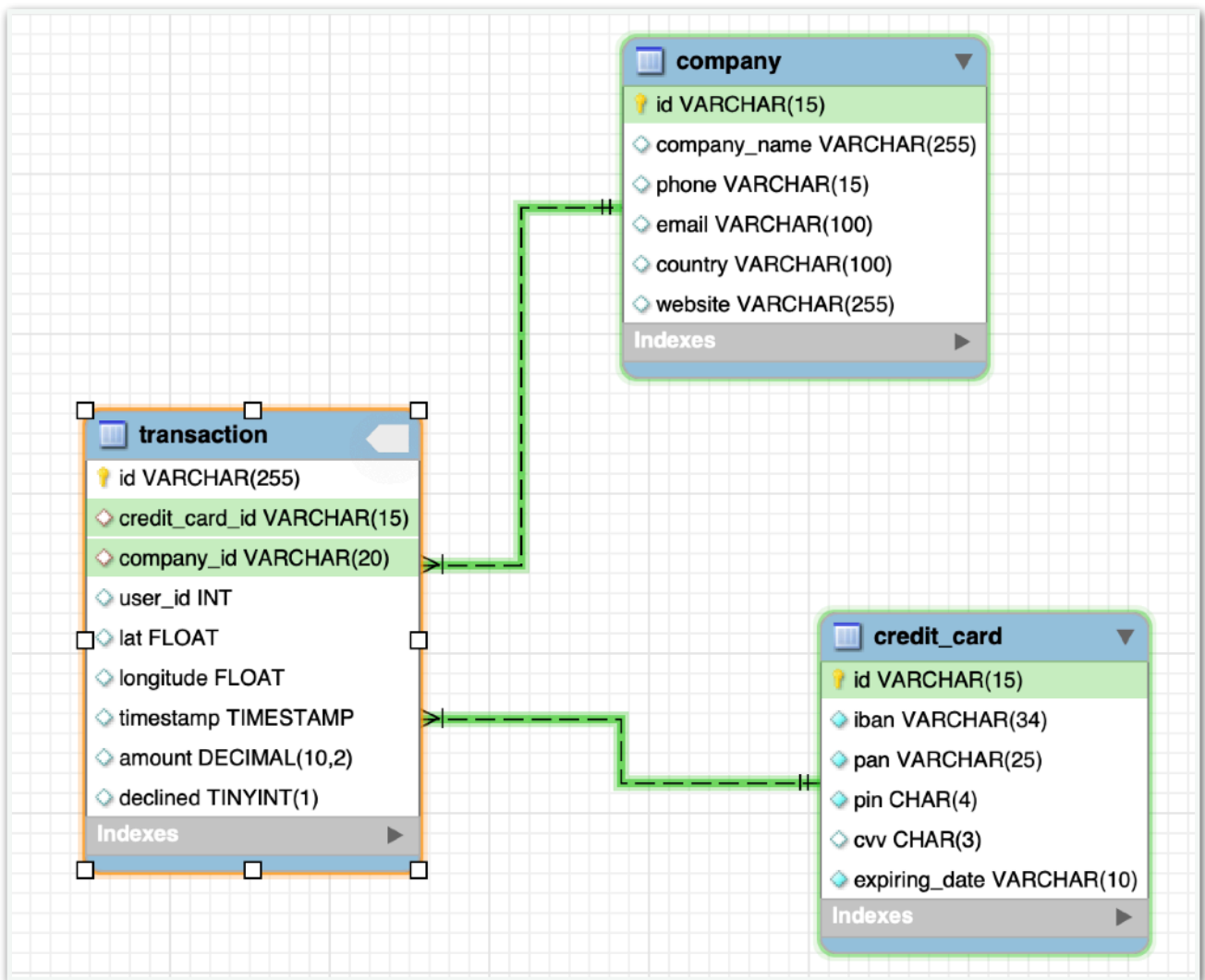


Patricia Proniewska

## Tarea S3.01. Manipulación de tablas

### Nivel 1

#### Ejercicio 1



La base de datos **transactions** está formada por **tres tablas** relacionadas en un esquema tipo relacional. Las relaciones son de uno a muchos (1:N): una empresa puede tener muchas transacciones, pero cada transacción pertenece a una sola empresa; y una tarjeta de crédito puede estar asociada a muchas transacciones, pero cada transacción está vinculada a una sola tarjeta de crédito.

