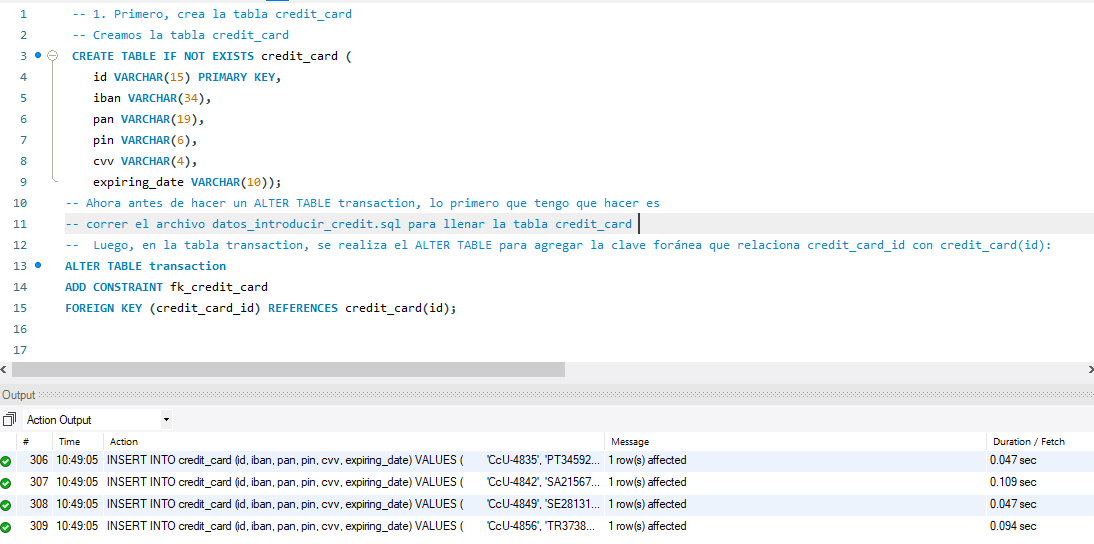
**Tarea S3.01. Manipulación de tablas**

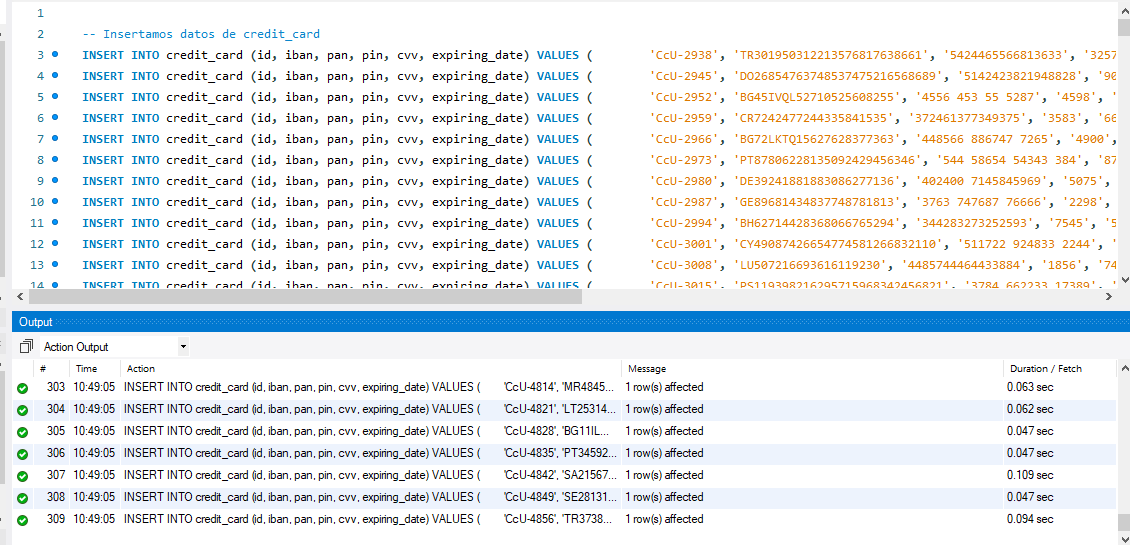
En este sprint, se simula una situación empresarial en la que debes realizar varias manipulaciones en las tablas de la base de datos. A su vez, tendrás que trabajar con índices y vistas. En esta actividad, continuarás trabajando con la base de datos que contiene información de una empresa dedicada a la venta de productos en línea. En esta tarea, empezarás a trabajar con información relacionada con tarjetas de crédito.

## Nivel 1

### **- Ejercicio 1**

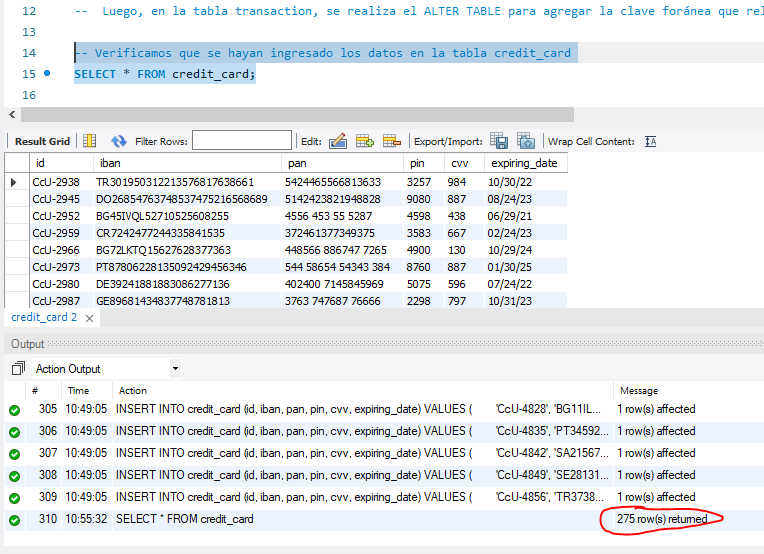
Tu tarea es diseñar y crear una tabla llamada "credit\_card" que almacene detalles cruciales sobre las tarjetas de crédito. La nueva tabla debe ser capaz de identificar de manera única cada tarjeta y establecer una relación adecuada con las otras dos tablas ("transaction" y "company"). Después de crear la tabla, será necesario que ingreses la información del documento denominado "dades\_introduir\_credit". Recuerda mostrar el diagrama y realizar una breve descripción del mismo.



