

ENTREGA SPRINT

FEDERICO URBINA

Curso especialización

IT ACADEMY

Descripción

En este sprint se simula una situación empresarial en la que deberás realizar diversas manipulaciones en las tablas de una base de datos. Además trabajarás con índices y vistas para optimizar consultas y organizar la información.

Continuarás trabajando con la base de datos que contiene información de un marketplace, un entorno similar a Amazon donde varias empresas venden sus productos a través de un canal online. En esta actividad, empezarás a trabajar con datos relacionados con tarjetas de crédito.

Añade las tablas al modelo según corresponda:

- Nivel 1: Tabla "credit_card"
- Nivel 3: Tabla "user"



Nivel 1

Ejercicio 1

Tu tarea es diseñar y crear una mesa 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 mesa será necesario que ingreses la información del documento denominado "dades_introducir_credit". Recuerda mostrar el diagrama y realizar una breve descripción del mismo.

1.a)

```

4 • CREATE TABLE Credit_card (
5           id          VARCHAR(15) PRIMARY KEY,
6           iban        VARCHAR(50),
7           pan         VARCHAR(50),
8           pin         VARCHAR(4),
9           cvv          INT,
10          expiring_date VARCHAR(20),
11          fecha_actual VARCHAR(20)
12
13      );
14
10%  ◁  28:13

```

Action Output	Time	Action	Response
1	17:23:41	USE transactions	0 row(s) affected
2	17:23:50	CREATE INDEX credit_ca...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3	17:23:55	CREATE TABLE Credit_c...	0 row(s) affected

1.b)

```

4998 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9578', 'XX991539646456110567870254', '8999808823061411', '2872', '772', '07/29/27');
4999 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9579', 'XX296393091587170202131236', '9690060468678689', '8379', '134', '12/25/25');
5000 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9580', 'XX781258889851950806677358', '5541182364498931', '9273', '737', '03/27/29');
5001 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9581', 'XX915670516405388124398147', '2624305470167630', '4336', '926', '06/29/25');
5002

```

Action Output	Time	Action	Response	Duration / Fetch Tir
4998	17:27:42	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9578', 'XX991539646456110567870254', '8999808823061411', '2872', '772', '07/29/27')	1 row(s) affected	0.00000 sec
5000	17:27:42	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9581', 'XX915670516405388124398147', '2624305470167630', '4336', '926', '06/29/25')	1 row(s) affected	0.00007 sec

```
1 •  SELECT count(id)
2   FROM transactions.Credit_card;
```

A screenshot of a database query results grid. The grid has one column labeled 'count(id)' and one row containing the value '5000'. The interface includes a top bar with zoom level (100%), a refresh icon, and a timestamp (31:2). Below the grid are buttons for 'Result Grid' and 'Filter Rows', and a search bar.

count(id)	5000
-----------	------

Operaciones realizadas:

- Creé la tabla credit_card con los campos indicados en el enunciado.
- Importé los registros proporcionados.
- Verifiqué la cantidad total de filas mediante un recuento.

Resultado clave:

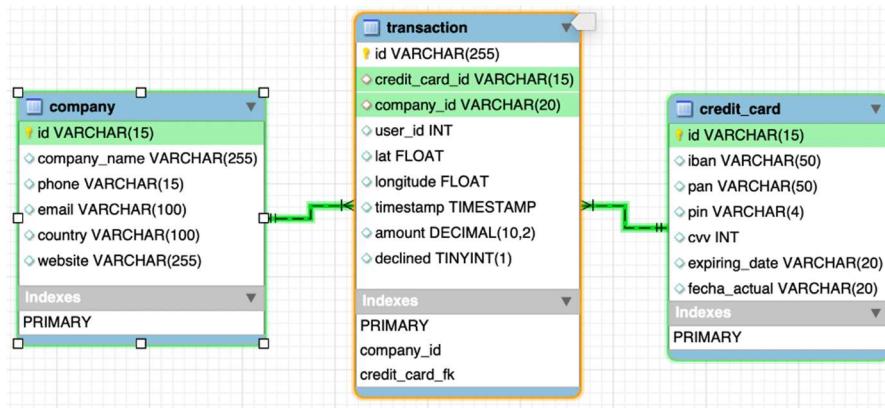
- La tabla credit_card contiene 5.000 registros.

