

Nivell 1

Exercici 1

La teva tasca és dissenyar i crear una taula anomenada "credit_card" que emmagatzemi detalls crucials sobre les targetes de crèdit. La nova taula ha de ser capaç d'identificar de manera única cada targeta i establir una relació adequada amb les altres dues taules ("transaction" i "company"). Després de crear la taula serà necessari que ingressis la informació del document denominat "dades_introduir_credit". Recorda mostrar el diagrama i realitzar una breu descripció d'aquest.

The screenshot shows a database management interface with a sidebar on the left displaying a tree view of the database structure. The main area displays SQL code for creating and altering the 'credit_card' table. The code includes a 'CREATE TABLE' statement with columns: id (varchar(20) NOT NULL PRIMARY KEY), iban (varchar(40)), pan (varchar(20)), pin (varchar(10)), cvv (varchar(4)), and expiring_date (date). It also includes an 'ALTER TABLE' statement to change the 'expiring_date' column to 'VARCHAR(15)' and a 'SELECT*' query to verify the data.

```
5 CREATE TABLE credit_card (  
6   id varchar(20) NOT NULL PRIMARY KEY,  
7   iban varchar(40),  
8   pan varchar(20),  
9   pin varchar(10),  
10  cvv varchar(4),  
11  expiring_date date  
12 );  
13  
14 -- Como al intentar cargar los datos en la tabla nos da error por el formato de la fecha cambio la variable  
15 ALTER TABLE transactions.credit_card  
16 CHANGE COLUMN expiring_date  
17 expiring_date VARCHAR(15);  
18  
19 SELECT*  
20 FROM credit_card; -- compruebo que los datos se han cargado correctamente esta vez
```

The 'Object Info' panel on the left shows the table structure for 'credit_card':

Columns:	
id	varchar(20) PK
iban	varchar(40)
pan	varchar(20)
pin	varchar(10)
cvv	varchar(4)
expiring_date	varchar(15)

The 'Result Grid' shows the data loaded into the table:

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TF301950312213576817638661	5424465566813633	3257	984	10/30/22
CcU-2945	DO26854763748537475216568689	5142423821948828	9080	887	08/24/23
CcU-2952	BQ459VQL5271052568255	4556 453 55 5287	4598	438	06/29/21

The 'Action Output' panel shows the execution of the 'SELECT*' query, returning 275 rows.

- Para poder introducir los datos en la tabla he tenido que cambiar el tipo de dato de la columna 'expiring_date' de DATE a VARCHAR (porque la fecha estaba en formato americano y MySQL no la reconoce como fecha).
- Posteriormente, intenté un cambio de formato de las fechas para que MySQL las aceptara y poder trabajar con ellas en formato DATE pero antes necesité deshabilitar momentáneamente las foreign keys y la opción de actualización segura, que habilité inmediatamente después de ejecutar la query porque esto no es recomendable tenerlo desactivado ya que podríamos afectar la base de datos completa de forma irreversible.

The screenshot shows the same database management interface as before, but with SQL code for disabling foreign keys and safe updates, converting the 'expiring_date' column to DATE, and then re-enabling the constraints.

```
26 SET FOREIGN_KEY_CHECKS=0; -- para desactivar las foreign keys  
27 SET SQL_SAFE_UPDATES=0; -- para deshabilitar la opción de actualización segura  
28  
29 -- Ahora cambio los valores de la fecha al modelo de fecha que SQL acepta para luego cambiar el tipo de variable:  
30 UPDATE transactions.credit_card  
31 SET expiring_date = DATE_FORMAT(STR_TO_DATE(expiring_date, '%m/%d/%Y'), '%Y-%m-%d') -- str_to_date cambia las fechas de formato string a date  
32 WHERE credit_card.id IS NOT null;  
33  
34 SET FOREIGN_KEY_CHECKS=1; -- vuelvo a activar las foreign keys  
35 SET SQL_SAFE_UPDATES=1; -- vuelvo a habilitar la opción de actualización segura  
36  
37 -- Comprobamos que los valores se han cambiado correctamente:  
38 SELECT*  
39 FROM credit_card;  
40  
41 -- Vuelvo a cambiar el tipo de variable a fecha:  
42 ALTER TABLE transactions.credit_card  
43 CHANGE COLUMN expiring_date  
44 expiring_date date;
```