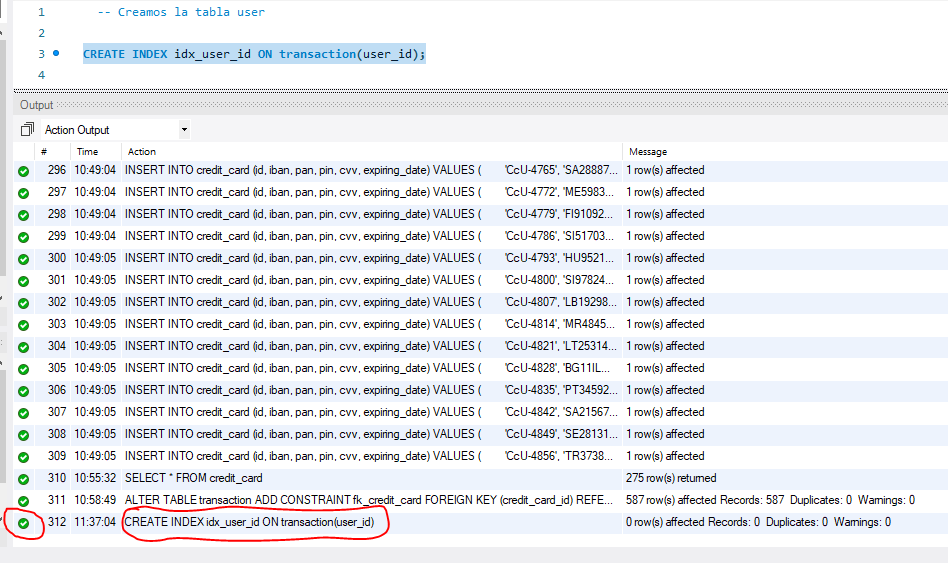


-- Verificamos que se hayan ingresado los datos en la tabla credit\_card

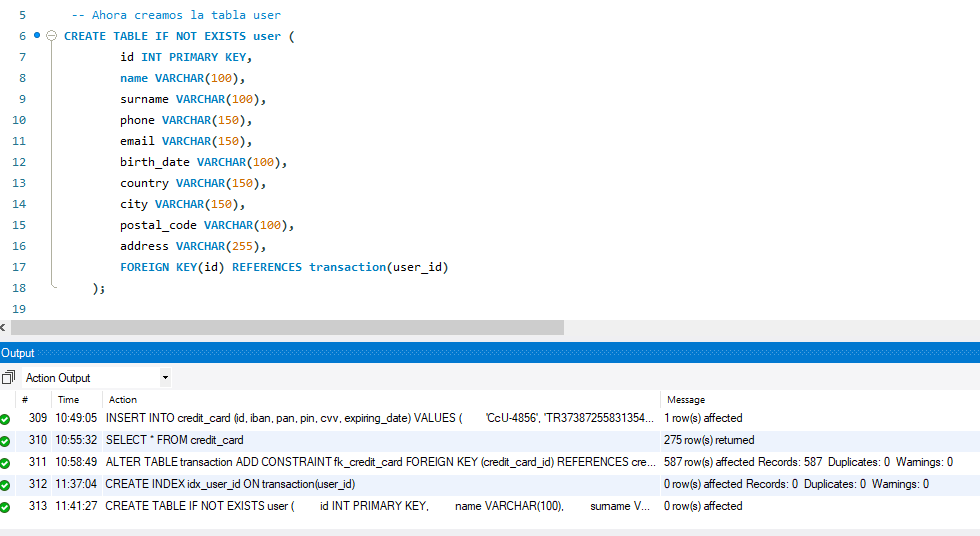
SELECT \* FROM credit\_card;



