
SPRINT 4 Tarea S4.01. Creación de Base de Datos

NIVEL 1

“Descarga los archivos CSV, estudíalos 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:”

Primer paso, creo la base de datos sobre la que vamos a trabajar, la llamaré operaciones.

```
1 CREATE DATABASE if not exists operaciones;
2
3 • USE operaciones;
4
```

#	Time	Action	Message
3	13:13:41	CREATE DATABASE if not exists operaciones	1 row(s) affected
4	13:14:06	USE operaciones	0 row(s) affected

Paso a crear la tabla de usuarios general (users_gral), que tendrá los datos de los usuarios de los distintos países. Es importante mencionar que las tablas user_uk, user_usa y user_ca comparten la misma estructura, lo que justifica una tabla unificada.

```
7 • CREATE TABLE IF NOT EXISTS users_gral (
8     id INT PRIMARY KEY,
9     name VARCHAR(100),
10    surname VARCHAR(100),
11    phone VARCHAR(150),
12    email VARCHAR(150),
13    birth_date VARCHAR(100),
14    country VARCHAR(150),
15    city VARCHAR(150),
16    postal_code VARCHAR(100),
17    address VARCHAR(255));
18
```

Time	SQL
17:47:11	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv' INTO TABLE users_gral FIELDS TERMINATED BY ';' ENCLOSED BY ''
17:44:48	CREATE TABLE IF NOT EXISTS users_gral (id INT PRIMARY KEY, name VARCHAR(100), surname VARCHAR(100), phone VARCHAR(150), email VARCHAR(150), birth_date VARCHAR(100), country VARCHAR(150), city VARCHAR(150), postal_code VARCHAR(100), address VARCHAR(255));

Creación de la estructura de las tablas ***credit_cards, products, companies***

```
CREATE TABLE IF NOT EXISTS credit_cards (  
  id VARCHAR(15) PRIMARY KEY,  
  user_id INT,  
  iban VARCHAR(80),  
  pan VARCHAR(80),  
  pin CHAR(4),  
  cvv CHAR(4),  
  track1 VARCHAR(70),  
  track2 VARCHAR(70),  
  expiring_date VARCHAR(80),  
  FOREIGN KEY (user_id) REFERENCES users_gral(id)  
);
```

```
CREATE TABLE IF NOT EXISTS products (  
  id VARCHAR(50) PRIMARY KEY,  
  product_name VARCHAR(50),  
  price DECIMAL(8,2),  
  colour CHAR(7),  
  weight SMALLINT,  
  warehouse_id VARCHAR(10));
```

```
100 CREATE TABLE IF NOT EXISTS companies (  
101   company_id VARCHAR(15) PRIMARY KEY,  
102   company_name VARCHAR(255),  
103   phone VARCHAR(15),  
104   email VARCHAR(100),  
105   country VARCHAR(100),  
106   website VARCHAR(255)  
107 );  
108
```

Paso a crear la tabla de hechos (tabla de transacciones)

```
89 CREATE TABLE IF NOT EXISTS transacciones (  
90     id VARCHAR(255) PRIMARY KEY,  
91     card_id VARCHAR(15) REFERENCES credit_cards(id),  
92     business_id VARCHAR(15) REFERENCES companies(company_id),  
93     timestamp TIMESTAMP,  
94     amount DECIMAL(10, 2),  
95     declined BOOLEAN,  
96     product_ids VARCHAR(50),  
97     user_id INT REFERENCES users_gral(id),  
98     lat FLOAT,  
99     longitude FLOAT  
100 );
```

Output

#	Time	Action	Message
14	12:09:25	CREATE TABLE IF NOT EXISTS companies (company_id VARCHAR(15) PRIMARY KEY, company_nam...	0 row(s) affected
15	12:12:49	CREATE TABLE IF NOT EXISTS transacciones (id VARCHAR(255) PRIMARY KEY, card_id VARCHAR(1...	0 row(s) affected

Me he dado cuenta más tarde, luego de haber creado la tabla transacciones, de que no había creado las claves foráneas. Por lo que paso a establecer las claves foráneas que harán de vínculo con las tablas de dimensiones respectivas (tabla companies, tabla users_gral y tabla credit_cards).

```
123 ###--- agregar clave foranea, me olvide de agregarlas cuando crear la tabla transacciones  
124  
125 ALTER TABLE transacciones  
126 ADD CONSTRAINT fk_card_id  
127 FOREIGN KEY (card_id) REFERENCES credit_cards(id),  
128 ADD CONSTRAINT fk_business_id  
129 FOREIGN KEY (business_id) REFERENCES companies(company_id);
```

Ahora para poder ingresar los datos en formato CSV, aplico **SHOW VARIABLES LIKE 'secure_file_priv'**; para que me pueda aparecer la ruta de acceso en donde debo poner los archivos para luego poder cargar los datos. Esa ruta de acceso la copio para utilizarla en la línea **LOAD DATA INFILE**.

Carga datos tabla **companies**

```
103 SHOW VARIABLES LIKE 'secure_file_priv';  
104  
105 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv'  
106 INTO TABLE companies  
107 FIELDS TERMINATED BY ','  
108 ENCLOSED BY ''''  
109 LINES TERMINATED BY '\n'  
110 IGNORE 1 ROWS;  
111  
112 select *  
113 from companies;
```

Result Grid

company_id	company_name	phone	email	country	website
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porrtitor.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
b-2254	Nascetur Ridiculus Mus Inc.	06 26 87 61 84	suspendisse.dui@icloud.net	United States	https://ebay.com/sub

companies 2 x

Output

#	Time	Action	Message
20	13:14:10	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv' INTO TABLE ...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
21	13:14:25	select * from companies	100 row(s) returned

Cargar los datos de usuarios en tabla **users_gral** (de los diferentes archivos de user como tienen la misma estructura, lo dejo unificado en una tabla de users_gral)

```
57 • SHOW VARIABLES LIKE 'secure_file_priv'; ### --- para poder ver la carpeta donde instalar los archivos.
58
59 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv'
60 INTO TABLE users_gral
61 FIELDS TERMINATED BY ','
62 ENCLOSED BY '"'
63 LINES TERMINATED BY '\r\n'
64 IGNORE 1 ROWS;
65
66 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv'
67 INTO TABLE users_gral
68 FIELDS TERMINATED BY ','
69 ENCLOSED BY '"'
70 LINES TERMINATED BY '\r\n'
71 IGNORE 1 ROWS;
72
73 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv'
74 INTO TABLE users_gral
75 FIELDS TERMINATED BY ','
76 ENCLOSED BY '"'
77 LINES TERMINATED BY '\r\n'
78 IGNORE 1 ROWS;
```

#	Time	Action	Message
42	17:49:56	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv' INTO TABLE us...	50 row(s) affected Records: 50 Deleted: 0
43	17:50:13	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv' INTO TABLE u...	150 row(s) affected Records: 150 Deleted: 0

Respecto a la carga de dato de la tabla **products**, al tener la columna price formato decimal, y el archivo figurar con símbolo “\$”, se aplica el comando: **SET price = REPLACE(@price, '\$', '')** para que me limpie ese carácter y se pueda cargar el tipo de dato como decimal.

```
55 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv'
56 INTO TABLE products
57 FIELDS TERMINATED BY ','
58 ENCLOSED BY '"'
59 LINES TERMINATED BY '\n'
60 IGNORE 1 ROWS
61 (id, product_name, @price, colour, weight, warehouse_id)
62 SET price = REPLACE(@price, '$', '');
63
```

#	Time	Action	Message
56	18:15:12	select * from credit_cards	275 row(s) returned
57	18:16:38	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv' INTO TABLE pro...	Error Code: 1366. Incorrect decimal value: '\$161.11' for column 'price' at row 1
58	18:53:17	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv' INTO TABLE pro...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

Paso a cargar los datos de tabla **transacciones**, en este caso tengo que poner delimitador el “punto y coma”, ya que el archivo CSV, tiene como carácter separador el “punto y coma”.

```
64 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'
65 INTO TABLE transacciones
66 FIELDS TERMINATED BY ';'
67 ENCLOSED BY '"'
68 LINES TERMINATED BY '\n'
69 IGNORE 1 ROWS;
```

#	Time	Action	Message
61	18:58:16	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv' INTO TABLE...	Error Code: 1261. Row 1 doesn't contain data for all columns
62	19:04:57	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv' INTO TABLE...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0

- EJERCICIO 1

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

```
126 • SELECT distinct t.user_id, u.name, u.surname, count(user_id) AS nro_transacc
127 FROM transacciones t
128 JOIN users_gral as u
129 ON t.user_id = u.id
130 WHERE t.user_id IN (
131     SELECT t1.user_id
132     FROM transacciones AS t1
133     GROUP BY t1.user_id
134     HAVING COUNT(user_id) >= 30
135 )
136 group by t.user_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

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

Result 13 x

Output

Action Output

#	Time	Action	Message
1	13:24:38	SELECT distinct t.user_id, u.name, u.surname, count(user_id) AS nro_transacc FROM transacciones t JOIN us...	4 row(s) returned

- EJERCICIO 2

“Muestra la media de amount por IBAN de las tarjetas de crédito a la compañía Donec Ltd, utiliza al menos 2 tablas.”