Relevancia:

- Permite continuar con los ejercicios que dependen de esta tabla.

Esquema:

La estructura del modelo corresponde a una base de datos relacional normalizada.



Las relaciones definidas son de tipo 1:N

company → transaction: una empresa puede tener múltiples transacciones asociadas.

credit_card → transaction: una tarjeta puede aparecer en varias transacciones.

- La tabla transaction contiene claves foráneas hacia company y credit_card.

Ejercicio 2

El departamento de Recursos Humanos ha identificado un error en el número de cuenta asociado a la tarjeta de crédito con ID CcU-2938. La información que debe mostrarse para este registro es: TR323456312213576817699999. Recuerda mostrar que el cambio se realizó.

Operaciones realizadas:

- Verifico antes de hacer cambios

```

16 •  SELECT IBAN
17      FROM credit_card
18      WHERE id = 'CcU-2938';
19
  
```

Result Grid

IBAN
TR301950312213576817638661

credit_card 1

Action Output

Time	Action
17:38:19	SELECT IBAN FROM credit_card WHERE id = 'CcU-2938' LIMIT 0, 50000

- Realizo el Update

```

20 •  UPDATE credit_card
21      SET iban = 'TR323456312213576817699999'
22      WHERE id = 'CcU-2938';
  
```

Action Output

Time	Action
17:40:41	UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE id = 'CcU-2938'

```
24 •   SELECT IBAN
25     FROM credit_card
26    WHERE id = 'CcU-2938';|
```

00% 24:26

Result Grid Filter Rows: Search Export:

IBAN
TR323456312213576817699999

credit_card 2

Action Output

Time	Action
17:43:09	SELECT IBAN FROM credit_card WHERE id = 'CcU-2938' LIMIT 0, 50000

- Verifico el cambio realizado

Resultado clave

- El campo iban quedó actualizado.

Relevancia

- Aplica un cambio directo sobre un registro concreto.

Ejercicio 3

En la tabla "transaction" ingresa una nueva transacción 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

Operaciones realizadas

```

• Ⓜ INSERT INTO transaction (
    id,
    credit_card_id,
    company_id,
    user_id,
    lat,
    longitude,
    amount,
    declined
)
VALUES (
    '108B1D1D-5B23-A76C-55EF-C568E49A99DD',
    'CcU-9999',
    'b-9999',
    9999,
    829.999,
    -117.999,
    111.11,
    0
);

```

00% 9:59 1 error found

Action Output

Action	Time	Response
19 11:26:42 INSERT INTO transaction (id,...		Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (

- Revisé la estructura de la tabla transaction con DESCRIBE para confirmar los nombres correctos de las columnas.
- Intenté insertar la transacción indicada en el enunciado utilizando los campos id, credit_card_id, company_id, user_id, lat, longitude, amount y declined.

- MySQL devolvió el error 1452 al no existir los identificadores b-9999 (company) y CcU-9999 (credit card).

En un proyecto real no insertaría una transacción sin la empresa y la tarjeta correspondientes, porque violaría la integridad referencial y afectaría la calidad del dato. Para poder completar el ejercicio y mantener la consistencia del modelo hablaría con las áreas correspondientes para que aporten los datos necesarios:

- `INSERT INTO company (id) VALUES ('b-9999');`
- `INSERT INTO credit_card (id) VALUES ('CcU-9999');`

00% 44:60 1 error found

Action Output

Action	Time	Response
27 12:13:11 INSERT INTO credit_card (id) VALUES... 1 row(s) affected		
28 12:13:24 INSERT INTO company (id) VALUES ('... 1 row(s) affected		

- Incorporaría los registros que faltaban inicialmente en ambas tablas, permitiendo así validar la inserción y continuar con el flujo requerido por el enunciado.

```

• INSERT INTO transaction (
    id,
    credit_card_id,
    company_id,
    user_id,
    lat,
    longitude,
    amount,
    declined
)
VALUES (
    '108B1D1D-5B23-A76C-55EF-C568E49A99DD',
    'CcU-9999',
    'b-9999',
    9999,
    829.999,
    -117.999,
    111.11,
    0
);
  
