

SPRINT 3

Nivel 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.

Para este primer ejercicio, lo primero que hacemos es crear la tabla credit card y establecemos que la primaryKey sea el id y también determinamos qué tipo de valor es el correspondiente a cada columna. Para que el id sea PK, tambien debemos asegurarnos que no pueda haber valores nulos en la tabla y lo hacemos mediante el código que mostramos a continuación.

```
18 CREATE TABLE credit_card (  
19     id VARCHAR(20) not null,  
20     iban VARCHAR(50),  
21     pan CHAR(50),  
22     pin CHAR(6),  
23     cvv CHAR(4),  
24     expiring_date VARCHAR(20),  
25     primary key(id));
```

Una vez que hemos creado la tabla, podemos añadir los datos del documento adjunto con los datos de la tabla credit_card

Lo siguiente que debemos hacer es la relación con la tabla transaction y para esto debemos establecer la Foreign Key en la tabla transaction. Pero al momento de intentar crear la FK nos da un error. Así que verificamos cómo fue creada la tabla transaction y nos encontramos que la tabla tiene un CONSTRAINT.

```
32 • show create table transaction;
```

Así que utilizamos el comando ALTER TABLE para eliminar el CONSTRAINT utilizando el comando DROP

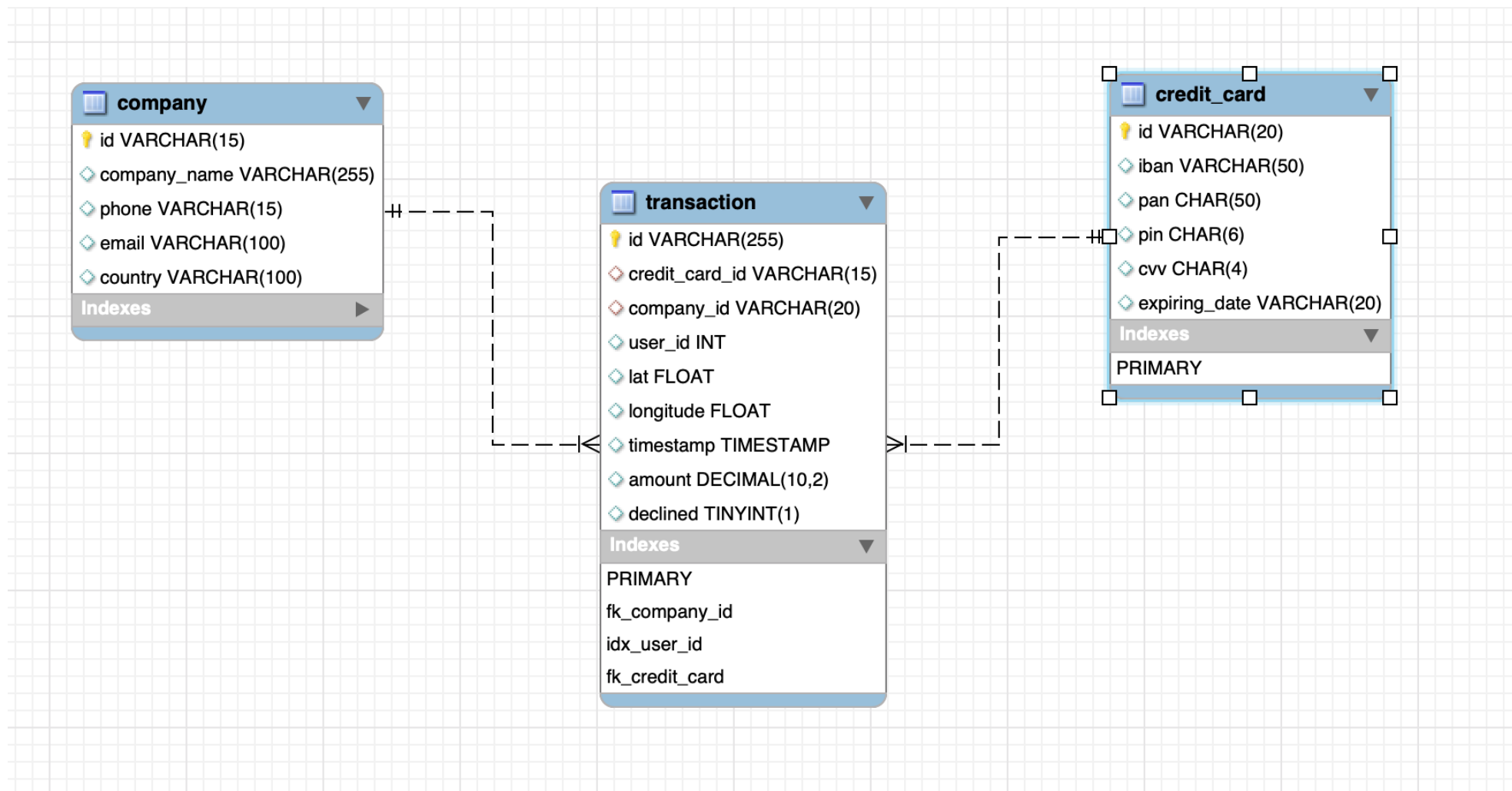
```
30 • ALTER TABLE transaction DROP CONSTRAINT transaction_ibfk_1;
```

