

SPRINT 3

En aquest sprint, es simula una situació empresarial en la qual has de realitzar diverses manipulacions en les taules de la base de dades. Al seu torn, hauràs de treballar amb índexs i vistes. En aquesta activitat, continuaràs treballant amb la base de dades que conté informació d'una empresa dedicada a la venda de productes en línia. En aquesta tasca, començaràs a treballar amb informació relacionada amb targetes de crèdit.

NIVELL 1

1.1.- 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.

Paso 1:

```
3 -- 1.1.- La teva tasca és dissenyar i crear una taula anomenada "credit_card" que emmagatzemi detalls crucials sobre les targetes de crèdit.
4 -- 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").
5 -- Després de crear la taula serà necessari que ingressis la informació del document denominat "dades_introduir_credit".
6 -- Recorda mostrar el diagrama i realitzar una breu descripció d'aquest.
7
8 • create table credit_card (
9     id varchar (255) primary key not null,           -- NOT NULL ya que una PK no puede adquirir este valor
10    iban varchar (255) null,
11    pan varchar (255) null,
12    pin int null,
13    cvv int null,
14    expiring_date varchar (10) null;                  -- Tipo varchar por el formato de la fecha no estandard mm/dd/aaaa
15
```

En esta primera query creo la tabla credit_card con la instrucción **CREATE TABLE** e indicando el nombre de los campos que la forman así como el tipo de datos admitidos en cada uno de ellos.

He utilizado *CONSTRAINT* (restricciones) como:

- **PRIMARY KEY:** identifico como PK el campo id de la tabla. Cada tarjeta dispondrá de un código de identificación único.
 - **NOT NULL:** para el campo id, ya que una PK no puede adquirir el valor NULL nunca.
 - **NULL:** el resto de campos los restrinjo a que si puedan adquirir el valor NULL ya que al importar datos externos, quizás fuese posible que no se aportaran todos y en ese caso el nuevo registro no se crearía.

En cuanto a los campos del tipo *VARCHAR()* he decidido permitir una cantidad de 255 dígitos para ‘asegurar’ que no sea una limitación en caso de tener que ampliar la tabla con datos externos.

Paso 2:

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree shows the 'transactions' schema selected. Under 'Tables', the 'credit_card' table is highlighted. The 'Object Info' tab displays the table's definition, including columns: id (varchar(255) PK), iban (varchar(255)), pan (varchar(255)), pin (int), cvv (int), and expiring_date (varchar(10)). The 'Session' tab shows the connection details.

In the main pane, a query editor window titled 'MNG_Sprint 03 v.1' contains the following SQL script:

```

1 • SELECT * FROM transactions.credit_card;

```

The results grid shows 275 rows of data inserted into the 'credit_card' table. The columns and their values are:

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR301950312213576817638661	542446566813633	3257	984	10/30/22
CcU-2945	DO2685476374853745216568689	5142423821948828	9080	887	08/24/23
CcU-2952	GRY24547724435841535	372461977349375	5888	438	08/24/23
CcU-2966	BG72LKD15627629429456346	448566887477265	4900	130	10/29/24
CcU-2973	PT8780628135092429456346	544 59654 54343 384	8760	887	01/30/25
CcU-2980	DE3924188183086277136	402400 7145845996	5075	596	07/24/24
CcU-2987	GE89881434837748781813	3763 747887 76866	2298	797	10/31/23
CcU-2994	RHR771442839066765294	34428327325293	7545	595	02/28/22

The 'Action Output' section shows the execution log with 275 entries, each indicating a successful insert operation. The log includes the time of execution, the action performed (INSERT INTO credit_card), and the number of rows affected (1 row(s) affected).

Una vez creada la tabla, procedo a completarla ejecutando el script facilitado para ello (*dades_introducir_credit.sql*).

Se completa con **275 registros** que facilitan datos de cada una de las tarjetas utilizadas para la realización de transacciones.