```

100% 9:59 1 error found

Action Output

	Time	Action	Response
30	12:16:23	INSERT INTO transaction (id,...)	1 row(s) affected

- Volví a ejecutar el INSERT y la transacción fue añadida correctamente.

```

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

- Verifiqué el registro añadido

100% 51:65 1 error found

Result Grid Filter Rows: Search Edit: Export/Import:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	NULL	111.11	0

transaction 7

Action Output

	Time	Action	Response
31	12:36:32	SELECT * FROM transaction WHERE id...	1 row(s) returned

Resultado clave

- La transacción se insertó correctamente.

Relevancia

- Muestra el comportamiento del sistema ante referencias inexistentes.

Ejercicio 4

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

Operaciones realizadas

- Verifico tabla y campo a eliminar

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	HULL	
iban	varchar(50)	YES		HULL	
pan	varchar(50)	YES		HULL	
pin	varchar(4)	YES		HULL	
cvv	int	YES		HULL	
expiring_date	varchar(20)	YES		HULL	
fecha_actual	varchar(20)	YES		HULL	

```

• DESCRIBE credit_card; # Verifico los campos de la tabla
• ALTER TABLE credit_card -- -----| Modificación Solicitada
  DROP COLUMN pan;

```

Action Output

Time	Action	Response
33 15:39:54	ALTER TABLE credit_card -- -----	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

- Usé la sentencia ALTER TABLE para eliminar la columna `pan`.

- Verifico nuevamente la estructura final

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	HULL	
iban	varchar(50)	YES		HULL	
pin	varchar(4)	YES		HULL	
cvv	int	YES		HULL	
expiring_date	varchar(20)	YES		HULL	
fecha_actual	varchar(20)	YES		HULL	

Resultado clave

- La columna pan fue eliminada.

Relevancia

- Evita almacenar un dato altamente confidencial
- Mantiene la tabla credit_card ajustada al diseño solicitado.
- Refuerza el uso de ALTER TABLE como herramienta para adaptar la estructura de una base de datos.

 Nivel 2

Ejercicio 1

Elimina de la tabla transaction el registro con ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de datos.

Operaciones realizadas

- `SELECT * # Verifico q el registro existe
FROM transaction
WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';`

- Verifico que el registro existe

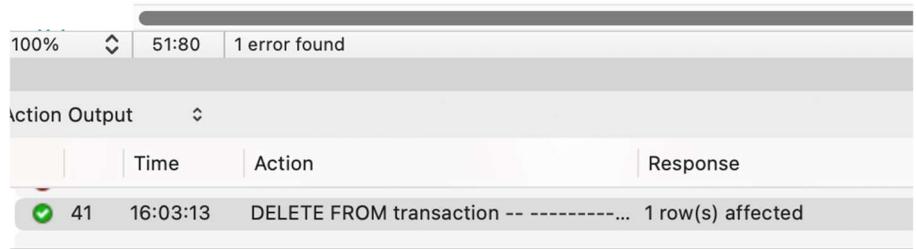


id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
000447FE-B650-4DCF-85DE-C7ED0EE1CAAD	CcS-5019	b-2370	438	41.5972	12.2218	2016-12-21 20:07:18	155.63	0

- `DELETE FROM transaction -- -----`

- `WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';`

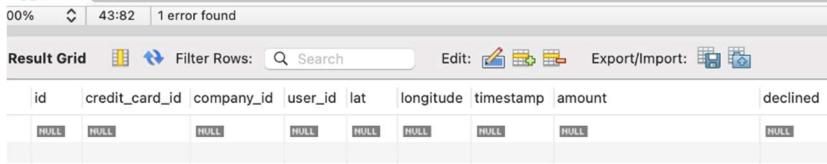
- Ejecutó la sentencia DELETE filtrando por la PK `id`.



Action	Time	Output	Response
41	16:03:13	DELETE FROM transaction -- -----... 1 row(s) affected	

- `SELECT * # Verifico q el registro NO existe
FROM transaction
WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';`

- Verifico que el registro NO existe



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

Resultado clave

- El registro con fue eliminado correctamente de la tabla transaction.

Relevancia

- Aplicación del uso correcto de DELETE + PK, evitando afectar múltiples registros.
- Refuerza buenas prácticas de control previo y verificación posterior al borrado.

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.

