

# 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.

```
1 • USE transactions;
2 • CREATE TABLE IF NOT EXISTS credit_card (
3     id VARCHAR(10) PRIMARY KEY,
4     iban VARCHAR(35),
5     pan VARCHAR(35),
6     pin CHAR(4),
7     cvv CHAR(3),
8     expiring_date VARCHAR(8));
9

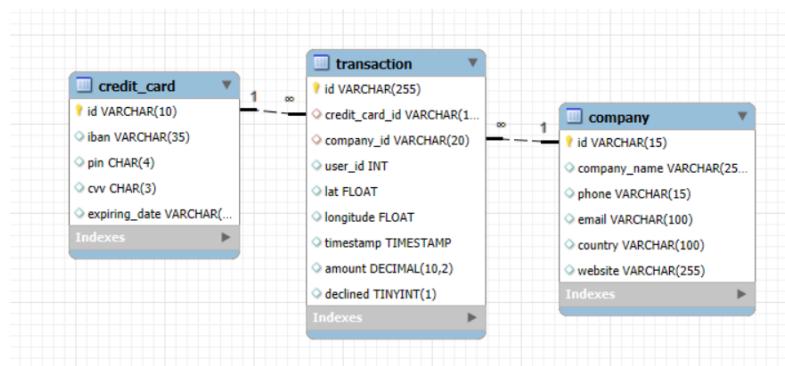
Output
Action Output
# Time Action Message
1 14:08:40 CREATE TABLE IF NOT EXISTS credit_card ( id VARCHAR(10) PRIMARY KEY, iban VARCHAR(35), pa... 0 rows affected

10 • ALTER TABLE transaction
11     ADD CONSTRAINT fk_transaction_credit_card_id
12         FOREIGN KEY (credit_card_id)
13             REFERENCES credit_card(id);
14
15

Output
Action Output
# Time Action Message
1 14:14:15 ALTER TABLE transaction ADD CONSTRAINT fk_transaction_credit_card_id FOREIGN KEY (credit_card_id)... 100000 rows affected Records: 100000 Duplicates: 0 Warnings: 0
```

**Nota 1:** Se creó la tabla 'credit\_card' con las columnas:

id como Primary Key, identifica de forma única cada tarjeta, iban, pan, pin, cvv, expiring\_date. Se añadió una Foreign Key en la tabla 'transaction' relacionando credit\_card\_id con id de la nueva tabla, lo que asegura la relación entre las transacciones y las tarjetas de crédito.



**Nota 2:** El diagrama muestra un modelo estrella donde la tabla 'transaction' funciona como tabla de hechos, ya que guarda las transacciones, las tablas 'credit\_card' y 'company' son tablas de dimensiones, porque aportan detalles adicionales a cada transacción. Así, cada transacción se relaciona con una tarjeta y con una empresa.

## Ejercicio 2

El departamento de Recursos Humanos ha identificado un error en el número de cuenta asociado a su tarjeta de crédito con ID CcU-2938. La información que debe mostrarse para este registro es: TR323456312213576817699999. Recuerda mostrar que el cambio se realizó.

The screenshot shows the MySQL Workbench interface. At the top, a code editor displays a script with two main parts: an UPDATE statement to change the iban for id 'CcU-2938' to 'TR323456312213576817699999', and a SELECT statement to retrieve the iban for id 'CcU-2938'. Below the code editor is a 'Result Grid' pane showing the output of the SELECT query, which is 'TR323456312213576817699999'. At the bottom, an 'Output' pane shows the execution log with three entries: the update at 19:17:07, the select at 19:17:12, and a message indicating 1 row(s) affected.

```
16  # Ejercicio 2
17  UPDATE credit_card SET iban = 'TR323456312213576817699999'
18  WHERE id = 'CcU-2938';
19
20 •  SELECT iban
21   FROM credit_card
22  WHERE id = 'CcU-2938';|
```

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

iban
TR323456312213576817699999

credit\_card 32 x

Output

Action Output
# Time Action
1 19:17:07 UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE id = 'CcU-2938'
2 19:17:12 SELECT iban FROM credit_card WHERE id = 'CcU-2938'

