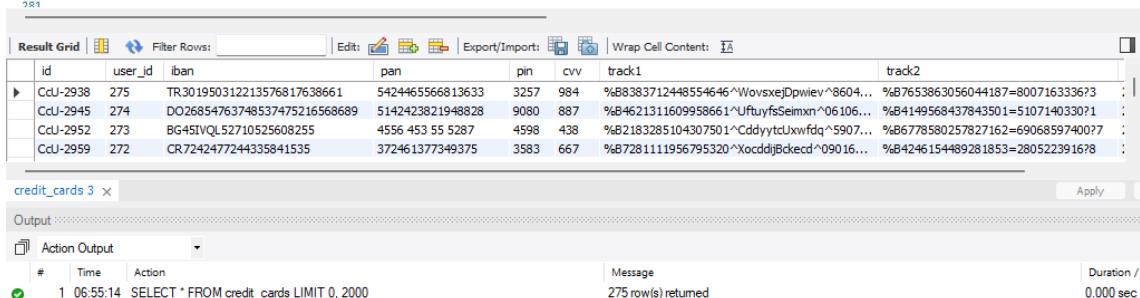
Podem veure que les dades s'han carregat correctament i hi ha 275 registres

Visualitzem la taula credit_cards i el seu contingut

```

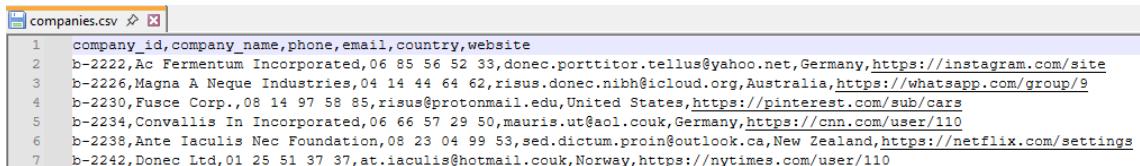
279 -- Visualitzem la importació de credit_cards.csv a taula credit_cards
280 • SELECT * FROM credit_cards;
281

```



	<code>id</code>	<code>user_id</code>	<code>iban</code>	<code>pan</code>	<code>pin</code>	<code>cvv</code>	<code>track1</code>	<code>track2</code>
▶	CcU-2938	275	TR301950312213576817638661	5424465566813633	3257	984	%B88383712448554646^WovsxeIDpwiev^8604...	%B7653863056044187=800716333673
	CcU-2945	274	DO2685476374853747521656689	5142423821948828	9080	887	%B84621311609958661^UftuyfsSeimxn^06106...	%B84149568437843501=510714033071
	CcU-2952	273	BG45IVQL52710525608255	4556 453 55 5287	4598	438	%B2183285104307501^CddytUxwfdq^5907...	%B6778580257827162=6906859740077
	CcU-2959	272	CR724277244335841535	372461377349375	3583	667	%B7281111956795320^XocddijBcked^09016...	%B84246154489281853=280522391678

Seguim visualitzant les dades de companies.csv



	<code>company_id</code>	<code>company_name</code>	<code>phone</code>	<code>email</code>	<code>country</code>	<code>website</code>
1	b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net,Germany	https://instagram.com/site	
2	b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibhici.cloud.org,Australia	https://whatsapp.com/group/9	
3	b-2230	Fuse Corp.	08 14 97 58 85	risus@protonmail.edu,United States	https://pinterest.com/sub/cars	
4	b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.co.uk,Germany	https://cnn.com/user/110	
5	b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca,New Zealand	https://netflix.com/settings	
6	b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.co.uk,Norway	https://nytimes.com/user/110	
7						

Analitzem els tipus de dades i les mides de les dades de companies.csv

<code>companies</code>	<code>company_id</code>	<code>company_name</code>	<code>phone</code>	<code>email</code>	<code>country</code>	<code>website</code>
Column	<code>company_id</code>					
Exemple	b-2578	Dui Quis Institute	06 93 28 72 81	luctus.sit.amet@yahoo	New Zealand	https://yahoo.com/en-us
Num ids	100					
MAX	6		35	14	38	14
						31

Crearem la taula companies tenint en compte els següents paràmetres:

Field	Type	Null	Key	Detall
<code>company_id</code>	varchar(20)	NO	PRIMARY KEY	ID de l'empresa (6 caràcters)
<code>company_name</code>	varchar(50)	YES		Nom de l'empresa (35 caràcters)
<code>phone</code>	varchar(50)	YES		Número de telèfon (14 caràcters)
<code>email</code>	varchar(100)	YES		Correu electrònic (38 caràcters)
<code>country</code>	varchar(100)	YES		País (14 caràcters)
<code>website</code>	varchar(200)	YES		URL de la web (31 caràcters)

```

293 • CREATE TABLE companies (
294     company_id VARCHAR(20) PRIMARY KEY,          -- ID de l'empresa (6 caràcters)
295     company_name VARCHAR(50),                   -- Nom de l'empresa (35 caràcters)
296     phone VARCHAR(50),                         -- Número de telèfon (14 caràcters)
297     email VARCHAR(100),                        -- Correu electrònic (38 caràcters)
298     country VARCHAR(100),                      -- País (14 caràcters)
299     website VARCHAR(200)                       -- URL de la web (31 caràcters)
300 );

```

Output:

Action Output	#	Time	Action	Message	Duration / Fetch
1 07:16:45	CREATE TABLE companies (company_id VARCHAR(20) PRIMARY KEY, - ID de ... 0 row(s) affected				0.875 sec

Visualitzem com ha quedat la taula amb un **DESCRIBE**

```

302 -- Visualitzem i verifiquem l'estrucció de la taula companies
303 • DESCRIBE companies;
304

```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:						
Field	Type	Null	Key	Default	Extra	
company_id	varchar(20)	NO	PRI	NULL		
company_name	varchar(50)	YES		NULL		
phone	varchar(50)	YES		NULL		
email	varchar(100)	YES		NULL		
country	varchar(100)	YES		NULL		
website	varchar(200)	YES		NULL		

Result 4 x

Output:

Action Output	#	Time	Action	Message	Duration /
1 07:19:02	DESCRIBE companies			6 row(s) returned	0.000 sec

Importem les dades de companies.csv a la taula **companies** amb la següent comanda:

```
LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\companies.csv'
```

INTO TABLE companies

FIELDS TERMINATED BY ',' → Els valors estan separats per comes

ENCLOSED BY "" → Els valors que tenen espais o caràcters especials estan entre cometes dobles

LINES TERMINATED BY '\r\n' → Les línies estan separades per salts de línia

IGNORE 1 ROWS → Ignorar la primera fila (capçalera)

