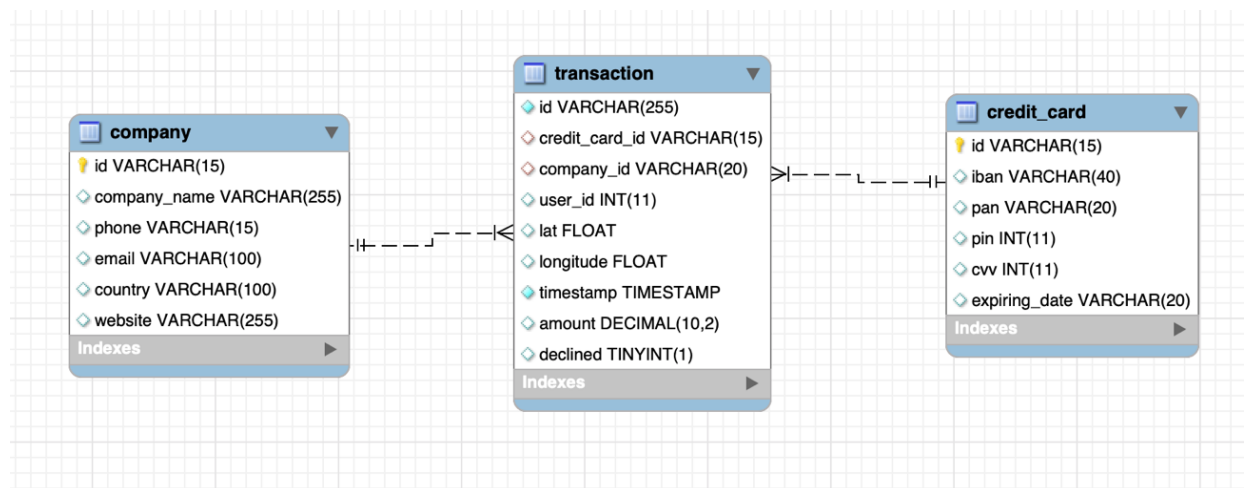


**Nivell 1****Exercici 1**

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.

```
8
9 CREATE TABLE credit_card(
10     id VARCHAR(15) PRIMARY KEY,
11     iban VARCHAR(40),
12     pan VARCHAR(20),
13     pin VARCHAR(4),
14     cvv INT,
15     expiring_date VARCHAR(20)
16 );
17
18 ALTER TABLE transaction ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
19
20 # Exercici 2: El departament de Recursos Humans ha identificat un error en el número de compte de l'usuari amb ID CcU-2938.
21 # La informació que ha de mostrar-se per a aquest registre és: R323456312213576817699999.
22 # Recorda mostrar que el canvi es va realitzar.
23
```

Time	Action	Response	Duration / Fetch Time
14:51:41	ALTER TABLE transaction ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id)	587 row(s) affected Records: 587 Duplicates: 0 War... 0.055 sec	



Al añadir la tabla "credit\_card" al diagrama, este se convierte en un diagrama de estrella con "transaction" como su tabla de hechos. "company" y "credit\_card" son las tablas de dimensiones.

## Anabel Martínez Ramírez

La Primary Key de “transaction” es “id” y está conectada con “company” mediante la Foreign Key “company\_id” y con “credit\_card” mediante la Foreign Key “credit\_card\_id”.

La Primary Key de “company” es “id”.

La Primary Key de “credit\_card” es “id”.

La relación entre “company” y “transaction” es de uno a muchos, ya que una empresa puede realizar muchas transacciones.

La relación entre “credit\_card” y “transaction” también es uno a muchos, ya que con una tarjeta de crédito se pueden realizar muchas transacciones.

