

# Nivell 1

## Exercici 1

La teva tasca és dissenyar i crear una taula anomenada "credit\_card" que emmagatzemi detalls crucials sobre les targetes de crèdit. La nova taula ha de ser capaç d'identificar de manera única cada targeta i establir una relació adequada amb les altres dues taules ("transaction" i "company"). Després de crear la taula serà necessari que ingressis la informació del document denominat "dades\_introduir\_credit". Recorda mostrar el diagrama i realitzar una breu descripció d'aquest.

Primero creamos la tabla nueva credit\_card y especificamos que la columna id es la Primary Key:

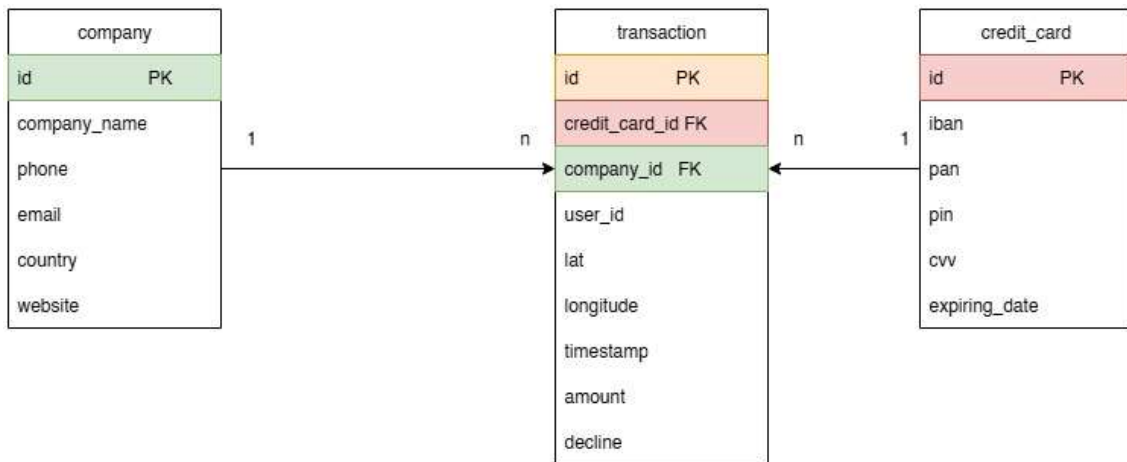
```
8 • ○ CREATE TABLE IF NOT EXISTS credit_card (  
9     id VARCHAR(8) PRIMARY KEY,  
10    iban VARCHAR(40),  
11    pan VARCHAR(20),  
12    pin VARCHAR(4),  
13    cvv VARCHAR(3),  
14    expiring_date VARCHAR(8)  
15    );
```

✓ 2 10:59:46 CREATE TABLE IF NOT EXISTS credit\_card ( id VARCHAR(8) PRIMARY KEY, ... 0 row(s) affected

Establecemos la relación entre tablas: en la **tabla transaction**, la columna credit\_card\_id es la clave foránea de la **tabla credit\_card**:

```
17 • ALTER TABLE transaction  
18     ADD CONSTRAINT fk_transaccion_tarjeta  
19     FOREIGN KEY (credit_card_id)  
20     REFERENCES credit_card(id);
```

ALTER TABLE transaction ADD CONSTRAINT fk\_transaccion\_tarjeta FOREI... 100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0



En el esquema añadimos una nueva tabla: credit\_card

En la tabla credit\_card se almacenan los datos de las tarjetas que han utilizado los usuarios para realizar sus compras. La tabla credit\_card se **relaciona** de forma directa con la de **transaction** en **1 – n**: tarjeta de crédito – muchas transacciones.

En cuanto a las claves primarias y foráneas:

- “id” es **Primary Key** en la tabla credit\_card y en la tabla transaction encontramos como **Foreign Key** “credit\_card\_id”

## Exercici 2

El departament de Recursos Humans ha identificat un error en el número de compte associat a la targeta de crèdit amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és: TR323456312213576817699999. Recorda mostrar que el canvi es va realitzar.

