

Tasca S4.01. Creació de Base de Dades

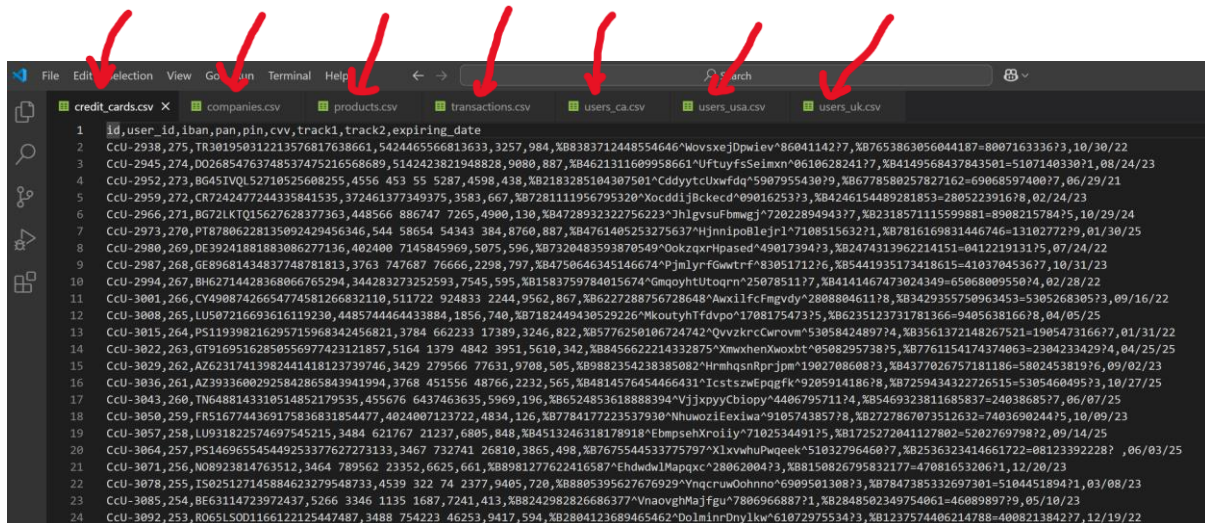
Nivel 1

Descarga los archivos CSV, estúdielas y diseña una base de datos con un esquema de estrella que contenga, al menos 4 tablas de las que puedas realizar las siguientes consultas:

Creamos primero la base de datos transacciones donde estarán todas las tablas que cargaremos desde los archivos CSV's.

Mediante use transacciones; pondremos por defecto esta base de datos.

Descargamos en nuestro ordenador los archivos CSV's y los visualizamos con Visual Studio Code.



Creamos las tablas con Create table y verificamos si existen con el "IF NOT EXISTS".

También creamos las Primary key en cada tabla y en las tablas products y users crearemos auto_increment, y tipo de dato INT


En la mayoría de caso creamos el tipo de dato VARCHAR (255), y en casos que observemos que son enteros: INT, en Price y amount: decimal (10,2), ya que manejaran datos de precios y montos, y en la columna timestamp tipo de dato TIMESTAMP, en las columnas de lat y longitude: FLOAT, ya que son coordenadas.

```

4 • CREATE DATABASE IF NOT EXISTS transacciones;
5 • USE transacciones;
6
7 -- creamos las tablas
8 • CREATE TABLE IF NOT EXISTS companies (
9     company_id VARCHAR(255) PRIMARY KEY,
10    company_name VARCHAR(255),
11    phone VARCHAR(255),
12    email VARCHAR(255),
13    country VARCHAR(255),
14    website VARCHAR(255)
15 );
16
17 • CREATE TABLE IF NOT EXISTS credit_cards (
18     id VARCHAR(255) PRIMARY KEY,
19     user_id INT,
20     iban VARCHAR(255),
21     pan VARCHAR(255),
22     pin VARCHAR(255),
23     cvv INT,
24     track1 VARCHAR(255),
25     track2 VARCHAR(255),
26     expiring_date VARCHAR(255)
27 );
28
29 • CREATE TABLE IF NOT EXISTS products (
30     id INT PRIMARY KEY AUTO_INCREMENT,
31     product_name VARCHAR(255),
32     price DECIMAL(10,2),
33     colour VARCHAR(255),
34     weight VARCHAR(255),
35     warehouse_id VARCHAR(255)
36 );
37

