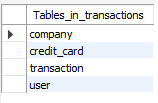
SPRINT 3

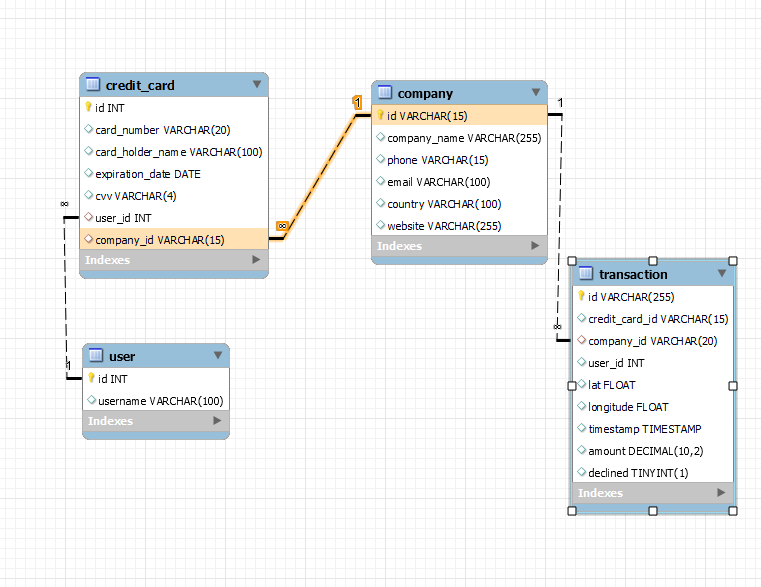
EJERCICIO 1

En este ejercicio lo que se pide es que creemos 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



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

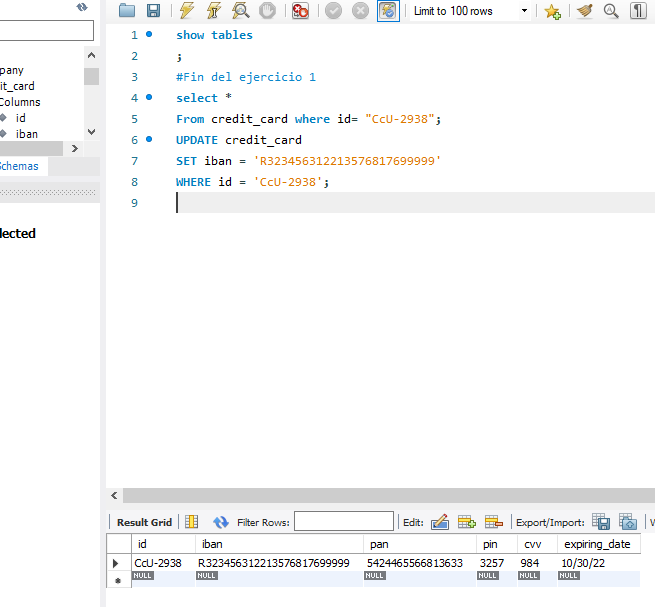


EJERCICIO 2

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



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

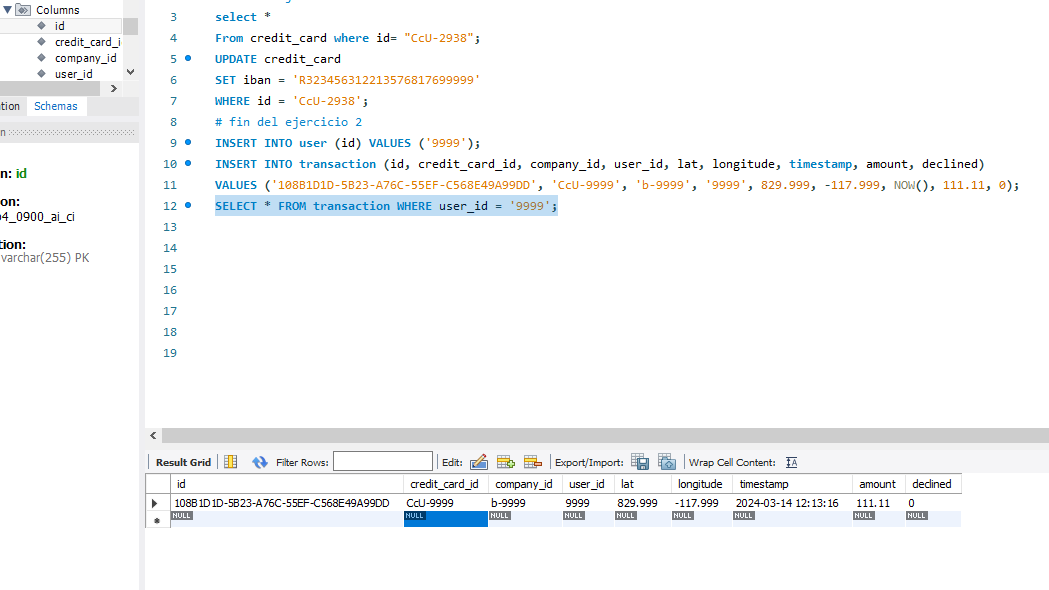
**UPDATE credit\_card SET iban = 'R323456312213576817699999' WHERE id = 'CcU-2938';**

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.



