

# ***SPRINT 3***

---

**Edoardo Brega**

# ★ 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 ingessis la informació del document denominat "dades\_introduir\_credit". Recorda mostrar el diagrama i realitzar una breu descripció d'aquest.

### ❖ creo la nueva tabla credit\_card

The screenshot shows a database management interface. On the left, a tree view displays the database structure, including tables like 'company', 'credit\_card', 'transaction', and 'user'. The 'credit\_card' table is selected. On the right, the SQL editor shows the following code:

```
1 CREATE INDEX idx_credit_card ON transaction(credit_card_id);
2 CREATE TABLE credit_card (
3     id VARCHAR(100) NOT NULL PRIMARY KEY,
4     iban VARCHAR(100),
5     pan VARCHAR(100),
6     pin int,
7     cvv int,
8     expiring_date VARCHAR(100),
9     FOREIGN KEY(id) REFERENCES transaction(credit_card_id)
10 );
11
```

Below the SQL editor, the 'Output' window shows the execution results:

#	Time	Action	Message
1	11:09:59	CREATE INDEX idx_credit_card ON transaction(credit_card_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
2	11:09:59	CREATE TABLE credit_card ( id VARCHAR(100) NOT NULL PRIMARY ...	0 row(s) affected

Below the output, the 'Table: credit\_card' structure is shown:

Columns:	
id	varchar(100) PK
iban	varchar(100)
pan	varchar(100)
pin	int
cvv	int
expiring_date	varchar(100)

- código:  

```
CREATE INDEX idx_credit_card ON transaction(credit_card_id);
CREATE TABLE credit_card (
    id VARCHAR(100) NOT NULL PRIMARY KEY,
    iban VARCHAR(100),
    pan VARCHAR(100),
    pin int,
    cvv int,
    expiring_date VARCHAR(100),
    FOREIGN KEY(id) REFERENCES transaction(credit_card_id)
);
```
- explicación:  
Empiezo añadiendo el index necesario para buscar rápidamente filas con valores de columna específicos.  
Creo nueva tabla 'Credit\_card'.  
A la columna 'Id' que identifica la tarjeta de credito le impongo que tenga un valor ('Not Null') y que sea la clave primaria ('Primary Key').  
Identifico la columna 'Expiring\_date' como tipo 'Varchar' porque los valores de ingresar en ese campo no están ordenados para el tipo 'Date'.  
Relaciono la columna 'Id' con la columna 'Credit\_card\_id' de la tabla 'Transaction' que está relacionada con la tabla 'Company' por las columnas 'Transaction.company\_id=company.id'.

## ❖ inserto los valores descargados

```

1
2 -- Insertamos datos de credit_card
3 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2938', 'TR301950312213576817638661', '542
4 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2945', 'DO26854763748537475216568689', '5
5 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2952', 'BG45IVQL52710525608255', '4556 45
6 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2959', 'CR7242477244335841535', '37246137
7 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2966', 'BG72LKTQ15627628377363', '448566
8 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2973', 'PT87806228135092429456346', '544
9 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2980', 'DE39241881883086277136', '402400
10 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2987', 'GE89681434837748781813', '3763 74
11 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2994', 'BH62714428368066765294', '3442832

```

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
✓ 267	11:10:57	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( ...	1 row(s) affected	0.000 sec
✓ 268	11:10:57	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( ...	1 row(s) affected	0.016 sec
✓ 269	11:10:57	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( ...	1 row(s) affected	0.000 sec
✓ 270	11:10:57	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( ...	1 row(s) affected	0.000 sec
✓ 271	11:10:57	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( ...	1 row(s) affected	0.000 sec
✓ 272	11:10:57	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( ...	1 row(s) affected	0.000 sec

## ❖ selecciono los valores ingresados con la fecha de tipo texto (VARCHAR)

```

1 • SELECT * FROM transactions.credit_card;

```

Result Grid						
Filter Rows:						
Edit: Export/Import: Wrap Cell Content:						
	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22
	CcU-2945	DO26854763748537475216568689	5142423821948828	9080	887	08/24/23
	CcU-2952	BG45IVQL52710525608255	4556 453 55 5287	4598	438	06/29/21
	CcU-2959	CR7242477244335841535	372461377349375	3583	667	02/24/23
	CcU-2966	BG72LKTQ15627628377363	448566 886747 7265	4900	130	10/29/24
	CcU-2973	PT87806228135092429456346	544 58654 54343 384	8760	887	01/30/25
	CcU-2980	DE39241881883086277136	402400 7145845969	5075	596	07/24/22
	CcU-2987	GE89681434837748781813	3763 747687 76666	2298	797	10/31/23
	CcU-2994	BH62714428368066765294	344283273252593	7545	595	02/28/22
	CcU-3001	CY49087426654774581266832110	511722 924833 2244	9562	867	09/16/22

credit_card 10				
Output				
Action Output				
#	Time	Action	Message	
✓ 276	11:23:40	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( ...	1 row(s) affected	
✓ 277	11:23:40	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( ...	1 row(s) affected	
✓ 278	11:23:59	SELECT * FROM transactions.credit_card LIMIT 0, 500	275 row(s) returned	

