

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.

```
-- Creo la tabla credit_card dentro de database transactions
```

```
use transactions;
```

```
CREATE TABLE credit_card (  
  id VARCHAR(15) PRIMARY KEY,  
  iban VARCHAR(40),  
  pan VARCHAR(20) UNIQUE,  
  pin CHAR(10),  
  cvv CHAR(5),  
  expiring_date VARCHAR (10)  
);
```

```
-- Insertamos datos de credit_card
```

```
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)  
VALUES ( 'CcU-2938', 'TR301950312213576817638661', '5424465566813633', '3257', '984', '10/30/22');  
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)  
VALUES ( 'CcU-2945', 'DO26854763748537475216568689', '5142423821948828', '9080', '887',  
'08/24/23');
```

```
*  
*  
*
```

```
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)  
VALUES ( 'CcU-4842', 'SA2156708581957118818229', '3774 636724 83250', '4655', '750', '11/11/24');  
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)  
VALUES ( 'CcU-4849', 'SE2813123487163628531121', '5223363813491514', '9992', '779', '03/21/25');  
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)  
VALUES ( 'CcU-4856', 'TR373872558313545667124286', '349528235713651', '9086', '974',  
'05/19/23');
```

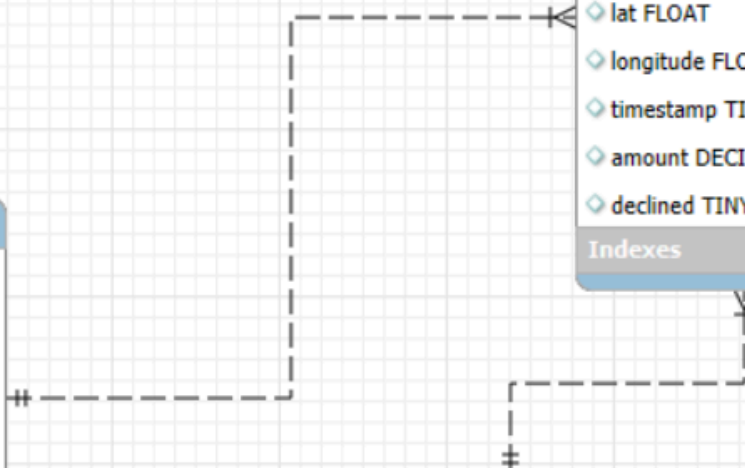
```
-- Modifico la table transaction para añadir el id de la table credit_card como Foreign Key
```

```
ALTER TABLE transaction  
ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
```

credit_card
id VARCHAR(15)
iban VARCHAR(40)
pan VARCHAR(20)
pin CHAR(10)
cvv CHAR(5)
expiring_date VARCHAR(10)
Indexes

company
id VARCHAR(15)
company_name VARCHAR(255)
phone VARCHAR(15)
email VARCHAR(100)
country VARCHAR(100)
website VARCHAR(255)
Indexes

transaction
id VARCHAR(255)
credit_card_id VARCHAR(15)
company_id VARCHAR(20)
user_id INT
lat FLOAT
longitude FLOAT
timestamp TIMESTAMP
amount DECIMAL(10,2)
declined TINYINT(1)
Indexes



- Ejercicio 2

El departament de Recursos Humans ha identificat un error en el número de compte de l'usuari amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és: R323456312213576817699999. Recorda mostrar que el canvi es va realitzar.

```
299
300 -- Este es el iban erroneo (antes del cambio):
301 • select * from credit_card
302 where id = 'Ccu-2938';
303
```

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

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

```
304 -- Modifico el iban para que sea igual a R323456312213576817699999
305 • update credit_card
306 set iban = 'R323456312213576817699999'
307 WHERE id = 'Ccu-2938';
308
309 -- Este es el iban correcto tras el cambio
310 • select * from credit_card
311 where id = 'Ccu-2938';
```

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

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
lat	829.999
longitud	-117.999
amunt	111.11
declined	0

-- Inicialmente, este script no funciona porque tengo que agregar la info del nuevo usuario tanto en tabla company como en tabla credit_card

```
insert into transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
values ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999,
111.11, 0);
```

```
-- Primero tengo que registrar a la nueva compañía b-9999 en la tabla company
insert into company (id)
values ('b-9999');
```

```
-- Luego registro la tarjeta de crédito en la tabla credit_card
insert into credit_card (id)
values ('CcU-9999');
```

```
-- Finalmente inserto la nueva transaccion que corresponde a la nueva company
```

```
insert into transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
values ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999,
111.11, 0);
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'company', 'credit_card', and 'transaction' tables listed. The main editor window shows a SQL script with the following queries:

```
320 • insert into credit_card (id)
321 values ('CcU-9999');
322
323 -- Finalmente inserto la nueva transaccion que corresponde a la nueva company
324 • insert into transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
325 values ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, 111.11, 0);
326
327 • select * from transaction
328 where id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD';
329
```

The 'Result Grid' at the bottom shows the output of the last query, displaying a single row of data for the transaction with ID '108B1D1D-5B23-A76C-55EF-C568E49A99DD'.

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	NULL	111.11	0

- Ejercicio 4

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

```
alter table credit_card  
drop column pan;
```

MySQL Workbench

dataanalytics x MySQL Model* x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator: lb_sprint03_nivel1* x transactions.transaction transactions.company

SCHEMAS

Filter objects

- Tables
 - company
 - Columns
 - id
 - company_name
 - phone
 - email
 - country
 - website
 - Indexes
 - Foreign Keys
 - Triggers
 - credit_card
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - transaction
 - Columns
 - Indexes
 - Foreign Keys

Administration Schemas

Information

```
327 • select * from transaction
328   where id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD';
329
330   -- Ejercicio 4
331   -- Eliminar la columna "pan" de la tabla credit_card
332 • alter table credit_card
333     drop column pan;
334
335 • select * from credit_card;
336
```

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

	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
	CcU-2966	BG72LKTQ15627628377363	4900	130	10/29/24
	CcU-2973	PT87806228135092429456346	8760	887	01/30/25
	CcU-2980	DE39241881883086277136	5075	596	07/24/22
	CcU-2987	GE89681434837748781813	2298	797	10/31/23
	CcU-2994	BH62714428368066765294	7545	595	02/28/22
	CcU-3001	CY49087426654774581266832110	9562	867	09/16/22

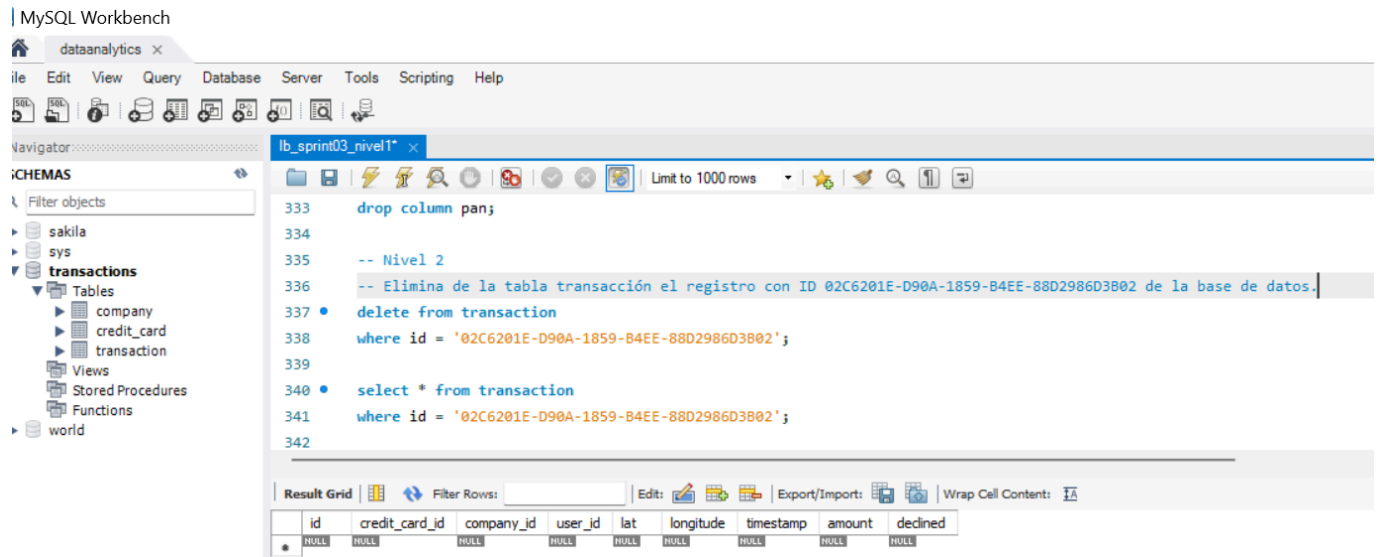
Nivel 2

Ejercicio 1

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

delete from transaction

```
where id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```



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.

```
create or replace view VistaMarketing as
select company_name, phone, country, round(avg(amount),2) as average
from company as c
inner join transaction
on company_id = c.id
where declined = 0
group by company_name, country, phone
order by average desc;

select * from vistamarketing;
```

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane shows a tree view with 'sakila' and 'sys' databases. Under 'sakila', there are 'Tables' (company, credit_card, transaction) and 'Views' (vistamarketing). The 'vistamarketing' view is selected, and its columns (company_name, phone, country, average) are listed. The main pane shows the SQL script for creating the view and selecting from it. The 'Result Grid' at the bottom displays the data for the 'vistamarketing' view, ordered by average purchase amount in descending order.

company_name	phone	country	average
Eget Ipsum Ltd	03 67 44 56 72	United States	481.86
Sed Id Limited	07 28 18 18 13	United States	477.51
Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.10
Nunc Sit Incorporated	07 28 42 63 63	Norway	461.83
Non Magna LLC	06 71 73 13 17	United Kingdom	458.74
Maecenas Malesuada Fringilla Inc.	09 38 53 76 61	Netherlands	451.29
Erat LLP	03 18 88 77 79	Netherlands	448.44
Tortor Nunc Commodo Company	05 35 92 77 16	United States	447.11
Sed Id Limited	08 47 56 71 57	Italy	444.16

Ejercicio 3

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

```
select * from vistamarketing
where country = 'Germany';
```

MySQL Workbench

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'company', 'credit_card', and 'transaction', and views like 'vistamarketing'. The 'vistamarketing' view is selected, showing its columns: 'company_name', 'phone', 'country', and 'average'. The main editor pane shows a SQL query with line numbers 350 to 360. The query is a SELECT statement that joins the 'company' table with the 'transaction' table, filters for 'declined = 0', groups by 'company_name', 'country', and 'phone', orders by 'average' in descending order, and then selects all columns from the 'vistamarketing' view where the 'country' is 'Germany'. The 'Result Grid' at the bottom displays the results of the query, showing a list of companies from Germany with their phone numbers and average values.

