

# 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 the MySQL Workbench interface with the SQL Editor tab selected. The code in the editor is:

```
1 • CREATE TABLE IF NOT EXISTS credit_card(
2     id VARCHAR(20) PRIMARY KEY,
3     iban VARCHAR(50),
4     pan VARCHAR(23),
5     pin VARCHAR(4),
6     cvv INT,
7     expiring_date VARCHAR(20)
8 );
```

The output pane shows a single query execution:

Action Output	#	Time	Action	Message	Duration / Fetch
58	18:12:43		SELECT c.id, c.company_name, COUNT(t.id) AS 'Quantitat transaccions', CASE W...	100 row(s) returned	0.203 sec / 0.000 sec

Primero de todo creamos la tabla 'credit\_card' en el modelo de bbdd con los campos necesarios una vez explorados los documentos que incluyen las variables a insertar posteriormente. Para decidir los formatos y longitud de las variables me he guiado por el diagrama mostrado al final del ejercicio.

The screenshot shows the MySQL Workbench interface with the SQL Editor tab selected. The code in the editor is:

```
15 • ALTER TABLE transaction
16     MODIFY credit_card_id VARCHAR(20);
```

The output pane shows a single query execution:

Action Output	#	Time	Action	Message	Duration / Fetch
61	12:43:13		ALTER TABLE transaction MODIFY credit_card_id VARCHAR(20)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec

El segundo paso es editar la longitud del campo transaction.credit\_card\_id a un VARCHAR(20) para que case con credit\_card.id ya que esta será la base para relacionar ambas tablas mediante FK.

```

4914 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9494', 'XX581297356475951935553149', '5835276613916752', '7584'
4915 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9495', 'XX616510915920361697753446', '3339471273973308', '2819'
4916 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9496', 'XX873923182575653187160905', '3953223868157415', '6977'
4917 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9497', 'XX217529100589886960261895', '5755321521940475', '4308'
4918 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9498', 'XX337430663495642356374437', '1071134241649571', '3595'
4919 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9499', 'XX587816282775716533001270', '8230722152930257', '9188'
4920 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9500', 'XX931549975010264549889600', '2952214276569369', '4381'
4921 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9501', 'XX423006644900567764288205', '2240994054607828', '3946'
4922 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9502', 'XX52284587513238148871125', '5870775395108393', '7570'
4923 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9503', 'XX722494123298844525324973', '4306205653252162', '9362'
4924 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9504', 'XX516461131257696114064733', '6326485423862980', '9741'
4925 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9505', 'XX182029659846530616245508', '5705192230160680', '4418'
4926 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9506', 'XX785470021993637196762007', '5457052633092708', '3606'
4927 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9507', 'XX219795828782133908686936', '5105076502286149', '6377'
4928 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9508', 'XX976778957929433265222544', '2890776023998872', '3636'
4929 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9509', 'XX557381936463717380922361', '9877809386446134', '1012'
4930 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9510', 'XX598426785316665786978181', '2872419912097418', '7324'

```

Output:

Action Output	#	Time	Action	Message	Duration / Fetch
	5073	13:23:36	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (CcS-9580...)	1 row(s) affected	0.000 sec
	5074	13:23:36	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (CcS-9581...)	1 row(s) affected	0.016 sec

El tercer y penúltimo paso es insertar los valores del documento *datos\_introducir\_sprint3\_credit.sql* a la tabla credit\_card. Este paso lo haremos desde una distinta pestaña de SQL para poder correrlo todo de una.

```

20 • SELECT COUNT(credit_card.id)
FROM credit_card;
21
22
23 • SELECT COUNT(DISTINCT credit_card.id)
FROM credit_card;

```

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

COUNT(DISTINCT credit_card.id)
5000

Result 7 x Read Only

Output:

Action Output	#	Time	Action	Message	Duration / Fetch
	5076	13:41:26	SELECT COUNT(credit_card.id) FROM credit_card	1 row(s) returned	0.031 sec / 0.000 sec
	5077	13:41:47	SELECT COUNT(DISTINCT credit_card.id) FROM credit_card	1 row(s) returned	0.062 sec / 0.000 sec

