

EXERCICIS SPRENT 3

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.

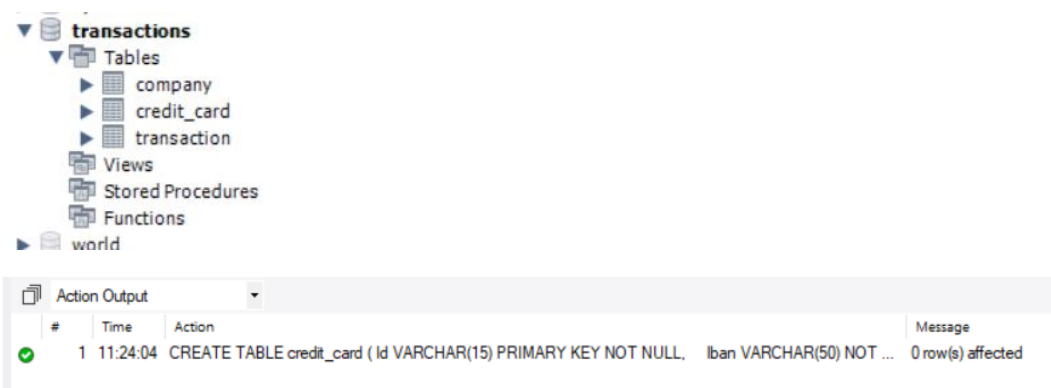
Antes de crear la nueva tabla, miramos datos que tenemos que introducir. Vemos que el formato de la fecha no es el optimo para MySQL, el formato que tiene es MM/DD/YY y en MySQL solo acepta formato YY/MM/DD. Si creamos la columna *Expiring_Date* como DATE nos dará error. La crearemos como texto y luego lo transformaremos.

Vamos a crear la nueva tabla en nuestra base de datos *transactions*. La nueva tabla se llamara *credit_card* y contendrá toda la información de las tarjetas de crédito con las cuales se hacen las transacciones. La tabla tendrá las siguientes columnas:

- Id VARCHAR(15) PRIMARY KEY NOT NULL
- Iban VARCHAR(50) NOT NULL
- Pan VARCHAR(50) NOT NULL
- Pin INT NOT NULL
- Cvv INT NOT NULL
- Expiring_Date VARCHAR(15) NOT NULL

```
4 CREATE TABLE credit_card (  
5     Id VARCHAR(15) PRIMARY KEY NOT NULL,  
6     Iban VARCHAR(50) NOT NULL,  
7     Pan VARCHAR(50) NOT NULL,  
8     PIN INT NOT NULL,  
9     CVV INT NOT NULL,  
10    expiring_date VARCHAR(15) NOT NULL -- Creamos como texto para salvar el formato diferente  
11 );
```

Aquí podemos ver como la nueva table ha sido creada



Introduzco los datos proporcionados en la nueva tabla.

#	Time	Action	Message
547	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4765', 'SA2888713...	1 row(s) affected
548	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4772', 'ME598320...	1 row(s) affected
549	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4779', 'FI9109231...	1 row(s) affected
550	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4786', 'SI5170310...	1 row(s) affected
551	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4793', 'HU952156...	1 row(s) affected
552	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4800', 'SI9782433...	1 row(s) affected
553	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4807', 'LB1929831...	1 row(s) affected
554	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4814', 'MR484528...	1 row(s) affected
555	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4821', 'LT2531475...	1 row(s) affected
556	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4828', 'BG111LMJ3...	1 row(s) affected
557	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4835', 'PT3459217...	1 row(s) affected
558	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4842', 'SA2156708...	1 row(s) affected
559	12:29:41	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4849', 'SE2812123...	1 row(s) affected

3 • `SELECT * FROM credit_card;`

Result Grid						
Filter Rows:						
Edit: Export/Import: Wr						
	Id	Iban	Pan	PIN	CVV	expiring_date
▶	CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22
	CcU-2945	DO26854763748537475216568689	5142423821948828	9080	887	08/24/23
	CcU-2952	BG45IVQL52710525608255	4556 453 55 5287	4598	438	06/29/21
	CcU-2959	CR7242477244335841535	372461377349375	3583	667	02/24/23
	CcU-2966	BG72LKTQ15627628377363	448566 886747 7265	4900	130	10/29/24
	CcU-2973	PT87806228135092429456346	544 58654 54343 384	8760	887	01/30/25
	CcU-2980	DE39241881883086277136	402400 7145845969	5075	596	07/24/22
	CcU-2987	GE89681434837748781813	3763 747687 76666	2298	797	10/31/23
	CcU-2994	BH62714428368066765294	344283273252593	7545	595	02/28/22
	CcU-3001	CY49087426654774581266832110	511722 924833 2244	9562	867	09/16/22
	CcU-3008	LU507216693616119230	4485744464433884	1856	740	04/05/25
	CcU-3015	PS119398216295715968342456821	3784 662233 17389	3246	822	01/31/22

561 12:31:28 `SELECT * FROM credit_card LIMIT 0, 1000`

275 row(s) returned

Ahora vamos a transformar los datos de expiring_date a formato YY/MM/DD.

```
-- Transformamos los datos de expiring_date a yy/mm/dd
-- Para hacer esto tengo que desconectar el modo seguro
```

- `SET SQL_SAFE_UPDATES = 0;`
- `UPDATE credit_card`
`SET expiring_date = STR_TO_DATE(expiring_date, '%m/%d/%y');`
- `SET SQL_SAFE_UPDATES = 1; -- Vuelvo a conectar el modo seguro`

563	12:38:16	SET SQL_SAFE_UPDATES = 0	0 row(s) affected
564	12:38:20	UPDATE credit_card SET expiring_date = STR_TO_DATE(expiring_date, '%m/%d/%y')	275 row(s) affected Rows matched: 275 Changed: 275 Warnings: 0
565	12:38:27	SET SQL_SAFE_UPDATES = 1	0 row(s) affected

2

3 • `SELECT * FROM credit_card;`

	Id	Iban	Pan	PIN	CVV	expiring_date
▶	CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	2022-10-30
	CcU-2945	DO26854763748537475216568689	5142423821948828	9080	887	2023-08-24
	CcU-2952	BG45IVQL52710525608255	4556 453 55 5287	4598	438	2021-06-29
	CcU-2959	CR7242477244335841535	372461377349375	3583	667	2023-02-24
	CcU-2966	BG72LKTQ15627628377363	448566 886747 7265	4900	130	2024-10-29
	CcU-2973	PT87806228135092429456346	544 58654 54343 384	8760	887	2025-01-30
	CcU-2980	DE39241881883086277136	402400 7145845969	5075	596	2022-07-24
	CcU-2987	GE80681434837748781813	3763 747687 76666	7098	767	2023-10-31

Ahora vamos a cambiar la columna *expiring_date* de tipo VARCHAR a tipo DATE

```
-- Vamos a pasar columna expiring_date de varchar a date
• ALTER TABLE credit_card
  MODIFY COLUMN expiring_date DATE;
```

567 12:46:40 ALTER TABLE credit_card MODIFY COLUMN expiring_date DATE 275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
CVV	int		NO			select,insert,update,references
expiring_date	date		YES			select,insert,update,references
Iban	varchar(50)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
Id	varchar(15)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
Pan	varchar(50)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
PIN	int		NO			select,insert,update,references

Al observar los datos también vemos que en la columna *Pan* hay algunos registros que tienen espacios. Para evitarnos problemas en un futuro vamos a eliminar esos espacios. Para poder hacerlo he tenido que desconectar el modo seguro. Lo he vuelto a conectar después.

```
25 -- Vamos a quitar los espacios de algunos registros de la columna Pan
26 -- Para hacer esto tengo que desconectar el modo seguro
27 • SET SQL_SAFE_UPDATES = 0;
28
29 • UPDATE credit_card SET Pan = REPLACE(Pan, ' ', '') WHERE Pan LIKE '% %';
30
31 • SET SQL_SAFE_UPDATES = 1; -- Vuelvo a conectar el modo seguro
```

569 12:56:44 SET SQL_SAFE_UPDATES = 0 0 row(s) affected

570 12:56:48 UPDATE credit_card SET Pan = REPLACE(Pan, ' ', '') WHERE Pan LIKE '% %' 187 row(s) affected Rows matched: 187 Changed: 187 Warnings: 0

571 12:57:05 SELECT * FROM credit_card LIMIT 0, 1000 275 row(s) returned

572 12:57:47 SET SQL_SAFE_UPDATES = 1 0 row(s) affected

3 • `SELECT * FROM credit_card;`

	Id	Iban	Pan	PIN	CVV	expiring_date
	CcU-3715	IS389474813577187701272239	4532634735833	7294	780	2024-06-14
	CcU-3722	SI50942306532997569	4716688698622567	3785	295	2024-08-10
	CcU-3729	MD5723087436783068347555	4024007128955884	8571	350	2025-03-02
	CcU-3736	LT273037680127756677	5218432888538272	9593	777	2023-02-06
	CcU-3743	BA542358041365401657	345537652543545	9314	656	2022-06-11
	CcU-3750	CY80138698304356192781019706	5175965455218886	5633	429	2022-11-29
	CcU-3757	HU63856737363800384694160203	4916668572485677	8557	789	2022-08-20
	CcU-3764	DO83239184883525656124257488	5352384759612247	6546	194	2022-01-18
	CcU-3771	CH6157615368246495325	4935132859765811	6253	637	2021-10-13
	CcU-3778	SE7813107767427189556484	343488771836851	9170	487	2024-04-17
	CcU-3785	ME38421331246858744050	5112727325633686	1755	987	2025-08-13

