

# 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. Creé una base de datos llamada sales

```
2 • CREATE DATABASE IF NOT EXISTS sales;  
3 • USE sales;  
4
```

✓	19	10:29:43	CREATE DATABASE IF NOT EXISTS sales	1 row(s) affected	0.063 sec
✓	20	10:29:43	USE sales	0 row(s) affected	0.016 sec

2. Revisé la estructura del file companies.csv. El separador es una coma. Los valores únicos que identifican la tabla están en la columna company\_id

```
|company_id,company_name,phone,email,country,website  
b-2222,Ac Fermentum Incorporated,06 85 56 52 33,donec.porttitor.tellus@yahoo.net,Germany,https://instagram.com/site  
b-2226,Magna A Neque Industries,04 14 44 64 62,risus.donec.nibh@icloud.org,Australia,https://whatsapp.com/group/9  
b-2230,Fusce Corp.,08 14 97 58 85,risus@protonmail.edu,United States,https://pinterest.com/sub/cars  
b-2234,Convallis In Incorporated,06 66 57 29 50,mauris.ut@aol.couk,Germany,https://cnn.com/user/110
```

3. Creo la tabla company con la PK company\_id. Otras columnas tienen el formato VARCHAR. He revisado la longitud de los valores en las columnas en Excel para determinar el tamaño adecuado de VARCHAR.

```
4  
5 -- Create table company  
6 • CREATE TABLE IF NOT EXISTS company (  
7     company_id VARCHAR(15) PRIMARY KEY,  
8     company_name VARCHAR(255),  
9     phone VARCHAR(15),  
10    email VARCHAR(100),  
11    country VARCHAR(100),  
12    website VARCHAR(255)  
13 );
```

## Output

Action Output

#	Time	Action	Message
✓ 1	10:37:01	CREATE TABLE IF NOT EXISTS company ( company_id VARCHAR(15) PRIMARY KEY, company...	0 row(s) affected

4. Inserto datos en la tabla company. En la captura de pantalla hay un error: debe ser FIELDS TERMINATED BY ',' en lugar de FIELDS TERMINATED BY ';;'

```
--
20 -- insert data to table company from csv file
21 • LOAD DATA LOCAL INFILE 'C:\Users\natab\OneDrive\Рабочий стол\Files database\companies.csv'
22 INTO TABLE company
23 FIELDS TERMINATED BY ';;'
24 LINES TERMINATED BY '\n'
25 IGNORE 1 ROWS;
```

Output

#	Time	Action	Message	Duration / Fetch
1	11:01:02	LOAD DATA LOCAL INFILE 'C:\Users\natab\OneDrive\Рабочий стол\Files database\companies.csv' INTO ...	Error Code: 3948. Loading local data is disabled; this must be enabled on both the client and server sides	0.000 sec

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

5. Check the secure file privileged directory

```
28 • SHOW VARIABLES LIKE 'secure_file_priv';
29
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Variable_name	Value			
secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\			

6. Move the csv file to the secure file privileged directory and insert data to the table company again. There is still the same error.

7. Check if local\_infile is enabled (on)

```
39 • SHOW VARIABLES LIKE 'local_infile';
40
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Variable_name	Value			
local_infile	OFF			

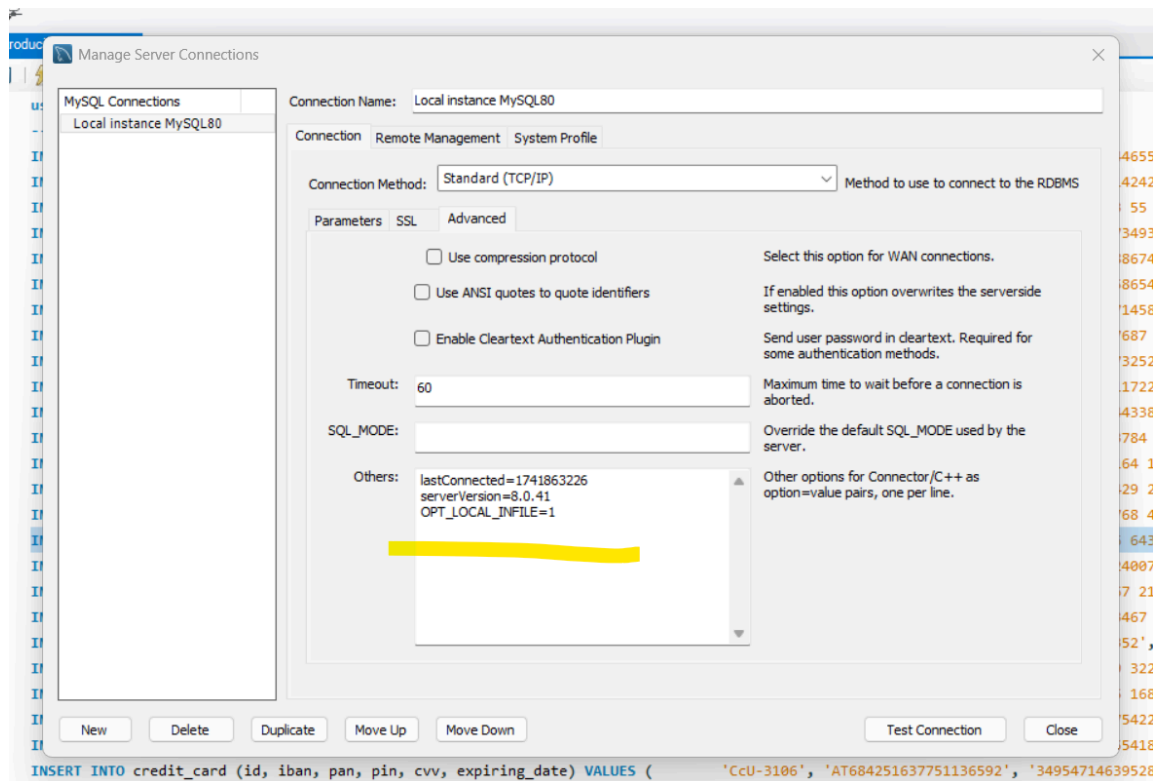
8. Enable local\_infile

```
43 • SET GLOBAL local_infile = 1;
```