❖ modifiko el tipo de valor contenido en la columna 'Expiring\_date'

```
16
17 • update credit_card
18   set expiring_date = str_to_date(expiring_date, '%m/%d/%Y');
19
```

Output			
Action Output			
#	Time	Action	Message
✓ 1	11:26:37	update credit_card set expiring_date = str_to_date(expiring_date, '%m/%d/%Y')	275 row(s) affected Rows matched: 275 Changed: 275 Warnings: 0

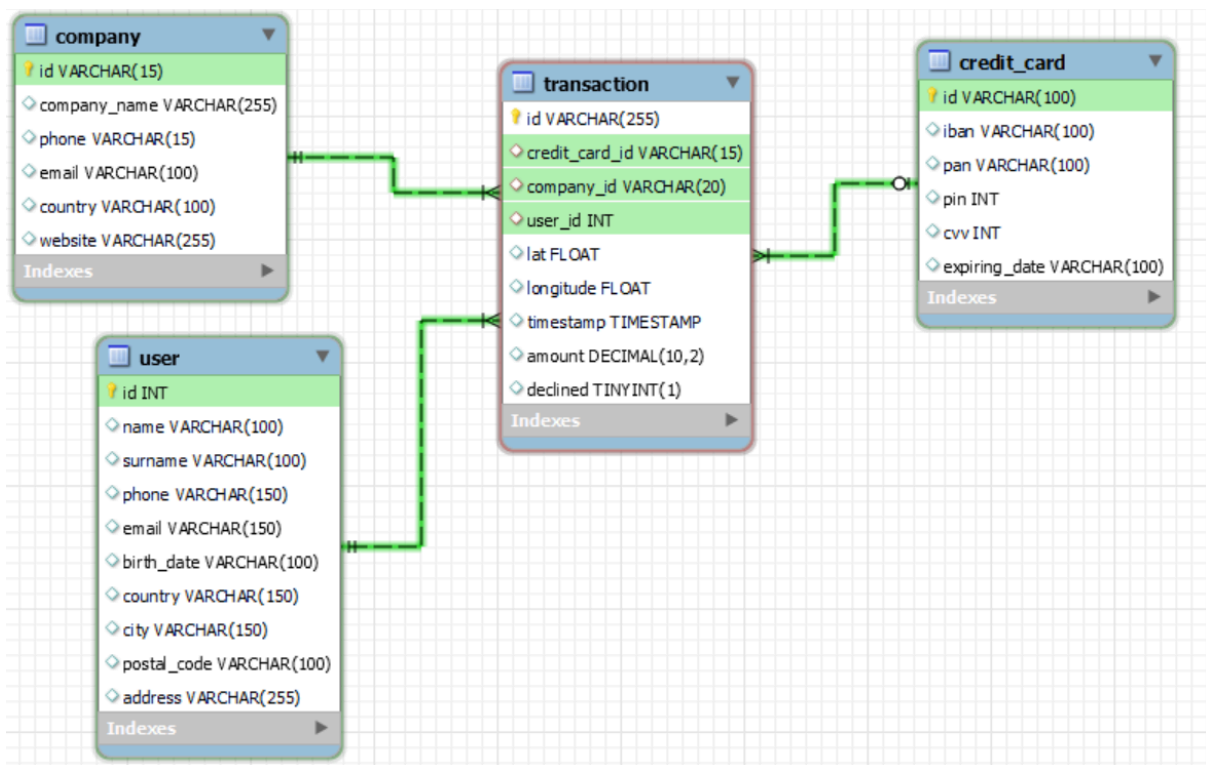
❖ muestro los valores de la tabla para verificar el cambio en la columna 'expiring\_date'

```
1 • SELECT * FROM transactions.credit_card;
2
```