(company_id, company_name, phone, email, country, website);

```

307 • LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\companies.csv'
308   INTO TABLE companies
309   FIELDS TERMINATED BY ','          -- Els valors estan separats per comes
310   ENCLOSED BY ""                 -- Els valors que tenen espais o caràcters especials estan entre cometes dobles
311   LINES TERMINATED BY '\r\n'       -- Les línies estan separades per salts de línia
312   IGNORE 1 ROWS                  -- Ignorar la primera fila (capçalera)
313   (company_id, company_name, phone, email, country, website);

```

Output:

Action Output	#	Time	Action	Message	Duration / Fetch
1 07:28:15	LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\compa... 100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0				0.125 sec

Podem veure que les dades s'han carregat correctament i hi ha 100 registres

Visualitzem la taula companies i el seu contingut

```

315 -- Visualitzem la importació de companies.csv a taula companies
316 • SELECT * FROM companies;
317

```

Result Grid Filter Rows: <input type="text"/> 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@cloud.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.pron@outlook.ca	New Zealand	https://netflix.com/settings	

companies 5 x

Output:

Action Output	#	Time	Action	Message	Duration /
1 07:32:11	SELECT * FROM companies LIMIT 0,2000			100 row(s) returned	0.000 sec ,

Seguim visualitzant les dades de products.csv

```
products.csv
1 id,product_name,price,colour,weight,warehouse_id
2 1,Direwolf Stannis,$161.11,#7c7c7c,1,WH-4
3 2,Tarly Stark,$9.24,#919191,2,WH-3
4 3,duel tourney Lannister,$171.13,#d8d8d8,1.5,WH-2
5 4,warden south duel,$71.89,#111111,3,WH-1
6 5,skywalker ewok,$171.22,#dbdbdb,3.2,WH-0
7 6,dooku solo,$136.60,#c4c4c4,0.8,WH--1
8 7,north of Casterly,$63.33,#b7b7b7,0.6,WH--2
9 8,Winterfell,$32.37,#383838,1.4,WH--3
10 9,Winterfell,$76.40,#b5b5b5,1.2,WH--4
```

Analitzem els tipus de dades i les mides de les dades de products.csv

products	column	id	product_name	price	colour	weight	warehouse_id
Exemple		39	the duel	\$151.78	#212121	1.5	WH-34
		4	warden south duel	\$71.89	#111111		3 WH-1
Num ids		100					
MAX		2		29		7	3
							6

Crearem la taula **products** tenint en compte els següents paràmetres:

Field	Type	Null	Key	Details
id	int	NO	PRIMARY KEY	ID del producte (2 díigits màxim)
product_name	varchar(100)	YES		Nom del producte (29 caràcters màxims)
price	decimal(7,2)	NO		Preu en format decimal (màxim 7 caràcters, incloent decimals)
colour	varchar(10)	YES		Codi de color hexadecimal (#XXXXXX → sempre 7 caràcters)
weight	decimal(5,1)	YES		Pes en format decimal (fins a 5 díigits amb 1 decimal)
warehouse_id	varchar(20)	YES		ID del magatzem (6 caràcters màxims)

```
329 • CREATE TABLE products (
330     id INT PRIMARY KEY,          -- ID del producte (2 díigits màxim)
331     product_name VARCHAR(100),    -- Nom del producte (29 caràcters màxims)
332     price DECIMAL(7,2) NOT NULL,  -- Preu en format decimal (màxim 7 caràcters, incloent decimals)
333     colour VARCHAR(10),          -- Codi de color hexadecimal (XXXXXX + sempre 7 caràcters)
334     weight DECIMAL(5,1),          -- Pes en format decimal (fins a 5 digits amb 1 decimal)
335     warehouse_id VARCHAR(20)      -- ID del magatzem (6 caràcters màxims)
336 );
```

Output : Action Output # Time Action Message Duration / Fetch
1 07:55:55 CREATE TABLE products (id INT PRIMARY KEY, -- ID del producte (2 digits... 0 row(s) affected 0.547 sec

Visualitzem com ha quedat la taula amb un **DESCRIBE**

```
338 -- Visualitzem i verifiquem l'estrucció de la taula products
339 • DESCRIBE products;
340
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
product_name	varchar(100)	YES		NULL	
price	decimal(7,2)	NO		NULL	
colour	varchar(10)	YES		NULL	
weight	decimal(5,1)	YES		NULL	
warehouse_id	varchar(20)	YES		NULL	

Result 6 × Output : Action Output # Time Action Message Duration / Fetch
1 07:57:58 DESCRIBE products 6 row(s) returned 0.047 sec /

Importem les dades de products.csv a la taula **products** amb la següent comanda:

```
LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\products.csv'
INTO TABLE products
```

FIELDS TERMINATED BY ',' → Els valors estan separats per comes

ENCLOSED BY "" → Els valors que tenen espais o caràcters especials estan entre cometes dobles

LINES TERMINATED BY '\n' → Les línies estan separades per salts de línia. Creat amb LINUX o MAC

IGNORE 1 ROWS → Ignorar la primera fila (capçalera)

(id, product_name, @price, colour, weight, warehouse_id) → Substituim el símbol \$ de la columna price per un valor buit ("") per eliminar-lo, utilitzant una variable temporal @price.

SET price = REPLACE(@price, '\$', ''); → @price guarda temporalment el valor de la columna price per poder modificar-lo amb REPLACE()

```
343 • LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\products.csv'
344   INTO TABLE products
345   FIELDS TERMINATED BY ','          -- Els valors estan separats per comes
346   ENCLOSED BY '"'                -- Els valors que tenen espais o caràcters especials estan entre cometes dobles
347   LINES TERMINATED BY '\n'        -- Les línies estan separades per salts de línia. Creat amb LINUX o MAC
348   IGNORE 1 ROWS                 -- Ignorar la primera fila (capçalera)
349   (id, product_name, @price, colour, weight, warehouse_id)    -- Substituir el símbol $ de la columna price per un valor buit ('') per eliminar-lo, utilitzant una var:
350   SET price = REPLACE(@price, '$', ''); -- @price guarda temporalment el valor de la columna price per poder modificar-lo amb REPLACE()
```

Output:

#	Time	Action	Message	Duration / Fetch
1	08:03:52	LOAD DATA INFILE C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\produ...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.141 sec

Podem veure que les dades s'han carregat correctament i hi ha 100 registres

Visualitzem la taula **products** i el seu contingut