Output

#	Time	Action	Message
1	11:08:12	SET GLOBAL local_infile = 1	0 row(s) affected

9. Use the solution from [the stackoverflow](#).  
Edit the connection, on the Connection tab, go to the 'Advanced' sub-tab, and in the 'Others:' box add the line 'OPT\_LOCAL\_INFILE=1'.



10. Inserto nuevamente los datos en la tabla company y verifica que se hayan agregado correctamente.

```
18 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\companies.csv'
19 INTO TABLE company
20 FIELDS TERMINATED BY ','
21 LINES TERMINATED BY '\\n'
22 IGNORE 1 ROWS;
23
```

43  
44 • SELECT \*  
45 FROM company;  
46

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

	company_id	company_name	phone	email	country	website
▶	b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net	Germany	https://instagram.com/site
	b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia	https://whatsapp.com/group/9
	b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/sub/cars
	b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany	https://cnn.com/user/110
	b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings
	b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.couk	Norway	https://nytimes.com/user/110
	b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
	b-2250	Amet Nulla Donec Corporation	07 15 25 14 74	mattis.integer.eu@protonmail.net	Italy	https://netflix.com/sub/cars
	b-2254	Nascetur Ridiculus Mus Inc.	06 26 87 61 84	suspendisse.dui@icloud.net	United States	https://ebay.com/sub
	b-2258	Vestibulum Lorem PC	02 02 87 33 40	aenean.massa.integer@aol.net	Belgium	https://pinterest.com/sub/cars

company 7 x

Output

Action Output

#	Time	Action	Message
✓ 1	13:07:04	SELECT * FROM company LIMIT 0, 5000	100 row(s) returned

11. Revisé la estructura de los archivos users\_ca.csv, users\_uk.csv y users\_usa.csv.

El separador de valores es una coma.

La columna birth\_date contiene fechas con coma (ej. "Mar 20, 2000" ), por eso están entre comillas dobles para mantenerlas como un solo valor.

La columna id contiene los identificadores únicos de los usuarios.

Como tres csv files contienen información sobre la misma dimensión (*user*), los combinaré en una única tabla *user*

```
id,name,surname,phone,email,birth_date,country,city,postal_code,address
201,Iola,Powers,018-139-4717,ante.blandit@outlook.edu,"Mar 20, 2000",Canada,Rigolet,V6T 6M7,154-5415 Auctor St.
202,Maxwell,Holden,045-402-7693,donec@hotmail.edu,"Dec 2, 1986",Canada,Murdochville,S7E 6E0,Ap #880-6372 Ultrices. St.
203,Jarrold,Fields,010-741-8105,sit.amet@google.couk,"Jan 6, 1982",Canada,Baddeck,K3X 6Z5,441-8969 Rhoncus Road
204,Emerson,Sharp,068-138-9383,ante.iaculis@outlook.ca,"Oct 15, 1994",Canada,Maple Creek,Y2C 9E6,"517-6759 Ut, Av."
```

12. Creo la tabla user y verifica que se haya creado correctamente. No estaba segura si debía crear la columna birth\_date como DATE o VARCHAR. Entiendo que en Power BI se puede cambiar el formato.

```

49 ● CREATE TABLE IF NOT EXISTS user (
50     id INT PRIMARY KEY,
51     name VARCHAR(100),
52     surname VARCHAR(100),
53     phone VARCHAR(150),
54     email VARCHAR(150),
55     birth_date VARCHAR(100),
56     country VARCHAR(150),
57     city VARCHAR(150),
58     postal_code VARCHAR(100),
59     address VARCHAR(255)
60 );
61
62

```

Output

Action Output

#	Time	Action	Message
1	13:24:37	CREATE TABLE IF NOT EXISTS user ( id INT PRIMARY KEY, name VARCHAR(100), surname...	0 row(s) affected

```

63 ● SELECT *
64 FROM user;
65
66

```

Result Grid

	id	name	surname	phone	email	birth_date	country	city	postal_code	address
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### 13. Inserto los datos en la tabla user desde users\_usa.csv

```

40 ● LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\users_usa.csv'
41 INTO TABLE user
42 FIELDS TERMINATED BY ','
43 ENCLOSED BY '"'
44 LINES TERMINATED BY '\\r\\n'
45 IGNORE 1 ROWS;

```

### 14. Inserto los datos en la tabla user desde users\_uk.csv

```

16 ● LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\users_uk.csv'
17 INTO TABLE user
18 FIELDS TERMINATED BY ','
19 ENCLOSED BY '"'
20 LINES TERMINATED BY '\\r\\n'
21 IGNORE 1 ROWS
22

