

SPRINT 3

Descripció

En aquest sprint, es simula una situació empresarial en la qual has de realitzar diverses manipulacions en les taules de la base de dades. Al seu torn, hauràs de treballar amb índexs i vistes. En aquesta activitat, continuaràs treballant amb la base de dades que conté informació d'una empresa dedicada a la venda de productes en línia. En aquesta tasca, començaràs a treballar amb informació relacionada amb targetes de crèdit.



Nivell 1

- Exercici 1

La teva tasca és dissenyar i crear una taula anomenada "credit_card" que emmagatzemi detalls crucials sobre les targetes de crèdit. La nova taula ha de ser capaç d'identificar de manera única cada targeta i establir una relació adequada amb les altres dues taules ("transaction" i "company"). Després de crear la taula serà necessari que ingressis la informació del document denominat "dades_introduir_credit". Recorda mostrar el diagrama i realitzar una breu descripció d'aquest.

```
1  -- Crear la taula credit_card
2
3  CREATE TABLE IF NOT EXISTS credit_card (
4      id VARCHAR(15) PRIMARY KEY,
5      iban VARCHAR(34),
6      pan VARCHAR(23),
7      pin VARCHAR(4),
8      cvv VARCHAR(4),
9      expiring_date VARCHAR(10)
10 );
```

Output

#	Time	Action	Message
1	22:34:54	CREATE TABLE IF NOT EXISTS credit_card (id VARCHAR(15) PRIMARY ...	0 row(s) affected

```

2  -- Insertamos datos de credit_card
3  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2938', 'TR301950312213576817638661', '5424465566813633',
4  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2945', 'D026854763748537475216568689', '5142423821948828'
5  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2952', 'B645IVQL52710525608255', '4556 453 55 5287', '455
6  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2959', 'CR7242477244335841535', '372461377349375', '3583'
7  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2966', 'BG72LKTQ15627628377363', '448566 886747 7265', '4
8  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2973', 'PT87806228135092429456346', '544 58654 54343 384'
9  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2980', 'DE39241881883086277136', '402400 7145845969', '56
10 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2987', 'GE89681434837748781813', '3763 747687 76666', '21
11 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2994', 'BH62714428368066765294', '344283273252593', '7545
12 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3001', 'CY49087426654774581266832110', '511722 924833 224
13 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3008', 'LU507216693616119230', '4485744464433884', '1856'
14 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3015', 'PS119398216295715968342456821', '3784 662233 1736
15 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3022', 'GT91695162850556977423121857', '5164 1379 4842 35
16 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3029', 'AZ62317413982441418123739746', '3429 279566 77631
17 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3036', 'AZ39336002925842865843941994', '3768 451556 48766

