Tasca S3.01 Manipulació de taules

Descripció

En aquest sprint, es simula una situació empresarial en la qual has de realitzar diverses manipulacions en les taules de la base de dades. Al seu torn, hauràs de treballar amb índexs i vistes. En aquesta activitat, continuaràs treballant amb la base de dades que conté informació d'una empresa dedicada a la venda de productes en línia. En aquesta tasca, començaràs a treballar amb informació relacionada amb targetes de crèdit.

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.

Creación de la tabla “credit_card”

- Creamos los campos que necesitaremos en la tabla siguiendo la estructura del archivo “dades_introduir_credit”.

- Asignamos el tipo de dato a cada campo condicionado por el tipo de dato que queremos introducir como registro.
- Asignamos el campo “id” como Primary Key de la tabla al tratarse de un campo único y que no puede admitir nulos. Además, nos servirá para relacionarlo con otras tablas de la base de datos.
- Asignamos al campo “iban” una key de tipo Unique al tratarse de un dato único que no admite duplicados (no pueden haber dos números IBAN iguales). De esta forma, si realizamos alguna búsqueda por este campo la consulta será más eficiente al tratarse de un campo indexado.
- Marcamos que el resto de campos puedan admitir nulos por precaución, ya que al no conocer de antemano el formato de los datos que introduciremos podríamos tener algún valor nulo.

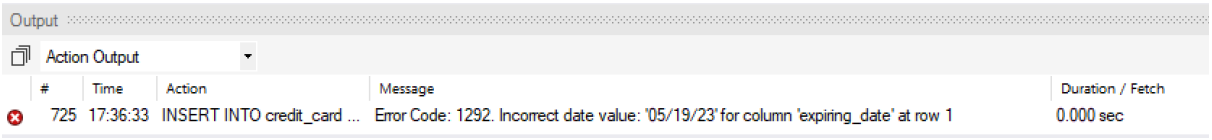
```
CREATE TABLE credit_card (
    id varchar(15) NOT NULL,
  iban varchar(50) NULL,
  pan varchar(100) NULL,
  pin smallint NULL,
  cvv smallint NULL,
  expiring_date date NULL,
  PRIMARY KEY(id),
  UNIQUE(iban)
  );
```

- Comprobamos que la tabla y sus campos se hayan creado correctamente según los parámetros especificados

```
DESCRIBE credit_card;
```

	Field	Type	Null	Key	Default	Extra
▶	id	varchar(15)	NO	PRI	NULL	
	iban	varchar(50)	YES	UNI	NULL	
	pan	varchar(100)	YES		NULL	
	pin	smallint	YES		NULL	
	cvv	smallint	YES		NULL	
	expiring_date	varchar(25)	YES		NULL	

- Introducimos los registros para la tabla “credit_card” y nos aparece el error 1292 indicando que hay algún tipo de problema entre el tipo de dato que admite el campo “expiring_date” y el tipo de dato introducido, copiado del archivo “dades_introduir_credit.



- Compruebo si la tabla devuelve resultados y esta vacía, aunque conserva su estructura.

```
SELECT *
FROM credit_card;
```

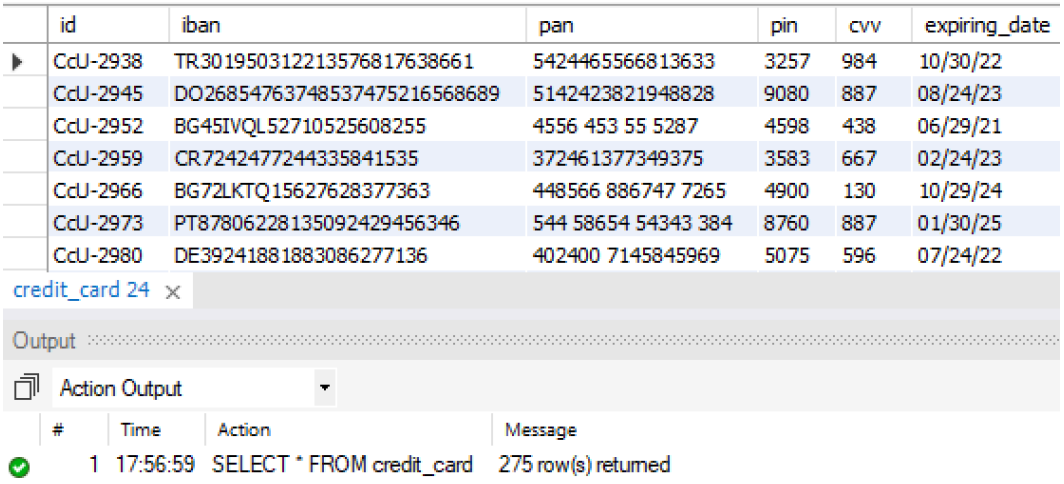
	id	iban	pan	pin	cvv	expiring_date
●	NULL	NULL	NULL	NULL	NULL	NULL

- Cambio el tipo de dato que admite el campo “expiring_date” a varchar, ya que los datos de esa columna no parecen respetar el formato de un tipo de

dato date (YYYY-MM-DD) en MySQL.

```
ALTER TABLE credit_card
MODIFY COLUMN expiring_date varchar(25) NULL;
```

- A continuación introducimos los datos de nuevo y esta vez se introducen sin devolver ningún error.
- Hacemos una consulta para obtener todos los resultados de la tabla y obtenemos 275 registros, que coinciden con el número de registros disponibles en el archivo “dades_introduir_credit”.



id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR320950312213576817638661	5424465566813633	3257	984	10/30/22
CcU-2945	DO28547637485374752165686 89	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

credit_card 24

Output

Action Output

#	Time	Action	Message
1	17:56:59	SELECT * FROM credit_card	275 row(s) returned

Relación entre las tablas “credit_card” y “transaction”

- Relacionamos ambas tablas de forma que el campo “credit_card_id” de la tabla “transaction”, será el FK del campo “id” de la tabla “credit_card”,Identificamos a esa constraint como “fk_transaction_credit”.
- Se trata de una relación 1 a N, donde un único registro en la tabla “credit_card” puede tener múltiples registros en la tabla “transaction”.

```
ALTER TABLE transaction
ADD CONSTRAINT fk_transaction_credit
FOREIGN KEY(credit_card_id) REFERENCES credit_card(id);
```

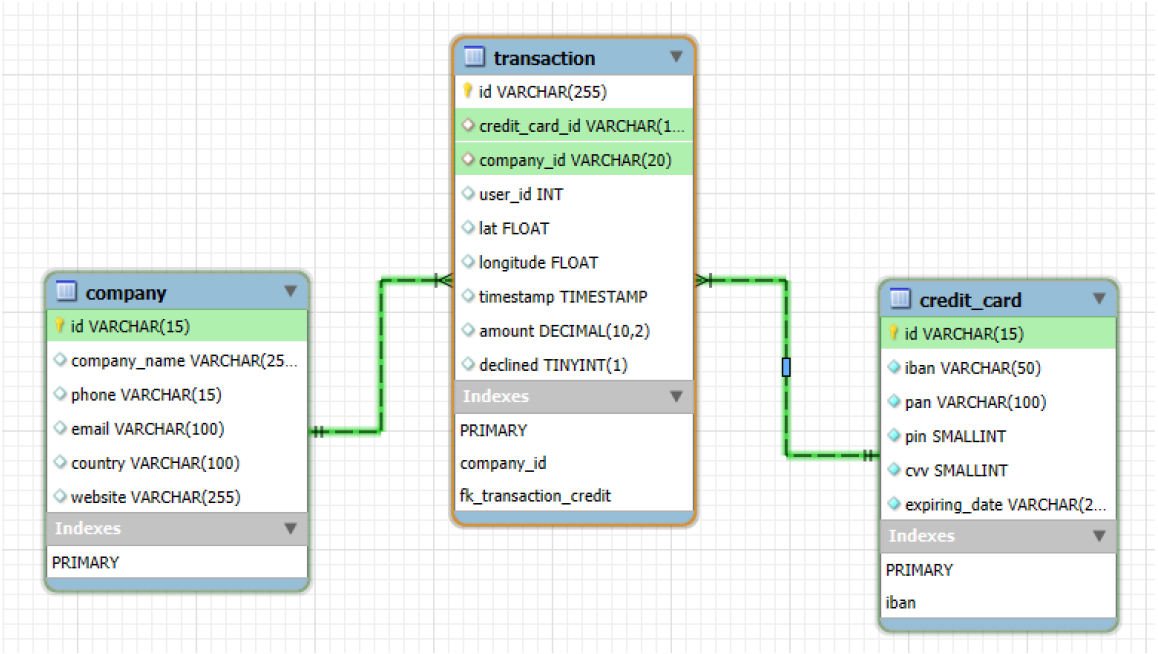
Estructura final de la tabla “credit_card”

La tabla “credit_card” consta de 6 campos:

Columna	Tipo de dato	Descripción
id	varchar(15) (PK)	Identificador de la tarjeta de crédito. Primary Key que se relacionará con la Foreign Key “credit_card_id” de la tabla “transaction”. Es de tipo varchar al combinar números y otros carácteres, como la FK a la que hará referencia.
iban	varchar(50) (UNIQUE)	Número único que combina el país de origen de la cuenta, la entidad bancaria y el número de la propia cuenta. Al ser un número único, le he asignado una KEY UNIQUE para optimizar las consultas mediante este campo.
pan	varchar(100)	Número que identifica al emisor de la tarjeta y a la cuenta bancaria relacionada. Podría funcionar como KEY UNIQUE, pero he considerado que tendría más sentido asignar esa key únicamente al campo “iban”.
pin	smallint	Código de seguridad numérico de la tarjeta de 4 dígitos. Le he asignado un valor de dato smallint al aceptar valores absolutos entre 0 y 65535.
ccv	smallint	Código de seguridad numérico de 3 dígitos para transacciones de forma online o telemática. También es smallint.
expiring_date	varchar(25)	Fecha de vencimiento de la tarjeta de crédito. Aunque quizá debería ser un valor de dato date, es varchar porque así estan formateados los registros introducidos para ese campo.

Diagrama de entidades y tipo de relación de “credit_card” con el resto de tablas

- El diagrama de entidades muestra una relación de 1 a N entre “credit_card” y “transaction”, en la cual un registro en “credit_card” puede tener más de un registro en “transaction”.
- La tabla “company” y “transaction” también tienen una relación de 1 a N.
- La tablas “credit_card” y “company” se relacionan entre ellas a través de la tabla “transaction”.
- La tabla “transaction” es la tabla de hechos, mientras que “company” y “credit_card” son las tablas de dimensiones.



Exercici 2

El departament de Recursos Humans ha identificat un error en el número de compte de l'usuari

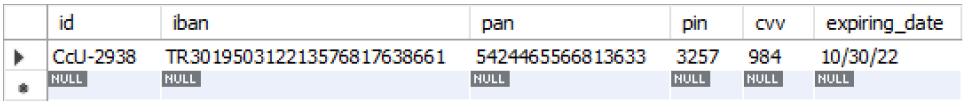
amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és:

- **R323456312213576817699999**

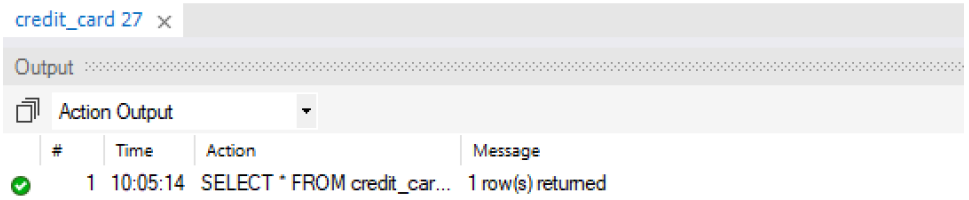
Recorda mostrar el canvi que es va realitzar.

