

SPRINT 3

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 forma 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 "datos_introducir_credit". Recuerda mostrar el diagrama y realizar una breve descripción del mismo.

Primero, creo la tabla **credit_card**:

```
8
9 • USE transactions;
10 • CREATE TABLE credit_card (
11     id VARCHAR(10) UNIQUE,
12     iban VARCHAR(255) NULL,
13     pan VARCHAR(255) NULL,
14     pin VARCHAR(4) NULL,
15     cvv VARCHAR(3) NULL,
16     expiring_date VARCHAR(8) NULL,
17     PRIMARY KEY (id)
18 );
```

Output				
#	Time	Action	Message	Duration / Fetch
693	11:48:42	USE transactions	0 row(s) affected	0.000 sec
694	11:49:19	CREATE TABLE credit_card (id VARCHAR(10) UNIQUE, iban VARCHAR(255) NULL, pan VARCHAR(255) NULL, pin VARCHAR(4) NULL, cvv VARCHAR(3) NULL, expiring_date VARCHAR(8) NULL, PRIMARY KEY (id))	0 row(s) affected	0.063 sec

El siguiente paso es cargar los datos a la tabla **credit_card**. Para ello abro el archivo con los datos:

Output				
#	Time	Action	Message	Duration / Fetch
968	11:54:26	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2938', 'TR301950312213576817638661', '5424465566813633', '3257', '984', '10/30/22')	1 row(s) affected	0.000 sec
969	11:54:26	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2945', '0026854763748537475216568689', '5142423821948828', '9080', '887', '08/24/23')	1 row(s) affected	0.000 sec

Y después visualizo la nueva tabla:

```
19
20 • SELECT * FROM credit_card;
```

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22
CcU-2945	DO26854763748537475216568689	5142423821948828	9080	887	08/24/23
CcU-2952	BG45IVQL52710525608255	4556453555287	4598	438	06/29/21
CcU-2959	CR724247724435841535	372461377349375	3583	667	02/24/23
CcU-2965	RG72LKTQ15627628377363	448569867477265	4900	130	10/29/24
CcU-2973	PT8780622813592429456346	5443865454343384	8760	887	01/30/25
CcU-2980	DE392418818308627713	4024007145845969	5075	596	07/24/22
CcU-2987	GE89681434837748781813	376374768776666	2298	797	10/31/23
CcU-2994	BH62714428368066765294	344283273252593	7545	595	02/28/22
CcU-3001	CY4908742654774581266832110	5117229248332244	9562	867	09/16/22
CcU-3008	LU507216693616119230	4485744464433884	1856	740	04/05/25
CcU-3015	PS119398216295715968342456821	378466223317389	3246	822	01/31/22
CcU-3022	GT91695162850556977423121857	5164137948423951	5610	342	04/25/25

credit_card 1 x

Action Output

#	Time	Action	Message	Duration / Fetch
969	11:54:26	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4856', 'TR3738725583...', 1 row(s) affected	0.000 sec	
970	11:55:54	SELECT * FROM credit_card	0.000 sec / 0.000 sec	

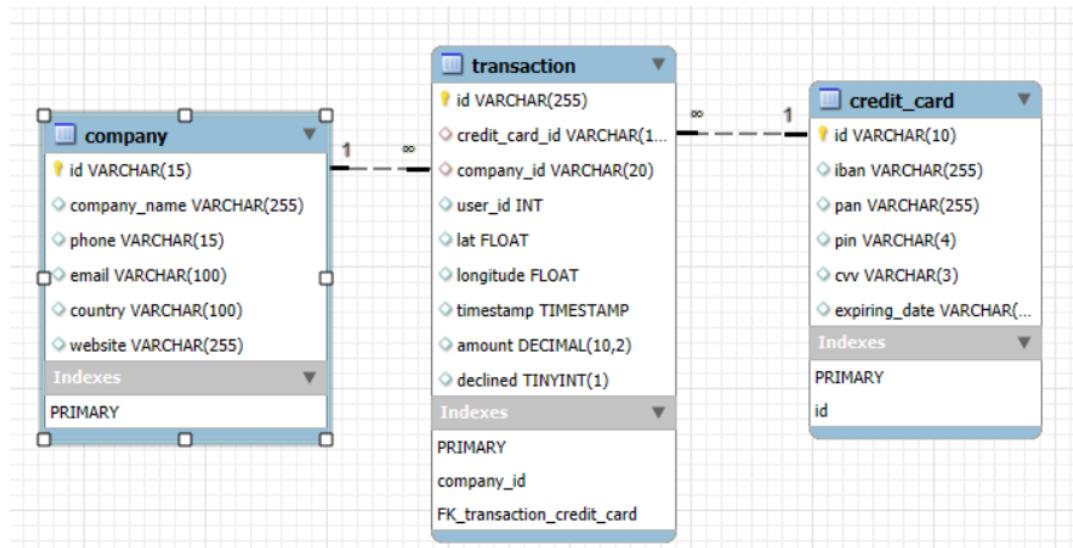
Para establecer las relaciones entre tablas introduzco el siguiente comando:

```
22 • ALTER TABLE transaction          -- Genero una nueva FK en la tabla transaction para poder relacionar credit_card con transaction
23   ADD CONSTRAINT FK_transaction_credit_card
24   FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
```

