

# Tasca S3.01. Manipulació de taules

Hecho por: Yatmelis Freites

## 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

```
CREATE TABLE credit_card (  
  id varchar(15) PRIMARY KEY,  
  iban varchar(34) NOT NULL unique,  
  pan varchar(25) NOT NULL,  
  pin INT(4) NOT NULL,  
  cvv INT(3) NOT NULL,  
  expiring_date varchar(8) NOT NULL  
);
```

1. Para este ejercicio, se escribe la declaración **CREATE** seguido de **TABLE** para indicar que se está **creando la tabla credit\_card**, tal y como pide el enunciado.
2. Dentro de la declaración se escriben los **datos de la columna id**, se caracteriza como dato tipo **varchar** (contiene números y letras y símbolos) y se indica la longitud del texto que estará contenido dentro de este campo, además, se restringe como **PRIMARY KEY** ya que es el campo que contiene valores que no se repiten y que permiten su identificación y relación con la tabla transaction.
3. Se escriben los datos del campo **iban** como tipo de dato **varchar** y admitiendo **34 caracteres**, este campo se marca como **NOT NULL** para indicar que no debe estar vacío y además se coloca una restricción como clave única (**UNIQUE**) ya que no existen dos números de cuenta iguales.
4. Se escriben los datos del campo **pan** como tipo de dato **varchar** y admitiendo **25 caracteres**, este campo se marca como **NOT NULL** para indicar que no debe estar vacío.

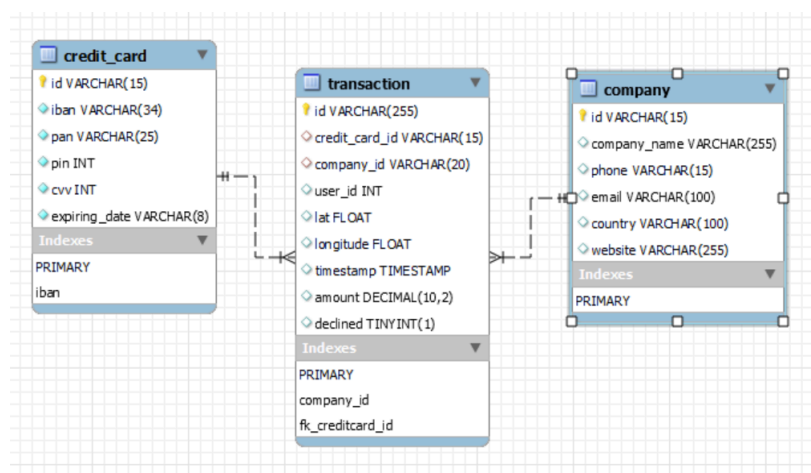
- Se escriben los datos para el campo **pin** como tipo de dato **INT** (numérico) y admitiendo **4 caracteres**, este campo se marca como **NOT NULL** para indicar que no debe estar vacío.
- Se escriben los datos para el campo **CVV** como tipo de dato **INT** (numérico) y admitiendo **3 caracteres**, este campo se marca como **NOT NULL** para indicar que no debe estar vacío.
- Se escriben los datos del campo **expiring\_date** como tipo de dato **varchar** y admitiendo **8 caracteres**, este campo se marca como **NOT NULL** para indicar que no debe estar vacío.

Después de creada la estructura de la tabla en la BBDD, ejecuto el código SQL del archivo “dades\_introducir\_credit” con lo que se llenan las filas y se llena la tabla.

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

	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	R323456312213576817699999	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
	CcU-3008	LU507216693616119230	4485744464433884	1856	740	04/05/25

El esquema de la BBDD MODIFICADO SE VE DE LA SIGUIENTE MANERA



La **relación** entre las tablas **credit\_card** y **transaction** es de **una a muchas** desde **credit\_card** a **transaction** ya que solo hay **una tarjeta de crédito** por cada **transacción**. Se encuentra el **iban** en la tabla **credit\_card** indexado como **clave única**.