Result Grid						
Filter Rows:						
Edit: Export/Import: Wrap Cell Content: A						
	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	2022-10-30
	CcU-2945	DO26854763748537475216568689	5142423821948828	9080	887	2023-08-24
	CcU-2952	BG451VQL52710525608255	4556 453 55 5287	4598	438	2021-06-29
	CcU-2959	CR7242477244335841535	372461377349375	3583	667	2023-02-24
Output						
Action Output						
#	Time	Action	Message			
✓ 1	11:26:37	update credit_card set expiring_date = str_to_date(expiring_date, '%m/%d/%Y')	275 row(s) affected Rows matched: 275 Changed: 275 Warnings: 0			
✓ 2	11:27:11	SELECT * FROM transactions.credit_card LIMIT 0, 500	275 row(s) returned			

- código:  
update credit\_card  
set expiring\_date = str\_to\_date(expiring\_date, '%m/%d/%Y');
- explicación:  
Modifico la columna 'Expiring\_date' para que el sistema la reconozca como fecha y poder trabajar con ella.

❖ **muestro el modelo creado**



- El modelo del diagrama es un modelo estrella. El modelo estrella es una técnica de modelado de datos que se utiliza para diseñar y optimizar almacenes de datos y data marts. Su nombre se debe a la forma que tiene el esquema lógico, que consta de una tabla central llamada tabla de hechos y varias tablas periféricas llamadas tablas de dimensiones. La tabla de hechos es 'Transaction', las otras son tablas de dimensiones.

Las relaciones entre las tablas son:

transaction.company\_id - N a 1 - company.id

transaction.credit\_card\_id - N a 1 - credit\_card.id

transaction.user\_id - N a 1 - user.id

Una relación 'N a 1' es aquella que vincula un número indeterminado de registros de la primera tabla con un único registro de la segunda, y vincula un registro de la segunda tabla con un número indeterminado de registros de la primera.

## → 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.

❖ selecciono el registro que hay que modificar

```
20
21 • select*from credit_card where id = 'CcU-2938';
22
```

Result Grid						
Filter Rows: <input type="text"/>						
Edit:						
Export/Import:						
Wrap Cell Content:						
	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	2022-10-30
*	NULL	NULL	NULL	NULL	NULL	NULL

credit_card 3 x			
Output			
Action Output			
#	Time	Action	Message
✓ 1	12:57:04	select*from credit_card where id = 'CcU-2938' LIMIT 0, 500	1 row(s) returned

❖ modifiko el registro y lo muestro a video

```
26
27 • update credit_card set iban = 'R323456312213576817699999'
28   where id = 'CcU-2938';
29 • select*from credit_card where id = 'CcU-2938';
30
```

Result Grid						
Filter Rows: <input type="text"/>						
Edit:						
Export/Import:						
Wrap Cell Content:						
	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	R323456312213576817699999	5424465566813633	3257	984	2022-10-30
*	NULL	NULL	NULL	NULL	NULL	NULL

credit_card 4 x			
Output			
Action Output			
#	Time	Action	Message
✓ 1	12:59:00	update credit_card set iban = 'R323456312213576817699999' where id = 'CcU-2938'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
✓ 2	12:59:00	select*from credit_card where id = 'CcU-2938' LIMIT 0, 500	1 row(s) returned

➤ código:

```
update credit_card set iban = 'R323456312213576817699999'
where id = 'CcU-2938';
```

➤ explicación:

Utilizo la vista coCon 'Update' puedo modificar uno o más registros de una tabla seleccionada con el valor definido con 'Set'. Con 'Where' limito el cambio a los registros que cumplen la condición.

## → 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

```
14
15
16
17 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES
18 ( '108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '999', '829.999', '-117.999', null , '111.11', '0');
19
20 • SELECT * FROM transactions.transaction WHERE id='108B1D1D-5B23-A76C-55EF-C568E49A99DD';
21
```

Result Grid									
Filter Rows:									
Edit: Export/Import: Wrap Cell Content:									
id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	
108B1D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	999	829.999	-117.999	NULL	111.11	0	
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

transaction 1 x									
Output									
Action Output									
#	Time	Action	Message	Duration / Fetch					
✓ 15	13:20:27	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, ti...	1 row(s) affected	0.000 sec					
✓ 16	13:21:46	SELECT * FROM transactions.transaction WHERE id='108B1D1D-5B23-A76C-55E...	1 row(s) returned	0.000 sec / 0.000 s					

- código:  
INSERT INTO transaction (id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, amount, declined) VALUES ( '108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '999', '829.999', '-117.999', null , '111.11', '0');
- explicación:  
Inserto los valores según el orden de la tabla. Dejo la columna 'Timestamp' en 'Null' porque no viene especificado el valor de insertar y la columna admite campos vacíos.

