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 mesa tiene que ser capaz de identificar de manera única cada tarjeta y establecer una relación adecuada con las otras dos mesas ("transaction" y "compañero"). Después de crear la tabla será necesario que ingreses la información del documento denominado "datos_introducir_credit". Recuerda mostrar el diagrama y realizar una breve descripción de este.

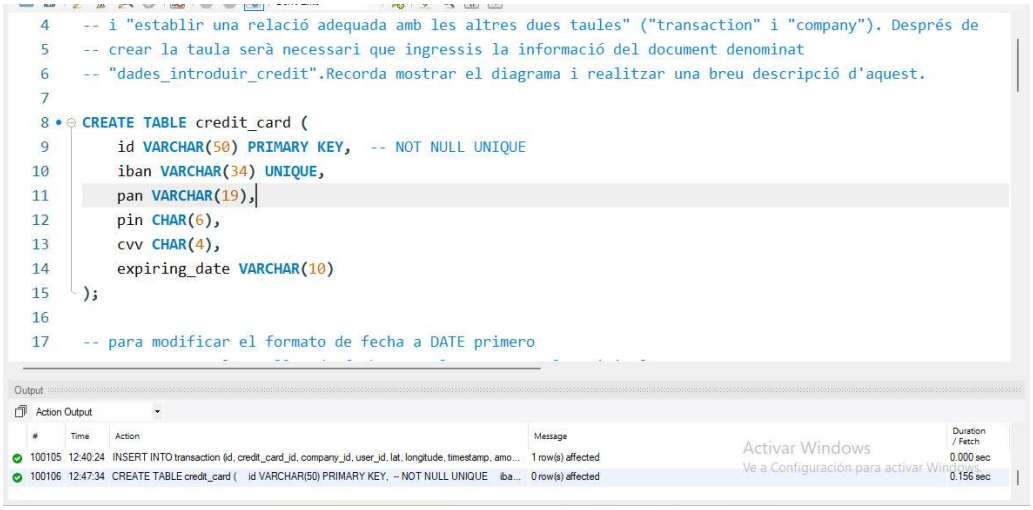

Figura 1.Creación de tabla credit_card

Tabla 1 .Estándares internacionales de longitudes de datos bancarios.

Campo	Estándar	Longitud / reglas
IBAN	ISO 13616	hasta 34 caracteres
PAN	ISO/IEC 7812	hasta 19 dígitos
IIN/BIN	ISO/IEC 7812	6–8 dígitos
PIN	ISO 9564	4–12 dígitos (general: 4 o 6)
CVV/CVC	PCI DSS + reglas de las marcas	3 (Visa/Mastercard), 4 (Amex)

Campo	**Estándar**	**Longitud / reglas**
Expiración	PCI DSS / EMV	MM/YY
BIC/SWIFT	ISO 9362	8 o 11 caracteres

El código crea la tabla **credit_card** en MySQL definiendo cada campo con su tipo y restricciones: un id como clave primaria, un iban único, y valores como PAN, PIN, CVV y fecha de expiración con longitudes basadas en estándares oficiales. La estructura está preparada para añadir relaciones con las tablas **transaction** y **company**, formando un modelo de datos organizado y consistente.

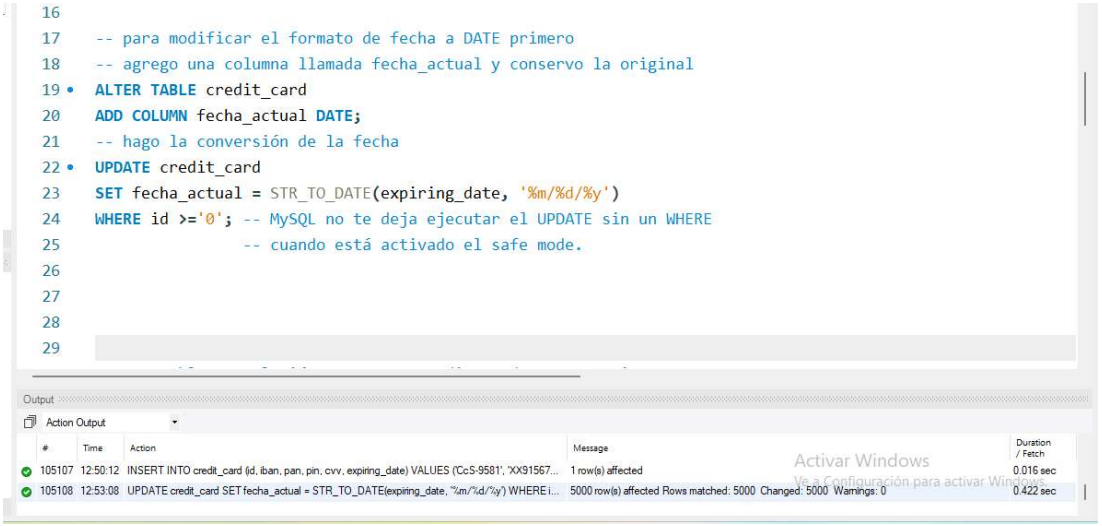

*Figura 2. Agregación de columna **fecha_actual** y converción de fecha al formato DATE.*

Se agregó columna **fecha_actual** como tipo DATE y convirtió la fecha almacenada en texto mediante STR_TO_DATE. Para poder ejecutar el UPDATE incluyó un WHERE, ya que el *safe mode* de MySQL lo exige.

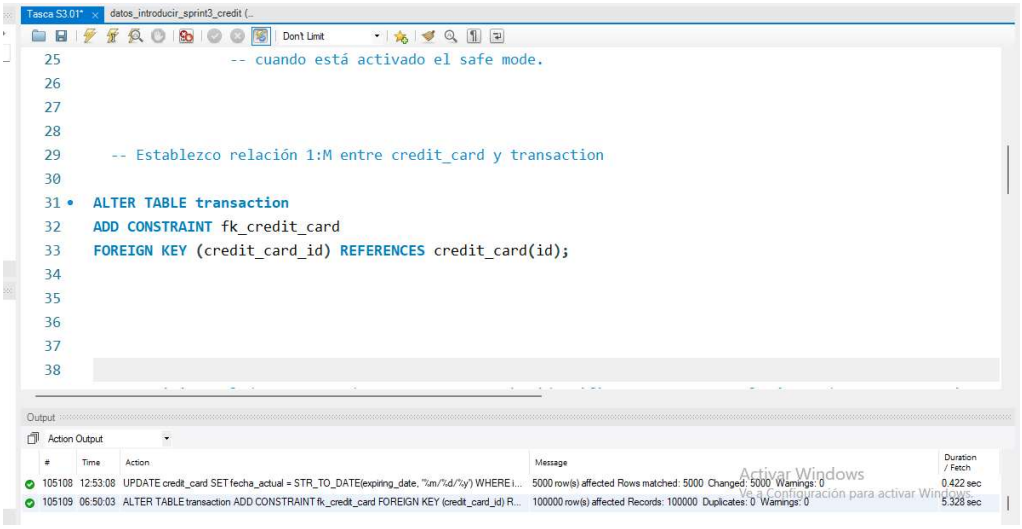

*Figura 3. Creación de relación entre la tabla **transaction** y **credit_card***