Paso 3:

```

16 • ALTER TABLE transaction          -- Genero nueva FK en la tabla transaction para poder relacionar
17   ADD CONSTRAINT FK_CreditCard
18     FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
19

```

100% 1:16

Action Output

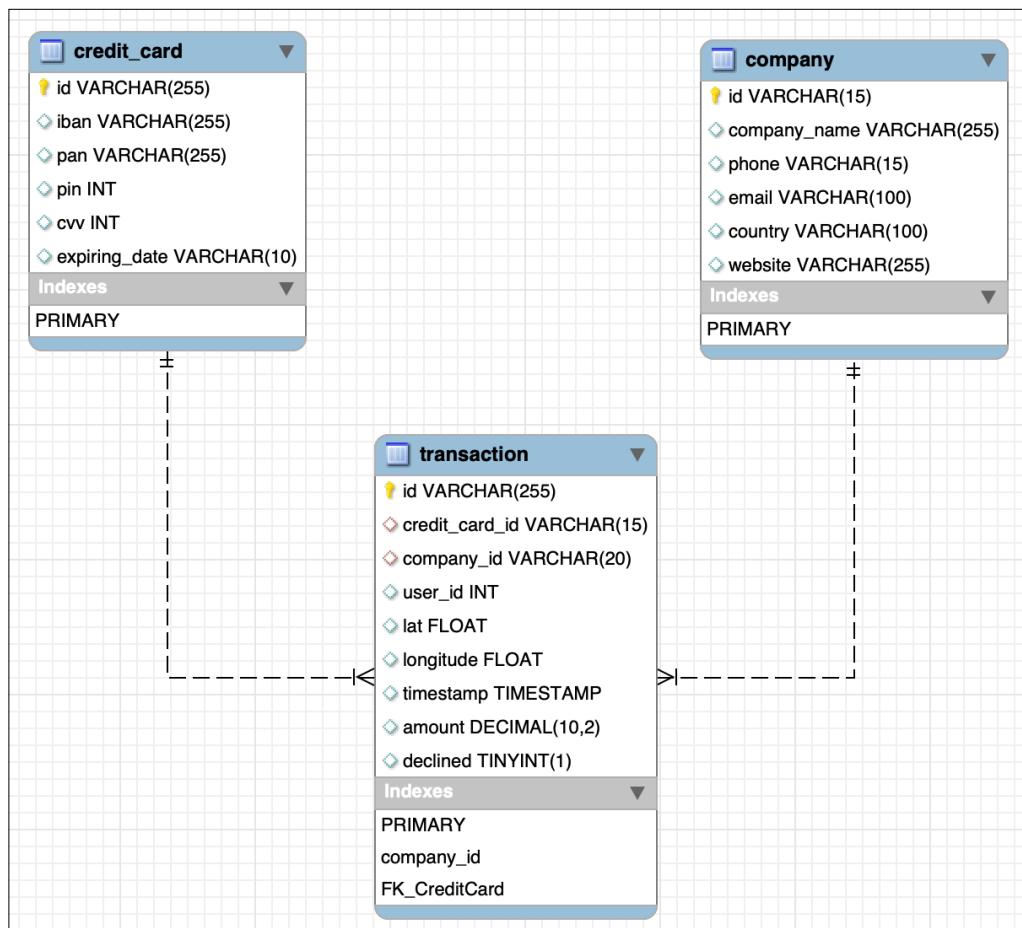
Time	Action	Response
14:31:50	ALTER TABLE transaction...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

La nueva tabla la integro en el modelo relacionándola con la tabla ***transaction***.

Lo hago a través del campo ***id*** de la tabla ***credit_card*** con el campo ***credit_card_id*** de la tabla ***transaction***, ya que ambos campos hacen referencia al identificador de cada una de las tarjetas de crédito.

Para poder efectuar la relación necesito definir una nueva ***Foreign Key (FK)*** en la tabla existente (***transaction***). Para ello utilizo la instrucción **ALTER TABLE** combinada con la **ADD CONSTRAINT**, para a la vez restringir el uso del campo ***credit_card_id*** como FK.

Quedando como modelo relacional el siguiente:



1.2.- El departament de Recursos Humans ha identificat un error en el número de compte de l'usuari amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és: R323456312213576817699999. Recorda mostrar que el canvi es va realitzar.

```

21 -- 1.2.- El departament de Recursos Humans ha identificat un error en el número de compte de l'usuari amb ID CcU-2938.
22 -- La informació que ha de mostrar-se per a aquest registre és: R323456312213576817699999. Recorda mostrar que el canvi es va realitzar.
23
24 • update credit_card          -- Modificación del IBAN asociado
25   set iban='R323456312213576817699999'
26   where id='CcU-2938';
27
28 • select id,iban,
29   case
30     when id!='CcU-2938' then 'No'
31     else 'IBAN actualizado'
32   end as 'ACTUALIZACION IBAN'
33   from credit_card;
34