```
a) 80 • CREATE VIEW VistaMarketing AS
81   SELECT
82     Company.company_name AS NombreEmpresa,
83     Company.phone AS Telefono,
84     Company.country AS Pais,
85     AVG(Transaction.amount) AS PromCompras
86   FROM transaction AS Transaction
87   JOIN company AS Company
88     ON Transaction.company_id = Company.id
89   GROUP BY
90     NombreEmpresa,
91     Telefono,
92     Pais;
93
```

Action Output			
	Time	Action	Response
1	18:01:15	CREATE VIEW VistaMarketing AS SELECT...	0 row(s) affected

b)

```
94 • SELECT *
95   FROM VistaMarketing
96   ORDER BY PromCompras DESC;
```

Result Grid			
NombreEmpresa	Telefono	Pais	PromCompras
Preum neque Corp.	07 11 48 55 28	Australia	270.198330
Urna Convallis Associates	06 01 24 77 04	United States	274.235011
At Associates	09 56 61 10 65	New Zealand	272.214870
Metus Vitae Associates	08 25 44 40 66	Australia	270.080965
Aliquet Diam Limited	02 76 61 47 46	United States	269.599181
Nec Luctus LLC	02 14 71 75 73	Norway	268.604837

VistaMarketing 3			
Action Output			
	Time	Action	Response
2	18:03:16	SELECT * FROM VistaMarketing ORDER BY...	101 row(s) returned

Operaciones realizadas

- Creé una vista VistaMarketing.
- Realicé un JOIN entre las tablas **transaction** y **company** utilizando el campo **company_id**.
- Calculé el promedio del importe de las transacciones mediante la función **AVG(amount)**.
- Agrupé los resultados por nombre de empresa, teléfono y país.
- Consulté la vista ordenando los resultados de mayor a menor promedio de compras.

Resultado clave

- La vista devuelve, para cada empresa, su nombre, teléfono, país y el promedio del importe de sus transacciones.

Relevancia

- Proporciona una vista agregada de las transacciones por empresa según lo solicitado en el ejercicio.

Ejercicio 3

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

