

SPRINT 3

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.

ID DE LA TARJETA (clave primaria)

NUMERO DE CUENTA (IBAN)

NUMERO DE LA TARJETA (pan)

CODIGO O PIN

CVV

CADUCIDAD

```
1  # EJERCICIO 1
2  #CREAR UNA TABLA
3  • USE transactions;
4  • CREATE TABLE credit_card (
5    id varchar(15) primary KEY, # PRIMARY KEYS
6    iban varchar(40) NULL,
7    pan varchar(25) NULL,
8    pin varchar(6) NULL,
9    cvv varchar(6) NULL,
10   expiring_date varchar(8) null);
```

Jump to
: content
abled. U
bar to
nt help
nt care
to tog
ic help.

Views
Stored Procedures
Functions

Administrative: schemas
Information

No object selected

31 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
32 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
33 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
34 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
35 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
36 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
37 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
38 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
39 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
40 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
41 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
42 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
Connect Help Stoppage

#	Time	Action	Message	Duration / Fetch
285	11:16:35	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)	1 row(s) affected	0.000 sec
286	11:16:35	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)	1 row(s) affected	0.000 sec
287	11:16:35	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)	1 row(s) affected	0.000 sec
288	11:16:35	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)	1 row(s) affected	0.000 sec
289	11:16:35	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)	1 row(s) affected	0.000 sec
290	11:16:35	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)	1 row(s) affected	0.016 sec

Client Info: Session

SPRINT 3

```
12 • alter table transaction
13 add constraint fk_credit_card
14 foreign key (credit_card_id)
15 references credit_card(id);
16
17 • show columns from credit_card;
```

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

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	HULL	
iban	varchar(40)	YES		HULL	
pan	varchar(25)	YES		HULL	
pin	varchar(6)	YES		HULL	
cvv	varchar(6)	YES		HULL	
expiring_date	varchar(8)	YES		HULL	

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

Field	Type	Null	Key	Default	Extra
id	varchar(255)	NO	PRI	HULL	
credit_card_id	varchar(15)	YES	MUL	HULL	
company_id	varchar(20)	YES	MUL	HULL	
user_id	int	YES		HULL	
lat	float	YES		HULL	
longitude	float	YES		HULL	
timestamp	timestamp	YES		HULL	
amount	decimal(10,2)	YES		HULL	

Result 2 x

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.

Se realizo una consulta para validar el id

SPRINT 3

```
11
12 • select *
13 from transactions.credit_card
14 where id = 'CcU-2938';
```

	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22
*	NULL	NULL	NULL	NULL	NULL	NULL

Para realizar el cambio se utilizó la siguiente instrucción

```
SQL File 5  tarea2* x datos_introducir_credit*
15 from transactions.credit_card
16 where id = 'CcU-2938';
17
18 • UPDATE transactions.credit_card
19 SET iban = 'R323456312213576817699999'
20 WHERE id = 'CcU-2938';
21
22
```

	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	R323456312213576817699999	5424465566813633	3257	984	10/30/22
*	NULL	NULL	NULL	NULL	NULL	NULL

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

SPRINT 3

Debido a que las tablas están vinculadas entre sí, es decir la tabla transacción depende de la tabla de compañía y de la de crédito se debe crear primero en ambas tablas.

En la siguiente se tiene la de company

```
26 FROM transactions.transaction;  
27  
28 • SELECT*  
29 FROM transactions.company  
30 where id='b-9999';  
31  
32  
33  
34  
35  
36
```

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

	id	company_name	phone	email	country	website
▶	b-9999	NULL	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

Ahora obtenemos la de credit_card

```
35 • SELECT*  
36 FROM transactions.credit_card  
37 where id='CcU-9999';  
38 • insert into transactions.credit_card (id)  
39 values ('CcU-9999');
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	id	iban	pan	pin	cvv	expiring_date
▶	CcU-9999	NULL	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

Para así insertar el registro en transacción

SPRINT 3

```
40
41 #Creo el nuevo usuario
42 • insert into transactions.transaction(id,credit_card_id,company_id, user_id, lat, longitude,timestamp, amount, declined)
43 values ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', now(), '111.11', '0');
44 #Visualizo la nueva transacción
45 • SELECT * FROM transaction
46 WHERE id = "108B1D1D-5B23-A76C-55EF-C568E49A99DD";
47
48
--
```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	2024-05-29 19:16:22	111.11	0

Utilice la función NOW() ya que es una función de fecha y hora, al usarla coloca o muestra la fecha y hora del momento que se efectúa o ejecuta la sentencia.

Ejercicio 4

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

Se muestra la tabla para realizar el cambio