```
352   -- Visualitzem la importació de products.csv a taula products
353 • SELECT * FROM products;
```

Result Grid:

id	product_name	price	colour	weight	warehouse_id
1	Direwolf Stannis	161.11	#7c7c7c	1.0	WH-4
2	Tarly Stark	9.24	#919191	2.0	WH-3
3	duel tourney Lannister	171.13	#d8d8d8	1.5	WH-2
4	warden south duell	71.89	#111111	3.0	WH-1
5	skywalker ewok	171.22	#dbdbdb	3.2	WH-0
6	dooku solo	136.60	#c4c4c4	0.8	WH--1

products 7

Output:

#	Time	Action	Message	Duration / Fetch
1	08:06:27	SELECT * FROM products LIMIT 0, 2000	100 row(s) returned	0.000 sec

Un cop hem creat totes les taules dimensionals crearem la **taula de fets transactions**

Seguim visualitzant les dades de transactions.csv

transactions.csv

id	card_id	business_id	timestamp	amount	declined	product_ids	user_id	lat	longitude
1	108B1D1D-5B23-A76C-55EF-C568E49A05DD	CcU-2938;b-2222;2021-07-07 17:43:16;293.57;0;59;275;83.7839152128;-178.860353536							
2	7DC26247-20EC-53FE-E555-B6C2E55CA5D5	CcU-2945;b-2226;2022-02-04 15:52:56;312.5;0;71							
3	72997E96-DC2C-A4D7-7C24-66C302F8EA5A	CcU-2952;b-2230;2022-01-30 15:16:36;239.87;0;97							
4	AB069F53-965E-A2A8-CE06-CA8C4FD92501	CcU-2959;b-2234;2021-04-15 13:37:18;60.99;0;11							
5	2F3B6AB6-147D-EB0B-FE8D-9A4E2EA9DBD5	CcU-2966;b-2238;2021-10-18 06:12:03;33.81;0;47							
6	5B0EEF86-B8A1-E5E1-27E7DC8CFS4A4	CcU-2973;b-2242;2022-01-06 01:44:48;42.82;0;23							
7	2B928E1C-EC14-A760-0A75-871477649D6A	CcU-2980;b-2246;2021-08-10 08:14:49;383.73;0;59							
8	063FB79-99EC-66FB-29F7-25726D1764A5	CcU-2987;b-2250;2022-01-06 21:25:27;92.61;0;47							
9									

Analitzem els tipus de dades i les mides de les dades de transactions.csv

transactions	id	card_id	business_id	timestamp	amount	declined	product_ids	user_id	lat	longitude
Column Exemple	3C291BE8-CA7 CcU-3652	b-2618		44376.60486	468.49	0	31	267	-1.8106E+11	1.08415E+11
	1DE327AB-29A CcU-3653	b-2618		44413.6614	440.13	0	73, 79, 97, 13	267	79346738688	9.03733E+11
Num ids	587									
MAX	36	8	6	16	6	1	14	3	13	14

Crearem la taula **transactions** tenint en compte els següents paràmetres:

Field	Type	Null	Key	Details
id	varchar(50)	NO	PRIMARY KEY	ID de la transacció (36 caràcters)
card_id	varchar(20)	YES	FOREIGN KEY	FOREIGN KEY a credit_cards
business_id	varchar(20)	YES	FOREIGN KEY	FOREIGN KEY A companies
timestamp	timestamp	YES		TIMESTAMP
amount	decimal(10,2)	YES		Import de la transacció (fins a 2 decimals, ex: 468.49)
declined	BOOLEAN	YES	TINYINT(1)	Valor que indica si la transacció ha estat rebutjada (0 = FALSE o 1 = TRUE)
product_ids	varchar(200)	YES		Llista d'IDs de productes (fins a 14 caràcters)
user_id	int	YES	FOREIGN KEY	FOREIGN KEY A users
lat	float	YES		Latitud amb alta precisió (fins a 14 caràcters, ex: -1,81059E+11)
longitude	float	YES		Longitud amb alta precisió (fins a 14 caràcters, ex: 1,08415E+11)

```

365      -- Creació taula transactions
366  ● CREATE TABLE transactions (
367      id VARCHAR(50) PRIMARY KEY,          -- UUID de la transacció (36 caràcters)
368      card_id VARCHAR(20),                 -- FOREIGN KEY a credit_cards
369      business_id VARCHAR(20),              -- FOREIGN KEY A companies
370      timestamp TIMESTAMP,                -- TIMESTAMP
371      amount DECIMAL(10, 2),               -- Import de la transacció (fins a 2 decimals, ex: 468.49)
372      declined BOOLEAN,                  -- Valor que indica si la transacció ha estat rebutjada (0 = FALSE o 1 = TRUE)
373      product_ids VARCHAR(200),            -- Llista d'IDs de productes (fins a 14 caràcters)
374      user_id INT,                      -- FOREIGN KEY A users
375      lat FLOAT,                       -- Latitud amb alta precisió (fins a 14 caràcters, ex: -1,81059E+11)
376      longitude FLOAT,                  -- Longitud amb alta precisió (fins a 14 caràcters, ex: 1,08415E+11)
377      FOREIGN KEY (card_id) REFERENCES credit_cards(id),    -- Creació FOREIGN KEY a taula credit_cards
378      FOREIGN KEY (business_id) REFERENCES companies(company_id), -- Creació FOREIGN KEY a taula companies
379      FOREIGN KEY (user_id) REFERENCES users(id)           -- Creació FOREIGN KEY a taula users
380  );
381
382
383
384

```

Output:

#	Time	Action	Message	Duration / Fetch
1	09:16:12	CREATE TABLE transactions (id VARCHAR(50) PRIMARY KEY,-- UUID de la transa...	0 row(s) affected	0.719 sec

Visualitzem com ha quedat la taula amb un **DESCRIBE**

```

382      -- Visualitzem i verifiquem l'estrucció de la taula transactions
383  ● DESCRIBE transactions
384 ;

```

Field	Type	Null	Key	Default	Extra
id	varchar(50)	NO	PRI	NULL	
card_id	varchar(20)	YES	MUL	NULL	
business_id	varchar(20)	YES	MUL	NULL	
timestamp	timestamp	YES		NULL	
amount	decimal(10,2)	YES		NULL	
declined	tinyint(1)	YES		NULL	
product_ids	varchar(200)	YES		NULL	
user_id	int	YES	MUL	NULL	
lat	float	YES		NULL	
longitude	float	YES		NULL	

Result: 9 ×

Output:

#	Time	Action	Message	Duration / F
1	09:23:56	DESCRIBE transactions	10 row(s) returned	0.000 sec /

Importem les dades de products.csv a la taula **transactions** amb la següent comanda:

```
LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\transactions.csv'
```

INTO TABLE transactions

FIELDS TERMINATED BY ';' → Els valors estan separats per PUNT I COMA!!

ENCLOSED BY "" → Els valors que tenen espais o caràcters especials estan entre cometes dobles

LINES TERMINATED BY '\r\n' → Les línies estan separades per salts de línia

IGNORE 1 ROWS → Ignorar la primera fila (capçalera)

(id, card_id, business_id, timestamp, amount, declined, product_ids, user_id, lat, longitude);