```

Result Grid Filter Rows: Search Export:

id	iban	ACTUALIZACION IBAN
CcU-2938	R323456312213576817699999	IBAN actualizado
CcU-2945	DO26854763748537475216568889	No
CcU-2952	BG45IVQL52710525608255	No
CcU-2959	CR7242477244335841535	No
CcU-2966	BG72LKTQ15627628377363	No
CcU-2973	PT87806228135092429456346	No
CcU-2980	DE39241881883066277136	No
CcU-2987	GFR86R143483774R7R1R13	No

Action Output

Time	Action	Response
16:01:38	update credit_card	-- Modificación del IBAN asociado set iban='R323456312213576817699999' where id='CcU... 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
16:01:45	select id,iban, case	-- Añadir mensaje modificación when id!='CcU-2938' then 'No' else 'IBAN actu... 275 row(s) returned

En las primeras líneas de código actualizo el campo iban del registro cuyo *id* es el del enunciado, mediante la sentencia **UPDATE** y el filtro **WHERE**.

La segunda parte del código corresponde a la query que devuelve la relación entre el *id* y el *iban* de cada tarjeta de crédito y además muestra si éste último ha sido modificado.

Utilizo la sentencia **CASE** para añadir ese campo *ACTUALIZACION IBAN* a la query.

1.3.- En la taula "transaction" ingressa 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

```
-- 1.3.- En la taula "transaction" ingressa un nou usuari amb la següent informació:
39      /*  Id      108B1D1D-5B23-A76C-55EF-C568E49A99DD
40      credit_card_id    CcU-9999
41      company_id      b-9999
42      user_id        9999
43      lat            829.999
44      longitude      -117.999
45      amount         111.11
46      declined       0
47
48  -- Paso 1:
49 •  insert into company(id)      -- Asegurar PK vs. FK entre las tablas company y transaction, para evitar problemas con los CONSTRAINT
50   values ('b-9999');
51
52 •  insert into credit_card(id)  -- Asegurar PK vs. FK entre las tablas credit_card y transaction, para evitar problemas con los CONSTRAINT
53   values ('CcU-9999');
54
55  -- Paso 2:
56 •  insert into transaction (id,credit_card_id,company_id,user_id,lat,longitude,amount,declined)      -- Inserto los valores en la tabla
57   values ('108B1D1D-5B23-A76C-55EF-C568E49A99DD','CcU-9999','b-9999',9999,829.999,-117.999,111.11,0);  -- referenciada
58
59  -- Paso 3:
60 •  select *                  -- Mostrar que los datos se han introducido correctamente
61   from transaction
62   where company_id='b-9999';
63
```

Result Grid Filter Rows: Search Edit: Export/Import:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	HULL	111.11	0
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

transaction 10

Action Output

Time	Action	Response
1 18:07:12	select *	-- Mostrar que los datos se han introducido correctamente from transaction where company_id='b-9999'

1 row(s) returned

Paso 1:

Para poder insertar un nuevo registro en una tabla, debo tener en cuenta las restricciones que puedan tener las llaves primaria (**PK**) y foránea (**FK**). Es por eso que en este paso aseguro que en todas las tablas con las que esta relacionada la tabla en la que insertamos el registro, incluyen las nuevas **PK** y **FK** que se incluyen en el nuevo registro.

Con la instrucción **INSERT INTO ... VALUES ...** es posible realizar esta acción.

Paso 2:

Una vez aseguramos que nuevas PK y FK están en las tablas 'padre' (en este caso company y credit_card), mediante la instrucción **INSERT INTO ... VALUES ...** actualizo la tabla transaction con un nuevo registro.

Paso 3:

Hago esta query para comprobar que los datos se han introducido correctamente, mediante el filtro **WHERE**.

1.4.- Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card. Recorda mostrar el canvi realitzat.

```

65 -- 1.4.- Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card. Recorda mostrar el canvi realitzat
66
67 • alter table credit_card drop column pan;
68
69 • show columns from credit_card;
70
100% 4:65

```

Field	Type	Null	Key	Default	Extra
id	varchar(255)	NO	PRI	NULL	
iban	varchar(255)	YES		NULL	
pin	int	YES		NULL	
cvv	int	YES		NULL	
expiring_date	varchar(10)	YES		NULL	

Result 11

Action Output	Time	Action	Response
1	18:21:51	alter table credit_card drop column pan	0 row(s) affected Records: 0 Duplic...
2	18:21:56	show columns from credit_card	5 row(s) returned

Con la instrucción **ALTER TABLE ... DROP COLUMN ...** es posible eliminar la columna de la tabla.