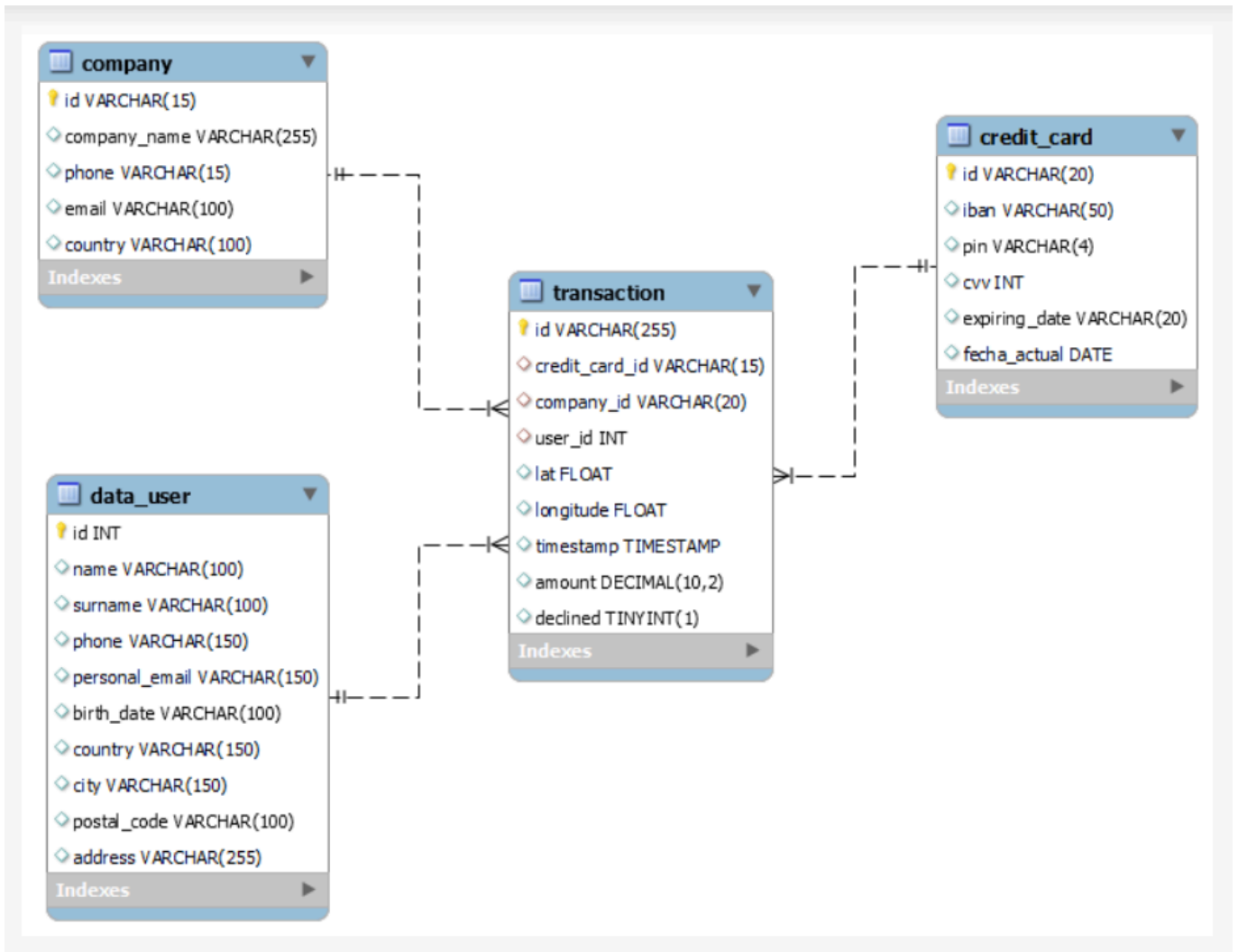
```
350 select company_name, phone, country, round(avg(amount),2) as average
351 from company as c
352 inner join transaction
353 on company_id = c.id
354 where declined = 0
355 group by company_name, country, phone
356 order by average desc;
357
358 • select * from vistamarketing
359 where country = 'Germany';
360
```

company_name	phone	country	average
Ac Industries	09 34 65 40 60	Germany	396.15
Auctor Mauris Corp.	05 62 87 14 41	Germany	308.99
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.57
Aliquam PC	01 45 73 52 16	Germany	280.34
Rutrum Non Inc.	02 66 31 61 09	Germany	266.90
Nunc Interdum Incorporated	05 18 15 48 13	Germany	242.95
Convallis In Incorporated	06 66 57 29 50	Germany	60.99
Augue Foundation	06 88 43 15 63	Germany	15.05

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:



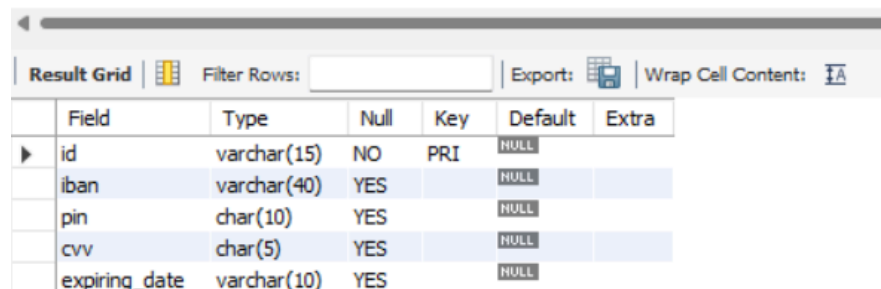
Recordatorio;