```

386  -- Importació de dades de transactions.csv a taula transactions
387
388  ● LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\transactions.csv'
389  INTO TABLE transactions
390  FIELDS TERMINATED BY ';'          -- Els valors estan separats per PUNT I COMA!!
391  ENCLOSED BY '\"'                 -- Els valors que tenen espais o caràcters especials estan entre cometes dobles
392  LINES TERMINATED BY '\\r\\n'       -- Les línies estan separades per salts de línia
393  IGNORE 1 ROWS                  -- Ignorar la primera fila (capçalera)
394  (id, card_id, business_id, timestamp, amount, declined, product_ids, user_id, lat, longitude);

```

Output:

#	Time	Action	Message	Duration / Fetch
1	09:32:11	LOAD DATA INFILE C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\transa...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0	0.734 sec

Podem veure que les dades s'han carregat correctament i hi ha 587 registres

Visualitzem la taula **transactions** i el seu contingut

```

396  -- Visualitzem la importació de transactions.csv a taula transactions
397  ● SELECT * FROM transactions;
398

```

Result Grid:

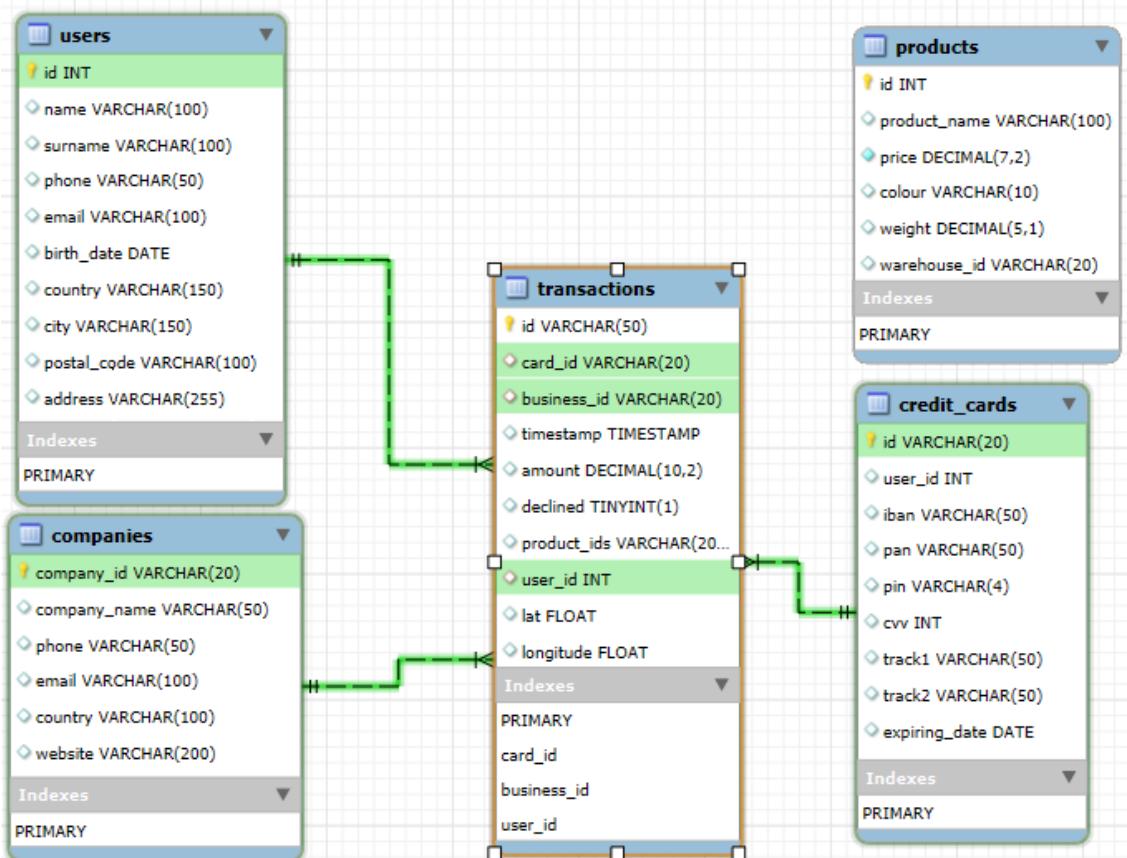
id	card_id	business_id	timestamp	amount	declined	product_ids	user_id	lat	longitude
02C6201E-D90A-1859-B4EE-88D2986D3802	CcU-2938	b-2362	2021-08-28 23:42:24	466.92	0	71, 1, 19	92	81.9185	-12.5276
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	2021-07-26 07:29:18	49.53	0	47, 97, 43	170	-43.9695	-117.525
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-A5E394FEBAA9	CcU-2959	b-2346	2021-10-26 23:00:01	279.93	0	43, 31	92	33.7381	158.298

transactions 10 ×

Output:

#	Time	Action	Message	Duration
1	09:34:24	SELECT * FROM transactions LIMIT 0, 2000	587 row(s) returned	0.000 s

Realitzem el diagrama EER per a veure com ha quedat **base_de_dades**



Podem veure que la base de dades anomenada **base_de_dades** té 5 taules principals:

- **users** (usuaris)
- **companies** (empreses)
- **credit_cards** (targetes de crèdit)
- **transactions** (transaccions)
- **products** (productes)

Taula	Clau Primària (PK)	Claus Foranes (FK)
users	id (INT)	-
companies	company_id (VARCHAR(20))	-
credit_cards	id (VARCHAR(20))	-
transactions	id (VARCHAR(50))	user_id (FK → users(id)), card_id (FK → credit_cards(id)), business_id (FK → companies(company_id))
products	id (INT)	-

Relacions Entre Taules

1. **users** → **credit_cards** (1:N)

Un usuari (users.id) pot tenir diverses targetes (credit_cards.user_id).

2. **users** → **transactions** (1:N)

Un usuari (users.id) pot fer moltes transaccions (transactions.user_id).

3. **credit_cards** → **transactions** (1:N)

Una targeta (credit_cards.id) pot estar associada a moltes transaccions (transactions.card_id).

4. companies → transactions (1:N)

Una empresa (companies.company_id) pot rebre moltes transaccions (transactions.business_id).

5. transactions → products (N:M, via product_ids)

Una transacció pot incloure múltiples productes.

Actualment, transactions.product_ids emmagatzema els productes com una cadena de text (mal disseny).

Hauria d'existir una taula intermitja transactions_products amb transaction_id i product_id com a claus foranes per normalitzar la relació.

- Exercici 1

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

Subconsulta interna (WHERE id IN (...))

Selecciona els user_id de la taula transactions.

Agrupa per user_id.

Filtra només els usuaris que tenen més de 30 transaccions (HAVING COUNT(id) > 30).

Consulta principal (SELECT u.id, u.name, u.surname, (...))

Cerca els usuaris (users.id, name, surname) que compleixen el criteri anterior.

Segona subconsulta ((SELECT COUNT(*) FROM transactions AS t WHERE t.user_id = u.id))

Per cada usuari trobat, compta quantes transaccions ha realitzat