```

Output

Action Output

#	Time	Action	Message
1	10:57:00	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\users_uk.csv' INTO TABLE ...	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0

15. Inserto los datos en la tabla user desde users\_ca.csv

```
54 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\users_ca.csv'
55 INTO TABLE user
56 FIELDS TERMINATED BY ','
57 ENCLOSED BY '"'
58 LINES TERMINATED BY '\\r\\n'
59 IGNORE 1 ROWS;
```

16. Verifico que todos los datos se hayan añadido en la tabla user.

The screenshot shows a database management tool interface. At the top, a SQL query is entered: `SELECT * FROM user;`. Below the query, a table of results is displayed. The table has 10 columns: `id`, `name`, `surname`, `phone`, `email`, `birth_date`, `country`, `city`, `postal_code`, and `address`. The data rows show various users with their details. Below the table, the 'Output' section shows the execution of the query: `SELECT * FROM user LIMIT 0, 5000` returned 275 rows in 0.000 seconds.

id	name	surname	phone	email	birth_date	country	city	postal_code	address
267	Ocean	Nelson	079-481-2745	aenean@yahoo.com	Dec 26, 1991	Canada	Charlottetown	85X 3P4	Ap #732-8357 Pede, Rd.
268	Clark	Olson	029-086-1867	nunc@icloud.net	Mar 15, 1987	Canada	Montague	S3Y 1W6	1315 Est Rd.
269	Haley	Fitzpatrick	055-871-6664	in.aliquet@outlook.org	Jan 10, 1996	Canada	Pangnirtung	R0Y 1E3	P.O. Box 914, 451 Nam Rd.
270	Elton	Roberson	096-325-5107	tristique.pharetra@google.net	Oct 12, 1990	Canada	McCallum	R0V 4P6	2857 Natoque Road
271	Leandra	Cherry	089-285-7016	lobortis quis@hotmail.ca	Sep 2, 1991	Canada	Gander	H6S 6M9	554-9293 Sollicitudin Av.
272	Hedwig	Gilbert	064-204-8788	sem eget@icloud.edu	Apr 16, 1991	Canada	Tuktoyaktuk	Q4C 3G7	P.O. Box 496, 5145 Sapien Road
273	Hilary	Ferguson	060-710-1604	sapien.molestie.ori@google.edu	Nov 3, 1981	Canada	Pangnirtung	1ZT 5G4	Ap #736-4628 Cras St.
274	Jameson	Hunt	024-732-2321	fringilla@protonmail.com	Jan 29, 1982	Canada	Township of Min...	B6V 6N4	224-4927 Praesent Ave
275	Kenyon	Hartman	082-871-7248	convallis.ante.lectus@yahoo.com	Aug 3, 1982	Canada	Richmond	R8H 2K2	8564 Facilisi. St.

17. Cambié la columna id para que sea autoincremental. Así, cada nuevo usuario recibe automáticamente un número único y no necesito asignarlo manualmente

```
58 • ALTER TABLE user MODIFY id INT NOT NULL AUTO_INCREMENT;
59
```

The screenshot shows the 'Output' section of the database management tool. It displays the execution of the SQL statement: `ALTER TABLE user MODIFY id INT NOT NULL AUTO_INCREMENT`. The message indicates that 275 rows were affected, with 0 duplicates and 0 warnings.

18. Revisé la estructura de credit\_cards.csv.

El separador de valores es una coma.

La columna id contiene los identificadores únicos de credit cards.

```
id,user_id,iban,pan,pin,cvv,track1,track2,expiring_date
CcU-2938,275,TR301950312213576817638661,5424465566813633,3257,984,%B8383712448554646^WovsxejDpwiev^86041142??,%B7653863056044187=
8007163336?3,10/30/22
CcU-2945,274,DO26854763748537475216568689,5142423821948828,9080,887,%B4621311609958661^UftuyfsSeimxn^0610628241??,%B4149568437843501=
5107140330?1,08/24/23
CcU-2952,273,BG45IVQL52710525608255,4556 453 55 5287,4598,438,%B2183285104307501^CddytcUxwfdq^5907955430?9,%B6778580257827162=
```

19. Creo la tabla credit\_card

```

CREATE TABLE IF NOT EXISTS credit_card (
  id VARCHAR(15) PRIMARY KEY,
  user_id INT,
  iban VARCHAR(35),
  pan VARCHAR(20),
  pin INT,
  cvv INT,
  track1 VARCHAR(255),
  track2 VARCHAR(255),
  expiring_date VARCHAR(8)
);

```

20. Inserto los datos en la tabla credit\_card desde credit\_cards.csv

```

51 LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\credit_cards.csv'
52 INTO TABLE credit_card
53 FIELDS TERMINATED BY ','
54 LINES TERMINATED BY '\n'
55 IGNORE 1 ROWS;
56
57

```

Output

Action Output

#	Time	Action	Message
5	11:35:46	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\credit_cards.csv' INTO T...	275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings: 0

21. Verifico que todos los datos se hayan añadido en la tabla credit\_card

```

56
57 -- check that data is added to credit_card
58 select *
59 FROM credit_card;