En esta actividad, es necesario que describas el "paso a paso" de las tareas realizadas. Es importante realizar descripciones sencillas, simples y fáciles de comprender. Para realizar esta actividad deberás trabajar con los archivos denominados "estructura_datos_user" y "datos_introducir_user"

Primero, voy a modificar la tabla credit_card

Campos de tabla credit_card ANTES de la modificación

```
301
302 -- Muestro campos de tabla credit_card
303 • show columns from credit_card;
304
305
```



	Field	Type	Null	Key	Default	Extra
▶	id	varchar(15)	NO	PRI	NULL	
	iban	varchar(40)	YES		NULL	
	pin	char(10)	YES		NULL	
	cvv	char(5)	YES		NULL	
	expiring_date	varchar(10)	YES		NULL	

Modifico campos id, iban, expiring_date de la tabla credit_card.

```
alter table credit_card
modify id varchar(20),
modify iban varchar(50),
modify expiring_date varchar(20);
```

Modifico campos pin, cvv de la tabla credit_card.

```
alter table credit_card
modify pin varchar(4),
modify cvv int;
```

Agrego campo fecha_actual a la tabla credit_card:

```
alter table credit_card add column fecha_actual date;
show columns from credit_card;
```

Campos de tabla credit_card DESPUES de las modificaciones:

MySQL Workbench

dataanalytics x

File Edit View Query Database Server Tools Scripting Help

Navigator: lb_sprint03_nivel1* lb_sprint03_nivel3* x

Limit to 1000 rows

SCHEMAS

Filter objects

- sakila
- sys
- transactions
 - Tables
 - company
 - credit_card
 - transaction
 - user
 - Views
 - vistamarketing
 - Stored Procedures
 - Functions
- world

Administration Schemas

Information

No object selected

```

305 -- Modifico campos id, iban, expiring_date de la tabla credit_card.
306 • alter table credit_card
307 modify id varchar(20),
308 modify iban varchar(50),
309 modify expiring_date varchar(20);
310
311 -- Modifico campos pin, cvv de la tabla credit_card.
312 • alter table credit_card
313 modify pin varchar(4),
314 modify cvv int;
315
316 • show columns from credit_card;
317
318 -- Agrego campo fecha_actual a la tabla credit_card
319 • alter table credit_card add column fecha_actual date;
320
321 • show columns from credit_card;
322

```

Result Grid

Filter Rows:

Export:

Wrap Cell Contents:

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	

Result 5 x

Eliminar el campo website de tabla company.

Tabla company **ANTES** de la eliminación del campo website:

```

318 -- Elimino el campo website de la tabla company
319 • show columns from company;
320

```

Result Grid

Filter Rows:

Export:

Wrap Cell Con

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	
website	varchar(255)	YES		NULL	

Tabla company **DESPUES** de la eliminacion del campo website:

alter table company

drop column website;

320

321 • show columns from company;

322

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	

Estructura de la tabla users

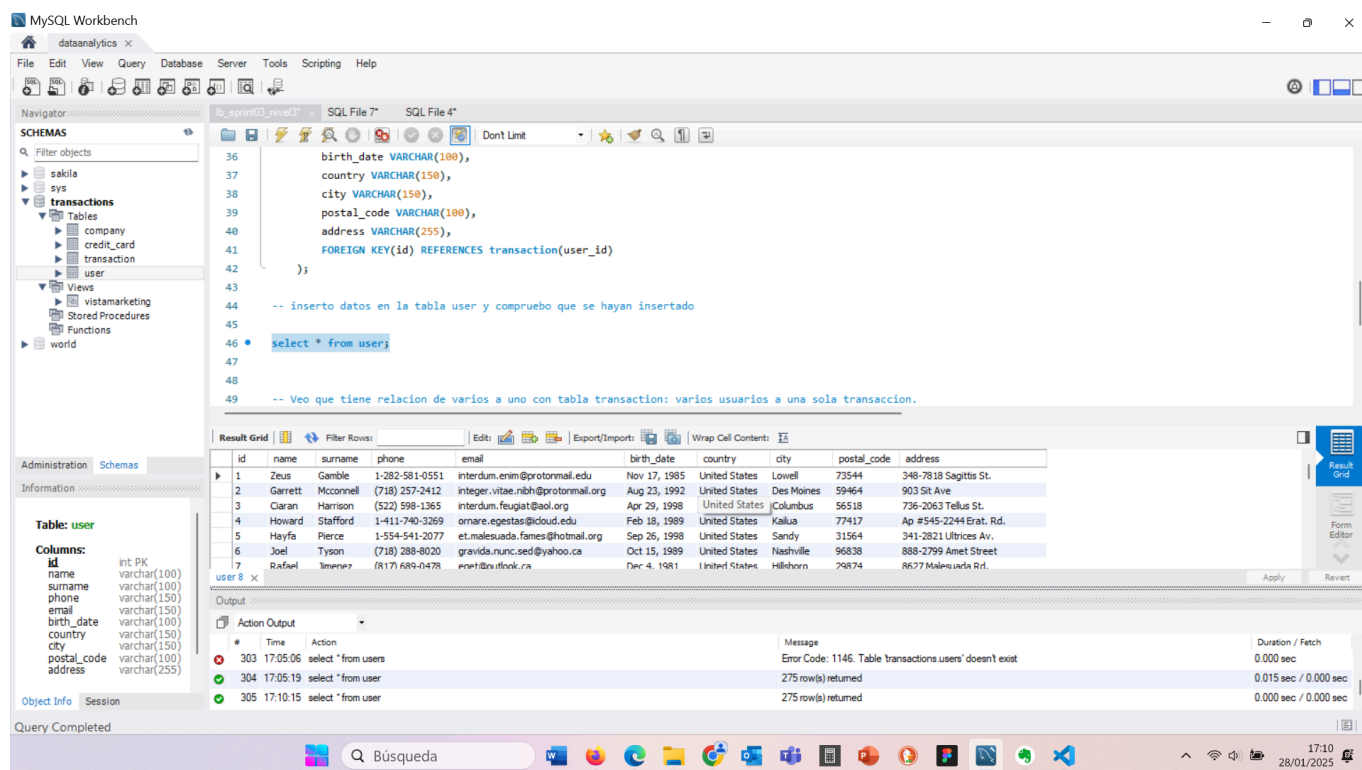
Primero, creo el index idx_user_id en la tabla transaction

```
CREATE INDEX idx_user_id ON transaction(user_id);
```

