

SPRINT 4

Lo primero que se nos pide que creamos una base de datos con esquema de estrella en la cual al menos 4 tablas pueden realizar diversas consultas

Table: companies

Columns:

Column Name	Data Type	Constraints
company_id	varchar(20)	PK
company_name	varchar(100)	
phone	varchar(20)	
email	varchar(100)	
country	varchar(50)	
website	varchar(255)	

```

1  CREATE TABLE Users (
2      id INT AUTO_INCREMENT PRIMARY KEY,
3      name VARCHAR(50),
4      surname VARCHAR(50),
5      phone VARCHAR(20),
6      email VARCHAR(100),
7      birth_date DATE,
8      country VARCHAR(50),
9      city VARCHAR(50),
10     postal_code VARCHAR(10),
11     address VARCHAR(255)
12 );
13 CREATE TABLE Companies (
14     company_id VARCHAR(20) PRIMARY KEY,
15     company_name VARCHAR(100),
16     phone VARCHAR(20),
17     email VARCHAR(100),
18     country VARCHAR(50),
19     website VARCHAR(255)
20 );
21 CREATE TABLE Credit_Cards (
22     id VARCHAR(20) PRIMARY KEY,
23     user_id INT,
24     iban VARCHAR(34),
25     pan VARCHAR(20),
26     pin INT,
27     cvv INT,
28     track1 VARCHAR(100),
29     track2 VARCHAR(100),
30     expiring_date TEXT,
31     CONSTRAINT fk_user_card FOREIGN KEY (user_id) REFERENCES Users(id)
32 );
33 CREATE TABLE Products (
34     id INT PRIMARY KEY,
35     product_name TEXT,
36     price TEXT,
37     colour TEXT,
38     weight DOUBLE,
39     warehouse_id TEXT
40 );
41 CREATE TABLE Transactions (
42     id CHAR(36) PRIMARY KEY,
43     card_id VARCHAR(20),
44     business_id VARCHAR(20),
45     timestamp DATETIME,
46     amount DOUBLE,
47     declined INT,
48     product_ids TEXT,
49     user_id INT,
50     lat DOUBLE,
51     longitude DOUBLE,
52     CONSTRAINT fk_card FOREIGN KEY (card_id) REFERENCES Credit_Cards(id),
53     CONSTRAINT fk_business FOREIGN KEY (business_id) REFERENCES Companies(company_id),
54     CONSTRAINT fk_user FOREIGN KEY (user_id) REFERENCES Users(id)
55 );
56 UPDATE Users
57 SET birth_date = STR_TO_DATE(birth_date, '%b %d, %Y');
58

```

Esta fue la estructura que he escogido para hacer mi base de datos y está compuesta por 5 tablas en la cual la principal ("transactions" tabla de hechos) la cual tiene datos clave y referencias a las otras tablas.

Alvaro Matias Santos Fuentes

Después de la creación de las tablas lo que hice para poder cargar los datos fue mediante Wizard import data ya que me fue imposible modificar los datos .imi específicamente la línea de secure

Las demás tablas (transactions, products, users, companies) son tablas de dimensiones en las cuales se almacenan diferentes parámetros y se clasifican según la aplicación que se le quiera dar.

La tabla credit_card_status es una tabla que cree a posteriori por eso no la tengo en cuenta