The 'Object Info' panel on the left shows the updated table structure for 'credit_card':

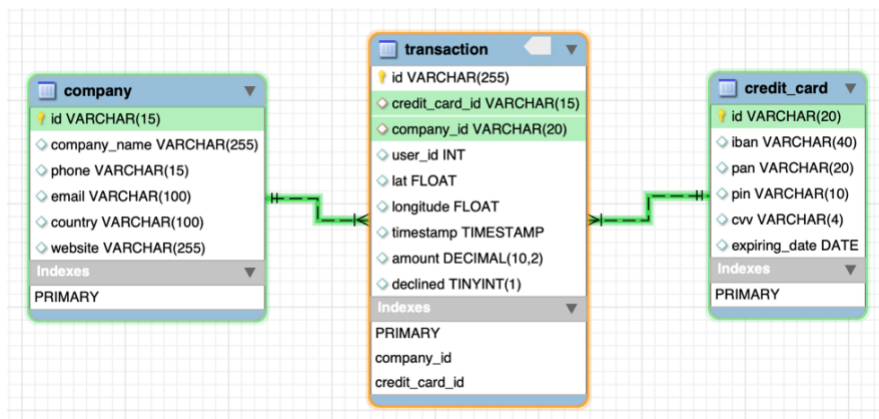
Columns:	
id	varchar(20) PK
iban	varchar(40)
pan	varchar(20)
pin	varchar(10)
cvv	varchar(4)
expiring_date	date

The 'Action Output' panel shows the execution of the 'ALTER TABLE' query, affecting 275 rows.

- Una vez cambiado el formato de las fechas volví a cambiar el tipo de dato de la columna 'expiring_date' de VARCHAR a DATE.
- Por último, relaciono las tablas 'credit_card' y 'transactions' a través de la PK (id) de 'credit_card' que se relaciona con la FK (credit_card_id) de 'transaction' que hemos de crear:



- Obtengo el siguiente diagrama:



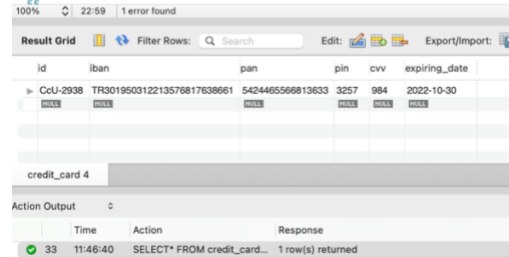
- En el diagrama observamos que la tabla de hechos sería 'transaction' y las tablas de dimensiones 'company' y 'credit_card'. La relación entre las tres tablas se establece a través de las Primary Keys de las tablas de dimensiones (id de 'company' e id de 'credit_card') y las Foreign Keys de la tabla de hechos: company_id que se relaciona con id de 'company' y credit_card_id que se relaciona con id de 'credit_card'.
- En el caso de nuestra nueva tabla 'credit_card' he definido el campo **id** como VARCHAR (20) por su relación con el campo credit_card_id de 'transaction' que también está definido de esa forma y el campo **expiring_date** como DATE ya que se trata de fechas. El **IBAN** (International Bank Account Number), utilizado para identificar de forma estandarizada cualquier cuenta sea del país que sea, es una serie alfanumérica formada por un máximo de 34 caracteres dependiendo del país, por eso lo he definido como VARCHAR (40) para darle algo de margen. El **CVV** o código de verificación de la tarjeta es un código de tres o cuatro números utilizado para verificar que el usuario de la compra es el poseedor de la tarjeta por eso lo he definido como VARCHAR (4) porque es el máximo número de caracteres que puede aceptar. El **PIN** es el código de seguridad de la tarjeta que he definido como VARCHAR (10) para darle margen, aunque suelen ser de 4 a 6 números. Por último, el **PAN** (Primary account number) que se refiere a la fila de números que aparece en la parte delantera de la tarjeta y está compuesto por entre 12 y 19 dígitos, por eso lo he definido como VARCHAR (20).