```

Después de insertar los datos en la tabla `credit_card` crearemos la relación entre las tablas `credit_card` y `transaction`. Las dos tablas se relacionan mediante la columna `id` de la tabla `credit_card` y la columna `credit_card_id` de la tabla `transaction` y definiendo la Foreign Key en la tabla `transaction` se crea esta relación.

```

14  -- Crear FK en la tabla transaction de la PK de la tabla credit_card
15  • ALTER TABLE transaction
16  ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);

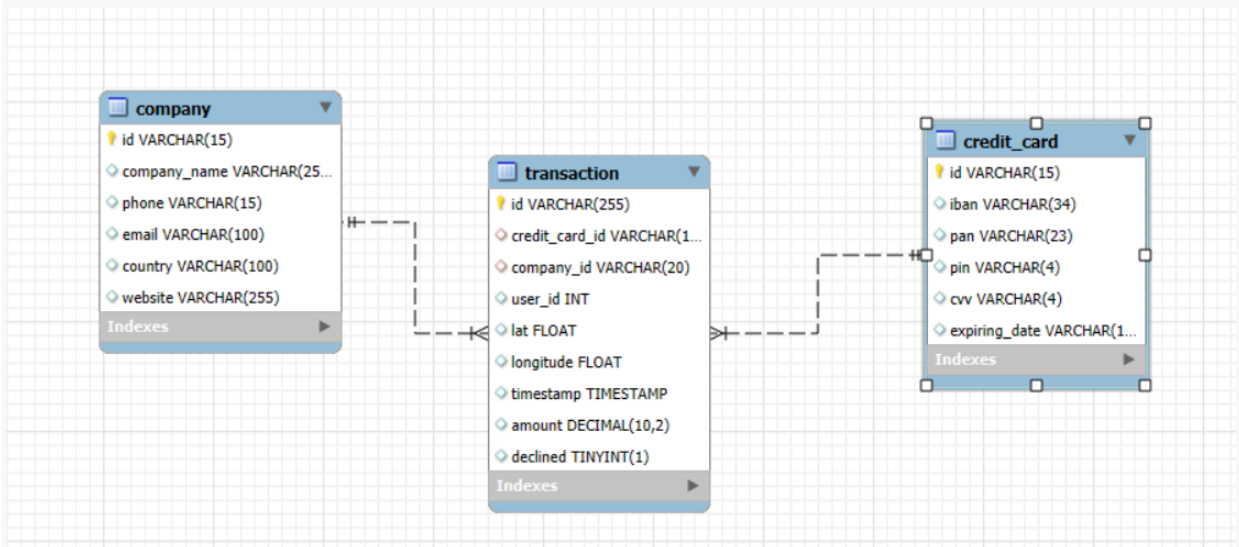
```

Output

Action Output

#	Time	Action	Message
1	23:15:16	ALTER TABLE transaction ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

El diagrama



Tenemos un esquema en estrella con la tabla **transaction** como **tabla de hechos** y las tablas **company** y **credit card** como tablas de **dimensiones**.

La tabla **company** se conecta con la tabla transaction mediante las variables **id** de la tabla company y **company_id** de la tabla transaction con una relación de **1:N** ya que una empresa puede realizar muchas transacciones. La tabla incluye las siguientes variables, descritas con detalle en el sprint anterior:

- id: tipo VARCHAR, es la **PK** de la tabla
- company_name: tipo VARCHAR
- phone: tipo VARCHAR
- email: tipo VARCHAR
- country: tipo VARCHAR
- website: tipo VARCHAR

La tabla **credit_card**, se relaciona con la tabla transaction con una relación de **1:N** ya que con una tarjeta de crédito se pueden realizar muchas transacciones. La relación se realiza mediante las variables **id** de la tabla credit_card y **credit_card_id** de la tabla transaction. La tabla incluye las variables:

- id: el identificador único de cada tarjeta de crédito, tipo VARCHAR, es la **PK** de la tabla
- iban: el número de cuenta bancaria asociada con cada tarjeta, el tipo de la variable es VARCHAR
- pan: el número identificador la cada tarjeta de crédito, tipo VARCHAR
- Pin: el código de seguridad de la tarjeta, tipo VARCHAR
- cvv: el código de seguridad de la tarjeta, tipo INT
- expiring_date: la fecha de caducidad de la tarjeta, tipo VARCHAR

La tabla **transaction** incluye las siguientes variables:

- id: tipo VARCHAR, es la **PK** de la tabla
- credit_card_id: tipo VARCHAR, es **FK** y conecta la tabla con la tabla credit_card
- company_id: tipo VARCHAR, es **FK** y conecta la tabla con la tabla company
- user_id: tipo INT
- lat: tipo FLOAT
- longitude: tipo FLOAT
- timestamp: tipo TIMESTAMP
- amount: tipo DECIMAL
- declined: tipo TINYINT

- Exercici 2

El departament de Recursos Humans ha identificat un error en el número de compte de l'usuari amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és: R323456312213576817699999. Recorda mostrar que el canvi es va realitzar.

```

18  -- Actualizar el número de cuenta del usuario con ID CcU-2938 a R323456312213576817699999
19  • UPDATE credit_card
20  SET iban = "R323456312213576817699999"
21  WHERE id = "CcU-2938";
22

