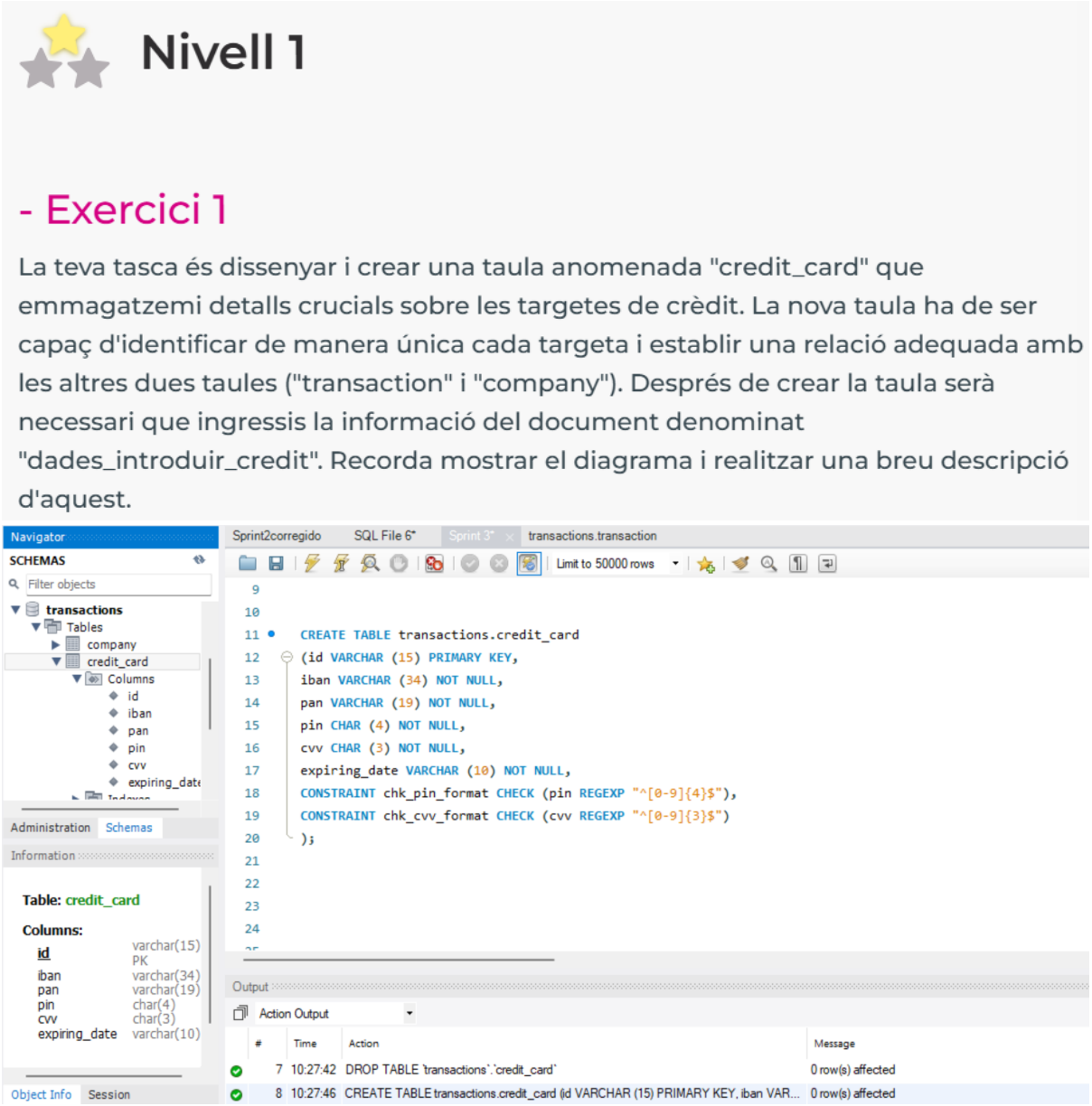
Sprint 3

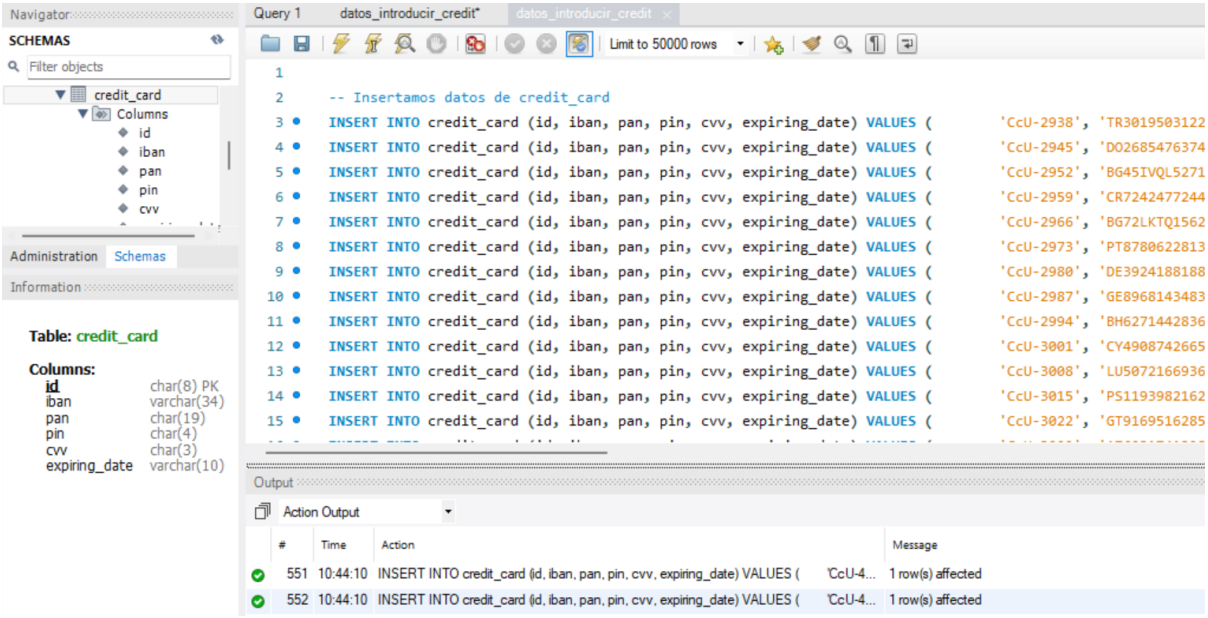
Manipulacion de Tablas



Se crea una tabla llamada credit_card en el esquema transactions.
Se le agregan los campos requeridos (id, iban, pan, pin, cvv, expiring_date) como columnas y se contrastan sus características en el archivo “dades_introduir_credit”, para ver que de que se componen. (Se puede comenzar con VARCHAR 255 y luego adaptar a los datos, sin embargo, he optado por primero observar los datos y sus características y luego crear la tabla acorde a sus valores)
Para poder ligar la nueva tabla con las otras existentes a través de su PrimaryKey (credit_card.id), se explora que columna de las otras tablas hace referencia a esta.
La tabla transaction se relaciona con su ForeignKey (credit_card_id) con la nueva tabla.
Para que se pueda relacionar esta nueva tabla de credit_card con la existente a través de su PrimaryKey, debe tener las mismas características que la ya existente ForeignKey

de transaction. (VARCHAR(15)).
Se tiene en cuenta las características específicas de cada columna, que valores pueden tener, cuantos dígitos serán permitidos, por si serán exclusivamente números o también pueden contener letras, por si son obligatorios u opcionales.
En los campos pin y cvv se le ha aplicado un check constraint para garantizar de que solo contengan números del 0-9, además de que cumplan con la longitud obligatoria de 4 y 3.
Se especifica la PrimaryKey, el id de esta tabla, esto garantiza la unicidad de este valor además de que implica la función NOT NULL.
En las otras columnas se agrega la función NOT NULL, para que los registros sean obligatorios y no puedan dejarse vacíos.

Posteriormente se le introducen todos los datos contenidos en el archivo "dades_introduir_credit"



Luego verifico si se han introducido los datos en la tabla de credit_card

```
23
24
25 •  SELECT *
26    FROM credit_card;
27
28
29
30
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22
CcU-2945	DO26854763748537475216568689	5142423821948828	9080	887	08/24/23
CcU-2952	BG45IVQL52710525608255	4556 453 55 5287	4598	438	06/29/21
CcU-2959	CR7242477244335841535	372461377349375	3583	667	02/24/23
CcU-2966	BG72LKTQ15627628377363	448566 886747 7265	4900	130	10/29/24
CcU-2973	PT87806228135092429456346	544 58654 54343 384	8760	887	01/30/25

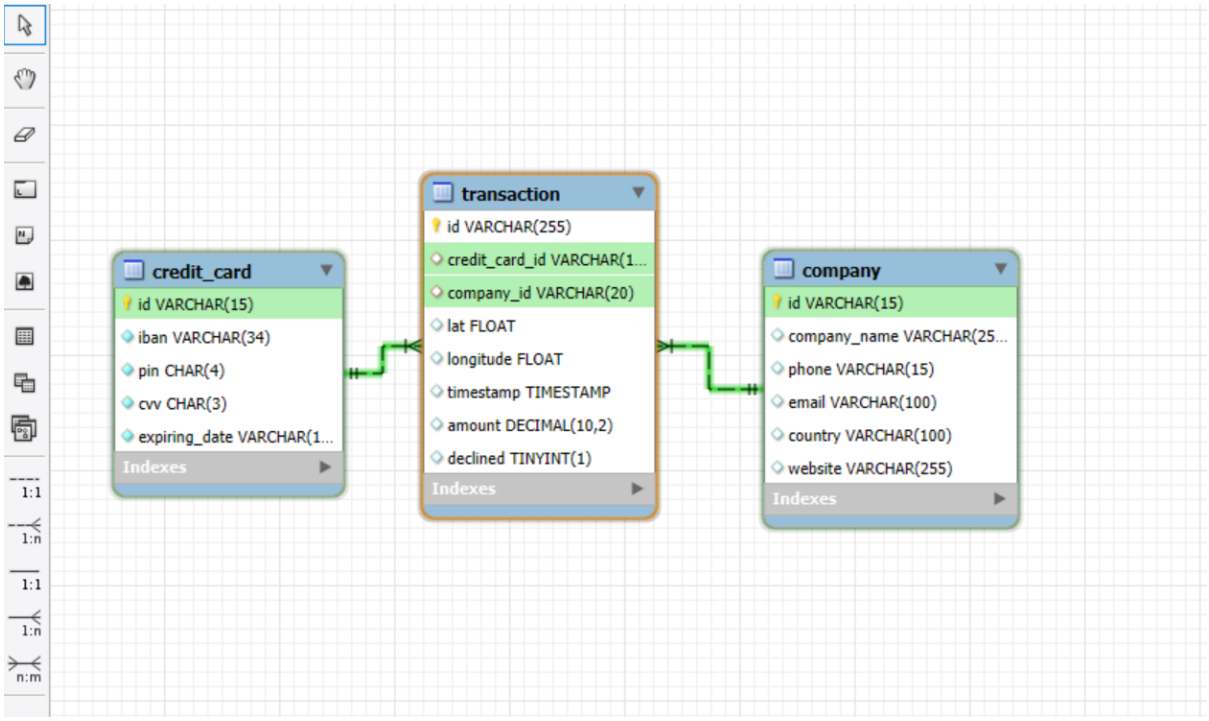
credit_card 2 | Apply

Output

Action Output

#	Time	Action	Message
8	10:27:46	CREATE TABLE transactions.credit_card (id VARCHAR (15) PRIMARY KEY, iban ...	0 row(s) affected
9	10:45:37	SELECT * FROM credit_card LIMIT 0, 50000	275 row(s) returned

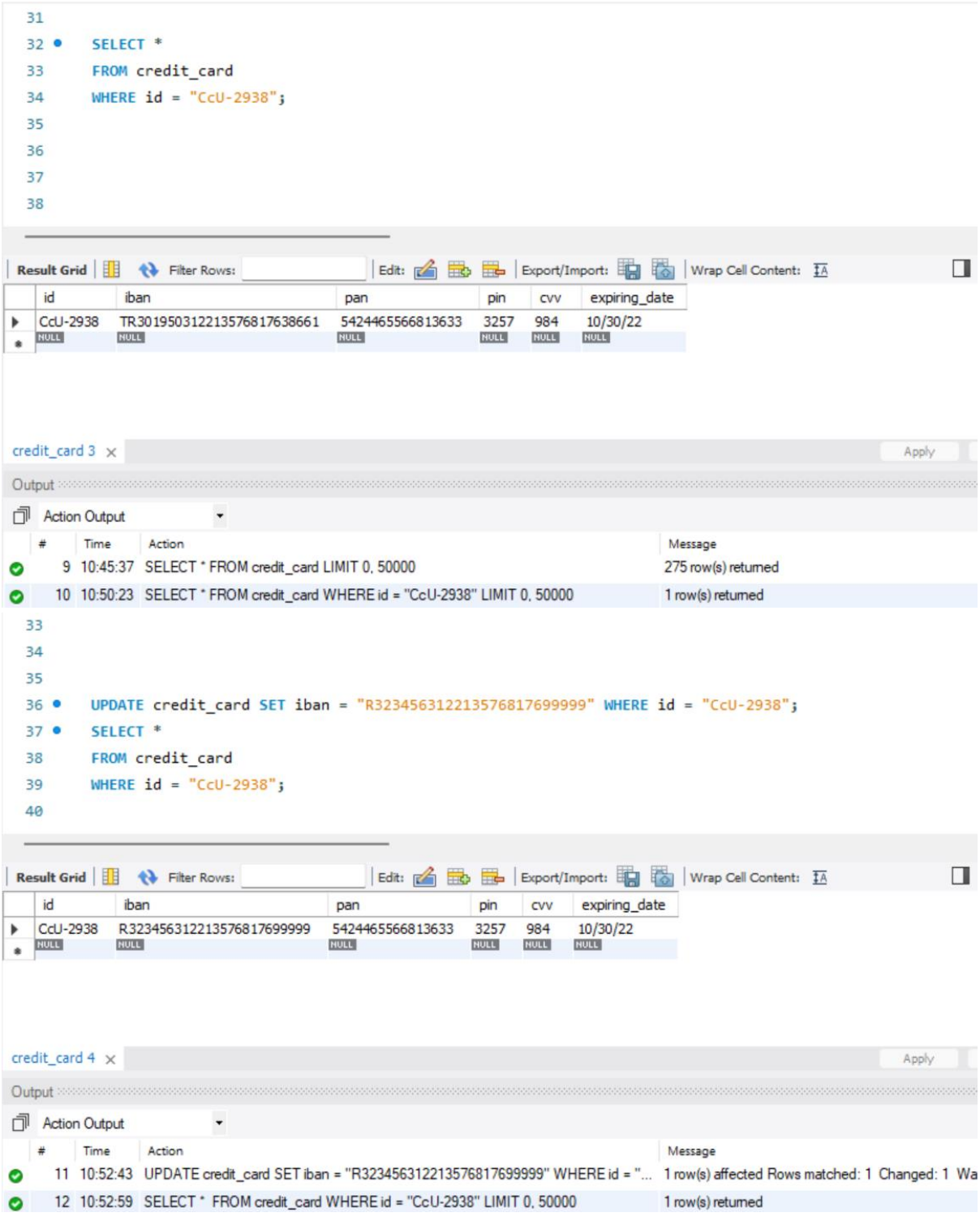
El diagrama resultante es:



- 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.

Se cambia el registro del iban del usuario con el id CcU-2938, al que inicialmente se le habían asignado estos valores:



```
31
32 •  SELECT *
33    FROM credit_card
34    WHERE id = "CcU-2938";
35
36
37
38
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22
NULL	NULL	NULL	NULL	NULL	NULL

credit_card 3 × Apply

Output

Action Output

#	Time	Action	Message
9	10:45:37	SELECT * FROM credit_card LIMIT 0, 50000	275 row(s) returned
10	10:50:23	SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 50000	1 row(s) returned

```
33
34
35
36 •  UPDATE credit_card SET iban = "R323456312213576817699999" WHERE id = "CcU-2938";
37 •  SELECT *
38    FROM credit_card
39    WHERE id = "CcU-2938";
40
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	iban	pan	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	5424465566813633	3257	984	10/30/22
NULL	NULL	NULL	NULL	NULL	NULL

credit_card 4 × Apply

Output

Action Output

#	Time	Action	Message
11	10:52:43	UPDATE credit_card SET iban = "R323456312213576817699999" WHERE id = "...	1 row(s) affected Rows matched: 1 Changed: 1 Wa
12	10:52:59	SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 50000	1 row(s) returned

Con el comando update, se cambia el número de cuenta de este usuario.
Primero se elige la tabla en la que se quiere realizar el cambio, luego el campo que se quiere ”colocar” (set) que en este caso es el iban y se inserta el valor que debe tener, después del =. Además, se tiene que especificar para quien se realiza este cambio, el usuario con el id CcU-2938. Siempre es recomendable usar la Primary-Key, para filtrar el registro en el que se debe efectuar el cambio, para que no se cambien más valores si estos están repetidos, a no ser que sea el propósito.

- Exercici 3

En la taula "transaction" ingressa un nou usuari amb la següent informació:

Id	108B1D1D-5B23-A76C-55EF-C568E49A99DD
credit_card_id	CcU-9999
company_id	b-9999
user_id	9999
lat	829.999
longitude	-117.999
amount	111.11
declined	0

```
50  USE company;
51  INSERT INTO company (id)
52  VALUES ("b-9999");
53  USE transactions;
54  INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
55  VALUES ("108B1D1D-5B23-A76C-55EF-C568E49A99DD", "CcU-9999", "b-9999", "9999", "829.999", "-117.999",
56  111.11, 0);
57  SELECT *
58  FROM transaction
59  WHERE company_id = "b-9999";
60
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	NULL	111.11	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 5 | Apply | Revert

Output

Action Output

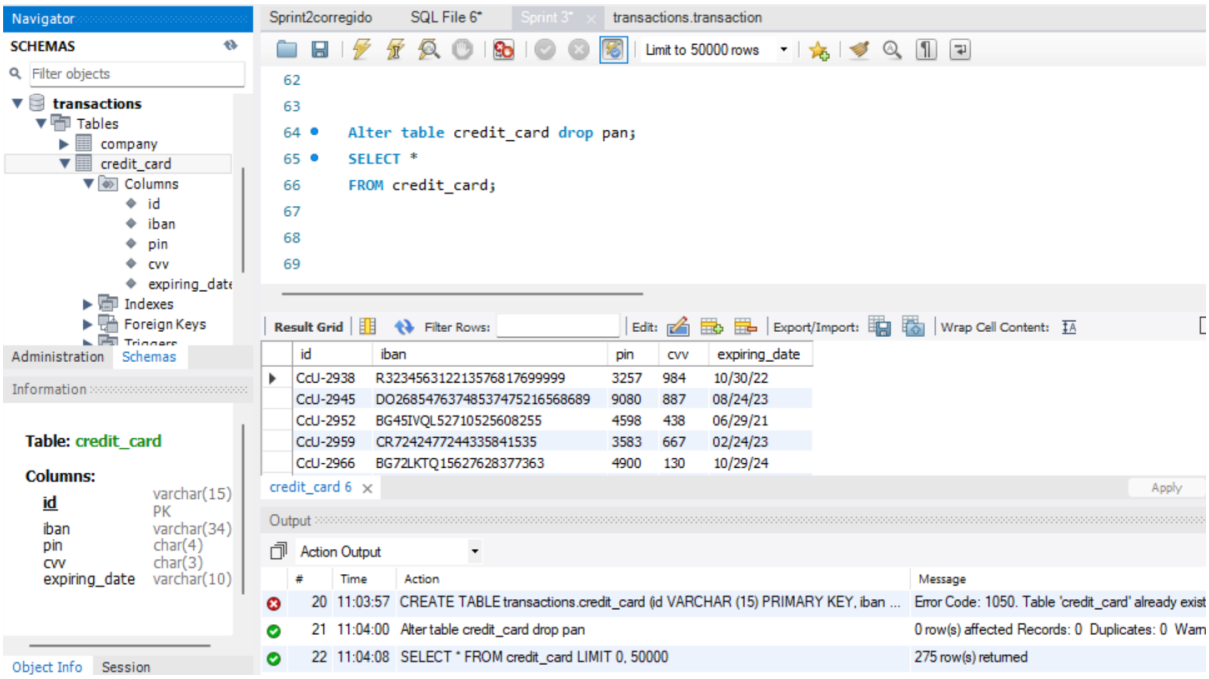
#	Time	Action	Message
17	10:59:13	INSERT INTO company (id) VALUES ("b-9999")	1 row(s) affected
18	10:59:28	INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, a...	1 row(s) affected

Para poder efectuar esta agregación a la tabla de transacciones, primeramente, hay

que agregarle a la tabla de company el company_id, ya que esta tabla está conectada con su Primary Key (los ids de las companys) a la tabla de transacciones.
Posteriormente se agregan los otros datos en la tabla de transacciones.
Podemos verificar que en la tabla de transacciones se han agregado los campos requeridos. El timestamp se ha quedado nulo ya que el dato no existe.

- Exercici 4

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



Nivell 2

Exercici 1

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

```
72
73 •  delete from transaction where id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
74 •  SELECT *
75    FROM transaction where id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
76
77
78
79
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

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

transaction 7 | Apply

Output

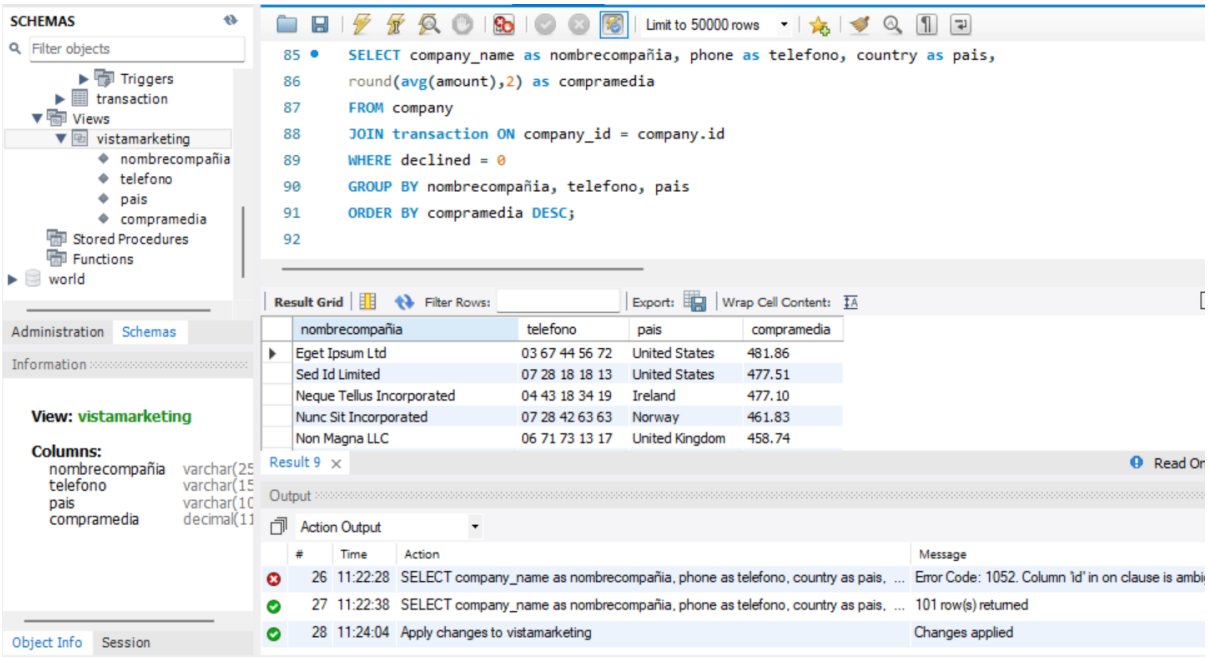
Action Output

#	Time	Action	Message
22	11:04:08	SELECT * FROM credit_card LIMIT 0, 50000	275 row(s) returned
23	11:19:01	delete from transaction where id = "02C6201E-D90A-1859-B4EE-88D2986D3B02"	0 row(s) affected
24	11:19:05	SELECT * FROM transaction where id = "02C6201E-D90A-1859-B4EE-88D2986D...	0 row(s) returned

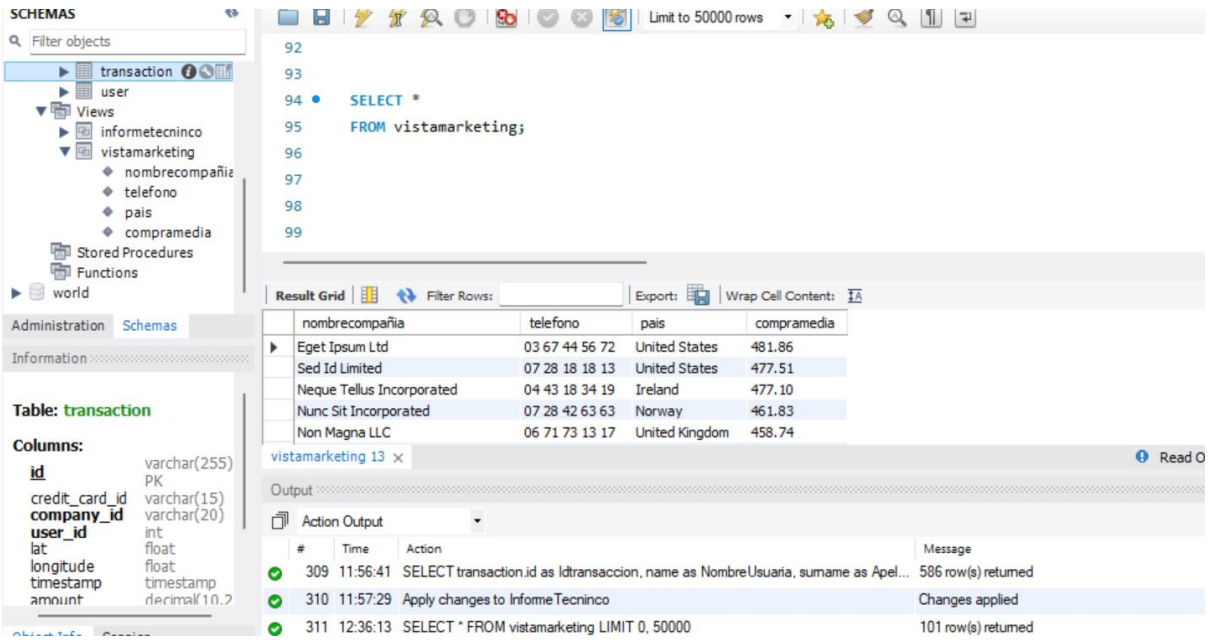
Aqui eliminamos el registro y luego verificamos que este eliminado.

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.



Primero se crea la tabla con los resultados que se quieren ver y luego se crea una Vista a la que podemos acceder con SELECT.



Exercici 3

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

```
97
98
99  SELECT *
100 FROM Vistamarketing
101 WHERE pais IN ("Germany");
102
103
104
```

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

nombrecompañia	telefono	pais	compramedia
Ac Industries	09 34 65 40 60	Germany	396.15
Auctor Mauris Corp.	05 62 87 14 41	Germany	308.99
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.57
Aliquam PC	01 45 73 52 16	Germany	280.34
Rutrum Non Inc.	02 66 31 61 09	Germany	266.90

Vistamarketing 10 ×

Output

Action Output

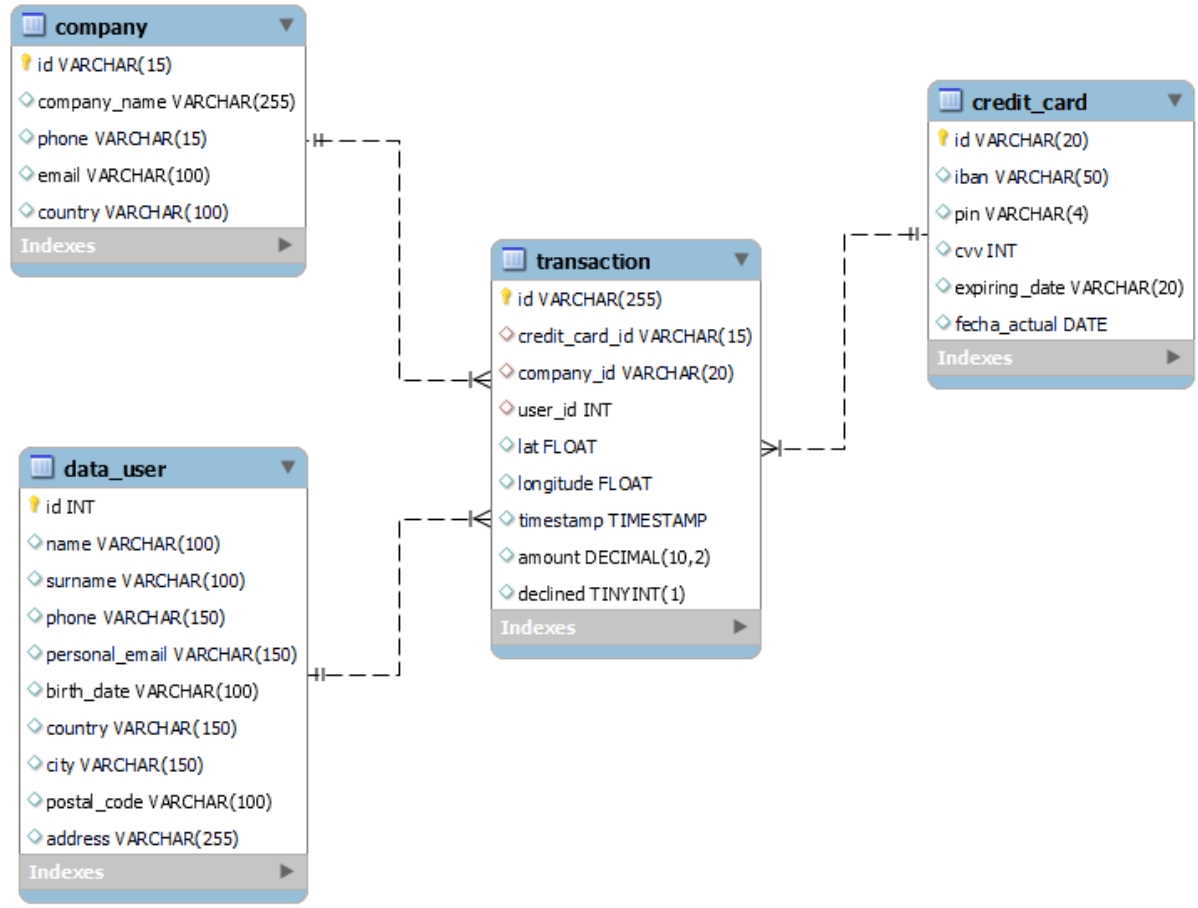
#	Time	Action	Message
27	11:22:38	SELECT company_name as nombrecompañia, phone as telefono, country as pais, ...	101 row(s) returned
28	11:24:04	Apply changes to vistamarketing	Changes applied
29	11:25:18	SELECT * FROM Vistamarketing WHERE pais IN ("Germany") LIMIT 0, 50000	8 row(s) returned

Desde la tabla creada anteriormente en vistamarketing, se filtran los resultados a través de la condición ”Germany”.

Nivell 3

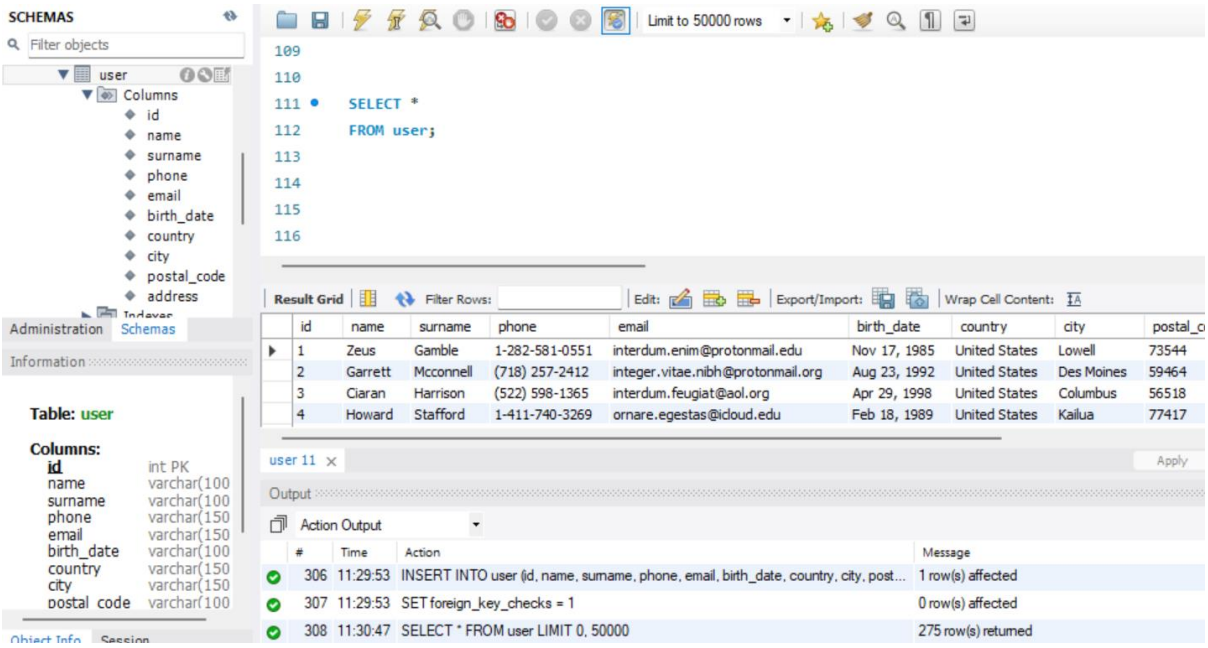
Exercici 1

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:



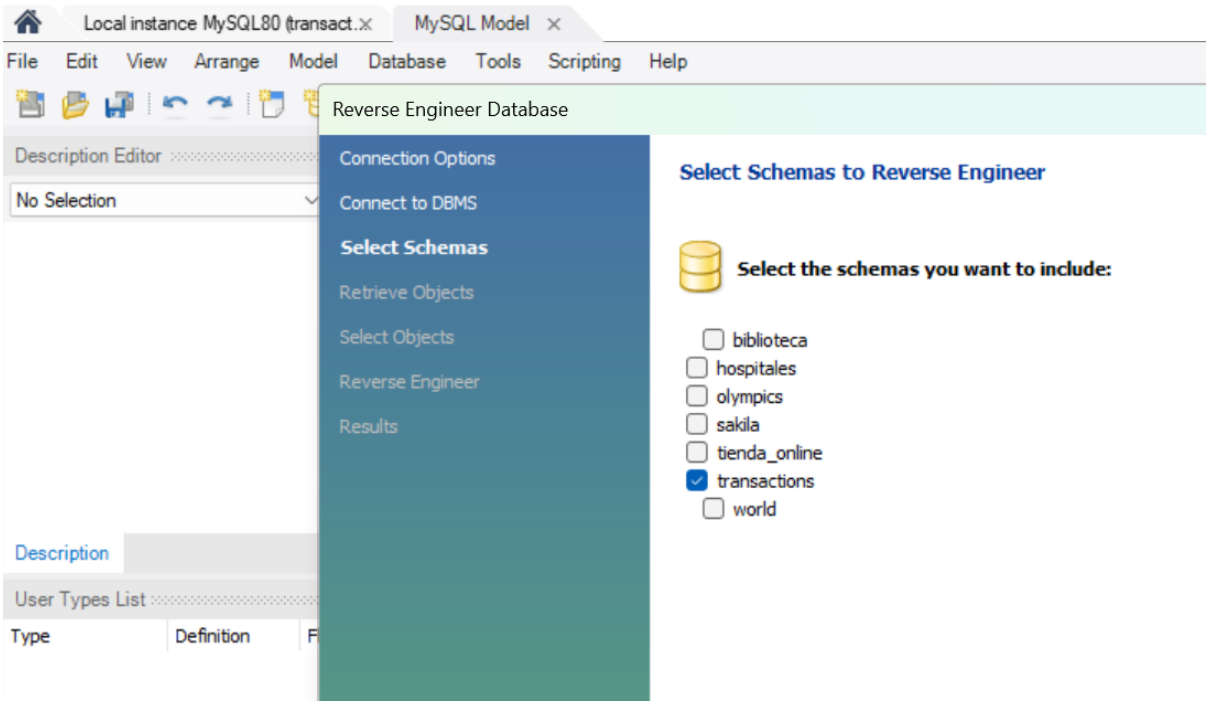
(El diagrama que quiero obtener)

Primero agrego la tabla user con los registros proporcionados.

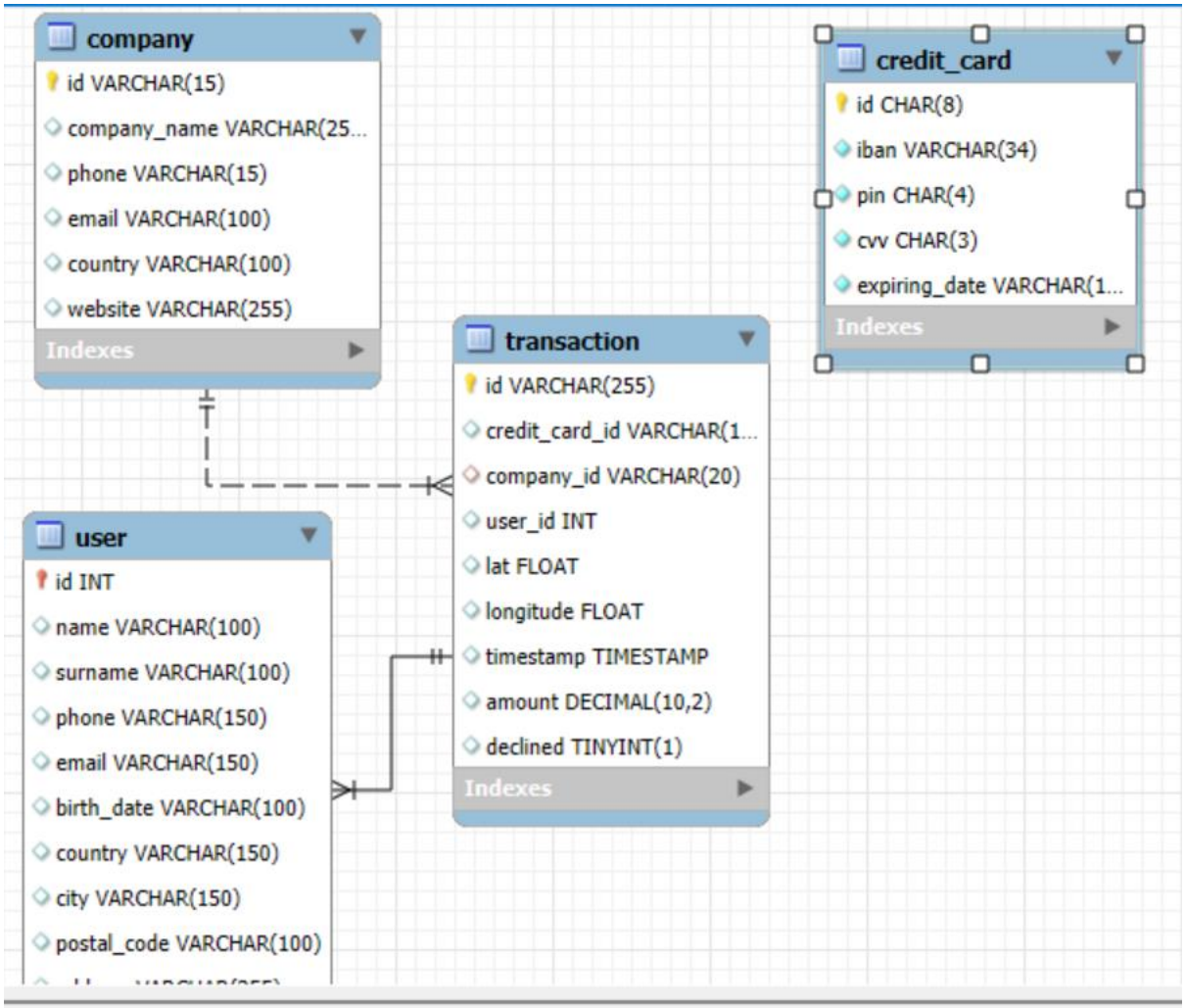


Luego visualizo el esquema a través de Reverse Engineer.

Entro a Database, Reverse Engineer y elijo la tabla en la que quiero administrar las conexiones:

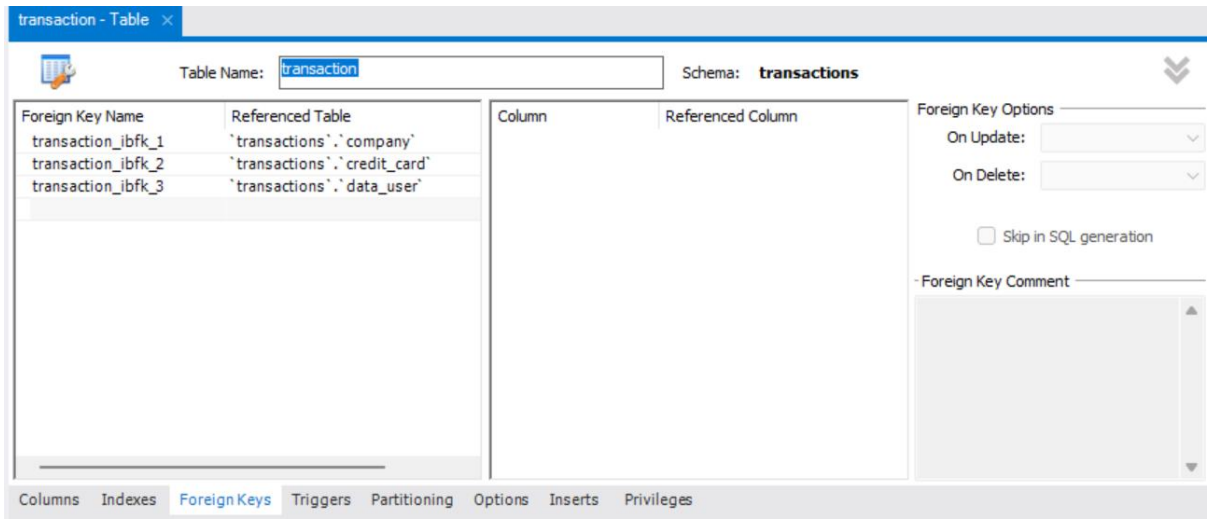


Llego a este diagrama:



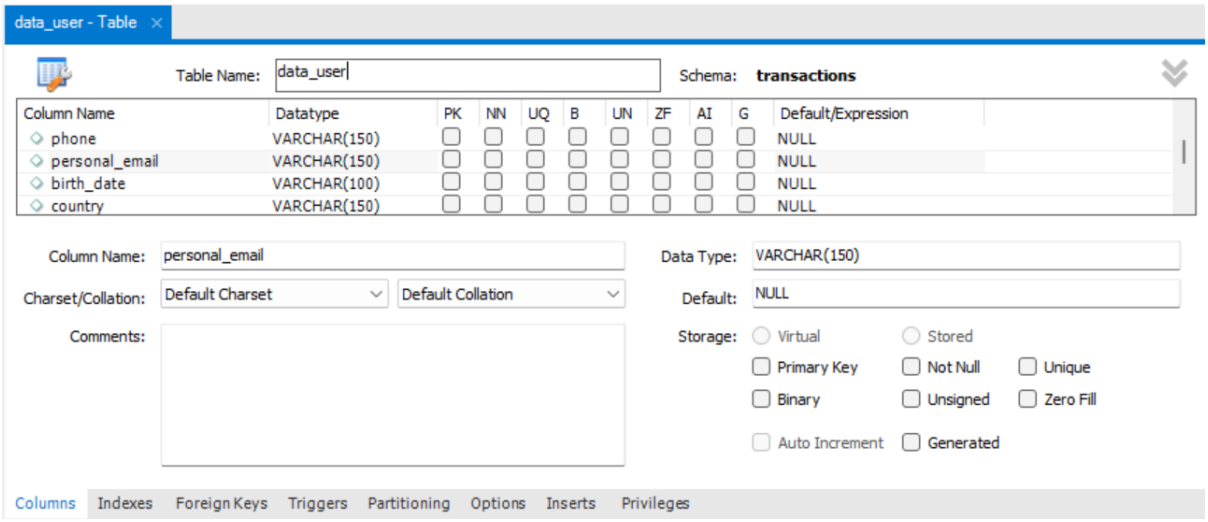
Realizo cambios en la tabla de transaction:

Aqui se administran las conexiones de las Foreign Keys de la tabla de hechos transaction.

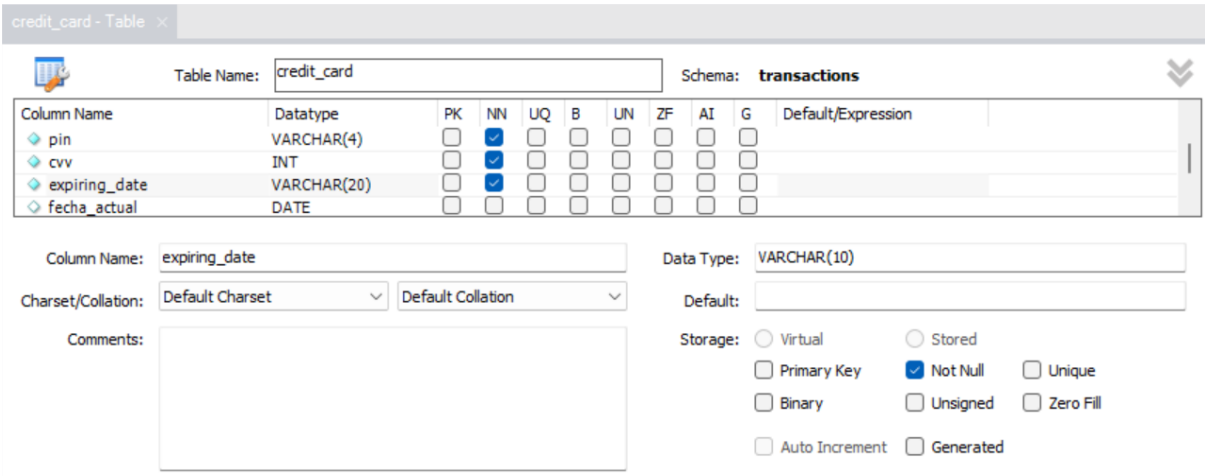


Luego realizo cambios en la tabla de user:

Hay que cambiar el nombre de la tabla y campos. Se pueden editár directamente para que coincida con los nombres del ejercicio. (nombre data_user y personal_email)

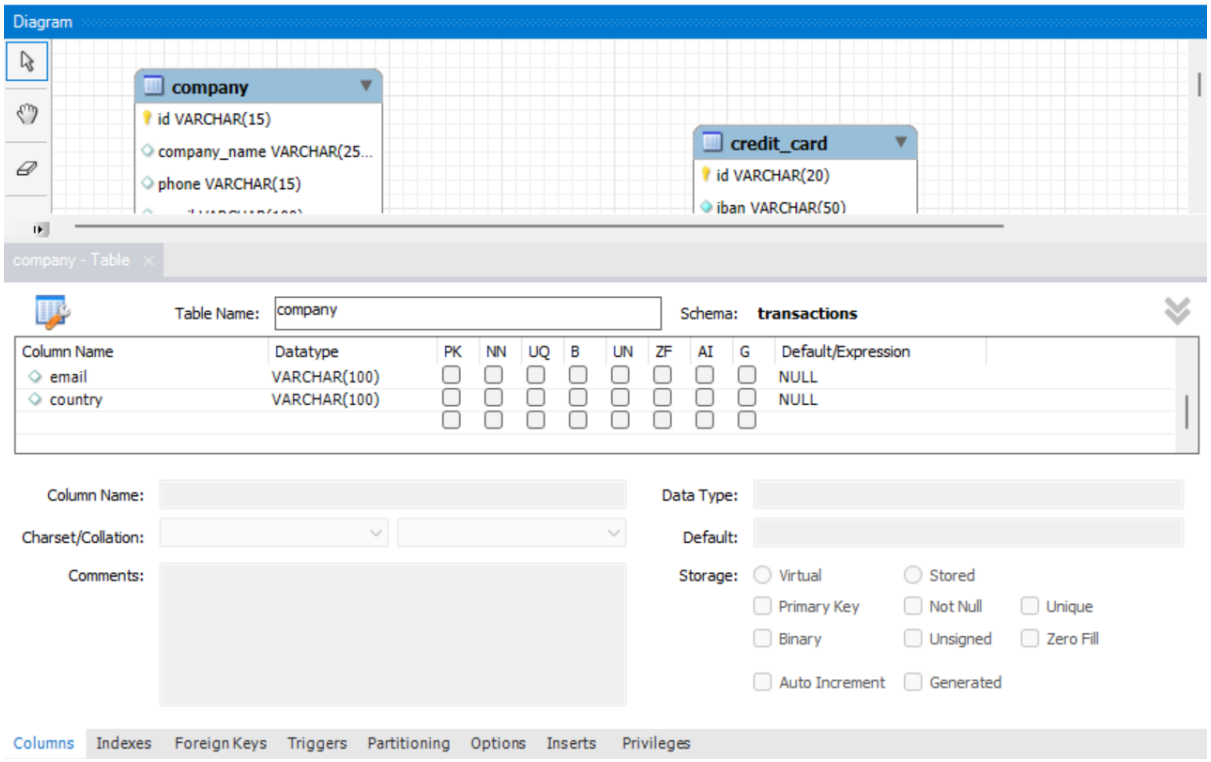


Realizo cambios en la tabla de credit_card:



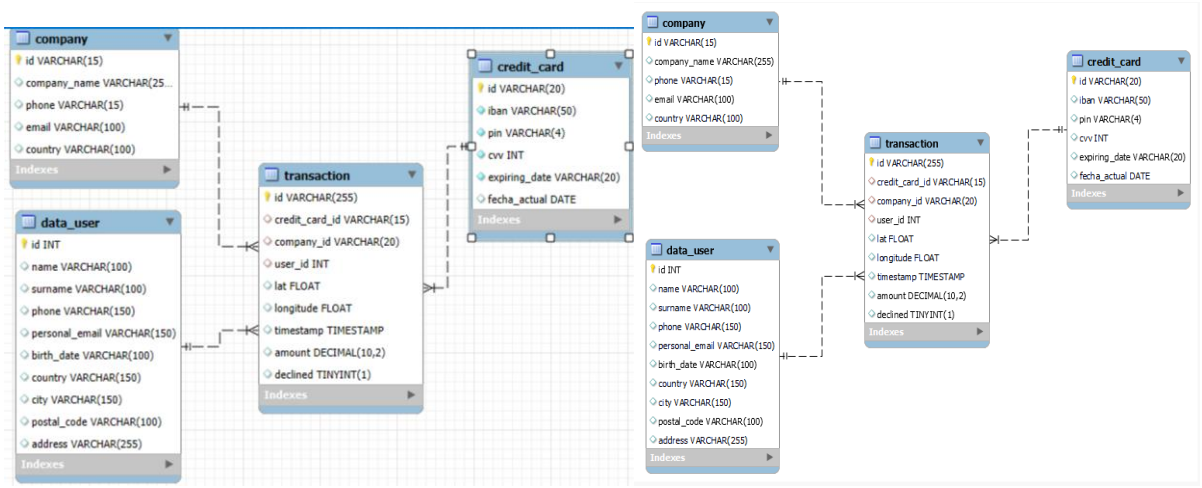
Cambio los Data_types en algunas columnas y agrego una columna de DATE.

Acomodo los datos en la tabla Company:

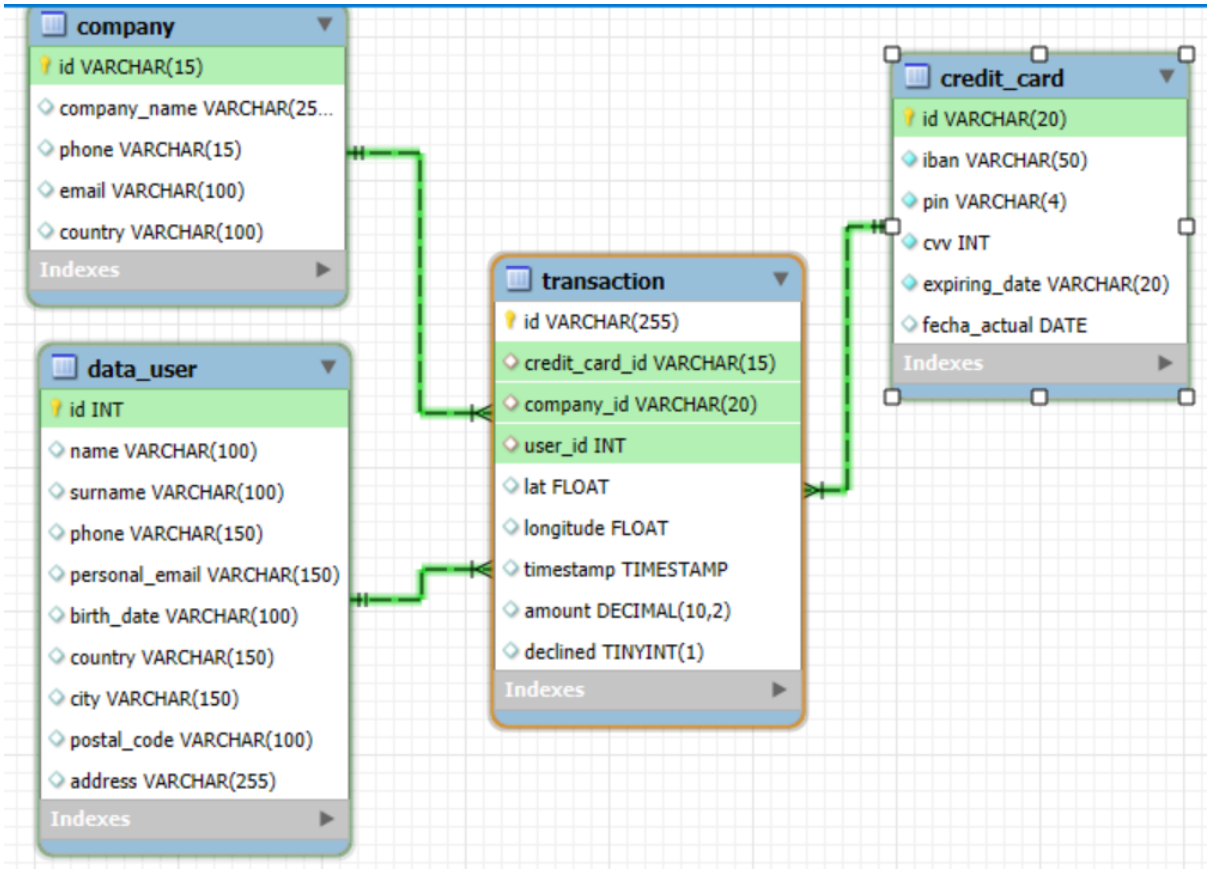


Borro la columna website.

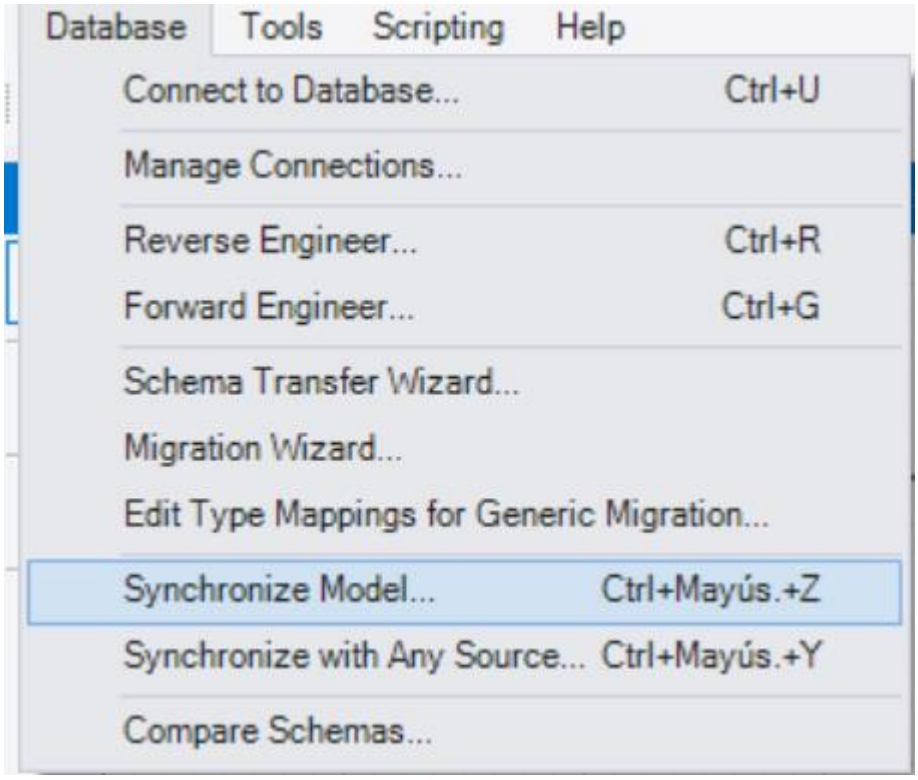
Obtengo este diagrama (Al lado la referencia):



El modelo demuestra estas relaciones con las PrimaryKeys:



Luego sincronizo el modelo de Reversee Engineer con el esquema



Me señala en que tablas se están realizando cambios:

Model and Database Differences

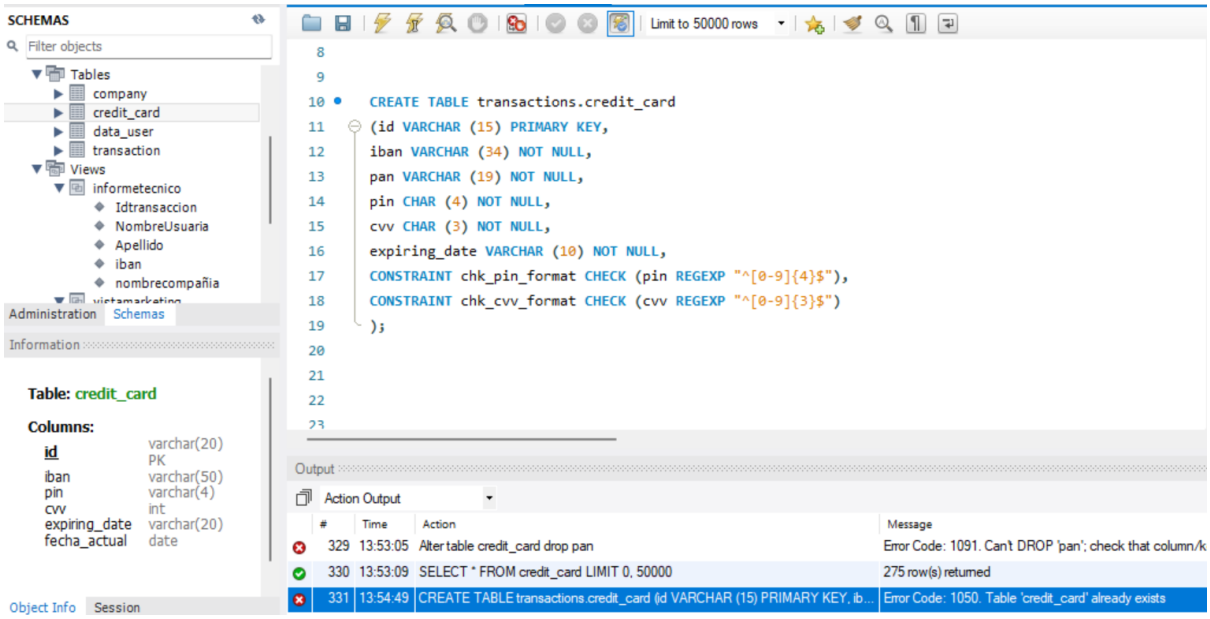
Double click arrows in the list to choose whether to ignore changes, update the model with database changes or vice-versa. You can also apply an action to multiple selected rows.

Model	Update	Source
transactions		transactions
company	→	company
credit_card	→	credit_card
transaction	→	transaction
data_user	→	user
informetecninco		informetecninco
vistamarketing		vistamarketing

Y me muestra el código que aplicara para realizar las consultas:

Preview Database Changes to be Applied

```
ALTER TABLE `transactions`.`company`
DROP COLUMN `website`;

ALTER TABLE `transactions`.`credit_card`
ADD COLUMN `fecha_actual` DATE NULL DEFAULT NULL AFTER `expiring_date`,
CHANGE COLUMN `id` `id` VARCHAR(20) NOT NULL ,
CHANGE COLUMN `iban` `iban` VARCHAR(50) NOT NULL ,
CHANGE COLUMN `pin` `pin` VARCHAR(4) NOT NULL ,
CHANGE COLUMN `cvv` `cvv` INT(11) NOT NULL ,
CHANGE COLUMN `expiring_date` `expiring_date` VARCHAR(20) NOT NULL ;

ALTER TABLE `transactions`.`transaction`
ADD INDEX `transaction_ibfk_2_idx` (`credit_card_id` ASC) VISIBLE,
ADD INDEX `transaction_ibfk_3_idx` (`user_id` ASC) VISIBLE;
;

ALTER TABLE `transactions`.`user`
CHANGE COLUMN `email` `personal_email` VARCHAR(150) NULL DEFAULT NULL , RENAME TO `transactions`.`data_user` ;

ALTER TABLE `transactions`.`transaction`
ADD CONSTRAINT `transaction_ibfk_2`
  FOREIGN KEY (`credit_card_id`)
  REFERENCES `transactions`.`credit_card` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `transaction_ibfk_3`
  FOREIGN KEY (`user_id`)
  REFERENCES `transactions`.`data_user` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;

```



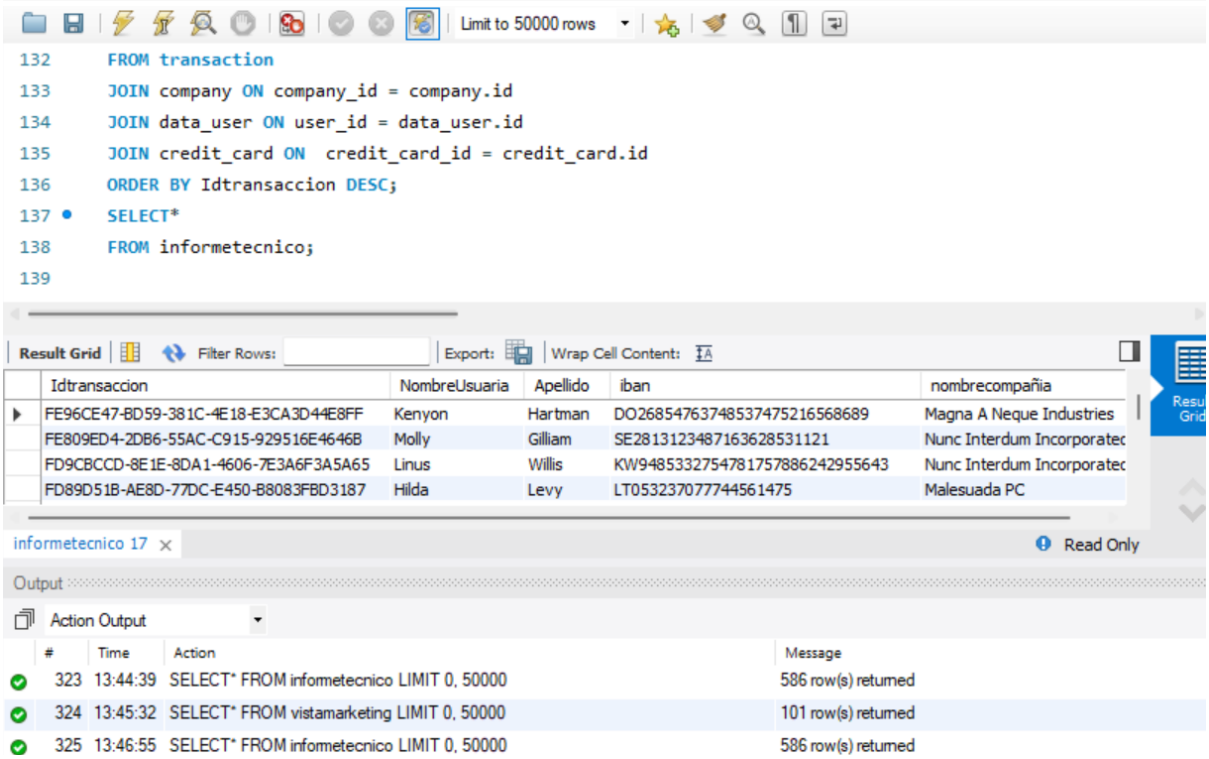
Aqui puedo ver que se han aplicado los cambiosa la tabla existente de credit_card teniendo en cuenta los datatypes indicados en este ejercicio.

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.



Aqui se visualizan todos los datos que deben incluirse en el InformeTecnico, además de que se le asignan nombres más explicativos. Luego se crea una nueva Vista.