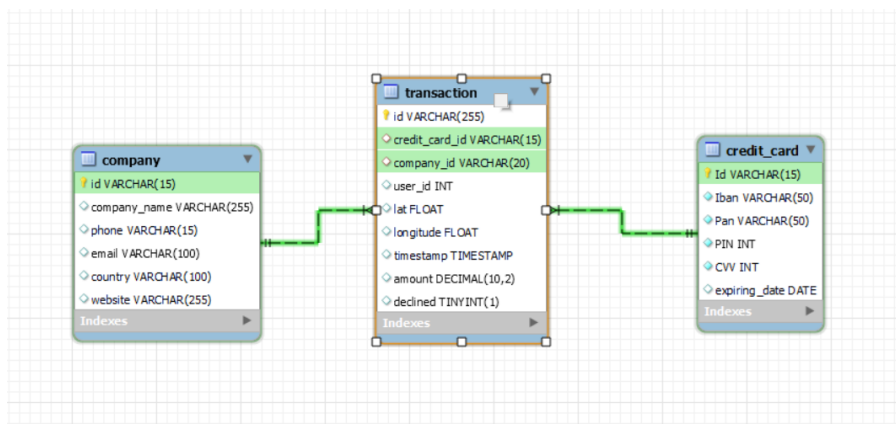
Ahora vamos a crear la relación entre las tablas *transaction* y *credit_card*. Vemos que la columna *Id* de *credit_card* es son los mismos datos que la columna *credit_card_id* de *transaction*. La *Id* de *credit_card* es la Primary Key de esta tabla. La *credit_card_id* de *transaction* es la Foreign Key de esta tabla que la une con la tabla *credit_card*. Asi lo voy a hacer.

```

33  -- vamos ha hacer la relacion entre transaction y credit_card
34  • ALTER TABLE transaction
35  ADD CONSTRAINT fk_credit_card_id
36  FOREIGN KEY (credit_card_id) REFERENCES credit_card(Id);
37

```

En el siguiente diagrama vemos como esta relacionada transaction y credit_card. Ya tenemos una base de datos con tres tablas relacionadas entre si con sus Primary Key y sus Foreign Key.



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.

Vamos a ver que numero de cuenta tiene ahora el cliente:

```

49  -- El usuario tiene ahora este numero de cuenta.
50  • SELECT Iban
51  FROM credit_card
52  WHERE id = 'Ccu-2938';
53

```

Result Grid

Iban
TR301950312213576817638661

Vamos a cambiarle el valor:

```

-- Vamos a cambiarle el valor
• UPDATE credit_card
SET Iban = 'R323456312213576817699999'
WHERE id = 'Ccu-2938';

```

3 13:54:03 UPDATE credit_card SET Iban = 'R323456312213576817699999' WHERE id = 'Ccu-2938' 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

Vamos a comprobar el nuevo numero de cuenta del cliente:

```
59 -- El usuario tiene ahora este numero de cuenta.
60 • SELECT Iban
61 FROM credit_card
62 WHERE id = 'Ccu-2938';
63
```

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

Iban
R323456312213576817699999

EXERCICI 3 (mal- modificado a continuació)

En la taula "transaction" ingresa 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

Para poder introducir este registro tengo que desactivar las restricciones de claves foráneas ya que ese id de compañía no esta en la tabla *company*. Inmediatamente después la vuelvo a activar.

```
74 -- Para insertarlo tengo que desactivar las restricciones de claves foraneas
75 • SET FOREIGN_KEY_CHECKS = 0;
76
77 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined)
78 VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, CURRENT_TIMESTAMP(), 111.11, 0);
79
80 • SET FOREIGN_KEY_CHECKS = 1; -- Vuelvo a activar las restricciones
```

✓	1	10:22:45	SET FOREIGN_KEY_CHECKS = 0	0 row(s) affected
✓	2	10:22:50	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined)	1 row(s) affected
✓	3	10:23:28	SET FOREIGN_KEY_CHECKS = 1	0 row(s) affected

Vemos el nuevo registro introducido.

