SPRINT 4

Partiendo de algunos archivos CSV diseñarás y crearás tu base de datos.

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

-- Creamos un nuevo schema/database para el sprint 4

```
CREATE SCHEMA IF NOT EXISTS Spr4;
```

-- Nos preparamos para crear las tablas y exportar los datos de los CSVs

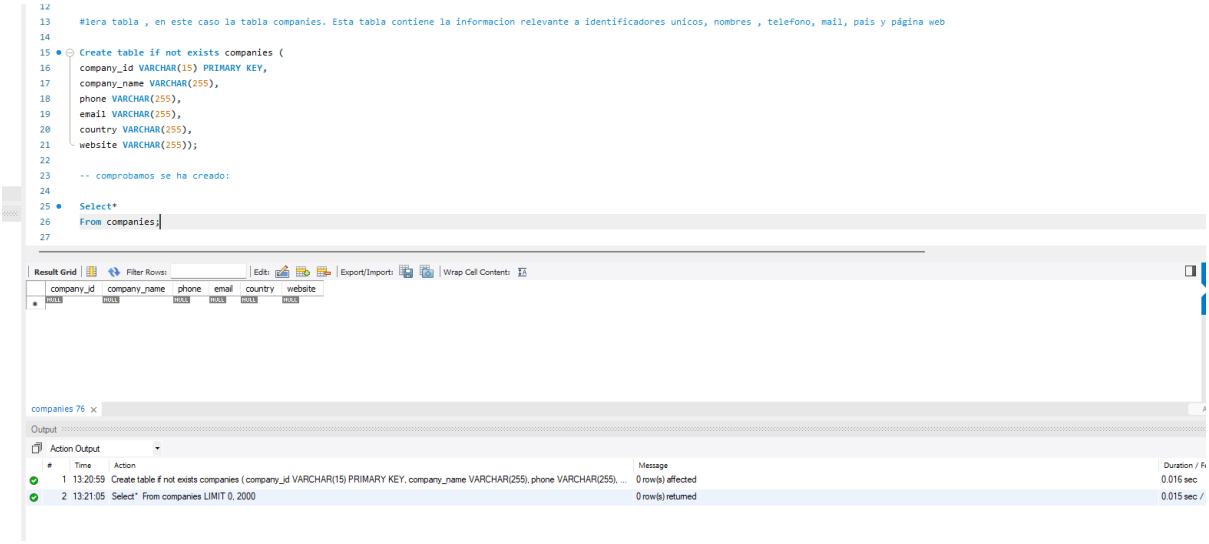
PRIMERA TABLA

En este caso la tabla companies. Esta tabla contiene la información relevante a identificadores únicos, nombres , teléfono, mail, país y página web

```
Create table if not exists companies (
company_id VARCHAR(15) PRIMARY KEY,
company_name VARCHAR(255),
phone VARCHAR(255),
email VARCHAR(255),
country VARCHAR(255),
website VARCHAR(255));
```

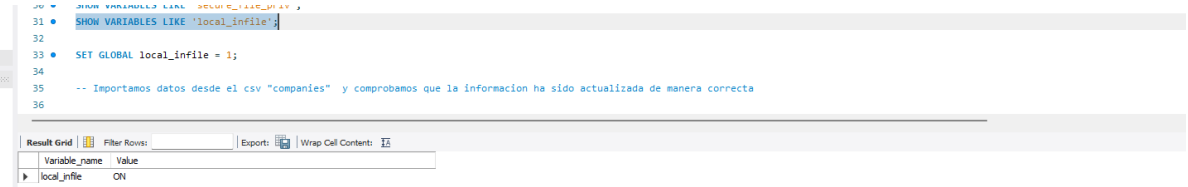
-- comprobamos se ha creado:

```
Select*
From companies;
```



-- Comprobamos el directorio de subida de datos

SHOW VARIABLES LIKE 'secure_file_priv';



SHOW VARIABLES LIKE 'local_infile';

-- Importamos datos desde el csv "companies" y comprobamos que la información ha sido actualizada de manera correcta

SET GLOBAL local_infile = 1;

Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv"

Into table spr4.companies

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

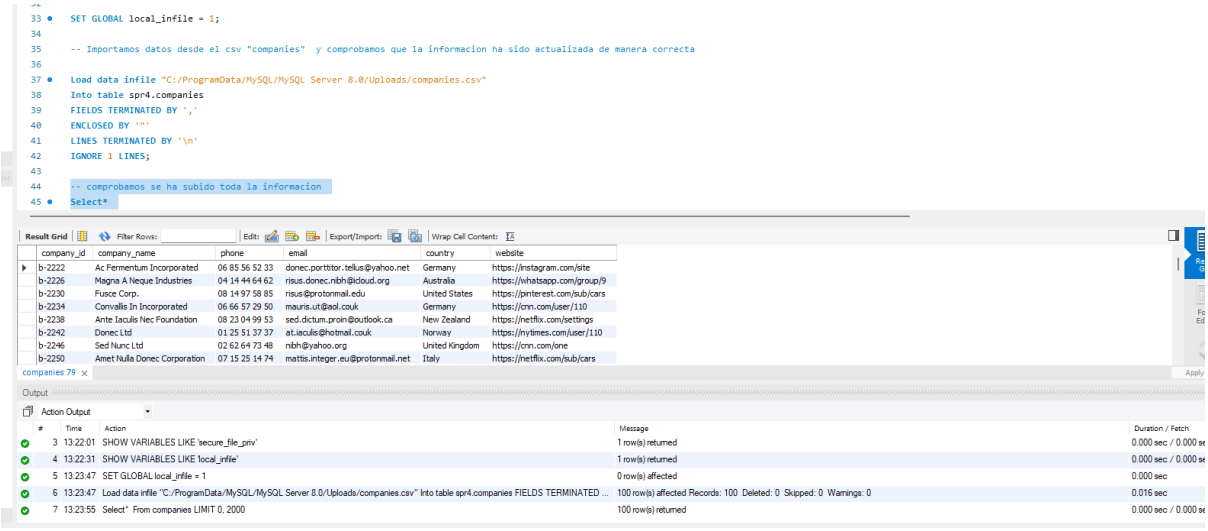
LINES TERMINATED BY '\n'

IGNORE 1 LINES;

-- comprobamos se ha subido toda la informacion

Select*

From companies;



```
SET GLOBAL local_infile = 1;

-- Importamos datos desde el csv "companies"  y comprobamos que la informacion ha sido actualizada de manera correcta

Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv"
Into table spr4.companies
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

-- comprobamos se ha subido toda la informacion
Select*
```

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

companies 79

Output

Action Output

#	Time	Action	Message	Duration / Fetch
3	13:22:01	SHOW VARIABLES LIKE 'secure_file_priv'	1 row(s) returned	0.000 sec / 0.000 se
4	13:22:31	SHOW VARIABLES LIKE 'local_infile'	1 row(s) returned	0.000 sec / 0.000 se
5	13:23:47	SET GLOBAL local_infile = 1	0 row(s) affected	0.000 sec
6	13:23:47	Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv" Into table spr4.companies FIELDS TERMINATED ...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.016 sec
7	13:23:55	Select* From companies LIMIT 0, 2000	100 row(s) returned	0.000 sec / 0.000 se

SEGUNDA TABLA

En este caso la tabla credit_cards . Esta tabla contiene información relevante respecto al identificador único de la tarjeta, identificador usuario (Este dato se podrá relacionar con otras tablas, y los datos concretos de estas tarjetas. Revisando el csv , es posible que se tengan corregir los siguientes campos:

- Datos de la columna user id, ,pan ,pin y cvv pasado a formato numerico

- Columna pan - eliminar los espacios en blanco

-La columna expiring_date tiene formato texto y formato fecha

--Creamos la tabla

Create table if not exists credit_cards (

id VARCHAR(15) primary key,

user_id VARCHAR(15),

iban VARCHAR(50),

pan VARCHAR (50),

pin VARCHAR(4),

cvv INT,

track1 VARCHAR (255),

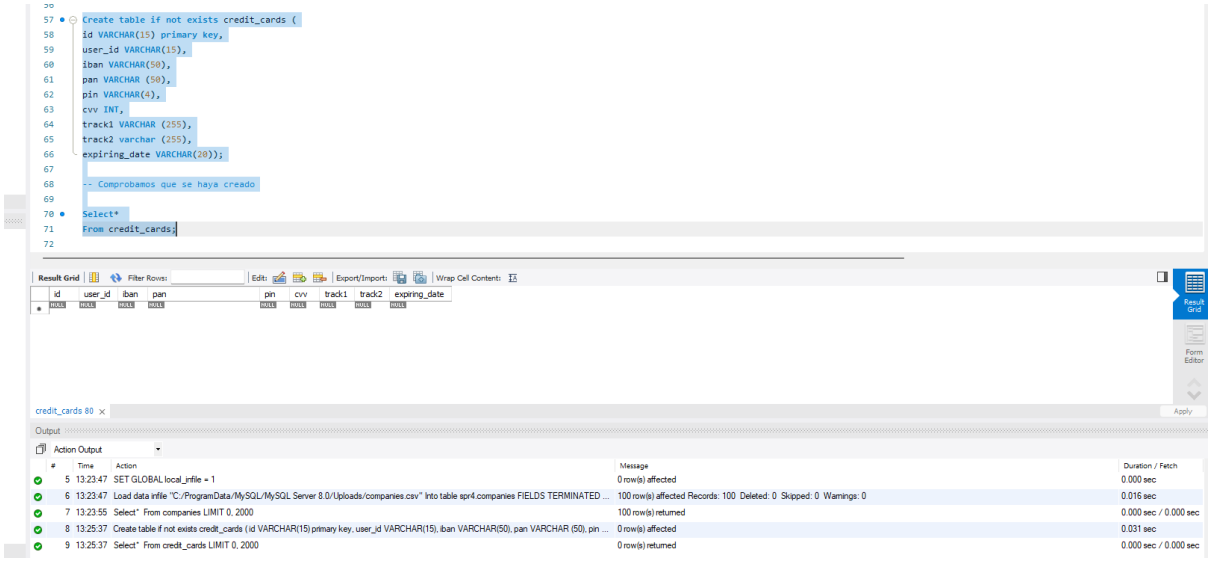
track2 varchar (255),

expiring_date VARCHAR(20));