Realizamos la modificación del registro:

```

28 -- Modificamos el registro
29 • UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE id = 'CcU-2938';
  
```

5101 12:11:22 update credit\_card set iban = 'TR323456312213576817699999' where id = 'CcU-2938' 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

Mostramos el cambio realizado:

```

24 -- Mostramos el cambio
25 • SELECT id, iban
26 FROM credit_card
27 WHERE id='CcU-2938';
  
```

id	iban
CcU-2938	TR323456312213576817699999

5103 12:15:18 SELECT id, iban FROM credit\_card WHERE id='CcU-2938' 1 row(s) returned

## Exercici 3

En la taula "transaction" ingresa un nou usuari amb la següent informació:

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

Con la información aportada en el enunciado, no podemos crear un nuevo usuario de forma directa ya que el **user\_id** indicado **no existe en la tabla company**.

El user\_id de la tabla transaction es **FOREIGN KEY** de la tabla company y para poder ingresar un nuevo usuario, el **user\_id debe de existir previamente** en la tabla company.

Lo mismo nos ocurrirá con la columna credit\_card\_id de la tabla credit\_card.

Por lo tanto, primero añadimos un **nuevo registro en la tabla company**, poniendo el user\_id indicado y con el resto de datos en NULL ya que no nos los han proporcionado.

```
40 • INSERT INTO company (id, company_name, phone, email, country, website)
41   VALUES ('b-9999', NULL, NULL, NULL, NULL, NULL);
```

5116 13:04:28 insert into company (id, company\_name, phone, email, country, website) values ('b-9999',.... 1 row(s) affected

Lo mismo ocurre en la tabla credit\_card. El valor credit\_card\_id es PRIMARY KEY de la tabla transaction y como nos han proporcionado la información del valor, creamos el registro en la tabla credit\_card:

```
48 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)
49   VALUES ('CcU-9999', NULL, NULL, NULL, NULL, NULL);
```

**\*\***Cómo hemos hablado, en la entrega anterior había un error de pantallazo de output y por lo tanto, no se puede aportar.

Por último, después de actualizar los registros de los Primary Key de todas las tablas, creamos el nuevo usuario en la tabla de hechos Transactions, con los valores

proporcionados. La información de la columna timestamp no se nos la proporciona, por lo que dejamos el valor en NULL. Vemos que entonces sí que nos permite agregar el nuevo usuario:

```
44 • INSERT INTO transaction (id,credit_card_id,company_id, user_id, lat, longitude, timestamp, amount, declined)
45 VALUES ('10881D1D-5B23-A76C-55EF-C568E49A99DD','CcU-9999', 'b-9999', 9999, 829.999, -117.999, NULL, 111.11, 0);
```

5117 13:05:04 insert into transaction(id,credit\_card\_id,company\_id, user\_id, lat, longitude, timestamp, am... 1 row(s) affected

## Exercici 4

Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit\_card. Recordra mostrar el canvi realitzat.

Eliminamos el registro:

```
47 • ALTER TABLE credit_card DROP COLUMN pan;
```

5118 13:21:07 ALTER TABLE credit\_card DROP COLUMN pan

Visualizamos los datos de la tabla y vemos que ya no aparece la columna "pan":

```
49 • SHOW COLUMNS FROM credit_card;
```

	Field	Type	Null	Key	Default	Extra
►	id	varchar(8)	NO	PRI	NULL	
	iban	varchar(40)	YES		NULL	
	pin	varchar(4)	YES		NULL	
	cvv	varchar(3)	YES		NULL	
	expiring_date	varchar(8)	YES		NULL	

5120 13:24:10 SHOW COLUMNS FROM credit\_card 5 row(s) returned

## Nivell 2

### Exercici 1

Elimina de la taula transaction el registre amb ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de dades.

Eliminamos el registro:

```
56 • DELETE FROM transaction
57 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
```

1 10:27:17 DELETE FROM transaction WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CA... 1 row(s) affected

Comprobamos que el registro con id “000447FE-B650-4DCF-85DE-C7ED0EE1CAAD” ya no existe:

```
59 • SELECT id
60 FROM transaction
61 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
```

	id
*	NULL

✓ 2 10:28:14 SELECT id FROM transaction WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD' 0 row(s) returned

## Exercici 2

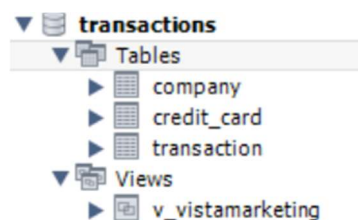
La secció de màrqueting desitja tenir accés a informació específica per a realitzar anàlisi i estratègies efectives. S'ha sol·licitat crear una vista que proporcioni detalls clau sobre les companyies i les seves transaccions. Serà necessària que creïs una vista anomenada VistaMarketing que contingui la següent informació: Nom de la companyia. Telèfon de contacte. País de residència. Mitjana de compra realitzat per cada companyia. Presenta la vista creada, ordenant les dades de major a menor mitjana de compra.

Creemos la vista solicitada. Añadimos alias para una correcta interpretación de datos:

```
77 • CREATE VIEW v_VistaMarketing AS
78 SELECT company_name AS Nombre_compañia, phone AS Telefono, country AS Pais, AVG (amount) AS Media_Compra
79 FROM company
80 JOIN transaction
81 ON company.id = transaction.company_id
82 WHERE declined = '0'
83 GROUP BY company.id
84 ORDER BY Media_Compra DESC;
```

✓ 9 11:33:54 CREATE VIEW v\_VistaMarketing AS SELECT company\_name AS Nombre\_compañia, phone AS Telefono, country AS Pais, AVG (amount) AS Media\_Compra 0 row(s) affected

Comprobamos en Views que se ha añadido la vista y realizamos consulta para comprobar que se visualiza correctamente:



79 • `SELECT * FROM v_VistaMarketing;`

	Nombre_compañia	Telefono	Pais	Media_Compra
▶	Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.911333
	Pretium Neque Corp.	07 77 48 55 28	Australia	275.578064
	Urna Convallis Associates	06 01 24 77 04	United States	273.566925
	At Associates	09 56 61 10 65	New Zealand	272.736985
	Metus Vitae Associates	08 25 44 40 66	Australia	270.051876
	Aliquet Diam Limited	02 76 61 47 46	United States	269.291145

✓ 13 23:00:23 `SELECT * FROM v_VistaMarketing`

101 row(s) returned

## Exercici 3

Filtra la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"

83 • `SELECT *`  
84 `FROM v_vistamarketing`  
85 `WHERE Pais = 'Germany';`

	Nombre_compañia	Telefono	Pais	Media_Compra
▶	Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.911333
	Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.319156
	Convallis In Incorporated	06 66 57 29 50	Germany	257.693651
	Ac Industries	09 34 65 40 60	Germany	255.169777
	Rutrum Non Inc.	02 66 31 61 09	Germany	255.137959
	Auctor Mauris Corp.	05 62 87 14 41	Germany	254.675099

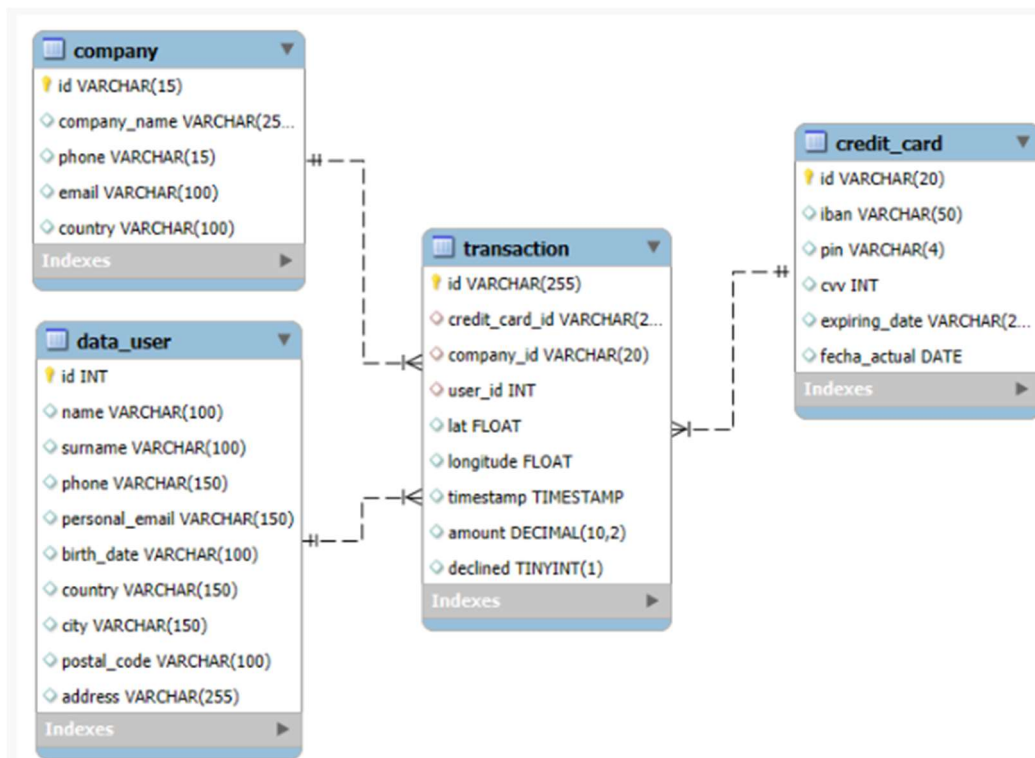
✓ 14 23:02:36 `SELECT * FROM v_vistamarketing WHERE Pais = 'Germany'`

8 row(s) returned

## Nivell 3

### Exercici 1

La setmana vinent tindràs una nova reunió amb els gerents de màrqueting. Un company del teu equip va realitzar modificacions en la base de dades, però no recorda com les va realitzar. Et demana que l'ajudis a deixar els comandos executats per a obtenir el següent diagrama:



Primero de todo, observamos que en el mapa hay una **nueva tabla “Data\_user”**, creamos la tabla y añadimos el archivo de datos de usuario a introducir.

```

1 CREATE TABLE IF NOT EXISTS user (
2     id CHAR(10) PRIMARY KEY,
3     name VARCHAR(100),
4     surname VARCHAR(100),
5     phone VARCHAR(150),
6     email VARCHAR(150),
7     birth_date VARCHAR(100),
8     country VARCHAR(150),
9     city VARCHAR(150),
10    postal_code VARCHAR(100),
11    address VARCHAR(255)
12 );
  
```

Relación entre tablas: Tabla user – id **PK**

Tabla transaction – user\_id **FK**

En este punto, no podemos establecer la relación entre tablas porque en un anterior ejercicio hemos añadido un nuevo registro. Creamos el id de usuario “9999”, que es la única información de que disponemos para la tabla user:

```

105 INSERT INTO user (id, name, surname, phone, personal_email, birth_date, country, city, postal_code, address)
106 VALUES ('9999', NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);
  
```



Ahora sí que podemos establecer la relación entre las dos tablas:

```
99     ALTER TABLE transaction
100     ADD CONSTRAINT fk_transaccion_usuario
101     FOREIGN KEY (user_id)
102     REFERENCES user(id);
```

✓ 4 16:46:53 ALTER TABLE transaction ADD CONSTRAINT fk\_transaccion\_usuario FOREIGN KEY (user\_id) REFERENCES user(id) 100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0

A continuación procedemos a realizar modificaciones para que los datos de las tablas queden de la misma forma:

## TABLA USER

Mostramos la información que contiene la tabla User y comparamos con la de nuestro compañero:

```
110 • SHOW COLUMNS FROM user;
```

Field	Type	Null	Key
id	char(10)	NO	PRI
name	varchar(100)	YES	
surname	varchar(100)	YES	
phone	varchar(150)	YES	
email	varchar(150)	YES	
birth_date	varchar(100)	YES	
country	varchar(150)	YES	
city	varchar(150)	YES	
postal_code	varchar(100)	YES	
address	varchar(255)	YES	



Comprobamos que en esta tabla, hay datos que no coinciden con el mapa que nos han proporcionado: nombre de tabla diferente, tipología de dato de la columna id diferente y nombre de la columna mail diferente. Hacemos las modificaciones oportunas:

Cambio de nombre de tabla

```
89 • ALTER TABLE user RENAME TO data_user;
```

✓ 5017 12:11:15 ALTER TABLE user RENAME TO data\_user 0 row(s) affected

Cambio tipo de dato en la columna id

```
91 • ALTER TABLE data_user
92     MODIFY COLUMN id INT;
```

✓ 5018 12:18:09 ALTER TABLE data\_user MODIFY COLUMN id INT 5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0



Cambio nombre en la columna email

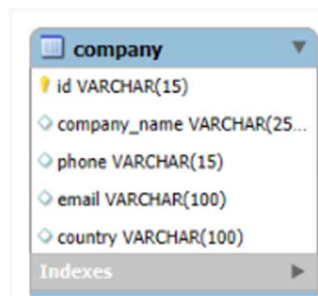
```
94 • ALTER TABLE data_user
95 CHANGE email personal_email VARCHAR(150);
```

✓ 5019 12:24:22 ALTER TABLE data\_user CHANGE email personal\_email VARCHAR(150) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

## TABLA COMPANY

```
97 • SHOW COLUMNS FROM company;
```

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	<b>NULL</b>	
company_name	varchar(255)	YES		<b>NULL</b>	
phone	varchar(15)	YES		<b>NULL</b>	
email	varchar(100)	YES		<b>NULL</b>	
country	varchar(100)	YES		<b>NULL</b>	
website	varchar(255)	YES		<b>NULL</b>	



En el diagrama proporcionado por nuestro compañero, no aparece la columna de website, así que eliminamos:

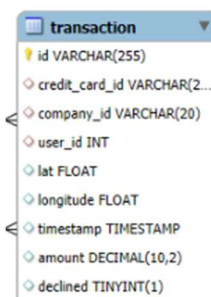
```
103 • ALTER TABLE company DROP COLUMN website;
```

✓ 5021 12:36:29 ALTER TABLE company DROP COLUMN website 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

## TABLA TRANSACTION

```
106 • SHOW COLUMNS FROM transaction;
```

Field	Type	Null	Key	Default	Extra
id	varchar(255)	NO	PRI	<b>NULL</b>	
credit_card_id	varchar(15)	YES		<b>NULL</b>	
company_id	varchar(20)	YES	MUL	<b>NULL</b>	
user_id	int	YES		<b>NULL</b>	
lat	float	YES		<b>NULL</b>	
longitude	float	YES		<b>NULL</b>	
timestamp	timestamp	YES		<b>NULL</b>	
amount	decimal(10,2)	YES		<b>NULL</b>	
declined	tinyint(1)	YES		<b>NULL</b>	



En el diagrama proporcionado, la cantidad de caracteres limite de credit\_card no se visibilizan completos (aparece 2...) No coincide con el límite que tenemos en nuestra

tabla, así que establecemos un número que empiece por 2 deberemos consultar cuál es el correcto.

```
128 • ALTER TABLE transaction
129     MODIFY credit_card_id VARCHAR(20);
```

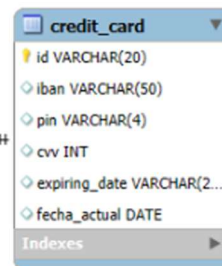
17 12:18:21 ALTER TABLE transaction MODIFY credit\_card\_id VARCHAR(20)

0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

## TABLA CREDIT CARD

```
109 • SHOW COLUMNS FROM credit_card;
```

	Field	Type	Null	Key	Default	Extra
▶	id	varchar(8)	NO	PRI	NULL	
	iban	varchar(40)	YES		NULL	
	pin	varchar(4)	YES		NULL	
	cvv	varchar(3)	YES		NULL	
	expiring_date	varchar(8)	YES		NULL	



credit_card	
id	VARCHAR(20)
iban	VARCHAR(50)
pin	VARCHAR(4)
cvv	INT
expiring_date	VARCHAR(20)
fecha_actual	DATE
Indexes	

Deberemos cambiar la longitud varchar de la columna id y de la columna iban. En expiring\_date no coincide la longitud pero nos pasa como anteriormente, no se muestra la cantidad exacta:

```
111 • ALTER TABLE credit_card
112     MODIFY id VARCHAR(20);
```

```
114 • ALTER TABLE credit_card
115     MODIFY iban VARCHAR(50);
```

```
140 • ALTER TABLE credit_card
141     MODIFY expiring_date VARCHAR(20);
```

También cambiamos el tipo de dato para la columna CVV:

```
118 • ALTER TABLE credit_card
119     MODIFY COLUMN cvv INT;
```

Por otro lado, en nuestra tabla original no tenemos la columna fecha\_actual, la añadimos.

Preguntaremos si los valores de esta columna los rellenaremos de forma manual, si debemos actualizarlos con la fecha actual o si se deben de ir rellenando a partir de ahora:

```
122 • ALTER TABLE credit_card
123     ADD COLUMN fecha_actual DATE;
```

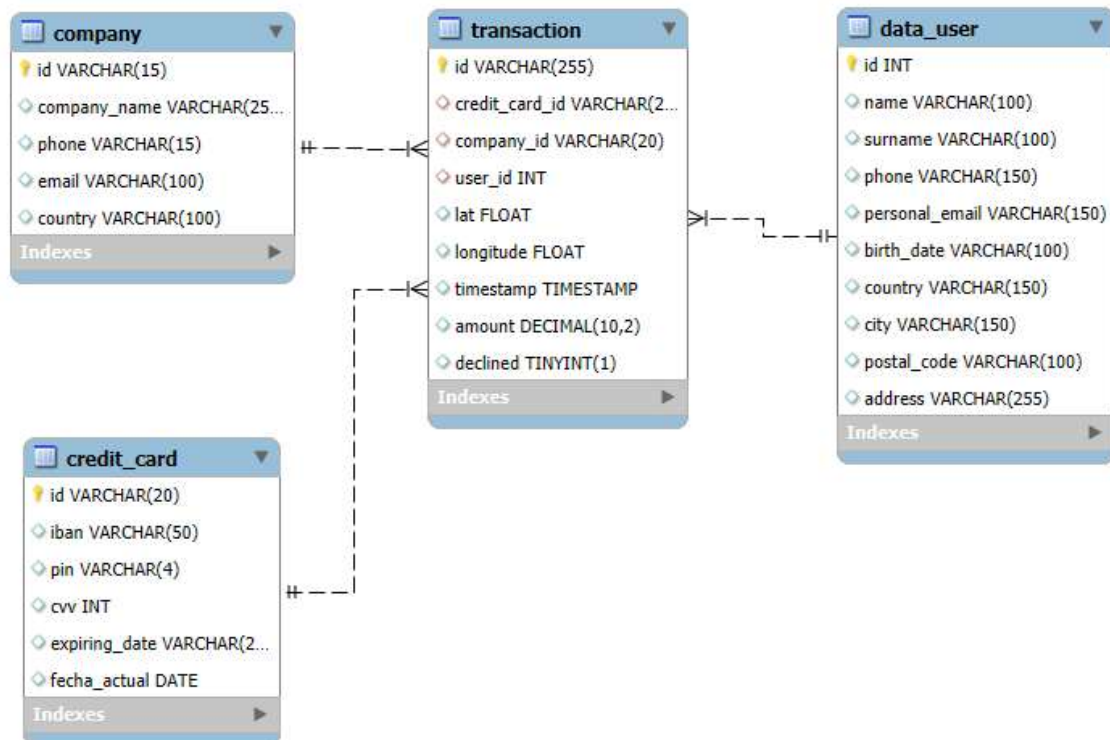
En el gráfico que nos proporciona nuestro compañero, vemos que tampoco se incluye la columna “pan” que ya habíamos eliminado anteriormente.

En el gráfico que nos muestra nuestro compañero, no aparece la vista VistaMarketing que habíamos creado anteriormente, así que eliminamos:

```
152 • DROP VIEW v_vistamarketing;
```

19 12:29:28 DROP VIEW v\_vistamarketing 0 row(s) affected

Finalmente, mostramos como quedan nuestras tablas después de los cambios:



## Exercici 2

L'empresa també us demana crear una vista anomenada "InformeTecnico" que contingui la següent informació:

- ID de la transacció
- Nom de l'usuari/ària
- Cognom de l'usuari/ària
- IBAN de la targeta de crèdit usada.
- Nom de la companyia de la transacció realitzada.
- Assegureu-vos d'incloure informació rellevant de les taules que coneixereu i utilitzeu àlies per canviar de nom columnes segons calgui.

Mostra els resultats de la vista, ordena els resultats de forma descendent en funció de la variable ID de transacció.

Creemos la vista:

```
129 • CREATE VIEW v_informetecnico AS
130 SELECT transaction.id AS ID_transacción, data_user.name AS Nombre_usuario, data_user.surname AS Apellido_usuario, iban AS IBAN, company_name
131 AS Nombre_compañía
132 FROM transaction
133 JOIN data_user ON transaction.user_id = data_user.id
134 JOIN credit_card ON transaction.credit_card_id = credit_card.id
135 JOIN company ON transaction.company_id = company.id
136 ORDER BY transaction.id DESC;
```

8 22:42:14 CREATE VIEW v\_InformeTecnico AS SELECT transaction.id AS ID\_transacción, data\_use... 0 row(s) affected

Comprobamos que se vea correctamente:

```
138 • SELECT *
139 FROM v_informetecnico;
```

	ID_transacción	Nombre_usuario	Apellido_usuario	IBAN	Nombre_compañía
▶	FFFD31D6-9495-47CE-B54A-7DB8E1CC274B	Bmrgli	Tprvmrc	XX794814451211289182490922	Turpis Company
	FFFCF76D-ECF0-4985-A2D0-B2A7B75998FC	Dfrled	Vilqcdl	XX636251701647892036676034	Amet Nulla Donec Corporation
	FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	Securp	Faofvqfy	XX162677143304223631437567	Nunc Interdum Incorporated
	FFFB270D-F53A-4D5D-9666-E5307C53CC84	Ggzjpa	Uirzjulh	XX395114267082019952567052	Viverra Donec Foundation
	FFF9E3CE-234E-408C-A8EF-F9CAD577224A	Yshimq	Zpsjsleed	XX8845462156537570367941	Convallis In Incorporated
	FFF9E178-6CD2-4DF9-99B0-49AE068809B1	Jevepx	Xwczwnm	XX321405515711654384711481	Mus Aenean Eget Foundation
	FFFC770D-17B5-4B4E-A5D0-58033AAA449F	Fdled	Ldfrsd	XX7814634333660030730496	Curabitur Aliquam Tincidunt

9 22:44:35 SELECT \* FROM v\_informetecnico 99999 row(s) returned