

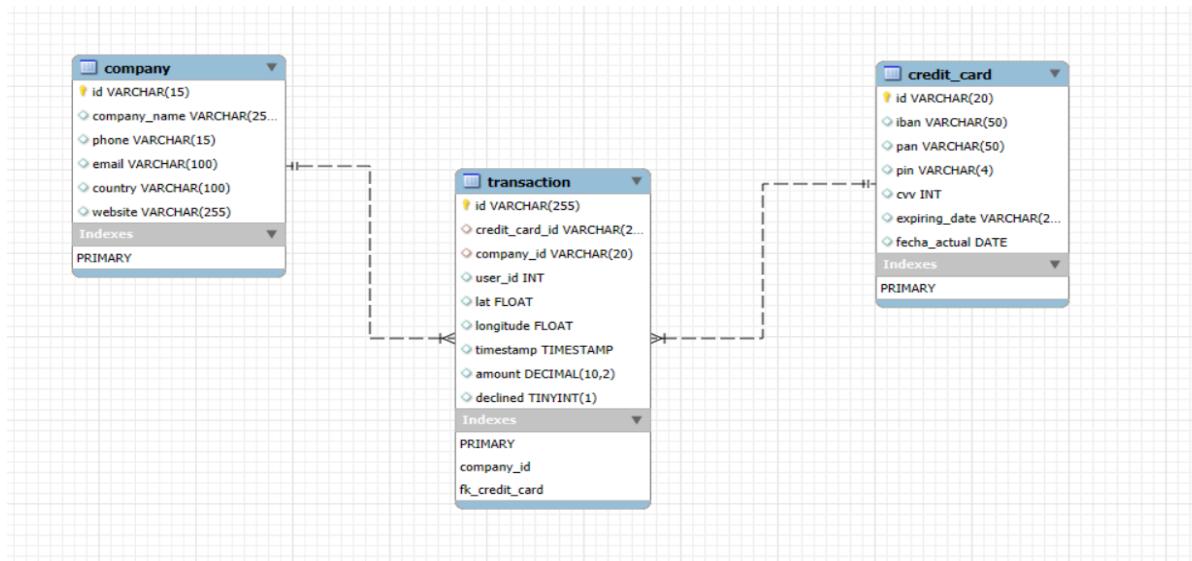
SPRINT 3



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.



ANÁLISIS MODELO Y TABLAS

Tenemos una base de datos que describe transacciones económicas de diferentes empresas, tenemos 3 tablas:

Primero de todo tenemos un modelo estrella formato da:

la tabla Transaction que es una tabla de hechos.

la tablas Company,Credit_Card que son de dimensiones.

Company tiene:

la columna ID como Primary.

Transaction tiene:

la columna ID como Primary,

la columna "company_id" como foreign key

la columna "credit_card_id" como foreign key

Credit_Card tiene:
la columna ID como Primary.

Relaciones de cardinalidad:

Las tablas están conectadas gracias a las relaciones 1 a n,
con la tabla de hechos en el medio.

Donde 1 es en Company(primary key "ID")
y N es en Transaction(foreign key "company_id")

donde 1 es en Credit_Card(primary key "ID")
y N es en Transaction(foreign key "credit_card_id")

Analysis Columnas:

Company:

id **varchar**(15)
company_name **varchar**(255)
phone **varchar**(15)
email **varchar**(100)
country **varchar**(100)
website **varchar**(255)

Transaction:

id **varchar**(255)
credit_card_id **varchar**(15)
company_name **varchar**(20)
user_id **int**
lat **float**
longitude **float**
timestamp **timestamp**
amount **decimal**(10,2)
declined **tinyint**(1)

Credit_card:

id **varchar**(20),
iban **varchar**(50),
pan **varchar**(50),
pin **varchar**(4),
cvv **int**,

```

    expiring_date varchar(20),
    fecha_actual date default (CURRENT_DATE));

```

Creación de la tabla credit_card:

```

45
46 • ALTER TABLE transaction MODIFY credit_card_id VARCHAR(20);
47
48 ##### CREO TABLA CREDIT CARD #####
49
50 • DROP TABLE IF EXISTS credit_card;
51
52 • CREATE TABLE IF NOT EXISTS credit_card (
53     id VARCHAR(20) PRIMARY KEY,
54     iban VARCHAR(50),
55     pan VARCHAR(50),
56     pin VARCHAR(4),
57     cvv INT ,
58     expiring_date VARCHAR(20),
59     fecha_actual DATE DEFAULT (CURRENT_DATE));
60
61 • ALTER TABLE transaction
62     ADD CONSTRAINT fk_credit_card
63     FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
64

```

Output

Action	Output		
#	Time	Action	Message
315388	12:41:25	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, decline...)	1 row(s) affected
315389	12:41:25	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, decline...)	1 row(s) affected
315390	12:41:25	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, decline...)	1 row(s) affected
315391	12:41:25	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, decline...)	1 row(s) affected
315392	12:41:33	ALTER TABLE transaction MODIFY credit_card_id VARCHAR(20)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
315393	12:42:06	CREATE TABLE IF NOT EXISTS credit_card (id VARCHAR(20) PRIMARY KEY, iban VARCHAR(50), pan V...	0 row(s) affected

