

## SPRINT 3 (MYSQL) MARINE FERNANDEZ

### NIVEL 1

### EJECICIO 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.

```
7 • USE transactions;
8 • CREATE TABLE IF NOT EXISTS credit_card (
9     id VARCHAR(8) PRIMARY KEY,
10     iban VARCHAR(34),
11     pan VARCHAR(19),
12     pin CHAR(4),
13     cvv CHAR(3),
14     expiring_date DATE
15 );
16
17 #Modificamos el tipo del campo expiring_date
18 • ALTER TABLE credit_card
19     MODIFY COLUMN expiring_date VARCHAR (10);
20
21 #Añadimos la relación entre la tabla transacción y credit_card
22 • ALTER TABLE transaction
23     ADD CONSTRAINT Fk CreditCard
```

Output

#	Time	Action	Message	Duration / Fetch
1	15:19:38	CREATE TABLE IF NOT EXISTS credit_card (id VARCHAR(8) PRIMARY KEY, iban VARCHAR(34), pan ...	0 row(s) affected	0.078 sec
2	15:19:41	ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR (10)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.109 sec

```
4981 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9561', 'XX842287760526866727040752', '5247195334352545', '4956', '326', '12/25/27');
4982 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9562', 'XX639604292186089282464578', '6116239298415107', '8566', '201', '04/25/29');
4983 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9563', 'XX392846417866192835897237', '2373131904258200', '5044', '675', '04/25/26');
4984 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9564', 'XX641179722562813208059896', '3101008088023097', '1822', '746', '03/26/29');
4985 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9565', 'XX642860021926055745163758', '5522454509376387', '3053', '879', '04/29/27');
4986 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9566', 'XX327046332926811278055630', '6161849983263025', '6889', '278', '06/27/25');
4987 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9567', 'XX856381783872538603280064', '0745947204130257', '4955', '190', '03/28/25');
4988 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9568', 'XX386536547822381431747659', '2296680186377148', '8326', '522', '05/27/26');
4989 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9569', 'XX305272728964047897361563', '0734570924276022', '6083', '424', '06/27/25');
4990 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9570', 'XX633942010258286181516573', '7233675965634598', '5619', '285', '03/27/28');
4991 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9571', 'XX484915382437141996579468', '1265066247957405', '6585', '104', '11/27/28');
4992 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9572', 'XX405448186269730388404508', '8296622848225053', '1995', '220', '07/27/29');
4993 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9573', 'XX965338931053088901924906', '9099040179637171', '2277', '224', '01/25/28');
4994 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9574', 'XX662761758361432686520775', '3171178331318656', '5084', '706', '11/28/27');
4995 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9575', 'XX499881216073626571196473', '8086033007377786', '7256', '660', '10/25/29');
4996 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9576', 'XX52971093058771264172007', '4554226978064107', '4884', '724', '06/28/29');
4997 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9577', 'XX158914078594800633147086', '8110437271856107', '5864', '772', '10/29/26');
4998 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9578', 'XX991539646456110567870254', '8999808823061411', '2872', '772', '07/29/27');
4999 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9579', 'XX296393091587170202131236', '9690060468678689', '8379', '134', '12/25/25');
5000 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9580', 'XX781258889851958086677358', '5541182364498931', '9273', '737', '03/27/29');
5001 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9581', 'XX915670516405388124398147', '2624305470167630', '4336', '926', '06/29/25');
5002
```

Output

#	Time	Action	Message	Duration / Fetch
5003	15:28:39	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9579', 'XX29639309158...	1 row(s) affected	0.016 sec
5004	15:28:39	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9580', 'XX78125888985...	1 row(s) affected	0.000 sec
5005	15:28:39	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9581', 'XX91567051640...	1 row(s) affected	0.000 sec

```

26 #Añadimos la relación entre la tabla transacción y credit_card
27 • ALTER TABLE transaction
28   ADD CONSTRAINT Fk_CreditCard
29   FOREIGN KEY (credit_card_id) REFERENCES credit_card (id);
30
31 #verificación de la correcta creación de la tabla
32 • DESCRIBE credit_card;
33
34
35

```

Field	Type	Null	Key	Default	Extra
id	varchar(8)	NO	PRI	NULL	
iban	varchar(34)	YES		NULL	
pan	varchar(19)	YES		NULL	
pin	char(4)	YES		NULL	
cvv	char(3)	YES		NULL	

