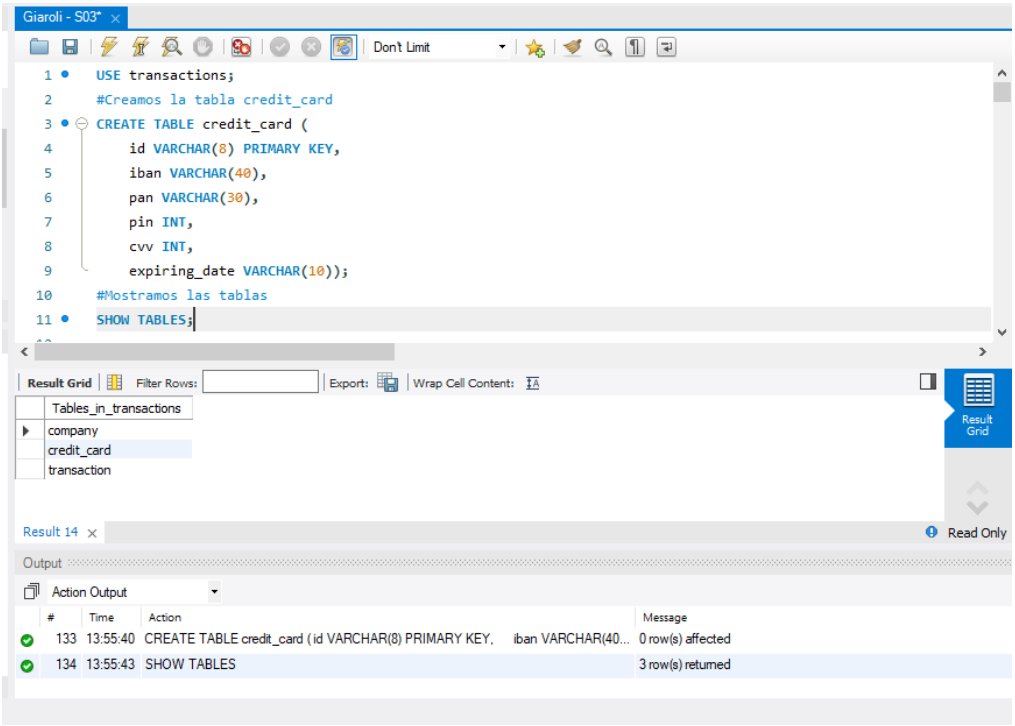
Nivel 1

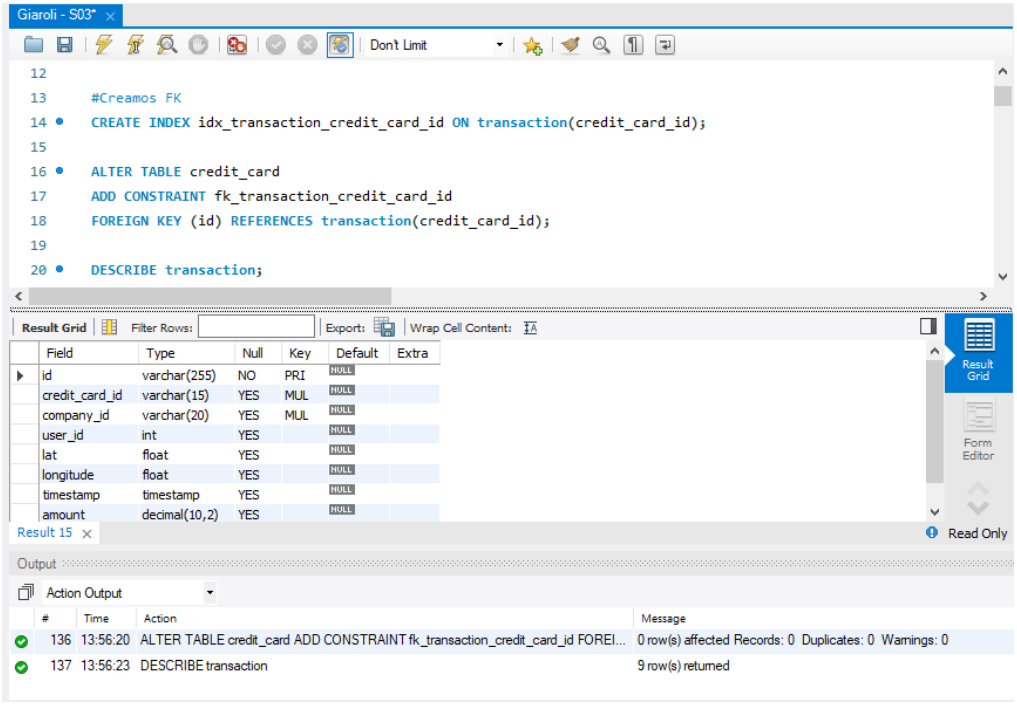
Ejercicio 1

Creamos la tabla “credit_card” con el siguiente comando:

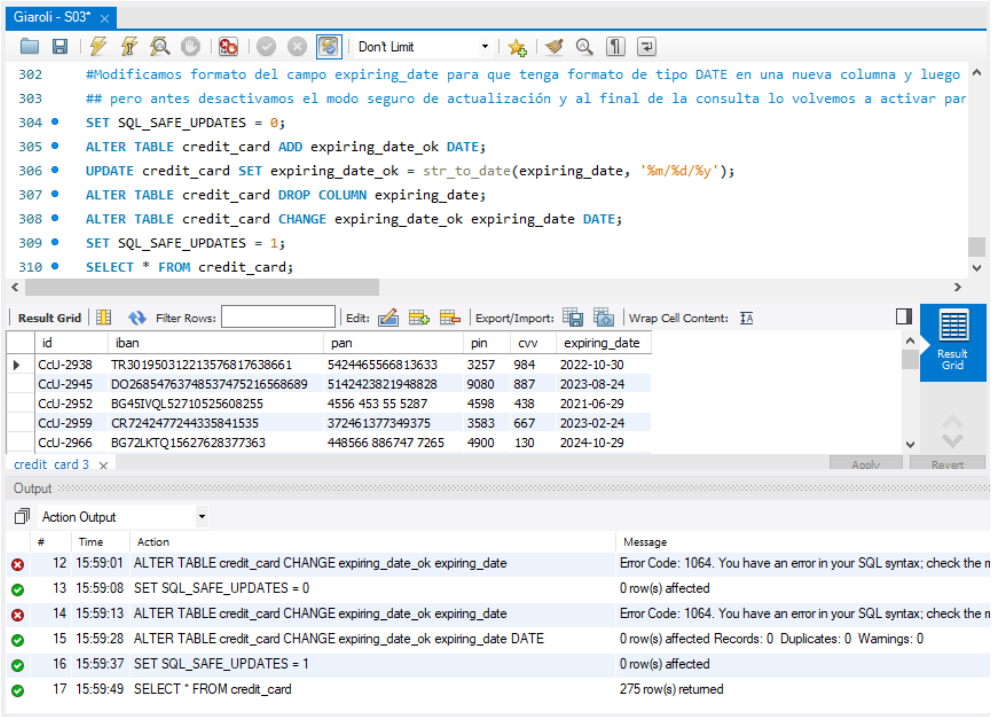


Decidimos que los campos id, iban y pan sean de tipo VARCHAR porque además de números pueden incluir espacios en blanco. Pin y cvv son de tipo INTEGER y expiring_date también será VARCHAR, pero luego de insertar los datos cambiaremos el formato para que puedan corresponderse a los de tipo DATE.

Definimos **id como PK**. Y en otra consulta, definimos también a id como la FK de la tabla transaction donde transaction.credit_card_id=credit_card.id. Ambas tablas tienen una **relación de N a 1 (transaction->credit_card)**, porque puede haber varias comprar realizadas con una misma tarjeta.



Una vez creada la tabla e ingresados los datos, podemos modificar los datos de expiring_date para lograr que sean de tipo DATE con los siguientes comandos.

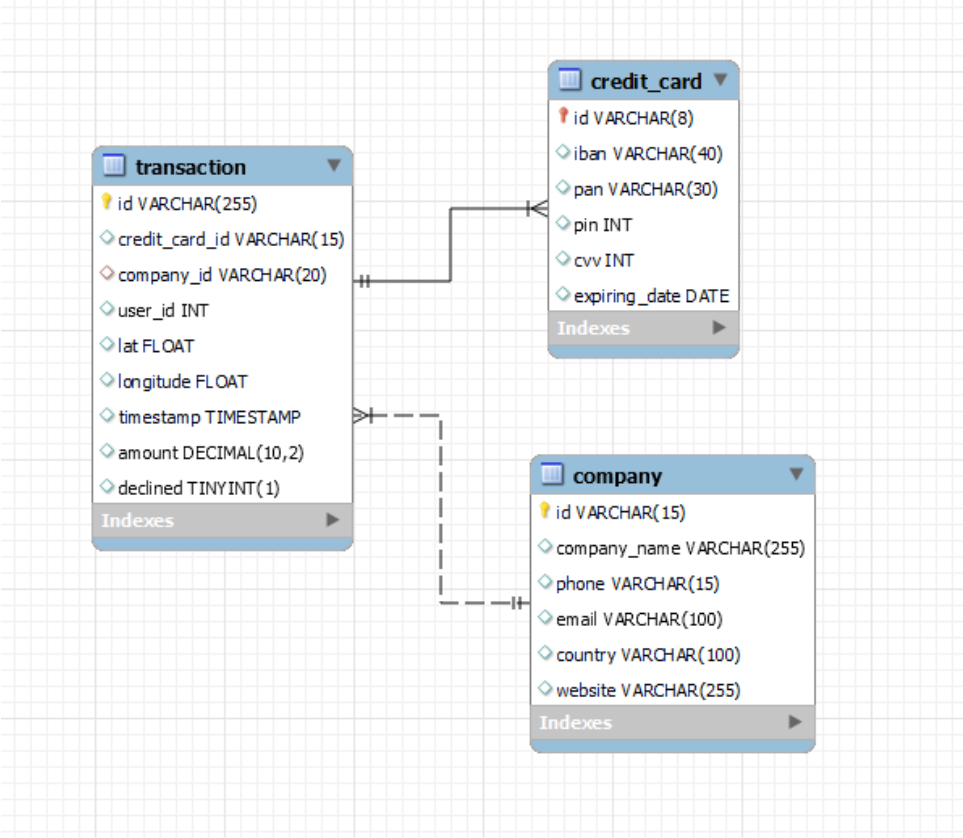


Para evitar futuros errores podríamos pedir al cliente que los nuevos datos de fechas de caducidad se registren con formato YYYY-MM-DD.

En el caso de que esto no sea posible, tal vez sería conveniente continuar con el formato VARCHAR o buscar alguna solución alternativa.

La **base de datos "Transactions"** queda entonces compuesta por **tres tablas: "Company", "Credit_Card" y "Transaction".**

DB TRANSACTIONS

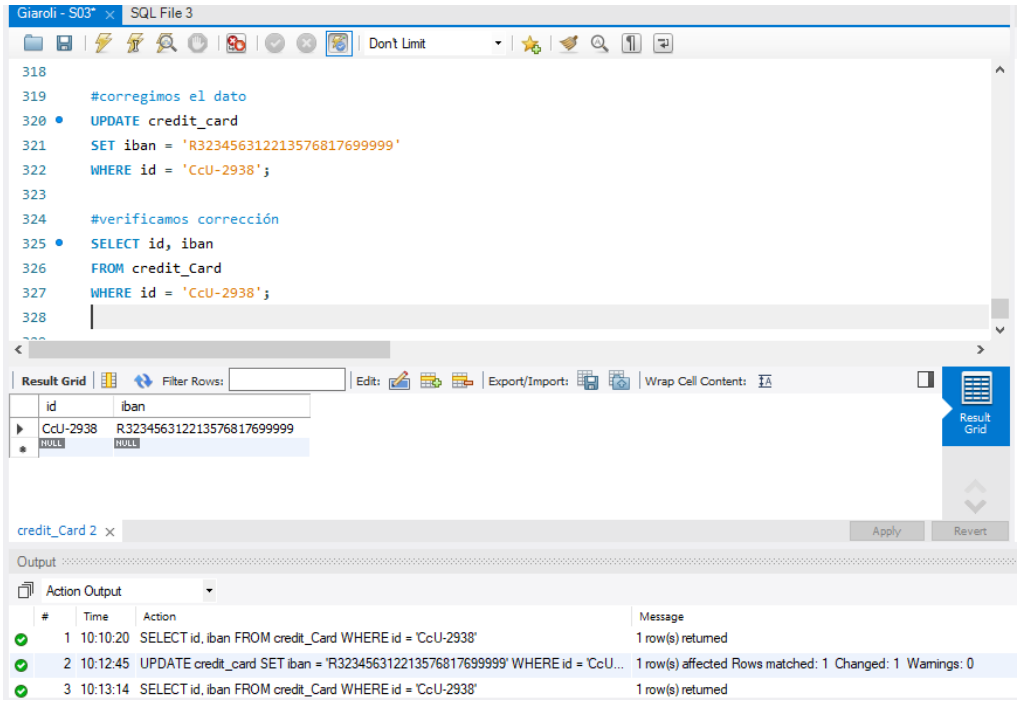


Company y Transaction ya han sido comentadas en el Sprint1. La nueva tabla incorporada al modelo es Credit_Card, compuesta por 6 campos que permiten identificar cada tarjeta de crédito por su id. Este campo actúa como PK de la tabla y a la vez como FK con la tabla "Transaction". Además, la tabla contiene información sobre el iban, pan, pin, cvv y fecha de caducidad de cada tarjeta.