```

Output

 Action Output

#	Time	Action	Message
✓ 4	19:47:41	CREATE DATABASE IF NOT EXISTS transacciones	1 row(s) affected
✓ 5	19:47:41	USE transacciones	0 row(s) affected
✓ 6	19:47:41	CREATE TABLE IF NOT EXISTS companies (company_id VARCHAR(255) PRIMARY KEY, company_name VA...	0 row(s) affected
✓ 7	19:47:41	CREATE TABLE IF NOT EXISTS credit_cards (id VARCHAR(255) PRIMARY KEY, user_id INT, iban VARCH...	0 row(s) affected
✓ 8	19:47:41	CREATE TABLE IF NOT EXISTS products (id INT PRIMARY KEY AUTO_INCREMENT, product_name VARCH...	0 row(s) affected

```

37
38 • CREATE TABLE IF NOT EXISTS transactions (
39     id VARCHAR(255) PRIMARY KEY,
40     card_id VARCHAR(255),
41     business_id VARCHAR(255),
42     timestamp TIMESTAMP,
43     amount DECIMAL(10, 2),
44     declined TINYINT,
45     product_ids INT,
46     user_id INT,
47     lat FLOAT,
48     longitude FLOAT
49 );
50
51
52 -- los campos de las tablas user_ca, user_uk, users_usa son iguales en estructura, así que uniremos los 3 archivos csv en una sola tabla users
53 • CREATE TABLE IF NOT EXISTS users (
54     id INT PRIMARY KEY AUTO_INCREMENT,
55     name VARCHAR(255),
56     surname VARCHAR(255),
57     phone VARCHAR(255),
58     email VARCHAR(255),
59     birth_date VARCHAR(255),
60     country VARCHAR(255),
61     city VARCHAR(255),
62     postal_code VARCHAR(255),
63     address VARCHAR(255)
64 );
65
66 -- Ejercicio 1
67 -- Realiza una subconsulta que muestre a todos los usuarios con más de 30 transacciones utilizando al menos 2 tablas.
68

```

Output				
Action Output				
#	Time	Action	Message	
✓ 1	20:11:18	CREATE TABLE IF NOT EXISTS transactions (id VARCHAR(255) PRIMARY KEY, card_id VARCHAR(255), b...	0 row(s) affected	
✓ 2	20:11:18	CREATE TABLE IF NOT EXISTS users (id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(255), surname...	0 row(s) affected	

Comprobamos que se hayan creado las tablas con describe:

- **Tabla company**

```

66 • DESC companies;
67

```

Result Grid							
Filter Rows:							
Export: Wrap Cell Content:							
	Field	Type	Null	Key	Default	Extra	
▶	company_id	varchar(255)	NO	PRI	NULL		
	company_name	varchar(255)	YES		NULL		
	phone	varchar(255)	YES		NULL		
	email	varchar(255)	YES		NULL		
	country	varchar(255)	YES		NULL		
	website	varchar(255)	YES		NULL		

Result 37				
Output				
Action Output				
#	Time	Action	Message	
✓ 1	18:30:47	DESC companies	6 row(s) returned	

- **Tabla credit_cards**

66 • DESC credit_cards;

67

Result Grid						
		Filter Rows:			Export:	Wrap Cell Content: IA
	Field	Type	Null	Key	Default	Extra
▶	id	varchar(255)	NO	PRI	NULL	
	user_id	int	YES		NULL	
	iban	varchar(255)	YES		NULL	
	pan	varchar(255)	YES		NULL	
	pin	varchar(255)	YES		NULL	
	cvv	int	YES		NULL	
	track1	varchar(255)	YES		NULL	
	track2	varchar(255)	YES		NULL	
	expiring_date	varchar(255)	YES		NULL	

Result 39 ×

Output

Action Output

#	Time	Action	Message
✓ 1	18:33:36	DESC credit_cards	9 row(s) returned

• Tabla products

66 • DESC products;

67

Result Grid						
		Filter Rows:			Export:	Wrap Cell Content: IA
	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	product_name	varchar(255)	YES		NULL	
	price	varchar(255)	YES		NULL	
	colour	varchar(255)	YES		NULL	
	weight	varchar(255)	YES		NULL	
	warehouse_id	varchar(255)	YES		NULL	

Result 41 ×

Output

Action Output

#	Time	Action	Message
✓ 1	18:35:55	DESC products	6 row(s) returned

• Tabla transactions

66 • DESC transactions;

67

Field	Type	Null	Key	Default	Extra
id	varchar(255)	NO	PRI	NULL	
card_id	varchar(255)	YES		NULL	
business_id	varchar(255)	YES		NULL	
timestamp	timestamp	YES		NULL	
amount	decimal(10,2)	YES		NULL	
declined	tinyint	YES		NULL	
product_ids	varchar(255)	YES		NULL	
user_id	int	YES		NULL	
lat	float	YES		NULL	
longitude	float	YES		NULL	

Result 45 x

Output

Action Output

#	Time	Action	Message
✓ 1	18:38:34	DESC transactions	10 row(s) returned

• Tabla users

66 • DESC users;

67

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

Result 46 x

Output

Action Output

#	Time	Action	Message
✓ 1	18:40:22	DESC users	10 row(s) returned

Introducción de datos en las tablas creadas

- Tabla companies

Previamente a la introducción de datos en las tablas, nos cercioramos de que no haya datos en las tablas, adjunto imagen abajo de una tabla companies, por ejemplo.

```
77 • SELECT * FROM companies; -- miramos si existe información en la tabla companies antes de cargar en introducir los datos.
78
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	company_id	company_name	phone	email	country	website
*	NULL	NULL	NULL	NULL	NULL	NULL

companies 48 x

Output

Action Output

#	Time	Action	Message
1	20:12:43	SELECT * FROM companies	0 row(s) returned

Después procederemos a la introducción de los datos:

Usamos Load data infile para la carga del archivo csv, se pone la ruta del archivo, especificamos que los campos estén separados por comas, y que tome los datos encerrados por "" como un campo, y que ignore la primera línea. Y finalmente, comprobamos nuevamente que hayan cargado todos los datos en la tabla.

