

SPRINT 3

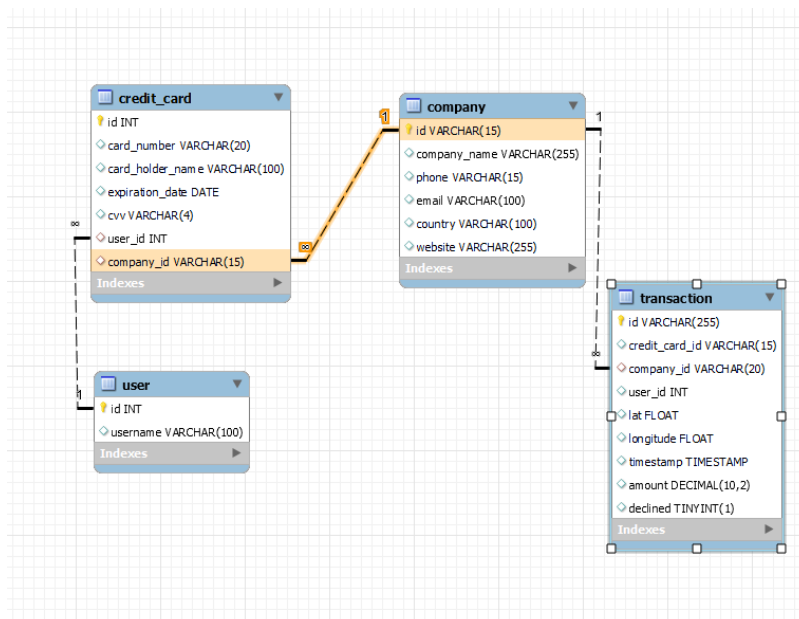
EJERCICIO 1

En este ejercicio lo que se pide es que creamos una tabla llamada `credit_card` y que almacene detalles cruciales sobre la tarjeta de crédito y luego hacer un diagrama en el cual se vea la interrelación entre ellas.

Lo primero que hice fue crear la tabla

Tables_in_transactions
company
credit_card
transaction
user

Luego hice el diagrama donde se ven las interacciones entre las tablas



EJERCICIO 2

En este ejercicio lo que tenemos que hacer es corregir el error de un ID en el cual la información que se muestra es errónea, ID CcU-2938 y debería contener R323456312213576817699999

Lo que hice fue buscar la información que contiene el ID CcU-2938

Al buscarlo me aparece información errónea en el iban y lo tenemos cambiar por el numero que recursos humanos nos ha dado

The screenshot shows a database management interface. On the left, a sidebar lists database objects: 'pany', 'it_card', 'Columns', 'id', 'iban', and 'schemas'. The main area displays a SQL script with the following lines:

```
1 • show tables
2 ;
3 #Fin del ejercicio 1
4 • select *
5 From credit_card where id= "CcU-2938";
6 • UPDATE credit_card
7 SET iban = 'R323456312213576817699999'
8 WHERE id = 'CcU-2938';
9
```

Below the script, a 'Result Grid' is shown with the following data:

	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	R323456312213576817699999	5424465566813633	3257	984	10/30/22
*	NULL	NULL	NULL	NULL	NULL	NULL

Lo primero que hice fue comprobar los datos que tenía ese ID y al comprobarlo el IBAN que me aparecía en la columna no correspondía con el iban que nos dieron desde recursos humanos,

select * From credit_card where id= "CcU-2938";

este codigo lo que hace es buscar el ID en especifico que en este caso es "CcU-2938" dentro la tabla **credit_card**

Luego al comprobar que los datos si estaban erróneos, los corregimos y mediante el código