```

409      -- Fem la consulta amb subconsultes
410 •   SELECT u.id, u.name, u.surname,
411       (SELECT COUNT(*) FROM transactions AS t WHERE t.user_id = u.id) AS total_transaccions
412   FROM users AS u
413   WHERE id IN (
414     SELECT user_id
415     FROM transactions
416     GROUP BY user_id
417     HAVING COUNT(id) > 30
418   );
  
```

Result Grid

	id	name	surname	total_transaccions
▶	92	Yehuda	Sharp	39
	267	Ocean	Nelson	52
	272	Hedwig	Gilbert	76
	275	Kenyon	Hartman	48

Result 11

Output

#	Time	Action	Message	Duration / F
1	10:17:45	SELECT u.id, u.name, u.surname, (SELECT COUNT(*) FROM transactions AS t WHERE t.user_id = u.id) AS total_transaccions FROM users AS u WHERE id IN (SELECT user_id FROM transactions GROUP BY user_id HAVING COUNT(id) > 30);	4 row(s) returned	0.000 sec /

Com que l'enunciat de l'exercici demana **explícitament** que es generi una subconsulta hem resolt l'exercici d'aquesta manera.

Tot i que com alternativa ho podríem fer amb JOIN per a més facilitat i optimització de la següent manera:

```
SELECT u.id, u.name, u.surname, COUNT(t.id) AS total_transactions
FROM users AS u
JOIN transactions AS t
ON u.id = t.user_id
GROUP BY u.id, u.name, u.surname
HAVING total_transactions > 30;
```

- Exercici 2

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

```
SELECT c.company_name, c_c.iban, ROUND(AVG(t.amount),2) AS mitjana_import
FROM companies AS C → Seleccionem el nom de l'empresa, l'IBAN de la targeta de crèdit i la mitjana de l'import de les transaccions
JOIN transactions AS t ON c.company_id = t.business_id → Unim la taula "companies" amb "transactions" mitjançant la clau forana "business_id"
JOIN users AS u ON u.id = t.user_id → Unim la taula "users" amb "transactions" mitjançant "user_id" per obtenir dades dels usuaris
JOIN credit_cards AS c_c ON c_c.id = t.card_id → Unim la taula "credit_cards" amb "transactions" mitjançant "card_id" per obtenir informació de les targetes de crèdit
WHERE c.company_name = 'Donec Ltd' → Filtre: només considerem transaccions fetes per l'empresa "Donec Ltd"
GROUP BY c.company_name, c_c.iban; → Agrupem les dades per empresa i IBAN per calcular la mitjana d'import de cada targeta associada a l'empresa
```

```
429 -- Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.
430
431 • SELECT c.company_name, c_c.iban, ROUND(AVG(t.amount),2) AS mitjana_import
432   FROM companies AS C
433   JOIN transactions AS t ON c.company_id = t.business_id
434   JOIN users AS u ON u.id = t.user_id
435   JOIN credit_cards AS c_c ON c_c.id = t.card_id
436   WHERE c.company_name = 'Donec Ltd'
437   GROUP BY c.company_name, c_c.iban;
```

Result Grid		
company_name	iban	mitjana_import
Donec Ltd	PT87806228135092429456346	203.72

Result 13				
Output				
Action Output			Message	Duration /
#	Time	Action		0.000 sec.
1	10:54:46	SELECT c.company_name, c_c.iban, ROUND(AVG(t.amount),2) AS mitjana_import FR...	1 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:

Creem la taula que contindrà les dades amb la següent comanda:

```
CREATE TABLE credit_card_estat (
```

```
    card_id VARCHAR(20) PRIMARY KEY,
```

```
    estat ENUM('activa', 'bloquejada') NOT NULL); → ENUM permet que hi hagi 2 possibilitats (en aquest cas)
    exclusivament anomenades 'activa' i 'bloquejada'
```

```
444      -- Creem la taula credit_card_estat que contindrà les dades
445
446 • CREATE TABLE credit_card_estat (
447     card_id VARCHAR(20) PRIMARY KEY,
448     estat ENUM('activa', 'bloquejada') NOT NULL -- ENUM permet que hi hagi 2 possibilitats exclusivament anomenades 'activa' i 'bloquejada'
449 );
450
```

Output:

#	Time	Action	Message	Duration / Fetch
1	11:43:57	CREATE TABLE credit_card_estat (card_id VARCHAR(20) PRIMARY KEY, estat ENUM('activa', 'bloquejada') NOT NULL);	0 row(s) affected	0.954 sec

Primerament i mitjançant **WINDOWS** farem una consulta que enumeri consecutivament cadascuna de les transaccions per a cada **card_id** de més recent a més antiga i a més mostri si aquesta va ser acceptada o rebutjada. Ho farem de la següent forma:

ROW_NUMBER() enumerarà consecutivament les transaccions (t.id) de cada card_id ordenant-les per timestamp de més recent a més antiga

PARTITION BY t.card_id dividirà les dades per card_id (cada targeta té la seva numeració independent).

```
SELECT t.card_id, t.id, t.timestamp, t.declined,
```

```
    ROW_NUMBER() OVER (PARTITION BY t.card_id ORDER BY t.timestamp DESC) AS numero_transaccio
```