```
41
42
43
44
45
46
47
48
49 #EJERCICIO 4
50 #Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla credit_card
51 • SELECT *
52 FROM transactions.credit_card;
53
54 • ALTER TABLE credit_card
55 DROP COLUMN pan;
56
--
```

id	iban	pan	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	5424465566813633	3257	984	10/30/22
CcU-2945	DO26854763748537475216568689	5142423821948828	9080	887	08/24/23
CcU-2952	BG45IVQL52710525608255	4556 453 55 5287	4598	438	06/29/21
CcU-2959	CR7242477244335841535	372461377349375	3583	667	02/24/23

Luego del cambio

```
48
49 #EJERCICIO 4
50 #Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla
51 • SELECT *
52 FROM transactions.credit_card;
53
54 • ALTER TABLE credit_card
55 DROP COLUMN pan;
56
--
```

id	iban	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	3257	984	10/30/22
CcU-2945	DO26854763748537475216568689	9080	887	08/24/23
CcU-2952	BG45IVQL52710525608255	4598	438	06/29/21
CcU-2959	CR7242477244335841535	3583	667	02/24/23

SPRINT 3

Nivel 2

Ejercicio 1

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

Antes de eliminarlo

```
57 # NIVEL 2
58 #Elimina de la tabla transacción el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de datos.
59 • select *
60 from transactions.transaction
61 where id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	92	81.9185	-12.5276	2021-08-28 23:42:24	466.92	0

Después

```
57 # NIVEL 2
58 #Elimina de la tabla transacción el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de datos.
59 • select *
60 from transactions.transaction
61 where id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
62
63 • delete
64 from transactions.transaction
65 where id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

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

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.

SPRINT 3

The screenshot shows a SQL IDE with a query editor and a results grid. The query in the editor is:

```
81 select company.id, company.company_name, company.phone, company.country, avg(amount, 2) as media
82 from transactions.company
83 join transactions.transaction on company.id = transaction.company_id
84 where transaction.declined = 0
85 group by company.id
86 order by media desc;
87
88 • SELECT *
89 FROM vistmarketing;
90
```

The results grid shows the following data:

company_name	phone	country	avg(amount)
Eget Ipsum Ltd	03 67 44 56 72	United States	481.860000
Sed Id Limited	07 28 18 18 13	United States	477.510000
Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.100000
Nunc Sit Incorporated	07 28 42 63 63	Norway	461.830000
Non Manna LLC	06 71 73 13 17	United Kingdom	458.740000

The output pane shows the following message:

```
1 11:31:07 SELECT * FROM vistmarketing LIMIT 0, 1000
Message
101 row(s) returned
```

Ejercicio 3

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

The screenshot shows a SQL IDE with a query editor and a results grid. The query in the editor is:

```
87 #ejercicio3
88 #Filtra la vista VistaMarketing para mostrar sólo las compañías que tienen su país de residencia en "Germany"
89 • SELECT * FROM transactions.vistmarketing WHERE country = 'Germany';
```

The results grid shows the following data:

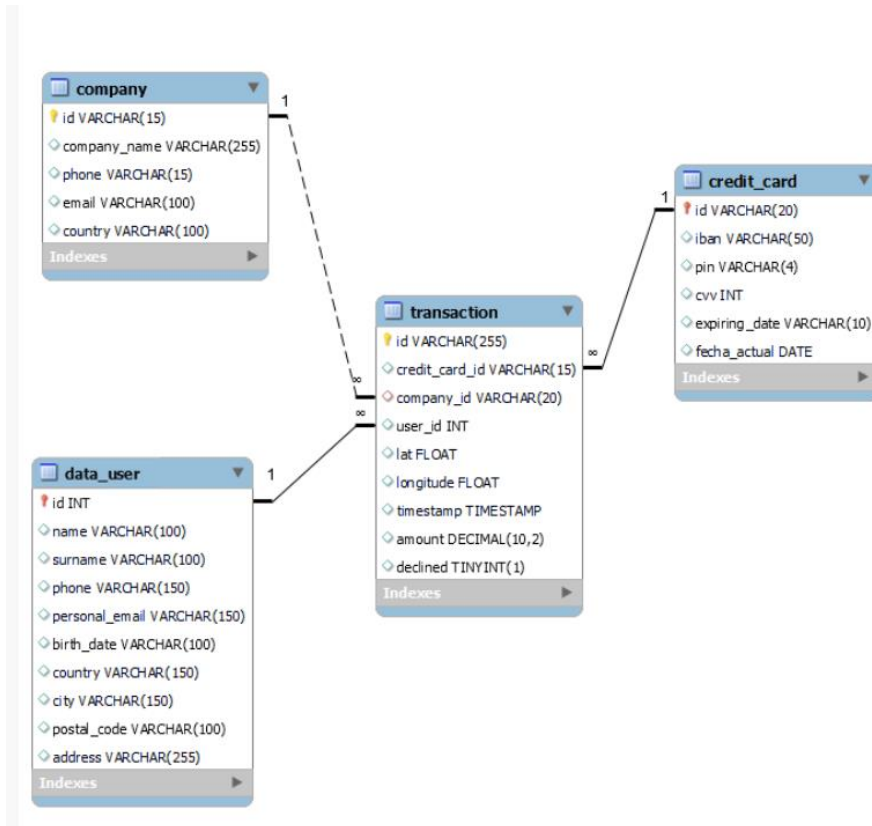
company_name	phone	country	avg(amount)
Auctor Mauris Corp.	05 62 87 14 41	Germany	308.990000
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.570000
Aliquam PC	01 45 73 52 16	Germany	280.340000
Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000

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:

SPRINT 3



En la Tabla company como se puede observar no se encuentra la columna website procedemos a eliminar ya que es uno de los cambios realizados