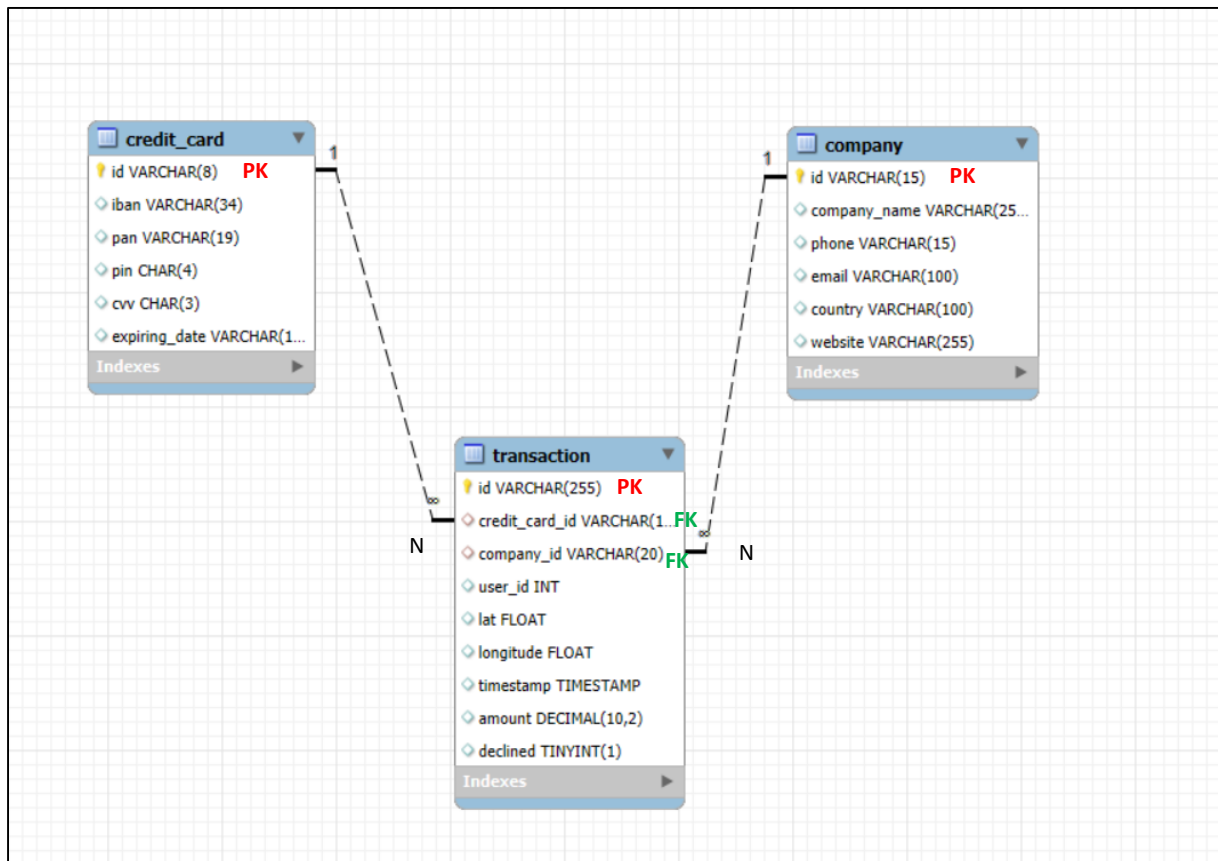
Result 2 x

#	Time	Action	Message	Duration / Fetch
5002	15:53:51	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9581', 'XX9156705164...	1 row(s) affected	0.000 sec
5003	15:54:17	ALTER TABLE transaction ADD CONSTRAINT Fk_CreditCard FOREIGN KEY (credit_card_id) REFER...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	3.360 sec
5004	15:54:41	DESCRIBE credit_card	6 row(s) returned	0.000 sec / 0.000 sec

Primero establecimos que estamos trabajando con la base de datos “transactions” y creamos la tabla “credit\_card”. Al poner el tipo de dato “DATE” nos da error al cargar el archivo con la información por lo cual tenemos que corregir el tipo de dato de DATE a VARCHAR.

Posteriormente establecemos la relación entre la tabla “credit\_car” y “transaction” configurando la variable “credit\_card\_id” de la tabla transaction como **FOREIGN KEY** en referencia a la **PRIMARY KEY** “id” de la tabla “credit\_car”. Escogí establecer esa FOREIGN KEY como CONSTRAINT es decir que no se puede tener un registro con un credit\_card\_id que no exista en la tabla credit\_card y eso asegura la integridad de la base de datos.

## Diagrama:



## Explicación del diagrama:

Esa base de datos se compone de 3 tablas : transaction, company y credit\_card.

La tabla company tiene información acerca de las empresas de ventas de productos en línea y la tabla transaction aparece la información acerca de las ventas por parte de esas empresas. En cuanto a la tabla credit\_card, almacena información sobre las tarjetas de crédito usadas para las ventas realizadas.