```
FROM transactions AS t;
```

```
469      -- ROW NUMBER() enumerarà consecutivament les transaccions (t.id) de cada card_id ordenant-les per timestamp de més recent a més antiga
470      -- PARTITION BY t.card_id dividirà les dades per card_id (cada targeta té la seva numeració independent).
471 • SELECT t.card_id, t.id, t.timestamp, t.declined,
472     ROW_NUMBER() OVER (PARTITION BY t.card_id ORDER BY t.timestamp DESC) AS numero_transaccio
473     FROM transactions AS t;
```

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

card_id	id	timestamp	declined	numero_transaccio
CCU-2938	DA247F32-9723-9B6B-8770-BC39C252CE15	2021-04-09 19:23:41	0	21
CCU-2938	85068503-9D65-6978-D3C4-8E48A1B9A3C	2021-04-01 07:27:49	0	22
CCU-2938	4DA5B1CE-E88C-23D1-AE6A-87589E4FB2D5	2021-03-28 05:01:44	0	23
CCU-2938	87D22108-C911-AC97-A3BE-B91C1DEEA279	2021-03-23 03:12:06	0	24
CCU-2945	7DC26247-20EC-53FE-E555-B6C2E55CA5D5	2022-02-04 15:52:56	0	1
CCU-2945	FE96CE47-BD59-381C-9E18-E3CA3D44E87F	2021-06-15 00:26:29	1	2
CCU-2952	72997E96-DC2C-A4D7-7C24-66C302F8AE5A	2022-01-30 15:16:36	0	1
CCU-2952	8768FDE2-A231-B916-8644-F70CD13CAF02	2021-05-06 05:33:39	1	2
CCU-2959	69D90229-AD26-43C3-5AAF-8D332D363EB2	2022-03-16 14:01:36	0	1
CCU-2959	95E6F578-3731-28A5-EB21-1A802D016BCB	2022-03-04 02:48:32	0	2
CCU-2959	6D2CA59B-C6E7-3938-CA4D-3603D23647A9	2022-02-28 00:10:50	0	3
CCU-2959	36CCD7CD-F90E-2EE9-58B8-576A04715373	2022-01-16 16:20:55	0	4
CCU-2959	2A37F5B3-1D83-38A4-6E94-B7407D91942C	2022-01-10 20:31:41	0	5
CCU-2959	5B3327C3-046C-EFF1-182A-C0B220923A98	2021-12-28 16:13:14	0	6
CCU-2959	88A74E4E-D317-3F69-86A2-57849C548E71	2021-12-16 17:20:36	0	7

Result 16 ×

Output:

#	Time	Action	Message	Duration / Fetch
1	11:41:30	SELECT t.card_id, t.id, t.timestamp, t.declined, ROW_NUMBER() OVER (PARTITION BY t.card_id ORDER BY t.timestamp DESC) AS numero_transaccio	587 row(s) returned	0.000 sec

Un cop ho tenim i sabem que funciona, busquem aquelles card_id en que **les últimes 3 transaccions** van ser '**declined**' (= 1). Ho fem incorporant una subconsulta (**en negreta**) amb la següent comanda:

Filtra targetes que hagin tingut almenys 3 transaccions rebutjades en tota la seva història

```
SELECT t.card_id, t.id, t.timestamp, t.declined,
```

```
    ROW_NUMBER() OVER (PARTITION BY t.card_id ORDER BY t.timestamp DESC) AS numero_transaccio
```

```

FROM transactions AS t
WHERE t.card_id IN (
    SELECT t.card_id
    FROM transactions AS t
    WHERE declined = 1
    GROUP BY card_id
    HAVING COUNT(*) >= 3
);

465 •   SELECT t.card_id, t.id, t.timestamp, t.declined,
466      ROW_NUMBER() OVER (PARTITION BY t.card_id ORDER BY t.timestamp DESC) AS numero_transaccio
467      FROM transactions AS t
468  WHERE t.card_id IN (
469      SELECT t.card_id
470      FROM transactions AS t
471      WHERE declined = 1
472      GROUP BY card_id
473      HAVING COUNT(*) >= 3
474 );

```

Result Grid				
card_id	id	timestamp	declined	numero_transaccio
1	1	2023-10-05 12:07:00	1	3

Result 18 x				
Output				
Action Output	#	Time	Action	Message
1 12:07:00 SELECT t.card_id,t.id,t.timestamp,t.declined, ROW_NUMBER() OVER (PARTI... 0 row(s) returned	1	12:07:00	SELECT t.card_id,t.id,t.timestamp,t.declined, ROW_NUMBER() OVER (PARTI... 0 row(s) returned	Duration: 0.000 sec

Però el que volem és determinar quines transaccions han estat rebutjades 3 vegades seguides. Per tant el següent pas serà filtrar les últimes 3 transaccions i comptar les rebutjades

Ho fem d'aquesta manera:

```

SELECT ranking.card_id, ranking.id, SUM(ranking.declined) AS suma_declined
FROM
(SELECT t.card_id, t.id, t.timestamp, t.declined,
ROW_NUMBER() OVER (PARTITION BY t.card_id ORDER BY t.timestamp DESC) AS
numero_transaccio → Assigna un número de fila (ROW_NUMBER() OVER (...)) a cada transacció,
ordenant per timestamp DESC dins de cada card_id (targeta)

FROM transactions AS t) AS ranking

WHERE ranking.numero_transaccio <= 3 → Filtrem només les últimes 3 transaccions per targeta
GROUP BY ranking.id Suma les transaccions rebutjades (declined = 1) per cada targeta
HAVING suma_declined = 3; → Retorna les transaccions en què les últimes 3 hagin estat rebutjades

```

```

478 -- Ara filarem només les últimes 3 transaccions de cada targeta i sumem els valors declined. Si la suma és 3, vol dir que les últimes 3
479
480 • SELECT ranking.card_id, ranking.id, SUM(ranking.declined) AS suma_declined
481   FROM
482   (SELECT t.card_id, t.id, t.timestamp, t.declined,
483    ROW_NUMBER() OVER (PARTITION BY t.card_id ORDER BY t.timestamp DESC) AS numero_transaccio
484    FROM transactions AS t) AS ranking
485 -- Ens quedem només amb les últimes 3 transaccions de cada targeta
486 WHERE ranking.numero_transaccio <=3
487 GROUP BY ranking.id
488 -- Si la suma de 'declined' és 3, vol dir que les últimes 3 transaccions han estat rebutjades
489 HAVING suma_declined = 3
490 ;

```

Result Grid		
card_id	id	suma_declined

Result 41					
Output					
Action Output			Message	Duration / F	
1	16:49:51	SELECT ranking.card_id,ranking.id, SUM(ranking.declined) AS suma_declined FROM ... 0 row(s) returned		0.016 sec	/

Seguidament volem determinar l'estat de cada targeta (activa o bloquejada)

```

WITH ultima_transaccio AS (
  SELECT ranking.card_id,
         SUM(ranking.declined) AS suma_declined
    FROM (
      SELECT card_id, declined,
             ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS numero_transaccio
        FROM transactions t
    ) AS ranking
   WHERE ranking.numero_transaccio <= 3
   GROUP BY ranking.card_id
)
SELECT
  COUNT(card_id) AS recompte_targetes,
  CASE
    WHEN suma_declined = 3 THEN 'bloquejada'
    ELSE 'activa'
  END AS estat
FROM ultima_transaccio
GROUP BY estat;

```

WITH ultima_transaccio AS (...) (Subconsulta temporal)

Filtra les últimes 3 transaccions per targeta (WHERE ranking.numero_transaccio <= 3).

Suma el nombre de transaccions rebutjades (SUM(declined)) per targeta.

Consulta principal:

Classifica cada targeta com a 'bloquejada' si ha tingut 3 transaccions rebutjades consecutives (suma_declined = 3), i com a 'activa' en cas contrari.

Compta quantes targetes hi ha en cada estat (COUNT(card_id)).

Agrupa per estat (GROUP BY estat).

Millora respecte a la consulta anterior : No retorna cada transacció, sinó un **resum total de targetes** per estat.

Ara si podem veure el resultat de la consulta