La base de datos queda conformada como un **modelo estrella**, con la tabla **Transaction como tabla de hechos** y las otras dos como tablas de dimensiones.

Ejercicio 2

Para actualizar el IBAN del usuario con ID CcU-2938 realizamos los siguientes comandos:

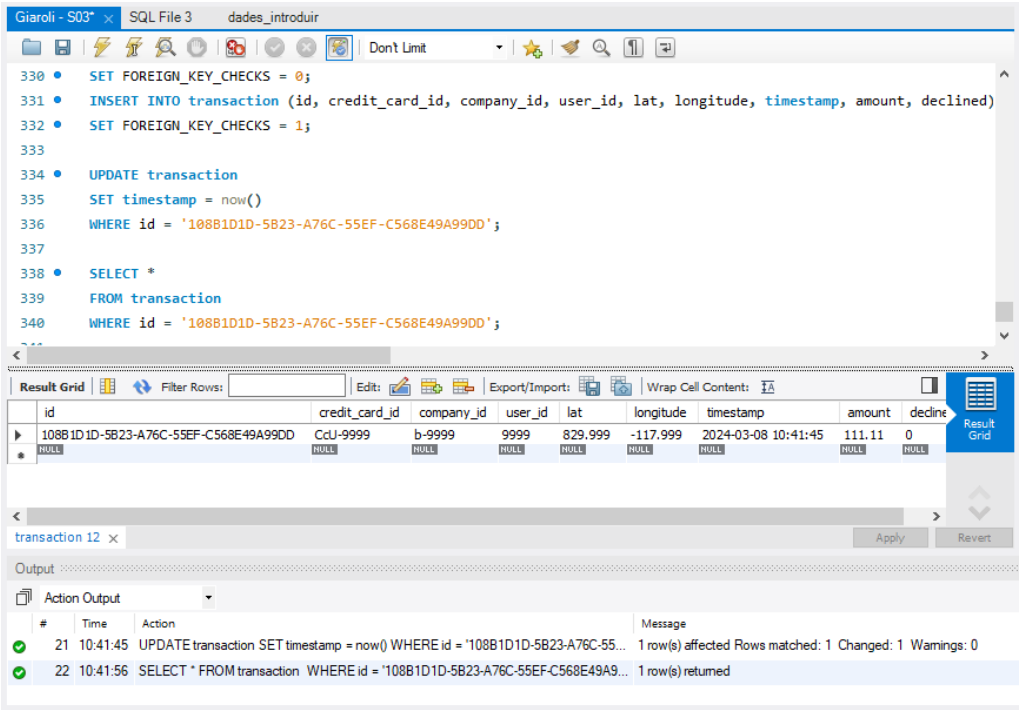


Ejercicio 3

Para poder ingresar un nuevo registro en la tabla Transaction, activamos y desactivamos el modo seguro de actualización con SET FOREIGN KEY CHECKS.

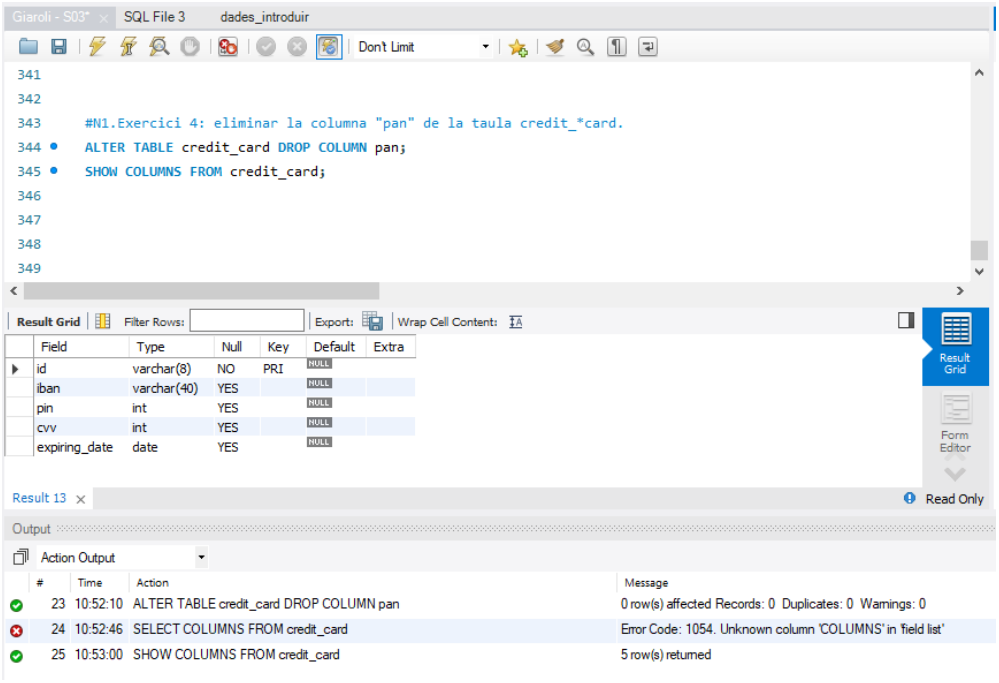
Luego actualizamos la tabla con el nuevo registro con el comando INSERT INTO. Al verificar los datos introducidos vemos que no se ha impreso timestamp así que lo hacemos en otra query (UPDATE).

Finalmente, revisamos que los datos introducidos sean correctos.