Se estableció la relación entre la tabla **transaction** y **credit_card** agregando una clave foránea mediante ALTER TABLE. Para ello creó la restricción **fk_credit_card**, la cual enlaza el campo credit_card_id de *transaction* con el campo id de *credit_card*. Con esta acción definió una relación de **uno a muchos**, donde una tarjeta puede tener múltiples transacciones asociadas, garantizando la integridad referencial dentro de la base de datos.

EJERCICIO 2

El departamento de Recursos Humanos ha identificado un error en el número de cuenta asociado a la tarjeta de crédito con ID CcU-2938. La información que tiene que mostrarse para este registro es: TR323456312213576817699999. Recuerda mostrar que el cambio se realizó.

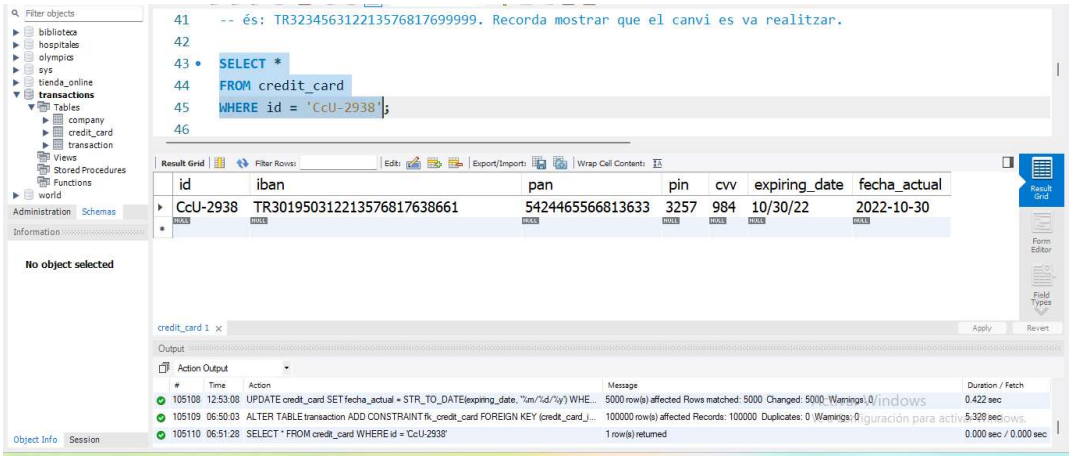

Figura 4. Identificación del error en el IBAN de la tarjeta con ID ***CcU-2938***

Detectado el error en el IBAN de la tarjeta con ID **CcU-2938**, figura 4, se procedió a corregirlo mediante una sentencia UPDATE que reemplazó el valor anterior por el nuevo número proporcionado por Recursos Humanos. Después ejecutó un SELECT filtrado por el mismo ID para comprobar que el cambio se aplicó correctamente., ver figura 5. Al visualizar nuevamente el registro en la tabla *credit_card*, pudo confirmar que el IBAN fue actualizado sin afectar el resto de los datos asociados a la tarjeta.

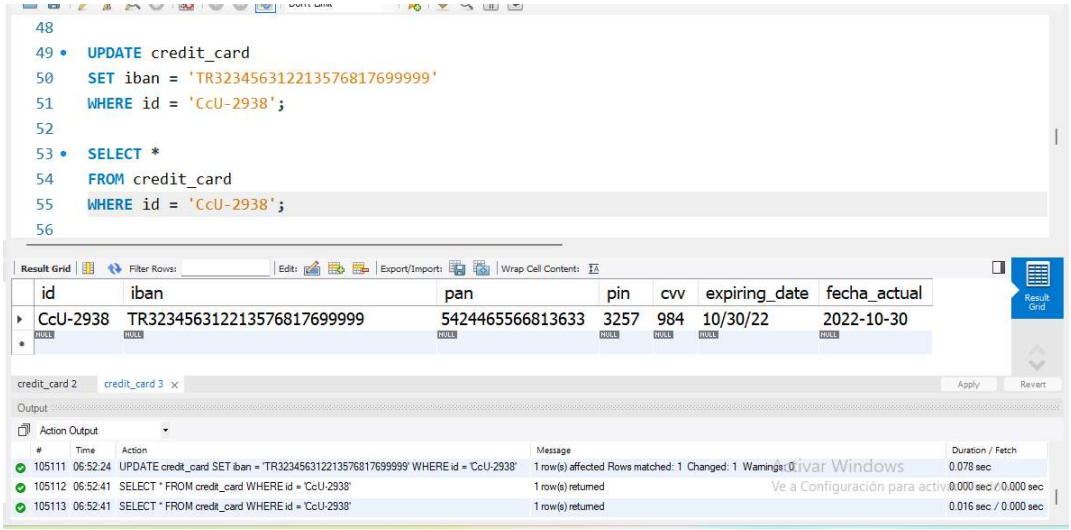

Figura 5 cambio datos ya existentes en registro específico de una tabla credit_card y comprobación de cambios..

EJERCICIO 3

En la tabla "transaction" ingresa una nueva transacción con la siguiente información:

Tabla 2. Datos de transacción a ingresar.