```

491 WITH ultima_transaccio AS (
492     SELECT ranking.card_id,
493             SUM(ranking.declined) AS suma_declined
494     FROM (
495         SELECT card_id, declined,
496                 ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS numero_transaccio
497         FROM transactions t
498     ) AS ranking
499     WHERE ranking.numero_transaccio <= 3
500     GROUP BY ranking.card_id
501 )
502
503     SELECT
504             COUNT(card_id) AS recompte_targetes,
505             CASE
506                 WHEN suma_declined = 3 THEN 'bloquejada'
507                 ELSE 'activa'
508             END AS estat
509     FROM ultima_transaccio
510     GROUP BY estat;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	
recompte_targetes	estat				
275	activa				

Result 25					
Output:					
Action Output	#	Time	Action	Message	Duration /
1 12:46:14 WITH ultima_transaccio AS (SELECT ranking.card_id,		SUM(ranking.decline...)	1 row(s) returned	0.032 sec.

Resum de les consultes:

1. Numerar les transaccions per cada targeta en ordre descendent.
2. Seleccionar targetes que han tingut almenys 3 transaccions rebutjades en total.
3. Filtrar només les últimes 3 transaccions per targeta i veure si totes han estat rebutjades.
4. Classificar targetes en 'bloquejada' o 'activa' segons si totes les últimes 3 transaccions han estat rebutjades.

Per últim fem un INSERT TO per incorporar les dades a la taula

```

514 • INSERT INTO credit_card_estat (card_id, estat)
515   WITH ultima_transaccio AS (
516     SELECT ranking.card_id,
517            SUM(ranking.declined) AS suma_declined
518    FROM (
519      SELECT card_id, declined,
520            ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS numero_transaccio
521       FROM transactions AS t
522    ) AS ranking
523   WHERE ranking.numero_transaccio <= 3
524   GROUP BY ranking.card_id
525  )
526   SELECT ranking.card_id,
527     CASE
528       WHEN suma_declined = 3 THEN 'bloquejada'
529       ELSE 'activa'
530     END AS estat
531   FROM ultima_transaccio AS ranking
532   GROUP BY ranking.card_id, estat;
  
```

Output:

#	Time	Action	Message	Duration / Fetc
1	16:53:49	INSERT INTO credit_card_estat (card_id, estat) WITH ultima_transaccio AS (SELEC...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0	0.187 sec

Exercici 1

Quantes targetes estan actives?

Fem la consulta a la taula credit_card_estat

```

543   -- Quantes targetes estan actives?
544 • SELECT COUNT(*) AS targetes_actives
545   FROM credit_card_estat
546   WHERE estat = 'activa';
  
```

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

targetes_actives
275

Result 43 x

Output:

#	Time	Action	Message	Duration / Fetc
1	16:55:56	SELECT COUNT(*) AS targetes_actives FROM credit_card_estat WHERE estat = 'acti... 1 row(s) returned		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.

En un sistema de vendes, una transacció (transaction_id) pot incloure diversos productes (product_id), i un producte pot aparèixer en múltiples transaccions. Com que no podem representar aquesta relació directament amb claus foranes en una única taula, necessitem una taula intermitja.

La clau primària (PRIMARY KEY(transaction_id, product_id)) és composta i assegura que cada parell transaction_id-product_id sigui únic.

Les claus foranes (FOREIGN KEY (transaction_id) REFERENCES transactions(id) i FOREIGN KEY (product_id) REFERENCES products(id)) garanteixen la integritat referencial, assegurant que només es poden registrar transaccions i productes existents.

CREATE TABLE transactions_products (→ Es crea una nova taula anomenada transactions_products.

transaction_id VARCHAR(50), → Es defineix la columna transaction_id com una cadena de fins a 50 caràcters, representant l'identificador d'una transacció.

product_id INT, → Es defineix la columna product_id com un enter, representant l'identificador d'un producte.

PRIMARY KEY(transaction_id, product_id), → Es defineix una clau primària composta entre transaction_id i product_id, assegurant que no hi hagi duplicats en la combinació de transaccions i productes.

FOREIGN KEY (transaction_id) REFERENCES transactions(id), → Es crea una clau forana que vincula transaction_id amb la columna id de la taula transactions, assegurant que només es puguin inserir valors existents en transactions.

FOREIGN KEY (product_id) REFERENCES products(id) → Es crea una clau forana que vincula product_id amb la columna id de la taula products, assegurant que només es puguin inserir valors existents en products

Creem la taula transactions_products

```

552      -- Creació taula intermitja transactions_products
553 • CREATE TABLE transactions_products (
554   transaction_id VARCHAR(50),
555   product_id INT,
556   PRIMARY KEY(transaction_id, product_id),
557   FOREIGN KEY (transaction_id) REFERENCES transactions(id),
558   FOREIGN KEY (product_id) REFERENCES products(id),
559 );
560
Output:
Action Output
# Time Action
Message
Duration / Fetch
0.953 sec
● 1 16:58:36 CREATE TABLE transactions_products (transaction_id VARCHAR(50), product_id INT... 0 row(s) affected

```

Visualitzem la estructura de la taula

```

561      -- Visualitzem la estructura i composició de la taula transactions_products
562 • DESCRIBE transactions_products;
563
Result Grid | Filter Rows: Export: Wrap Cell Content: □
Field Type Null Key Default Extra
transaction_id varchar(50) NO PRI NULL
product_id int NO PRI NULL

```

Result 44 x

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:04:41	DESCRIBE transactions_products	2 row(s) returned	0.000 sec /

Genera la següent consulta:

Exercici 1

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

L'objectiu de l'exercici és determinar quantes vegades s'ha venut cada producte. Per això, hem de descompondre les dades i després comptar les aparicions de cada producte.

