

SPRINT 3

DATA ANALYST

RAFAEL FONS BIBILONI

13-Mayo-2025

NIVEL I - EJERCICIO 1:

```
4 NIVEL I - Ejercicio 1
5 Tu tarea es diseñar y crear una tabla llamada "credit_card" que almacene detalles cruciales sobre las tarjetas de crédito. La nueva tabla
6 debe ser capaz de identificar de forma única cada tarjeta y establecer una relación adecuada con las otras dos tablas ("transaction" y "company").
7 Después de crear la tabla será necesario que ingreses la información del documento denominado "datos_introducir_credit". Recuerda mostrar el
8 diagrama y realizar una breve descripción del mismo.
9 */
10
11 • CREATE TABLE credit_card(
12     id VARCHAR(20) NOT NULL PRIMARY KEY,
13     iban VARCHAR(40) UNIQUE,
14     pan VARCHAR(20),
15     pin VARCHAR(4),
16     cvv VARCHAR(3),
17     expiring_date DATE
18 );
19
20 -- modiflico el tipo de dato del campo "expiring_date" porque no cumple con el formato de fecha de mysql = YYYY-MM-DD
21 • ALTER TABLE credit_card
22     MODIFY COLUMN expiring_date VARCHAR(8);
23
24 -- agreggo la relacion entre el PK de "credit_card" y la FK de "transaction":
25 • ALTER TABLE transaction
26     ADD FOREIGN KEY (credit_card_ID) REFERENCES credit_card(id);
27
28 • DESCRIBE credit_card;
```

Result Grid Filter Rows: Search Export:

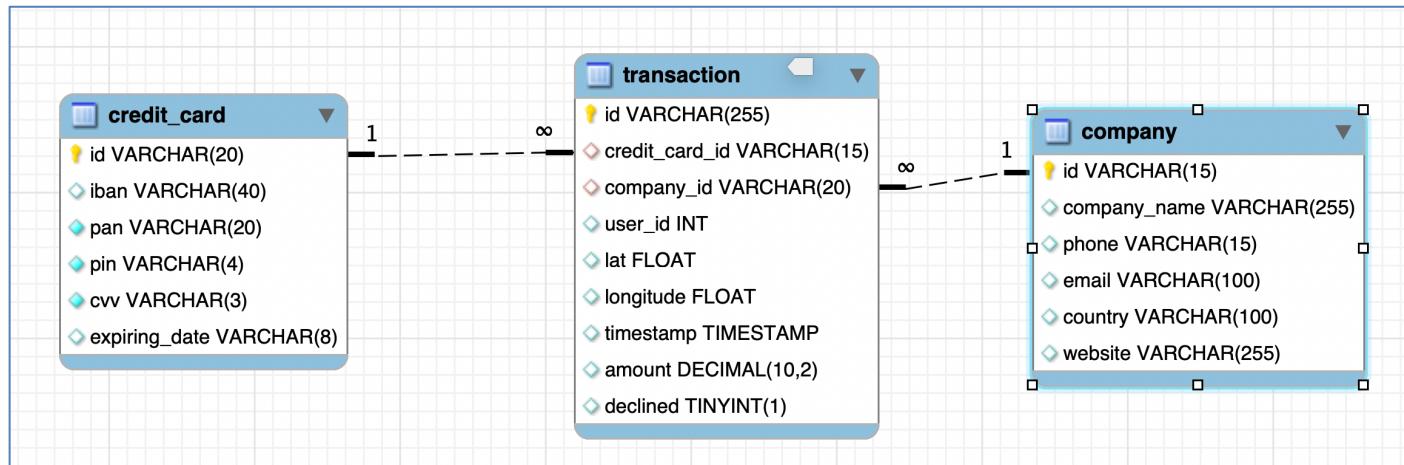
Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI	NULL	
iban	varchar(40)	YES	UNI	NULL	
pan	varchar(20)	YES		NULL	
pin	varchar(4)	YES		NULL	
cvv	varchar(3)	YES		NULL	
expiring_date	varchar(8)	YES		NULL	

credit_card 1 Result 7

Action Output

Time	Action	Response
387 16:59:01	DESCRIBE credit_card	6 row(s) returned

El modelo contiene ahora tiene 3 tablas:



La nueva tabla “credit_card” están vinculadas por el campo “**credit_card.id**” (PRIMARY KEY) = “**transaction.credit_card_id**” (FOREIGN KEY) en una relación 1 a muchos. Es decir, en la tabla transaction puede contener muchos registros de una misma tarjeta de crédito.

