

SPRINT 3

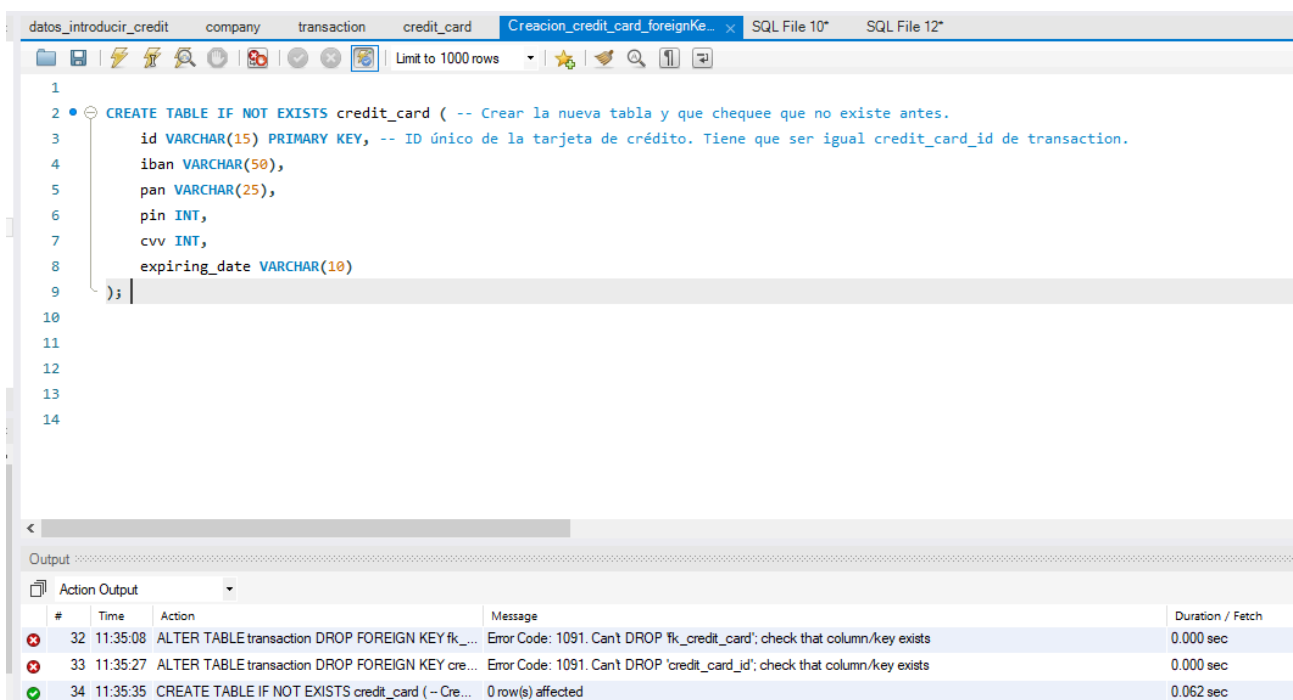
Nivel 1

Ejercicio 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.

He creado la tabla a partir de las columnas y datos de "dades_introduir_credit".

Para establecer la relación con la tabla transaction, la modifiqué utilizando ADD CONSTRAINT porque ya estaba creada. Esto permite agregar la clave foránea después y asignarle un nombre específico a la restricción.



```
1
2 CREATE TABLE IF NOT EXISTS credit_card ( -- Crear la nueva tabla y que chequee que no existe antes.
3   id VARCHAR(15) PRIMARY KEY, -- ID único de la tarjeta de crédito. Tiene que ser igual credit_card_id de transaction.
4   iban VARCHAR(50),
5   pan VARCHAR(25),
6   pin INT,
7   cvv INT,
8   expiring_date VARCHAR(10)
9 );
10
11
12
13
14
```

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|---|------------------|
| 32 | 11:35:08 | ALTER TABLE transaction DROP FOREIGN KEY fk_... | Error Code: 1091. Can't DROP 'fk_credit_card'; check that column/key exists | 0.000 sec |
| 33 | 11:35:27 | ALTER TABLE transaction DROP FOREIGN KEY cre... | Error Code: 1091. Can't DROP 'credit_card_id'; check that column/key exists | 0.000 sec |
| 34 | 11:35:35 | CREATE TABLE IF NOT EXISTS credit_card (-- Cre... | 0 row(s) affected | 0.062 sec |

```

1
2 • CREATE TABLE IF NOT EXISTS credit_card ( -- Crear la nueva tabla y que chequee que no existe antes.
3     id VARCHAR(15) PRIMARY KEY, -- ID único de la tarjeta de crédito. Tiene que ser igual credit_card_id de transaction.
4     iban VARCHAR(50),
5     pan VARCHAR(25),
6     pin INT,
7     cvv INT,
8     expiring_date VARCHAR(10)
9 );
10
11 • ALTER TABLE transaction
12 ADD CONSTRAINT fk_credit_card
13 FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
14
15
16
17

```

Output:

| # | Time | Action | Message |
|-----|----------|---------------------------|--|
| 311 | 11:38:49 | INSERT INTO credit_car... | 1 row(s) affected |
| 312 | 11:38:49 | INSERT INTO credit_car... | 1 row(s) affected |
| 313 | 11:41:48 | CREATE TABLE IF NOT ... | 0 row(s) affected, 1 warning(s): 1050 Table 'credit_card' already exists |
| 314 | 11:41:48 | ALTER TABLE transaccio... | 587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0 |

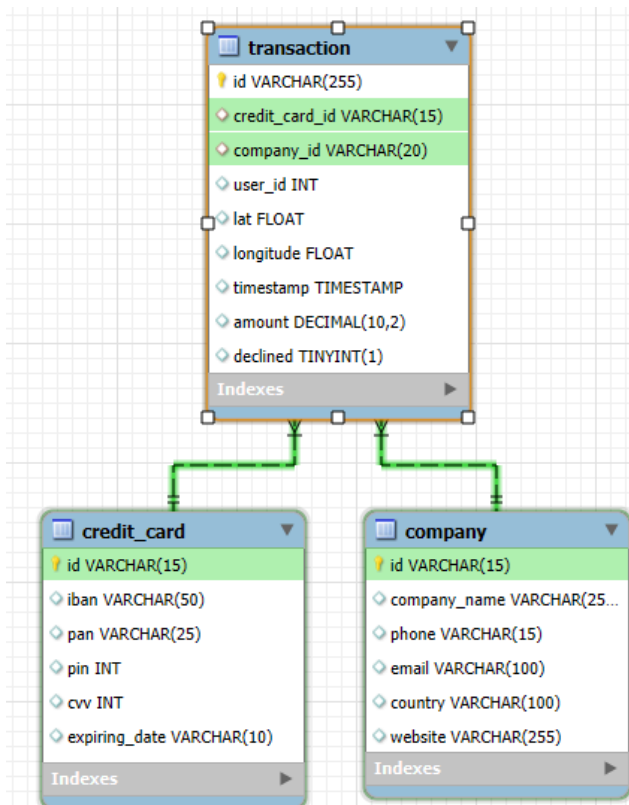
Los registros a introducir:

```

1 • SET foreign_key_checks = 0;
2
3 -- Insertamos datos de user
4 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
5 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
6 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
7 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
8 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
9 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
10 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
11 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
12 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
13 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
14 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
15 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
16 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
17 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
18 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
19 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
20 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (

```

Diagrama:



Aquí se pueden ver 3 tablas: transaction, company y la nueva credit_card.

En la tabla transaction, la columna **id** es la clave primaria, que identifica cada transacción de forma única.

Ahora tiene dos foreign keys:

credit_card_id que está conectada con la columna **id** de la tabla **credit_card**, entonces cada transacción está asociada a una tarjeta de crédito.

company_id estaba conectada con la columna **id** de **company**, entonces cada transacción está vinculada con una empresa.

Tabla credit_card: **id** es la clave primaria. Está relacionada con la tabla **transaction** a través de **credit_card_id**. Esto significa que una tarjeta puede estar asociada a varias transacciones.

Tabla company: **id** es la clave primaria, que identifica a cada empresa. Una compañía/empresa puede tener muchas

transacciones. Está relacionada con la tabla **transaction** con la columna **company_id**.

Ejercicio 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 un dato de una registro se usa update, se escoge qué tabla se quiere modificar se hace set para decirle cuales son los datos a modificar mediante el nombre de la columna y luego con where se determina cuál registro, generalmente utilizando la clave primaria o algún otro criterio, pero mejor la clave primaria para asegurar de que solo se modifique ese registro en específico.

⇒ Demostrar cambio realizado:

Dame todos los campos de la tabla credit_card en donde el id es igual a CcU: 2938.

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
5 • UPDATE credit_card
6 SET iban = 'R323456312213576817699999'
7 WHERE id = 'CcU-2938';
8
9 • SELECT * FROM credit_card WHERE id = 'CcU-2938';
10
11
12
```

The results pane displays a table with the following data:

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

The output pane shows the execution log with the following messages:

| # | Time | Action | Message |
|---|----------|--|--|
| 2 | 13:07:15 | update credit_card set id = 'ID CcU-2938' where iban ... | Error Code: 1054. Unknown column 'R323456312213576817699999' in 'where clause' |
| 3 | 13:25:26 | UPDATE credit_card SET iban = 'R32345631221357... | Error Code: 1054. Unknown column 'CcU' in 'where clause' |
| 4 | 13:26:31 | UPDATE credit_card SET iban = 'R32345631221357... | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 |
| 5 | 13:27:47 | SELECT * FROM transactions.credit_card LIMIT 0, 1... | 275 row(s) returned |
| 6 | 13:29:29 | UPDATE credit_card SET iban = 'R32345631221357... | 0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0 |
| 7 | 13:29:29 | SELECT * FROM credit_card WHERE id = 'CcU-293... | 1 row(s) returned |

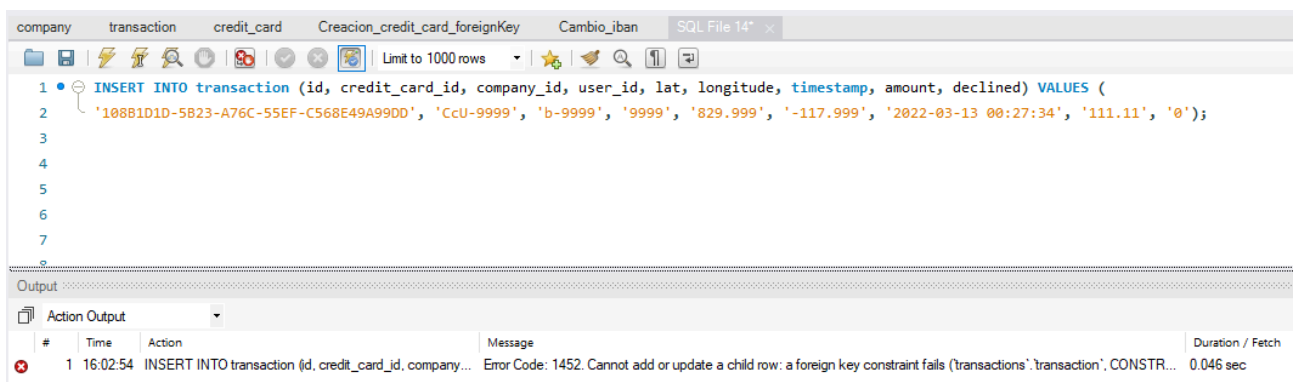
Ejercicio 3

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 |

Cuando he intentado insertar la fila, me ha salido un error:

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails
(`transactions`.`transaction`, CONSTRAINT `transaction_ibfk_1` FOREIGN KEY (`company_id`) REFERENCES `company` (`id`))



```
company transaction credit_card Creacion_credit_card_foreignKey Cambio_iban SQL File 14" x
Limit to 1000 rows
1 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES (
2   '108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '2022-03-13 00:27:34', '111.11', '0');
3
4
5
6
7
8
Output
Action Output
# Time Action Message Duration / Fetch
1 16:02:54 INSERT INTO transaction (id, credit_card_id, company... Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('transactions`.`transaction`, CONSTR... 0.046 sec
```

Esto quiere decir que hay un fallo, que se está intentado actualizar una fila en transaction pero no tiene relación con la tabla company. Entonces hay que crear el registro en la tabla company con id = b-9999 y crear un registro en la tabla credit_card para el id CcU-9999

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails
(`transactions`.`#sql-17e4_40`, CONSTRAINT `transaction_ibfk_2` FOREIGN KEY (`user_id`) REFERENCES `data_user` (`id`))

Podemos chequear si es cierto, con `Select * From`, cualquiera de los datos que queremos introducir.

Y luego con `Insert into ()` y valores introducimos los datos.

The screenshot shows a SQL editor with the following queries:

```
1 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES (  
2   '10881D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '2022-03-13 00:27:34', '111.11', '0');  
3  
4 • SELECT * FROM company WHERE id = 'b-9999';  
5  
6 • INSERT INTO company (id, company_name, phone, email, country, website)  
7   VALUES ('b-9999', 'NombreSprint3 Nivel1_ejercicio 3', '123456789', 'info@empresaSprint3_1_3.com', 'Sprint3_1_3', 'www.empresa_Sprint3_1_3.com');  
8  
9 • SELECT * FROM credit_card WHERE id = 'CcU-9999';  
10  
11 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  
12   'CcU-9999', 'IBAN-9999', 'PAN-9999', 1234, 987, '2024-12-29');
```

The output table shows the execution results:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|---|------------------|
| 1 | 16:22:22 | INSERT INTO transaction (id, credit_card_id, company_id,... | Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('transactions'.transaction', CONSTRAINT ... | 0.016 sec |
| 2 | 16:24:04 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring... | 1 row(s) affected | 0.000 sec |

Aquí se ha registrado el nuevo usuario en la tabla transaction:

The screenshot shows the same SQL editor with the following queries:

```
1 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES (  
2   '10881D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '2022-03-13 00:27:34', '111.11', '0');  
3  
4 • SELECT * FROM company WHERE id = 'b-9999';  
5  
6 • INSERT INTO company (id, company_name, phone, email, country, website)  
7   VALUES ('b-9999', 'NombreSprint3 Nivel1_ejercicio 3', '123456789', 'info@empresaSprint3_1_3.com', 'Sprint3_1_3', 'www.empresa_Sprint3_1_3.com');  
8  
9 • SELECT * FROM credit_card WHERE id = 'CcU-9999';  
10  
11 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  
12   'CcU-9999', 'IBAN-9999', 'PAN-9999', 1234, 987, '2024-12-29');
```

The output table shows the execution results:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|---|------------------|
| 1 | 16:22:22 | INSERT INTO transaction (id, credit_card_id, company_id,... | Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('transactions'.transaction', CONSTRAINT ... | 0.016 sec |
| 2 | 16:24:04 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring... | 1 row(s) affected | 0.000 sec |
| 3 | 16:25:02 | INSERT INTO transaction (id, credit_card_id, company_id,... | 1 row(s) affected | 0.016 sec |

Comprobación de registro:

The screenshot shows a SQL IDE with a script titled "SQL File 14". The script contains the following SQL statements:

```

5
6 • INSERT INTO company (id, company_name, phone, email, country, website)
7   VALUES ('b-9999', 'NombreSprint3 Nivel1_ejercicio 3', '123456789', 'info@empresaSprint3_1_3.com', 'Sprint3_1_3', 'www.empresa_Spr
8
9 • SELECT * FROM credit_card WHERE id = 'CcU-9999';
10
11 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
12   'CcU-9999', 'IBAN-9999', 'PAN-9999', 1234, 987, '2024-12-29');
13
14 • SELECT * FROM transaction WHERE company_id = 'b-9999';
  
```

The "Result Grid" shows the results of the queries:

| id | credit_card_id | company_id | user_id | lat | longitude | timestamp | amount | declined |
|--------------------------------------|----------------|------------|---------|---------|-----------|---------------------|--------|----------|
| 108B1D1D-SB23-A76C-55EF-C568E49A99DD | CcU-9999 | b-9999 | 9999 | 829.999 | -117.999 | 2022-03-13 00:27:34 | 111.11 | 0 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

The "Output" section shows the "Action Output" for the script:

| # | Time | Action | Message |
|---|----------|--|---|
| 1 | 16:22:22 | INSERT INTO transaction (id, credit_card_id, company... | Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('transactions', 'transaction', 'CONSTRAI... |
| 2 | 16:24:04 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expin... | 1 row(s) affected |
| 3 | 16:25:02 | INSERT INTO transaction (id, credit_card_id, company... | 1 row(s) affected |
| 4 | 16:27:07 | SELECT * FROM transaction WHERE company_id = 'b-... | 1 row(s) returned |

Ejercicio 4

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

The screenshot shows a SQL IDE with a script titled "SQL File 15". The script contains the following SQL statements:

```

1 • ALTER TABLE credit_card DROP COLUMN pan;
2
3 • SHOW COLUMNS FROM credit_card;
  
```

The "Result Grid" shows the results of the queries:

| Field | Type | Null | Key | Default | Extra |
|---------------|-------------|------|-----|---------|-------|
| id | varchar(15) | NO | PRI | NULL | |
| iban | varchar(50) | YES | | NULL | |
| pin | int | YES | | NULL | |
| cvv | int | YES | | NULL | |
| expiring_date | varchar(10) | YES | | NULL | |

The "Output" section shows the "Action Output" for the script:

| # | Time | Action | Message |
|---|----------|---|--|
| 1 | 16:32:16 | ALTER TABLE credit_card DROP COLUMN pan | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 |
| 2 | 16:32:57 | SHOW COLUMNS FROM credit_card | 5 row(s) returned |

Nivel 2

Ejercicio 1

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

Primero he chequeado que existiera y luego lo he eliminado.

Es importante utilizar el where para saber qué registros eliminar. Si no se incluye una condición, se eliminan todos los registros de la tabla.

The screenshot shows a SQL IDE interface with a query editor and an output window. The query editor contains the following SQL code:

```
1  -- Elimina de la taula transaction el registre amb ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de dades.
2
3  • SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
4
5  • DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
6
7
```

Below the query editor, there is a "Result Grid" section showing a table with columns: id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, and declined. The table contains one row with all values set to NULL.

The "Output" window shows the execution results:

| # | Time | Action | Message |
|---|----------|--|-------------------|
| 3 | 16:38:18 | SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02' LIMIT 0, 10... | 1 row(s) returned |
| 4 | 16:38:49 | DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02' | 1 row(s) affected |
| 5 | 16:39:30 | SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02' LIMIT 0, 10... | 0 row(s) returned |

Ejercicio 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.

⇒ Creamos la vista y escogemos las columnas necesarias, hacemos el join y agrupamos.

```
1 /*Vista sobre las compañías y sus transacciones, llamada VistaMarketing
2  Nom de la companyia. Telèfon de contacte. País de residència. Mitjana de compra realitzat per cada companyia.
3  Ordenada de mayor a menor por promedio de compra.*/
4
5 • CREATE VIEW VistaMarketing AS
6 SELECT company_name,
7        phone,
8        country,
9        ROUND(AVG(amount),2) AS promedio_compra
10 FROM company c
11 JOIN transaction t ON c.id = t.company_id
12 WHERE declined = '0'
13 GROUP BY company_id;
```

Output

| # | Time | Action | Message |
|---|----------|--|-------------------|
| 1 | 12:03:21 | CREATE VIEW VistaMarketing AS SELECT company_name, ph... | 0 row(s) affected |

Para poder visualizar la View le pedimos con select y en este caso en orden descendente.

```
14
15 • SELECT *
16 FROM VistaMarketing
17 ORDER BY promedio_compra DESC;
```

Result Grid

| | company_name | phone | country | promedio_compra |
|---|---------------------------|----------------|----------------|-----------------|
| ▶ | 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 |

VistaMarketing 3 x

Output

| # | Time | Action | Message |
|---|----------|---|---------------------|
| 2 | 12:04:02 | SELECT * FROM VistaMarketing WHERE country = '... | 8 row(s) returned |
| 3 | 12:05:43 | SELECT * FROM VistaMarketing ORDER BY promed... | 101 row(s) returned |

Ejercicio 3

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

The screenshot shows a SQL IDE interface with a toolbar at the top. The main editor displays SQL code for creating and querying a view named VistaMarketing. The code is as follows:

```
4
5 CREATE VIEW VistaMarketing AS
6 SELECT company_name,
7        phone,
8        country,
9        ROUND(AVG(amount),2) AS promedio_compra
10 FROM company c
11 JOIN transaction t ON c.id = t.company_id
12 WHERE declined = '0'
13 GROUP BY company_id;
14
15 SELECT *
16 FROM VistaMarketing
17 WHERE country = 'Germany'
18 ORDER BY promedio_compra DESC;
```

Below the editor, the 'Result Grid' tab is active, showing a table with 5 columns: company_name, phone, country, and promedio_compra. The table contains 5 rows of data, all from Germany.

| company_name | phone | country | promedio_compra |
|---------------------------|----------------|---------|-----------------|
| 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 |

At the bottom, the 'Output' tab shows the execution log. It contains two entries:

| # | Time | Action | Message |
|---|----------|--|---------------------|
| 3 | 12:05:43 | SELECT * FROM VistaMarketing ORDER BY promedio_compra DESC | 101 row(s) returned |
| 4 | 12:06:31 | SELECT * FROM VistaMarketing WHERE country = 'Germany' | 8 row(s) returned |

Nivel 3

Ejercicio 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:

Proceso de modificación de bases de datos:

- Crear la estructura para la nueva tabla: User
 - ⇒ En este caso la tenemos creada, la abrimos con "File" / "Open SQL Script" y la ejecutamos.
 - Primero se crea un índice en la columna user_id para hacer que las consultas sean más rápidas.
- Después se importan los registros de la tabla "datos_introducir_user(1)"

The screenshot shows a SQL IDE with a script editor and an output window. The script editor contains the following SQL commands:

```
-- Creamos la tabla user
CREATE INDEX idx_user_id ON transaction(user_id);
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)
);
```

The output window shows the execution results:

| # | Time | Action | Message | Duration / Fetch |
|-----|----------|--|---------------------|-------------------|
| 620 | 12:01:12 | INSERT INTO user (id, name, surname, phone, email, birth_date, ... | 1 row(s) affected | 0.000 sec |
| 621 | 12:01:26 | SELECT * FROM transactions.user LIMIT 0, 1000 | 275 row(s) returned | 0.000 sec / 0.000 |
| 622 | 12:02:27 | SET foreign_key_checks = 1 | 0 row(s) affected | 0.000 sec |

The screenshot shows a SQL IDE with a script editor and an output window. The script editor contains the following SQL commands:

```
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
  "2", "Garrett", "Mcconnel
  "3", "Ciaran", "Harrison"
  "4", "Howard", "Stafford"
  "5", "Hayfa", "Pierce", "
  "6", "Joel", "Tyson", "(7
  "7", "Rafael", "Jimenez",
  "8", "Nissim", "Franks",
  "9", "Mannix", "Mcclain",
  "10", "Robert", "Mccarthy
  "11", "Joan", "Baird", "(
  "12", "Benedict", "Wheele
  "13", "Allegra", "Stanton
  "14", "Sara", "Flynn", "1
  "15", "Noelani", "Patrick
  "16", "Eric", "Roth", "1-
```

The output window shows the execution results:

| # | Time | Action | Message | Duration / Fetch |
|-----|----------|--|---------------------|-------------------|
| 620 | 12:01:12 | INSERT INTO user (id, name, surname, phone, email, birth_date, ... | 1 row(s) affected | 0.000 sec |
| 621 | 12:01:26 | SELECT * FROM transactions.user LIMIT 0, 1000 | 275 row(s) returned | 0.000 sec / 0.000 |
| 622 | 12:02:27 | SET foreign_key_checks = 1 | 0 row(s) affected | 0.000 sec |

Para tener el mismo diagrama se tienen que modificar algunos campos de las tablas:

En tabla credit_card añadir nueva columna fecha_actual con tipo de dato DATE

```
ALTER TABLE credit_card
ADD fecha_actual DATE;
```

En tabla credit_card cambiamos el tipo de datos de los campos:

id a VARCHAR(20)

pin VARCHAR(4) */

```
8 • ALTER TABLE credit_card
9   MODIFY id VARCHAR(20) not null,
10  MODIFY pin VARCHAR(4) null default null;
```

En tabla company eliminar la columna website:

```
• ALTER TABLE company
  DROP COLUMN website;
```

Cambiar el nombre de la tabla user a data_user

```
17 • RENAME TABLE user to data_user;
18
```

En tabla data_user cambiar columna de email a personal_email:

```
20 • ALTER TABLE data_user
21   CHANGE email personal_email VARCHAR(150);
```

Crear el foreign key para cambiar la relación con la tabla transaction:

```
25 • ALTER TABLE transactions.credit_card
26   ADD CONSTRAINT fk_credit_card_transaction
27   FOREIGN KEY (id) REFERENCES transactions.transaction (credit_card_id)
28   ON DELETE RESTRICT -- No permite eliminar una tarjeta de crédito si hay transacciones relacionadas.
29   ON UPDATE CASCADE; -- Si el id de una tarjeta de crédito cambia, ese cambio se hace automáticamente a la tabla transaction.
```

Crear correctamente la foreign key de transaction:

The screenshot shows a SQL IDE with a script editor and an output window. The script editor contains the following SQL commands:

```
1
2 • ALTER TABLE data_user
3   DROP FOREIGN KEY data_user_ibfk_1;
4
5 • ALTER TABLE transaction
6   ADD FOREIGN KEY (user_id) REFERENCES data_user(id);
7
8
```

The output window shows the following log:

| # | Time | Action | Message | Duration / Fetch |
|-----|----------|---|---|------------------|
| 318 | 16:00:51 | INSERT INTO user (id, name, surname, phone, email, birth_date, country, ... | Error Code: 1146. Table 'transactions.user' doesn't exist | 0.000 sec |
| 319 | 16:03:09 | ALTER TABLE transaction ADD FOREIGN KEY (user_id) REFERENCES d... | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 | 0.016 sec |

Diagrama antes de las modificaciones:

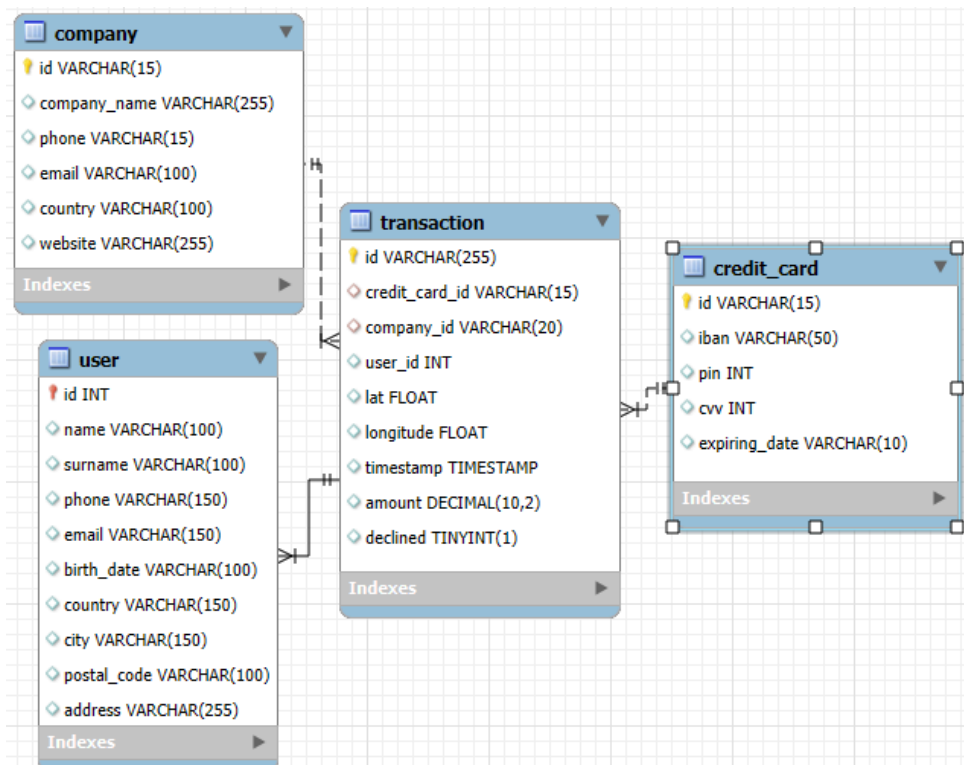
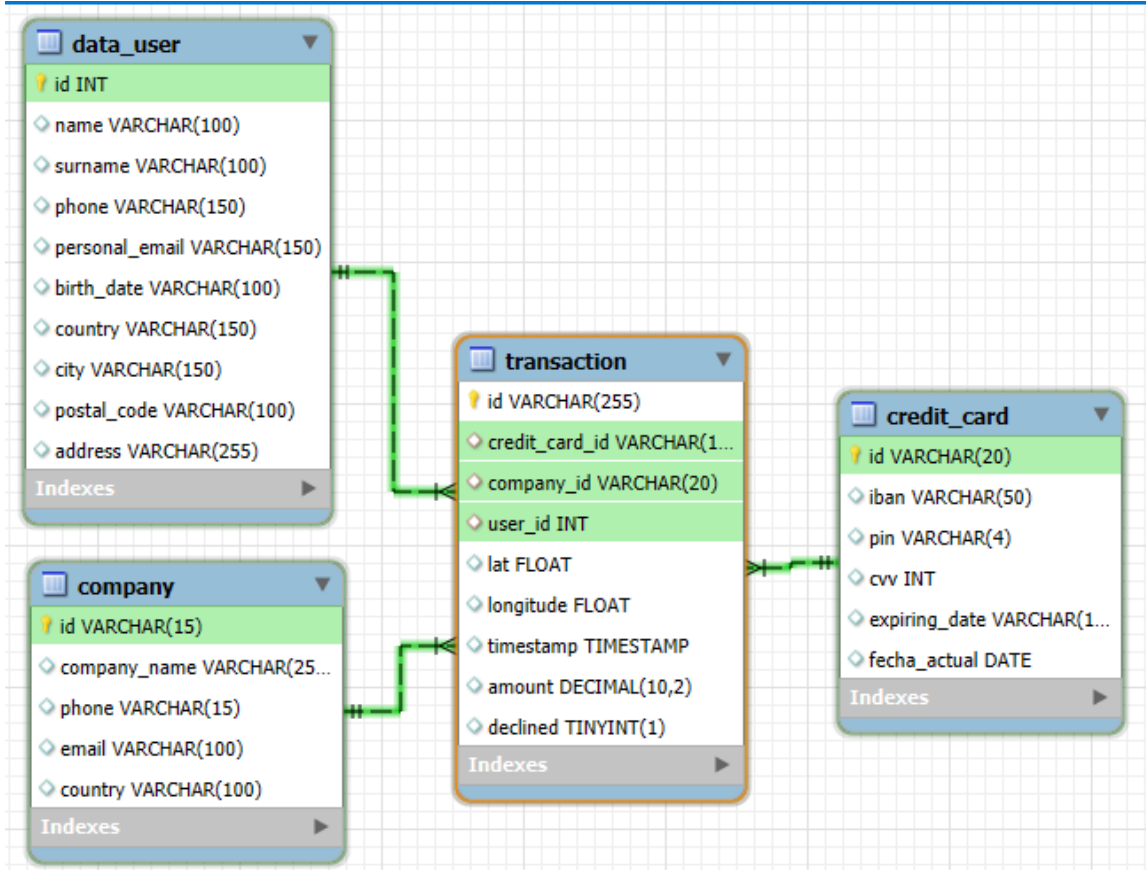


Diagrama después de las modificaciones:



Ejercicio 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.

⇒ Cuando generaba la vista me salía Null en el campo Nombre completo y email_contacto. Salido un error:

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`transactions`.`#sql-17e4_40`, CONSTRAINT `transaction_ibfk_2` FOREIGN KEY (`user_id`) REFERENCES `data_user` (`id`))

Esto quiere decir que hay un fallo. Entonces hay que crear el registro en la tabla data_user = 9999.

Lo comprobamos con buscando los user_id en transaction que no tengan un usuario correspondiente en la tabla data_user.

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query is:

```
1  
2 • SELECT user_id  
3 FROM transaction  
4 WHERE user_id NOT IN (SELECT id FROM data_user);  
5
```

The results pane shows a single row with the value 9999.

transaction 10 x

Output

Action Output

| # | Time | Action | Message |
|---|----------|--|-------------------|
| 1 | 16:40:27 | SELECT user_id FROM transaction WHERE user_id NOT IN (SELECT id F... | 1 row(s) returned |

Comprobamos que está el registro:

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query is:

```
3 FROM transaction  
4 WHERE user_id NOT IN (SELECT id FROM data_user);  
5  
6 • INSERT INTO data_user (id)  
7 VALUES ('9999');  
8  
9 • SELECT * FROM data_user WHERE id = '9999';  
10
```

The results pane shows a single row with the value 9999.

data_user 11 x

Output

Action Output

| # | Time | Action | Message |
|---|----------|---|-------------------|
| 2 | 16:42:10 | INSERT INTO data_user (id) VALUES ('9999') | 1 row(s) affected |
| 3 | 16:42:29 | SELECT * FROM data_user WHERE id = '9999' LIMIT 0, 1000 | 1 row(s) returned |

Procedemos a crear de nuevo la vista:

```

6 CREATE VIEW InformeTecnico AS
7 SELECT transaction.id AS ID_Transaccion,
8        CONCAT(data_user.name, " ", data_user.surname) AS 'Nombre Completo',
9        data_user.personal_email AS email_contacto,
10       credit_card.iban AS 'Numero Tarjeta',
11       transaction.amount AS 'Importe transaccion',
12       transaction.timestamp AS Fecha,
13       transaction.declined AS Estado,
14       company.company_name AS 'Nombre Empresa',
15       company.country AS 'Pais Empresa'
16 FROM transaction
17 LEFT JOIN data_user ON transaction.user_id = data_user.id
18 RIGHT JOIN credit_card ON transaction.credit_card_id = credit_card.id
19 LEFT JOIN company ON transaction.company_id = company.id;

```

| ID_Transaccion | Nombre Completo | email_contacto | Numero Tarjeta | Importe transaccion | Fecha | Es |
|--------------------------------------|-----------------|---------------------------------|--------------------------------|---------------------|---------------------|----|
| FE96CE47-8D59-381C-4E18-E3CA3D44E8FF | Kenyon Hartman | convallis.ante.lectus@yahoo.com | DO26854763748537475216568689 | 480.13 | 2021-06-15 00:26:29 | 1 |
| FE809ED4-2DB6-55AC-C915-929516E4646B | Molly Gilliam | donec@outlook.co.uk | SE2813123487163628531121 | 219.83 | 2021-11-09 21:35:40 | 0 |
| FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65 | Linus Willis | ultrices.posuere@yahoo.co.uk | KW9485332754781757886242955643 | 42.32 | 2021-06-13 11:41:17 | 0 |
| FE80951B-AE8D-77DC-F450-B8083EBD3187 | Hilda Leary | et libero@yahoo.co.uk | KW9485332754781757886242955643 | 42.32 | 2021-06-13 11:41:17 | 0 |

Output: Action Output

| # | Time | Action | Message | Duration / Fets |
|---|----------|--|-------------------|-----------------|
| 6 | 16:48:35 | CREATE VIEW InformeTecnico AS SELECT transaction.id AS ID_Transac... | 0 row(s) affected | 0.015 sec |

Esta vista llamada InformeTecnico se utiliza para mostrar información importante de las transacciones en un formato más sencillo y comprensible. Combina datos de todas las tablas que tenemos ahora: transaction, data_user, credit_card y company.

Se usa LEFT JOIN para las tablas data_user y company porque queremos mostrar todas las transacciones, incluso si no hay un usuario o empresa asociados. Esto garantiza que las transacciones aparezcan, aunque falten datos de usuario o empresa.

En cambio, se usó RIGHT JOIN para credit_card porque queremos asegurarnos de que todas las tarjetas de crédito involucradas aparezcan, incluso si no están vinculadas a una transacción específica. Esto es útil cuando queremos que todos los registros de tarjetas estén presentes, sin importar si han sido usadas en una transacción o no.

Y visualizamos:

```

21
22 • SELECT *
23 FROM InformeTecnico
24 ORDER BY ID_Transaccion DESC;
25
26

```

| ID_Transaccion | Nombre Completo | email_contacto | Numero Tarjeta | Importe transaccion | Fecha | Estado |
|--------------------------------------|-----------------|---------------------------------|---------------------------------|---------------------|---------------------|--------|
| FE96CE47-8D59-381C-4E18-E3CA3D44E8FF | Kenyon Hartman | convallis.ante.lectus@yahoo.com | DO26854763748537475216568689 | 480.13 | 2021-06-15 00:26:29 | 1 |
| FE809ED4-2DB6-55AC-C915-929516E46468 | Molly Giliam | donec@outlook.couk | SE2813123487163628531121 | 219.83 | 2021-11-09 21:35:40 | 0 |
| FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65 | Linus Willis | ultrices.posuere@yahoo.couk | KW9485332754781757886242955643 | 42.32 | 2021-06-13 11:41:17 | 0 |
| FD89D51B-AE8D-77DC-E450-B8083FBD3187 | Hilda Levy | et.libero@yahoo.org | LT053237077744561475 | 200.72 | 2022-03-16 02:35:05 | 0 |
| FD2E8957-414B-8EEC-E9AD-59AA7A8A6290 | Hedwig Gilbert | sem.eget@icloud.edu | GE84848451582810541526 | 78.29 | 2022-03-13 00:27:34 | 0 |
| FCE2A89A-271D-2BDC-9E49-8DD92A373391 | Hakeem Alford | adpiscing.ligula@google.edu | MD1234119525145401270486 | 335.56 | 2022-02-06 22:48:41 | 0 |
| FBD7E0D6-BA68-F5BC-0CA9-EA4B8760100C | Hedwig Gilbert | sem.eget@icloud.edu | MU4132333444534342541344788855 | 207.09 | 2021-04-29 14:17:50 | 1 |
| FAC76A80-8448-69AA-E892-426C2F12621C | Slade Poole | amet@icloud.com | MT05JWCF58868200575771634583813 | 304.95 | 2021-05-30 21:10:55 | 0 |
| FAAD3FFC-1A17-E141-43D3-359A5BA7CB3B | Hedwig Gilbert | sem.eget@icloud.edu | GE90157928843338134463 | 149.84 | 2021-10-24 20:16:23 | 0 |
| FA053936-75D8-85FA-490D-9B624E1B920A | Hedwig Gilbert | sem.eget@icloud.edu | GT02497653655330848247645975 | 151.32 | 2021-07-06 10:18:35 | 0 |
| F85A7D75-2778-9D75-D776-3F41A828DE88 | Sarah Beck | vitae.risus@aol.couk | VG1468087984174645729577 | 135.93 | 2021-10-10 01:46:36 | 0 |

Output

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|---------------------|---------------------|
| 6 | 16:48:35 | CREATE VIEW InformeTecnico AS SELECT transaction.id AS ID_Transaccion, | 0 row(s) affected | 0.015 sec |
| 7 | 16:48:43 | SELECT * FROM InformeTecnico ORDER BY ID_Transaccion DESC LIMIT 0, 1000 | 587 row(s) returned | 0.031 sec / 0.000 s |

Al ver la columna Estado, me pareció que podría estar mejor visualizada con “Aprobada” o “Declinada”. Modifico la vista, usando Create or replace view y la columna Estado con la función CASE:

```

177
178 • CREATE OR REPLACE VIEW InformeTecnico AS
179 SELECT transaction.id AS ID_Transaccion,
180        CONCAT(data_user.name, " ", data_user.surname) AS 'Nombre Completo',
181        data_user.personal_email AS email_contacto,
182        credit_card.iban AS 'Numero Tarjeta',
183        transaction.amount AS 'Importe transaccion',
184        transaction.timestamp AS Fecha,
185        CASE
186            WHEN transaction.declined = 0 THEN 'Aprobada'
187            WHEN transaction.declined = 1 THEN 'Declinada'
188        END AS Estado,
189        company.company_name AS 'Nombre Empresa',
190        company.country AS 'Pais Empresa'
191 FROM transaction
192 LEFT JOIN data_user ON transaction.user_id = data_user.id
193 RIGHT JOIN credit_card ON transaction.credit_card_id = credit_card.id
194 LEFT JOIN company ON transaction.company_id = company.id;
195

```

| ID_Transaccion | Nombre Completo | email_contacto | Numero Tarjeta | Importe transaccion | Fecha | Estado | Nomb |
|--------------------------------------|-----------------|---------------------------------|--------------------------------|---------------------|---------------------|-----------|--------|
| FE96CE47-8D59-381C-4E18-E3CA3D44E8FF | Kenyon Hartman | convallis.ante.lectus@yahoo.com | DO26854763748537475216568689 | 480.13 | 2021-06-15 00:26:29 | Declinada | Magna |
| FE809ED4-2DB6-55AC-C915-929516E46468 | Molly Giliam | donec@outlook.couk | SE2813123487163628531121 | 219.83 | 2021-11-09 21:35:40 | Aprobada | Nunc I |
| FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65 | Linus Willis | ultrices.posuere@yahoo.couk | KW9485332754781757886242955643 | 42.32 | 2021-06-13 11:41:17 | Aprobada | Nunc I |

InformeTecnico 3 x

Output

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|---------------------|---------------------|
| 2 | 10:47:37 | CREATE OR REPLACE VIEW InformeTecnico AS SELECT transaction.id AS ID_Transaccion, | 0 row(s) affected | 0.047 sec |
| 3 | 10:47:56 | SELECT * FROM InformeTecnico ORDER BY ID_Transaccion DESC LIMIT 0, 1000 | 587 row(s) returned | 0.015 sec / 0.000 s |