Creación de la clave foránea:

```

61 • ALTER TABLE transaction
62     ADD CONSTRAINT fk_credit_card
63     FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
64
65
66 #Exercici 2
67 #El departament de Recursos Humans ha identificat un error en el número de compte associat a la targeta de crèdit
68 #amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és: TR323456312213576817699999. Recorda mostrar que el canvi es va
69 • SELECT * FROM credit_card
70 WHERE id = 'CcU-2938';
71
72 • UPDATE credit_card
73 SET iban = 'TR323456312213576817699999'

```

Output

Action	Output			
#	Time	Action	Message	Duration / Re
320390	12:45:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9577', 'XX158914078594808...', ...)	1 row(s) affected	0.000 sec
320391	12:45:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9578', 'XX991539646456110...', ...)	1 row(s) affected	0.000 sec
320392	12:45:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9579', 'XX296393091587170...', ...)	1 row(s) affected	0.000 sec
320393	12:45:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9580', 'XX781258889851950...', ...)	1 row(s) affected	0.000 sec
320394	12:45:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9581', 'XX915670516405388...', ...)	1 row(s) affected	0.000 sec
320395	12:45:25	ALTER TABLE transaction ADD CONSTRAINT fk_credit_card FOREIGN KEY (credit_card_id) REFERENCES...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	1.532 sec

COMENTARIO:

He cambiado el valor del formato(varchar) de la columna “credit_card_id” de la tabla transaction a 20 (antes era 15) para una mejor funcionalidad

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.

ANTES:

```
66 #Exercici 2
67 #El departament de Recursos Humans ha identificat un error en el número de compte associat a la targeta de crèdit
68 #amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és: TR323456312213576817699999. Recorda mostrar que el canvi es va
69 • SELECT * FROM credit_card
70 WHERE id = 'CcU-2938'
71
72 • UPDATE credit_card
73 SET iban = 'TR323456312213576817699999'
74 WHERE id = 'CcU-2938';
75
76 #Exercici 3
77 #En la taula "transaction" ingressa una nova transacció amb la següent informació:
```

Result Grid						
Filter Rows: <input type="text"/> Edit: Export/Import: Wrap Cell Content:						
id	iban	pan	pin	cvv	expiring_date	fecha_actual
CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22	2025-12-09
HULL	HULL	HULL	HULL	HULL	HULL	HULL

credit_card 7						
Output: Action Output						
#	Time	Action	Message			
320391	12:45:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9578', 'XX99153964645611...')	1 row(s) affected			
320392	12:45:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9579', 'XX29639309158717...')	1 row(s) affected			
320393	12:45:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9580', 'XX78125888985195...')	1 row(s) affected			
320394	12:45:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9581', 'XX91567051640538...')	1 row(s) affected			
320395	12:45:25	ALTER TABLE transaction ADD CONSTRAINT fk_credit_card FOREIGN KEY (credit_card_id) REFERENCES...	100000 row(s) affected	Records: 100000	Duplicates: 0	Warnings: 0
320396	12:47:07	SELECT * FROM credit_card WHERE id = 'CcU-2938' LIMIT 0, 200	1 row(s) returned			

DESPUÉS:

```
66 #exercici 2
67 #El departament de Recursos Humans ha identificat un error en el número de compte associat a la targeta de crèdit
68 #amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és: TR323456312213576817699999. Recorda mostrar que el canvi es va
69 • SELECT * FROM credit_card
70 WHERE id = 'CcU-2938';
71
72 • UPDATE credit_card
73 SET iban = 'TR323456312213576817699999'
74 WHERE id = 'CcU-2938';
75
76 #Exercici 3
77 #En la taula "transaction" ingressa una nova transacció amb la següent informació:
```

Result Grid						
Filter Rows: <input type="text"/> Edit: Export/Import: Wrap Cell Content:						
id	iban	pan	pin	cvv	expiring_date	fecha_actual
CcU-2938	TR323456312213576817699999	5424465566813633	3257	984	10/30/22	2025-12-09
HULL	HULL	HULL	HULL	HULL	HULL	HULL

credit_card 8						
Output: Action Output						
#	Time	Action	Message			
320393	12:45:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9580', 'XX78125888985195...')	1 row(s) affected			
320394	12:45:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9581', 'XX91567051640538...')	1 row(s) affected			
320395	12:45:25	ALTER TABLE transaction ADD CONSTRAINT fk_credit_card FOREIGN KEY (credit_card_id) REFERENCES...	100000 row(s) affected	Records: 100000	Duplicates: 0	Warnings: 0
320396	12:47:07	SELECT * FROM credit_card WHERE id = 'CcU-2938' LIMIT 0, 200	1 row(s) returned			
320397	12:47:30	UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE id = 'CcU-2938'	1 row(s) affected	Rows matched: 1	Changed: 1	Warnings: 0
320398	12:47:33	SELECT * FROM credit_card WHERE id = 'CcU-2938' LIMIT 0, 200	1 row(s) returned			

