

Pasos previos al ejercicio.

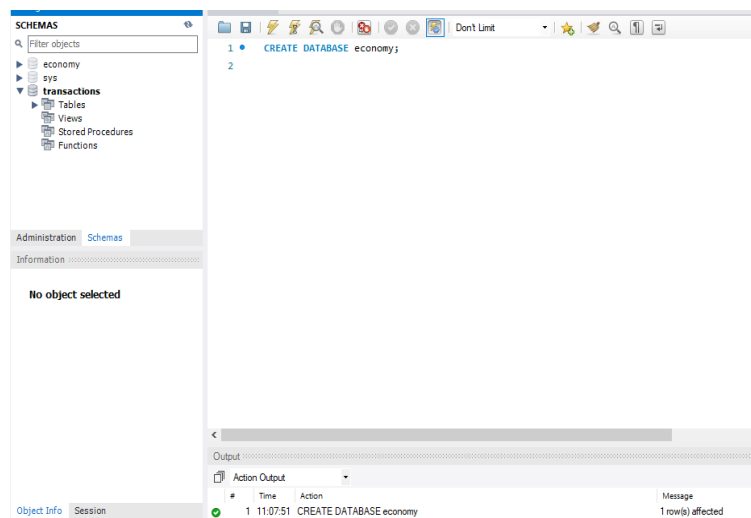
Creación de schema por código.
CREATE DATABASE economy;

Primero de todo intente coger como referencia de los datos la anterior tabla(Sprint3) que guardaba similitud. También intente comprobar directamente en los archivos .csv utilizando calc (=TIPO (campo+fila).Solo arrojaba 0 numérico y 1 texto. Después de muchas pruebas donde por ejemplo int me lo indicaba como caducado. Entonces he ido a lo seguro con varchar. La medidas (255) las he utilizado sobretodo en las PK y Fk.

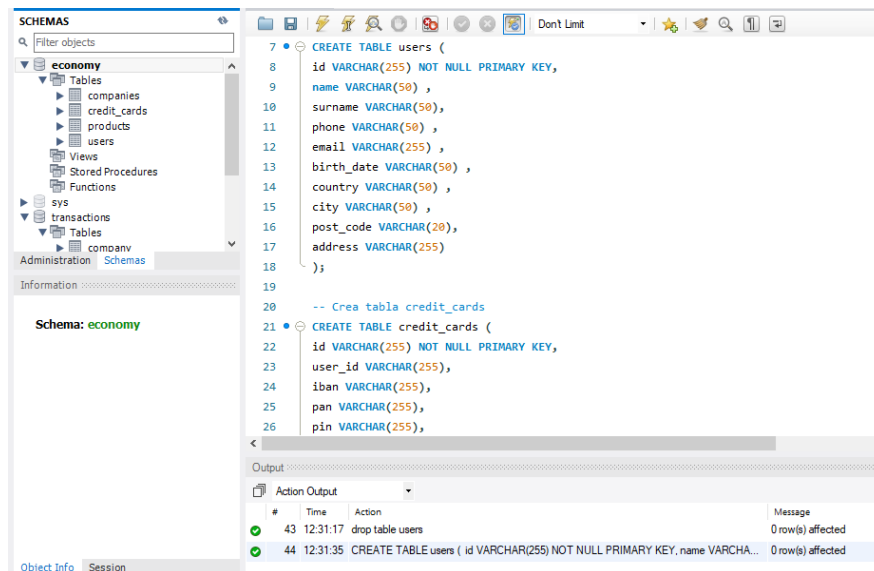
Es cierto que podía ajustar más , pero sin ninguna directriz extra , tanto de evolución de las tablas , empresas , volumen de operaciones o alguna evidencia a medio o largo plazo , he utilizado los numero mas altos para esos campos. Creo que podía utilizar floats en algunos sitios y los utilice . Pero he tenido problemas no causado por esto , mas bien eran errores de escritura de los campos o no estaban bien posicionado los campos . Esto me paso con la tabla transactions.

Errores así me hicieron decantarme con varchar y descartar tipos de datos como decimal para amount o float para longitud / lat. Según la inspección de la tabla anterior de transactions del sprint 3 pues esos datos y tuve algún que otro problema . Se pueden revisar con varias opciones por comando o de manera gráfica.

La tabla transactions la borre varias veces por querer hacerla parecida a la del anterior sprint, pero tenia campo diferentes entre otras cosas .



Creación tabla Users.

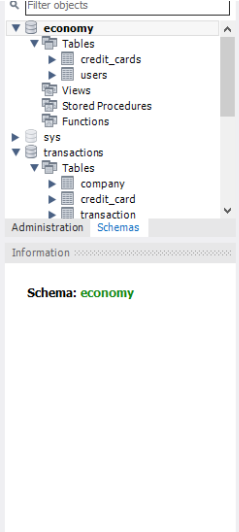


En esta tabla cargue todos los datos de las tres tablas , sin crear unas provisionales. Todos los formatos y campos eran iguales. Solo modificaba la parte final del archivo usa.csv, ca.csv , etc.

Tabla credit_cards

SPRINT 4

Abel

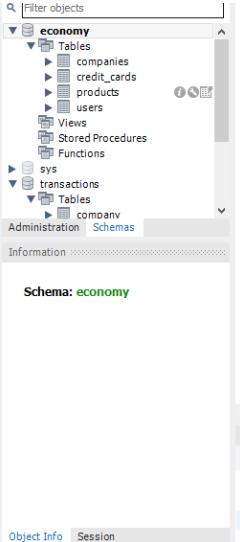


```
10 country VARCHAR(50) ,
11 city VARCHAR(50) ,
12 post_code VARCHAR(20),
13 address VARCHAR(255)
14 );
15
16 -- Crea tabla credit_cards
17 CREATE TABLE credit_cards (
18 id VARCHAR(255) NOT NULL PRIMARY KEY,
19 user_id VARCHAR(255),
20 iban VARCHAR(255),
21 pan VARCHAR(255),
22 pin VARCHAR(255),
23 cvv VARCHAR(50) ,
24 track1 VARCHAR(255) ,
25 track2 VARCHAR(255) ,
26 expiring_date VARCHAR(50)
27 );
28
29 -- Crea tabla companies
```

#	Time	Action	Message
33	12:16:16	drop table products	0 row(s) affected
34	12:18:34	CREATE TABLE credit_cards (id VARCHAR(255) NOT NULL PRIMARY KEY, user_id V...	0 row(s) affected

Tabla products

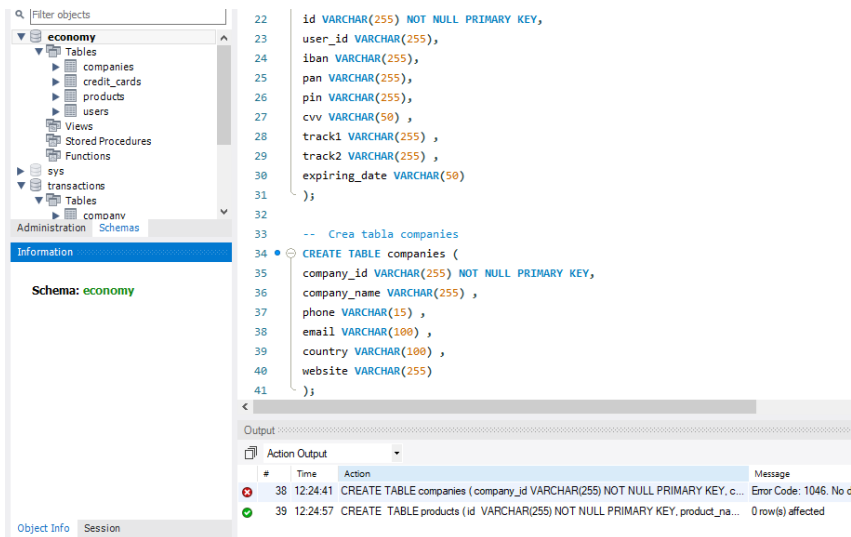
Esta tabla tenia pensado créala y después conectarla a través de una tabla intermedia para no tener una relación $n \rightarrow m$. Esto puede dar problemas de duplicidades o problemas en el futuro . Se puede trabajar si, pero como he comentado antes , en proyectos muy grandes puede ocasionar problemas. Pero al final descarte la idea al no ser necesaria.



```
34 CREATE TABLE companies (
35 company_id VARCHAR(255) NOT NULL PRIMARY KEY,
36 company_name VARCHAR(255) ,
37 phone VARCHAR(15) ,
38 email VARCHAR(100) ,
39 country VARCHAR(100) ,
40 website VARCHAR(255)
41 );
42
43 -- Crea tabla productos
44 CREATE TABLE products (
45 id VARCHAR(255) NOT NULL PRIMARY KEY,
46 product_name VARCHAR(255) ,
47 price VARCHAR(20) ,
48 colour VARCHAR(255) ,
49 weight VARCHAR(255) ,
50 warehouse_id VARCHAR(255)
51 );
52
53
```

#	Time	Action	Message
41	12:29:42	drop table products	0 row(s) affected
42	12:29:53	CREATE TABLE products (id VARCHAR(255) NOT NULL PRIMARY KEY, product_na...	0 row(s) affected

Tabla companies



A continuación muestro la manera de cargar los archivo .csv.