En este esquema, la variable **company.id** es la **Primary Key (PK)** de la tabla **company**. Es única para cada empresa y sirve como su identificador principal.

La variable **credit\_card.id** es la **Primary Key (PK)** de la tabla **credit\_card**, que identifica de forma única cada tarjeta de crédito registrada.

Las variables **transaction.company\_id** y **transaction.credit\_card\_id** son **Foreign Keys (FK)** que establecen las relaciones entre las tablas:

- **transaction.company\_id** hace referencia a **company.id** para indicar a qué empresa pertenece cada transacción.
- **transaction.credit\_card\_id** hace referencia a **credit\_card.id** para indicar con qué tarjeta de crédito se realizó la transacción.

**Tabla company:** guarda la información de cada empresa.

- id - identifica de forma única a cada empresa (PK) / VARCHAR(15)
- company\_name - nombre / VARCHAR(255)
- phone - teléfono / VARCHAR(15)
- email - correo electrónico / VARCHAR(100)
- country - país donde de la empresa / VARCHAR(100)
- website - página web / VARCHAR(255)

**Tabla transaction:** registra cada operación realizada con las empresas.

- id - identifica de forma única cada transacción (PK) / VARCHAR(255)
- credit\_card\_id - identificador de la tarjeta de crédito (FK) / VARCHAR(15)
- company\_id - identificador de la empresa asociada a la transacción (FK) / VARCHAR (20)
- user\_id - identificador para el usuario que realizó la transacción / INT
- lat - latitud de la ubicación de la transacción / FLOAT
- longitude - longitud de la ubicación de la transacción / FLOAT
- timestamp - fecha y hora de la transacción / TIMESTAMP
- amount - cantidad de dinero en la transacción / DECIMAL
- declined - indica si la transacción fue rechazada / TINYINT(1)

**Tabla credit\_card:** Almacena los datos esenciales de las tarjetas de crédito.

- id – identifica de forma única cada tarjeta (PK) / VARCHAR(15)
- iban – número IBAN de la cuenta / VARCHAR(34)

- pan – número de la tarjeta / VARCHAR(25)
- pin – código PIN de seguridad / CHAR(4)
- cvv – código de verificación / CHAR(3)
- expiring\_date – fecha de caducidad de la tarjeta / VARCHAR(10)

```

-- Creación de la tabla credit_card
CREATE TABLE credit_card(
    id VARCHAR(15) PRIMARY KEY,
    iban VARCHAR(34) NOT NULL,
    pan VARCHAR(25) NOT NULL,
    pin CHAR(4) NOT NULL,
    cvv CHAR(3),
    expiring_date VARCHAR(10) NOT NULL
);

-- Creación de las relaciones entre las tablas
• ALTER TABLE transaction
  ADD CONSTRAINT FK_transaction_creditcard
  FOREIGN KEY (credit_card_id)
  REFERENCES credit_card(id);

• ALTER TABLE transaction
  ADD CONSTRAINT FK_Transaction_Company
  FOREIGN KEY (company_id)
  REFERENCES company(id);

```

100% 18:33

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

id	iban	pin	cvv	expiring_date
CcS-4859	XX7826930491423553609370	4983	277	11/26/26
CcS-4860	XX5559590368835304645299	6876	661	07/27/27
CcS-4861	XX2035182877195191627307	5710	398	04/25/26
CcS-4862	XX4774721462463645409758	4042	174	11/27/26
CcS-4863	XX1476829664245046207111	5969	449	12/27/29
CcS-4864	XX8380298893385731196159	8481	139	02/28/26

credit\_card 20

Action Output

Time	Action
50... 10:51:07	SELECT * FROM credit_card

## 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.

```

39 • UPDATE credit_card
40 SET credit_card.iban = 'TR323456312213576817699999'
41 WHERE credit_card.id = 'CcU-2938';
42
43 • SELECT *
44 FROM credit_card
45 WHERE credit_card.id = 'CcU-2938';
46
100% 35:45

```

**Result Grid** Filter Rows: Search Edit: Export/Import:

id	iban	pin	cvv	expiring_date
CcU-2938	TR323456312213576817699999	3257	984	10/30/22
NULL	NULL	NULL	NULL	NULL

credit\_card 21

**Action Output**

	Time	Action	Response
50...	10:53:03	SELECT * FROM credit_card WHERE credit_card.id = 'CcU-2938'	1 row(s) returned