Exercici 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.

- Primero observo los datos del usuario con id: CcU-2938.

```
52 • SELECT*
53 FROM credit_card
54 WHERE id = 'CcU-2938'; -- primero observo los datos del usuario
```



id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	2022-10-30

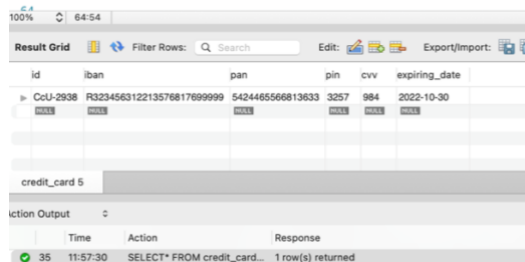
credit_card 4

Action Output

Time	Action	Response
33 11:46:40	SELECT* FROM credit_card...	1 row(s) returned

- Posteriormente, usando la declaración UPDATE selecciono el iban, cambio su valor en el registro del usuario con id: CcU-2938 y compruebo que el cambio se ha efectuado.

```
56 -- Cambio el iban del usuario:
57 • UPDATE credit_card
58 SET iban = 'R323456312213576817699999'
59 WHERE id = 'CcU-2938';
60
61 • SELECT*
62 FROM credit_card
63 WHERE id = 'CcU-2938'; -- compruebo que el cambio se ha efectuado
```



id	iban	pan	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	5424465566813633	3257	984	2022-10-30

credit_card 5

Action Output

Time	Action	Response
35 11:57:30	SELECT* FROM credit_card...	1 row(s) returned

Exercici 3

En la taula "transaction" ingresa un nou usuari amb la següent informació:

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

- Para ingresar un nuevo usuario en la tabla 'transaction' primero necesité crear el id de la compañía en la tabla 'company', así como el id de la tarjeta de crédito en la tabla 'credit_card' porque estas tablas están relacionadas a través de las PK de 'company' y 'credit_card' que son FK en 'transaction'.

```

12 • SELECT *
13 FROM transactions.company
14 WHERE id = 'b-9999';

```

id	company_name	phone	email	country	website
b-9999					

company 2

Action Output

Time	Action	Response
60 13:34:46	SELECT * FROM t...	1 row(s) returned

```

14 • SELECT *
15 FROM credit_card
16 WHERE id = 'CcU-9999';

```

id	iban	pan	pin	cvv	expiring_date
CcU-9999					

credit_card 4

Action Output

Time	Action	Response
395 18:27:59	SELECT * FROM credit_card...	1 row(s) returned

- Para añadir el nuevo usuario como no tengo datos para la columna 'timestamp' nombro las columnas a las que afectará nuestra fila de datos y añado los valores correspondientes de las columnas en orden para evitar errores:

```

85 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
86 VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, 111.11, 0);
87
88 -- Compruebo que se haya insertado bien nuestra fila en la tabla transaction:
89 • SELECT *
90 FROM transaction
91 WHERE 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		111.11	0

transaction 26

Action Output

Time	Action	Response
397 18:33:45	SELECT * FROM transac...	1 row(s) returned

Exercici 4

Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_*card. Recordats mostrar el canvi realitzat.

```

89 • ALTER TABLE credit_card
90 DROP COLUMN pan; -- Elimino la columna 'pan' de 'credit_card'
91
92 • SELECT *
93 FROM credit_card; -- Compruebo que la he eliminado correctamente

```

id	iban	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	3257	984	2022-10-30
CcU-2945	DO26854763748537475216568689	9080	887	2023-08-24
CcU-2952	BG45IVQL52710525608255	4598	438	2021-06-29
CcU-2959	CH7242477244335841535	3583	667	2023-02-24
CcU-2966	BG72LKQ15627626377363	4900	130	2024-10-29
CcU-2973	PT87806228135092429456346	6790	887	2025-01-30
CcU-2980	DE39241861883066277136	5075	596	2023-07-24
CcU-2987	GE89681434837748781813	2298	797	2023-10-31
CcU-2994	BH62714428368066765294	7545	595	2022-02-28

credit_card 13

Action Output

Time	Action	Response
63 13:40:13	SELECT * FROM...	155 row(s) returned

Nivell 2

Exercici 1

Elimina de la taula transaction el registre amb ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de dades.