Ahora se creará la tabla user, antes de ello creamos el Índice: idx\_user\_id



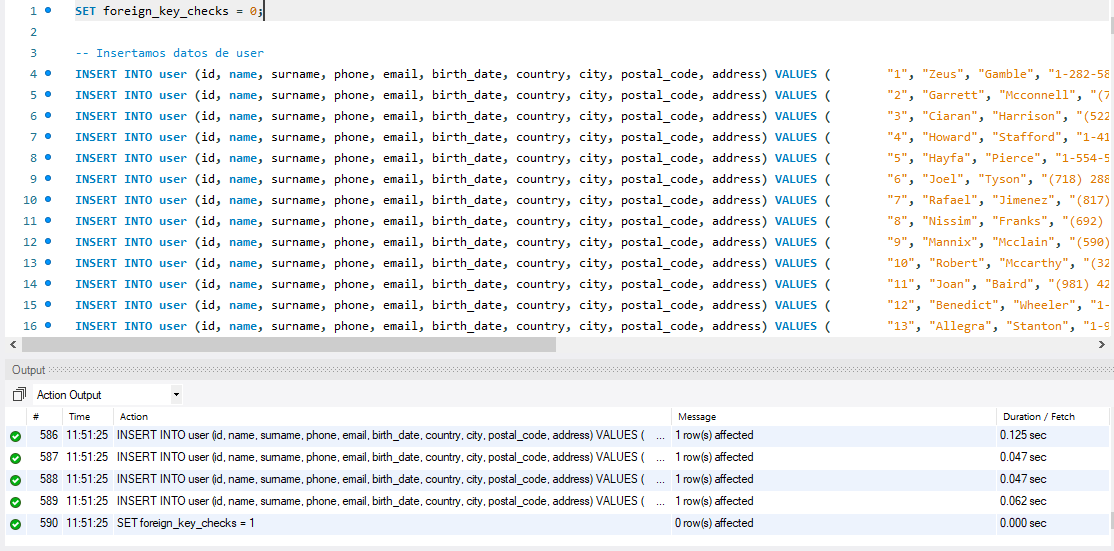
Ahora creamos la tabla: user



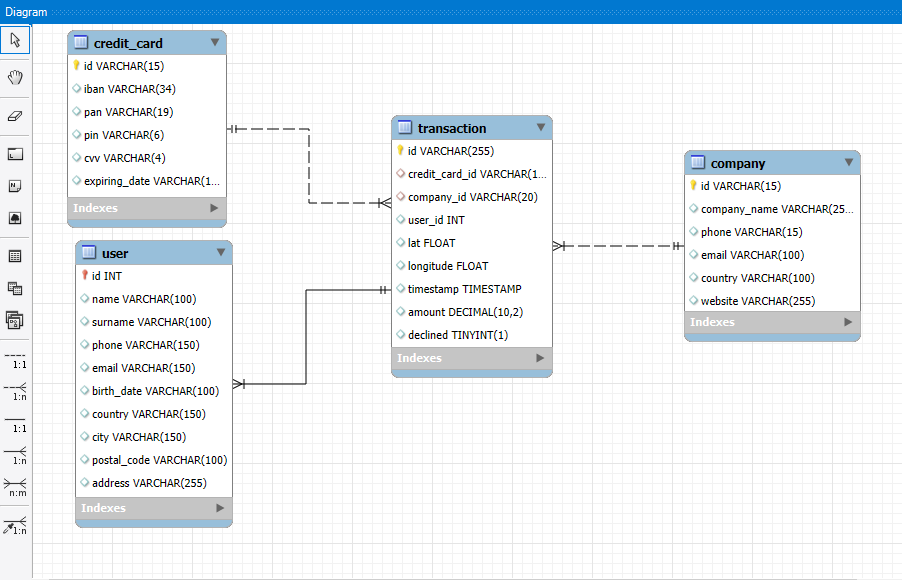
Ahora insertamos registros en la tabla user

Se ejecuta el archivo datos\_introducir\_user (1).sql, que **inserta registros** en la tabla user.”

El archivo datos\_introducir\_user (1).sql contiene las sentencias necesarias para insertar registros en la tabla user. **Se desactivan temporalmente las verificaciones de claves foráneas con SET foreign\_key\_checks = 0; para permitir una carga segura de los datos, y se reactivan al final con SET foreign\_key\_checks = 1; para asegurar la integridad referencial del sistema.**



Descripción breve del diagrama



**Tablas de la base de datos transactions**

En el diagrama se pueden ver **cuatro tablas**: transaction, credit\_card, user y company. A continuación, se explica cada una:

1. **Transaction**

* **Es la tabla principal** que registra todas las transacciones realizadas.
* **Campos clave**:
  + id → identificador único (clave primaria).
  + credit\_card\_id → relaciona cada transacción con una tarjeta de crédito.
  + company\_id → indica en qué empresa se hizo la transacción.
  + user\_id → indica qué usuario la realizó.
  + lat, longitude → coordenadas geográficas.
  + timestamp → fecha y hora.
  + amount → monto de la transacción.
  + declined → si fue rechazada o no.
* **Relaciones**:
  + credit\_card\_id → FK hacia credit\_card(id)
  + company\_id → FK hacia company(id)
  + user\_id → FK hacia user(id)

**Es el centro del diagrama**, desde donde parten las relaciones hacia las otras tablas.

1. **credit\_card**

* Almacena los datos de las tarjetas usadas en las transacciones.
* **Campos clave**:
  + id (clave primaria)
  + iban, pan, pin, cvv, expiring\_date
* **Relación**:
  + Su campo id es referenciado desde transaction(credit\_card\_id)

**Tiene una relación 1:N con transaction**:  
Una tarjeta puede tener **muchas transacciones**.

1. **user**

* Contiene la información personal del usuario.
* **Campos clave**:
  + id (clave primaria)
  + name, surname, email, phone, etc.
* **Relación**:
  + Su campo id es referenciado desde transaction(user\_id)

**Relación 1:N con transaction**:  
Un usuario puede hacer **muchas transacciones**.

1. **company**

* Representa las empresas donde se realizaron las compras.
* **Campos clave**:
  + id (clave primaria)
  + name, industry, country, etc.
* **Relación**:
  + Su campo id es referenciado desde transaction(company\_id)

**Relación 1:N con transaction**:  
Una empresa puede recibir **muchas transacciones**.

**Descripción del diagrama**

El diagrama muestra la estructura de la base de datos Transaction, compuesta por cuatro tablas principales. La tabla transaction funciona como núcleo, ya que se conecta con user, credit\_card y company a través de claves foráneas. Las relaciones son todas de tipo 1:N, es decir, un usuario, tarjeta o empresa puede estar vinculada a múltiples transacciones. Esta estructura permite un registro completo de cada operación financiera, incluyendo quién la hizo, con qué tarjeta, en qué empresa y bajo qué coordenadas geográficas.”