Una vez, eliminado el CONSTRAINT, generamos de nuevo las FK de la tabla Transaction mediante el código ALTER TABLE Y ADD CONSTRAINT.

```
34 • ALTER TABLE transaction
35   ADD CONSTRAINT fk_credit_card
36   FOREIGN KEY (credit_card_id)
37   REFERENCES credit_card(id);
38
```

```
39 • ALTER TABLE transaction
40   ADD CONSTRAINT fk_company_id
41   FOREIGN KEY (company_id)
42   REFERENCES company(id);
```

Así que una vez generadas las FK, sabemos que las tablas están relacionadas y podemos generar el modelo utilizando el Reverse Engineer.



En el modelo podemos observar las relaciones entre las tablas y los tipos de datos que contienen,

Exercici 2

El departament de Recursos Humans ha identificat un error en el número de compte associat a la targeta de crèdit amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és: TR323456312213576817699999. Recorda mostrar que el canvi es va realitzar

Para este ejercicio, primero nos aseguramos que el campo exista haciendo una selección sencilla.

```
67 SELECT *
68 FROM credit_card
69 WHERE id = "CcU-2938";
```

✓ 20343 13:33:13 SELECT * FROM credit_card WHERE id = "CcU-2938" 1 row(s) returned

Una vez que confirmamos el dato, utilizamos en comando UPDATE para modificar el campo de la tabla deseada y con SET establecemos el nuevo valor. Y filtramos con un WHERE dónde indicamos a qué valor se aplicará el cambio.

```
71 UPDATE credit_card
72 SET iban = "TR323456312213576817699999"
73 WHERE id = "CcU-2938";
```

	id	iban	pan	pin	cvv	expiring_date	
▶	CcU-2938	TR323456312213576817699999	5424465566813633	3257	984	10/30/22	

✓ 20344 13:34:00 UPDATE credit_card SET iban = "TR323456312213576817699999"... 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

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

Para este ejercicio, primero tenemos que agregar los campos "credit_card_id" y "company_id" en las tablas "credit_card" y "company" respectivamente, ya que ambos campos son Foreign Keys de la tabla "transaction" de lo contrario no podremos añadir la información que nos piden. Para hacer ésto utilizamos la función INSERT INTO y seleccionamos la tabla y el campo dónde queremos introducir la información y con VALUES establecemos el valor del campo.

```
75
76 • INSERT INTO credit_card (id)
77   VALUES ("CcU-9999");
78
79 • INSERT INTO company (id)
80   VALUES ("b-9999");
```

✓ 20346 10:22:31 INSERT INTO credit_card (id) VALUES ("CcU-9999") 1 row(s) affected