Con la instrucción **SHOW COLUMNS** compruebo que la columna pan se ha eliminado de la tabla **credit_card** (también lo podría haber comprobado con la instrucción **DESCRIBE**).

NIVELL 2

2.1.- Elimina de la taula transaction el registre amb ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de dades.

```

74 -- 2.1.- Elimina de la taula transaction el registre amb ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de dades.
75
76 • delete from transaction where id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
77
78 • select *
79   from transaction
80   where id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
81
82

```

Result Grid Filter Rows: Search Edit: Export/Import:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 1

Action Output

	Time	Action	Response
1	20:53:23	delete from transaction where id='02C6201E-D90A-1859-B4EE-88D2986D3B02'	1 row(s) affected
2	20:59:27	select * from transaction where id='02C6201E-D90A-1859-B4EE-88D2986D3B02'	0 row(s) returned

Con la instrucción **DELETE** combinada con el filtro **WHERE** elimino el registro de la tabla transaction cuyo *id*=02C6201E-D90A-1859-B4EE-88D2986D3B02.

Con la siguiente query compruebo que el registro se ha eliminado.

2.2.- 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.

```

83 -- 2.2.- La secció de màrqueting desitja tenir accés a informació específica per a realitzar anàlisi i estratègies efectives.
84 -- S'ha sol·licitat crear una vista que proporcioni detalls clau sobre les companyies i les seves transaccions.
85 -- Serà necessària que creïs una vista anomenada VistaMarketing que contingui la següent informació:
86 -- Nom de la companyia.
87 -- Telèfon de contacte.
88 -- País de residència.
89 -- Mitjana de compra realitzat per cada companyia.
90 -- Presenta la vista creada, ordenant les dades de major a menor mitjana de compra.
91
92 • create view VistaMarketing as
93   select company_name,phone,country,round(avg(amount),2) as 'MEDIA DE COMPRAS' -- Query que devuelve los datos solicitados
94   from company
95   join transaction on company.id=transaction.company_id
96   group by company_name,phone,country
97   order by round(avg(amount),2) desc;
98
100% 4:83
Action Output
Time Action Response
1 21:04:02 create view VistaMarketing as select company_name,phone,country,round(avg(amount),2) as 'MEDIA DE COMPRAS' -- Query que d... 0 row(s) affected

```

Con la sentencia **CREATE VIEW** más la query que la acompaña se genera la vista.
La query está formada por una **JOIN** entre las tablas **company** y **transaction**.

La vista queda almacenada dentro de la base de datos (o **SCHEMA**) **transactions**, dentro de la carpeta **Views**.

company_name	phone	country	MEDIA DE COMPRAS
Eget Ipsum Ltd	03 67 44 56 72	United States	473.08
Non Magna LLC	06 71 73 13 17	United Kingdom	468.35
Sed Id Limited	07 28 18 18 13	United States	461.21
Justo Eu Arcu Ltd	08 42 56 71 52	Italy	443.64
Eget Tincidunt Dui Institute	05 35 93 32 44	Netherlands	442.52
Viverra Donec Foundation	03 33 12 32 73	United Kingdom	442.28
Vestibulum Lorem PC	02 02 87 33 40	Belgium	434.06
Aliquet Diam Limited	02 76 61 47 46	United States	425.64
Maecenas Malesuada Fringilla Inc.	09 38 53 76 61	Netherlands	408.62
Non Ante LLP	08 89 47 65 08	Sweden	407.79
Egestas Nunc Sed Limited	06 01 02 70 47	Italy	406.11
Nunc Sit Incorporated	07 28 42 63 63	Norway	405.36

2.3.- Filtra la vista VistaMarketing per a mostrar només les companies que tenen el seu país de residència en "Germany".

```

100 -- 2.3.- Filtra la vista VistaMarketing per a mostrar només les companies que tenen el seu pais de residència en "Germany"
101 • select *
102   from VistaMarketing
103   where country='Germany';
104
100% 4:100

```

Result Grid Filter Rows: Search Export:

company_name	phone	country	MEDIA DE COMPRAS
Aliquam PC	01 45 73 52 16	Germany	385.27
Ac Industries	09 34 65 40 60	Germany	289.65
Rutrum Non Inc.	02 66 31 61 09	Germany	266.90
Nunc Interdum Incorporated	05 18 15 48 13	Germany	244.03
Augue Foundation	06 88 43 15 63	Germany	240.80
Ac Fermentum Incorporated	06 85 56 52 33	Germany	206.47
Auctor Mauris Corp.	05 62 87 14 41	Germany	184.31
Convallis In Incorporated	06 66 57 29 50	Germany	156.73

