S4.01. Creación de Base de Datos

Nivel 1

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

```
-- Creamos la base de datos y tablas
```

```
4      -- Creamos la base de datos
5      DROP DATABASE IF EXISTS transactions_db;
6      CREATE DATABASE transactions_db;
7      USE transactions_db;
8       -- (columnas) id,name,surname,phone,email,birth_date,country,city,postal_code,address
9       --            id,name,surname,phone,email,birth_date,country,city,postal_code,address
10       -- Creamos la tabla user
11     CREATE TABLE IF NOT EXISTS user (
12         id INT PRIMARY KEY,
13         name VARCHAR(100),
14         surname VARCHAR(100),
15         phone VARCHAR(150),
16         email VARCHAR(150),
17         birth_date DATE,
18         country VARCHAR(150),
19         city VARCHAR(150),
20         postal_code VARCHAR(100),
21         address VARCHAR(255)
22     );
```

Output

Action Output

#	Time	Action	Message
1	01:10:07	DROP DATABASE IF EXISTS transactions_db	7 row(s) affected
2	01:10:12	CREATE DATABASE transactions_db	1 row(s) affected
3	01:10:12	USE transactions_db	0 row(s) affected
4	01:10:12	CREATE TABLE IF NOT EXISTS user (id INT PRIMARY KEY, name VARCHAR(100), su...	0 row(s) affected

```
-- company_id,company_name,phone,email,country,website
  -- Creamos la tabla company
```

```sql
CREATE TABLE IF NOT EXISTS company (
    company_id VARCHAR(15) PRIMARY KEY,
    company_name VARCHAR(255),
    phone VARCHAR(15),
    email VARCHAR(100),
    country VARCHAR(100),
    website VARCHAR(255)
);
```

Output

Action Output

#	Time	Action	Message
3	01:10:12	USE transactions_db	0 row(s) affected
4	01:10:12	CREATE TABLE IF NOT EXISTS user (id INT PRIMARY KEY, name VARCHAR(100), ...	0 row(s) affected
5	01:13:33	CREATE TABLE IF NOT EXISTS company (company_id VARCHAR(15) PRIMARY KEY, ...	0 row(s) affected

```
-- id,user_id,iban,pan,pin,cvv,track1,track2,expiring_date
  -- Creamos la tabla credit_card
```

```sql
CREATE TABLE IF NOT EXISTS credit_card (
    id VARCHAR(20) PRIMARY KEY,
    user_id INT,
    iban VARCHAR(50),
    pan VARCHAR(20),
    pin VARCHAR(4),
    cvv VARCHAR(4),
    track1 VARCHAR(255),
    track2 VARCHAR(255),
    expiring_date DATE
);
```

Output

Action Output

#	Time	Action	Message
4	01:10:12	CREATE TABLE IF NOT EXISTS user (id INT PRIMARY KEY, name VARCHAR(100), ...	0 row(s) affected
5	01:13:33	CREATE TABLE IF NOT EXISTS company (company_id VARCHAR(15) PRIMARY KEY, ...	0 row(s) affected
6	01:15:07	CREATE TABLE IF NOT EXISTS credit_card (id VARCHAR(20) PRIMARY KEY, user_id I...	0 row(s) affected

-- id card_id business_id timestamp amount declined product_ids user_id lat longitude
  -- Creamos la tabla transactions

```sql
CREATE TABLE IF NOT EXISTS transactions (
    id VARCHAR(255) PRIMARY KEY,
    card_id VARCHAR(20),
    business_id VARCHAR(15),
    timestamp timestamp,
    amount DECIMAL(10,2),
    declined BOOLEAN,
    product_ids VARCHAR(255),
    user_id INT,
    lat VARCHAR(50),
    longitude VARCHAR(50),
    FOREIGN KEY (card_id) REFERENCES credit_card(id),
    FOREIGN KEY (business_id) REFERENCES company(company_id),
    FOREIGN KEY (user_id) REFERENCES user(id)
);
```

Output

Action Output

#	Time	Action	Message
4	01:18:13	CREATE TABLE IF NOT EXISTS company (company_id VARCHAR(15) PRIMARY KEY, ...	0 row(s) affected
5	01:18:18	CREATE TABLE IF NOT EXISTS credit_card (id VARCHAR(20) PRIMARY KEY, user_id I...	0 row(s) affected
6	01:35:19	CREATE TABLE IF NOT EXISTS transactions (id VARCHAR(255) PRIMARY KEY, card_i...	0 row(s) affected