La línea discontinua refiere a que la relación no es rígida/estricta. Es decir que la FK no es obligatoria (permite NULL).

NIVEL I - EJERCICIO 2:

```

30  /*
31   * NIVEL 1 - Ejercicio 2
32   * El departamento de Recursos Humanos ha identificado un error en el número de cuenta del usuario con ID CcU-2938.
33   * La información que debe mostrarse para este registro es: R323456312213576817699999. Recuerda mostrar que el cambio se realizó.
34   */
35
36   -- primero verifico que el usuario exista
37 • SELECT *
38   FROM credit_card
39   WHERE id="CcU-2938";
40
41   -- actualizo los dato del usuario
42 • UPDATE credit_card
43   SET iban="R323456312213576817699999"
44   WHERE id="CcU-2938";
45

```

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

id	iban	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	3257	984	10/30/22
HULL	HULL	HULL	HULL	HULL

credit_card 2

Action Output

Time	Action	Response
6 11:00:34	SELECT * FROM credit_card WHERE id="CcU-2938" LIMIT 0, 1000	1 row(s) returned
7 11:00:34	UPDATE credit_card SET iban="R323456312213576817699999" WHERE id="CcU-2938"	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0

NIVEL I - EJERCICIO 3:

```

42  /*
43   * NIVEL 1 - Ejercicio 3
44   * En la tabla "transaction" ingresa un nuevo usuario con la siguiente información:
45   */
46
47   -- primero agrego la compañía "b-9999":
48 • INSERT INTO company(id, company_name, phone, email, country, website)
49   VALUES ("b-9999", "", "", "", "", "");
50
51   -- primero agrego la tarjeta "CcU-9999":
52 • INSERT INTO credit_card(id, iban, pan, pin, cvv, expiring_date)
53   VALUES ("CcU-9999", "", "", "", "", "");
54
55 • INSERT INTO transaction(id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined)
56   VALUES ("108B1D1D-5B23-A76C-55EF-C568E49A99DD", "CcU-9999", "b-9999", "9999", 829.999, -117.999, 2025-05-09 00:00:00, 111.11, 0);
57
58 • SELECT *
59   FROM transaction
60   WHERE id="108B1D1D-5B23-A76C-55EF-C568E49A99DD";
61

```

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	2025-05-09 00:00:00	111.11	0
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

X credit_card 1 transaction 6

Action Output

Time	Action	Response
383 16:56:25	DELETE FROM credit_card... 1 row(s) affected	
384 16:56:29	INSERT INTO credit_card(... 1 row(s) affected	
385 16:56:36	INSERT INTO transaction(... 1 row(s) affected	
386 16:56:40	SELECT * FROM transactio... 1 row(s) returned	

NIVEL I - EJERCICIO 4:

```

66  /*
67   NIVEL I - Ejercicio 4
68   Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla credit_card. Recuerda mostrar el cambio realizado
69  */
70   -- primero veo los campos de la tabla para ver si existe y verificar su nombre:
71 • DESCRIBE credit_card;
72
73   -- elimino el campo "pan"
74 • ALTER TABLE credit_card
75   DROP pan;
76
77   -- verifico la estructura final de la tabla:
78 • DESCRIBE credit_card;
79
00% 9:62

```

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI	HULL	
iban	varchar(40)	YES	UNI	HULL	
pin	varchar(4)	YES		HULL	
cvv	varchar(3)	YES		HULL	
expiring_date	varchar(8)	YES		HULL	

Result 4 Result 5

Action Output

Time	Action	Response
11:05:26	DESCRIBE credit_card	6 row(s) returned
11:05:26	ALTER TABLE credit_card DROP pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
11:05:26	DESCRIBE credit_card	5 row(s) returned