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

Primer paso creo un trigger:

```
83  DELIMITER $$  
84  
85 • CREATE TRIGGER insert_user_creditcard  
86   BEFORE INSERT ON transaction  
87   FOR EACH ROW  
88   BEGIN  
89     IF NOT EXISTS( SELECT 1 FROM company WHERE id = new.company_id )  
90     THEN INSERT INTO company(id)  
91       VALUES(new.company_id);  
92     END IF;  
93  
94     IF NOT EXISTS( SELECT 1 FROM credit_card WHERE id = new.credit_card_id )  
95     THEN INSERT INTO credit_card(id)  
96       VALUES(new.credit_card_id);  
97     END IF;  
98   END$$  
99  
100  DELIMITER ;  
101  
102 • ## Despues haber creado el trigger que actualiza automaticamente las tablas company y credit card puedo insertar la nueva transaccion ###  
103  
104   ## Antes cambio la columna timestamp, para que tenga como DEFAULT dia y hora actual en automatico(sino me queda vacio el campo, cuando hago el insert de la nueva t  
105  
106   ALTER TABLE transaction MODIFY timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP;  
107  
108   ### ahora puedo insertar tranquilamente ###  
109  
110 •   INSERT INTO transaction(id,credit_card_id,company_id,user_id,lat,longitude,amount,declined)  
111   VALUES ("108B1D1D-5B22-A76C-55EF-C568E49A99DD","CcU-3999","b-9999",9399,829.999,-117.999,111.11,0);  
112  
113 •   SELECT * FROM transaction
```

Output ::

Action	Output		
#	Time	Action	Message
320398	12:47:33	SELECT * FROM credit_card WHERE id = 'CcU-2938' LIMIT 0, 200	1 row(s) returned
320399	12:48:52	CREATE TRIGGER insert_user_creditcard BEFORE INSERT ON transaction FOR EACH ROW BEGIN IF N... 0 row(s) affected	

Segundo paso pongo default en columna timestamp:

```
106 ALTER TABLE transaction MODIFY timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP;
107
108 ### ahora puedo insertar tranquilamente ###
109
110 • INSERT INTO transaction(id,credit_card_id,company_id,user_id,lat,longitude,amount,declined)
111   VALUES ("108B1D1D-5B22-A76C-55EF-C568E49A99DD","CcU-3999","b-9939",9399,829.999,-117.999,111.11,0);
112
113 • SELECT * FROM transaction
114 WHERE id = "108B1D1D-5B22-A76C-55EF-C568E49A99DD";
115
116
117 #Exercici 4
118 #Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card. Recorda mostrar el c
119 • SELECT * FROM credit_card;
```

Output

#	Time	Action	Message
320399	12:48:52	CREATE TRIGGER insert_user_creditcard BEFORE INSERT ON transaction FOR EACH ROW BEGIN IF N...	0 row(s) affected
320400	12:50:50	ALTER TABLE transaction MODIFY timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Tercero paso introducir nueva transacción:

```
110 • INSERT INTO transaction(id,credit_card_id,company_id,user_id,lat,longitude,amount,declined)
111   VALUES ("108B1D1D-5B22-A76C-55EF-C568E49A99DD","CcU-3999","b-9939",9399,829.999,-117.999,111.11,0);
112
113 • SELECT * FROM transaction
114 WHERE id = "108B1D1D-5B22-A76C-55EF-C568E49A99DD";
115
116
117 #Exercici 4
118 #Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card. Recorda mostrar el c
119 • SELECT * FROM credit_card;
120
121 • ALTER TABLE credit_card DROP COLUMN pan;
122
```

Output

#	Time	Action	Message
320400	12:50:50	ALTER TABLE transaction MODIFY timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
320401	12:52:55	INSERT INTO transaction(id,credit_card_id,company_id,user_id,lat,longitude,amount,declined) VALUES ("1...	1 row(s) affected

COMENTARIO:

He tenido que añadir en las tablas company y credit_card los valores id referentes a la nueva transacción, para poder hacer el insert en la tabla transaction.
El problema a solucionar fue que teniendo una clave foránea no nos deja añadir directamente un valor en la tabla hija sin tenerlo en la madre.
Por eso he creado el trigger.

Exercici 4

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

ANTES:

```
117 #Exercici 4
118 #Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card. Recorda mostrar el canvi realitzat
119 • SELECT * FROM credit_card;
120
121 • ALTER TABLE credit_card DROP COLUMN pan;
```

id	iban	pan	pin	cvv	expiring_date	fecha_actual
Cc5-4857	XX4857591835292505850771	2314242385113924	1819	467	09/27/25	2025-12-09
Cc5-4858	XX8581768137002436094025	6582720299715533	3964	817	12/28/28	2025-12-09
Cc5-4859	XX7826930491423553609370	8861684536289642	4983	277	11/26/26	2025-12-09
Cc5-4860	XX5559590368835304645299	2481155515498459	6876	661	07/27/27	2025-12-09
Cc5-4861	XX2035182877195191627307	1308930301149557	5710	398	04/25/26	2025-12-09
Cc5-4862	XX4774721462463645409758	6715617009807829	4042	174	11/27/26	2025-12-09
Cc5-4863	XX1476829664245046207111	3140879819451394	5969	449	12/27/29	2025-12-09

DESPUÉS:

```
117 #Exercici 4
118 #Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card. Recorda mostrar el canvi real
119 • SELECT * FROM credit_card;
120
121 • ALTER TABLE credit_card DROP COLUMN pan;
```

id	iban	pin	cvv	expiring_date	fecha_actual
Cc5-4857	XX4857591835292505850771	1819	467	09/27/25	2025-12-09
Cc5-4858	XX8581768137002436094025	3964	817	12/28/28	2025-12-09
Cc5-4859	XX7826930491423553609370	4983	277	11/26/26	2025-12-09
Cc5-4860	XX5559590368835304645299	6876	661	07/27/27	2025-12-09
Cc5-4861	XX2035182877195191627307	5710	398	04/25/26	2025-12-09
Cc5-4862	XX4774721462463645409758	4042	174	11/27/26	2025-12-09
Cc5-4863	XX1476829664245046207111	5969	449	12/27/29	2025-12-09



Nivell 2

Exercici 1

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

```
1  #Exercici 1
2  #Elimina de la taula transaction el registre amb ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de dades.
3
4 • DELETE FROM transaction
5 WHERE id = "000447FE-B650-4DCF-85DE-C7ED0EE1CAAD";
6
7 #Exercici 2
8 #La secció de màrqueting desitja tenir accés a informació específica per a realitzar anàlisi i estratègies efectives.
9 #S'ha sol·licitat crear una vista que proporcioni detalls clau sobre les companyies i les seves transaccions.
10 #Serà necessària que creïs una vista anomenada VistaMarketing que contingui la següent informació: Nom de la companyia.
11 #Telèfon de contacte. País de residència. Mitjana de compra realitzat per cada companyia. Presenta la vista creada, ordenant les dades
12
13 • CREATE VIEW VistaMarketing AS (
14     SELECT c.company_name, c.phone, c.country,round(avg(t.amount),2) as avg_sales
15     FROM company AS c
16     JOIN transaction AS t
17     ON c.id = t.company_id
18     GROUP BY c.company_name, c.phone, c.country
19     ORDER BY avg(t.amount) DESC);
20
21 • select * from VistaMarketing;
22 #Exercici 3
23 #Filtre la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"
24
```

Output:

#	Time	Action	Message
320399	12:48:52	CREATE TRIGGER insert_user_creditcard BEFORE INSERT ON transaction FOR EACH ROW BEGIN IF N...	0 row(s) affected
320400	12:50:50	ALTER TABLE transaction MODIFY timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
320401	12:52:55	INSERT INTO transaction(id,credit_card_id,company_id,user_id,lat,longitude,amount,declined) VALUES ("1...	1 row(s) affected
320402	12:54:59	SELECT * FROM credit_card LIMIT 0, 200	200 row(s) returned
320403	12:55:17	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
320404	12:55:22	SELECT * FROM credit_card LIMIT 0, 200	200 row(s) returned
320405	12:56:32	DELETE FROM transaction WHERE id = "000447FE-B650-4DCF-85DE-C7ED0EE1CAAD"	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.

```
12
13 • CREATE VIEW VistaMarketing AS (
14     SELECT c.company_name, c.phone, c.country, round(avg(t.amount),2) as avg_sales
15     FROM company AS c
16     JOIN transaction AS t
17     ON c.id = t.company_id
18     GROUP BY c.company_name, c.phone, c.country
19     ORDER BY avg(t.amount) DESC);
20
21 • SELECT * FROM VistaMarketing;
22 #Exercici 3
23 #Filtre la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"
24
25 • SELECT company_name
26 FROM VistaMarketing
27 ORDER BY avg_sales DESC;
```

The screenshot shows the SQL editor with the following code and results:

```
Result Grid | Filter Rows: Export: Wrap Cell Content: □
company_name phone country avg_sales
▶ Ac Fermentum Incorporated 06 85 56 52 33 Germany 284.87
Pretium Neque Corp. 07 77 48 55 28 Australia 276.16
Urna Convallis Associates 06 01 24 77 04 United States 274.24
At Associates 09 56 61 10 65 New Zealand 272.21
Metus Vitae Associates 08 25 44 40 66 Australia 270.08
Aliquet Diam Limited 02 76 61 47 46 United States 269.60
```