Con los comandos anteriores vemos como el total de los registros de la tabla credit\_card es igual al total de los registros distintos lo que la convierte en una tabla de dimensiones común la cual está relacionada con la tabla de hechos transactions de 1 a N.

```

15 • ALTER TABLE transaction
ADD CONSTRAINT fk_transaction_credit
FOREIGN KEY (credit_card_id)
REFERENCES credit_card(id);

```

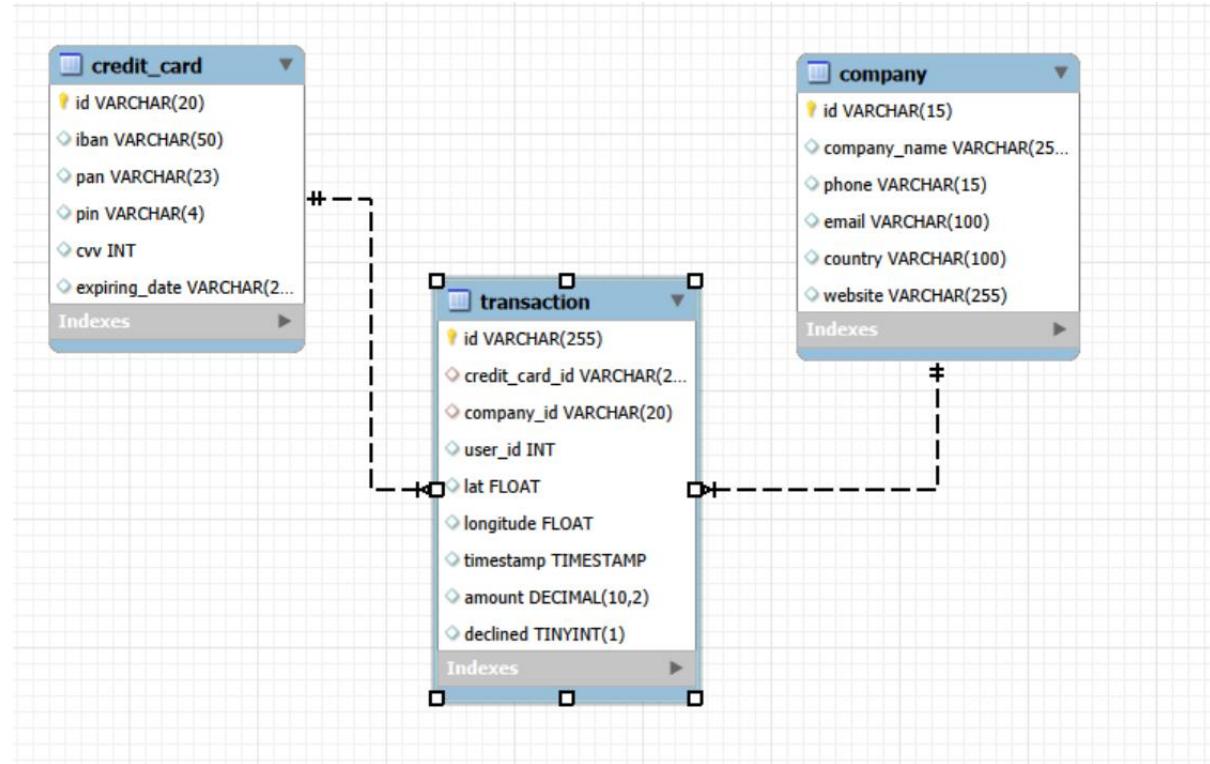
Output:

Action Output	#	Time	Action	Message	Duration / Fetch
	5074	13:23:36	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (CcS-9581...)	1 row(s) affected	0.016 sec
	5075	13:26:26	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_credit FOREIGN KEY...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	5.781 sec

Finalmente le mandamos a SQL que queremos crear una FK en transactions.credit\_card\_id y que tenga como referencia credit\_card.id. Es decir que todos los valores en el campo hijo

y FK transactions.credit\_card\_id esté previamente incluida en el campo madre credit\_card.id.