1	SELECT *
2	FROM transaction
3	WHERE id='108B1D1D-5B23-A76C-55EF-C568E49A99DD';

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	2025-01-21 10:22:50	111.11	0

MODIFICACIO 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

Para que me coja la nueva información, para que no me fallen las Foreign Keys tengo que crear las *id* de *credit_card_id* en la tabla *credit_card*, *company_id* en la tabla *company* y *user_id* de la tabla *data_user*:

222	•	INSERT INTO credit_card (id, Iban, PIN, CVV, expiring_date, fecha_actual)
223		VALUES ('CcU-9999', '9999999999999999', '9999', '999', '2099-12-12', CURRENT_DATE);
224		

✓	8	20:33:41	INSERT INTO credit_card (id, Iban, PIN, CVV, expiring_date, fecha_actual) VALUES ('CcU-9999', '999...	1 row(s) affected
---	---	----------	---	-------------------


```

231 • INSERT INTO company (id, company_name, phone, email, country)
232 VALUES ('b-9999', 'Nueve', '99999999', '9999@99.com', 'Noveno');
233

```

10 20:39:43 INSERT INTO company (id, company_name, phone, email, country) VALUES (b-9999, 'Nueve', '99999... 1 row(s) affected

Tengo que insertar también este id en la tabla `data_user` ya que en la modificación esta tabla ya la tengo creada y si no lo hiciera me daría error otra vez. Si lo hubiera hecho correcto en un inicio, como la tabla `data_user` no estaba creada no hubiera hecho falta hacerlo.

```

239 -- vamos a crear el mismo user_id en la tabla data_user
240 • INSERT INTO data_user (id)
241 VALUES ('9999');

```

15 20:52:41 INSERT INTO data_user (id) VALUES ('9999') 1 row(s) affected

Comprobamos la tabla `credit_cards`

```

226 • SELECT *
227 FROM credit_card
228 WHERE id='CcU-9999';
229

```

Result Grid						
Filter Rows:						
	Id	Iban	PIN	CVV	expiring_date	fecha_actual
▶	CcU-9999	9999999999999999	9999	999	2099-12-12	2025-01-26
	NULL	NULL	NULL	NULL	NULL	NULL

Comprobamos la tabla `company`

```

235 • SELECT *
236 FROM company
237 WHERE id='b-9999';
238

```

Result Grid					
Filter Rows:					
	id	company_name	phone	email	country
▶	b-9999	Nueve	99999999	9999@99.com	Noveno
*	NULL	NULL	NULL	NULL	NULL

Comprobamos la tabla `data_user`

```

244 • SELECT *
245 FROM data_user
246 WHERE id='9999';
247

```

id	name	surname	phone	personal_email	birth_date	country	city	postal_code	address
9999	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Ahora ya podemos ingresar el nuevo registro en la tabla *transaction*:

```

INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined)
VALUES ('10881D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, CURRENT_TIMESTAMP(), 111.11, 0);

```

17 20:56:14 INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, d... 1 row(s) affected

Comprobamos el nuevo registro de *transaction*:

```

253 • SELECT *
254 FROM transaction
255 WHERE id='10881D1D-5B23-A76C-55EF-C568E49A99DD';
256

```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
10881D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	2025-01-26 20:56:14	111.11	0

EXERCICI 4

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

Aquí vemos como es la tabla *credit_card*:

```

91 • SELECT * FROM credit_card;

```

Id	Iban	Pan	PIN	CVV	expiring_date
CcU-2938	R323456312213576817699999	5424465566813633	5424465566813633	887	2022-10-30
CcU-2945	DO26854763748537475216568689	5142423821948828	5080	887	2023-08-24
CcU-2952	BG45IVQL52710525608255	4556453555287	4598	438	2021-06-29
CcU-2959	CR7242477244335841535	372461377349375	3583	667	2023-02-24
CcU-2966	BG72LKTQ15627628377363	4485668867477265	4900	130	2024-10-29
CcU-2973	PT87806228135092429456346	5445865454343384	8760	887	2025-01-30
CcU-2980	DE39241881883086277136	4024007145845969	5075	596	2022-07-24
CcU-2987	GE89681434837748781813	376374768776666	2298	797	2023-10-31
CcU-2994	BH62714428368066765294	344283273252593	7545	595	2022-02-28

Vamos a eliminar la columna *Pan*.

```

3 -- Vamos a eliminar la columna PAN
4 • ALTER TABLE credit_card DROP COLUMN Pan;

```


9 10:55:46 ALTER TABLE credit_card DROP COLUMN Pan 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Vamos a ver como ha quedado la tabla *credit_card*:

```
97 • SELECT * FROM credit_card;
```

	Id	Iban	PIN	CVV	expiring_date
▶	CdJ-2938	R323456312213576817699999	3257	984	2022-10-30
	CdJ-2945	DO26854763748537475216568689	9080	887	2023-08-24
	CdJ-2952	BG45IVQL52710525608255	4598	438	2021-06-29
	CdJ-2959	CR7242477244335841535	3583	667	2023-02-24
	CdJ-2966	BG72LKTQ15627628377363	4900	130	2024-10-29
	CdJ-2973	PT87806228135092429456346	8760	887	2025-01-30
	CdJ-2980	DE39241881883086277136	5075	596	2022-07-24
	CdJ-2987	GE89681434837748781813	2298	797	2023-10-31
	CdJ-2994	BH62714428368066765294	7545	595	2022-02-28

NIVELL 2

EXERCICI 1

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

Vamos a ver ese registro:

```
103 • SELECT *
104 FROM transaction
105 WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

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

Vamos a eliminar el registro:

```
107 -- Vamos a eliminar el registro
108 • DELETE FROM transaction
109 WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

12 11:30:26 DELETE FROM transaction WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B02' 1 row(s) affected

Vamos a ver si aun existe ese registro

```
112 • SELECT *
113 FROM transaction
114 WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

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

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.

Vamos a crear la vista:

```
1 • CREATE VIEW VistaMarketing AS
2 SELECT c.company_name AS Nom_Companyia, c.phone AS Telefon_Contacte,
3        c.country AS Pais_Residencia, ROUND(AVG(t.amount), 2) AS Mitjana_Compra
4 FROM company c
5 JOIN transaction t ON c.id = t.company_id
6 GROUP BY c.company_name, c.phone, c.country
7 ORDER BY Mitjana_Compra DESC;
```

1 10:44:37 CREATE VIEW VistaMarketing AS SELECT c.company_name AS Nom_Companyia, c.phone AS Telefon_... 0 row(s) affected

Vamos a ver la vista creada:

```
133 • SELECT *
134 FROM vistamarketing;
135
```

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

1 12:02:32 SELECT * FROM vistamarketing LIMIT 0, 1000 100 row(s) returned

Views

- vistamarketing
- Stored Procedures
- Functions

EXERCICI 3

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

Tengo que tener en cuenta que en la vista *country* lo he llamado *Pais_Residencia*.

```

139 • SELECT *
140 FROM vistamarketing
141 WHERE Pais_Residencia='Germany';
142

```

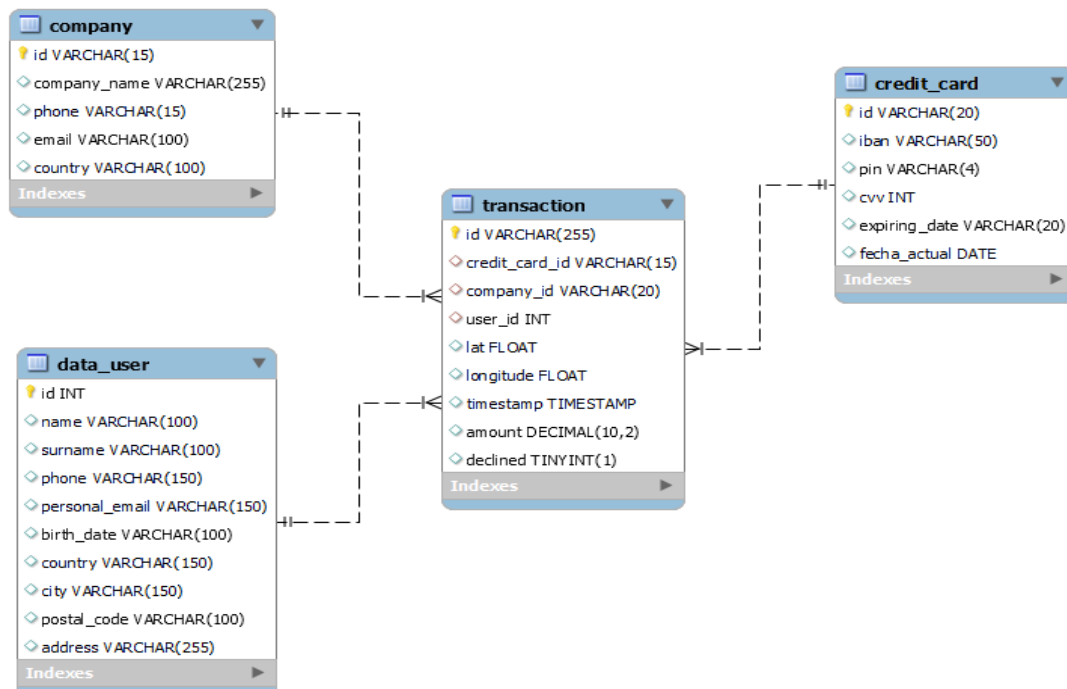
	Nom_Companyia	Telefon_Contacte	Pais_Residencia	Mitjana_Compra
▶	Aliquam PC	01 45 73 52 16	Germany	385.27
	Ac Industries	09 34 65 40 60	Germany	289.65
	Rutrum Non Inc.	02 66 31 61 09	Germany	266.90
	Nunc Interdum Incorporated	05 18 15 48 13	Germany	244.03
	Augue Foundation	06 88 43 15 63	Germany	240.80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

3 12:08:17 SELECT * FROM vistamarketing WHERE Pais_Residencia='Germany' LIMIT 0, 1000 8 row(s) returned

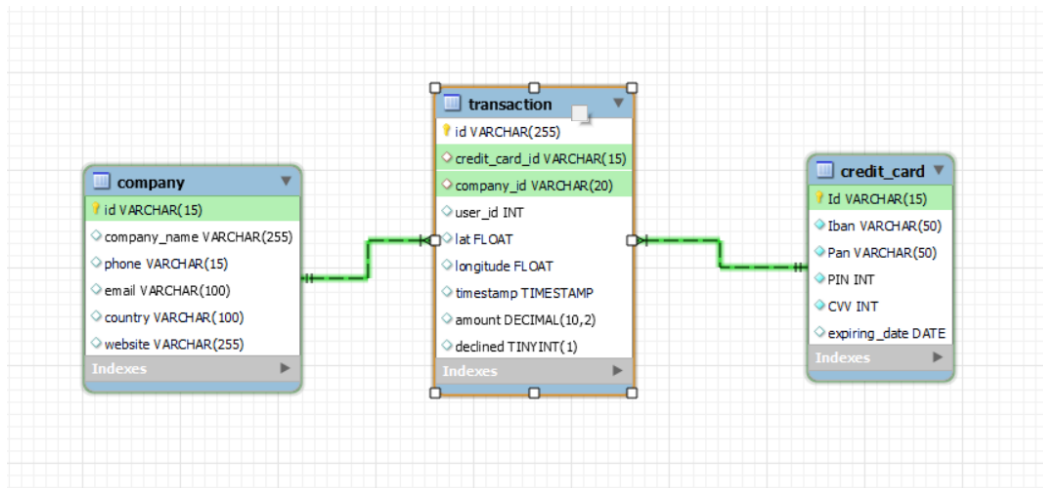
NIVELL 3

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 diagrama de mostra



Actualmente tenemos el siguiente esquema:



Vemos varias diferencia entre los dos esquemas.

Tabla company:

Se ha eliminado la columna *website*. Vamos a hacerlo:

```

150 • ALTER TABLE company DROP COLUMN website;
151

```

Output

#	Time	Action	Message
1	12:37:29	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Tables

- company
 - Columns
 - id
 - company_name
 - phone
 - email
 - country
 - Indexes

Tabla transaction:

No hay que hacer nada. Están iguales.

Tabla credit_card:

Tengo que cambiar varias cosas:

La columna *id* tengo que pasarla de VARCHAR(15) a VARCHAR(20)

```

154 • ALTER TABLE credit_card
155   MODIFY COLUMN Id VARCHAR(20);
156

```

1 12:49:04 ALTER TABLE credit_card MODIFY COLUMN Id VARCHAR(20) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

La columna *pin* la pasamos de INT a VARCHAR(4)

```

158 • ALTER TABLE credit_card
159     MODIFY COLUMN PIN VARCHAR(4);
160

```

1 12:53:54 ALTER TABLE credit_card MODIFY COLUMN PIN VARCHAR(4)

275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

La columna expiring_date de DATE a VARCHAR(20)

```

162 • ALTER TABLE credit_card
163     MODIFY COLUMN expiring_date VARCHAR(20);
164

```

1 12:57:49 ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR(20)

275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

Creamos una nueva columna llamada fecha_actual del tipo DATE

```

166 • ALTER TABLE credit_card
167     ADD COLUMN fecha_actual DATE;
168

```

1 13:02:30 ALTER TABLE credit_card ADD COLUMN fecha_actual DATE

0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

comprobación



Ya hemos modificado las tablas que teníamos ya creadas. Ahora tendríamos que crear una nueva tabla llamada *data_user*. Nos proporcionan los códigos y los datos para crear esta nueva tabla, y los ejecutamos.

Creamos la tabla

```

1 13:17:38 CREATE INDEX idx_user_id ON transaction(user_id)
2 13:17:38 CREATE TABLE IF NOT EXISTS user ( id INT PRIMARY KEY, name VARCHAR(100), sum...

```

0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

0 row(s) affected

Introducimos los datos

```

3 13:20:35 SET foreign_key_checks = 0
4 13:20:35 INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VA...
5 13:20:35 INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VA...
6 13:20:35 INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VA...
7 13:20:35 INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VA...
8 13:20:35 INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VA...
9 13:20:35 INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VA...
10 13:20:35 INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VA...
11 13:20:35 INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VA...
12 13:20:35 INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VA...
13 13:20:35 INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VA...
14 13:20:35 INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VA...
15 13:20:35 INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VA...

```

0 row(s) affected

1 row(s) affected

1 row(s) affected

1 row(s) affected

1 row(s) affected

1 row(s) affected

1 row(s) affected

1 row(s) affected

1 row(s) affected

1 row(s) affected

1 row(s) affected

1 row(s) affected

1 row(s) affected

Comprobamos la nueva tabla