Detalle de cada tabla:

### Company

- Id : identificador de la empresa (es único)
- Company\_name: nombre de la empresa
- Phone: número de telefono de la empresa
- Email: correo electrónico de la empresa
- Country: país donde se encuentra la empresa
- Website: página web de la empresa

### Transaction

- Id: identificador de la transacción (es único?)
- Credit\_car\_id: identificador de la tarjeta usada para la venta
- Company\_id: identificador de la empresa que concluyó la venta
- User\_id: identificador del usuario que realizó la transacción

- Lat/longitude: coordenadas geográficas de la transacción
- Timestamp: fecha de la transacción
- Amount: importe de la venta
- Declined: indica si la transacción fue cancelada o no (0 significa que se concluyó, 1 significa que fue cancelada)

### **Credit\_card**

- Id : identificador de la tarjeta bancaria
- Iban : número de cuenta asociado a la tarjeta bancaria
- Pin : contraseña de la tarjeta bancaria
- Pan : número completo de la tarjeta bancaria, lo cual aparece en la tarjeta
- Cvv : número de verificación que se usa para realizar compras, asegura que el comprador tenga la tarjeta física.
- Expiring\_date : fecha de caducidad de la tarjeta bancaria

### Relación entre las tablas:

La tabla company tiene como **Primary Key** la columna "Id".

La tabla transaction tiene como **Primary Key** la columna "Id" y se relaciona con la tabla company mediante la columna "company\_id" que viene a ser **Foreign Key**.

Entre las tablas company y transaction tenemos una relación de **1 a N** es decir que el "Id" de una empresa puede estar relacionado con varias transacciones (mediante la columna "company\_id" de la tabla transaction) pero una transacción solo puede estar vinculada a una única empresa.

La tabla credit\_card tiene como **Primary Key** la columna "Id" y se relaciona con la tabla transaction mediante la columna "credit\_card\_id" donde es **Foreign Key**.

Entre las tablas credit\_card y transaction tenemos una relación de **1 a N** es decir que el "Id" de una tarjeta bancaria puede estar relacionado con varias transacciones (mediante la columna "credit\_card\_id" de la tabla transaction) pero una transacción solo puede estar vinculada con una tarjeta bancaria.

## 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 a database management interface with the following components:

- SQL Editor:** Contains the following SQL code:

```
41 • UPDATE credit_card SET iban = "TR323456312213576817699999" WHERE id = "CcU-2938";
42
43 #verificacion
44 • SELECT iban
45 FROM credit_card
46 WHERE id = "CcU-2938";
47
```
- Result Grid:** Displays the result of the SELECT query as a table with one row:

iban
TR323456312213576817699999
- Output Panel:** Shows the execution log with two entries:

#	Time	Action	Message	Duration /
1	16:02:19	UPDATE credit_card SET iban = "TR323456312213576817699999" WHERE id = "CcU-2938"	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
2	16:02:22	SELECT iban FROM credit_card WHERE id = "CcU-2938"	1 row(s) returned	0.000 sec

## EJERCICIO3

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
lat	829.999
longitude	-117.999
amount	111.11
declined	0

```

60 #Insertamos el id de credit card proporcionado en la tabla credit_card
61 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)
62   VALUES ("CcU-9999", NULL, NULL, NULL, NULL, NULL);
63
64 #Insertamos el id de la empresa proporcionado en la tabla company
65 • INSERT INTO company (id, company_name, phone, email, country, website)
66   VALUES ("b-9999", NULL, NULL, NULL, NULL, NULL);
67
68 #Finalmente podemos ingresar los datos relativos a la transacción
69 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined)
70   VALUES ("10881D1D-5B23-A76C-55EF-C568E49A99DD", "CcU-9999", "b-9999", "9999", "829.999", "-117.999", NULL, "111.11", "0");
71
72 #verificacion

```

Output

#	Time	Action	Message	Duration / Fe
1	16:04:21	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ("CcU-9999", NULL, NULL, NULL, NULL, NULL);	1 row(s) affected	0.016 sec
2	16:04:23	INSERT INTO company (id, company_name, phone, email, country, website) VALUES ("b-9999", NULL, NULL, NULL, NULL, NULL);	1 row(s) affected	0.016 sec
3	16:04:30	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ("10881D1D-5B23-A76C-55EF-C568E49A99DD", "CcU-9999", "b-9999", "9999", "829.999", "-117.999", NULL, "111.11", "0");	1 row(s) affected	0.015 sec

```

72 #verificacion
73 • SELECT *
74   FROM transaction
75   WHERE id = "10881D1D-5B23-A76C-55EF-C568E49A99DD";
76
77 #FINCICLO 4

```

Result Grid

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
10881D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	NULL	111.11	0

transaction 4 x

Output

#	Time	Action	Message	Duration / Fe
1	16:05:08	SELECT * FROM transaction WHERE id = "10881D1D-5B23-A76C-55EF-C568E49A99DD"	1 row(s) returned	0.000 sec /

Al intentar introducir los datos nos encontramos con el mensaje error 1452 ya que la Foreign Key company\_id no existe en la tabla company (donde se relaciona con "Id" y es Primary Key). En ese caso hemos insertado los datos de la company b-9999 en la tabla company rellenando los datos que faltan con NULL.

Hemos realizado una operación similar con la tabla credit\_card, insertando una fila con los datos de la credit\_card CcU-9999.

