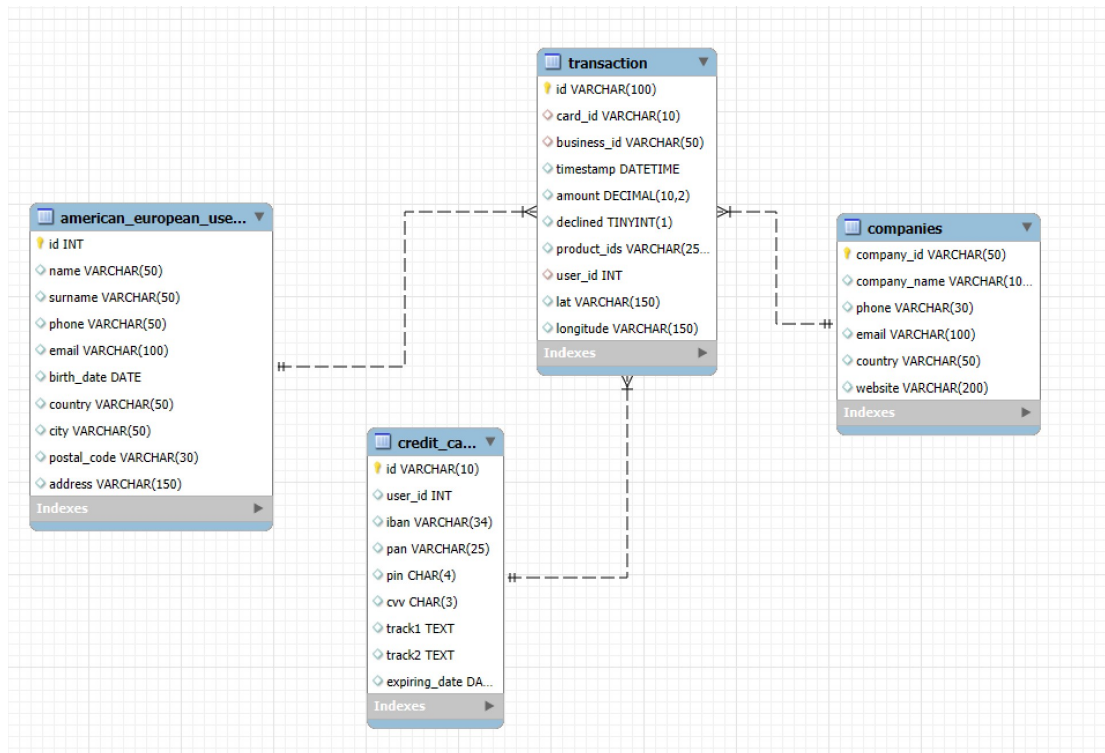


Alumna: Mariela Giménez Riveros

Crear la base de datos con sus respectivas tablas



CREACIÓN TABLAS Y CARGAR DATOS:

Tabla de "AMERICAN_USERS_RAW" he creado para tener una tabla temporal ya que "birth_date" esta en formato de string y luego creé la Tabla de american_users y insertar allí el birth_date de tipo Date.

```

CREATE TABLE american_users_raw (
    id INT,
    name VARCHAR(100),
    surname VARCHAR(100),
    phone VARCHAR(50),
    email VARCHAR(150),
    birth_date VARCHAR(50), -- formato tipo 'Nov 17, 1985'
    country VARCHAR(100),
    city VARCHAR(100),
    postal_code VARCHAR(20),
    address VARCHAR(255)
);

#cargar datos...
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.4/Uploads/american_users.csv'
INTO TABLE american_users_raw
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

```

```

CREATE TABLE american_users (
    id INT,
    name VARCHAR(100),
    surname VARCHAR(100),
    phone VARCHAR(50),
    email VARCHAR(150),
    birth_date DATE,
    country VARCHAR(100),
    city VARCHAR(100),
    postal_code VARCHAR(20),
    address VARCHAR(255)
);

#Insertar los datos de american_users_raw a american_users
INSERT INTO american_users
SELECT
    id,
    name,
    surname,
    phone,
    email,
    STR_TO_DATE(birth_date, '%b %d, %Y') AS birth_date,
    country,
    city,
    postal_code,
    address
FROM american_users_raw;