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

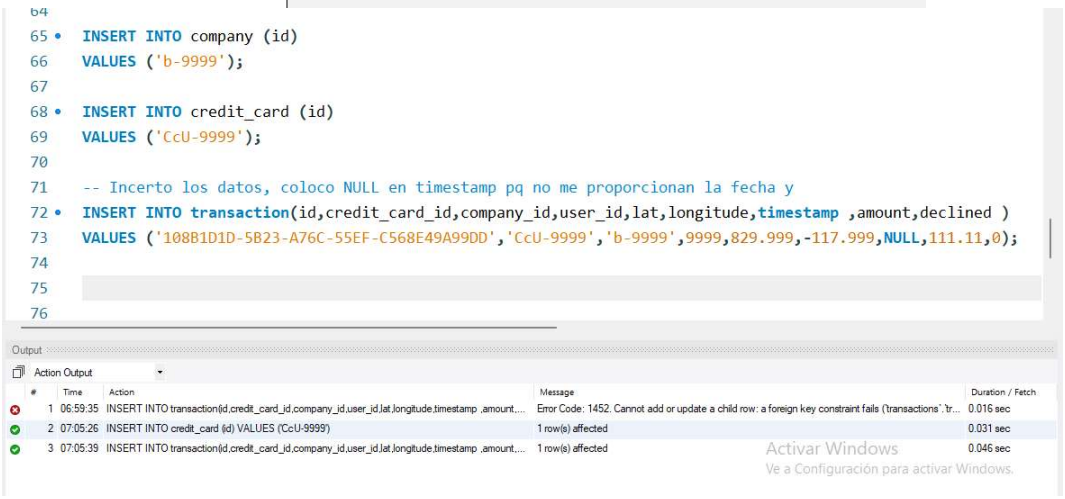

Figura 6. Ingreso de nueva transacción

Se intentó insertar el registro inicialmente, pero el sistema devolvió el error 1452, lo que indicó que todavía no existían los valores correspondientes al credit_card_id o company_id requeridos por la transacción. Para cumplir con las claves foráneas, primero se insertaron los datos de una tarjeta en la tabla *credit_card* con el id 'CcU-9999' y los de una compañía en la tabla *company* con el id 'b-9999'. Una vez creados estos registros, se añadió la nueva transacción mediante INSERT INTO transaction, verificando que credit_card_id y company_id coincidieran con los valores ya existentes. Al corregir el orden de inserción, se evitó el error de integridad referencial y se pudo registrar

correctamente la operación con todos sus campos, incluyendo coordenadas, monto e indicador de rechazo.

EJERCICIO 4

Desde recursos humanos te solicitan eliminar la columna "pan" de la mesa credit_cardo. Recuerda mostrar el cambio realizado.

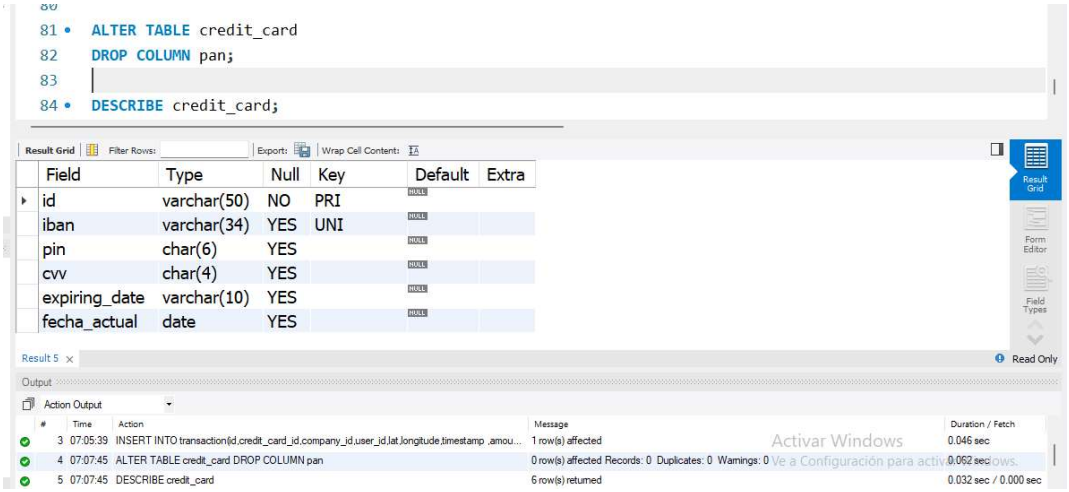

Figura 7.- Eliminación de la columna PAN

Para atender la solicitud de Recursos Humanos, se procedió a eliminar la columna pan de la tabla *credit_card* utilizando el comando ALTER TABLE credit_card DROP COLUMN pan;. Después se ejecutó DESCRIBE credit_card para mostrar la estructura actualizada de la tabla y confirmar que la columna había sido removida correctamente, tal y como lo muestra la figura 7. Con esto, dejó la tabla ajustada a los nuevos requerimientos sin afectar el resto de los datos existentes.

NIVEL 2

EJERCICIO 1

Elimina de la Tabla transaction el registro con ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de datos.

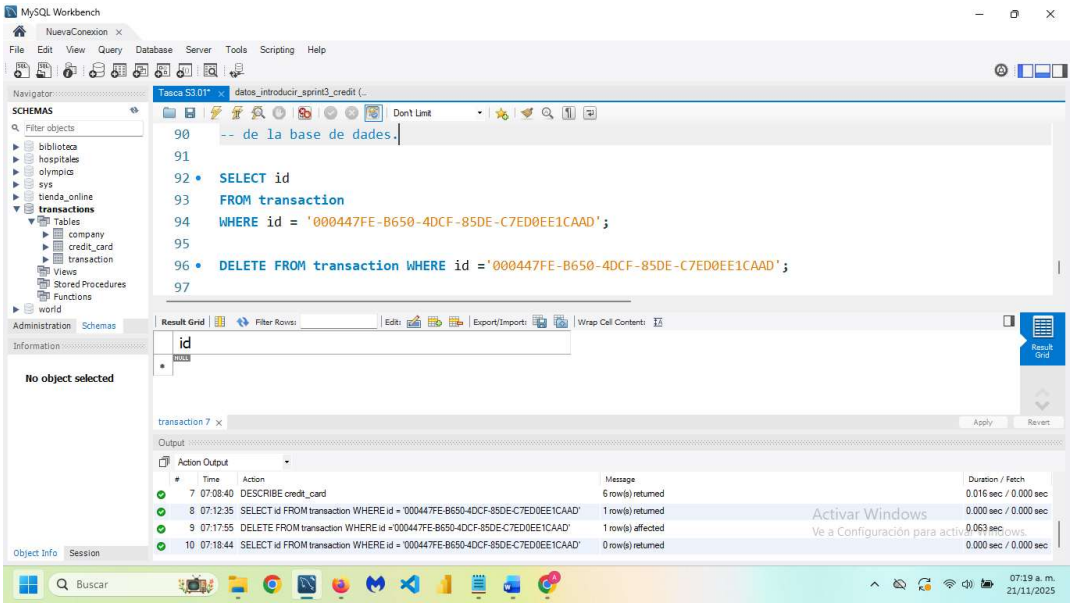

Figura 8. Eliminación de la Tabla transaction registro con ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD

Se procedió a localizar el registro de la tabla transaction donde id era 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD mediante una consulta SELECT, verificando así que el dato existía. Una vez confirmado, ejecutó la sentencia DELETE FROM transaction con el mismo identificador para eliminarlo de la base de datos. Finalmente se volvió a ejecutar un SELECT para comprobar que el registro ya no aparecía, ver figura 8, confirmando que la eliminación se realizó correctamente.