## Ejercicio 3

En la tabla "transaction" ingresa una nueva transacción.

```

50 -- 1. Comprobar si existen los registros de la tarjeta y la empresa
51 • SELECT * FROM credit_card WHERE id = 'CcU-9999';
52 • SELECT * FROM company WHERE id = 'b-9999';
53
54 -- 2. Crear la empresa asociada a la transacción (si no existe - como en nuestro caso)
55 • INSERT INTO company (id, company_name, phone, email, country, website)
56 VALUES ('b-9999', 'Test Company', '00000000', 'test@test.com', 'Spain', 'www.test.com');
57
58 -- 3. Crear la tarjeta de crédito de la transacción (si no existe - como en nuestro caso)
59 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)
60 VALUES ('CcU-9999', 'TEST123456789', '9999888877776666', '1234', '999', '12/30/30');
61
62 -- 4. Insertar la nueva transacción en la tabla "transaction"
63 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
64 VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, 111.11, 0);
65
66 • SELECT *
67 FROM transaction
68 WHERE transaction.id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD';
69
100% 63:68

```

**Result Grid** Filter Rows: Search Edit: Export/Import:

id	credit_card...	company_id	user_id	lat	longitude	timestamp	amount	declined
----	----------------	------------	---------	-----	-----------	-----------	--------	----------

```

63  /*
64  En un primer momento había creado la empresa y la tarjeta de crédito con datos aleatorios inventados
65  para poder insertar la nueva transacción. Más adelante haciendo el P2P me di cuenta de que no se deben
66  introducir datos falsos en una base de datos, y que los valores desconocidos se
67  registran como NULL.
68
69  Por eso, modifiqué la estructura de la tabla credit_card para permitir valores NULL
70  en las columnas necesarias, y en lugar de crear nuevos registros, actualicé los existentes
71  utilizando UPDATE para dejar los campos en NULL. Por eso, en esta captura se muestran
72  comandos UPDATE en lugar de INSERT, reflejando la corrección del error anterior.
73  */
74
75  • DESCRIBE company;
76  • DESCRIBE credit_card;
77
78  -- que creado variables que no permiten NULL en tabla credit_card. Voy a arreglarlo:
79
80  • ALTER TABLE credit_card
81    MODIFY iban VARCHAR(34) NULL,
82    MODIFY pin CHAR(4) NULL,
83    MODIFY expiring_date VARCHAR(10) NULL;

```

```

-- 2. Actualizar la empresa asociada a la transacción (solo dejar el ID y resto NULL)
UPDATE company
SET company_name = NULL,
    phone = NULL,
    email = NULL,
    country = NULL,
    website = NULL
WHERE id = 'b-9999';

-- 3. Actualizar la tarjeta de crédito de la transacción (solo dejar el ID y resto NULL)
• UPDATE credit_card
  SET iban = NULL,
      pin = NULL,
      cvv = NULL,
      expiring_date = NULL
  WHERE id = 'CcU-9999';

• DESCRIBE company;
• DESCRIBE credit_card;

```

100% 18:103

Result Grid



Filter Rows:

Export:

	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	

```

110  -- Comprobar resultado final:
111
112  • SELECT *
113  FROM transaction
114  WHERE transaction.id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD';
115

```

100% 63:114

Result Grid Filter Rows: Search Edit: Export/Import:

id	credit_card...	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
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 66

Action Output

Time	Action	Response
5171 11:35:57	MySQLWorkbench in WHERE transaction.id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD'	1 row(s) returned

## Ejercicio 4

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

```

119 • ALTER TABLE credit_card
120 DROP COLUMN pan;
121
122 • DESCRIBE credit_card;
123

```

100% 22:122

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
▶ id	varchar(15)	NO	PRI	NULL	
iban	varchar(34)	YES		NULL	
pin	char(4)	YES		NULL	
cvv	char(3)	YES		NULL	
expiring_date	varchar(10)	YES		NULL	
fecha_actual	date	YES		NULL	

Result 67

Action Output

Time	Action	Response
5172 11:41:25	DESCRIBE credit_card	6 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.