```
100 • DELETE FROM transaction
101 WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
102
100% 1:103
Action Output
Time Action Response
65 13:51:30 DELETE FROM transaction WHERE id = '02C6201E-D90A-1... 1 row(s) affected
```

- Comprobamos que el registro se haya eliminado correctamente:

```
103 • SELECT *
104 FROM transaction
105 WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
106
100% 34:98
Result Grid
id credit_card_id company_id user_id lat longitude timestamp amount declined
transaction 15
Action Output
Time Action Response
66 13:53:20 SELECT * FROM transaction WHERE id = '02C6201E-D90A-1... 0 row(s) returned
```

- Como no nos aparece ningún registro con ese ID en la tabla 'transaction' suponemos que la eliminación ha sido correcta.

Exercici 2

La secció de màrqueting desitja tenir accés a informació específica per a realitzar anàlisi i estratègies efectives. S'ha sol·licitat crear una vista que proporcioni detalls clau sobre les companyies i les seves transaccions. Serà necessària que creïs una vista anomenada VistaMarketing que contingui la següent informació: Nom de la companyia. Telèfon de contacte. País de residència. Mitjana de compra realitzat per cada companyia. Presenta la vista creada, ordenant les dades de major a menor mitjana de compra.

```
117 • CREATE VIEW VistaMarketing AS
118 SELECT c.company_name AS nombre, ANY_VALUE(c.phone) AS telefono,
119 ANY_VALUE(c.country) AS pais, ROUND(AVG(t.amount), 2) AS media_de_compra
120 FROM company AS c
121 JOIN transaction AS t ON c.id = t.company_id
122 WHERE t.declined = 0 -- filtro para descartar las ventas declinadas
123 GROUP BY 1; -- Creo la vista
124
125 • SELECT *
126 FROM VistaMarketing
127 ORDER BY media_de_compra DESC; -- Visualizo la vista ordenada de mayor a menor Media de compra
128
100% 76:141
Result Grid
nombre telefono pais media_de_compra
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
VistaMarketing 66
Action Output
Time Action Response
27... 13:46:13 SELECT * FROM VistaMarketing ORDER BY media_de_compra DE... 101 row(s) returned
```

- Agrupo por el nombre de la compañía y utilizo la función ANY_VALUE(), que me da cualquier valor (si hubiera más de un valor posible), para el teléfono y el país, ya que si no me da error al estar agrupando solo por uno de los campos del SELECT. (Esto lo hago después de comprobar que el teléfono y el país es siempre el mismo para cada compañía en cada uno de los registros).
- Utilizo el filtro WHERE para descartar las compras que hayan sido declinadas ya que estas incrementarían la media de compra falsamente (porque son compras no realizadas).

Exercici 3

Filtra la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"