```

Output

Action Output

#	Time	Action	Message
✓ 1	10:37:23	UPDATE credit_card SET iban = "R323456312213576817699999" WHERE id = "CcU-2938"	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

```

23  -- Mostar el cambio realizado
24  • SELECT *
25  FROM credit_card
26  WHERE id = "CcU-2938";

```

Result Grid

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

credit card 2 ×

Output

Action Output

#	Time	Action	Message
✓ 1	10:40:31	SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 1000	1 row(s) returned

- Exercici 3

En la taula "transaction" ingressa un nou usuari amb la següent informació:

Id	108B1D1D-5B23-A76C-55EF-C568E49A99DD
credit_card_id	CcU-9999
company_id	b-9999
user_id	9999
lat	829.999
longitude	-117.999
amount	111.11

declined	0
----------	---

```

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

```

Output

#	Time	Action	Message
1	11:25:05	INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ("1...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (transactions.transaction, CO...

Para poder añadir los datos del nuevo usuario se tienen que actualizar las PK de las tablas company y credit_card (id) añadiendo los valores correspondientes.

Este paso es necesario por la relación entre las tablas, insertar un nuevo valor en una FK cuando el mismo valor no existe en la PK de la tabla de referencia no es posible.

Primero insertaré en la tabla company el nuevo valor b-9999 en id.

Luego insertaré en la tabla credit_card el nuevo valor CcU-9999 en id

Como paso final, añadiré los datos del nuevo usuario en la tabla transaction

```

36 • INSERT INTO company (id) VALUE ("b-9999");
37 • INSERT INTO credit_card (id) VALUE ("CcU-9999");
38 • INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
39 VALUES ("108B1D1D-5B23-A76C-55EF-C568E49A99DD", "CcU-9999", "b-9999", "9999", "829.999", "-117.999", "111.11", "0");

```

Output

#	Time	Action	Message
1	11:26:20	INSERT INTO company (id) VALUE ("b-9999")	1 row(s) affected
2	11:26:20	INSERT INTO credit_card (id) VALUE ("CcU-9999")	1 row(s) affected
3	11:26:20	INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ("1...	1 row(s) affected

```

41 • SELECT *
42 FROM transaction
43 WHERE credit_card_id = "CcU-9999";

```

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	NULL	111.11	0
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 3

Output

#	Time	Action	Message
1	11:29:00	SELECT * FROM transaction WHERE credit_card_id = "CcU-9999" LIMIT 0, 1000	1 row(s) returned

- Exercici 4

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

```
45  -- eliminar columna pan de la tabla credit_card
46 • ALTER TABLE credit_card
47  DROP COLUMN pan;
48  -- mostrar el cambio en la tabla
49 • SHOW COLUMNS
50  FROM credit_card;
51
```

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

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

Result 1 x

Output

Action Output

#	Time	Action	Message
✓ 1	23:17:48	SHOW COLUMNS FROM credit_card	6 row(s) returned



Nivell 2

Exercici 1

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

```
52  -- eliminar de la tabla transaction el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02
53 • DELETE FROM transaction
54  WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
```

Output

Action Output

#	Time	Action	Message
✓ 1	11:39:31	DELETE FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02"	1 row(s) affected

```

56  -- revisar si el registro se eliminó correctamente
57  •  SELECT *
58     FROM transaction
59     WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";