## Ejercicio 3

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

```
164 • SELECT *
165 FROM vista_marketing
166 WHERE vista_marketing.pais = 'Germany'
167 ORDER BY nombre_compania ASC;
168
```

100% 30:167

Result Grid Filter Rows: Search Export:

nombre_compania	telefono	pais	promedio_compra
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.867160
Ac Industries	09 34 65 40 60	Germany	255.147288
Aliquam PC	01 45 73 52 16	Germany	253.136923
Auctor Mauris Corp.	05 62 87 14 41	Germany	254.765518
Augue Foundation	06 88 43 15 63	Germany	253.505000
Convallis In Incorporated	06 66 57 29 50	Germany	257.745376
Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.319156
Rutrum Non Inc.	02 66 31 61 09	Germany	255.136927

vista\_marketing 69

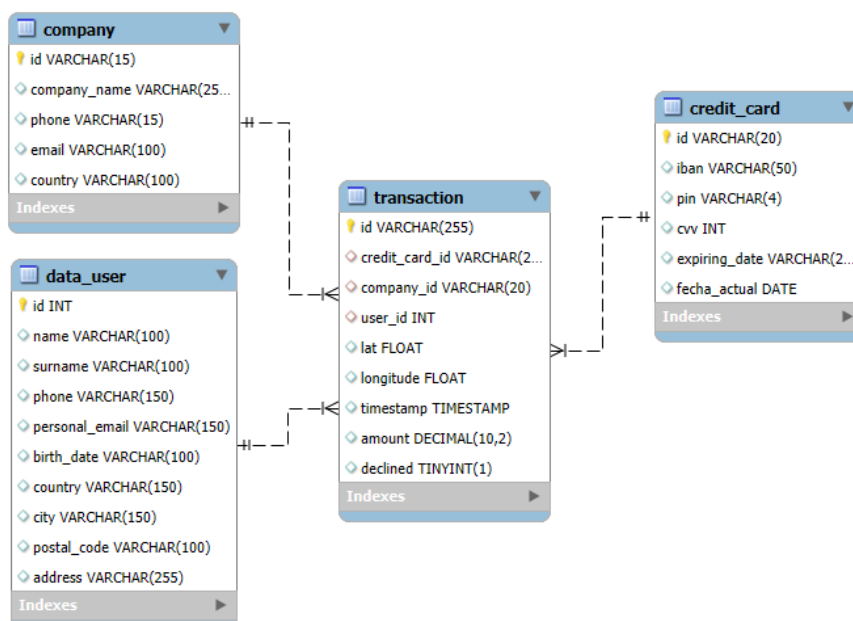
Action Output

Time	Action	Response
5174 12:35:51	SELECT * FROM vista_marketing WHERE vista_marketing.pais = 'Germany' ORDER BY nombre_compania ASC	8 row(s) returned

## Nivel 3

### Ejercicio 1

La próxima semana tendrás una nueva reunión con los gerentes de marketing. Un compañero de tu equipo realizó modificaciones en la base de datos, pero no recuerda cómo las realizó. Te pide que le ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama:





```

178 -- 1. Crear la tabla "user":
179 • CREATE TABLE IF NOT EXISTS user (
180     id CHAR(10) PRIMARY KEY,
181     name VARCHAR(100),
182     surname VARCHAR(100),
183     phone VARCHAR(150),
184     email VARCHAR(150),
185     birth_date VARCHAR(100),
186     country VARCHAR(150),
187     city VARCHAR(150),
188     postal_code VARCHAR(100),
189     address VARCHAR(255)
190 );
191
192 -- 2. Comprobar y modificar el tipo de datos de transaction.user_id (FK).
193 -- Columnas: transaction.user_id y user.id, deben tener el mismo tipo de dato como el el diagrama: (CHAR(10))
194 -- para que la relación FK funcione correctamente.
195
196 • ALTER TABLE transaction
197   MODIFY COLUMN user_id CHAR(10);
198
199 -- 3. Renombrar tabla 'user' to 'data_user':
200 • RENAME TABLE user TO data_user;
201
202 -- 4. Abrir el archivo "datos introducir sprint3 user.sql"
203 -- Insertar los datos del archivo "datos_introducir_user.sql" y comprobar el resultado:
204
205 • SELECT *
206   FROM data_user;