Lo correcto hubiera sido evitar usar "NULL" y reemplazarlo por un texto como "DESCONOCIDO" por si quisiéramos filtrar por un dato en concreto a futuro, sin embargo, como tenemos diferentes tipos de datos (fechas, series de números) ha sido imposible hacerlo,

Por otro lado, la tabla transaction se compone de 8 columnas cuando los datos de la transacción por añadir solo se componen de 7 elementos, el ante penúltimo, timestamp no está proporcionado. Por lo tanto lo hemos añadido como NULL ya que desconocemos la fecha y hora de la dicha transacción.

## EJERCICIO 4

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

```
77 #EJERCICIO 4
78 #Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla credit_card. Recuerda mostrar el cambio realizado.
79 • ALTER TABLE credit_card
80 DROP COLUMN pan;
81
82 #verificacion
83 • SELECT*
84 FROM credit_card;
```

Result Grid

	id	iban	pin	cvv	expiring_date
▶	CcS-4857	XX4857591835292505850771	1819	467	09/27/25
	CcS-4858	XX8581768137002436094025	3964	817	12/28/28
	CcS-4859	XX7826930491423553609370	4983	277	11/26/26
	CcS-4860	XX5559590368835304645299	6876	661	07/27/27
	CcS-4861	XX2035182877195191627307	5710	398	04/25/26

credit card 5 x

Output

#	Time	Action	Message	Duration / Fe
✓ 1	16:06:27	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.062 sec
✓ 2	16:06:31	SELECT* FROM credit_card	5001 row(s) returned	0.000 sec / 0

## NIVEL 2

### EJERCICIO 1

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

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.

```
90 #Antes de eliminar, verificamos que existe antes ese registro
91 • SELECT*
92 FROM transaction
93 WHERE id="000447FE-B650-4DCF-85DE-C7ED0EE1CAAD";
94
95 • DELETE FROM transaction
96 WHERE id ="000447FE-B650-4DCF-85DE-C7ED0EE1CAAD";
97
```

Output

#	Time	Action	Message	Duration / Fetch
1	16:07:19	SELECT* FROM transaction WHERE id="000447FE-B650-4DCF-85DE-C7ED0EE1CAAD"	1 row(s) returned	0.000 sec / 0.00
2	16:07:27	DELETE FROM transaction WHERE id ="000447FE-B650-4DCF-85DE-C7ED0EE1CAAD"	1 row(s) affected	0.031 sec

```
98 #verificacion:
99 • SELECT*
100 FROM transaction
101 WHERE id="000447FE-B650-4DCF-85DE-C7ED0EE1CAAD";
102
```

Result Grid

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined

transaction 8 x

Output

#	Time	Action	Message	Duration / Fe
1	16:08:14	SELECT* FROM transaction WHERE id="000447FE-B650-4DCF-85DE-C7ED0EE1CAAD"	0 row(s) returned	0.000 sec / 0

Antes de eliminar el registro verificamos que existe en la base de datos, posteriormente lo eliminamos y averiguamos que haya sido efectivo.



## 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.

```
110 • CREATE VIEW VistaMarketing AS
111 SELECT
112     c.company_name,
113     c.phone,
114     c.country,
115     ROUND(AVG(t.amount),2) AS Media_compras
116 FROM company AS c
117 JOIN transaction AS t
118 ON c.id=t.company_id
119 WHERE declined = 0
120 GROUP BY c.company_name, c.phone, c.country
121 ;
122
123 • SELECT *
124 FROM VistaMarketing
125 ORDER BY Media_compras DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: [12](#)

	company_name	phone	country	Media_compras
▶	Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.91
	Pretium Neque Corp.	07 77 48 55 28	Australia	275.58
	Urna Convalis Associates	06 01 24 77 04	United States	273.57
	At Associates	09 56 61 10 65	New Zealand	272.74
	Metus Vitae Associates	08 25 44 40 66	Australia	270.05

VistaMarketing 9 x

Output

#	Time	Action	Message	Duration /
✓ 1	16:08:54	CREATE VIEW VistaMarketing AS SELECT c.company_name, c.phone, c.country, ROUND(AV...	0 row(s) affected	0.031 sec
✓ 2	16:08:58	SELECT * FROM VistaMarketing ORDER BY Media_compras DESC	101 row(s) returned	0.515 sec /

## EJERCICIO 3

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

```
130 • SELECT*
131 FROM VistaMarketing
132 WHERE country = "Germany";
```

Result Grid

company_name	phone	country	Media_compras
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.91
Convallis In Incorporated	06 66 57 29 50	Germany	257.69
Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.32
Augue Foundation	06 88 43 15 63	Germany	253.56
Ac Industries	09 34 65 40 60	Germany	255.17

VistaMarketing 10 x

Output

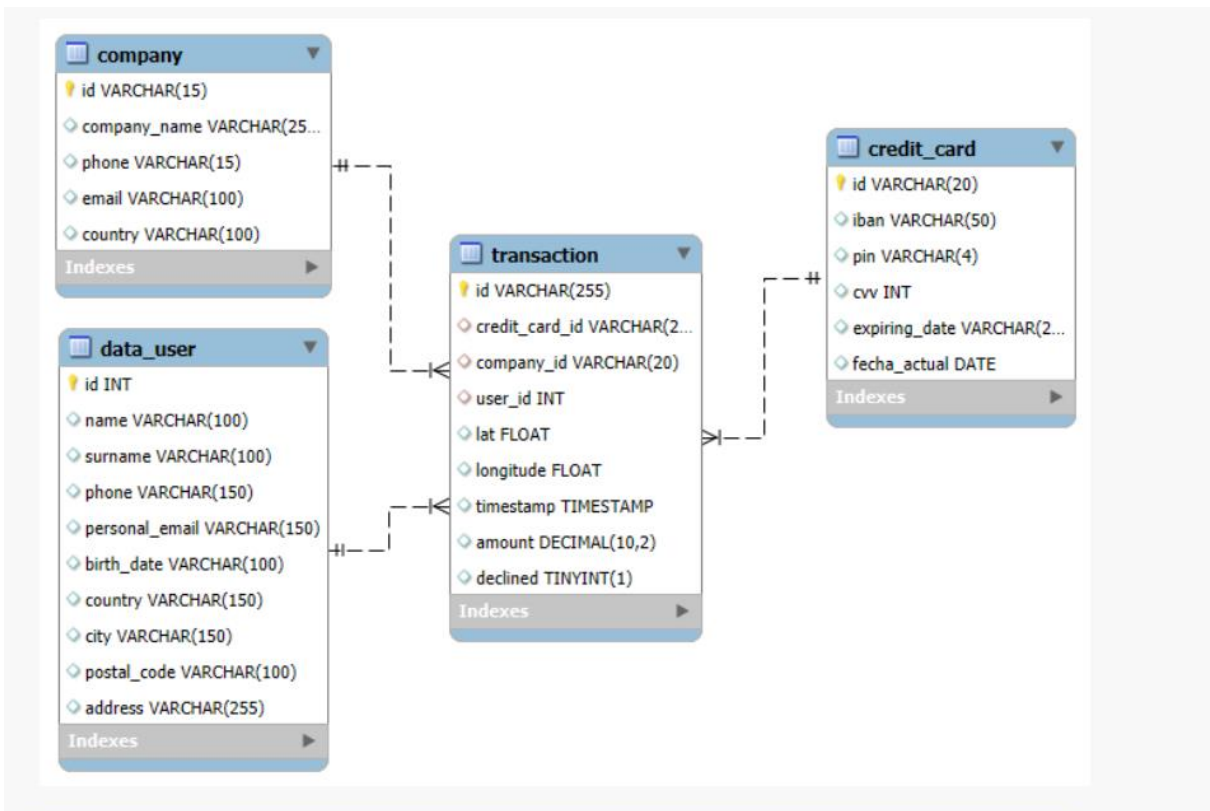
Action Output

#	Time	Action	Message	Duration / Fe
1	16:09:57	SELECT* FROM VistaMarketing WHERE country = "Germany"	8 row(s) returned	0.156 sec / 0

## NIVEL 2

### 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:



Procedemos a cargar los datos de la nueva tabla users:

```
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 );
```

Output

#	Time	Action	Message	Duration / Fetch
1	16:15:53	CREATE TABLE IF NOT EXISTS user (id CHAR(10) PRIMARY KEY, name VARCHAR(100), surname VA...	0 row(s) affected	0.031 sec