### 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.*

The screenshot shows a database management interface with the following components:

- SQL Editor:** Contains two SQL statements:

```
23  
24 UPDATE transactions.credit_card  
25 SET iban = 'R323456312213576817699999'  
26 WHERE id = 'CcU-2938';  
27  
28 SELECT *  
29 FROM credit_card  
30 WHERE id = 'CcU-2938';  
31
```
- Result Grid:** Displays the results of the SQL execution. The first row shows the updated record for 'CcU-2938' with the new IBAN. The second row shows the result of the SELECT query, which is a single row with all columns set to NULL.

id	iban	pan	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	5424465566813633	3257	984	10/30/22
NULL	NULL	NULL	NULL	NULL	NULL
- Action Output:** Shows the execution details of the SQL statements.

	Time	Action	Response	Duration / Fetch Time
1	14:52:25	UPDATE transactions.credit_card SET iban = 'R323456312213576817699999' WHERE id = 'CcU-2938';	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.00053 sec
2	14:52:28	SELECT * FROM credit_card WHERE id = 'CcU-2938' LIMIT 0, 1000	1 row(s) returned	0.00037 sec / 0.0000...

### Exercici 3:

En la taula "transaction" ingressa 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

```

33
34 INSERT INTO transactions.credit_card (id)
35 VALUES ('CcU-9999');
36
37 INSERT INTO transactions.company (id)
38 VALUES ('b-9999');
39
40 INSERT INTO transactions.transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
41 VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '117.999', '111.11', '0');
42
43 SELECT *
44 FROM transaction
45 WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD';
46
47 # Exercici 4
    
```

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	2025-05-07 14:53:14	111.11	0

Time	Action	Response	Duration / Fetch Time
14:53:07	INSERT INTO transactions.credit_card (id) VALUES ('CcU-9999')	1 row(s) affected	0.00047 sec
14:53:10	INSERT INTO transactions.company (id) VALUES ('b-9999')	1 row(s) affected	0.0011 sec
14:53:14	INSERT INTO transactions.transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)	1 row(s) affected	0.0021 sec
14:53:17	SELECT * FROM transaction WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD' LI...	1 row(s) returned	0.0022 sec / 0.00001...

### Exercici 4:

Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit\_card. Recordra mostrar el canvi realitzat.

```

51 ALTER TABLE credit_card DROP COLUMN pan;
52
53 SELECT *
54 FROM credit_card;
55
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	BGA5IVQL52710525608255	4598	438	08/29/21
CcU-2959	CR7242477244335841535	3583	667	02/24/23
CcU-2966	BG72LKTQ15827628377363	4900	130	10/29/24
CcU-2973	PT87806228135092429456346	8760	887	01/30/25
CcU-2980	DE39241881883086277136	5075	696	07/24/22
CcU-2987	GE89681434837748781813	2298	797	10/31/23
CcU-2994	BH62714428368066765294	7545	695	02/28/22
CcU-3001	CY49087426654774581266832110	9562	867	09/16/22
CcU-3008	LU507216693616119230	1856	740	04/05/25
CcU-3015	PS119398216295715968342456821	3246	822	01/31/22
CcU-3022	GT91695162850556977423121857	5610	342	04/25/25
CcU-3029	AZ62317413982441418123739746	9708	505	09/02/23
CcU-3036	AZ39336002925842865843941994	2232	565	10/27/25
CcU-3043	TN6488143310514852179535	5969	196	06/07/25
CcU-3050	FR5167744368175836831854477	4834	126	10/09/23
CcU-3057	IR3318225748947545215	1805	848	09/14/25

Time	Action	Response	Duration / Fetch Time
14:54:31	ALTER TABLE credit_card DROP COLUMN pan	276 row(s) affected Records: 276 Duplicates: 0 War...	0.047 sec
14:54:35	SELECT * FROM credit_card LIMIT 0, 1000	276 row(s) returned	0.0015 sec / 0.00009...

## Nivell 2

### Exercici 1:

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

The screenshot shows a database management interface with a SQL editor and a results pane. The SQL editor contains the following code:

```
60
61 DELETE FROM transaction
62 WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
63
64 SELECT *
65 FROM transaction
66 WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
67
```

The results pane displays a table with the following columns: id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, amount, and declined. The table is currently empty, showing NULL values for all columns.

Below the table, the 'Action Output' section shows the results of the SQL execution:

	Time	Action	Response	Duration / Fetch Time
✓ 1	14:55:49	DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'	1 row(s) affected	0.00078 sec
✓ 2	14:55:54	SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02' LI...	0 row(s) returned	0.0011 sec / 0.00001...

## Exercici 2:

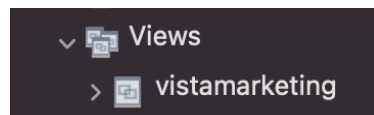
La secció de màrqueting desitja tenir accés a informació específica per a realitzar anàlisi i estratègies efectives. S'ha sol·licitat crear una vista que proporcioni detalls clau sobre les companyies i les seves transaccions. Serà necessària que creïs una vista anomenada VistaMarketing que contingui 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.

```

73
74 • CREATE VIEW VistaMarketing AS
75   SELECT c.company_name AS Empresa, c.phone AS Telefono, c.country as Pais, ROUND(AVG(amount), 2) AS Media_Compras
76   FROM company c
77   JOIN transaction t
78   ON c.id = t.company_id
79   WHERE declined = 0
80   GROUP BY c.id
81   ORDER BY AVG(amount) DESC;
82
100% 54:71

```

	Time	Action	Response	Duration / Fetch Time
3	14:57:04	CREATE VIEW VistaMarketing AS SELECT c.company_name AS Empresa, c.phone AS Telefon...	0 row(s) affected	0.023 sec



```

1 • SELECT * FROM transactions.vistamarketing;

```

Empresa	Telefono	Pais	Media_Compras
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
Justo Eu Arcu Ltd	08 42 56 71 52	Italy	444.16
Pede Cum Ltd	07 62 26 48 38	Norway	442.32
Vestibulum Lorem PC	02 02 87 33 40	Belgium	428.40
Mauris Institute	05 29 60 36 87	Sweden	427.71
Aliquet Diam Limited	02 76 61 47 46	United States	425.64
Mus Aenean Eget Foundation	06 25 15 52 43	Sweden	419.97

	Time	Action	Response	Duration / Fetch Time
4	14:57:52	SELECT * FROM transactions.vistamarketing LIMIT 0, 1000	101 row(s) returned	0.0022 sec / 0.00005...

\* **COMENTARIO:** En esta query, he elegido incluir el “WHERE declined = 0” porque la pregunta especifica que se debe realizar una media de las compras realizadas por cada empresa, y yo con eso entiendo que, para ser una compra, la transacción no debe estar rechazada.

### Exercici 3:

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

The screenshot shows a database management tool interface. At the top, a SQL query is entered in a text area:

```
85
86 SELECT *
87 FROM transactions.VistaMarketing
88 WHERE Pais = 'Germany';
```

Below the query, the 'Result Grid' displays the results of the query. The grid has four columns: 'Empresa', 'Telefono', 'Pais', and 'Media\_Compras'. There are 8 rows of data, all with 'Germany' as the country.

Empresa	Telefono	Pais	Media_Compras
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

Below the result grid, the 'Action Output' section shows the execution details of the query:

	Time	Action	Response	Duration / Fetch Time
5	14:58:46	SELECT * FROM transactions.VistaMarketing WHERE Pais = 'Germany' LIMIT 0, 1000	8 row(s) returned	0.0033 sec / 0.00001...

## 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".*

Para llegar al resultado obtenido, se deben realizar los siguientes pasos:

- En primer lugar, vamos a abrir el archivo "estructura\_datos\_user.sql" y lo vamos a ejecutar en su totalidad dándole al icono del rayo. Con esto, ya tendremos la estructura de la tabla creada en nuestra base de datos.

