

## SPRINT 4

### Modelado SQL creación de BASES DE DATOS

#### DESCRIPCION

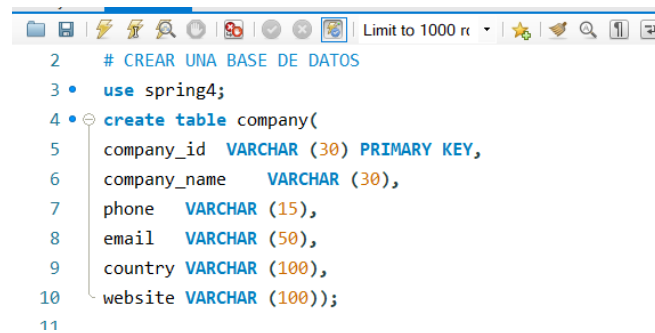
#### NIVEL 1

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 las siguientes consultas:

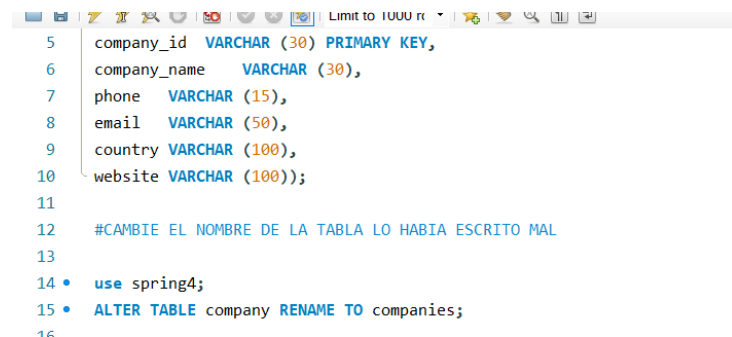
#### BASE DE DATOS SPRING4

#### Creando la tabla COMPANIES



```
2  # CREAMOS UNA BASE DE DATOS
3  • use spring4;
4  • create table company(
5      company_id VARCHAR (30) PRIMARY KEY,
6      company_name VARCHAR (30),
7      phone VARCHAR (15),
8      email VARCHAR (50),
9      country VARCHAR (100),
10     website VARCHAR (100));
11
```

El nombre lo coloque mal hice un rename



```
5  company_id VARCHAR (30) PRIMARY KEY,
6  company_name VARCHAR (30),
7  phone VARCHAR (15),
8  email VARCHAR (50),
9  country VARCHAR (100),
10 website VARCHAR (100));
11
12 #CAMBIE EL NOMBRE DE LA TABLA LO HABIA ESCRITO MAL
13
14 • use spring4;
15 • ALTER TABLE company RENAME TO companies;
16
```

Al introducir la información tuve problemas con el campo **company\_name** ; ya que en la fila 31 del CSV el tamaño era demasiado largo para el tamaño definido de esa columna , así que realice cambio.

Para introducir la data tuve que realizar varios cambios en la configuración de mysql para que lograra subir la informacion del link correspondiente.

## SPRINT 4

```
20 • LOAD DATA INFILE 'C:\\ProgramData\\test\\Uploads\\companies.csv'
21 INTO TABLE companies
22 FIELDS TERMINATED BY ','
23 ENCLOSED BY '"'
24 LINES TERMINATED BY '\n'
25 IGNORE 1 LINES;
26
27 • ALTER TABLE companies MODIFY COLUMN company_name VARCHAR(100);
28 • select *
```

company_id	company_name	phone	email	country	website
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porrtitor.telus@yahoo.net	Germany	https://instagram.com/site
b-2226	Magna A Neque Industries	04 14 44 64 62	rsus.donec.nibh@icloud.org	Australia	https://whatsapp.com/group/9
b-2230	Fusce Corp.	08 14 97 58 85	rsus@protonmail.edu	United States	https://pinterest.com/sub/cars
b-2234	Convello In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany	https://cnn.com/user/110
b-2238	Anite Jaculis 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.jaculis@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 1 x

Output

Continuo con la creación de la **tabla credi\_card**, me guie de la base de datos anterior para lo tamaños.

```
30
31 • create table credit_card(
32   id varchar(20),
33   user_id varchar(100),
34   iban varchar(50),
35   pan varchar(100),
36   pin varchar(100),
37   cvv varchar(100),
38   track1 varchar(100),
39   track2 varchar(100),
40   expiring_date varchar(100));
41
42
43 • ALTER TABLE credit_card RENAME TO credit_cards;
44
45
```