```
79 • LOAD DATA
80 INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\companies.csv'
81 INTO TABLE companies
82 FIELDS TERMINATED BY ','
83 ENCLOSED BY '"'
84 IGNORE 1 ROWS;
85
86 • SELECT * FROM companies; -- comprobamos que se hayan cargado en introducido los datos en la tabla companies
87
```

Result Grid						Filter Rows:		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@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-2260	Uterque Lorem DC	03 03 87 33 40	suspendisse.integer@aol.net	Belgium	https://linkedin.com/sub/cars								

Output			
Action Output			
#	Time	Action	Message
1	20:17:17	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\companies.csv' INTO TABLE comp...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
2	20:17:17	SELECT * FROM companies	100 row(s) returned

- Tabla credit_cards

Comprobamos que no exista datos en la tabla credit_cards.

```
98 • SELECT * FROM credit_cards;
99
```

Result Grid									
	id	user_id	iban	pan	pin	cvv	track1	track2	expiring_date
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

credit_cards 58 x credit_cards 59			
Output			
Action Output			
#	Time	Action	Message
1	20:55:35	SELECT * FROM credit_cards	0 row(s) returned

Cargamos los datos del archivo csv correspondiente a credit_cards e introducimos los datos, especificamos que separe los campos terminados en comas y que todo lo que este entre comillas lo considere un campo, e ignore la primera fila, al tener los encabezados ya creados en la creación de la tabla vacía. Verificamos nuevamente que hayan cargado todos los datos en la tabla.

```
88 -- tabla credit_cards
89 • SELECT * FROM credit_cards;
90
91 • LOAD DATA
92 INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\credit_cards.csv'
93 INTO TABLE credit_cards
94 FIELDS TERMINATED BY ','
95 ENCLOSED BY '"'
96 IGNORE 1 ROWS;
97
98 • SELECT * FROM credit_cards;
99
```

Result Grid									
Filter Rows:									
Edit: Export/Imports: Wrap Cell Contents:									
id	user_id	iban	pan	pin	cvv	track1	track2	expiring_date	
CcU-2938	275	TR301950312213576817638661	5424465566813633	3257	984	%B8383712448554646^WovsxejDpwiev^8604...	%B7653863056044187=800716333623	10/30/22	
CcU-2945	274	DO26854763748537475216568689	5142423821948828	9080	887	%B4621311609958661^UftuyfsSeimxn^06106...	%B4149568437843501=510714033021	08/24/23	
CcU-2952	273	BG45IVQL52710525608255	4556 453 55 5287	4598	438	%B2183285104307501^CddyrtUxwfdq^5907...	%B6778580257827162=6906859740077	06/29/21	
CcU-2959	272	CR7242477244335841535	372461377349375	3583	667	%B7281111956795320^XocddjBckecd^09016...	%B4246154489281853=280522391678	02/24/23	
CcU-2966	271	BG72LKTQ15627628377363	448566 886747 7265	4900	130	%B4728932322756223^JhlgvsuFbmwigj^7202...	%B2318571115599881=890821578475	10/29/24	
CcU-2973	270	PT87806228135092429456346	544 58654 54343 384	8760	887	%B4761405253275637^HjnnipoBlejrl^7108515...	%B7816169831446746=1310277279	01/30/25	
CcU-2980	269	DE39241881883086277136	402400 7145845969	5075	596	%B7320483593870549^OokzqrHpsed^4901...	%B2474313962214151=041221913175	07/24/22	
CcU-2987	268	GE89681434837748781813	3763 747687 76666	2298	797	%B4750646345146674^PjmlryfGwvtrf^83051...	%B5441935173418615=410370453677	10/31/23	
CcU-2994	267	BH62714428368066765294	344283273252593	7545	595	%B1583759784015674^GmqoyhtUtoqrn^2507...	%B4141467473024349=6506800955074	02/28/22	
credit_cards 58 credit_cards 59 x									
Output									
Action Output									
#	Time	Action	Message						
1	20:55:35	SELECT * FROM credit_cards	0 row(s) returned						
2	20:55:35	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\credit_cards.csv' INTO TABLE credit_cards	275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings: 0						
3	20:55:35	SELECT * FROM credit_cards	275 row(s) returned						

- Tabla products

Hacemos lo mismo que para los anteriores, verificamos que no tenga datos.

Cargamos el archivo csv de products, especificamos que este separado por comas cada campo, que incluya lo que está encerrado entre comillas como un campo, y que ignore la primera fila.

Verificamos nuevamente que hayan cargado todos los datos en la tabla.

```

100  -- tabla products
101  • SELECT * FROM products;
102
103  • LOAD DATA
104  INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\products.csv'
105  INTO TABLE products
106  FIELDS TERMINATED BY ','
107  ENCLOSED BY '"'
108  IGNORE 1 ROWS;
109
110  • SELECT * FROM products;
111

```

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	1.4	WH--3
9	Winterfell	\$76.40	#b5b5b5	1.2	WH--4

#	Time	Action	Message
5	21:10:39	SELECT * FROM products	0 row(s) returned
6	21:12:53	SELECT * FROM products	0 row(s) returned
7	21:12:53	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\products.csv' INTO TABLE produ...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
8	21:12:53	SELECT * FROM products	100 row(s) returned

- Tabla transactions

Hacemos en este caso un cambio en las especificaciones de la introducción, ya que los datos están separados por punto y coma (;) entonces lo cambiamos campos terminados en “;”.

Finalmente comprobamos que los datos se hayan cargado e introducido en la tabla transactions .


```

112 -- table transactions
113 • SELECT * FROM transactions;
114
115 • LOAD DATA
116 INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\transactions.csv'
117 INTO TABLE transactions
118 FIELDS TERMINATED BY ','
119 ENCLOSED BY '"'
120 LINES TERMINATED BY '\\n'
121 IGNORE 1 ROWS;
122
123 • SELECT * FROM transactions;
124

```

Result Grid										
Filter Rows:										
Edit: Export/Import: Wrap Cell Content:										
	id	card_id	business_id	timestamp	amount	declined	product_ids	user_id	lat	longitude
▶	02C6201E-D90A-1859-B4EE-88D2986D3802	CdU-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	CdU-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	CdU-2987	b-2250	2022-01-06 21:25:27	92.61	0	47, 67, 31, 5	275	-81.2227	-129.05
	0668296C-CD89-A883-76BC-2E4C44F8C8AE	CdU-3743	b-2618	2022-01-26 02:07:14	394.18	0	89, 83, 79	265	-34.3593	-100.556
	06CD9AA5-9B42-D684-DDDD-A5E394FEB499	CdU-2959	b-2346	2021-10-26 23:00:01	279.93	0	43, 31	92	33.7381	158.298
	07A46D48-31A3-7E87-65B9-0DA902AD109F	CdU-3225	b-2386	2021-06-28 21:11:42	340.87	1	47, 23	272	38.8342	92.1905
	09DE92CE-6F27-2B87-13B5-9385B2B388E2	CdU-3071	b-2298	2021-05-11 20:40:06	303.05	1	67, 7	275	71.1706	10.5757
	0A476ED9-0C13-1962-F87B-D3563924B539	CdU-4359	b-2302	2022-02-26 20:33:54	430.49	0	29, 41, 11	221	-56.4901	114.801
	0BEB80B7-9D66-1707-CE4B-9DC7E71914B5	CdU-3141	b-2338	2022-03-04 14:54:35	288.81	1	19, 41, 29, 3	272	23.3264	-13.6037
	0C7C3A22-0047-3B01-846D-7B5500A17098	CdU-3200	b-2424	2021-04-10 20:59:41	102.44	1	80, 31	272	63.3515	68.6667
transactions 68 transactions 69 x										
Output										
Action Output										
#	Time	Action	Message							
✓	1 22:02:32	SELECT * FROM transactions	0 row(s) returned							
✓	2 22:02:32	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\transactions.csv' INTO TABLE trans...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0							
✓	3 22:02:32	SELECT * FROM transactions	587 row(s) returned							

- Tabla users

Para montar esta tabla, hay 3 archivos csv que tienen la misma estructura de campos, es por eso por lo que adjuntamos una a una cada archivo a la tabla para formar una sola llamada users.

Los datos están separados por comas, y en este caso especificamos:

“ LINES TERMINATED BY '\\r\\n' ”, este permite leer archivos csv que provienen de Windows, solo hemos usado este formato para estos archivos de users, en el resto no ha hecho falta.

```

125 -- tabla users
126 • SELECT * FROM users;
127
128 • LOAD DATA
129 INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_uk.csv'
130 INTO TABLE users
131 FIELDS TERMINATED BY ','
132 ENCLOSED BY '"'
133 LINES TERMINATED BY '\\r\\n'
134 IGNORE 1 ROWS;
135
136 • LOAD DATA

```

id	name	surname	phone	email	birth_date	country	city	postal_code	address
1									

users 2 x

Output

Action Output

#	Time	Action	Message
1	10:21:46	SELECT * FROM users	0 row(s) returned
2	10:21:46	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_uk.csv' INTO TABLE users FI...	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0

Se comprueba que se hayan ingresado los datos:

```

136 • SELECT * FROM users;
137

```

id	name	surname	phone	email	birth_date	country	city	postal_code	address
151	Meghan	Hayden	0800 746 6747	arcu.vel@hotmail.ca	Jul 2, 1980	United Kingdom	Tullibody	A1Y 3TC	Ap #432-4493 Aliquet Rd.
152	Hakeem	Alford	(0111) 367 0184	adipiscing.ligula@google.edu	Sep 30, 1979	United Kingdom	Kettering	O21 7JV	551-8930 Lobortis Street
153	Keegan	Pugh	(016977) 3851	sodales.nisi@aol.org	Jul 27, 1994	United Kingdom	Whitehaven	HQ8V 7YP	Ap #312-5898 Consectetur St.
154	Cooper	Bullock	(021) 2521 6627	et@outlook.net	Nov 2, 1986	United Kingdom	Prestegne	U18 0DN	872-1866 Pede Rd.
155	Joshua	Russell	055 4409 5286	justo.nec.ante@outlook.edu	Jan 23, 1984	United Kingdom	Hatfield	B5H 5CS	Ap #285-4727 Auctor. Av.
156	Remedios	Case	055 3114 1566	mollis.phasellus.libero@aol.com	Oct 9, 1994	United Kingdom	North Berwick	QR0 8CW	479-3690 Turpis Road

users 3 x

Output

Action Output

#	Time	Action	Message
2	10:21:46	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_uk.csv' INTO TABLE users ...	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0
3	10:35:11	SELECT * FROM users	50 row(s) returned

Luego continuamos ingresando los datos de los archivos csv restantes:

```

137
138 • LOAD DATA
139 INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_ca.csv'
140 INTO TABLE users
141 FIELDS TERMINATED BY ','
142 ENCLOSED BY '"'
143 LINES TERMINATED BY '\\r\\n'
144 IGNORE 1 ROWS;
145
146 • LOAD DATA
147 INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_usa.csv'
148 INTO TABLE users
149 FIELDS TERMINATED BY ','
150 ENCLOSED BY '"'
151 LINES TERMINATED BY '\\r\\n'
152 IGNORE 1 ROWS;
153
154 • SELECT * FROM users;
155

```

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

users 4 x

Output

Action Output

#	Time	Action	Message
2	10:37:07	LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_usa.csv' INTO TABLE users...	150 row(s) affected Records: 150 Deleted: 0 Skipped: 0 Warnings: 0
3	10:37:07	SELECT * FROM users	275 row(s) returned

CREAR CONSTRAINT

Para relacionar las tablas crearemos las constraint de la foreign key, relacionaremos las tablas users, companies y credit_cards con la tabla transactions.

```

157 -- se crean los constraint para relacionar las tablas
158 • ALTER TABLE transactions
159 ADD CONSTRAINT fk_trans_users
160 FOREIGN KEY(user_id) REFERENCES users(id);
161
162 • ALTER TABLE transactions
163 ADD CONSTRAINT fk_trans_comp
164 FOREIGN KEY(business_id) REFERENCES companies(company_id);
165
166 • ALTER TABLE transactions
167 ADD CONSTRAINT fk_trans_credit
168 FOREIGN KEY(card_id) REFERENCES credit_cards(id);
169

```

Output

Action Output

#	Time	Action	Message
1	10:40:36	ALTER TABLE transactions ADD CONSTRAINT fk_trans_users FOREIGN KEY(user_id) REFERENCES users(id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
2	10:40:36	ALTER TABLE transactions ADD CONSTRAINT fk_trans_comp FOREIGN KEY(business_id) REFERENCES companie...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
3	10:40:37	ALTER TABLE transactions ADD CONSTRAINT fk_trans_credit FOREIGN KEY(card_id) REFERENCES credit_cards(id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

Para relacionar la tabla productos con transactions, nos da un error:

'Error Code: 3780. Referencing column 'product_ids' and referenced column 'id' in foreign key constraint 'fk_trans_prod' are incompatible'.

Este error se refiere a que las columnas que queremos relacionar en ambas tablas son incompatibles y se refiere a que en la tabla transactions la columna product_ids contiene varios valores en cada celda, que son los productos comprados en esa transaccion.