```
select *
```

```
from transactions.company;
```

```
ALTER TABLE company
```

```
DROP COLUMN website;
```


SPRINT 3

```
98 # Tabla company se debe eliminar la columna web site
99 • select *
100 from transactions.company;
101
102 • ALTER TABLE company
103 DROP COLUMN website;
```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Contents
	id	company_name	phone	email	country
▶	b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.telus@yahoo.net	Germany
	b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia
	b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States
	b-2234	Convalis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany

Se realizo el cambio a la tabla USER

rename table user to data_user;

show tables;

```
103 DROP COLUMN website;
104
105 # cambiar el nombre a la tabla USER
106
107 • rename table user to data_user;
108
109 • show tables;
```

Result Grid	
Filter Rows: <input type="text"/>	
Export:	
Wrap Cell Content:	
Tables_in_transactions	
▶	company
	credit_card
	data_user
	transaction

En la tabla DATA_USER renombrar email por personal_email

ALTER TABLE data_user CHANGE email personal_email VARCHAR(150);

```
109 • show tables;
110 # renombrar en la tabla data_user email por personal_email
111 • ALTER TABLE data_user CHANGE email personal_email VARCHAR(150);
112
113 • SELECT *
```

Result Grid										
Filter Rows: <input type="text"/>										
Edit:										
Export/Import:										
Wrap Cell Content:										
	id	name	surname	phone	personal_email	birth_date	country	city	postal_code	address
▶	1	John	Smith	1 202 201 2051	john.smith@protonmail.edu	Nov 17, 1985	United States	London	73544	340 7010 G...

Generalmente en una tabla no debe usarse la clave primaria como foreign key, se realiza el cambio

SPRINT 3

En la tabla CREDI_CARD cambiar los siguientes campos, eliminar el campo PAN y modificar el campo fecha actual con tipo DATE

Id a varchar(20)

Iban a VARCHAR(50)

pin a VARCHAR(4)

cvv a INT

expiring_date a VARCHAR (10).

