

## 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
24   FOREIGN KEY (credit_card_id) REFERENCES credit_card (id);
25
26 #verificacion de la correcta creación de la tabla
27 • DESCRIBE credit_card;
28
29
```

Result Grid

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	

Result 18 x

Output

Action Output

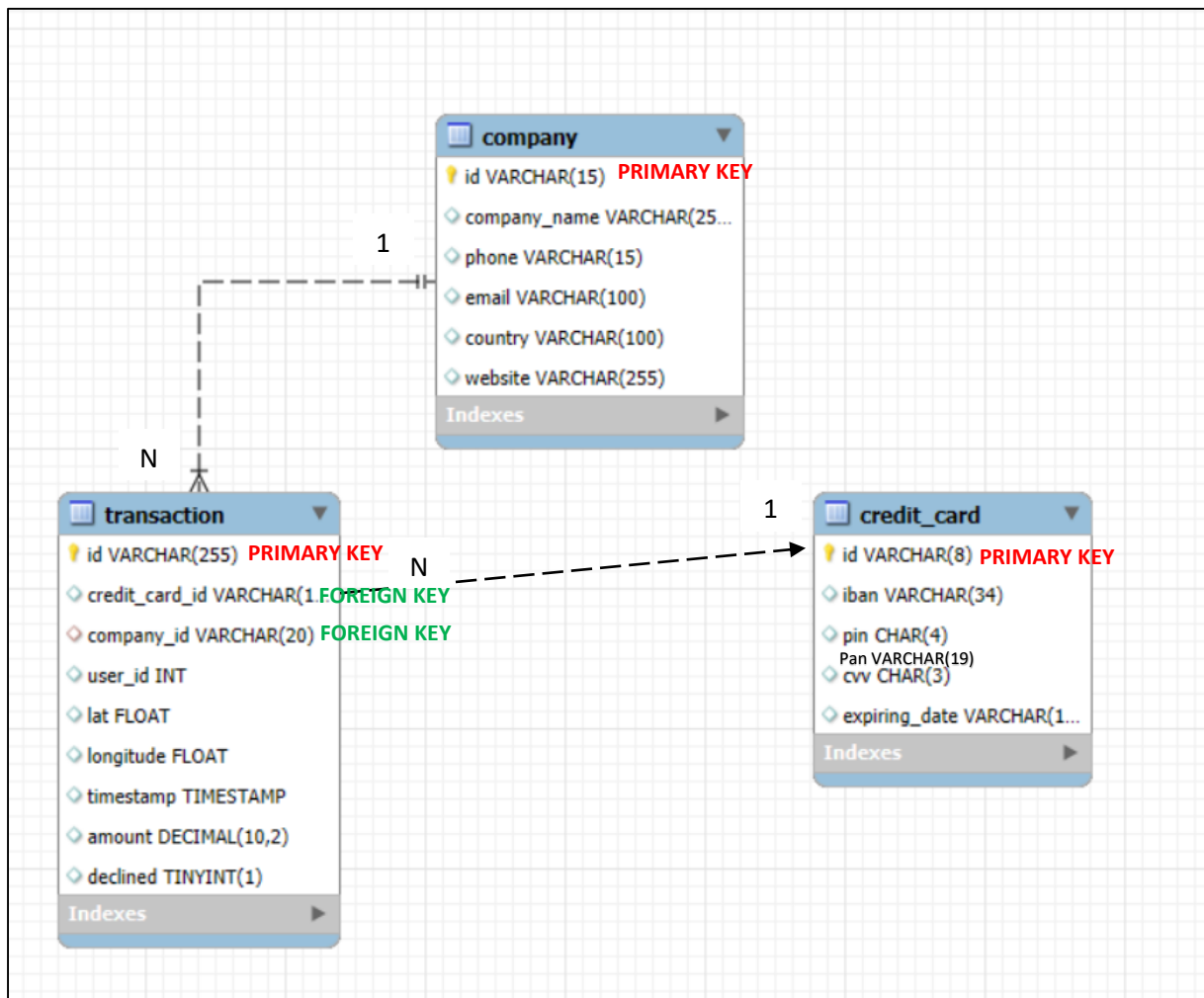
#	Time	Action	Message
38	12:01:00	DESCRIBE credit_card	6 row(s) returned

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 teléfono 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. At the top, a SQL query is entered in a text area:

```
36 • UPDATE credit_card SET iban = "TR323456312213576817699999" WHERE id = "CcU-2938";
37
38 #verificacion
39 • SELECT iban
40 FROM credit_card
41 WHERE id = "CcU-2938";
42
43
```

Below the query, the 'Result Grid' is displayed. It has a 'Filter Rows' field and buttons for 'Export' and 'Wrap Cell Content'. The grid shows a single row with the column 'iban' and the value 'TR323456312213576817699999'.

At the bottom, the 'Output' section is visible, showing 'Action Output' with a table of execution details:

#	Time	Action	Message
40	13:03:44	SELECT iban FROM credit_card WHERE id = "CcU-2938"	1 row(s) returned

## 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