Resultante EER Diagram:



Como vemos, tabla credit\_card creada con las variables mencionadas anteriormente. PK = id. Relacionada con transaction mediante la FK credit\_card\_id. Relación de 1 a N.

## 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.

The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window containing the following SQL script:

```
28 •  SELECT credit_card.iban
29   FROM credit_card
30  WHERE credit_card.id = 'CcU-2938';
31
32 •  UPDATE credit_card
33    SET iban = 'TR323456312213576817699999'
34  WHERE credit_card.id = 'CcU-2938';
35
```

Below the code editor is a result grid titled "Result Grid" which displays a single row of data:

iban
TR323456312213576817699999

At the bottom of the interface is an "Output" window titled "credit card 10 x". It contains a table titled "Action Output" with the following data:

#	Time	Action	Message	Duration / Fetch
5080	20:51:34	UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE credit...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec
5081	20:51:37	SELECT credit_card.iban FROM credit_card WHERE credit_card.id = 'CcU-2938'	1 row(s) returned	0.000 sec / 0.000 sec

Antes de hacer el UPDATE de la fila con credit\_card.id = 'CcU-2938' he usado un SELECT para confirmar que el credit\_card.iban efectivamente era distinto al proporcionado en el enunciado. Posterior a correr el UPDATE, he vuelto a correr el SELECT para confirmar que he modificado de forma correcta la fila.

## Exercici 3

En la taula "transaction" ingressa una nova transacció 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

```
39 •   SELECT id
40     FROM credit_card
41    WHERE id = 'CcU-9999';
42
43 •   SELECT id
44     FROM company
45    WHERE id = 'b-9999';
46
```

Primero veo si existe credit\_card.id = CcU-9999 en la tabla padre.  
Lo mismo con company.id = b-9999. Como no existen tengo que crearlos.

```
49 •   INSERT INTO company (id)
50     VALUES ('b-9999');
51
52 •   INSERT INTO credit_card (id)
53     VALUES ('CcU-9999');
54
```

Como ambas tablas permiten NULL en DEFAULT me permite añadir una fila la cual solo tenga información de la PK.

```

58 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined)
59 VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, '2026-02-18 21:13:21', 111.11, 0);
60
61 • SELECT *
62 FROM transaction
63 WHERE credit_card_id = 'CcU-9999';

```

The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window with the above SQL code. Below it is a 'Result Grid' table with one row of data:

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	2026-02-18 21:13:21	111.11	0

Below the table is an 'Output' pane titled 'transaction 22'. It contains two log entries:

- # 5099 21:31:21 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', 9999, 829.999, -117.999, '2026-02-18 21:13:21', 111.11, 0); Duration / Fetch: 0.000 sec / 0.000 sec. Message: Error Code: 1064. You have an error in your SQL syntax; check the manual that came with MySQL for the right syntax to use near 'CcU-9999' at line 1.
- # 5100 21:31:37 SELECT \* FROM transaction WHERE credit\_card\_id = 'CcU-9999'; Duration / Fetch: 0.000 sec / 0.000 sec. Message: 1 row(s) returned.

Finalmente insertamos la información proporcionada por el enunciado y comprobamos con SELECT que efectivamente se ha introducido correctamente.

Nótese que el campo de TIMESTAMP fue añadido manualmente porque (1) el enunciado no nos proporcionaba la fecha de la transacción, y (2) el TIMESTAMP no estaba automatizado.

## Exercici 4

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

```

65 • ALTER TABLE credit_card
66 DROP COLUMN pan;
67
68 • SELECT *
69 FROM credit_card;

```

The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window with the above SQL code. Below it is a 'Result Grid' table with five rows of data:

id	iban	pin	cvv	expiring_date
CcS-4857	XX4857591835292505850771	1819	467	09/27/25
CcS-4858	XX8581768137002436094025	3964	817	12/28/28
CcS-4859	XX7826930491423553609370	4983	277	11/26/26
CcS-4860	XX5559590368835304645299	6876	661	07/27/27
CcS-4861	XX2035182877195191627307	5710	398	04/25/26

Below the table is an 'Output' pane titled 'credit\_card 23'. It contains three log entries:

- # 5101 21:43:25 ALTER TABLE credit\_card DROP COLUMN pan Duration / Fetch: 0.031 sec. Message: 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0.
- # 5102 21:43:32 SELECT \* FROM credit\_card Duration / Fetch: 0.000 sec / 0.000 sec. Message: 5001 row(s) returned.