La tabla credit\_card almacena información esencial de cada tarjeta de crédito utilizada por los usuarios para realizar transacciones. Esta tabla tiene una relación directa con la tabla transaction, ya que cada transacción está vinculada a una tarjeta específica.

**Estructura de la tabla credit\_card:**

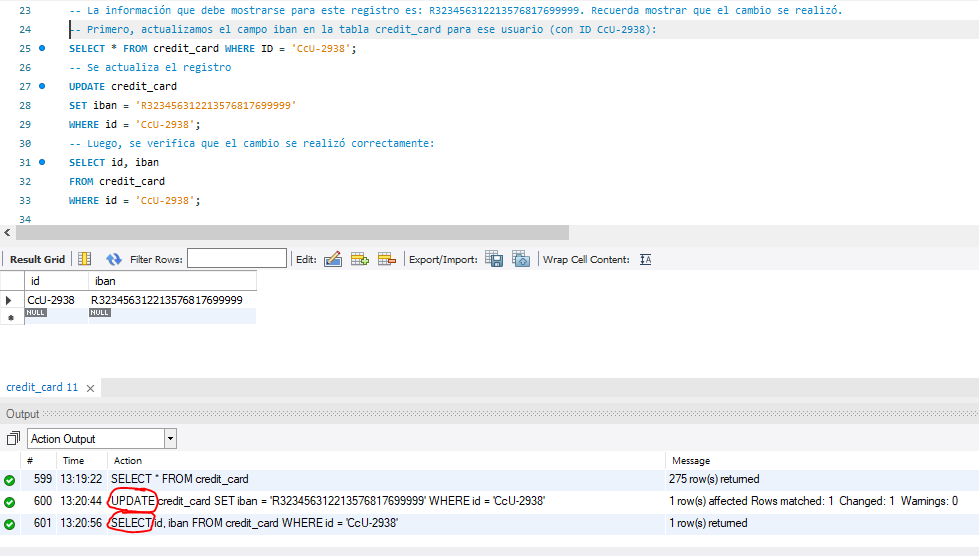
| **Campo** | **Tipo** | **Descripción** |
| --- | --- | --- |
| id | VARCHAR(15) | Identificador único de la tarjeta (PK) |
| iban | VARCHAR(34) | Número IBAN asociado a la cuenta |
| pan | VARCHAR(19) | Número de la tarjeta (Primary Account Number) |
| pin | VARCHAR(6) | Número secreto de seguridad |
| cvv | VARCHAR(4) | Código de verificación de la tarjeta |
| expiring\_date | VARCHAR(10) | Fecha de vencimiento de la tarjeta |

La clave primaria es el campo id, y este se utiliza como **clave foránea** en la tabla transaction, asegurando que cada transacción esté asociada a una tarjeta válida.

### **- Ejercicio 2**

El departamento de Recursos Humanos ha identificado un error en el número de cuenta del usuario con ID CcU-2938. La información que debe mostrarse para este registro es: R323456312213576817699999. Recuerda mostrar que el cambio se realizó.

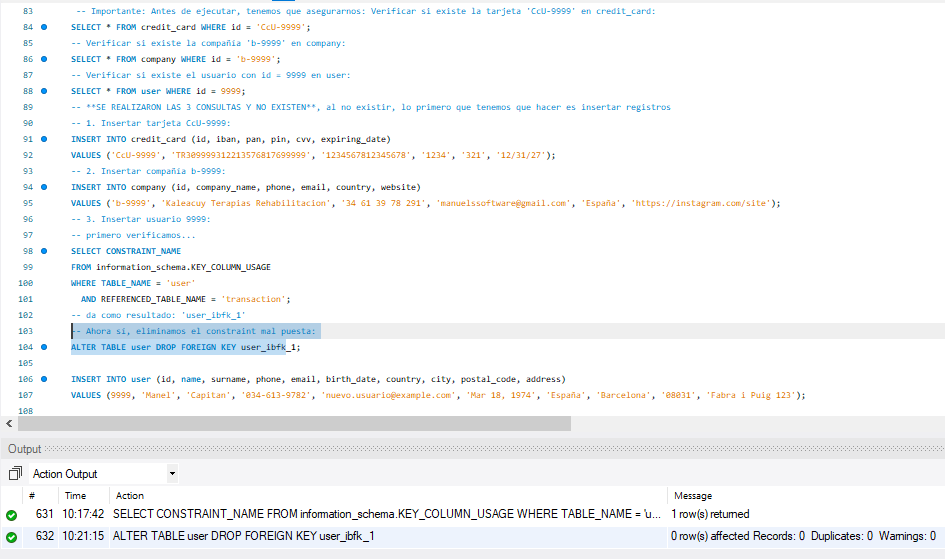
--Primero, actualizamos el campo iban en la tabla credit\_card para ese usuario (con ID CcU-2938):