```

98 •  SELECT *
99   FROM VistaMarketing
100  WHERE pais = 'Germany'
101 ORDER BY PromCompras DESC;
102
100% 27:101
Result Grid Filter Rows: Search Export:
NombreEmpresa Telefono Pais PromCompras
Ac Fermentum Incorporated 06 85 56 52 33 Germany 284.867160
Nunc Interdum Incorporated 05 18 15 48 13 Germany 259.319156
Convallis In Incorporated 06 66 57 29 50 Germany 257.745376
Ac Industries 09 34 65 40 60 Germany 255.147288
Rutrum Non Inc. 02 66 31 61 09 Germany 255.136927
Auctor Mauris Corp 05 62 R7 14 41 Germanv 254.76551R

```

VistaMarketing 4

Action Output	Time	Action	Response
3	18:14:29	SELECT * FROM VistaMarketing WHERE pais = 'Germany'...	8 row(s) returned

Operaciones realizadas

- Consulté la vista VistaMarketing.
- Aplicué un filtro mediante **WHERE** para seleccionar únicamente las empresas cuyo país de residencia es Germany.

Resultado clave

- Se muestran únicamente las empresas con país Germany, junto con su nombre, teléfono, país y promedio de transacciones.

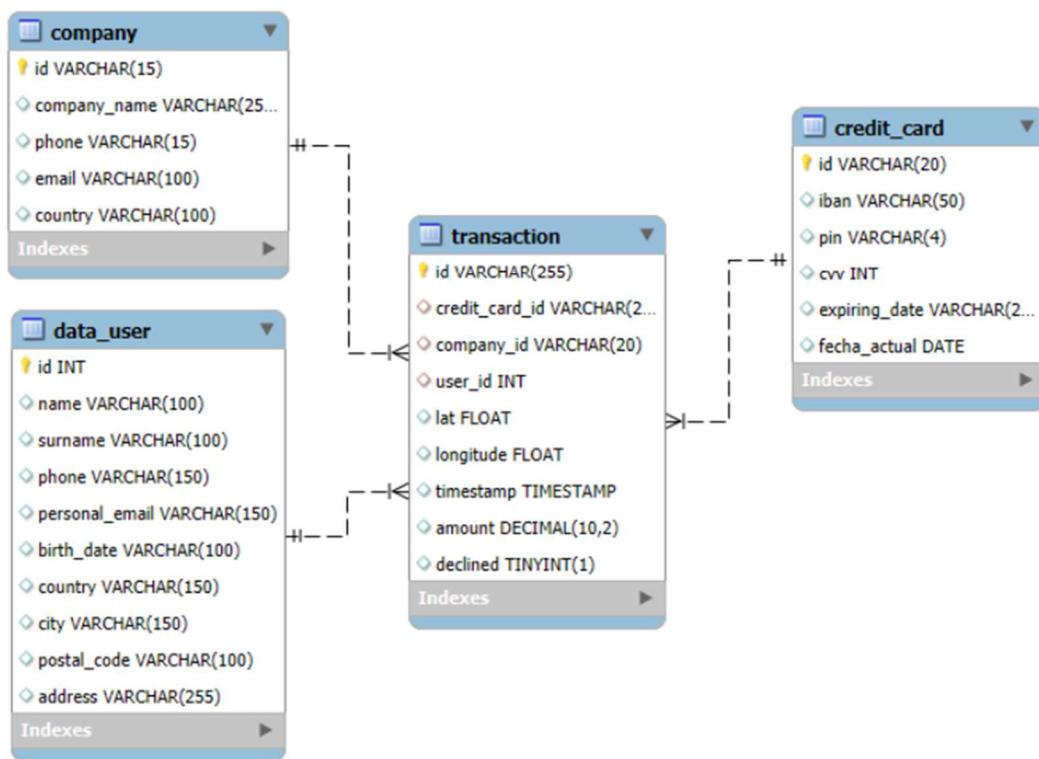
Relevancia

- Permite obtener los registros que cumplen la condición indicada sin modificar la definición de la vista.

 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:



Operaciones realizadas

```

104
105 • CREATE TABLE IF NOT EXISTS data_user (
106   id CHAR(10) PRIMARY KEY,
107   name VARCHAR(100),
108   surname VARCHAR(100),
109   phone VARCHAR(150),
110   email VARCHAR(150),
111   birth_date VARCHAR(100),
112   country VARCHAR(150),
113   city VARCHAR(150),
114   postal_code VARCHAR(100),
115   address VARCHAR(255)
116 );
117
  
```

Action Output: 7 rows affected

Action	Time	Response
CREATE TABLE IF NOT EXISTS data_user (18:37:58	id CHAR(... 0 row(s) affected

- Creé la tabla **data_user** a partir del archivo proporcionado.

```

1 • select count(id)
2 from data_user;
  
```

Result Grid		Filter Rows: <input type="text"/> Search
	count(id)	Resets all sorted columns
	5000	

- Importé los datos correspondientes y verifiqué la carga.

120 -----| Verifico cambios a realizar en Tabla Company

121 • DESCRIBE company;

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	

- Verifico que sobra el campo company.website, por lo tanto usé la cláusula DROP para deshacerme de él.

122 • ALTER TABLE company DROP website;

123 • DESCRIBE company;

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	

Action Output	Time	Action
1	19:11:23	ALTER TABLE company DROP website
2	19:11:27	DESCRIBE company

- Compruebo los cambios realizados

- Verifiqué la estructura final de la tabla mediante DESCRIBE.

124 -----| Verifico cambios a realizar en Tabla data_user

125 • DESCRIBE data_user;

126

Field	Type	Null	Key	Default	Extra
id	char(10)	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
surname	varchar(100)	YES		NULL	
phone	varchar(150)	YES		NULL	
email	varchar(150)	YES		NULL	
birth_date	varchar(100)	YES		NULL	
country	varchar(150)	YES		NULL	
city	varchar(150)	YES		NULL	
postal_code	varchar(100)	YES		NULL	
address	varchar(255)	YES		NULL	

- Encontré que había que renombrar el campo `email`, lo cual hice con la cláusula RENAME y con Modify Column el tipo de id q ha de ser `int` en lugar de `char(10)`. Seguidamente comprobé el cambio realizado.