```
170 • SELECT *
171 FROM user;
172
```

	id	name	surname	phone	email	birth_date	country	city	postal_code	address
▶	1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.edu	Nov 17, 1985	United States	Lowell	73544	348-7818 Sagittis St.
	2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Des Moines	59464	903 Sit Ave
	3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Columbus	56518	736-2063 Tellus St.
	4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.edu	Feb 18, 1989	United States	Kailua	77417	Ap #545-2244 Erat. Rd.
	5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	Sep 26, 1998	United States	Sandy	31564	341-2821 Ultrices Av.
	6	Joel	Tyson	(718) 288-8020	gravida.nunc.sed@yahoo.ca	Oct 15, 1989	United States	Nashville	96838	888-2799 Amet Street
	7	Rafael	Jimenez	(817) 689-0478	egest@outlook.ca	Dec 4, 1981	United States	Hillsboro	29874	8627 Malesuada Rd.
	8	Nissim	Franks	(692) 157-3469	egestas.aliquam.fringilla@google.ca	Aug 1, 1993	United States	Jackson	61750	Ap #251-7144 Integer St.

✓ 1 13:24:31 SELECT * FROM user LIMIT 0, 1000

275 row(s) returned

Pero hay dos nombres diferentes.

La tabla se llama *user* y la tendría que llamar *data_user*

```
174 • RENAME TABLE user TO data_user;
175
```

Output

Action Output

#	Time	Action
✓ 1	13:31:58	RENAME TABLE user TO data_user

▼ transactions

▼ Tables

- company
- credit_card
- data_user
- transaction

▼ Views

La nueva tabla tiene una columna llamada *email* y se tendría que llamar *personal_email*

```
177 • ALTER TABLE data_user
178 CHANGE COLUMN email personal_email VARCHAR(150);
```

Output

Action Output

#	Time	Action
✓ 1	13:38:40	ALTER TABLE data_user CHANGE COLUMN email personal_email VARCHAR(150)



Vamos a comprobar que el esquema haya quedado igual

Al comprobarlo hemos visto que se ha producido un error al definir la foreign key de la tabla *transaction* que nos relaciona con la primary key, que es la *id*, de la tabla *data_user*.

Erroneo- Modificado a continuacion

Rectificamos este hecho. Para hacerlo debemos desconectar la seguridad de Foreign Key. Inmediatamente después lo volvemos a conectar.