```

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	NULL

transaction 5 x

Output

Action Output

#	Time	Action	Message
1	11:41:16	SELECT * FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	0 row(s) returned

Exercici 2

La secció de màrqueting desitja tenir accés a informació específica per a realitzar anàlisi i estratègies efectives. S'ha sol·licitat crear una vista que proporcioni detalls clau sobre les companyies i les seves transaccions. Serà necessària que creïs una vista anomenada VistaMarketing que contingui la següent informació: Nom de la companyia. Telèfon de contacte. País de residència. Mitjana de compra realitzat per cada companyia. Presenta la vista creada, ordenant les dades de major a menor mitjana de compra.

```

62 -- Crear VistaMarketing
63 • CREATE VIEW VistaMarketing AS
64 SELECT c.company_name, c.phone, c.country, ROUND(AVG(tr.amount),2) AS avg_purchase
65 FROM company c
66 JOIN transaction tr
67 ON c.id = tr.company_id
68 WHERE tr.declined = 0
69 GROUP BY c.id
70 ORDER BY avg_purchase DESC;
71 -- Mostrar VistaMarketing
72 • SELECT *
73 FROM VistaMarketing;
74

```

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

	company_name	phone	country	avg_purchase
▶	Eget Ipsum Ltd	03 67 44 56 72	United States	481.86
	Sed Id Limited	07 28 18 18 13	United States	477.51
	Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.10
	Nunc Sit Incorporated	07 28 42 63 63	Norway	461.83
	Non Magna LLC	06 71 73 13 17	United Kingdom	458.74
	Maecenas Malesuada Fringilla Inc.	09 38 53 76 61	Netherlands	451.29
	Erat LLP	03 18 88 77 79	Netherlands	448.44
	Tortor Nunc Commoda Company	05 35 92 77 16	United States	447.11
	Tristique Eget Arcu Ltd	08 47 56 71 52	Italy	444.16

VistaMarketing 2 x

Output

Action Output ▼

#	Time	Action	Message
1	23:23:49	CREATE VIEW VistaMarketing AS SELECT c.company_name, c.phone, c.country, ROUND(AVG(tr.amount),2) ...	0 row(s) affected
2	23:23:49	SELECT * FROM VistaMarketing LIMIT 0, 1000	101 row(s) returned