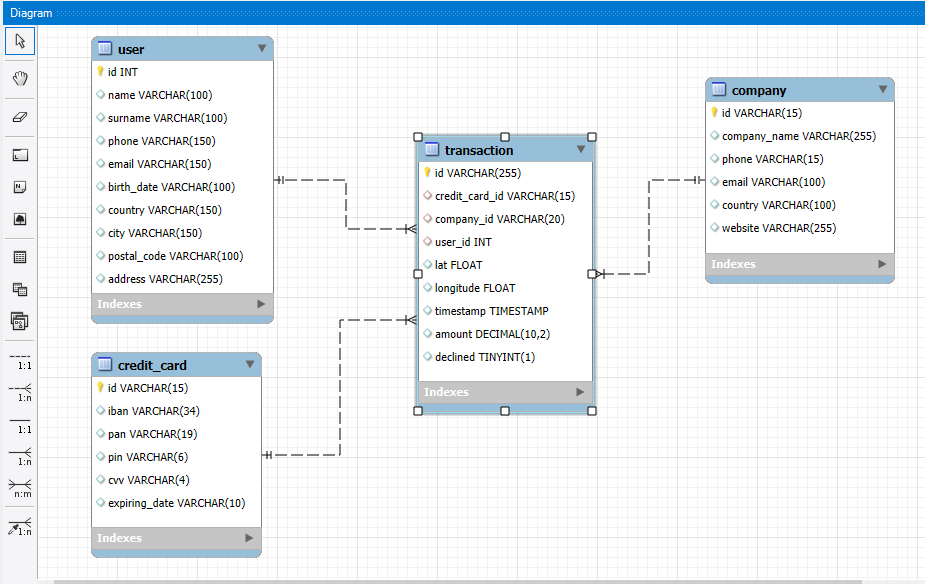
### **- Ejercicio 3**

En la tabla "transaction" ingresa un nuevo usuario con la siguiente información:

|  |  |
| --- | --- |
| Id | 108B1D1D-5B23-A76C-55EF-C568E49A99DD |
| credit\_card\_id | CcU-9999 |
| company\_id | b-9999 |
| user\_id | 9999 |
| lat | 829.999 |
| longitude | -117.999 |
| amount | 111.11 |
| declined | 0 |



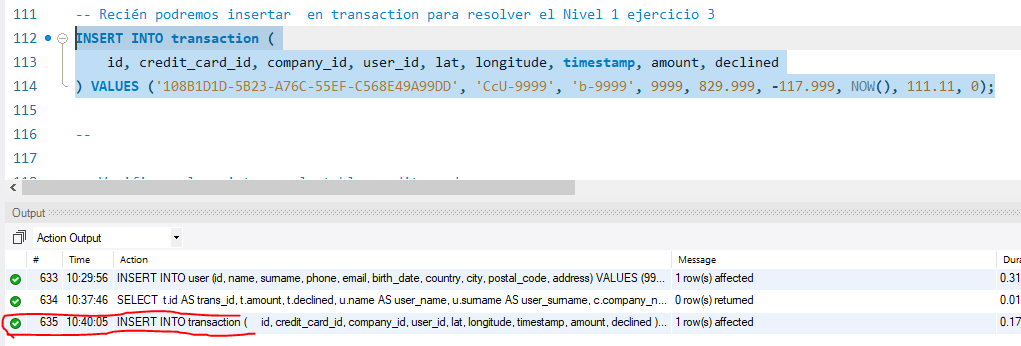
**Diagrama**



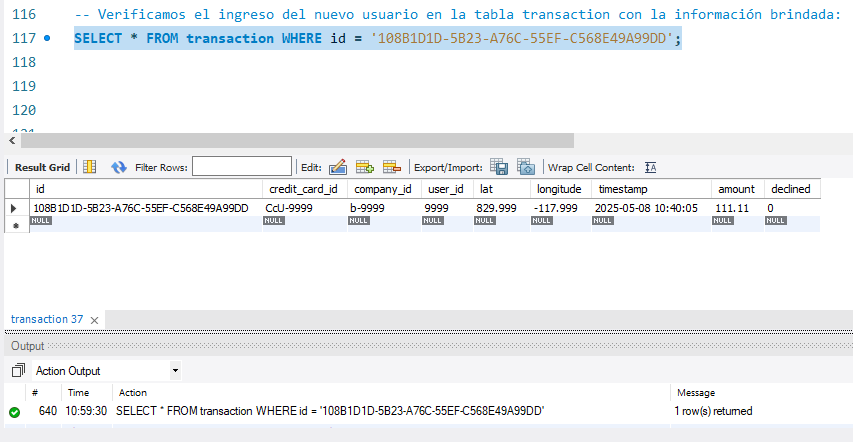
**-- Ahora si ingresaremos al nuevo usuario con "id": '9999'**