```

Result Grid

id	user_id	iban	pan	pin	cvv	track1	track2	expiring_date
CdU-4793	10	HU95215627749276573565556322	471662 767641 7624	7216	848	%87755121567434815^Pfzyp0Mrjdl^860221...	%87412375289135315=390436037575	11/09/23
CdU-4800	9	SI97824334522161436	5455 7952 5528 3322	3745	886	%86267823435548784^Hrfdddd.hnrap^51114...	%81811225531223836=120476246378	05/23/20
CdU-4807	8	LB19298318715580851625676971	4539 4326 8269 4216	8596	626	%88875994714521289^BdmgotvElcnz^36022...	%82436382423563134=410774591475	04/07/22
CdU-4814	7	MR4845282437847152280636374	374471619343357	8790	124	%87721393967157957^AdwifddGdtmcq^4102...	%86168514385624483=9408131572	12/19/20
CdU-4821	6	LT253147505686466784	453987 7873842836	9000	867	%84662497837622681^PawpivvQntntrh^6702...	%84538665274571178=190917113171	07/15/20
CdU-4828	5	BG111LMJ30149367569464	4485252735942	2789	942	%87763464626517588^KmgovvFiovcv^5003...	%84651147303850314=210585916172	09/04/22
CdU-4835	4	PT34592171131763200132583	3723 677744 22550	1149	680	%86221188915952608^EnfmdemGugvnr^580...	%88187762874317668=910791978176	01/08/24
CdU-4842	3	SA2156708581957118818229	3774 636724 83250	4655	750	%8216216733758821^CszvsvPjqcfr^950818...	%82517312164209886=5404789179	11/11/24
CdU-4849	2	SE2813123487163628531121	5223363813491514	9992	779	%88844154447682199^JunfxuIzherj^380534...	%82623983651705584=020755122679	03/21/25
CdU-4856	1	TR373872558313545667124286	349528235713651	9086	974	%84251431580521172^BjklrXqdide^2001250...	%88171116145537298=750285292677	05/19/23
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

credit\_card 7

Output

Action Output

#	Time	Action	Message
1	11:44:23	select * FROM credit_card LIMIT 0, 5000	275 row(s) returned

22. Revisé la estructura de products.csv

El separador de valores es una coma.

La columna id contiene los identificadores únicos de los productos.

Los precios tienen un símbolo de dólar. Además, como representan valores monetarios, pueden tener un máximo de dos decimales después de la coma.

```

x      x      id,product_name,price,colour,weight,warehouse_id
1,Direwolf Stannis,$161.11,#7c7c7c,1,WH-4
2,Tarly Stark,$9.24,#919191,2,WH-3
3,duel tourney Lannister,$171.13,#d8d8d8,1.5,WH-2
4,warden south duel,$71.89,#111111,3,WH-1
5,skywalker ewok,$171.22,#dbdbdb,3.2,WH-0
6,dooku solo,$136.60,#c4c4c4,0.8,WH--1
7,north of Casterly,$63.33,#b7b7b7,0.6,WH--2
8 Winterfell $32 37 #383838 1 1 WH--3

```

23. Creo la tabla product.

```

61 -- create table product
62 CREATE TABLE IF NOT EXISTS product (
63     id INT AUTO_INCREMENT PRIMARY KEY,
64     product_name VARCHAR(300),
65     price DECIMAL(10,2),
66     colour VARCHAR(10),
67     weight DECIMAL(6,2),
68     warehouse_id VARCHAR(10)
69 );
70

```

Output			
Action Output			
#	Time	Action	Message
1	12:11:35	CREATE TABLE IF NOT EXISTS product (id INT AUTO_INCREMENT PRIMARY KEY, product_name VARCHAR...	0 row(s) affected

24. Inserta datos en la tabla product.

Uso SET price = REPLACE(price, '\$', '') + 0 para reemplazar el símbolo de dólar (\$) con una cadena vacía (") y sumo 0 para convertirlo en un valor decimal.

Hay un error:: Error Code: 1366. Incorrect decimal value: '\$161.11' for column 'price' at row 1.

Este error ocurre porque REPLACE() solo funciona con datos de tipo char, varchar.

```

70 -- Insert data to product
71
72 LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\uploads\products.csv'
73 INTO TABLE product
74 FIELDS TERMINATED BY ','
75 LINES TERMINATED BY '\n'
76 IGNORE 1 ROWS
77 (id, product_name, price, colour, weight, warehouse_id)
78 SET price = REPLACE(price, '$', '') + 0;
79

```

Output			
Action Output			
#	Time	Action	Message
1	12:18:11	LOAD DATA INFILE 'C:\ProgramData\MySQL\MySQL Server 8.0\uploads\products.csv' INTO TABLE ...	Error Code: 1366. Incorrect decimal value: '\$161.11' for column 'price' at row 1
			Duration / Fetch
			0.000 sec

25. Para poder reemplazar el símbolo de dólar cambio tipo de datos de price a VARCHAR(50)



```
80 • ALTER TABLE product MODIFY COLUMN price VARCHAR(50);
81
```

Output			
Action Output			
#	Time	Action	Message
✖	1 12:18:11	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv' INTO TABLE ...	Error Code: 1366. Incorrect decimal value: '\$161.11' for column 'price' at row 1
✔	2 12:20:13	ALTER TABLE product MODIFY COLUMN price VARCHAR(50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

26. Inserto otra vez datos a product. Esta vez sin errores.

```
84 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv'
85 INTO TABLE product
86 FIELDS TERMINATED BY ','
87 LINES TERMINATED BY '\n'
88 IGNORE 1 ROWS
89 (id, product_name, price, colour, weight, warehouse_id)
90 SET price = REPLACE(price, '$', '') + 0;
```