-- Se procederá a cargar los archivos csv
-- Antes de cargar verificamos si está activado 'local_infile', para poder cargar los archivos

```sql
-- Se procederá a cargar los archivos csv
-- Antes de cargar verificamos si está activado 'local_infile', para poder cargar los archivos
SHOW VARIABLES LIKE 'local_infile';

-- tenemos que activar local_infile en el servidor
-- Activa la opción para todas las sesiones nuevas, pero no afecta la sesión actual.
SET GLOBAL local_infile = 'ON';

-- El siguiente comando nos ayuda a conocer la ruta en donde se deben colocar
-- los archivos *.csv para proceder con la carga de data
SHOW VARIABLES LIKE "secure_file_priv";
```

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

Variable_name	Value
secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\

transactions 104 Result 105 Result 106

Output

Action Output

#	Time	Action	Message
8	01:55:19	SHOW VARIABLES LIKE 'local_infile'	1 row(s) returned
9	01:55:19	SET GLOBAL local_infile = 'ON'	0 row(s) affected
10	01:55:19	SHOW VARIABLES LIKE "secure_file_priv"	1 row(s) returned

Se comienza a cargar el archivo de los usuarios comenzando por users_usa.csv, ya que tiene los ID desde 1 hasta 150

```
83  LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv'
84  INTO TABLE user
85  FIELDS TERMINATED BY ','
86  ENCLOSED BY '"'
87  LINES TERMINATED BY '\r\n'
88  IGNORE 1 ROWS
89  (id, name, surname, phone, email, @birth_date, country, city, postal_code, address)
90  SET birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y');
```

Output

Action Output

#	Time	Action	Message
9	01:55:19	SET GLOBAL local_infile = 'ON'	0 row(s) affected
10	01:55:19	SHOW VARIABLES LIKE "secure_file_priv"	1 row(s) returned
11	01:59:53	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv' IN...	150 row(s) affected Records: 150

Se continúa con el archivo users_uk.csv, ya que tiene los ID desde 151 hasta 200

```
94  -- Se continúa con el archivo users_uk.csv, ya que tiene los ID desde 151 hasta 200
95  LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv'
96  INTO TABLE user
97  FIELDS TERMINATED BY ','
98  ENCLOSED BY '"'
99  LINES TERMINATED BY '\r\n'
100 IGNORE 1 ROWS
101 (id, name, surname, phone, email, @birth_date, country, city, postal_code, address)
102 SET birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y');
```

Output

Action Output

#	Time	Action	Message
10	01:55:19	SHOW VARIABLES LIKE "secure_file_priv"	1 row(s) returned
11	01:59:53	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv' IN...	150 row(s) affected Records: 150
12	02:02:39	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv' IN...	50 row(s) affected Records: 50 De

Se continúa con el archivo users_ca.csv, ya que tiene los ID desde 201 hasta 275

```
107 •  LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv'
108    INTO TABLE user
109    FIELDS TERMINATED BY ','
110    ENCLOSED BY '"'
111    LINES TERMINATED BY '\r\n'
112    IGNORE 1 ROWS
113    (id, name, surname, phone, email, @birth_date, country, city, postal_code, address)
114    SET birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y');
115
116 •  SELECT * FROM user;
117
```

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

id	name	surname	phone	email	birth_date	country	city	postal_code
1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.edu	1985-11-17	United States	Lowell	73544
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	1992-08-23	United States	Des Moines	59464
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	1998-04-29	United States	Columbus	56518
4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.edu	1989-02-18	United States	Kailua	77417
5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	1998-09-26	United States	Sandy	31564

user 107

Output

Action Output

#	Time	Action	Message
12	02:02:39	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv' IN...	50 row(s) affected Records: 50 Deleted: 0
13	02:03:34	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv' IN...	75 row(s) affected Records: 75 Deleted: 0
14	02:03:40	SELECT * FROM user	275 row(s) returned

-- Se continúa con la carga en la tabla company

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

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

company 108

Output

Action Output

#	Time	Action	Message
14	02:03:40	SELECT * FROM user	275 row(s) returned
15	02:05:08	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv' I...	100 row(s) affected Records: 100 Deleted: 0
16	02:05:13	SELECT * FROM company	100 row(s) returned