Esto me dio algún dolor de cabeza que otro, por pensar que descargaría directamente si colocaba una ruta absoluta , estuvieran donde estuvieran. Pero tenia otro procedimiento.

Primero con este comando SHOW VARIABLE LIKE 'secure_file_priv',muestra la ruta para colocar los archivos a subir a workbench.

LOAD DATA INFILE – descarga de aquí.

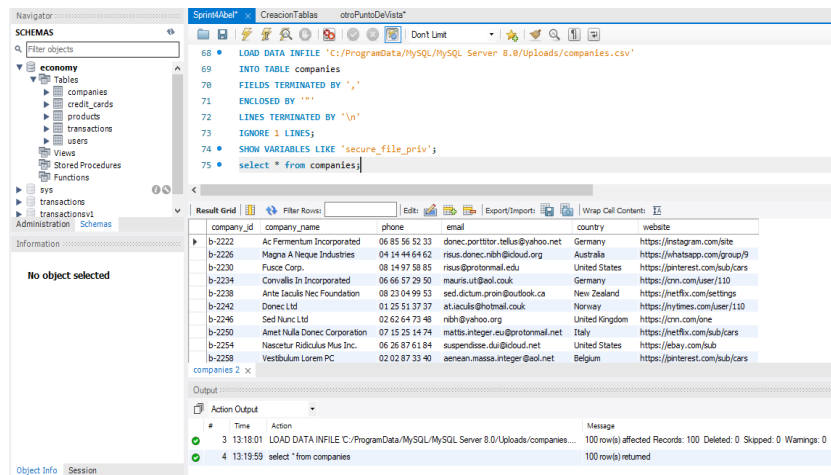
INTO TABLE (en esta tabla)

Algunos archivos tenia sus particularidades en FIELDS TERMINATED BY “,” - “;” - como se delimita cada campo en sus filas.

ENCLOSED BY “” para determinar opción específica entre comillas dobles ,

LINES TERMINATED BY “\n” o “\r\n” y alguna combinación mas que probé , pero esto tenia que ver con la terminación de la línea.

IGNORE _ LINES Para ignorar las líneas del encabezado que quieras. Muchas veces pueden contener títulos o algún tipo de dato no necesario.



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Schemas' tree is expanded to 'economy', showing tables like 'companies', 'credit_cards', 'products', 'transactions', 'users', 'views', 'stored_procedures', and 'functions'. The 'credit_cards' table is selected. In the center, a query window shows a SQL script:

```

75 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv'
76 INTO TABLE credit_cards
77 FIELDS TERMINATED BY ','
78 ENCLOSED BY '"'
79 LINES TERMINATED BY '\n'
80 IGNORE 1 LINES;
81
82 SELECT * FROM credit_cards;
83

```

Below the script, the 'Result Grid' shows a table with columns: id, user_id, iban, pan, pin, cvv, track1, track2. The table contains 275 rows of data. At the bottom, the 'Output' pane shows the execution results:

```

1 15:36:55 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.c... 275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings: 0
2 15:40:37 SELECT* FROM credit_cards 275 row(s) returned

```

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Schemas' tree is expanded to 'economy', showing tables like 'companies', 'credit_cards', 'products', 'transactions', 'users', 'views', 'stored_procedures', and 'functions'. The 'users' table is selected. In the center, a query window shows a SQL script:

```

98 INTO TABLE transactions
99 FIELDS TERMINATED BY ','
100 ENCLOSED BY '"'
101 LINES TERMINATED BY '\n'
102 IGNORE 1 LINES;
103
104 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv'
105 INTO TABLE users
106 FIELDS TERMINATED BY ','
107 ENCLOSED BY '"'
108 LINES TERMINATED BY '\r\n'
109 IGNORE 1 ROWS;
110
111
112
113
114
115
116 SHOW VARIABLES LIKE 'secure_file_priv';

```

Below the script, the 'Output' pane shows the execution results:

```

20 16:17:56 select * from transactions 0 row(s) returned
21 16:19:16 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.c... 150 row(s) affected Records: 150 Deleted: 0 Skipped: 0 Warnings: 0

```

Se pueden crear una tabla users y otra tres . Comporten el mismo formato y tipos de datos. Se puede utilizar una unión , pero creo que en esta ocasión, importar los datos directamente me dio buen resultado.