```
116 #En la tabla CREDI_CARD cambiar los siguientes campos Id a varchar(20
117 • ALTER TABLE transactions.credit_card MODIFY COLUMN id VARCHAR(20);
118 • ALTER TABLE credit_card MODIFY COLUMN iban VARCHAR(50);
119 • ALTER TABLE credit_card MODIFY COLUMN pin VARCHAR(4);
120 • ALTER TABLE credit_card MODIFY COLUMN cvv INT;
121 • ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR(10);
122 #Eliminar el campo pan.
123 • ALTER TABLE credit_card DROP COLUMN pan;
124 -- Agrego el campo fecha_actual con tipo DATE
125 • ALTER TABLE credit_card ADD fecha_actual DATE;
126 #Visualización:
127 • SHOW COLUMNS
128 FROM credit card;
```

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

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI		
iban	varchar(50)	YES			
pin	varchar(4)	YES			
cvv	int	YES			
expiring_date	varchar(10)	YES			
fecha_actual	date	YES			

Ejercicio 2

La empresa también te solicita crear una vista llamada "InformeTecnico" que contenga la siguiente información:

SPRINT 3

- o ID de la transacción
- o Nombre del usuario/a
- o Apellido del usuario/a
- o IBAN de la tarjeta de crédito usada.
- o Nombre de la compañía de la transacción realizada.
- o Asegúrate de incluir información relevante de ambas tablas y utiliza alias para cambiar de nombre columnas según sea necesario.

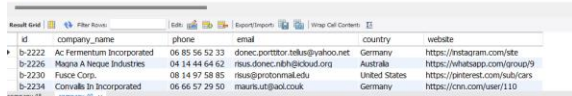
Muestra los resultados de la vista, ordena los resultados de forma descendente en función de la variable ID de transacción.

Para este ejercicio

Se valida los campos de la tabla COMPANYY

Antes

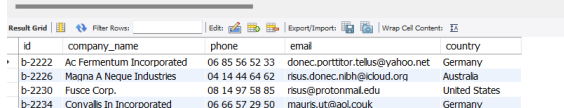
```
93 # Tabla company se debe eliminar la columna web site
94 * select *
95 from transactions.company;
96
97 * ALTER TABLE company
98 DROP COLUMN website;
99
100
```



id	company_name	phone	email	country	website
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porrtitor.telus@yahoo.net	Germany	https://instagram.com/site
b-2226	Magna A Neque Industries	04 14 44 64 62	rsus.donec.nibh@icloud.org	Australia	https://whatsapp.com/group/9
b-2230	Fusce Corp.	08 14 97 58 85	rsus@protonmail.edu	United States	https://pinterest.com/sub/cars
b-2234	Convalis In Incorporated	06 66 57 29 50	mauris.ut@aol.co.uk	Germany	https://cnn.com/user/110

Después

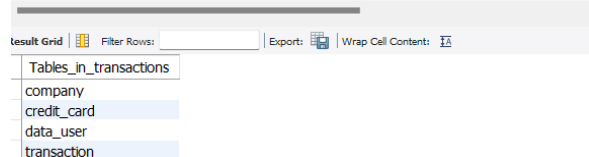
```
--
91 #NIVEL3
92 # EJERCICIO 1
93 # Tabla company se debe eliminar la columna web site
94 * select *
95 from transactions.company;
96
97 * ALTER TABLE company
98 DROP COLUMN website;
99
--
```



id	company_name	phone	email	country
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porrtitor.telus@yahoo.net	Germany
b-2226	Magna A Neque Industries	04 14 44 64 62	rsus.donec.nibh@icloud.org	Australia
b-2230	Fusce Corp.	08 14 97 58 85	rsus@protonmail.edu	United States
b-2234	Convalis In Incorporated	06 66 57 29 50	mauris.ut@aol.co.uk	Germany

Renombrar la tabla USER

```
101
102 * rename table user to data_user;
103
104 * show tables;
105
```



Tables_in_transactions
company
credit_card
data_user
transaction

No he podido agregar la fk, debido a que no agregue el usuario en el ejercicio número 2 y no había Fk solventando el error

SPRINT 3

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

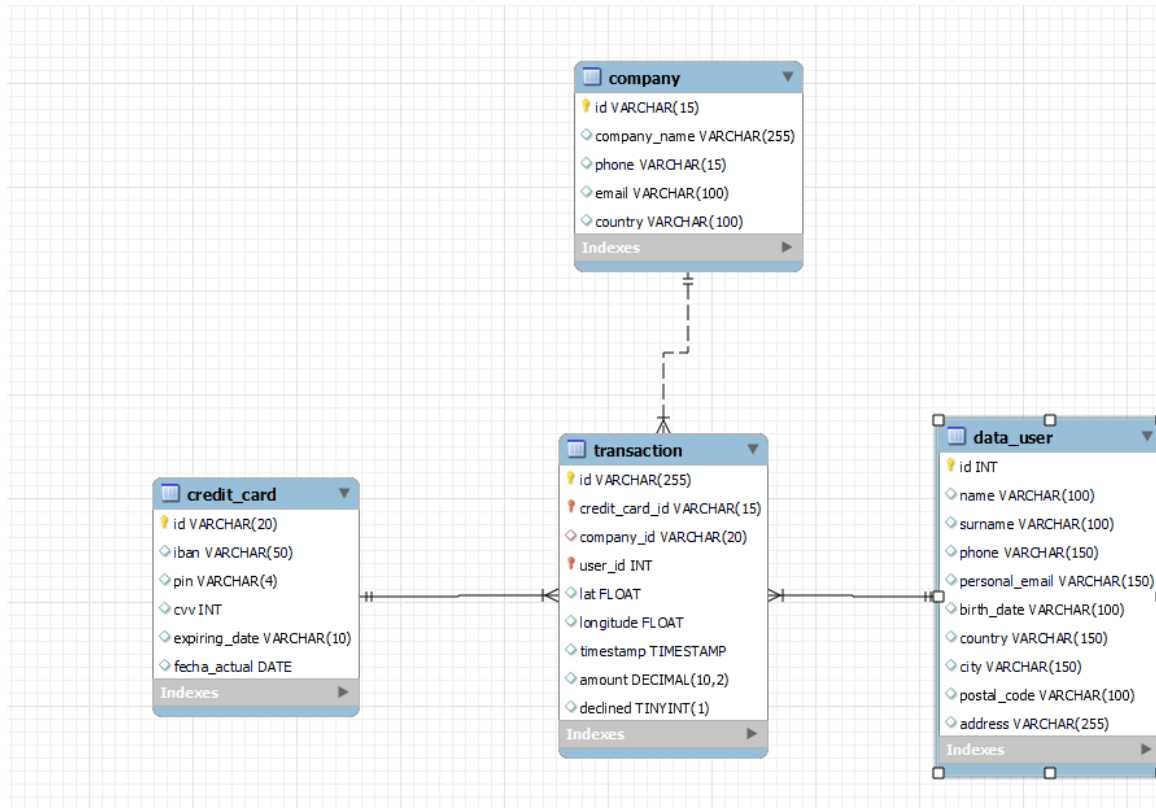
```
1 • SELECT user_id
2 FROM transaction
3 WHERE user_id NOT IN (SELECT id FROM data_user);
4
```

The result grid at the bottom shows a single column named `user_id` with one row containing the value `9999`.

user_id
9999

El esquema quedo d la siguiente forma

SPRINT 3



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.

SPRINT 3

Find Q| vista