Para poder hacerlo tuve que modificar la estrutctura de datos ya que como la tenia 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 tenia 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 tenia que ser

EJERCICIO 4

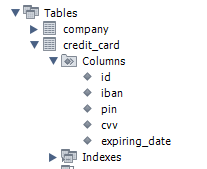
Desde recursos humanos se nos pide que elimenemos la columna pan de la tabla credit\_card

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



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



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

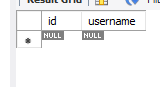
Use la siguiente función para borrarlo

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

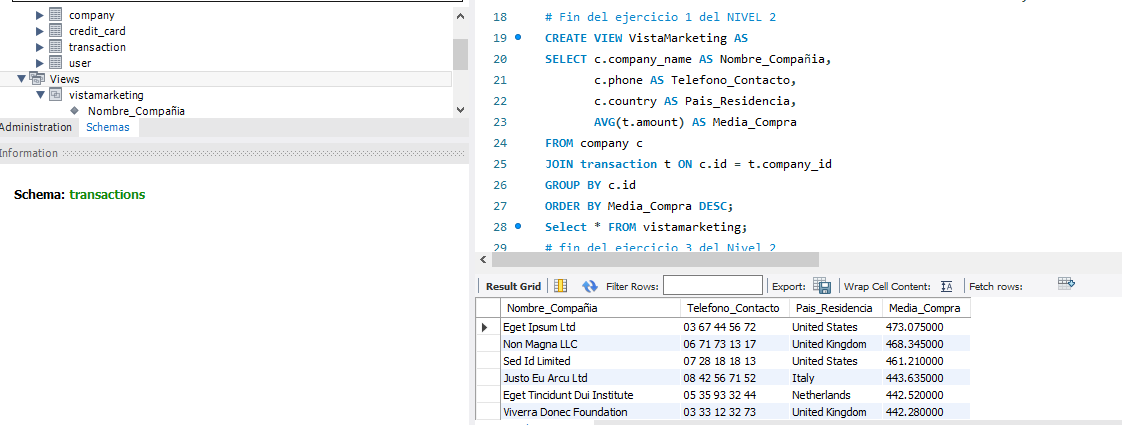
Luego al buscarlo me aparecen las casillas sin rellenar



RSULTADO : 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



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 VistaMarqueting

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

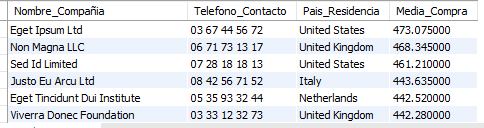
**GROUP BY c.id**

**ORDER BY Media\_Compra DESC;**

Despues mediante la función

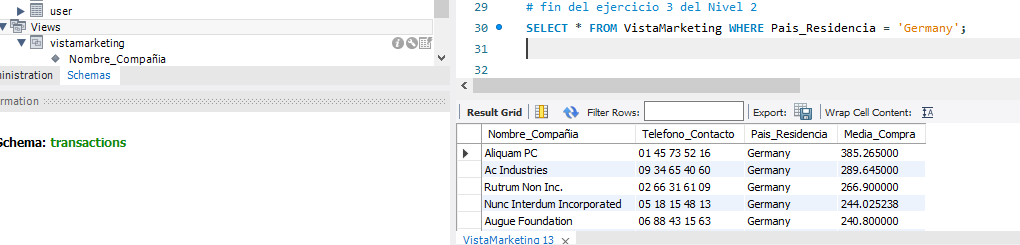
**Select \* FROM vistamarketing;**

Comprobamos que el programa nos da los datos como queremos

Resultado: 

EJERCICIO 3

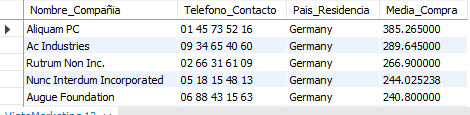
En este ejercicio se nos pide que filtemos el vista que creamos en el ejercicio anterior para que solo muestre las compañias que han hecho alguna transaccion y esten como residentes en Germany( Alemania )



Para este ejercicio use la el sguiente 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 compañias que tiene como pais de residencia Germany

Resultado: 

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 funcion lo que hace es buscar detrno de la tabla Company todos los países que sean del país Germany(Alemania)



Resultado: 