## → exercici 4

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

```
7 • ALTER TABLE credit_card DROP pan;  
8 • select* from credit_card;  
9
```

The screenshot shows a database management interface. At the top, there's a toolbar with icons for 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. Below the toolbar, a table displays the contents of the 'credit\_card' table. The table has columns: id, iban, pin, cvv, and expiring\_date. The data is as follows:

id	iban	pin	cvv	expiring_date
CcJ-2938	R323456312213576817699999	3257	984	2022-10-30
CcJ-2945	DO26854763748537475216568689	9080	887	2023-08-24
CcJ-2952	BG45IVQL52710525608255	4598	438	2021-06-29
CcJ-2959	CR7242477244335841535	3583	667	2023-02-24
CcJ-2966	BG72LKTQ15627628377363	4900	130	2024-10-29

Below the table, there's a section labeled 'credit card 1 x' with a sub-section 'Output'. The 'Output' section shows the results of the SQL commands executed:

#	Time	Action	Message
17	13:37:29	ALTER TABLE credit_card DROP pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
18	13:37:49	select* from credit_card LIMIT 0, 500	275 row(s) returned

- código:  
ALTER TABLE credit\_card DROP pan;
- explicación:  
Con 'Alter Table' selecciono la tabla que quiero modificar y con 'DROP' elimino la columna que especifico despues.

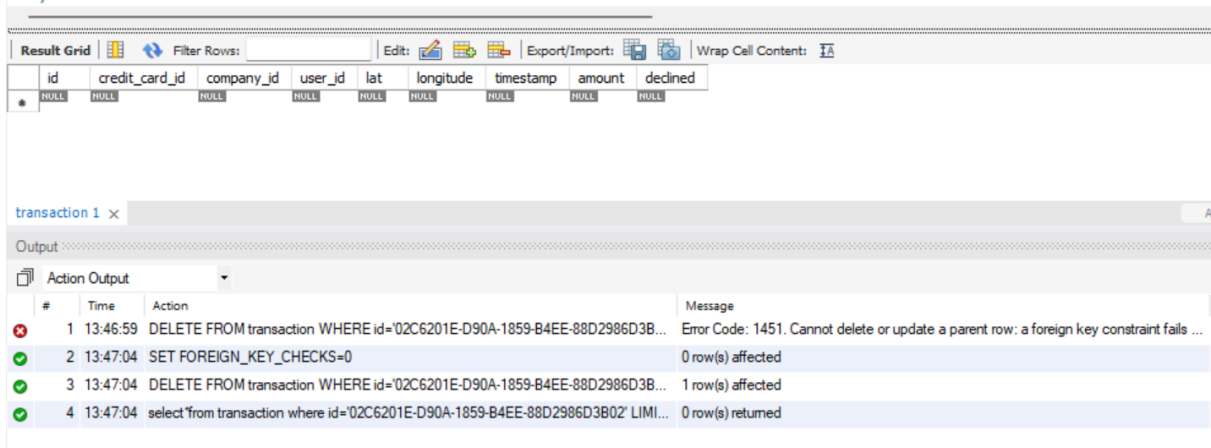


## ★ NIVELL 2

### → exercici 1

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

```
2
3 • DELETE FROM transaction WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
4 • SET FOREIGN_KEY_CHECKS=0;
5 • DELETE FROM transaction WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
6 • select*from transaction where id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
7
```



#	Time	Action	Message
1	13:46:59	DELETE FROM transaction WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B...	Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ...
2	13:47:04	SET FOREIGN_KEY_CHECKS=0	0 row(s) affected
3	13:47:04	DELETE FROM transaction WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B...	1 row(s) affected
4	13:47:04	select*from transaction where id='02C6201E-D90A-1859-B4EE-88D2986D3B02' LIM...	0 row(s) returned

- código:  
DELETE FROM transaction WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
- explicación:  
Para poder eliminar un registro primero tengo que hacer "SET FOREIGN\_KEY\_CHECKS=0". Después con 'Delete' selecciono la tabla que quiero cancelar y con la cláusula 'Where' especifico cuáles registros eliminar.  
Si intentara eliminar el registro sin modificar el control 'Foreing key' me daría el error que se ve en la primera riga de 'Action output'.

## → 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.

### ❖ creo la vista

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

- SCHEMAS** (Left Panel): Shows a tree view of the database structure. Under 'Views', a new view named 'vistamarketing' has been created. It contains four columns: 'Nom de la companyia', 'Telèfon de contacte', 'País de residència', and 'Mitjana de compra'.
- Code Editor** (Center): Contains the SQL script for creating the view:

```
17
18 • CREATE VIEW VistaMarketing AS
19 SELECT company_name 'Nom de la companyia', phone 'Telèfon de contacte',
20        country 'País de residència', avg(amount) 'Mitjana de compra'
21 FROM company
22 JOIN transaction ON transaction.company_id = company.id
23 WHERE declined = 0
24 GROUP BY company_id
25 ORDER BY avg(amount) desc;
26
```
- Output** (Bottom Panel): Shows the execution results. A table with columns '#', 'Time', 'Action', and 'Message' displays the following row:

#	Time	Action	Message
1	10:39:48	CREATE VIEW VistaMarketing AS SELECT company_name 'Nom de la companyia'...	0 row(s) affected

❖ muestro el contenido de la vista

```
1 • SELECT * FROM transactions.vistamarketing;
2
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Nom de la companyia	Telèfon de contacte	País de residència	Mitjana de compra
▶	Eget Ipsum Ltd	03 67 44 56 72	United States	481.860000
	Sed Id Limited	07 28 18 18 13	United States	477.510000
	Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.100000
	Nunc Sit Incorporated	07 28 42 63 63	Norway	461.830000

stamarketing 1 x

Output

Action Output

#	Time	Action	Message
✓ 1	10:39:48	CREATE VIEW VistaMarketing AS SELECT company_name'Nom de la companyi...',phone'Telèfon de contacte',country'País de residència',avg(amount)'Mitjana de compra' FROM company join transaction on transaction.company_id = company.id where declined = 0 group by company_id order by avg(amount) desc;	0 row(s) affected
✓ 2	10:41:06	SELECT * FROM transactions.vistamarketing LIMIT 0, 500	100 row(s) returned

- código:  
CREATE VIEW VistaMarketing AS  
SELECT company\_name'Nom de la companyia',phone'Telèfon de contacte',  
country'País de residència',avg(amount)'Mitjana de compra'  
FROM company  
join transaction on transaction.company\_id = company.id  
where declined = 0  
group by company\_id  
order by avg(amount) desc;
- explicación:  
Para crear una nueva vista utilizo 'Create view' seguido del nombre de la vista.  
Selecciono las columnas que quiero mostrar de la tabla 'Company' y a través de la relacion  
'transaction.company\_id = company.id' con la tabla 'Transaction' puedo mostrar la media de  
las transacciones agrupadas por compañía.  
Hago un 'Join' simple (Inner) teniendo en cuenta que ya está comprobado por la etrutura de  
las tablas y por los ejercicios precedentes que no hay compañías sin transacciones y no hay  
trasacciones sin compañía.  
En caso contrario, se podría utilizar 'Full Join' para incluir todos los valores, aunque no esten  
relacionados.

### → exercici 3

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

❖ selecciono los valores filtrados