```
1 -- Creamos la tabla user
2
3 CREATE INDEX idx_user_id ON transaction(user_id);
4
5 CREATE TABLE IF NOT EXISTS user (
6     id INT PRIMARY KEY,
7     name VARCHAR(100),
8     surname VARCHAR(100),
9     phone VARCHAR(150),
10    email VARCHAR(150),
11    birth_date VARCHAR(100),
12    country VARCHAR(150),
13    city VARCHAR(150),
14    postal_code VARCHAR(100),
15    address VARCHAR(255),
16    FOREIGN KEY(id) REFERENCES transaction(user_id)
17 );
18
19
20
21
```

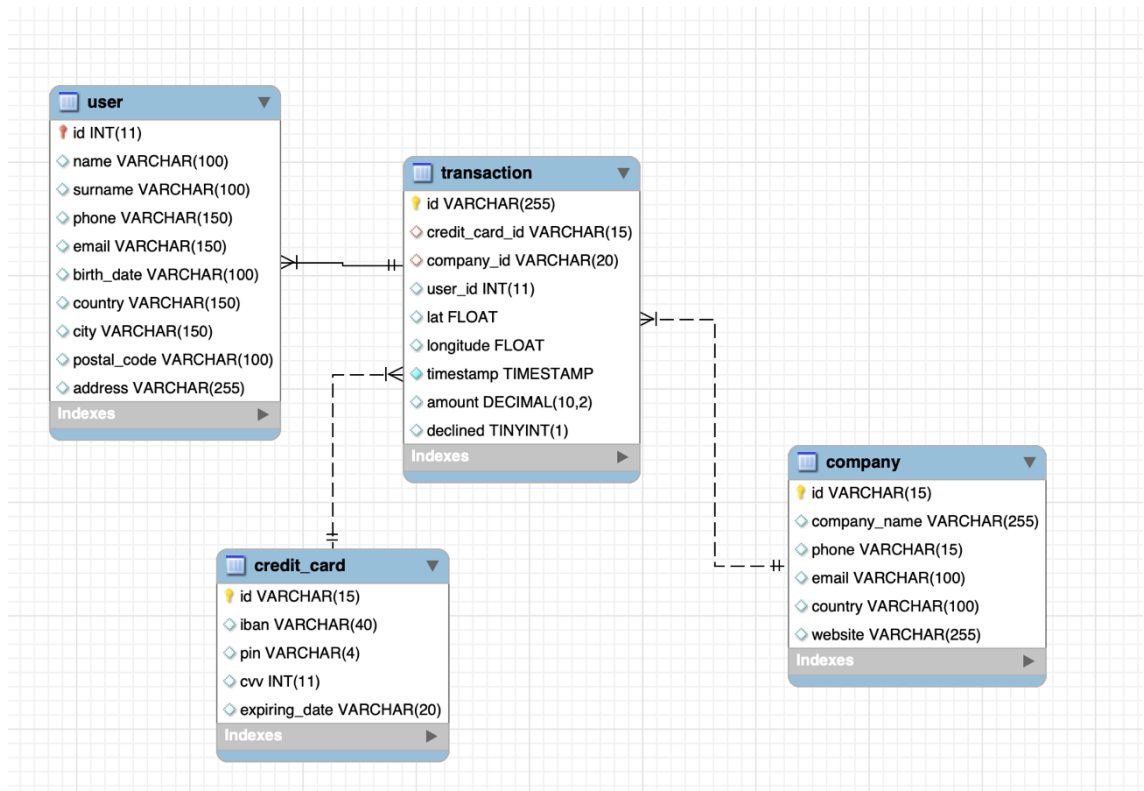
	Time	Action	Response	Duration / Fetch Time
7	15:02:38	CREATE TABLE IF NOT EXISTS user ( id INT PRIMARY KEY, name VARCHAR(100),...	0 row(s) affected	0.0061 sec

A continuación, abrimos el archivo “datos\_introducir\_user(1).sql” y lo vamos a ejecutar en su totalidad dándole al icono del rayo.