Exercici 3

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


```

76  -- Mostrar de VistaMarketing solo las empresas de Alemania
77  •  SELECT *
78      FROM VistaMarketing
79      WHERE country = "Germany";

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
company_name	phone	country	avg_purchase
Ac Industries	09 34 65 40 60	Germany	396.150000
Auctor Mauris Corp.	05 62 87 14 41	Germany	308.990000
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.570000
Aliquam PC	01 45 73 52 16	Germany	280.340000
Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000
Nunc Interdum Incorporated	05 18 15 48 13	Germany	242.947692
Convallis In Incorporated	06 66 57 29 50	Germany	60.990000
Augue Foundation	06 88 43 15 63	Germany	15.050000

VistaMarketing 7 x

Output :

Action Output

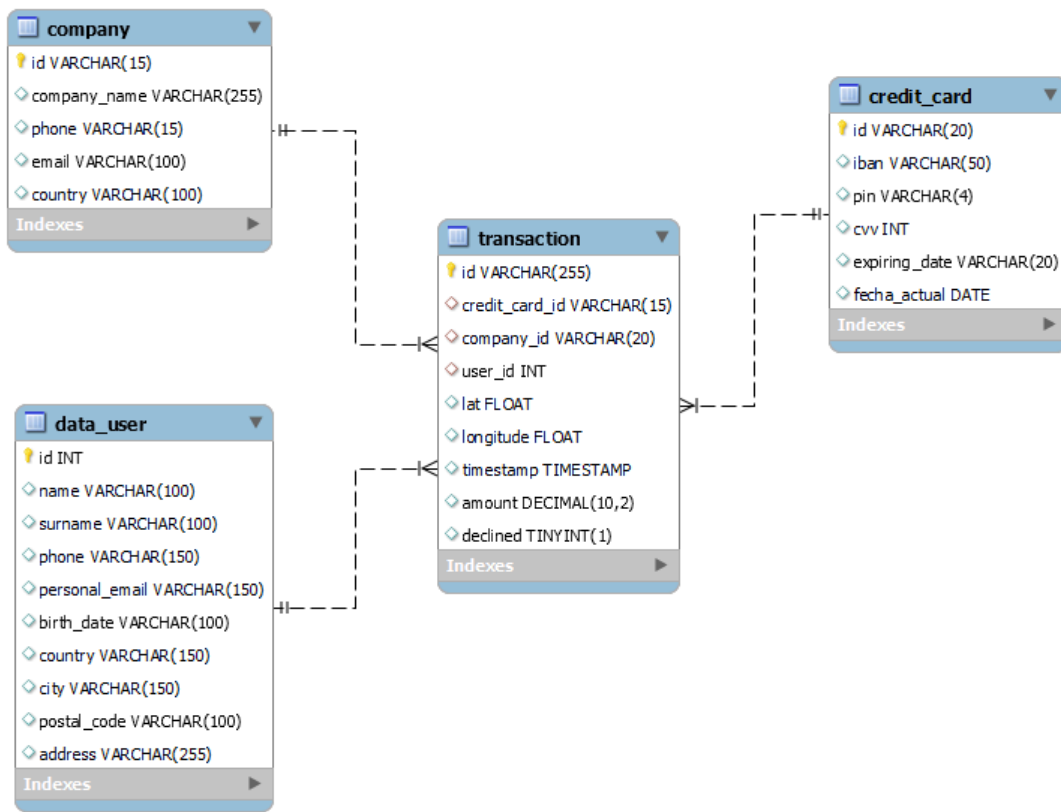
#	Time	Action	Message
1	11:53:45	SELECT * FROM VistaMarketing WHERE country = "Germany" LIMIT 0, 1000	8 row(s) returned



Nivell 3

Exercici 1

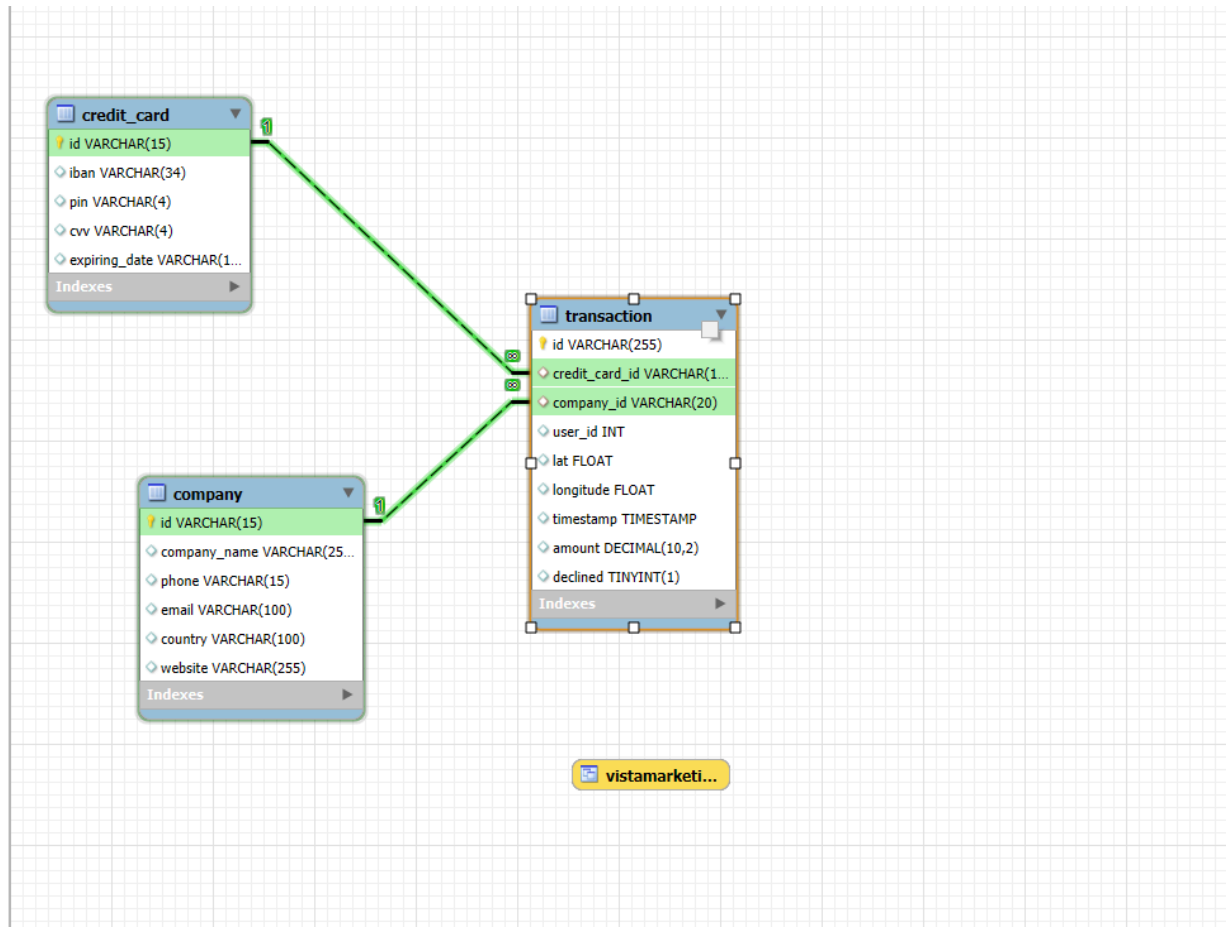
La setmana vinent tindràs una nova reunió amb els gerents de màrqueting. Un company del teu equip va realitzar modificacions en la base de dades, però no recorda com les va realitzar. Et demana que l'ajudis a deixar els comandos executats per a obtenir el següent diagrama:



Recordatori

En aquesta activitat, és necessari que descriguis el "pas a pas" de les tasques realitzades. És important realitzar descripcions senzilles, simples i fàcils de comprendre. Per a realitzar aquesta activitat hauràs de treballar amb els arxius denominats "estructura_dades_user" i "dades_introduir_user"

Como punto de referencia, a continuación se muestra el diagrama de la bbdd antes de los cambios realizados



Los cambios realizados en cada tabla son los siguientes:

- En la tabla company:

Se elimina la columna website

- En la tabla credit_card:

1. `id` VARCHAR(15) → VARCHAR(20)
2. `iban` VARCHAR(34) → VARCHAR(50)
3. `cvv` VARCHAR(4) → INT
4. `expiring_date` VARCHAR(10) → VARCHAR(20)
5. Crear nueva columna `fecha_actual`(DATE)

- Crear tabla user ejecutando los archivos "estructura_dades_user" i "dades_introduir_user":

1. Nombre de la tabla user → data_user
2. `email` VARCHAR(150) → `personal_email` VARCHAR(150)

- En la tabla transaction:

Definir user_id como FK, relacionándola con la PK id de la tabla data_user

```

84 -- Eliminar la columna website de la tabla company
85 • ALTER TABLE company
86 DROP COLUMN website;
87 -- Revisar el cambio
88 • 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	

Result 1 ×

Output

#	Time	Action	Message
✓ 1	22:16:30	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 2	22:17:41	DESCRIBE company	5 row(s) returned

```

90 -- modificar el max.de caracteres permitido a 20 en id de la tabla credit_card
91 • ALTER TABLE credit_card
92 MODIFY COLUMN id VARCHAR(20);
93
94 -- modificar el iban de la tabla credit_card para permitir hasta 50 caracteres
95 • ALTER TABLE credit_card
96 MODIFY COLUMN iban VARCHAR(50);
97
98 -- modificar el tipo de la variable cvv a INT en la tabla credit_card
99 • ALTER TABLE credit_card
100 MODIFY COLUMN cvv INT;
101
102 -- modificar el máximo de caracteres en expiring_date de la tabla credit_card a 20
103 • ALTER TABLE credit_card
104 MODIFY COLUMN expiring_date VARCHAR(20);
105
106 -- crear nueva columna (fecha_actual(DATE)) en la tabla credit_card
107 • ALTER TABLE credit_card
108 ADD COLUMN fecha_actual DATE;

```

Output

#	Time	Action	Message
✓ 1	22:45:11	ALTER TABLE credit_card MODIFY COLUMN id VARCHAR(20)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 2	22:45:11	ALTER TABLE credit_card MODIFY COLUMN iban VARCHAR(50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 3	22:45:11	ALTER TABLE credit_card MODIFY COLUMN cvv INT	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
✓ 4	22:45:11	ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR(20)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 5	22:45:11	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

```

110  -- revisar los cambios en la tabla credit card
111  • DESCRIBE credit_card;

```

Result Grid						
		Filter Rows:	Export:		Wrap Cell Content: I	
	Field	Type	Null	Key	Default	Extra
▶	id	varchar(20)	NO	PRI	NULL	
	iban	varchar(50)	YES		NULL	
	pin	varchar(4)	YES		NULL	
	cvv	int	YES		NULL	
	expiring_date	varchar(20)	YES		NULL	
	fecha_actual	date	YES		NULL	

Result 2 ×

Output

Action Output

#	Time	Action	Message
✓ 1	22:47:55	DESCRIBE credit_card	6 row(s) returned

```

113  -- Crear tabla user ejecutando el archivo "estructura_dades_user"
114
115  • CREATE INDEX idx_user_id ON transaction(user_id);
116
117  • CREATE TABLE IF NOT EXISTS user (
118      id INT PRIMARY KEY,
119      name VARCHAR(100),
120      surname VARCHAR(100),
121      phone VARCHAR(150),
122      email VARCHAR(150),
123      birth_date VARCHAR(100),
124      country VARCHAR(150),
125      city VARCHAR(150),
126      postal_code VARCHAR(100),
127      address VARCHAR(255),
128      FOREIGN KEY(id) REFERENCES transaction(user_id)
129  );

```

Output

Action Output

#	Time	Action	Message
✓ 1	23:00:57	CREATE INDEX idx_user_id ON transaction(user_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 2	23:00:57	CREATE TABLE IF NOT EXISTS user (id INT PRIMARY KEY, name VARCHAR(100), surname V...	0 row(s) affected

```

1  -- Insertar los datos en la tabla user utilizando el archivo datos_introducir_user(1).sql
2
3  • SET foreign_key_checks = 0;
4
5  -- Insertamos datos de user
6  • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
7  • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
8  • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
9  • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
10 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
11 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
12 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
13 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
14 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
15 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
16 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
17 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
18 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
19 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (

```

```

"1", "Zeus", "Gamble", "1-282-581-0551
"2", "Garrett", "Mcconnell", "(718) 25
"3", "Ciaran", "Harrison", "(522) 598-
"4", "Howard", "Stafford", "1-411-740-
"5", "Hayfa", "Pierce", "1-554-541-207
"6", "Joel", "Tyson", "(718) 288-8020"
"7", "Rafael", "Jimenez", "(817) 689-0
"8", "Nissim", "Franks", "(692) 157-34
"9", "Mannix", "Mcclain", "(590) 883-2
"10", "Robert", "Mccarthy", "(324) 746
"11", "Joan", "Beird", "(981) 429-8106
"12", "Benedict", "Wheeler", "1-515-82
"13", "Allegra", "Stanton", "1-927-753
"14", "Sara", "Flynn", "1-311-646-9333

```

```

133  -- cambiar el nombre de la tabla user a data_user y de la columna email a personal_email
134 • ALTER TABLE user
135   RENAME TO data_user,
136   RENAME COLUMN email TO personal_email;

```

Output

Action Output

#	Time	Action	Message
✓ 1	23:44:34	ALTER TABLE user RENAME TO data_user, RENAME COLUMN email TO personal_email	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

```

138  -- revisar los cambios en la tabla data_user
139 • DESCRIBE data_user;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	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 4 ×

Output

Action Output

#	Time	Action
✓ 1	23:50:41	DESCRIBE data_user