Message  
0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0  
1 row(s) returned

## Ejercicio 3

En la tabla "transaction" ingresa una nueva transacción con la siguiente información:

Id 108B1D1D-5B23-A76C-55EF-C568E49A99DD, credit\_card\_id CcU-9999, company\_id b-9999, user\_id 9999, late 829.999, longitud -117.999, amount 111.11, declined 0

The screenshot shows the MySQL Workbench interface. At the top, a code editor displays a script with three INSERT statements: one for the 'company' table with id 'b-9999', one for the 'credit\_card' table with id 'CcU-9999', and one for the 'transaction' table with all specified fields. Below the code editor is an 'Output' pane showing the execution log with three entries corresponding to the insertions.

```
20  # Ejercicio 3
21 •  INSERT INTO company (id) VALUES ('b-9999');
22 •  INSERT INTO credit_card (id) VALUES ('CcU-9999');
23 •  INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
24  VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, 111.11 , 0);|
```

Output

Action Output
# Time Action
1 14:29:51 INSERT INTO company (id) VALUES (b-9999)
2 14:29:55 INSERT INTO credit_card (id) VALUES (CcU-9999)
3 14:30:00 INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES (10...

Message  
1 row(s) affected  
1 row(s) affected  
1 row(s) affected

**Nota:** Antes de ingresar la nueva transacción se creó el 'company\_id' en la tabla 'company' y el 'credit\_card\_id' en la tabla 'credit\_card' ya que no existían y la transacción depende de estas Foreign Key. Como la tabla 'transaction' contiene datos tanto de 'company' como de 'credit\_card', si estos registros no estaban en las tablas correspondientes no se podía añadir la transacción. Finalmente, se inserta la transacción con todos sus datos (id, credit\_card\_id, company\_id, user\_id, lat, longitude, amount, declined).

## Ejercicio 4

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

```
27  # Ejercicio 4
28 • ALTER TABLE credit_card DROP COLUMN pan;
29 • SHOW COLUMNS from credit_card;
```

The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window with the following SQL statements:

```
27  # Ejercicio 4
28 • ALTER TABLE credit_card DROP COLUMN pan;
29 • SHOW COLUMNS from credit_card;
```

Below the code editor is a 'Result Grid' table showing the columns of the 'credit\_card' table. The table has the following structure:

Field	Type	Null	Key	Default	Extra
id	varchar(10)	NO	PRI	NULL	
iban	varchar(35)	YES		NULL	
pin	char(4)	YES		NULL	
cvv	char(3)	YES		NULL	
expiring_date	varchar(8)	YES		NULL	

At the bottom of the interface, there is a 'Result 2' tab which displays the output of the 'SHOW COLUMNS' and 'ALTER TABLE' commands. The output table shows the following log entries:

#	Time	Action	Message
1	14:19:36	SHOW COLUMNS from credit_card	6 row(s) returned
2	14:19:44	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3	14:19:57	SHOW COLUMNS from credit_card	5 row(s) returned

## Nivel 2

### Ejercicio 1

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

```
30  # NIVEL 2
31  # Ejercicio 1
32 • DELETE FROM transaction
33 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
```

The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window with the following SQL statement:

```
30  # NIVEL 2
31  # Ejercicio 1
32 • DELETE FROM transaction
33 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
```

Below the code editor is an 'Output' table showing the log entry for the DELETE command. The table shows the following log entry:

#	Time	Action	Message
1	14:31:47	DELETE FROM transaction WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD'	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.

```
35  # Ejercicio 2
36 • CREATE VIEW `VistaMarketing`AS
37   SELECT c.company_name, c.phone, c.country, ROUND(AVG(t.amount),2) media_compras
38   FROM transaction t
39   JOIN company c ON t.company_id = c.id
40   GROUP BY c.id, c.company_name
41   ORDER BY media_compras DESC;
42
43 • SELECT * FROM transactions.vistamarketing;
--
```