```
92 • SELECT c.id, c.iban, AVG(t.amount)
93 FROM transacciones AS t
94 JOIN credit_cards AS c
95 ON t.card_id = c.id
96 WHERE t.business_id = (
97     SELECT c1.company_id
98     FROM companies as c1
99     WHERE c1.company_name = "Donec Ltd")
100 group by c.id, c.iban;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

id	iban	AVG(t.amount)
CdU-2973	PT87806228135092429456346	203.715000

Result 63 x

Output

Action Output

#	Time	Action	Message
1	14:35:44	SELECT c.id, c.iban, AVG(t.amount) FROM transacciones AS t JOIN credit_cards AS c ON t	

NIVEL 2

“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:”

Para crear una nueva tabla, opto por duplicar la tabla transacciones ya existente, y comenzar a trabajar sobre la nueva tabla duplicada.

```
35 • CREATE TABLE transaccionesv2 AS SELECT * FROM transacciones;
36
```

Output

#	Time	Action	Message
90	11:44:56	describe transacciones	10 row(s) returned
91	20:18:53	CREATE TABLE transaccionesv2 AS SELECT * FROM transacciones	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

Agrego las claves primarias y secundarias a esta nueva tabla de transacciones.

```
36
37 • ALTER TABLE transaccionesv2
38   ADD PRIMARY KEY (id),
39   ADD CONSTRAINT fk_card_id_v2
40   FOREIGN KEY (card_id) REFERENCES credit_cards(id),
41   ADD CONSTRAINT fk_business_id_v2
42   FOREIGN KEY (business_id) REFERENCES companies(company_id),
43   ADD CONSTRAINT fk_user_id_v2
44   FOREIGN KEY (user_id) REFERENCES users_gral(id);
45
```

Output

#	Time	Action	Message
8	21:24:27	ALTER TABLE transaccionesv2 ADD PRIMARY KEY (id), ADD CONSTRAINT fk_card_id FOREIGN KEY (... Error Code: 1826. Duplicate foreign key constraint name 'fk_card_id'	
9	21:27:53	ALTER TABLE transaccionesv2 ADD PRIMARY KEY (id), ADD CONSTRAINT fk_card_id_v2 FOREIGN K...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

- EJERCICIO 1

¿Cuántas tarjetas están activas?

Y para verificar si la consulta cumple con el requerimiento solicitado

```
79 • WITH clasificacion AS(
80   SELECT card_id,
81   SUM(declined) OVER(PARTITION BY card_id ORDER BY timestamp ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS Acum
82   FROM transaccionesv2)
83   SELECT card_id, count(DISTINCT card_id) as Recuento
84   FROM clasificacion
85   WHERE card_id NOT IN (
86     SELECT card_id
87     FROM clasificacion
88     GROUP BY card_id
89     HAVING MAX(Acum) >= 3)
90   group by card_id ;
```

Result Grid

card_id	Recuento
CdU-2938	1
CdU-2945	1
CdU-2952	1
CdU-2959	1
CdU-2966	1

Result 95 x

Output

#	Time	Action	Message
1	12:55:53	WITH clasificacion AS(SELECT card_id, SUM(declined) OVER(PARTITION BY card_id ORDER BY timesta...	275 row(s) returned

He agregado otros registros a la tabla transaccionesv2, de forma tal que el acumulado de decline llegara a 3, y poder comprobar si la consulta me aparta a un lado dicha credit_card, y así ha sido.

```
58 • INSERT INTO transaccionesv2
59 (id, card_id, business_id, timestamp, amount, declined, product_ids, user_id, lat, longitude)
60 VALUES
61 (
62 '69090229-AD26-43C3-SAAF-8D332D383E63', 'CcU-2959', 'b-2362', '2022-03-17 14:01:00', 100, 1, '67, 29', 92, 40753650688, 1297231412224),
63 (
64 '69090229-AD26-43C3-SAAF-8D332D383E64', 'CcU-2959', 'b-2362', '2022-03-18 14:01:00', 100, 1, '67, 29', 92, 40753650688, 1297231412224),
65 (
66 '69090229-AD26-43C3-SAAF-8D332D383E65', 'CcU-2959', 'b-2362', '2022-03-19 14:01:00', 100, 1, '67, 29', 92, 40753650688, 1297231412224);
```

Ejemplo, al agregar nuevas operaciones declined a la credit_card CcU-2959, supera las 3 operaciones declined, y luego de ejecutar la consulta, no la considera, ya la deja afuera del resultado final.

```
74 • WITH clasificacion AS(
75     SELECT card_id,
76            SUM(declined) OVER(PARTITION BY card_id ORDER BY timestamp ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS Acum
77     FROM transaccionesv2)
78 SELECT card_id, count(DISTINCT card_id)
79 FROM clasificacion
80 WHERE card_id NOT IN (
81     SELECT card_id
82     FROM clasificacion
83     GROUP BY card_id
84     HAVING MAX(Acum) >= 3)
85 group by card_id ;
```

Result Grid

card_id	count(DISTINCT card_id)
CcU-2938	1
CcU-2945	1
CcU-2952	1
CcU-2966	1
CcU-2973	1

Result 92 x

Output

Action Output

#	Time	Action	Message
1	12:44:14	WITH clasificacion AS(SELECT card_id, SUM(declined) OVER(PARTITION BY card_id ORDER BY timesta...	274 row(s) returned