Output			
Action Output			
#	Time	Action	Message
✔	1 12:44:52	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\products.csv' INTO TABLE pro...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

27. Cambio el tipo de dato de la columna price a decimal

```
93 • ALTER TABLE product MODIFY COLUMN price DECIMAL(10,2);
```

Output			
Action Output			
#	Time	Action	Message
✔	1 12:49:50	ALTER TABLE product MODIFY COLUMN price DECIMAL(10,2)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0
			Duration / Fetch 0.094 sec

28. En lugar de usar el símbolo \$ dentro del precio, agregó una nueva columna llamada currency. Esta columna guarda la información de la moneda como texto (USD), para saber que todos los precios están en dólares.

```

96 • ALTER TABLE product ADD COLUMN currency VARCHAR(3) DEFAULT 'USD';
97
98 • select *
99 FROM product;
100

```

Result Grid							
Filter Rows:							
Edit: Export/Import: Wrap Cell Content:							
	id	product_name	price	colour	weight	warehouse_id	currency
▶	1	Direwolf Stannis	161.11	#7c7c7c	1.00	WH-4	USD
	2	Tarly Stark	9.24	#919191	2.00	WH-3	USD
	3	duel tourney Lannister	171.13	#d8d8d8	1.50	WH-2	USD
	4	warden south duel	71.89	#111111	3.00	WH-1	USD
	5	skywalker ewok	171.22	#dbdbdb	3.20	WH-0	USD
	6	dooku solo	136.60	#c4c4c4	0.80	WH--1	USD
	7	north of Casterly	63.33	#b7b7b7	0.60	WH--2	USD
	8	Winterfell	32.37	#383838	1.40	WH--3	USD
	9	Winterfell	76.40	#b5b5b5	1.20	WH--4	USD
	10	Karstark Dorne	119.52	#f4f4f4	2.40	WH--5	USD
	11	Karstark Dorne	49.70	#141414	2.70	WH--6	USD

product 2 x			
Output			
Action Output			
#	Time	Action	Message
✓ 1	12:49:50	ALTER TABLE product MODIFY COLUMN price DECIMAL(10,2)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0
✗ 2	12:52:21	ALTER TABLE products ADD COLUMN currency VARCHAR(3) DEFAULT 'USD'	Error Code: 1146. Table 'sales.products' doesn't exist
✓ 3	12:52:32	ALTER TABLE product ADD COLUMN currency VARCHAR(3) DEFAULT 'USD'	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 4	12:53:02	select * FROM product LIMIT 0, 5000	100 row(s) returned

## 29. Revisé la estructura de transactions.csv

El separador de valores es ;.

La columna id contiene los identificadores únicos de los productos.

```

id;card_id;business_id;timestamp;amount;declined;product_ids;user_id;lat;longitude
108B1D1D-5B23-A76C-55EF-C568E49A05DD;CCU-2938;b-2222;2021-07-07 17:43:16;293.57;0;59;275;83.7839152128;-178.860353536
7DC26247-20EC-53FE-E555-B6C2E55CA5D5;CCU-2945;b-2226;2022-02-04 15:52:56;312.5;0;71, 41;275;58.9367181312;-76.8171099136
72997E96-DC2C-A4D7-7C24-66C302F8AE5A;CCU-2952;b-2230;2022-01-30 15:16:36;239.87;0;97, 41, 3;275;43.3584055296;-17.6579677184
AB069F53-965E-A2A8-CE06-CA8C4FD92501;CCU-2959;b-2234;2021-04-15 13:37:18;60.99;0;11, 13, 61, 29;275;1.6481916928;-158.0065729536

```

## 30. Creo la tabla transaction. La tabla referencia las tablas credit\_card, company, user (FKs)

```

111 -- Creamos la tabla transaction
112 • CREATE TABLE IF NOT EXISTS transaction (
113     id VARCHAR(255) PRIMARY KEY,
114     credit_card_id VARCHAR(15),
115     company_id VARCHAR(15),
116     user_id INT,
117     lat DECIMAL (12,10),
118     longitude DECIMAL(13,10),
119     timestamp TIMESTAMP,
120     amount DECIMAL(10, 2),
121     declined BOOLEAN,
122     product_ids VARCHAR(100),
123     FOREIGN KEY (credit_card_id) REFERENCES credit_card(id),
124     FOREIGN KEY (company_id) REFERENCES company(company_id),
125     FOREIGN KEY (user_id) REFERENCES user(id)
126 );
127

```

Output			
Action Output			
#	Time	Action	Message
✓ 1	10:56:51	CREATE TABLE IF NOT EXISTS transaction ( id VARCHAR(255) PRIMARY KEY, credit_card_id ...	0 row(s) affected

## 31. Inserto los datos en la tabla transaction.

El orden de las columnas en el CSV es diferente al de la tabla transaction, por eso en LOAD DATA INFILE especifiqué el orden explícitamente entre paréntesis

```

130 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\transactions.csv'
131 INTO TABLE transaction
132 FIELDS TERMINATED BY ','
133 LINES TERMINATED BY '\\n'
134 IGNORE 1 ROWS
135 (id, credit_card_id, company_id, timestamp, amount, declined, product_ids, user_id, lat, longitude);
136
137 • select *
138 from transaction;
139

```

Output

#	Time	Action	Message
1	11:02:27	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\uploads\\transactions.csv' INTO TA...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0