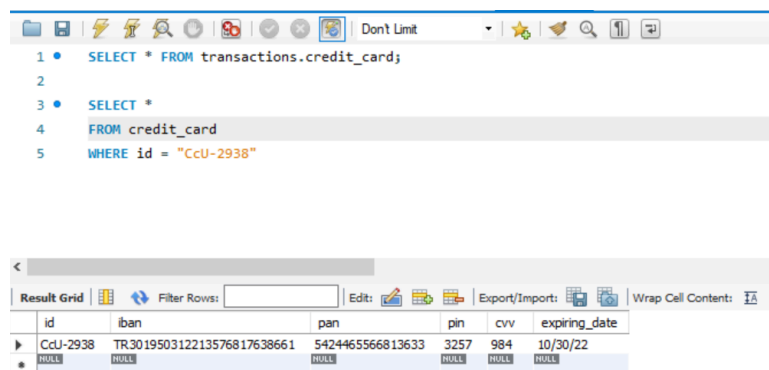
La **relación** entre las tablas **company** y **transaction** es de **una a muchas de company hacia transaction** ya que solo hay una compañía por cada transacción.

La **tabla de hechos** es la **tabla transaction** en este caso, que se relaciona con la tabla de dimensión **company** desde **transaction.company\_id(FK) a company.id (PK)** y con la **tabla de dimensión credit\_card** desde **transaction.credit\_card\_id(FK) a credit\_card.id(PK)**.

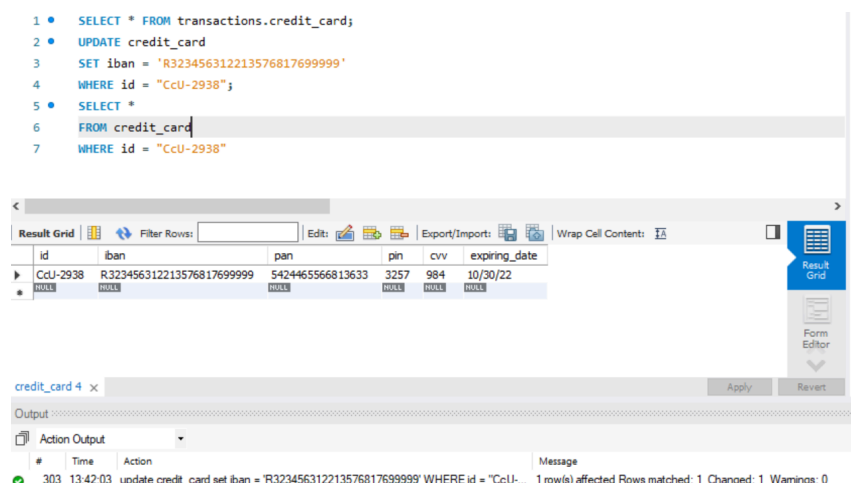
Por otra parte la PK de la tabla transaction es transaction.id

## - 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.



1. Esta query selecciona (**SELECT**) todos los datos de la tabla **credit\_card** haciendo un filtro con **WHERE** para encontrar el **id** que sea igual a **CcU-2938**, donde se muestra el **número de iban erróneo**



1. Se escribe una query para actualizar el iban (**UPDATE**) **credit\_card** para hacer referencia a la tabla a manipular. **SET iban (para indicar el campo a cambiar) = "R323456312213576817699999"** haciendo un filtro con **WHERE** para encontrar el **id que sea igual a CcU-2938**, donde se realizará el cambio.
2. Para confirmar que se ha realizado el cambio, se escribe la query inicial.

## - 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

```

27 /*Exercici 3
28 En la taula "transaction" ingresa un nou usuari amb la següent informació:*/
29 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
30   VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999',
31   '111.11', '0');
32 -- company_id 'b-9999' no existe en la tabla company, por lo que debes insertar y todo queda nulo

```

output

#	Time	Action	Message
1	10:13:13	SELECT * FROM transactions.company	100 row(s) returned
2	10:16:21	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, am...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (tra

1. Para este ejercicio, se escribe la declaración **INSERT INTO transaction** para indicar que se debe insertar los datos indicados en el enunciado dentro de la tabla **transaction** con **VALUES** se indica entre paréntesis los valores para los campos indicados en **INSERT INTO**.
2. Al ejecutar la query muestra un error relacionado con el **company\_id** (FK), no existe un PK en la tabla **company** que relacione esta fila de datos, por lo que debo insertar este id en la tabla **company**
3. También alteré las tablas y los valores default para las columnas **iban**, **pin**, **cvv** y **expiring\_date** para que me permita añadir una columna y dejar el resto de los