**-- Recién podremos insertar en transaction para resolver el Nivel 1 ejercicio 3**

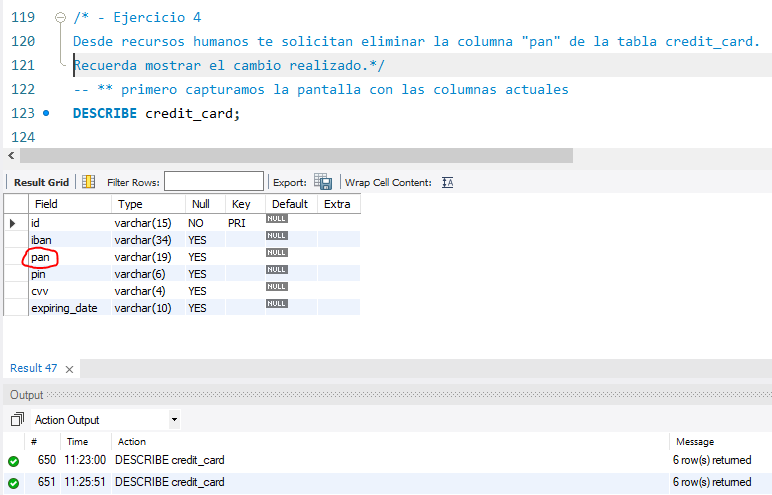


**-- Verificamos el ingreso del nuevo usuario en la tabla transaction con la información brindada:**

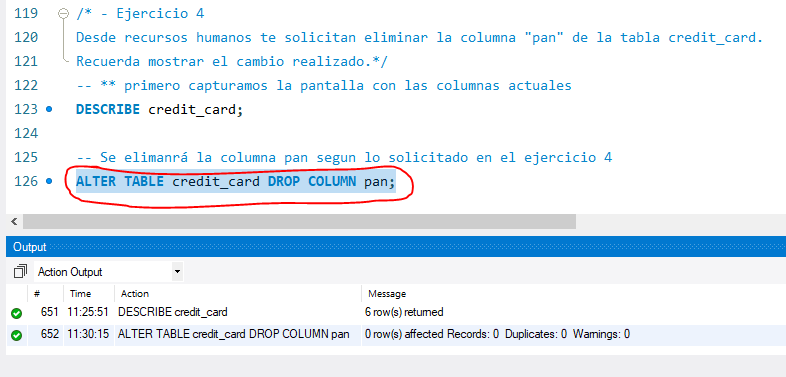


### **- Ejercicio 4**

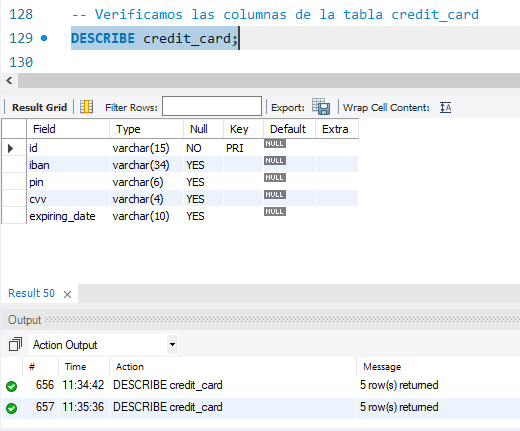
Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla credit\_\*card. Recuerda mostrar el cambio realizado.



**-- Se elimina la columna pan según lo solicitado en el ejercicio 4**



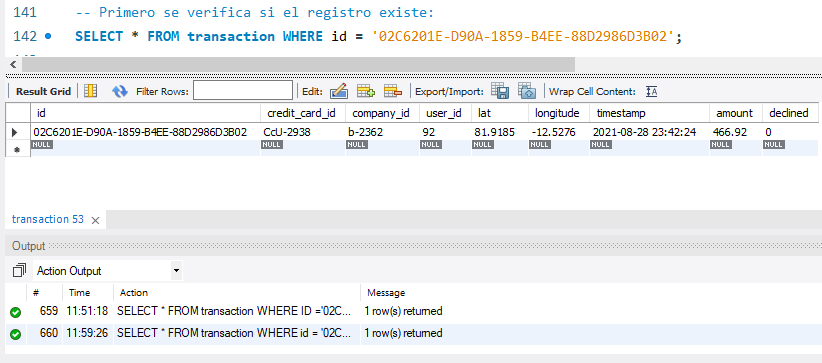
**-- Verificamos las columnas de la tabla credit\_card, en la cual ya no se visualiza la columna ‘pan’, según lo solicitado**

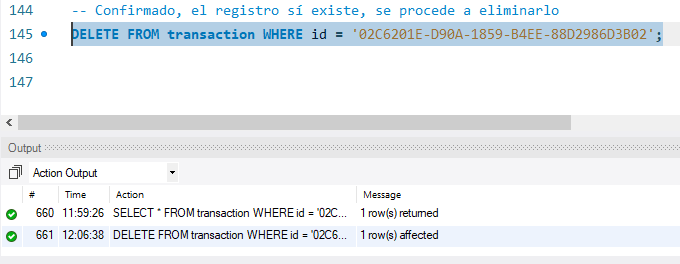


## Nivel 2

### **Ejercicio 1**

Elimina de la tabla transaction el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de datos.

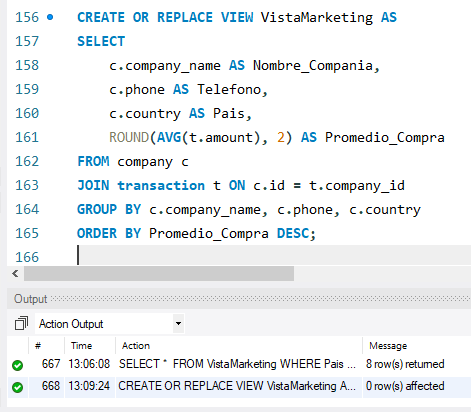




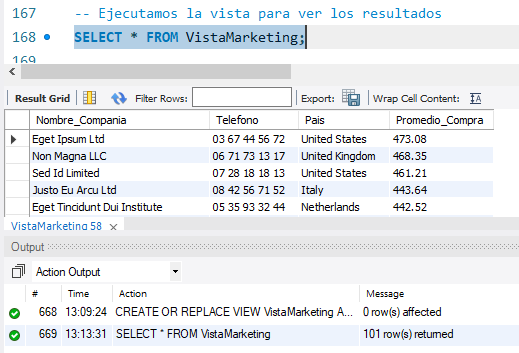
### 

### **Ejercicio 2**

La sección de marketing desea tener acceso a información específica para realizar análisis y estrategias efectivas. Se ha solicitado crear una vista que proporcione detalles clave sobre las compañías y sus transacciones. Será necesaria que crees una vista llamada VistaMarketing que contenga la siguiente información: Nombre de la compañía. Teléfono de contacto. País de residencia. Promedio de compra realizado por cada compañía. Presenta la vista creada, ordenando los datos de mayor a menor media de compra.