32. La columna product\_ids tiene varios ids de productos en una misma transacción, pero para organizar mejor los datos, hay que normalizarla. Por eso, creo una tabla intermedia (transaction\_product) donde cada fila tendrá solo un transaction\_id y un product\_id. Así, cada producto se almacena por separado y es más fácil hacer consultas en SQL.

	id	product_ids
▶	02C6201E-D90A-1859-B4EE-88D2986D3B02	71, 1, 19
	0466A42E-47CF-8D24-FD01-C0B689713128	47, 97, 43
	063FBA79-99EC-66FB-29F7-25726D1764A5	47, 67, 31, 5
	0668296C-CDB9-A883-76BC-2E4C44F8C8AE	89, 83, 79
	06CD9AA5-9B42-D684-DDDD-A5E394FEBA99	43, 31

33. Create transaction\_product table.

Esta tabla intermedia transaction\_product normaliza la relación muchos a muchos entre transacciones y productos. La clave primaria compuesta (PRIMARY KEY (transaction\_id, product\_id)) garantiza que cada combinación de transacción y producto sea única, evitando duplicados. Además, las claves foráneas aseguran que transaction\_id exista en la tabla transaction y product\_id en la tabla product, manteniendo la integridad de los datos.

```

142 • CREATE TABLE IF NOT EXISTS transaction_product(
143     transaction_id VARCHAR(255),
144     product_id INT,
145     PRIMARY KEY (transaction_id, product_id), -- composite PK
146     FOREIGN KEY (transaction_id) REFERENCES transaction(id),
147     FOREIGN KEY (product_id) REFERENCES product(id)
148 );
149
150
151

```

Output

#	Time	Action	Message
1	11:05:43	CREATE TABLE IF NOT EXISTS transaction_product(transaction_id VARCHAR(255), product_id INT, PRI...	0 row(s) affected

34. Inserto los datos en la tabla transaction\_product table

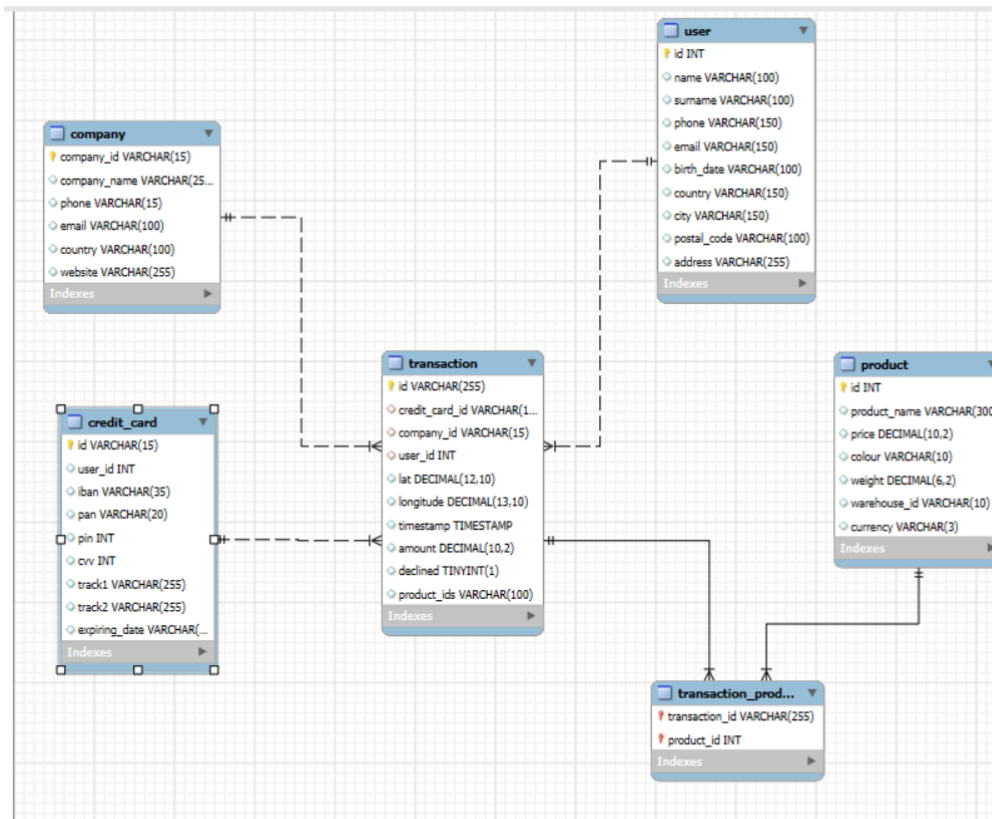
```

151 • INSERT INTO transaction_product (transaction_id, product_id)
152   SELECT t.id, p.id
153   FROM transaction t
154   JOIN product p ON FIND_IN_SET(p.id, REPLACE (t.product_ids, ' ', '' )) > 0;
155
156
157

```

Output			
Action Output			
#	Time	Action	Message
1	11:05:43	CREATE TABLE IF NOT EXISTS transaction_product(transaction_id VARCHAR(255), product_id INT, PRI...	0 row(s) affected
2	11:15:17	INSERT INTO transaction_product (transaction_id, product_id) SELECT t.id, p.id FROM transaction t JOIN ...	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0

## Final diagram:



## Estructura del Star Schema en esta base de datos:

- Fact table (transaction)
  - Es el centro del esquema y almacena eventos clave - transacciones.
  - Se conecta con tablas de dimensión company, credit\_card, user a través de claves foráneas.
- Dimension tables
  - User
  - Company
  - Credit\_card
  - product

- Tabla intermedia (transaction\_product) que normaliza la relación entre transaction y product.