```

2. Crear la Tabla EUROPEAN_USERS_RAW he creado para tener una tabla temporal ya que "birth_date" esta en formato de string y luego creé la Tabla de EUROPEAN_USERS y insertar allí el birth_date de tipo Date. Utilice STR_TO_DATE() para convertir el valor de string en Date(fecha)

El '%b %e, %Y' indica el formato en el que está escrita la fecha en la cadena.

```

CREATE TABLE european_users_raw (
    id INT,
    name VARCHAR(100),
    surname VARCHAR(100),
    phone VARCHAR(50),
    email VARCHAR(150),
    birth_date VARCHAR(50), -- fecha como texto
    country VARCHAR(100),
    city VARCHAR(100),
    postal_code VARCHAR(20),
    address VARCHAR(255)
);
#cargar los datos
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.4/Uploads/european_users.csv'
INTO TABLE european_users_raw
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

```

```

CREATE TABLE european_users (
    id INT PRIMARY KEY,
    name VARCHAR(50),
    surname VARCHAR(50),
    phone VARCHAR(50),
    email VARCHAR(100),
    birth_date DATE,
    country VARCHAR(50),
    city VARCHAR(50),
    postal_code VARCHAR(30),
    address VARCHAR(150)
);

#insertar datos
INSERT INTO european_users
SELECT
    id,
    name,
    surname,
    phone,
    email,
    STR_TO_DATE(birth_date, '%b %e, %Y') AS birth_date,
    country,
    city,
    postal_code,
    address
FROM european_users_raw;

```

Fusionar las tablas temporales a la tabla AMERICAN_EUROPEAN_USERS porque tienen las mismas columnas y era innecesario tener tablas por separado. Para añadir los valores a la tabla Utilice la función INSERT().

```

CREATE TABLE american_european_users (
    id INT PRIMARY KEY,
    name VARCHAR(50),
    surname VARCHAR(50),
    phone VARCHAR(50),
    email VARCHAR(100),
    birth_date DATE,
    country VARCHAR(50),
    city VARCHAR(50),
    postal_code VARCHAR(30),
    address VARCHAR(150)
);

-- 3. Insertar ambos en AMERICAN_EUROPEAN_USERS:
INSERT INTO american_european_users
SELECT * FROM american_users;

INSERT INTO american_european_users
SELECT * FROM european_users;

select count(*) from american_european_users;

```

CREANDO LA TABLA CREDIT_CARD_RAW temporal porque la fecha esta de tipo cadena y haré lo mismo que las tablas anteriores, luego se inserta en la tabla CREDIT_CARD

```

CREATE TABLE credit_card_raw (
    id VARCHAR(10),
    user_id INT,
    iban VARCHAR(34),
    pan VARCHAR(25),
    pin CHAR(4),
    cvv CHAR(3),
    track1 TEXT,
    track2 TEXT,
    expiring_date VARCHAR(10) -- fecha como texto, ej: 10/30/22
);

#cargar datos
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.4/Uploads/credit_cards.csv'
INTO TABLE credit_card_raw
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

```

```

CREATE TABLE credit_card (
    id VARCHAR(10) PRIMARY KEY,
    user_id INT,
    iban VARCHAR(34),
    pan VARCHAR(25),
    pin CHAR(4),
    cvv CHAR(3),
    track1 TEXT,
    track2 TEXT,
    expiring_date DATE
);

INSERT INTO credit_card
SELECT
    id,
    user_id,
    iban,
    pan,
    pin,
    cvv,
    track1,
    track2,
    STR_TO_DATE(expiring_date, '%m/%d/%y') AS expiring_date
FROM credit_card_raw;

```

CREANDO LA TABLA COMPANIES y cargar datos

```

CREATE TABLE companies (
    company_id VARCHAR(50) PRIMARY KEY,
    company_name VARCHAR(100),
    phone VARCHAR(30),
    email VARCHAR(100),
    country VARCHAR(50),
    website VARCHAR(200)
);

#cargar datos
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.4/Uploads/companies.csv'
INTO TABLE companies
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