-- Se carga la tabla credit_card

```
128 ● LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv'
129   INTO TABLE credit_card
130   FIELDS TERMINATED BY ','
131   LINES TERMINATED BY '\n'
132   IGNORE 1 ROWS
133   (id, user_id, iban, pan, pin, cvv, track1, track2, @expiring_date)
134   SET expiring_date = STR_TO_DATE(@expiring_date, '%m/%d/%y');
135
136 ● SELECT * FROM credit_card;
137
```

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

id	user_id	iban	pan	pin	cvv	track1	track2	expiring_date
CcU-2938	275	TR301950312213576817638661	5424465566813633	3257	984	%B8383712448554646^WovsxejDpwiev^8604...	%B7653863056044187=800716333673	2022-10-30
CcU-2945	274	DO268547637485374752165686 89	5142423821948828	9080	887	%B4621311609958661^UftuyfsSeimxn^06106...	%B4149568437843501=510714033071	2023-08-24
CcU-2952	273	BG45IVQL52710525608255	4556 453 55 5287	4598	438	%B2183285104307501^CddyytcUxwfdq^5907...	%B6778580257827162=690685974007?7	2021-06-29
CcU-2959	272	CR7242477244335841535	372461377349375	3583	667	%B7281111956795320^XocddjBckecd^09016...	%B4246154489281853=280522391678	2023-02-24
CcU-2966	271	BG72LKTQ15627628377363	448566 886747 7265	4900	130	%B4728932322756223^JhlgvsuFbmwgj^7202...	%B2318571115599881=890821578475	2024-10-29

credit_card 109 | Apply

Output

Action Output

#	Time	Action	Message	Duration /
16	02:05:13	SELECT * FROM company	100 row(s) returned	0.000 sec
17	02:06:07	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv' ...	275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings: 0	0.500 sec
18	02:06:12	SELECT * FROM credit_card	275 row(s) returned	0.000 sec

-- Carga de la tabla transactions

```
138   -- Carga de la tabla transactions
139   LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'
140   INTO TABLE transactions
141   FIELDS TERMINATED BY ';'
142   LINES TERMINATED BY '\r\n'
143   IGNORE 1 ROWS;
144
145 ● select * from transactions;
146
147   /* Nivel 1
```

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-88D2986D3B02	CcU-2938	b-2362	2021-08-28 23:42:24	466.92	0	71, 1, 19	92	81.9184589824	-12.5275561984
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	2021-07-26 07:29:18	49.53	0	47, 97, 43	170	-43.9694885888	-117.5251835904
063FBA79-99EC-66FB-29F7-25726D1764A5	CcU-2987	b-2250	2022-01-06 21:25:27	92.61	0	47, 67, 31, 5	275	-81.222680576	-129.049879552
0668296C-CDB9-A883-76BC-2E4C44F8C8AE	CcU-3743	b-2618	2022-01-26 02:07:14	394.18	0	89, 83, 79	265	-34.3593055232	-100.555928064
06CD9AA5-9B42-D684-DDDD-A5E394FEBA99	CcU-2959	b-2346	2021-10-26 23:00:01	279.93	0	43, 31	92	33.7381445632	158.298210304

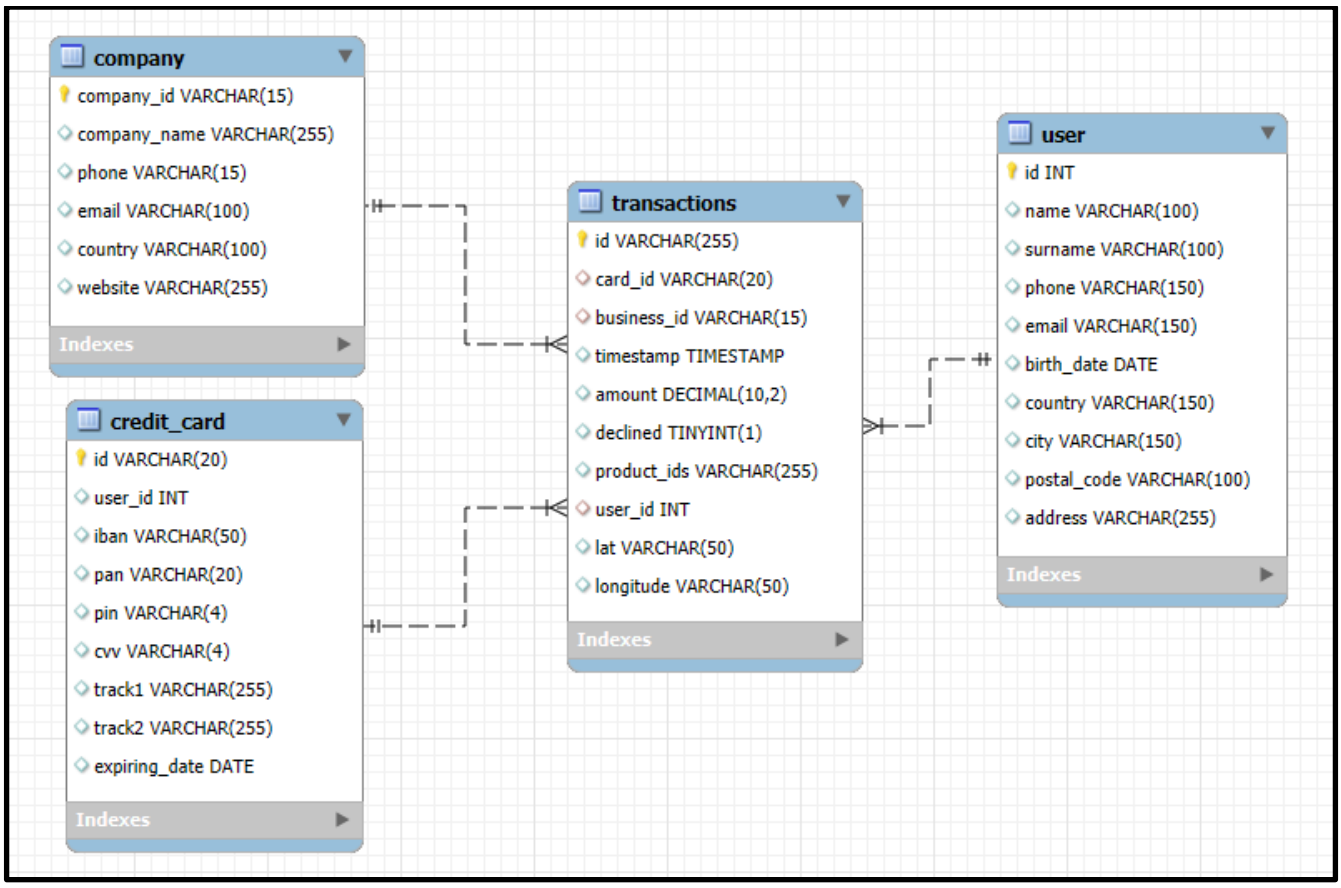
transactions 110

Output

Action Output

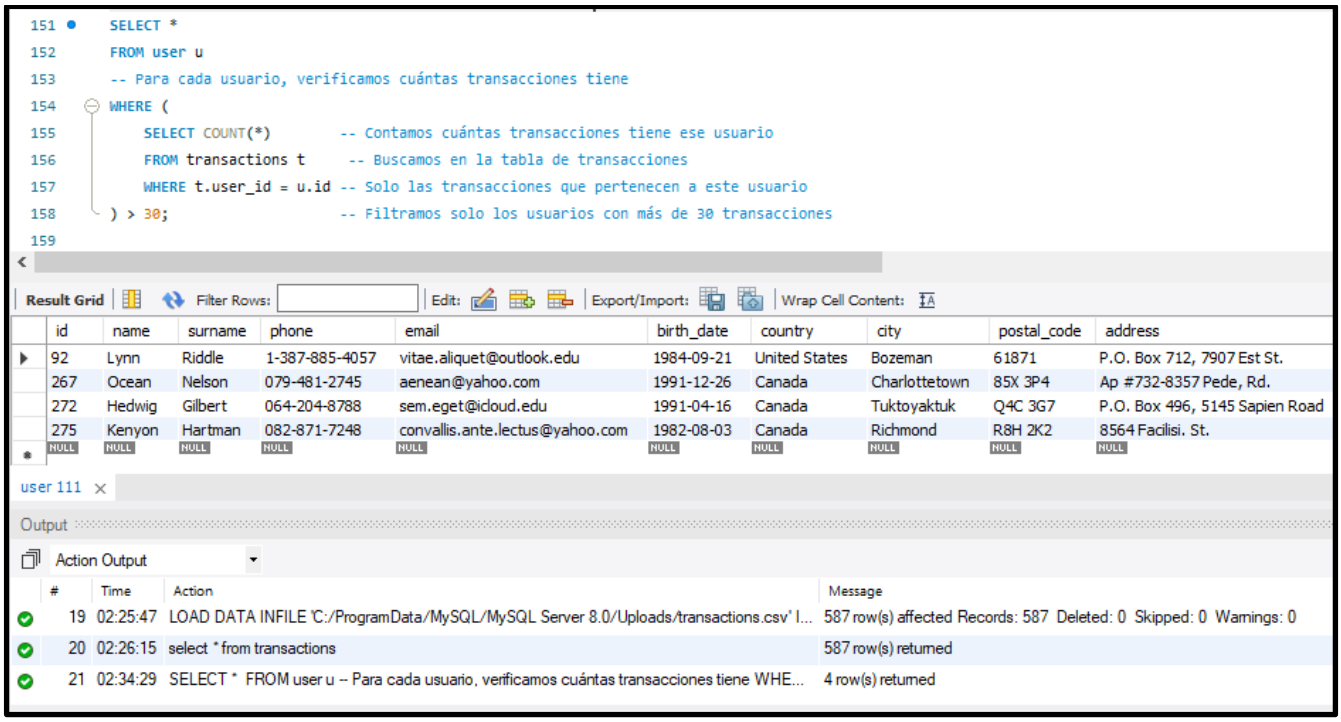
#	Time	Action	Message
18	02:06:12	SELECT * FROM credit_card	275 row(s) returned
19	02:25:47	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv' I...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0
20	02:26:15	select * from transactions	587 row(s) returned

Diagrama de la base de datos transactions_db con las 4 tablas creadas



Nivel 1

- **Ejercicio 1**
  Realiza una subconsulta que muestre todos los usuarios con más de 30 transacciones utilizando al menos 2 tablas.



```
151 •  SELECT *
152    FROM user u
153    -- Para cada usuario, verificamos cuántas transacciones tiene
154    WHERE (
155        SELECT COUNT(*)        -- Contamos cuántas transacciones tiene ese usuario
156        FROM transactions t     -- Buscamos en la tabla de transacciones
157        WHERE t.user_id = u.id -- Solo las transacciones que pertenecen a este usuario
158    ) > 30;                    -- Filtramos solo los usuarios con más de 30 transacciones
159
```

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

id	name	surname	phone	email	birth_date	country	city	postal_code	address
92	Lynn	Riddle	1-387-885-4057	vitae.aliquet@outlook.edu	1984-09-21	United States	Bozeman	61871	P.O. Box 712, 7907 Est St.
267	Ocean	Nelson	079-481-2745	aenean@yahoo.com	1991-12-26	Canada	Charlottetown	85X 3P4	Ap #732-8357 Pede, Rd.
272	Hedwig	Gilbert	064-204-8788	sem.eget@icloud.edu	1991-04-16	Canada	Tuktoyaktuk	Q4C 3G7	P.O. Box 496, 5145 Sapien Road
275	Kenyon	Hartman	082-871-7248	convallis.ante.lectus@yahoo.com	1982-08-03	Canada	Richmond	R8H 2K2	8564 Facilisi. St.
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