NIVEL II - EJERCICIO 1:

```

80  /*
81   NIVEL II - Ejercicio 1
82   Elimina de la tabla transacción el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de datos.
83  */
84   -- verifico que el registro exista:
85 • SELECT *
86   FROM transaction
87   WHERE id="02C6201E-D90A-1859-B4EE-88D2986D3B02";
88
89   -- elimino el registro:
90 • DELETE FROM transaction
91   WHERE id="02C6201E-D90A-1859-B4EE-88D2986D3B02";
92
93   -- verifico que realmente se haya eliminado el registro:
94 • SELECT* FROM transaction
95   WHERE id="02C6201E-D90A-1859-B4EE-88D2986D3B02";
96
00% 1:76

```

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

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

transaction 8 transaction 9

Action Output

Time	Action	Response
11:09:26	SELECT * FROM transaction WHERE id="02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	1 row(s) returned
11:09:26	DELETE FROM transaction WHERE id="02C6201E-D90A-1859-B4EE-88D2986D3B02"	1 row(s) affected
11:09:26	SELECT* FROM transaction WHERE id="02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	0 row(s) returned

NIVEL II - EJERCICIO 2:

```
86 | NIVEL II – Ejercicio 2
87 | La sección de marketing desea tener acceso a información específica para realizar análisis y estrategias efectivas.
88 | Se ha solicitado crear una vista que proporcione detalles clave sobre las compañías y sus transacciones. Será necesaria que crees
89 | una vista llamada VistaMarketing que contenga la siguiente información: Nombre de la compañía. Teléfono de contacto. País de residencia.
90 | Media de compra realizado por cada compañía. Presenta la vista creada, ordenando los datos de mayor a menor promedio de compra.
91 | */
92 |
93 | -- creo la vista solicitada:
94 • CREATE VIEW VistaMarketing AS
95 | SELECT c.company_name, c.phone, c.country, ROUND(AVG(t.amount),2) AS media_compra
96 | FROM transaction t
97 | JOIN company c
98 | ON t.company_id = c.id
99 | GROUP BY c.company_name, c.phone, c.country
100 | ORDER BY media_compra DESC;
101 |
102 | -- muestro la vista
103 • SELECT * FROM vistamarketing;
```

Result Grid Filter Rows: Search Export:

company_name	phone	country	media_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 Tinidunt Duis Institute	05 35 93 32 44	Netherlands	442.52
Viverra Donec Foundation	03 33 12 32 73	United Kingdom	442.28
Vestibulum Lorem PC	02 02 87 33 40	Belgium	434.06
Aliquet Diam Limited	02 76 61 47 46	United States	425.64
Maecenas Malesuada Fringilla Inc.	09 38 53 76 61	Netherlands	408.62
Non Ante LLP	08 89 47 65 08	Sweden	407.79
Egestas Nunc Sed Limited	06 01 02 70 47	Italy	406.11
Nunc Sit Incorporated	07 28 42 63 63	Norway	405.36
Mauris Aenean Industries	04 14 44 64 62	Australia	396.32

Action Output

Time	Action	Response
10:40:07	CREATE VIEW VistaMarketing AS SELECT c.company_name, c.phone, c.country, ROUND(AVG(t.amount),2) AS media_compra FROM transaction t JOIN comp...	0 row(s) affected
10:40:44	SELECT * FROM sprint3.vistamarketing LIMIT 0, 1000	101 row(s) returned
10:41:12	SELECT * FROM vistamarketing LIMIT 0, 1000	101 row(s) returned

SPRINT 3

DATA ANALYST

RAFAEL FONS BIBILONI
13-Mayo-2025

NIVEL II - EJERCICIO 3:

```
107  /*
108   NIVEL II - Ejercicio 3
109   Filtra la vista VistaMarketing para mostrar sólo las compañías que tienen su país de residencia en "Germany"
110  */
111
112 •  SELECT *
113   FROM vistamarketing
114   WHERE country = "Germany";
115
```

Result Grid | Filter Rows: Search | Export:

company_name	phone	country	media_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
Ac Fermentum Incorporated	06 85 56 52 33	Germany	206.47
Auctor Mauris Corp.	05 62 87 14 41	Germany	184.31
Convallis In Incorporated	06 66 57 29 50	Germany	156.73

vistamarketing 5

Action Output | Time | Action | Response

13	10:43:24	SELECT * FROM vistamarketing WHERE country = "Germany" LIMIT 0, 1000	8 row(s) returned
----	----------	--	-------------------