VistaMarketing 3

Action Output

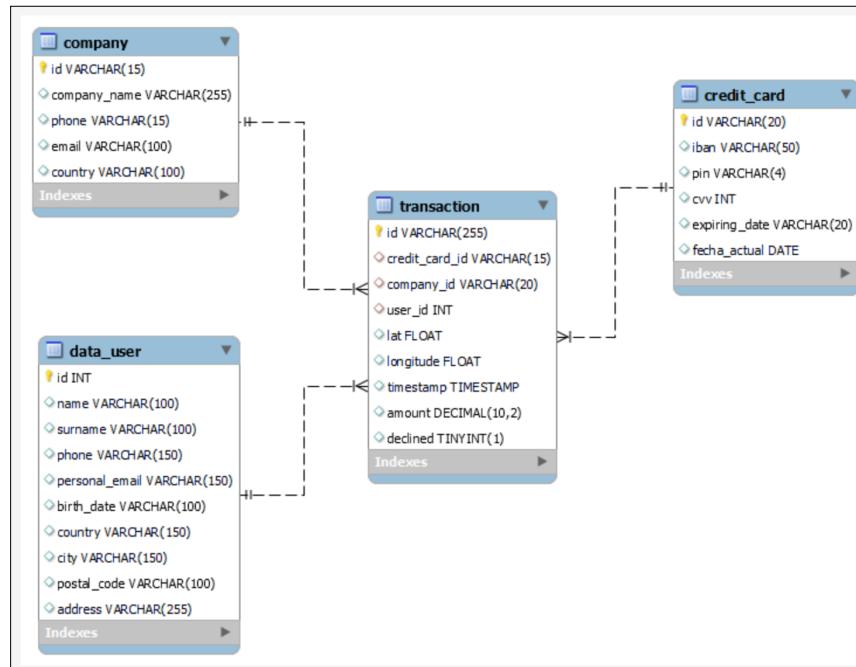
Time	Action	Response
21:11:21	select * from VistaMarketing where country='Germany'	8 row(s) returned

Una vez generada la vista, se pueden realizar **consultas** sobre ella.

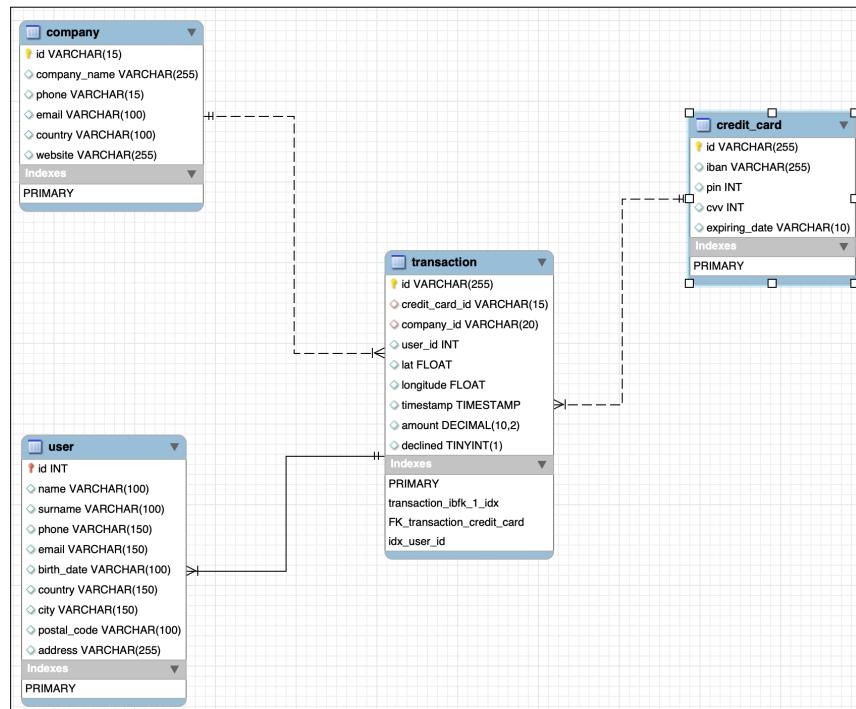
Con el filtro **WHERE** seleccionamos los registros de la vista cuyo campo country contiene la palabra **Germany**.