para poder cargar los datos tuve que modificar una columna de users

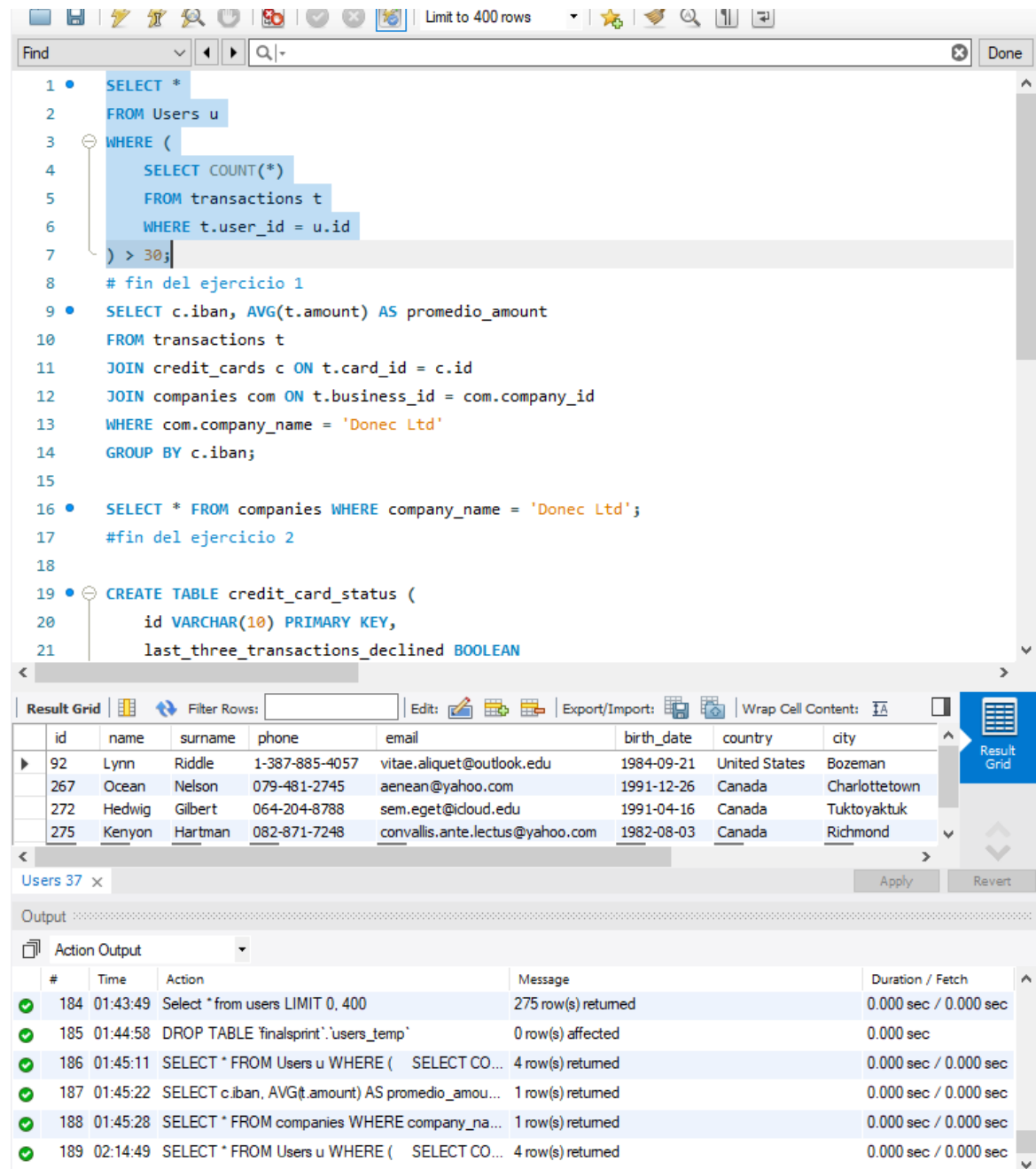
UPDATE Users

SET birth_date = STR_TO_DATE(birth_date, '%b %d, %Y')

para que pudiera entender los datos de fechas que teníamos dentro de los datos a carga

EJERCICIO 1

En este ejercicio se nos pide realizar una subconsulta en la cual muestre a todos los usuarios con mas de 30 transacciones utilizando al menos dos tablas.



The screenshot shows a SQL IDE interface. The top part is a query editor with the following SQL code:

```

1 • SELECT *
2   FROM Users u
3  WHERE (
4      SELECT COUNT(*)
5      FROM transactions t
6      WHERE t.user_id = u.id
7  ) > 30;
8  # fin del ejercicio 1
9 • SELECT c.iban, AVG(t.amount) AS promedio_amount
10  FROM transactions t
11  JOIN credit_cards c ON t.card_id = c.id
12  JOIN companies com ON t.business_id = com.company_id
13  WHERE com.company_name = 'Donec Ltd'
14  GROUP BY c.iban;
15
16 • SELECT * FROM companies WHERE company_name = 'Donec Ltd';
17 #fin del ejercicio 2
18
19 • CREATE TABLE credit_card_status (
20     id VARCHAR(10) PRIMARY KEY,
21     last_three_transactions_declined BOOLEAN