```

141 -- crear FK user_id en tabla transaction, conectar con tabla data_user por PK id
142 • ALTER TABLE transaction
143 ADD FOREIGN KEY (user_id) REFERENCES data_user(id);

```

Output

Action Output

#	Time	Action	Message
1	10:20:24	ALTER TABLE transaction ADD FOREIGN KEY (user_id) REFERENCES data_user(id)	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (transactions`.`#sql-16c4_;

```

145 -- para resolver el error 1452 hay que ver si existen registros en user_id de transaction que no existen en id de data_user
146 • SELECT tr.user_id
147 FROM transaction tr
148 LEFT JOIN data_user du
149 on tr.user_id = du.id
150 WHERE du.id IS NULL;

```

Result Grid

user_id
9999

Result 9

Output

Action Output

#	Time	Action	Message
1	10:49:54	SELECT tr.user_id FROM transaction tr LEFT JOIN data_user du on tr.user_id = du.id WHERE du.id IS NULL LI...	1 row(s) returned

```

152 -- Añadir user_id 9999 en data_user para poder crear la FK en transaction
153 • INSERT INTO data_user(id) VALUES ("9999");

```

Output

Action Output

#	Time	Action	Message
1	10:50:57	INSERT INTO data_user(id) VALUES ("9999")	1 row(s) affected

```

155 -- crear la FK user_id en tabla transaction, conectar con tabla data_user por PK id
156 • ALTER TABLE transaction
157 ADD FOREIGN KEY (user_id) REFERENCES data_user(id);
158

```

Output

Action Output

#	Time	Action	Message
1	10:52:41	ALTER TABLE transaction ADD FOREIGN KEY (user_id) REFERENCES data_user(id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

```
159  -- consultando el esquema de data_user aparece que el id es PK y FK, eliminar el FK de la tabla
160  • ALTER TABLE data_user
161  DROP FOREIGN KEY data_user_ibfk_1;
162
163
164
```

Output :

Action Output

#	Time	Action	Message
✓ 1	11:30:32	ALTER TABLE data_user DROP FOREIGN KEY data_user_ibfk_1	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Revisando el diagrama EER con el del enunciado del ejercicio ahora aparecen iguales, tras la ejecución de todos los comandos anteriores

Exercici 2

L'empresa també et sol·licita crear una vista anomenada "InformeTecnico" que contingui la següent informació:

- ID de la transacció
- Nom de l'usuari/ària
- Cognom de l'usuari/ària
- IBAN de la targeta de crèdit usada.
- Nom de la companyia de la transacció realitzada.
- Assegura't d'incloure informació rellevant de totes dues taules i utilitza àlies per a canviar de nom columnes segons sigui necessari.

Mostra els resultats de la vista, ordena els resultats de manera descendent en funció de la variable ID de transaction.


```

163 -- Crear vista "InformeTecnico"
164 • CREATE VIEW InformeTecnico AS
165 SELECT tr.id AS transaction_id, du.name AS user_name, du.surname AS user_surname, cc.iban, c.company_name
166 FROM transaction tr
167 JOIN data_user du
168 ON tr.user_id = du.id
169 JOIN credit_card cc
170 ON tr.credit_card_id = cc.id
171 JOIN company c
172 ON tr.company_id = c.id
173 ORDER BY transaction_id DESC;
174
175 -- mostrar la vista InformeTecnico
176 • SELECT *
177 FROM informetecnico;
178

```

transaction_id	user_name	user_surname	iban	company_name
FE96CE47-BD59-381C-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-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCE2AB9A-271D-2BDC-9E49-8DD92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated

informetecnico 12 x

Output			
Action Output			
#	Time	Action	Message
✓ 1	11:55:11	CREATE VIEW InformeTecnico AS SELECT tr.id AS transaction_id, du.name AS user_name, du.surname AS u...	0 row(s) affected
✓ 2	11:55:11	SELECT * FROM informetecnico LIMIT 0, 1000	587 row(s) returned