

## - Exercici 1

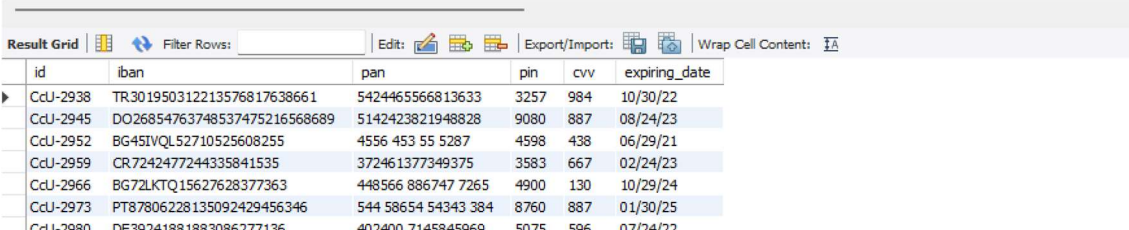
La teva tasca és dissenyar i crear una taula anomenada "credit\_card" que emmagatzemi detalls crucials sobre les targetes de crèdit. La nova taula ha de ser capaç d'identificar de manera única cada targeta i establir una relació adequada amb les altres dues taules ("transaction" i "company"). Després 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.

En este caso, comprobamos, los datos que nos van a proporcionar, una vez vistos usamos la función **Create Table**, creamos cada columna con el tipo de dato correspondiente para cada columna.

Por ejemplo, **varchar** si es una cade de **string**, **int** si es un entero, etc. Además, también establecemos la relación con la tabla "Transaction" a través de la primary key de credit card que es la id y la foreign key de en transaction en credit\_card\_id.

Añadimos not null en las columnas necesarias para establecer que no pueden haber valores errores en dichas columnas.

```
1 • CREATE TABLE credit_card (  
2     id VARCHAR(15) PRIMARY KEY,  
3     iban VARCHAR(100) NOT NULL,  
4     pan VARCHAR(255) NOT NULL,  
5     pin VARCHAR(50) NOT NULL,  
6     cvv VARCHAR(50) NOT NULL,  
7     expiring_date VARCHAR(255) NOT NULL  
8 );  
9  
10 #Aqui introducimos el Script para los datos de la tabla  
11  
12 #luego introducimos los valores que nos han proporcionado en el otro script y comprobamos:  
13 • SELECT *  
14 FROM credit_card;
```



	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

Podemos ver con la query que todo ha salido como esperábamos.

Además de la query de como relacionar las tablas:

```
#Añadimos la Foreign KEY a la tabla transaction  
ALTER TABLE transaction ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
```

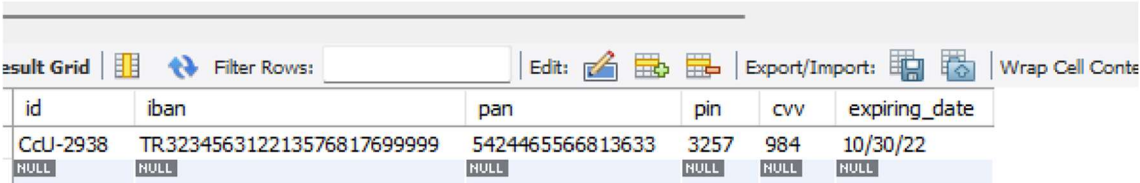
## - Exercici 2

El departament de Recursos Humans ha identificat un error en el número de compte de l'usuari amb el: IBAN CcU-2938. Es requereix actualitzar la informació que identifica un compte bancari a nivell internacional (identificat com "IBAN"): TR323456312213576817699999. Recorda mostrar que el canvi es va realitzar.

En este caso, usamos la función **update** para indicar que queremos alterar un valor de una tabla. Usamos la función set seguido de la columna que queremos alterar, pero sin olvidar un **filtro where** de donde queremos modificar el valor exactamente.