```
1 SET foreign_key_checks = 0;
2
3 -- Insertamos datos de user
4 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "1", "Zeus",
5 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "2", "Garrett
6 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "3", "Ciaran
7 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "4", "Howard
8 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "5", "Hayfa",
9 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "6", "Joel",
10 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "7", "Rafael"
11 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "8", "Nissim"
12 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "9", "Mannix"
13 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "10", "Robert
14 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "11", "Joan",
15 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "12", "Benedi
```

	Time	Action	Response	Duration / Fetch Time
270	15:04:05	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code,...	1 row(s) affected	0.00033 sec
271	15:04:05	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code,...	1 row(s) affected	0.00036 sec
272	15:04:05	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code,...	1 row(s) affected	0.00039 sec
273	15:04:05	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code,...	1 row(s) affected	0.00034 sec
274	15:04:05	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code,...	1 row(s) affected	0.00033 sec
275	15:04:05	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code,...	1 row(s) affected	0.00072 sec
276	15:04:05	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code,...	1 row(s) affected	0.00035 sec
277	15:04:05	SET foreign_key_checks = 1	0 row(s) affected	0.00011 sec

Con esto, ya habremos rellenado nuestra base de datos. Observaremos que, al hacerlo, el diagrama obtenido es distinto del que debería ser, es el siguiente:



Por ello, para corregirlo, vamos a seguir los pasos siguientes:

- En primer lugar, vemos que la Foreign Key se ha vinculado erróneamente en el código en esta nueva tabla “user\_id”, vamos a eliminarla mediante el siguiente comando:

```
94
95 ALTER TABLE user
96 DROP FOREIGN KEY user_ibfk_1;
97
100% 30:96
Action Output
Time Action Response Duration / Fetch Time
1 17:02:31 ALTER TABLE user DROP FOREIGN KEY user_ibfk_1 275 row(s) affected Records: 275 Duplicates: 0 War... 0.041 sec
```

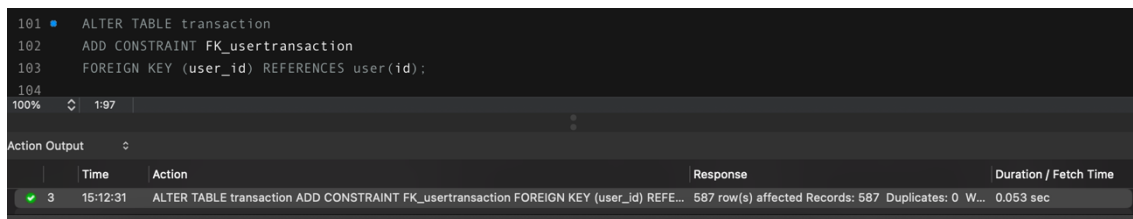
- A continuación, vamos a introducir el usuario “9999” que habíamos creado en un ejercicio anterior en la tabla “user\_id” para que no haya una inconsistencia entre las claves. Esto se hace mediante el comando:

```
98 INSERT INTO transactions.user(id)
99 VALUES ('9999');
100
101 ALTER TABLE transaction
102
100% 1:95
Action Output
Time Action Response Duration / Fetch Time
2 15:10:51 INSERT INTO transactions.user(id) VALUES ('9999') 1 row(s) affected 0.00046 sec
```



Ahora, vamos a crear de nuevo la Foreign Key con el código:

```
101 ALTER TABLE transaction
102 ADD CONSTRAINT FK_usertransaction
103 FOREIGN KEY (user_id) REFERENCES user(id);
104
```



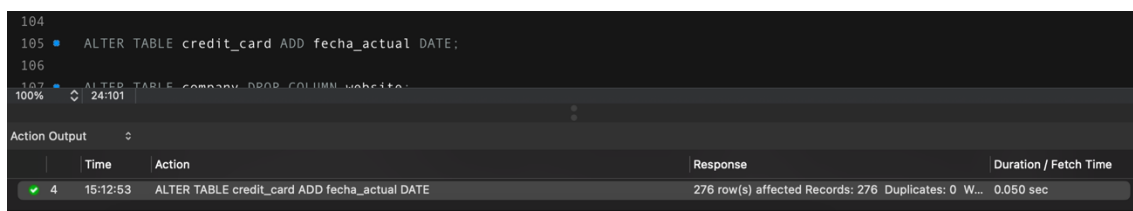
	Time	Action	Response	Duration / Fetch Time
3	15:12:31	ALTER TABLE transaction ADD CONSTRAINT FK_usertransaction FOREIGN KEY (user_id) REFERENCE...	587 row(s) affected Records: 587 Duplicates: 0 W...	0.053 sec

Y ya tenemos el diagrama correctamente vinculado con las Foreign Key y las Primary Key correctas.

Ahora nos queda comprobar que el contenido de las tablas sea el mismo, y vemos que no lo es. Para ello, debemos realizar las siguientes correcciones:

- En primer lugar, la tabla “credit\_card” ha sido alterada y se le ha añadido una nueva columna llamada “fecha\_actual”, por lo que vamos a realizar el siguiente comando para añadirle una nueva columna:

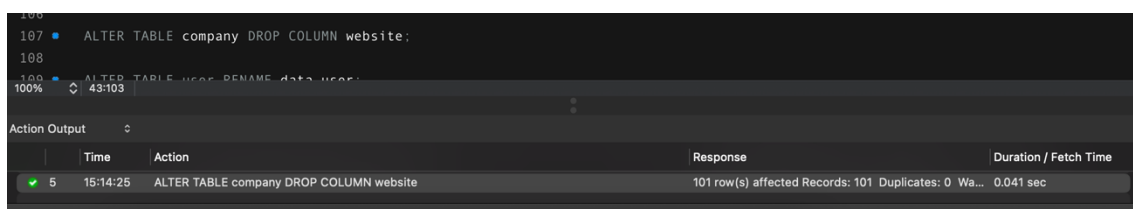
```
104
105 ALTER TABLE credit_card ADD fecha_actual DATE;
106
107 ALTER TABLE company DROP COLUMN website;
108
```



	Time	Action	Response	Duration / Fetch Time
4	15:12:53	ALTER TABLE credit_card ADD fecha_actual DATE	276 row(s) affected Records: 276 Duplicates: 0 Wa...	0.050 sec

- A continuación, vemos que de la tabla “company” se ha eliminado la columna “website”, lo cual hacemos con el código:

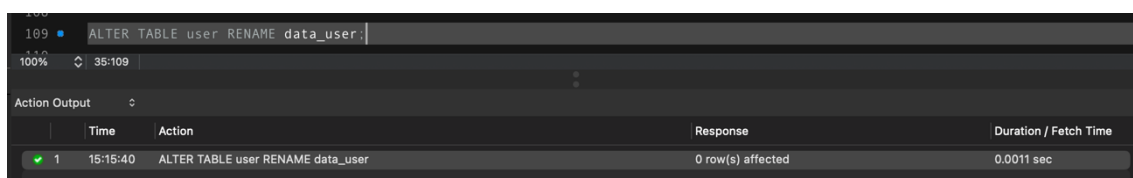
```
109 ALTER TABLE company DROP COLUMN website;
110
111 ALTER TABLE user RENAME data_user;
112
```



	Time	Action	Response	Duration / Fetch Time
5	15:14:25	ALTER TABLE company DROP COLUMN website	101 row(s) affected Records: 101 Duplicates: 0 Wa...	0.041 sec

- También se ha renombrado la tabla “user”, que ahora se debería llamar “data\_user”. Esto lo hacemos de la siguiente forma:

```
109 ALTER TABLE user RENAME data_user;
```



	Time	Action	Response	Duration / Fetch Time
1	15:15:40	ALTER TABLE user RENAME data_user	0 row(s) affected	0.0011 sec

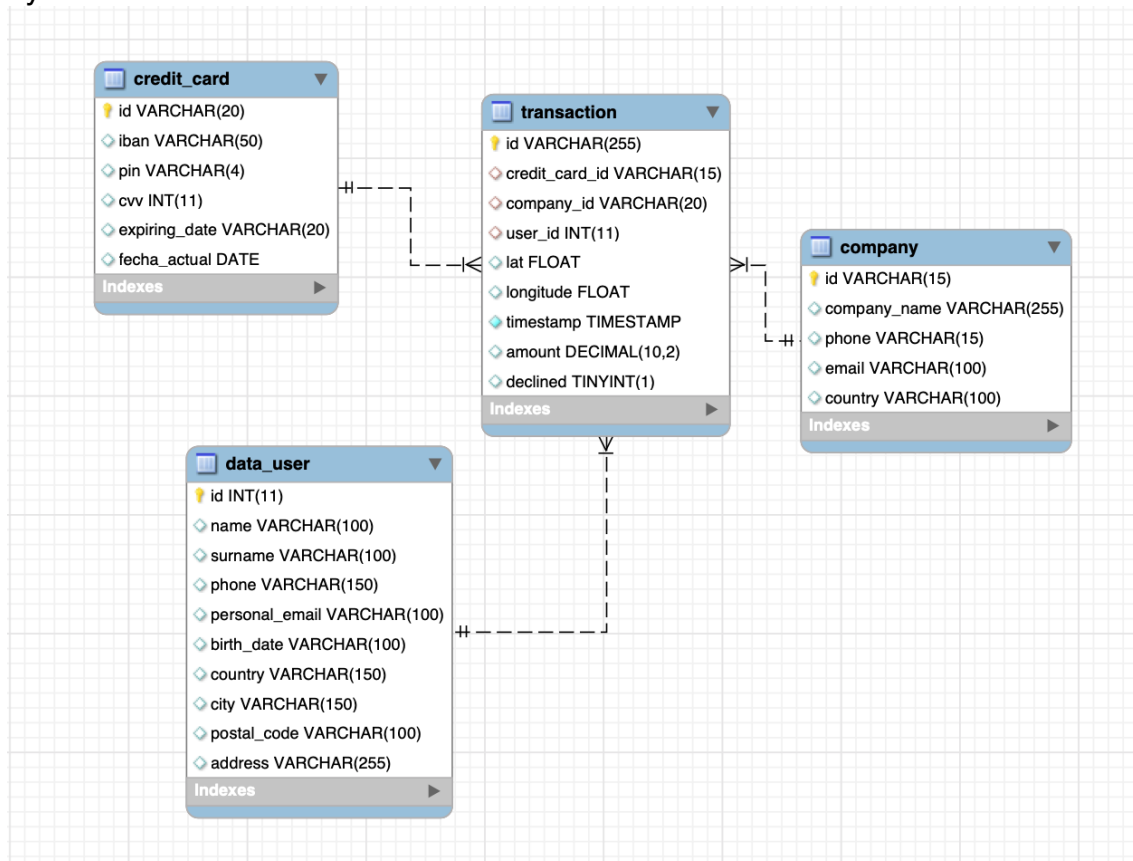
- Dentro de la tabla “data\_user”, se ha cambiado el nombre de la columna “email”, que ahora se llama “personal\_email”. Esto se hace de la siguiente manera:

```
111 ALTER TABLE data_user CHANGE email personal_email VARCHAR(100);
112
113
100% 49:113
Action Output
Time Action Response Duration / Fetch Time
1 15:16:08 ALTER TABLE data_user CHANGE email personal_email VARCHAR(100) 276 row(s) affected Records: 276 Duplicates: 0 W... 0.040 sec
```

- Por ultimo, en la tabla credit\_card, hay dos columnas en las que se ha tenido que modificar la longitud del tipo de variable VARCHAR, en “iban” e “id”. Esto se hace mediante este código:

```
113 ALTER TABLE credit_card MODIFY COLUMN iban VARCHAR(50);
114 ALTER TABLE credit_card MODIFY COLUMN id VARCHAR(20);
100% 54:114
Action Output
Time Action Response Duration / Fetch Time
1 15:16:33 ALTER TABLE credit_card MODIFY COLUMN iban VARCHAR(50) 276 row(s) affected Records: 276 Duplicates: 0 W... 0.038 sec
2 15:16:36 ALTER TABLE credit_card MODIFY COLUMN id VARCHAR(20) 276 row(s) affected Records: 276 Duplicates: 0 W... 0.012 sec
```

Así es como se obtiene un diagrama como el que nuestro compañero realizó ayer:



## 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.