Le agregue su clave primaria

```
56 # agregue su clave primaria
57 • ALTER TABLE credit_cards
58   ADD PRIMARY KEY (id);
59
```

Output

#	Time	Action	Message
1	11:30:41	ALTER TABLE credit_cards ADD PRIMARY KEY (id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Realizo el mismo procedimiento

## SPRINT 4

```
48
49 • LOAD DATA INFILE 'C:\ProgramData\test\uploads\credit_cards.csv'
50 INTO TABLE credit_cards
51 FIELDS TERMINATED BY ','
52 ENCLOSED BY '"'
53 LINES TERMINATED BY '\n'
54 IGNORE 1 LINES;
55
56
57 • select *
58 from spring4.credit_cards;
```

id	user_id	ban	pan	pin	cvv	track1	track2	expiring_date
CCU-2938	275	TR301950312213576817638661	5424465566813633	3257	984	%08383712448554646~WovvxeD...	%07653863056044187~80071633...	10/30/22
CCU-2945	274	D026854763748537475216568689	5142423821948828	9080	887	%084621311609958661~UbuYvfe...	%04149568437843501~51071403...	08/24/23
CCU-2952	273	06450VQL52710525608255	4556 453 55 5287	4598	438	%02181285104307501~Codyvte...	%06778080257827162~69068897...	06/29/21
CCU-2959	272	C87242477244335841535	372461377348375	3583	467	%07281111956795320~Vocud88...	%04246154480281853~28652239...	02/24/23
CCU-2966	271	0C724KTJ21562762377363	448566 886747 7265	4900	130	%04728932322756223~Jhqvusf...	%02318571115599881~89082157...	10/29/24
CCU-2973	270	PT87806228135092424956346	544 58654 54343 384	8760	887	%04761405253275637~Hhnpq08...	%07816169831446746~1310277279	01/30/25
CCU-2980	269	DC32041881863886277136	402400 7145845969	5075	596	%0720482553875649~Ockezp...	%02474313962214151~04122191...	07/24/22
CCU-2987	268	GE39681434837748781813	3763 747687 76666	2298	797	%047506464345146674~PmlyvGw...	%05441935172418615~41037045...	10/31/23

credit\_cards 2 x

Importe las tablas de usuarios **USERS\_CA**, **USERS\_USA**, **USERS\_UK** y realice una nueva tabla que la llame **USERS**, y realice una unión esto lo hice debido a que las tres tablas tienen la misma información.

```
60 #creando tabla USERS
61 • use spring4;
62 • create table users (
63   id VARCHAR (100) PRIMARY KEY,
64   name varchar (100),
65   surname varchar (100),
66   phone varchar (100),
67   email varchar (100),
68   birth_date varchar (100),
69   country varchar (100),
70   city varchar (100),
71   postal_code varchar (100),
72   address varchar (100));
73
74
75
```

Realizo la **UNION**

```
Query 1 | Users | Users2 | TAPEA1*
75 -- Insertar datos combinados de user1, user2, y user3 en users
76
77 • INSERT INTO users (id,name, surname,phone,email,birth_date,country,city,postal_code,address)
78   SELECT id,name, surname,phone,email,birth_date,country,city,postal_code,address FROM users_ca
79   UNION ALL
80   SELECT id,name, surname,phone,email,birth_date,country,city,postal_code,address FROM users_uk
81   UNION ALL
82   SELECT id,name, surname,phone,email,birth_date,country,city,postal_code,address from users_usa;
83
84 • select *
85   from users;
86
```