-- y comprobamos que se haya creado

Select*

From credit_cards;



-- Importamos datos desde el csv "credit_card" y comprobamos que la información ha sido actualizada de manera correcta

SET GLOBAL local_infile = 1;

Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv"

Into table spr4.credit_cards

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

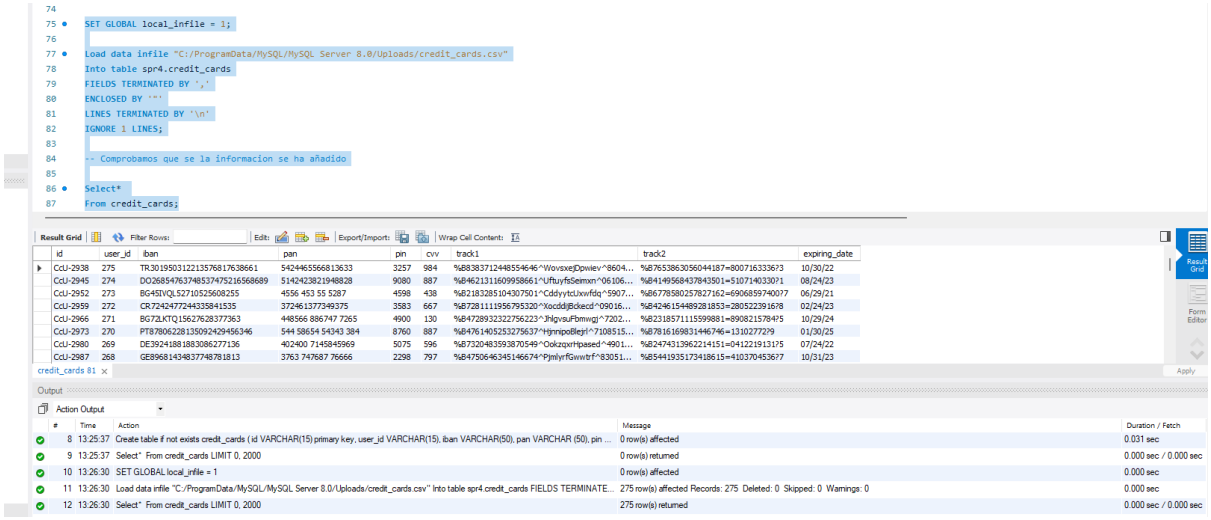
LINES TERMINATED BY '\n'

IGNORE 1 LINES;

-- Comprobamos que se la información se ha añadido

Select*

From credit_cards;



-- Realizamos modificiones en la segunda tabla - Formato numerico ID

ALTER TABLE credit_cards MODIFY COLUMN user_id INT;

-- Eliminar espacios en blanco PAN - utilizamos trim y replace para asegurarnos de que no quedan espacios blancos dentro ni al empezar ni al terminar

SET SQL_SAFE_UPDATES = 0;

UPDATE credit_cards

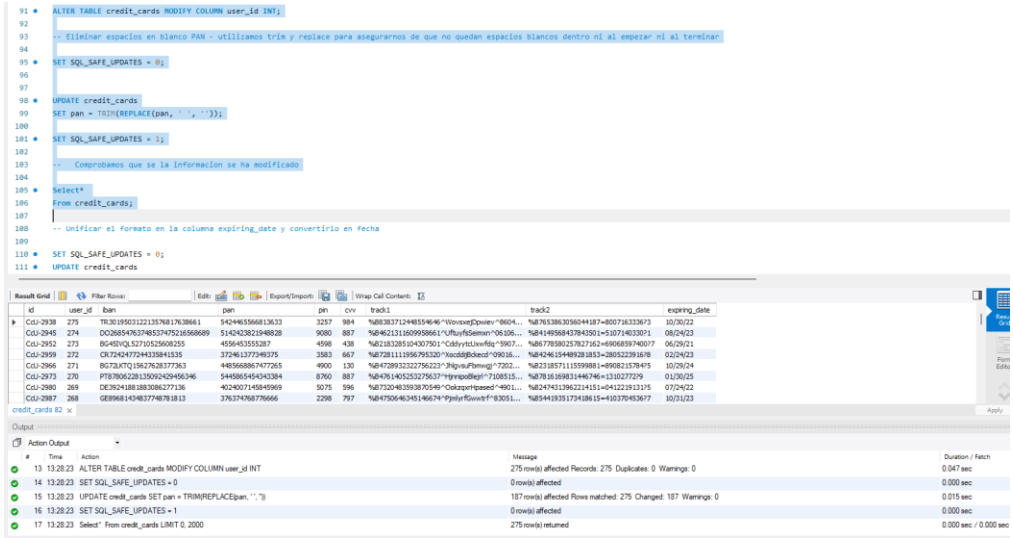
SET pan = TRIM(REPLACE(pan, ' ', ''));

SET SQL_SAFE_UPDATES = 1;

-- Comprobamos que se la información se ha modificado

Select*

From credit_cards;

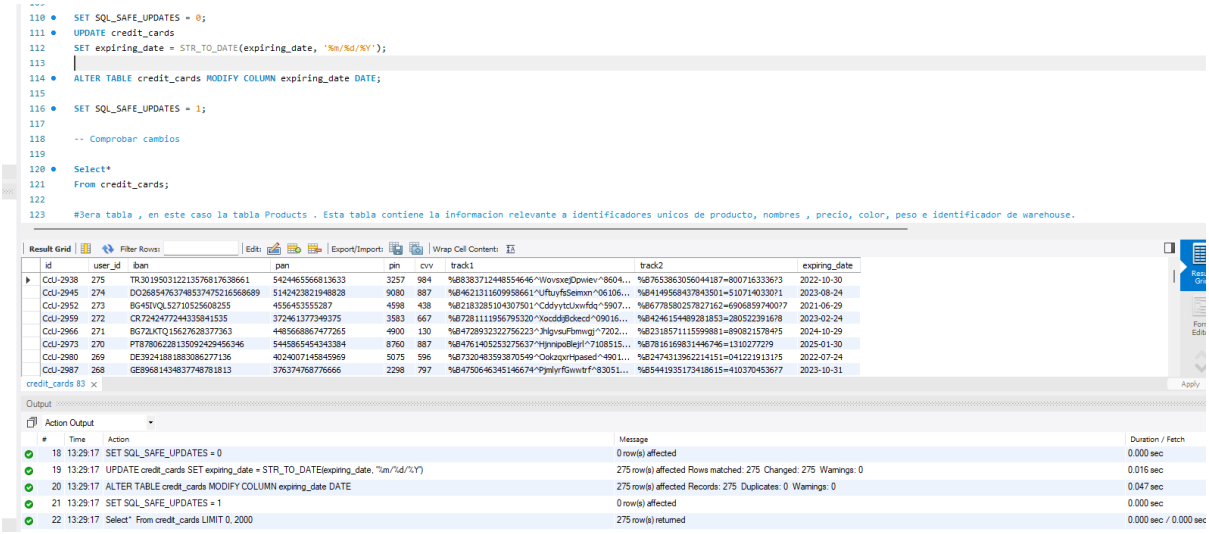


-- Unificar el formato en la columna expiring_date y convertirlo en fecha

```
SET SQL_SAFE_UPDATES = 0;

UPDATE credit_cards

SET expiring_date = STR_TO_DATE(expiring_date, '%m/%d/%Y');

ALTER TABLE credit_cards MODIFY COLUMN expiring_date DATE;

SET SQL_SAFE_UPDATES = 1;
```

-- Comprobar cambios

```
Select*

From credit_cards;
```



TERCERA TABLA

En este caso la tabla Products . Esta tabla contiene la información relevante a identificadores unicos de producto, nombres , precio, color, peso e identificador de warehouse.

