

SPRINT 3

Manipulació de taules

Juanjo MONTERO

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.

La taula creada **credit_card** està formada pels camps *id*, *iban*, *pin*, *pan*, *cvv* i *expiring_date* tots ells amb tipus de dades VARCHAR (amb longitud depenent del de seu contigut) excepte el valor *cvv* que té tipus de dades INT.

```
CREATE TABLE IF NOT EXISTS transactions.credit_card (  
    id VARCHAR(15) PRIMARY KEY,  
    iban VARCHAR(50),  
    pin VARCHAR(4),  
    pan VARCHAR(25),  
    cvv INT,  
    expiring_date VARCHAR(20)  
);  
  
-- després de poblar la taula credit_card amb el contingut de datos_introducir_credit.sql  
  
ALTER TABLE transaction  
ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
```

✓	1	13:10:13	CREATE TABLE IF NOT EXISTS transactions.credit_card (...	0 row(s) affected
---	---	----------	---	-------------------

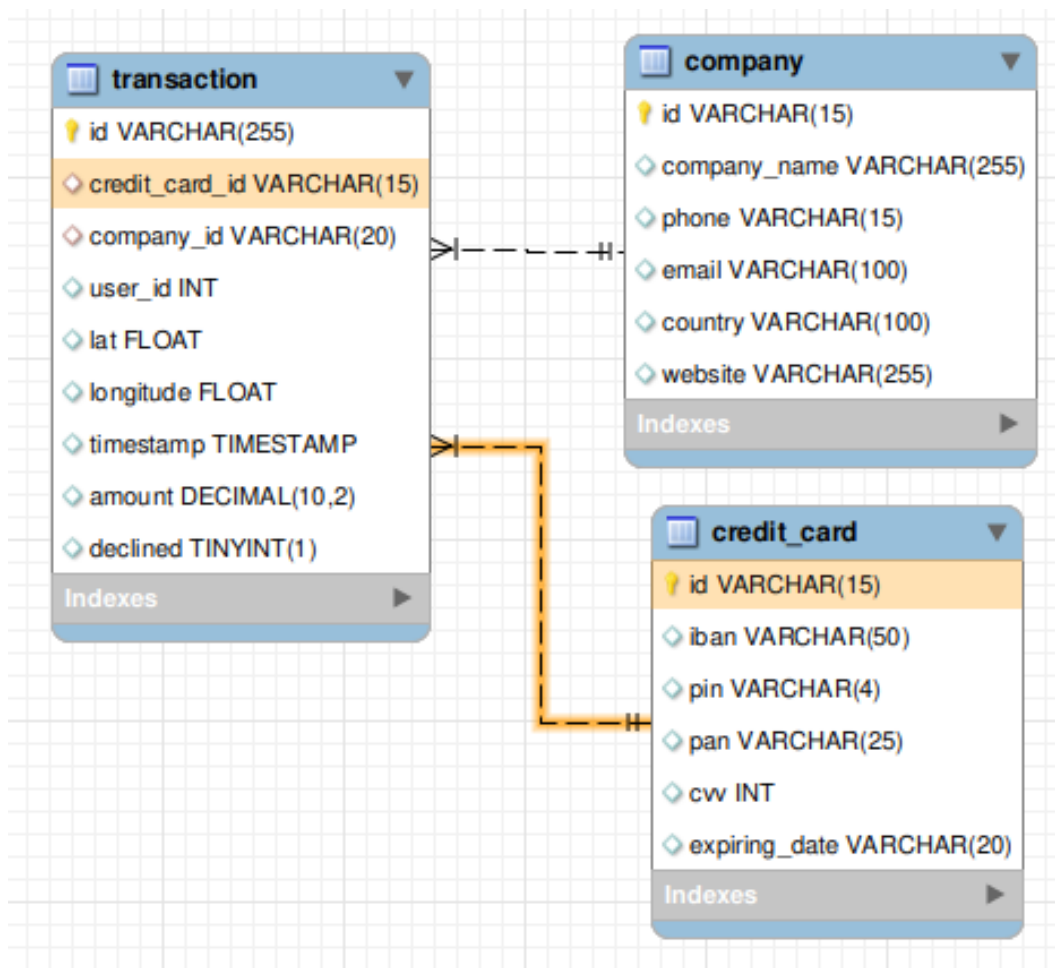
S'introdueixen les dades fent servir l'script "dades_introduir_credit"

✓	1	13:11:22	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) ...	1 row(s) affected
✓	2	13:11:22	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) ...	1 row(s) affected
✓	3	13:11:22	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) ...	1 row(s) affected
* * * * *				
✓	273	13:11:24	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) ...	1 row(s) affected
✓	274	13:11:24	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) ...	1 row(s) affected
✓	275	13:11:24	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) ...	1 row(s) affected

Un cop poblada la taula *credit_card* amb les dades facilitades, s'estableix entre les columnes ***credit_card.id (primary key)*** i ***transaction.credit_card_id (foreign key)*** una relació 1-to-n. Això és així ja que cada transacció tindrà una única targeta de crèdit, però cada targeta es pot fer servir en més d'una transacció

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

El diagrama ER de la base de dades, queda de la següent manera:



NIVELL 1 - 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.

Es fa la modificació a través d'un UPDATE filtrant pel Id del compte.

```
1 • UPDATE credit_card
2   SET iban = "R323456312213576817699999"
3   WHERE id = "CcU-2938";
```

Action Output ▼				
#	Time	Action	Message	Duration
1	13:32:58	UPDATE credit_card SET iban = "R3...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0,0060 s

Després de la modificació amb UPDATE, es comprova que els canvis s'han realitzant correctament amb una ordre SELECT.

```
1 • SELECT * FROM credit_card
2   WHERE id = "CcU-2938";
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

#	id	iban	pin	pan	cvv	expiring_date
1	CcU-2938	R323456312213576817699999	3257	5424465566813633	984	10/30/22
*	NULL	NULL	NULL	NULL	NULL	NULL

credit_card 4

Apply

Revert

Action Output

#	Time	Action	Message	Duration
1	13:35:36	SELECT * FROM credit_card WHERE ...	1 row(s) returned	0,00095

NIVELL 1 - 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

Es deshabiliten les comprovacions de foreign keys, degut a que s'estàn ingressant foreign keys que encara no existeixen com primary keys en les seves respectives taules.

L'ordre que es fa servir és `SET foreign_key_checks = 0;`

Un cop fet això, s'afegeix el nou registre fent servir INSERT INTO.

```
1 • SET foreign_key_checks = 0;
2
3 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
4   VALUES ("108B1D1D-5B23-A76C-55EF-C568E49A99DD", "CcU-9999", "b-9999", 9999, 829.999, -117.999, 111.11, 0 );
5
```

Action Output

	#	Time	Action	Message	Duration / Fetch
✓	1	17:09:14	SET foreign_key_checks = 0	0 row(s) affected	0,00021 sec
✓	2	17:09:23	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)	1 row(s) affected	0,0059 sec

Després d'inserir les dades, es torna a habilitar la comprovació de claus forànies fent servir `SET foreign_key_checks = 1;`

Alternativament, es recomana crear la companyia amb *id* b-9999, per tal de mantenir la integritat referencial de la base de dades.

NIVELL 1 - Exercici 4

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

S'elimina la columna *pan* mitjançant l'ordre DROP COLUMN.

1	•	ALTER TABLE credit_card
2		DROP COLUMN pan;

	#	Time	Action	Message	Duration / Fetch
✓	1	14:17:19	DESCRIBE credit_card	6 row(s) returned	0,0016 sec / 0,00003...
✓	2	14:22:36	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0,206 sec
✓	3	14:22:55	DESCRIBE credit_card	5 row(s) returned	0,0031 sec / 0,00002...

A continuació es mostra la descripció de les columnes abans i després de l'eliminació.

Estat inicial de la taula:

1	•	DESCRIBE credit_card;
---	---	-----------------------

#	Field	Type	Null	Key	Default	Extra
1	id	varchar(15)	NO	PRI	NULL	
2	iban	varchar(50)	YES		NULL	
3	pin	varchar(4)	YES		NULL	
4	pan	varchar(25)	YES		NULL	
5	cvv	int	YES		NULL	
6	expiring_date	varchar(20)	YES		NULL	

Estat final de la taula (sense *pan*):

1	•	DESCRIBE credit_card;
---	---	-----------------------

#	Field	Type	Null	Key	Default	Extra
1	id	varchar(15)	NO	PRI	NULL	
2	iban	varchar(50)	YES		NULL	
3	pin	varchar(4)	YES		NULL	
4	cvv	int	YES		NULL	
5	expiring_date	varchar(20)	YES		NULL	

NIVELL 2 – Exercici 1

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

Fent ús de l'ordre DELETE i filtrant per la *id* de transacció, s'elimina el registre sol·licitat.