```

CREANDO LA TABLA DE TRANSACTION

```
CREATE TABLE transaction (  
    id VARCHAR(100) PRIMARY KEY,  
    card_id VARCHAR(10),  
    business_id VARCHAR(50),  
    timestamp DATETIME,  
    amount DECIMAL(10,2),  
    declined BOOLEAN,  
    product_ids VARCHAR(255),  
    user_id INT,  
    lat VARCHAR(150),  
    longitude VARCHAR(150),  
    FOREIGN KEY (card_id) REFERENCES credit_card(id),  
    FOREIGN KEY (business_id) REFERENCES companies(company_id),  
    FOREIGN KEY (user_id) REFERENCES american_european_users(id)  
);  
  
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.4/Uploads/transactions.csv'  
INTO TABLE transaction  
FIELDS TERMINATED BY ';'   
ENCLOSED BY ''''  
LINES TERMINATED BY '\n'  
IGNORE 1 ROWS;
```

EJERCICIO 1

1 Realizar una subconsulta que muestre todas los usuarios con más de 80 transacciones utilizando al menos 2 tablas.


```

3 • SELECT u.name
4 FROM american_european_users u
5 WHERE id IN (
6     SELECT COUNT(t.user_id) AS total_transactions
7     FROM transaction t
8     #where t.amount >= 80
9     GROUP BY t.user_id
10    having total_transactions >= 80
11    order by total_transactions desc
12 ) ;

```

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

name
Acton
Serina
Kim
Neil

american_european_users 135 x

Output

Action Output

#	Time	Action	Message
1	23:37:16	SELECT u.name FROM american_european_users u WHERE id IN (SELECT COUNT(t.user_id) AS total...	4 row(s) returned

Ejercicio 2.

2.1- Muestra la mediana de amount por IBAN de las tarjetas de creditos a la compa a DONEC LTD. Utiliza al menos 2 tablas.

```

16 • SELECT
17     cc.iban, round(AVG(t.amount),2) as mediana_iban
18 FROM transaction t
19 JOIN credit_card cc ON t.card_id = cc.id
20 JOIN companies c ON t.business_id = c.company_id
21 WHERE c.company_name = 'Donec Ltd'
22 group by cc.iban
23 #having round(AVG(t.amount),2)
24 ;

```

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

iban	mediana_iban
XX911406401125586307586805	356.25
SK9446370242474562577506	142.96
XX776752917845952975555640	257.37
XX413827362289719304908990	139.59
XX347787246070769610780308	240.41

Result 136 x

Output

Action Output

#	Time	Action	Message
1	23:39:02	SELECT cc.iban, round(AVG(t.amount),2) as mediana_iban FROM transaction t JOIN credit_card cc ON t...	371 row(s) returned

Nivell 2

Crea una nueva tabla que refleje el estado de las tarjetas de creditos basada en las ultimas 3 transacciones van a ser declinadas y genera la siguiente consulta:

```
CREATE TABLE card_status (  
    card_id varchar(10) PRIMARY KEY,  
    status VARCHAR(10),  
    FOREIGN KEY (card_id) REFERENCES credit_card(id)  
);
```

Insertar los datos en la tabla, como ya esta insertado he hecho un count para ver los datos que se han añadido en la tabla card_status

```
34 • INSERT INTO card_status (card_id, status)  
35 SELECT  
36     card_id AS credit_card_id,  
37     CASE  
38         WHEN SUM(CASE WHEN declined = 1 THEN 1 ELSE 0 END) = 3 THEN 'bloqueada'  
39         ELSE 'activa'  
40     END AS status  
41 FROM (  
42     SELECT  
43         card_id,  
44         declined,  
45         ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS fila  
46     FROM transaction  
47 ) t  
48 WHERE fila <= 3  
49 GROUP BY card_id;  
50  
51 • select count(*) from card_status;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
count(*)			
5000			

result 106 x

Output

Action Output

#	Time	Action	Message
1	20:05:58	select count(*) from card_status LIMIT 0, 1000	1 row(s) returned

Ejercicio 1

1.1- Cuantas tarjetas estan activas?