Creo la tabla users con el script del archivo estructura_datos_user.sql.

```
CREATE TABLE IF NOT EXISTS user (  
  id INT 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),  
  FOREIGN KEY(id) REFERENCES transaction(user_id)  
);
```

Inserto datos del archivo en la tabla user y compruebo que se hayan insertado



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```
36 birth_date VARCHAR(100),
37 country VARCHAR(150),
38 city VARCHAR(150),
39 postal_code VARCHAR(100),
40 address VARCHAR(255),
41 FOREIGN KEY(id) REFERENCES transaction(user_id)
42 );
43
44 -- inserto datos en la tabla user y compruebo que se hayan insertado
45
46 select * from user;
47
48
49 -- Veo que tiene relacion de varios a uno con tabla transaction: varios usuarios a una sola transaction.
```

The result grid shows the data for the 'user' table:

#	id	name	surname	phone	email	birth_date	country	city	postal_code	address
1	Zeus	Gamble		1-282-581-0551	interdum.enim@protonmail.edu	Nov 17, 1985	United States	Lowell	73544	348-7818 Sagitts St.
2	Garrett	Mcconnell		(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Des Moines	59464	903 5lt Ave
3	Cieran	Harrison		(322) 398-1365	interdum.feugiat@eol.org	Apr 29, 1998	United States	Columbus	56518	736-2063 Tellus St.
4	Howard	Stafford		1-411-740-3269	omare.egetas@icloud.edu	Feb 18, 1989	United States	Kakua	77417	Ap #545-2244 Erat. Rd.
5	Hayfa	Pierce		1-554-541-2077	et.malesuada.fames@hotmail.org	Sep 26, 1998	United States	Sandy	31564	341-2821 Ultrices Av.
6	Joel	Tyson		(718) 288-8020	gravida.nunc.sed@yahoo.ca	Oct 15, 1989	United States	Nashville	96838	888-2799 Amet Street
7	Rafael	Jimenez		(817) 693-0478	eneit@outlook.ca	Dec 4, 1981	United States	Hillbom	29874	8677 Malesuada Rd.

The output pane shows the following messages:

```
303 17:05:06 select * from users
Error Code: 1146. Table transactions.users doesn't exist
0.000 sec

304 17:05:19 select * from user
275 row(s) returned
0.015 sec / 0.000 sec

305 17:10:15 select * from user
275 row(s) returned
0.000 sec / 0.000 sec
```

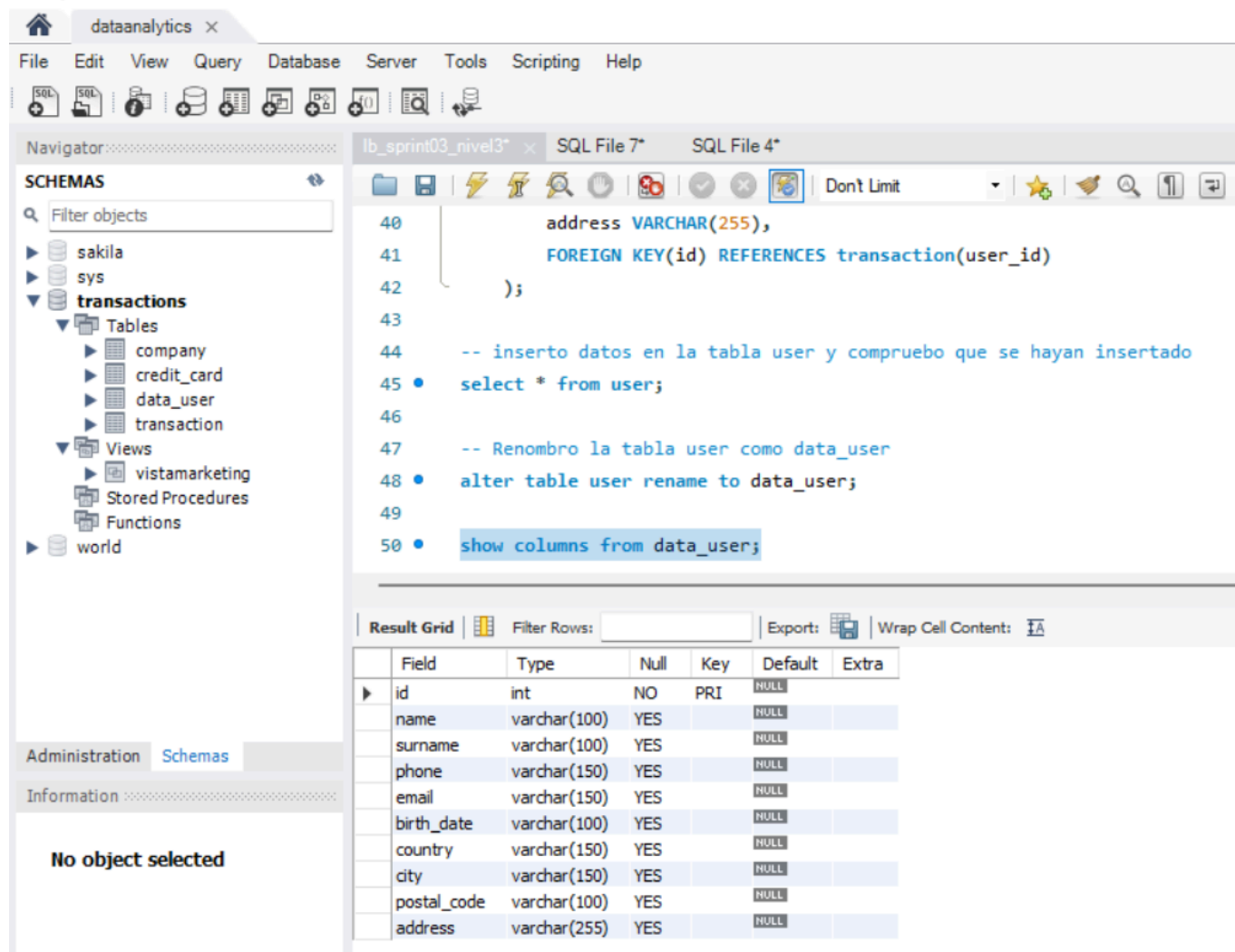
Renombro la tabla user como data_user

alter table user rename to data_user;

Ahora tengo que cambiar el nombre del campo email a personal_email en la tabla data_user

Campos de data_user ANTES de modificar nombre de campo email

MySQL Workbench



The screenshot shows the MySQL Workbench interface. On the left, the 'Navigator' pane displays the 'dataanalytics' database schema. The 'SCHEMAS' section is expanded, showing the 'transactions' database. Under 'Tables', the 'data_user' table is listed. The 'Result Grid' at the bottom shows the structure of the 'data_user' table.

SQL Query:

```
40         address VARCHAR(255),
41         FOREIGN KEY(id) REFERENCES transaction(user_id)
42     );
43
44     -- inserto datos en la tabla user y compruebo que se hayan insertado
45     • select * from user;
46
47     -- Renombro la tabla user como data_user
48     • alter table user rename to data_user;
49
50     • show columns from data_user;
```

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	
	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	

Renombro campo email como personal_email en la tabla data_user

alter table data_user rename column email to personal_email;

Campos de data_user DESPUES de modificar nombre de campo email

MySQL Workbench

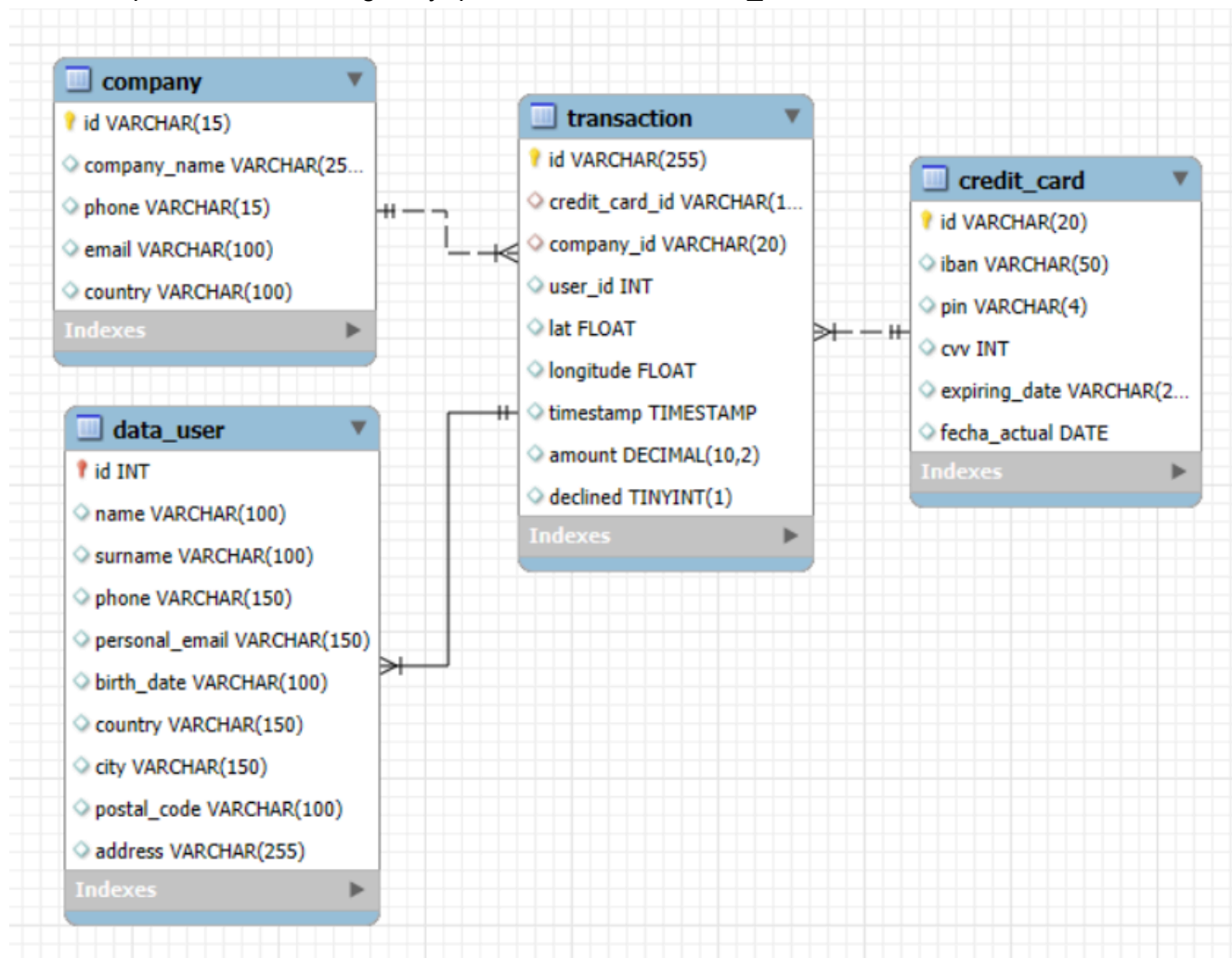
The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with a search filter 'Filter objects'. The 'transactions' schema is expanded, showing tables like 'company', 'credit_card', 'data_user', and 'transaction'. The main editor window shows a SQL script with the following queries:

```
45 • select * from user;
46
47 -- Renombro la tabla user como data_user
48 • alter table user rename to data_user;
49
50 • show columns from data_user;
51
52 -- Renombro campo email como personal_email en la tabla data_user y compru
53 • alter table data_user rename column email to personal_email;
54
55 • show columns from data_user;
```

The 'Result Grid' at the bottom displays the output of the last query, showing the columns of the 'data_user' table:

	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	

Veo el diagrama entidad relación de las tablas . Es incorrecto. La relación entre la tabla data_user y transaction no puede ser de varias a una. Una sola transacción no puede tener varios usuarios. Tenemos que eliminar la foreign key que está en la tabla data_user.



-- Tenemos que eliminar foreign key de la tabla data_user. Es incorrecta.
alter table data_user drop foreign key data_user_ibfk_1;

Luego compruebo en el script de la tabla data_user que efectivamente se eliminó esta foreign key.

MySQL Workbench

The screenshot shows the MySQL Workbench interface with the 'dataanalytics' database selected. The 'Navigator' pane on the left shows the 'transactions' database structure, including tables like 'company', 'credit_card', 'data_user', and 'transaction'. The main editor displays the DDL for 'transactions.data_user'.

```
DDL for transactions.data_user

1 CREATE TABLE `data_user` (
2   `id` int NOT NULL,
3   `name` varchar(100) DEFAULT NULL,
4   `surname` varchar(100) DEFAULT NULL,
5   `phone` varchar(150) DEFAULT NULL,
6   `personal_email` varchar(150) DEFAULT NULL,
7   `birth_date` varchar(100) DEFAULT NULL,
8   `country` varchar(150) DEFAULT NULL,
9   `city` varchar(150) DEFAULT NULL,
10  `postal_code` varchar(100) DEFAULT NULL,
11  `address` varchar(255) DEFAULT NULL,
12  PRIMARY KEY (`id`)
13 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

MySQL Workbench

The screenshot shows the MySQL Workbench interface with the 'dataanalytics' database selected. The main editor displays a series of SQL commands to rename a column and drop a foreign key.

```
51
52 -- Renombro campo email como personal_email en la tabla data_user y compruebo cambios
53 • alter table data_user rename column email to personal_email;
54
55 • show columns from data_user;
56
57 -- Tenemos que eliminar foreign key de la tabla data_user. Es incorrecta. Luego compruebo en su script.
58 • alter table data_user drop foreign key data_user_ibfk_1;
59
60 • show create table data_user;
```

The 'Result Grid' pane at the bottom shows the output of the 'show create table data_user;' command, displaying the full DDL for the 'data_user' table.

Table	Create Table
data_user	CREATE TABLE `data_user` (`id` int NOT NULL, `name` varchar(100) DEFAULT NULL, `surname` varchar(100) DEFAULT NULL, `phone` varchar(150) DEFAULT NULL, `personal_email` varchar(150) DEFAULT NULL, `birth_date` varchar(100) DEFAULT NULL, `country` varchar(150) DEFAULT NULL, `city` varchar(150) DEFAULT NULL, `postal_code` varchar(100) DEFAULT NULL, `address` varchar(255) DEFAULT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

The 'Administration' pane on the left shows the 'Schemas' tab, with the 'Table: data_user' selected. The 'Columns' section lists the columns: 'id' (int PK), 'name' (varchar(100)), and 'surname' (varchar(100)).

[illegible]

Modifico la table transaction para añadirle el id de la tabla data_user como Foreign Key

```
alter table transaction  
add constraint fk_user_id  
foreign key (user_id) references data_user(id);
```

Reviso en el script y si se ha creado.