NIVELL 3

3.1.- 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:



Comparando el esquema del enunciado con el diagrama de relación que se obtiene de la actualización de las tablas a través de los archivo auxiliares, será necesario realizar varios cambios en las tablas así como en alguna que otra relación entre ellas.



Cambio del tipo de relación entre las tablas data-user y transaction:

Si compruebo como se ha generado la tabla data_user veo que existe un **CONSTRAINT...FOREIGN KEY** con la tabla transaction entre los campos data_user.id y transaction.user_id. El sentido de esta relación no es el correcto, ya que debe ser al revés, es decir, de 1:N (un usuario puede realizar varias transacciones).

```

153  -- Cambio de tipo de relación entre las tablas credit_card y transaction
154 • ALTER TABLE user           -- Borrar la constraint de data_user
155   DROP FOREIGN KEY user_ibfk_1;
156
157 • insert into user (id)      -- Añadir el nuevo usuario (9999) del ejercicio 1.3 (de lo contrario no se cumple la restricción)
158   values (9999);
159
160 • ALTER TABLE transaction    -- Crear una nueva FK en transaction
161   ADD CONSTRAINT FK_transaction_user_id
162   FOREIGN KEY (user_id) REFERENCES user(id);
163
100%  C | 4:153 |
```

Action Output ▾

Time	Response
1 12:14:15	... 9 row(s) returned

Debo rehacer la relación, primero eliminando el **CONSTRAINT...FOREIGN KEY...** de la tabla user y generando uno nuevo en la tabla transaction. Esto lo realizo con el primer **ALTER TABLE** combinado con **DROP FOREIGN KEY**.

Al intentar volver a generar la relación mediante una nueva **FOREIGN KEY** (FK) en la tabla transaction no se realizada la operación por detectarse un fallo con el **CONSTRAINT** y se debía a que en la tabla user faltaba incluir el id (9999) generado como nuevo en el ejercicio 1.3. Es por eso que mediante la instrucción **INSERT INTO** incluyo el *id* 9999.

Finalmente, con un nuevo **ALTER TABLE...ADD CONSTRAINT...FOREIGN KEY** consigo restablecer la relación entre ambas tablas.

Cambios en la tabla company:

En la modificación no aparece el campo (o atributo) `website`.

Lo elimino con la sentencia **ALTER TABLE ... DROP COLUMN ...** y seguidamente compruebo mediante un **DESCRIBE** que la operación se ha realizado.

```

112    -- Cambios tabla company
113    -- Eliminar el campo website
114 • alter table company drop column website; -- Instrucción para eliminar el campo
115 • describe company;                      -- Comprobación de que el campo ha sido eliminado
116
 100%  4:112

```

Result Grid Filter Rows: Search Export:

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

Result 3

Action Output

	Time	Action	Response
1	13:13:18	describe company	6 row(s) returned
2	13:13:56	alter table company drop column website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3	13:14:19	describe company	5 row(s) returned
4	13:14:53	describe company	5 row(s) returned

Cambios en la tabla credit_card:

```

118 -- Cambiar el parámetro de tamaño para las VARCHAR de los campos id(255->20)
119 • ALTER TABLE transaction                                -- 'Rompo' la relación entre tablas, ya que el campo es FK de otra tabla
120   DROP CONSTRAINT FK_transaction_credit_card ;
121
122 • alter table credit_card modify column id varchar(20);    -- Modifico el tipo del campo
123
124 • ALTER TABLE transaction                                -- Genero de nuevo la FK en la tabla transaction para poder relacionar
125   ADD CONSTRAINT FK_transaction_credit_card
126     FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
127
128 -- Cambiar el parámetro de tamaño para las VARCHAR de los campos iban(255->50) y expiring_date(10->20)
129 • alter table credit_card modify column iban varchar(50);
130 • alter table credit_card modify column expiring_date varchar(20);
131
132 -- Cambiar el tipo al campo pin(INT->VARCHAR(4))
133 • alter table credit_card modify column pin varchar(4);
134
135 -- Añadir un nuevo campo fecha_actual(DATE)
136 • alter table credit_card add fecha_actual date null;
137
138 -- Compruebo los cambios realizados
139 • describe credit_card;
140

```

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI	<input type="button" value="NULL"/>	
iban	varchar(50)	YES		<input type="button" value="NULL"/>	
pin	varchar(4)	YES		<input type="button" value="NULL"/>	
cvv	int	YES		<input type="button" value="NULL"/>	
expiring_date	varchar(20)	YES		<input type="button" value="NULL"/>	
fecha_actual	date	YES		<input type="button" value="NULL"/>	

Result 3

Action Output

Time	Action	Response
16:03:36	describe credit_card	6 row(s) returned

En esta tabla, los cambios son relacionados con el cambio del tipo de datos almacenados en las columnas de la tabla, que se pueden modificar con la instrucción **ALTER TABLE**.