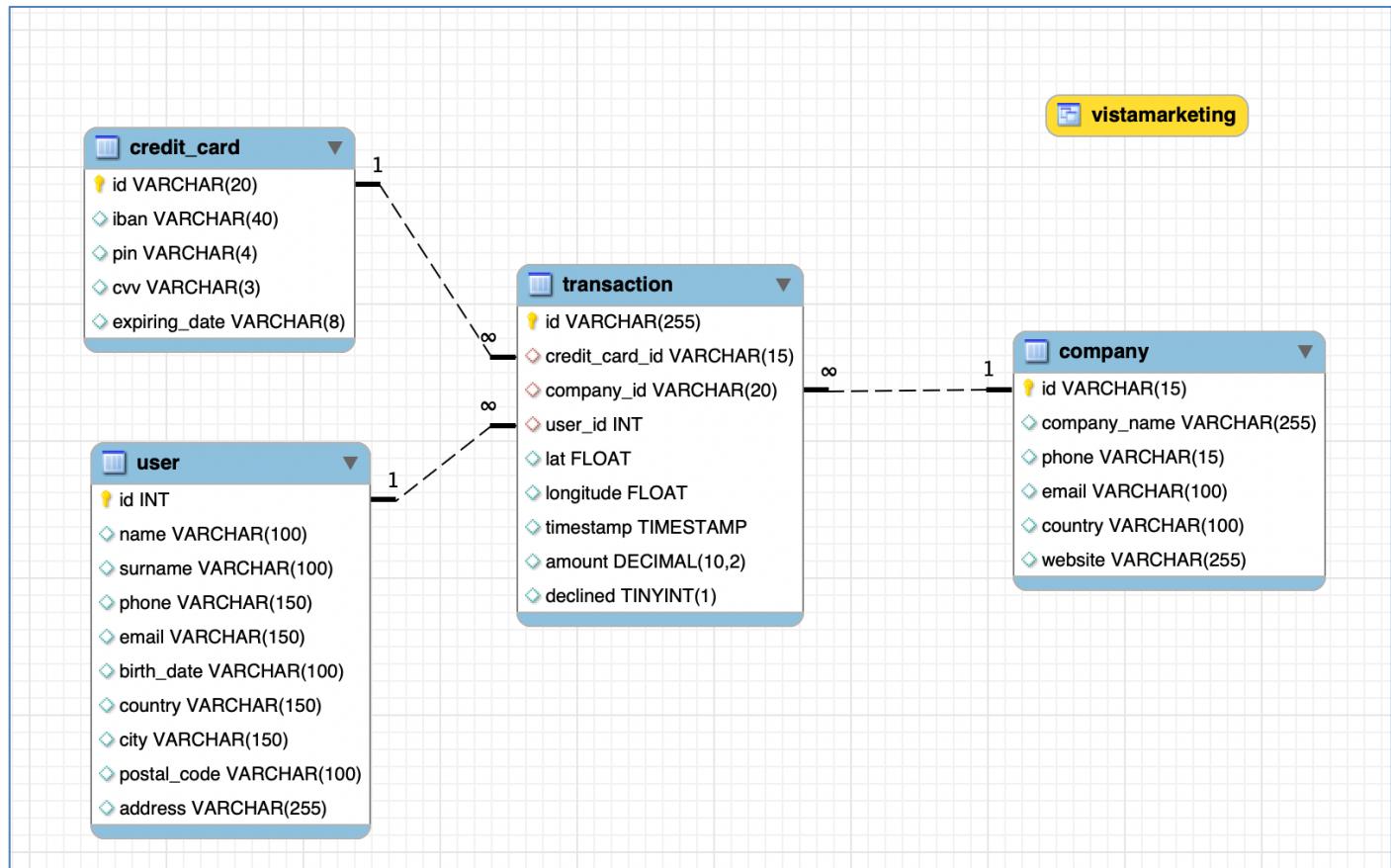
NIVEL III - EJERCICIO 1:

```
15  NIVEL III - Ejercicio 1|
16  La próxima semana tendrás una nueva reunión con los gerentes de marketing. Un compañero de tu equipo realizó modificaciones en la base de datos,
17  pero no recuerda cómo las realizó. Te pide que le ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama:
18  */
19
20  -- Creamos el índice para el campo "user_id" de la tabla transaction. De lo contrario no puedo agregar la vinculación de l FK centre ambas tablas.
21 •  CREATE INDEX idx_user_id ON transaction(user_id);
22  -- Ahora si podemos crear la tabla user
23 •  CREATE TABLE IF NOT EXISTS user (
24      id INT PRIMARY KEY,
25      name VARCHAR(100),
26      surname VARCHAR(100),
27      phone VARCHAR(150),
28      email VARCHAR(150),
29      birth_date VARCHAR(100),
30      country VARCHAR(150),
31      city VARCHAR(150),
32      postal_code VARCHAR(100),
33      address VARCHAR(255),
34      FOREIGN KEY(id) REFERENCES transaction(user_id)
35  );
36
37  -- la relación entre user y transaction es incorrecta. Debería se 1:n de user a transaction.
38 •  ALTER TABLE transaction
39  ADD FOREIGN KEY(user_id) REFERENCES user(id);
40
41  -- elimino la restriccion de FK de la tabla "user" que estaban en el script original. Primero ejecute un "show create table" para ver el nombre de
42 •  ALTER TABLE user
43  DROP CONSTRAINT user_ibfk_1;
44
45  -- verifico que la constraint haya sido borrada
46 •  SHOW CREATE table user;
47
48  -- inserto los datos en la tabla user usando el script "datos_introducir_user (1)"
```