## - Exercici 1

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

Junto las tablas user y transaction, que tienen en común el campo id de usuario. Luego, GROUP BY user\_id) y cuento cuántas transacciones tiene cada usuario. Después, filtro los usuarios con más de 30 transacciones.

```
160 • SELECT t.user_id,
161         COUNT(t.user_id) AS transactions_count
162     FROM user u
163         JOIN transaction t
164         ON u.id = t.user_id
165     GROUP BY t.user_id
166     HAVING transactions_count > 30
167     ORDER BY transactions_count ASC;
```

Result Grid		
Filter Rows:		
Export:   Wrap Cell Content:		
	user_id	transactions_count
▶	92	39
	275	48
	267	52
	272	76

Result 64 x				
Output				
Action Output				
#	Time	Action	Message	
✓ 1	13:21:11	SELECT t.user_id, COUNT(t.user_id) AS transactions_count FROM user u JOIN transaction t	... 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.**

**Shows the average amount per IBAN of credit cards in the company Donec Ltd, use at least 2 tables.**

Uno las tablas company, transaction y credit\_card y filtro solo las transacciones de la empresa Donec Ltd. Después, GROUP BY company\_id, credit\_card\_id, iban y cálculo avg amount de las transacciones para cada IBAN. Como el formato de amount en la tabla transaction es decimal(10,2), uso RUND para devolver AVG(amount).

```

171 • SELECT
172     t.company_id,
173     t.credit_card_id,
174     c.company_name,
175     cc.iban,
176     ROUND(AVG(t.amount),2) AS avg_amount
177 FROM company c
178     JOIN transaction t
179         ON c.company_id = t.company_id
180         AND c.company_name = "Donec Ltd"
181     JOIN credit_card cc
182         ON cc.id = t.credit_card_id
183 GROUP BY t.company_id,
184         t.credit_card_id,
185         cc.iban;
186

```

Result Grid					
	company_id	credit_card_id	company_name	iban	avg_amount
▶	b-2242	CcU-2973	Donec Ltd	PT87806228135092429456346	203.72

## Nivell 2

Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions van ser declinades i genera la següent consulta:

```

259 • CREATE TABLE credit_card_status
260 (
261     credit_card_id VARCHAR(15) PRIMARY KEY,
262     card_status ENUM('Active', 'Inactive') NOT NULL,
263     FOREIGN KEY (credit_card_id) REFERENCES credit_card(id) ON DELETE CASCADE
264 );
265

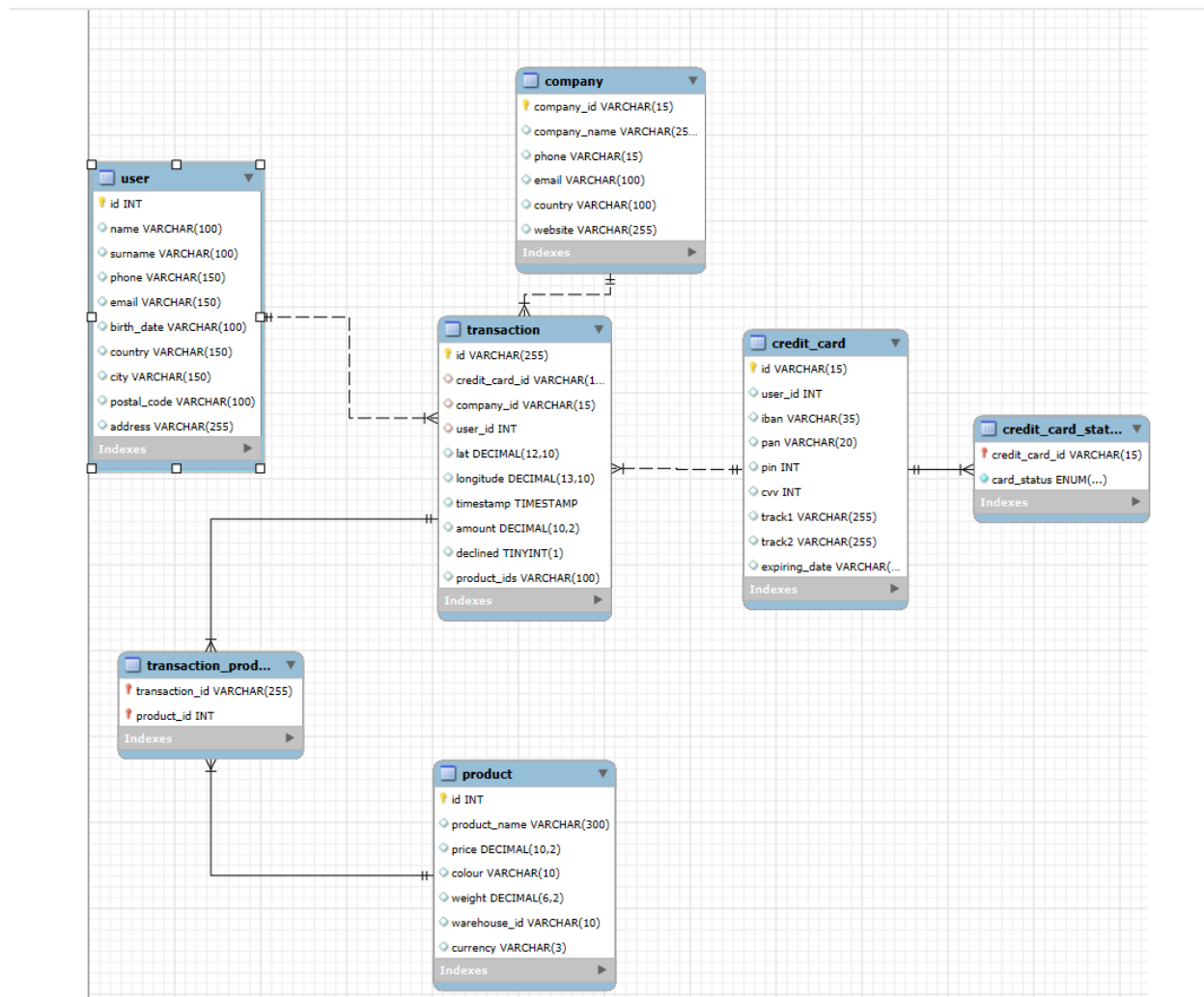
```