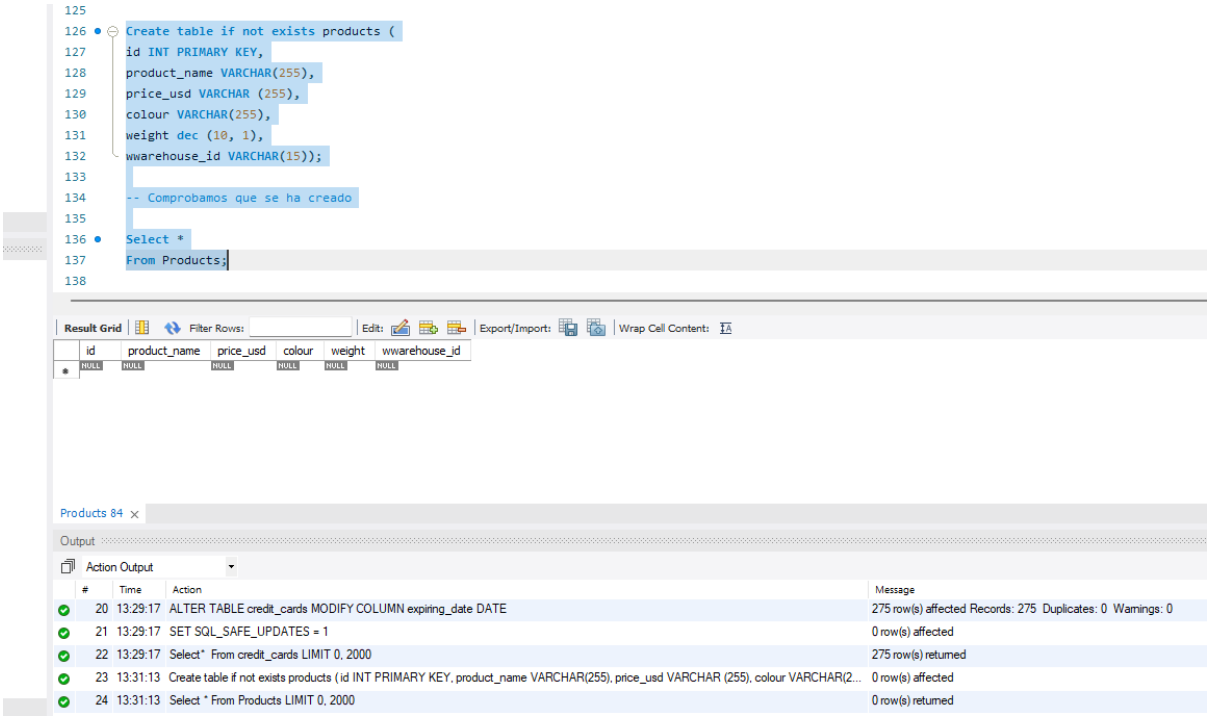
-- Revisando el archivo CSV antes de importar los datos, hemos observado que el campo precio y id está en formato texto, por lo que crearemos la columna ID como INT y luego modificaremos los datos de la columna price_usd

```
Create table if not exists products (

id INT PRIMARY KEY,

product_name VARCHAR(255),

price_usd VARCHAR (255),

colour VARCHAR(255),

weight dec (10, 1),

wwarehouse_id VARCHAR(15))
```

-- Comprobamos que se ha creado

Select *

From products;



-- Cargamos la informacion ddel csv products

SET GLOBAL local_infile = 1;

Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv"

Into table spr4.products

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

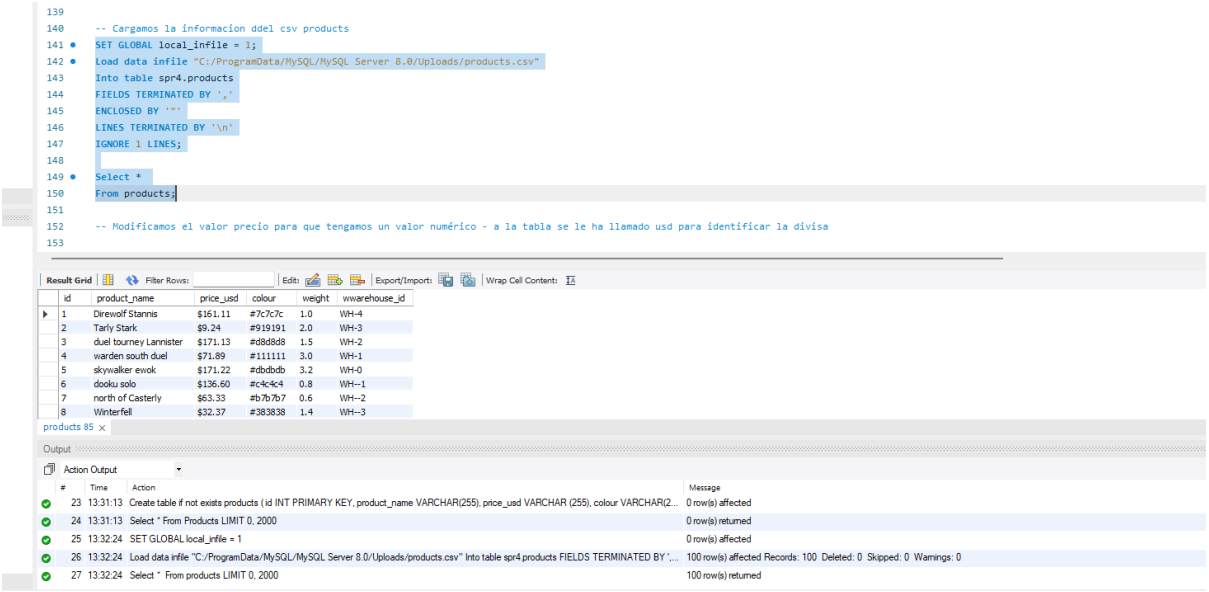
LINES TERMINATED BY '\n'

IGNORE 1 LINES;

--Comprobamos que se ha subido la informacion

Select *

From products;



```
-- Cargamos la informacion ddel csv products
SET GLOBAL local_infile = 1;
Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv"
Into table spr4.products
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

Select *
From products;

-- Modificamos el valor precio para que tengamos un valor numérico - a la tabla se le ha llamado usd para identificar la divisa
```

id	product_name	price_usd	colour	weight	wwarehouse_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 duel	$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
7	north of Casterly	$63.33	#b7b7b7	0.6	WH--2
8	Winterfell	$32.37	#383838	1.4	WH--3

products 85

Output

Action Output

#	Time	Action	Message
23	13:31:13	Create table if not exists products (id INT PRIMARY KEY, product_name VARCHAR(255), price_usd VARCHAR (255), colour VARCHAR(2...	0 row(s) affected
24	13:31:13	Select * From Products LIMIT 0, 2000	0 row(s) returned
25	13:32:24	SET GLOBAL local_infile = 1	0 row(s) affected
26	13:32:24	Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv" Into table spr4.products FIELDS TERMINATED BY ',...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
27	13:32:24	Select * From products LIMIT 0, 2000	100 row(s) returned

-- Modificamos el valor precio para que tengamos un valor numérico - a la tabla se le ha llamado usd para identificar la divisa

SET SQL_SAFE_UPDATES = 0;

UPDATE products

SET price_usd = substr(price_usd,2);

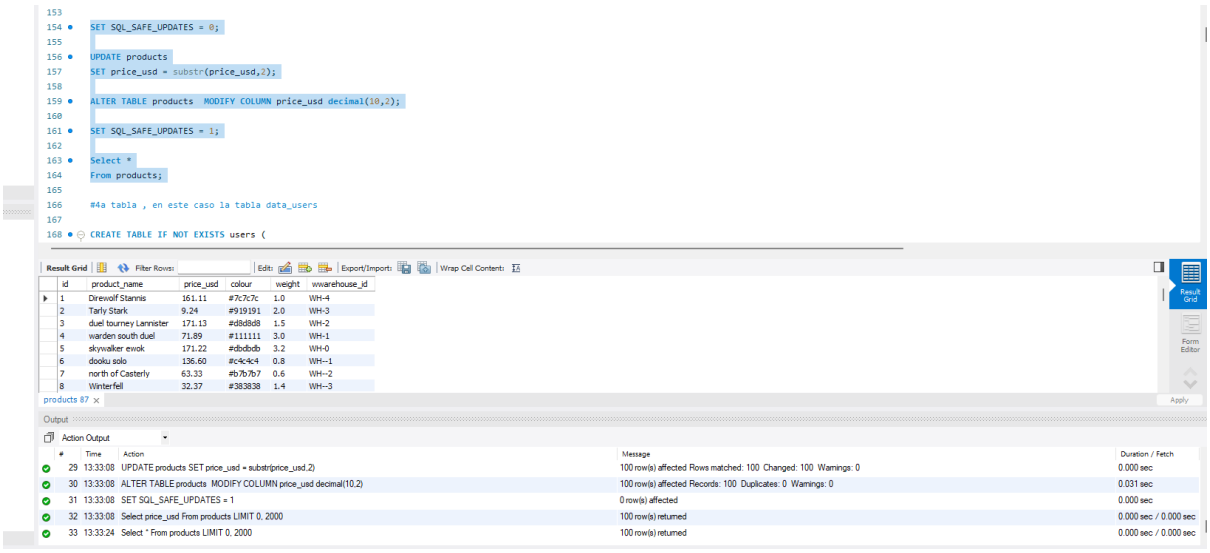
ALTER TABLE products MODIFY COLUMN price_usd decimal(10,2);

SET SQL_SAFE_UPDATES = 1;