- Solicitamos todos los datos disponibles dela usuario sobre el cual hay que modificar algún registro para identificarlo.

```
SELECT *
FROM credit_card
WHERE id  = "CcU-2938";
```



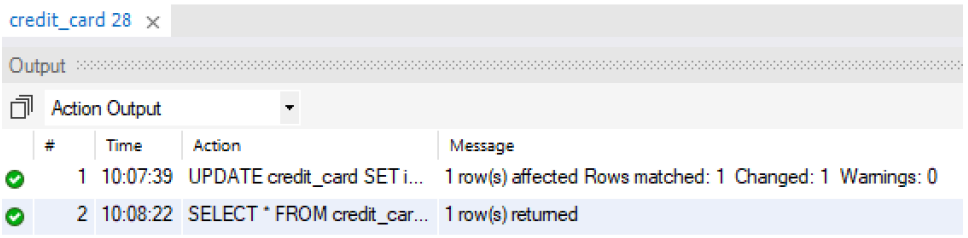
id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22
NULL	NULL	NULL	NULL	NULL	NULL



- Introducimos los parámetros para cambiar el campo "iban" con la información facilitada
- Comprobamos que la actualización del registro se ha realizado correctamente

```
UPDATE credit_card
SET iban = "R323456312213576817699999"
WHERE id = "CcU-2938";
```

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



Exercici 3

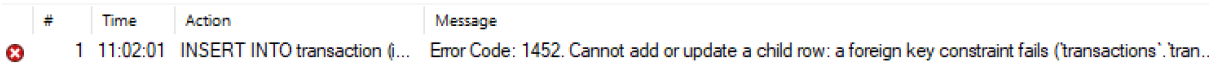
En la taula "transaction" ingressa un nou usuari amb la següent informació:

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

- Añadimos el registro en la tabla “transaction”

```
INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitud
VALUES ("108B1D1D-5B23-A76C-55EF-C568E49A99DD", "CcU-9999", "b-99
```

- Nos devuelve el error 1452, indicando que no ha podido realizar la actualización debido a un fallo en la relación entre la tabla “transaction” y “company”

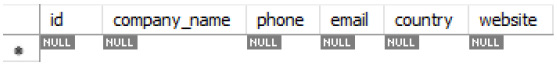


#	Time	Action	Message
1	11:02:01	INSERT INTO transaction (i...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`transactions`.`tran..

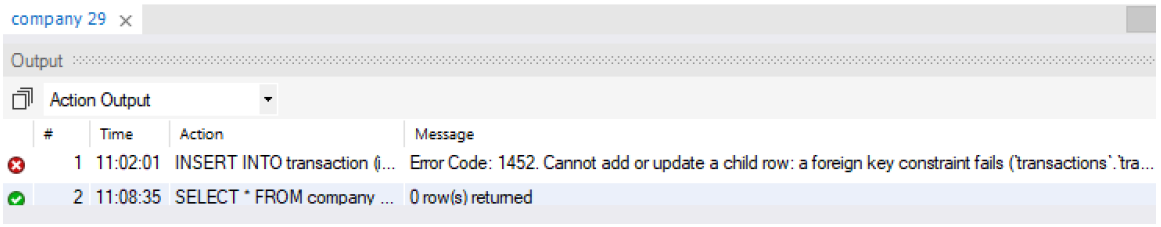
- Ambas tablas están relacionados por el PK “id” de la tabla “company”, que se relaciona con la FK “company_id” de la tabla “transaction”
- Comprobamos que el identificador de la empresa que hemos de introducir existe en la tabla “company” ya que funciona a modo de directorio de todas las empresas que pueden tener o no transacciones registradas en “transaction”.

```
SELECT *
FROM company
WHERE id = b-9999;
```

- No obtenemos ningún resultado, por lo que entendemos que la empresa con id “b-9999” no existe



id	company_name	phone	email	country	website
NULL	NULL	NULL	NULL	NULL	NULL



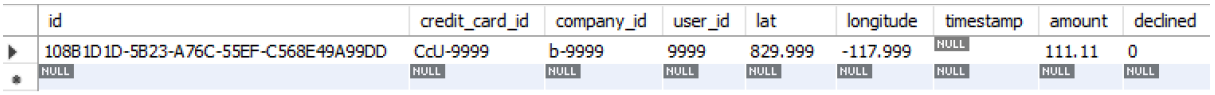
- Procedemos a dar de alta a esa empresa en “company”, pero como únicamente sabemos su identificador solamente registraremos ese campo, manteniendo el resto en nulos.
- A continuación, añadiremos la tarjeta de crédito con la que se ha realizado la transacción a nuestra tabla de dimensiones “credit_card”, ya que tampoco está registrada aún.

```
-- Alta de la empresa en "company"
SELECT INTO company (id)
VALUES ("b-9999");

-- Alta del identificador de la tarjeta de crédito en "credit_card"
SELECT INTO credit_card (id)
VALUES ("CcU-9999");
```

- Finalmente, podemos añadir el nuevo registro a “transaction” sin obtener ningún error.
- Realizamos una búsqueda por el identificador de la transacción para comprobar que se ha registrado correctamente.

```
SELECT *
FROM transaction
WHERE id = "108B1D1D-5B23-A76C-55EF-C568E49A99DD";
```



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

Exercici 4

Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card.

Recorda mostrar el canvi realitzat.

- Eliminamos el campo “pan”

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

- Confirmamos que la columna se ha eliminado correctamente

```
SELECT *
FROM credit_card;
```

	id	iban	pin	cvv	expiring_date
▶	CcU-2938	R323456312213576817699999	3257	984	10/30/22
	CcU-2945	DO268547637485374752165686 89	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

credit_card 38 ×

Output

Action Output

#	Time	Action	Message
2	12:02:57	ALTER TABLE credit_card...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3	12:03:12	SELECT * FROM credit_card	276 row(s) returned

NIVELL 2

Exercici 1

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

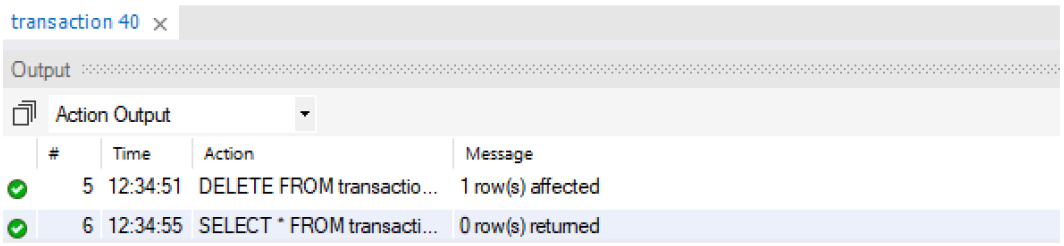
- Eliminamos el registro filtrando el id mediante WHERE

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

- Comprobamos que ese registro se ha eliminado correctamente y observamos que no nos devuelve ningún resultado.

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

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



transaction 40

Output

Action Output

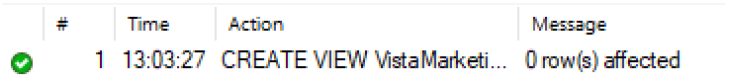
#	Time	Action	Message
5	12:34:51	DELETE FROM transactio...	1 row(s) affected
6	12:34:55	SELECT * FROM transacti...	0 row(s) returned

Exercici 2

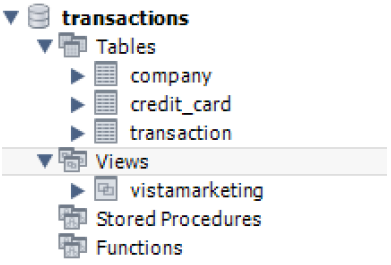
La secció de màrqueting desitja tenir accés a informació específica per a realitzar anàlisi i estratègies efectives. S'ha sol·licitat crear una vista que proporcioni detalls clau sobre les companyies i les seves transaccions. Serà necessària que creïs una vista anomenada VistaMarketing que contingui 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.

```
CREATE VIEW VistaMarketing AS
SELECT
   company_name,
  phone,
  country,
  ROUND(AVG(amount), 2) AS avg_amount
FROM company AS c
```

```
JOIN transaction AS t
    ON  c.id = t.company_id
WHERE declined = 0                    -- Filtramos por las transacciones no declina
GROUP BY company_name, phone, country;
```



#	Time	Action	Message
1	13:03:27	CREATE VIEW VistaMarketi...	0 row(s) affected



- Comprobamos que podemos usar la vista “vistamarketing”

```
SELECT *
FROM vistamarketing
ORDER BY avg_amount DESC;
```

company_name	phone	country	avg_amount
Eget Ipsum Ltd	03 67 44 56 72	United States	481.86
Sed Id Limited	07 28 18 18 13	United States	477.51
Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.10
Nunc Sit Incorporated	07 28 42 63 63	Norway	461.83
Non Magna LLC	06 71 73 13 17	United Kingdom	458.74

Result 54

Output

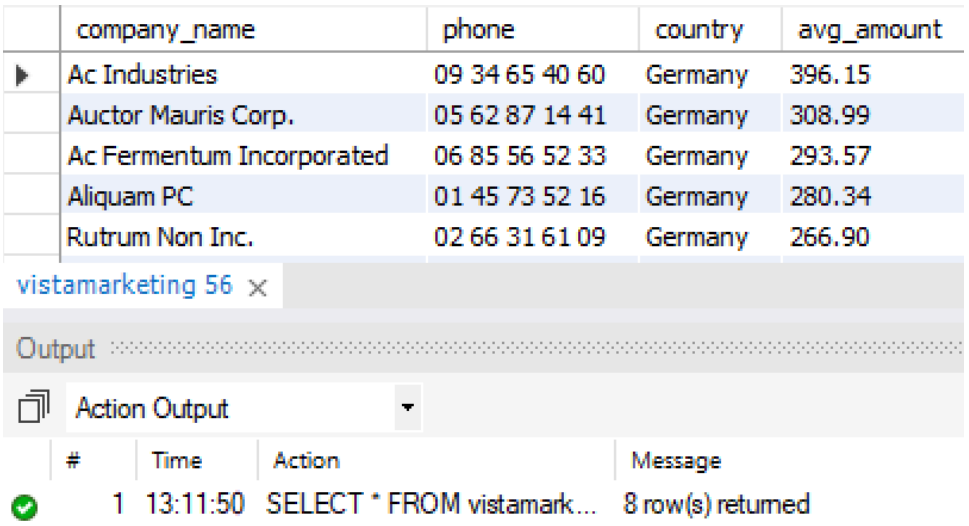
Action Output

#	Time	Action	Message
2	13:08:31	SELECT * FROM vistamar...	101 row(s) returned
3	13:08:54	SELECT company_name, ...	101 row(s) returned

Exercici 3

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

```
SELECT *
FROM vistamarketing
WHERE country = "Germany";
```

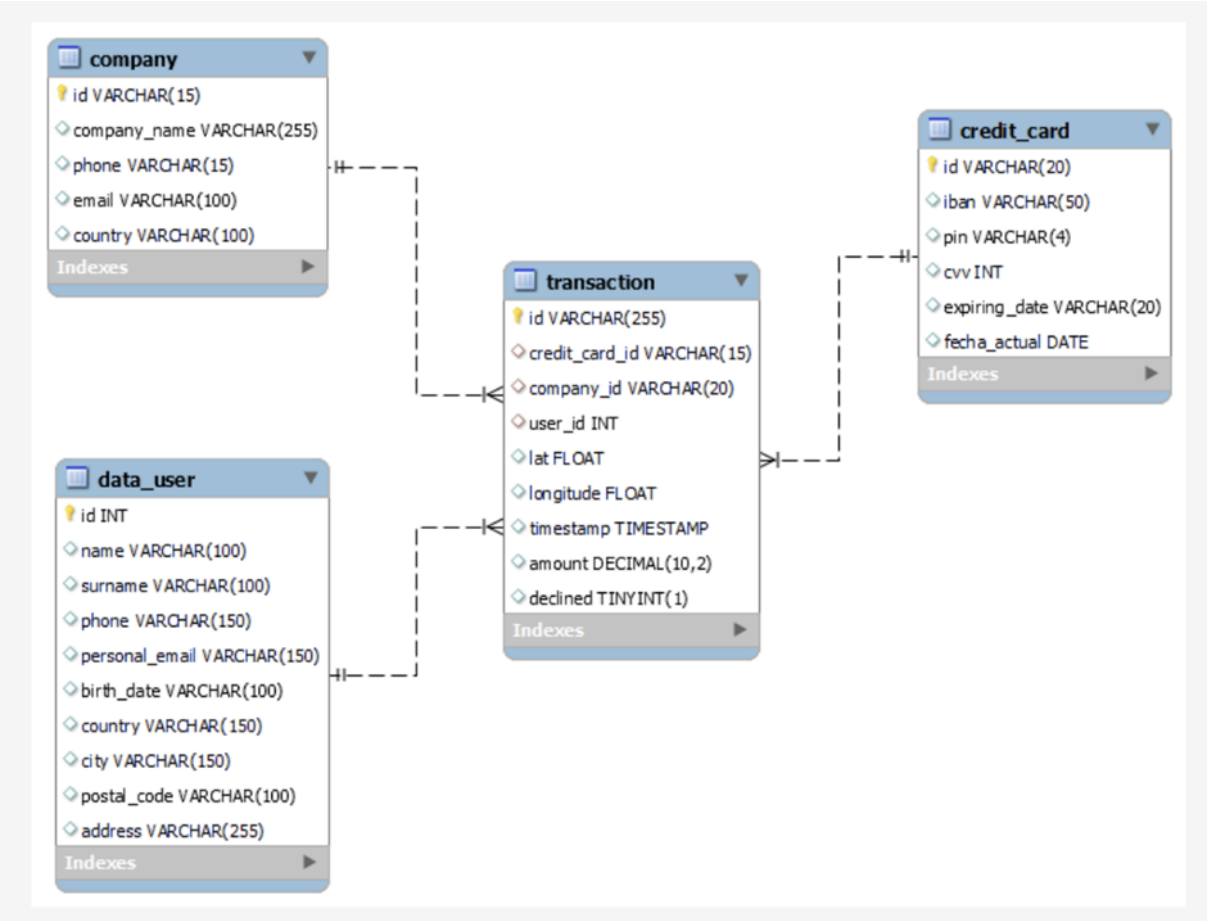


company_name	phone	country	avg_amount
Ac Industries	09 34 65 40 60	Germany	396.15
Auctor Mauris Corp.	05 62 87 14 41	Germany	308.99
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.57
Aliquam PC	01 45 73 52 16	Germany	280.34
Rutrum Non Inc.	02 66 31 61 09	Germany	266.90

NIVELL 3

Exercici 1

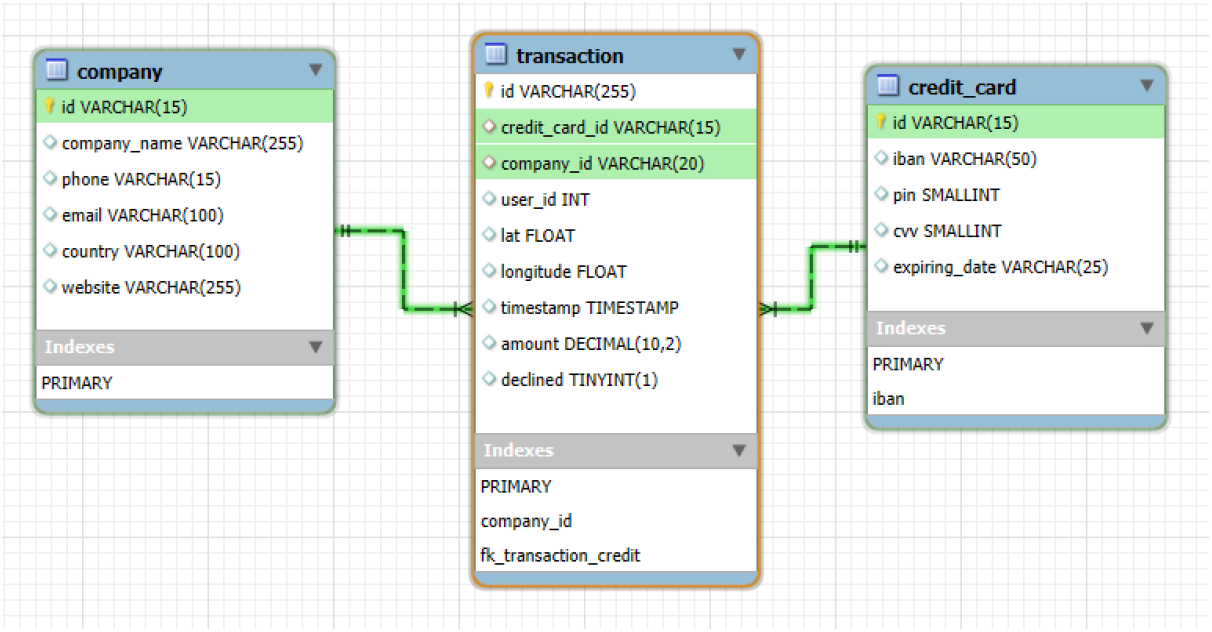
La setmana vinent tindràs una nova reunió amb els gerents de màrqueting. Un company del teu equip va realitzar modificacions en la base de dades, però no recorda com les va realitzar. Et demana que l'ajudis a deixar els comandos executats per a obtenir el següent diagrama:



Recordatori

En aquesta activitat, és necessari que 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"

Diagrama de entidades actual de TRANSACTIONS



Cambios a realizar:

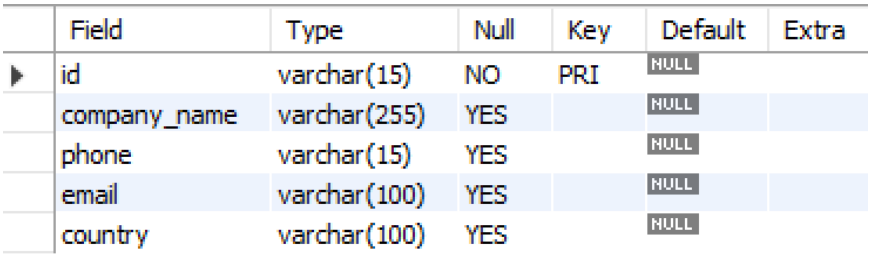
- Tabla “company”
  - Eliminar campo “website”
- Tabla “credit_card”
  - Modificar longitud de varchar en campo “id”
  - Modificar tipo de dato en campo “pin”
  - Modificar tipo de dato en campo cvv
  - Modificar tipo de dato en campo “expiring_date”
  - Agregar campo “fecha_actual” como date
- Crear tabla “data_user” e introducir registros
  - Cambiar nombre campo “email”
  - Crear relación con “transaction”

Modificación de la tabla “company”

- Eliminamos el campo “website” y comprobamos que se ha eliminado correctamente

```
ALTER TABLE company
DROP website;

DESCRIBE company;     -- Comprobamos que la columnna se ha eliminado
```



Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	NULL	
company_name	varchar(255)	YES		NULL	
phone	varchar(15)	YES		NULL	
email	varchar(100)	YES		NULL	
country	varchar(100)	YES		NULL	

Modificación de la tabla “credit_card”

- Modificar longitud de varchar en campo “id” para que sea varchar(20). Seguirá siendo NOT NULL porque es la Primary Key de la tabla.

```
ALTER TABLE credit_card
MODIFY COLUMN id varchar(20) NOT NULL;
```

- Modificamos el tipo de dato en campo “pin”. Cambiamos a varchar(4) e indicamos que siga admitiendo nulos.

```
ALTER TABLE credit_card
MODIFY COLUMN pin varchar(4) NULL;
```

- Modificamos tipo de dato en campo “cvv” a integer.

```
ALTER TABLE credit_card
MODIFY COLUMN cvv int NULL;
```

- Modificamos el campo “expiring_date” para que sea de tipo varchar(20)

```
ALTER TABLE credit_card
MODIFY COLUMN expiring_date varchar(20) NULL;
```

- Agregamos el campo “fecha_actual” como date. Indicamos que puede admitir valores NULL.

```
ALTER TABLE credit_card
ADD fecha_actual date NULL;
```

- Comprobamos como queda la tabla después de realizar los cambios.

```
DESCRIBE credit_card;
```

	Field	Type	Null	Key	Default	Extra
▶	id	varchar(20)	NO	PRI	NULL	
	iban	varchar(50)	YES	UNI	NULL	
	pin	varchar(4)	YES		NULL	
	cvv	int	YES		NULL	
	expiring_date	varchar(20)	YES		NULL	
	fecha_actual	date	YES		NULL	

Creación de la tabla “data_user” e introducción de los registros

- Seleccionamos los datos del documento “estructura_datos_user” y introducimos el código facilitado.
- A continuación, introducimos los registros de “datos_introducir_user”.

```
/* Convertimos a INDEX el campo user_id de la tabla
transactions para realizar la relación entre tablas*/

CREATE INDEX idx_user_id
ON transaction(user_id);

-- Creamos la tabla
CREATE TABLE IF NOT EXISTS user (
	id INT PRIMARY KEY,
	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),
    FOREIGN KEY(id) REFERENCES transaction(user_id)
);
```

- Cambiamos el nombre de la tabla para que coincida con el diagrama facilitado.

```
ALTER TABLE user
RENAME TO data_user;
```

- Cambiar el nombre del campo “email” a “personal_email”.

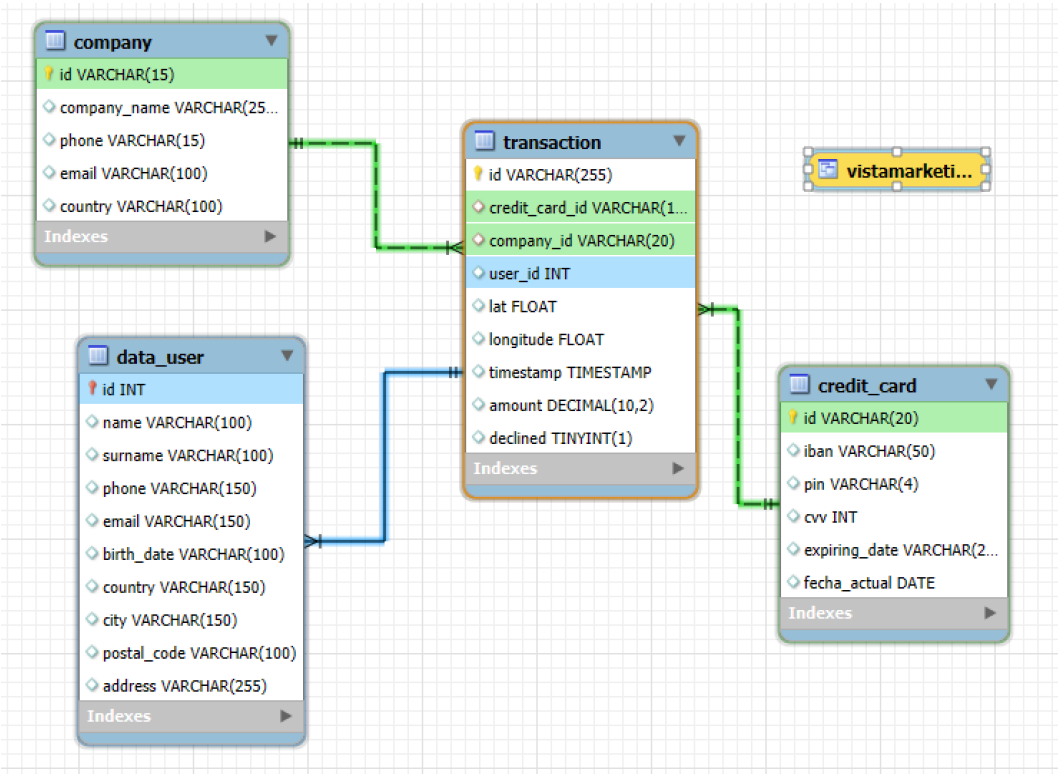
```
ALTER TABLE data_user
RENAME COLUMN email TO personal_email;
```

- Comprobamos que la tabla se ha creado y renombrado de forma adecuada.

```
DESCRIBE data_user;
```

	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	
	name	varchar(100)	YES		NULL	
	surname	varchar(100)	YES		NULL	
	phone	varchar(150)	YES		NULL	
	email	varchar(150)	YES		NULL	
	birth_date	varchar(100)	YES		NULL	
	country	varchar(150)	YES		NULL	
	city	varchar(150)	YES		NULL	
	postal_code	varchar(100)	YES		NULL	
	address	varchar(255)	YES		NULL	

DIAGRAMA ENTIDADES



Observamos que la relación entre “data_user” y “transaction” no es la correcta. Debería ser una relación de 1 a N, en la cual deberíamos tener 1 registro único

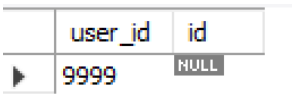
en “data_user” que a su vez podría repetirse múltiples registros en “transaction”.

Sin embargo, en el actual diagrama tenemos una relación N a 1. La relación está invertida y no es la correcta. Deduzco que hay un usuario con registro en la tabla “transactions” que no tiene registro creado en “data_user”, de ahí la relación invertida.

Para realizar la comprobación, hacemos un LEFT JOIN de “transactions” con “data_user”, donde la tabla “transactions” es la tabla principal.

Seleccionaremos los campos “user_id” de “transactions” y “id” de “data_user”, y filtraremos para obtener NULL a través del campo “id”.De esta forma, obtendremos NULL en aquellos campos que estén en “transactions” pero no en “data_user” y podremos observar si falta aún registro en la segunda tabla.

```
SELECT
   t.user_id,
   d.id
FROM transaction AS t
LEFT JOIN data_user AS d
 ON d.id = t.user_id
WHERE d.id IS NULL;
```



Observamos que el campo “id”, que pertenece a “data_user”, nos devuelve un valor nulo, por lo que el “user_id” 9999 de la tabla “transactions” no figura en la otra tabla.

Es el mismo usuario del cual hemos introducido datos de transacción en un ejercicio anterior.

Definir la relación correcta entre tablas

- Eliminamos la actual relación.

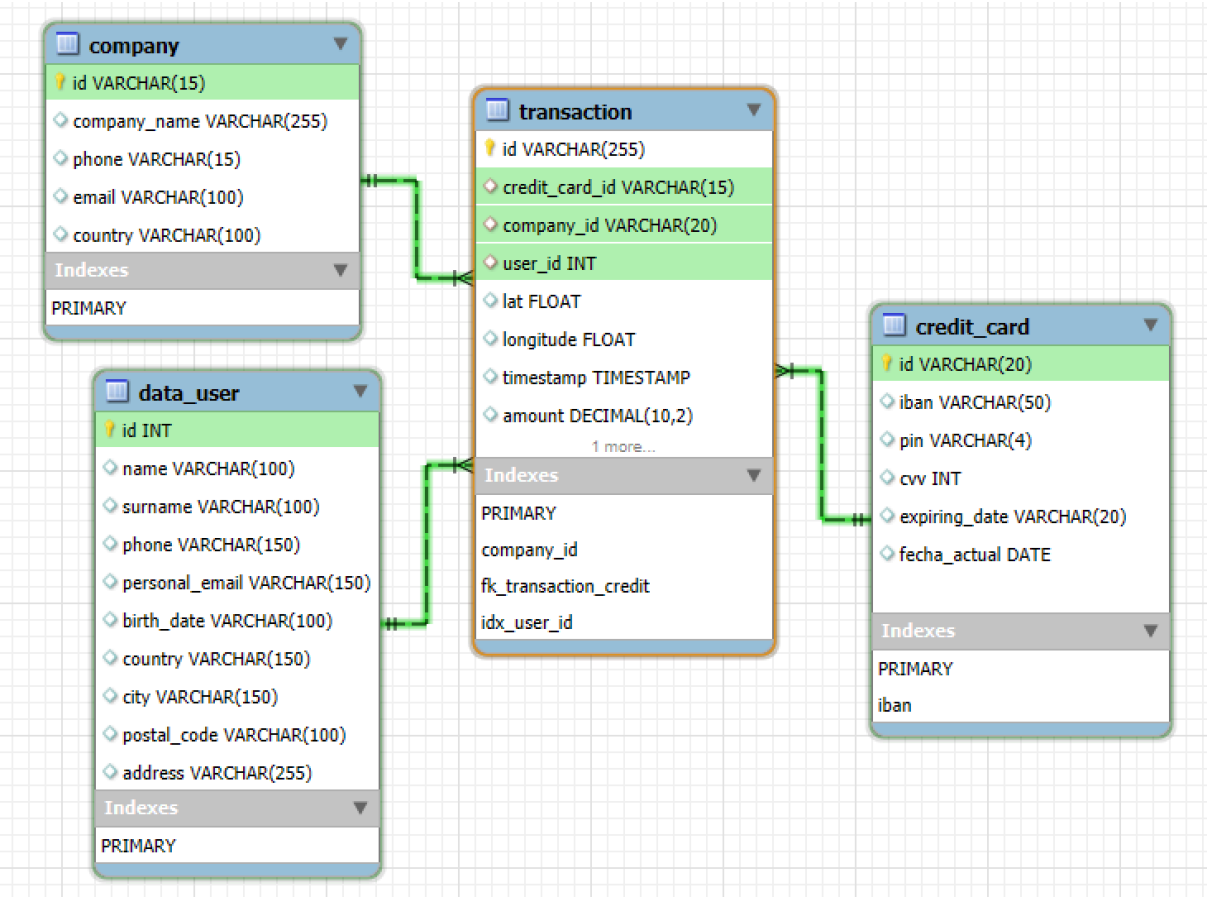
```
ALTER TABLE data_user
DROP FOREIGN KEY data_user_ibfk_1;
```

- Registramos el usuario 9999 en la tabla “data_user” y lo asignamos el campo “id”.

```
INSERT INTO data_user (id)
VALUES ("9999");
```

- Relacionamos “data_user” mediante el campo “id”, PK de esta tabla, que se relaciona con su FK “user_id” de la tabla “transaction”

```
ALTER table transaction
ADD CONSTRAINT fk_datau_transaction
FOREIGN KEY(user_id) REFERENCES data_user(id);
```



El diagrama final nos muestra un esquema en estrella donde “transaction” es la tabla de hechos y “company”, “data_user” y “credit_card” son las tablas de dimensiones, relacionándose todas ellas en una relación de 1 a N con “transaction.

Exercici 2

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

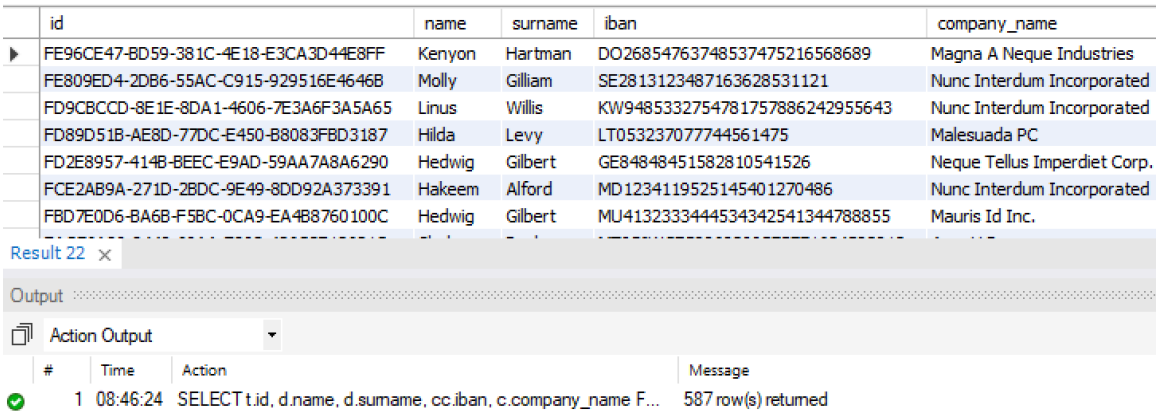
- ID de la transacció
- Nom de l'usuari/ària
- Cognom de l'usuari/ària
- IBAN de la targeta de crèdit usada.
- Nom de la companyia de la transacció realitzada.

- 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.

- Creamos la consulta con los parámetros especificados y probamos que funcione correctamente
- En este caso, al tratarse de un informe técnico con información relevante queremos que figuren todas las transacciones realizadas, hayan sido declinadas o no.

```
SELECT
    t.id,
    d.name,
    d.surname,
    cc.iban,
    c.company_name,
    t.declined
FROM transaction AS t
JOIN data_user AS d
    ON t.user_id = d.id
JOIN company AS c
    ON t.company_id = c.id
JOIN credit_card AS cc
    ON t.credit_card_id = cc.id;
```



id	name	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-BA6B-F5BC-0CA9-EA4B8760100C	Hedwig	Gilbert	MU413233344453434254134478885 5	Mauris Id Inc.

Result 22

Output

Action Output

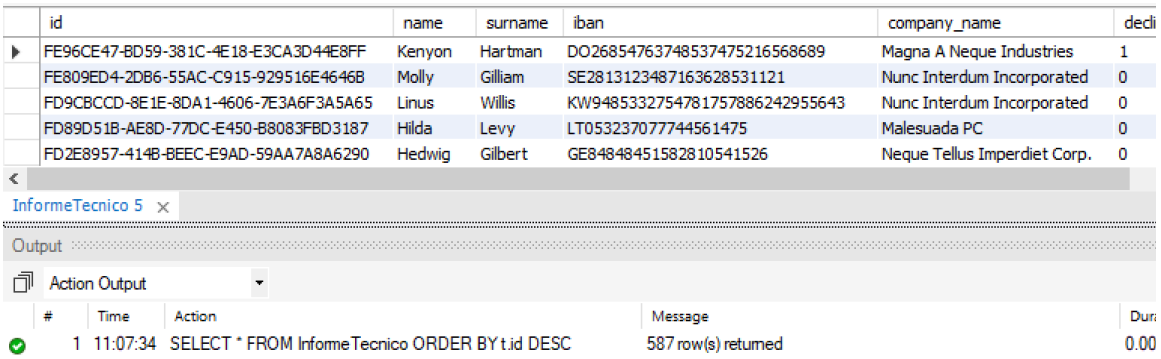
#	Time	Action	Message
1	08:46:24	SELECT t.id, d.name, d.surname, cc.iban, c.company_name F...	587 row(s) returned

- Creamos la vista con CREATE VIEW ... AS

```
CREATE VIEW InformeTecnico AS
SELECT
    t.id,
    d.name,
    d.surname,
    cc.iban,
    c.company_name
FROM transaction AS t
JOIN data_user AS d
    ON t.user_id = d.id
JOIN company AS c
    ON t.company_id = c.id
JOIN credit_card AS cc
    ON t.credit_card_id = cc.id;
```

- Probamos la vista para asegurarnos que nos devuelve los resultados correctos

```
SELECT *
FROM InformeTecnico
ORDER BY t.id DESC;
```



id	name	surname	iban	company_name	decli
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries	1
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE281312348716362853112l	Nunc Interdum Incorporated	0
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated	0
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC	0
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.	0

InformeTecnico 5

Output

Action Output

#	Time	Action	Message	Dur
1	11:07:34	SELECT * FROM InformeTecnico ORDER BY t.id DESC	587 row(s) returned	0.00