Pas 1: Desglossar la llista de product_ids i normalitzar les dades

Pas 2: Comptar les vegades que cada producte ha estat venut

Pas 1. Inserir dades desglossades a transactions_products

```

567 •      INSERT INTO transactions_products (transaction_id, product_id)
568   SELECT
569     t.id AS transaction_id,
570     CAST(SUBSTRING_INDEX(SUBSTRING_INDEX(t.product_ids, ',', n.n), ',', -1) AS UNSIGNED) AS product_id
571   FROM
572     transactions t
573   JOIN
574     (SELECT 1 AS n UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5) n
575   WHERE
576     CHAR_LENGTH(t.product_ids) - CHAR_LENGTH(REPLACE(t.product_ids, ',', '')) >= n.n - 1;
577   -- Aquest codi desglossa la llista de product_ids per separar els productes que hi ha a cada transacció. L'ús de SUBSTRING_INDEX permet sepe
578

```

Output:

#	Time	Action	Message	Duration / Fetch
1	19:01:51	INSERT INTO transactions_products (transaction_id, product_id) SELECT t.id AS tra...	1457 row(s) affected Records: 0 Warnings: 0	0.937 sec

INSERT INTO transactions_products (transaction_id, product_id) → Insereix dades a la taula **transactions_products**, que serveix com a taula intermitja entre transactions i products

SELECT

t.id AS transaction_id, → Selecciona l'id de cada transacció per mantenir la relació amb els productes venuts

CAST(SUBSTRING_INDEX(SUBSTRING_INDEX(t.product_ids, ',', n.n), ',', -1) AS UNSIGNED) AS product_id

SUBSTRING_INDEX(t.product_ids, ',', n.n) → Retorna els primers n.n elements de la llista separada per comes. Exemple: Per **t.product_ids = "5,12,8"**, si n.n = 2, el resultat és "5,12".

SUBSTRING_INDEX(..., ',', -1) → Agafa només l'últim element de la cadena resultant. Exemple: "5,12" → 12.

CAST(... AS UNSIGNED) → Converteix l'ID del producte a un número per evitar errors.

FROM

transactions t → Agafem les transaccions de la taula **transactions**

JOIN

(SELECT 1 AS n UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5) n

→ Simula una seqüència de valors (1,2,3,4,5) per iterar sobre les posicions dins de **product_ids** i extreure cada element.

WHERE

CHAR_LENGTH(t.product_ids) - CHAR_LENGTH(REPLACE(t.product_ids, ',', '')) >= n.n - 1;

CHAR_LENGTH(t.product_ids) - CHAR_LENGTH(REPLACE(t.product_ids, ',', ''))

→ Compta el nombre de comes a la cadena (product_ids) per saber quants productes hi ha.

>= n.n - 1

→ Filtra perquè només es considerin les posicions vàlides.

Pas 2: Comptar les vegades que cada producte ha estat venut

```

579 -- Pas 2: Comptar les vegades que cada producte ha estat venut.
580 -- Ara podem comptar les vegades que s'ha venut cada producte
581 • SELECT tp.product_id, COUNT(tp.product_id) AS total_productes_venuts, p.price
582   FROM transactions_products AS tp
583   JOIN products AS p
584   ON p.id = tp.product_id
585   GROUP BY 1;

```

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

product_id	total_productes_venuts	price
1	61	161.11
2	65	9.24
3	51	171.13
5	49	171.22
7	54	63.33

Result 49 ×

Output:

Action Output	#	Time	Action	Message	Duration /
1 19:04:50	SELECT tp.product_id, COUNT(tp.product_id) AS total_productes_venuts, p.price FR...	26 row(s) returned	0.000 sec ,		

SELECT tp.product_id, COUNT(tp.product_id) AS total_productes_venuts, p.price

→ **tp.product_id**: Selecciona l'ID del producte.

COUNT(tp.product_id) AS total_productes_venuts

→ Compta les vegades que apareix un producte a **transactions_products**, és a dir, el nombre de vendes.

p.price → Mostra el preu de cada producte des de la taula products

FROM transactions_products AS tp

→ Treballem sobre la taula intermitja **transactions_products**

JOIN products AS p

ON p.id = tp.product_id

→ Fem un **JOIN** amb la taula **products** per afegir el preu de cada producte.

GROUP BY 1;

→ Agrupa per tp.product_id per comptar les vendes de cada producte

Reptes, dificultats afrontades i aspectes a millorar

- Localitzar la ruta on MySQL permet llegir/escriure arxius mitjançant SHOW VARIABLES LIKE 'secure_file_priv'; i donar permisos per a que SQLServer pugui accedir als arxius mitjançant

SET GLOBAL local_infile = 1;

- Ús de **LOAD DATA INFILE** per a importar arxius .csv

- A les comandes per a importar .csv ús de **FIELDS TERMINATED BY ','** quan els valors estan separats per comes o **FIELDS TERMINATED BY ';'** quan els valors estan separats per punt i coma.

ENCLOSED BY ' ' ' quan els valors tenen espais o caràcters especials estan entre cometes dobles i

LINES TERMINATED BY '\r\n' per indicar que les línies estan separades per salts de línia amb arxiu creat amb Windows.

LINES TERMINATED BY '\n' per indicar que les línies estan separades per salts de línia però l'arxiu ha estat creat amb LINUX o MAC

- Ús de @ per a importar dades d'un CSV i necessitem convertir el tipus de dada, assignant els valors a una variable i aplicant una conversió i també

SET xxx = STR_TO_DATE(@); → Convertim el valor de la variable @xxx (en format text a un format DATE)

- Ús de **SET preu = REPLACE(@preu, '\$·/€/£', '')**; → @preu guarda temporalment el valor de la columna price per poder modificar-lo amb REPLACE() i així eliminar la moneda i poder fer càlculs numèrics amb el camp.

- Ús de les funcions de **finestra (window functions)** que permeten realitzar càlculs sobre un conjunt de files sense agregar-les.

Conceptes clau:

- **ROW_NUMBER()**: Assigna un número únic a cada fila dins d'una partició.
- **OVER**: Defineix sobre quin conjunt de files actua la funció de finestra.
- **PARTITION BY**: Divideix les dades en grups perquè la funció s'apliqui a cada grup per separat.

- Ús de les taula pont o intermèdia que s'utilitzen en bases de dades relacionals per gestionar relacions molts a molts (M:N) entre dues taules principals. Aquesta taula conté, com a mínim, les claus primàries de les dues taules que es volen relacionar, actuant com a claus foranes.

Gràcies a aquest enfocament, es pot mantenir la normalització de la base de dades, evitant la redundància i facilitant consultes eficients. Això permet gestionar millor les relacions complexes, com l'assignació de múltiples entitats a diferents grups o categories.

-Ús de **WITH** que s'utilitza per definir **Expressions de Taula Comuna (CTE - Common Table Expressions)**, que són subconsultes temporals que milloren la llegibilitat i eficiència de les consultes complexes.

Després de **WITH**, es defineix una subconsulta que es pot reutilitzar dins de la consulta principal.