--Comprobamos el cambio

Select price_usd

From products;



```
SET SQL_SAFE_UPDATES = 0;

UPDATE products
SET price_usd = substr(price_usd,2);

ALTER TABLE products  MODIFY COLUMN price_usd decimal(10,2);

SET SQL_SAFE_UPDATES = 1;

Select *
From products;

#4a tabla , en este caso la tabla data_users

CREATE TABLE IF NOT EXISTS users (
```

id	product_name	price_usd	colour	weight	wwarehouse_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 duel	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
7	north of Casterly	63.33	#b7b7b7	0.6	WH--2
8	Winterfell	32.37	#383838	1.4	WH--3

products 87

Output

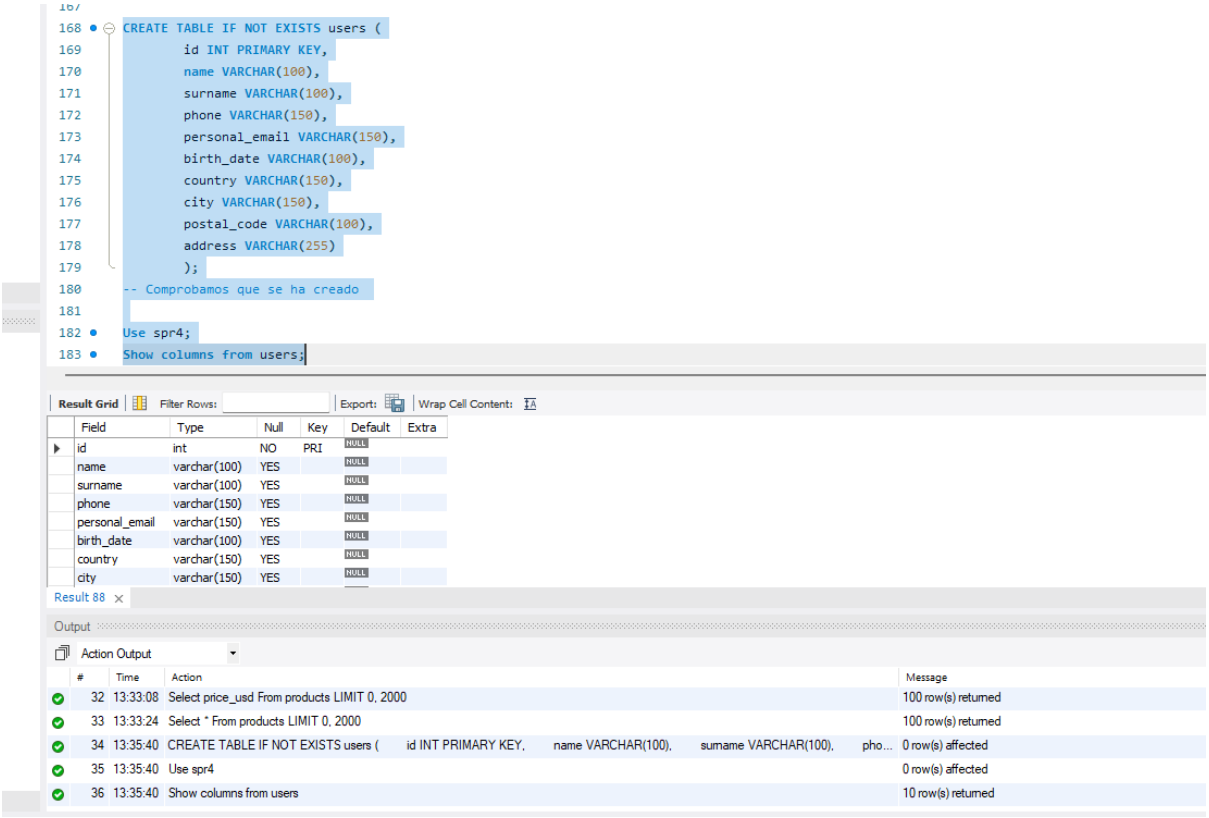
Action Output

#	Time	Action	Message	Duration / Fetch
29	13:33:08	UPDATE products SET price_usd = substr(price_usd,2)	100 row(s) affected Rows matched: 100 Changed: 100 Warnings: 0	0.000 sec
30	13:33:08	ALTER TABLE products MODIFY COLUMN price_usd decimal(10,2)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0	0.031 sec
31	13:33:08	SET SQL_SAFE_UPDATES = 1	0 row(s) affected	0.000 sec
32	13:33:08	Select price_usd From products LIMIT 0, 2000	100 row(s) returned	0.000 sec / 0.000 sec
33	13:33:24	Select * From products LIMIT 0, 2000	100 row(s) returned	0.000 sec / 0.000 sec

CUARTA TABLA

Creamos la tabla users en este caso la tabla data_user

```
CREATE TABLE IF NOT EXISTS users (
    id INT PRIMARY KEY,
    name VARCHAR(100),
    surname VARCHAR(100),
    phone VARCHAR(150),
    personal_email VARCHAR(150),
    birth_date VARCHAR(100),
    country VARCHAR(150),
    city VARCHAR(150),
    postal_code VARCHAR(100),
    address VARCHAR(255)
    );
-- Comprobamos que se ha creado

Use spr4;
Show columns from users;
```



```
CREATE TABLE IF NOT EXISTS users (
        id INT PRIMARY KEY,
        name VARCHAR(100),
        surname VARCHAR(100),
        phone VARCHAR(150),
        personal_email VARCHAR(150),
        birth_date VARCHAR(100),
        country VARCHAR(150),
        city VARCHAR(150),
        postal_code VARCHAR(100),
        address VARCHAR(255)
        );
-- Comprobamos que se ha creado

Use spr4;
Show columns from users;
```

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

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
surname	varchar(100)	YES		NULL	
phone	varchar(150)	YES		NULL	
personal_email	varchar(150)	YES		NULL	
birth_date	varchar(100)	YES		NULL	
country	varchar(150)	YES		NULL	
city	varchar(150)	YES		NULL	

Result 88

Output

Action Output

#	Time	Action	Message
32	13:33:08	Select price_usd From products LIMIT 0, 2000	100 row(s) returned
33	13:33:24	Select * From products LIMIT 0, 2000	100 row(s) returned
34	13:35:40	CREATE TABLE IF NOT EXISTS users (id INT PRIMARY KEY, name VARCHAR(100), surname VARCHAR(100), pho...	0 row(s) affected
35	13:35:40	Use spr4	0 row(s) affected
36	13:35:40	Show columns from users	10 row(s) returned

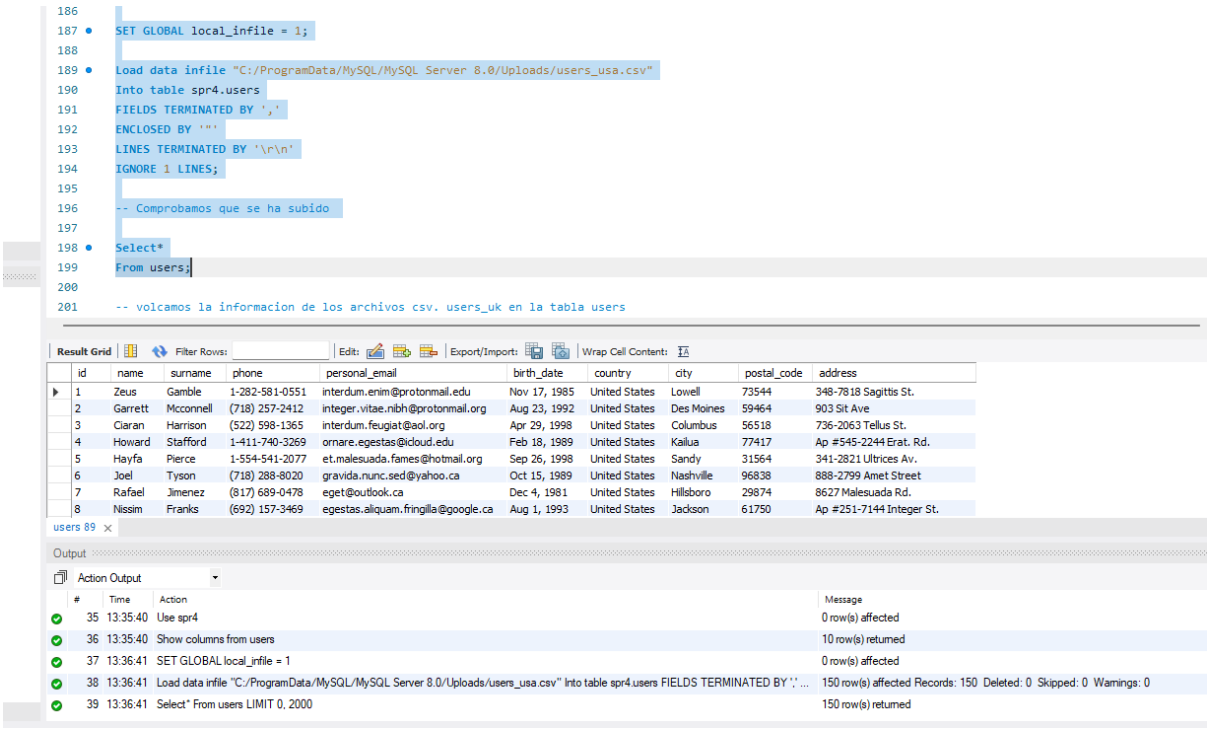
-- A continuación volcaremos toda la informacion de los 3 csv. en una tabla

-- volcamos la información de los archivos csv users_usa

```
SET GLOBAL local_infile = 1;
Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv"
Into table spr4.users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
```

-- Comprobamos que se ha subido

```
Select*
From users;
```



-- volcamos la informacion de los archivos csv. users_uk en la tabla users

```
Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv"
Into table spr4.users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
```

-- volcamos la informacion de los archivos csv. users_ca en la tabla users

```
Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv"
Into table spr4.users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
```

-- Comprobamos que se ha subido

Select*

From users;

```
Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv"
Into table spr4.users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;

-- volcamos la informacion de los archivos csv. users_ca en la tabla users

Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv"
Into table spr4.users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;

-- Comprobamos que se ha subido

Select*
From users;
```

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

id	name	surname	phone	personal_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
7	Rafael	Jimenez	(817) 689-0478	eget@outlook.ca	Dec 4, 1981	United States	Hillsboro	29874	8627 Malesuada Rd.
8	Nissim	Franks	(692) 157-3469	egestas.aliquam.fringilla@google.ca	Aug 1, 1993	United States	Jackson	61750	Ap #251-7144 Integer St.