```
174 • SELECT product_ids
175 FROM TRANSACTIONS;
176
177
178 • SELECT n_id
```

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

product_ids
71, 1, 19
47, 97, 43
47, 67, 31, 5
89, 83, 79
43, 31
47, 23
67, 7
29, 41, 11
19, 41, 29, 3
89, 31
83, 43, 73, 61
7, 47, 17
27, 12

TRANSACTIONS 5 x

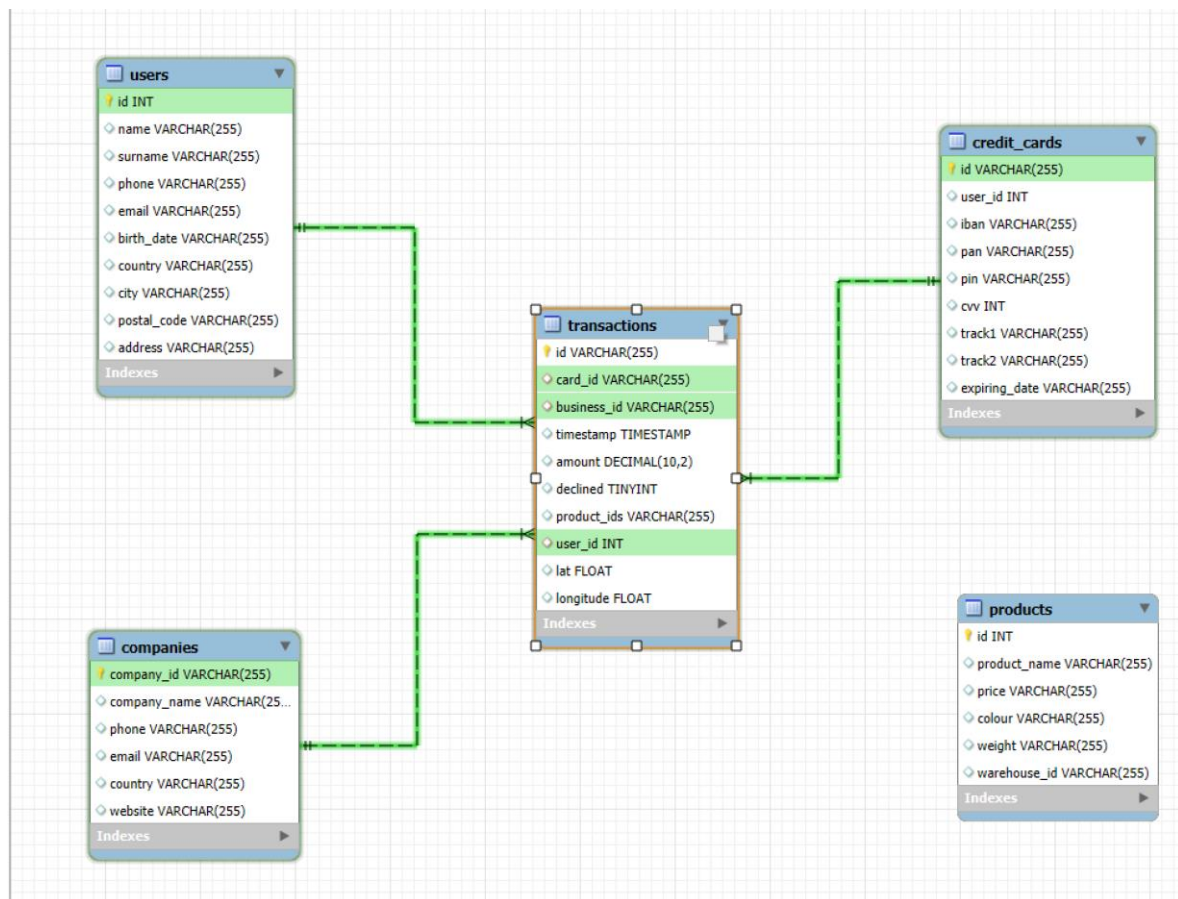
Output

Action Output

#	Time	Action	Message
✓ 1	11:02:27	SELECT product_ids FROM TRANSACTIONS	587 row(s) returned

Para nuestro proyecto, dejaremos momentáneamente sin relacionar la tabla products, y quedaría nuestro diagrama de la siguiente forma:

Sería un modelo en estrella donde la tabla de hechos es transactions y las tablas dimensiones serían: users, credit_cards, companies.



- Exercici 1

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

```

177 -- - Ejercicio 1
178 -- Realiza una subconsulta que muestre a todos los usuarios con más de 30 transacciones utilizando al menos 2 tablas.
179 • SELECT u.id, COUNT(*) AS Num_Transacciones
180 FROM transactions t
181 INNER JOIN users u ON t.user_id = u.id
182 GROUP BY u.id
183 HAVING Num_Transacciones > 30
184 ORDER BY Num_Transacciones DESC;
185
  
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
id	Num_Transacciones			
272	76			
267	52			
275	48			
92	39			

Result 6 x			
Output			
Action Output			
#	Time	Action	Message
1	11:08:24	SELECT u.id, COUNT(*) AS Num_Transacciones FROM transactions t INNER JOIN users u ON t.user_id = u.id GRO...	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.

```
187 -- - Ejercicio 2
188 -- Muestra la media de amount por IBAN de las tarjetas de crédito en la compañía Donec Ltd., utiliza por lo menos 2 tablas.
189
190 • SELECT round(AVG(t.amount),2) AS Media_Amount_Donet
191 FROM transactions t
192 INNER JOIN credit_cards cr ON t.card_id = cr.id
193 WHERE t.business_id = (SELECT company_id FROM companies WHERE company_name = 'Donec Ltd');
194
195 -- TRUNCATE transactions;
```

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

Media_Amount_Donet
203.72

Result 7 x

Output

Action Output

#	Time	Action	Message
1	11:17:36	SELECT round(AVG(t.amount),2) AS Media_Amount_Donet FROM transactions t INNER JOIN credit_cards cr ON t.ca...	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:

```

201 -- Nivell 2
202 -- Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat
203 -- en si les últimes tres transaccions van ser declinades i genera la següent consulta:
204 • CREATE TABLE activ_card AS
205 SELECT card_id,
206 CASE
207 WHEN sum(declined) >= 3 THEN 'no activo'
208 ELSE 'activo'
209 END AS actividad
210 FROM (SELECT card_id, declined,
211 ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS row_num
212 FROM transactions
213 ) AS hist_card
214 WHERE row_num <=3
215 GROUP BY card_id;
216
217 • SELECT * FROM activ_card;
218

```

Result Grid			
	card_id	actividad	
▶	CdU-2938	activo	
	CdU-2945	activo	
	CdU-2952	activo	
	CdU-2959	activo	
	CdU-2966	activo	
	CdU-2973	activo	
	CdU-2980	activo	
	CdU-2987	activo	
	CdU-2994	activo	
	CdU-3001	activo	
	CdU-3008	activo	
	CdU-3015	activo	

activ_card 43 x

Output			
Action Output			
#	Time	Action	Message
✓ 1	12:08:30	CREATE TABLE activ_card AS SELECT card_id, CASE WHEN sum(declined) >= 3 THEN 'no activo' ELSE '...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0
✓ 2	12:08:30	SELECT * FROM activ_card	275 row(s) returned