```
54 • select count(*) from card_status where status = "activa";
```

count(*)
4995

result 107 x

Output

Action Output

#	Time	Action	Message
1	20:10:00	select count(*) from card_status where status = "activa" LIMIT 0, 1000	1 row(s) returned

Nivel 3

3.1- Crea una tabla con la cual pueda unir los datos del nuevo archivo "Products.csv con la base de datos creada, teniendo en cuenta en cuenta que desde transaction tiene products_ids. Genera la siguiente consulta

Creando tabla temporal para cargar los datos porque el price tiene el \$

```
CREATE TABLE products_temp (
    id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price VARCHAR(20),
    colour VARCHAR(50),
    weight DECIMAL(10,2),
    warehouse_id VARCHAR(10)
);
```

```
#cargando los datos a la tabla temporal
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.4/Uploads/products.csv'
INTO TABLE products_temp
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

```

CREATE TABLE products (
    id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10,2),
    colour VARCHAR(50),
    weight DECIMAL(10,2),
    warehouse_id varchar(10)
);

#insertar los datos de la tabla temporal a la tabla de products
INSERT INTO products (id, product_name, price, colour, weight, warehouse_id)
SELECT
    id,
    product_name,
    CAST(REPLACE(price, '$', '' ) AS DECIMAL(10,2)) AS price,
    colour,
    weight,
    warehouse_id
FROM products_temp;

```

Crear la tabla de puente que conecta la transaction y products

```

98 CREATE TABLE transaction_products (
99     transaction_id varchar(100),
100     product_id INT,
101     FOREIGN KEY (transaction_id) REFERENCES transaction(id),
102     FOREIGN KEY (product_id) REFERENCES products(id),
103     PRIMARY KEY (transaction_id, product_id)
104 );
105
106 SHOW COLUMNS FROM products; # mirar los nombres y tipos de la tabla

```

Result Grid						
Filter Rows:		Export:		Wrap Cell Content: IA		
	Field	Type	Null	Key	Default	Extra
►	id	int	NO	PRI	NULL	
	product_name	varchar(100)	YES		NULL	
	price	decimal(10,2)	YES		NULL	
	colour	varchar(50)	YES		NULL	
	weight	decimal(10,2)	YES		NULL	
	warehouse_id	varchar(10)	YES		NULL	

insertar los datos en la tabla puente y limpiar los datos de products ya que es una lista y puede haber espacios entre medios, he optado por usar la función JSON_TABLE puedes usar la función JSON_TABLE para convertir datos JSON para poder leer mejor, lo que

facilita su consulta y manipulación.

```
INSERT INTO transaction_products (transaction_id, product_id)
SELECT
    t.id AS transaction_id,
    CAST(j.value AS UNSIGNED) AS product_id
FROM transaction t,
JSON_TABLE(
    CONCAT(
        '["',
        REPLACE(REPLACE(t.product_ids, ' ', ''), ', ', '"', ''),
        '"]'
    ),
    '$[*]' COLUMNS (value VARCHAR(10) PATH '$')
) AS j;
```

Ejercicio 1

3.1- Necesitamos conocer el numero de veces que se han vendido cada producto.

```
125 • select p.id, count(pt.product_id) as product_sold, p.product_name
126 from transaction_products pt join products p
127 on pt.product_id = p.id
128 group by p.id, p.product_name
129 order by product_sold Desc;
130
131 • SHOW COLUMNS FROM transaction_products; # mirar los nombres y tipos de la tabla
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
id	product_sold	product_name			
52	2654	riverlands the duel			
29	2635	Tully maester Tarly			
21	2609	duel Direwolf			
16	2608	the duel warden			
66	2601	mustafar jinn			
87	2598	sith Jade			
33	2597	duel warden			
48	2597	rock Renly in			
23	2593	riverlands north			
68	2589	Stark Karstark			
88	2587	Stannis warden so...			
4	2584	warden south duel			

result 109 x

Output

Action Output

#	Time	Action	Message
1	20:31:28	select p.id, count(pt.product_id) as product_sold, p.product_name from transaction_products pt join products ...	100 row(s) returned