users 90

Output

Action Output

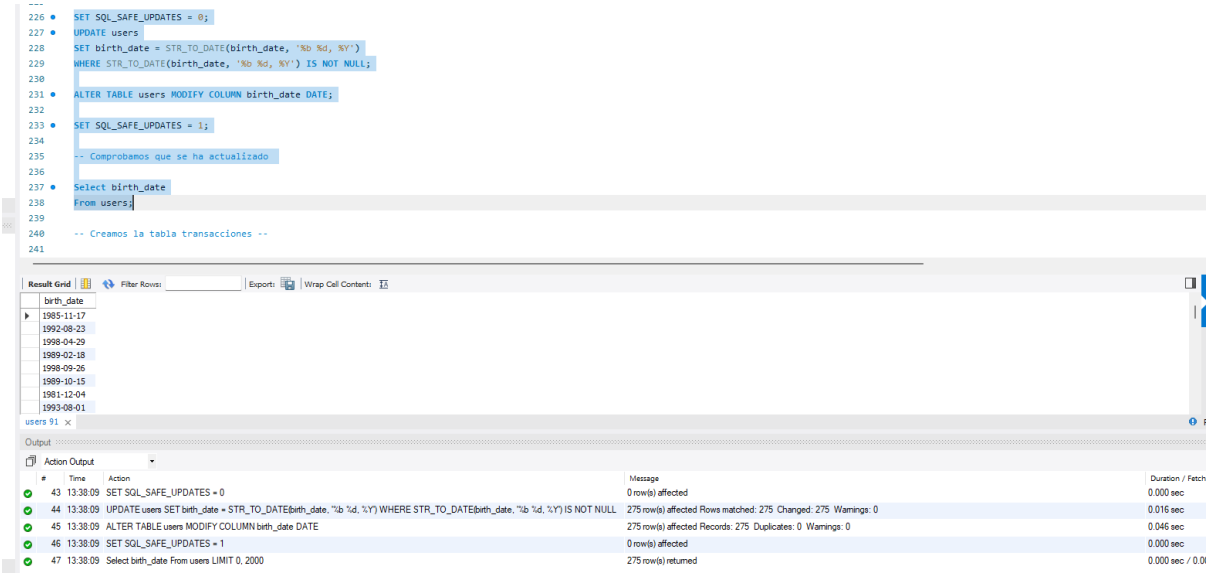
#	Time	Action	Message	Duration
38	13:36:41	Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv" Into table spr4.users FIELDS TERMINATED BY ',' ...	150 row(s) affected Records: 150 Deleted: 0 Skipped: 0 Warnings: 0	0.015 se
39	13:36:41	Select* From users LIMIT 0, 2000	150 row(s) returned	0.000 se
40	13:37:16	Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv" Into table spr4.users FIELDS TERMINATED BY ',' E...	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0	0.016 se
41	13:37:20	Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv" Into table spr4.users FIELDS TERMINATED BY ',' E...	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0	0.000 se
42	13:37:20	Select* From users LIMIT 0, 2000	275 row(s) returned	0.000 se

-- se puede observar en la tabla que la información de fecha de nacimiento no se ha subido como fecha - Realizamos el cambio

```
SET SQL_SAFE_UPDATES = 0;

UPDATE users

SET birth_date = STR_TO_DATE(birth_date, '%b %d, %Y')

WHERE STR_TO_DATE(birth_date, '%b %d, %Y') IS NOT NULL;

ALTER TABLE users MODIFY COLUMN birth_date DATE;

SET SQL_SAFE_UPDATES = 1;
```

-- Comprobamos que se ha actualizado

```
Select birth_date

From users;
```

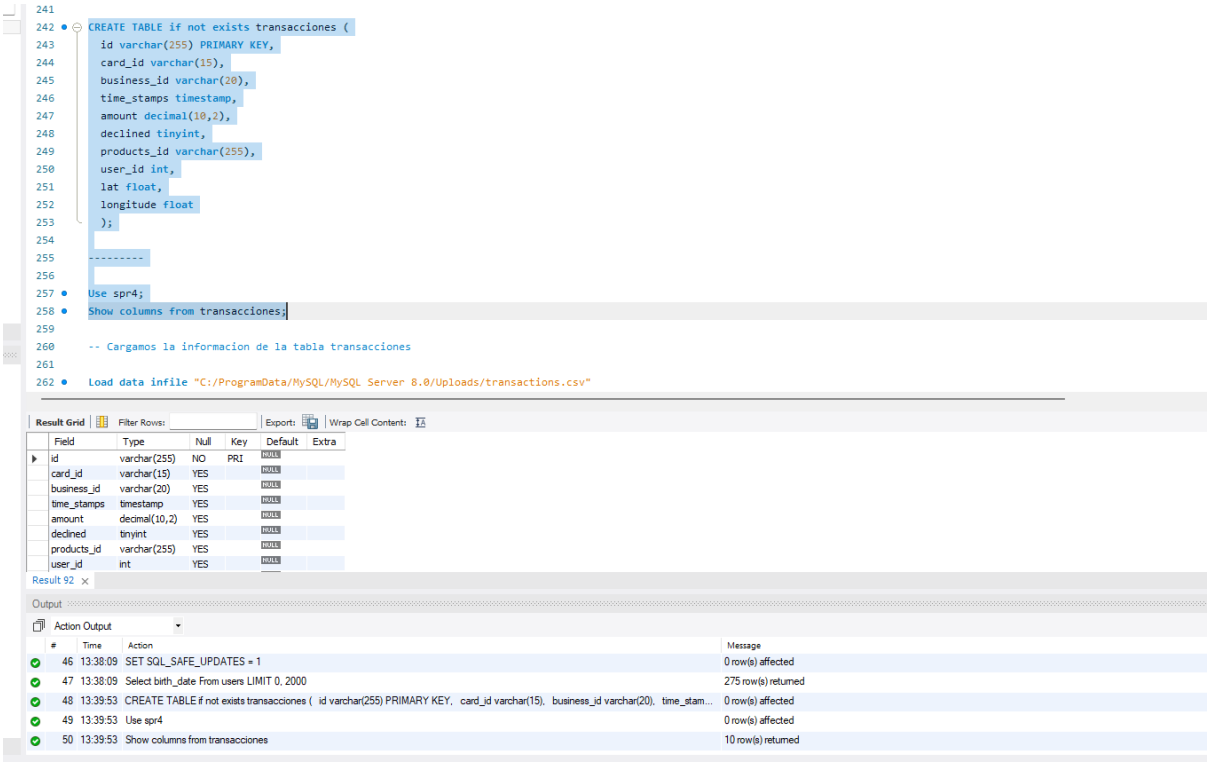


QUINTA TABLA

-- Creamos la tabla transacciones --

```
CREATE TABLE if not exists transacciones (
 id varchar(255) PRIMARY KEY,
 card_id varchar(15),
 business_id varchar(20),
 time_stamps timestamp,
 amount decimal(10,2),
 declined tinyint,
 products_id varchar(255),
 user_id int,
 lat float,
 longitude float
 );
```

```
Use spr4;
Show columns from transacciones;
```

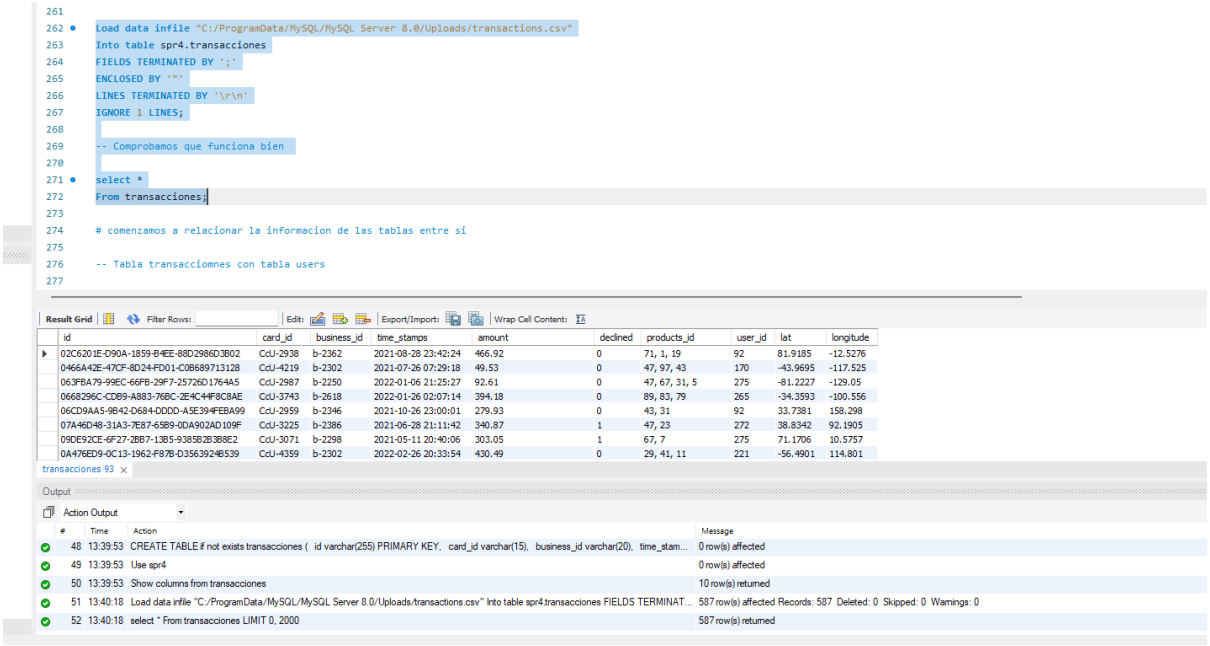


-- Cargamos la informacion de la tabla transacciones

```
Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv"
Into table spr4.transacciones
FIELDS TERMINATED BY ';'
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
```