```
125 • DESCRIBE data_user;
126 • ALTER TABLE data_user RENAME COLUMN email TO personal_email;
127 • ALTER TABLE data_user MODIFY COLUMN id int;
128 • DESCRIBE data_user;
```

129 --- ----- | Verifico cambios a realizar en Tabla data_user

Result Grid Filter Rows: Search Export:

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

Result 11

- Mostré los campos de la tabla Credit_card y realicé los cambios necesarios para que quede según el formato necesario para la presentación.

```
128 --- ----- | Verifico cambios a realizar en Tabla Credit_card
129 • DESCRIBE Credit_card;
```

100% 22:129

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	NULL	
iban	varchar(50)	YES		NULL	
pan	varchar(50)	YES		NULL	
pin	varchar(4)	YES		NULL	
cvv	int	YES		NULL	
expiring_date	varchar(20)	YES		NULL	
fecha_actual	varchar(20)	YES		NULL	

- ```
131 • ALTER TABLE credit_card MODIFY COLUMN fecha_actual DATE;
132 • ALTER TABLE credit_card MODIFY COLUMN id VARCHAR (20);
133 • ALTER TABLE credit_card MODIFY COLUMN pin VARCHAR (4) ;
134 • ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR (10);
```

100% 55:132

Action Output

|     | Time     | Action                                  | Response                                                     |
|-----|----------|-----------------------------------------|--------------------------------------------------------------|
| ✓ 1 | 10:13:40 | ALTER TABLE credit_card MODIFY COLUM... | 5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0 |
| ✓ 2 | 10:14:37 | ALTER TABLE credit_card MODIFY COLUM... | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0       |
| ✓ 3 | 10:14:46 | ALTER TABLE credit_card MODIFY COLUM... | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0       |
| ✓ 4 | 10:14:52 | ALTER TABLE credit_card MODIFY COLUM... | 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0       |

- Comprobé q los cambios solicitados por código quedaran evidenciados en la descripción de la tabla.

- ```
136 • DESCRIBE Credit_card; -- ----- | Verifico cambios realizados en Tabla Credit_card
137
```

100% 55:132

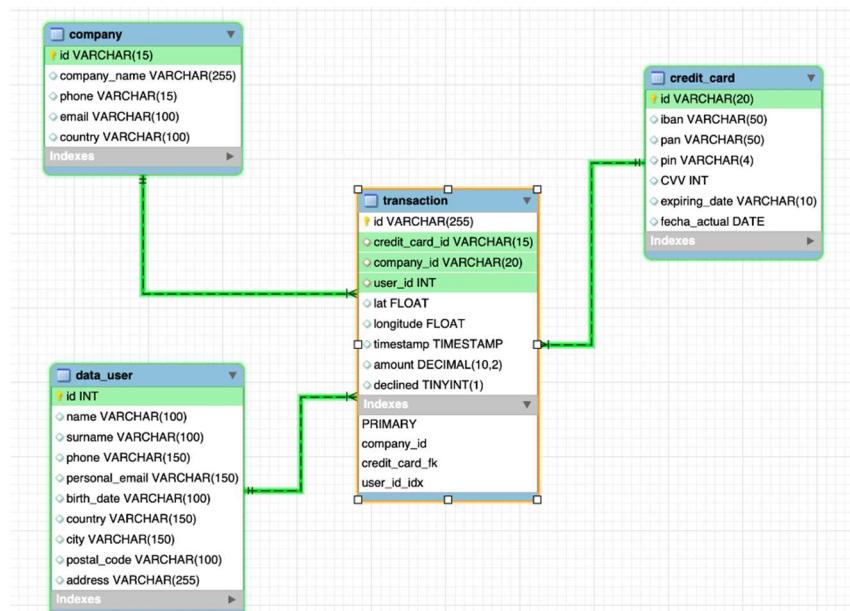
Result Grid Filter Rows: Search Export:

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

Result 10

Action Output

	Time	Action	Response
✓ 1	10:25:39	DESCRIBE Credit_card	7 row(s) returned



Ejercicio 2

La empresa también os pide 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.
- Asegúrese de incluir información relevante de las tablas que conoceréis y utilice alias para cambiar de nombre columnas según sea necesario.

Muestra los resultados de la vista, ordena los resultados de forma descendente en función de la variable ID de transacción.