Una vez creada esta tabla la relaciono con Credit_cards

```

215 -- Relaciono esta tabla con el modelo
216 • ALTER TABLE activ_card
217 ADD CONSTRAINT fk_credit FOREIGN KEY(card_id) REFERENCES credit_cards(id);
218

```

Output			
Action Output			
#	Time	Action	Message
✓ 14	11:21:54	SELECT * FROM transacciones.credit_cards	275 row(s) returned
✓ 15	11:22:34	SELECT * FROM activ_card	275 row(s) returned
✓ 16	11:26:42	ALTER TABLE activ_card ADD CONSTRAINT fk_credit FOREIGN KEY(card_id) REFERENCES credit_cards(id)	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

- Exercici 1

Quantes targetes estan actives?

```

219 -- - Ejercici 1
220 -- Quantas targetes estan actives?
221 • SELECT count(*)
222 FROM activ_card
223 WHERE actividad = 'activo';

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
count(*)			
275			

Result 44 x

Output

Action Output

#	Time	Action	Message
✓ 1	12:10:28	SELECT count(*) FROM activ_card WHERE actividad = 'activo'	1 row(s) returned

Nivel 3

Crea una tabla con la que podamos unir los datos del nuevo archivo products.csv con la base de datos creada, teniendo en cuenta que desde transaction tienes product_ids. Genera la siguiente consulta:

Se ha separado la columna products_ids de la tabla transactions, ya que en cada celda había los id de productos vendidos en una transacción separados por comas, entonces se han separado con Substring_index, respetando la transacción a la que pertenece, con esto se han creado una tabla llamada trans_product.


```

226 • CREATE TABLE trans_product AS
227 SELECT id,
228     SUBSTRING_INDEX(product_ids, ',', 1) AS producto
229 FROM transactions
230 WHERE product_ids LIKE '%'
231 UNION ALL
232     SELECT id,
233     SUBSTRING_INDEX(SUBSTRING_INDEX(product_ids, ',', -2), ',', -1) AS producto
234 FROM transactions
235 WHERE product_ids LIKE '%,%'
236 UNION ALL
237     SELECT id,
238     SUBSTRING_INDEX(SUBSTRING_INDEX(product_ids, ',', 2), ',', -1) AS producto
239 FROM transactions
240 WHERE product_ids LIKE '%,%,%'
241 UNION ALL
242     SELECT id,
243     SUBSTRING_INDEX(SUBSTRING_INDEX(product_ids, ',', -2), ',', 1) AS producto
244 FROM transactions
245 WHERE product_ids LIKE '%,%,%,%';
246
247 • SELECT *
248 FROM trans_product;

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	id	producto				
	02C6201E-D90A-1859-B4EE-88D2986D3B02	71				
	0466A42E-47CF-8D24-FD01-C0B689713128	47				
	063FBA79-99EC-66FB-29F7-25726D1764A5	47				
	0668296C-CD89-A883-76BC-2E4C4F8C8AE	89				
	06CD9AA5-9B42-D684-DDDD-A5E394FEB9A9	43				
	07A46D48-31A3-7E87-65B9-0DA902AD109F	47				
	09DE92CE-6F27-2BB7-13B5-9385B2B3B8E2	67				

trans_product 91 x

Output		
#	Time	Action
1	20:31:17	CREATE TABLE trans_product AS SELECT id, SUBSTRING_INDEX(product_ids, ',', 1) AS producto FROM transactions
2	20:31:17	SELECT * FROM trans_product

Message	
1457 row(s) affected	Records: 1457 Duplicates: 0 Warnings: 0
1457 row(s) returned	

Con esta nueva tabla creada, la relacionaremos con transactions y con products, para establecer una entidad – relación y que sirva de nexo o tabla puente para relacionarlas.

Así que modificamos el tipo de dato en la columna producto de la tabla trans_product a INT, para que se iguale en el tipo a la columna id de la tabla products.

También creamos una primary key compuesta en la tabla trans_product con los dos campos que tiene esta tabla (id y producto)

```

255
256 • ALTER TABLE trans_product ADD CONSTRAINT primary key (id,producto);
257

```

Output		
#	Time	Action
1	11:31:38	ALTER TABLE trans_product ADD CONSTRAINT primary key (id,producto)

Message	
0 row(s) affected	Records: 0 Duplicates: 0 Warnings: 0

Y finalmente, para relacionar Trans_product con trasactions y con products creamos las constraint de foreign keys.