campos con valores default. Esto lo hice de la siguiente manera: escribiendo la declaración **ALTER TABLE**, indicando el nombre de la tabla a modificar e indicando luego con **ALTER COLUMN** seguido del nombre de la columna a modificar, y posteriormente indicando con **SET DEFAULT** el valor que tomará por defecto. **NULL** corresponde a los tipos de dato varchar y 0 para los datos numéricos

```
35 • ALTER TABLE credit_card
36     ALTER COLUMN iban
37     SET DEFAULT 'Null';
38 • ALTER TABLE credit_card
39     ALTER COLUMN pin
40     SET DEFAULT '0';
41 • ALTER TABLE credit_card
42     ALTER COLUMN cvv
43     SET DEFAULT '0';
44 • ALTER TABLE credit_card
45     ALTER COLUMN expiring_date
46     SET DEFAULT 'NULL';
```

4. **credit\_card\_id 'CcU-9999'** no existe en la tabla **credit\_card**, por lo que debo insertar esa PK en su tabla. Declaro **INSERT INTO** indicando la tabla y la columna entre **paréntesis(id)** y luego indico con **VALUES** el valor que va en la columna id

```
INSERT INTO credit_card (id)
VALUES ('CcU-9999');
```

5. Por último, ejecuto de nuevo la query inicial y los datos se agregan así a la tabla **transaction de la siguiente manera**

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

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount
10881D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	NULL	111.11

## - Exercici 4

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

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

id	iban	pan	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	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
CcU-3008	LU507216693616119230	4485744464433884	1856	740	04/05/25

1. Esta query selecciona (**SELECT**) todos los datos de la tabla **credit\_card**.

```
1 • SELECT *
2 FROM transactions.credit_card;
3
4 • ALTER TABLE credit_card
5 DROP COLUMN pan;
```

id	iban	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	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

2. Esta query altera la estructura de la tabla (**ALTER TABLE**) **credit\_card**, y con **DROP COLUMN** se indica que se debe borrar la columna **pan**

3. Para confirmar que se ha realizado el cambio, se escribe la query inicial.

## Nivell 2

### Exercici 1

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

```
1 • SELECT * FROM transactions.transaction;
2 • SELECT *
3 FROM transaction
4 WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
5
6
```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	d
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	92	81.9185	-12.5276	2021-08-28 23:42:24	466.92	0

1. Esta query selecciona (**SELECT**) **todos los datos** de la tabla **transaction** haciendo un filtro con **WHERE** para encontrar el **id** que **sea igual a** 02C6201E-D90A-1859-B4EE-88D2986D3B02, donde se realizará el cambio.

```

1 • SELECT * FROM transactions.transaction;
2 • SELECT *
3 FROM transaction
4 WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
5
6 • DELETE FROM transaction
7 WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
8

```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Contents:

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

2. Se escribe una query para borrar (**DELETE**) de la tabla transaction y se filtra con **WHERE** para encontrar el **id** que sea igual a 02C6201E-D90A-1859-B4EE-88D2986D3B02.
3. Para confirmar que se ha realizado el cambio, se escribe la query inicial.

## 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.

```

11 • SELECT company.company_name, company.phone, company.country,
12 AVG(transaction.amount) AS mediana_compra_por_compañia
13 FROM company
14 JOIN transaction
15 ON company.id = transaction.company_id
16 GROUP BY company.company_name, company.phone, company.country
17 ORDER BY AVG(transaction.amount) DESC

```

company_name	phone	country	mediana_compra_por_compañia
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

4. Para crear la vista, primero se crea una query donde se selecciona (**SELECT**) **company.company\_name, company.phone y company.country** de la **tabla company**, además de la mediana de **transaction.amount** de la tabla de la

tabla **transaction** usando la función **AVG()** y catalogando con el alias **mediana\_compra\_por\_compañía**, se realiza un **JOIN** de la tabla **company** con la tabla **transaction** que se unen mediante **company.id = transaction.company\_id**.

