Tarea S3.01. Manipulación de Tablas

En este sprint se simula una situación empresarial en la cual deberás realizar varias manipulaciones en las tablas de una base de datos. Además, trabajarás con índices y vistas para optimizar consultas y organizar la información.
Continuarás trabajando con la base de datos que contiene información de un marketplace, un entorno similar a Amazon donde varias empresas venden sus productos a través de un canal en línea. En esta actividad, comenzarás a trabajar con datos relacionados con tarjetas de crédito.

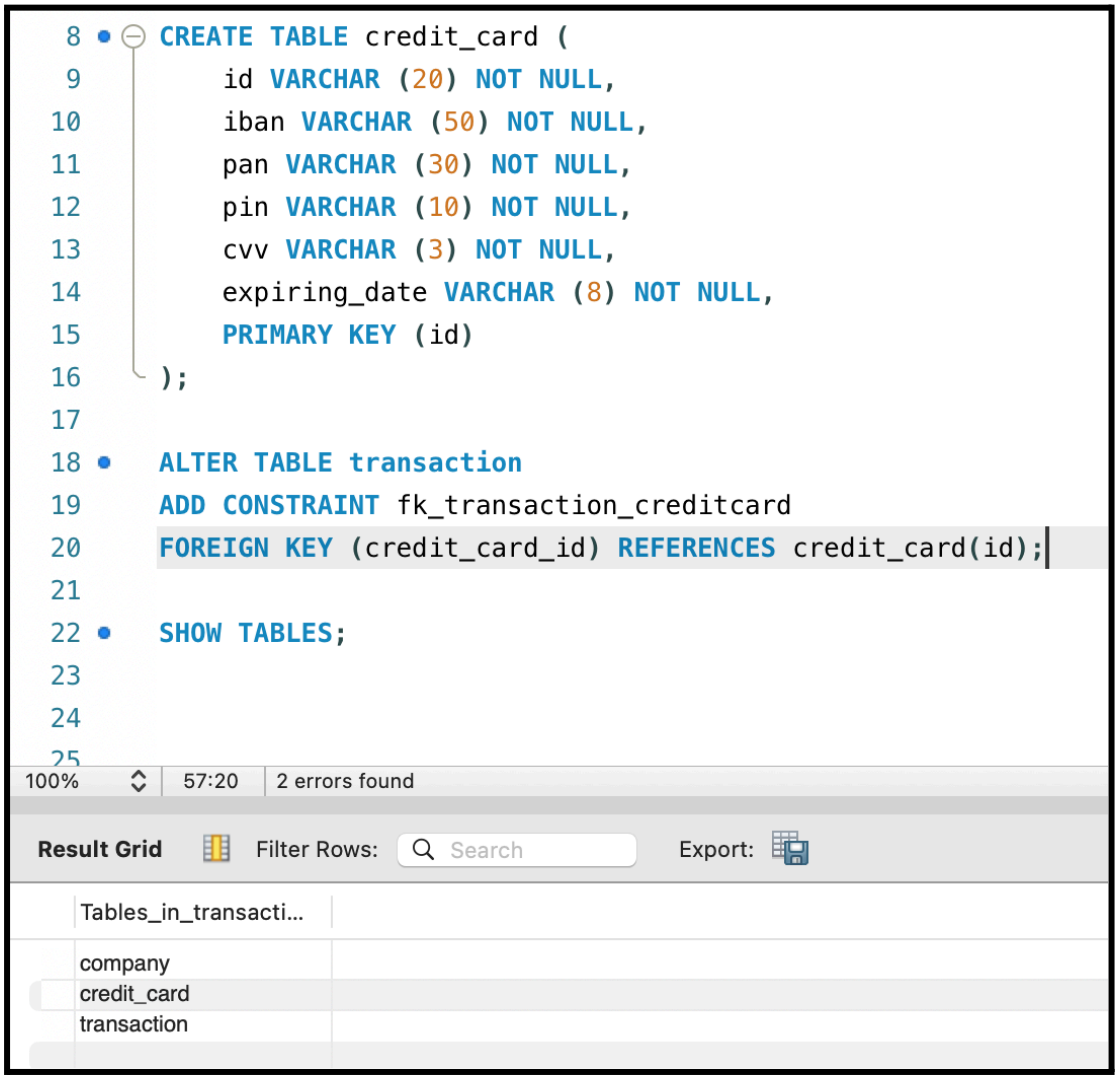
Agrega las tablas al modelo según corresponda:

- Nivel 1: Tabla "credit_card"
- Nivel 3: Tabla "user"