Una vez que hemos creado los campos de las Foreign Keys, ahora si que podemos introducir el resto de la información a la tabla transaction y utilizaremos de nuevo la función INSERT INTO seleccionando la tabla y los valores de la fila en las columnas correspondientes.

```
82 • INSERT INTO transaction (id,credit_card_id,company_id,user_id,lat,longitude,amount,declined)
83   VALUES ("108B1D1D-5B23-A76C-55EF-C568E49A99DD","CcU-9999","b-9999",9999,829.999,-117.999,111.11,0);
84
```

✓ 20348 10:33:35 INSERT INTO transaction (id,credit_card_id,company_id,user_id,lat,lo... 1 row(s) affected

Exercici 4

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

Para este ejercicio, primero haremos la selección de todos los campos de la tabla para hacer la comprobación de la tabla y sus columnas.

```
85 SELECT *
86 FROM credit_card;
87
88
```

	id	iban	pan	pin	cvv	expiring_date	
▶	CcS-4857	NULL	NULL	NULL	NULL	NULL	
	CcS-4858	NULL	NULL	NULL	NULL	NULL	
	CcS-4859	NULL	NULL	NULL	NULL	NULL	
	CcS-4860	NULL	NULL	NULL	NULL	NULL	
	CcS-4861	NULL	NULL	NULL	NULL	NULL	
	CcS-4862	NULL	NULL	NULL	NULL	NULL	
	CcS-4863	NULL	NULL	NULL	NULL	NULL	
	CcS-4864	NULL	NULL	NULL	NULL	NULL	

✓ 20350 10:43:40 SELECT * FROM credit_card 5001 row(s) returned

Y luego utilizamos la función ALTER TABLE dónde seleccionamos la tabla que queremos modificar y con la función DROP COLUMN seleccionamos la columna que queremos eliminar.

```
89 ALTER TABLE credit_card
90 DROP COLUMN pan;
91
```

	id	iban	pin	cvv	expiring_date	
	CcS-4859	NULL	NULL	NULL	NULL	
	CcS-4860	NULL	NULL	NULL	NULL	
	CcS-4861	NULL	NULL	NULL	NULL	
	CcS-4862	NULL	NULL	NULL	NULL	
	CcS-4863	NULL	NULL	NULL	NULL	
	CcS-4864	NULL	NULL	NULL	NULL	
	CcS-4865	NULL	NULL	NULL	NULL	
	CcS-4866	NULL	NULL	NULL	NULL	

✓ 13 18:52:04 ALTER TABLE credit_card DROP COLUMN pan 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Nivell 2

Exercici 1

Elimina de la taula transaction el registre amb ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de dades.

Primero comprobamos que la transacción existe mediante una selección simple filtrando mediante WHERE y con el id de la transacción.

```
92 • SELECT *
93 FROM transaction
94 WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
```

Y luego utilizamos de nuevo la función DELETE, especificamos la tabla de la que queremos borrar el campo y volvemos a filtrar utilizando el WHERE junto con el id del registro que queremos eliminar.

```
96 • DELETE FROM transaction
97 WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
```

```
✓ 15 18:56:22 DELETE FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE... 1 row(s) affected
```

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.

Para éste ejercicio, utilizamos el comando CREATE VIEW seguido del nombre que queremos utilizar y utilizamos AS para crear la vista solicitada. Luego realizamos la selección de los campos requeridos (nombre, teléfono, país y media de ventas) y ordenamos de la forma requerida. Para crear la vista es necesario hacer la agrupación de todos los campos que no sean una función de agregación para poder luego hacer consultas en la vista de lo contrario, si nos permite realizar la vista, pero no hacer consultas.

```
108 CREATE VIEW VistaMarketing AS
109 SELECT company.company_name,company.phone,company.country,AVG(transaction.amount) as media_compras
110 FROM company
111 JOIN transaction
112 ON company.id=transaction.company_id
113 GROUP BY company.company_name,company.phone,company.country
114 ORDER BY AVG(transaction.amount) DESC;
```

✓ 25398 10:46:50 CREATE VIEW VistaMarketing AS SELECT company.company_name,c... 0 row(s) affected