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. Media de compra realizado por cada compañía. Presenta la vista creada, ordenando los datos de mayor a menor media de compra.

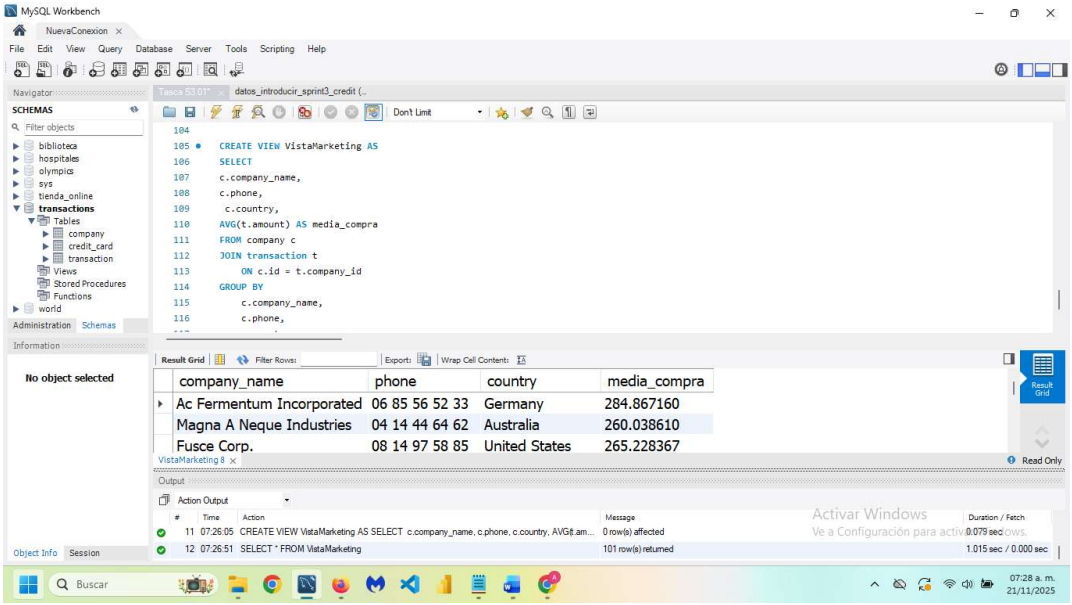

Figura 9. Creación de vista VistaMarketing

En la figura 9 se observa que se creó la vista VistaMarketing para proporcionar al área de marketing información relevante sobre cada compañía y sus transacciones. La vista muestra el nombre de la empresa, su teléfono de contacto, el país de residencia y la media de compra obtenida a partir de sus transacciones, utilizando una combinación de las tablas company y transaction. Después de generar la vista, se ordenó los resultados de mayor a menor según la media de compra, permitiendo que el departamento de marketing visualice de inmediato qué compañías realizan mayores volúmenes de compra y facilitando así sus análisis y estrategias.

EJERCICIO 3

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

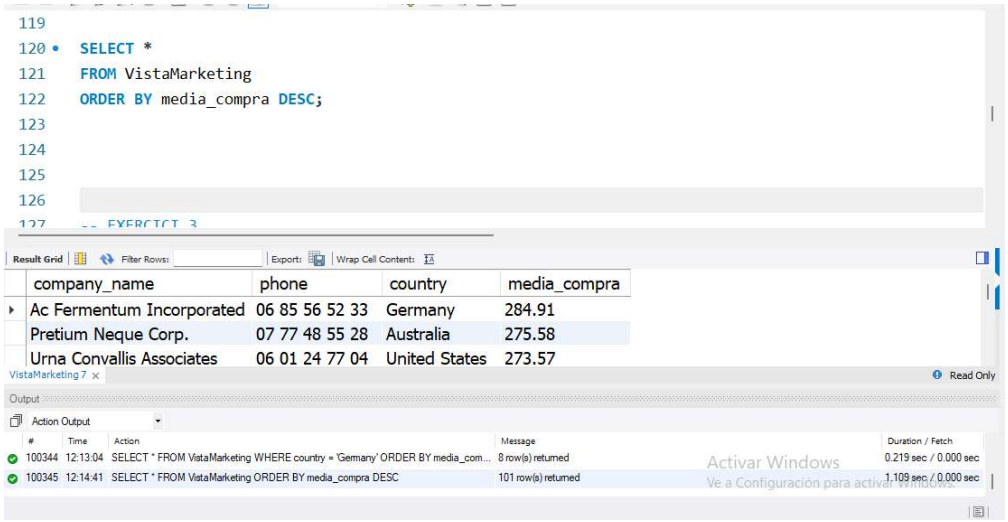

Figura 10 Filtro de la vista VistaMarketing por país "Germany".

Filtró la vista **VistaMarketing** para mostrar exclusivamente las compañías cuyo país de residencia es *Germany*, utilizando una sentencia SELECT con la condición WHERE country = 'Germany'. Luego ordenó los resultados de mayor a menor según la media de compra.