Ejercicio 4

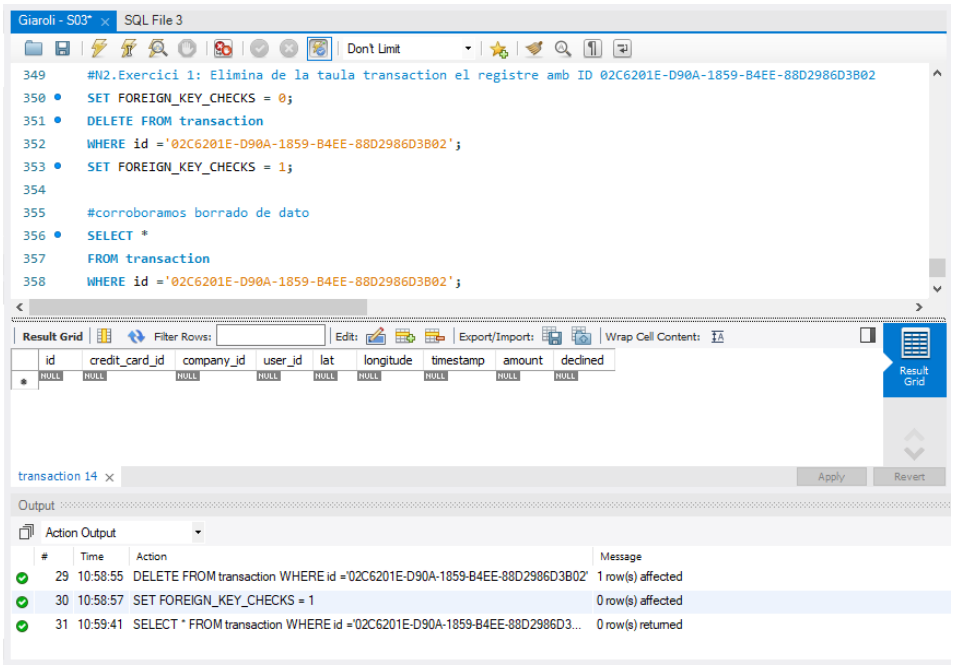
Para eliminar la columna pan de la tabla credit_card realizamos los siguientes comandos:



Nivel 2

Ejercicio 1

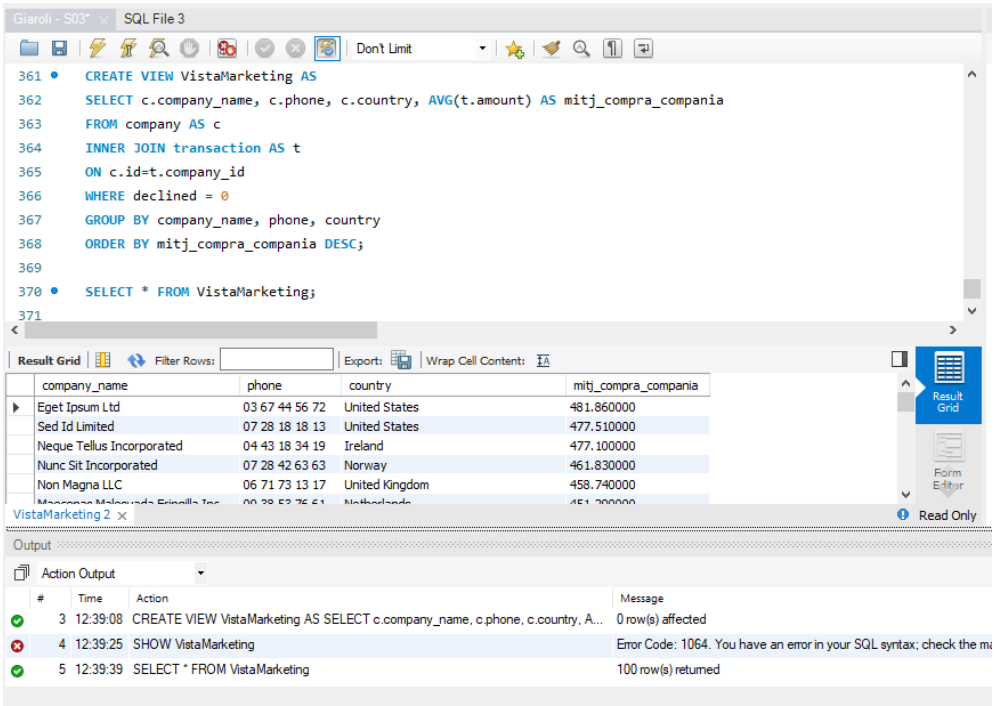
Para eliminar de la tabla Transaction el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02, realizamos lo siguiente:



Ejercicio 2

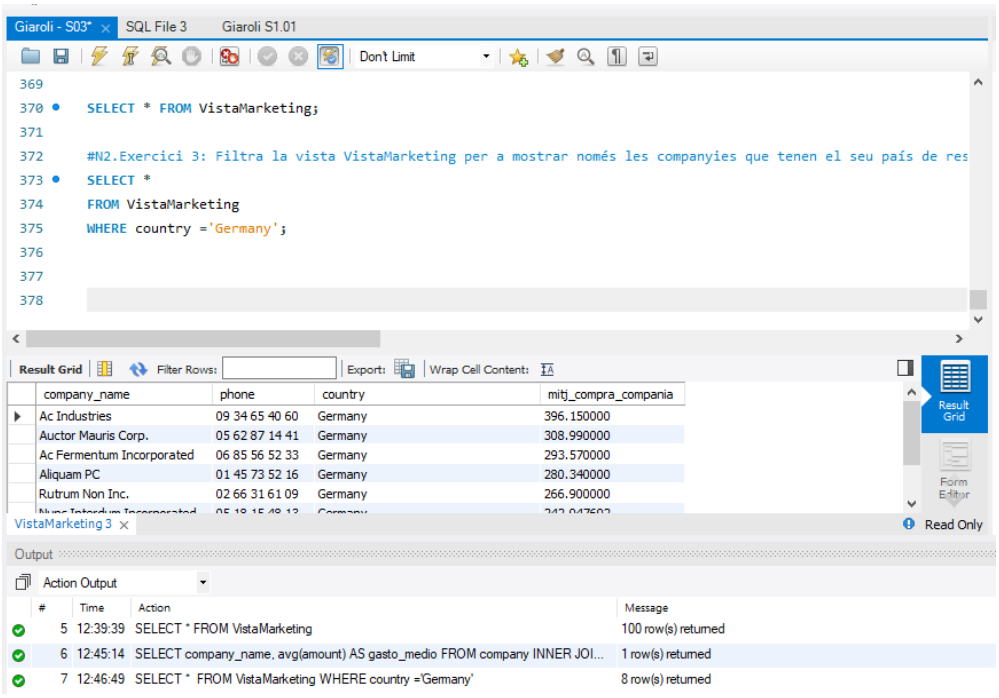
Para crear la vista VistaMarketing tenemos que crear una vista compleja, ya que utilizaremos datos de dos tablas, con funciones de agregación y agrupación.