Abajo vemos como comprobamos si la información se ha actualizado.

```
20
21      #ex02
22      • UPDATE credit_card
23      SET iban = "TR323456312213576817699999"
24      WHERE id="CcU-2938";
25
26      #COMPROBACION DEL CAMBIO:
27      • SELECT *
28      FROM credit_card
29      WHERE id= "CcU-2938";
```



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

## - Exercici 3

En la taula "transaction" ingresa 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

```
39 • INSERT INTO company (id)
40   VALUES ('b-9999');
41 • INSERT INTO credit_card (id)
42   VALUES ('CcU-9999');
43   #insertamos los valores requeridos.
44 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount,declined)
45   VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD','CcU-9999','b-9999','9999','829.999','-117.999',null,'111.11','0');
46
47   #Comprobacion del cambio:
48 • SELECT *
49   FROM transaction
50   WHERE company_id = 'b-9999';
```

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	NULL	111.11	0

En este caso tenemos dos opciones para proceder con la introducción de los datos:

1. Deshabilitar la FK ente company y transaction para poder introducir el valor sin necesidad de modificar ambas tablas.
2. Añadir y dejar prepara la tabla company y credit\_card con la id de la nueva compañía y tarjeta respectivamente.

Yo he decido optar por la segunda opción. Una vez hecho esto usamos las función **Insert into** seguido del nombre de la tabla para indicar los valores que vamos a introducir de dicha tabla. Después con **Values()** especificamos los valores para cada columna nombrada.

## - Exercici 4

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

```
53      #Ex04
54      • ALTER TABLE credit_card
55      DROP COLUMN pan;
56      #Comprobamos:
57      • SELECT *
58      FROM credit_card;
```

Result Grid

Filter Rows:

Edit:

Export/Import:

id	iban	pin	cvv	expiring_date
CcU-2938	TR323456312213576817699999	3257	984	10/30/22
CcU-2945	DO26854763748537475216568689	9080	887	08/24/23
CcU-2952	BG45IVQL52710525608255	4598	438	06/29/21
CcU-2959	CR7242477244335841535	3583	667	02/24/23
CcU-2966	BG72LKTQ15627628377363	4900	130	10/29/24
CcU-2973	PT87806228135092429456346	8760	887	01/30/25
CcU-2980	DE39241881883086277136	5075	596	07/24/22

Para este ejercicio usamos la función **Alter Table** que nos deja modificar la estructura de una tabla, y con la función **Drop** especificamos que queremos eliminar un elemento en este caso **Column** "pan".

## Nivell 2

### Exercici 1

Elimina el registre amb IBAN 02C6201E-D90A-1859-B4EE-\*88D2986D3B02 de la base de dades.

```
60 #Nivel 2 Ex 01
61 • DELETE
62 FROM transaction
63 WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
64
65 #Comprobamos:
66 • SELECT *
67 FROM transaction
68 WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```



The screenshot shows a database interface with a toolbar and a result grid. The toolbar includes buttons for 'result Grid', 'Filter Rows', 'Edit', and 'Export/Import'. The result grid displays the following columns and their values:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

En este caso eliminamos un registro usando la función **Delete** con un filtro **Where** para indicar que deseamos eliminar. **Importante el where o borramos la tabla entera.**

## 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

```
71 • CREATE VIEW VistaMarketing AS
72 SELECT
73     company.company_name AS Compañia,
74     company.phone AS telefono,
75     company.country AS pais,
76     AVG(transaction.amount) AS promedio
77 FROM company
78 JOIN transaction ON company.id = transaction.company_id
79 GROUP BY Compañia,telefono,pais
80 ORDER BY promedio DESC;
81
82 #Comprobamos:
83 • SELECT *
84 FROM VistaMarketing;
85
```

Compañia	telefono	pais	promedio
Eget Ipsum Ltd	03 67 44 56 72	United States	473.075000
Non Magna LLC	06 71 73 13 17	United Kingdom	468.345000
Sed Id Limited	07 28 18 18 13	United States	461.210000
Justo Eu Arcu Ltd	08 42 56 71 52	Italy	443.635000
Eget Tincidunt Dui Institute	05 35 93 32 44	Netherlands	442.520000
Viverra Donec Foundation	03 33 12 32 73	United Kingdom	442.280000
Vestibulum Lorem PC	02 02 87 33 40	Belgium	434.060000

Para crear una vista, usamos la función Create seguida de View para indicar que es una vista, Seleccionamos los campos requeridos y comprobamos la vista.

**\*\*Este fue uno de los ejercicios corregidos y esta vez lo he hecho sin subconsulta para que sea mas eficaz como se me dijo\*\***

## Exercici 3

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

Aquí simplement seleccionamos todo de la vista y aplicamos un filtro de país en clausula Where.