id	name	surname	phone	email	birth_date	country	city	postal_code	address
177	Leandra	Carpenter	07624 223711	nunc@poodo.net	Aug 20, 2000	United Kingd...	Keeth	XASV 0RZ	Ap #681-4983 Nec. Av.
178	Guine...	Kemp	070 1590 7780	suacp4@poodo.net	Jun 18, 1987	United Kingd...	Kinnaske	X08 15H	Ap #801-3785 Lacus. St.
179	Stuart	Smal	0169771 2504	sodasw4e.enaticloud.net	Apr 21, 1983	United Kingd...	Beangotto...	AC98 35E	P.O. Box 484, 610 Et Ave
18	Russell	Jimenez	(657) 779-2438	orc@outbook.edu	Aug 26, 1993	United States	Hattiesburg	75647	4095 Quam Rd.
180	Amelia	Valenzuela	(016977) 9972	ac.nuka@poodo.co.uk	Feb 27, 1986	United Kingd...	Stockport	BK0M 3FO	2729 Feugiat. Road
181	Kenneth	O'Brien	076 8212 1580	curius@poodo.com	Jan 15, 1995	United Kingd...	St. Andre...	DS 2TG	Ap #659-5780 Ede. Rd.
182	Dane	Shepard	0975 985 5842	vehicla@hotmail.net	Oct 13, 1999	United Kingd...	Sekrah	TBO 70U	P.O. Box 148, 5146 Paccrat. ...
183	Germa...	Whitehead	(016977) 7528	malesuada.malesuada.integer...	Oct 20, 1982	United Kingd...	Alva	GCOY 2EW	895-5715 Ipsum. Road

users 1 x

Crear la tabla **TRANSACTIONS**

## SPRINT 4

```
87
88
89 # creando tabla transaccion id card_id business_id timestamp amount declined product_ids user_id lat longitude
90 • create table transactions (
91   id varchar(100),
92   card_id varchar(100),
93   business_id varchar(100),
94   timestamp varchar(100),
95   amount varchar(100),
96   declined varchar(100),
97   product_ids varchar(100),
98   user_id varchar(100),
99   lat varchar(100),
100  longitude varchar(100));
101
```

Se introdujeron los datos en la tabla

```
91 | id varchar(100),
92 | card_id varchar(100),
93 | business_id varchar(100),
94 | timestamp varchar(100),
95 | amount varchar(100),
96 | declined varchar(100),
97 | product_ids varchar(100),
98 | user_id varchar(100),
99 | lat varchar(100),
100 | longitude varchar(100));
101
102 • LOAD DATA INFILE 'C:\\ProgramData\\test\\uploads\\transactions.csv'
103 INTO TABLE transactions
104 FIELDS TERMINATED BY ';' -- En este caso los campos estan separados por ";" y no por ","
105 ENCLOSED BY '"'
106 LINES TERMINATED BY '\n\r'
107 IGNORE 1 ROWS

```

Output

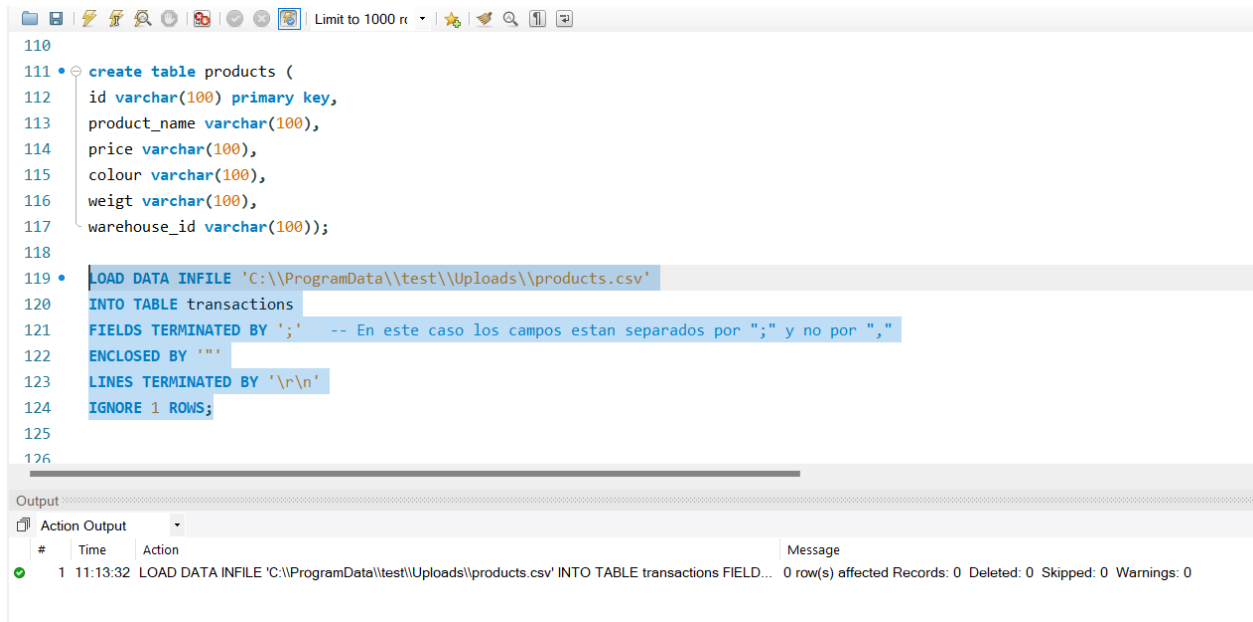
#	Time	Action	Message
1	10:45:19	LOAD DATA INFILE 'C:\\ProgramData\\test\\uploads\\transactions.csv' INTO TABLE transactions FI...	507 row(s) affected Records: 507 Deleted: 0 Skipped: 0 Warnings: 0