```
4977 VALUES ( "4876", "Vocekj", "Gvstzepo", "+61-439-9962", "vocekj.gvstzepo@example.com", "Jun 13, 1996", "Canada", "Kitchener", "H2A 0A1", "461 Gvstzepo St");
4978 VALUES ( "4878", "Iowvee", "Wysqpdjdj", "+30-169-1101", "iowvee.wysqpdjdj@example.com", "Jan 22, 1962", "United States", "San Antonio", "78201", "54 Wysqpdjdj S
4979 VALUES ( "4882", "Vntada", "Ggypxftg", "+32-826-6272", "vntada.ggypxftg@example.com", "Dec 16, 1969", "United States", "San Jose", "95101", "979 Ggypxftg St"
4980 VALUES ( "4887", "Bgtmrv", "Dnumnpnf", "+38-919-4272", "bgtmrv.dnumnpnf@example.com", "Dec 17, 1997", "United States", "Houston", "77001", "778 Dnumnpnf St")
4981 VALUES ( "4892", "Eeatkh", "Ayyimaoj", "+78-549-9530", "eeatkh.ayyimaoj@example.com", "Mar 25, 1962", "United States", "Chicago", "60601", "337 Ayyimaoj St")
4982 VALUES ( "4896", "Tuefdm", "Jcwbuih", "+98-532-2398", "tuefdm.jcwbuih@example.com", "May 19, 2001", "Canada", "Kitchener", "H2A 0A1", "194 Jcwbuih St");
4983 VALUES ( "4899", "Vpdxvy", "Wrbxrosf", "+71-381-4018", "vpdxvy.wrbxrosf@example.com", "Jul 28, 1974", "United States", "San Diego", "92101", "678 Wrbxrosf St
4984 VALUES ( "4901", "Wevsom", "Xnrzrmzw", "+87-131-1665", "wevsom.xnrzrmzw@example.com", "Oct 17, 1968", "United States", "San Antonio", "78201", "563 Xnrzrmzw
4985 VALUES ( "4906", "Qhsaiv", "Kybjjewd", "+99-885-3733", "qhsaiv.kybjjewd@example.com", "Jun 17, 1974", "United States", "San Diego", "92101", "316 Kybjjewd St
4986 VALUES ( "4914", "Efysfr", "Ihzumypa", "+91-980-2052", "efysfr.ihzumypa@example.com", "Jan 11, 1956", "United States", "Chicago", "60601", "91 Ihzumypa St");
4987 VALUES ( "4920", "Vmafsk", "Mksiulvo", "+51-202-2467", "vmafsk.mksiulvo@example.com", "Oct 24, 1958", "United States", "San Antonio", "78201", "344 Mksiulvo
4988 VALUES ( "4927", "Npecth", "Izfujzmh", "+60-230-4439", "npecth.izfujzmh@example.com", "Jan 13, 1979", "United States", "Houston", "77001", "422 Izfujzmh St")
4989 VALUES ( "4928", "Paasvv", "Kdxrcgta", "+75-743-6074", "paasvv.kdxrcgta@example.com", "Feb 17, 1991", "United States", "Phoenix", "85001", "788 Kdxrcgta St")
4990 VALUES ( "4931", "Vrhgue", "Npzdnefa", "+89-915-1161", "vrhgue.npzdnefa@example.com", "Feb 26, 1950", "United States", "Los Angeles", "90001", "252 Npzdnefa
4991 VALUES ( "4939", "Nzultp", "Llylwendk", "+94-736-5751", "nzultp.llylwendk@example.com", "Mar 5, 1999", "United States", "Philadelphia", "19101", "477 Llylwendk
4992 VALUES ( "4945", "Pimdet", "Zvagjirk", "+91-727-8072", "pimdet.zvagjirk@example.com", "Jul 1, 1969", "Canada", "Toronto", "M5A 1A1", "45 Zvagjirk St");
4993 VALUES ( "4947", "Ypyafn", "Tfiiwpnz", "+95-878-8855", "ypyafn.tfiiwpnz@example.com", "Aug 1, 1985", "United States", "Los Angeles", "90001", "360 Tfiiwpnz S
4994 VALUES ( "4950", "Akxgbk", "Fxtbrlub", "+96-670-3225", "akxgbk.fxtbrlub@example.com", "Jan 3, 1984", "Canada", "Vancouver", "V5K 0A1", "19 Fxtbrlub St");
4995 VALUES ( "4952", "Fufhmv", "Sbgtpspk", "+61-136-1383", "fufhmv.sbgtpspk@example.com", "Jul 22, 1960", "Canada", "Montreal", "H1A 0A1", "570 Sbgtpspk St");
4996 VALUES ( "4961", "Phwjfi", "Myxbvzkf", "+34-594-1874", "phwjfi.myxbvzkf@example.com", "Nov 2, 1972", "United States", "New York", "10001", "790 Myxbvzkf St")
4997 VALUES ( "4977", "Gjldyg", "Eggqecse", "+62-591-5059", "gjldyg.eggqecse@example.com", "Aug 16, 1986", "United States", "Chicago", "60601", "537 Eggqecse St")
4998 VALUES ( "4980", "Asgaxi", "Btardzti", "+39-155-6819", "asgaxi.btardzti@example.com", "Jul 24, 1999", "Canada", "Winnipeg", "R2C 0A1", "394 Btardzti St");
4999 VALUES ( "4983", "Eremdc", "Orekppbr", "+59-545-4710", "eremdc.orekppbr@example.com", "Apr 5, 1953", "Canada", "Winnipeg", "R2C 0A1", "635 Orekppbr St");
5000 VALUES ( "4999", "Omjnoj", "Rmuqvngx", "+32-354-7682", "omjnoj.rmuqvngx@example.com", "Aug 26, 1996", "United States", "San Diego", "92101", "437 Rmuqvngx St
```

Output

#	Time	Action	Message	Duration / Fetch
5000	16:17:34	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VAL...	1 row(s) affected	0.000 sec
5001	16:17:34	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VAL...	1 row(s) affected	0.000 sec

Para este ejercicio procedemos primero a realizar los cambios a nivel de las estructuras de las tablas:

### Modificaciones de la tabla company:

```
137 #modificaciones tabla company
138 • ALTER TABLE company
139 DROP COLUMN website;
140
141 #verificación
142 • DESCRIBE company;
143
144 #modificación tabla credit_card;
```

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 11 x

Output

#	Time	Action	Message	Duration / Fetch
1	16:10:36	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec
2	16:10:41	DESCRIBE company	5 row(s) returned	0.000 sec / 0.000

Se ha eliminado la columna “website”.

### Modificaciones de la tabla credit\_card :