- **id**: debe ser del mismo tipo pero de tamaño diferente, debe pasar de **VARCHAR(255)** a **VARCHAR(20)**. Al ser un campo utilizado para relacionar tablas (forma parte del *CONSTRAINT FK* de la tabla *transaction*) primero debo eliminar dicha relación, modificar el dato y volver a crear la relación entre las tablas *transaction* y *credit_card*.
- **iban**: debe ser del mismo tipo pero de tamaño diferente, debe pasar de **VARCHAR(255)** a **VARCHAR(50)**.
- **pin**: en este caso cambia por completo de tipo de dato, debe pasar de **INT** a **VARCHAR(4)**.
- **expiring_date**: debe ser del mismo tipo pero de tamaño diferente, debe pasar de **VARCHAR(10)** a **VARCHAR(20)**.

Adicionalmente hay que crear un nuevo atributo (o campo) a la tabla, del tipo DATE:

- **fecha_actual**: del tipo DATE que puede adquirir valores NULL, por si se tuviese que actualizar de manera externa la tabla y no se pasará ningún valor para este campo.

Cambios en la tabla user:

```

141  -- Cambios tabla user
142  -- Cambiar el nombre de la tabla user->data_user
143 • alter table user
144  rename to data_user;
145
146  -- Cambiar el nombre del campo email->personal_email
147 • alter table data_user
148  rename column email to personal_email;
149
150  -- Comprobación de los cambios realizados
151 • describe data_user;
152
153
 100% 4:141

```

Result Grid Filter Rows: Search Export:

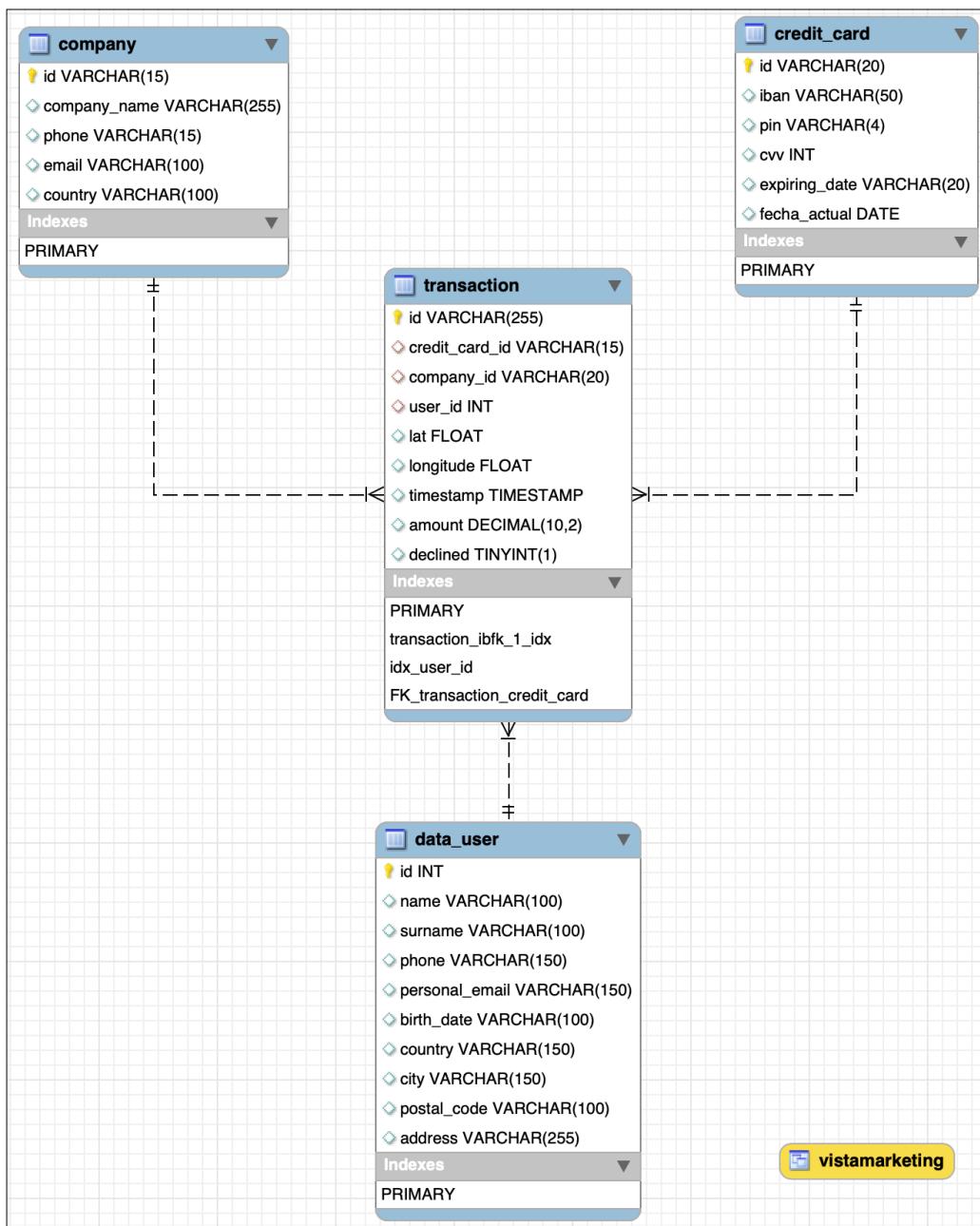
Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
surname	varchar(100)	YES		NULL	
phone	varchar(150)	YES		NULL	
personal_email	varchar(150)	YES		NULL	
birth_date	varchar(100)	YES		NULL	
country	varchar(150)	YES		NULL	
city	varchar(150)	YES		NULL	
postal_code	varchar(100)	YES		NULL	
address	varchar(255)	YES		NULL	