```

132 SELECT *
133 FROM VistaMarketing
134 WHERE Pais = 'Germany';

```

nombre	telefono	pais	media_de_compra
Nunc Intertium Incorporated	05 18 15 48 13	Germany	242.95
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.57
Rutrum Non Inc.	02 66 31 61 09	Germany	266.90
Ac Industries	09 34 65 40 80	Germany	396.15
Conwallis In Incorporated	06 66 57 29 50	Germany	60.99
Auctor Mauris Corp.	05 60 87 14 41	Germany	308.99
Augue Foundation	06 88 43 15 63	Germany	15.05
Aliquam PC	01 45 73 52 16	Germany	280.34

VistaMarketing 63

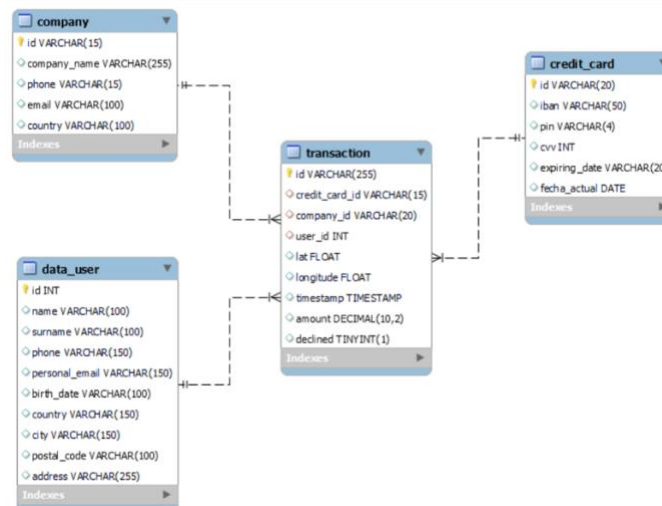
Action Output

Time	Action	Response
2421 12:26:30	SELECT * FROM VistaMarketing WHERE Pais = 'Germany'	8 row(s) returned

Nivell 3

Exercici 1

La setmana vinent tindràs una nova reunió amb els gerents de màrqueting. Un company del teu equip va realitzar modificacions en la base de dades, però no recorda com les va realitzar. Et demana que l'ajudis a deixar els comandos executats per a obtenir el següent diagrama:



Recordatori: En aquesta activitat, és necessari que descriguis el "pas a pas" de les tasques realitzades. És important realitzar descripcions senzilles, simples i fàcils de comprendre. Per a realitzar aquesta activitat hauràs de treballar amb els arxius denominats "estructura_dades_user" i "dades_introduir_user".

- Lo primero es cambiar los tipos de los campos de la tabla 'credit_card' y añadirle una nueva columna llamada fecha_actual que sea de tipo DATE:

```

143 • ALTER TABLE transactions.credit_card
144 CHANGE COLUMN iban
145 iban VARCHAR(50);
146
147 • ALTER TABLE transactions.credit_card
148 CHANGE COLUMN pin
149 pin VARCHAR(4);
150
151 • ALTER TABLE transactions.credit_card
152 CHANGE COLUMN cvv
153 cvv INT;
154
155 • ALTER TABLE transactions.credit_card
156 CHANGE COLUMN expiring_date
157 expiring_date VARCHAR(20);
158
159 -- También he de añadir una nueva columna llamada 'fecha_actual' que sea de tipo DATE
160 • ALTER TABLE transactions.credit_card
161 ADD fecha_actual DATE;
162
100% 1:142

```

Time	Action	Response
24...	ALTER TABLE transactions.credit_card ADD fecha_actual DATE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

- Y eliminar la columna 'website' de la tabla 'company':

```

169
170 • ALTER TABLE company
171 DROP COLUMN website;
172
173
100% 53:167

```

Time	Action	Response
737 11:46:58	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

- Ahora hemos de crear la tabla 'user', renombrarla a 'data_user' y cambiar el nombre de la columna 'email' por 'personal_email':

```

169 • CREATE INDEX idx_user_id ON transaction(user_id); -- repetir en credit_card
170
171 • CREATE TABLE IF NOT EXISTS user (
172     id INT PRIMARY KEY,
173     name VARCHAR(100),
174     surname VARCHAR(100),
175     phone VARCHAR(150),
176     email VARCHAR(150),
177     birth_date VARCHAR(100),
178     country VARCHAR(150),
179     city VARCHAR(150),
180     postal_code VARCHAR(100),
181     address VARCHAR(255),
182     FOREIGN KEY(id) REFERENCES transaction(user_id)
183 );
184
185 -- Renombrar la tabla a data_user:
186 • RENAME TABLE user
187 TO data_user;
188
189 -- Y cambiar el nombre de la columna 'email' de la tabla 'data_user' por 'personal_email'
190 • ALTER TABLE data_user
191 CHANGE COLUMN email
192 personal_email VARCHAR(150);
193
100% 30:178

```

Time	Action	Response
2712 13:06:09	ALTER TABLE data_user CHANGE COLUMN email personal_email...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

- y la visualizamos:

196 • SELECT *

197 FROM data_user;

100% 18/194

	id	name	surname	phone	personal_email	birth_date	country	city	postal_code	address
> 1	Zeus	Gamble		1-282-581-0551	interdum.enim@protonmail.edu	Ncv 17, 1985	United States	Lowell	73544	348-7819 Sagitts St.
2	Garrett	Mcconnell		(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Des Moines	59464	903 Sit Ave
3	Claran	Harrison		(522) 598-1365	interdum.leugiat@aol.org	Apr 29, 1998	United States	Columbus	56518	736-2063 Tellus St.
4	Howard	Stafford		1-411-740-3269	omara.egestas@icloud.edu	Feb 18, 1989	United States	Kailua	77417	Ap #545-2244 Erat. Rd.
5	Hayla	Pierce		1-554-941-2077	et.malesuada.lames@hotmail.org	Sep 26, 1998	United States	Sandy	31564	341-2821 Ultrices Av.

data_user 64

Action Output

Time	Action	Response
2713 13:10:46	SELECT * FROM data_user	275 row(s) returned

- Como al intentar hacer el diagrama la relación entre las tablas 'data_user' y 'transaction' no es correcta: introducimos el usuario que existe en 'transaction' pero no en 'data user', borramos la relación existente (incorrecta) y volvemos a establecer la relación:

```

196 • INSERT INTO data_user (id)
197 VALUES ('9999');
198
199 -- Borro la FK existente:
200 • ALTER TABLE data_user
201 DROP FOREIGN KEY data_user_ibfk_1;
202
203 -- y vuelvo a establecer la relación:
204 • ALTER TABLE transaction
205 ADD FOREIGN KEY (user_id) REFERENCES data_user(id);

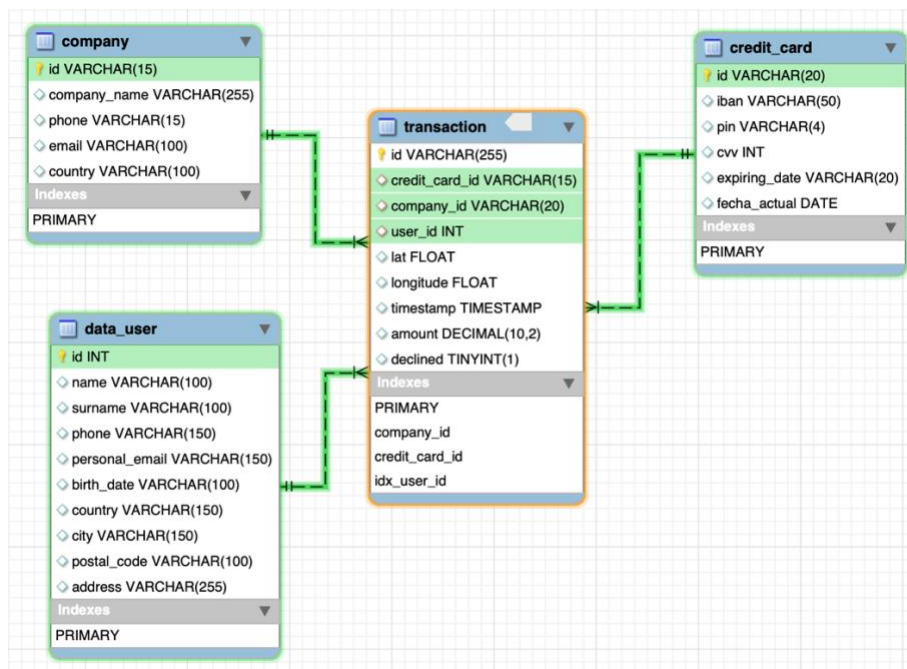
```

100% 33/194

Action Output

Time	Action	Response
725 10:31:52	ALTER TABLE transaction ADD FOREIGN KEY (user_id) REFERENCES data_user(id)	587 row(s) affected Records

- Obtenemos el siguiente diagrama:



Que es igual al que teníamos que crear y en el que podemos observar la relación entre las cuatro tablas de la base de datos 'transactions', así como sus PK y FK.

Exercici 2

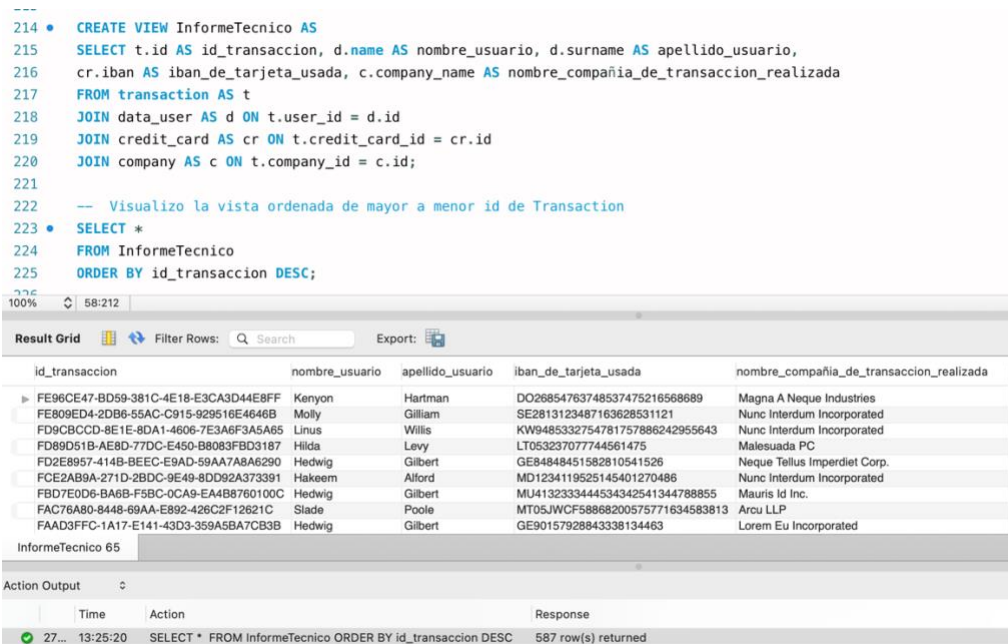
L'empresa també et sol·licita crear una vista anomenada "InformeTecnico" que contingui la següent informació:

- ID de la transacció
- Nom de l'usuari/ària
- Cognom de l'usuari/ària
- IBAN de la targeta de crèdit usada.
- Nom de la companyia de la transacció realitzada.

Assegura't d'incloure informació rellevant de totes dues taules i utilitza àlies per a canviar de nom columnes segons sigui necessari.

Mostra els resultats de la vista, ordena els resultats de manera descendent en funció de la variable ID de transaction.