-- Comprobamos que funciona bien

```
select *
From transacciones;
```



```
261
262 •  Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv"
263    Into table spr4.transacciones
264    FIELDS TERMINATED BY ';'
265    ENCLOSED BY '"'
266    LINES TERMINATED BY '\r\n'
267    IGNORE 1 LINES;
268
269    -- Comprobamos que funciona bien
270
271 •  select *
272    From transacciones;
273
274    # comenzamos a relacionar la informacion de las tablas entre si
275
276    -- Tabla transacciomnes con tabla users
277
```

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

id	card_id	business_id	time_stamps	amount	declined	products_id	user_id	lat	longitude
02C6201E-D90A-1859-B4EE-88D2986D3B02	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-A5E394FEBA99	CcU-2959	b-2346	2021-10-26 23:00:01	279.93	0	43, 31	92	33.7381	158.298
07A46D48-31A3-7E87-65B9-0DA902AD109F	CcU-3225	b-2386	2021-06-28 21:11:42	340.87	1	47, 23	272	38.8342	92.1905
09DE92CE-6F27-2BB7-13B5-9385B2B3B8E2	CcU-3071	b-2298	2021-05-11 20:40:06	303.05	1	67, 7	275	71.1706	10.5757
0A476ED9-0C13-1962-F87B-D3563924B539	CcU-4359	b-2302	2022-02-26 20:33:54	430.49	0	29, 41, 11	221	-56.4901	114.801

transacciones 93 ×

Output

Action Output

#	Time	Action	Message
48	13:39:53	CREATE TABLE if not exists transacciones (id varchar(255) PRIMARY KEY, card_id varchar(15), business_id varchar(20), time_stam...	0 row(s) affected
49	13:39:53	Use spr4	0 row(s) affected
50	13:39:53	Show columns from transacciones	10 row(s) returned
51	13:40:18	Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv" Into table spr4.transacciones FIELDS TERMINAT...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0
52	13:40:18	select * From transacciones LIMIT 0, 2000	587 row(s) returned

CREACION DE RELACIONES ENTRE TABLAS

comenzamos a relacionar la información de las tablas entre si

-- Tabla transacciones con tabla users

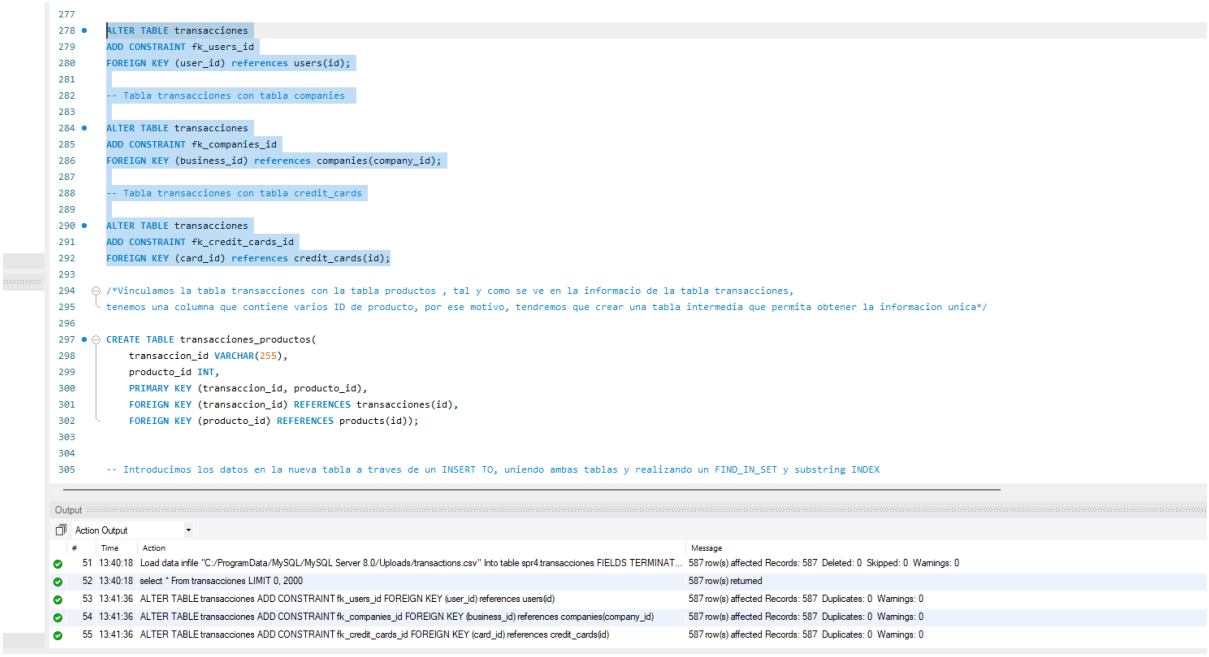
```
ALTER TABLE transacciones
ADD CONSTRAINT fk_users_id
FOREIGN KEY (user_id) references users(id);
```

-- Tabla transacciones con tabla companies

```
ALTER TABLE transacciones
ADD CONSTRAINT fk_companies_id
FOREIGN KEY (business_id) references companies(company_id);
```

-- Tabla transacciones con tabla credit_cards

```
ALTER TABLE transacciones
ADD CONSTRAINT fk_credit_cards_id
FOREIGN KEY (card_id) references credit_cards(id);
```



```
277
278 • ALTER TABLE transacciones
279   ADD CONSTRAINT fk_users_id
280   FOREIGN KEY (user_id) references users(id);
281
282   -- Tabla transacciones con tabla companies
283
284 • ALTER TABLE transacciones
285   ADD CONSTRAINT fk_companies_id
286   FOREIGN KEY (business_id) references companies(company_id);
287
288   -- Tabla transacciones con tabla credit_cards
289
290 • ALTER TABLE transacciones
291   ADD CONSTRAINT fk_credit_cards_id
292   FOREIGN KEY (card_id) references credit_cards(id);
293
294   /*Vinculamos la tabla transacciones con la tabla productos , tal y como se ve en la informacio de la tabla transacciones,
295   tenemos una columna que contiene varios ID de producto, por ese motivo, tendremos que crear una tabla intermedia que permita obtener la informacion unica*/
296
297 • CREATE TABLE transacciones_productos(
298       transaccion_id VARCHAR(255),
299       producto_id INT,
300       PRIMARY KEY (transaccion_id, producto_id),
301       FOREIGN KEY (transaccion_id) REFERENCES transacciones(id),
302       FOREIGN KEY (producto_id) REFERENCES products(id));
303
304
305    -- Introducimos los datos en la nueva tabla a traves de un INSERT TO, uniendo ambas tablas y realizando un FIND_IN_SET y substring INDEX
```

Output

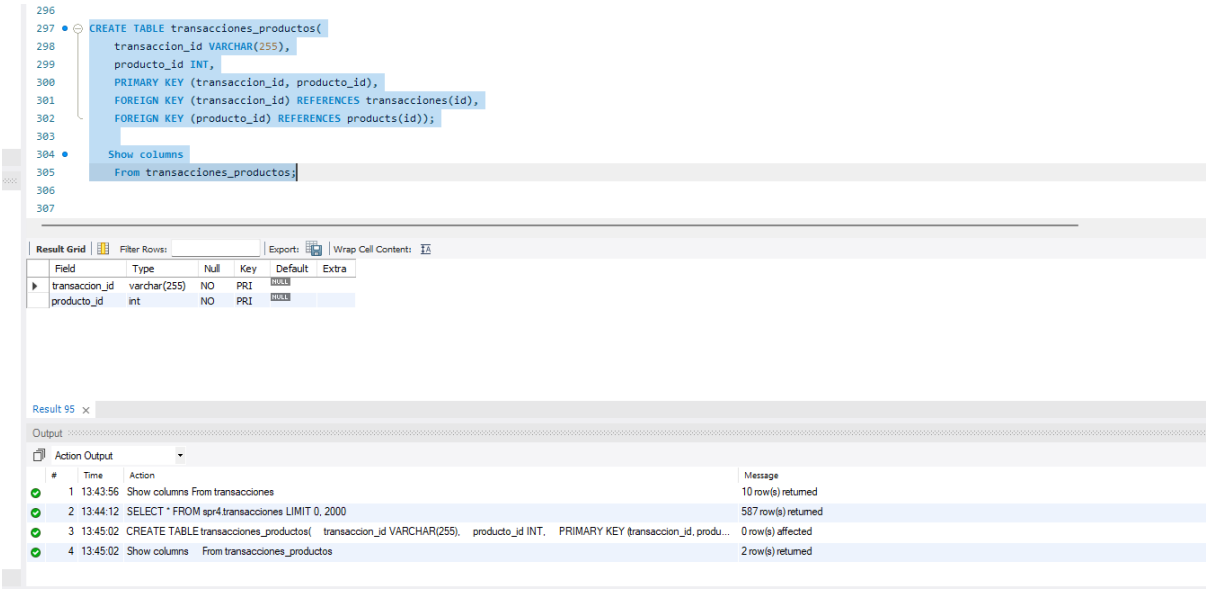
Action Output

#	Time	Action	Message
51	13:40:18	Load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv" Into table spr4.transacciones FIELDS TERMINAT...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0
52	13:40:18	select * From transacciones LIMIT 0, 2000	587 row(s) returned
53	13:41:36	ALTER TABLE transacciones ADD CONSTRAINT fk_users_id FOREIGN KEY (user_id) references users(id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
54	13:41:36	ALTER TABLE transacciones ADD CONSTRAINT fk_companies_id FOREIGN KEY (business_id) references companies(company_id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
55	13:41:36	ALTER TABLE transacciones ADD CONSTRAINT fk_credit_cards_id FOREIGN KEY (card_id) references credit_cards(id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