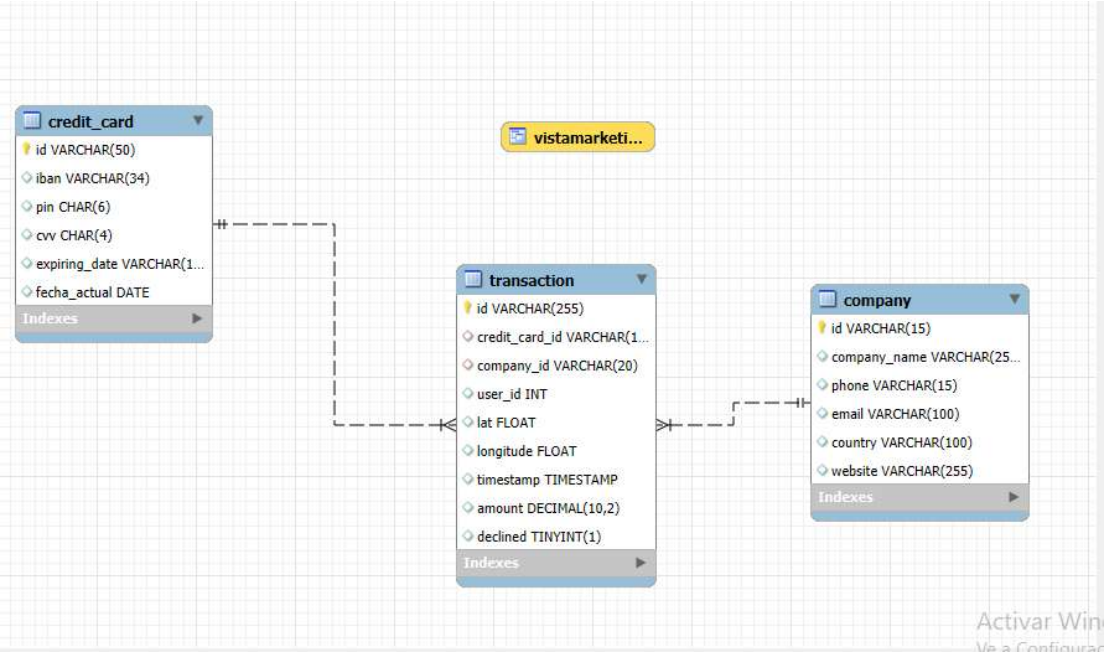

Figura 11. Diagrama resultado del nivel 2

NIVEL 3

Ejercicio 1

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

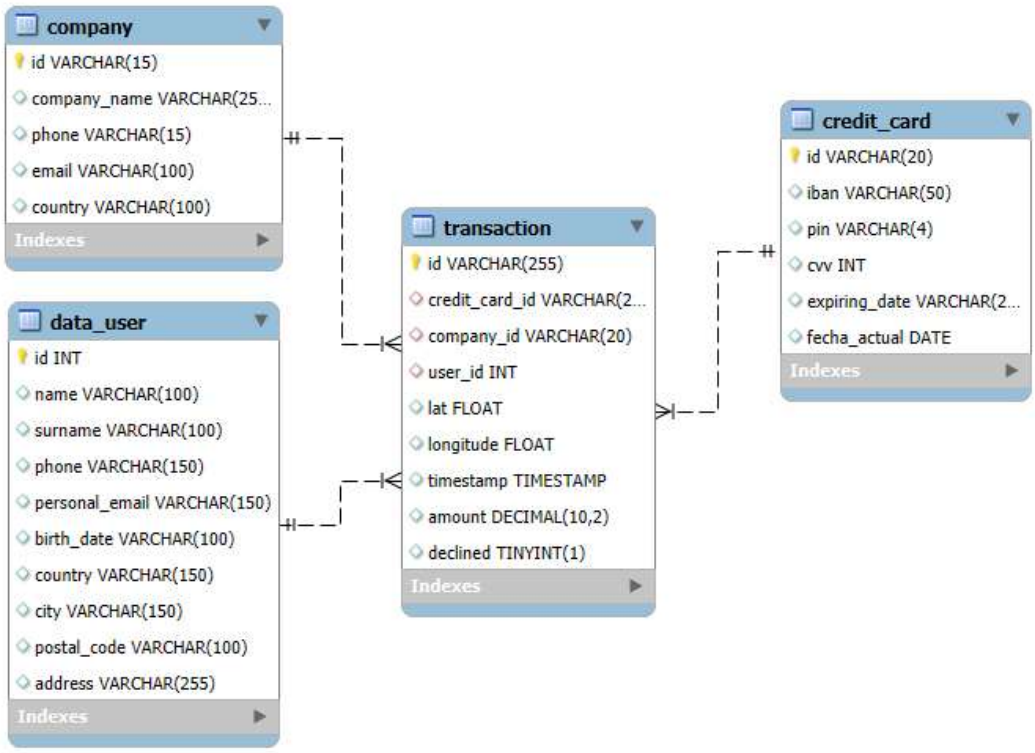

Figura 12.Diagrama ejercicio nivel 3

Para verificar los cambios realizados se verificó el diagrama actual con el diagrama resultados hechos hasta el nivel 2. Evidenciándose la creación de una niueva tabla inicialmente, data_user

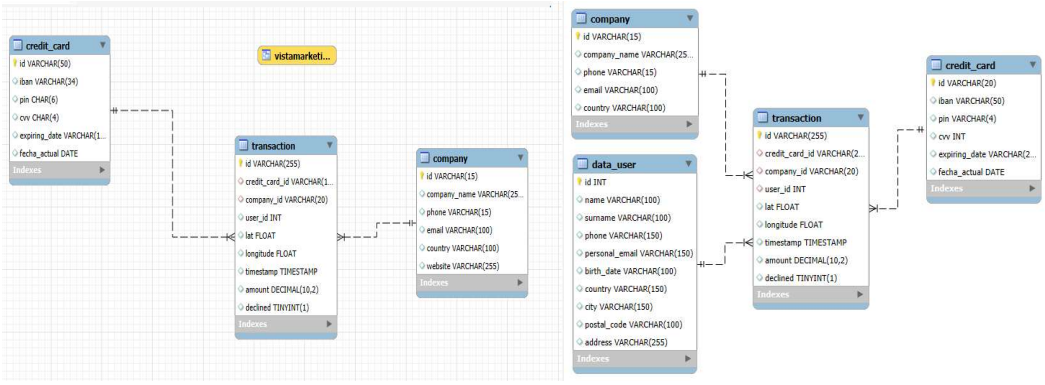

Figura 13. Diagramas antes y después de los cambios.