Action Output

#	Time	Action	Message	Duration / Fetch
970	11:55:54	SELECT * FROM credit_card	275 row(s) returned	0.000 sec / 0.000 sec
971	11:58:20	ALTER TABLE transaction -- Genero una nueva FK en la tabla transaction para poder relacionar cre... 587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0	0.125 sec

Y finalmente puedo ver el diagrama del modelo creado:



La base de datos sigue teniendo un modelo estrella, en el que la tabla de hechos es **transaction**, y las de dimensiones **company** y la nueva tabla añadida, **credit_card**.

Esta nueva tabla tiene diferentes variables:

- *id*, que es la PRIMARY KEY, de tipo varchar, limitada a 10 caracteres
- *iban*, de tipo varchar, limitada a 255 caracteres
- *pan*, de tipo varchar, limitada a 255 caracteres
- *pin*, de tipo varchar, limitada a 4 caracteres
- *cvv*, de tipo varchar, limitada a 3 caracteres
- *expiring_date*, de tipo varchar, limitada a 8 caracteres

La tabla **credit_card** se relaciona con **transaction**, mediante su PRIMARY KEY *id*, que es la FOREIGN KEY de la tabla **transaction** (variable *credit_card_id*). De esta manera se establece una relación de uno a muchos, ya que una sola tarjeta de crédito puede hacer muchas transacciones.

- Ejercicio 2

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.

```
29
30 • UPDATE credit_card
31     SET iban = 'R323456312213576817699999'
32     WHERE id = 'CcU-2938';
33
```

The screenshot shows the MySQL Workbench interface. In the 'Output' pane, there is an 'Action Output' tab. It displays two rows of information:

#	Time	Action	Message	Duration / Fetch
971	11:58:20	ALTER TABLE transaction	- Genero una nueva FK en la tabla transaction para poder relacionar cre...	0.125 sec
972	12:11:26	UPDATE credit_card SET iban = 'R323456312213576817699999' WHERE id = 'CcU-2938'	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec

Se verifica el cambio:

```
34     #Se verifica el cambio:
35 • SELECT id, iban,
36     CASE
37         WHEN id = 'CcU-2938' THEN 'Datos actualizado'
38         ELSE 'No'
39     END AS 'Actualización'
40     FROM credit_card;
41
```

The screenshot shows the MySQL Workbench interface. In the 'Output' pane, there is an 'Action Output' tab. It displays two rows of information:

#	Time	Action	Message	Duration / Fetch
979	12:19:43	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('transactions'.transaction', ...)	0.000 sec
980	12:24:10	SELECT id, iban, CASE WHEN id = 'CcU-2938' THEN 'Datos actualizado' ELSE 'No' END AS 'Actualizaci...	275 row(s) returned	0.000 sec / 0.000

- 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
lato	829.999
longitud	-117.999
amunt	111.11
declined	0

Primero se ha de crear la empresa en las tablas **company** y **credit_card**, ya que en ellas no existe y todas las tablas están unidas y el cambio afecta a datos incluidos en las tres tablas. Primero lo hago en la tabla **company**:

```
40 # Ejercicio 3: En la tabla "transaction" ingresa un nuevo usuario con la siguiente información:  
41 -- Primero se ha de crear la empresa, ya que no existe en la tabla company,  
42 -- y ambas tablas (transaction y company) están unidas por id  
43 • INSERT INTO company (id)  
44 VALUES ('b-9999');
```

Output			
#	Time	Action	Message
973	12:12:48	SELECT id, iban FROM credit_card WHERE id = 'CcU-2938'	1 row(s) returned
974	12:15:08	INSERT INTO company (id) VALUES ('b-9999')	1 row(s) affected