VistaMarketing 3 ×

Output

#	Time	Action	Message
320405	12:56:32	DELETE FROM transaction WHERE id = "000447FE-B650-4DCF-85DE-C7ED0EE1CAAD"	1 row(s) affected
320406	12:58:01	CREATE VIEW VistaMarketing AS (SELECT c.company_name, c.phone, c.country,round(avg(t.amount),2) ...)	0 row(s) affected
320407	12:58:01	SELECT * FROM VistaMarketing LIMIT 0, 200	101 row(s) returned

Exercici 3

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

```
22 #Exercici 3
23 #Filtre la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"
24
25 • SELECT company_name
26 FROM VistaMarketing
27 WHERE country = "Germany";
```

The screenshot shows the SQL editor with the following code and results:

```
Result Grid | Filter Rows: Export: Wrap Cell Content: □
company_name
▶ Ac Fermentum Incorporated
Nunc Interdum Incorporated
Convallis In Incorporated
Ac Industries
Rutrum Non Inc.
Auctor Mauris Corp.
```

VistaMarketing 4 ×

Output

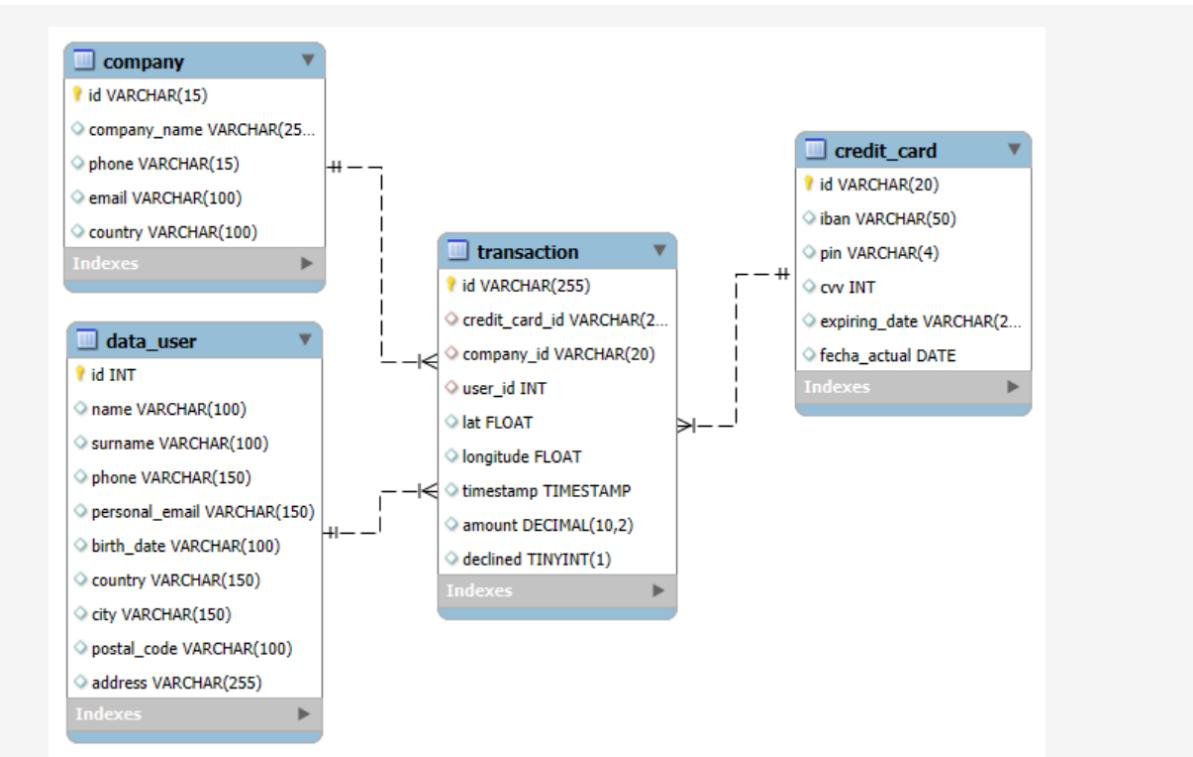
#	Time	Action	Message
320406	12:58:01	CREATE VIEW VistaMarketing AS (SELECT c.company_name, c.phone, c.country,round(avg(t.amount),2) ...)	0 row(s) affected
320407	12:58:01	SELECT * FROM VistaMarketing LIMIT 0, 200	101 row(s) returned
320408	12:59:13	SELECT company_name FROM VistaMarketing WHERE country = "Germany" LIMIT 0, 200	8 row(s) returned



Nivell 3

Exercici 1

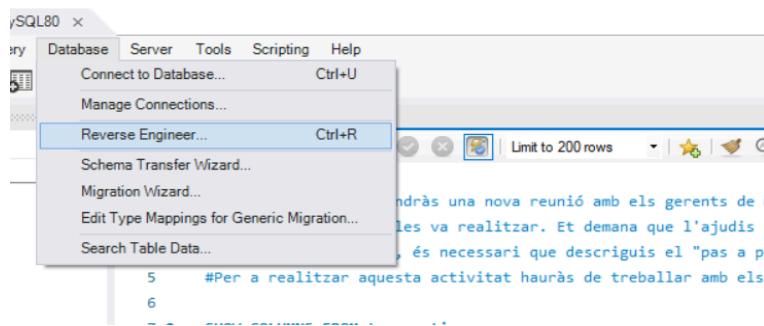
La setmana vinent tindràs una nova reunió amb els gerents de màrqueting. Un company del teu equip va realitzar modificacions en la base de dades, però no recorda com les va realitzar. Et demana que l'ajudis a deixar els comandos executats per a obtenir el següent diagrama:



Puntos clave:

1. He importado la base de datos descargandola desde el enunciado de lo sprint 2, con adentro la bbdd “transactions”, las dos tablas “company y transaction” y los relativos valores.
2. He creado manualmente la tabla “credit_card” teniendo como referencias los valores a insertar presentes en el fichero ‘datos_introducir_sprint3_credit.sql’ y el diagrama que tenemos arriba, presente en el sprint3. He cambiado la columna timestamp, para que tenga como DEFAULT dia y hora actual en automático.
3. Para poder inserir los valores y crear la clave foránea en la tabla “Hija”, tenemos que tener los valores en la tabla “Madre”, así que he creado un Trigger para poder añadir estos valores en automático en las tablas company y credit_card, así que podemos añadir nuevas transacciones a la tabla transaction.

4. Después he añadido la clave foránea al campo “credit_card_id” tabla “transaction”, con referencia a la tabla “credit_card” campo “id”.
5. He importado la tabla user y sus valores trámite los ficheros: ‘ datos introducir sprint3 user.sql’ y ‘ estructura datos user.sql’
6. He tenido que modificar el formato de la columna “id” en la tabla “user” desde “char” a INT, igual que la columna que en la tabla de “transaction” para poder crear la clave foránea entre estas dos tablas.
7. **Comentario extra:** he tenido que modificar el TRIGGER para poder añadir en automático los valores en todas las tablas credit_card, user y company una vez que hemos creado la tabla user.
8. **Diagrama :** para crear el diagrama hice Reverse Engineer desde Database,pongo una captura de donde se encuentra :



Primer paso para crear la clave foránea en User:

```

9    *** para poder hacer la fk tengo cambiar el formato en la tabla user y anadir en la que hace manualmente o modificar el trigger anterior
10
11 • ALTER TABLE user MODIFY id int;
12
13    ### manualmente ###
14
15    # INSERT INTO user(id)
16    # VALUES (9999);
17
18    ##### trigger modificado ####
19
20 • DROP TRIGGER IF EXISTS insert_user_creditcard;
21
22 DELIMITER $$;
23
24 • CREATE TRIGGER insert_user_creditcard
25     BEFORE INSERT ON transaction
26     FOR EACH ROW
27     RFGTN

```

Output:

#	Time	Action	Message
325409	13:02:14	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUE...	1 row(s) affected
325410	13:02:14	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUE...	1 row(s) affected
325411	13:05:54	ALTER TABLE user MODIFY id int	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0

Segundo paso cambio el trigger anterior:

```
22 DELIMITER $$  
23  
24 • CREATE TRIGGER insert_user_creditcard  
25   BEFORE INSERT ON transaction  
26   FOR EACH ROW  
27   BEGIN  
28     IF NOT EXISTS( SELECT 1 FROM company WHERE id = new.company_id )  
29     THEN INSERT INTO company(id)  
30       VALUES(new.company_id);  
31     END IF;  
32  
33     IF NOT EXISTS( SELECT 1 FROM user WHERE id = new.user_id )  
34     THEN INSERT INTO user(id)  
35       VALUES(new.user_id);  
36     END IF;  
37  
38     IF NOT EXISTS( SELECT 1 FROM credit_card WHERE id = new.credit_card_id )  
39     THEN INSERT INTO credit_card(id)  
40       VALUES(new.credit_card_id);  
41     END IF;  
42   END$$  
43  
44 DELIMITER ;  
45  
46 • ##### Tener que reiniciar todo el registro del nivel 1 ####
```

Tercero paso inserisco otra vez todo el registro:

```
46 • ##### tengo que reinserir todo el registro del nivel 1 #####
47 DELETE FROM transaction
48 WHERE id = ("108B1D10-5B22-A76C-55EF-C568E49A99DD");
49
50 • INSERT INTO transaction(id,credit_card_id,company_id,user_id,lat,longitude,amount,declined)
51 VALUES ("108B1D10-5B22-A76C-55EF-C568E49A99DD","CcU-3999","b-9939",9399,829.999,-117.999,111.11,0);
52
53 ##### ahora puedo anadir la clave foranea #####
54
55 • ALTER TABLE transaction
56 ADD CONSTRAINT fk_user
57 FOREIGN KEY (user_id) REFERENCES user(id);
58
59 #Exercici 2
60 #L'empresa també us demana crear una vista anomenada "InformeTecnico" que contingui la següent informació:
61
62 #ID de la transacció
63 #Nom de l'usuari/ària
64 #Nom en nom de l'usuari/ària
```