Le agregue su clave primaria, ya que no lo había hecho al crear la tabla

```
111 • DROP TABLE users_ca;
112   # agregue su clave primaria
113 • ALTER TABLE transactions
114   ADD PRIMARY KEY (id);
115
116   # debo realizar la conexcion entre las tablas
117 • ALTER TABLE transactions
```

Crear la tabla **PRODUCTS**

## SPRINT 4



```
110
111 • create table products (
112     id varchar(100) primary key,
113     product_name varchar(100),
114     price varchar(100),
115     colour varchar(100),
116     weight varchar(100),
117     warehouse_id varchar(100));
118
119 • LOAD DATA INFILE 'C:\\ProgramData\\test\\Uploads\\products.csv'
120 INTO TABLE transactions
121 FIELDS TERMINATED BY ';' -- En este caso los campos estan separados por ";" y no por ","
122 ENCLOSED BY '"'
123 LINES TERMINATED BY '\\r\\n'
124 IGNORE 1 ROWS;
125
126
```

Output

#	Time	Action	Message
1	11:13:32	LOAD DATA INFILE 'C:\\ProgramData\\test\\Uploads\\products.csv' INTO TABLE transactions FIELD...	0 row(s) affected Records: 0 Deleted: 0 Skipped: 0 Warnings: 0

Realizo la conexión de las tablas

```
# debo realizar la conexion entre las tablas
• alter table transactions
  add constraint fk_credit_cards
  foreign key (card_id)
  references credit_cards(id);

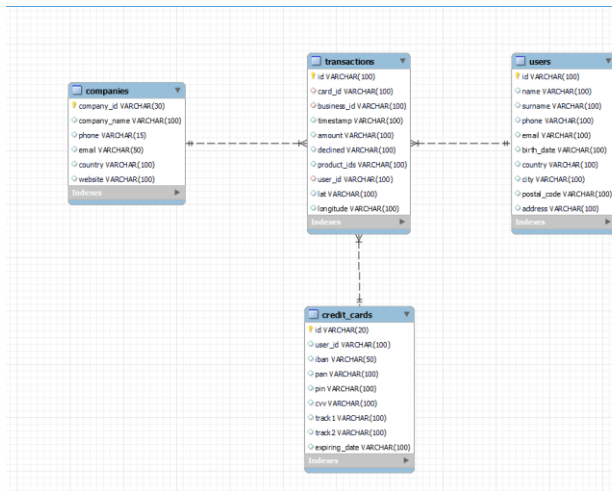
• alter table transactions
  add constraint fk_users
  foreign key (user_id)
  references users(id);

• alter table transactions
  add constraint fk_companies
  foreign key (business_id)
  references companies(company_id);
```

tion Output

#	Time	Action	Message
1	11:56:28	alter table transactions add constraint fk_companies foreign key (business_id) references compani...	Error Code: 3734. Failed to add the foreign key constraint. Missing
2	11:59:14	alter table transactions add constraint fk_companies foreign key (business_id) references compani...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
3	11:59:36	alter table transactions add constraint fk_users foreign key (user_id) references users(id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
4	11:59:42	alter table transactions add constraint fk_credit_cards foreign key (card_id) references credit_cards...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

