

Tarea S3.01. Manipulación de tablas

En este sprint, se simula una situación empresarial en la que debes realizar varias manipulaciones en las tablas de la base de datos. A su vez, tendrás que trabajar con índices y vistas. En esta actividad, continuarás trabajando con la base de datos que contiene información de una empresa dedicada a la venta de productos en línea. En esta tarea, empezarás a trabajar con información relacionada con tarjetas de crédito.

Nivel 1

- Ejercicio 1

Tu tarea es diseñar y crear una tabla llamada "credit_card" que almacene detalles cruciales sobre las tarjetas de crédito. La nueva tabla debe ser capaz de identificar de manera única cada tarjeta y establecer una relación adecuada con las otras dos tablas ("transaction" y "company"). Después de crear la tabla, será necesario que ingreses la información del documento denominado "dades_introducir_credit". Recuerda mostrar el diagrama y realizar una breve descripción del mismo.

```
1  -- 1. Primero, crea la tabla credit_card
2  -- Creamos la tabla credit_card
3  CREATE TABLE IF NOT EXISTS credit_card (
4      id VARCHAR(15) PRIMARY KEY,
5      iban VARCHAR(34),
6      pan VARCHAR(19),
7      pin VARCHAR(6),
8      cvv VARCHAR(4),
9      expiring_date VARCHAR(10));
10
11 -- Ahora antes de hacer un ALTER TABLE transaction, lo primero que tengo que hacer es
12 -- correr el archivo datos_introducir_credit.sql para llenar la tabla credit_card
13 -- Luego, en la tabla transaction, se realiza el ALTER TABLE para agregar la clave foránea que relaciona credit_card_id con credit_card(id):
14 ALTER TABLE transaction
15 ADD CONSTRAINT fk_credit_card
16 FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
17
```

| # | Time | Action | Message | Duration / Fetch |
|-----|----------|---|-------------------|------------------|
| 306 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4835', 'PT34592... | 1 row(s) affected | 0.047 sec |
| 307 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4842', 'SA21567... | 1 row(s) affected | 0.109 sec |
| 308 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4849', 'SE28131... | 1 row(s) affected | 0.047 sec |
| 309 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4856', 'TR3738... | 1 row(s) affected | 0.094 sec |

```
1
2  -- Insertamos datos de credit_card
3  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2938', 'TR301950312213576817638661', '5424465566813633', '3257
4  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2945', 'D026854763748537475216568689', '5142423821948828', '98
5  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2952', 'B645IVQL52710525608255', '4556 453 55 5287', '4598', '
6  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2959', 'CR7242477244335841535', '372461377349375', '3583', '66
7  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2966', 'B672LKTQ15627628377363', '448566 886747 7265', '4900',
8  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2973', 'PT87806228135092429456346', '544 58654 54343 384', '87
9  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2980', 'DE39241881883086277136', '402400 7145845969', '5075',
10 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2987', 'GE89681434837748781813', '3763 747687 76666', '2298',
11 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2994', 'BH62714428368066765294', '344283273252593', '7545', '5
12 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3001', 'CY49087426654774581266832110', '511722 924833 2244', '
13 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3008', 'LU507216693616119230', '4485744644433884', '1856', '74
14 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3015', 'PS119398216295715068342456821', '3784 662233 17389', '

```

| # | Time | Action | Message | Duration / Fetch |
|-----|----------|---|-------------------|------------------|
| 303 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4814', 'MR4845... | 1 row(s) affected | 0.063 sec |
| 304 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4821', 'LT25314... | 1 row(s) affected | 0.062 sec |
| 305 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4828', 'BG111IL... | 1 row(s) affected | 0.047 sec |
| 306 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4835', 'PT34592... | 1 row(s) affected | 0.047 sec |
| 307 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4842', 'SA21567... | 1 row(s) affected | 0.109 sec |
| 308 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4849', 'SE28131... | 1 row(s) affected | 0.047 sec |
| 309 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4856', 'TR3738... | 1 row(s) affected | 0.094 sec |

-- Verificamos que se hayan ingresado los datos en la tabla credit_card

SELECT * FROM credit_card;

```
12 -- Luego, en la tabla transaction, se realiza el ALTER TABLE para agregar la clave foránea que re
13
14 -- Verificamos que se hayan ingresado los datos en la tabla credit_card
15 • SELECT * FROM credit_card;
16
```

| | id | iban | pan | pin | cvv | expiring_date |
|---|----------|------------------------------|---------------------|------|-----|---------------|
| ▶ | CcU-2938 | TR301950312213576817638661 | 5424465566813633 | 3257 | 984 | 10/30/22 |
| | CcU-2945 | DO26854763748537475216568689 | 5142423821948828 | 9080 | 887 | 08/24/23 |
| | CcU-2952 | BG45IVQL52710525608255 | 4556 453 55 5287 | 4598 | 438 | 06/29/21 |
| | CcU-2959 | CR7242477244335841535 | 372461377349375 | 3583 | 667 | 02/24/23 |
| | CcU-2966 | BG72LKTQ15627628377363 | 448566 886747 7265 | 4900 | 130 | 10/29/24 |
| | CcU-2973 | PT87806228135092429456346 | 544 58654 54343 384 | 8760 | 887 | 01/30/25 |
| | CcU-2980 | DE39241881883086277136 | 402400 7145845969 | 5075 | 596 | 07/24/22 |
| | CcU-2987 | GE89681434837748781813 | 3763 747687 76666 | 2298 | 797 | 10/31/23 |

credit_card 2 x

Output

| # | Time | Action | Message |
|-------|----------|---|---------------------|
| ✓ 305 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4828', 'BG11IL... | 1 row(s) affected |
| ✓ 306 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4835', 'PT34592... | 1 row(s) affected |
| ✓ 307 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4842', 'SA21567... | 1 row(s) affected |
| ✓ 308 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4849', 'SE28131... | 1 row(s) affected |
| ✓ 309 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4856', 'TR3738... | 1 row(s) affected |
| ✓ 310 | 10:55:32 | SELECT * FROM credit_card | 275 row(s) returned |

Ahora se creará la tabla user, antes de ello creamos el Índice: idx_user_id

```
1 -- Creamos la tabla user
2
3 • CREATE INDEX idx_user_id ON transaction(user_id);
4
```

Output

| # | Time | Action | Message |
|-------|----------|---|--|
| ✓ 296 | 10:49:04 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4765', 'SA28887... | 1 row(s) affected |
| ✓ 297 | 10:49:04 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4772', 'ME5983... | 1 row(s) affected |
| ✓ 298 | 10:49:04 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4779', 'FI91092... | 1 row(s) affected |
| ✓ 299 | 10:49:04 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4786', 'SI51703... | 1 row(s) affected |
| ✓ 300 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4793', 'HU9521... | 1 row(s) affected |
| ✓ 301 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4800', 'SI97824... | 1 row(s) affected |
| ✓ 302 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4807', 'LB19298... | 1 row(s) affected |
| ✓ 303 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4814', 'MR4845... | 1 row(s) affected |
| ✓ 304 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4821', 'LT25314... | 1 row(s) affected |
| ✓ 305 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4828', 'BG11IL... | 1 row(s) affected |
| ✓ 306 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4835', 'PT34592... | 1 row(s) affected |
| ✓ 307 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4842', 'SA21567... | 1 row(s) affected |
| ✓ 308 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4849', 'SE28131... | 1 row(s) affected |
| ✓ 309 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4856', 'TR3738... | 1 row(s) affected |
| ✓ 310 | 10:55:32 | SELECT * FROM credit_card | 275 row(s) returned |
| ✓ 311 | 10:58:49 | ALTER TABLE transaction ADD CONSTRAINT fk_credit_card FOREIGN KEY (credit_card_id) REFE... | 587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0 |
| ✓ 312 | 11:37:04 | CREATE INDEX idx_user_id ON transaction(user_id) | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 |

Ahora creamos la tabla: user

```
5 -- Ahora creamos la tabla user
6 CREATE TABLE IF NOT EXISTS user (
7     id INT PRIMARY KEY,
8     name VARCHAR(100),
9     surname VARCHAR(100),
10    phone VARCHAR(150),
11    email VARCHAR(150),
12    birth_date VARCHAR(100),
13    country VARCHAR(150),
14    city VARCHAR(150),
15    postal_code VARCHAR(100),
16    address VARCHAR(255),
17    FOREIGN KEY(id) REFERENCES transaction(user_id)
18 );
19
```

Output

| # | Time | Action | Message |
|-----|----------|--|--|
| 309 | 10:49:05 | INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4856', 'TR37387255831354... | 1 row(s) affected |
| 310 | 10:55:32 | SELECT * FROM credit_card | 275 row(s) returned |
| 311 | 10:58:49 | ALTER TABLE transaction ADD CONSTRAINT fk_credit_card FOREIGN KEY (credit_card_id) REFERENCES cre... | 587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0 |
| 312 | 11:37:04 | CREATE INDEX idx_user_id ON transaction(user_id) | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 |
| 313 | 11:41:27 | CREATE TABLE IF NOT EXISTS user (id INT PRIMARY KEY, name VARCHAR(100), surname V... | 0 row(s) affected |

Ahora insertamos registros en la tabla user

Se ejecuta el archivo datos_introducir_user (1).sql, que **inserta registros** en la tabla user.”

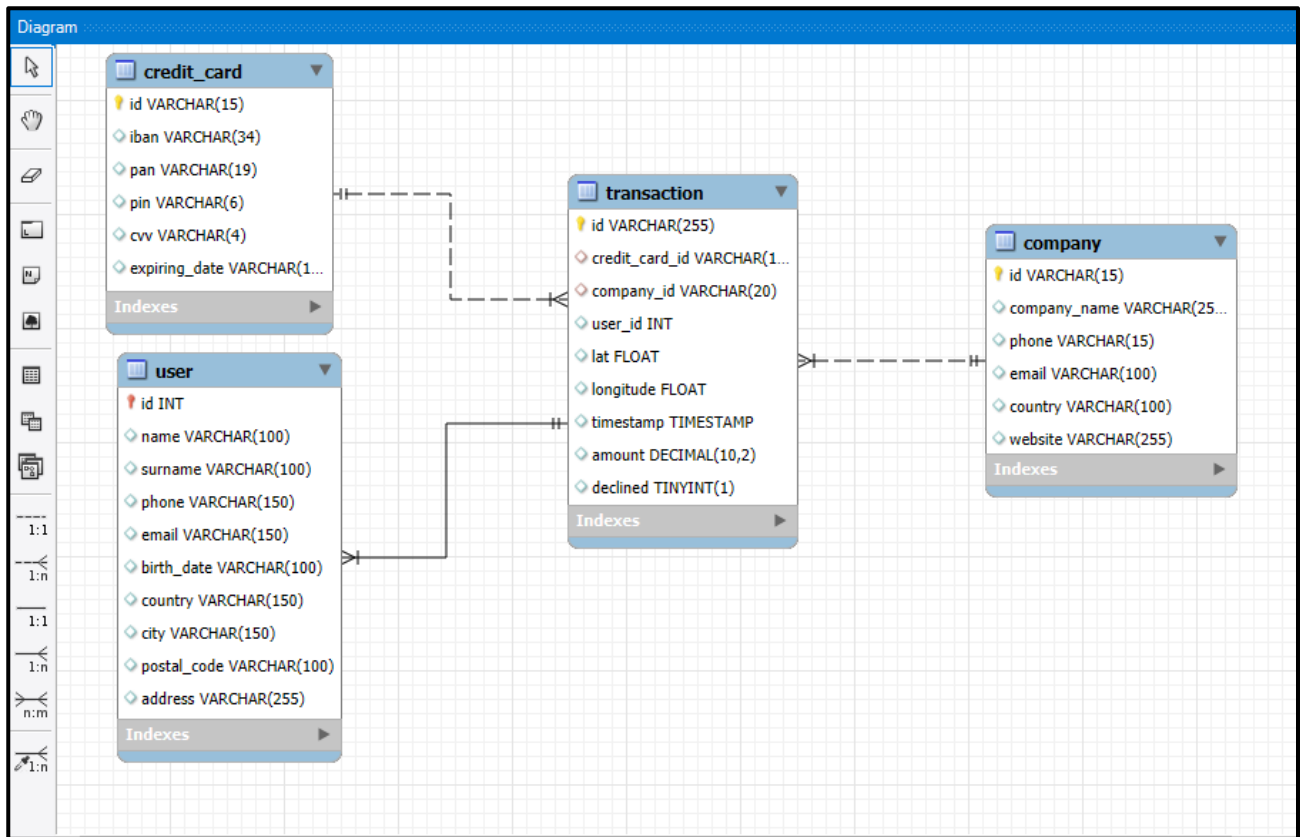
El archivo datos_introducir_user (1).sql contiene las sentencias necesarias para insertar registros en la tabla user. **Se desactivan temporalmente las verificaciones de claves foráneas con SET foreign_key_checks = 0 ; para permitir una carga segura de los datos, y se reactivan al final con SET foreign_key_checks = 1 ; para asegurar la integridad referencial del sistema.**

```
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 ( "1", "Zeus", "Gamble", "1-282-58
5 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "2", "Garrett", "Mcconnell", "(7
6 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "3", "Ciaran", "Harrison", "(522
7 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "4", "Howard", "Stafford", "1-41
8 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "5", "Hayfa", "Pierce", "1-554-5
9 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "6", "Joel", "Tyson", "(718) 288
10 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "7", "Rafael", "Jimenez", "(817)
11 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "8", "Nissim", "Franks", "(692)
12 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "9", "Mannix", "Mcclain", "(590)
13 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "10", "Robert", "Mccarthy", "(32
14 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "11", "Joan", "Baird", "(981) 42
15 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "12", "Benedict", "Wheeler", "1-
16 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "13", "Allegra", "Stanton", "1-9
```

Output

| # | Time | Action | Message | Duration / Fetch |
|-----|----------|--|-------------------|------------------|
| 586 | 11:51:25 | INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (... | 1 row(s) affected | 0.125 sec |
| 587 | 11:51:25 | INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (... | 1 row(s) affected | 0.047 sec |
| 588 | 11:51:25 | INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (... | 1 row(s) affected | 0.047 sec |
| 589 | 11:51:25 | INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (... | 1 row(s) affected | 0.062 sec |
| 590 | 11:51:25 | SET foreign_key_checks = 1 | 0 row(s) affected | 0.000 sec |

Descripción breve del diagrama



Tablas de la base de datos transactions

En el diagrama se pueden ver **cuatro tablas**: **transaction**, **credit_card**, **user** y **company**. A continuación, se explica cada una:

1) **Transaction**

- **Es la tabla principal** que registra todas las transacciones realizadas.
- **Campos clave:**
 - **id** → identificador único (clave primaria).
 - **credit_card_id** → relaciona cada transacción con una tarjeta de crédito.
 - **company_id** → indica en qué empresa se hizo la transacción.
 - **user_id** → indica qué usuario la realizó.
 - **lat, longitude** → coordenadas geográficas.
 - **timestamp** → fecha y hora.
 - **amount** → monto de la transacción.
 - **declined** → si fue rechazada o no.
- **Relaciones:**
 - **credit_card_id** → FK hacia **credit_card(id)**
 - **company_id** → FK hacia **company(id)**
 - **user_id** → FK hacia **user(id)**

Es el centro del diagrama, desde donde parten las relaciones hacia las otras tablas.

2) `credit_card`

- Almacena los datos de las tarjetas usadas en las transacciones.
- **Campos clave:**
 - `id` (clave primaria)
 - `iban`, `pan`, `pin`, `cvv`, `expiring_date`
- **Relación:**
 - Su campo `id` es referenciado desde `transaction(credit_card_id)`

Tiene una relación 1:N con `transaction`:

Una tarjeta puede tener **muchas transacciones**.

3) `user`

- Contiene la información personal del usuario.
- **Campos clave:**
 - `id` (clave primaria)
 - `name`, `surname`, `email`, `phone`, etc.
- **Relación:**
 - Su campo `id` es referenciado desde `transaction(user_id)`

Relación 1:N con `transaction`:

Un usuario puede hacer **muchas transacciones**.

4) `company`

- Representa las empresas donde se realizaron las compras.
- **Campos clave:**
 - `id` (clave primaria)
 - `name`, `industry`, `country`, etc.
- **Relación:**
 - Su campo `id` es referenciado desde `transaction(company_id)`

Relación 1:N con `transaction`:

Una empresa puede recibir **muchas transacciones**.

Descripción del diagrama

El diagrama muestra la estructura de la base de datos `Transaction`, compuesta por cuatro tablas principales. La tabla `transaction` funciona como núcleo, ya que se conecta con `user`, `credit_card` y `company` a través de claves foráneas. Las relaciones son todas de tipo 1:N, es decir, un usuario, tarjeta o empresa puede estar vinculada a múltiples transacciones. Esta estructura permite un registro completo de cada operación financiera, incluyendo quién la hizo, con qué tarjeta, en qué empresa y bajo qué coordenadas geográficas.”

La tabla `credit_card` almacena información esencial de cada tarjeta de crédito utilizada por los usuarios para realizar transacciones. Esta tabla tiene una relación directa con la tabla `transaction`, ya que cada transacción está vinculada a una tarjeta específica.

Estructura de la tabla `credit_card`:

| Campo | Tipo | Descripción |
|---------------|-------------|---|
| id | VARCHAR(15) | Identificador único de la tarjeta (PK) |
| iban | VARCHAR(34) | Número IBAN asociado a la cuenta |
| pan | VARCHAR(19) | Número de la tarjeta (Primary Account Number) |
| pin | VARCHAR(6) | Número secreto de seguridad |
| cvv | VARCHAR(4) | Código de verificación de la tarjeta |
| expiring_date | VARCHAR(10) | Fecha de vencimiento de la tarjeta |

La clave primaria es el campo `id`, y este se utiliza como **clave foránea** en la tabla `transaction`, asegurando que cada transacción esté asociada a una tarjeta válida.

- Ejercicio 2

El departamento de Recursos Humanos ha identificado un error en el número de cuenta del usuario con ID CcU-2938. La información que debe mostrarse para este registro es: R323456312213576817699999. Recuerda mostrar que el cambio se realizó.

--Primero, actualizamos el campo `iban` en la tabla `credit_card` para ese usuario (con ID CcU-2938):

```
23 -- La información que debe mostrarse para este registro es: R323456312213576817699999. Recuerda mostrar que el cambio se realizó.
24 |-- Primero, actualizamos el campo iban en la tabla credit_card para ese usuario (con ID CcU-2938):
25 • SELECT * FROM credit_card WHERE ID = 'CcU-2938';
26 -- Se actualiza el registro
27 • UPDATE credit_card
28   SET iban = 'R323456312213576817699999'
29   WHERE id = 'CcU-2938';
30 -- Luego, se verifica que el cambio se realizó correctamente:
31 • SELECT id, iban
32   FROM credit_card
33   WHERE id = 'CcU-2938';
34
```

Result Grid

| id | iban |
|----------|---------------------------|
| CcU-2938 | R323456312213576817699999 |
| NULL | NULL |

credit_card 11 x

Output

Action Output

| # | Time | Action | Message |
|-----|----------|---|--|
| 599 | 13:19:22 | SELECT * FROM credit_card | 275 row(s) returned |
| 600 | 13:20:44 | UPDATE credit_card SET iban = 'R323456312213576817699999' WHERE id = 'CcU-2938' | 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 |
| 601 | 13:20:56 | SELECT id, iban FROM credit_card WHERE id = 'CcU-2938' | 1 row(s) returned |

- Ejercicio 3

En la tabla "transaction" ingresa un nuevo usuario con la siguiente información:


| | |
|----------------|--------------------------------------|
| 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 |

```

83      -- Importante: Antes de ejecutar, tenemos que asegurarnos: Verificar si existe la tarjeta 'CcU-9999' en credit_card:
84      SELECT * FROM credit_card WHERE id = 'CcU-9999';
85      -- Verificar si existe la compañía 'b-9999' en company:
86      SELECT * FROM company WHERE id = 'b-9999';
87      -- Verificar si existe el usuario con id = 9999 en user:
88      SELECT * FROM user WHERE id = 9999;
89      -- **SE REALIZARON LAS 3 CONSULTAS Y NO EXISTEN**, al no existir, lo primero que tenemos que hacer es insertar registros
90      -- 1. Insertar tarjeta CcU-9999:
91      INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)
92      VALUES ('CcU-9999', 'TR309999312213576817699999', '1234567812345678', '1234', '321', '12/31/27');
93      -- 2. Insertar compañía b-9999:
94      INSERT INTO company (id, company_name, phone, email, country, website)
95      VALUES ('b-9999', 'Kaleacuy Terapias Rehabilitacion', '34 61 39 78 291', 'manuelsoftware@gmail.com', 'España', 'https://instagram.com/site');
96      -- 3. Insertar usuario 9999:
97      -- primero verificamos...
98      SELECT CONSTRAINT_NAME
99      FROM information_schema.KEY_COLUMN_USAGE
100     WHERE TABLE_NAME = 'user'
101           AND REFERENCED_TABLE_NAME = 'transaction';
102      -- da como resultado: 'user_ibfk_1'
103      -- Ahora sí, eliminamos el constraint mal puesta:
104      ALTER TABLE user DROP FOREIGN KEY user_ibfk_1;
105
106      INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address)
107      VALUES (9999, 'Manel', 'Capitan', '034-613-9782', 'nuevo.usuario@example.com', 'Mar 18, 1974', 'España', 'Barcelona', '08031', 'Fabra i Puig 123');
108

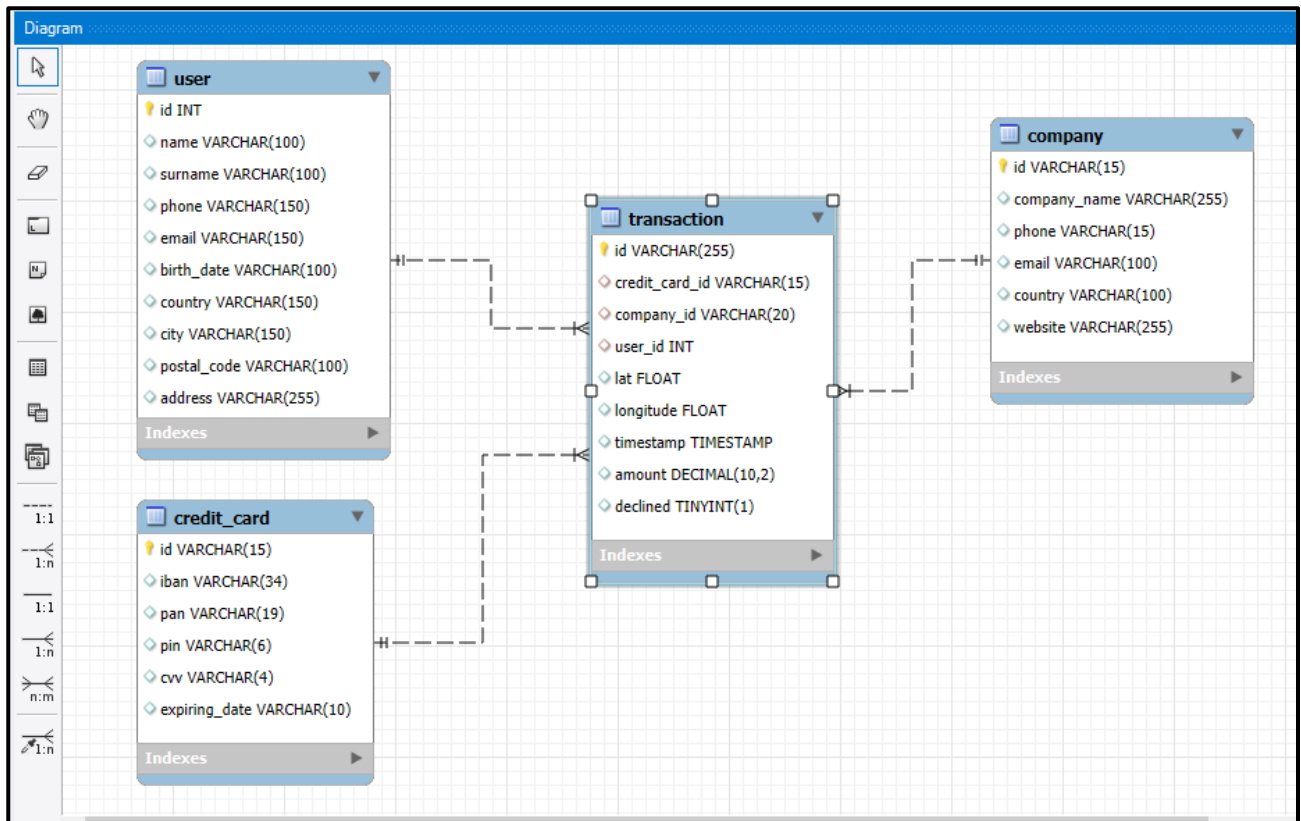
```

Output :


 Action Output

| | # | Time | Action | Message |
|---|-----|----------|--|--------------------------------|
| ✓ | 631 | 10:17:42 | SELECT CONSTRAINT_NAME FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_NAME = 'u... | 1 row(s) returned |
| ✓ | 632 | 10:21:15 | ALTER TABLE user DROP FOREIGN KEY user_ibfk_1 | 0 row(s) affected Records: 0 D |

Diagrama



-- Ahora si ingresaremos al nuevo usuario con "id": '9999'

```

98 • SELECT CONSTRAINT_NAME
99   FROM information_schema.KEY_COLUMN_USAGE
100  WHERE TABLE_NAME = 'user'
101        AND REFERENCED_TABLE_NAME = 'transaction';
102  -- da como resultado: 'user_ibfk_1'
103  -- Ahora sí, eliminamos el constraint mal puesta:
104 • ALTER TABLE user DROP FOREIGN KEY user_ibfk_1;
105
106  -- Ahora si ingresaremos al nuevo usuario con "id": '9999'
107 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address)
108  VALUES (9999, 'Manel', 'Capitan', '034-613-9782', 'nuevo.usuario@example.com', 'Mar 18, 1974', 'España', 'Barcelona', '08031', ' ');
109
110
111
112
113
114
115
  
```

| # | Time | Action | Message |
|-----|----------|---|--|
| 631 | 10:17:42 | SELECT CONSTRAINT_NAME FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_NAME = 'u... | 1 row(s) returned |
| 632 | 10:21:15 | ALTER TABLE user DROP FOREIGN KEY user_ibfk_1 | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 |
| 633 | 10:29:56 | INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (99... | 1 row(s) affected |

-- Recién podremos insertar en transaction para resolver el Nivel 1 ejercicio 3

```
111 -- Recién podremos insertar en transaction para resolver el Nivel 1 ejercicio 3
112 • INSERT INTO transaction (
113     id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined
114 ) VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, NOW(), 111.11, 0);
115
116 --
117
```

Output

Action Output

| # | Time | Action | Message | Dura |
|-----|----------|---|-------------------|------|
| 633 | 10:29:56 | INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (99... | 1 row(s) affected | 0.31 |
| 634 | 10:37:46 | SELECT t.id AS trans_id, t.amount, t.declined, u.name AS user_name, u.surname AS user_surname, c.company_n... | 0 row(s) returned | 0.01 |
| 635 | 10:40:05 | INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined)... | 1 row(s) affected | 0.17 |

-- Verificamos el ingreso del nuevo usuario en la tabla transaction con la información brindada:

```
116 -- Verificamos el ingreso del nuevo usuario en la tabla transaction con la información brindada:
117 • SELECT * FROM transaction WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD';
118
119
120
121
```

Result Grid

| | 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 | 2025-05-08 10:40:05 | 111.11 | 0 |
| * | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

transaction 37 x

Output

Action Output

| # | Time | Action | Message |
|-----|----------|---|-------------------|
| 640 | 10:59:30 | SELECT * FROM transaction WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD' | 1 row(s) returned |

- Ejercicio 4

Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla credit_card. Recuerda mostrar el cambio realizado.

```
119  /* - Ejercicio 4
120      Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla credit_card.
121      Recuerda mostrar el cambio realizado.*/
122      -- ** primero capturamos la pantalla con las columnas actuales
123  •  DESCRIBE credit_card;
124
```

Result Grid

| Field | Type | Null | Key | Default | Extra |
|---------------|-------------|------|-----|---------|-------|
| id | varchar(15) | NO | PRI | NULL | |
| iban | varchar(34) | YES | | NULL | |
| pan | varchar(19) | YES | | NULL | |
| pin | varchar(6) | YES | | NULL | |
| cvv | varchar(4) | YES | | NULL | |
| expiring_date | varchar(10) | YES | | NULL | |

Result 47 x

Output

Action Output

| # | Time | Action | Message |
|-----|----------|----------------------|-------------------|
| 650 | 11:23:00 | DESCRIBE credit_card | 6 row(s) returned |
| 651 | 11:25:51 | DESCRIBE credit_card | 6 row(s) returned |

-- Se elimina la columna pan según lo solicitado en el ejercicio 4

```
119  /* - Ejercicio 4
120      Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla credit_card.
121      Recuerda mostrar el cambio realizado.*/
122      -- ** primero capturamos la pantalla con las columnas actuales
123  •  DESCRIBE credit_card;
124
125      -- Se eliminará la columna pan según lo solicitado en el ejercicio 4
126  •  ALTER TABLE credit_card DROP COLUMN pan;
```

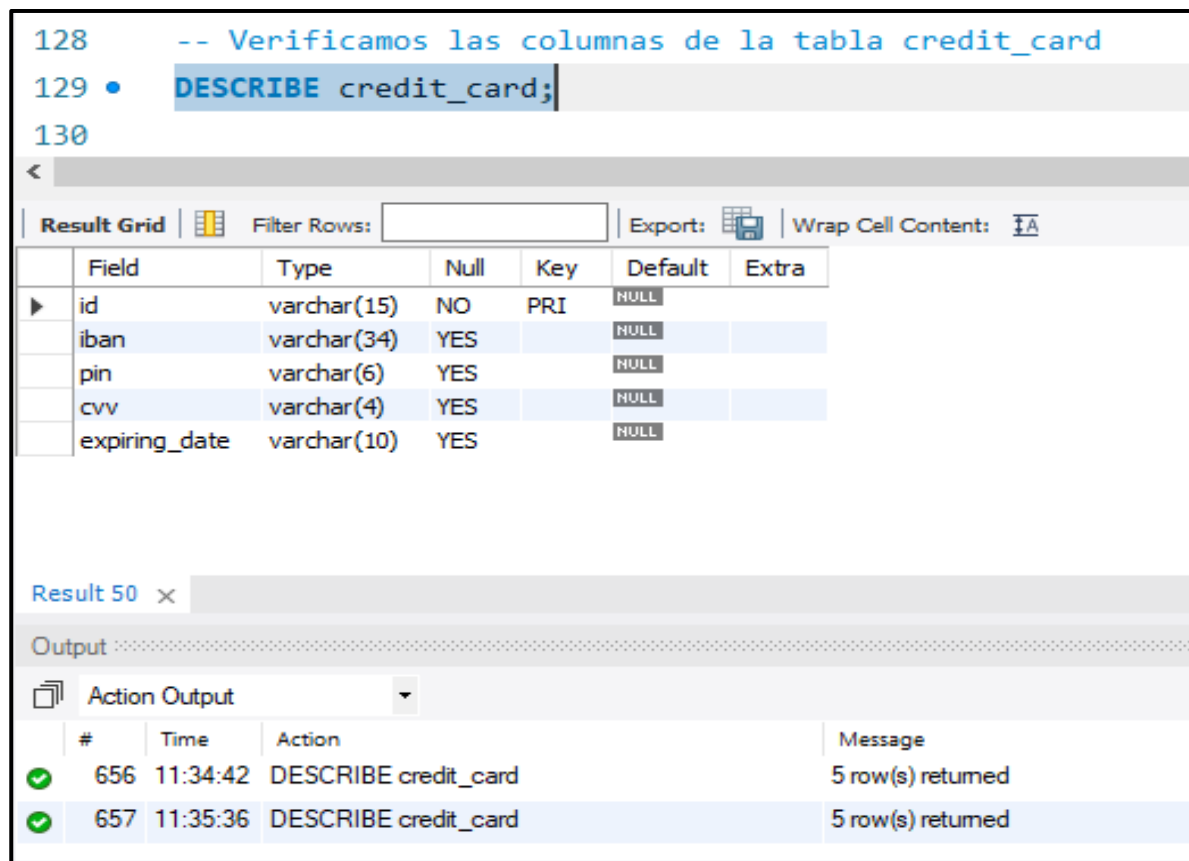
Output

Action Output

| # | Time | Action | Message |
|-----|----------|---|--|
| 651 | 11:25:51 | DESCRIBE credit_card | 6 row(s) returned |
| 652 | 11:30:15 | ALTER TABLE credit_card DROP COLUMN pan | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 |

-- Verificamos las columnas de la tabla credit_card, en la cual ya no se visualiza la columna 'pan', según lo solicitado

```
128 -- Verificamos las columnas de la tabla credit_card
129 • DESCRIBE credit_card;
130
```



| Field | Type | Null | Key | Default | Extra |
|---------------|-------------|------|-----|---------|-------|
| id | varchar(15) | NO | PRI | NONE | |
| iban | varchar(34) | YES | | NONE | |
| pin | varchar(6) | YES | | NONE | |
| cvv | varchar(4) | YES | | NONE | |
| expiring_date | varchar(10) | YES | | NONE | |

Result 50 x

Output

Action Output

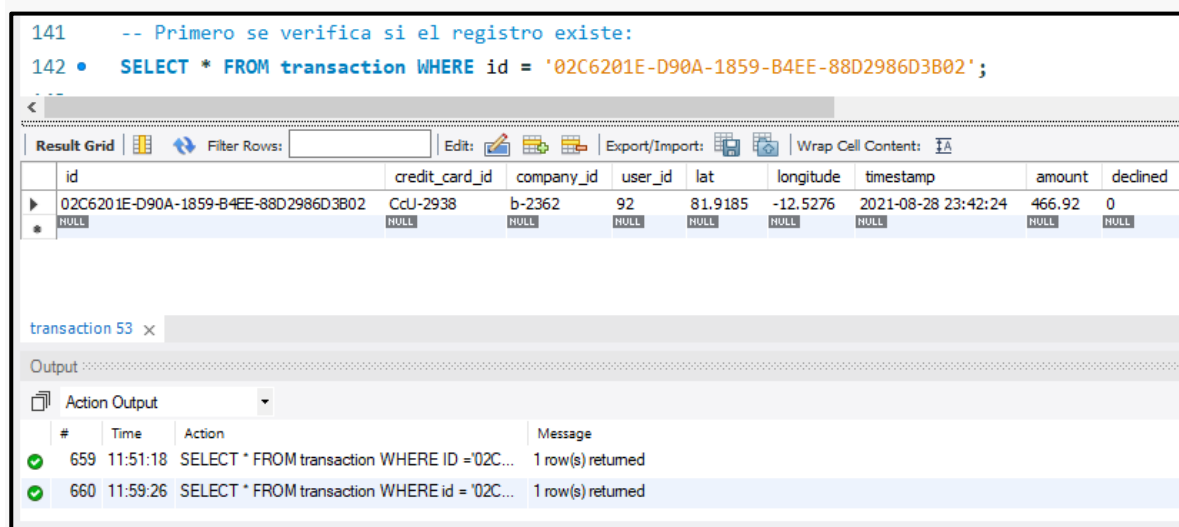
| # | Time | Action | Message |
|-----|----------|----------------------|-------------------|
| 656 | 11:34:42 | DESCRIBE credit_card | 5 row(s) returned |
| 657 | 11:35:36 | DESCRIBE credit_card | 5 row(s) returned |

Nivel 2

Ejercicio 1

Elimina de la tabla transaction el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de datos.

```
141 -- Primero se verifica si el registro existe:
142 • SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```



| id | credit_card_id | company_id | user_id | lat | longitude | timestamp | amount | declined |
|--------------------------------------|----------------|------------|---------|---------|-----------|---------------------|--------|----------|
| 02C6201E-D90A-1859-B4EE-88D2986D3B02 | CcU-2938 | b-2362 | 92 | 81.9185 | -12.5276 | 2021-08-28 23:42:24 | 466.92 | 0 |

transaction 53 x

Output

Action Output

| # | Time | Action | Message |
|-----|----------|--|-------------------|
| 659 | 11:51:18 | SELECT * FROM transaction WHERE ID =02C... | 1 row(s) returned |
| 660 | 11:59:26 | SELECT * FROM transaction WHERE id = '02C... | 1 row(s) returned |

```

144 -- Confirmando, el registro sí existe, se procede a eliminarlo
145 • DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
146
147

```

Output

Action Output

| # | Time | Action | Message |
|-------|----------|--|-------------------|
| ✓ 660 | 11:59:26 | SELECT * FROM transaction WHERE id = '02C... | 1 row(s) returned |
| ✓ 661 | 12:06:38 | DELETE FROM transaction WHERE id = '02C6... | 1 row(s) affected |

```

149 -- verificamos que se haya eliminado
150 • SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
151

```

Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Content:

| id | credit_card_id | company_id | user_id | lat | longitude | timestamp | amount | declined |
|----|----------------|------------|---------|------|-----------|-----------|--------|----------|
| * | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

transaction 6 x

Output

Action Output

| # | Time | Action | Message |
|------|----------|---|---------------------|
| ✓ 17 | 09:06:10 | SELECT * FROM transactions.infometecnico | 587 row(s) returned |
| ✓ 18 | 09:06:32 | SELECT * FROM transactions.infometecnico | 587 row(s) returned |
| ✓ 19 | 11:06:12 | Select * from transaction where declined = '1' | 87 row(s) returned |
| ✓ 20 | 11:09:29 | SELECT * FROM transactions.infometecnico | 587 row(s) returned |
| ✓ 21 | 11:28:34 | SELECT * FROM transactions.vistamarketing | 101 row(s) returned |
| ✓ 22 | 12:19:00 | SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02' | 0 row(s) returned |

Ejercicio 2

La sección de marketing desea tener acceso a información específica para realizar análisis y estrategias efectivas. Se ha solicitado crear una vista que proporcione detalles clave sobre las compañías y sus transacciones. Será necesaria que crees una vista llamada VistaMarketing que contenga la siguiente información: Nombre de la compañía. Teléfono de contacto. País de residencia. Promedio de compra realizado por cada compañía. Presenta la vista creada, ordenando los datos de mayor a menor media de compra.

```

156 • CREATE OR REPLACE VIEW VistaMarketing AS
157 SELECT
158     c.company_name AS Nombre_Compania,
159     c.phone AS Telefono,
160     c.country AS Pais,
161     ROUND(AVG(t.amount), 2) AS Promedio_Compra
162 FROM company c
163 JOIN transaction t ON c.id = t.company_id
164 GROUP BY c.company_name, c.phone, c.country
165 ORDER BY Promedio_Compra DESC;
166

```

<

Output



Action Output

| | # | Time | Action | Message |
|---|-----|----------|---|-------------------|
| ✓ | 667 | 13:06:08 | SELECT * FROM VistaMarketing WHERE Pais ... | 8 row(s) returned |
| ✓ | 668 | 13:09:24 | CREATE OR REPLACE VIEW VistaMarketing A... | 0 row(s) affected |

-- Ejecutamos la vista para ver los resultados

167 -- Ejecutamos la vista para ver los resultados

168 • **SELECT * FROM VistaMarketing;**

169


<

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

| | Nombre_Compania | Telefono | Pais | Promedio_Compra |
|---|------------------------------|----------------|----------------|-----------------|
| ▶ | Eget Ipsum Ltd | 03 67 44 56 72 | United States | 473.08 |
| | Non Magna LLC | 06 71 73 13 17 | United Kingdom | 468.35 |
| | Sed Id Limited | 07 28 18 18 13 | United States | 461.21 |
| | Justo Eu Arcu Ltd | 08 42 56 71 52 | Italy | 443.64 |
| | Eget Tincidunt Dui Institute | 05 35 93 32 44 | Netherlands | 442.52 |

VistaMarketingo 58 x

Output

 Action Output ▼

| | # | Time | Action | Message |
|---|-----|----------|--|---------------------|
| ✓ | 668 | 13:09:24 | CREATE OR REPLACE VIEW VistaMarketing A... | 0 row(s) affected |
| ✓ | 669 | 13:13:31 | SELECT * FROM VistaMarketing | 101 row(s) returned |

Ejercicio 3

Filtra la vista VistaMarketing para mostrar sólo las compañías que tienen su país de residencia en "Germany"

173 • **SELECT ***

174 **FROM VistaMarketing**

175 **WHERE Pais = 'Germany';**

176


<

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

| | Nombre_Compania | Telefono | Pais | Promedio_Compra |
|---|----------------------------|----------------|---------|-----------------|
| ▶ | Aliquam PC | 01 45 73 52 16 | Germany | 385.27 |
| | Ac Industries | 09 34 65 40 60 | Germany | 289.65 |
| | Rutrum Non Inc. | 02 66 31 61 09 | Germany | 266.90 |
| | Nunc Interdum Incorporated | 05 18 15 48 13 | Germany | 244.03 |
| | Augue Foundation | 06 88 43 15 63 | Germany | 240.80 |

VistaMarketingo 59 x

Output

 Action Output ▼

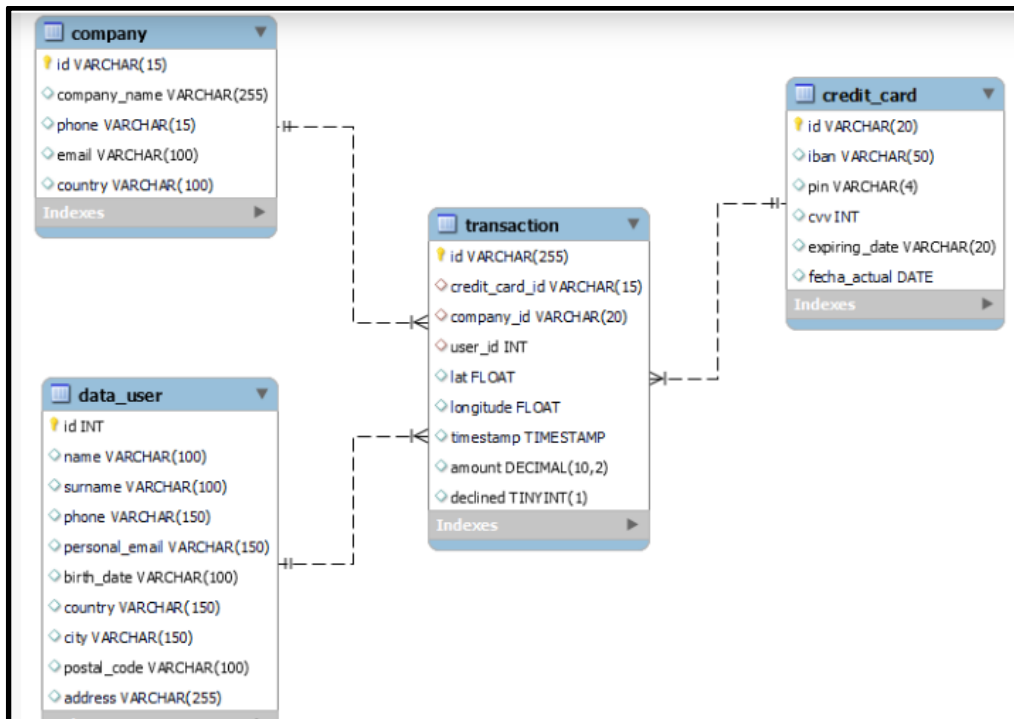
| | # | Time | Action | Message |
|---|-----|----------|---|---------------------|
| ✓ | 669 | 13:13:31 | SELECT * FROM VistaMarketing | 101 row(s) returned |
| ✓ | 670 | 13:19:49 | SELECT * FROM VistaMarketing WHERE Pais ... | 8 row(s) returned |

Nivel 3

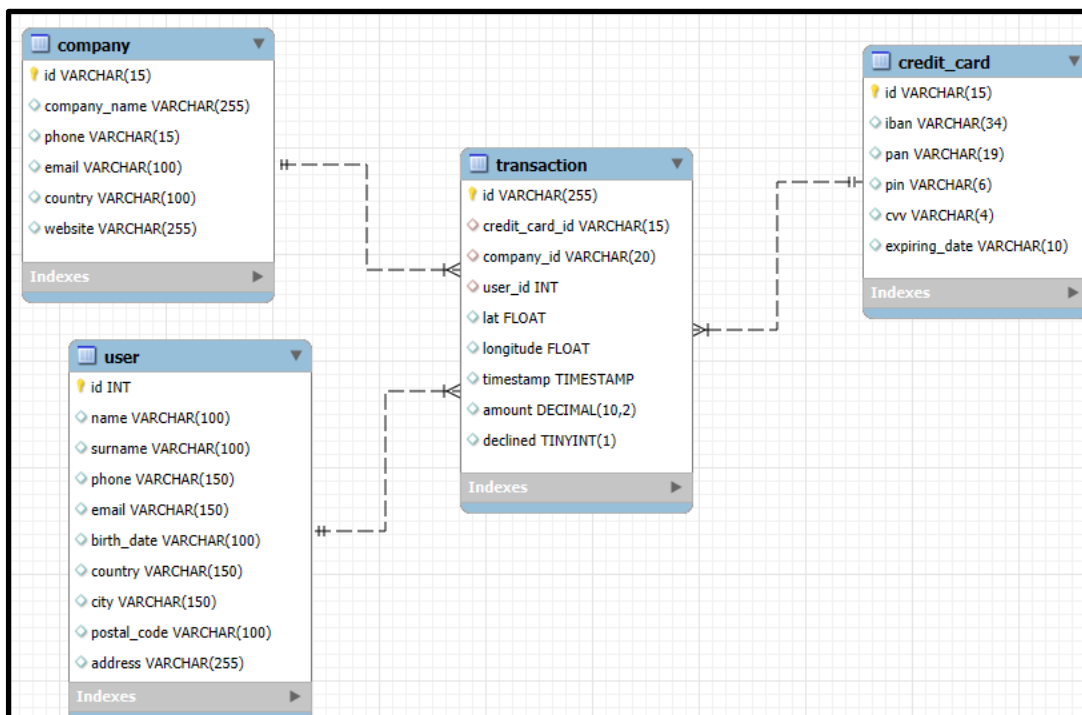
Ejercicio 1

La semana próxima tendrás una nueva reunión con los gerentes de marketing. Un compañero de tu equipo realizó modificaciones en la base de datos, pero no recuerda cómo las realizó. Te pide que le ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama:

El diagrama debe quedar tal como está en la imagen

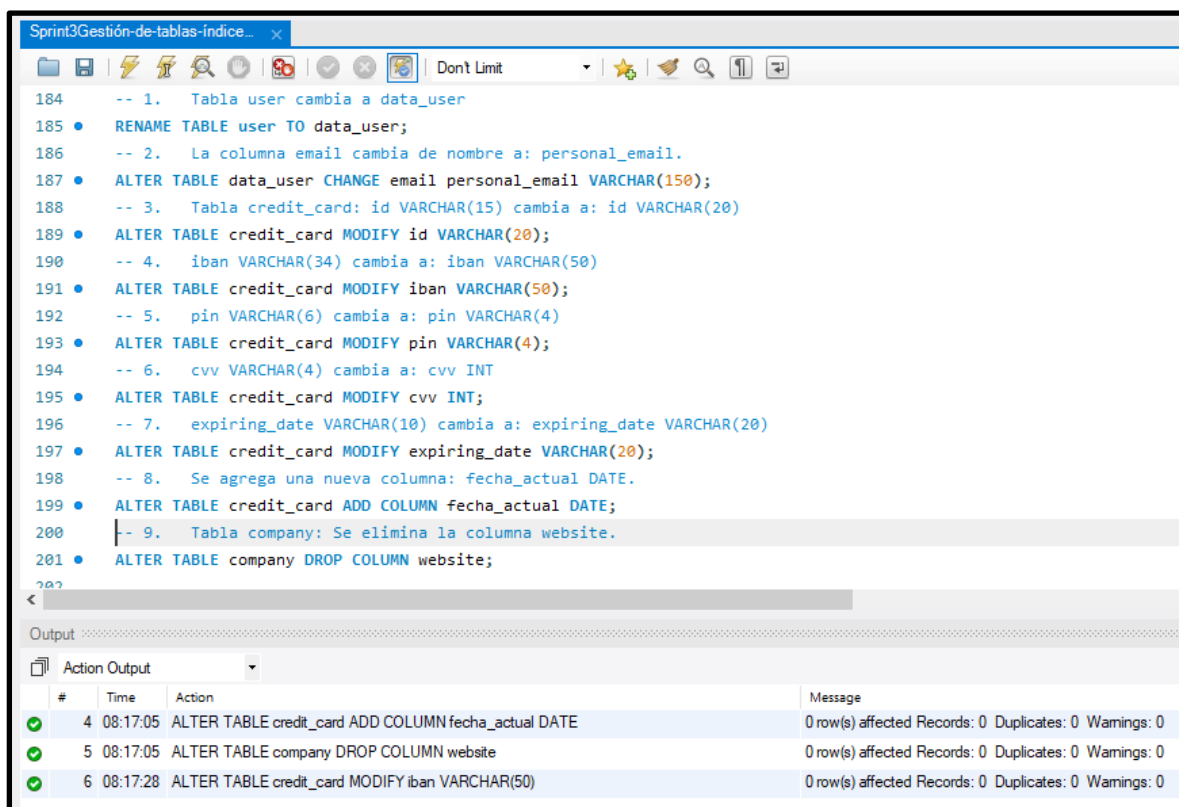


El diagrama que actualmente tenemos es:



Comparando los dos diagramas, se procede a ordenar las tablas como nos piden y comparamos las diferencias:

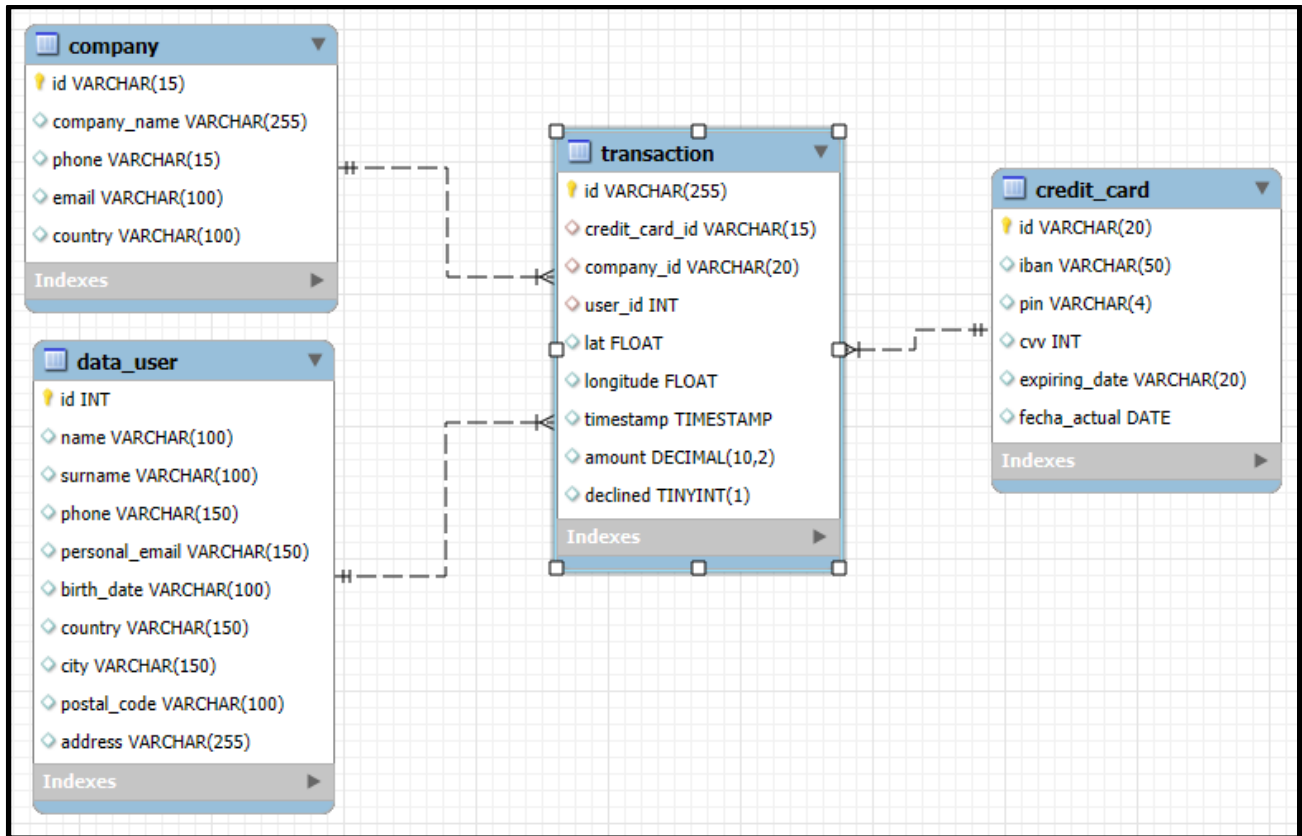
1. Tabla user cambia a: data_user
RENAME TABLE user TO data_user;
2. La columna email cambia de nombre a: personal_email.
ALTER TABLE data_user CHANGE email personal_email VARCHAR(150);
3. Tabla credit_card: id VARCHAR(15) cambia a: id VARCHAR(20)
ALTER TABLE credit_card MODIFY id VARCHAR(20);
4. iban VARCHAR(34) cambia a: iban VARCHAR(50)
ALTER TABLE credit_card MODIFY iban VARCHAR(50);
5. pin VARCHAR(6) cambia a: pin VARCHAR(4)
ALTER TABLE credit_card MODIFY pin VARCHAR(4);
6. cvv VARCHAR(4) cambia a: cvv INT
ALTER TABLE credit_card MODIFY cvv INT;
7. expiring_date VARCHAR(10) cambia a: expiring_date VARCHAR(20)
ALTER TABLE credit_card MODIFY expiring_date VARCHAR(20);
8. Se agrega una nueva columna: fecha_actual DATE.
ALTER TABLE credit_card ADD COLUMN fecha_actual DATE;
9. Tabla company: Se elimina la columna website.
ALTER TABLE company DROP COLUMN website;



```
184 -- 1. Tabla user cambia a data_user
185 • RENAME TABLE user TO data_user;
186 -- 2. La columna email cambia de nombre a: personal_email.
187 • ALTER TABLE data_user CHANGE email personal_email VARCHAR(150);
188 -- 3. Tabla credit_card: id VARCHAR(15) cambia a: id VARCHAR(20)
189 • ALTER TABLE credit_card MODIFY id VARCHAR(20);
190 -- 4. iban VARCHAR(34) cambia a: iban VARCHAR(50)
191 • ALTER TABLE credit_card MODIFY iban VARCHAR(50);
192 -- 5. pin VARCHAR(6) cambia a: pin VARCHAR(4)
193 • ALTER TABLE credit_card MODIFY pin VARCHAR(4);
194 -- 6. cvv VARCHAR(4) cambia a: cvv INT
195 • ALTER TABLE credit_card MODIFY cvv INT;
196 -- 7. expiring_date VARCHAR(10) cambia a: expiring_date VARCHAR(20)
197 • ALTER TABLE credit_card MODIFY expiring_date VARCHAR(20);
198 -- 8. Se agrega una nueva columna: fecha_actual DATE.
199 • ALTER TABLE credit_card ADD COLUMN fecha_actual DATE;
200 -- 9. Tabla company: Se elimina la columna website.
201 • ALTER TABLE company DROP COLUMN website;
```

Output

| # | Time | Action | Message |
|-----|----------|--|--|
| ✓ 4 | 08:17:05 | ALTER TABLE credit_card ADD COLUMN fecha_actual DATE | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 |
| ✓ 5 | 08:17:05 | ALTER TABLE company DROP COLUMN website | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 |
| ✓ 6 | 08:17:28 | ALTER TABLE credit_card MODIFY iban VARCHAR(50) | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 |



Ejercicio 2

La empresa también te solicita crear una vista llamada "InformeTecnico" que contenga la siguiente información:

- ID de la transacción
- Nombre del usuario/a
- Apellido del usuario/a
- IBAN de la tarjeta de crédito usada.
- Nombre de la compañía de la transacción realizada.
- Asegúrate de incluir información relevante de ambas tablas y utiliza alias para cambiar de nombre columnas según sea necesario.

Muestra los resultados de la vista, ordena los resultados de manera descendente en función de la variable ID de transaction.

```
204 /* Ejercicio 2: La empresa también te solicita crear una vista llamada "InformeTecnico" que contenga la siguiente información:
205 ID de la transacción
206 Nombre del usuario/a
207 Apellido del usuario/a
208 IBAN de la tarjeta de crédito usada.
209 Nombre de la compañía de la transacción realizada.
210 Asegúrate de incluir información relevante de ambas tablas y utiliza alias para cambiar de nombre columnas según sea necesario.
211 Muestra los resultados de la vista, ordena los resultados de manera descendente en función de la variable ID de transaction. */
212
213 • CREATE OR REPLACE VIEW InformeTecnico AS
214 SELECT t.id AS ID_Transaccion, u.name AS Nombre_del_Usuario, u.surname AS Apellido_del_Usuario,
215 cr.iban AS IBAN_Tarjeta, c.company_name AS Nombre_de_la_Compañia
216 FROM data_user u
217 JOIN transaction t ON u.id = t.user_id
218 JOIN credit_card cr ON t.credit_card_id = cr.id
219 JOIN company c ON t.company_id = c.id
220 ORDER BY ID_Transaccion DESC;
```

Output

| # | Time | Action | Message |
|------|----------|--|---------------------|
| ✓ 14 | 08:53:01 | SELECT t.id AS ID_Transaccion, u.name AS Nombre_Usuario, u.surname AS Apellido_Usuari... | 587 row(s) returned |
| ✓ 15 | 09:02:54 | SELECT t.id AS ID_Transaccion, u.name AS Nombre_del_Usuario, u.surname AS Apellido_de... | 587 row(s) returned |
| ✓ 16 | 09:03:51 | CREATE OR REPLACE VIEW InformeTecnico AS SELECT t.id AS ID_Transaccion, u.name ... | 0 row(s) affected |

SELECT * FROM transactions.informetecnico;

```
222 • SELECT * FROM transactions.informetecnico;
223
```

Result Grid

| ID_Transaccion | Nombre_del_Usuario | Apellido_del_Usuario | IBAN_Tarjeta | Nombre_de_la_Compañia |
|--------------------------------------|--------------------|----------------------|--------------------------------|------------------------------|
| FE96CE47-8D59-381C-4E18-E3CA3D44E8FF | Kenyon | Hartman | DO26854763748537475216568689 | Magna A Neque Industries |
| FE809ED4-2DB6-55AC-C915-929516E4646B | Molly | Gilliam | SE2813123487163628531121 | Nunc Interdum Incorporated |
| FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65 | Linus | Willis | KW9485332754781757886242955643 | Nunc Interdum Incorporated |
| FD89D51B-AE8D-77DC-E450-B8083FBD3187 | Hilda | Levy | LT053237077744561475 | Malesuada PC |
| FD2E8957-414B-8EEC-E9AD-59AA7A8A6290 | Hedwig | Gilbert | GE84848451582810541526 | Neque Tellus Imperdiet Corp. |

informetecnico 5 x

Output

| # | Time | Action | Message |
|------|----------|--|---------------------|
| ✓ 16 | 09:03:51 | CREATE OR REPLACE VIEW InformeTecnico AS SELECT t.id AS ID_Transaccion, u.name ... | 0 row(s) affected |
| ✓ 17 | 09:06:10 | SELECT * FROM transactions.informetecnico | 587 row(s) returned |
| ✓ 18 | 09:06:32 | SELECT * FROM transactions.informetecnico | 587 row(s) returned |