Aquí efectuamos la selección de todos los campos de la vista para ver cómo queda.

```
114 SELECT * FROM VistaMarketing;
```

	company_name	phone	country	media_compras	
▶	Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.801906	
■	Pretium Neque Corp.	07 77 48 55 28	Australia	275.321348	
■	Urna Convallis Associates	06 01 24 77 04	United States	274.090893	
■	At Associates	09 56 61 10 65	New Zealand	271.836379	
■	Aliquet Diam Limited	02 76 61 47 46	United States	269.934753	
■	Metus Vitae Associates	08 25 44 40 66	Australia	269.730585	
■	Neque Tellus Incorporated	04 43 18 34 19	Ireland	268.295532	
■	Eget Ipsum Ltd	03 67 44 56 72	United States	267.984372	

✓ 20360 11:34:36 SELECT * FROM VistaMarketing 101 row(s) returned

Exercici 3

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

Para éste ejercicio, hacemos la selección de todos los campos de la VISTA que hemos creado anteriormente y filtramos los resultados utilizando WHERE delimitando el país a "Germany"

```
118 • SELECT *
119     FROM VistaMarketing
120     WHERE country = "Germany";
121
```

	company_name	phone	country	media_compras	
▶	Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.801906	
	Nunc Interdum Incorporated	05 18 15 48 13	Germany	258.376749	
	Convallis In Incorporated	06 66 57 29 50	Germany	257.612111	
	Ac Industries	09 34 65 40 60	Germany	255.192324	
	Rutrum Non Inc.	02 66 31 61 09	Germany	255.144344	
	Auctor Mauris Corp.	05 62 87 14 41	Germany	254.675536	
	Augue Foundation	06 88 43 15 63	Germany	253.488327	
	Aliquam PC	01 45 73 52 16	Germany	253.304385	

✓ 20361 12:17:16 SELECT * FROM VistaMarketing WHERE country = "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:

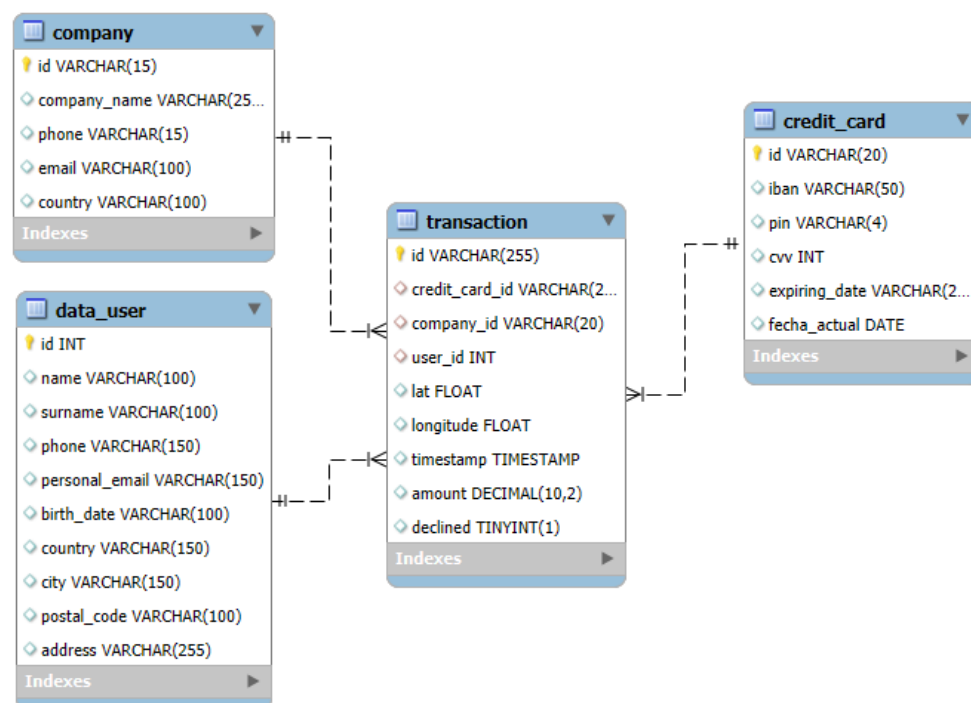


Diagrama modelo

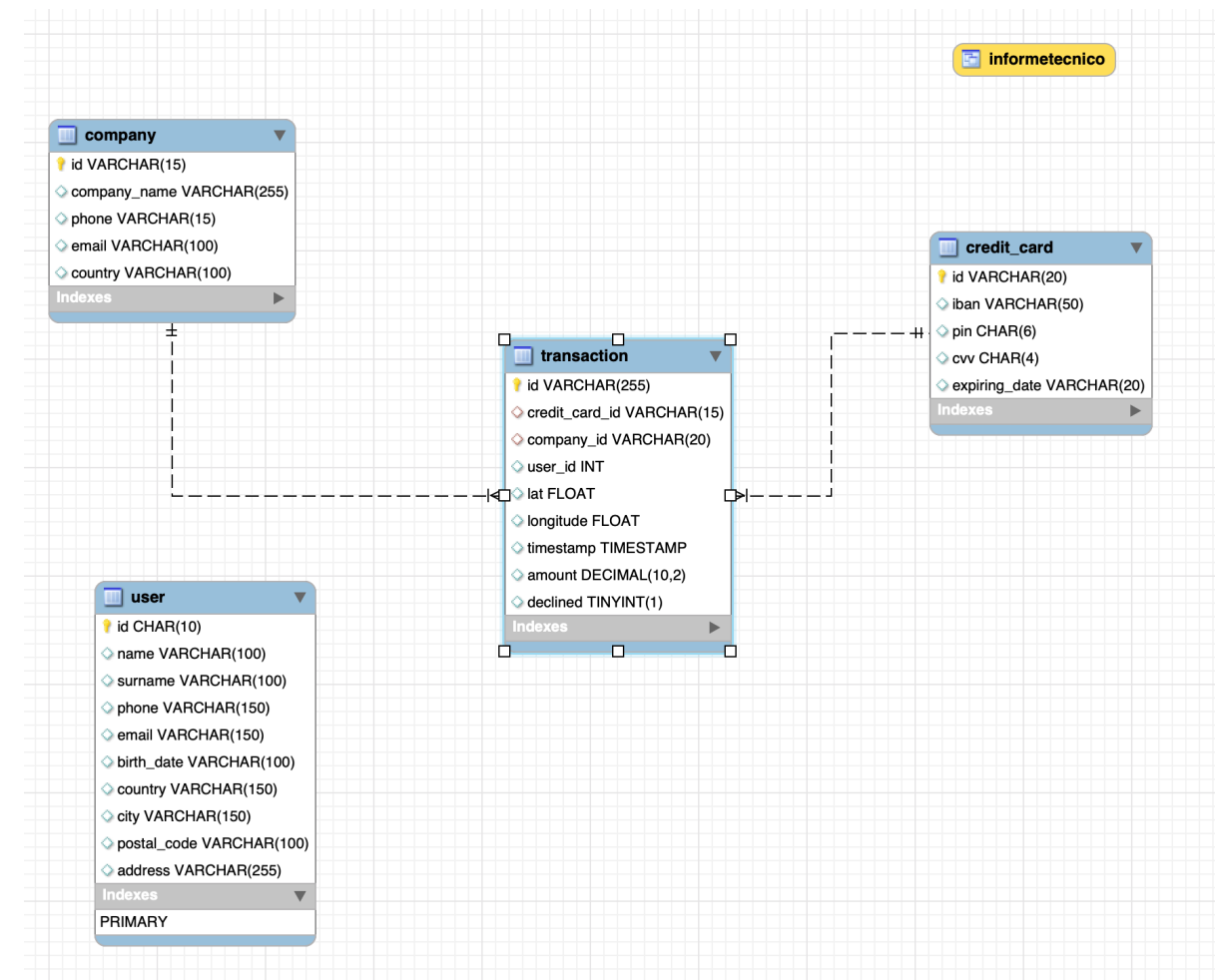


Diagrama inicial

Lo primero que hemos realizado en este ejercicio es crear la tabla user con los documentos adjuntos. Lo que nos ha generado el diagrama inicial realizado en reverse engineer. Lo primero que observamos es que los nombres de las tablas user y data_user son distintos. También observamos que las tablas no están vinculadas. En la tabla compañía no tenemos la columna website. También podemos observar que los tipos de dato o su longitud son distintos en varias columnas y la tabla credit card tiene una columna extra que se llama fecha actual y que la fecha de expiración no tiene formato fecha.

Lo primero que haremos será cambiar el nombre a la tabla user. Para ello utilizamos el comando RENAME TABLE.

```
122 * RENAME TABLE user TO data_user;
```

```
✓ 25363 19:05:16 RENAME TABLE user TO data_user 0 row(s) affected
```

Lo siguiente que intentamos hacer fue crear una Foreign Key para vincular las tablas pero obtuvimos el error 3780

```
135 * ALTER TABLE transaction
136 ADD CONSTRAINT fk_data_user_id
137 FOREIGN KEY (user_id)
138 REFERENCES data_user(id);
```

```
✗ 25366 19:13:02 ALTER TABLE transaction ADD CONSTRAINT fk_data_user_id FOREIGN... Error Code: 3780. Referencing column 'user_id' and referenced column 'id' in fo
```

Este error lo que nos indica es que no se puede crear la Foreign Key porque el tipo de dato es distinto en las columnas que intentamos vincular.

Así que primero debemos de cambiar el tipo de dato de la columna id de tabla user_data a "INT", para que sea igual al user_id de la tabla transaction

```
124 * ALTER TABLE data_user MODIFY COLUMN id INT;
```

```
✓ 25367 19:19:31 ALTER TABLE data_user MODIFY COLUMN id INT 5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0
```

Aprovecharemos para hacer el resto de modificaciones a las otras tablas para dejarlas según el modelo que nos piden.

Utilizaremos la función ALTER TABLE para hacer las modificaciones a las columnas correspondientes. Lo primero que haremos será borrar la columna website de la tabla company con el comando DROP COLUMN.

```
122 • ALTER TABLE company
123     DROP COLUMN website;
```

Lo siguiente es utilizar ALTER TABLE para modificar el tipo de dato de las columnas y para ello utilizamos MODIFY COLUMN.

```
125 • ALTER TABLE credit_card MODIFY COLUMN pin VARCHAR(4);
```

```
127 • ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR(20);
```

Por último utilizaremos ALTER TABLE y la función ADD COLUMN para crear la columna fecha_actual en la tabla credit card

```
133 • ALTER TABLE credit_card
134     ADD COLUMN fecha_actual DATE;
```

Una vez realizadas las transformaciones a las tablas volvemos a intentar crear la Foreign KEY, pero ahora nos da el error 1452

```
✖ 25368 19:20:27 ALTER TABLE transaction ADD CONSTRAINT fk_data_user_id FOREIGN... Error Code: 1452. Cannot add or update a child row: a foreign key constraint fa
```

Este error lo que nos indica es que no se puede crear la Foreign Key porque los datos en las tablas no son concordantes. Eso quiere decir que hay datos en la segunda tabla que no son en la primera y que como la primera tabla tiene una Primary Key no se puede ejecutar la acción. Así que tenemos que encontrar los datos que faltan/sobran. Para ello haremos una selección de los user_id de la tabla transactions y los comparamos con los id de la tabla data_user.

```
150 • SELECT transaction.user_id
151     FROM transaction
152     WHERE transaction.user_id IS NOT NULL
153     AND transaction.user_id NOT IN (SELECT data_user.id FROM data_user)
154     GROUP BY transaction.user_id;
```

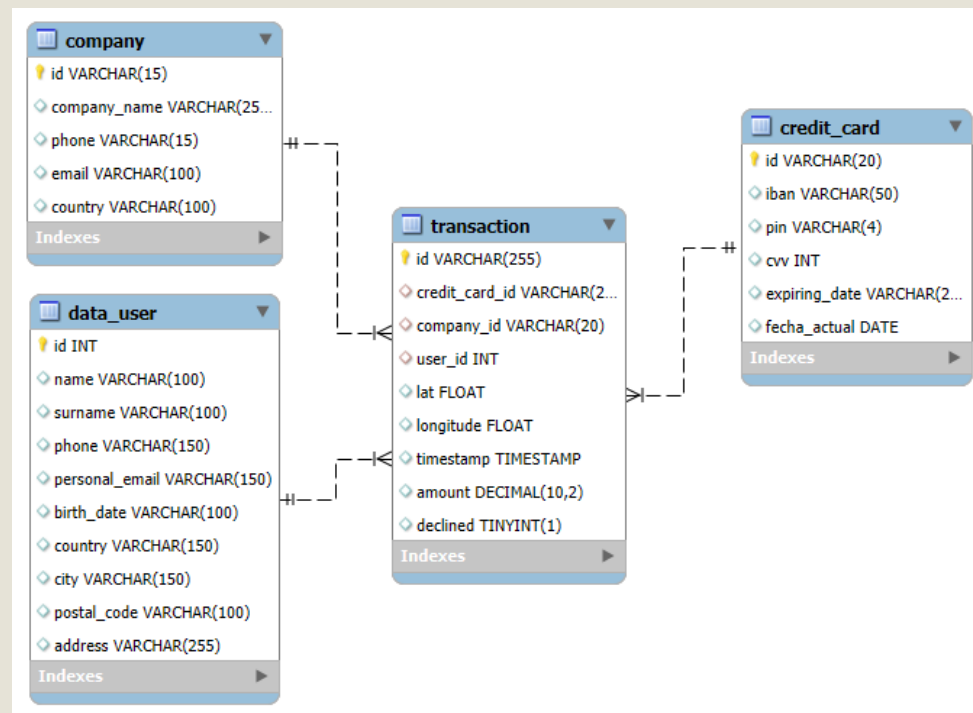



Diagrama modelo

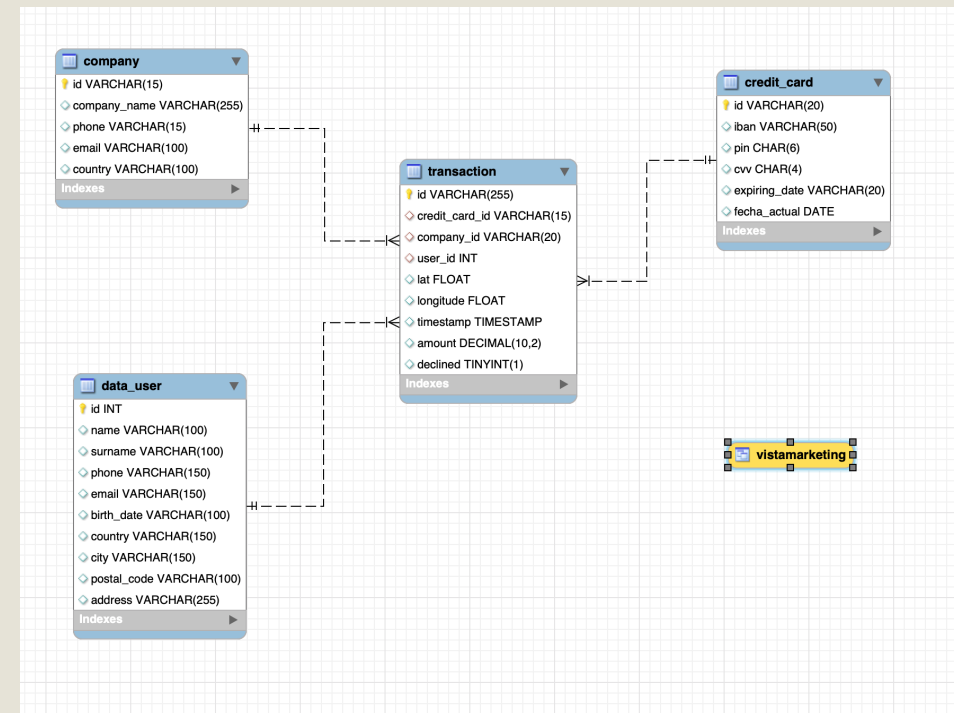


Diagrama final

Y aquí tenemos el resultado final. La columna pin de la tabla credit_card, la hemos dejado en CHAR(6) ya que muchas tarjetas actualmente utilizan pin de 6 dígitos. Y la columna CVV la hemos creado CHAR(4) porque hay algunas tarjetas que ya utilizan cvv de 4 dígitos y también porque si utilizáramos INT, habría un problema con las tarjetas que comienzan por el número cero(0) cuando se realicen búsquedas o comparaciones.

