Descripció

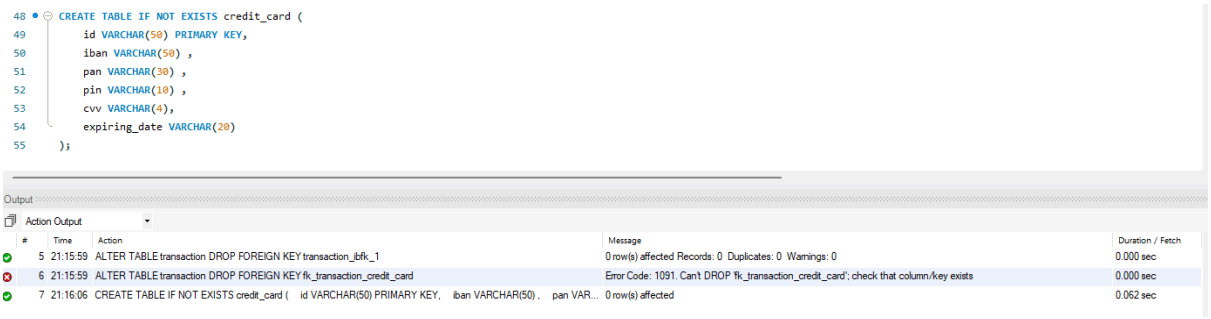
En aquest sprint, es simula una situació empresarial en la qual has de realitzar diverses manipulacions en les taules de la base de dades. Al seu torn, hauràs de treballar amb índexs i vistes. En aquesta activitat, continuaràs treballant amb la base de dades que conté informació d'una empresa dedicada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

- En este caso para crear la tabla **credit_card** usamos la siguiente query, la tabla que creamos tiene la siguientes variables:



id (VARCHAR) es la clave primaria que identifica las tarjetas de crédito utilizadas.

IBAN (VARCHAR) es el número de IBAN perteneciente a cada tarjeta de crédito.

PAN (VARCHAR)

PIN y CVV (VARCHAR) son los números de PIN y CVV de cada tarjeta mientras que **EXPIRING_DATE** es la fecha de vencimiento de cada una. Luego cargamos la información

desde **‘dades_introduir_credit’**

```
-- Insertamos datos de credit_card
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-2938', 'TR301950312213576817638661', '5424465566813633', '3257', '984', '10/30/22');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-2945', 'DO28547637485374752216568689', '5142423821948828', '9080', '887', '08/24/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-2952', 'BG45IVQL52710525608255', '4556 453 55 5287', '4598', '438', '06/29/21');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-2959', 'CR7242477244335841535', '372461377349375', '3583', '667', '02/24/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-2966', 'BG72LKTQ15627628377363', '448566 886747 7265', '4900', '130', '10/29/24');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-2973', 'PT8780622813509242945634 6', '544 58654 54343 384', '8760', '887', '01/30/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-2980', 'DE39241881883086277136', '402400 7145845969', '5075', '596', '07/24/22');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-2987', 'GE89681434837748781813', '3763 747687 76666', '2298', '797', '10/31/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-2994', 'BH62714428368066765294', '344283273252593', '7545', '595', '02/28/22');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3001', 'CY4908742665477458126683 2110', '511722 924833 2244', '9562', '867', '09/16/22');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3008', 'LU507216693616119230', '4485744464433884', '1856', '740', '04/05/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3015', 'PS1193982162957159683424568 21', '3784 662233 17389', '3246', '822', '01/31/22');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3022', 'GT9169516285055697742312185 7', '5164 1379 4842 3951', '5610', '342', '04/25/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3029', 'AZ623174139824414181237397 46', '3429 279566 77631', '9708', '505', '09/02/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3036', 'AZ393360029258428658439419 94', '3768 451556 48766', '2232', '565', '10/27/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3043', 'TN648814331051485217953 5', '455676 6437463635', '5969', '196', '06/07/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3050', 'FR5167744369175836831854477', '4024007123722', '4834', '126', '10/09/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3057', 'LU9318225746975452 15', '3484 621767 21237', '6805', '848', '09/14/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3064', 'PS1469655454492533776272731 33', '3467 732741 26810', '3865', '498', '06/03/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3071', 'NO8923814763512', '3464 789562 23352', '6625', '661', '12/20/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3078', 'IS0251271458846232795487 33', '4539 322 74 2377', '9405', '720', '03/08/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3085', 'BE63114723972437', '5266 3346 1135 1687', '7241', '413', '05/10/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3092', 'RO65LSOD1166122125447487', '3488 754223 46253', '9417', '594', '12/19/22');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3099', 'PT2610527535682370553721 8', '448 55418 98863 789', '5612', '564', '01/22/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3106', 'AT684251637751136592', '349547146395283', '9733', '209', '01/27/24');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3113', 'IE26LCGT47732173572752', '341834822877471', '9011', '287', '06/12/21');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3120', 'RS7265576666616623714 4', '527646 533375 6577', '7658', '265', '01/16/21');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3127', 'PT8353346143864434281686 4', '4716 443 46 4368', '8038', '924', '01/16/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3134', 'BG23MYJQ52668951824779', '5146 3453 9766 2168', '7260', '935', '08/24/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-3141', 'CH4437804777669672438', '3775 626726 45261', '2923', '330', '05/11/24');
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
280	21:17:29	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4842', 'SA2156708581957118818...	1 row(s) affected	0.000 sec
281	21:17:29	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4849', 'SE2813123487163628531...	1 row(s) affected	0.000 sec
282	21:17:29	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4856', 'TR373872558313545667...	1 row(s) affected	0.000 sec

y así ya tenemos la tabla con los datos correspondientes, y ejecutamos el siguiente comando para crear la FK que relaciona la tabla que creamos con la tabla ya existente **transaction:**

```
-- EJECUTAMOS ESTE CODIGO PARA ESTABLECER LA RELACION CON LA TABLA TRANSACTION

ALTER TABLE transaction
ADD CONSTRAINT fk_transaction_credit_card
FOREIGN KEY (credit_card_id)
REFERENCES credit_card(id)
ON DELETE SET NULL;
--
```

Output

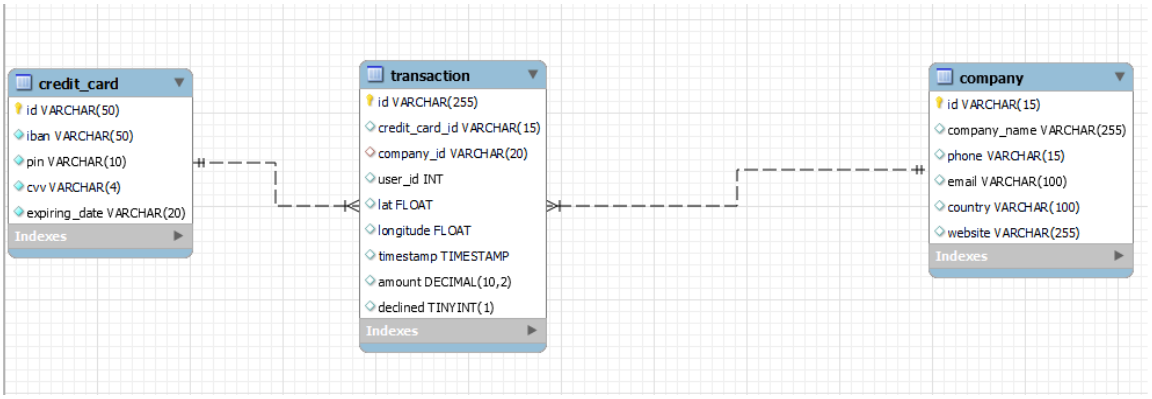
Action Output

#	Time	Action	Message	Duration / Fetch
287	21:18:45	DELETE FROM transaction WHERE id ="02C6201E-D90A-1859-B4EE-88D2986D3B02"	0 row(s) affected	0.000 sec
288	21:18:50	SELECT * from transaction WHERE id="02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
289	21:19:03	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_credit_card FOREIGN KEY (credit_card_id) REFERE...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0	0.140 sec

De esta manera quedarían establecidas las siguientes relaciones:

credit_card - transaction= 1:n

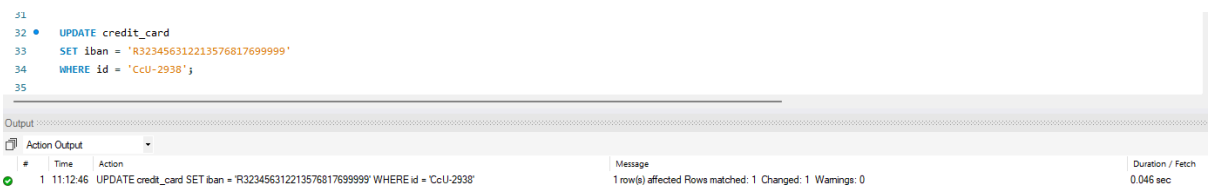
company-transaction= 1:n



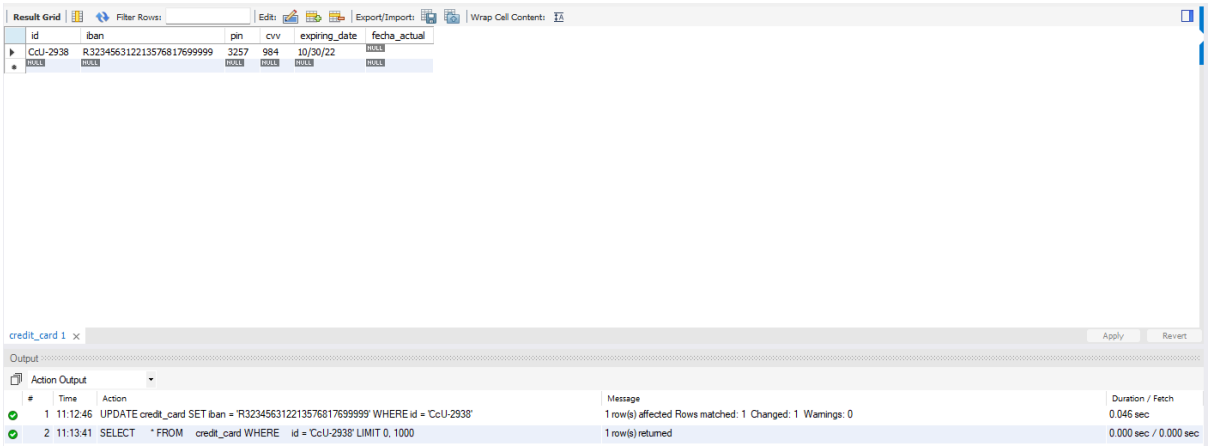
En este caso la tabla de hechos es la tabla 'transaction' mientras que company y credit_card son tablas dimensionales, la relación en este caso es UNO a MUCHOS en ambos casos (una tarjeta puedo tener muchas transacciones y una compañía puede haber realizado muchas transacciones)

- 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.



- Para modificar este campo utilizamos la función UPDATE que modifica la tabla **credit_card** donde está la columna **pan** y que contiene el campo a modificar, ubicamos este dato usando el filtro **WHERE** dónde ubicamos el número de id que contiene ese numero de pan en este caso siendo 'Ccu-2938'.



- 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

```
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);
```

```
583 10:33:05 INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ('108... Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`transactions`.`transaction`, CONS... 0.031 sec
```

- Para ingresar esta nueva información recurrimos a la función INSERT INTO en este caso a la tabla **transaction** pero devuelve un error ya que hay campos que no se registran en la tabla **credit_card** y en la tabla **company** así que primero debemos ingresar los datos allí mediante INSERT INTO en ambas tablas y rellenamos los datos faltantes para poder realizar el ingreso de datos a la tabla transaction

```
INSERT INTO credit_card (id, iban, pin, cvv, expiring_date)
VALUES ('CcU-9999', '5424465566813633', '3257', '984', '2025-10-30');
INSERT INTO company (id, company_name)
VALUES ('b-9999', 'Empresa Nueva');
```

```
44    -- EJERCICIO 3
45    -- (PRIMERO INSERTAMOS LOS SIGUIENTES DATOS PARA QUE LA QUERY FUNCIONE--
46    INSERT INTO credit_card (id, iban, pin, cvv, expiring_date)
47    VALUES ('CcU-9999', '5424465566813633', '3257', '984', '2025-10-30');
48    INSERT INTO company (id, company_name)
49    VALUES ('b-9999', 'Empresa Nueva');
50
51    -- AHORA SI PODEMOS INGRESAR LOS NUEVOS DATOS A LA TABLA TRANSACTION
52
53    INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
54    VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', 829.999, -117.999, 111.11, 0);
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
711	11:35:11	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VAL...	1 row(s) affected	0.000 sec
712	11:35:11	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VAL...	1 row(s) affected	0.000 sec
713	11:35:11	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VAL...	1 row(s) affected	0.000 sec
714	11:35:11	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VAL...	1 row(s) affected	0.000 sec
715	11:35:11	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VAL...	1 row(s) affected	0.000 sec

Verificamos que los datos hayan sido ingresados correctamente dentro de la tabla transaction:

```
75    -- CORROBORAMOS QUE LOS DATOS SEAN CORRECTOS
76    SELECT * FROM transaction
77    WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD';
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
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 7 | Apply | Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
11	11:36:44	SELECT * FROM credit_card WHERE id = 'CcU-9999' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
12	11:38:00	SELECT * from transaction WHERE user_id ='b-9999' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
13	11:39:47	SELECT * FROM transaction WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Buscar 11:40

- Exercici 4

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

- Para eliminar la columna pan utilizamos el método ALTER TABLE y DROP COLUMN **'pan'**,

```
68    -- EJERCICIO 4
69    -- ELIMINAMOS LA COLUMNA pan
70
71 •  ALTER TABLE credit_card DROP COLUMN pan;
72
73    -- CORROBORAMOS QUE HAYA SIDO CORRECTAMENTE ELIMINADA
74
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1543	19:02:31	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.109 sec

y verificamos que el cambio haya sido realizado

```
89
90 •  DESCRIBE credit_card;
91
```

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

Field	Type	Null	Key	Default	Extra
id	varchar(50)	NO	PRI	NULL	
iban	varchar(50)	YES		NULL	
pin	varchar(10)	YES		NULL	
cvv	varchar(4)	YES		NULL	
expiring_date	varchar(20)	YES		NULL	
fecha_actual	date	YES		NULL	

Result 8 | Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
12	11:38:00	SELECT * from transaction WHERE user_id ='b-9999' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
13	11:39:47	SELECT * FROM transaction WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
14	11:41:46	DESCRIBE credit_card	6 row(s) returned	0.016 sec / 0.000 sec

Buscar 11:41 26/2/2025

Nivell 2

Exercici 1

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

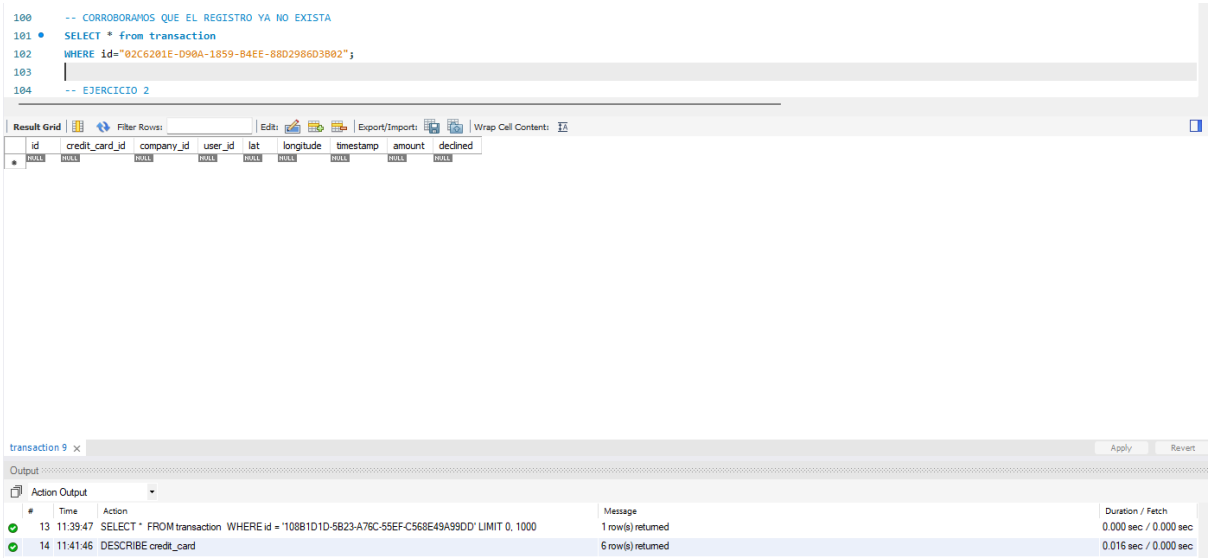
- Para eliminar la transacción con el correspondiente número de ID utilizamos el método DELETE from la tabla **transaction** y el filtro WHERE para ubicar ese número de ID

```
--
81 •  DELETE FROM transaction
82    WHERE id ="02C6201E-D90A-1859-B4EE-88D2986D3B02";
83
84    -- CORROBORAMOS QUE EL REGISTRO YA NO EXISTA
85 •  SELECT * from transaction
86    WHERE id="02C6201E-D90A-1859-B4EE-88D2986D3B02";
```

Output

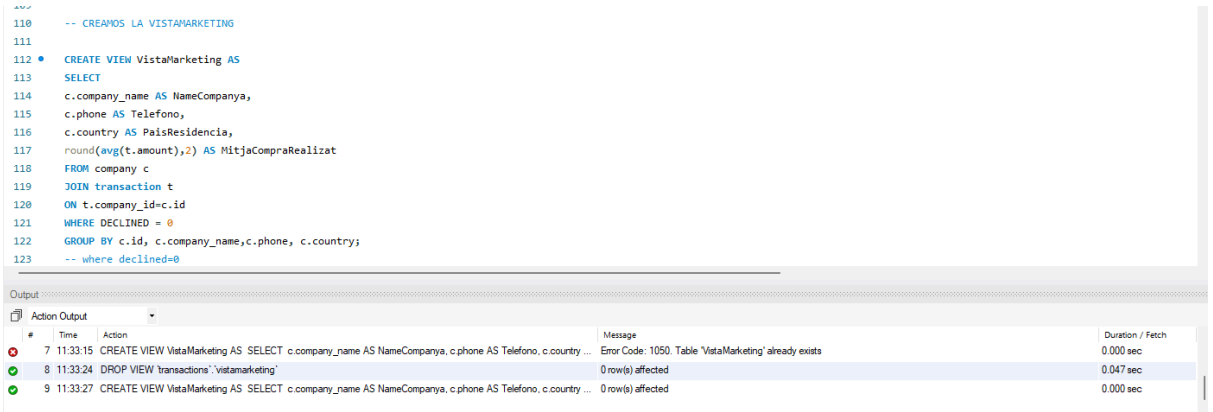
Action Output

#	Time	Action	Message	Duration / Fetch
1543	19:02:31	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.109 sec
1544	19:02:44	SELECT * from transaction WHERE id="02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
1545	19:02:53	DELETE FROM transaction WHERE id ="02C6201E-D90A-1859-B4EE-88D2986D3B02"	1 row(s) affected	0.032 sec



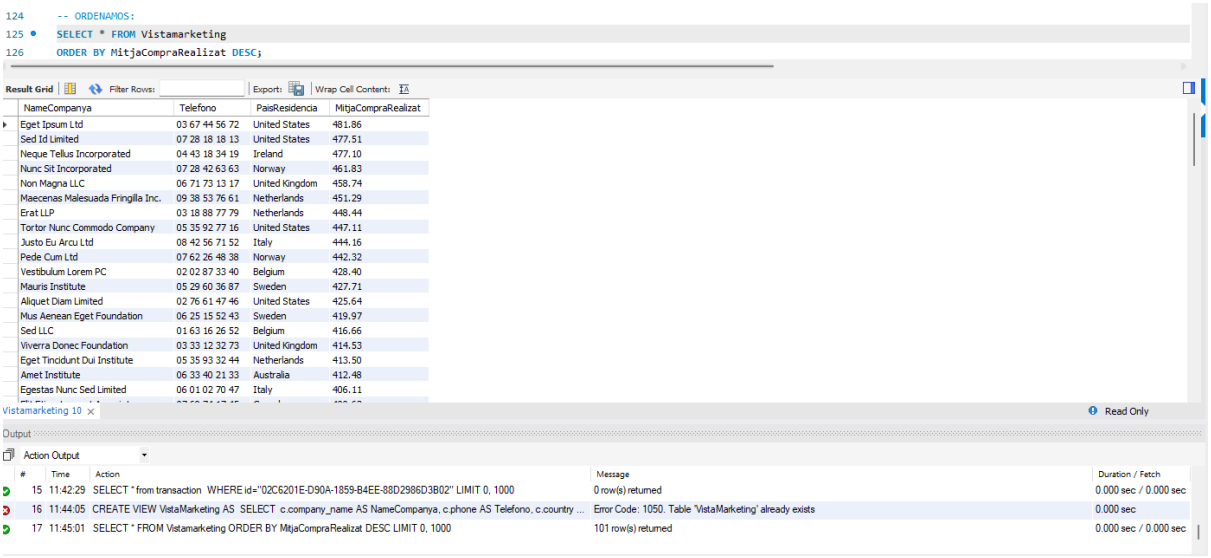
Exercici 2

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.



- Para crear la vista llamada VistaMarketing utilizamos CREATE VIEW que nos permite seleccionar los datos específicos de las tablas que queremos utilizar, en este caso de la tabla **company** tomamos los datos: **company_name, phone y country**, mientras que de la tabla **transaction** utilizamos la columna **amount** para saber la media de compra realizada por cada compañía realizando la media con **AVG,** (utilizamos round para que redondee la cantidad de decimales a 2) para utilizar ambas columnas realizamos un JOIN que podemos realizar mediante el id de las compañías en ambas tablas, nos aseguramos con **WHERE** que filtre las transacciones que fueron declinadas y para

presentarlo en el formato en que lo piden utilizamos ORDER BY(...) DESC para que muestre los datos de mayor a menor.



```
124    -- ORDENAMOS:
125    SELECT * FROM Vistamarketing
126    ORDER BY MitjaCompraRealizat DESC;
```

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

NameCompanya	Telefono	PaisResidencia	MitjaCompraRealizat
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
Maecenas Malesuada Fringilla Inc.	09 38 53 76 61	Netherlands	451.29
Erat LLP	03 18 88 77 79	Netherlands	448.44
Tortor Nunc Commodo Company	05 35 92 77 16	United States	447.11
Justo Eu Arcu Ltd	08 42 56 71 52	Italy	444.16
Pede Cum Ltd	07 62 26 48 38	Norway	442.32
Vestibulum Lorem PC	02 02 87 33 40	Belgium	428.40
Mauris Institute	05 29 60 36 87	Sweden	427.71
Aliquet Diam Limited	02 76 61 47 46	United States	425.64
Mus Aenean Eget Foundation	06 25 15 52 43	Sweden	419.97
Sed LLC	01 63 16 26 52	Belgium	416.66
Viverra Donec Foundation	03 33 12 32 73	United Kingdom	414.53
Eget Tincidunt Dui Institute	05 35 93 32 44	Netherlands	413.50
Amet Institute	06 33 40 21 33	Australia	412.48
Egestas Nunc Sed Limited	06 01 02 70 47	Italy	406.11

Vistamarketing 10 | Read Only

Output

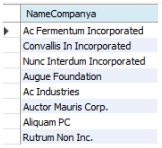
Action Output

#	Time	Action	Message	Duration / Fetch
15	11:42:29	SELECT * from transaction WHERE id="02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
16	11:44:05	CREATE VIEW VistaMarketing AS SELECT c.company_name AS NameCompanya, c.phone AS Telefono, c.country ...	Error Code: 1050. Table 'VistaMarketing' already exists	0.000 sec
17	11:45:01	SELECT * FROM Vistamarketing ORDER BY MitjaCompraRealizat DESC LIMIT 0, 1000	101 row(s) returned	0.000 sec / 0.000 sec

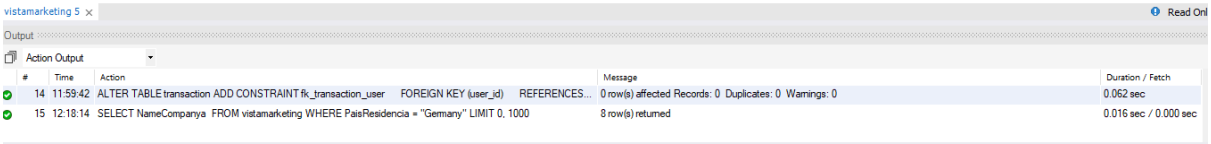
Exercici 3

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

```
SELECT NameCompanya
FROM vistamarketing
WHERE PaisResidencia = "Germany";
```



NameCompanya
Ac Fermentum Incorporated
Convallis In Incorporated
Nunc Interdum Incorporated
Augue Foundation
Ac Industries
Auctor Mauris Corp.
Aliquam PC
Rutrum Non Inc.



vistamarketing 5 | Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
14	11:59:42	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_user FOREIGN KEY (user_id) REFERENCES...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.062 sec
15	12:18:14	SELECT NameCompanya FROM vistamarketing WHERE PaisResidencia = "Germany" LIMIT 0, 1000	8 row(s) returned	0.016 sec / 0.000 sec

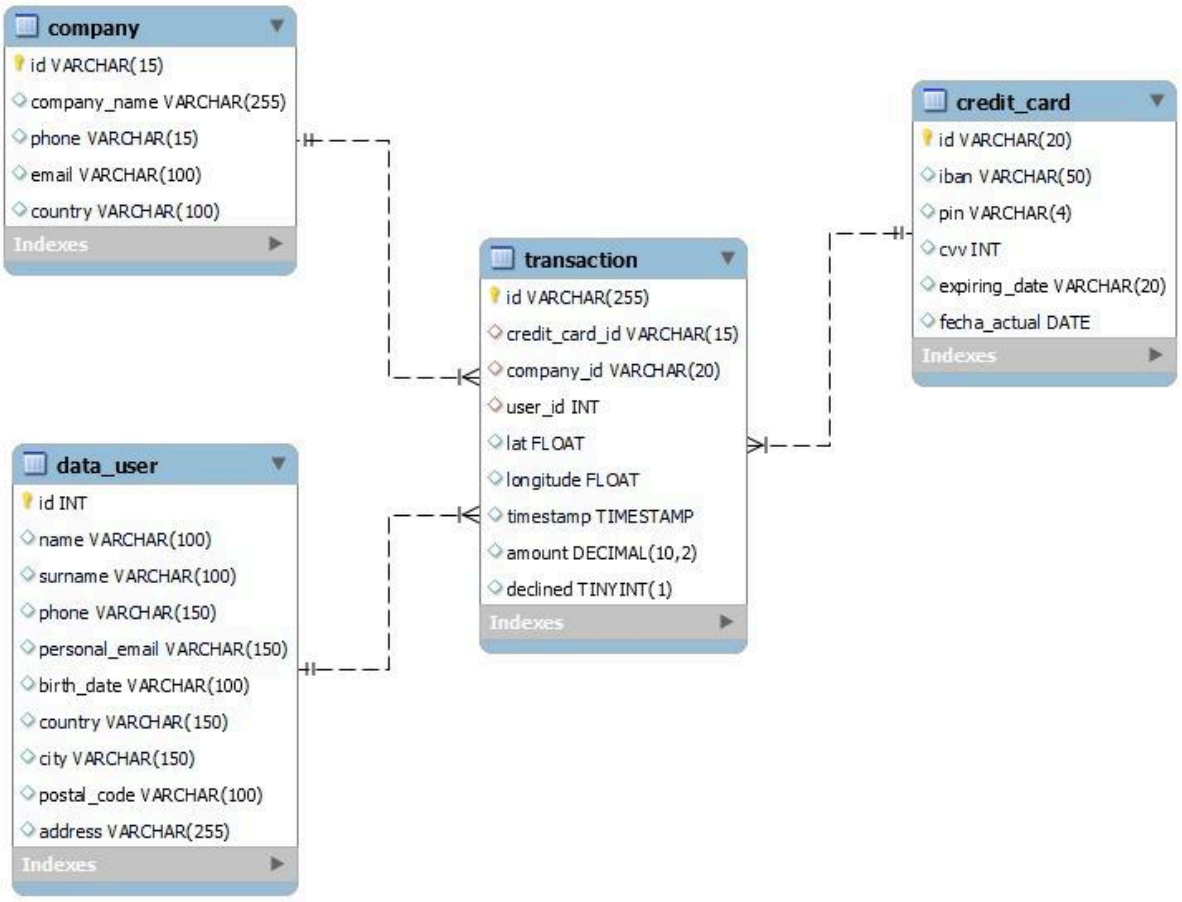
Para este filtro seleccionamos queremos mostrar el nombre de la compañia que residen en alemania para eso seleccionamos por namecompanya dentro de la vista que creamos llamada

vistamarketing y filtramos con un where paisresidencia = germany que nos devolvera los datos que queremos mostrar.

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 pbtenir 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"

- En primera instancia vemos que todas las relaciones son relaciones débiles por eso están identificadas con líneas rectas discontinuas, por debemos modificar este tipo de relación (fuerte a débil)
- 1º PASO: Cargar la tabla user

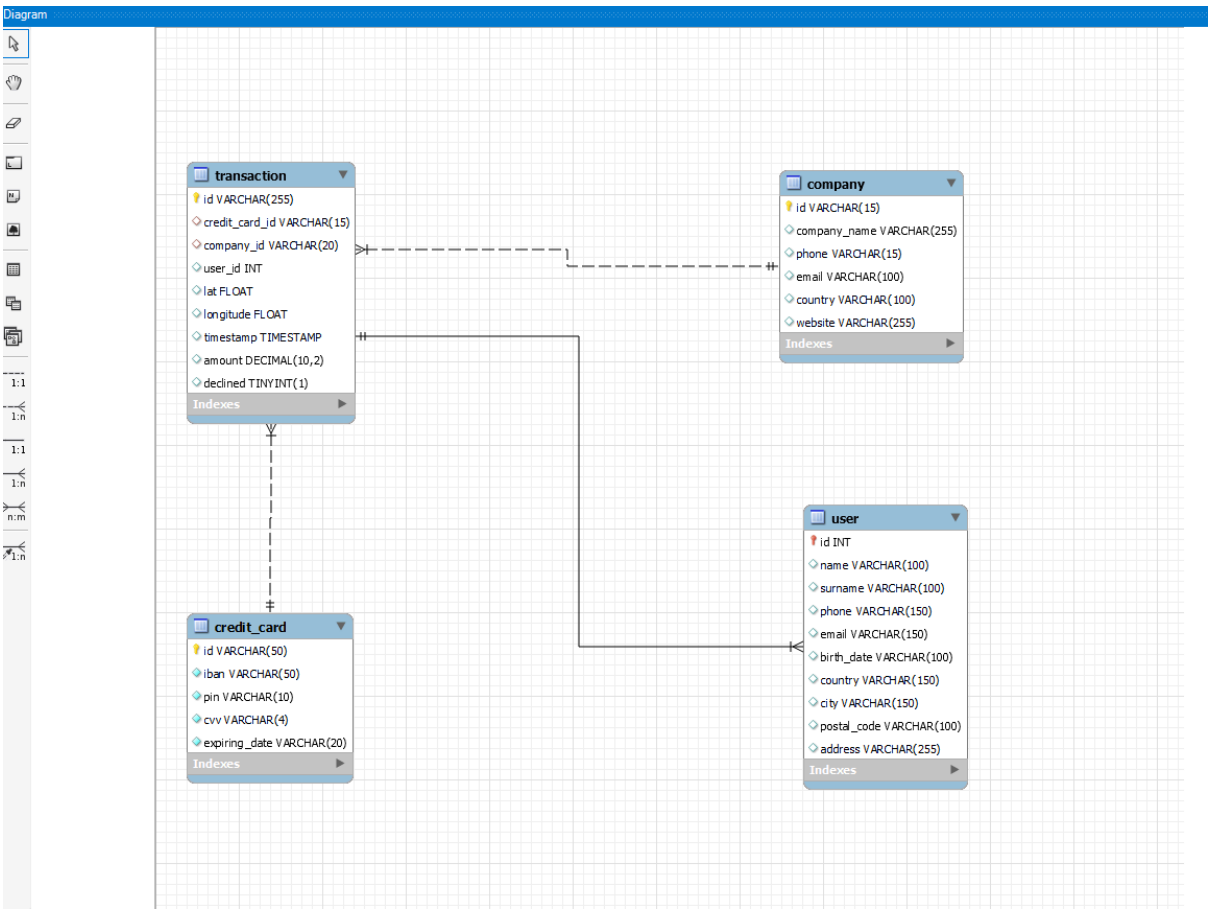
```
121    -- CARGAMOS LA TABLA USER ORIGINAL CON EL SIGUIENTE COMANDO
122    CREATE INDEX idx_user_id ON transaction(user_id);
123
124    CREATE TABLE IF NOT EXISTS user (
125            id INT PRIMARY KEY,
126            name VARCHAR(100),
127            surname VARCHAR(100),
128            phone VARCHAR(150),
129            email VARCHAR(150),
130            birth_date VARCHAR(100),
131            country VARCHAR(150),
132            city VARCHAR(150),
133            postal_code VARCHAR(100),
134            address VARCHAR(255),
135            FOREIGN KEY(id) REFERENCES transaction(user_id)
136        );
```

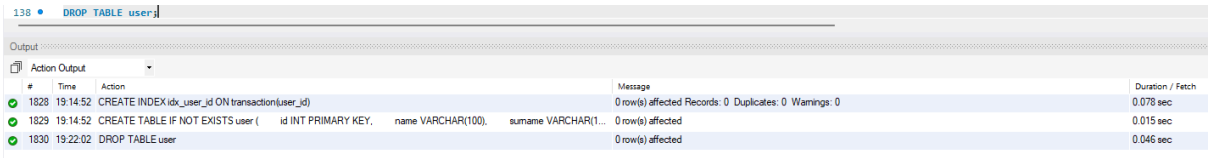
Output

Action Output

#	Time	Action	Message	Duration / Fetch
1827	19:14:45	DROP TABLE `transactions`.`user`	0 row(s) affected	0.047 sec
1828	19:14:52	CREATE INDEX idx_user_id ON transaction(user_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.078 sec
1829	19:14:52	CREATE TABLE IF NOT EXISTS user (id INT PRIMARY KEY, name VARCHAR(100), surname VARCHAR(1...	0 row(s) affected	0.015 sec

Pero si cargamos este comando el diagrama que obtenemos es el siguiente:





- 2º paso :
- Cargamos los datos desde ‘datos_introduir_user’.

```
SET foreign_key_checks = 0;

-- Insertamos datos de user
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "1", "Zeus", "Gamble", "1-282-581-0551", "interdum.enim@protonmail.edu", "Nov 1
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "2", "Garrett", "Mcconnell", "(718) 257-2412", "integer.vitae.nibh@protonmail.c
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "3", "Ciaran", "Harrison", "(522) 598-1365", "interdum.feugiat@aol.org", "Apr 2
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "4", "Howard", "Stafford", "1-411-740-3269", "ornare.egestas@icloud.edu", "Feb
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "5", "Hayfa", "Pierce", "1-554-541-2077", "et.malesuada.fames@hotmail.org", "Se
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "6", "Joel", "Tyson", "(718) 288-8020", "gravida.nunc.sed@yahoo.ca", "Oct 15, 1
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "7", "Rafael", "Jimenez", "(817) 689-0478", "eget@outlook.ca", "Dec 4, 1981",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "8", "Nissim", "Franks", "(692) 157-3469", "egestas.aliquam.fringilla@google.ca
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "9", "Mannix", "Mcclain", "(590) 883-2184", "aliquam.nisl@outlook.com", "Jan 24
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "10", "Robert", "Mccarthy", "(324) 746-6771", "fermentum@protonmail.com", "Apr
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "11", "Joan", "Baird", "(981) 429-8106", "et@outlook.net", "Feb 25, 1990",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "12", "Benedict", "Wheeler", "1-515-824-2855", "tincidunt.donec.vitae@hotmail.c
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "13", "Allegra", "Stanton", "1-927-753-6488", "proin.eget@protonmail.ca", "May
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "14", "Sara", "Flynn", "1-311-646-9333", "integer@outlook.net", "Dec 27, 1988",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "15", "Noelani", "Patrick", "1-723-488-5894", "sem.magna@google.com", "Sep 17,
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "16", "Eric", "Roth", "1-218-549-8253", "lorem.sit@yahoo.net", "Sep 7, 1988",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "17", "Bruce", "Gill", "(744) 732-8628", "metus@aol.net", "Mar 4, 1990",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "18", "Russell", "Jimenez", "(657) 779-2438", "orci@outlook.edu", "Aug 26, 199
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "19", "Nicholas", "Travis", "1-330-223-9652", "libero.dui@hotmail.com", "Jul 15
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "20", "Kelsey", "Bates", "(653) 724-4754", "ullamcorper.nisl@aol.com", "May 6,
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "21", "Hall", "Reeves", "(241) 759-9235", "erat.eget@hotmail.edu", "Jul 22, 198
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "22", "Allistair", "Holmes", "1-265-323-0812", "donec.tempor.est@protonmail.com
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "23", "Kelsie", "Bass", "1-837-832-5631", "consequat@google.ca", "Apr 2, 1990",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "24", "Nolan", "Cash", "(273) 334-3785", "nam@hotmail.com", "Sep 9, 1994",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "25", "Wanda", "Campbell", "(702) 823-5535", "sagittis@google.couk", "May 31, 1
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "26", "Aquila", "Strickland", "1-246-231-5495", "enim.sit@icloud.com", "Sep 28,
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "27", "Diana", "Williamson", "1-285-365-7779", "id.nunc@google.com", "Dec 14, 1
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (       "28", "Elmo", "Cain", "1-663-583-6021", "nec.metus.facilisis@google.org", "Oct
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1755	19:11:12	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("205", "S...	1 row(s) affected	0.000 sec
1756	19:11:12	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("206", "...	1 row(s) affected	0.000 sec
1757	19:11:12	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("207", "Y...	1 row(s) affected	0.000 sec
1758	19:11:12	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("208", "B...	1 row(s) affected	0.000 sec

- 3º paso:
- Para modificar la relación entre user y transaction realizamos los siguiente con los comandos ALTER TABLE … DROP FOREIGN KEY para eliminar la FK existente en user

```
-- ELIMINAR LA FK DE LA TABLA USER
ALTER TABLE user
DROP FOREIGN KEY user_ibfk_1;

--
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
589	20:28:06	SET foreign_key_checks = 1	0 row(s) affected	0.000 sec
590	20:29:34	Error loading schema content	Error Code: 1356 View 'transactions.informetecnico' references invalid table(s) or column(s) or function(s) or definer/invoker of ...	
591	20:31:12	ALTER TABLE user DROP FOREIGN KEY user_ibfk_1	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec

- 4º Paso:
- Cambiamos el nombre de la tabla user por data_user con ALTER_TABLE

```
-- CAMBIAMOS EL NOMBRE DE LA TABLA USER POR DATA_USER
ALTER TABLE user RENAME data_user;
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2110	19:26:08	ALTER TABLE transaction ADD CONSTRAINT idx_user_id FOREIGN KEY (user_id) REFERENCES data_user(id) O...	Error Code: 1824. Failed to open the referenced table 'data_user'	0.000 sec
2111	19:28:47	SELECT * FROM transactions.user LIMIT 0, 1000	275 row(s) returned	0.016 sec / 0.000 sec
2112	19:29:13	ALTER TABLE user RENAME data_user	0 row(s) affected	0.047 sec

- 5º Paso:
- Con el siguiente comando nos aseguramos de revisar que no falten registros ni en la tabla user ni transaction (es decir que no haya transacciones sin un usuario creado)

```
184 •  SELECT user_id FROM transaction WHERE user_id NOT IN (SELECT id FROM data_user);  -- ASI VEMOS SI NOS FALTA UN REGISTRO DENTRO DE LA TABLA Y LO AGREGAMOS:
185
```

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

user_id
9999

transaction 5

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2111	19:28:47	SELECT * FROM transactions.user LIMIT 0, 1000	275 row(s) returned	0.016 sec / 0.000 sec
2112	19:29:13	ALTER TABLE user RENAME data_user	0 row(s) affected	0.047 sec
2113	19:31:32	SELECT user_id FROM transaction WHERE user_id NOT IN (SELECT id FROM data_user) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

- 6º Paso:
- Este comando arroja el siguiente resultado (esta transacción había sido creada anteriormente pero no tenemos datos del usuario) entonces de la siguiente manera cargamos los datos que nos faltan:

```
129 •  INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address)
130    VALUES (9999, 'Rocio', 'Rivero', '603515267', 'email@ejemplo.com', '2000-01-01', 'Spain', 'Barcelona', '08004', 'xxxxxx');
```

Output

Action Output

#	Time	Action	Message
1170	10:10:54	SELECT user_id FROM transaction WHERE user_id NOT IN (SELECT id FROM user) LIMIT 0, 1000	1 row(s) returned
1171	10:12:18	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (99...	1 row(s) affected

- 7º Paso:
- Una vez hecho esto ya podemos ejecutar la query donde relacionamos la tabla **transaction** con la tabla que cargamos (user)

```
156 •  ALTER TABLE transaction
157    ADD CONSTRAINT fk_user_transaction
158    FOREIGN KEY (user_id)
159    REFERENCES data_user(id)
160    ON DELETE SET NULL;
161
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
882	20:38:19	INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (9999, '...	1 row(s) affected	0.015 sec
883	20:38:41	ALTER TABLE transaction ADD CONSTRAINT fk_user_transaction FOREIGN KEY (user_id) REFERENCES user(id) ON ...	Error Code: 1824. Failed to open the referenced table 'user'	0.015 sec
884	20:40:01	ALTER TABLE transaction ADD CONSTRAINT fk_user_transaction FOREIGN KEY (user_id) REFERENCES data_user(id) ...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0	0.140 sec

y procedemos a limpiar los registros que tenemos para que solo queden los requeridos en el diagrama:

- 8ºPaso

- En la tabla **credit_card** agregamos la columna **fecha_actual** de la siguiente manera:

```
155
156 • ALTER TABLE credit_card
157   ADD COLUMN fecha_actual DATE;
158
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1206	10:41:30	alter table credit_card drop column fecha_actual	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec
1207	10:41:59	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE DEFAULT current_timestamp	Error Code: 1067. Invalid default value for 'fecha_actual'	0.000 sec
1208	10:42:12	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE DEFAULT current_date	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server ...	0.000 sec
1209	10:42:37	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE DEFAULT timestamp	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server ...	0.000 sec
1210	10:45:42	alter table credit_card drop column fecha_actual	Error Code: 1091. Can't DROP 'fecha_actual'; check that column/key exists	0.000 sec
1211	10:46:10	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.062 sec

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

id	iban	pin	cvv	expiring_date	fecha_actual
CcU-2938	TR301950312213576817638661	3257	984	10/30/22	NULL
CcU-2945	DO26854763748537475216568689	9080	887	08/24/23	NULL
CcU-2952	BG45IVQL52710525608255	4598	438	06/29/21	NULL
CcU-2959	CR7242477244335841535	3583	667	02/24/23	NULL
CcU-2966	BG72LKTQ15627628377363	4900	130	10/29/24	NULL
CcU-2973	PT87806228135092429456346	8760	887	01/30/25	NULL
CcU-2980	DE39241881883086277136	5075	596	07/24/22	NULL
CcU-2987	GE89681434837748781813	2298	797	10/31/23	NULL
CcU-2994	BH62714428368066765294	7545	595	02/28/22	NULL
CcU-3001	CY490874266547745812668321 10	9562	867	09/16/22	NULL
CcU-3008	LU507216693616119230	1856	740	04/05/25	NULL

credit_card 1 | Apply | Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1207	10:41:59	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE DEFAULT current_timestamp	Error Code: 1067. Invalid default value for 'fecha_actual'	0.000 sec
1208	10:42:12	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE DEFAULT current_date	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server ...	0.000 sec
1209	10:42:37	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE DEFAULT timestamp	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server ...	0.000 sec
1210	10:45:42	alter table credit_card drop column fecha_actual	Error Code: 1091. Can't DROP 'fecha_actual'; check that column/key exists	0.000 sec
1211	10:46:10	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.062 sec
1212	10:47:13	SELECT * FROM transactions.credit_card LIMIT 0, 1000	276 row(s) returned	0.000 sec / 0.000 sec

- 9º Paso:

En la tabla **data_user** le cambiamos el nombre a la columna email que ahora se llamará **personal_email:**

```
174 • ALTER TABLE data_user
175   RENAME COLUMN email TO personal_email;
176
177
178
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
3	10:52:47	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec
4	10:59:42	ALTER TABLE data_user RENAME COLUMN email TO personal_email	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec

- 10º Paso:
- Por último eliminamos la columna **website** de la tabla **company:**

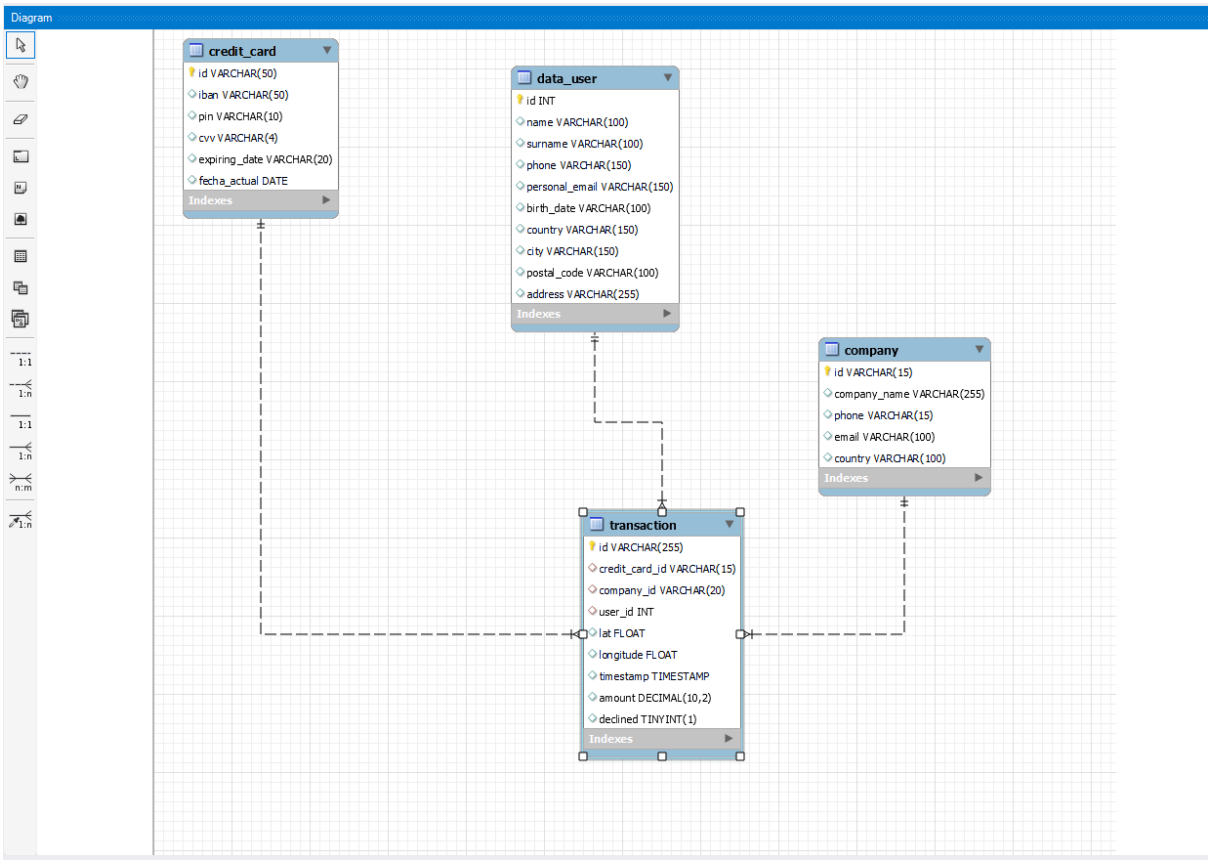
```
171 • ALTER TABLE company
172   DROP COLUMN website;
173
174
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	10:51:21	SELECT * FROM transactions.credit_card LIMIT 0, 1000	276 row(s) returned	0.000 sec / 0.015 sec
3	10:52:47	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec

Con estos cambios realizados y ejecutados deberíamos obtener el diagrama pedido:



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.

- Para crear esta vista llamada InformeTecnico seguimos los pasos que realizamos anteriormente para crear la VistaMarketing

```sql
CREATE VIEW InformeTecnico AS
SELECT
t.id AS IdTransaccion,
u.name AS NombreUser,
u.surname AS ApellidoUser,
cc.iban AS IbanTarjeta,
c.company_name AS NombreCompanya
FROM transaction t
JOIN user u ON u.id=t.user_id
JOIN credit_card cc ON cc.id=t.credit_card_id
JOIN company c ON t.company_id=c.id;

SELECT *
FROM informetecnico
ORDER BY IdTransaccion DESC;
```

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

IdTransaccion	NombreUser	ApellidoUser	IbanTarjeta	NombreCompanya
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.
FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF5886820057577163458381 3	Arcu LLP
FAAD3FFC-1A17-E141-43D3-359A5BA7CB3B	Hedwig	Gilbert	GE9015792884333813446 3	Lorem Eu Incorporated
FA053936-75D8-85FA-490D-9B624E1B920A	Hedwig	Gilbert	GT0249765365533084824764597 5	Non Justo Corp.
F85A7D75-2778-9D75-D776-3F41A828DE88	Sarah	Beck	VG1468087984174645729577	Ut Semper Foundation
F843DC08-CCB5-2444-1B4E-5966289FBA8B	Jasper	Landry	VG1468087984174645729577	Ut Semper Foundation
F5ACD74B-4275-5AA1-2414-6EF417636B98	Nora	Reeves	MD1234119525145401270486	Nunc Interdum Incorporated
F56FCA4A-0039-9F64-7376-85632B91121B	Lynn	Riddle	CR7242477244335841535	Ut Semper Foundation
F55B3CE1-3379-E0BF-5AB9-6F4CC2C5479C	Sonya	Mckee	EE541536644818872885	Arcu LLP
F4BCAE41-3B8E-EA8D-9C24-466F7CEB9F9A	Chester	Haynes	CY9426353740501548118862557 6	Malesuada PC
F2B3F645-2F6D-F891-9D05-33DBACF58DF4	Heather	Burks	SM6022751049715477062682363	Malesuada PC

informetecnico 4 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
9	11:07:05	SELECT NameCompanya FROM vistamarketing WHERE PaisResidencia = "Germany" LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 se
10	11:12:56	SELECT * FROM informetecnico ORDER BY IdTransaccion DESC LIMIT 0, 1000	586 row(s) returned	0.015 sec / 0.000 se

En este caso seleccionamos datos de la tabla **transaction** (id), de la tabla **user** (Nombre y apellido) y de la tabla **company** seleccionamos el numero de IBAN y el nombre de la compañía, para poder obtener todos estos datos realizamos JOIN de la tabla **user** con la tabla transaction donde seleccionamos los id, la tabla credit card con transaction también podemos encontrar las claves de los id de las tarjetas de crédito y por último company con transaction con los respectivos id de cada compañía y para mostrar los datos como nos es pedido finalmente ordenamos los resultados por ORDER BY idtransaction en orden DESC.