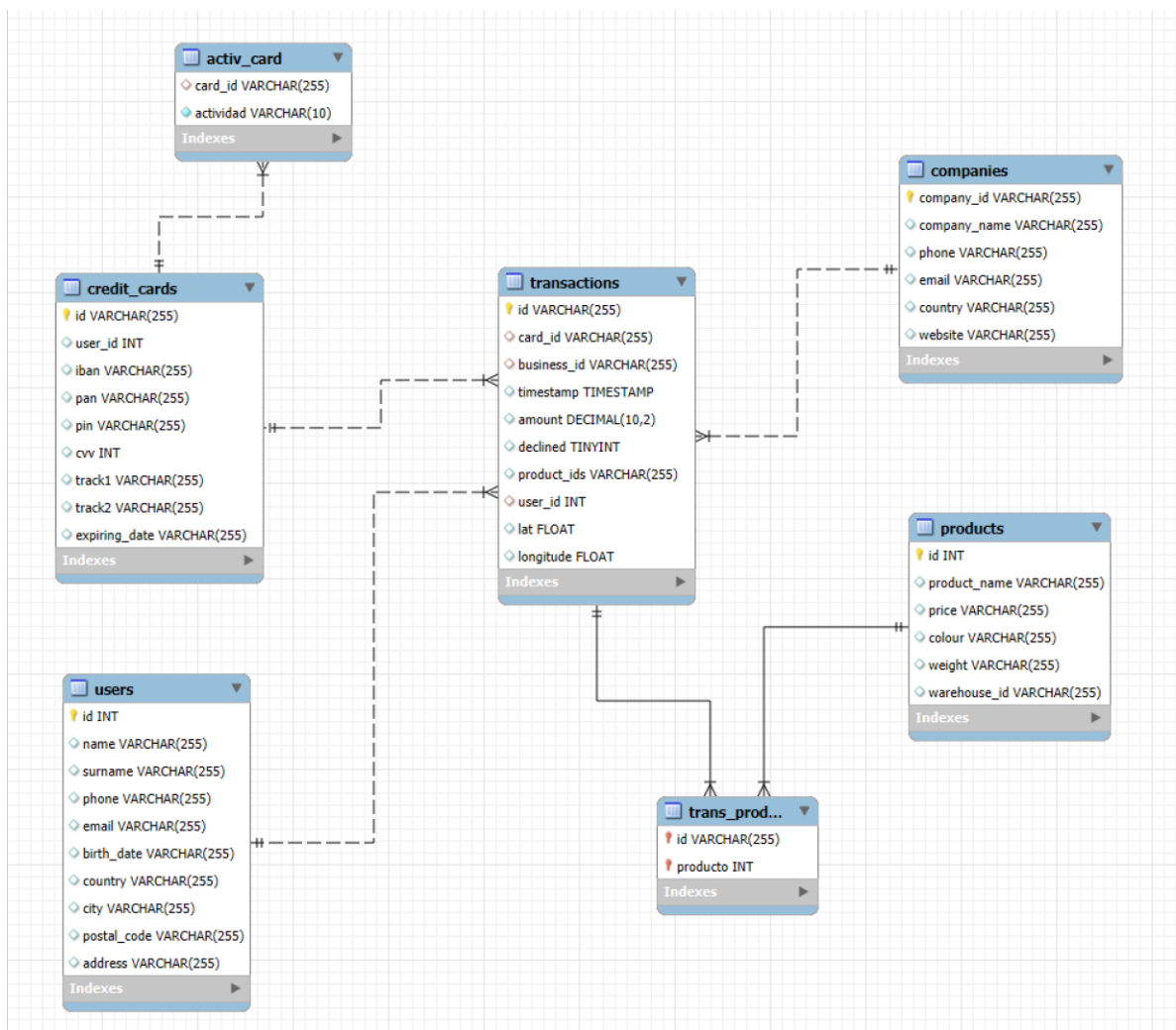
```

253 • ALTER TABLE trans_product MODIFY producto INT; -- modificar tipo de dato a INT
254
255 • ALTER TABLE trans_product
256 ADD CONSTRAINT fk_product FOREIGN KEY(producto) REFERENCES products(id);
257
258 • ALTER TABLE trans_product
259 ADD CONSTRAINT fk_id FOREIGN KEY (id) REFERENCES transactions(id);
260
261

```

#	Time	Action	Message
1	22:42:35	ALTER TABLE trans_product MODIFY producto INT	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0
2	22:42:35	ALTER TABLE trans_product ADD CONSTRAINT fk_product FOREIGN KEY(producto) REFERENCES products(id)	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0
3	22:42:36	ALTER TABLE trans_product ADD CONSTRAINT fk_id FOREIGN KEY (id) REFERENCES transactions(id)	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0

Aquí muestro el diagrama final y sus relaciones.



Ejercicio 1

Necesitamos conocer el número de veces que se ha vendido cada producto.

Aquí mostramos cuantas veces se ha vendido cada producto,

Mostramos un listado de productos con la cantidad de ventas de cada producto que se ha vendido, excluimos las ventas hechas y que hayan sido rechazadas.

```
269 • SELECT tp.producto as product, COUNT(tp.id) AS VECES_VENDIDO
270 FROM trans_product tp
271 INNER JOIN transactions t ON tp.id = t.id
272 WHERE declined = '0'
273 GROUP BY product
274 ORDER BY product;
275
```

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

	product	VECES_VENDIDO
▶	1	51
	2	56
	3	43
	5	42
	7	44
	11	40
	13	51
	17	54
	19	44
	23	60

Result 18 x

Output

Action Output

#	Time	Action	Message
✓ 1	11:46:29	SELECT tp.producto as product, COUNT(tp.id) AS VECES_VENDIDO FROM trans_product tp INNER JOIN transacti...	26 row(s) returned