SPRINT 4

Abel

Navigator

SCHEMAS

Filter objects

Tables

- economy
 - companies
 - credit_cards
 - transactions
 - users
- sys
- transactions
- transactionsv1

Administration Schemas

Information

Schema: economy

CreationTables otroPuntoDeVista

```
105 INTO TABLE users
106 FIELDS TERMINATED BY ','
107 ENCLOSED BY '"'
108 LINES TERMINATED BY '\r\n'
109 IGNORE 1 ROWS;
110
111 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv'
112 INTO TABLE users
113 FIELDS TERMINATED BY ','
114 ENCLOSED BY '"'
115 LINES TERMINATED BY '\r\n'
116 IGNORE 1 ROWS;
117
118 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv'
119 INTO TABLE users
120 FIELDS TERMINATED BY ','
121 ENCLOSED BY '"'
122 LINES TERMINATED BY '\r\n'
123 IGNORE 1 ROWS;
124
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
22	16:21:14	LOAD DATA INFILE C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0	0.016 sec
23	16:22:37	LOAD DATA INFILE C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0	0.015 sec

Object Info Session

Navigator

SCHEMAS

Filter objects

Tables

- economy
 - companies
 - credit_cards
 - transactions
 - users
- sys
- transactions
- transactionsv1

Administration Schemas

Information

Schema: economy

CreationTables otroPuntoDeVista

```
99 FIELDS TERMINATED BY ','
100 ENCLOSED BY '"'
101 LINES TERMINATED BY '\n'
102 IGNORE 1 LINES;
103
104 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv'
105 INTO TABLE users
106 FIELDS TERMINATED BY ','
107 ENCLOSED BY '"'
108 LINES TERMINATED BY '\r\n'
109 IGNORE 1 ROWS;
110
111 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv'
112 INTO TABLE users
113 FIELDS TERMINATED BY ','
114 ENCLOSED BY '"'
115 LINES TERMINATED BY '\r\n'
116 IGNORE 1 ROWS;
117
118
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
21	16:19:16	LOAD DATA INFILE C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv	150 row(s) affected Records: 150 Deleted: 0 Skipped: 0 Warnings: 0	0.015 sec
22	16:21:14	LOAD DATA INFILE C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0	0.016 sec

Object Info Session

Navigator

SCHEMAS

Filter objects

Tables

- economy
 - companies
 - credit_cards
 - transactions
 - users
- sys
- transactions
- transactionsv1

Administration Schemas

Information

Schema: economy

CreationTables otroPuntoDeVista

```
118 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv'
119 INTO TABLE users
120 FIELDS TERMINATED BY ','
121 ENCLOSED BY '"'
122 LINES TERMINATED BY '\r\n'
123 IGNORE 1 ROWS;
124
125 • SELECT * FROM users;
126
```

Result Grid

id	name	surname	phone	email	birth_date	country	city	post_code	address
1	Zeus	Gamble	1-282-581-0551	interdum.erim@protonmail.eu	Nov 17, 1985	United States	Lowell	73544	348-7818 Sagitts St.
10	Robert	McCarthy	(324) 746-6771	fermentum@protonmail.com	Apr 30, 1984	United States	Eugene	85526	P.O. Box 773, 3594 Omare St.
100	Melode	Mclean	1-677-221-7152	rsus.varius@google.ca	Sep 15, 1989	United States	College	11838	Ap #644-6492 Sagitts St.
101	Sarah	Beck	(398) 661-4545	vibac.rissa@iail.co.uk	Apr 9, 1983	United States	Great Falls	67129	66-5-6047 In Rd.
102	Jasper	Landry	1-397-765-1118	consectetur.eusmod@aol.org	Apr 16, 1982	United States	Columbus	11595	Ap #374-7325 Sodales Rd.
103	Upton	Chavez	(227) 785-6484	eusmod.est@aol.ca	Mar 15, 1986	United States	Essex	95631	1990 Vel. Av.
104	Martha	Barlow	(732) 326-5448	vulputate@hotmail.net	Oct 29, 1988	United States	Chicago	41512	Ap #311-7103 In Avenue
105	Hedrem	Rose	(838) 313-6727	urna@icloud.com	Nov 28, 1983	United States	Tacoma	99632	8034 Terbor. Road
106	Tanner	Valenzuela	1-346-421-3135	nasectur.ridiculus@google.net	Apr 6, 1993	United States	Naperville	31130	Ap #114-2616 Fusce Road
107	Victor	Valencia	(239) 969-1938	non.enim@hotmail.co.uk	May 1, 1998	United States	Warren	15158	Ap #182-9926 At St.

users 4 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
23	16:22:37	LOAD DATA INFILE C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0	0.015 sec
24	16:25:35	SELECT * FROM users	275 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SPRINT 4