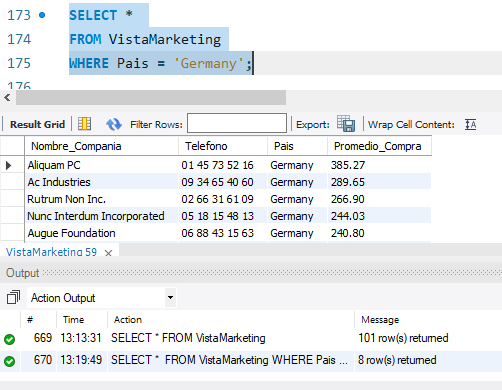


-- Ejecutamos la vista para ver los resultados



### **Ejercicio 3**

**Filtra la vista VistaMarketing para mostrar sólo las compañías que tienen su país de residencia en "Germany"**

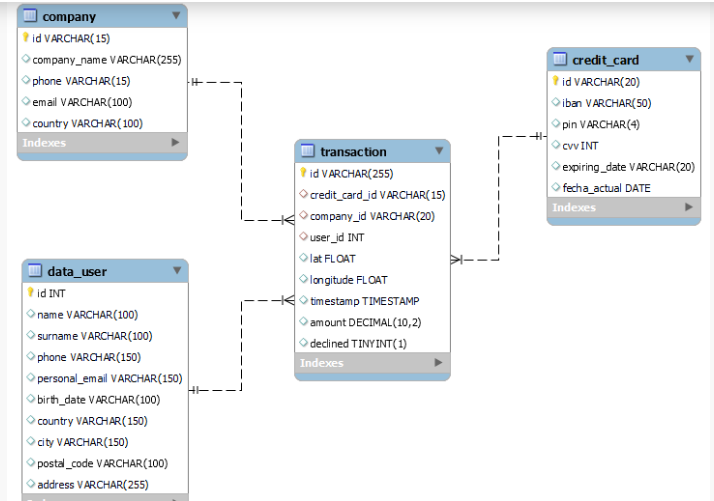


## Nivel 3

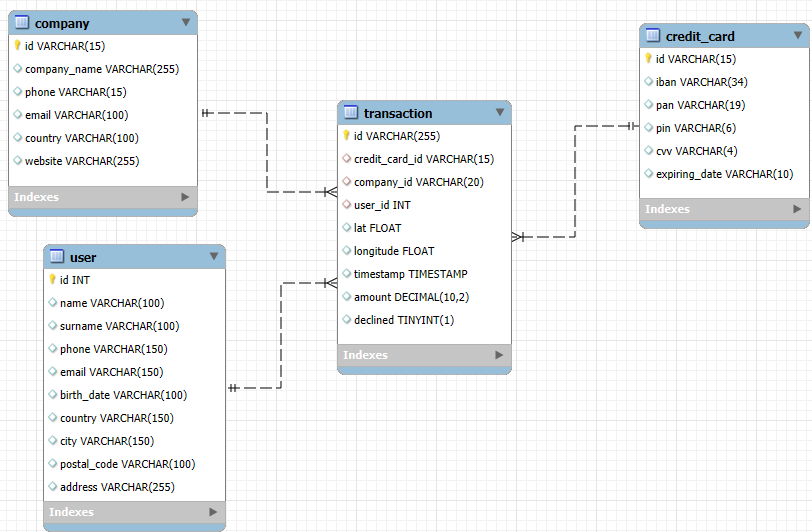
### **Ejercicio 1**

La semana próxima tendrás una nueva reunión con los gerentes de marketing. Un compañero de tu equipo realizó modificaciones en la base de datos, pero no recuerda cómo las realizó. Te pide que le ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama:

**El diagrama debe quedar tal como está en la imagen**



**El diagrama que actualmente tenemos es:**



**Comparando los dos diagramas, se procede a ordenar las tablas como nos piden y comparamos las diferencias:**

1. Tabla user cambia a: data\_user

RENAME TABLE user TO data\_user;

1. La columna email cambia de nombre a: personal\_email.

ALTER TABLE data\_user CHANGE email personal\_email VARCHAR(150);

1. Tabla credit\_card: id VARCHAR(15) cambia a: id VARCHAR(20)

ALTER TABLE credit\_card MODIFY id VARCHAR(20);

1. iban VARCHAR(34) cambia a: iban VARCHAR(50)

ALTER TABLE credit\_card MODIFY iban VARCHAR(50);

1. pin VARCHAR(6) cambia a: pin VARCHAR(4)

ALTER TABLE credit\_card MODIFY pin VARCHAR(4);

1. cvv VARCHAR(4) cambia a: cvv INT

ALTER TABLE credit\_card MODIFY cvv INT;