```
180 -- Modificamos foreign key de la columna user_id de la tabla transaction
181 -- Desconectamos seguridad para Foreign Key
182 • SET FOREIGN_KEY_CHECKS = 0;
183 -- Anulamos foreign key crado
184 • ALTER TABLE data_user
185 DROP FOREIGN KEY data_user_ibfk_1;
186 -- Creamos nuevo foreign key
187 • ALTER TABLE transaction
188 ADD CONSTRAINT fk_user_id FOREIGN KEY (user_id) REFERENCES data_user(id);
189
190 • SET FOREIGN_KEY_CHECKS = 1; -- Volvemos a activar la seguridad
```

MODIFICACION

Al arrastrar el error del ejercicio nivel 1 ejercicio 3 me daba error al ejecutar el cambio de foreign key. No se debe utilizar el desconectar las foreign key. Si no hubiera tenido ese error no tendría que hacer estos cambios.

El ejercicio tendría que haber sido como lo siguiente:

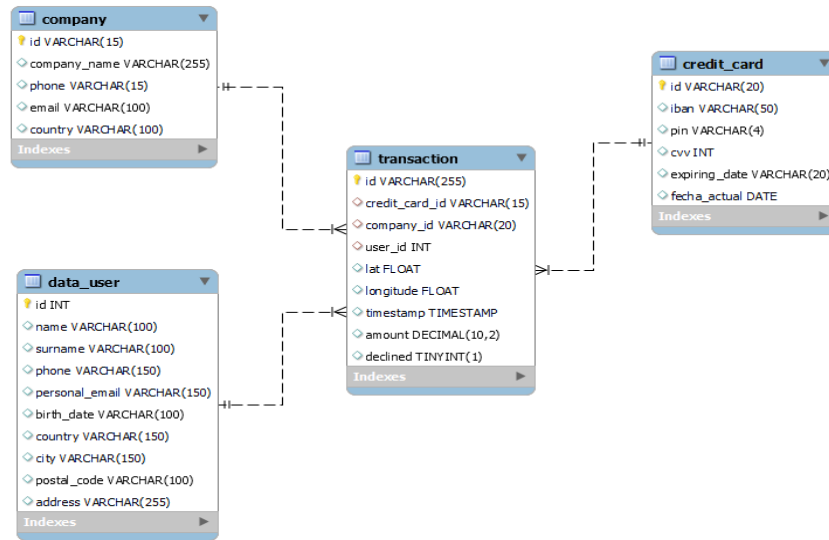
```
259 -- primero tenemos que saber que nombre tiene nuestra foreign key
260 • SHOW CREATE TABLE data_user;
261
262 -- Anulamos foreign key creado
263 • ALTER TABLE data_user
264 DROP FOREIGN KEY data_user_ibfk_1;
265
266 -- Creamos nuevo foreign key
267 • ALTER TABLE transaction
268 ADD CONSTRAINT fk_user_id FOREIGN KEY (user_id) REFERENCES data_user(id);
269
```

NOTA

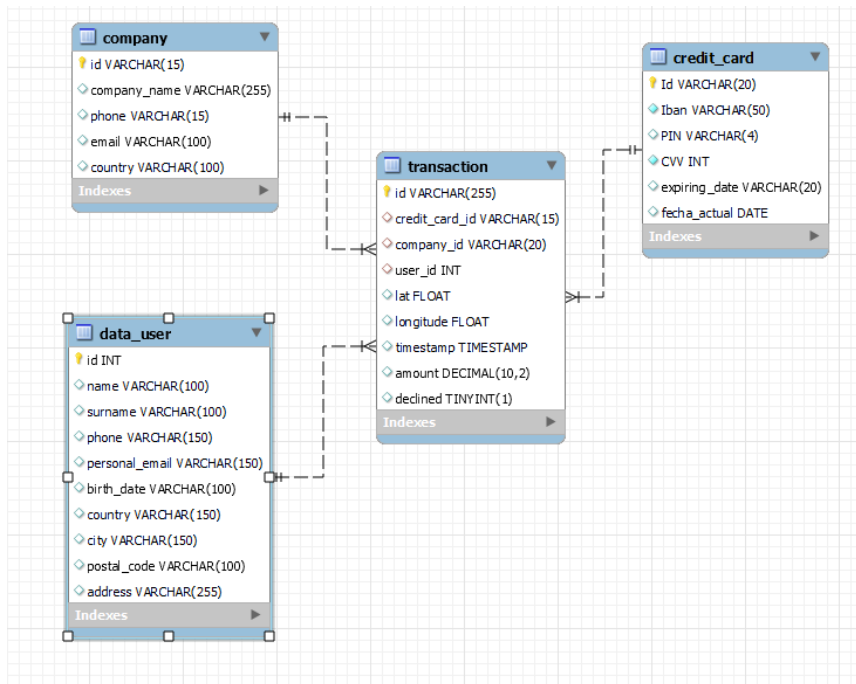
No puedo hacer las comprobaciones ya que a esta altura del ejercicio lo tengo tan avanzado que volver atrás es complicado.

Ahora compararemos los dos esquemas para ver si los resultados son correctos

Esquema propuesto:



Esquema realizado:



Los dos esquemas son iguales. Todo ok.

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.

Creemos la vista:

```
CREATE VIEW InformeTecnico AS
SELECT t.id AS IdTransaccion, d.name AS NomClient, d.surname AS CognomClient,
       cc.iban AS IbanTarjeta, c.company_name AS NomEmpresa
FROM transaction t
JOIN data_user d ON t.user_id = d.id
JOIN credit_card cc ON t.credit_card_id = cc.id
JOIN company c ON t.company_id = c.id
ORDER BY t.id DESC;
```

1 18:02:35 CREATE VIEW InformeTecnico AS SELECT t.id AS IdTransaccion, d.name AS NomClient, d.surname AS ... 0 row(s) affected

Vamos a comprobar su existencia:

212 • SELECT *

213 FROM informetecnico;

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

	IdTransaccion	NomClient	CognomClient	IbanTarjeta	NomEmpresa
▶	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
	FBD7E0D6-8A6B-F5BC-0CA9-EA4B8760100C	Hedwig	Gilbert	MU4132333444534342541344788855	Mauris Id Inc.
	FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF58868200575771634583813	Arcu LLP
	FAAD3FFC-1A17-E141-43D3-359A5BA7CB3B	Hedwig	Gilbert	GE90157928843338134463	Lorem Eu Incorporated
	FA053936-75D8-85FA-490D-9B624E1B920A	Hedwig	Gilbert	GT02497653655330848247645975	Non Justo Corp.

▼ **transactions**

- ▼ **Tables**
 - company
 - credit_card
 - data_user
 - transaction
- ▼ **Views**
 - informetecnico
 - IdTransaccion
 - NomClient
 - CognomClient
 - IbanTarjeta
 - NomEmpresa
 - vistamarketing
- ▼ **Stored Procedures**
- ▼ **Functions**