Exercici 2

L'empresa també us demana 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.
- Assegureu-vos d'incloure informació rellevant de les taules que coneixereu i utilitzeu àlies per canviar de nom columnes segons calgui.

Mostra els resultats de la vista, ordena els resultats de forma descendent en funció de la variable ID de transacció.

```
166 • CREATE VIEW InformeTecnico AS
167 SELECT transaction.id,data_user.name as nombre ,data_user.surname as apellido,data_user.country as pais, credit_ca
168 FROM transaction
169 JOIN data_user
170 ON data_user.id=transaction.user_id
171 JOIN credit_card
172 ON credit_card.id=transaction.credit_card_id
173 JOIN company
174 ON company.id=transaction.company_id
175 GROUP BY transaction.id,data_user.name ,data_user.surname, credit_card.iban,company.company_name,transaction.decli
176 ORDER BY transaction.id DESC;
```

	id	nombre	apellido	pais	iban	compañia	declined	amount
▶	FFFD31D6-9495-47CE-B54A-7DB8E1CC274B	Bmrgli	Tprvvmrc	United Kingdom	XX794814451211289182490922	Turpis Company	0	74.54
	FFFCF76D-ECF0-4985-A2D0-B2A7B75998FC	Dfrled	Vilqcjdl	Netherlands	XX636251701647892036676034	Amet Nulla Donec Corporation	0	148.91
	FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	Securp	Faofvqfy	Sweden	XX162677143304223631437567	Nunc Interdum Incorporated	0	234.22
	FFFB270D-F53A-4D5D-9666-E5307C53CC84	Ggzjpa	Uirzjulh	Portugal	XX395114267082019952567052	Viverra Donec Foundation	0	349.13
	FFF9E3CE-234E-408C-A8EF-F9CAD577224A	Yshimq	Zpsjsleed	Germany	XX8845462156537570367941	Convallis In Incorporated	0	247.39
	FFF9E178-6CD2-4DF9-99B0-49AE068809B1	Jevepx	Xwcwzwnm	Netherlands	XX321405515711654384711481	Mus Aenean Eget Foundation	0	438.13
	FFF867C9-17B5-4B1F-AFD9-F8023AAA449E	Fqlngd	Lvhfqyxi	United Kingdom	XX278446342932680979729426	Cras Vehicula Aliquet Industries	0	448.63
	FFF7042D-18C6-4DDD-823C-4D90A4AC8F26	Njoraa	Egsqcuii	Portugal	XX405009272572550082027209	Placerat LLP	0	163.35
	FFF660D4-4244-47F6-9210-E5D1DCB99DB0	Lopzaj	Itgryfay	United States	XX63376659736627454015125	Pede Cum Ltd	0	288.76
	FFF5C660-4441-436D-BD27-E6C53B618622	Gmnbnu	Oxdvhkl	United States	XX237820256172646394016483	At Associates	0	53.31

✓	25388	12:31:23	CREATE VIEW InformeTecnico AS SELECT transaction.id,data_user.na...	0 row(s) affected
✓	25389	12:31:45	SELECT * FROM InformeTecnico	100586 row(s) returned

Para éste ejercicio hemos creado la vista InformeTecnico con la información que nos solicitan utilizando el comando CREATE VIEW y también hemos añadido la columna “país” de la tabla data_user, que nos puede ayudar a realizar análisis más detallados sobre los países desde dónde se realizan las compras. Y hemos añadido tambien la columna declined y amount de la tabla transaction. Que nos pueden servir para realizar análisis sobre el total de las ventas o si hubiera algún problema o fraude con las transacciones declinadas.

Corrección P2P: Jorge Tecocha