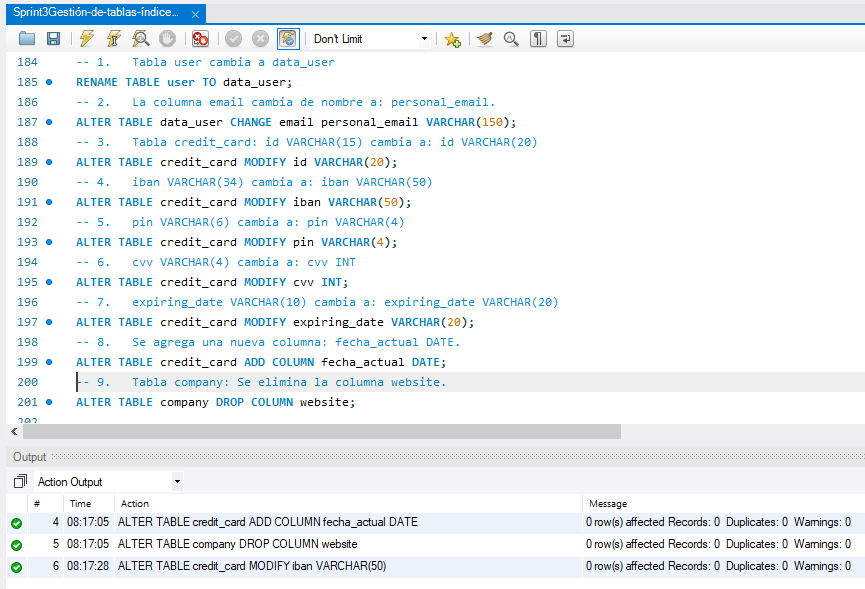
1. expiring\_date VARCHAR(10) cambia a: expiring\_date VARCHAR(20)

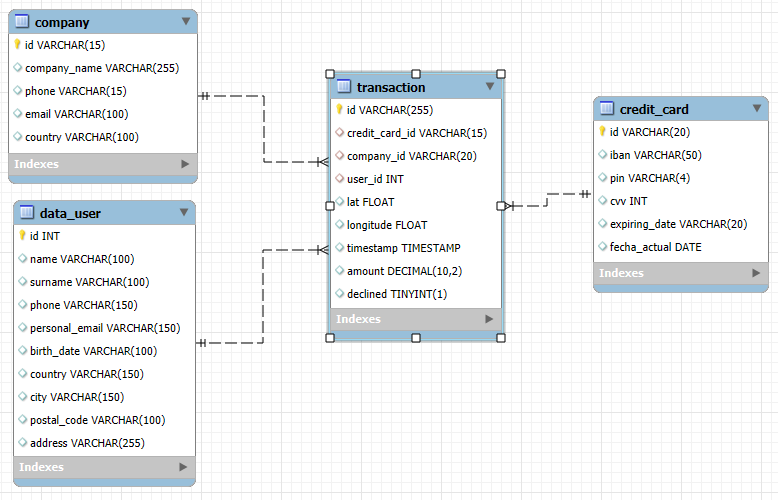
ALTER TABLE credit\_card MODIFY expiring\_date VARCHAR(20);

1. Se agrega una nueva columna: fecha\_actual DATE.

ALTER TABLE credit\_card ADD COLUMN fecha\_actual DATE;

1. Tabla company: Se elimina la columna website.

ALTER TABLE company DROP COLUMN website;

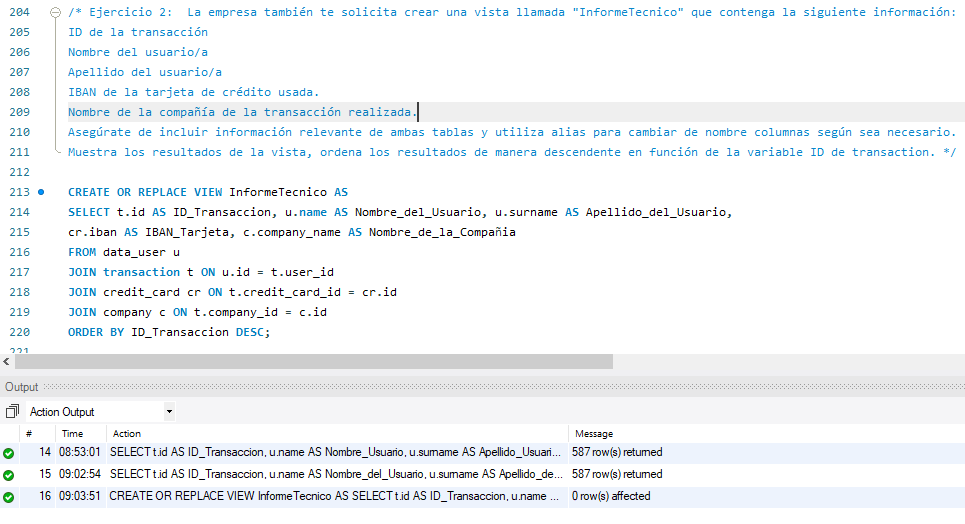


### **Ejercicio 2**

La empresa también te solicita crear una vista llamada "InformeTecnico" que contenga la siguiente información:

* ID de la transacción
* Nombre del usuario/a
* Apellido del usuario/a
* IBAN de la tarjeta de crédito usada.
* Nombre de la compañía de la transacción realizada.
* Asegúrate de incluir información relevante de ambas tablas y utiliza alias para cambiar de nombre columnas según sea necesario.

Muestra los resultados de la vista, ordena los resultados de manera descendente en función de la variable ID de transaction.



**SELECT** \* **FROM** transactions.informetecnico;