```
86      #Nivel 2 Ex03
87      SELECT *
88      FROM VistaMarketing
89      WHERE país = 'Germany';
90
91      #Ex08
```

Result Grid				
Filter Rows:		Export:		
Wrap Cell Content:				
Compañia	telefono	pais	promedio	
Aliquam PC	01 45 73 52 16	Germany	385.265000	
Ac Industries	09 34 65 40 60	Germany	289.645000	
Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000	
Nunc Interdum Incorporated	05 18 15 48 13	Germany	244.025238	
Augue Foundation	06 88 43 15 63	Germany	240.800000	
Ac Fermentum Incorporated	06 85 56 52 33	Germany	206.465000	
Auctor Mauris Corp.	05 62 87 14 41	Germanv	184.310000	

# 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 les següents modificacions (s'espera que realitzin 6 canvis):

Primero creamos el indice y la tabla con sus correspondientes datos

```
91 #Nivel 3 Ex01
92 #Primero creamos el indice y la tabla data_user que alamecena los datos de usuario
93 • CREATE INDEX idx_user_id ON transaction(user_id);
94 • CREATE TABLE IF NOT EXISTS user (
95     id INT PRIMARY KEY,
96     name VARCHAR(100),
97     surname VARCHAR(100),
98     phone VARCHAR(150),
99     email VARCHAR(150),
100     birth_date VARCHAR(100),
101     country VARCHAR(150),
102     city VARCHAR(150),
103     postal_code VARCHAR(100),
104     address VARCHAR(255)
105 );
106
```

Después modificamos las tablas, primero añadimos la columna fecha\_actual luego borramos la columna website de company.

```
#Modificamos la tabla credit_card para añadir la columna fecha_actual
• ALTER TABLE credit_card
  ADD COLUMN fecha_actual DATE;

#Eliminamos la columna website de la tabla company
• ALTER TABLE company
  DROP COLUMN website;
```

Procedemos a modificar las columnas correspondientes de la tabla credit\_card.

```
#Modificamos algunas columnas de credit_card
ALTER TABLE credit_card
MODIFY COLUMN id VARCHAR(20),
MODIFY COLUMN iban VARCHAR(50),
MODIFY COLUMN pin VARCHAR(4),
MODIFY COLUMN expiring_date VARCHAR(10),
MODIFY COLUMN cvv int; #Apunto que en la anterior correccion se me aviso que este tipo de dato podria
#dar error si hay 0 en la izquierda pero el ejercicio lo requiere
```



#Apunto que en la anterior corrección se me aviso que este tipo de dato en csv podría dar error si hay 0 en la izquierda pero el ejercicio lo requiere

Luego introducimos el script con lo datos cargados.

```
93 # introducimos los datos de user (proximo ejercicio linea 377)
94 • SET foreign_key_checks = 0;
95
96 -- Insertamos datos de user
97 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "1", "Zeus",
98 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "2", "Garret
99 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "3", "Ciaran
100 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "4", "Howard
101 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "5", "Hayfa"
102 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "6", "Joel",
103 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "7", "Rafael
104 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "8", "Nissim
105 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "9", "Mannix
106 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "10", "Nelsa
```

Por ultimo modificamos la tabla User para ponerle su nuevo nombre data\_user

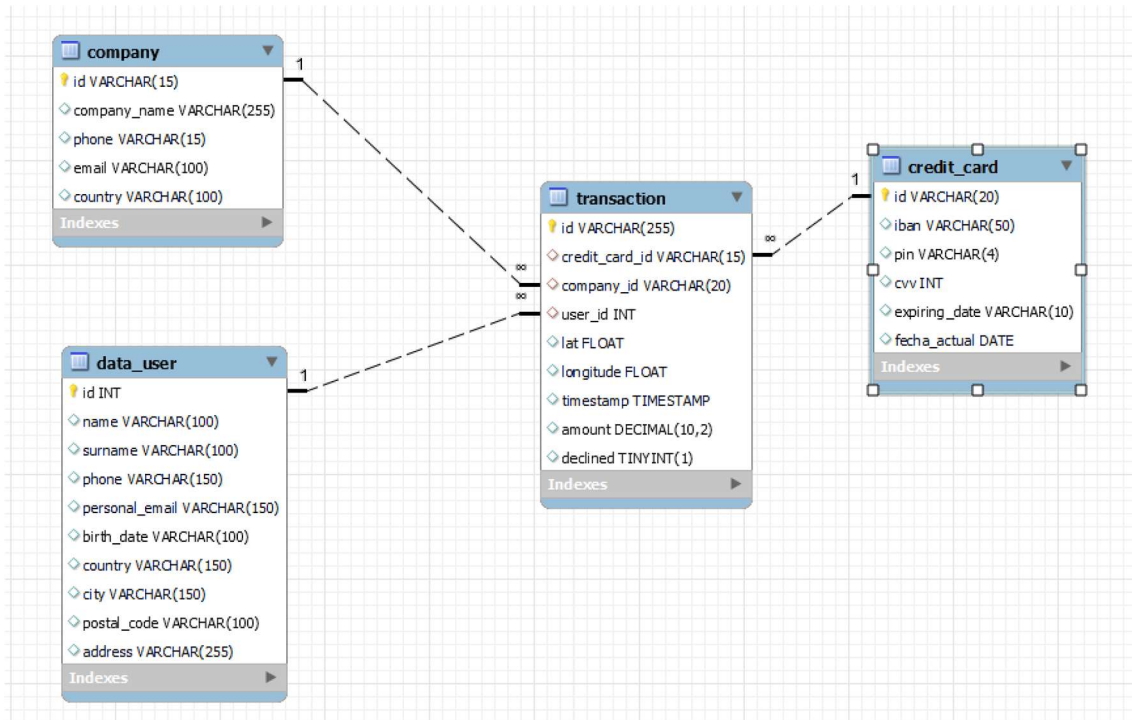
```
#renombramos la tabla user, al nombre correspondiente, data_user
rename table user to data_user;
```

También cambiamos la columna mail a personal\_mail