## Exercici 1

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

```
74 •  SELECT *
75   FROM transaction
76 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
77
78 •  DELETE FROM transaction
79 WHERE id = "000447FE-B650-4DCF-85DE-C7ED0EE1CAAD";
```

Result Grid	Filter Rows:	Edits:	Export/Import:	Wrap Cell Content:																		
<table border="1"><thead><tr><th>id</th><th>credit_card_id</th><th>company_id</th><th>user_id</th><th>lat</th><th>longitude</th><th>timestamp</th><th>amount</th><th>declined</th></tr></thead><tbody><tr><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></tbody></table>	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	NULL												
id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined														
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL														

transaction 25 x

Action Output	#	Time	Action	Message	Duration / Fetch
5104	10:37:24		DELETE FROM transaction WHERE id = "000447FE-B650-4DCF-85DE-C7ED0EE1... 1 row(s) affected		0.016 sec
5105	10:37:27		SELECT * FROM transaction WHERE id = "000447FE-B650-4DCF-85DE-C7ED0EE... 0 row(s) returned		0.000 sec / 0.000 sec

Corremos el SELECT antes y después para asegurarnos (1) que existe la row de dicha transacción y (2) para confirmar que se eliminó correctamente.

## 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.

```

84 • CREATE VIEW VistaMarketing AS
85   SELECT
86     c.company_name,
87     c.phone,
88     c.country,
89     ROUND(AVG(t.amount), 2) AS Mitjana_de_compres
90   FROM transaction t
91   INNER JOIN company c
92     ON t.company_id = c.id
93   WHERE declined = 0
94   GROUP BY c.id, c.company_name, c.phone, c.country;
95

```

Output:

Action Output	#	Time	Action	Message	Duration / Fetch
	10199	13:01:04	SELECT * FROM VistaMarketing ORDER BY Mitjana_de_compres DESC	101 row(s) returned	0.844 sec / 0.000 sec
	10200	13:03:50	SELECT * FROM VistaMarketing ORDER BY Mitjana_de_compres DESC	101 row(s) returned	0.812 sec / 0.000 sec

```

97 • SELECT *
98   FROM VistaMarketing
99   ORDER BY Mitjana_de_compres DESC;
100

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
company_name	phone	country	Mitjana_de_compres
► Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.91
Pretium Neque Corp.	07 77 48 55 28	Australia	275.58
Urna Convallis Associates	06 01 24 77 04	United States	273.57
At Associates	09 56 61 10 65	New Zealand	272.74
Metus Vitae Associates	08 25 44 40 66	Australia	270.05

VistaMarketing 44 × Read Only

Output:

Action Output	#	Time	Action	Message	Duration / Fetch
	10200	13:03:50	SELECT * FROM VistaMarketing ORDER BY Mitjana_de_compres DESC	101 row(s) returned	0.812 sec / 0.000 sec
	10201	13:04:26	SELECT * FROM VistaMarketing ORDER BY Mitjana_de_compres DESC	101 row(s) returned	0.844 sec / 0.000 sec

## Exercici 3

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

```
92 •  SELECT *  
93      FROM VistaMarketing  
94      WHERE country = 'Germany';  
95
```

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

company_name	phone	country	Mitjana de compres
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.91
Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.32
Convallis In Incorporated	06 66 57 29 50	Germany	257.69
Ac Industries	09 34 65 40 60	Germany	255.17

VistaMarketing 30 x 

Output:

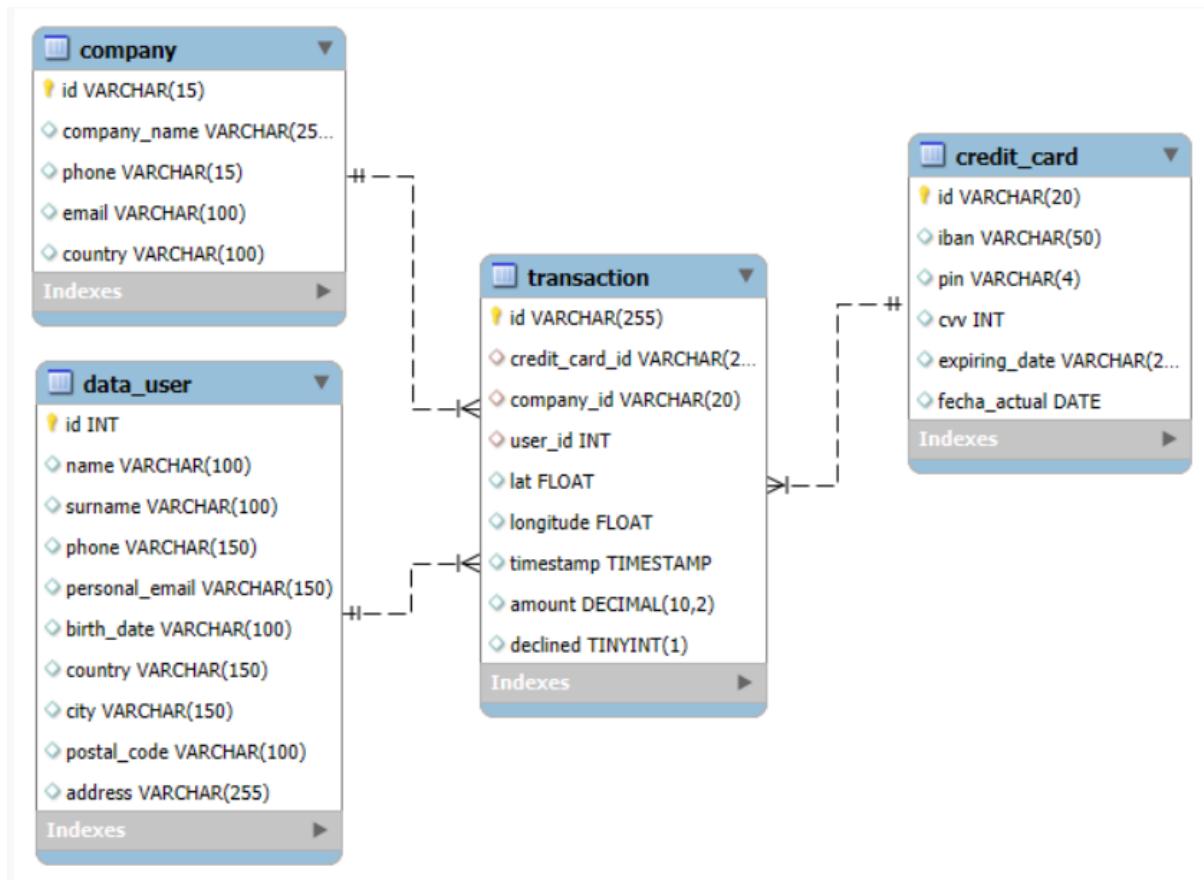
Action Output

#	Time	Action	Message	Duration / Fetch
✖ 5114	10:58:07	SELECT * FROM VistaMarketing WHERE c.country = 'Germany'	Error Code: 1054. Unknown column 'c.country' in 'where clause'	0.000 sec
✔ 5115	10:58:20	SELECT * FROM VistaMarketing WHERE country = 'Germany'	8 row(s) returned	0.141 sec / 0.000 sec

**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:



He dividido los pasos para llegar a dicho diagrama de la siguiente forma:

```
100    #Eliminamos el campo website de la tabla company
101 •  ALTER TABLE company
102     DROP COLUMN website;
103    #Eliminamos la VistaMarketing
104 •  DROP VIEW VistaMarketing;
105    #Creamos el campo fecha_actual en la tabla credit_card
106 •  ALTER TABLE credit_card
107     ADD COLUMN fecha_actual DATE;
```

Output		
#	Time	Action
5117	11:40:16	DROP VIEW VistaMarketing
5118	11:40:19	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE

```

109      #Creamos la tabla data_user
110 • CREATE TABLE IF NOT EXISTS data_user(
111     id INT PRIMARY KEY,
112     name VARCHAR(100),
113     surname VARCHAR(100),
114     phone VARCHAR(100),
115     personal_email VARCHAR(150),
116     birth_date VARCHAR(100),
117     country VARCHAR(150),
118     city VARCHAR(150),
119     postal_code VARCHAR(100),
120     address VARCHAR(255)
121 );
122
123

```

#### Output:

Action Output	#	Time	Action	Message	Duration / Fetch
	5118	11:40:19	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
	5119	11:51:26	CREATE TABLE IF NOT EXISTS data_user(id INT PRIMARY KEY, name VARCH...	0 row(s) affected	0.032 sec

```

109      #Creamos la tabla data_user
110      #(notese que el nombre de la tabla y del campo email son distintos al diagrama, posteriormente realizaremos los cambios necesarios):
111
112 • CREATE TABLE IF NOT EXISTS user(
113     id INT PRIMARY KEY,
114     name VARCHAR(100),
115     surname VARCHAR(100),
116     phone VARCHAR(100),
117     email VARCHAR(150),
118     birth_date VARCHAR(100),
119     country VARCHAR(150),
120     city VARCHAR(150),
121     postal_code VARCHAR(100),
122     address VARCHAR(255)
123 );

```

#### Output:

Action Output	#	Time	Action	Message	Duration / Fetch
	10121	11:57:01	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal...	1 row(s) affected	0.000 sec

```

125      #Modificamos el nombre del campo 'email' a 'personal_email'
126 • ALTER TABLE user
127     RENAME COLUMN email TO personal_email;
128
129      #Modificamos el nombre de la tabla 'user' a 'data_user'
130 • RENAME TABLE `user` TO data_user;
131

```

#### Output:

Action Output	#	Time	Action	Message	Duration / Fetch
	10125	12:07:02	DROP TABLE data_user	0 row(s) affected	0.031 sec
	10126	12:07:05	RENAME TABLE 'user' TO data_user	0 row(s) affected	0.032 sec

Debido a un pequeño error al introducir el tipo de campo en data\_user (antes user) he tenido que modificar el lenght del VARCHAR del campo phone:

```

132 • ALTER TABLE data_user
133     MODIFY phone VARCHAR(150);
134

```

#### Output:

Action Output	#	Time	Action	Message	Duration / Fetch
	10126	12:07:05	RENAME TABLE 'user' TO data_user	0 row(s) affected	0.032 sec
	10127	12:11:33	ALTER TABLE data_user MODIFY phone VARCHAR(150)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec

```

171 • CREATE VIEW InformeTecnico AS
172     SELECT t.id, du.name, du.surname, cc.iban, c.company_name, ROUND(SUM(t.amount), 2) AS total_amount_per_user, TT.total_income_per_company, TT.
173     FROM transaction t
174     INNER JOIN data_user du ON t.user_id = du.id
175     INNER JOIN credit_card cc ON t.credit_card_id = cc.id
176     INNER JOIN company c ON t.company_id = c.id
177     INNER JOIN (SELECT tr.company_id, ROUND(SUM(tr.amount), 2) AS total_income_per_company, ROUND(AVG(tr.amount), 2) AS average_buy_per_client
178     FROM transaction tr
179     GROUP BY tr.company_id) TT ON t.company_id = TT.company_id
180     GROUP BY t.id, du.name, du.surname, cc.iban, c.company_name
181     ORDER BY t.id DESC;

```

\*El id que faltaba por registrar era solamente 1, y pertenecía a la transacción que tuvimos que añadir manualmente en el anterior ejercicio.

```

183 • SELECT*
184     FROM InformeTecnico;
185

```

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

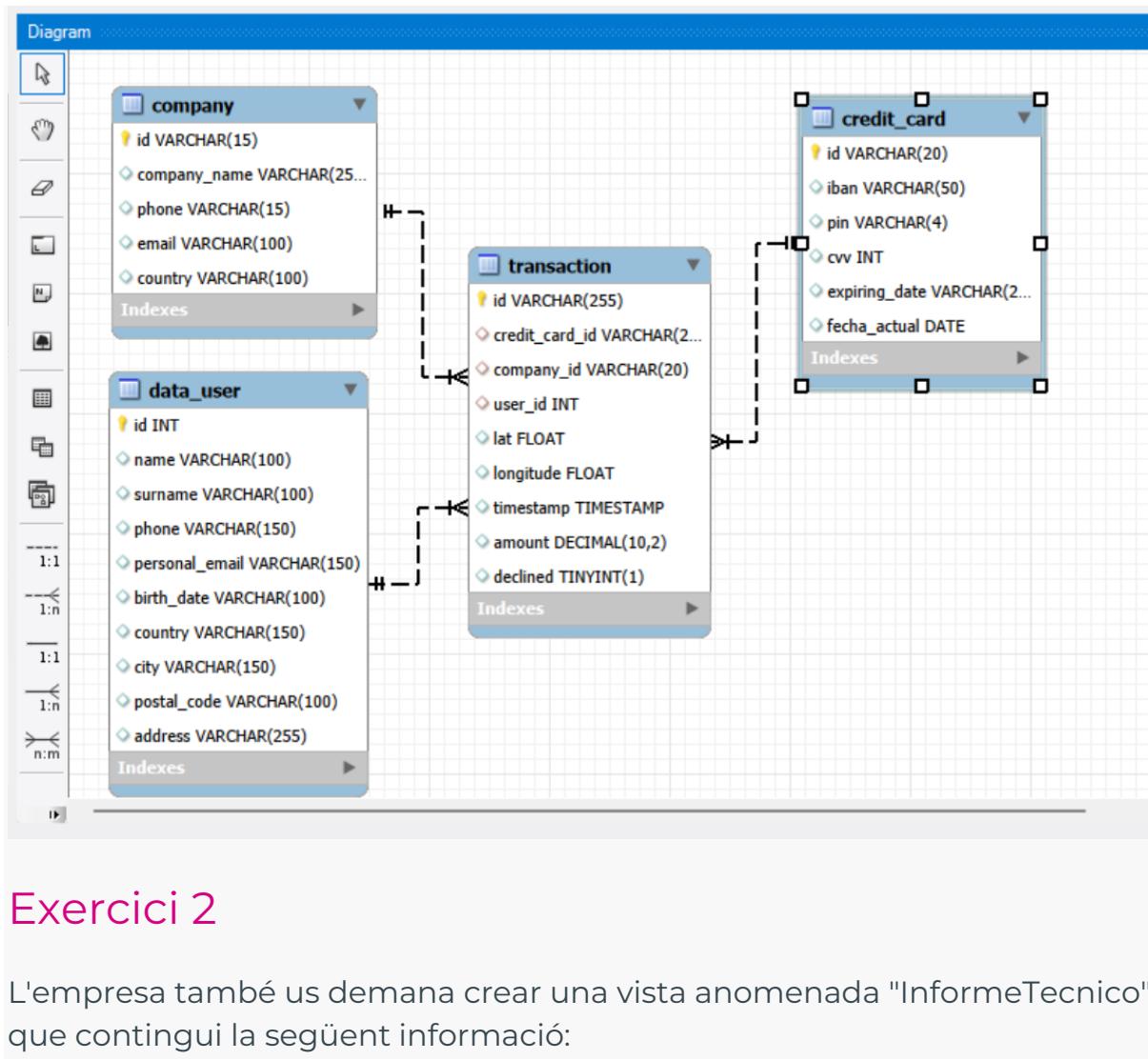
	id	name	surname	iban	company_name	total_amount_per_user	total_income_per_company	average
▶	FFFD31D6-9495-47CE-B54A-7DB8E1CC274B	Bmrgli	Tprvmmrc	XX794814451211289182490922	Turpis Company	74.54	387959.15	261.71
	FFFCF76D-ECF0-4985-A2D0-82A7B75998FC	Dfried	Vlqcdl	XX636251701647892036676034	Amet Nulla Donec Corporation	148.91	390331.32	258.33
	FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	Securp	Faofvqfy	XX162677143304223631437567	Nunc Interdum Incorporated	234.22	414651.33	259.32
	FFFB270D-F53A-4D5D-9666-E5307C53CC84	Ggzjpa	Uirzjulh	XX395114267082019952567052	Viverra Donec Foundation	349.13	396517.15	257.98

InformeTecnico 40 × Read Only

Action Output

#	Time	Action	Message	Duration / Fetch
✓	10:33:14	CREATE VIEW InformeTecnico AS SELECT t.id, du.name, du.surname, cc.iban, c.c...	0 row(s) affected	0.016 sec
✓	10:33:18	SELECT* FROM InformeTecnico	100000 row(s) returned	4.688 sec / 0.297 sec

Diagrama final coincidente con el propuesto por el enunciado:



## 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 descendente en funció de la variable ID de transacció.