Y después hago lo mismo en la tabla **credit_card**:

```
54 • INSERT INTO credit_card (id)  
55 VALUES ('CcU-9999');  
56  
57 -- Romper relación para poder operar:
```

Output			
#	Time	Action	Message
1005	12:50:07	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('transactions'.transaction', ...)
1006	12:53:35	INSERT INTO credit_card (id) VALUES (CcU-9999)	1 row(s) affected

Ahora se puede añadir la nueva transacción:

```
57 -- Finalmente se puede añadir la nueva transacción  
58 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)  
59 VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, 111.11, 0);  
60
```

Output			
#	Time	Action	Message
1006	12:53:35	INSERT INTO credit_card (id) VALUES (CcU-9999)	1 row(s) affected
1007	12:56:32	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ...	1 row(s) affected

Y visualizamos el nuevo registro:

```
61 -- Y visualizamos el nuevo registro:  
62 • SELECT *  
63 FROM transaction  
64 WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD';  
65
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons for filtering, editing, and exporting data. Below the toolbar is a 'Result Grid' table showing a single row of data from the 'transaction' table. The columns are: id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, and declined. The data for the first row is: id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD', credit_card_id = 'CdU-9999', company_id = 'b-9999', user_id = '9999', lat = '829.999', longitude = '-117.999', timestamp = 'NULL', amount = '111.11', and declined = '0'. Below the table is a 'transaction 5' output window. It contains a table titled 'Action Output' with two rows. Row 1 (red X) is a failed query: '1009 12:58:18 SELECT * FROM transaction WHERE transaction_id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD''. The message column says 'Error Code: 1054. Unknown column 'transaction_id' in 'w''. Row 2 (green checkmark) is a successful query: '1010 12:58:28 SELECT * FROM transaction WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD''. The message column says '1 row(s) returned'.

- Ejercicio 4

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

```
65 • ALTER TABLE credit_card  
66     DROP COLUMN pan;  
67
```

Visualizamos el resultado mediante:

```
68 -- Se visualiza el resultado:  
69 • SHOW COLUMNS FROM credit_card;
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons for filtering, editing, and exporting data. Below the toolbar is a 'Result Grid' table showing the columns of the 'credit_card' table. The columns listed are: id, iban, pin, cvv, and expiring_date. Below the table is a 'Result 4' output window. It contains a table titled 'Action Output' with two rows. Row 1 (red X) is a failed query: '993 12:40:23 ALTER TABLE credit_card DROP COLUMN pan'. The message column says 'Error Code: 1091. Can't'. Row 2 (green checkmark) is a successful query: '994 12:40:30 SHOW COLUMNS FROM credit_card'. The message column says '5 row(s) returned'.

Nivel 2

Ejercicio 1

Elimina de la tabla transacción el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de datos.

```
79 • DELETE FROM transaction  
80 WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';  
81
```

Output			
#	Time	Action	Message
① 1013	13:00:16	DELETE FROM transaction WHERE transaction_id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'	Error Code: 1054. Unknown column 'transaction_id' in 'where clause'
② 1014	13:00:27	DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'	1 row(s) affected

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 promedio de compra.

```
88 • CREATE VIEW VistaMarketing AS  
89   SELECT c.company_name, c.phone, c.country, AVG(t.amount) AS Media_compras  
90   FROM company c  
91   JOIN transaction t  
92   ON c.id = t.company_id  
93   GROUP BY c.company_name, c.phone, c.country  
94   ORDER BY Media_compras DESC;  
95  
96 • SELECT * FROM VistaMarketing;
```

Result Grid			
company_name	phone	country	Media_compras
Eget Ipsum Ltd	03 67 44 56 72	United States	473.075000
Non Magna LLC	06 71 73 13 17	United Kingdom	468.345000
Sed Id Limited	07 28 18 18 13	United States	461.210000
Justo Eu Arcu Ltd	08 42 56 71 52	Italy	443.635000
Eget Tincidunt Duis Institute	05 35 93 32 44	Netherlands	442.520000

VistaMarketing 7 ×

Output

```
1016 13:07:24 CREATE VIEW VistaMarketing AS SELECT c.company_name, c.phone, c.country, AVG(t.amount) AS Media_compras; 0 row(s) affected  
1017 13:07:32 SELECT * FROM VistaMarketing 101 row(s) returned
```