Output

Action Output

#	Time	Action	Message
✓ 1	16:40:34	CREATE TABLE credit_card_status ( credit_card_id VARCHAR(15) PRIMARY KEY, card_status EN...	0 row(s) affected

Diagrama después de crear la tabla credit\_card\_status.



Inserto los datos en la tabla desde select.

- CTE ranked\_transactions: asigna un número de orden a cada transacción por tarjeta de crédito, ordenándolas de más reciente a más antigua.
- CTE transaction\_card\_status: verifica si las 3 transacciones más recientes de una tarjeta fueron rechazadas (declined = 3). Si esto ocurre, la tarjeta se marca como "Inactive", de lo contrario, se mantiene como "Active".
- al hacer el JOIN, se asigna el estado de la tarjeta desde transaction\_card\_status. Si no tiene estado, se le pone 'Active' por defecto con COALESCE, asegurando que todas las tarjetas tengan un estado.

```

191
192 • INSERT INTO credit_card_status (credit_card_id, card_status)
193 WITH ranked_transactions AS (
194 SELECT
195     credit_card_id,
196     declined,
197     RANK() OVER(PARTITION BY credit_card_id ORDER BY timestamp DESC) AS transaction_rank
198 FROM transaction
199 ),
200 transaction_card_status AS (
201 SELECT
202     credit_card_id,
203     CASE WHEN SUM(declined) = 3 THEN "Inactive" ELSE "Active" END AS status
204 FROM ranked_transactions
205 WHERE transaction_rank <= 3
206 GROUP BY credit_card_id
207 )
208 SELECT
209     cc.id,
210     COALESCE(tsc.status, 'Active') AS status
211 FROM credit_card cc
212 LEFT JOIN transaction_card_status tsc
213     ON cc.id = tsc.credit_card_id;
214
215 • SELECT
216     product_id,
217     COUNT(transaction_id) AS total_sales
218 FROM transaction_product
219 GROUP BY product_id;
220
221
222
223

```

Output

#	Time	Action	Message
1	23:08:00	INSERT INTO credit_card_status (credit_card_id, card_status) WITH ranked_transactions AS ( SELECT credit_card_id, declined, RANK() OVER(PA...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

```

214
215 • SELECT *
216 FROM credit_card_status
217 LIMIT 10

```

Result Grid

credit_card_id	card_status
CdU-2938	Active
CdU-2945	Active
CdU-2952	Active
CdU-2959	Active
CdU-2966	Active
CdU-2973	Active
CdU-2980	Active
CdU-2987	Active
CdU-2994	Active
CdU-3001	Active

credit\_card\_status 85 x

Output





#	Time	Action	Message
1	23:08:00	INSERT INTO credit_card_status (credit_card_id, card_status) WITH ranked_transactions AS ( SELECT credit_card_id, declined, RANK() OVER(PA...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0
2	23:09:05	SELECT * FROM credit_card_status LIMIT 10	10 row(s) returned

## Exercici 1

Quantes targetes estan actives?



```
214
215 • SELECT COUNT(*) as active_cards_count
216 FROM credit_card_status
217 WHERE status = "Active"
```

Result Grid			Filter Rows: <input type="text"/>	Export: 
	active_cards_count			
	275			

## Nivell 3

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

- Esta tabla ya fue creada en el ejercicio anterior. La tabla transaction\_product ya conecta las transacciones con los productos mediante product\_id.

## Exercici 1

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

- La tabla transaction\_product registra cada producto por transacción, por lo que solo es necesario agrupar por product\_id y usar COUNT(transaction\_id) para contar cuántas veces se ha vendido cada producto.

```

285 • SELECT
286     product_id,
287     COUNT(transaction_id) AS total_sales
288 FROM transaction_product
289 GROUP BY product_id;
290

```

Result Grid		
Filter Rows: <input type="text"/>		
Export:		
Wrap Cell Content:		
product_id	total_sales	
1	61	
2	65	
3	51	
5	49	
7	54	

Result 80 x

Output

Action Output

#	Time	Action	Message
1	22:33:28	SELECT product_id, COUNT(transaction_id) AS total_sales FROM transaction_product GROUP BY prod...	26 row(s) returned