```

174 • CREATE VIEW InformeTecnico AS
175     SELECT
176         t.id AS id_transaction,
177         du.name AS nombre_cliente,
178         du.surname AS apellido_cliente,
179         cc.iban,
180         c.company_name,
181         ROUND(SUM(t.amount), 2) AS total_amount_per_user,
182         TT.total_income_per_company,
183         TT.average_buy_per_client
184     FROM transaction t
185     INNER JOIN data_user du ON t.user_id = du.id
186     INNER JOIN credit_card cc ON t.credit_card_id = cc.id
187     INNER JOIN company c ON t.company_id = c.id
188     INNER JOIN (
189         SELECT tr.company_id, ROUND(SUM(tr.amount), 2) AS total_income_per_company, ROUND(AVG(tr.amount), 2) AS average_buy_per_client
190         FROM transaction tr
191         GROUP BY tr.company_id) TT ON t.company_id = TT.company_id
192     GROUP BY t.id, du.name, du.surname, cc.iban, c.company_name
193     ORDER BY t.id DESC;

```

En esta vista recogimos todos los campos mencionados en el ejercicio y además, le incluimos la suma del total de transacciones realizadas de dicho cliente (granulado por nombre y apellidos). Adicionalmente mediante una subconsulta le añadimos la suma total y la media por empresa para así poder comparar el peso que tiene cada una de las transacciones y/o clientes en las operaciones de su propia empresa con intención de, por ejemplo, consolidar cuentas.

```

195 • SELECT*
196     FROM InformeTecnico;
197

```

	id	name	surname	iban	company_name	total_amount_per_user	total_income_per_company	average
▶	FFFD31D6-9495-47CE-B54A-7D88E1CC274B	Bmrgli	Tprvvmrc	XX794814451211289182490922	Turpis Company	74.54	387595.15	261.71
	FFFCF76D-ECF0-4985-A2D0-82A7B75998FC	Dfried	Vlqcjdl	XX636251701647892036676034	Amet Nulla Donec Corporation	148.91	390331.32	258.33
	FFFC9E8D-2C7-4ADE-98F2-7533EFDF126	Securp	Faofovafy	XX162677143304223631437567	Nunc Interdum Incorporated	234.22	414651.33	259.32
	FFFB270D-F53A-4D5D-9666-E5307C53CC84	Ggzjpa	Uirzjulh	XX395114267082019952567052	Viverra Donec Foundation	349.13	396517.15	257.98

InformeTecnico 45 × Read Only

Output:

#	Time	Action	Message	Duration / Fetch
10201	13:04:26	SELECT * FROM VistaMarketing ORDER BY Mitjana_de_compres DESC	101 row(s) returned	0.844 sec / 0.000 sec
10202	13:21:40	SELECT* FROM InformeTecnico	100000 row(s) returned	3.594 sec / 0.188 sec