```

121
122 CREATE VIEW InformeTecnico AS
123 SELECT t.id AS ID_transaccion, u.name AS nombre, u.surname AS apellido, cc.iban, c.company_name AS empresa
124 FROM transaction t
125 JOIN company c
126 ON t.company_id = c.id
127 JOIN data_user u
128 ON u.id = t.user_id
129 JOIN credit_card cc
130 ON cc.id = t.credit_card_id;
  
```

Object Info Session

Schema: transactions

Action Output

	Time	Action	Response	Duration / Fetch Time
1	15:18:50	CREATE VIEW InformeTecnico AS SELECT t.id AS ID_transaccion, u.name AS nombre, u.surna...	0 row(s) affected	0.020 sec

```

1 SELECT * FROM transactions.informetecnico;
  
```

Result Grid

ID_transaccion	nombre	apellido	iban	empresa
8BCE9908-6D44-B7DB-D363-D468A2DF3A25	Roth	Cook	GI98IMNL122444875373267	Lorem Eu Incorporated
A2CD7827-4C8D-9E71-8549-26283307BF1	Duncan	Romero	GI98IMNL122444875373267	Lorem Eu Incorporated
44D0BBB1-975E-86AA-75DA-BDAFED65F368	Phyllis	Holt	GI98IMNL122444875373267	Lorem Eu Incorporated
DA7A5D78-448A-678E-4836-465C8B963F4E	Beverly	Burt	GI98IMNL122444875373267	Lorem Eu Incorporated
E3E941D3-0866-2E6F-85D7-A2F21E79A215	Irma	Whitehead	BH86814579285516920703	Lorem Eu Incorporated
18B1AD88-F155-45B7-D2EA-4738407435D6	Yao	Emerson	BH86814579285516920703	Lorem Eu Incorporated
2660D3B8-0B2C-AEBA-A891-95433968CF71	Bert	Juarez	BH86814579285516920703	Lorem Eu Incorporated
B31AC3B8-1529-B0CE-C385-41CDF671921	Kenneth	Morrison	BH86814579285516920703	Lorem Eu Incorporated

informetecnico 1

Action Output

	Time	Action	Response	Duration / Fetch Time
1	15:18:50	CREATE VIEW InformeTecnico AS SELECT t.id AS ID_transaccion, u.name AS nombre, u.surna...	0 row(s) affected	0.020 sec
2	15:19:26	SELECT * FROM transactions.informetecnico LIMIT 0, 1000	587 row(s) returned	0.0063 sec / 0

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

```

134 SELECT *
135 FROM informetecnico
136 ORDER BY ID_transaccion DESC;
  
```

Result Grid

ID_transaccion	nombre	apellido	iban	empresa
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BE0C-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCE2AB9A-271D-2BDC-9E49-8DD92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated
FBD7E0D6-BA6B-F5BC-0CA9-EA4B8760100C	Hedwig	Gilbert	MU4132333444534342541344788855	Mauris Id Inc.
FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF58868200575771634583813	Arcu LLP
FAAD3FFC-1A17-E141-43D3-359A5BA7CB3B	Hedwig	Gilbert	GE9015792884338134463	Lorem Eu Incorporated
FA053936-75D8-85FA-490D-9B624E1B920A	Hedwig	Gilbert	GT02497653655330848247645975	Non Justo Corp.
F85A7D75-2778-9D75-D776-3F41A828DE88	Sarah	Beck	VG1468087984174845729577	Ut Semper Foundation
F843D08-CCB5-2444-1B4E-5966289FBA8B	Jasper	Landry	VG1468087984174845729577	Ut Semper Foundation

informetecnico 1

Action Output

	Time	Action	Response	Duration / Fetch Time
3	15:20:03	SELECT * FROM informetecnico ORDER BY ID_transaccion DESC LIMIT 0, 1000	587 row(s) returned	0.0069 sec / 0.00014...