```
1 • DELETE FROM transaction
2   WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
3 • SELECT COUNT(*) FROM transaction;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
#	COUNT(*)			
1	587			

	#	Time	Action	Message	Duration / Fetch
✓	1	14:37:58	SELECT COUNT(*) FROM transaction LIMIT 0, 1000	1 row(s) returned	0,0011 sec / 0,00001...
✓	2	14:38:07	DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'	1 row(s) affected	0,0068 sec
✓	3	14:38:14	SELECT COUNT(*) FROM transaction LIMIT 0, 1000	1 row(s) returned	0,00094 sec / 0,0000...

NIVELL 2 - Exercici 2

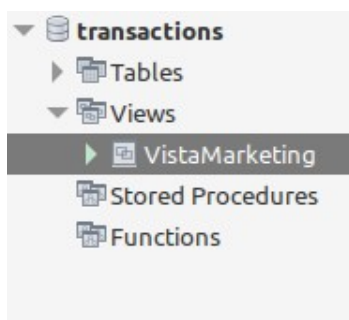
La secció de màrqueting desitja tenir accés a informació específica per a realitzar anàlisi i estratègies efectives. S'ha sol·licitat crear una vista que proporcioni detalls clau sobre les companyies i les seves transaccions. Serà necessària que creïs una vista anomenada VistaMarketing que contingui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

Es crea la vista demanada a través de l'ordre CREATE VIEW.

```
1 • CREATE VIEW VistaMarketing AS
2     SELECT
3         company_name AS "Companyia",
4         phone AS "Telèfon",
5         country AS "País",
6         ROUND(AVG(COALESCE(amount,0)),2) AS "Mitjana de compres"
7     FROM transaction
8     LEFT JOIN company ON company.id = transaction.company_id
9     WHERE transaction.declined = 0
10    GROUP BY 1, 2, 3
11    ORDER BY 4 DESC;
```



1 • `SELECT * FROM transactions.VistaMarketing;`

#	Companyia	Telèfon	País	Mitjana de compres
1	Eget Ipsum Ltd	03 67 44 56 72	United States	481.86
2	Sed Id Limited	07 28 18 18 13	United States	477.51
3	Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.10
4	Nunc Sit Incorporated	07 28 42 63 63	Norway	461.83
5	Non Magna LLC	06 71 73 13 17	United Kingdom	458.74
6	Maecenas Malesuada Fringilla Inc.	09 38 53 76 61	Netherlands	451.29
7	Erat LLP	03 18 88 77 79	Netherlands	448.44
8	Tortor Nunc Commodo Company	05 35 92 77 16	United States	447.11
9	Justo Eu Arcu Ltd	08 42 56 71 52	Italy	444.16
10	Pede Cum Ltd	07 62 26 48 38	Norway	442.32

Es fa servir l'orde COALESCE per tractar com a zero qualsevol possible valor NULL i tenir-lo en compte en el càlcul de la mitjana.

Es filtren aquelles transaccions que han estat rebutjades (*declined* = 1).

NIVELL 2 - Exercici 3

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

Afegint una clàusula WHERE es filtren els resultats de la vista *VistaMarketing*.

```
1 • SELECT * FROM transactions.VistaMarketing
2 WHERE `País` = "Germany";
```

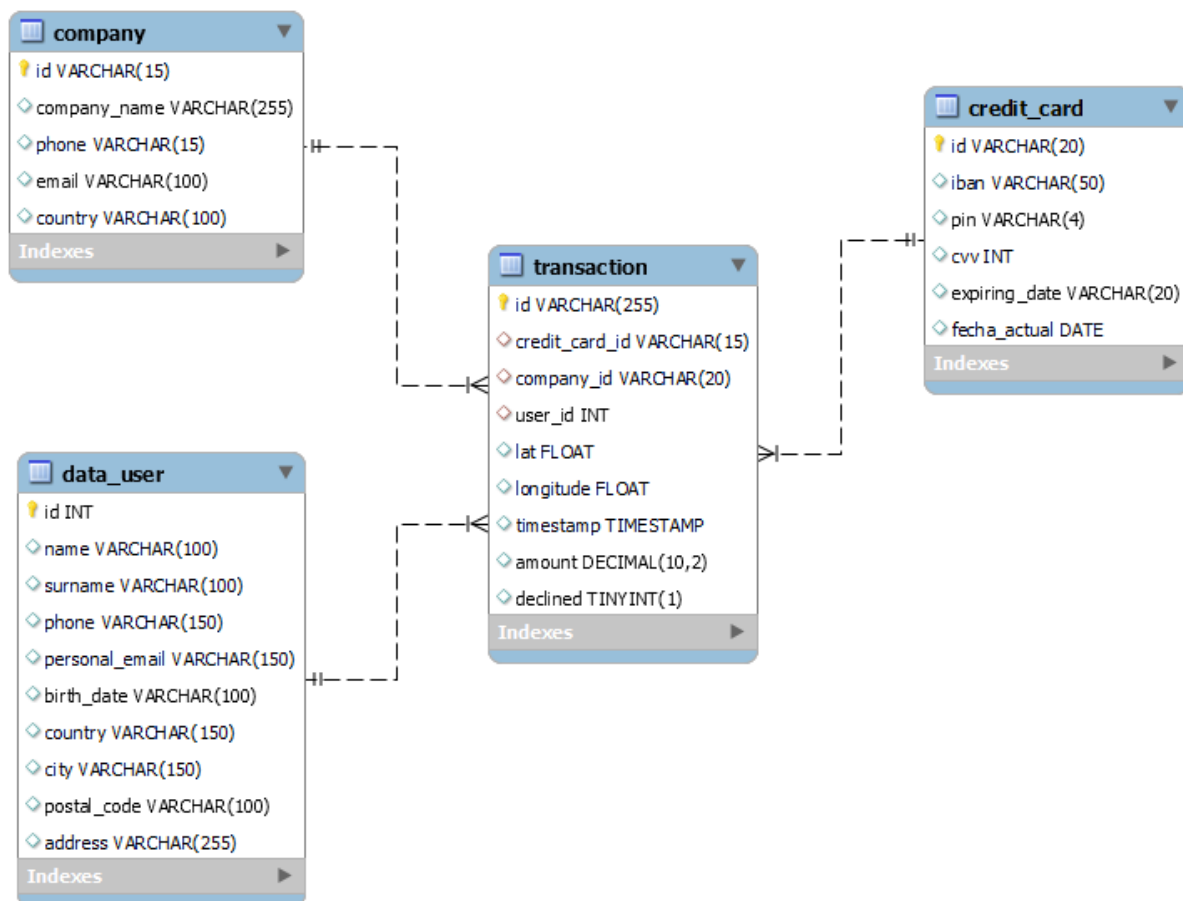
Result Grid					
		Filter Rows:			
		Export:	Wrap Cell Content: IA		
#	Companyia	Telèfon	País	Mitjana de compres	
1	Aliquam PC	01 45 73 52 16	Germany	385.27	
2	Ac Industries	09 34 65 40 60	Germany	289.65	
3	Rutrum Non Inc.	02 66 31 61 09	Germany	266.90	
4	Nunc Interdum Incorporated	05 18 15 48 13	Germany	244.03	
5	Augue Foundation	06 88 43 15 63	Germany	240.80	
6	Ac Fermentum Incorporated	06 85 56 52 33	Germany	206.47	
7	Auctor Mauris Corp.	05 62 87 14 41	Germany	184.31	
8	Convallis In Incorporated	06 66 57 29 50	Germany	156.73	
VistaMarketing 5 ✕					
Action Output ▼					
	#	Time	Action	Message	
✓	1	10:30:34	SELECT * FROM transactions.VistaMarketing WHERE `País` = "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"

(!) Atenció

Es proposen **dos mètodes diferents** per resoldre aquesta activitat. El primer mètode crea tot l'esquema desde la base, segons el disseny proposat.

En el segon mètode, es parteix de l'estat de la base de dades i es modifiquen cada una de les difències que existeixen amb el diagrama ER desitjat.

NIVELL 3 – Exercici 1 (Alternativa A – Creació de l'esquema des de zero)

```
CREATE DATABASE IF NOT EXISTS transactions;  
USE transactions;
```

Es crea la base de dades **transactions**.
La resta de comandaments s'apliquen sobre aquesta base de dades.

```
CREATE TABLE IF NOT EXISTS data_user (  
  id INT PRIMARY KEY,  
  name VARCHAR(100),  
  surname VARCHAR(100),  
  phone VARCHAR(150),  
  personal_email VARCHAR(150),  
  birth_date VARCHAR(100),  
  country VARCHAR(150),  
  city VARCHAR(150),  
  postal_code VARCHAR(100),  
  address VARCHAR(255)  
);
```

Es crea la taula **data_user**.

Aquesta taula compta amb clau primària **id** que identifica de manera única cada usuari.

```
CREATE TABLE IF NOT EXISTS credit_card (  
  id VARCHAR(20) PRIMARY KEY,  
  iban VARCHAR(50),  
  pin VARCHAR(4),  
  cvv INT,  
  expiring_date VARCHAR(20),  
  fecha_actual DATE  
);
```

Es crea la taula **credit_card**.

Aquesta taula compta amb clau primària **id** que identifica de manera única cada targeta de crèdit.

```
CREATE TABLE IF NOT EXISTS company (  
  id VARCHAR(15) PRIMARY KEY,  
  company_name VARCHAR(255),  
  phone VARCHAR(15),  
  email VARCHAR(100),  
  country VARCHAR(100)  
);
```

Es crea la taula **company**.

Aquesta taula compta amb clau primària **id** que identifica de manera única cada companyia.

```
CREATE TABLE IF NOT EXISTS transaction (  
  id VARCHAR(255) PRIMARY KEY,  
  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),  
  FOREIGN KEY (credit_card_id) REFERENCES credit_card(id),  
  FOREIGN KEY (company_id) REFERENCES company(id),  
  FOREIGN KEY (user_id) REFERENCES data_user(id)  
);
```

Es crea la taula **transacció**.

Aquesta taula compta amb clau primària **id** que identifica de manera única cada transacció.