/*Vinculamos la tabla transacciones con la tabla productos , tal y como se ve en la información de la tabla transacciones,

tenemos una columna que contiene varios ID de producto, por ese motivo, tendremos que crear una tabla intermedia que permita obtener la informacion unica*/

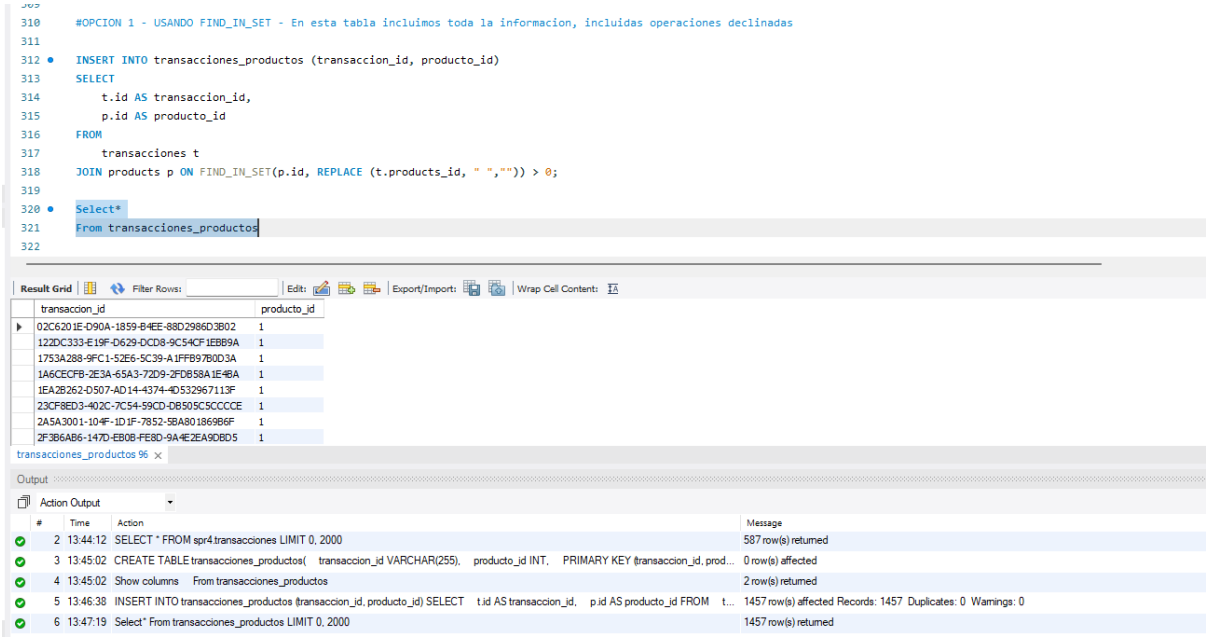
```
CREATE TABLE transacciones_productos(
    transaccion_id VARCHAR(255),
    producto_id INT,
    PRIMARY KEY (transaccion_id, producto_id),
    FOREIGN KEY (transaccion_id) REFERENCES transacciones(id),
    FOREIGN KEY (producto_id) REFERENCES products(id));
```



-- Introducimos los datos en la nueva tabla a traves de un INSERT TO, uniendo ambas tablas y realizando un FIND_IN_SET

#OPCION 1 - USANDO FIND_IN_SET - En esta tabla incluimos toda la información, incluidas operaciones declinadas

```
INSERT INTO transacciones_productos (transaccion_id, producto_id)
SELECT
  t.id AS transaccion_id,
  p.id AS producto_id
FROM
  transacciones t
JOIN products p ON FIND_IN_SET(p.id, REPLACE (t.products_id, " ",""")) > 0;
```



Hay otra OPCION 2 usando substrings - dejamos el código a continuación.

```
INSERT INTO transacciones_productos (transaccion_id, producto_id)
SELECT
  id,
  SUBSTRING_INDEX(
    SUBSTRING_INDEX(products_id, ',', 1),
    ',',
    -1
  ) AS primer_primer_id
FROM
  transacciones
WHERE
  CHAR_LENGTH(products_id) - CHAR_LENGTH(REPLACE(products_id, ',', '')) >= 0;


INSERT INTO transacciones_productos (transaccion_id, producto_id)
SELECT
  id,
  SUBSTRING_INDEX(
    SUBSTRING_INDEX(products_id, ',', 2),
```

```
    ',',
    -1
  ) AS segundo_producto_id
FROM
  transacciones
WHERE
  CHAR_LENGTH(products_id) - CHAR_LENGTH(REPLACE(products_id, ',', '')) >= 1;



INSERT INTO transacciones_productos (transaccion_id, producto_id)
SELECT
  id,
  SUBSTRING_INDEX(
    SUBSTRING_INDEX(products_id, ',', 3),
    ',',
    -1
  ) AS tercer_producto_id
FROM
  transacciones
WHERE
  CHAR_LENGTH(products_id) - CHAR_LENGTH(REPLACE(products_id, ',', '')) >= 2;

INSERT INTO transacciones_productos (transaccion_id, producto_id)
  SELECT
  id,
  SUBSTRING_INDEX(
    SUBSTRING_INDEX(products_id, ',', 4),
    ',',
    -1
  ) AS cuarto_producto_id
```

FROM

transacciones

WHERE

CHAR_LENGTH(products_id) - CHAR_LENGTH(REPLACE(products_id, ',', '')) >= 3; */

#Ahora que tenemos la base de datos creada y relacionada entre si , procederemos a realizar las consultas

NIVEL 1

Ejercicio 1.1

-- Realiza una subconsulta que muestre a todos los usuarios con más de 30 transacciones utilizando al menos 2 tablas.

-- UTILIZANDO join

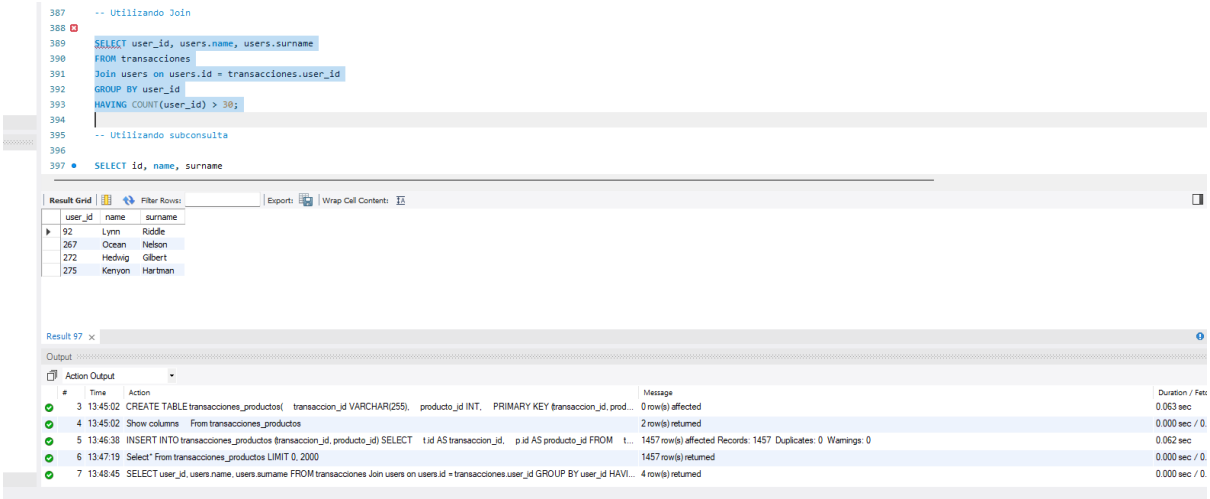
SELECT user_id, users.name, users.surname

FROM transacciones

Join users on users.id = transacciones.user_id

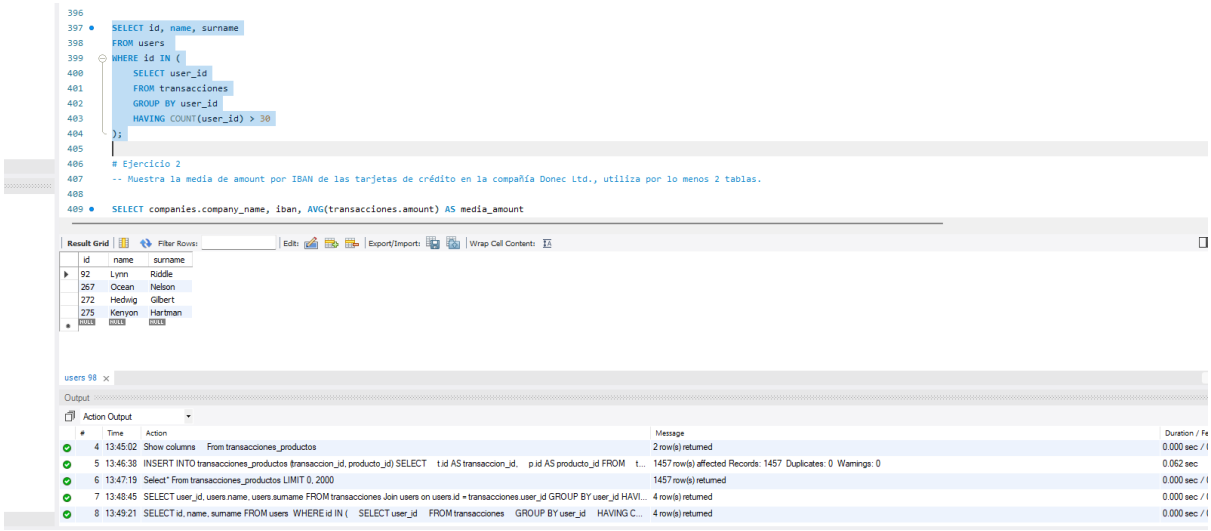
GROUP BY user_id

HAVING COUNT(user_id) > 30;