Action Output | Time | Action | Response

300	11:08:04	SELECT * FROM sprint3.user LIMIT 0, 1000	275 row(s) returned
-----	----------	--	---------------------

A continuación copio el modelo ahora con las 4 tablas, pero la estructura del modelo de datos difieren con el indicado en el ejercicio.



Se debe modificar varios aspectos de la estructura del modelo de datos:

1. Cambiar nombre tabla “user” por “data_user”:

```

161      # 1.Cambiar nombre tabla "user" por "data_user"
162 •      RENAME TABLE user TO data_user;
163
164 •      DESCRIBE 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	
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 10

tion Output

Time	Action	Response
21 11:09:26	SELECT* FROM transaction WHERE id="02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	0 row(s) returned
22 11:22:30	RENAME TABLE user TO data_user	0 row(s) affected
23 11:22:55	DESCRIBE data_user	10 row(s) returned

2. Cambiar el nombre del campo “email” de la tabla “user” por “personal_email”

```

166      # 2.Cambiar el nombre del campo "email" de la tabla "user" por "personal_email"
167 •      ALTER TABLE data_user
168      CHANGE email personal_email VARCHAR(150);
169
170 •      DESCRIBE data_user;
171
00%   55:160 | 1 error found

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

Action Output

Time	Action	Response
24 11:25:48	ALTER TABLE data_user CHANGE email personal_email VARCHAR(150)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
25 11:25:48	DESCRIBE data_user	10 row(s) returned

3. Eliminar campo “website” de la tabla “company”

```

172      # 3.Eliminar campo "website" de la tabla "company"
173 •      DESCRIBE company;
174
175 •      ALTER TABLE company
176      DROP website;
177
178 •      DESCRIBE company;
00%   34:162 | 1 error found

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	

Result 13 Result 14

Action Output

Time	Action	Response
27 11:27:44	DESCRIBE company	6 row(s) returned
28 11:27:44	ALTER TABLE company DROP website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
29 11:27:44	DESCRIBE company	5 row(s) returned

4. Agregar el campo “fecha_actual DATE” en la tabla “credit_card”

```

180      # 4.Agregar el campo "fecha_actual DATE" en la tabla "credit_card"
181 •      ALTER TABLE credit_card
182      ADD fecha_actual DATE;
183
184 •      describe credit_card;
185
00%   9:171 | 1 error found