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

company_name	phone	country	media_compras
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.87
Pretium Neque Corp.	07 77 48 55 28	Australia	276.16
Urna Convallis Associates	06 01 24 77 04	United States	274.24
At Associates	09 56 61 10 65	New Zealand	272.21
Metus Vitae Associates	08 25 44 40 66	Australia	270.08
Aliquet Diam Limited	02 76 61 47 46	United States	269.60
Nec Luctus LLC	02 14 71 75 73	Norway	268.60
Neque Tellus Incorporated	04 43 18 34 19	Ireland	267.85
Tortor Nunc Commodo Company	05 35 92 77 16	United States	267.84
Cras Consulting	07 50 10 85 63	Belgium	267.44

vistamarketing 7 ×

Output

Action Output

#	Time	Action	Message
1	15:25:57	CREATE VIEW `VistaMarketing`AS SELECT c.company_name, c.phone, c.country, ROUND(AVG(t.amount),2) ...	0 row(s) affected
2	15:26:00	SELECT * FROM transactions.vistamarketing	101 row(s) returned

## Ejercicio 3

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

```
45  #Ejercicio 3
46 • SELECT * FROM transactions.vistamarketing
47 WHERE country = 'Germany';
```

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

company_name	phone	country	media_compras
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.87
Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.32
Convallis In Incorporated	06 66 57 29 50	Germany	257.75
Ac Industries	09 34 65 40 60	Germany	255.15
Rutrum Non Inc.	02 66 31 61 09	Germany	255.14
Auctor Mauris Corp.	05 62 87 14 41	Germany	254.77
Augue Foundation	06 88 43 15 63	Germany	253.51
Aliquam PC	01 45 73 52 16	Germany	253.14

vistamarketing 9 ×

Output

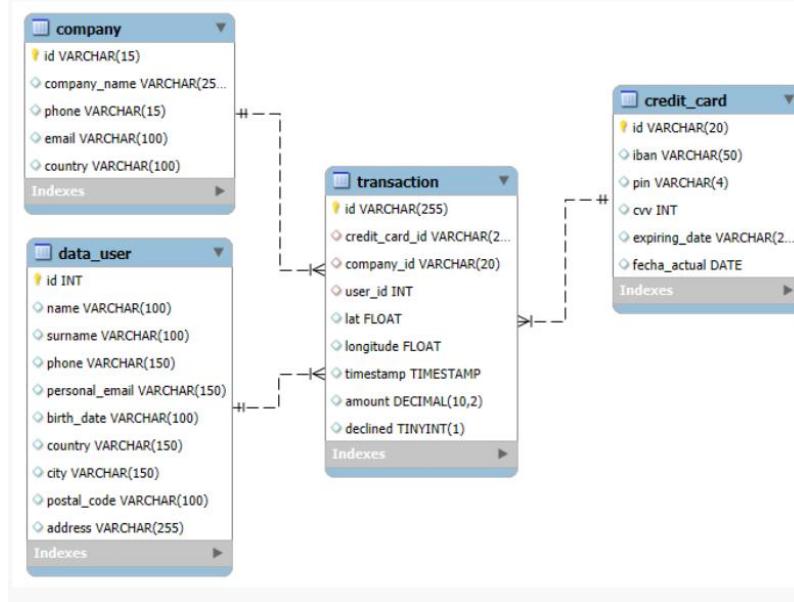
Action Output

#	Time	Action	Message
1	15:30:19	SELECT * FROM transactions.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:



1. Se crea la tabla ‘user’ con sus columnas, se establece id como la PRIMARY KEY.

```
55 • ⚡ CREATE TABLE IF NOT EXISTS user (
56     id CHAR(10) PRIMARY KEY,
57     name VARCHAR(100),
58     surname VARCHAR(100),
59     phone VARCHAR(150),
60     email VARCHAR(150),
61     birth_date VARCHAR(100),
62     country VARCHAR(150),
63     city VARCHAR(150),
64     postal_code VARCHAR(100),
65     address VARCHAR(255)
66 );
67
```

Output ::::::::::::

Action Output	#	Time	Action	Message
	1	16:17:05	CREATE TABLE IF NOT EXISTS user ( id CHAR(10) 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) );	0 row(s) affected

2. se añade toda la información de la tabla ‘user’ mediante INSERT INTO.
3. Antes de crear la Foreign Key entre user\_id de la tabla ‘transaction’ y id de la tabla ‘user’, se eliminaron registros en ‘transaction’ que contenían valores de user\_id que no existían en la tabla ‘user’. En este caso, se identificaron por el company\_id.

```
67
68 •  DELETE FROM transaction
69      WHERE company_id = 'b-9999';
70
```

Output :

Action Output

#	Time	Action
---	------	--------

1	16:18:58	DELETE FROM transaction WHERE company_id = 'b-9999'
---	----------	---

Message

1 row(s) affected

4. Se modificó el tipo de dato de id de la tabla ‘user’ para que coincidiera con user\_id de la tabla ‘transaction’ (de CHAR(10) a INT), ya que los tipos deben coincidir para establecer una relación de Foreign Key.

```
71 •  ALTER TABLE user
72      MODIFY id INT;
73
```

Output :

Action Output

#	Time	Action
---	------	--------

1	16:20:46	ALTER TABLE user MODIFY id INT
---	----------	--------------------------------

Message

5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0

5. Se crea la FOREIGN KEY

```
74 •  ALTER TABLE transaction
75      ADD CONSTRAINT fk_transaction_user
76          FOREIGN KEY (user_id)
77              REFERENCES user(id);
78
```

Output :

Action Output

#	Time	Action
---	------	--------

1	16:21:59	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_user FOREIGN KEY (user_id) REFERENCES user(id)	99999 row(s) affected Records: 99999 Duplicates: 0 Warnings: 0
---	----------	--	--

## Ejercicio 2

La empresa también le 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.

Asegúrese de incluir información relevante de las tablas que conocerá y utilice 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.

```
70  # Ejercicio 2
71 • CREATE VIEW `InformeTecnico` AS
72   SELECT t.id AS Transaction_ID, u.name AS User_Name, u.surname AS User_Surname,
73          cc.iban AS Credit_card_iban, c.company_name AS Company_Name
74   FROM transaction t
75   JOIN user u ON t.user_id = u.id
76   JOIN credit_card cc ON t.credit_card_id = cc.id
77   JOIN company c ON t.company_id = c.id
78   ORDER BY Transaction_ID DESC;
79
80 • SELECT * FROM transactions.informetecnico;
```

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

Transaction_ID	User_Name	User_Surname	Credit_card_iban	Company_Name
FFFD31D6-9495-47CE-854A-7DB8E1CC274B	Bmrgli	Tprvrmrc	XX794814451211289182490922	Turpis Company
FFFCF78D-ECF0-4985-A2D0-B2A7B75998FC	Dfled	Vlqjdl	XX636251701647892036676034	Amet Nulla Donec Corporation
FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	Securp	Faofqfy	XX162677143304223631437567	Nunc Interdum Incorporated
FFFB270D-F53A-4D5D-9666-E5307C53CC84	Gzzipa	Urzjulh	XX395114267082019952567052	Viverra Donec Foundation
FFFB9E3CE-234E-408C-A8EF-F9CAD577224A	Yshimq	Zpsjleed	XX8849462156537570367941	Convallis In Incorporated
FFF9E178-6CD2-4DF9-99B0-49AE06880961	Jevpx	Xwcnzwmm	XX321405515711654384711481	Mus Aenean Eget Foundation
FFFB867C9-17B5-4B1F-ADF9-F8023AA449E	Fqngd	Lvhfqxi	XX278446342932680979729426	Cras Vehicula Aliquet Industries
FFFB7A42D9-18F6-44DD-A73C-4B90A44CRP7K	Nirvaa	Frennici	YY4n5nn077757755nn0n0n0n0n0n0n0	Placerat IIP

informetecnico 27 x

Output

Action Output

#	Time	Action	Message
7	18:20:28	CREATE VIEW `InformeTecnico` AS SELECT t.id AS Transaction_ID, u.name AS User_Name, u.surname AS ...	0 row(s) affected
8	18:20:35	SELECT * FROM transactions.informetecnico	99999 row(s) returned