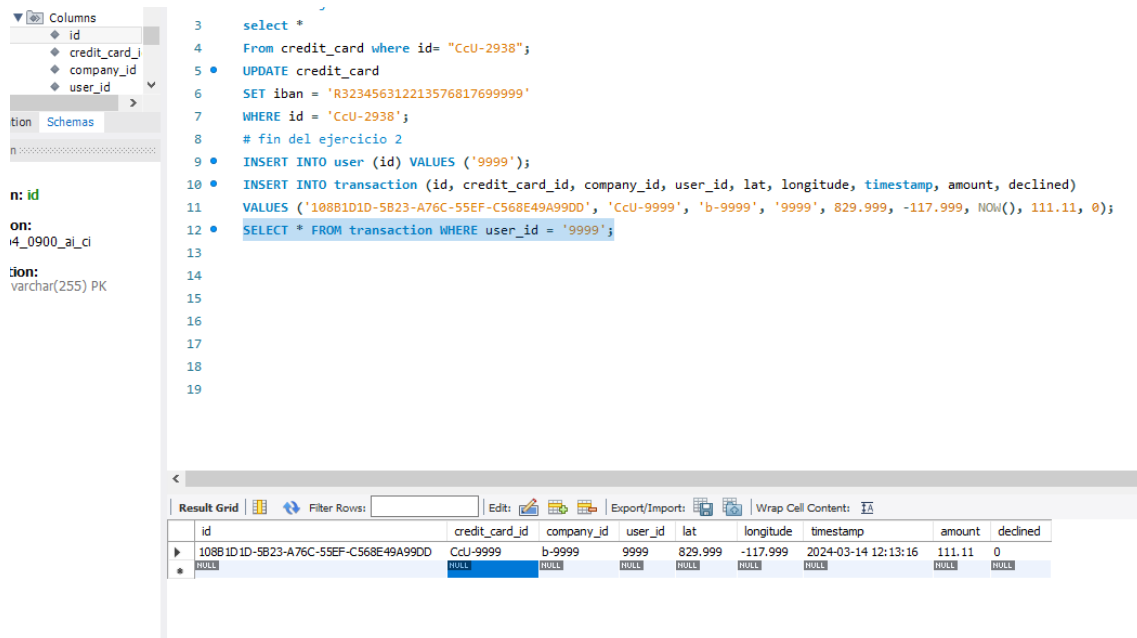
UPDATE credit_card SET iban = 'R323456312213576817699999' WHERE id = 'CcU-2938';

Este codigo lo que hace es hacer un cambio en la columna IBAN en la cual coincida el ID 'CcU-2938'

RESULTADO: como se puede comprobar en la foto los cambios han sido realizados correctamente, el iban ha sido reemplazado y solucionado el problema con recursos humanos

EJERCICIO 3

En este ejercicio lo que se pide es que ingresemos un usuario nuevo con sus correspondientes datos.



The screenshot shows a database management tool interface. On the left, there's a 'Columns' panel with a tree view showing 'id', 'credit_card_id', 'company_id', and 'user_id'. Below it, a 'Schemas' panel shows a table 'transaction' with columns 'id', 'credit_card_id', 'company_id', 'user_id', 'lat', 'longitude', 'timestamp', 'amount', and 'declined'. The main area displays SQL queries:

```

3 select *
4 from credit_card where id= "CcU-2938";
5 UPDATE credit_card
6 SET iban = 'R323456312213576817699999'
7 WHERE id = 'CcU-2938';
8 # fin del ejercicio 2
9 INSERT INTO user (id) VALUES ('9999');
10 INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined)
11 VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A990D', 'CcU-9999', 'b-9999', '9999', 829.999, -117.999, NOW(), 111.11, 0);
12 SELECT * FROM transaction WHERE user_id = '9999';
13
14
15
16
17
18
19

```

Below the queries, a 'Result Grid' is shown with the following data:

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶	108B1D1D-5B23-A76C-55EF-C568E49A990D	CcU-9999	b-9999	9999	829.999	-117.999	2024-03-14 12:13:16	111.11	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Para poder hacerlo tuve que modificar la estructura de datos ya que como la tenía había conflictos entre tablas, tuve que cambiar el valor de la columna Id de USER que INT por VARCHAR , después tuve que crearle un usuario en la tabla USER ya que tenía un estatus de convivencia en le cual si no esta la información de usuario en USER no deja cargar el resto de información en las demás tablas.

Luego mediante la función INSERT INTO TRANSACTION añadimos todas las columnas que queremos cambiar y mediante la función VALUES ponemos la información correspondiente a cada campo

RESULTADO: como se puede ver en la imagen se ha cargado la información de la tabla como tenía que ser

EJERCICIO 4

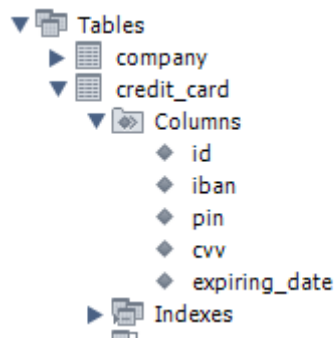
Desde recursos humanos se nos pide que elimenemos la columna pan de la tabla credit_card

Para ello utilizaremos la función **ALTER TABLE credit_card DROP COLUMN pan;**

Esta función lo que hace es alterar la tabla eliminando la columna de pan

✓ 393 12:33:27 ALTER TABLE credit_card DROP COLUMN pan

Después de esto actualizamos las columnas y no aparece.