Result Grid Filter Rows: Search Export:

```

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI	NULL	
iban	varchar(40)	YES	UNI	NULL	
pin	varchar(4)	YES		NULL	
cvv	varchar(3)	YES		NULL	
expiring_date	varchar(8)	YES		NULL	
fecha_actual	date	YES		NULL	

Result 15

Action Output

Time	Action	Response
30 11:29:16	ALTER TABLE credit_card ADD fecha_actual DATE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
31 11:29:16	describe credit_card	6 row(s) returned

5. Cambiar algunos tipos de datos de los campos de la tabla “credit_card”:

- a. IBAN VARCHAR(50)
- b. expiring_date VARCHAR(20)

```

182      # 5.Cambiar algunos tipos de datos de los campos de la tabla "credit_card":
183 •      ALTER TABLE credit_card
184          CHANGE iban IBAN VARCHAR(50),
185          CHANGE expiring_date expiring_date VARCHAR(20);
186
187 •      describe credit_card;

```

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI	NULL	
IBAN	varchar(50)	YES	UNI	NULL	
pin	varchar(4)	YES		NULL	
cvv	varchar(3)	YES		NULL	
expiring_date	varchar(20)	YES		NULL	
fecha_actual	date	YES		NULL	

Result 18

Action Output

Time	Action	Response
40 11:44:12	describe credit_card	6 row(s) returned

C. CVV INT

```

189      -- para modificar el campo "cvv" debo modificar el registro que ingrese con id "CcU-9999", porque en ese campo ingrese un string vacío "".
190      -- primero actualizo ese valor:
191 •      UPDATE credit_card
192          SET cvv=NULL WHERE id = "CcU-9999";
193
194 •      ALTER TABLE credit_card
195          CHANGE cvv cvv INT;
196
197 •      describe credit_card;
198 /* 15:187 */

```

Result Grid Filter Rows: Search Export:

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

Result 19

Action Output

Time	Action	Response
41 11:45:47	UPDATE credit_card SET cvv=NULL WHERE id = "CcU-9999"	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
42 11:45:47	ALTER TABLE credit_card CHANGE cvv cvv INT	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
43 11:45:47	describe credit_card	6 row(s) returned

NIVEL III - EJERCICIO 2:

```

199 NIVEL III - Ejercicio 2
200 La empresa también te solicita crear una vista llamada "InformeTecnico" que contenga la siguiente información:
201
202 ID de la transacción
203 Nombre del usuario/a
204 Apellido del usuario/a
205 IBAN de la tarjeta de crédito usada.
206 Nombre de la compañía de la transacción realizada.
207 Asegúrate de incluir información relevante de ambas tablas y utiliza alias para cambiar de nombre columnas según sea necesario.
208 Muestra los resultados de la vista, ordena los resultados de forma descendente en función de la variable ID de transacción.
209 */
210 -- creo la vista
211 • CREATE VIEW InformeTecnico AS
212     SELECT t.id AS id_transaction, u.name AS user_name, u.surname AS user_surname, cd.iban AS IBAN, c.company_name AS company_name
213     FROM transaction t
214     JOIN data_user u
215     ON t.user_id = u.id
216     JOIN credit_card cd
217     ON t.credit_card_id = cd.id
218     JOIN company c
219     ON t.company_id = c.id
220 ;

```

Result Grid Filter Rows: Search Export:

id_transaction	user_name	user_surname	IBAN	company_name
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9C8CCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-B8083FB03187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCF9A9QA-071D-2BDC-0E49-8DD92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated

informetecnico 20

Action Output

Time	Action	Response
47 11:48:20	CREATE VIEW InformeTecnico AS SELECT t.id AS id_transaction, u.name AS user_name, u.surname AS user_surname, cd.iban AS IBAN, c.compa...	0 row(s) affected

Muestro el view creado:

```

222 -- muestro la vista y ordeno por ID:
223 • SELECT *
224     FROM informetecnico
225     ORDER BY id_transaction DESC;
226

```

Result Grid Filter Rows: Search Export:

id_transaction	user_name	user_surname	IBAN	company_name
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9C8CCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-B8083FB03187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCF9A9QA-071D-2BDC-0E49-8DD92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated

informetecnico 20

Action Output

Time	Action	Response
48 11:48:48	SELECT * FROM informetecnico ORDER BY id_transaction DESC LIMIT 0, 1000	586 row(s) returned