```

100% 16:206

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

id	name	surname	phone	personal_email	birth_date	country	city	postal_code	address
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nipn@protonmail.org	Aug 23, 1992	United States	Philadelphia	19101	903 Sit Ave
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Houston	77001	736-2063 Tellus St.
4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.edu	Feb 18, 1989	United States	Phoenix	85001	Ap #545-2244 Erat. Rd.
5	Hayla	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	Sep 26, 1998	United States	Philadelphia	19101	341-2821 Ultrices Av.
6	Joel	Tyson	(718) 288-8020	gravida.nunc.sed@yahoo.ca	Oct 15, 1989	United States	San Jose	95101	688-2799 Amet Street
7	Rafael	Jimenez	(817) 689-0478	en@outlook.ca	Dec 4, 1981	United States	Chicago	60601	8627 Malesuada Rd.

data\_user 7

Action Output

Time	Action	Response
15 13:37:45	SELECT * FROM data_user	5001 row(s) returned

```

208 -- 5. Preparar los datos para crear las relacion (FK) entre tablas:
209
210 -- Comprobar si hay usuarios que existen en tabla TRANSACTION y no existan en tabla USER
211
212 • SELECT DISTINCT user_id
213   FROM transaction
214  WHERE user_id IS NOT NULL
215  AND user_id NOT IN (SELECT id FROM user);
216
217 -- Añadir los datos de prueba de usuario que faltaba:
218
219 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address)
220   VALUES ('9999', 'testname', 'testsurname', '000000000', 'user@test.com', '2001-01-01', 'Spain', 'Barcelona', '08000', 'Poblenou Street 123');
221
222 -- 6. Crear relacion (FK) entre las tablas (después de crear el usuario que faltaba, podemos establecer la FK sin errores.):
223
224 • ALTER TABLE transaction
225   ADD CONSTRAINT FK_Transaction_user
226   FOREIGN KEY (user_id)
227   REFERENCES user(id);
228
229 • DESCRIBE transaction;
230

```

100% 22:229

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
company_id	varchar(20)	YES	MUL	NULL	
user_id	int	YES	MUL	NULL	
lat	float	YES		NULL	
longitude	float	YES		NULL	
timestamp	timestamp	YES		NULL	
amount	decimal(10,2)	YES		NULL	
declined	tinyint(1)	YES		NULL	

Result 9

Action Output

Time	Action	Response
17 13:45:19	DESCRIBE transaction	9 row(s) returned

```

-- 7. Quitar la columna "website" de la tabla company:
232
233 • ALTER TABLE company
234 DROP COLUMN website;
235
236 -- Comprobar que la columna se quitó bien
237 • DESCRIBE company;
238
239 -- 8. Añadir columna nueva "fecha_actual" en la tabla credit_card (tipo DATE)
240
241 • ALTER TABLE credit_card
242 ADD COLUMN fecha_actual DATE;
243
244 -- Comprobar que la columna se añadió bien
245 • DESCRIBE credit_card;
246
247 -- 9. Otra forma de cambiar el nombre de la columna "email" a "personal_email" en la tabla data_user
248
249 • ALTER TABLE data_user
250 RENAME COLUMN email TO personal_email;
251
252 -- Comprobar que el cambio se hizo bien
253 • DESCRIBE data_user;

```

100% 20:253

Result Grid						
Filter Rows:		Search		Export:		
Field	Type	Null	Key	Default	Extra	
id	int	NO	PRI			
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		
Result 10						
Action Output						
	Time	Action				Response
18	13:47:10	DESCRIBE data_user				10 row(s) returned

```

255 -- 10. Cambiar los tipos y longitudes de datos para que sean iguales al diagrama deseado:
256
257 -- Me di cuenta de que antes cometi un error antes creando la relación.
258 -- Puse la columna "id" en data_user como CHAR(10), pero debería ser INT.
259 -- Por eso, ahora tengo que soltar la FK, cambiar el tipo y volver a crearla.
260
261 -- Quitar la Foreign Key para poder cambiar el tipo de datos
262 • ALTER TABLE transaction
263 DROP FOREIGN KEY FK_Transaction_user;
264
265 -- Cambiar el tipo del id de data_user a INT
266 • ALTER TABLE data_user
267 MODIFY id INT NOT NULL;
268
269 -- Cambiar también el tipo de user_id en transaction a INT (debe ser igual)
270 • ALTER TABLE transaction
271 MODIFY user_id INT;
272
273 -- Volver a crear la Foreign Key correctamente con el tipo INT
274 • ALTER TABLE transaction
275 ADD CONSTRAINT FK_Transaction_user
276 FOREIGN KEY (user_id)
277 REFERENCES data_user(id);
278
279 -- Comprobar que todo está correcto
280 • DESCRIBE data_user;
281 • DESCRIBE transaction;