En este caso tendremos en cuenta al booleano Declined ya que el enunciado pide las compras realizadas.



Ejercicio 3

Para filtrar de la vista VistaMarketing sólo a las compañías con sede en Alemania realizamos la siguiente consulta:

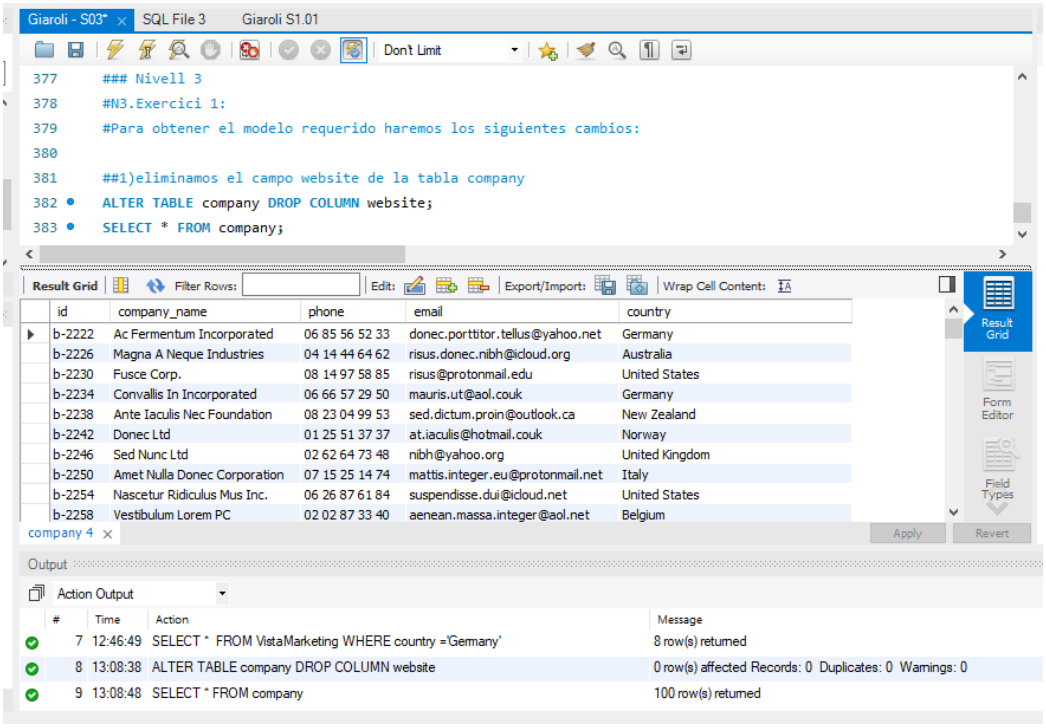


Nivel 3

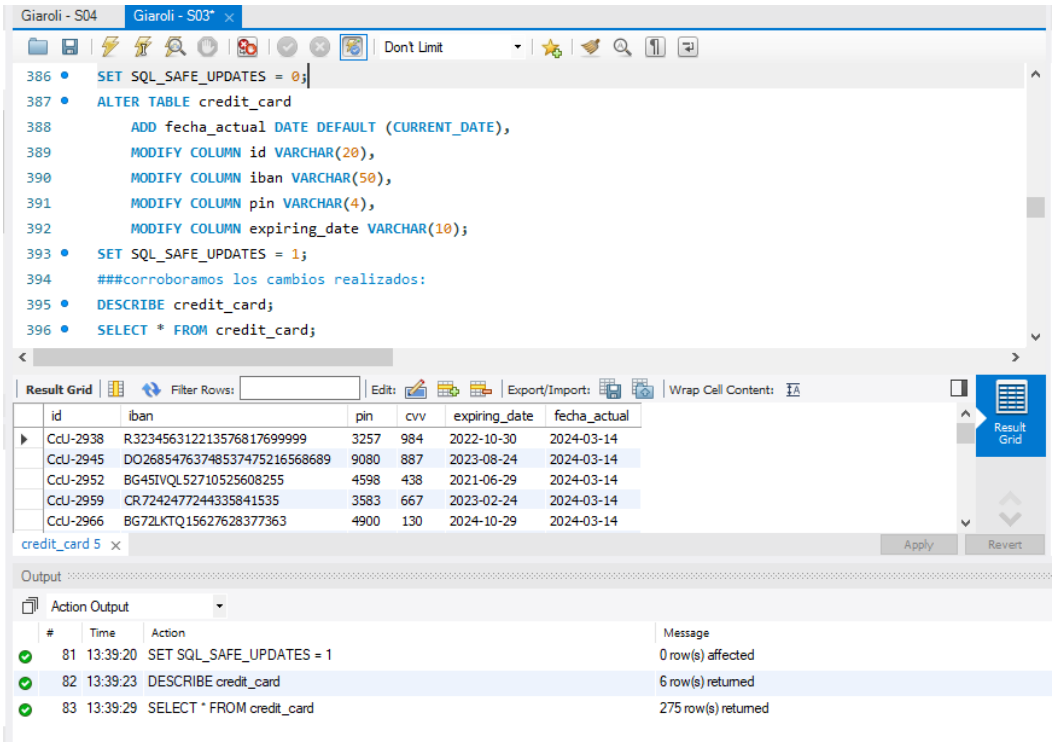
Ejercicio 1

Para obtener el modelo requerido haremos los siguientes cambios:

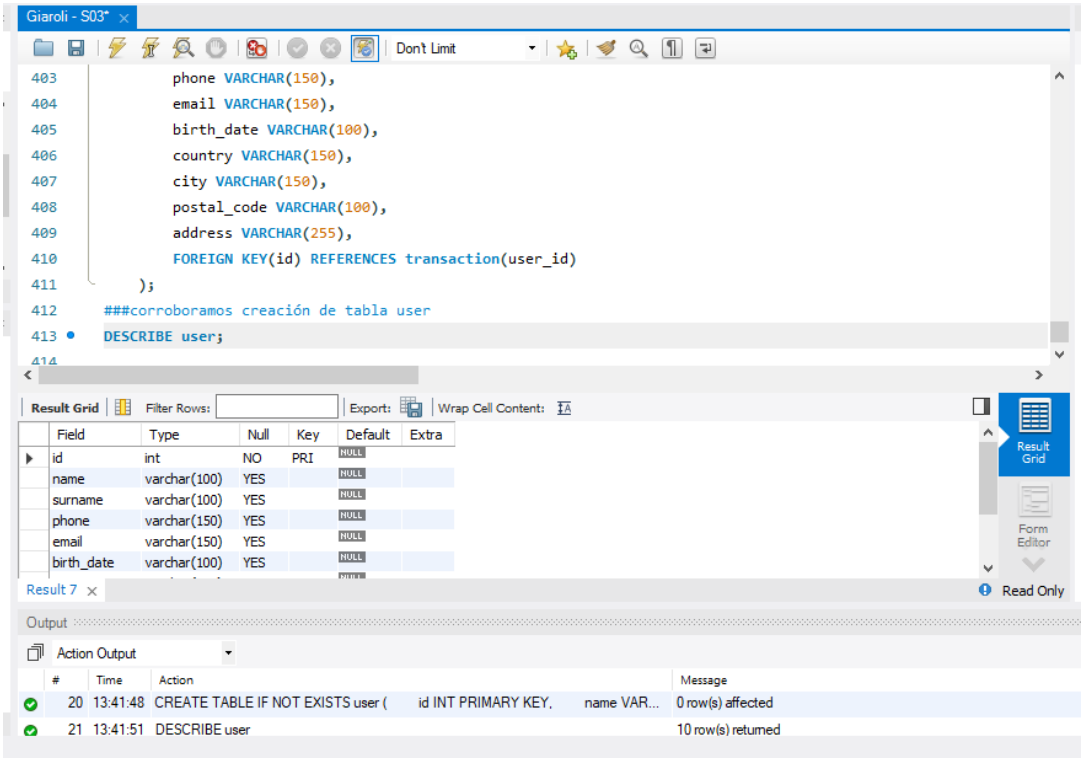
1) eliminamos el campo website de la tabla company



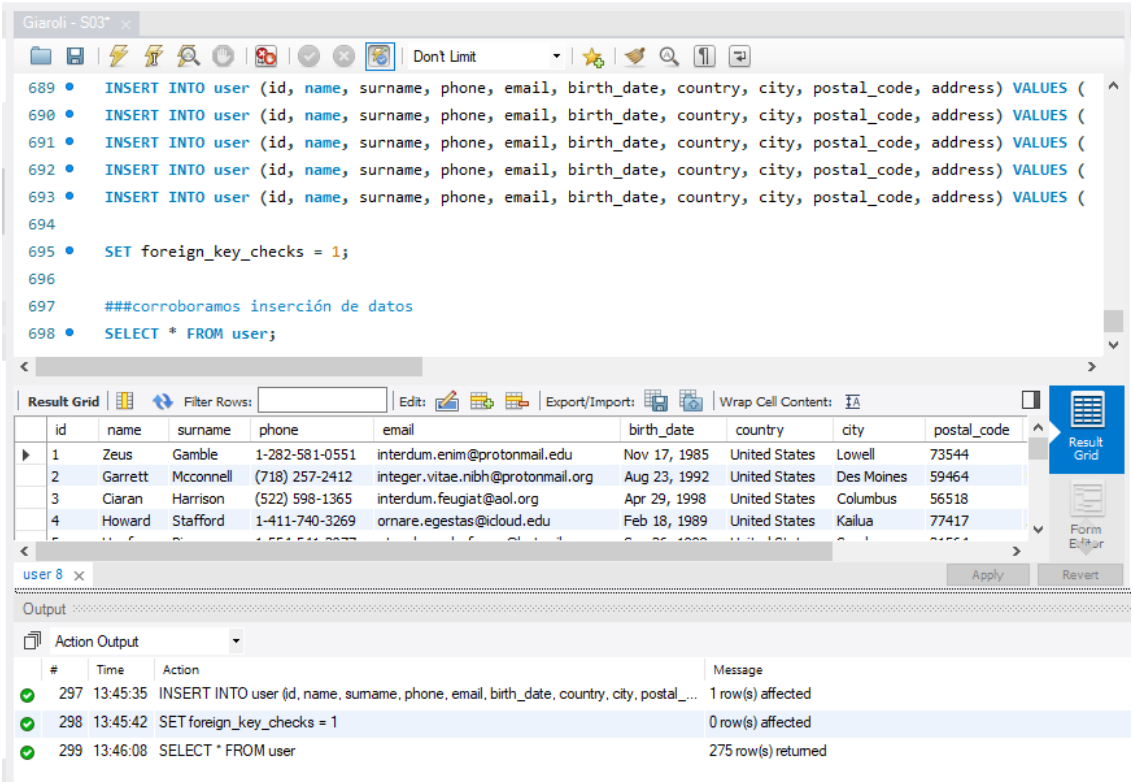
2) en la tabla credit_card cambiamos el formato de 4 campos y añadimos uno nuevo.