```
214 • CREATE VIEW InformeTecnico AS
215 SELECT t.id AS id_transaccion, d.name AS nombre_usuario, d.surname AS apellido_usuario,
216 cr.iban AS iban_de_tarjeta_usada, c.company_name AS nombre_compañia_de_transaccion_realizada
217 FROM transaction AS t
218 JOIN data_user AS d ON t.user_id = d.id
219 JOIN credit_card AS cr ON t.credit_card_id = cr.id
220 JOIN company AS c ON t.company_id = c.id;
221
222 -- Visualizo la vista ordenada de mayor a menor id de Transaction
223 • SELECT *
224 FROM InformeTecnico
225 ORDER BY id_transaccion DESC;
```



id_transaccion	nombre_usuario	apellido_usuario	iban_de_tarjeta_usada	nombre_compañia_de_transaccion_realizada
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9CBCC0-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCE2AB9A-271D-2BDC-9E49-8DD92A373391	Hakeem	Allford	MD1234119525145401270486	Nunc Interdum Incorporated
FBD7E0D6-BA6B-F5BC-0CA9-EA4B8760100C	Hedwig	Gilbert	MU4132333444534342541344788855	Mauris Id Inc.
FAC76A80-84A8-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF5886820057771634583813	Arcu LLP
FAAD3FFC-1A17-E141-43D3-359A5BA7CB3B	Hedwig	Gilbert	GE90157928843338134463	Lorem Eu Incorporated

InformeTecnico 65

Time	Action	Response
27...	SELECT * FROM InformeTecnico ORDER BY id_transaccion DESC	587 row(s) returned

*He intentado tener en cuenta la recomendación de utilizar alias para hacer que el código sea más legible y fácil de mantener (usando por ejemplo c y t en lugar de escribir company y transaction repetidamente).