MySQL Workbench

dataanalytics x

File Edit View Query Database Server Tools Scripting Help

Navigator: ib_sprint03_nivel3* SQL File 7* transactions.transaction x

SCHEMAS

Filter objects

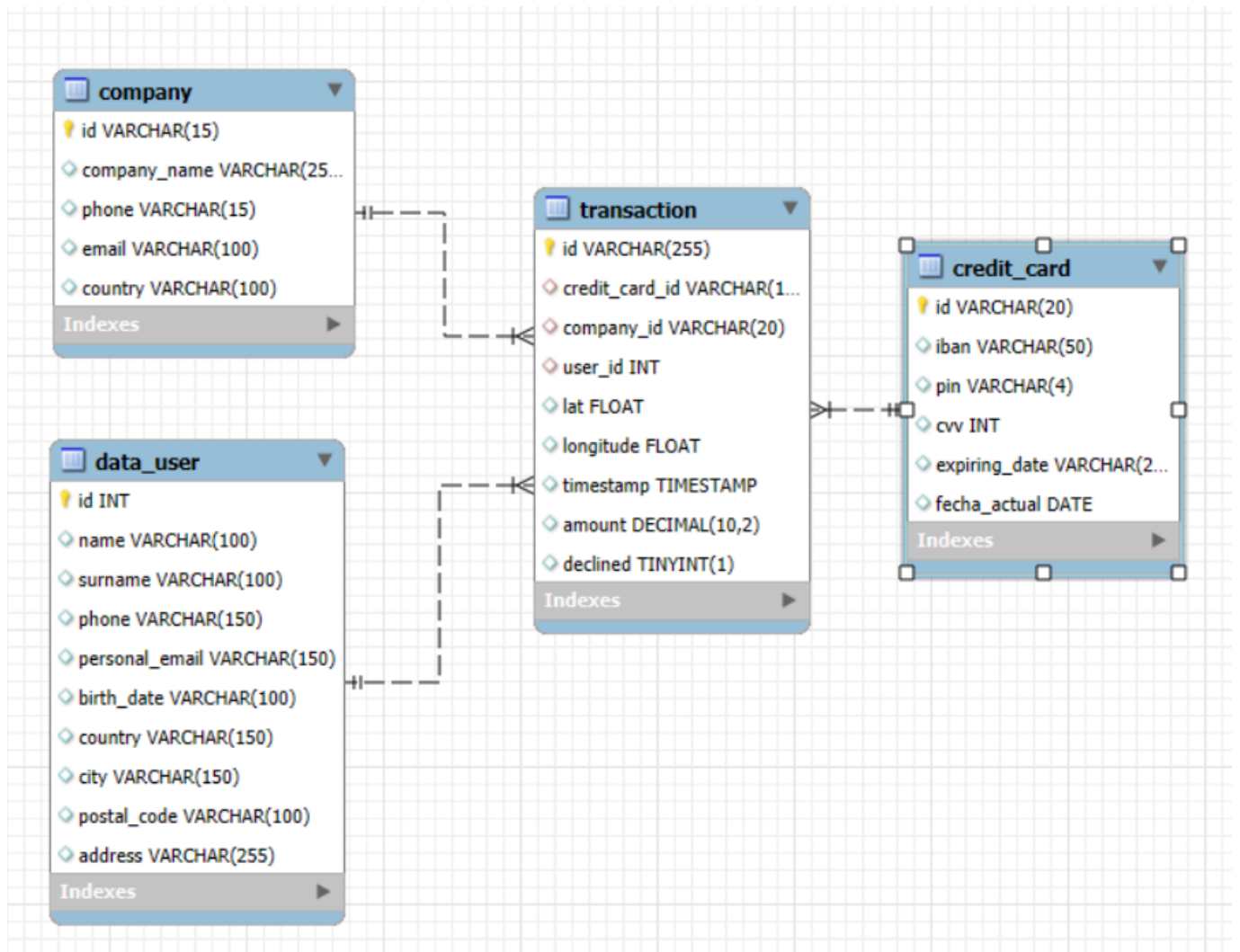
- sakila
- sys
- transactions
 - Tables
 - company
 - credit_card
 - data_user
 - transaction
 - Views
 - vistamarketing
 - Stored Procedures
 - Functions
- world

Administration Schemas Information

DDL for transactions.transaction

```
1 CREATE TABLE `transaction` (  
2   `id` varchar(255) NOT NULL,  
3   `credit_card_id` varchar(15) DEFAULT NULL,  
4   `company_id` varchar(20) DEFAULT NULL,  
5   `user_id` int DEFAULT NULL,  
6   `lat` float DEFAULT NULL,  
7   `longitude` float DEFAULT NULL,  
8   `timestamp` timestamp NULL DEFAULT NULL,  
9   `amount` decimal(10,2) DEFAULT NULL,  
10  `declined` tinyint(1) DEFAULT NULL,  
11  PRIMARY KEY (`id`),  
12  KEY `company_id` (`company_id`),  
13  KEY `credit_card_id` (`credit_card_id`),  
14  KEY `idx_user_id` (`user_id`),  
15  CONSTRAINT `fk_user_id` FOREIGN KEY (`user_id`) REFERENCES `data_user` (`id`),  
16  CONSTRAINT `transaction_ibfk_1` FOREIGN KEY (`company_id`) REFERENCES `company` (`id`),  
17  CONSTRAINT `transaction_ibfk_2` FOREIGN KEY (`credit_card_id`) REFERENCES `credit_card` (`id`)  
18 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Voy a ver el diagrama entidad relación para comprobar que todo esté OK



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.

```
create view informetecnico as
select t.id as transaction_id, du.name , du.surname, cc.iban, c.company_name
from transaction as t
join company as c
on c.id = t.company_id
join data_user as du
on du.id = t.user_id
join credit_card as cc
on cc.id = t.credit_card_id
order by t.id desc;
```