```

100% 22:281

Result Grid						
Filter Rows:		Search		Export:		
Field	Type	Null	Key	Default	Extra	
id	varchar(255)	NO	PRI	NULL		
credit_card_id	varchar(20)	YES	MUL	NULL		
company_id	varchar(20)	YES	MUL	NULL		
user_id	int	YES	MUL	NULL		
lat	float	YES		NULL		
longitude	float	YES		NULL		
timestamp	timestamp	YES		NULL		
amount	decimal(10,2)	YES		NULL		
declined	tinyint(1)	YES		NULL		
Result 12						
Action Output						
	Time	Action				Response
20	13:50:03	DESCRIBE transaction				9 row(s) returned

```

284 -- Cambiar el tipo de la columna "id" en la tabla data_user a INT (clave principal)
285 • ALTER TABLE data_user
286   MODIFY id INT;
287
288 -- 11. Eliminar las vistas:
289 • DROP VIEW IF EXISTS informe_tecnico;
290 • DROP VIEW IF EXISTS vista_marketing;
291
292
293 -- 12. Modificar los tipos de datos en tabla 'credit_card' para que coincidan con diagrama deseado:
294 • ALTER TABLE credit_card
295   MODIFY COLUMN id VARCHAR(20),
296   MODIFY COLUMN iban VARCHAR(50),
297   MODIFY COLUMN pin VARCHAR(4),
298   MODIFY COLUMN cvv INT,
299   MODIFY COLUMN expiring_date VARCHAR(20),
300   MODIFY COLUMN fecha_actual DATE;
301
302 -- Comprobar el resultado:
303 • DESCRIBE credit_card;
304
305 -- 13. Modificar los tipos de datos en tabla 'transaction' para que coincidan con diagrama deseado:
306 • ALTER TABLE `transaction`
307   MODIFY COLUMN credit_card_id VARCHAR(20);
308
309 -- Comprobar el resultado:
310 • DESCRIBE transaction;
311
312 -- 14. Visualizar datos finales:
313 • SELECT *
314   FROM data_user;

```

100% 16:314

Result Grid

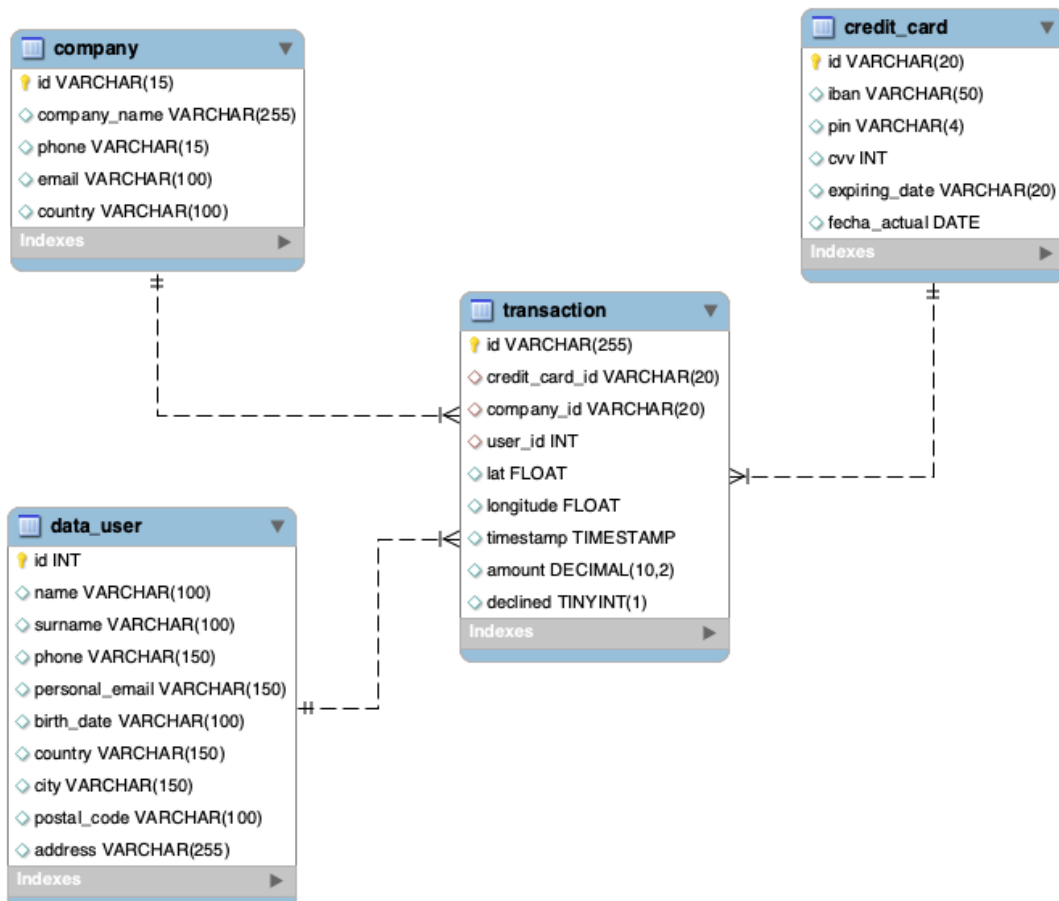
id	name	surname	phone	personal_email	birth_date	country	city	postal_code	address
3	Claran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Houston	77001	736-2063 Tellus St.
4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.edu	Feb 18, 1989	United States	Phoenix	85001	Ap #545-2244 Erat. Rd.
5	Hayla	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	Sep 26, 1998	United States	Philadelphia	19101	341-2821 Ultrices Av.