```

55 #Insertamos el id de credit card proporcionado en la tabla credit_card
56 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)
57 VALUES ("CcU-9999", NULL, NULL, NULL, NULL, NULL);
58
59 #Insertamos el id de la empresa proporcionado en la tabla company
60 • INSERT INTO company (id, company_name, phone, email, country, website)
61 VALUES ("b-9999", NULL, NULL, NULL, NULL, NULL);
62
63 #Finalmente podemos ingresar los datos relativos a la transacción
64 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined)
65 VALUES ("10881D1D-5B23-A76C-55EF-C568E49A99DD", "CcU-9999", "b-9999", "9999", "829.999", "-117.999", NULL, "111.11", "0");
66
67 #verificacion
68 • SELECT *
69 FROM transaction
70 WHERE id = "10881D1D-5B23-A76C-55EF-C568E49A99DD";
71

```

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
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 22 x

Output

Action Output

#	Time	Action	Message
42	13:13:37	SELECT * FROM transaction WHERE id = "10881D1D-5B23-A76C-55EF-C568E49A99DD"	1 row(s) returned

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.

```
74 • ALTER TABLE credit_card
75   DROP COLUMN pan;
76
77   #verificacion
78 • SELECT*
79   FROM credit_card;
```

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

	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 24 x

Output

Action Output

#	Time	Action	Message
✓ 44	13:22:40	SELECT* FROM credit_card	5001 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.

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.

```
85  #Antes de eliminar, verificamos que existe antes ese registro
86  • SELECT*
87  FROM transaction
88  WHERE id="000447FE-B650-4DCF-85DE-C7ED0EE1CAAD";
89
90  • DELETE FROM transaction
91  WHERE id ="000447FE-B650-4DCF-85DE-C7ED0EE1CAAD";
92
93  #verificacion:
94  • SELECT*
95  FROM transaction
96  WHERE id="000447FE-B650-4DCF-85DE-C7ED0EE1CAAD";
97
```

Result Grid

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

transaction 2 x

Output

Action Output

#	Time	Action	Message
1	15:15:38	SELECT* FROM transaction WHERE id="000447FE-B650-4DCF-85DE-C7ED0EE1CAAD"	0 row(s) returned

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.

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

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

	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 Convallis 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 4 x

Output

Action Output

#	Time	Action	Message
✓ 3	15:17:25	SELECT * FROM VistaMarketing ORDER BY Media_compras DESC	101 row(s) returned



## EJERCICIO 3

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

```
125 • SELECT*
126 FROM VistaMarketing
127 WHERE country = "Germany";
```

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

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 6 x

Output

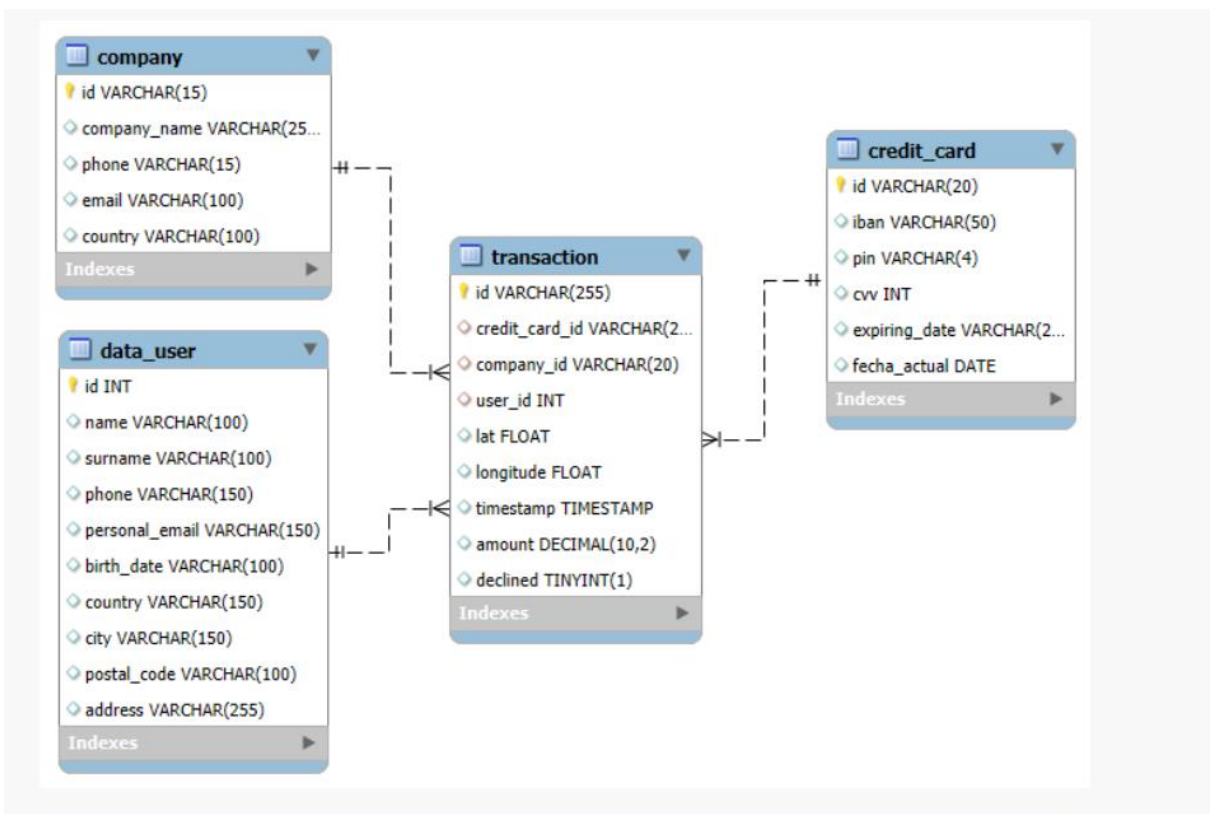
Action Output

#	Time	Action	Message
5	15:18:55	SELECT* FROM VistaMarketing WHERE country = "Germany"	8 row(s) returned

## 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:



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

### Modificaciones de la tabla company:

```
134 • ALTER TABLE company
135     DROP COLUMN website;
136
137     #verificación
138 • DESCRIBE company;
139
140     #modificación tabla credit card;
```

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

	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 8 x

Output :

Action Output

#	Time	Action	Message
✓ 10	15:21:06	DESCRIBE company	5 row(s) returned

Se ha eliminado la columna “website”.

## Modificaciones de la tabla credit\_card :

```
140 #modificación tabla credit_card:
141 • ALTER TABLE credit_card
142   MODIFY COLUMN id VARCHAR (20);
143
144 • ALTER TABLE credit_card
145   MODIFY COLUMN iban VARCHAR (50);
146
147 • ALTER TABLE credit_card
148   MODIFY COLUMN pin VARCHAR (4);
149
150 • ALTER TABLE credit_card
151   MODIFY COLUMN cvv INT;
152
153 • ALTER TABLE credit_card
154   ADD COLUMN fecha_actual DATE;
155
156 #verificación
157 • DESCRIBE credit_card;
```

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

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 10 x

Output

Action Output

#	Time	Action	Message
12	15:25:12	DESCRIBE credit_card	6 row(s) returned

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:

```
159 #modificaciones tabla user
160 • RENAME TABLE user TO data_user;
161
162 • ALTER TABLE data_user
163   MODIFY COLUMN id INT;
164
165 • ALTER TABLE data_user
166   RENAME COLUMN email TO personal_email;
167
168 #verificación
169 • DESCRIBE data_user;
170
```

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

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	

Result 12 x

Output

Action Output

#	Time	Action	Message
16	15:32:46	DESCRIBE data_user	10 row(s) returned

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:

```
171 #añadir el id company de la transaccion que nos pidieron añadir en el ejercicio nivel 1 para poder configurar la FK con la tabla User_id
172 • INSERT INTO data_user(id, name, surname, phone, personal_email, birth_date, country, city, postal_code, address)
173   VALUES(9999, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);
174
175 #verificación
176 • SELECT*
177   FROM data_user
178   WHERE id = 9999;
179
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

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
18	15:36:56	SELECT* FROM data_user WHERE id = 9999	1 row(s) returned

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:

```
182 #modificaciones foreign keys en la tabla transaction
183
184 • ALTER TABLE transaction
185   ADD CONSTRAINT Fk_UserId
186   FOREIGN KEY (user_id) REFERENCES data_user (id);
187
188 #verificación
189 • DESCRIBE transaction;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

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 16 x

Output

Action Output

#	Time	Action	Message
20	15:44:01	DESCRIBE transaction	9 row(s) returned

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.

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

219	SELECT*
220	FROM InformeTecnico
221	ORDER BY ID_transaccion DESC;
222	

ID_transaccion	Fecha_transaccion	Nombre_usuario	Apellido_usuario	Iban_tarjeta_credito_usada	Nombre_compania_transaccion_realizada	Pais_compania	Total_Importe_Ventas
FFFD31D6-9495-47CE-B54A-7D88E1CC274B	2021-08-05	Bmrgli	Tprvvmrc	XX794814451211289182490922	Turpis Company	Netherlands	387595.15
FFFCF76D-ECF0-4985-A2D0-B2A7B75998FC	2023-06-17	Dfired	Vilqjdl	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
FFFB270D-F53A-4D5D-9666-E5307C53CC84	2024-05-13	Ggzjpa	Uirzjhl	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	Xwiczwm	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-18C6-4DD0-823C-4D90A4AC8F26	2017-07-20	Njoraa	Egsquili	XX405009272572550082027209	Placerat LLP	Netherlands	384850.79
FFF660D4-4244-47F6-9210-ESD1DCB99DB0	2017-05-18	Lopzaj	Itgryfay	XX63376659736627454015125	Pede Cum Ltd	Norway	115902.10
FFF5C660-4441-436D-BD27-E6C53B618622	2018-11-30	Gmrbnu	Oxdvhl	XX237820256172646394016483	At Associates	New Zealand	125763.27

#	Time	Action	Message	Du
22	15:54:31	SELECT* FROM InformeTecnico ORDER BY ID_transaccion DESC	100000 row(s) returned	9.3

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 realizada categorizado por empresas así como el recuento total de las transacciones de esas.