```

Below the query editor is a "Result Grid" showing the results of the first query. The grid has columns: id, name, surname, phone, email, birth_date, country, city. The results are:

id	name	surname	phone	email	birth_date	country	city
92	Lynn	Riddle	1-387-885-4057	vitae.aliquet@outlook.edu	1984-09-21	United States	Bozeman
267	Ocean	Nelson	079-481-2745	aenean@yahoo.com	1991-12-26	Canada	Charlottetown
272	Hedwig	Gilbert	064-204-8788	sem.eget@idoud.edu	1991-04-16	Canada	Tuktoyaktuk
275	Kenyon	Hartman	082-871-7248	convallis.ante.lectus@yahoo.com	1982-08-03	Canada	Richmond

Below the result grid is an "Output" section showing the execution log. The log contains the following entries:

#	Time	Action	Message	Duration / Fetch
184	01:43:49	Select * from users LIMIT 0, 400	275 row(s) returned	0.000 sec / 0.000 sec
185	01:44:58	DROP TABLE 'finalsprint'. 'users_temp'	0 row(s) affected	0.000 sec
186	01:45:11	SELECT * FROM Users u WHERE (SELECT CO...	4 row(s) returned	0.000 sec / 0.000 sec
187	01:45:22	SELECT c.iban, AVG(t.amount) AS promedio_amou...	1 row(s) returned	0.000 sec / 0.000 sec
188	01:45:28	SELECT * FROM companies WHERE company_na...	1 row(s) returned	0.000 sec / 0.000 sec
189	02:14:49	SELECT * FROM Users u WHERE (SELECT CO...	4 row(s) returned	0.000 sec / 0.000 sec

Lo que hice en este ejercicio fue usar la función SELECT y que seleccioné todas las columnas de la misma y delante de la cláusula WHERE y una subconsulta en la que le pedimos que nos de los resultados que tengan el mismo id en las dos tablas y también mas de 30 transacciones.

Ejercicio 2

En este ejercicio se nos pide que mostremos la media de amount por iban de las tarjetas de crédito de la compañía Donec Ltd.

Query 1 SQL File 1*

Limit to 400 rows

Find

```

4      SELECT COUNT(*)
5      FROM transactions t
6      WHERE t.user_id = u.id
7  ) > 30;
8  # fin del ejercicio 1
9  • SELECT c.iban, AVG(t.amount) AS promedio_amount
10 FROM transactions t
11 JOIN credit_cards c ON t.card_id = c.id
12 JOIN companies com ON t.business_id = com.company_id
13 WHERE com.company_name = 'Donec Ltd'
14 GROUP BY c.iban;
15
16 • SELECT * FROM companies WHERE company_name = 'Donec Ltd';
17 #fin del ejercicio 2
18
19 • CREATE TABLE credit_card_status (
20     id VARCHAR(10) PRIMARY KEY,
21     last_three_transactions_declined BOOLEAN
22 );
23 -- Primero, seleccionamos el ID de la tarjeta de crédito y evaluamos si las última
24 -- Luego, insertamos los resultados en la tabla credit_card_status.

```

Result Grid

iban	promedio_amount
PT87806228135092429456346	203.715

Result 40

Output

Action Output

#	Time	Action	Message
✓ 187	01:45:22	SELECT c.iban, AVG(t.amount) AS promedio_amou...	1 row(s) returned
✓ 188	01:45:28	SELECT * FROM companies WHERE company_na...	1 row(s) returned
✓ 189	02:14:49	SELECT * FROM Users u WHERE (SELECT CO...	4 row(s) returned
✓ 190	02:20:55	SELECT c.iban, AVG(t.amount) AS promedio_amou...	1 row(s) returned
✓ 191	02:22:52	SELECT * FROM companies WHERE company_na...	1 row(s) returned
✓ 192	02:23:04	SELECT c.iban, AVG(t.amount) AS promedio_amou...	1 row(s) returned

Lo primero que hice fue mediante la función SELECT llamar a Credit_cards.iban, después usamos la función AVG(transactions amount) y a este se le dará el nombre de promedio_amount

A continuación, mediante la función FROM llamamos a Transactions y le damos el alias de T

la información saldrá de TRANSACTIONS y mediante Join uniremos la información de las tablas CREDIT_CARD a la cual se le dará el alias de C y COMPANY a la cual se le dará el ALIAS de COM la correlación que tienen estas tablas es el id para que nos dé el iban de uno y la media del otro.

Después mediante la cláusula WHERE especificamos que esta consulta solo nos de los resultados del nombre de compañía especificado que en este caso es: Donec Ltd

Solución: iban PT87806228135092429456346', y este es el promedio de amount del iban de las tarjetas de la empresa '203.715'