Cuarto paso creo la clave foránea:

```
55 • ALTER TABLE transaction
56   ADD CONSTRAINT fk_user
57     FOREIGN KEY (user_id) REFERENCES user(id);
58
59 #Exercici 2
60 #L'empresa també us demana crear una vista anomenada "InformeTecnico" que contingui la següent informació:
61
62 #ID de la transacció
63 #Nom de l'usuari/ària
64 #Cognom de l'usuari/ària
65 #IBAN de la targeta de crèdit usada.
66 #Nom de la companyia de la transacció realitzada.
67 #Assegureu-vos d'incloure informació rellevant de les taules que coneixereu i utilitzeu àlies per canviar de nom columnes segons calgui.
68 #Mostra els resultats de la vista, ordena els resultats de forma descendente en funció de la variable ID de transacció.
69 • DROP VIEW IF EXISTS InformeTecnico;
70
71 • CREATE VIEW InformeTecnico AS(
72   SELECT u.id AS user_id,
73     CONCAT(u.name, ' ', u.surname) AS full_name,
74     c.iban,
75     co.name AS company_name
76   FROM user u
77   JOIN credit_card c ON u.id = c.user_id
78   JOIN company co ON c.company_id = co.id
79   ORDER BY u.id DESC;
```

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.

```
71 • CREATE VIEW InformeTecnico AS(
72     SELECT u.id AS user_id,
73             CONCAT(u.name, ' ', u.surname) AS full_name_user,
74             cc.id AS credit_card_id,
75             cc.iban,
76             c.id AS company_id,
77             c.company_name,
78             t.id AS id_transaction,
79             t.amount,
80             date(t.timestamp) AS date_trasaction
81     FROM transaction AS t
82     JOIN user AS u
83     ON t.user_id = u.id
84     JOIN credit_card AS cc
85     ON t.credit_card_id = cc.id
86     JOIN company AS c
87     ON t.company_id = c.id
88     ORDER BY id_transaction DESC;
89
90 • SELECT * FROM InformeTecnico;
```

Result Grid | Filter Rows: | Exports: | Wrap Cell Content: | Fetch rows: | Result Grid | Form Editor | Read Only | Con

user_id	full_name_user	credit_card_id	iban	company_id	company_name	id_transaction	amount	date
1103	Bmrgli Tprvrmrc	Ccs-5684	XX794814451211289182490922	b-2450	Turpis Company	FFFD31D6-9495-47CE-B54A-7DB8E1CC274B	74,54	2021
680	Dfried Vilqjdl	Ccs-5261	XX636251701647892036676034	b-2250	Amet Nulla Donec Corporation	FFFCF76D-ECF0-4985-A2D0-82A7B75998FC	148,91	2023
1008	Securp Faofvqfy	Ccs-5589	XX162677114304223631437567	b-2302	Nunc Interdum Incorporated	FFFC9E8D-27C7-4ADE-98F2-7533EF0F126	234,22	2019
3074	Ggjjoa Urzjulh	Ccs-7655	XX395114267082019952567052	b-2522	Viverra Donec Foundation	FFFB27D0-F53A-405D-9666-E5307C53CC84	349,13	2024
429	Yshmq Zpsjsleed	Ccs-5010	XX8845462156537570367941	b-2234	Convallis In Incorporated	FFF9E3CE-234E-408C-A8EF-F9CAD577224A	247,39	2022
4990	Ieuenx Xwrwzwm	Crs-8971	XX37140551571165438471481	h-7266	Misk Aenean Enat Foundation	FFF9F178-4CD7-4DF9-99R1-49AF068809R1	498,13	2023

Output

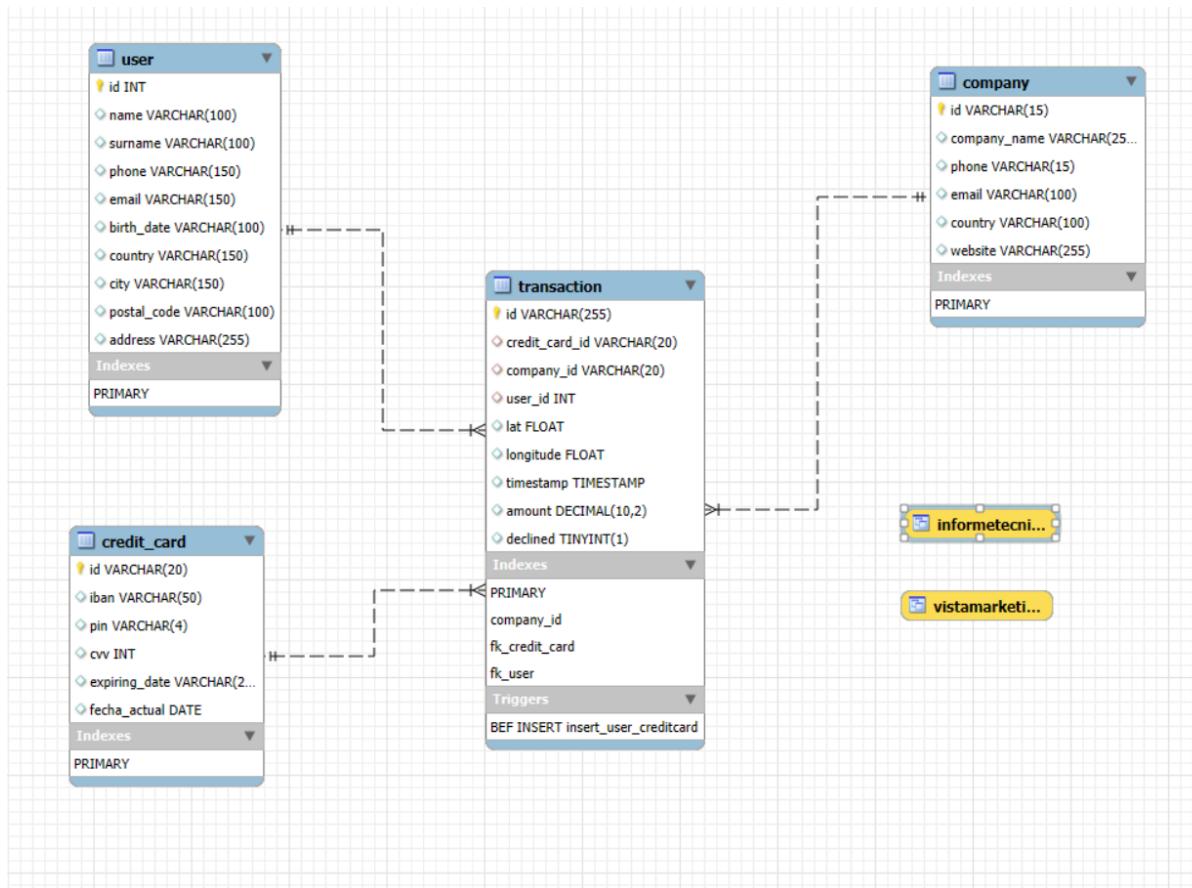
Action Output

#	Time	Action	Message
325417	13:13:24	DROP VIEW IF EXISTS InformeTecnico	0 row(s) affected, 1 warning(s): 1051 Unknown table transactions.informetecnico'
325418	13:13:24	CREATE VIEW InformeTecnico AS (SELECT u.id AS user_id, CONCAT(u.name, ' ', u.surname) AS full_name_user, cc.id AS credit_card_id, cc.iban, c.id AS company_id, c.company_name, t.id AS id_transaction, t.amount, date(t.timestamp) AS date_trasaction FROM transaction AS t JOIN user AS u ON t.user_id = u.id JOIN credit_card AS cc ON t.credit_card_id = cc.id JOIN company AS c ON t.company_id = c.id ORDER BY id_transaction DESC);	0 row(s) affected
325419	13:13:24	SELECT * FROM InformeTecnico LIMIT 0, 200	200 row(s) returned

COMMENTARIO:

He creado la vista como pide el enunciado más algunos campos que pienso ser interesantes para el informe.

DIAGRAMA FINAL PERSONAL



COMMENTARIO:

Tenemos una base de datos que describe transacciones económicas de diferentes empresas,

tenemos 4 tablas:

Primero de todo tenemos un modelo **estrella** formato da:

la tabla Transaction que es una tabla de hechos.

las tablas Company,Credit_Card,Data_User que son de dimensiones.

Company tiene:

la columna ID como Primary.

Transaction tiene:

la columna ID como Primary,

la columna "company_id" como foreign key

la columna "credit_card_id" como foreign key

la columna "user_id" como foreign Key

Credit_Card tiene:
la columna ID como Primary.

Data_User tiene:
la columna ID como Primary.

Cardinalidad:
Las tablas están conectadas gracias a las relaciones 1 a n,
con la tabla de hechos en el medio(transaction).
donde 1 es en Company(primary key "ID")
y N es en Transaction(foreign key "company_id"
donde 1 es en Credit_Card(primary key "ID")
y N es en Transaction(foreign key "credit_card_id"
donde 1 es en Data_User(primary key "ID")
y N es en Transaction(foreign key "user_id"

EXTRA:
Podemos ver las dos vistas creadas : “InformeTecnico” y “VistaMarketing”

Analisis Columnas:

Company:

```
id varchar(15)
company_name varchar(255)
phone varchar(15)
email varchar(100)
country varchar(100)
website varchar(255)
```

Transaction:

```
id varchar(255)
credit_card_id varchar(20)
company_name varchar(20)
user_id int
lat float
longitude float
timestamp timestamp default (CURRENT_TIMESTAMP)
```

```
amount decimal(10,2)
declined tinyint(1)
```

Credit_card:

```
id varchar(20),
iban varchar(50),
pin varchar(4),
cvv int ,
expiring_date varchar(20),
fecha_actual date default (CURRENT_DATE);
```

User:

```
id int,
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)
```