RESULTADO: ha sido eliminado de la tabla .

NIVEL 2

EJERCICIO 1

En este ejercicio se nos pide que eliminemos un registro de la base de datos

```
✓ 401 12:49:12 DELETE FROM user WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'
✓ 402 12:49:15 SELECT * FROM user WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02' LIMIT 0, 100
```

Par ello lo que hice fue buscar el id y borrarlo de USER ID

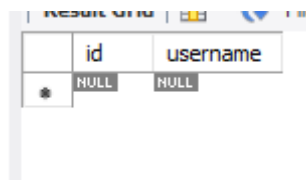
Use la siguiente función para borrarlo

```
DELETE FROM user WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B0'
```

Esta funcion lo que hace es buscar dentro de la table user el Id en e especifico y borrarlo

Para corroborar que esta echo bien, buscamos el id en user y no debería de aparecer

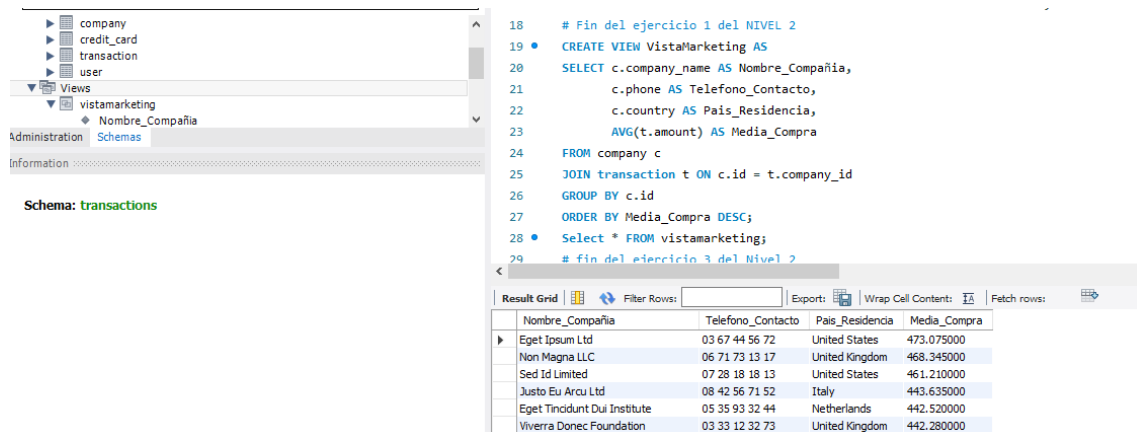
Luego al buscarlo me aparecen las casillas sin rellenar



RESULTADO : Se ha eliminado con éxito

Ejercicio 2

En este ejercicio se nos pide que creemos un modelo vista que contenga información de nombre de compañía, teléfono de contacto, país de residencia, media de compra realizado por cada compañía y presentar los datos ordenados de menor a mayor según el promedio de compra



The screenshot shows a database management tool interface. On the left, a tree view displays the database structure, including tables like 'company', 'credit_card', 'transaction', and 'user', and a view named 'vistamarketing'. The 'Schemas' tab is selected, showing the 'transactions' schema. On the right, a SQL editor displays the following code:

```

18 # Fin del ejercicio 1 del NIVEL 2
19 • CREATE VIEW VistaMarketing AS
20 SELECT c.company_name AS Nombre_Compañia,
21        c.phone AS Telefono_Contacto,
22        c.country AS Pais_Residencia,
23        AVG(t.amount) AS Media_Compra
24 FROM company c
25 JOIN transaction t ON c.id = t.company_id
26 GROUP BY c.id
27 ORDER BY Media_Compra DESC;
28 • Select * FROM vistamarketing;
29 # fin del ejercicio 3 del Nivel 2
    
```

Below the SQL editor, a 'Result Grid' shows the data returned by the query:

Nombre_Compañia	Telefono_Contacto	Pais_Residencia	Media_Compra
Eget Ipsum Ltd	03 67 44 56 72	United States	473.075000
Non Magna LLC	06 71 73 13 17	United Kingdom	468.345000
Sed Id Limited	07 28 18 18 13	United States	461.210000
Justo Eu Arcu Ltd	08 42 56 71 52	Italy	443.635000
Eget Tincidunt Dui Institute	05 35 93 32 44	Netherlands	442.520000
Viverra Donec Foundation	03 33 12 32 73	United Kingdom	442.280000

Lo primero que hice fue crear **CREATE VIEW VistaMarketing AS**

Esto lo que hace es ordenarle al programa que cree una vista con el nombre de VistaMarketing

Luego seleccionamos los apartados que queremos que esta incluya y le damos un alias a cada uno para que sea más entendible, toda esta información la sacamos de la tabla Company y transaction que van estar vinculadas mediante la función

JOIN transaction t ON c.id = t.company_id

Luego mediante las siguientes dos funciones le pedimos al programa que nos agrupe los id y que sean clasificados segun la media de sus transacciones

GROUP BY c.id

ORDER BY Media_Compra DESC;

Despues mediante la función

Select * FROM vistamarketing;

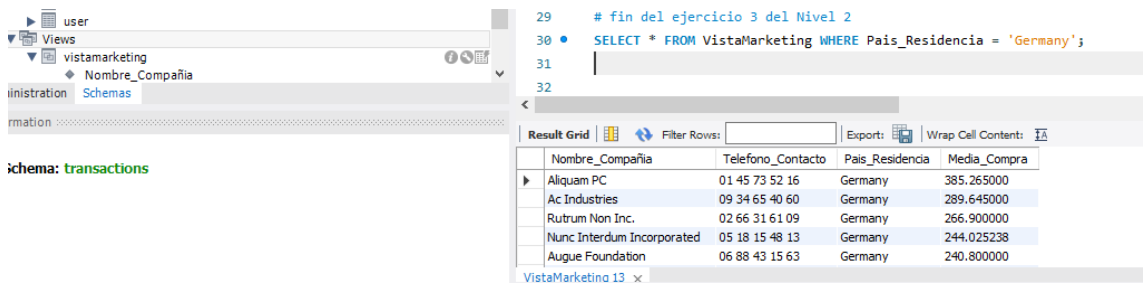
Comprobamos que el programa nos da los datos como queremos

Resultado:

Nombre_Compañia	Telefono_Contacto	Pais_Residencia	Media_Compra
Eget Ipsum Ltd	03 67 44 56 72	United States	473.075000
Non Magna LLC	06 71 73 13 17	United Kingdom	468.345000
Sed Id Limited	07 28 18 18 13	United States	461.210000
Justo Eu Arcu Ltd	08 42 56 71 52	Italy	443.635000
Eget Tincidunt Dui Institute	05 35 93 32 44	Netherlands	442.520000
Viverra Donec Foundation	03 33 12 32 73	United Kingdom	442.280000

EJERCICIO 3

En este ejercicio se nos pide que filtremos el vista que creamos en el ejercicio anterior para que solo muestre las compañías que han hecho alguna transaccion y esten como residentes en Germany(Alemania)



Para este ejercicio use la el siguiente codigo:

SELECT * FROM VistaMarketing WHERE Pais_Residencia = 'Germany';

Este codigo lo que hace es pedirle al programa que seleccione todas la columnas de la tabla y que solo nos muestre informacion de las compañías que tiene como pais de residencia Germany

	Nombre_Compañia	Telefono_Contacto	Pais_Residencia	Media_Compra
▶	Aliquam PC	01 45 73 52 16	Germany	385.265000
	Ac Industries	09 34 65 40 60	Germany	289.645000
	Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000
	Nunc Interdum Incorporated	05 18 15 48 13	Germany	244.025238
	Augue Foundation	06 88 43 15 63	Germany	240.800000

Resultado: VistaMarketing 13

EJERCICIO 3


En este ejercicio se nos pide que demos el nombre de las empresas que tienen su residencia fiscal en Alemania

Para ello lo primero que hice fue buscar las empresas de Alemania dentro de la base de datos.

Mediante la función: `SELECT * FROM company WHERE country = 'Germany';`

Esta función lo que hace es buscar dentro de la tabla Company todos los países que sean del país Germany(Alemania)

```
• SELECT * FROM company WHERE country = 'Germany';  
# fin del ejercicio 3 del Nivel 2
```

Resultado:  286 10:53:45 `SELECT * FROM company WHERE country = 'Germany' LIMIT 0, 100`