Se evidenciaron los siguientes cambios entre los diagramas, ver tabla 3

Tabla 3. Cambios realizados al diagrama 1 evidenciados en diagrama 2

Elemento	Diagrama 1	Diagrama 2	**Cambio realizado**
Nueva tabla agregada	No existe tabla data_user	Aparece tabla data_user	**Se agregó una nueva entidad** llamada data_user con información personal del usuario.
Campos de la tabla data_user	—	id, name, surname, phone, personal_email, birth_date, country, postal_code, address	**Nueva estructura completa** de datos del usuario.
Relación transaction → data_user	transaction.user_id existe pero no tenía referencia visual	transaction.user_id se conecta claramente a data_user.id	**Se formalizó la relación** entre transacciones y usuarios.
Tabla credit_card	id (VARCHAR50), iban (VARCHAR24), pin (CHAR5), cvv (CHAR5), expiring_date VARCHAR/fecha_actual DATE	id VARCHAR(20), iban VARCHAR(50), pin VARCHAR(4), cvv INT, expiring_date VARCHAR(20), fecha_actual DATE	**Cambios en tipos de datos y longitudes** (reducción en id, ampliación de iban, ajuste de pin y cvv).
Tabla company	Campos iguales pero sin orden estandarizado	Igual estructura pero organizada en nuevo orden No aparece el campo website	**Reorganización de campos**, sin cambios de estructura. Se borra el campo website
Relación transaction → company	Existía vínculo	Se mantiene vinculo	**Se mantiene la relación**
Tipos de datos en transaction	id VARCHAR(255), lat FLOAT, long FLOAT, timestamp TIMESTAMP, amount DECIMAL(10,2), declined TINYINT(1)	Igual, sin cambios	**Sin modificaciones**.
Relación transaction → credit_card	credit_card_id VARCHAR(?) apuntaba a credit_card.id	Se mantiene igual	**Sin cambios**.

Se comenzó creando la data data_user, sin embargo, inicialmente no se creó con ese nombre, sino con la estructura proporcionada en los archivos adjuntos del ejercicio para poder introducir los valores proporcionados y luego hacer los cambios sobre la tabla user, ver figura 14.

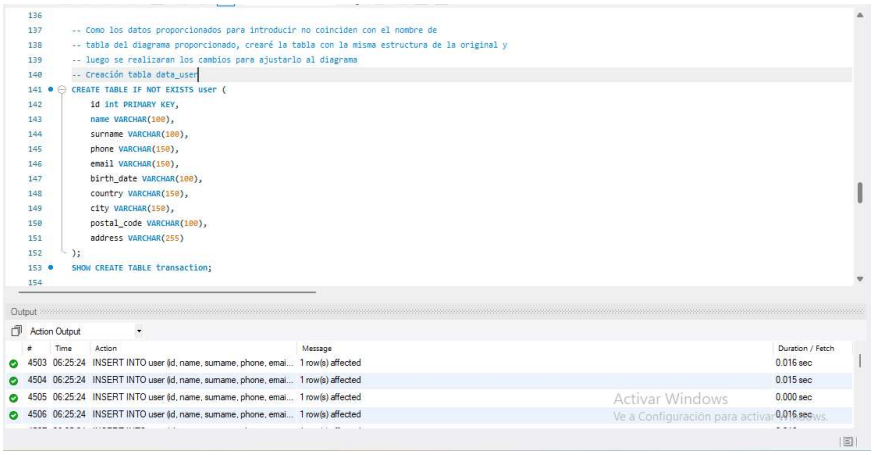

Figura 14. Creación de tabla user

En la figura 15 podemos observar el cambio del nombre de la tabla user a data_user y de la columna email a personal_email

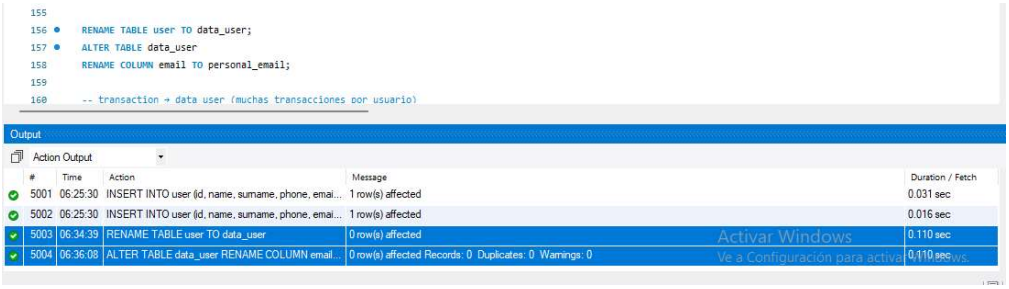

Figura 15.Cambio de nombre de tabla user a data_user y de la columna email a personal_email

En la figura 16 se puede observar la creación de la relación transaction →data_user (muchas transacciones por usuario).

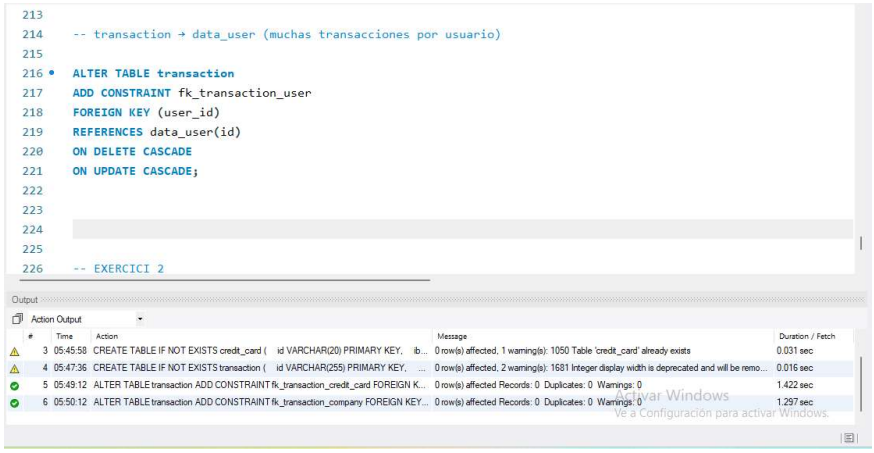

Figura 16.Creación de relación transaction → data_user (muchas transacciones por usuario)

El siguiente paso fue hacer los cambios en la tabla credit_card estos fueron de tipo de datos y longitudes, ver figura 17.