5. Luego agrupo con **GROUP BY** por **company.company\_name**, **company.phone** y **company.country** y ordeno mediante **ORDER BY** por la mediana de **transaction.amount** de manera descendente como **AVG(transaction.amount) DESC**

```

• CREATE VIEW VistaMarketing AS
SELECT company.company_name, company.phone, company.country,
AVG(transaction.amount) AS mediana_compra_por_compañía
FROM company
JOIN transaction
ON company.id = transaction.company_id
GROUP BY company.company_name, company.phone, company.country
ORDER BY AVG(transaction.amount) DESC;



```

6. Después de tener la query, se escribe la declaración **CREATE VIEW** y se escribe el nombre que se indica en el enunciado: **VistaMarketing AS** para indicar que la vista es de la query previamente escrita

```

1 • SELECT * FROM transactions.vistamarketing;

```

Result Grid			
Filter Rows:		Export: 	
Wrap Cell Content: 			
company_name	phone	country	mediana_compra_por_compañía
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

7. Luego de tener la vista creada, se puede escribir una query para seleccionar (**SELECT**) todos los elementos de **vistamarketing** de una manera más sencilla.



## Exercici 3

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

```
1 • SELECT *
2 FROM transactions.vistamarketing
3 WHERE country = "Germany";
```

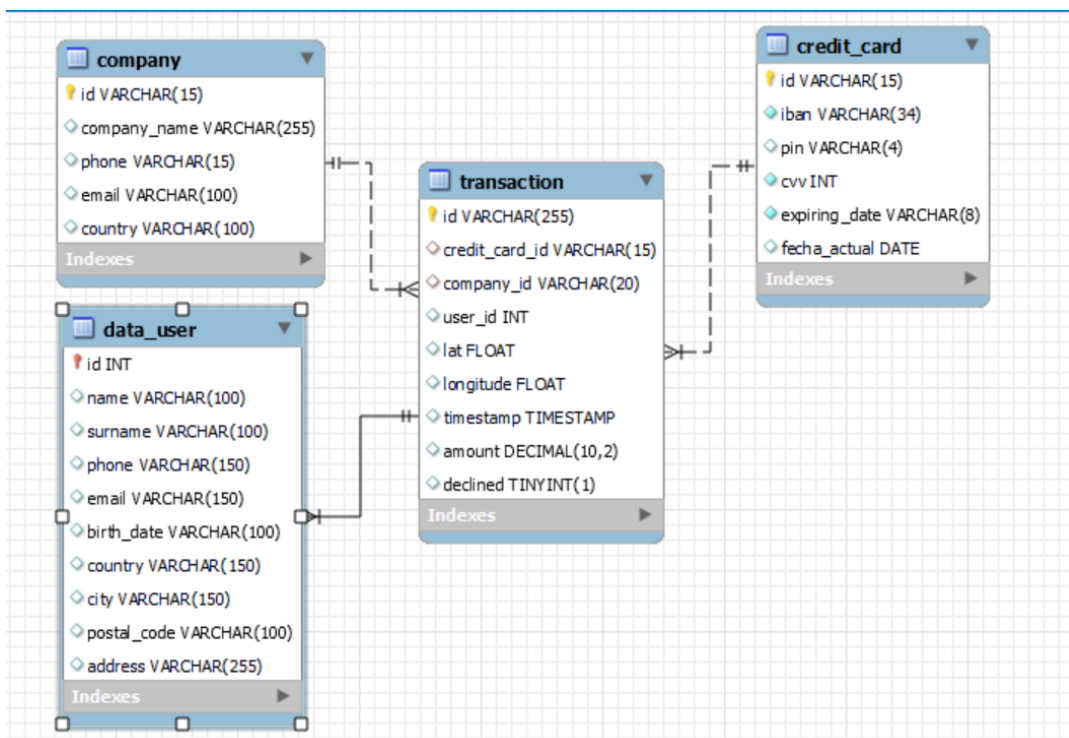
company_name	phone	country	mediana_compra_por_compañia
Aliquam PC	01 45 73 52 16	Germany	385.265000
Ac Industries	09 34 65 40 60	Germany	289.645000
Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000
Nunc Interdum Incorporated	05 18 15 48 13	Germany	244.025238
Augue Foundation	06 88 43 15 63	Germany	240.800000
Ac Fermentum Incorporated	06 85 56 52 33	Germany	206.465000
Auctor Mauris Cras	05 62 87 14 41	Germany	184.310000

Acá selecciono (**SELECT**) el **TODOS LOS DATOS** de la vista **vistamarketing** y se realiza un filtro con **WHERE** la columna **country** es igual a **"Germany"**.