```
#Modificamos la columna email a personal_email
alter table data_user
change email personal_email VARCHAR(150);
```

Ahora establecemos la relación entre las tablas, debemos recordar que en un ejercicio anterior introducimos el user 9999 en la tabla transaction por ello debemos introducir o comprobar que este en la tabla data\_user antes de establecer la relación

```
#Establecemos la relacion entre las tablas transactions y data_user
#Para ello previamente debemos introucir el usuario que hemos creado antes en la tabla transaccion
INSERT INTO data_user (id)
VALUES('9999');
ALTER TABLE transaction ADD FOREIGN KEY (user_id) REFERENCES data_user(id);
```



## 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.

En este caso, volvemos a usar Create View para crear la vista, luego en este caso he suado joins para unir las tablas y seleccionar los datos requeridos. aparte he añadido otros datos como el país y el monto de las transacciones, usuarios y compañía

```

CREATE VIEW InformeTecnico AS
SELECT transaction.id AS id_transaccion,
transaction.amount AS monto_transaccion,
data_user.name AS nombre_usuario,
data_user.surname AS apellido,
data_user.country AS Pais_usuario,
credit_card.iban,
company.company_name AS compañía,
company.country AS Pais_Compañía
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 company.id = transaction.company_id;

```

“Nota: he separado las columnas, una por línea para mayor claridad, también he añadido más columnas aparte de las específicas que se pide, como indica en el último punto”

Comprobamos el resultado:

```

162 • SELECT *
163 FROM InformeTecnico;

```

Result Grid     Filter Rows:   Export:  Wrap Cell Content:							
	id_transaccion	monto_transaccion	nombre_usuario	apellido	Pais_usuario	iban	compañía
▶	108B1D1D-5B23-A76C-55EF-C568E49A05DD	293.57	Kenyon	Hartman	Canada	TR323456312213576817699999	Ac Fermentum Inco
	EA2C3281-C9C1-A387-44F8-729FB4B51C76	119.36	Kenyon	Hartman	Canada	TR323456312213576817699999	Ac Fermentum Inco
	7DC26247-20EC-53FE-E555-B6C2E55CA5D5	312.50	Kenyon	Hartman	Canada	DO26854763748537475216568689	Magna A Neque Inc
	FE96CE47-8D59-381C-4E18-E3CA3D44E8FF	480.13	Kenyon	Hartman	Canada	DO26854763748537475216568689	Magna A Neque Inc
	72997E96-DC2C-A4D7-7C24-66C302F8AE5A	239.87	Kenyon	Hartman	Canada	BG45IVQL52710525608255	Fusce Corp.
	8768FDE2-A231-B916-8644-F7DCD13CAFC2	460.38	Kenyon	Hartman	Canada	BG45IVOL52710525608255	Fusce Corp.