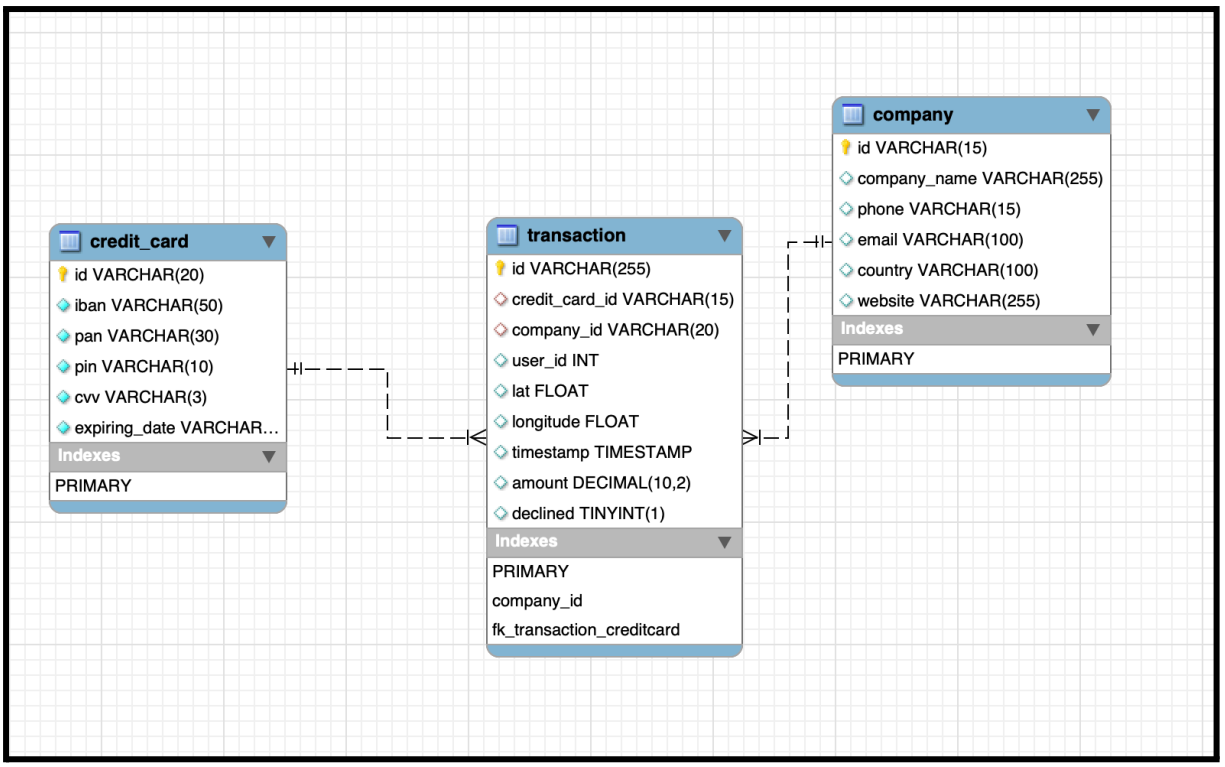
Nivel 1

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 de este.



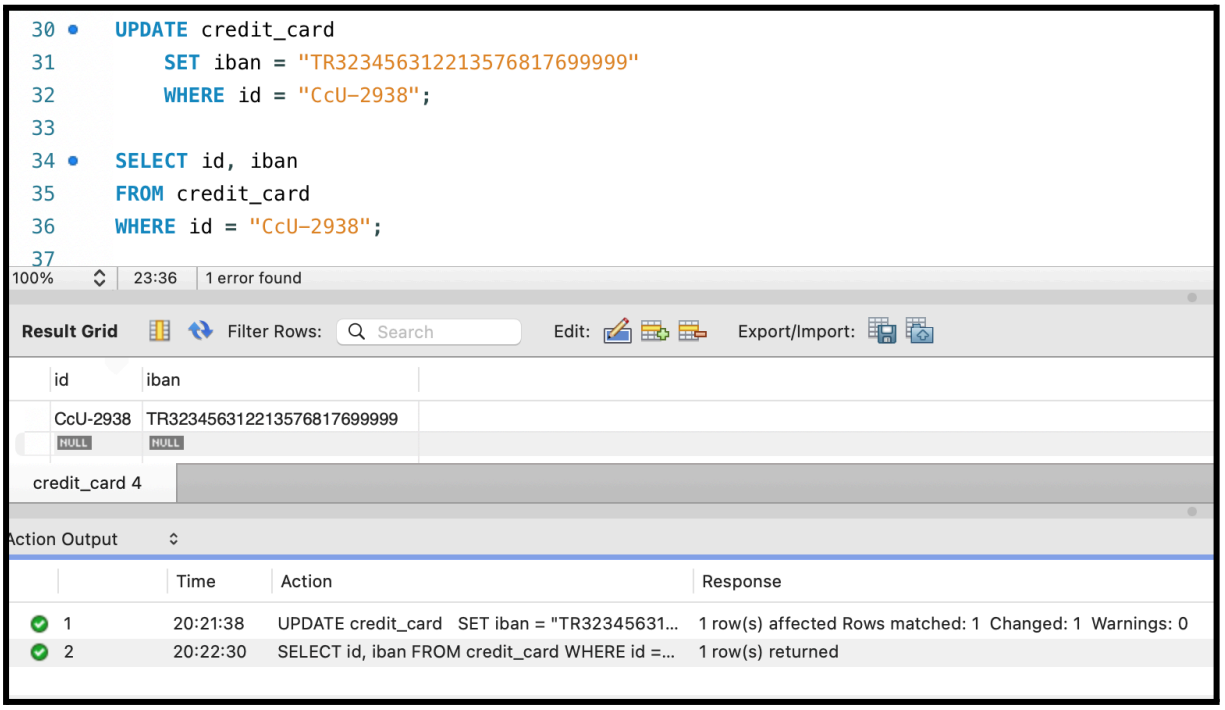
Para crear la tabla *credit_card* utilice el comando ***CREATE TABLE***, detallando los campos que quiero que incluya con los respectivos tipos de datos e indicando que el campo *id* sea la ***PRIMARY KEY***. Para saber que campos introducir me fije en el script "dades_introduir_credit".
Seguidamente utilice el comando ***ALTER TABLE*** en la tabla *transaction* para añadir un ***CONSTRAINT*** que va a ser que el campo *credit_card_id* funcione como ***FOREIGN KEY*** y se relacione con la tabla *credit_card*.