Ejercicio 3

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

```
99  # Ejercicio 3: Filtra la vista VistaMarketing para mostrar sólo las compañías que tienen su país de
100 # residencia en "Germany"
101 • SELECT *
102 FROM VistaMarketing
103 WHERE country = 'Germany';
104
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

company_name	phone	country	Media_compras
Aliquam PC	01 45 73 52 16	Germany	385.265000
Ac Industries	09 34 65 40 60	Germany	289.645000
Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000
Nunc Interdum Incorporated	05 18 15 48 13	Germany	244.025238
Augue Foundation	06 88 43 15 63	Germany	240.800000
Ac Fermentum Incorporated	06 85 56 52 33	Germany	206.465000
Auctor Mauris Corp.	05 62 87 14 41	Germany	184.310000
Convallis In Incorporated	06 66 57 29 50	Germany	156.730000

VistaMarketing 8 x

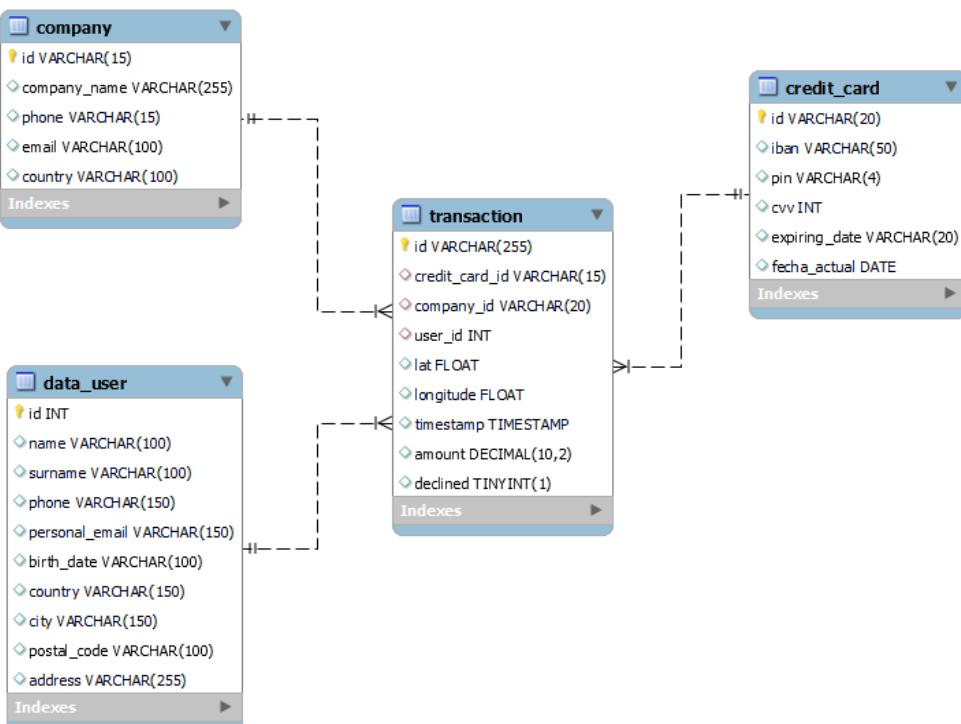
Output:

Action Output	#	Time	Action	Message
1017	13:07:32	SELECT * FROM VistaMarketing		101 row(s) returned
1018	13:10:13	SELECT * FROM VistaMarketing WHERE country = 'Germany'		8 row(s) returned

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 realizó. Te pide que le ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama:



En primer lugar accedo al diagrama actual para comprobar las diferencias en el modelo actual y el deseado. Para ello entro desde la barra de menús en Database → Reverse Engineer y selecciono la base de datos **transactions**. Anoto las diferencias entre las tablas, las claves y las relaciones y procedo a cambiarlas.

Primero creo la tabla **user** y vuelco en ella los datos ejecutando los archivos que vienen dados, y hago los cambios pertinentes.

```
113  -- Cambio el nombre a data_user con rename
114 • ALTER TABLE user RENAME TO data_user;
115
116
117
```

Output			
#	Time	Action	Message
1298	13:24:26	SET foreign_key_checks = 1	0 row(s) affected
1299	13:28:35	ALTER TABLE user RENAME TO data_user	0 row(s) affected

```
116  -- Cambio el nombre del campo email a personal_email
117 • ALTER TABLE data_user
118  CHANGE COLUMN email personal_email VARCHAR(150) NULL;
119
120
```

Output			
#	Time	Action	Message
1300	13:33:02	ALTER TABLE data_user CHANGE COLUMN personal_email email VARCHAR(150) NULL	Error Code: 1054. Unknown column 'personal_email' in 'data_user'
1301	13:33:14	ALTER TABLE data_user CHANGE COLUMN email personal_email VARCHAR(150) NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Finalmente compruebo la tabla **data_user**:

```
120  -- Compruebo con describe:
121 • DESCRIBE data_user;
122
```

Result Grid					
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	

Result 9 ×

Output

Action Output			
#	Time	Action	Message
1301	13:33:14	ALTER TABLE data_user CHANGE COLUMN email personal_email VARCHAR(150) NULL	0 row(s) affected Recor
1302	13:38:01	DESCRIBE data_user	10 row(s) returned

Para poder continuar con los cambios, he de crear en la tabla **data_user** el nuevo usuario (el del ejercicio 1.3) para poder vincular más adelante las tablas **transaction** y **data_user**:

```
135 -- Creo en la tabla data_user el nuevo usuario (del ej. 1.3):
136 • INSERT INTO data_user (id)
137   VALUES (9999);
138
```

Output:

Action Output			
#	Time	Action	Message
✓ 1310	16:24:53	describe data_user	10 row(s) returned
✓ 1311	16:25:37	INSERT INTO data_user (id) VALUES (9999)	1 row(s) affected

Creo ahora la relación entre la tabla **transaction** y la tabla **data_user**, con su Foreign Key:

```
120 -- Creo la relación entre data_user y transaction con la FK correspondiente:
121 • ALTER TABLE transaction
122   ADD CONSTRAINT FK_transaction_data_user
123     FOREIGN KEY (user_id) REFERENCES data_user(id);
124
```

Output:

Action Output			
#	Time	Action	Message
✓ 2	17:17:12	USE transactions	0 row(s) affected
✓ 3	17:17:39	ALTER TABLE transaction ADD CONSTRAINT FK_transaction_data_user FOREIGN KEY (user_id) REFERE...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

Después voy a la tabla **company** y borro el campo **website** mediante:

```
123 -- Cambios en tabla company:
124 • ALTER TABLE company
125   DROP COLUMN website;
126
127 • DESCRIBE company;
```

Result Grid:

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 10 x

Output:

Action Output			
#	Time	Action	Message
✗ 1306	13:46:25	DESCRIBE company -- Cambios en la tabla credit_card: -- Libero las relaciones entre transaction y credit...	Error Code: 1064. You have
✓ 1307	13:46:40	DESCRIBE company	5 row(s) returned

En cuanto a la tabla **credit_card** requiere de algunos cambios en la configuración de sus variables, pero antes he de romper las relaciones con la tabla **transaction**, para poder operar:

```
--  
134 • ALTER TABLE transaction DROP FOREIGN KEY FK_transaction_credit_card;
```

```
135
```

Output		
#	Time	Action
1307	13:46:40	DESCRIBE company
1308	16:20:36	ALTER TABLE transaction DROP FOREIGN KEY FK_transaction_credit_card

Comienzo con los cambios en la configuración de las variables. Primero cambio la longitud de la variable *id*:

```
145 -- Cambio de la longitud de la variable id:
```

```
146 • ALTER TABLE credit_card MODIFY COLUMN id VARCHAR(20);
```

Output		
#	Time	Action
1312	16:27:19	ALTER TABLE transaction ADD CONSTRAINT FK_transaction_user_id FOREIGN KEY (user_id) REFERENCES user(id)
1313	16:40:35	ALTER TABLE credit_card MODIFY COLUMN id VARCHAR(20)

Después cambio la variable *iban*, reduciendo su longitud a 50 caracteres:

```
148 -- Reduzco la variable iban a 50 caracteres:
```

```
149 • ALTER TABLE credit_card MODIFY COLUMN iban VARCHAR(50);
```

Output		
#	Time	Action
1313	16:40:35	ALTER TABLE credit_card MODIFY COLUMN id VARCHAR(20)
1314	16:42:13	ALTER TABLE credit_card MODIFY COLUMN iban VARCHAR(50)

La variable *cvv*, la convierto a numero entero (INT):

```
151 -- Variable cvv a INT:
```

```
152 • ALTER TABLE credit_card MODIFY COLUMN cvv INT;
```

Output		
#	Time	Action
1314	16:42:13	ALTER TABLE credit_card MODIFY COLUMN iban VARCHAR(50)
1315	16:44:25	ALTER TABLE credit_card MODIFY COLUMN cvv INT

La variable `expiring_date` pasa a tener una longitud de 20 caracteres mediante el comando:

```
154 -- Cambio la longitud de la variable expiring_date a 20 caracteres:  
155 • ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR(20);
```

The screenshot shows the MySQL Workbench interface with the 'Output' tab selected. It displays two successful operations: 'ALTER TABLE credit_card MODIFY COLUMN cvv INT' at 16:44:25 and 'ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR(20)' at 16:46:51. The 'Message' pane indicates 276 rows affected and 0 warnings for the first operation, and 0 rows affected and 0 warnings for the second.

Y después añado el nuevo campo `fecha_actual`:

```
157 -- Crear nuevo campo fecha_actual:  
158 • ALTER TABLE credit_card ADD fecha_actual DATE NULL;  
159  
160
```

The screenshot shows the MySQL Workbench interface with the 'Output' tab selected. It displays two successful operations: 'ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR(20)' at 16:46:51 and 'ALTER TABLE credit_card ADD fecha_actual DATE NULL' at 16:50:27. The 'Message' pane indicates 0 rows affected and 0 warnings for both operations.

Finalmente vuelvo a generar la relación entre las tablas `credit_card` y `transactions`, con su Foreign Key correspondiente:

```
139 -- Creo la relación entre credit_card y transaction con la FK correspondiente:  
140 • ALTER TABLE transaction  
141 ADD CONSTRAINT FK_transaction_credit_card  
142 FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);  
143
```

The screenshot shows the MySQL Workbench interface with the 'Output' tab selected. It displays two successful operations: 'ALTER TABLE transaction ADD CONSTRAINT FK_transaction_user FOREIGN KEY (user_id) REFERENCES user(id)' at 17:05:48 and 'ALTER TABLE transaction ADD CONSTRAINT FK_transaction_credit_card FOREIGN KEY (credit_card_id) REFERENCES credit_card(id)' at 17:09:29. The 'Message' pane indicates 587 rows affected and 0 warnings for both operations.

Para acabar se ha de comprobar que los campos quedan como se solicita mediante:

```
160 -- Compruebo:  
161 • DESCRIBE credit_card;
```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. It displays the structure of the `credit_card` table with the following columns:

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI	NULL	
iban	varchar(50)	YES		NULL	
pin	varchar(4)	YES		NULL	
cvv	int	YES		NULL	
expiring_date	varchar(20)	YES		NULL	
fecha_actual	date	YES		NULL	

Below the result grid, the 'Output' tab shows the execution of 'DESCRIBE COMPANY' and 'DESCRIBE TRANSACTION' commands, with 5 and 9 rows returned respectively.

Ejercicio 2

La empresa también te solicita 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.

Asegúrate de incluir información relevante de ambas tablas y utiliza alias para cambiar de nombre columnas según sea necesario.

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

```
172 • CREATE VIEW InformeTecnico AS
173   SELECT t.id AS ID_Transaccion, du.name AS nombre_usuario, du.surname AS apellido_usuario,
174         cc.iban AS num_iban, c.company_name AS nombre_empresa
175   FROM transaction t
176   JOIN data_user du
177   ON t.user_id = du.id
178   JOIN credit_card cc
179   ON t.credit_card_id = cc.id
180   JOIN company c
181   ON c.id = t.company_id
182   ORDER BY ID_Transaccion DESC;
```

Output		
#	Time	Action
6	17:25:17	SELECT * FROM transactions.transaction
7	17:25:43	CREATE VIEW InformeTecnico AS SELECT t.id AS ID_Transaccion, du.name AS nombre_usuario, du.suma...

Visualizamos la nueva vista InformeTecnico:

```
184 • SELECT * FROM InformeTecnico;
```

Result Grid				
ID_Transaccion	nombre_usuario	apellido_usuario	num_iban	nombre_empresa
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9C8CCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7ABA6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCE2BA9A-271D-2BDC-9E49-8DD92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated
FBD7E006-B6A6-F5BC-0CA9-EA48B760100C	Hedwig	Gilbert	MU4132333444534342541344788855	Mauris Id Inc.
FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF58868200575771634583813	Arcu LLP
FAAAD3FC-1A17-E141-43D3-359A5BA7CB38	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
FR43NC0A-C85-7444-1R4F-5966289FRA8R	Iasner	Iandriv	VG1468087984174645729577	Ilit Semper Foundation

Output		
#	Time	Action
7	17:25:43	CREATE VIEW InformeTecnico AS SELECT t.id AS ID_Transaccion, du.name AS nombre_usuario, du.suma...
8	17:26:27	SELECT * FROM InformeTecnico