

Sprint.3 NIVEL 1

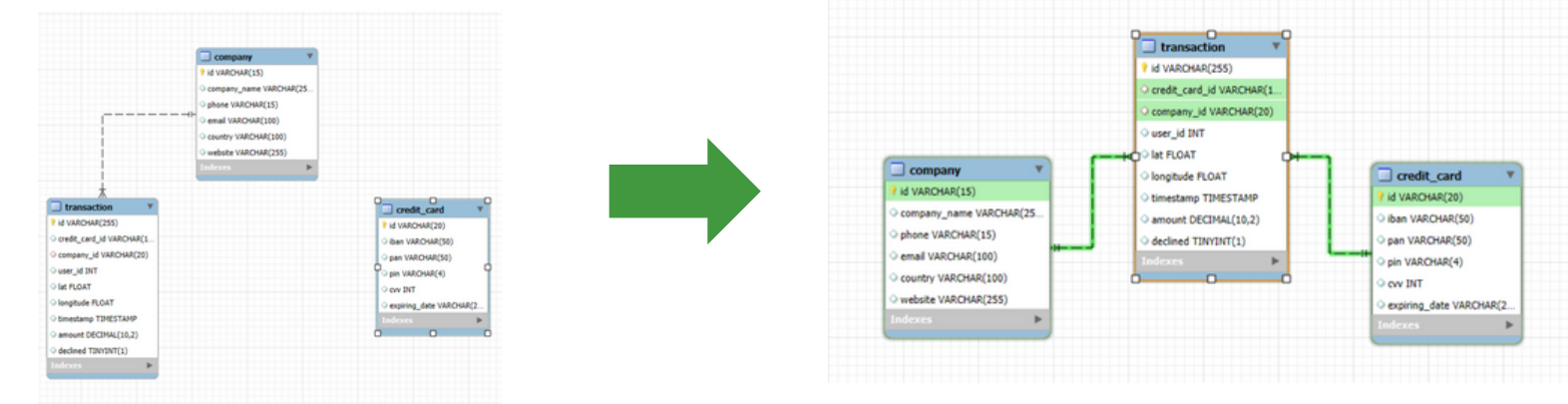
Carla Cociña

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_introducir_credit". Recuerda mostrar el diagrama y realizar una breve descripción de este.

Lo primero que hice, fue mirar el esquema para ver que tipo de datos necesitaba en cada columna de la tabla credit_card.

Al hacer el reverse veo que la tabla transaction ya tiene su FK ahora falta conectarla, por lo que genero un alter table y agrego la FK. Ahora las tres tablas están conectadas desde transaction.



```
2 -- NIVEL 1
3
4 -- Ejercicio 1
5
6 • USE transactions;
7
8 • CREATE TABLE credit_card
9 (id varchar(20) primary key not null,
10 iban varchar (50),
11 pan varchar (50),
12 pin varchar(4),
13 cvv int,
14 expiring_date varchar(20));
15
16 /* Al hacer el reverse despues de crear la tabla y cargar
17 los datos se puede ver que no esta conectada PK/FK con transaccions */
18
19 • ALTER TABLE transaction add foreign key (credit_card_id) references credit_card(id);
20
21 /* Como la foreign key en transaccions ya estaba hecha
22 hice un alter table para generar la referencia y ligarlas */
23
```

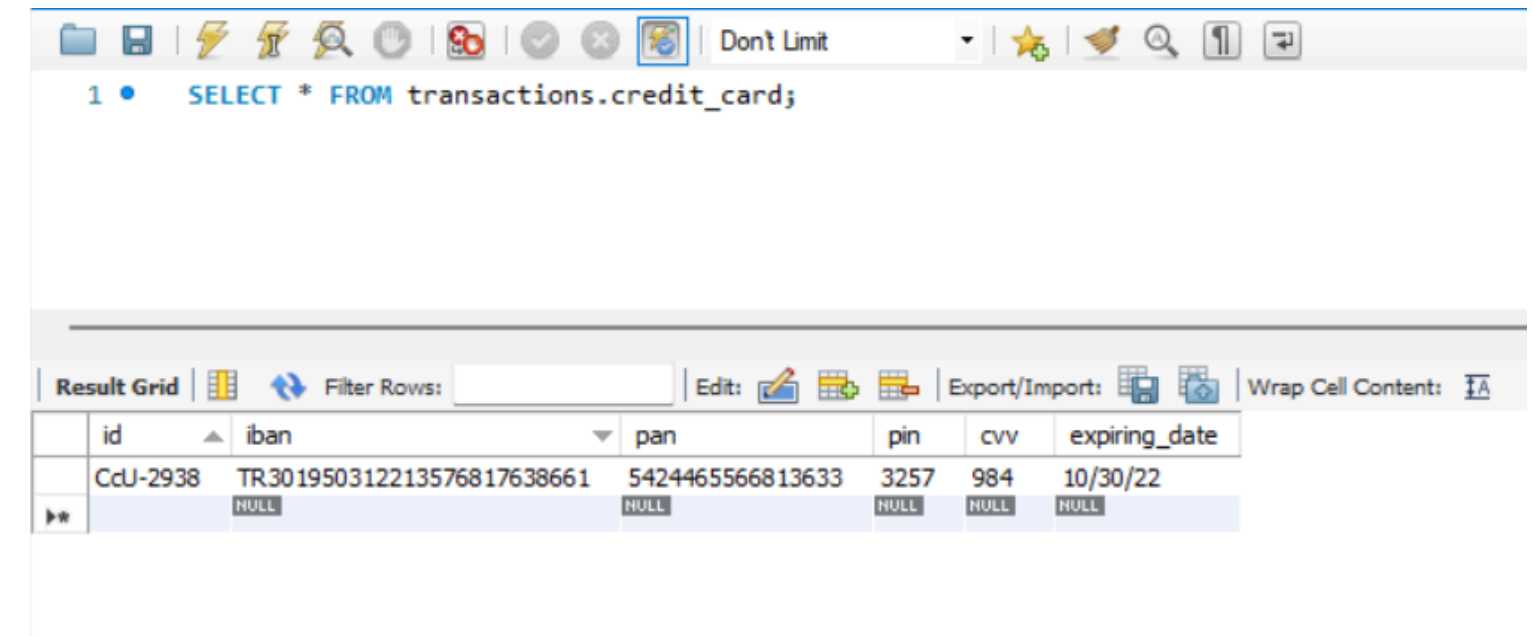
```
2 -- Insertamos datos de credit_card
3 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
4 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
5 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
6 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
7 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
8 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
9 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
10 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
11 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
```

```
'CcU-2938', 'TR301950312213576817638661', '5424465566813633', '3257', '984', '10/30/22');
'CcU-2945', 'D026854763748537475216568689', '5142423821948828', '9080', '887', '08/24/23');
'CcU-2952', 'BG45IVQL52710525608255', '4556 453 55 5287', '4598', '438', '06/29/21');
'CcU-2959', 'CR7242477244335841535', '372461377349375', '3583', '667', '02/24/23');
'CcU-2966', 'BG72LKTQ15627628377363', '448566 886747 7265', '4900', '130', '10/29/24');
'CcU-2973', 'PT87806228135092429456346', '544 58654 54343 384', '8760', '887', '01/30/25');
'CcU-2980', 'DE39241881883086277136', '402400 7145845969', '5075', '596', '07/24/22');
'CcU-2987', 'GE89681434837748781813', '3763 747687 76666', '2298', '797', '10/31/23');
'CcU-2994', 'BH62714428368066765294', '344283273252593', '7545', '595', '02/28/22');
```

Ejercicio 2

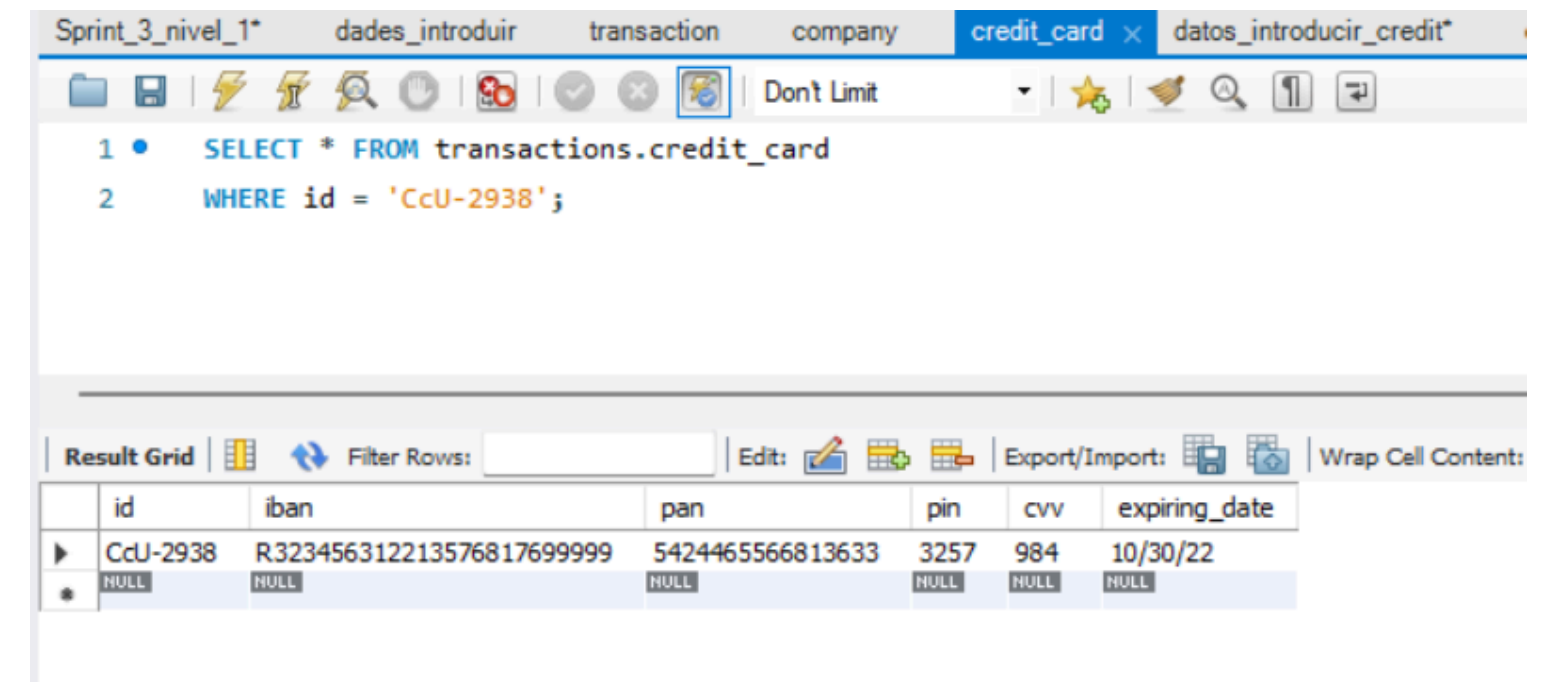
Me piden cambiar un dato en la tabla credit_card que tiene un error, lo hago con UPDATE

```
26
27 • UPDATE credit_card
28 SET iban = 'R323456312213576817699999'
29 WHERE id = 'CcU-2938';
```



The screenshot shows a SQL query editor with a toolbar at the top. The query is: `1 • SELECT * FROM transactions.credit_card;`. Below the query, the 'Result Grid' is displayed with columns: id, iban, pan, pin, cvv, and expiring_date. The first row shows data for 'CcU-2938'. The second row shows NULL values for all columns.

	id	iban	pan	pin	cvv	expiring_date
	CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22
▶		NULL	NULL	NULL	NULL	NULL



The screenshot shows a SQL query editor with a toolbar at the top. The query is: `1 • SELECT * FROM transactions.credit_card` and `2 WHERE id = 'CcU-2938';`. Below the query, the 'Result Grid' is displayed with columns: id, iban, pan, pin, cvv, and expiring_date. The first row shows data for 'CcU-2938'. The second row shows NULL values for all columns.

	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	R323456312213576817699999	5424465566813633	3257	984	10/30/22
*	NULL	NULL	NULL	NULL	NULL	NULL

Ejercicio 3

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

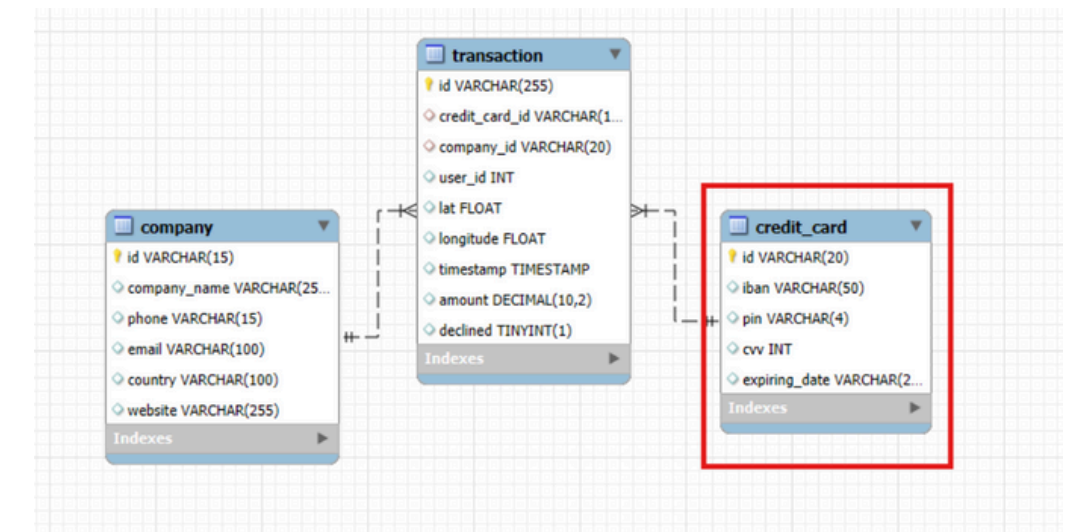
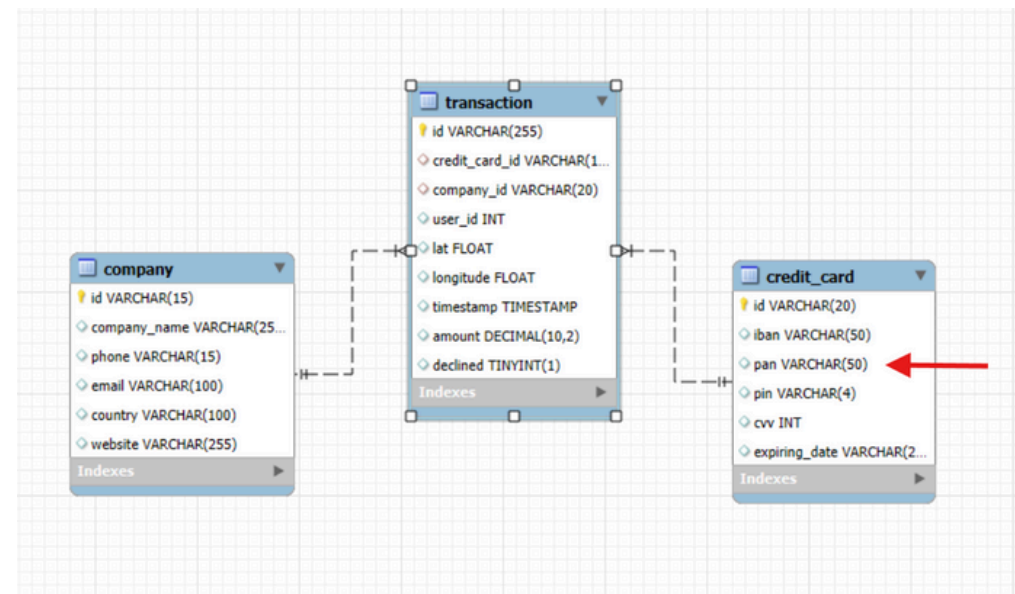
Como no proporcionan el timestamp, lo quito del código para que cada dato vaya en el orden correcto. Si tuviera todos los datos, no sería necesario detallar las columnas. Tengo que desbloquear restricciones para insertarlo ya que genera error, después vuelvo a bloquearlas

```
33 -- para insertar los datos lo hago quitando timestamp para que se apliquen los datos en cada tabla
34 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
35 VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '111.11', '0');
36
37 -- Cuando intento ejecutarlo genera error code 1452
38
39 • SET FOREIGN_KEY_CHECKS=0;
40
41 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
42 VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '111.11', '0');
43
44 -- Vuelvo a bloquear la foreign key
45
46 • SET FOREIGN_KEY_CHECKS=1;
```

Ejercicio 4

Eliminar la columna pan de credit_card

```
50 • ALTER TABLE credit_card
51 DROP COLUMN pan;
```



Sprint.3 NIVEL 2

Ejercicio 1

Eliminar el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la tabla de transacciones.

```
-  
6 • DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';  
7  
8 • SELECT id  
9 FROM transaction  
10 where id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

```
2  
3 • SELECT * FROM transactions.transaction  
4 WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

Result Grid				
	id	credit_card_id	company_id	user_id
▶	02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	92
*	NULL	NULL	NULL	NULL



```
2  
3 • SELECT * FROM transactions.transaction  
4 WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

Result Grid							
	id	credit_card_id	company_id	user_id	lat	longitude	timestamp
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Ejercicio 2

Se ha solicitado crear una vista que proporcione detalles clave sobre las compañías y sus transacciones.

Crear una vista llamada **VistaMarketing** que contenga la siguiente información:

- Nombre de la compañía
- Teléfono de contacto
- País de residencia
- Promedio de compra realizada por cada compañía ordenada de forma descendiente

```
24 • CREATE VIEW VistaMarketing AS
25   SELECT c.company_name, c.phone, c.country, round(avg(t.amount),2) AS Promedio_Compra
26   FROM company c JOIN transaction t ON c.id = t.company_id
27   WHERE t.declined = 0
28   GROUP BY c.company_name, c.phone, c.country
29   ORDER BY Promedio_Compra DESC;
```

Ejercicio 3

Filtrar desde la VistaMarketing por Alemania

```
33 • SELECT *
34   FROM VistaMarketing
35   WHERE country = 'Germany';
```

Resultado VistaMarketing

3

4 • SELECT * FROM transactions.vistamarketing;

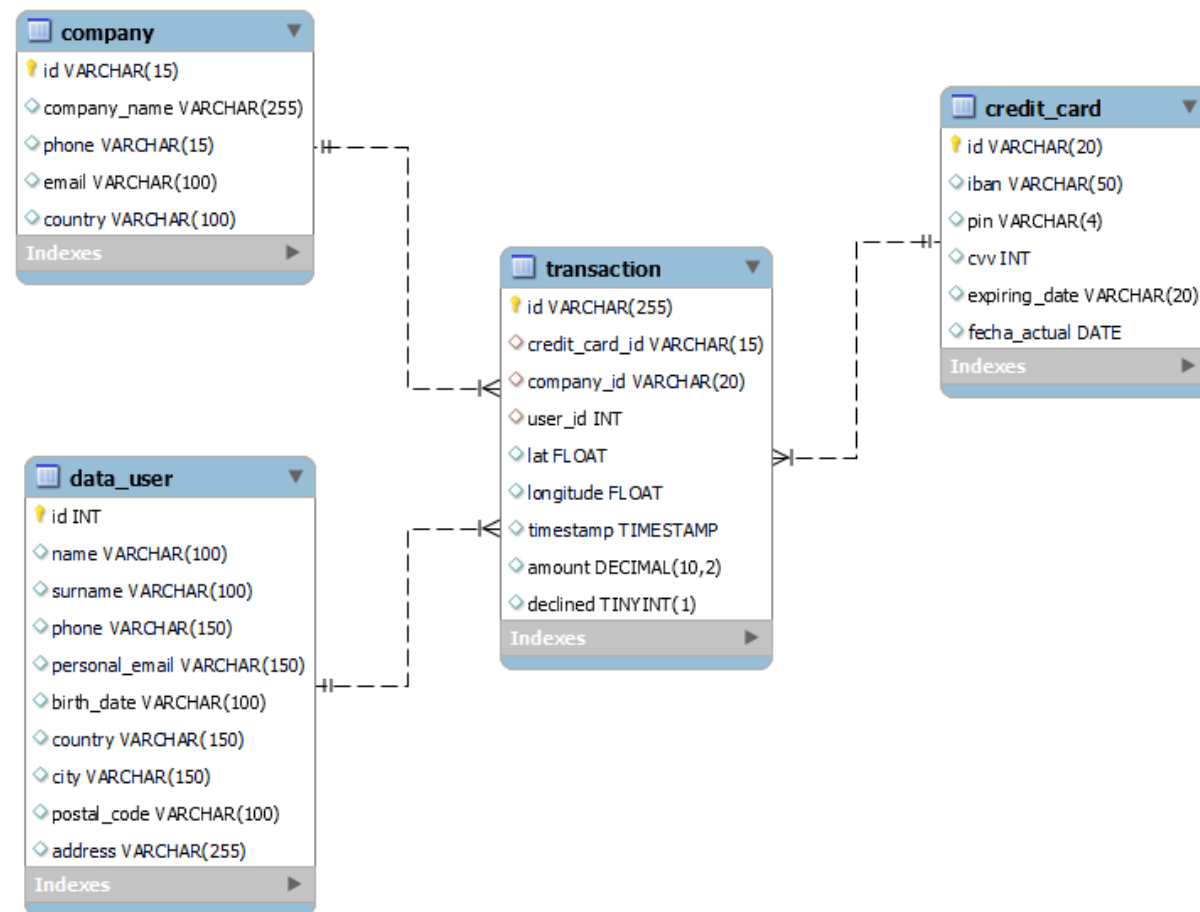
	company_name	phone	country	Promedio_Compra
▶	Eget Ipsum Ltd	03 67 44 56 72	United States	481.86
	Sed Id Limited	07 28 18 18 13	United States	477.51
	Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.10
	Nunc Sit Incorporated	07 28 42 63 63	Norway	461.83
	Non Magna LLC	06 71 73 13 17	United Kingdom	458.74
	Maecenas Malesuada Fringilla Inc.	09 38 53 76 61	Netherlands	451.29
	Erat LLP	03 18 88 77 79	Netherlands	448.44
	Tortor Nunc Commodo Company	05 35 92 77 16	United States	447.11
	Justo Eu Arcu Ltd	08 42 56 71 52	Italy	444.16
	Pede Cum Ltd	07 62 26 48 38	Norway	442.32
	Vestibulum Lorem PC	02 02 87 33 40	Belgium	428.40
	Mauris Institute	05 29 60 36 87	Sweden	427.71
	Aliquet Diam Limited	02 76 61 47 46	United States	425.64
	Mus Aenean Eget Foundation	06 25 15 52 43	Sweden	419.97
	Sed LLC	01 63 16 26 52	Belgium	416.66
	Viverra Donec Foundation	03 33 12 32 73	United Kingdom	414.53
	Eget Tincidunt Dui Institute	05 35 93 32 44	Netherlands	413.50
	Amet Institute	06 33 40 21 33	Australia	412.48

Sprint.3 NIVEL 3

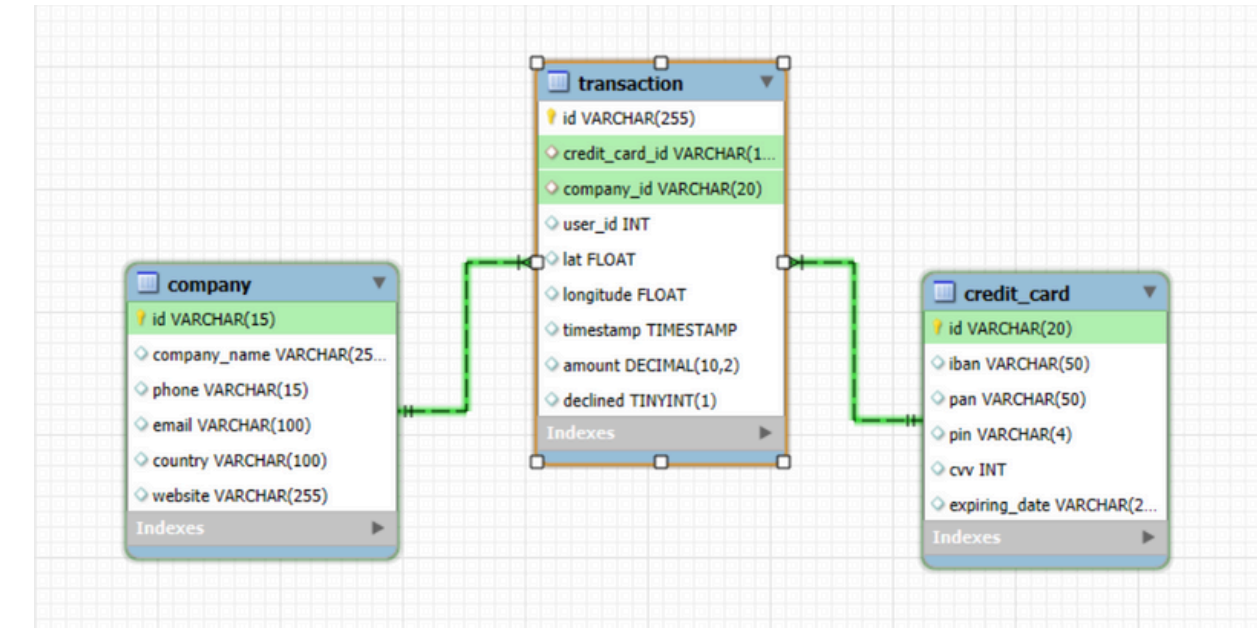
Ejercicio 1

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 cómo las hizo. Te pide que lo ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama:

El esquema que piden



Esquema actual



Con un reverse reviso los dos diagramas, hago una lista de lo que le falta y que necesita modificación:

- 1- Hay una nueva tabla llamada Data_User
- 2- En company hay que eliminar la columna website
- 3- Credit card tiene una nueva columna llamada 'fecha_actual' DATE
- 4- En la nueva tabla Data_user hay que cambiar el nombre de la columna email

1- Con el script **estructura_datos_user** hay que crear la nueva tabla que se llama user, no la puedo cambiar a data_user como sale en el diagrama hasta que cargue todos los datos.

Al ejecutar el script, la foreign key se posiciona mal. Para comprobar el esquema hago el reverse y en vez de hacer 1:N hace N:1, lo borro del CREATE TABLE y agrego después de crear la nueva tabla con ALTER TABLE para que genere el enlace 1:N con transaction usando como referencia user(id)

Agregue la data del script **datos_introducir_user** y se cargaron 275 rows

En el diagrama el compañero hizo un cambio de nombre en la tabla user y lo cambio a data_user

- `RENAME TABLE user TO data_user;`

Este cambio generó un error en el diagrama en donde aparecía la foreign key dos veces
con SHOW CREATE TABLE pude ver donde estaba el error y transaction_ibfk estaba dos veces por lo que hice drop al constraint repetido

- `alter table transaction drop CONSTRAINT transaction_ibfk_4;`

- `CREATE INDEX idx_user_id ON transaction(user_id);`

- `CREATE TABLE IF NOT EXISTS user (
 id INT PRIMARY KEY,
 name VARCHAR(100),
 surname VARCHAR(100),
 phone VARCHAR(150),
 email VARCHAR(150),
 birth_date VARCHAR(100),
 country VARCHAR(150),
 city VARCHAR(150),
 postal_code VARCHAR(100),
 address VARCHAR(255));`

`FOREIGN KEY(id) REFERENCES transaction(user_id)`

`ALTER TABLE transaction add foreign key (user_id) references user(id);`

2- En company hay que eliminar la columna website

```
ALTER TABLE company
DROP COLUMN website;
```

Compruebo que la he eliminado correctamente

	id	company_name	phone	email	country
▶	b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net	Germany
	b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia

3- Credit card tiene una nueva columna llamada 'fecha_actual' DATE

```
ALTER TABLE credit_card
ADD COLUMN fecha_actual DATE DEFAULT(CURRENT_DATE);
```

6 •

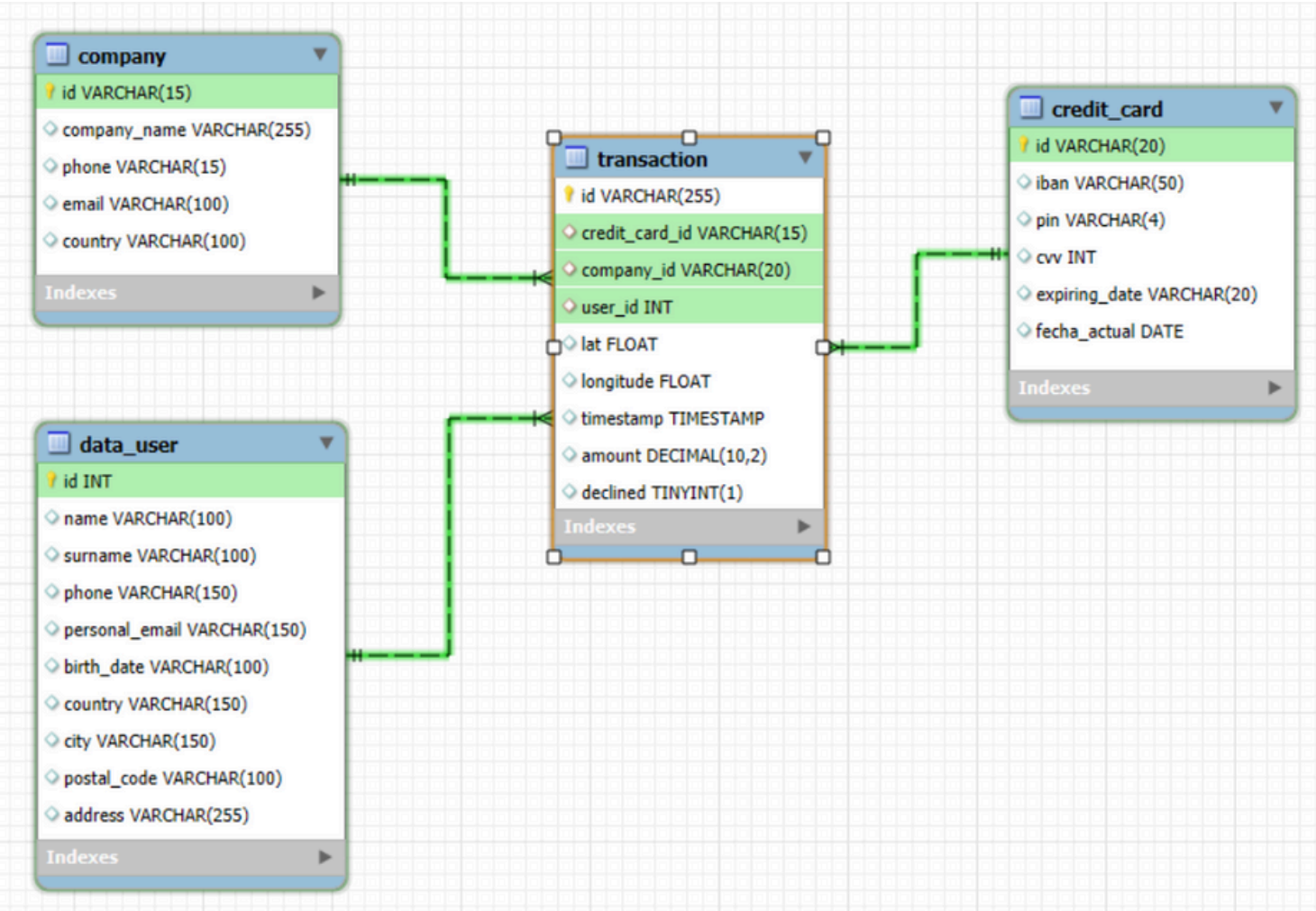
```
SELECT * FROM transactions.credit_card;
```

	id	iban	pin	cvv	expiring_date	fecha_actual
▶	CcU-2938	R323456312213576817699999	3257	984	10/30/22	2025-01-26
	CcU-2945	DO26854763748537475216568689	9080	887	08/24/23	2025-01-26

4- Cambio el nombre de la columna email de data_user a personal_email

- ```
ALTER TABLE data_user
RENAME COLUMN email TO personal_email;
```

Finalmente así queda el esquema con las modificaciones





## Ejercicio 2

crear una vista llamada "InformeTecnico" con ID de la transacción, nombre, apellido, iban de la tarjeta , nombre compañía ordenado por ID de transaccion de forma descendiente

```
CREATE VIEW InformeTecnico AS
SELECT t.id, d.name, d.surname, cr.iban, c.company_name
FROM company c JOIN transaction t ON (c.id = t.company_id)
JOIN data_user d ON (d.id = t.user_id)
JOIN credit_card cr ON (cr.id = t.credit_card_id)
ORDER BY t.id DESC;
```

## Resultado vista InformeTecnico

5 • `SELECT * FROM transactions.informetecnico;`

|   | id                                   | name    | surname | iban                            | company_name                 |
|---|--------------------------------------|---------|---------|---------------------------------|------------------------------|
| ▶ | FE96CE47-BD59-381C-4E18-E3CA3D44E8FF | Kenyon  | Hartman | DO26854763748537475216568689    | Magna A Neque Industries     |
|   | FE809ED4-2DB6-55AC-C915-929516E4646B | Molly   | Gilliam | SE2813123487163628531121        | Nunc Interdum Incorporated   |
|   | FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65 | Linus   | Willis  | KW9485332754781757886242955643  | Nunc Interdum Incorporated   |
|   | FD89D51B-AE8D-77DC-E450-B8083FBD3187 | Hilda   | Levy    | LT053237077744561475            | Malesuada PC                 |
|   | FD2E8957-414B-BEEC-E9AD-59AA7A8A6290 | Hedwig  | Gilbert | GE84848451582810541526          | Neque Tellus Imperdiet Corp. |
|   | FCE2A89A-271D-2BDC-9E49-8DD92A373391 | Hakeem  | Alford  | MD1234119525145401270486        | Nunc Interdum Incorporated   |
|   | FBD7E0D6-BA6B-F5BC-0CA9-EA4B8760100C | Hedwig  | Gilbert | MU4132333444534342541344788855  | Mauris Id Inc.               |
|   | FAC76A80-8448-69AA-E892-426C2F12621C | Slade   | Poole   | MT05JWCF58868200575771634583813 | Arcu LLP                     |
|   | FAAD3FFC-1A17-E141-43D3-359A5BA7CB3B | Hedwig  | Gilbert | GE90157928843338134463          | Lorem Eu Incorporated        |
|   | FA053936-75D8-85FA-490D-9B624E1B920A | Hedwig  | Gilbert | GT02497653655330848247645975    | Non Justo Corp.              |
|   | F85A7D75-2778-9D75-D776-3F41A828DE88 | Sarah   | Beck    | VG1468087984174645729577        | Ut Semper Foundation         |
|   | F843DC08-CCB5-2444-1B4E-5966289FBA8B | Jasper  | Landry  | VG1468087984174645729577        | Ut Semper Foundation         |
|   | F5ACD74B-4275-5AA1-2414-6EF417636B98 | Nora    | Reeves  | MD1234119525145401270486        | Nunc Interdum Incorporated   |
|   | F56FCA4A-0039-9F64-7376-85632B91121B | Lynn    | Riddle  | CR7242477244335841535           | Ut Semper Foundation         |
|   | F55B3CE1-3379-E0BF-5AB9-6F4CC2C5479C | Sonya   | Mckee   | EE541536644818872885            | Arcu LLP                     |
|   | F4BCAE41-3B8E-EA8D-9C24-466F7CEB9F9A | Chester | Haynes  | CY94263537405015481188625576    | Malesuada PC                 |
|   | F2B3E645-2E6D-E891-9D05-33DBACE58DE4 | Heather | Burks   | SM6022751049715477062682363     | Malesuada PC                 |
|   | F28E106B-5418-4667-9514-2E2A823ECC65 | Hedwig  | Gilbert | RO76DAFO6583348580208155        | Pede Ultrices Ltd            |
|   | F233E386-5A74-63B3-8111-09FC9BA38011 | Nero    | Mills   | HU95215627749276573565556322    | Arcu LLP                     |
|   | F22B8361-F2CD-8C41-DDCA-14604F175CDB | Alia    | Cherry  | MD5733887426782868247555        | Magna Incubitor              |

InformeTecnico 1 x

Output

Action Output

| #   | Time     | Action                                                                                                     | Message             |
|-----|----------|------------------------------------------------------------------------------------------------------------|---------------------|
| 664 | 19:14:14 | SELECT t.id, d.name, d.surname, cr.iban, c.company_name FROM company c JOIN transaction t ON (c.id = t.... | 587 row(s) returned |
| 665 | 19:14:53 | SELECT t.id, d.name, d.surname, cr.iban, c.company_name FROM company c JOIN transaction t ON (c.id = t.... | 587 row(s) returned |