## SPRINT 4



### - Ejercicio 1

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

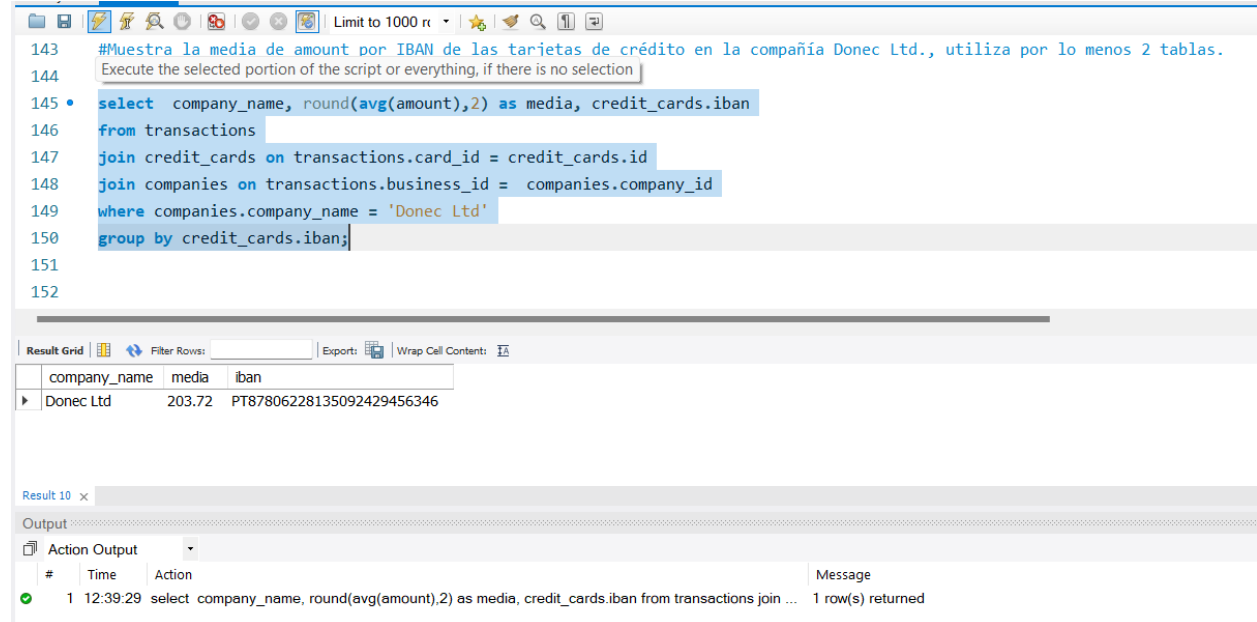
```
132
133 #EJERCICIO 1
134 #MOSTRAR TODOS LOS USUARIOS CON MAS DE 30 TRANSACCIONES
135 • SELECT users.id, users.name, users.surname, count(transactions.id) as cont
136 FROM USERS
137 join transactions on transactions.user_id= users.id
138 group by users.id, users.name, users.surname
139 having cont > 30;
140
141
```

	id	name	surname	cont
▶	267	Ocean	Nelson	52
	272	Hedwig	Gilbert	76
	275	Kenyon	Hartman	48
	92	Lynn	Riddle	39

### - Ejercicio 2

## SPRINT 4

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



The screenshot shows a SQL IDE interface. The top pane contains a SQL query with line numbers 143 to 152. The query is as follows:

```
143 #Muestra la media de amount por IBAN de las tarjetas de crédito en la compañía Donec Ltd., utiliza por lo menos 2 tablas.
144 Execute the selected portion of the script or everything, if there is no selection
145 • select company_name, round(avg(amount),2) as media, credit_cards.iban
146 from transactions
147 join credit_cards on transactions.card_id = credit_cards.id
148 join companies on transactions.business_id = companies.company_id
149 where companies.company_name = 'Donec Ltd'
150 group by credit_cards.iban;
151
152
```

The bottom pane shows the 'Result Grid' with the following data:

company_name	media	iban
Donec Ltd	203.72	PT87806228135092429456346

Below the result grid, the 'Output' pane shows the 'Action Output' for the query, indicating that 1 row(s) were returned.

## NIVEL 2

Se procede a realizar la creación de la tabla, con la estructura y los datos que se definirán en la consulta SELECT, el campo card\_id es el identificador de las tarjetas, lo uso para realizar un CASE y así definir el estado de la tarjeta, cuento la filas por transacción y así valido la cantidad de transacciones por tarjeta si tiene menos de 3 se considera activa, el contador para sumar las transacciones q estén declined como las que no estén y las compara con el numero 3, si se cumple la tres transacciones recientes fueron declined entonces es inactiva.

En la subconsulta, se coloca el identificador de tarjetas (card\_id) y el (declined) para saber si esta declined o no,

El ROW\_NUMBER me asignan número de fila a cada transacción dentro de cada grupo de card\_id.

OVER(PARTITION BY card\_id ORDER BY timestamp DESC este comando me divide las filas en particiones por card\_id y las ordena por timestamp en orden descendente.

AS Rank\_Transaccion: este comando asigna el resultado de ROW\_NUMBER() a la columna Rank\_Transaccion, indicando el orden de las transacciones más recientes.

Para luego filtrar las transacciones mas recientes

## SPRINT 4

```
184 SELECT
185     card_id,
186     CASE
187         WHEN COUNT(*) < 3 THEN 'Activa'
188         WHEN SUM(declined) = 3 THEN 'Inactiva'
189         ELSE 'Activa'
190     END AS status
191 FROM (
192     SELECT
193         card_id,
194         declined,
195         ROW_NUMBER() OVER(PARTITION BY card_id ORDER BY timestamp DESC) AS Rank_Transaccion
196     FROM transactions
197 ) AS Transacciones_Ordenadas
198 WHERE Rank_Transaccion <= 3
199 GROUP BY card_id;
```

Output:

#	Time	Action	Message
1	11:31:33	CREATE TABLE card_status AS SELECT card_id, CASE WHEN COUNT(*) < 3 THEN 'A...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

Resumen del proceso completo

1. La subconsulta ordena todas las transacciones por tarjeta.
2. Se asigna un número de fila a cada transacción para identificar las tres más recientes.
3. Se filtran las transacciones para mantener solo las tres más recientes por tarjeta.
4. Se agrupan las transacciones por tarjeta.

## Ejercicio 1

¿Cuántas tarjetas están activas?



## SPRINT 4

```
199 GROUP BY card_id;
200
201 #Ejercicio 1
202 #¿Cuántas tarjetas están activas?
203 • SELECT * FROM card_status;
204
205 • SELECT count(card_id)
206 FROM card_status
207 WHERE status = "Activa";
```

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

count(card_id)
275

Result 36 x

Output

Action Output

#	Time	Action	Message
1	12:42:19	SELECT count(card_id) FROM card_status WHERE status = "Activa" LIMIT 0, 1000	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

```
208
209 #NIVEL 3
210 #Crea una tabla con la que podamos unir los datos del nuevo archivo products.csv con la base de datos creada,
211 # teniendo en cuenta que desde transaction tienes product_ids. Genera la siguiente consulta:
212
213 • CREATE TABLE inter_products
214 select id, product_ids
215 from transactions;
216
217
```

Output

Action Output

#	Time	Action	Message
1	11:13:11	CREATE TABLE inter_products select id, product_ids from transactions	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

Visualizo la tabla

## SPRINT 4

```
213 • CREATE TABLE inter_products
214     select id, product_ids
215     from transactions;
216
217
218
```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	id	product_ids			
▶	02C6201E-D90A-1859-B4EE-88D29...	71, 1, 19			
	0466A42E-47CF-8D24-FD01-C0B68...	47, 97, 43			
	063FBA79-99EC-66FB-29F7-25726D...	47, 67, 31, 5			
	0668296C-CDB9-A883-76BC-2E4C4...	89, 83, 79			
	06CD9AA5-9B42-D684-DDDD-A5E3...	43, 31			
	07A46D48-31A3-7E87-65B9-0DA90...	47, 23			
	09DE92CE-6F27-2BB7-13B5-9385B...	67, 7			
	0A476ED9-0C13-1962-F87B-D3563...	29, 41, 11			

transactions 2 x

Para lograr desglosar los `product_ids` se realizó una subconsulta (`numbers`) que genera en este caso los números del 1 al 5. Esto nos ayudará a desglosar hasta 5 números separados por comas en cada fila de **PRODUCT\_ID**.

Utilizamos la función **SUBSTRING\_INDEX** para desglosar la cadena de **PRODUCT\_ID** por comas. La función **SUBSTRING\_INDEX** devuelve una parte específica de una cadena hasta un delimitador especificado.