Abel

The screenshot shows the SQL Developer interface with the 'Schemas' pane on the left and the 'SQL Editor' on the right. The 'Schemas' pane shows a tree structure for the 'economy' schema, including tables like 'companies', 'credit_cards', 'users', 'sys', 'transactions', and 'transactions_v1'. The 'SQL Editor' contains the following SQL code:

```
-- tabla transactions
CREATE TABLE transactions (
  id VARCHAR(255) NOT NULL PRIMARY KEY,
  card_id VARCHAR(255),
  business_id VARCHAR(255),
  timestamp VARCHAR(20),
  amount VARCHAR(20),
  declined VARCHAR(1),
  product_ids VARCHAR(255),
  user_id VARCHAR(255),
  lat VARCHAR(255),
  longitude VARCHAR(255)
);
describe transactions;
select * from transactions;
drop table transactions;
```

The 'Output' pane at the bottom shows the execution results:

#	Time	Action	Message
43	11:41:53	Drop table transactions	0 row(s) affected
44	11:41:57	CREATE TABLE transactions (id VARCHAR(255) NOT NULL PRIMARY KEY, card_id ...	0 row(s) affected

The screenshot shows the SQL Developer interface with the 'Schemas' pane on the left and the 'SQL Editor' on the right. The 'Schemas' pane shows a tree structure for the 'economy' schema, including tables like 'companies', 'credit_cards', 'users', 'sys', 'transactions', and 'transactions_v1'. The 'SQL Editor' contains the following SQL code:

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'
INTO TABLE transactions
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
SELECT * FROM transactions;
```

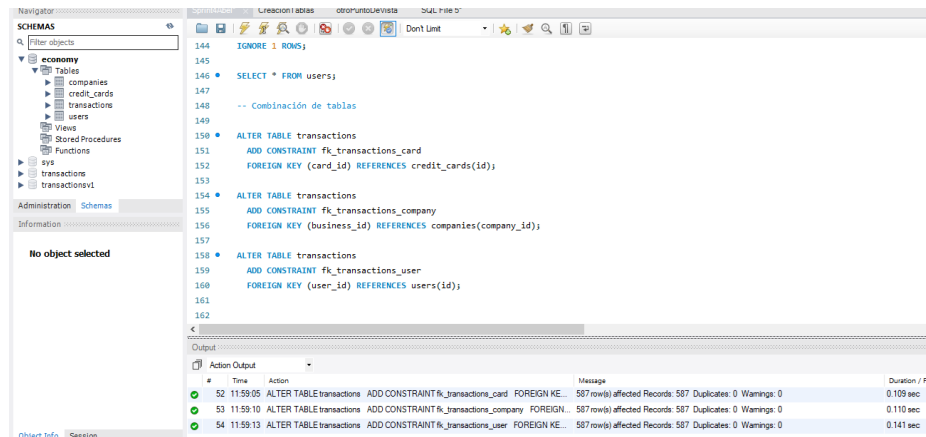
The 'Result Grid' shows the data loaded into the 'transactions' table:

id	card_id	business_id	timestamp	amount	declined	product_ids	user_id	lat	longitude
03C5301E-096A-1859-84EE-80C2086D3B02	CdU-2938	b-2362	2021-08-18 22:42:24	466.92	0	71, 1, 19	92	81.918458824	-12.527561894
9466443E-47CF-4024-FD01-C06689713128	CdU-4219	b-2302	2021-07-26 07:29:18	49.53	0	47, 97, 43	170	-43.969488588	-117.5251835904
063FBA79-99EC-66FB-29F7-23726D1764A5	CdU-2987	b-2250	2022-01-06 21:25:27	92.61	0	47, 67, 31, 5	275	-81.222680576	-129.048879552
0668296C-CD89-A883-78BC-2E4C4F8C8AE	CdU-3743	b-2618	2022-01-26 02:07:14	394.18	0	89, 83, 79	265	-34.3593055232	-100.559928064
06CD9AA5-8642-0684-0000-ASE394FEBAA9	CdU-2959	b-2346	2021-10-26 23:00:01	279.93	0	43, 31	92	33.7381445632	158.298210304
07A46D48-21A3-7B37-638F-0DA902AD109F	CdU-3225	b-2386	2021-06-28 21:11:42	340.87	1	47, 23	272	36.8341336528	92.190549408
09259DCE-4F27-28F7-13E5-938362B38E21	CdU-3071	b-2298	2021-05-11 20:40:06	303.05	1	67, 7	275	71.1705612288	10.5765762996
0A476ED9-0C13-1962-F876-035639246539	CdU-4359	b-2302	2022-02-26 20:33:54	430.49	0	29, 41, 11	221	-56.490077696	114.8012215648
0EB8087-9D66-1707-CE4E-9DC7E71914B5	CdU-3141	b-2338	2022-03-04 14:54:35	288.81	1	19, 41, 29, 3	272	23.3264027648	-13.6036868096
0C7C3A33-9947-38C1-846D-78E3D0D17598	CdU-3309	b-2434	2021-04-10 20:58:41	103.44	1	89, 31	272	63.3614763008	-68.6666633216

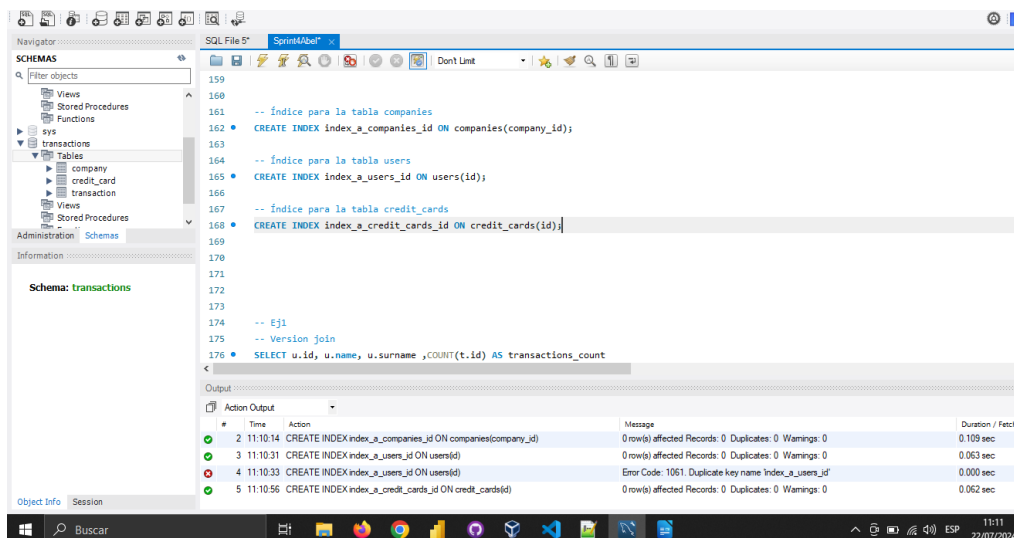
The 'Output' pane at the bottom shows the execution results:

#	Time	Action	Message
45	11:43:54	LOAD DATA INFILE C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0
46	11:43:59	SELECT * FROM transactions	587 row(s) returned

Relación de tablas , donde se efectúa una alteración/modificación de la tabla X.
 Restricción de la tabla donde se le añade un texto y el texto da una idea de la operación.
 Por ultimo se especifica la Fk y se establece la referencia con la otra tabla y el campo.



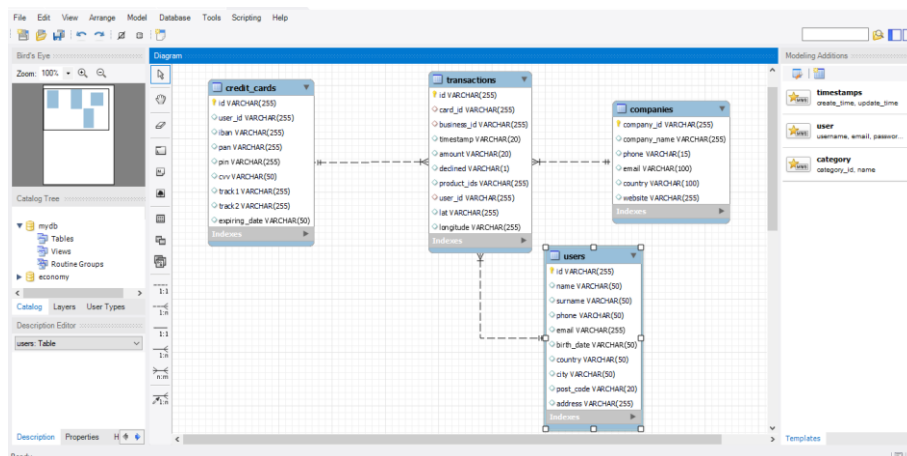
Creación de la índices para una mejora en la optimización de las búsquedas.
 CREATE_INDEX nombre_como_referencia ON nombre_de_tabla(campo).
 Se utiliza el índice para una mayor velocidad y optimización de las consultas.



Esquema
estrella ,

tipo
con la tabla

transactions como tabla de hechos. Porque tiene un mayor numero de métricas. Además que es la tabla donde explica una historia de lo que esta pasando.



EJ 1-Realiza una subconsulta que muestre a todos los usuarios con más de 30 transacciones utilizando al menos 2 tablas.
 -Versión Join. Versión rápida para ver lo resultados finales y comparar. Ya que el ejercicio, entiendo que pide una subconsulta.
 -Versión subconsulta
 Repito dos veces la misma subconsulta una en el select, el campo obtenido de la consulta interior. Después utilizo la misma subconsulta para filtrar.

The screenshot shows a SQL IDE interface with a query editor and a results grid. The query is as follows:

```

237 ORDER BY transactions_count DESC;*/
238
239
240 -- Version Subquery
241 SELECT u.id, u.name, u.surname,
242 (SELECT COUNT(ts.id)
243 FROM transactions ts
244 WHERE ts.user_id = u.id) AS transactions_count
245 FROM users u
246 WHERE (SELECT COUNT(ts.id)
247 FROM transactions ts
248 WHERE ts.user_id = u.id) > 30
249 ORDER BY transactions_count DESC;
250
251 SELECT u.id, u.name, u.surname, tsc.transactions_count
  
```

The results grid shows the following data:

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

The Action Output section shows the execution of the query:

#	Time	Action	Message
13	11:21:35	SELECT u.id, u.name, u.surname, (SELECT COUNT(ts.id) FROM transactions t...	4 row(s) returned
14	11:21:38	SELECT u.id, u.name, u.surname, (SELECT COUNT(ts.id) FROM transactions t...	4 row(s) returned

EJ 2-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 with a query editor and a results grid. The query is as follows:

```

181
182 -- EJ 2
183
184 SELECT cc.iban, c.company_name, ROUND(AVG(t.amount), 2) AS media
185 FROM credit_cards cc
186 JOIN transactions t
187 ON cc.id = t.card_id
188 JOIN companies c
189 ON t.business_id = c.company_id
190 WHERE c.company_name = 'Donec Ltd'
191 GROUP BY cc.iban, c.company_name;
  
```

The results grid shows the following data:

iban	company_name	media
PF8706628812002420463246	Donec Ltd	203.72

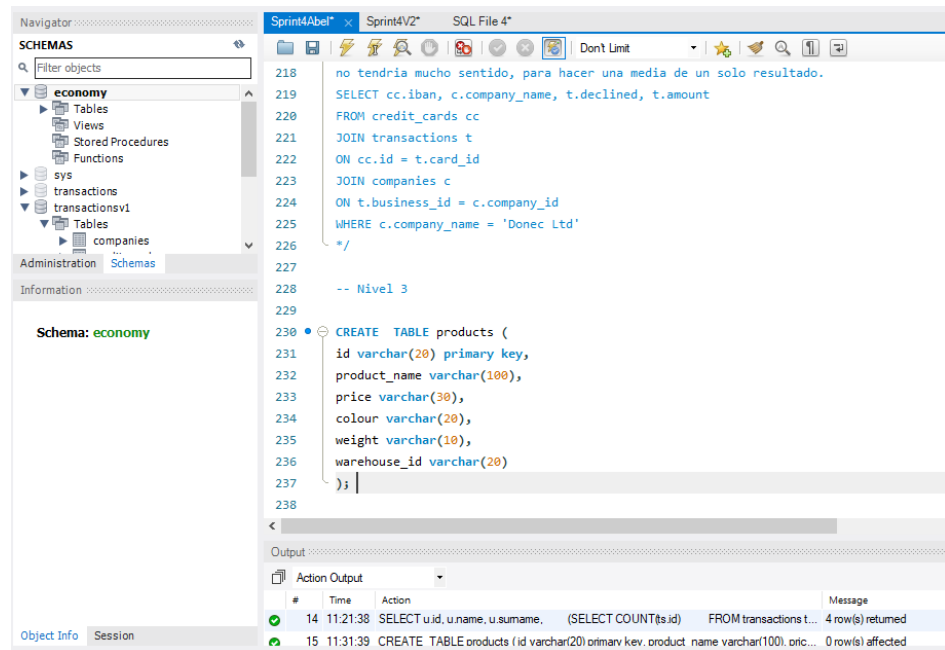
The Action Output section shows the execution of the query:

#	Time	Action	Message
1	21:38:37	SELECT u.id, u.name, u.surname, COUNT(*) AS transactions_count FROM users u JOIN transactions t ON u.id = t.user_id GROUP BY u.id HAVING...	4 row(s) returned
2	21:41:35	SELECT cc.iban, c.company_name, ROUND(AVG(t.amount), 2) AS media FROM credit_cards cc JOIN transactions t ON cc.id = t.card_id JOIN com...	1 row(s) returned

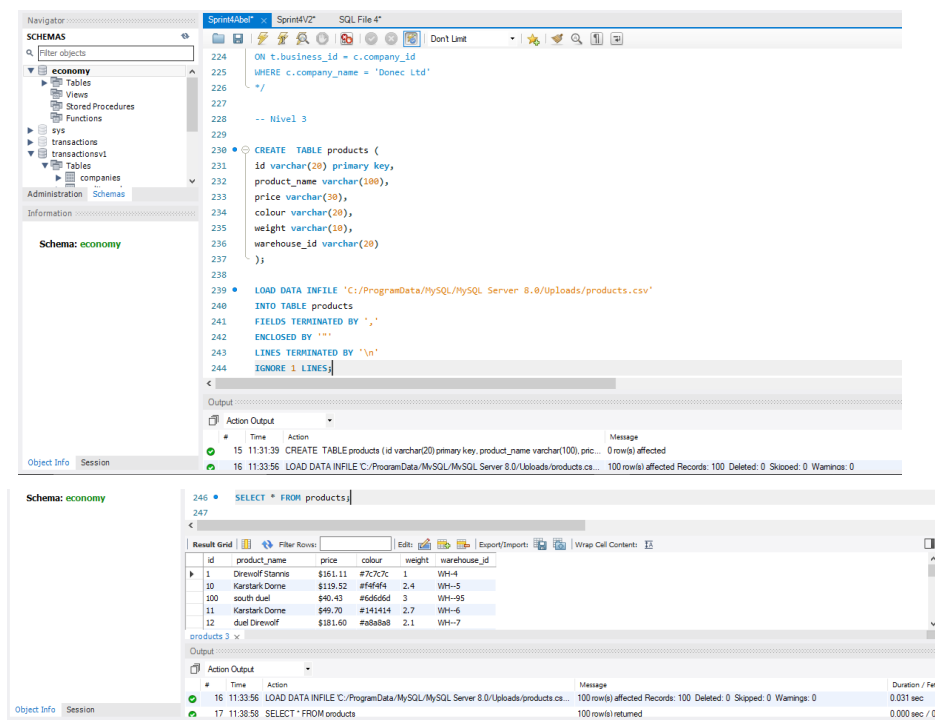
Debido a las pocas operaciones, decidí incluir todas las operaciones del campo decline, para mostrar el funcionamiento de AVG.

Nivel III

Creación tabla productos



Cargo datos a la tabla productos.



Creación tabla intermedia trans_productos

SPRINT 4

Abel

```

249 • create table trans_products (
250   id_transaction varchar(50) NOT NULL,
251   id_product varchar(20) NOT NULL,
252   primary key (id_transaction, id_product),
253   foreign key (id_transaction) references transactions(id),
254   foreign key (id_product) references products(id));

```

Output

#	Time	Action	Message
17	11:38:58	SELECT * FROM products	100 row(s) returned
18	11:40:52	create table trans_products (id_transaction varchar(50) NOT NULL, id_product varchar(20) NOT NULL, primary key (id_transaction, id_product), foreign key (id_transaction) references transactions(id), foreign key (id_product) references products(id));	0 row(s) affected

Cargar datos con insert into

```

256 • insert into trans_products (
257   id_transaction, id_product)
258   select t.id, p.id
259   from transactions t
260   inner join products p on find_in_set(p.id, replace(t.product_ids, ','))
261   where declined = 0;

```

Output

#	Time	Action	Message	Duration / Fetch
18	11:40:52	create table trans_products (id_transaction varchar(50) NOT NULL, id_product varchar(20) NOT NULL, primary key (id_transaction, id_product), foreign key (id_transaction) references transactions(id), foreign key (id_product) references products(id));	0 row(s) affected	0.078 sec
19	11:42:48	insert into trans_products (id_transaction, id_product) select t.id, p.id from transactions t inner join products p on find_in_set(p.id, replace(t.product_ids, ',')) where declined = 0;	1236 row(s) affected Records: 1236 Duplicates: 0 Warnings: 0	0.125 sec

Nivel 3
Ej 1

Forma
esquema ,
ha

Schema: economy

```

263 • SELECT * FROM trans_products;

```

Result Grid

id_transaction	id_product
02C6201E-090A-1859-84EE-88D2986D3B02	1
122DC333-E19F-0629-DC08-9C54CF1EBB9A	1
1763A788-4EC1-10F6-5C70-A1FFB978D13A	1

```

266 • select p.id, p.product_name, count(id_transaction) as total_transaction
267   from trans_products t
268   inner join products p
269   on t.id_product = p.id
270   group by p.id, p.product_name
271   order by total_transaction desc;

```

Result Grid

id	product_name	total_transaction
23	riverlands north	60
67	Winterfell	59
2	Tarly Stark	56
17	skywalker evok sth	54
43	duel	54

Output

#	Time	Action	Message
20	11:43:53	SELECT * FROM trans_products	1236 row(s) returned
21	11:45:47	select p.id, p.product_name, count(id_transaction) as total_transaction from trans_products t inner join products p on t.id_product = p.id group by p.id, p.product_name order by total_transaction desc;	26 row(s) returned

final del
ahora se

transformado en un modelo copo de nieve, donde algunas dimensiones interactúan entre si.