data\_user 16

Action Output

	Time	Action	Response
✓ 24	13:54:47	SELECT * FROM data_user	5001 row(s) returned

Crear diagrama EER para comprobar los datos finales con el diagrama deseado:



## Ejercicio 2

La empresa también le pide crear una vista llamada "InformeTecnico" que contenga la siguiente información:

- ID de la transacción
- Nombre del usuario/a
- Apellido del usuario/a
- IBAN de la tarjeta de crédito usada.
- Nombre de la compañía de la transacción realizada.
- Asegúrese de incluir información relevante de las tablas que conocerá y utilice alias para cambiar de nombre columnas según sea necesario.

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

```
320 • CREATE OR REPLACE VIEW informe_tecnico AS
321 SELECT
322     transaction.id AS ID_transaccion,
323     data_user.name AS Nombre_usuario,
324     data_user.surname AS Apellido_usuario,
325     credit_card.iban AS IBAN_tarjeta,
326     company.company_name AS Nombre_compania
327 FROM transaction
328 JOIN data_user
329     ON transaction.user_id = data_user.id
330 JOIN credit_card
331     ON transaction.credit_card_id = credit_card.id
332 JOIN company
333     ON transaction.company_id = company.id
334 WHERE transaction.declined = 0;
335
336 -- Mostrar resultados de la vista
337 • SELECT *
338 FROM informe_tecnico
339 ORDER BY ID_transaccion DESC;
```

100% 30:339

Result Grid					
Filter Rows: Search					
Export: Fetch rows:					
ID_transaccion	Nombre_usuario	Apellido_usuario	IBAN_tarjeta	Nombre_compania	
FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	Securp	Faolvqly	XX162677143304223631437567	Nunc Interdum Incorporated	
FFFB270D-F53A-4D5D-9666-E5307C53CC84	Ggzjpa	Uirzjulh	XX395114267082019952567052	Viverra Donec Foundation	
FFF9E3CE-234E-408C-A8EF-F9CAD577224A	Yshimq	Zpsjsleed	XX8845462156537570367941	Convallis In Incorporated	
FFF9E178-6CD2-4DF9-99B0-49AE068809B1	Jovepx	Xwcwzwnm	XX321405515711654384711481	Mus Aenean Eget Foundation	
FFF867C9-17B5-4B1F-AFD9-F8023AAA449E	Fqjngd	Lvhfqyxi	XX278446342932680979729426	Cras Vehicula Aliquet Industries	
FFF7042D-18C6-4DDD-823C-4D90A4AC8F26	Njoraa	Egsqucui	XX405009272572550082027209	Placerat LLP	
FFF660D4-4244-47F6-9210-E5D1DCB99DB0	Lopzaj	Itgrylay	XX63376659736627454015125	Pede Cum Ltd	
FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	Securp	Faolvqly	XX162677143304223631437567	Nunc Interdum Incorporated	
informe_tecnico 18					
Action Output					
Time Action Response					
35	14:03:55	SELECT * FROM informe_tecnico ORDER BY ID_transaccion DESC			99763 row(s) returned