## Nivell 3

### Exercici 1

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



## Recordatori

En aquesta activitat, és necessari que descrigueu el "pas a pas" de les tasques realitzades. És important realitzar descripcions senzilles, simples i fàcils de comprendre. Per a realitzar aquesta activitat hauràs de treballar amb els arxius denominats "estructura\_dades\_user" i "dades\_introduir\_user"

En primer lugar revisaría los archivos de la BBDD, con lo que encuentro el archivo "estructura\_dades\_user" y lo ejecuto

```
estructura_datos_user x
-- Creamos la tabla user
1
2
3 • CREATE INDEX idx_user_id ON transaction(user_id);
4
5 CREATE TABLE IF NOT EXISTS user (
6     id INT PRIMARY KEY,
7     name VARCHAR(100),
8     surname VARCHAR(100),
9     phone VARCHAR(150),
10    email VARCHAR(150),
11    birth_date VARCHAR(100),
12    country VARCHAR(150),
13    city VARCHAR(150),
14    postal_code VARCHAR(100),
15    address VARCHAR(255),
16    FOREIGN KEY(id) REFERENCES transaction(user_id)
17 );
```

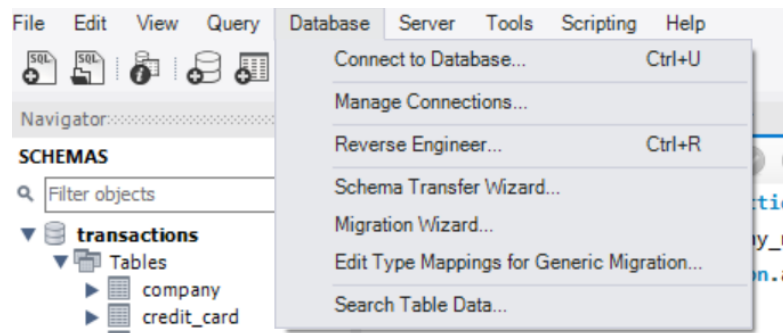
Con esto se crea la estructura de la base de datos

The screenshot shows a database management interface. On the left, the 'SCHEMAS' pane displays the 'user' table structure with columns: id (INT PK), name (varchar(100)), surname (varchar(100)), phone (varchar(150)), email (varchar(150)), birth\_date (varchar(100)), country (varchar(150)), and city (varchar(150)). The main pane shows the execution of a script named 'datos\_introduir\_user(1)'. The script includes a SET statement for foreign key checks, followed by a series of INSERT statements into the 'user' table, and finally a SET statement to turn foreign key checks back on. The output pane shows the execution results, indicating that the scripts were executed successfully and the data was inserted into the table.

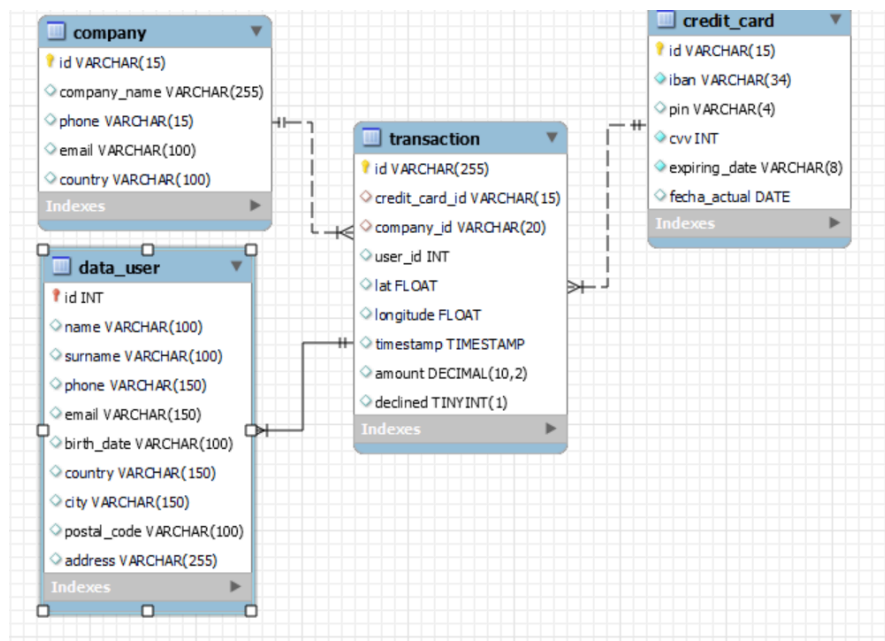
Luego ejecuto el archivo "dades\_introduir\_user" y lo ejecuto, con lo que se llenan las filas de la tabla

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

id	name	surname	phone	email	birth_date	country	city	postal_co
1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.edu	Nov 17, 1985	United States	Lowell	73544
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Des Moines	59464
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Columbus	56518
4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.edu	Feb 18, 1989	United States	Kailua	77417
5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	Sep 26, 1998	United States	Sandy	31564
6	Intel	Tuenn	(718) 788-8070	gravida.nunc.sed@uolun.ca	Oct 15, 1989	United States	Nashville	66838



Con Reverse Engineer se crea el modelo, y se compara con el diagrama que se solicita inicialmente y se observan los cambios que se deben realizar en la BBDD

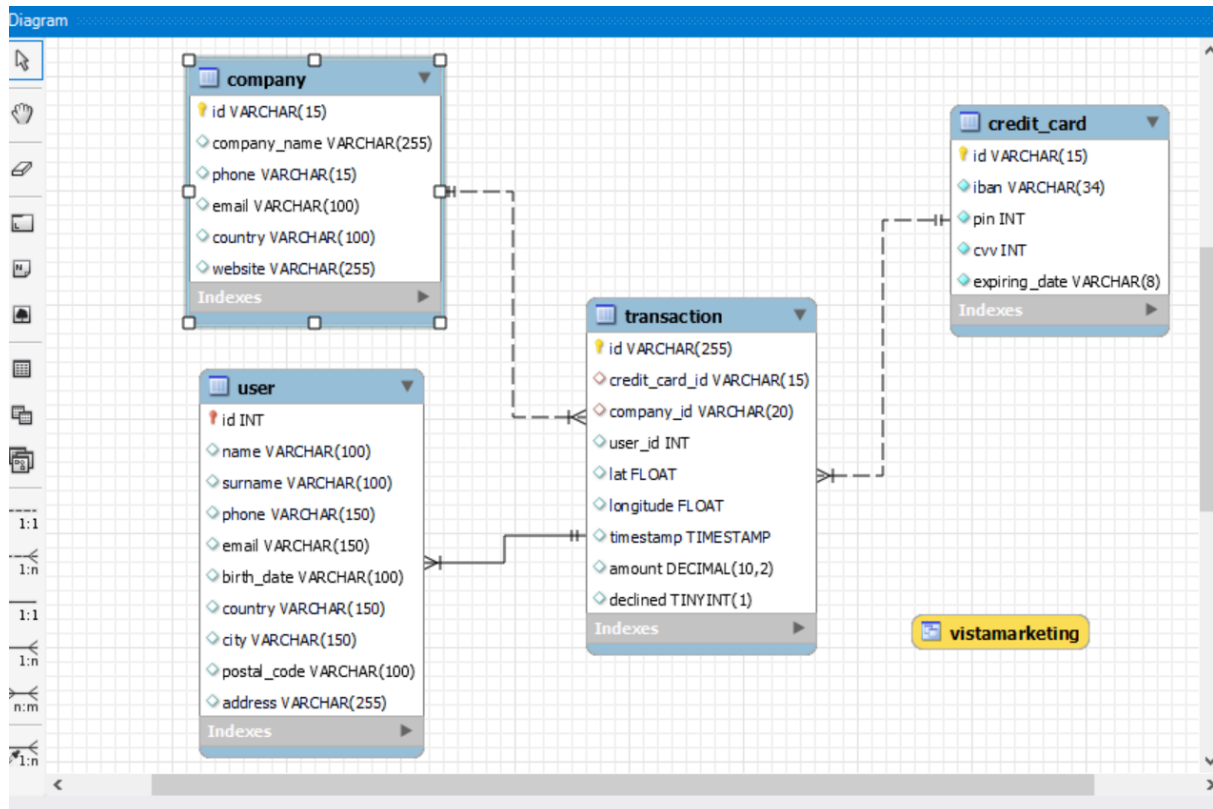


Se realizan las queries para realizar cambios en las tablas

- `ALTER TABLE company  
DROP COLUMN website;`
  - `RENAME TABLE user TO data_user;`
  - `ALTER TABLE credit_card MODIFY pin varchar(4);`
  - `ALTER TABLE credit_card ADD COLUMN fecha_actual DATE; /`
- a. Con ALTER TABLE se indica que se modificará la tabla company, para eliminar la columna website ya que no aparece en el diagrama modelo entregado, se escribe DROP COLUMN website.
  - b. La tabla user debe cambiar su nombre a data\_user y se realiza el cambio con el comando RENAME TABLE, indicando el nombre de la tabla a cambiar TO para indicar el nombre a usar data\_usere
  - c. Con ALTER TABLE se indica que se modificará la tabla credit\_card, para modificar el tipo de valor contenido en la columna pin mediante MODIFY, indicando el nombre de la columna a modificar y escribiendo el tipo de valor nuevo (VARCHAR) y el número de caracteres.

- d. Con ALTER TABLE se indica que se modificará la tabla credit\_card, para AGREGAR una nueva columna mediante ADD COLUMN fecha\_actual y se escribe DATE para determinar el tipo de dato que va dentro de la columna.

Finalmente se actualiza la BBDD y con el Reverse Engineer se genera el esquema corregido



## Exercici 2

L'empresa també et sol·licita crear una vista anomenada "InformeTecnico" que contingui la següent informació:

- ID de la transacció
- Nom de l'usuari/ària
- Cognom de l'usuari/ària
- IBAN de la targeta de crèdit usada.
- Nom de la companyia de la transacció realitzada.
- Assegura't d'incloure informació rellevant de totes dues taules i utilitza àlies per a canviar de nom columnes segons sigui necessari.

Mostra els resultats de la vista, ordena els resultats de manera descendent en funció de la variable ID de transaction.

1. Para crear la vista, primero se crea una query donde se selecciona (**SELECT**) **transaction.id** y se renombra con el alias **AS id\_transaction**, **data\_user.name** y se renombra con el alias **AS nombre\_usuario**, **data\_user.surname** y se renombra con el alias **AS apellido\_usuario**, **credit\_card.iban** y se renombra con el alias **AS iban\_tarjeta**, **company.company\_name** y se renombra con el alias **AS nombre\_compañia**,

**company.country** y se renombra con el alias **AS pais\_compañía** de la tabla **FROM data\_user**, se realiza un **JOIN** de la tabla **company** con la tabla **transaction** que se unen mediante **data\_user.id = transaction.user\_id**. Se realiza un segundo **JOIN** de la tabla **credit\_card** con la tabla **transaction**, que se unen mediante **credit\_card.id = transaction.credit\_card\_id**. Se realiza un último **JOIN** de la tabla **company** que se unen mediante **company.id = transaction.company\_id**

2. Luego se ordena de manera descendente con el comando **ORDER BY transaction.id DESC;**

```

2 • SELECT
3     transaction.id AS id_transaction,
4     data_user.name AS nombre_usuario,
5     data_user.surname AS apellido_usuario,
6     credit_card.iban AS iban_tarjeta,
7     company.company_name AS nombre_compañía,
8     company.country AS pais_compañía
9 FROM data_user
10 JOIN transaction
11 ON data_user.id = transaction.user_id
12 JOIN credit_card
13 ON credit_card.id = transaction.credit_card_id
14 JOIN company
15 ON company.id = transaction.company_id
16 ORDER BY transaction.id DESC;

```

id_transaction	nombre_usuario	apellido_usuario	iban_tarjeta	nombre_compañía	pais_compañía
FE96CE47-8D59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries	Australia
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gillam	SE2813123487163628531121	Nunc Interdum Incorporated	Germany
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated	Germany
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC	Ireland

3. Después de creada la query, se escribe **CREATE VIEW InformeTecnico AS** encima de esta

```

CREATE VIEW InformeTecnico AS
SELECT
    transaction.id AS id_transaction,
    data_user.name AS nombre_usuario,
    data_user.surname AS apellido_usuario,
    credit_card.iban AS iban_tarjeta,
    company.company_name AS nombre_compañía,
    company.country AS pais_compañía
FROM data_user
JOIN transaction
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
ORDER BY transaction.id DESC;

```

4. Al seleccionar todos los elementos de la nueva vista creada, se presentan los siguientes resultados:

1 • `SELECT * FROM transactions.informetecnico;`

Result Grid					Filter Rows:	Export:	Wrap Cell Content:
id_transaction	nombre_usuario	apellido_usuario	iban_tarjeta	nombre_compañia			
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Indust			
FE809ED4-2DB6-55AC-C915-929516E46468	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorpo			
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorpo			
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC			
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwia	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet			