

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.

En primer lugar he creado la tabla user utilizando el archivo estructura_datos_user de los recursos.

Luego he cargado en dicha tabla los valores que nos han dado en el archivo datos_introducir_user.

Posteriormente he creado la tabla credit_card:

- `CREATE INDEX idx_credit_card_id ON transaction(credit_card_id);`

- `CREATE TABLE credit_card_1(
 `id` VARCHAR(15) PRIMARY KEY,
 `iban` VARCHAR(100) NOT NULL,
 `pan` VARCHAR(255) NOT NULL,
 `pin` INT NOT NULL,
 `cvv` INT NOT NULL,
 `expiring_date` VARCHAR(255) NOT NULL,
 FOREIGN KEY(id) REFERENCES transaction(credit_card_id)
);`

Finalmente he importado todos los datos que figuraban en el archivo datos_introducir_credit agregándole al comienzo la primera línea:

SET foreign_key_checks = 0;

Y al final la línea:

SET foreign_key_checks = 1;

Con lo que podemos ver cómo se han cargado los datos:

Result Grid

Filter Rows:

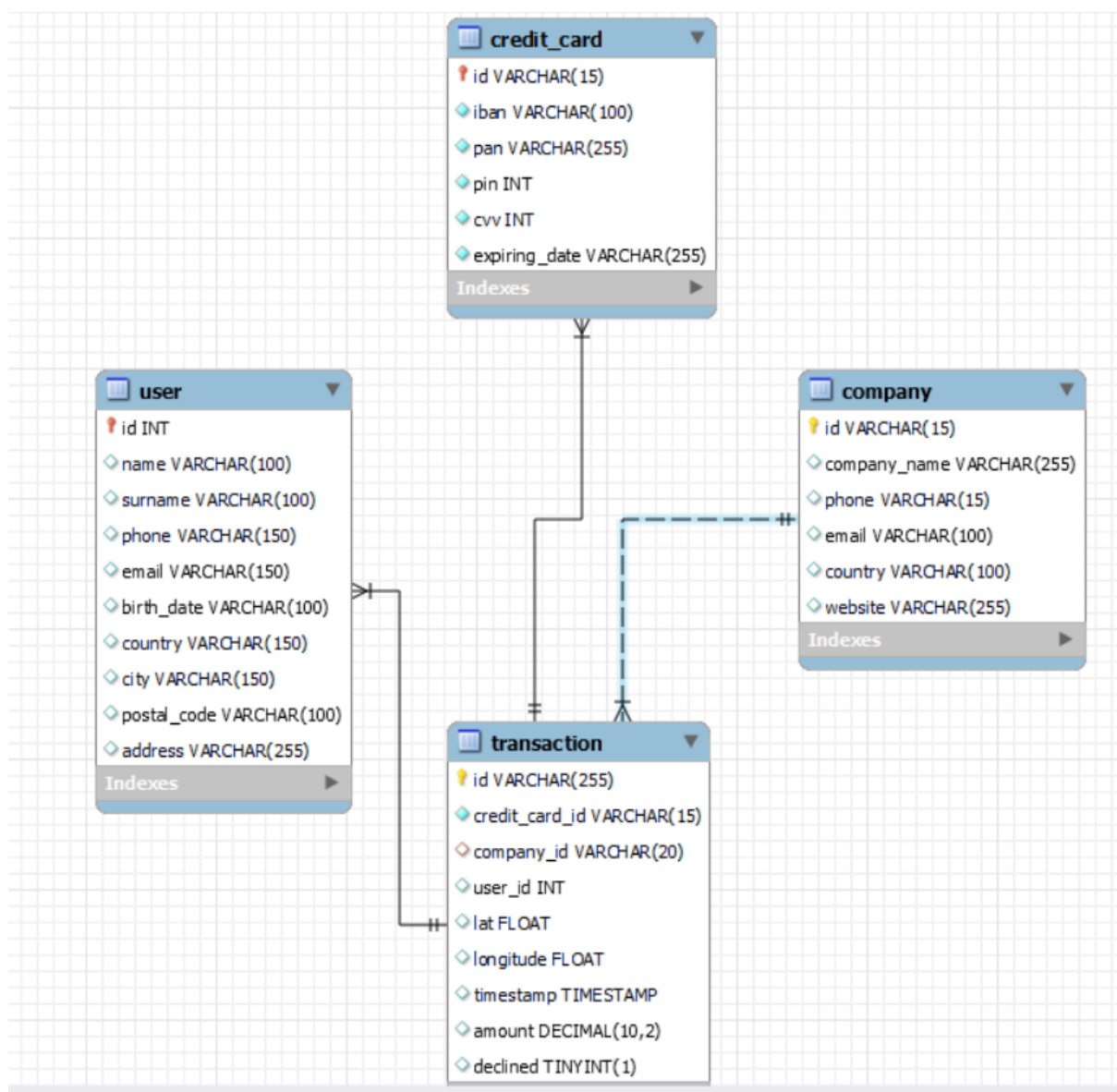
Edit:

Export/Import:

Wrap

	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
	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

Podemos ver el diagrama generado con sus relaciones:




- 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.

Como primer paso realizo la consulta de la cuenta IBAN del usuario con id = Ccu-2938

```
SELECT iban
FROM transactions.credit_card
WHERE id = "Ccu-2938";
```

Confirmando que el número de IBAN de dicho usuario no es el que debería ser:




Result Grid		 Filter Rows:
	iban	
▶	TR301950312213576817638661	

Entonces debemos realizar el cambio en el sistema. Para ello ejecutamos:

```
UPDATE credit_card SET IBAN = "TR323456312213576817699999"
WHERE id = "Ccu-2938";
```

Para confirmar que el dato ha sido modificado:

```
SELECT iban
FROM transactions.credit_card
WHERE id = "Ccu-2938";
```

Result Grid				Filter Rows:	
	iban				
	TR323456312213576817699999				

- 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

Primero quiero aclarar que falta un campo entre los datos que es el "timestamp". He consultado con la profesora y me ha dicho que se les ha olvidado el dato, y que lo deje sin contemplar. Pero para que esté correctamente deberíamos incluir todos los campos que forman parte de la tabla.

Para poder incluir dichos datos, como la tabla transaction está relacionada con otras tablas debemos deshabilitar las conexiones entre ellas.

Luego ejecutar el código para incluir los datos y rehabilitar la restricción para que no puedan realizarse cambios.

```
SET FOREIGN_KEY_CHECKS=0;
INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
  values ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, 111.11, 0);
SET FOREIGN_KEY_CHECKS=1;
```

Hay que tener en cuenta también el tipo de dato a incorporar para que mantenga el formato y lo pueda incluir en la tabla correctamente.

Para confirmar que los datos se hayan cargado correctamente decidí ejecutar esta query:

```
SELECT *
FROM transaction
WHERE credit_card_id = 'CcU-9999';
```

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

Confirmando que se ha cargado correctamente. Cabe aclarar también que en el caso de intentar ejecutar nuevamente la query para insertar esos datos el sistema no nos va a dejar hacerlo porque la primary key ya existe y no nos deja duplicarla pues deben ser valores únicos.

- Exercici 4

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


```
ALTER TABLE credit_card DROP COLUMN pan;
```

Con esto eliminamos la columna de la tabla.

Para confirmar que el cambio se ha realizado ejecutamos la siguiente query:


```
SELECT *  
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
	CcU-2987	GE89681434837748781813	2298	797	10/31/23
	CcU-2994	BH62714428368066765294	7545	595	02/28/22
	CcU-3001	CY49087426654774581266832110	9562	867	09/16/22

Con esto confirmamos que hemos quitado la columna pan.

Nivell 2

Exercici 1

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

Primero prefiero consultar por el registro a eliminar:

```
SELECT *  
FROM transaction  
WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

Result Grid									
Filter Rows:									
Edit:									
Export/Import:									
Wrap Cell Content:									
	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶	02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	92	81.9185	-12.5276	2021-08-28 23:42:24	466.92	0

Entonces procedo a eliminarlo:

```
DELETE FROM transaction  
WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

Confirmo que se haya eliminado:

```
SELECT *  
FROM transaction  
WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

Result Grid									
Filter Rows:									
Edit:									
Export/Import:									
Wra									
	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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 escribo el query que va a traer la información que me pide el sector de Marketing. Para luego con esa query generar la vista.

```

SELECT company.company_name, company.phone, company.country, avg(transaction.amount)
FROM transactions.company
JOIN transactions.transaction
ON company.id = transaction.company_id
GROUP BY company.company_name, company.phone, company.country
ORDER BY avg(transaction.amount) DESC;

```

El resultado que arroja se ve así: (son 100 líneas)

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
company_name	phone	country	avg(transaction.amount)
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
Aliquet Diam Limited	02 76 61 47 46	United States	425.640000
Maecenas Malesuada Fringilla Inc.	09 38 53 76 61	Netherlands	408.620000
Non Ante LLP	08 89 47 65 08	Sweden	407.790000

Entonces creamos la vista:

credit_card
transaction
company
vistamarketing - View
vistamarketing

Name: vistamarketing

The name of the view is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

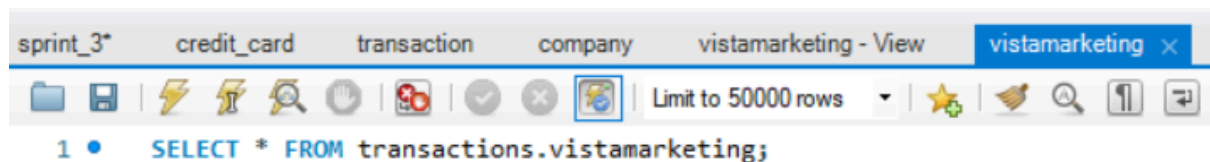
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

```

CREATE
ALGORITHM = UNDEFINED
DEFINER = `root`@`localhost`
SQL SECURITY DEFINER
VIEW `vistamarketing` AS
SELECT
  `company`.`company_name` AS `company_name`,
  `company`.`phone` AS `phone`,
  `company`.`country` AS `country`,
  AVG(`transaction`.`amount`) AS `avg(transaction.amount)`
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

```

Cuando consultamos lo que se ha generado en la vista vemos que es lo mismo que antes:



company_name	phone	country	avg(transaction.amount)
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
Aliquet Diam Limited	02 76 61 47 46	United States	425.640000
Maecenas Malesuada Fringilla Inc.	09 38 53 76 61	Netherlands	408.620000
Non Ante LLP	08 89 47 65 08	Sweden	407.790000

Exercici 3

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

En este caso he modificado la vista para que solo muestre los datos de Alemania. He puesto el renglón de HAVING.

Name: vistamarketing

The name of the view is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:



```
1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5     VIEW `vistamarketing` AS
6     SELECT
7         `company`.`company_name` AS `company_name`,
8         `company`.`phone` AS `phone`,
9         `company`.`country` AS `country`,
10        AVG(`transaction`.`amount`) AS `avg(transaction.amount)`
11    FROM
12        ( `company`
13      JOIN `transaction` ON ((`company`.`id` = `transaction`.`company_id`)))
14    GROUP BY `company`.`company_name` , `company`.`phone` , `company`.`country`
15    HAVING (`company`.`country` = 'Germany')
16    ORDER BY AVG(`transaction`.`amount`) DESC
```

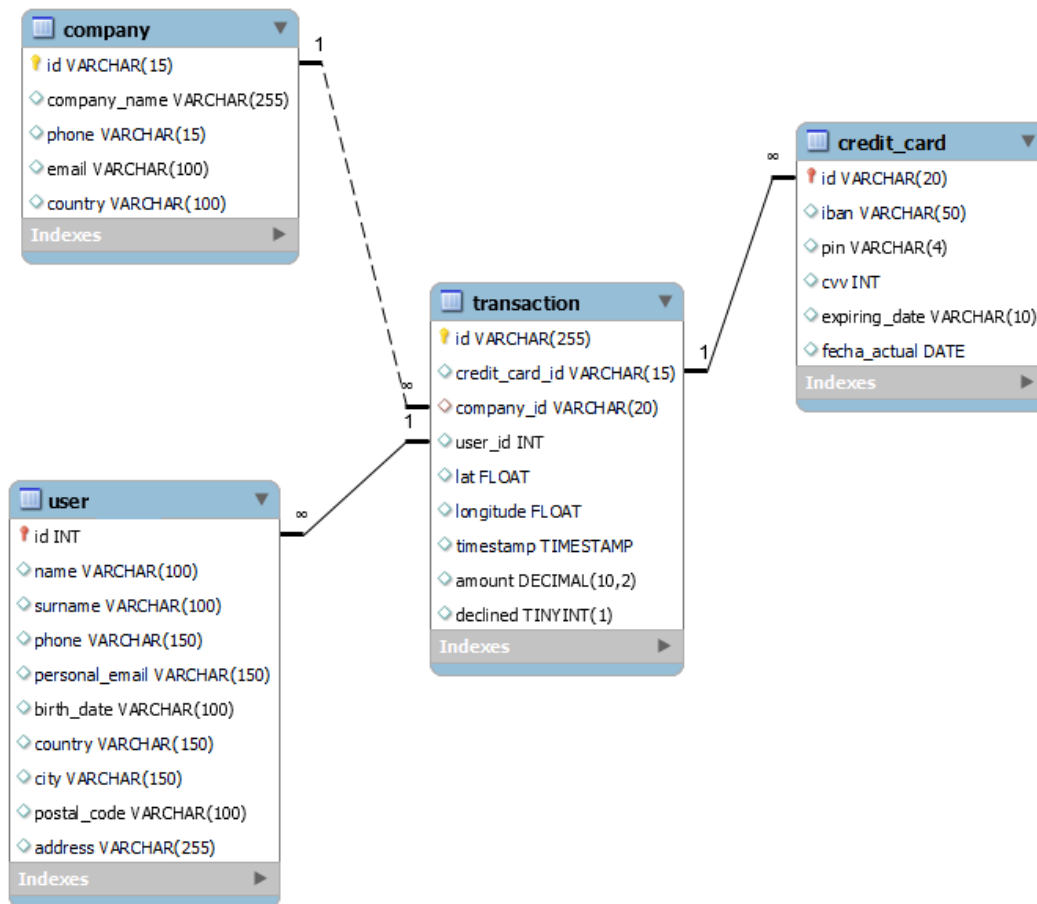
sprint_3*	credit_card	transaction	company	vistamarketing - View	vistamarketing x
<div> </div> <div>Limit to 50000 rows</div> <div> </div>					
1 • <code>SELECT * FROM transactions.vistamarketing;</code>					
<div> <div>Result Grid</div> <div> Filter Rows: <input type="text"/> </div> <div> Export: Wrap Cell Content: </div> </div>					
	company_name	phone	country	avg(transaction.amount)	
▶	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	Germany	184.310000	
	Convallis In Incorporated	06 66 57 29 50	Germany	156.730000	

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):



Recordatori

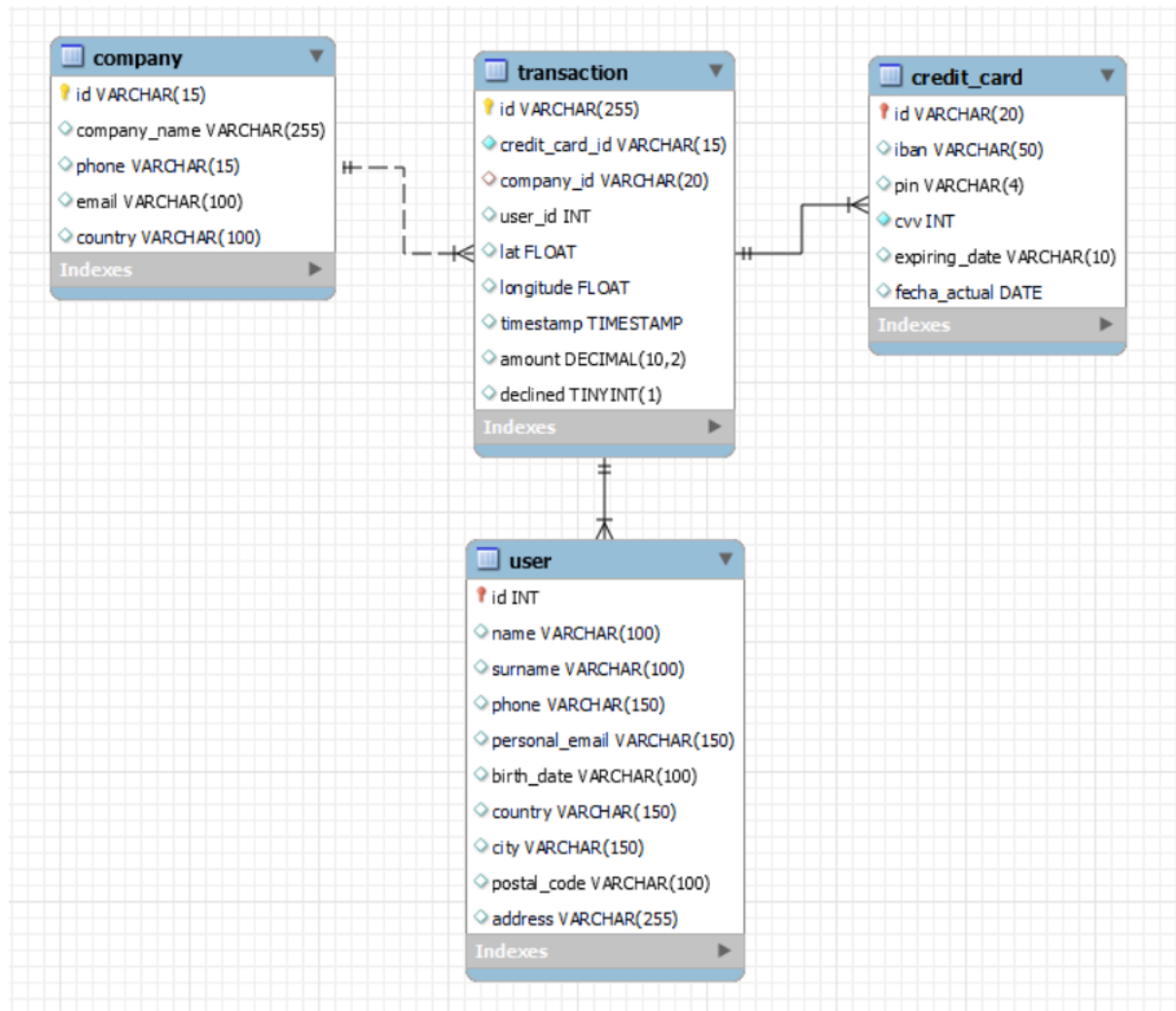
En aquesta activitat, és necessari que descriguis el "pas a pas" de les tasques realitzades. És important realitzar descripcions senzilles, simples i fàcils de comprendre. Per a realitzar aquesta activitat hauràs de treballar amb els arxius denominats "estructura_dades_user" i "dades_introduir_user"

Los comandos que habría que ejecutar son los siguientes:

```
ALTER TABLE user change email personal_email VARCHAR(150);
ALTER TABLE credit_card add column fecha_actual DATE;
ALTER TABLE company DROP column website;
ALTER TABLE credit_card MODIFY id VARCHAR(20);
ALTER TABLE credit_card MODIFY iban VARCHAR(50);
ALTER TABLE credit_card MODIFY pin VARCHAR(4);
ALTER TABLE credit_card MODIFY expiring_date VARCHAR(10);
```

[illegible][illegible][illegible]

Quedaría así



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.

Primero creo la query que voy a usar para esa vista:

```

SELECT transaction.id AS transaction_id, user.name AS user_name, user.surname AS user_surname, credit_card.iban, company.company_name
FROM transactions.transaction
JOIN transactions.company
ON transaction.company_id = company.id
JOIN transactions.user
ON transaction.user_id = user.id
JOIN transactions.credit_card
ON transaction.credit_card_id = credit_card.id
GROUP BY transaction.id, user.id, user.name, user.surname, credit_card.iban, company.company_name
ORDER BY transaction_id DESC;

```

Compruebo que funcione:

transaction_id	user_name	user_surname	iban	company_name
FE96CE47-8D59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCE2AB9A-271D-2BDC-9E49-8DD92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated
FBD7E0D6-BA6B-F5BC-0CA9-EA4B876D100C	Hedwig	Gilbert	MU4132333444534342541344788855	Mauris Id Inc.
FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF58868200575771634583813	Arcu LLP
FAAD3FFC-1A17-E141-43D3-359A5BA7CB3B	Hedwig	Gilbert	GE90157928843338134463	Lorem Eu Incorporated
FA053936-75D8-85FA-490D-9B624E1B920A	Hedwig	Gilbert	GT02497653655330848247645975	Non Justo Corp.

Entonces creamos la vista y se genera esto:

```

CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `informetecnico` AS
  SELECT
    `transaction`.`id` AS `transaction_id`,
    `user`.`name` AS `user_name`,
    `user`.`surname` AS `user_surname`,
    `credit_card`.`iban` AS `iban`,
    `company`.`company_name` AS `company_name`
  FROM
    (((`transaction`
    JOIN `company` ON ((`transaction`.`company_id` = `company`.`id`)))
    JOIN `user` ON ((`transaction`.`user_id` = `user`.`id`)))
    JOIN `credit_card` ON ((`transaction`.`credit_card_id` = `credit_card`.`id`)))
  GROUP BY `transaction`.`id`, `user`.`id`, `user`.`name`, `user`.`surname`, `credit_card`.`iban`, `company`.`company_name`
  ORDER BY `transaction_id` DESC

```

sprint_3* transaction credit_card company user informetecnico - View informetecnico x

Limit to 50000 rows

1 • SELECT * FROM transactions.informetecnico;

transaction_id	user_name	user_surname	iban	company_name
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCE2AB9A-271D-2BDC-9E49-8DD92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated
FBD7E0D6-8A6B-F5BC-0CA9-EA4B8760100C	Hedwig	Gilbert	MU4132333444534342541344788855	Mauris Id Inc.
FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF58868200575771634583813	Arcu LLP
FAAD3FFC-1A17-E141-43D3-359A5BA7CB3B	Hedwig	Gilbert	GE90157928843338134463	Lorem Eu Incorporated
FA053936-75D8-85FA-490D-9B624E1B920A	Hedwig	Gilbert	GT02497653655330848247645975	Non Justo Corp.

Arroja 586 líneas en el orden solicitado.