```

145 • CREATE VIEW InformeTecnico AS
146   SELECT
147     t.id          AS ID,
148     d.name        AS NOMBRE,
149     d.surname     AS APELLIDO,
150     c.iban        AS IBAN,
151     Co.company_name AS EMPRESA
152   FROM transaction t
153   JOIN data_user d
154     ON t.user_id = d.id
155   JOIN Credit_card c
156     ON t.credit_card_id = c.id
157   JOIN company Co
158     ON t.company_id = Co.id;
159
160
161 • SELECT *
162   FROM InformeTecnico
163   ORDER BY t.id DESC;
164
165

```

100% 20:163

Result Grid Filter Rows: Search Export: Fetch rows:

ID	NOMBRE	APELLIDO	IBAN	EMPRESA
FFEE3CCD-570E-4832-A549-B51590ACE...	Tlrxr	Zrqkefve	XX271299628636232349308310	Sapien Nunc Pulvinar LLP
FFED6464-4990-49E5-837F-18E2DDC50765	Mpkkar	Cxkaqbogg	XX669765463980171165992580	Dui Cras Associates
FFED3451-EFEE-4EA1-884A-643C968C95...	Zysgzp	Pprhmopx	XX436742306265236220597816	Tristique Neque Venenatis Ins...
FFECBAA07-4369-4603-88E6-39E612783CCB	Zikgjv	Xacvswcm	XX255104722606040277752468	Ut Semper Foundation
FFEC7298-D1E3-4B14-B3AB-1A397245D...	Agxbvj	Ydlqjbif	XX793894724937164823298857	Mus Aenean Eget Foundation
FFEA573F-E92A-430B-B0BA-8D4B580123...	Amfofp	Gryrdhpp	XX177617019374017529450303	Interdum Feugiat Sed Associa...
FFEA5425-DD40-4861-BA5D-9F19DCE00...	Olnnxq	Rglpnmvp	XX461336779805852619026178	Cras Consulting
FFE8D93D-CD93-4B60-9D07-FAD4478AC...	Pqsyxf	Nrfmetzn	XX499881216073626571196473	Non Justo Corp.
FFE7F6DF-4747-484F-A4D9-307A20C23D...	Dane	Shepard	AE640696354928782425103	Netus Et Malesuada Ltd
FFE7879B-7741-40A3-A53E-ACD9630A1A...	Wggcyy	Nrwjfiqs	XX203497096655051171132884	Augue Foundation
FFE4CB2C-DF9A-482B-82B1-166F860D13...	Wbebaο	Iaamqvrx	XX844933261515037464617096	Ac Libero Inc.
FFE46A91-7183-4F01-9D15-905E96FF8801	Cakzav	Mycdeeyk	XX177535118533621174262541	Dictum Eu Corp.
FFE23DC5-048A-478D-AA8E-F8D0228B7...	Dsrcqz	Xdeedtup	XX619742474117426975704055	Tempor Diam Institute
FFE1AA6E-F08A-43F8-A156-B0813D4F23E5	Diymej	Thoeaumy	XX29436503404677987288112	Tempor Diam Institute

Operaciones realizadas:

- Creé la vista InformeTecnico.
- Realicé los JOIN necesarios entre transaction, data_user, credit_card y company.
- Seleccioné los campos solicitados en el enunciado.

Resultado clave

- La vista devuelve el identificador de la transacción, el nombre y apellido del usuario, el IBAN y el nombre de la empresa, ordenados por identificador de forma descendente.

Relevancia

- Devuelve la información solicitada en una única vista.