-- Utilizando subconsulta

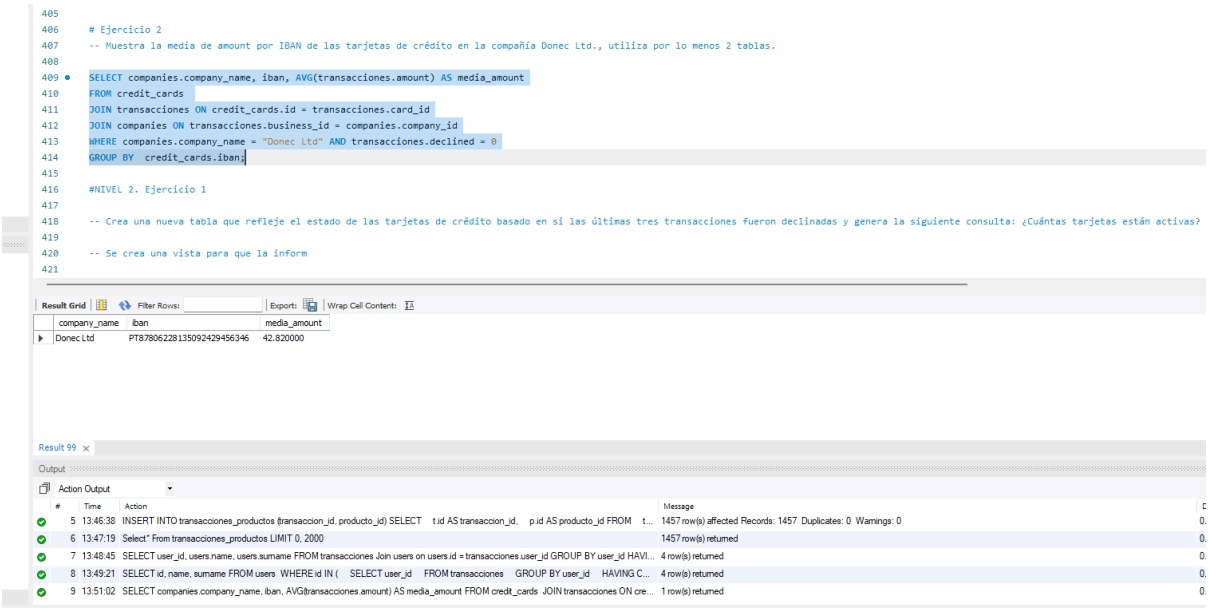
```
SELECT id, name, surname
FROM users
WHERE id IN (
    SELECT user_id
    FROM transacciones
    GROUP BY user_id
    HAVING COUNT(user_id) > 30
);
```



Ejercicio 2

-- Muestra la media de amount por IBAN de las tarjetas de crédito en la compañía Donec Ltd., utiliza por lo menos 2 tablas.

```
SELECT companies.company_name, iban, AVG(transacciones.amount) AS media_amount
FROM credit_cards
JOIN transacciones ON credit_cards.id = transacciones.card_id
JOIN companies ON transacciones.business_id = companies.company_id
WHERE companies.company_name = "Donec Ltd" AND transacciones.declined = 0
GROUP BY  credit_cards.iban;
```



```
405
406     # Ejercicio 2
407     -- Muestra la media de amount por IBAN de las tarjetas de crédito en la compañía Donec Ltd., utiliza por lo menos 2 tablas.
408
409 •   SELECT companies.company_name, iban, AVG(transacciones.amount) AS media_amount
410     FROM credit_cards
411     JOIN transacciones ON credit_cards.id = transacciones.card_id
412     JOIN companies ON transacciones.business_id = companies.company_id
413     WHERE companies.company_name = "Donec Ltd" AND transacciones.declined = 0
414     GROUP BY  credit_cards.iban;
415
416     #NIVEL 2. Ejercicio 1
417
418     -- Crea una nueva tabla que refleje el estado de las tarjetas de crédito basado en si las últimas tres transacciones fueron declinadas y genera la siguiente consulta: ¿Cuántas tarjetas están activas?
419
420     -- Se crea una vista para que la inform
421
```

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

company_name	iban	media_amount
Donec Ltd	PT87806228135092429456346	42.820000

Result 99

Output

Action Output

#	Time	Action	Message	D
5	13:46:38	INSERT INTO transacciones_productos (transaccion_id, producto_id) SELECT t.id AS transaccion_id, p.id AS producto_id FROM t...	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0	0.
6	13:47:19	Select* From transacciones_productos LIMIT 0, 2000	1457 row(s) returned	0.
7	13:48:45	SELECT user_id, users.name, users.surname FROM transacciones Join users on users.id = transacciones.user_id GROUP BY user_id HAVI...	4 row(s) returned	0.
8	13:49:21	SELECT id, name, surname FROM users WHERE id IN (SELECT user_id FROM transacciones GROUP BY user_id HAVING C...	4 row(s) returned	0.
9	13:51:02	SELECT companies.company_name, iban, AVG(transacciones.amount) AS media_amount FROM credit_cards JOIN transacciones ON cre...	1 row(s) returned	0.

#NIVEL 2.

Ejercicio 1

-- Crea una nueva tabla que refleje el estado de las tarjetas de crédito basado en si las últimas tres transacciones fueron declinadas y genera la siguiente consulta: ¿Cuántas tarjetas están activas?

```
Create table Creditcardstatus as

Select card_id, IF(sum(declined)=3, "deactivated", "active") as Status

FROM  ( Select card_id, transacciones.time_stamps, transacciones.declined, rank() over
(partition by transacciones.card_id order by time_stamps DESC) as ranking

        From transacciones

) as status_ranking

Where ranking <4

Group by card_id;
```

```
421
422 •  Create table Creditcardstatus as
423    Select card_id, IF(sum(declined)=3, "deactivated", "active") as Status
424    FROM  ( Select card_id, transacciones.time_stamps, transacciones.declined, rank() over (partition by transacciones.card_id order by time_stamps DESC) as ranking
425        From transacciones
426    ) as status_ranking
427    Where ranking <4
428    Group by card_id;
429
430
431    -- Realizamos la consulta
432
433 •  Select *
434    From Creditcardstatus
435    where status = "active";
436
437    #NIVEL 3
438
439    -- Ya hemos creado la tabla intermedia en pasos anteriores para crear nuestra base de datos - Realizamos la consulta.
440
441 •  SELECT products.id, product_name, count(producto_id) as total_ventas
```

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

card_id	Status
CcU-2938	active
CcU-2945	active
CcU-2952	active
CcU-2959	active
CcU-2966	active
CcU-2973	active
CcU-2980	active
CcU-2987	active

Creditcardstatus 101

Output

Action Output

#	Time	Action	Message
8	13:49:21	SELECT id, name, surname FROM users WHERE id IN (SELECT user_id FROM transacciones GROUP BY user_id HAVING C...	4 row(s) returned
9	13:51:02	SELECT companies.company_name, iban, AVG(transacciones.amount) AS media_amount FROM credit_cards JOIN transacciones ON cre...	1 row(s) returned
10	13:52:03	Create table Creditcardstatus as Select card_id, IF(sum(declined)=3, "deactivated", "active") as Status FROM (Select card_id, transaccion...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0
11	13:52:07	Select count(status) From Creditcardstatus where status = "active" LIMIT 0, 2000	1 row(s) returned
12	13:52:43	Select * From Creditcardstatus where status = "active" LIMIT 0, 2000	275 row(s) returned

-- Realizamos la consulta

Select count(status)

From Creditcardstatus

where status = "active";

```
430
431    -- Realizamos la consulta
432
433 •  Select count(status)
434    From Creditcardstatus
435    where status = "active";
```

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

count(status)
275

Result 102

Output

Action Output

#	Time	Action	Message
9	13:51:02	SELECT companies.company_name, iban, AVG(transacciones.amount) AS media_amount FROM credit_cards JOIN transacciones ON cre...	1 row(s) returned
10	13:52:03	Create table Creditcardstatus as Select card_id, IF(sum(declined)=3, "deactivated", "active") as Status FROM (Select card_id, transaccion...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0
11	13:52:07	Select count(status) From Creditcardstatus where status = "active" LIMIT 0, 2000	1 row(s) returned
12	13:52:43	Select * From Creditcardstatus where status = "active" LIMIT 0, 2000	275 row(s) returned
13	13:53:18	Select count(status) From Creditcardstatus where status = "active" LIMIT 0, 2000	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:

-- Ya hemos creado la tabla intermedia en pasos anteriores para crear nuestra base de datos - tabla transacciones_productos

Ejercicio 1

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

--Realizamos la consulta.

```
SELECT products.id, product_name, count(producto_id) as total_ventas
From transacciones_productos
Join products on transacciones_productos.producto_id = products.id
Group by producto_id
ORDER BY total_ventas DESC;
```