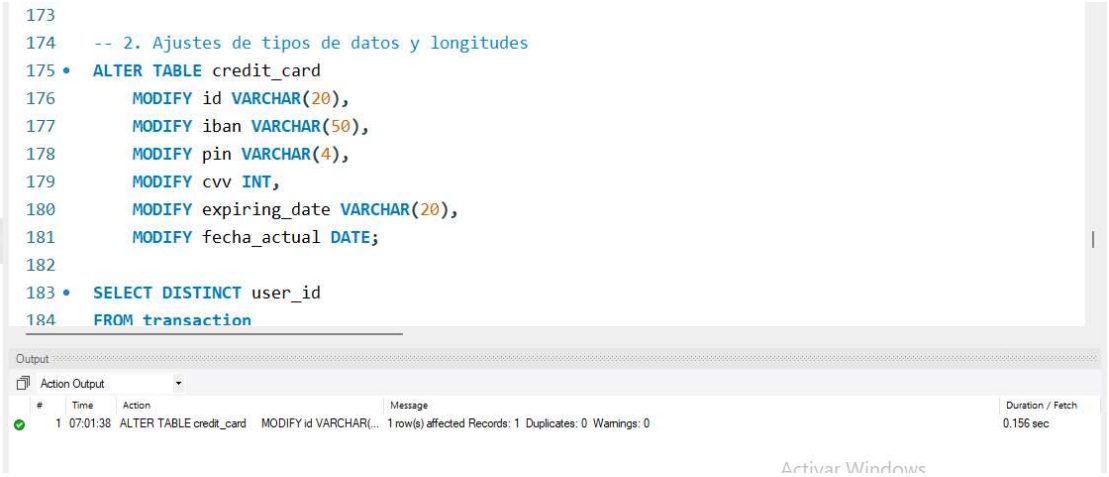

Figura 17. Modificación de longitudes y tipo de datos

Se evaluó la integridad referencial garantizar que las relaciones entre tablas sean **coherentes y válidas**, debido a que en ejercicios anteriores se realizaron modificaciones y se necesita saber si hay datos huérfanos, ver código en figura 18. En este caso se encontró que en la tabla data_user no está el id =9999 y se procede a insertar, ya que la tabla padre (data_user) debe tener los mismos id que en la tabla hija (transaction) y asi crear la FK.

Es importante destacar que inicialmente se había presentado la situación de que se pudo crear la FK entre data_user y transaction sin hacer la validación referencial lo que no debe suceder si están activas las FK. Es por este sentido se borra la FK creada inicialmente y

se activan las restricciones de las Fk y se realiza correctamente la adición de la FK correctamente, ver figura 19.

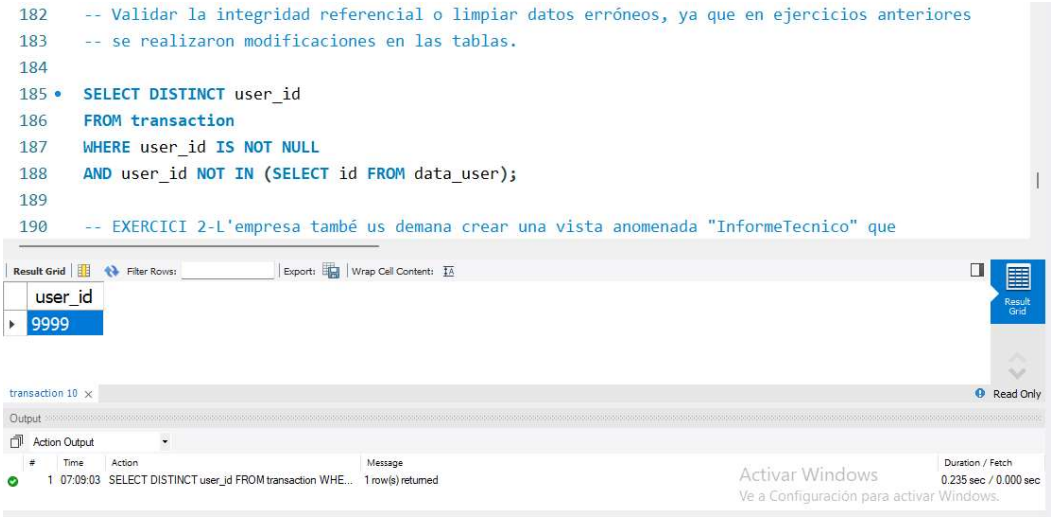

Figura 18. Evaluación de integridad referencial

Evidentemente no me debió dejar crear la FK y es por eso que se evidencia que estaban desactivadas las restricciones de foreign key ya que obliga a que cada transaction.user_id exista en data_user.id y por tanto daría error.

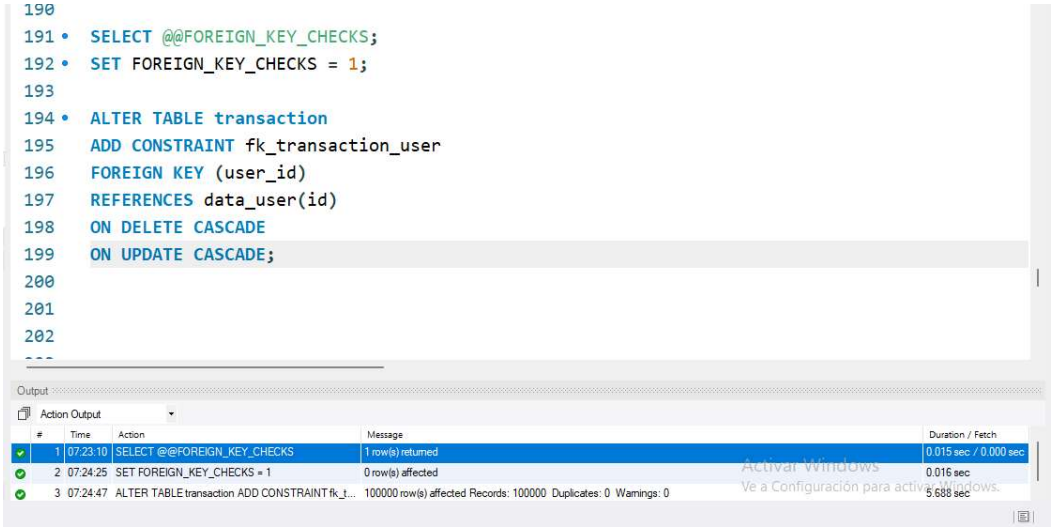

Figura 19. Activación de las restricciones de la FK y creación de la FK entre transación y data_user

Figura 20. Drop de la columna website

Adicionalmente en los cambios se puede observar que en la tabla company no existe el campo website

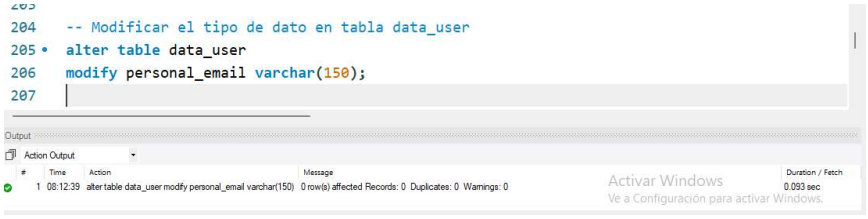

Figura 21. Modificación de longitud de dato en la tabla data_user del campo personal_email

En la tabla 21 podemos observar el cambio de la longitud de dato de personal_email

EJERCICIO 2.