Aquí, `inter_products.product_id` es la columna que contiene los números separados por comas que queremos desglosar. `numbers.n` se refiere a los números generados en nuestra subconsulta, lo que nos permite acceder a cada número individual dentro de la cadena.

Luego, utilizamos otra función **SUBSTRING\_INDEX** para obtener el último número de la cadena desglosada por comas.

El **TRIM** se utiliza para eliminar cualquier espacio en blanco adicional alrededor del número, asegurando que la conversión sea correcta.

Usamos **CAST** para convertir este número de cadena a un número entero sin signo (**UNSIGNED**).

Finalmente, seleccionamos `ID` de `inter_products` y el número desglosado como **PRODUCT\_ID**, y unimos esta selección con nuestra subconsulta de `numbers` utilizando **JOIN**

Aquí, `CHAR_LENGTH` y `REPLACE` se utilizan para contar el número de comas en `product_id` y asegurarnos de desglosar todos los números separados por comas hasta la posición `n`.

Finalmente, insertamos los datos desglosados (`ID` y `PRODUCT_ID`) en la tabla `productos_desglosados`.

## SPRINT 4

```
216 --
217
218
219 -- Desglosar los PRODUCT_IDS y luego insertar en productos_desglosados
220 • INSERT INTO productos_desglosados (ID, PRODUCT_ID)
221 SELECT
222     ID,
223     CAST(TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(inter_products.product_ids, ',', numbers.n), ',', -1)) AS UNSIGNED) AS PRODUCT_IDS
224 FROM
225     inter_products
226 JOIN
227     (SELECT 1 n UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5) AS numbers
228     ON CHAR_LENGTH(inter_products.product_ids) - CHAR_LENGTH(REPLACE(inter_products.product_ids, ',', '')) >= numbers.n - 1;
229
```

```
229
230 • select *
231 from productos_desglosados
232
233
234
```

Result Grid

Filter Rows:

Exports

Wrap Cell Contents

Fetch rows:

	id	product_id
▶	02C6201E-D90A-1859-B4EE-88D29...	19
	02C6201E-D90A-1859-B4EE-88D29...	1
	02C6201E-D90A-1859-B4EE-88D29...	71
	0466A42E-47CF-8D24-FD01-C0B68...	43
	0466A42E-47CF-8D24-FD01-C0B68...	97

productos\_desglosados 13

Output

Action Output

#	Time	Action	Message
✓ 1	12:43:26	select * from productos_desglosados LIMIT 0, 1000	1000 row(s) returned

Se procede a realizar las conexiones

```
231 • select *
232 from productos_desglosados;
233 # VAMOS A REALIZAR LAS PK Y FK
234
235 • alter table products modify column id int;
236
237 • ALTER TABLE productos_desglosados
238 ADD PRIMARY KEY(id,product_id),
239 ADD FOREIGN KEY (id) REFERENCES transactions(id),
240 ADD FOREIGN KEY (product_id) REFERENCES products(id);
241
242
```

Output		Filter Rows:	Exports	Wrap Cell Contents:	Fetch rows:
Action Output					
#	Time	Action	Message		
✓ 1	12:01:27	ALTER TABLE productos_desglosados ADD PRIMARY KEY(id,product_id), ADD FOREIGN KEY (id) REFEREN...	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0		

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

## SPRINT 4

Por ultimo se necesita conocer la cantidad de productos se realizo el join de la tabla creada con la tabla producto para llegar al siguiente resultado,

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

244

245 • `SELECT COUNT(*) FROM products;`

246

247 • `SELECT products.id, products.product_name, COUNT(productos_desglosados.product_id) AS product_count`

248 `FROM products`

249 `LEFT JOIN productos_desglosados ON products.id = productos_desglosados.product_id`



250 `GROUP BY products.id, products.product_name`

251 `ORDER BY product_count DESC;`

252

253


254

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

	id	product_name	product_count
▶	23	riverlands north	68
	67	Winterfell	68
	79	Direwolf riverlands the	66
	2	Tarly Stark	65
	43	duel	65
	47	Tully	60

Result 7 x

Output

 Action Output

#	Time	Action	Message
✓ 1	12:40:01	SELECT products.id, products.product_name, COUNT(productos_desglosados.product_id) AS product_c...	100 row(s) returned

## SPRINT 4