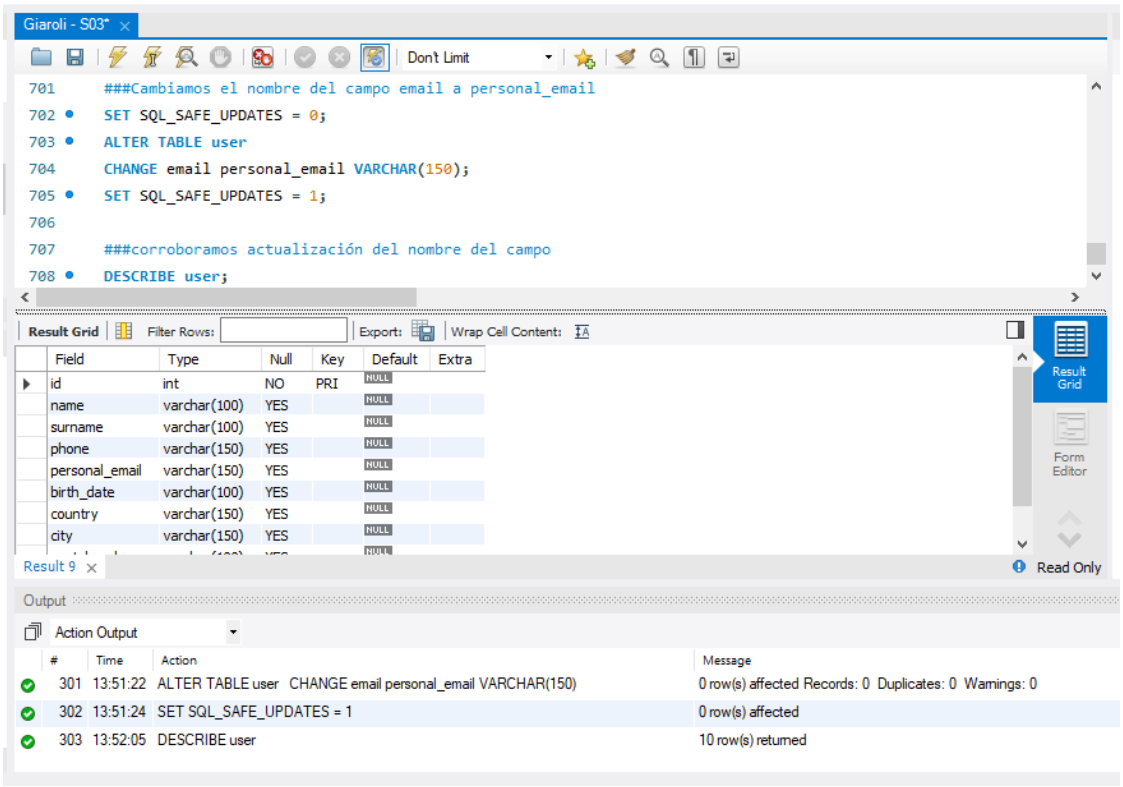
3) Creamos la tabla User



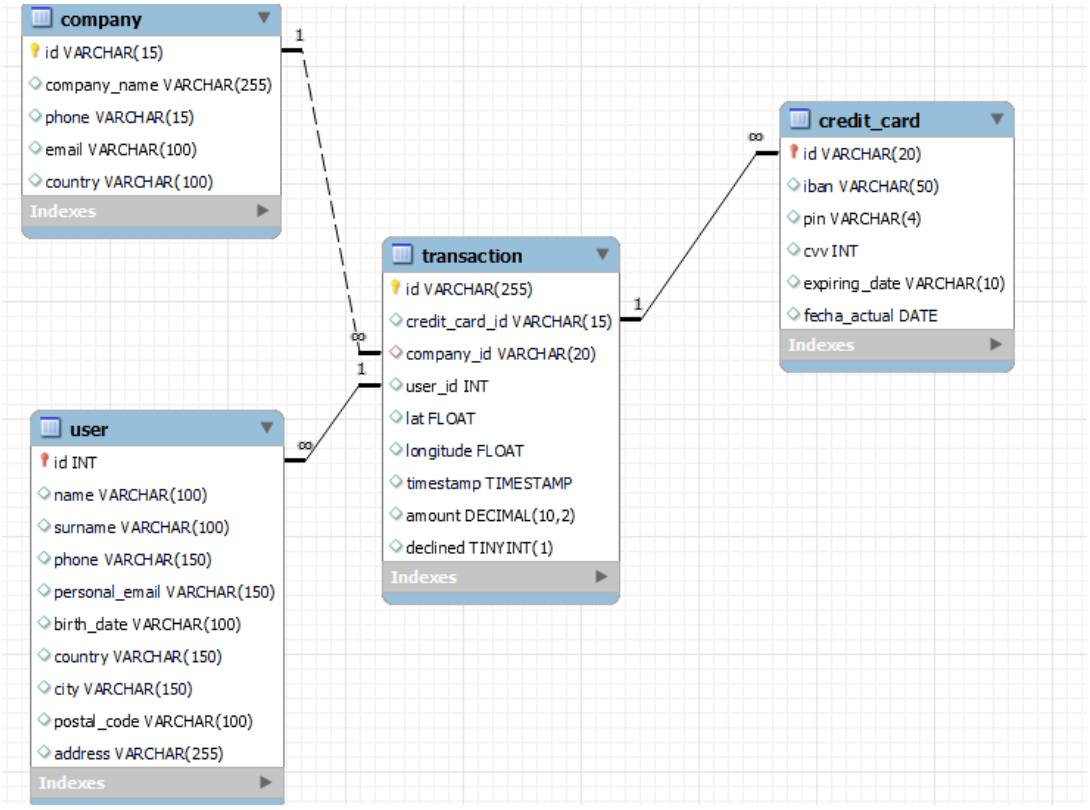
a. Insertamos los datos



b. Cambiamos el nombre del campo email a personal_email

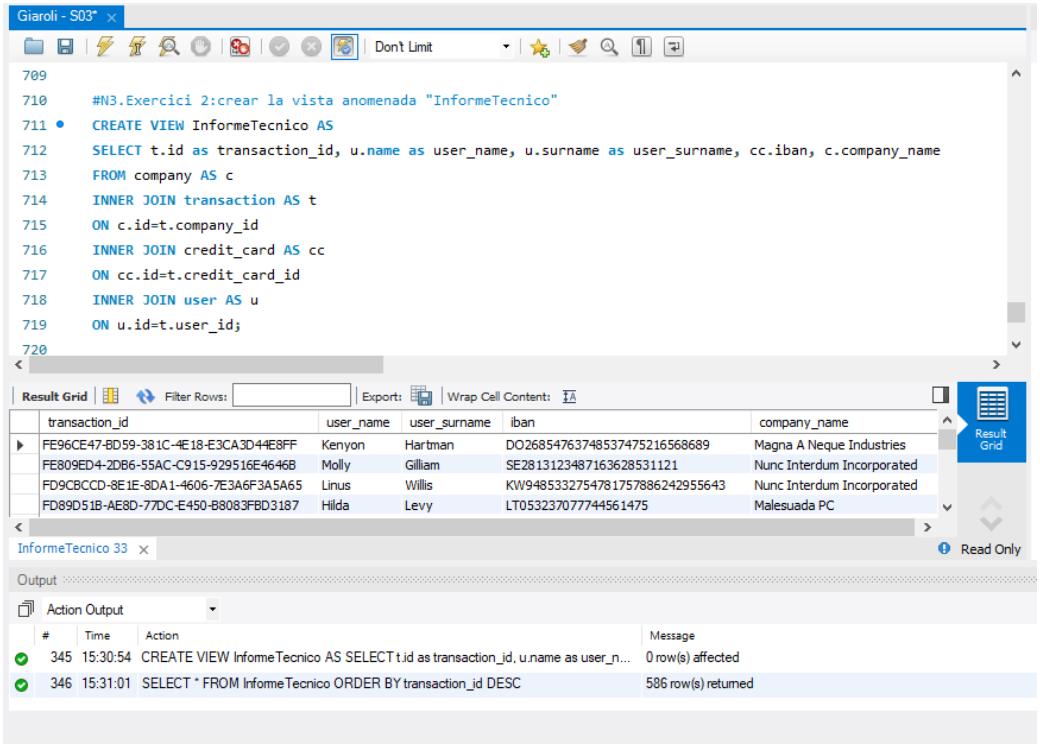


4) Establecemos las relaciones entre tablas como el modelo requerido y eliminamos del dibujo del Diagrama: la VistaMarketing y el Indice de la FK de la tabla transaction para que los diagramas coincidan. Adjuntamos el archivo del diagrama al GitHub.