```
144 #modificación tabla credit_card:
145 • ALTER TABLE credit_card
146 MODIFY COLUMN id VARCHAR (20);
147
148 • ALTER TABLE credit_card
149 MODIFY COLUMN iban VARCHAR (50);
150
151 • ALTER TABLE credit_card
152 MODIFY COLUMN pin VARCHAR (4);
153
154 • ALTER TABLE credit_card
155 MODIFY COLUMN cvv INT;
156
157 • ALTER TABLE credit_card
158 ADD COLUMN fecha_actual DATE;
159
160 #verificación
```

Output

#	Time	Action	Message	Duration / Fetch
1	16:10:36	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec
2	16:10:41	DESCRIBE company	5 row(s) returned	0.000 sec / 0.000
3	16:11:29	ALTER TABLE credit_card MODIFY COLUMN id VARCHAR (20)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
4	16:11:33	ALTER TABLE credit_card MODIFY COLUMN iban VARCHAR (50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
5	16:11:46	ALTER TABLE credit_card MODIFY COLUMN pin VARCHAR (4)	5001 row(s) affected Records: 5001 Duplicates: 0 Warnings: 0	0.188 sec
6	16:11:52	ALTER TABLE credit_card MODIFY COLUMN cvv INT	5001 row(s) affected Records: 5001 Duplicates: 0 Warnings: 0	0.172 sec
7	16:11:58	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec

```

160 #verificación
161 • DESCRIBE credit_card;
162

```

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(10)	YES		NULL	
fecha_actual	date	YES		NULL	

Result 12 ×

Output

#	Time	Action	Message	Duration / Fetch
1	16:12:41	DESCRIBE credit_card	6 row(s) returned	0.000 sec / 0.000

Se ha cambiado el tipo de datos de las columnas “id”, “iban”, “pin” y “cvv”.

Se ha añadido la columna “fecha\_actual”.

### Modificaciones de la tabla user:

```

163 #modificaciones tabla user
164 • RENAME TABLE user TO data_user;
165
166 • ALTER TABLE data_user
167   MODIFY COLUMN id INT;
168
169 • ALTER TABLE data_user
170   RENAME COLUMN email TO personal_email;
171
172 #verificación
173 • DESCRIBE data_user;
174

```

Output

#	Time	Action	Message	Duration / Fetch
1	16:18:27	RENAME TABLE user TO data_user	0 row(s) affected	0.453 sec
2	16:18:30	ALTER TABLE data_user MODIFY COLUMN id INT	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0	5.172 sec
3	16:18:36	ALTER TABLE data_user RENAME COLUMN email TO personal_email	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.094 sec

```

172 #verificación
173 • DESCRIBE data_user;
174

```

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	

Result 13 ×

Output

#	Time	Action	Message	Duration / Fetch
1	16:19:12	DESCRIBE data_user	10 row(s) returned	0.000 sec / 0.000

Se ha cambiado el nombre de la tabla de user a data\_user.

Se ha cambiado el tipo de dato de la columna “id”.

Se ha cambiado el nombre de la columna “email” a “personal\_email”.

## Manipulación previa a la configuración de las relaciones entre tablas:

```
175 #añadir el id company de la transacción que nos pidieron añadir en el ejercicio nivel 1 para poder configurar la FK con la tabla User_id
176 • INSERT INTO data_user(id, name, surname, phone, personal_email, birth_date, country, city, postal_code, address)
177 VALUES(9999, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);
178
179 #verificación
180 • SELECT*
181 FROM data_user
182 WHERE id = 9999;
```

id	name	surname	phone	personal_email	birth_date	country	city	postal_code	address
9999	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

data\_user 14 x

Output

Action Output

#	Time	Action	Message	Duration / F
1	16:20:14	INSERT INTO data_user(id, name, surname, phone, personal_email, birth_date, country, city, postal_code, address) VALUES(9999, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);	1 row(s) affected	0.016 sec
2	16:20:19	SELECT* FROM data_user WHERE id = 9999	1 row(s) returned	0.000 sec /

Se han añadido los datos del user con id 9999 en la tabla data\_user rellenando los datos que faltan con NULL en base a la transacción que tuvimos que ingresar a la base de datos en el Ejercicio 3, nivel1.

## Modificaciones Foreign Key en la tabla transaction:

```
184 #modificaciones foreign keys en la tabla transaction
185 • ALTER TABLE transaction
186 ADD CONSTRAINT Fk_UserId
187 FOREIGN KEY (user_id) REFERENCES data_user (id);
188
189 #verificación
190 • DESCRIBE transaction;
```

Field	Type	Null	Key	Default	Extra
id	varchar(255)	NO	PRI	NULL	
credit_card_id	varchar(15)	YES	MUL	NULL	
company_id	varchar(20)	YES	MUL	NULL	
user_id	int	YES	MUL	NULL	
lat	float	YES		NULL	
longitude	float	YES		NULL	

Result 15 x

Output

Action Output

#	Time	Action	Message	Duration / Fets
1	16:21:00	ALTER TABLE transaction ADD CONSTRAINT Fk_UserId FOREIGN KEY (user_id) REFERENCES data_user (id);	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	3.718 sec
2	16:21:08	DESCRIBE transaction	9 row(s) returned	0.016 sec / 0

Se ha añadido la relación de Foreign key de la columna “user\_id”.

## 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.

```
202 • CREATE VIEW InformeTecnico AS
203 SELECT
204     t.id AS ID_transaccion,
205     DATE(t.timestamp) AS Fecha_transaccion,
206     U.name AS Nombre_usuario,
207     U.surname AS Apellido_usuario,
208     c.iban AS Iban_tarjeta_credito_usada,
209     c1.company_name AS Nombre_compania_transaccion_realizada,
210     c1.country AS Pais_compania,
211     SUM(t.amount) OVER(PARTITION BY c1.company_name) AS Total_Importe_Ventas_Por_Compania,
212     COUNT(t.id) OVER(PARTITION BY c1.company_name) AS Total_Cantidad_Ventas_Por_Compania
213 FROM transaction AS t
214 JOIN data_user AS u
215 ON t.user_id = u.id
216 JOIN credit_card AS c
217 ON c.id = t.credit_card_id
218 JOIN company AS c1
219 ON c1.id = t.company_id;
```

Output

#	Time	Action	Message	Duration / Fetch
1	16:22:31	CREATE VIEW InformeTecnico AS SELECT t.id AS ID_transaccion, DATE(t.timestamp) AS Fecha_tra...	0 row(s) affected	0.032 sec

```
221 • SELECT*
222 FROM InformeTecnico
223 ORDER BY ID_transaccion DESC;
224
```

Result Grid

ID_transaccion	Fecha_transaccion	Nombre_usuario	Apellido_usuario	Iban_tarjeta_credito_usada	Nombre_compania_transaccion_realizada	Pais_compania	Total_Importe_Ventas_Por_C
FFFD31D6-9495-47CE-B54A-7DB8E1CC274B	2021-08-05	Bmrgli	Tprvnmrc	XX794814451211289182490922	Turpis Company	Netherlands	387595.15
FFFCF76D-ECF0-4985-A2D0-82A7B75998FC	2023-06-17	Dfried	Vlqcdl	XX636251701647892036676034	Amet Nulla Donec Corporation	Italy	390331.32
FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	2019-10-30	Securp	Faofvqfy	XX162677143304223631437567	Nunc Interdum Incorporated	Germany	414651.33
FFF8270D-F53A-4D5D-9666-E5307C53CC84	2024-05-13	Ggzjpa	Urzjul	XX395114267082019952567052	Viverra Donec Foundation	United Kingdom	396517.15
FFF9E3CE-234E-408C-A8EF-F9CAD577224A	2022-12-17	Yshimq	Zpsjsleed	XX8845462156537570367941	Convallis In Incorporated	Germany	390226.50
FFF9E178-6CD2-4DF9-9980-49AE068809B1	2023-02-08	Jevepx	Xwcvzwnm	XX321405515711654384711481	Mus Aenean Eget Foundation	Sweden	400133.75
FFF867C9-17B5-4B1F-AFD9-F8023AAA449E	2015-04-12	Fqlngd	Lvhfqyxi	XX278446342932680979729426	Cras Vehicula Aliquet Industries	Netherlands	391432.93
FFF7042D-18C6-4DD0-B23C-4D90A4AC8F26	2017-07-20	Njoraa	Egsqcui	XX40500927272550082027209	Placerat LLP	Netherlands	384850.79
FFF660D4-4244-47F6-9210-E5D1DCB990B0	2017-05-18	Lopzaj	Itgryfay	XX63376659736627454015125	Pede Cum Ltd	Norway	115902.10
FFF5C660-4441-436D-BD27-E6C53B618622	2018-11-30	Gmnbnu	Oxdvhkdl	XX237820256172646394016483	At Associates	New Zealand	125763.27
FFF54F54-B439-41F0-BDD2-F7332DC1ACAD	2021-03-18	Gqcfyy	Mpifrltn	XX802723943240147612158718	Enim Condimentum Ltd	United Kingdom	392394.69
FFF42F7D-7A0D-46E2-AF72-5996903F8AA9	2017-01-19	Dkiugq	Ycsbpryu	XX926442301555195974199541	At Pede Corp.	Italy	396416.41

InformeTecnico 16 x

Output

#	Time	Action	Message	Duration / Fetch
1	16:22:31	CREATE VIEW InformeTecnico AS SELECT t.id AS ID_transaccion, DATE(t.timestamp) AS Fecha_tra...	0 row(s) affected	0.032 sec

Se ha procedido a realizar la VIEW, añadiendo columnas con datos que se han considerado relevantes como la fecha de la transacción y el país de la compañía que realizó la transacción.

Además se han usados funciones ventanas para mostrar el valor del importe total de todas las ventas realizadas categorizado por empresas así como el recuento total de las transacciones de esas.