Action Output

Time	Action	Response
1 16:39:26	alter table user rename to data_user	0 row(s) affected
2 16:41:13	alter table data_user rename column email to personal_email	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3 16:41:45	describe data_user	10 row(s) returned

En esta tabla, los cambios son relacionados con el cambio del nombre de una columna y de propia tabla, que se pueden modificar con la sentencia **ALTER TABLE**, combinada con la instrucción **RENAME** y **RENAME COLUMN**

El esquema final es el siguiente:



3.2.- L'empresa també et sol·licita 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.

Assegura't d'incloure informació rellevant de totes dues taules i utilitza àlies per a canviar de nom columnes segons sigui necessari.

Mostra els resultats de la vista, ordena els resultats de manera descendente en funció de la variable ID de transaction.

```
-- 3.2.- L'empresa també et sol·licita crear una vista anomenada "InformeTecnico" que contingui la següent informació:
168 -- ID de la transacció
169 -- Nom de l'usuari/ària
170 -- Cognom de l'usuari/ària
171 -- IBAN de la targeta de crèdit usada.
172 -- Nom de la companyia de la transacció realitzada.
173 -- Assegura't d'incloure informació rellevant de totes dues taules i utilitza àlies per a canviar de nom columnes segons sigui necessari.
174 -- Mostra els resultats de la vista, ordena els resultats de manera descendente en funció de la variable ID de transaction.

175
176
177 • create view InformeTecnico as
178 select transaction.id as idTransaction,data_user.name as userName,data_user.surname as userSurname,credit_card.iban as ibanCreditCard,company.company_name as companyName
179 from transaction
180 join data_user on transaction.user_id=data_user.id
181 join credit_card on transaction.credit_card_id=credit_card.id
182 join company on transaction.company_id=company.id
183 order by transaction.id DESC;
184
100% 62:181
185
```

Action	Output	Response
Time	Action	
1	13:00:56	create view InformeTecnico as select transaction.id as idTransaction,data_user.name as userName,data_user.surname as userSurname,credit_card.iban as ibanCreditCard,company.company_name as companyName... 0 row(s) affected

Sobre el modelo definitivo, mediante la sentencia **CREATE VIEW** genero la vista solicitada.

Con la instrucción **SELECT** selecciono los campo a incluir en la vista, renombrándolos con la ayuda del comando **AS**.

Son necesarias hasta tres JOIN para acceder a las tablas que contienen los campos que se incluyen en la vista:

- ID de la transacción en la tabla *transaction*.
- Nombre del usuario/a en la tabla *data_user*.
- Apellido del usuario/a en la tabla *data_user*.
- IBAN de la tarjeta de crédito usada en la tabla *credit_card*.
- Nombre de la compañía de la transacción realizada en la tabla *company*.