```
1 • SELECT * FROM transactions.vistamarketing
2   where `País de residència`='germany';
```

Result Grid				
Filter Rows:		Export:	Wrap Cell Content:	
	Nom de la companyia	Telèfon de contacte	País de residència	Mitjana de compra
▶	Ac Industries	09 34 65 40 60	Germany	396.150000
	Auctor Mauris Corp.	05 62 87 14 41	Germany	308.990000
	Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.570000
	Aliquam PC	01 45 73 52 16	Germany	280.340000

vistamarketing 2 x

Output

Action Output

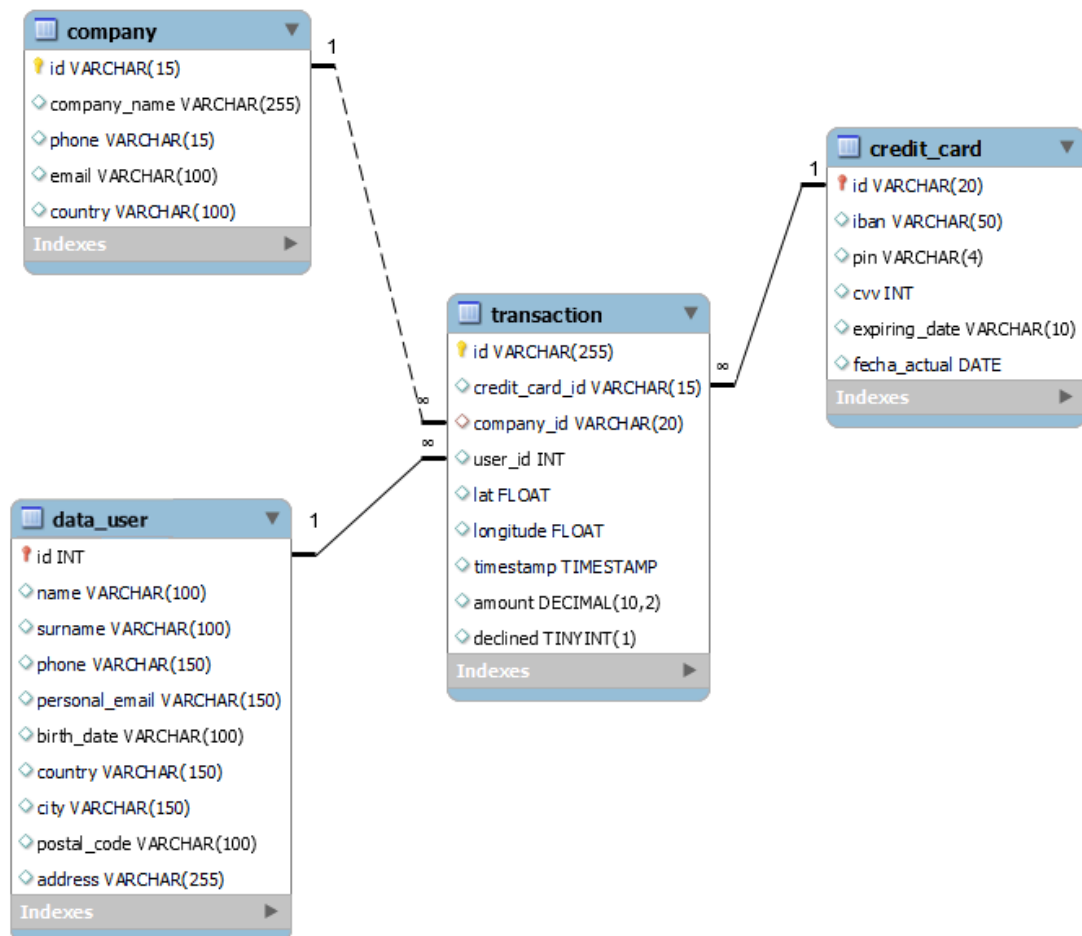
#	Time	Action	Message
✓ 1	11:00:14	SELECT * FROM transactions.vistamarketing where `País de residència`='german...	8 row(s) returned

- código:  
SELECT \* FROM transactions.vistamarketing  
where `País de residència`='germany';
- explicación:  
Utilizo la vista como si fuera una tabla: selecciono todos los registros que cumplen la cláusula contenida en el 'Where'.

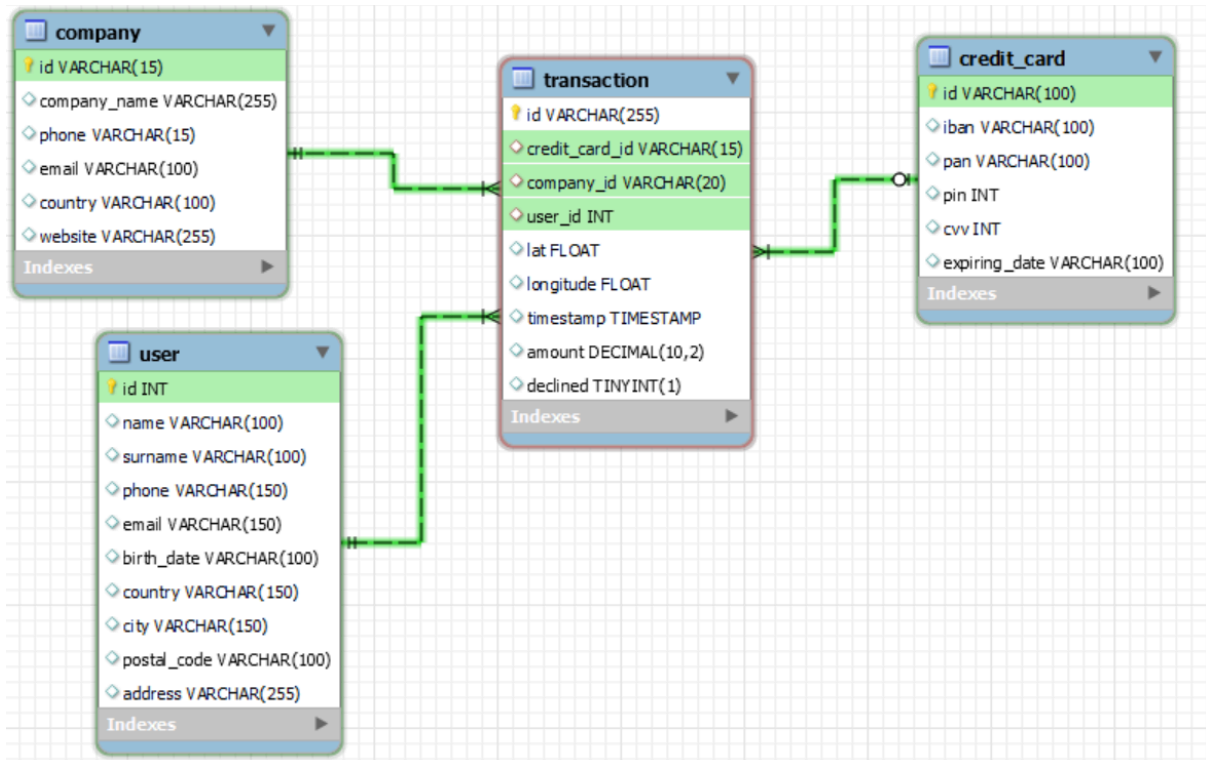
## ★ NIVELL 3

### → 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:



❖ muestro el diagrama de base



❖ muestro el código

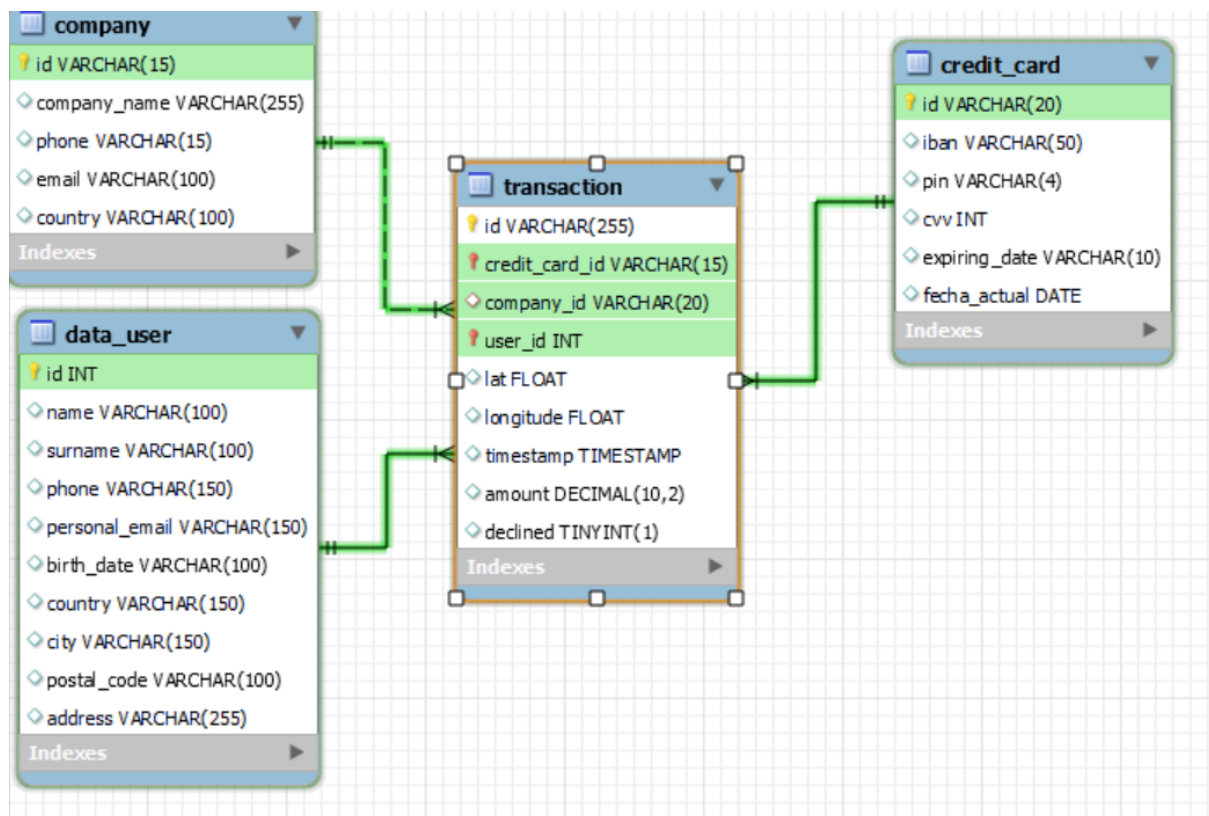
```

1 • SET FOREIGN_KEY_CHECKS=0;
2   -- tabla company elimino columna 'website'
3 • ALTER TABLE company DROP website;
4   -- tabla user cambio nombre tabla
5 • alter table user rename data_user;
6   -- tabla user cambio nombre columna 'email'
7 • alter table data_user rename column email to personal_email;
8   -- tabla credit_card modifiko los tipos de columnas
9 • alter table credit_card modify id VARCHAR(20);
10 • alter table credit_card modify iban VARCHAR(50);
11 • alter table credit_card modify pin VARCHAR(4);
12 • alter table credit_card modify expiring_date VARCHAR(10);
13   -- tabla credit card añado columna 'fecha_actual'
14 • alter table credit_card add column fecha_actual date;
15 --
  
```

Output

#	Time	Action	Message
✓ 20	10:36:07	alter table credit_card modify iban VARCHAR(50)	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0
✓ 21	10:36:16	alter table credit_card modify pin VARCHAR(4)	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0
✓ 22	10:36:57	alter table credit_card modify expiring_date VARCHAR(10)	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0
✓ 23	10:39:03	alter table credit_card add column fecha_actual date	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

❖ muestro el resultado



➤ código:

```
SET FOREIGN_KEY_CHECKS=0;
alter table company drop website;
alter table user rename data_user;
alter table data_user rename column email to personal_email;
alter table credit_card modify id VARCHAR(20);
alter table credit_card modify iban VARCHAR(50);
alter table credit_card modify pin VARCHAR(4);
alter table credit_card modify expiring_date VARCHAR(10);
alter table credit_card add column fecha_actual date;
```

➤ explicación:

Primero desactivo el control de las claves externas para poder hacer las modificaciones.  
 Para eliminar la columna 'website' utilizo 'alter table \*\* drop \*\*'.  
 Para renombrar la tabla 'user' utilizo 'alter table \*\* rename \*\*'.  
 Para renombrar la columna 'email' utilizo 'alter table \*\* rename column \*\* to \*\*'.  
 Para cambiar los tipos de columnas en la tabla 'credit\_card' utilizo 'alter table \*\* modify \*\* \*\*'.  
 Para añadir una columna a la tabla 'credit\_card' utilizo 'alter table \*\* add column \*\* \*\*'.

## → 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.

❖ muestro el codigo

```
12
13 • CREATE VIEW InformeTecnico AS
14 SELECT transaction.id 'ID de la transacció',data_user.name'Nom de l'usuari/ària',
15 data_user.surname'Cognom de l'usuari/ària',credit_card.iban'IBAN de la targeta de crèdit usada',
16 company.company_name'Nom de la companyia de la transacció realitzada'
17 FROM transaction
18 join data_user on data_user.id=transaction.user_id
19 join credit_card on credit_card.id=transaction.credit_card_id
20 join company on transaction.company_id = company.id
21 where declined = 0
22 order by transaction.id desc;
23
```

Output

Action Output

#	Time	Action	Message
1	11:24:56	CREATE VIEW InformeTecnico AS SELECT transaction.id 'ID de la transacció',data_...	0 row(s) affected

❖ muestro el resultado

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

Result Grid

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
FE809ED4-2DB6-55AC-C915-929516E46468	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
FD2E8957-4148-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCE2AB9A-271D-2BDC-9E49-8DD92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated
FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF58868200575771634583813	Arcu LLP

informetecnico 1 x

Output

Action Output

#	Time	Action	Message	Du
1	11:26:44	SELECT * FROM transactions.informetecnico LIMIT 0, 500	499 row(s) returned	0.0



➤ código:

```
CREATE VIEW InformeTecnico AS
SELECT transaction.id 'ID de la transacció',data_user.name'Nom de l'usuari/ària',
data_user.surname'Cognom de l'usuari/ària',credit_card.iban'IBAN de la targeta de crèdit
usada',company.company_name'Nom de la companyia de la transacció realitzada'
FROM transaction
join data_user on data_user.id=transaction.user_id
join credit_card on credit_card.id=transaction.credit_card_id
join company on transaction.company_id = company.id
where declined = 0
order by transaction.id desc;
```

➤ explicación:

Para crear una nueva vista utilizo 'Create View \*\* as'.  
Después selecciono todas las columnas necesarias a mostrar las informaciones requeridas.  
Utilizo 'Join' para poder mostrar las informaciones de todas las tablas relacionadas.  
Añado la cláusula 'Where' para no incluir las transacciones denegadas.  
Ordeno todo por el Id de la transacción.