La empresa también os pide crear una vista llamada "InformeTecnico" que contenga la siguiente información:

- ID de la transacción
- Nombre del usuario/aria
- Apellido del usuario/aria
- IBAN de la tarjeta de crédito usada.
- Nombre de la compañía de la transacción realizada.

Aseguraos de incluir información relevante de las tablas que conoceréis y utilizáis alias para cambiar de nombre columnas según haga falta.

Muestra los resultados de la vista, ordena los resultados de forma descendente en función de la variable ID de transacción.

Se creó la vista y se llamó para mostrar la tabla con los resultados. Ver figura 22 y 23.

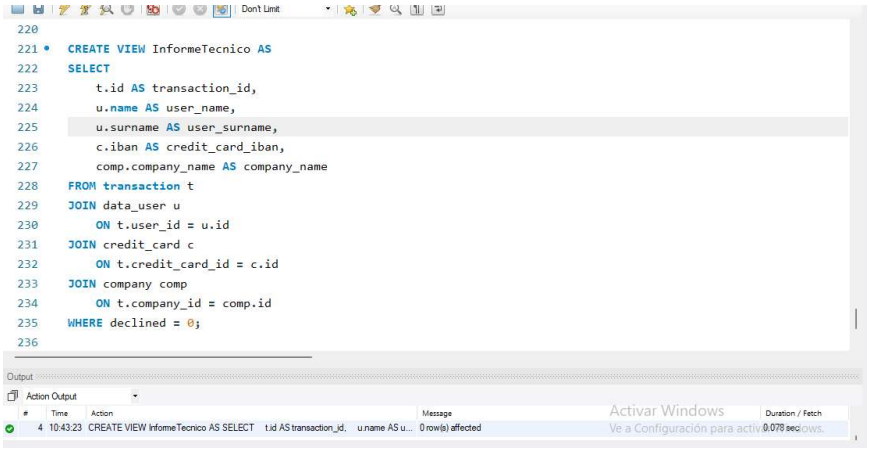

Figura 22. Creación de la vista "InformeTecnico"

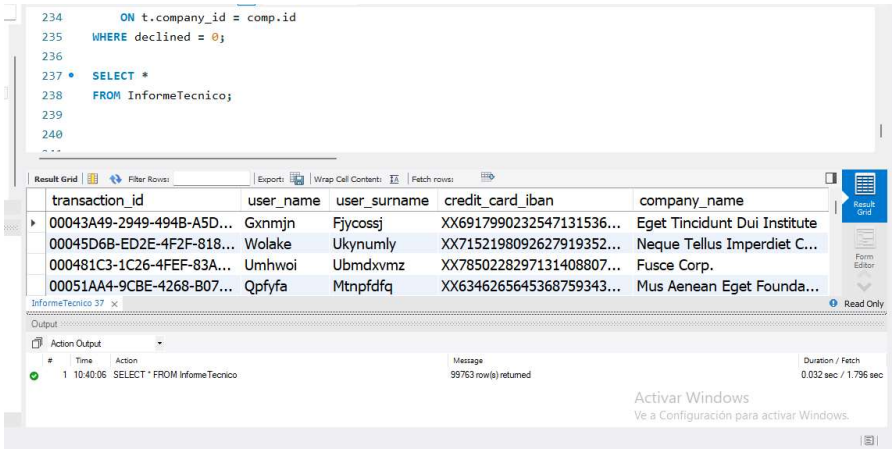

Figura 23. Muestra de "InformeTecnico"

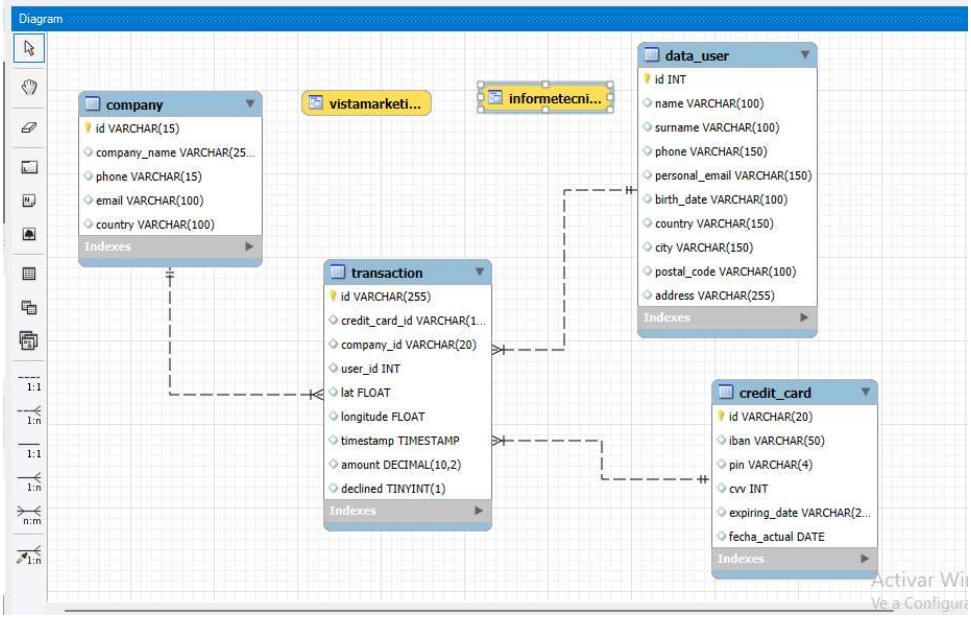

Figura 24. Diagrama Nivel 3