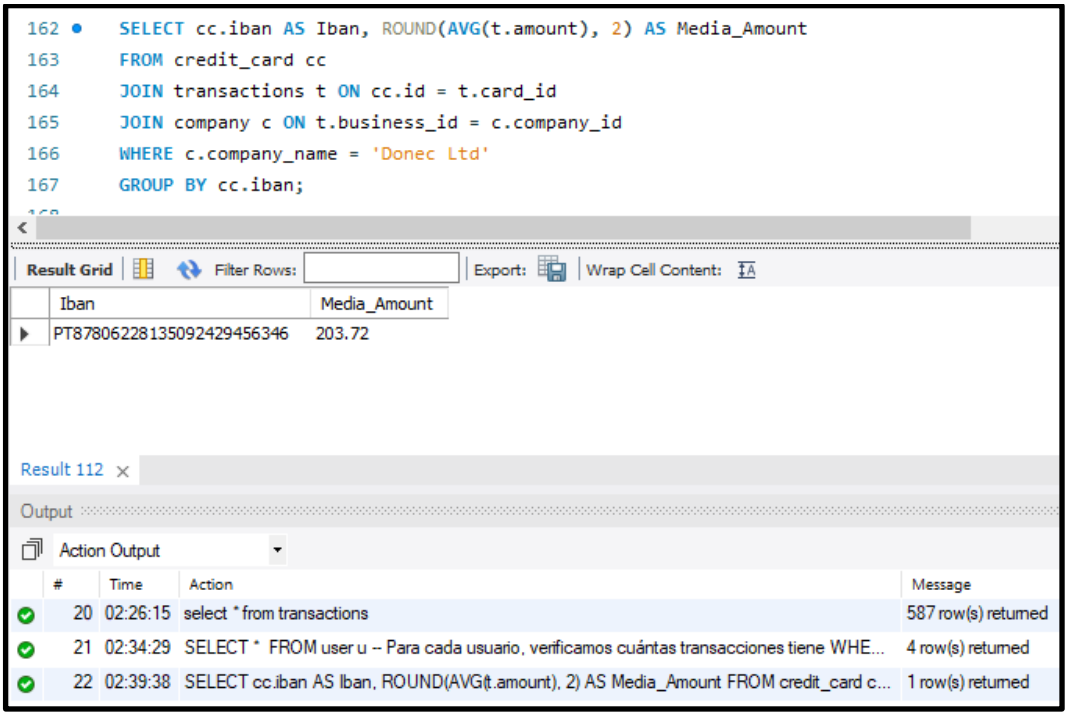
user 111

Output

Action Output

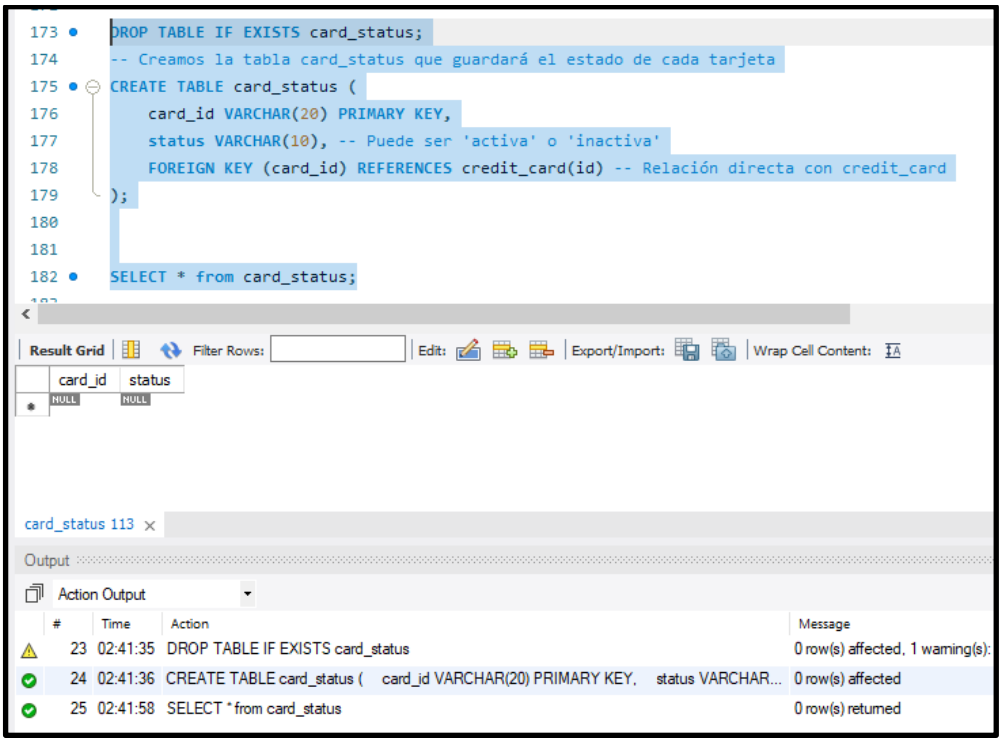
#	Time	Action	Message
19	02:25:47	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv' I...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0
20	02:26:15	select * from transactions	587 row(s) returned
21	02:34:29	SELECT * FROM user u -- Para cada usuario, verificamos cuántas transacciones tiene WHE...	4 row(s) returned

- **Ejercicio 2**
  Muestra la media de `amount` por IBAN de las tarjetas de crédito en la compañía **Donec Ltd**, utilizando al menos 2 tablas.



Nivel 2

Crea una nueva tabla que refleje el estado de las tarjetas de crédito basándote en si las últimas tres transacciones fueron rechazadas y genera la siguiente consulta:



```sql
-- Insertamos una fila por cada tarjeta indicando si está activa o inactiva
-- Insertamos el estado de cada tarjeta según sus 3 últimas transacciones
INSERT INTO card_status (card_id, status)
SELECT card_id,
    CASE
        WHEN SUM(declined) = 3 THEN 'inactiva'  -- Las 3 últimas transacciones fueron rechazadas
        ELSE 'activa'                           -- Al menos una fue aceptada
    END AS status
FROM (
    -- Numeramos las transacciones por tarjeta desde la más reciente
    SELECT card_id, declined,
    -- Enumeramos las transacciones por tarjeta, desde la más reciente
        ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS fila
    FROM transactions
    WHERE card_id IS NOT NULL -- Filtramos las transacciones que tienen un card_id no nulo
) AS ultimas
-- Nos quedamos con las 3 más recientes por tarjeta
WHERE fila <= 3 -- Filtramos para quedarnos solo con las 3 últimas transacciones por tarjeta
GROUP BY card_id; -- Agrupamos por card_id para calcular el estado de cada tarjeta
```

Output

Action Output

#	Time	Action	Message
25	02:41:58	SELECT * from card_status	0 row(s) returned
26	02:48:20	INSERT INTO card_status (card_id, status) SELECT card_id, CASE WHEN SUM(dec...	275 row(s) affected Records: 275 Du

```sql
SELECT * from card_status;
```

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

card_id	status
CcU-2938	activa
CcU-2945	activa
CcU-2952	activa
CcU-2959	activa
CcU-2966	activa

card_status 114

Output

Action Output

#	Time	Action	Message
26	02:48:20	INSERT INTO card_status (card_id, status) SELECT card_id, CASE WHEN SUM(dec...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0
27	02:54:23	SELECT * from card_status	275 row(s) returned

- **Ejercicio 1**
  ¿Cuántas tarjetas están activas?

```sql
/*  Ejercicio 1
¿Cuántas tarjetas están activas? */

SELECT COUNT(*) AS tarjetas_activas
FROM card_status
WHERE status = 'activa';
```

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

tarjetas_activas
275

Result 115

Output

Action Output

#	Time	Action	Message
26	02:48:20	INSERT INTO card_status (card_id, status) SELECT card_id, CASE WHEN SUM(dec...	275 row(s) affected
27	02:54:23	SELECT * from card_status	275 row(s) returned
28	02:56:14	SELECT COUNT(*) AS tarjetas_activas FROM card_status WHERE status = 'activa'	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 la tabla `transactions` tienes `product_ids`. Genera la siguiente consulta:

```sql
-- creamos latabla products
DROP TABLE IF EXISTS products;
CREATE TABLE products (
    id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10,2),         -- Precio como número decimal
    currency_symbol CHAR(1),     -- Nuevo campo para almacenar el símbolo '$'
    colour VARCHAR(20),
    weight DECIMAL(8,2),
    warehouse_id VARCHAR(10)
);
```

Output

Action Output

#	Time	Action	Message
29	02:58:44	DROP TABLE IF EXISTS products	0 row(s) affected, 1 warning(s):
30	02:58:44	CREATE TABLE products (id INT PRIMARY KEY, product_name VARCHAR(100), pri...	0 row(s) affected

-- Carga de la tabla products

```
229    -- Carga de la tabla products
230    -- carga el CSV y se convierte el campo price y se crea una nueva columna para el símbolo de moneda
231    LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv'
232    INTO TABLE products
233    FIELDS TERMINATED BY ','
234    OPTIONALLY ENCLOSED BY '"'
235    LINES TERMINATED BY '\n'
236    IGNORE 1 ROWS
237    (@id, @product_name, @price_raw, @colour, @weight, @warehouse_id)
238    SET
239        id = @id,
240        product_name = @product_name,
241        price = CAST(SUBSTRING(@price_raw, 2) AS DECIMAL(10,2)), -- Quitamos el símbolo $
242        currency_symbol = SUBSTRING(@price_raw, 1, 1),           -- Guardamos el símbolo $
243        colour = @colour,
244        weight = @weight,
245        warehouse_id = @warehouse_id;
246
247
```

Output

Action Output

#	Time	Action	Message
30	02:58:44	CREATE TABLE products (id INT PRIMARY KEY, product_name VARCHAR(100), pri...	0 row(s) affected
31	03:01:37	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv' INT...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

```
247    SELECT * FROM products;
248
```

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

id	product_name	price	currency_symbol	colour	weight	warehouse_id
1	Direwolf Stannis	161.11	$	#7c7c7c	1.00	WH-4
2	Tarly Stark	9.24	$	#919191	2.00	WH-3
3	duel tourney Lannister	171.13	$	#d8d8d8	1.50	WH-2
4	warden south duel	71.89	$	#111111	3.00	WH-1
5	skywalker ewok	171.22	$	#dbdbdb	3.20	WH-0

products 116

Output

Action Output

#	Time	Action	Message
31	03:01:37	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv' INT...	100 row(s) affected Records:
32	03:07:07	SELECT * FROM products	100 row(s) returned

Como las tablas products y transaction son de muchos a muchos, se creará una tabla intermedio: “transactions_product”

```
250     --  Como las tablas products y transactions son de muchos amuchos , se creará una tablaintermedio: transactions_product
251     -- Creación de la tabla transactions_product
252 •   CREATE TABLE IF NOT EXISTS transactions_product (
253         transactions_id VARCHAR(255),
254         product_id INT,
255         PRIMARY KEY (transactions_id, product_id),
256         FOREIGN KEY (transactions_id) REFERENCES transactions(id),
257         FOREIGN KEY (product_id) REFERENCES products(id)
258     );
259
```

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

transactions_id	product_id
NULL	NULL

transactions_product 117

Output

Action Output

#	Time	Action	Message
32	03:07:07	SELECT * FROM products	100 row(s) returned
33	03:09:13	CREATE TABLE IF NOT EXISTS transactions_product (transactions_id VARCHAR(255), ...	0 row(s) affected
34	03:10:18	SELECT * FROM transactions_product	0 row(s) returned

-- Carga de la tabla intermedia transactions_product:
-- Insertamos transactions_id - product_id en la tabla intermedia

```
264 •   INSERT INTO transactions_product (transactions_id, product_id)
265     SELECT
266         t.id AS transactions_id,
267         p.id AS product_id
268     FROM transactions t
269     JOIN products p
270         ON FIND_IN_SET(p.id, t.product_ids) > 0; -- Utilizamos FIND_IN_SET para buscar product_ids en la lista separada por comas
271
272 •   SELECT * FROM transactions_product;
273
```

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

transactions_id	product_id
122DC333-E19F-D629-DCD8-9C54CF1EBB9A	1
1753A288-9FC1-52E6-5C39-A1FFB97B0D3A	1
2F499B4D-4DC7-B337-010D-8B7471812A80	1
331A8A52-52D4-D323-0388-1A97C982E441	1
41A2942C-D6E1-B164-2A45-D7ED17237A3A	1

transactions_product 118

Output

Action Output

#	Time	Action	Message
34	03:10:18	SELECT * FROM transactions_product	0 row(s) returned
35	03:13:57	INSERT INTO transactions_product (transactions_id, product_id) SELECT t.id AS transacti...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
36	03:14:40	SELECT * FROM transactions_product	587 row(s) returned

-- Nivel 3, Ejercicio 1
-- Necesitamos conocer el número de veces que se ha vendido cada producto.
-- Número de veces que se ha vendido cada producto

```sql
SELECT
    product_id,
    COUNT(*) AS cantidad_vendida
FROM transactions_product
GROUP BY product_id
ORDER BY cantidad_vendida DESC;
```

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

product_id	cantidad_vendida
23	32
79	31
7	30
43	29
61	28

Result 119

Output

Action Output

#	Time	Action	Message
35	03:13:57	INSERT INTO transactions_product (transactions_id, product_id) SELECT t.id AS transacti...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
36	03:14:40	SELECT * FROM transactions_product	587 row(s) returned
37	03:18:10	SELECT product_id, COUNT(*) AS cantidad_vendida FROM transactions_product GR...	26 row(s) returned

DIAGRAMA FINAL DE LA BASE DE DATOS TRANSACTIONS_DB