```
154 #vista llamada "InformeTecnico"
155 • create view informetecnico as
156     select transaction.id,data_user.name, data_user.surname,credit_card.iban, company.company_name
157     from transactions.transaction
158     join transactions.data_user on transaction.id = data_user.id
159     join transactions.credit_card on transaction.credit_card_id = credit_card.id
160     join transactions.company on transaction.company_id = company.id;
161
162 • SELECT *
163 FROM informetecnico
164 ORDER BY transaction.id DESC;
165
```

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

	id	name	surname	iban	company_name
▶	9FBB3D61-D3C2-E5BB-4BC3-6CC83...	Mannix	Mcclain	SI90187218272592688	Aliquam PC
	9F698741-2C27-2C18-9CD3-E3C7F...	Mannix	Mcclain	CR7242477244335841535	Lorem Eu Incorporated
	9EE639A3-2836-87C2-207F-AF3941...	Mannix	Mcclain	VG1468087984174645729577	Ut Semper Foundation
	9DED76AE-FE36-ACC3-7377-AE98A...	Mannix	Mcclain	FI7847379817377733	Malesuada PC
	9D4AB77C-9F51-FD95-EAB4-A3716...	Mannix	Mcclain	SE8424235059254817742452	Nunc Interdum Incorporated
	9D388ED9-558B-EF5A-E4B3-C9668...	Mannix	Mcclain	BH62714428368066765294	Enim Condimentum Ltd
	9CEA5069-EEDB-11B8-F8DE-9E858...	Mannix	Mcclain	SE2813123487163628531121	Nunc Interdum Incorporated
	9CB62BC2-FFA6-449F-59E4-379EDA...	Mannix	Mcclain	HR5688747222861164805	Erat LLP

data_user 69 | informetecnico 85 x

Output

NIVEL 1

EJERCICIO 2

Como olvide pegar el diagrama realizado en el ejercicio N°1 cree un nuevo schema y realice alguna cambios para ese ejercicio

Realizando mejoras en los campos de la tabla credi_card, se realizaron cambios a la fecha como no es posible modificar directamente un campo de tipo VARCHAR a DATE, en una tabla existente de MYSQL, ya que los datos tienen estructuras diferentes, se creo una nueva columna a la tabla para almacenar las fechas en formato DATE

La misma nombro como fecha_expiracion

Esto se debe a que los tipos de datos VARCHAR a DATE tienen estructuras y propósitos diferentes.

SPRINT 3

```
18
19 #nueva columna
20 • ALTER TABLE credit_card
21   ADD fecha_caducidad DATE;
22 #se actualizan los valores del campo
23 • UPDATE credit_card
24   SET fecha_caducidad = STR_TO_DATE(expiring_date, "%m/%d/%Y");
25 # Ahora se elimina el campo
26 • ALTER TABLE credit_card
27   DROP COLUMN expiring_date;
28 #Cambio el nombre de la columna nuevamente a expiring_date
29 • ALTER TABLE credit_card CHANGE fecha_caducidad expiring_date DATE;
30
```

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI		
iban	varchar(40)	YES			
pan	varchar(25)	YES			
pin	varchar(6)	YES			
cvv	varchar(6)	YES			
expiring_date	date	YES			