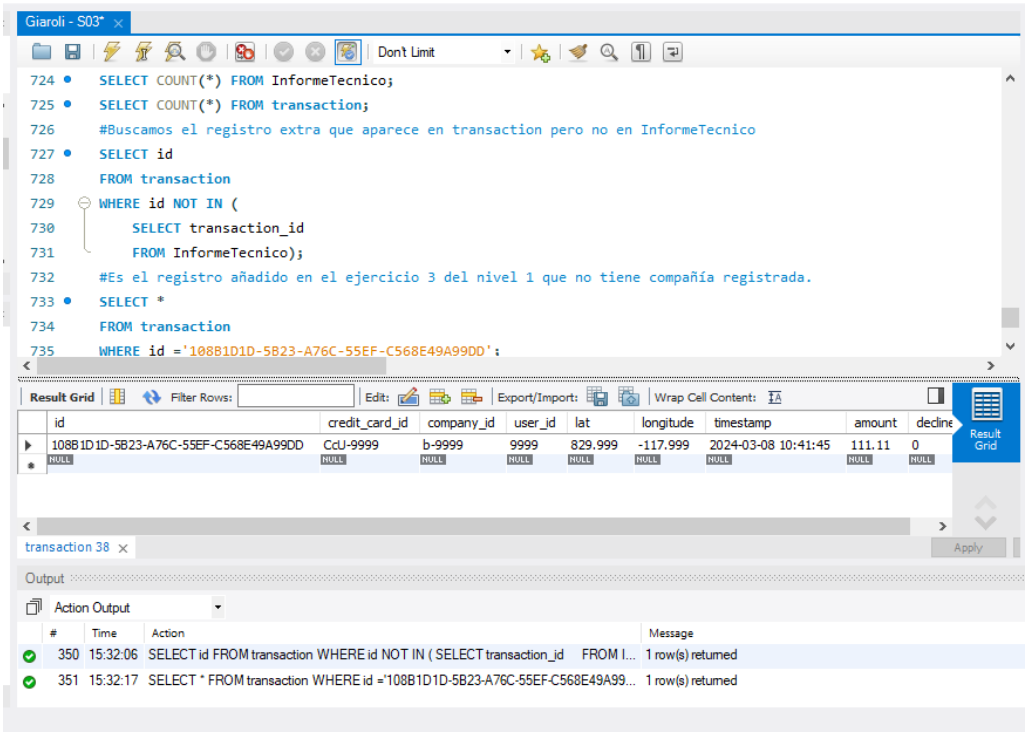


Ejercicio 2

Creamos la vista InformeTecnico con la siguiente consulta:

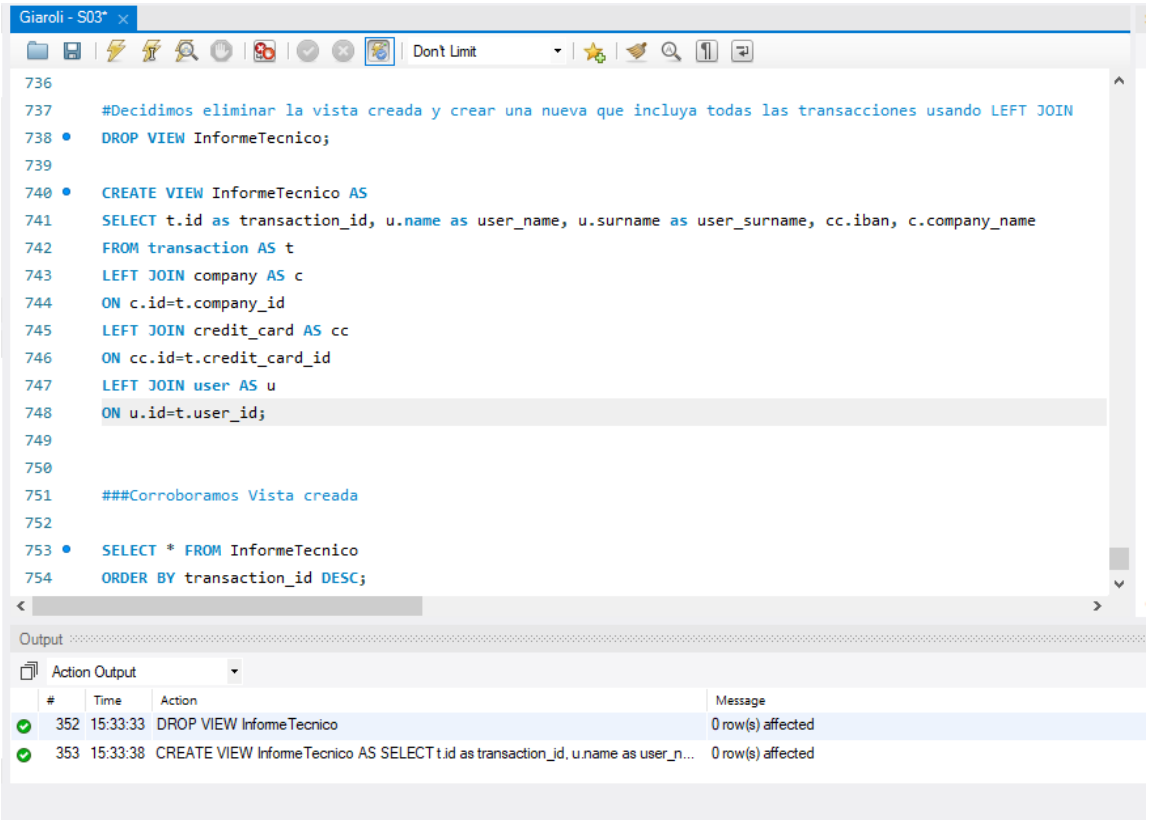


Corroboramos que la cantidad de datos de la tabla transaction coincida con la de InformeTecnico para verificar que no haya errores.



Encontramos que Transaction tiene un registro más que InformeTecnico, es el registro añadido en el ejercicio 3 del nivel 1.

Decidimos eliminar la vista creada y crear una nueva haciendo una Left Join para incluir todas las transacciones:



Verificamos la vista creada: