

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_introduir_credit". Recuerda mostrar el diagrama y realizar una breve descripción de este.

```

1 -- SPRINT 3 - NIVEL 1 - EJERCICIO 1
2 -- Estado inicial: Verificar tablas existentes
3
4 USE transactions;
5
6 -- Ver qué tablas tenemos
7 • SHOW TABLES;

```

Result Grid

| Tables_in_transactions |
|------------------------|
| company |
| transaction |

Action Output

| | Time | Action | Response | Duration / Fetch Time |
|---|----------|------------------|-------------------|-------------------------|
| 1 | 13:52:33 | USE transactions | 0 row(s) affected | 0.00033 sec |
| 2 | 13:52:33 | SHOW TABLES | 2 row(s) returned | 0.0015 sec / 0.00000... |

Explicación: Primero miro qué tablas ya existen en mi base de datos. Veo que tengo "company" y "transaction", pero no hay "credit_card" todavía. Perfecto, así confirmo que voy a crear la tabla desde cero.

```

10 • CREATE TABLE credit_card (id VARCHAR(20) PRIMARY KEY,
11      iban VARCHAR(50), pan VARCHAR(20), pin VARCHAR(10),
12      cvv VARCHAR(5), expiring_date VARCHAR(10));

```

Action Output

| | Time | Action | Response | Duration / Fetch Time |
|---|----------|--|----------|-----------------------|
| 1 | 14:03:38 | CREATE TABLE credit_card (id VA... 0 row(s) affected | | 0.018 sec |

Explicación: Creo la tabla "credit_card" con todos los campos necesarios. La tabla se creó correctamente.

```

15 • DESCRIBE credit_card;
16

```

Result Grid

| Field | Type | Null | Key | Default | Extra |
|---------------|-------------|------|-----|---------|-------|
| id | varchar(20) | NO | PRI | NULL | |
| iban | varchar(50) | YES | | NULL | |
| pan | varchar(20) | YES | | NULL | |
| pin | varchar(10) | YES | | NULL | |
| cvv | varchar(5) | YES | | NULL | |
| expiring_date | varchar(10) | YES | | NULL | |

Action Output

| | Time | Action | Response | Duration / Fetch Time |
|---|----------|----------------------|-------------------|-------------------------|
| 1 | 14:11:54 | DESCRIBE credit_card | 6 row(s) returned | 0.0013 sec / 0.00001... |

Explicación: Verifico que la tabla tiene los 6 campos correctos y que "id" es la clave primaria.

```

1 • USE transactions;
2
3 -- Insertamos datos de credit_card
4 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2938'
5 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2945'
6 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2952'
7 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2959'
8 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2966'
9 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2973'
10 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2980'
11 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2987'
12 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2994'

```

| Action Output | | Action | | | | Response | | Duration / Fetch Time | |
|---------------|----------|--------------------------------------|--|-------------------|--|----------|--|-----------------------|--|
| | Time | Action | | | | | | | |
| ✓ 1 | 10:15:50 | USE transactions | | 0 row(s) affected | | | | 0.015 sec | |
| ✓ 2 | 10:15:50 | INSERT INTO credit_card (id, iban... | | 1 row(s) affected | | | | 0.017 sec | |
| ✓ 3 | 10:15:50 | INSERT INTO credit_card (id, iban... | | 1 row(s) affected | | | | 0.00050 sec | |
| ✓ 4 | 10:15:50 | INSERT INTO credit_card (id, iban... | | 1 row(s) affected | | | | 0.00022 sec | |
| ✓ 5 | 10:15:50 | INSERT INTO credit_card (id, iban... | | 1 row(s) affected | | | | 0.00033 sec | |
| ✓ 6 | 10:15:50 | INSERT INTO credit_card (id, iban... | | 1 row(s) affected | | | | 0.00015 sec | |
| ✓ 7 | 10:15:50 | INSERT INTO credit_card (id, iban... | | 1 row(s) affected | | | | 0.00014 sec | |

Explicación: Adicionei os dados das tarjetas de crédito na tabla, uma por uma.

| 18 • SELECT COUNT(*) as total_registros FROM credit_card; | | | | | | | | | |
|---|----------|------------------------------------|--|-------------------|--|----------|--|-------------------------|--|
| Action Output | | Time | | Action | | Response | | Duration / Fetch Time | |
| | Time | Action | | | | | | | |
| ✓ 1 | 10:51:53 | SELECT COUNT(*) as total_regist... | | 1 row(s) returned | | | | 0.0014 sec / 0.00000... | |

Explicación: Verifiquei quantas tarjetas foram inseridas. Total: 275 tarjetas.

| 20 • DESCRIBE transaction; | | | | | | | | | |
|----------------------------|----------|----------------------|--|-------------------|--|----------|--|-------------------------|--|
| Action Output | | Time | | Action | | Response | | Duration / Fetch Time | |
| | Time | Action | | | | | | | |
| ✓ 1 | 11:03:17 | DESCRIBE transaction | | 9 row(s) returned | | | | 0.0017 sec / 0.00000... | |

| 22 • DESCRIBE company; | | | | | | | | | |
|------------------------|----------|------------------|--|-------------------|--|----------|--|-------------------------|--|
| Action Output | | Time | | Action | | Response | | Duration / Fetch Time | |
| | Time | Action | | | | | | | |
| ✓ 1 | 11:04:34 | DESCRIBE company | | 6 row(s) returned | | | | 0.0021 sec / 0.00000... | |

Explicación: Verifico a estrutura das tabelas transaction e company para ver como conectá-las com credit_card.

| 25 • SHOW CREATE TABLE transaction; | | | | | | | | | |
|-------------------------------------|----------|-------------------------------|--|-------------------|--|----------|--|--------------------------|--|
| Action Output | | Time | | Action | | Response | | Duration / Fetch Time | |
| | Time | Action | | | | | | | |
| ✓ 1 | 11:15:11 | SHOW CREATE TABLE transaction | | 1 row(s) returned | | | | 0.00062 sec / 0.00000... | |

Explicación: Verifico si ya existen relaciones entre las tablas antes de crear nuevas

```

28 • ALTER TABLE transaction
29   ADD CONSTRAINT fk_credit_card
30     FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);

```

100% ◊ 20:25

Action Output ◊

| Time | Action | Response | Duration / Fetch Time |
|---|----------------------------------|---|-----------------------|
| 1 11:42:53 | ALTER TABLE transaction ADD C... | 587 row(s) affected Records: 587 Duplicates: 0 War... | 0.044 sec |
| 33 • ALTER TABLE transaction | | | |
| 34 ADD CONSTRAINT fk_company | | | |
| 35 FOREIGN KEY (company_id) REFERENCES company(id); | | | |

100% ◊ 14:30

Action Output ◊

| Time | Action | Response | Duration / Fetch Time |
|------------|----------------------------------|---|-----------------------|
| 1 11:45:46 | ALTER TABLE transaction ADD C... | 587 row(s) affected Records: 587 Duplicates: 0 War... | 0.033 sec |

Explicación: Creo las relaciones entre las tablas usando Foreign Keys para conectar credit_card y company con transaction.

```

38 • SHOW CREATE TABLE transaction;

```

100% ◊ 24:33

Result Grid Filter Rows: Q Search Export: Result Grid

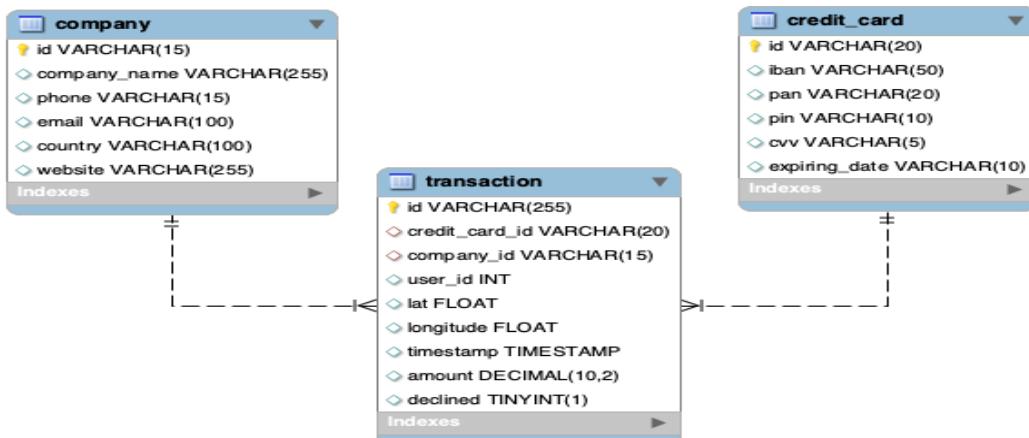
| Table | Create Table |
|-------------|---|
| transaction | CREATE TABLE `transaction` (`id` varchar(25... |

Result 13 Read Only

Action Output ◊

| Time | Action | Response | Duration / Fetch Time |
|------------|-------------------------------|-------------------|-------------------------|
| 1 11:50:45 | SHOW CREATE TABLE transaction | 1 row(s) returned | 0.0019 sec / 0.00002... |

Explicación: Verifico que las relaciones entre tablas se crearon correctamente



Explicación: Diagrama de la base de datos mostrando las relaciones entre credit_card, transaction y company

La tabla credit_card almacena información de las tarjetas de crédito de los clientes. Contiene 6 campos: id (clave primaria), iban, pan (número de tarjeta), pin, cvv y fecha de vencimiento. Se relaciona con la tabla transaction a través del campo credit_card_id, permitiendo vincular cada transacción con su tarjeta correspondiente. La tabla contiene 275 registros de tarjetas de crédito.

IT ACADEMY

Nivel 1

Ejercicio 2

- 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"). Despues de crear la tabla será necesario que ingreses la información del documento denominado "dades_introduir_credit". Recuerda mostrar el diagrama y realizar una breve descripción de este.

The screenshot shows the MySQL Workbench interface with a result grid. The query executed is:

```
2 • SELECT * FROM credit_card WHERE id = 'CcU-2938';
```

The result grid displays one row of data:

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

Action Output shows the following log entry:

| Time | Action | Response | Duration / Fetch Time |
|----------|----------------------------------|-------------------|-------------------------|
| 12:27:01 | SELECT * FROM credit_card WHE... | 1 row(s) returned | 0.00071 sec / 0.0000... |

Explicación: Verifico que las relaciones entre tablas se crearon correctamente

The screenshot shows the MySQL Workbench interface with a result grid. The query executed is:

```
5 • UPDATE credit_card  
6 SET iban = 'R323456312213576817699999'  
7 WHERE id = 'CcU-2938';
```

Action Output shows the following log entry:

| Time | Action | Response | Duration / Fetch Time |
|----------|-------------------------------------|--|-----------------------|
| 12:32:22 | UPDATE credit_card SET iban = '...' | 0 row(s) affected Rows matched: 1 Changed: 0 Warn... | 0.00051 sec |

Explicación: Actualizo el IBAN del usuario CcU-2938 con el valor correcto proporcionado por Recursos Humanos

The screenshot shows the MySQL Workbench interface with a result grid. The query executed is:

```
10 • SELECT * FROM credit_card WHERE id = 'CcU-2938';
```

The result grid displays one row of data:

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

Action Output shows the following log entry:

| Time | Action | Response | Duration / Fetch Time |
|----------|----------------------------------|-------------------|-------------------------|
| 12:37:37 | SELECT * FROM credit_card WHE... | 1 row(s) returned | 0.00087 sec / 0.0000... |

Explicación: Consulto los datos antes del cambio, ejecuto la actualización y verifico que el cambio se realizó correctamente

IT ACADEMY

Nivel 1

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"

2 • DESCRIBE transaction;

| Field | Type | Null | Key | Default | Extra |
|-----------------------------|----------------------------|------|-----|-------------------|-------|
| <code>id</code> | <code>varchar(255)</code> | NO | PRI | <code>NULL</code> | |
| <code>credit_card_id</code> | <code>varchar(20)</code> | YES | MUL | <code>NULL</code> | |
| <code>company_id</code> | <code>varchar(15)</code> | YES | MUL | <code>NULL</code> | |
| <code>user_id</code> | <code>int</code> | YES | | <code>NULL</code> | |
| <code>lat</code> | <code>float</code> | YES | | <code>NULL</code> | |
| <code>longitude</code> | <code>float</code> | YES | | <code>NULL</code> | |
| <code>timestamp</code> | <code>timestamp</code> | YES | | <code>NULL</code> | |
| <code>amount</code> | <code>decimal(10,2)</code> | YES | | <code>NULL</code> | |
| <code>declined</code> | <code>tinyint(1)</code> | YES | | <code>NULL</code> | |

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|----------------------|-------------------|------------------------|
| 12:50:33 | DESCRIBE transaction | 9 row(s) returned | 0.0038 sec / 0.0000... |

Explicación: Verifico la estructura de la tabla transaction antes de insertar el nuevo registro

```
5 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)
6   VALUES ('CcU-9999', 'XX9999999999999999', '9999999999999999', '9999', '999', '12/31/25')
7
```

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|--|----------|-----------------------|
| 13:25:36 | INSERT INTO credit_card (id, iban... 1 row(s) affected | | 0.0094 sec |

Explicación: Creo la tarjeta de crédito CcU-9999 necesaria para realizar la transacción

```
9 • INSERT INTO company (id, company_name, phone, email, country, website)
10  VALUES ('b-9999', 'Empresa Test', '999-999-999', 'test@empresa.com', 'España', 'www.test.e...
```

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|--|----------|-----------------------|
| 10:40:53 | INSERT INTO company (id, comp... 1 row(s) affected | | 0.0054 sec |

```
13 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount,
14   VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.
```

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|---|----------|-----------------------|
| 10:44:28 | INSERT INTO transaction (id, cre... 1 row(s) affected | | 0.0027 sec |

Explicación: Inserto la transacción en la tabla transaction con todos los datos especificados

IT ACADEMY

Nivel 1

Ejercicio 4

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

2 • DESCRIBE credit_card;

| Field | Type | Null | Key | Default | Extra |
|----------------------------|--------------------------|------|-----|-------------------|-------|
| <code>id</code> | <code>varchar(20)</code> | NO | PRI | <code>NULL</code> | |
| <code>iban</code> | <code>varchar(50)</code> | YES | | <code>NULL</code> | |
| <code>pan</code> | <code>varchar(20)</code> | YES | | <code>NULL</code> | |
| <code>pin</code> | <code>varchar(10)</code> | YES | | <code>NULL</code> | |
| <code>cvv</code> | <code>varchar(5)</code> | YES | | <code>NULL</code> | |
| <code>expiring_date</code> | <code>varchar(10)</code> | YES | | <code>NULL</code> | |

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|----------------------|-------------------|-------------------------|
| 10:54:17 | DESCRIBE credit_card | 6 row(s) returned | 0.0061 sec / 0.00001... |

Explicación: Verifico la estructura actual de la tabla credit_card antes de eliminar la columna pan

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

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|--|----------|-----------------------|
| 11:00:11 | ALTER TABLE credit_card DROP... 0 row(s) affected Records: 0 Duplicates: 0 Warnings... 0.019 sec | | |

Explicación: Elimino la columna pan de la tabla credit_card según la solicitud de Recursos Humanos

8 • DESCRIBE credit_card;

| Field | Type | Null | Key | Default | Extra |
|---------------|-------------|------|-----|---------|-------|
| id | varchar(20) | NO | PRI | NULL | |
| iban | varchar(50) | YES | | NULL | |
| pin | varchar(10) | YES | | NULL | |
| cvv | varchar(5) | YES | | NULL | |
| expiring_date | varchar(10) | YES | | NULL | |

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|----------------------|-------------------|-------------------------|
| 11:03:43 | DESCRIBE credit_card | 5 row(s) returned | 0.0018 sec / 0.00001... |

Explicación: Verifico que la columna pin se eliminó correctamente de la tabla credit_card.



- Nivel 2

- Ejercicio 1

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

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

| id | credit... | company_id | user_id | lat | longitude | timestamp | amount | declined | |
|--------------------------------------|-----------------|------------|---------|-----|-----------|-----------|---------------------|----------|---|
| 02C6201E-D90A-1859-B4EE-88D2986D3B02 | CcU-2... b-2362 | HULL | HULL | 92 | 81.9185 | -12.5276 | 2021-08-28 23:42:24 | 466.92 | 0 |

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|--|-------------------|-------------------------|
| 11:13:38 | SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'; | 1 row(s) returned | 0.0034 sec / 0.00001... |

Explicación: Verifico que el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 existe antes de eliminarlo

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

| id | credit... | company_id | user_id | lat | longitude | timestamp | amount | declined | |
|--------------------------------------|-----------------|------------|---------|-----|-----------|-----------|---------------------|----------|---|
| 02C6201E-D90A-1859-B4EE-88D2986D3B02 | CcU-2... b-2362 | HULL | HULL | 92 | 81.9185 | -12.5276 | 2021-08-28 23:42:24 | 466.92 | 0 |

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|--|-------------------|-----------------------|
| 11:19:09 | DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'; | 1 row(s) affected | 0.0030 sec |

Explicación: Elimino el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la tabla transaction

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

| id | credit... | company_id | user_id | lat | longitude | timestamp | amount | declined |
|--------------------------------------|-----------------|------------|---------|------|-----------|-----------|--------|----------|
| 02C6201E-D90A-1859-B4EE-88D2986D3B02 | CcU-2... b-2362 | HULL | HULL | HULL | HULL | HULL | HULL | HULL |

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|--|-------------------|------------------------|
| 11:23:34 | SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'; | 0 row(s) returned | 0.00059 sec / 0.000... |

Explicación: Verifico que el registro se eliminó correctamente - debe mostrar 0 resultados



- Nivel 2

- 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 necesario que crees una vista llamada VistaMarketing que contenga la siguiente información:
 - Nombre de la compañía
 - Teléfono de contacto
 - País de residencia
 - Media de compra realizada por cada compañía

Presenta la vista creada, ordenando los datos de mayor a menor media de compra.

2 • DESCRIBE company;

100% ◇ 33:1

Result Grid Filter Rows: Search Export:

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

Result 1 Read Only

Action Output ◇

| Time | Action | Response | Duration / Fetch Time |
|------------|------------------|-------------------|-------------------------|
| 1 11:40:28 | DESCRIBE company | 6 row(s) returned | 0.0029 sec / 0.00000... |

Explicación: Verifico la estructura de la tabla company para crear la vista VistaMarketing

5 • DESCRIBE transaction;

100% ◇ 27:4

Result Grid Filter Rows: Search Export:

| Field | Type | Null | Key | Default | Extra |
|----------------|---------------|------|-----|---------|-------|
| id | varchar(255) | NO | PRI | NULL | |
| credit_card_id | varchar(20) | YES | MUL | NULL | |
| company_id | varchar(15) | YES | MUL | NULL | |
| user_id | int | YES | | NULL | |
| lat | float | YES | | NULL | |
| longitude | float | YES | | NULL | |
| timestamp | timestamp | YES | | NULL | |
| amount | decimal(10,2) | YES | | NULL | |
| declined | tinyint(1) | YES | | NULL | |

Result 2 Read Only

Action Output ◇

| Time | Action | Response | Duration / Fetch Time |
|------------|----------------------|-------------------|-------------------------|
| 1 11:45:19 | DESCRIBE transaction | 9 row(s) returned | 0.0026 sec / 0.00001... |

Explicación: Verifico la estructura de la tabla transaction para obtener los datos de amount (cantidad de compra)

```
8 • CREATE VIEW VistaMarketing AS
9   SELECT c.company_name AS 'Nombre de la compañía',
10      c.phone AS 'Teléfono de contacto',
11      c.country AS 'País de residencia',
12      AVG(t.amount) AS 'Media de compra'
13  FROM company c
14  JOIN transaction t ON c.id = t.company_id
15  GROUP BY c.id, c.company_name, c.phone, c.country
16  ORDER BY AVG(t.amount) DESC;
```

100% ◇ 1:3

Action Output ◇

| Time | Action | Response | Duration / Fetch Time |
|------------|----------------------------------|-------------------|-----------------------|
| 1 12:28:32 | CREATE VIEW VistaMarketing AS... | 0 row(s) affected | 0.018 sec |

Explicación: Creo la vista VistaMarketing que combina datos de company y transaction para mostrar información clave para marketing

19 • SELECT * FROM VistaMarketing;

100% ◇ 1:17

Result Grid Filter Rows: Search Export:

| Nombre de la compañía | Teléfono de conta... | País de residencia | Media de comp... |
|-------------------------------|----------------------|--------------------|------------------|
| Eget Ipsum Ltd | 03 67 44 56 72 | United States | 473.075000 |
| Non Magna LLC | 06 71 73 13 17 | United Kingdom | 468.345000 |
| Sed Id Limited | 07 28 18 18 13 | United States | 461.210000 |
| Justo Eu Arcu Ltd | 08 42 56 71 52 | Italy | 443.635000 |
| Eget Tincidunt Duis Institute | 05 35 93 32 44 | Netherlands | 442.520000 |
| Viverra Donec Foundation | 03 33 12 32 73 | United Kingdom | 442.280000 |

VistaMarketing 3 Read Only

Action Output ◇

| Time | Action | Response | Duration / Fetch Time |
|------------|-----------------------------------|---------------------|-------------------------|
| 1 12:40:55 | SELECT * FROM VistaMarketing L... | 101 row(s) returned | 0.0074 sec / 0.00002... |

Explicación: Muestro los resultados de la vista VistaMarketing con las empresas ordenadas por media de compra de mayor a menor



- Nivel 2
- Ejercicio 3

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

2 • SELECT * FROM VistaMarketing WHERE `País de residencia` = 'Germany';

100% ◇ 41:1

Result Grid Filter Rows: Search Export:

| Nombre de la compañía | Teléfono de conta... | País de residencia | Media de comp... |
|----------------------------|----------------------|--------------------|------------------|
| Alliquam PC | 01 45 73 52 16 | Germany | 385.265000 |
| Ac Industries | 09 34 65 40 60 | Germany | 289.645000 |
| Rutrum Non Inc. | 02 66 31 61 09 | Germany | 266.900000 |
| Nunc Interdum Incorporated | 05 18 15 48 13 | Germany | 244.025238 |
| Augue Foundation | 06 88 43 15 63 | Germany | 240.800000 |
| Ac Fermentum Incorporated | 06 85 56 52 33 | Germany | 206.465000 |
| Auctor Mauris Corp. | 05 62 87 14 41 | Germany | 184.310000 |
| Convallis In Incorporated | 06 66 57 29 50 | Germany | 156.730000 |

VistaMarketing 2 Read Only

Action Output ◇

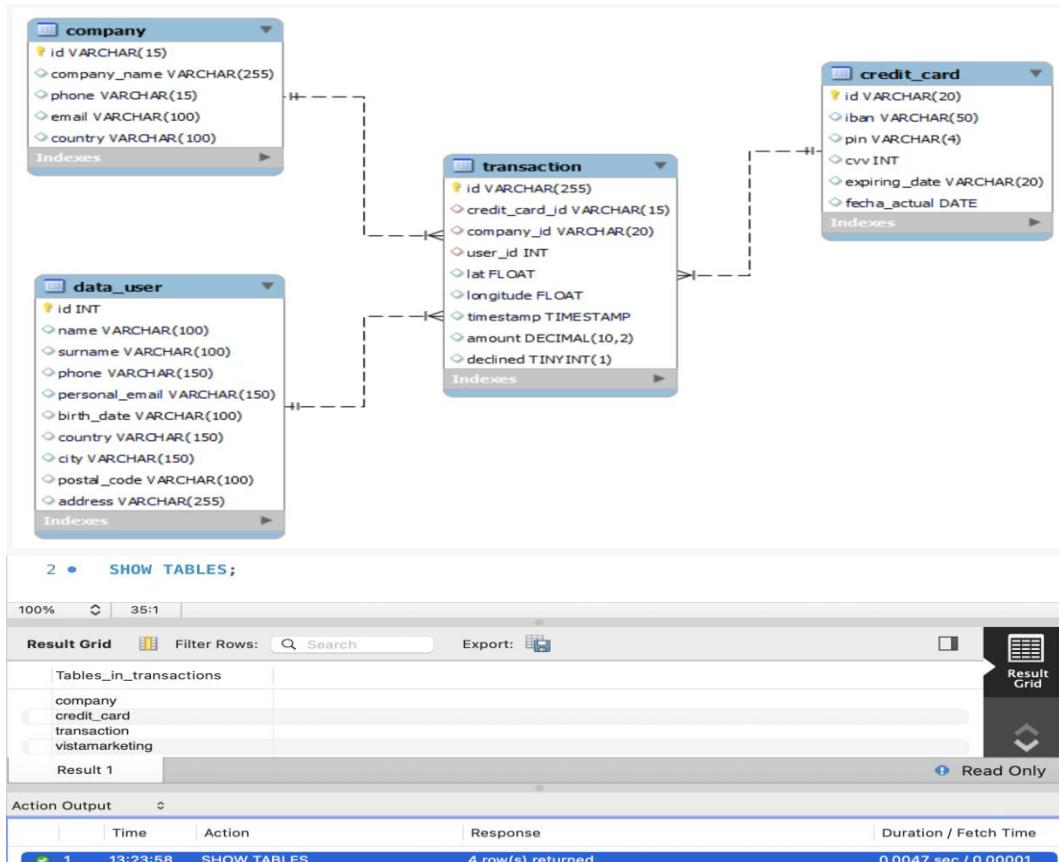
| Time | Action | Response | Duration / Fetch Time |
|------------|---------------------------------|-------------------|-------------------------|
| 1 13:05:37 | SELECT * FROM VistaMarketing... | 8 row(s) returned | 0.0021 sec / 0.00001... |

Explicación: Filtra la vista VistaMarketing para mostrar únicamente las empresas con país de residencia en Germany

Nivel 3

Ejercicio 1

- La semana que viene 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:



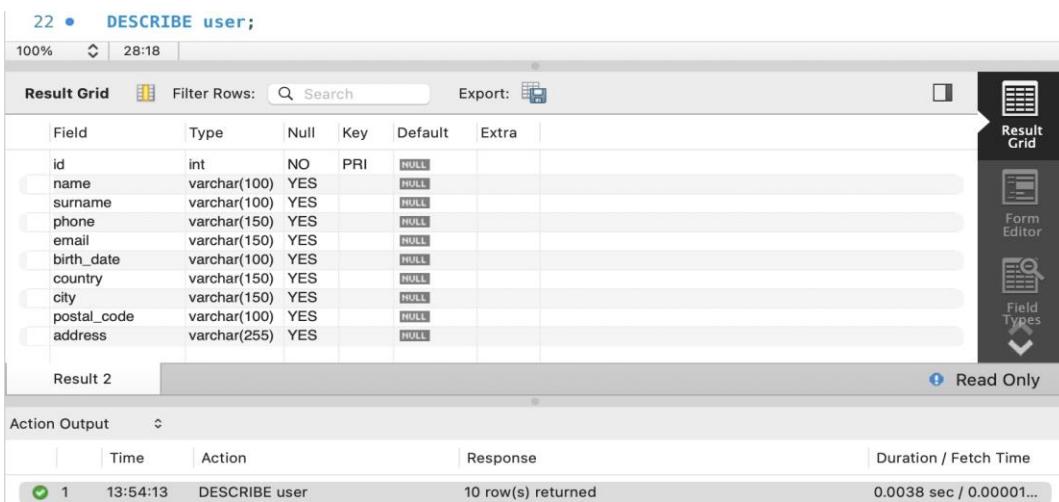
Explicación: Verifico qué tablas existen actualmente en la base de datos para identificar cuáles necesito crear



Explicación: Creo un índice en la columna user_id de la tabla transaction para optimizar las búsquedas antes de crear la tabla user



Explicación: Creo la tabla user sin foreign key constraint para evitar errores de referencia



Explicación: Creo la tabla user sin foreign key constraint para evitar errores de referencia

```
25 • SET foreign_key_checks = 0;
100% 9:21

Action Output
Time Action Response Duration / Fetch Time
1 14:02:45 SET foreign_key_checks = 0 0 row(s) affected 0.0016 sec

100% 28:1

Action Output
Time Action Response Duration / Fetch Time
1 14:06:52 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code) 1 row(s) affected 0.0049 sec
2 14:06:52 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code) 1 row(s) affected 0.00078 sec
3 14:06:52 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code) 1 row(s) affected 0.00057 sec
```

Explicación: Desactivo temporalmente la verificación de foreign keys para poder insertar los datos de usuarios sin errores de referencia

```
4 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code)
5 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code)
6 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code)
7 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code)
100% 28:1

Action Output
Time Action Response Duration / Fetch Time
1 14:06:52 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code) 1 row(s) affected 0.0049 sec
2 14:06:52 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code) 1 row(s) affected 0.00078 sec
3 14:06:52 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code) 1 row(s) affected 0.00057 sec
```

Explicación: Inserto los primeros 3 usuarios en la tabla user con toda su información personal

```
33 • SELECT * FROM user;
100% 20:29

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid
id name surname phone email birth_date country city postal_code
1 Zeus Gamble 1-282-581-0551 interdum.enim@protonmail.edu Nov 17, 1985 United States Lowell 73544
2 Garrett McConnell (718) 257-2412 integer.vitae.nibh@protonmail.org Aug 23, 1992 United States Des Moines 59464
3 Ciaran Harrison (522) 598-1365 interdum.feugiat@aol.org Apr 29, 1998 United States Columbus 56518
NULL NULL NULL NULL NULL NULL NULL NULL NULL
user 4 Apply

Action Output
Time Action Response Duration / Fetch Time
1 14:12:14 SELECT * FROM user LIMIT 0, 1000 3 row(s) returned 0.00060 sec / 0.000...
```

Explicación: Verifico que los 3 primeros usuarios se insertaron correctamente en la tabla user

```
36 • SELECT COUNT(*) AS 'Total de usuarios' FROM user;
100% 21:32

Result Grid Filter Rows: Search Export: Result Grid
Total de usuarios...
275
Result 6 Read Only

Action Output
Time Action Response Duration / Fetch Time
1 14:21:10 SELECT COUNT(*) AS 'Total de us... 1 row(s) returned 0.00073 sec / 0.000...
```

Explicación: Verifico el número total de usuarios insertados en la tabla user

```
39 • SET foreign_key_checks = 1;
100% 22:32

Action Output
Time Action Response Duration / Fetch Time
1 14:24:47 SET foreign_key_checks = 1 0 row(s) affected 0.00028 sec

Action Output
Time Action Response Duration / Fetch Time
1 14:27:11 SHOW TABLES 5 row(s) returned 0.0014 sec / 0.00001...
```

Explicación: Verifico todas las tablas existentes para confirmar que tenemos la estructura completa del diagrama solicitado

```
42 • SHOW TABLES;
100% 1:40

Result Grid Filter Rows: Search Export: Result Grid
Tables_in_transactions
company
credit_card
transaction
user
vistamarketing
Result 8 Read Only

Action Output
Time Action Response Duration / Fetch Time
1 14:27:11 SHOW TABLES 5 row(s) returned 0.0014 sec / 0.00001...
```



```
44 • DESCRIBE transaction;
45 100% 12:41

Result Grid Filter Rows: Search Export: Result Grid
Field Type Null Key Default Extra
id varchar(255) NO PRI NULL
credit_card_id varchar(20) YES MUL NULL
company_id varchar(15) YES MUL NULL
user_id int YES MUL NULL
lat float YES NULL
longitude float YES NULL
timestamp timestamp YES NULL
amount decimal(10,2) YES NULL
declined tinyint(1) YES NULL
Result 10 Read Only

Action Output
Time Action Response Duration / Fetch Time
1 14:43:03 DESCRIBE transaction 9 row(s) returned 0.0019 sec / 0.00001...
```

Explicación: Analizo la estructura de la tabla transaction para identificar los campos que se conectan con otras tablas. Veo que tiene credit_card_id, company_id y user_id para las conexiones

```

46 •   SELECT
47     CONSTRAINT_NAME,
48     TABLE_NAME,
49     COLUMN_NAME,
50     REFERENCED_TABLE_NAME,
51     REFERENCED_COLUMN_NAME
52   FROM INFORMATION_SCHEMA.KEY_COLUMN_USAGE
53   WHERE TABLE_SCHEMA = 'transactions'
54   AND REFERENCED_TABLE_NAME IS NOT NULL;
55

```

Result Grid

| CONSTRAINT_NAME | TABLE_NAME | COLUMN_NAME | REFERENCED_TABLE_NAME | REFERENCED_COLUMN_NAME |
|-----------------|-------------|----------------|-----------------------|------------------------|
| fk_company | transaction | company_id | company | id |
| fk_credit_card | transaction | credit_card_id | credit_card | id |

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|----------------------------|-------------------|--------------------------|
| 14:45:08 | SELECT CONSTRAINT_NAME,... | 2 row(s) returned | 0.0015 sec / 0.000000... |

Explicación: Verifico qué conexiones foreign key existen. Solo hay 2: company y credit_card. Falta user

Result Grid

| Table | Create Table |
|-------------|--|
| transaction | CREATE TABLE 'transaction' ('id' varchar(25...) |

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|-------------------------------|-------------------|------------------------|
| 10:59:36 | SHOW CREATE TABLE transaction | 1 row(s) returned | 0.034 sec / 0.00046... |

Explicación: Veo exactamente cómo está creada la tabla transaction con todas sus conexiones y índices

```

56 •   ALTER TABLE transaction
57     ADD CONSTRAINT fk_user
58     FOREIGN KEY (user_id) REFERENCES user(id);

```

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|-----------------------------------|---|-----------------------|
| 14:54:38 | ALTER TABLE transaction ADD CO... | 587 row(s) affected Records: 587 Duplicates: 0 War... 0.036 sec | |

Explicación: Intento crear la conexión faltante entre transaction y user

```

60 •   SELECT DISTINCT t.user_id
61   FROM transaction t
62   LEFT JOIN user u ON t.user_id = u.id
63   WHERE u.id IS NULL
64   AND t.user_id IS NOT NULL;

```

Result Grid

| user_id |
|---------|
| 9999 |

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|-----------------------------------|-------------------|-------------------------|
| 14:50:52 | SELECT DISTINCT t.user_id FROM... | 1 row(s) returned | 0.0019 sec / 0.00000... |

Explicación: Busco qué usuarios están en transaction pero NO están en la tabla user

```

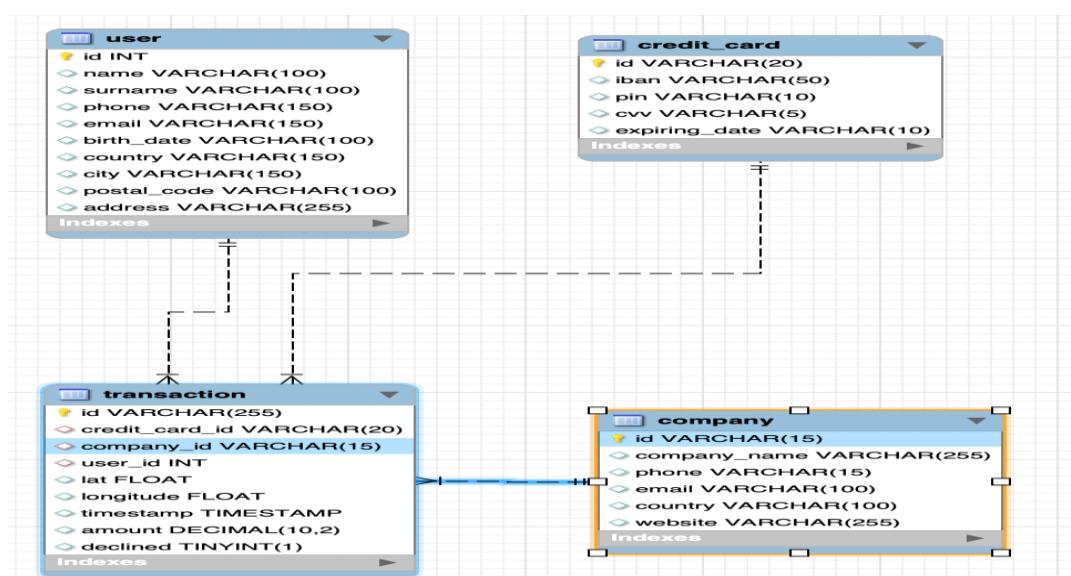
66 •   INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code,
67   VALUES (9999, 'Usuario', 'Desconocido', '000-000-0000', 'unknown@email.com', 'Jan 1, 1900',
68

```

Action Output

| Time | Action | Response | Duration / Fetch Time |
|----------|--------------------------------------|-------------------|-----------------------|
| 14:53:40 | INSERT INTO user (id, name, surna... | 1 row(s) affected | 0.0051 sec |

Explicación: Creo el usuario 9999 que faltaba para que todas las transacciones tengan su usuario



Explicación: Genero el diagrama visual final con todas las conexiones completadas

Nivel 3

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 todas las tablas y utiliza alias para cambiar de nombre las columnas según sea necesario.

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

| 2 • DESCRIBE transaction; | | | | | | |
|-----------------------------|---------------|------|-----|-------------------|-------|--|
| Result Grid | | | | | | |
| Field | Type | Null | Key | Default | Extra | |
| <code>id</code> | varchar(255) | NO | PRI | <code>NULL</code> | | |
| <code>credit_card_id</code> | varchar(20) | YES | MUL | <code>NULL</code> | | |
| <code>company_id</code> | varchar(15) | YES | MUL | <code>NULL</code> | | |
| <code>user_id</code> | int | YES | MUL | <code>NULL</code> | | |
| <code>lat</code> | float | YES | | <code>NULL</code> | | |
| <code>longitude</code> | float | YES | | <code>NULL</code> | | |
| <code>timestamp</code> | timestamp | YES | | <code>NULL</code> | | |
| <code>amount</code> | decimal(10,2) | YES | | <code>NULL</code> | | |
| <code>declined</code> | tinyint(1) | YES | | <code>NULL</code> | | |

Result 1 Read Only

Action Output Duration / Fetch Time

| | | | |
|----------|----------------------|-------------------|-------------------------|
| Time | Action | Response | Duration / Fetch Time |
| 11:37:22 | DESCRIBE transaction | 9 row(s) returned | 0.0075 sec / 0.00001... |

Explicación: Veo qué campos tiene transaction para conectar con las otras tablas.

| 5 • DESCRIBE user; | | | | | | |
|--------------------------|--------------|------|-----|-------------------|-------|--|
| Result Grid | | | | | | |
| Field | Type | Null | Key | Default | Extra | |
| <code>id</code> | int | NO | PRI | <code>NULL</code> | | |
| <code>name</code> | varchar(100) | YES | | <code>NULL</code> | | |
| <code>surname</code> | varchar(100) | YES | | <code>NULL</code> | | |
| <code>phone</code> | varchar(150) | YES | | <code>NULL</code> | | |
| <code>email</code> | varchar(150) | YES | | <code>NULL</code> | | |
| <code>birth_date</code> | varchar(100) | YES | | <code>NULL</code> | | |
| <code>country</code> | varchar(150) | YES | | <code>NULL</code> | | |
| <code>city</code> | varchar(150) | YES | | <code>NULL</code> | | |
| <code>postal_code</code> | varchar(100) | YES | | <code>NULL</code> | | |
| <code>address</code> | varchar(255) | YES | | <code>NULL</code> | | |

Result 2 Read Only

Action Output Duration / Fetch Time

| | | | |
|----------|---------------|--------------------|-------------------------|
| Time | Action | Response | Duration / Fetch Time |
| 11:41:48 | DESCRIBE user | 10 row(s) returned | 0.0029 sec / 0.00001... |

Explicación: Veo qué campos tiene user para obtener nombre y apellido

| 8 • DESCRIBE credit_card; | | | | | | |
|----------------------------|-------------|------|-----|-------------------|-------|--|
| Result Grid | | | | | | |
| Field | Type | Null | Key | Default | Extra | |
| <code>id</code> | varchar(20) | NO | PRI | <code>NULL</code> | | |
| <code>iban</code> | varchar(50) | YES | | <code>NULL</code> | | |
| <code>pin</code> | varchar(10) | YES | | <code>NULL</code> | | |
| <code>cvv</code> | varchar(5) | YES | | <code>NULL</code> | | |
| <code>expiring_date</code> | varchar(10) | YES | | <code>NULL</code> | | |

Result 3 Read Only

Action Output Duration / Fetch Time

| | | | |
|----------|----------------------|-------------------|-------------------------|
| Time | Action | Response | Duration / Fetch Time |
| 11:51:11 | DESCRIBE credit_card | 5 row(s) returned | 0.0021 sec / 0.00000... |

Explicación: Veo qué campos tiene credit_card para obtener el IBAN

| 11 • DESCRIBE company; | | | | | | |
|---------------------------|--------------|------|-----|-------------------|-------|--|
| Result Grid | | | | | | |
| Field | Type | Null | Key | Default | Extra | |
| <code>id</code> | varchar(15) | NO | PRI | <code>NULL</code> | | |
| <code>company_name</code> | varchar(255) | YES | | <code>NULL</code> | | |
| <code>phone</code> | varchar(15) | YES | | <code>NULL</code> | | |
| <code>email</code> | varchar(100) | YES | | <code>NULL</code> | | |
| <code>country</code> | varchar(100) | YES | | <code>NULL</code> | | |
| <code>website</code> | varchar(255) | YES | | <code>NULL</code> | | |

Result 9 Read Only

Action Output Duration / Fetch Time

| | | | |
|----------|------------------|-------------------|-------------------------|
| Time | Action | Response | Duration / Fetch Time |
| 12:02:13 | DESCRIBE company | 6 row(s) returned | 0.0013 sec / 0.00001... |

Explicación: Veo qué campos tiene company para obtener el nombre de la compañía

```

14 • CREATE VIEW InformeTecnico AS
15     SELECT t.id AS 'ID Transacción',
16             u.name AS 'Nombre Usuario',
17             u.surname AS 'Apellido Usuario',
18             cc.iban AS 'IBAN Tarjeta',
19             c.company_name AS 'Nombre Compañía'
20     FROM transaction t
21     JOIN user u ON t.user_id = u.id
22     JOIN credit_card cc ON t.credit_card_id = cc.id
23     JOIN company c ON t.company_id = c.id
24     ORDER BY t.id DESC;

```

100% ▾ 1:9

Action Output ▾

| | Time | Action | Response | Duration / Fetch Time |
|---|----------|------------------------------------|-------------------|-----------------------|
| 1 | 13:00:45 | CREATE VIEW InformeTecnico AS S... | 0 row(s) affected | 0.012 sec |

Explicación: Creo la vista que une las 4 tablas y muestra la información solicitada

27 • SELECT * FROM InformeTecnico;

100% ▾ 15:22

Result Grid Filter Rows: Search Export:

| ID Transacción | Nombre Usuario | Apellido Usuario | IBAN Tarjeta | Nombre Compañía |
|--------------------------------------|----------------|------------------|---------------------------------|-----------------|
| FE96CE47-BD59-381C-4E18-E3CA3D44E8FF | Kenyon | Hartman | DO26854763748537475216568689 | Magna A N... |
| FE809ED4-2DB6-55AC-C915-929516E4646B | Molly | Gilliam | SE2813123487163628531121 | Nunc Inter... |
| FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65 | Linus | Willis | KW9485332754781757886242955643 | Nunc Inter... |
| FD89D51B-AE8D-77DC-E450-B8083FB3187 | Hilda | Levy | LT053237077744561475 | Malesuada... |
| FD2E8957-414B-BEEC-E9AD-59AA7A8A6290 | Hedwig | Gilbert | GE64848451582810541526 | Neque Tell... |
| FCE2AB9A-271D-25DC-9E49-8DD92A373391 | Hakeem | Alford | MD1234119525145401270486 | Nunc Inter... |
| FBD7E0D6-BA6B-F5BC-0CA9-EA4B8760100C | Hedwig | Gilbert | MU41323334453434254134478855 | Mauris Id I... |
| FAC76A80-8448-69AA-E892-426C2F12621C | Slade | Poole | MT05JWCF58868200575771634583813 | Arcu LLP... |
| FAAD3FFC-1A17-E141-43D3-359A5BA7CB3B | Hedwig | Gilbert | GE90157928843338134463 | Lorem Eu... |
| FA053936-75D8-85FA-490D-9B624E1B920A | Hedwig | Gilbert | GT02497653655330848247645975 | Non Justo... |

InformeTecnico 10

Action Output ▾

| | Time | Action | Response | Duration / Fetch Time |
|---|----------|------------------------------------|---------------------|--------------------------|
| 1 | 13:06:06 | SELECT * FROM InformeTecnico Li... | 587 row(s) returned | 0.012 sec / 0.00021 s... |

Explicación: Muestro todos los datos de la vista InformeTecnico ordenados por ID descendente

30 • SHOW FULL TABLES WHERE table_type = 'VIEW';

100% ▾ 36:26

Result Grid Filter Rows: Search Export:

| Tables_in_transactions | Table_type |
|------------------------|------------|
| informetecnico | VIEW |
| vistamarketing | VIEW |

Result 11

Action Output ▾

| | Time | Action | Response | Duration / Fetch Time |
|---|----------|---------------------------------|-------------------|-------------------------|
| 1 | 13:14:29 | SHOW FULL TABLES WHERE table... | 2 row(s) returned | 0.0039 sec / 0.00001... |

Explicación: Verifico que la vista InformeTecnico se creó y está disponible