1	N
credit_card.id	credit_card_id
data_user.id	user_id
Company.id	company_id

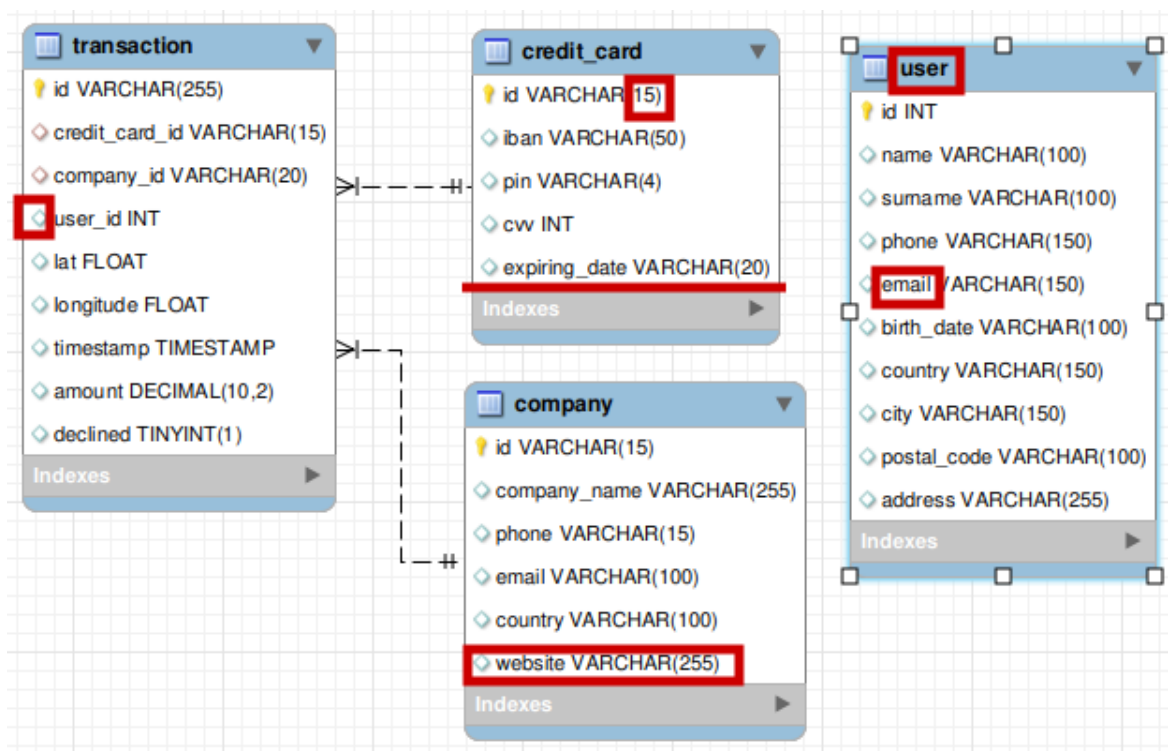
A més a més, es defineixen tres claus forànies (**credit_card_id**, **company_id**, **user_id**) que relacionen les taules **credit_card**, **company** i **data_user** amb la taula **transacció**.

Aquesta solució és menys propensa a arrossegat errors que poguessin existir en les taules i les seves relacions, donat que es desconeixen els canvis que s'havien realitzat prèviament. No obstant, s'ha de tenir en compte al introduir les dades que algunes columnes (com **pan**) han estat eliminades, per la qual cosa es recomana fer servir l'alternativa B.

NIVELL 3 – Exercici 1 (Alternativa B – Modificació de l'esquema existent)

<pre>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));</pre>	<p>En la creació de la taula <i>user</i> utilitzant l'arxiu proporcionat <i>estructura_datos_user.sql</i> es detecta un problema amb la declaració de la clau forània, que està referenciant a la mateixa taula <i>user</i> en lloc de referenciar la taula <i>transaction</i>.</p> <p>S'elimina la línia mostrada per corregir-ho. S'establirà aquesta FOREIGN KEY correctament més endavant.</p>
--	---

Es detecten 6 diferències entre el model actual i l'esquema que es desitja:



Diferència 1:

Es modifica el tipus de dada per *credit_card.id* de VARCHAR(15) a VARCHAR 20

```
ALTER TABLE credit_card MODIFY COLUMN id VARCHAR(20)
```

0 row(s) affected
Records: 0 Duplicates: 0 Warnings: 0

Diferència 2:

Es canvia el nom de la taula *user* per *data_user*

```
ALTER TABLE user RENAME TO data_user
```

0 row(s) affected

Diferència 3:

Es canvia el nom de la columna *data_user.email* per *data_user.personal_email*

```
ALTER TABLE data_user RENAME COLUMN email TO personal_email
```

0 row(s) affected
Records: 0 Duplicates: 0 Warnings: 0

Diferència 4:

S'elimina la columna *company.website*

```
ALTER TABLE company DROP COLUMN website
```

0 row(s) affected
Records: 0 Duplicates: 0 Warnings: 0

Diferència 5:

S'afegeix la columna *credit_card.fecha_actual* amb un tipus de dades DATE

```
ALTER TABLE credit_card ADD fecha_actual DATE
```

0 row(s) affected
Records: 0 Duplicates: 0 Warnings: 0

Diferència 6:

Finalment, s'estableix la relació entre les taules *transaction* i *data_user*.
user_id és clau forània referenciant *data_user.id*

Es desactiva momentàniament la comprovació de claus forànies ja que existeixen transaccions amb usuaris que encara no consten en la taula *data_user*.

```
SET foreign_key_checks = 0
```

0 row(s) affected

```
ALTER TABLE transaction ADD FOREIGN KEY (user_id) REFERENCES data_user(id)
```

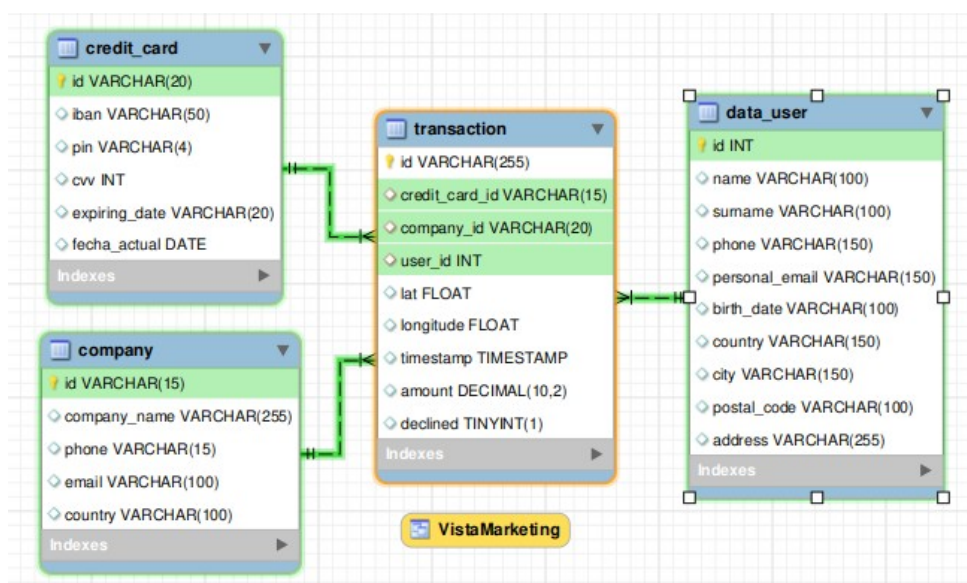
0 row(s) affected
Records: 0 Duplicates: 0 Warnings: 0

```
SET foreign_key_checks = 1
```

0 row(s) affected

NIVELL 3 – Exercici 1 (Conclusió)

Indistintament de la solució escollida, l'esquema ER final de la base de dades transactions queda com segueix:



NIVELL3 - 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.

```
1 • CREATE VIEW InformeTecnico AS
2 SELECT
3     transaction.id AS "Identificador de Transacció",
4     data_user.name AS "Nom",
5     data_user.surname AS "Cognom",
6     credit_card.iban AS "IBAN",
7     company.company_name AS "Companyia"
8 FROM transaction
9 JOIN data_user ON transaction.user_id = data_user.id
10 JOIN credit_card ON transaction.credit_card_id = credit_card.id
11 JOIN company ON transaction.company_id = company.id
12 ORDER BY 1 DESC;
```

Action Output ▼

	#	Time	Action	Message
✓	1	13:50:29	CREATE VIEW InformeTecnico AS SELECT transaction.id AS "Identificador de Tra...	0 row(s) affected

```
1 • SELECT * FROM transactions.InformeTecnico;
```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

#	Identificador de Transacció	Nom	Cognom	IBAN	Companyia
1	FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
2	FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
3	FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955...	Nunc Interdum Incorporated
4	FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
5	FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
6	FCE2AB9A-271D-2BDC-9E49-8DD92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated
7	FBD7E0D6-BA6B-F5BC-0CA9-EA4B8760100C	Hedwig	Gilbert	MU4132333444534342541344788...	Mauris Id Inc.
8	FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF588682005757716345...	Arcu LLP
9	FAAD3FFC-1A17-E141-43D3-359A5BA7CB3B	Hedwig	Gilbert	GE90157928843338134463	Lorem Eu Incorporated
10	FA053936-75D8-85FA-490D-9B624E1B920A	Hedwig	Gilbert	GT02497653655330848247645975	Non Justo Corp.

InformeTecnico 1 ✕

Action Output ▼

	#	Time	Action	Message
✓	1	13:50:29	CREATE VIEW InformeTecnico AS SELECT transaction.id AS "Identificador de Tra...	0 row(s) affected
✓	2	13:53:17	SELECT * FROM transactions.InformeTecnico LIMIT 0, 1000	586 row(s) returned