Finalmente la estructura de la base de datos queda así. La tabla transaction funciona como tabla de hechos y queda relacionada por un lado con la tabla *credit_card* con una relación de N a 1. Y por otro lado se relaciona con la tabla *company* también con una relación de N a 1.

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 debe mostrarse para este registro es: TR323456312213576817699999. Recuerda mostrar que el cambio se realizó.



Para poder realizar el cambio de este dato del campo específico que nos piden utilizo el comando ***UPDATE*** en la tabla *credit_card*, para indicar cuál el dato nuevo que reemplace al anterior escribo ***SET*** con el campo iban. Finalmente se necesita filtrar con un ***WHERE*** para poder saber específicamente la fila que quiero modificar, si no me modifica todas las filas de la tabla con el dato que está en el ***SET***.

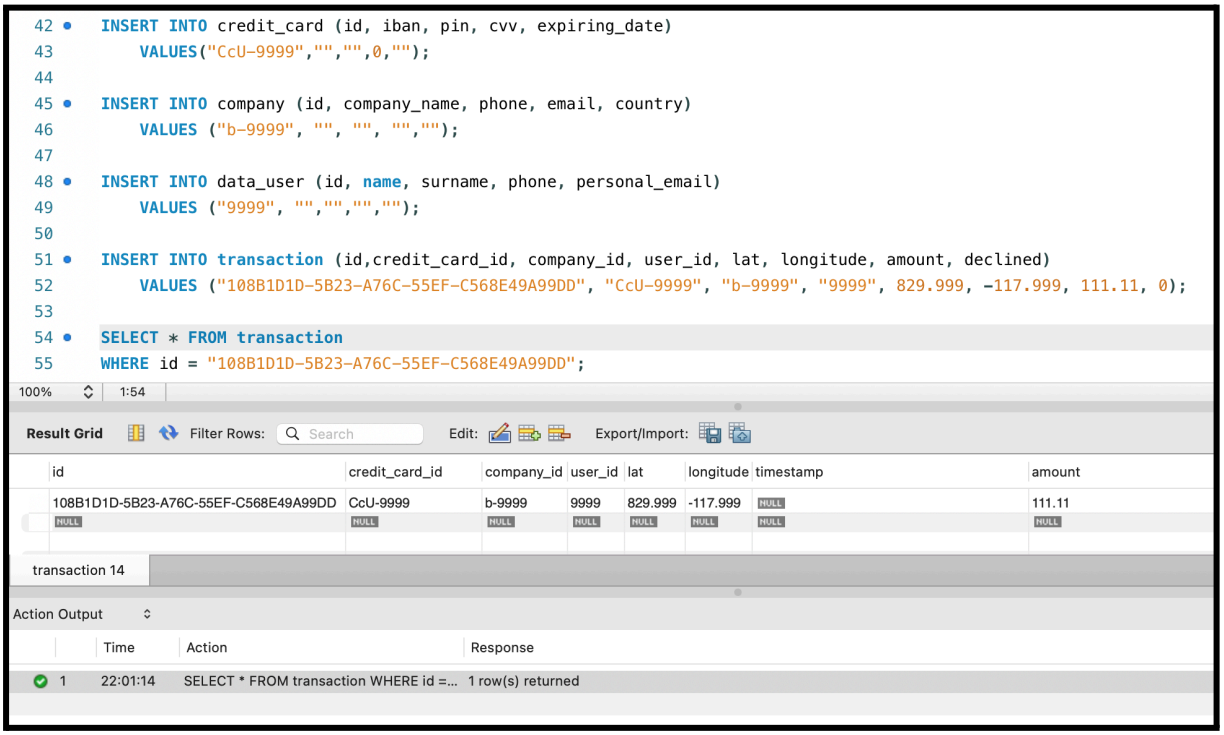
Hago un SELECT de *id* e *iban* filtrando por el *id* para realizar la verificación.

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



Para poder insertar el nuevo registro en la tabla *transaction* primero es necesario garantizar que las ***FOREIGN KEY*** existan en sus respectivas tablas. Por eso:

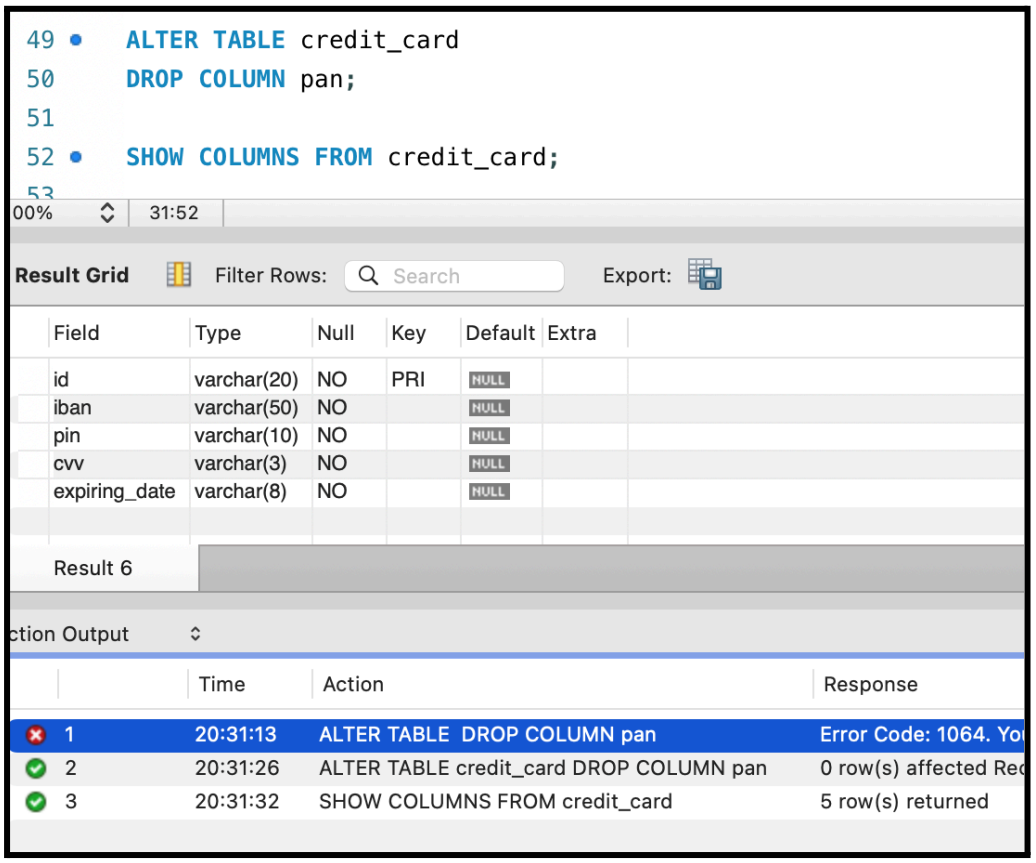
1. Inserto un nuevo registro en la tabla *credit_card* con el `id = "CcU-9999"`.
2. Inserto un nuevo registro en la tabla *company* con el `id = "b-9999"`.
3. Inserto un nuevo registro en la tabla *data_user* con el `id = "9999"`.

Así me aseguro de que los identificadores de tarjeta, compañía y usuario estén creados y puedan relacionarse en la tabla *transaction*.

Finalmente, hago el ***INSERT*** en la tabla *transaction* con toda la información pedida en el enunciado. Para verificar que el nuevo registro se ingresó correctamente, utilizo un ***SELECT*** filtrado por el `id` específico.

Ejercicio 4

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



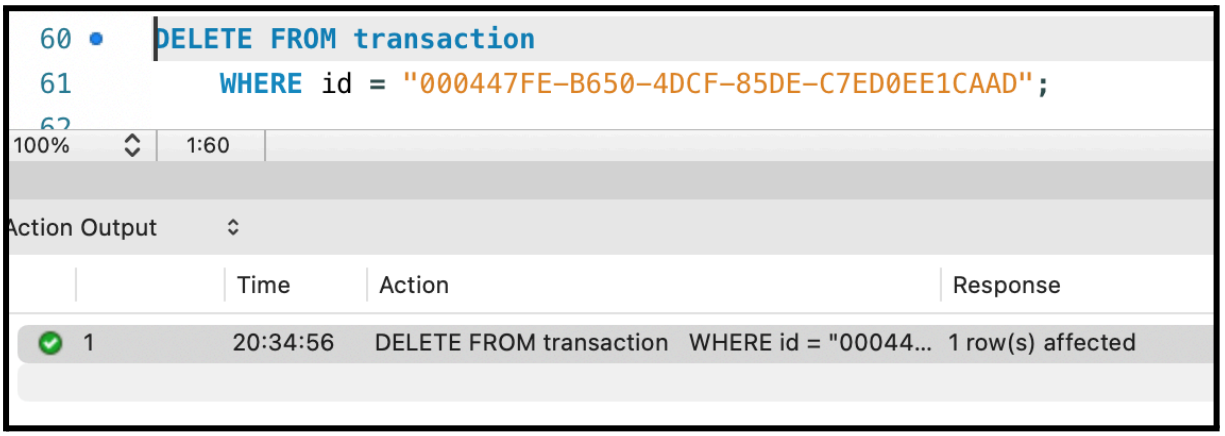
Para poder eliminar un campo de una tabla utilizo el comando ***ALTER TABLE***, que me permite modificar la estructura de la tabla. En este caso aplico la cláusula ***DROP COLUMN*** seguida del nombre de la columna que quiero quitar (*pan*).

Una vez eliminada, hago un **SHOW COLUMNS FROM *credit_card*** para comprobar que la columna ya no forma parte de la tabla. De esta forma se puede evidenciar que el cambio se realizó correctamente.

Nivel 2

Ejercicio 1

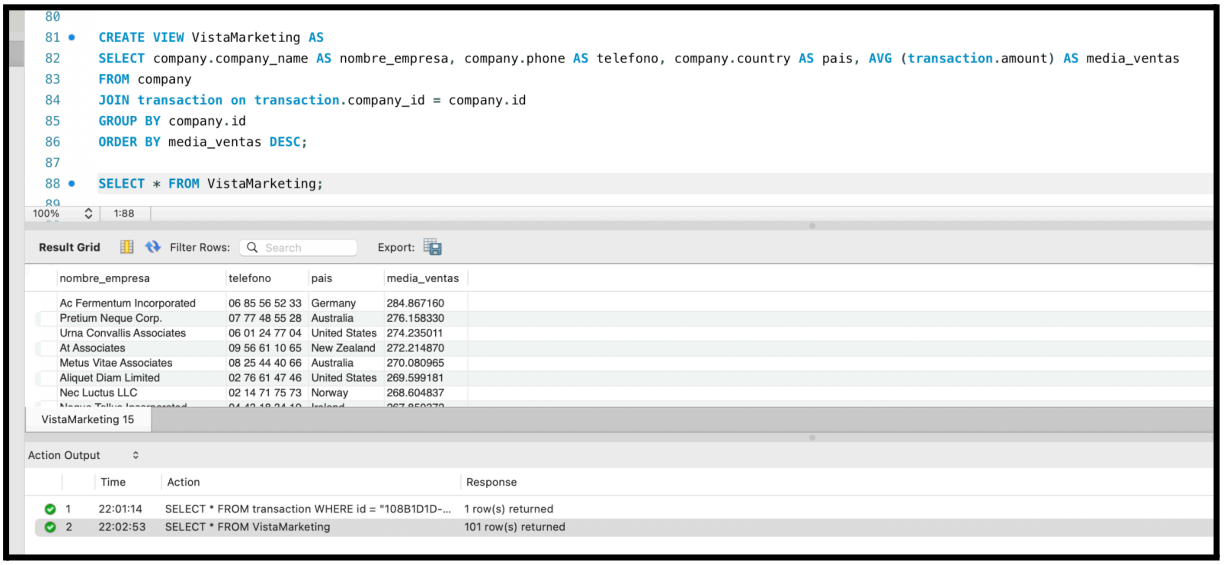
Elimina de la tabla transaction el registro con ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de datos.



Para borrar un registro específico dentro de una tabla utilizo el comando ***DELETE FROM***, indicando primero la tabla de origen *transaction*.
Después, agrego la condición ***WHERE*** con el campo *id* y el valor exacto que quiero eliminar. Esto es muy importante porque, si no se usa el ***WHERE***, se eliminarían todos los registros de la tabla.

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 necesario que crees una vista llamada VistaMarketing que contenga la siguiente información: nombre de la compañía, teléfono de contacto, país de residencia, media de compra realizada por cada compañía. Presenta la vista creada, ordenando los datos de mayor a menor media de compra.



```
80
81 •  CREATE VIEW VistaMarketing AS
82    SELECT company.company_name AS nombre_empresa, company.phone AS telefono, company.country AS pais, AVG (transaction.amount) AS media_ventas
83    FROM company
84    JOIN transaction on transaction.company_id = company.id
85    GROUP BY company.id
86    ORDER BY media_ventas DESC;
87
88 •  SELECT * FROM VistaMarketing;
89
```

100% 1:88

Result Grid | Filter Rows: Search | Export:

nombre_empresa	telefono	pais	media_ventas
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.867160
Pretium Neque Corp.	07 77 48 55 28	Australia	276.158330
Urna Convallis Associates	06 01 24 77 04	United States	274.235011
At Associates	09 56 61 10 65	New Zealand	272.214870
Metus Vitae Associates	08 25 44 40 66	Australia	270.080965
Aliquet Diam Limited	02 76 61 47 46	United States	269.599181
Nec Luctus LLC	02 14 71 75 73	Norway	268.604837

VistaMarketing 15

Action Output

	Time	Action	Response
1	22:01:14	SELECT * FROM transaction WHERE id = "108B1D1D-...	1 row(s) returned
2	22:02:53	SELECT * FROM VistaMarketing	101 row(s) returned

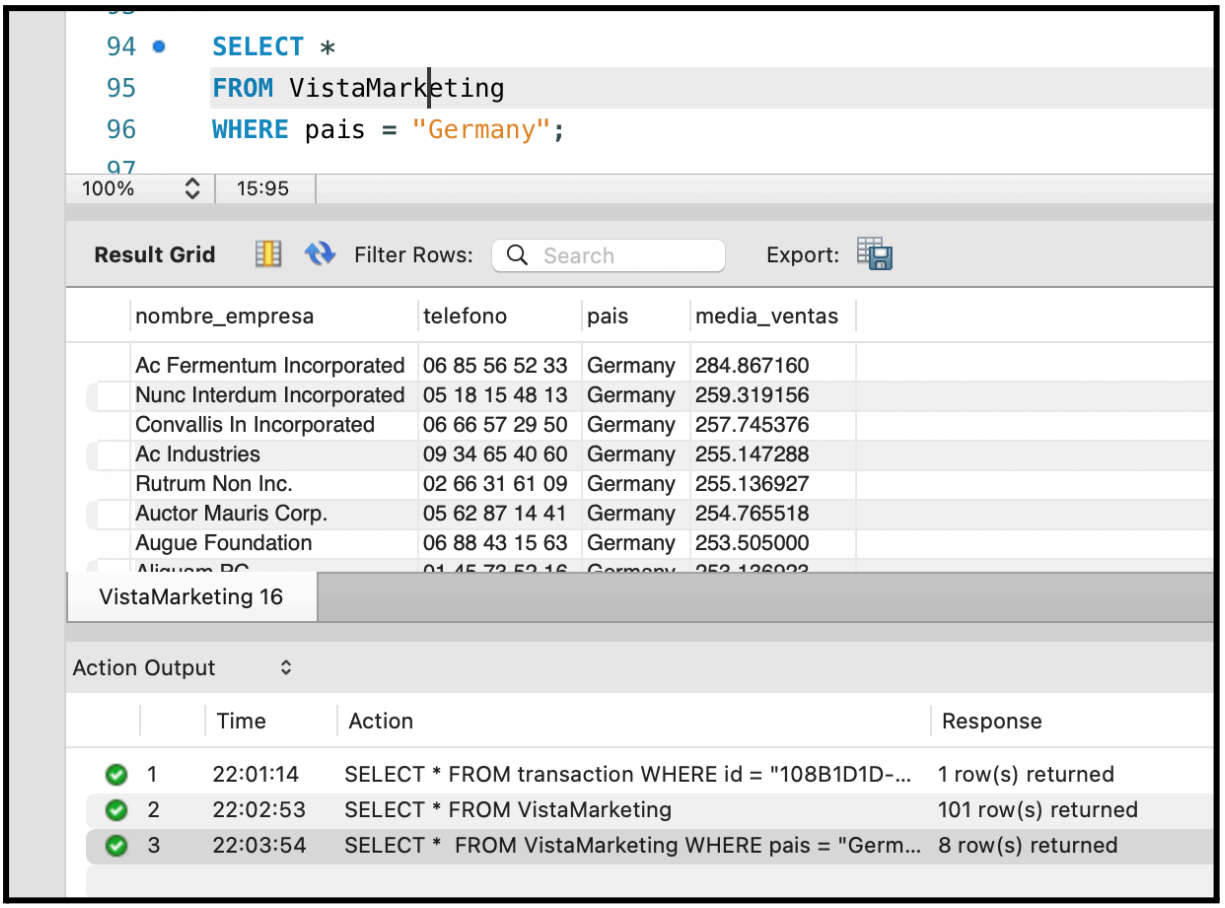
Para resolver este ejercicio cree una vista con el comando ***CREATE VIEW***, lo que permite guardar una consulta con nombre y reutilizarla fácilmente. Dentro del ***SELECT*** incluyo las columnas solicitadas con sus respectivos alias.

Luego utilizo un ***JOIN*** entre la tabla *company* y la tabla ***transaction*** para relacionar cada transacción con su respectiva compañía.

Finalmente, aplico ***GROUP BY*** *company.id* para que la media de ventas se calcule por cada empresa, y ***ORDER BY media_ventas DESC*** para ordenar los resultados desde la media más alta a la más baja.

Ejercicio 3

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

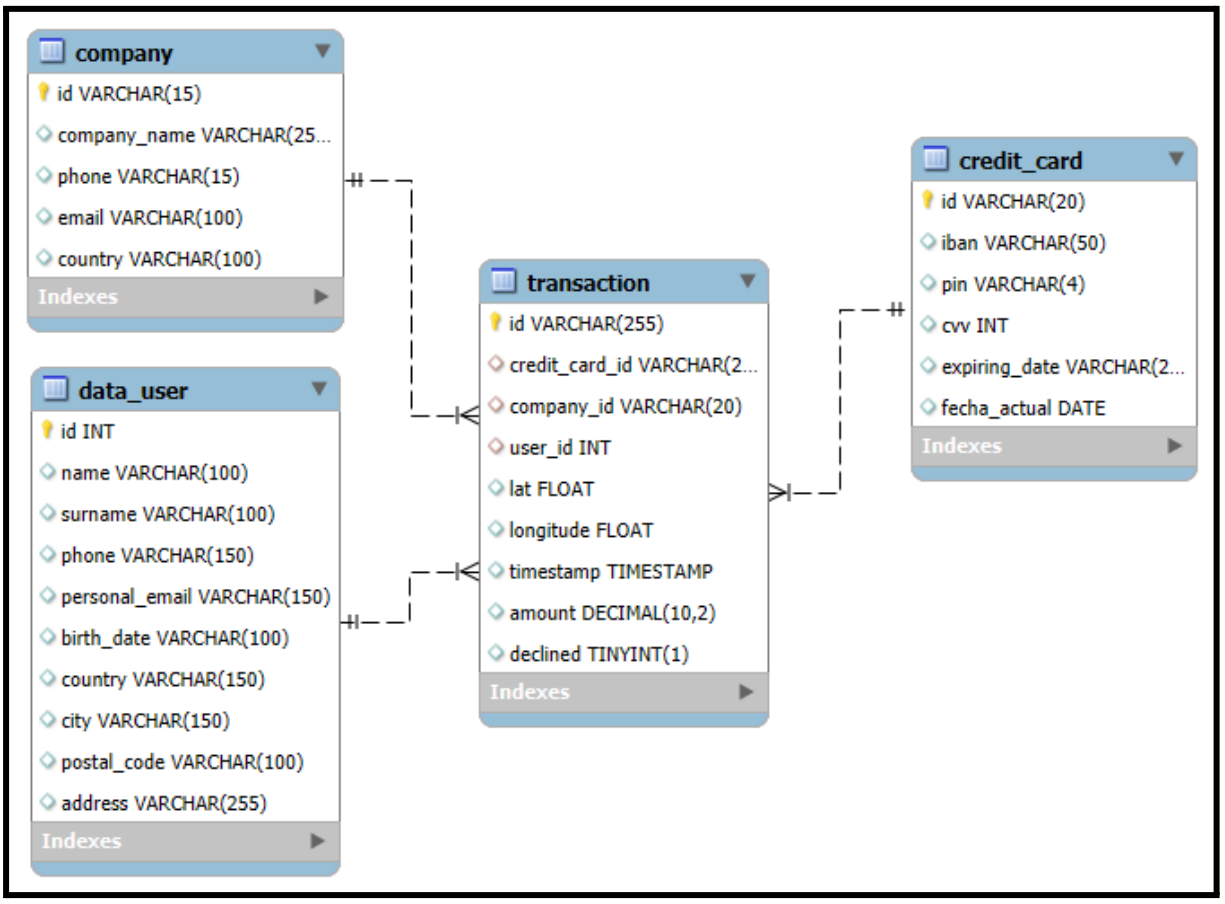


Para restringir los resultados, utilizo el filtro ***WHERE***, indicando que solo se muestren aquellas filas en las que el campo *country* tenga el valor `"Germany"`.

Nivel 3

Ejercicio 1

La semana que viene 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 hizo. Te pide que lo ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama:



```sql
RENAME TABLE user to data_user;

ALTER TABLE data_user
MODIFY id INT,
CHANGE email personal_email VARCHAR(150) NOT NULL;

ALTER TABLE company
DROP COLUMN website;
```

Primero paso a renombrar la tabla ***user*** a ***data_user*** con ***RENAME TABLE***, adaptando el nombre al del diagrama.
En ***data_user***, cambio el tipo de dato de la columna ***id*** a ***INT*** y renombro la columna ***email*** a ***personal_email*** con la cláusula ***CHANGE***, además de ajustar la longitud a `VARCHAR(150)` y marcarla como ***NOT NULL***.
En la tabla ***company***, elimino la columna ***website*** usando ***DROP COLUMN***.

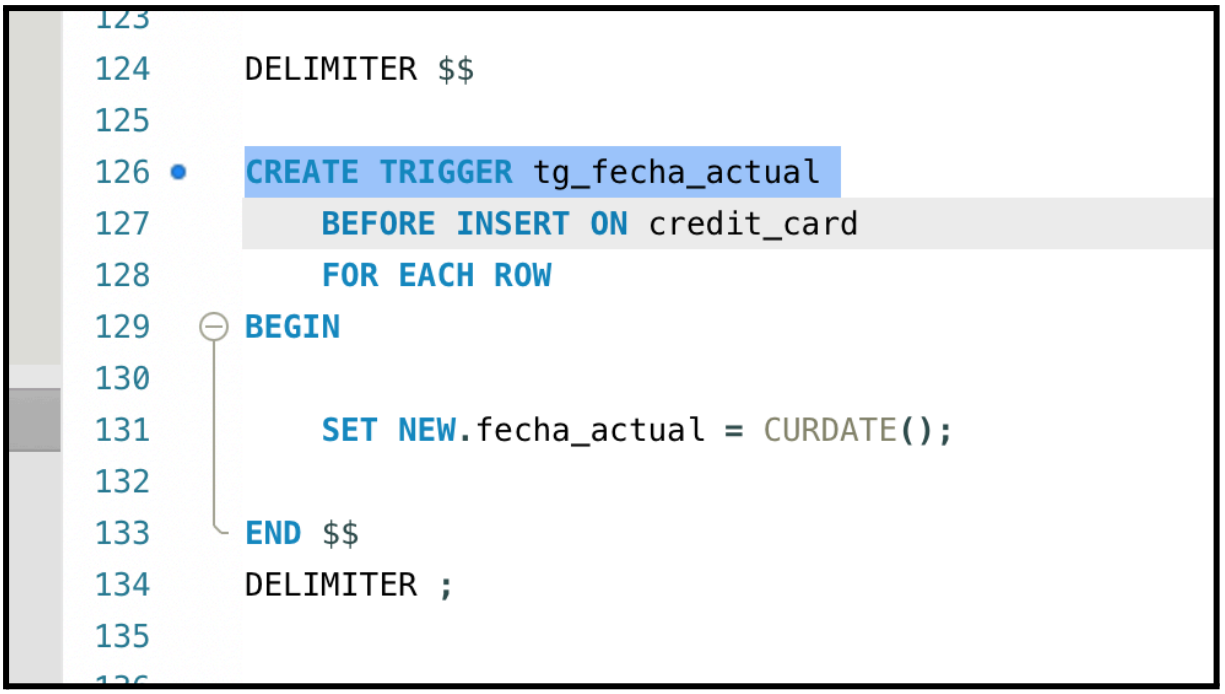
```sql
ALTER TABLE transaction
MODIFY declined TINYINT(1),
MODIFY user_id INT,
ADD CONSTRAINT fk_transaction_user
    FOREIGN KEY (user_id) REFERENCES data_user(id);

ALTER TABLE credit_card
MODIFY pin VARCHAR(4),
MODIFY cvv INT,
MODIFY expiring_date VARCHAR(20),
ADD fecha_actual DATE;
```

En la tabla ***transaction***, con el comando ***MODIFY*** ajusto los tipos de datos:

- ***declined*** a ***TINYINT(1)***
- ***user_id*** a ***INT***
- Además, añado una restricción de **FOREIGN KEY** para relacionar ***user_id*** con el campo ***id*** de ***data_user***.

En la tabla **credit_card** realizo las siguientes modificaciones:

- ajusto ***pin*** a `VARCHAR(4)`
- ***cvv*** lo cambio a **INT**
- cambio la longitud de ***expiring_date*** a `VARCHAR(20)`
- Añado un nuevo campo ***fecha_actual*** de tipo ***DATE***.



```
123
124     DELIMITER $$
125
126     CREATE TRIGGER tg_fecha_actual
127         BEFORE INSERT ON credit_card
128         FOR EACH ROW
129     BEGIN
130
131         SET NEW.fecha_actual = CURDATE();
132
133     END $$
134     DELIMITER ;
135
```

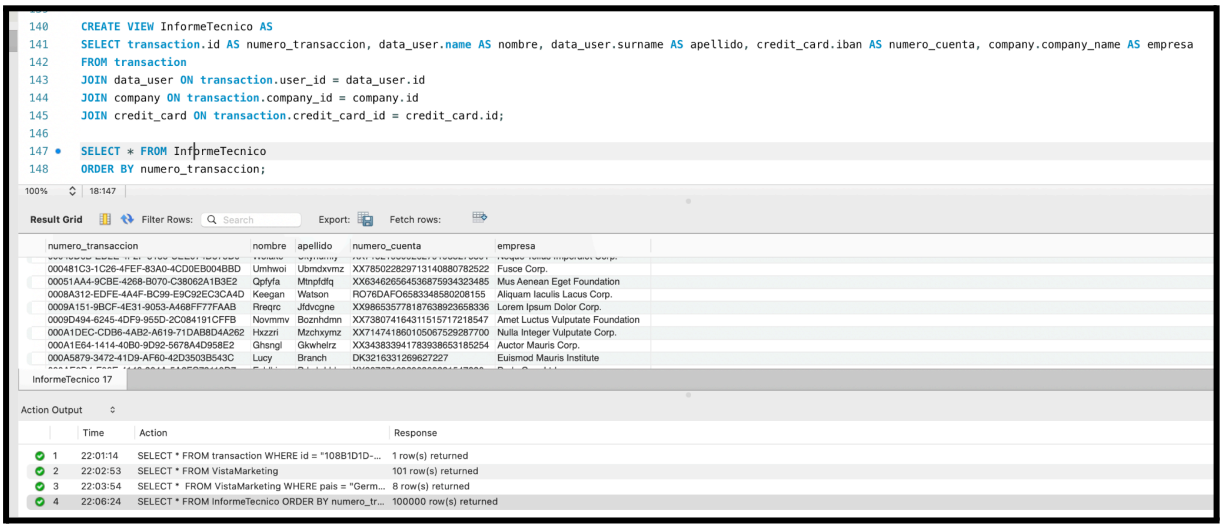
Finalmente, creo un ***trigger*** llamado ***tg_fecha_actual***. Este trigger se ejecuta antes de cada ***INSERT*** en la tabla ***credit_card*** y establece automáticamente la fecha del sistema con la función **CURDATE()** en la columna ***fecha_actual***.

Ejercicio 2

La empresa también les pide 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

Asegúrense de incluir información relevante de las tablas que conozcan y utilicen alias para cambiar el nombre de columnas según sea necesario.
Muestra los resultados de la vista, ordena los resultados de forma descendente en función de la variable ID de transacción.



```
140  CREATE VIEW InformeTecnico AS
141  SELECT transaction.id AS numero_transaccion, data_user.name AS nombre, data_user.surname AS apellido, credit_card.iban AS numero_cuenta, company.company_name AS empresa
142  FROM transaction
143  JOIN data_user ON transaction.user_id = data_user.id
144  JOIN company ON transaction.company_id = company.id
145  JOIN credit_card ON transaction.credit_card_id = credit_card.id;
146
147  SELECT * FROM InformeTecnico
148  ORDER BY numero_transaccion;
```

100% 18:147

Result Grid — Filter Rows: Search — Export: — Fetch rows:

numero_transaccion	nombre	apellido	numero_cuenta	empresa
000481C3-1C26-4FEF-83A0-4CD0EB004B8D	Umhwoi	Ubmdxvmz	XX7850228297131408807825 22	Fusce Corp.
00051AA4-9CBE-4268-B070-C38062A1B3E2	Qpfyfa	Mtnpfdfq	XX6346265645368759343234 85	Mus Aenean Eget Foundation
0008A312-EDFE-4A4F-BC99-E9C92EC3CA4D	Keegan	Watson	RO76DAFO6583348580208155	Aliquam Iaculis Lacus Corp.
0009A151-9BCF-4E31-9053-A468FF77FAAB	Rreqrc	Jfdvcgne	XX9865357781876389236583 36	Lorem Ipsum Dolor Corp.
0009D494-6245-4DF9-955D-2C084191CFFB	Novmmv	Boznhdmn	XX7380741643115157172185 47	Amet Luctus Vulputate Foundation
000A1DEC-CDB6-4AB2-A619-71DAB8D4A262	Hxzzri	Mzchxymz	XX7147418601050675292877 00	Nulla Integer Vulputate Corp.
000A1E64-1414-40B0-9D92-567BA4D958E2	Ghsngl	Gkwhelrz	XX3438339417839386531852 54	Auctor Mauris Corp.
000A5879-3472-41D9-AF60-42D3503B543C	Lucy	Branch	DK3216331269627227	Euismod Mauris Institute

InformeTecnico 17

Action Output

	Time	Action	Response
1	22:01:14	SELECT * FROM transaction WHERE id = "108B1D1D-...	1 row(s) returned
2	22:02:53	SELECT * FROM VistaMarketing	101 row(s) returned
3	22:03:54	SELECT * FROM VistaMarketing WHERE pais = "Germ...	8 row(s) returned
4	22:06:24	SELECT * FROM InformeTecnico ORDER BY numero_tr...	100000 row(s) returned