

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.

```

• USE transactions;
• DROP TABLE IF EXISTS credit_card;
• CREATE TABLE IF NOT EXISTS credit_card (
  id VARCHAR(10) PRIMARY KEY, -- Identificador únic de la targeta
  iban VARCHAR(34) NOT NULL, -- El IBAN pot tenir fins 34 caracters segons standard
  pan VARCHAR(20) NOT NULL, -- PAN (Primary Account Number, 16 dígitos)
  pin CHAR(4) NOT NULL, -- PIN (4 dígitos)
  cvv CHAR(3) NOT NULL, -- CVV (3 dígitos)
  expiring_date VARCHAR(10) NOT NULL -- Data de caducitat
);

```

Output

#	Time	Action	Message
1	12:53:14	CREATE TABLE IF NOT EXISTS credit_card (id VARCHAR(10) PRIMARY KEY, -- Identificador únic de la tar...	0 row(s) affected

```

/* No puc carregar els registres de "datos_introducir_sprint3_credit". Si la columna és VARCHAR(10) hauria d'acceptar la cadena tal qual;
com no ho fa reviso els errors/warnings o si la transacció va ser revertida. En un altre full SQL comprobo:
SELECT @@sql_mode;
i carrego temporalment:
SET @@sql_mode = 'NO_ENGINE_SUBSTITUTION'
per poder carregar els registres de les targetes de crèdit*/

```

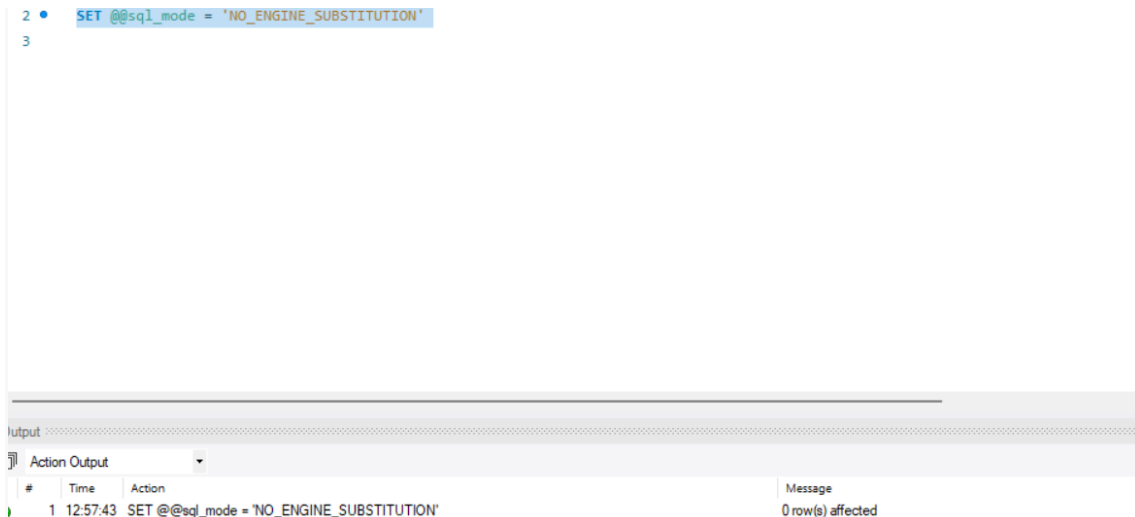
```

1 • SELECT @@sql_mode;
2
3

```

Result Grid

@@sql_mode
ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION



SELECT @@sql_mode;

Serveix per **veure quines regles i validacions té actives el servidor MySQL** en aquell moment.

Aquestes regles (el *sql_mode*) determinen, per exemple:

- com es tracten les dates invàlides (0000-00-00),
- si es permet inserir dades truncades,
- si s'aplica compatibilitat amb l'estàndard SQL,
- si el servidor treballa en mode estricte (STRICT_TRANS_TABLES, STRICT_ALL_TABLES).

SET @@sql_mode = 'NO_ENGINE_SUBSTITUTION';

Canvia el *sql_mode* de la **sessió actual**.

Amb aquesta configuració:

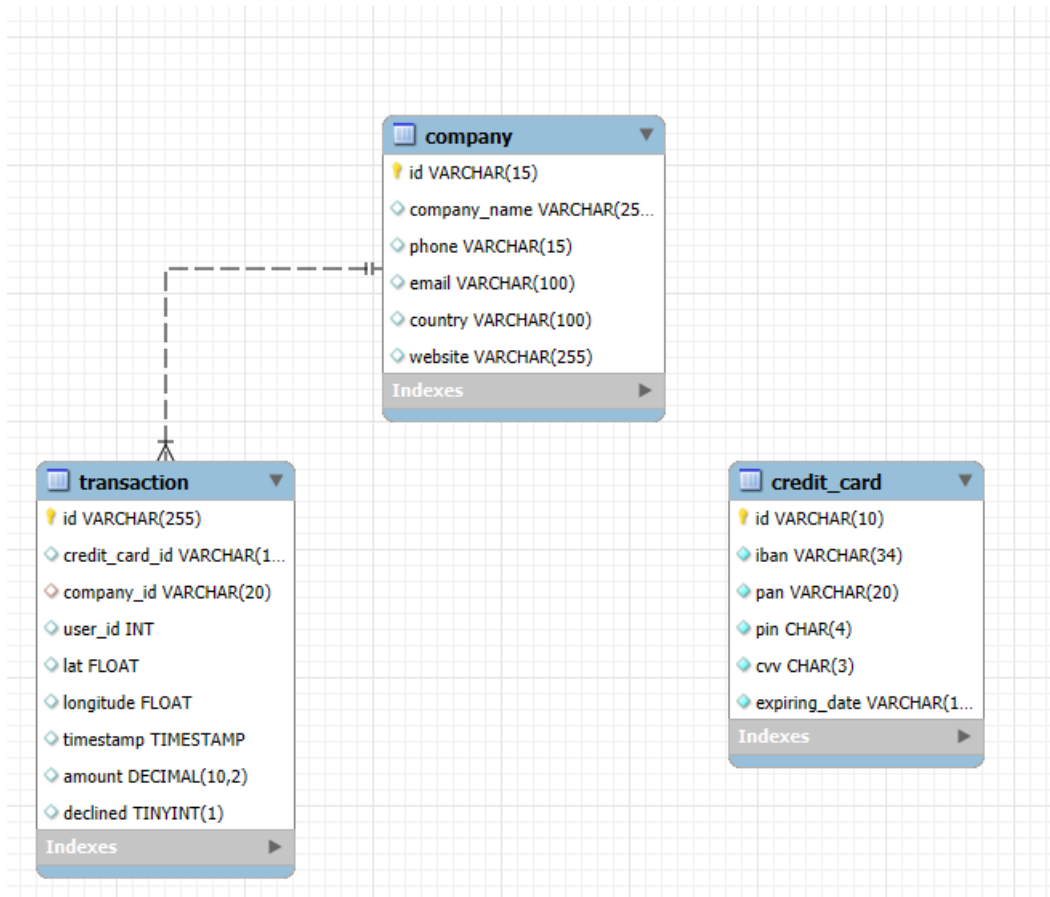
- Només es manté la regla **NO_ENGINE_SUBSTITUTION** (no substituir motors de taula inexistents).
- Es desactiven altres regles, incloses les de mode estricte.

Això implica que MySQL permetrà:

- inserir **dates invàlides** (les guardarà com 0000-00-00),
- **truncar textos** sense avisar,
- acceptar valors que normalment generarien error.

Treure STRICT_TRANS_TABLES pot provocar que es **desin dades incorrectes sense adonar-nos**. Per això s'utilitza **@@sql_mode = 'NO_ENGINE_SUBSTITUTION'** per carregar dades perquè es temporal fins que .

Ara ja podem carregar “datos_introducir_sprint3_credit” pero no tenim creades les relacions entre la nova taula “credit_card” i les altres dues tal com mostra el diagrama següent:



Nota: Abans de crear la **clau forànea**, comprovo que no hi ha valors “orfes”, que tots els valors de transaction.credit_card_id tenen correspondència en credit_card.id. Si hi haguessin s’haurien de corregir.

```

7 • SELECT t.credit_card_id
8 FROM transaction t
9 LEFT JOIN credit_card c
10 ON t.credit_card_id = c.id
11 WHERE c.id IS NULL AND t.credit_card_id IS NOT NULL;
12

```

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

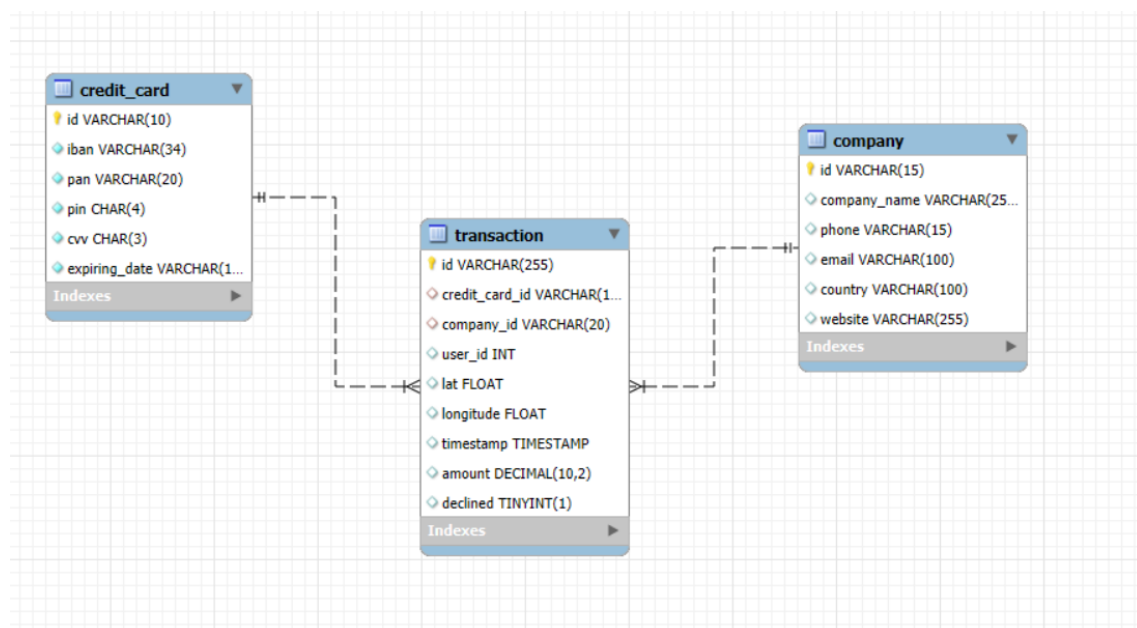
#	Time	Action	Message
1	17:46:32	SELECT t.credit_card_id FROM transaction t LEFT JOIN credit_card c ON t.credit_card_id = c.id	0 row(s) returned

Creem les relacions entre la clau primària de la taula “credit_card” (id) i la clau forànea de transaction (credit_card_id)

```
ALTER TABLE transaction
ADD CONSTRAINT fk_transaction_credit_card_id
FOREIGN KEY (credit_card_id)
REFERENCES credit_card(id);
```

Action Output		
Time	Action	Message
1 19:44:32	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_credit_card_id FOREIGN KEY (credit_card_id) ...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0

I ara ja tenim el model relacional següent amb les tres taules:



Exercici 2

El departament de Recursos Humans ha identificat un error en el número de compte associat a la targeta de crèdit amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és: TR3234563122135768176999999. Recorda mostrar que el canvi es va realitzar.

Identifiquem la informació de la targeta CcU-2938 en la taula credit_card.

```

1 • SELECT *
2 FROM credit_card
3 WHERE id ="CcU-2938";
4

```

Result Grid

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

credit_card 4 x

Output

Action Output

#	Time	Action
1	20:19:05	SELECT * FROM credit_card WHERE id ="CcU-2938"

Actualitzem les dades (UPDATE/SET)

```

39 • UPDATE credit_card
40 SET iban = 'TR323456312213576817699999'
41 WHERE id = 'CcU-2938';-- actualitzem les dades
42

```

Input

Action Output

#	Time	Action	Message
1	20:24:06	UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE id = 'CcU-2938'	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0

Mostrem que s'ha fet el canvi

```

43 • SELECT *
44 FROM credit_card
45 WHERE id ="CcU-2938";-- mostrem que s'ha fet el canvi

```

Result Grid

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR323456312213576817699999	5424465566813633	3257	984	10/30/22
NULL	NULL	NULL	NULL	NULL	NULL

credit_card 2 x

Output

Action Output

#	Time	Action	Message
1	20:25:18	SELECT * FROM credit_card WHERE id ="CcU-2938"	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

Es necessari crear primer el registre de la companyia amb id b-9999 i la crèdit_card CcU-9999 perquè la transacció està lligada a una companyia (i també a una credit_card)

```
-- es necessari introduir primer la companyia en la taula company i la credit card en la per introduir una transacció
INSERT INTO company (
  id,
  company_name,
  phone,
  email,
  country,
  website
) VALUES (
  'b-9999',
  NULL,
  NULL,
  NULL,
  NULL,
  NULL
);
```

tion Output

Time	Action	Message
1 20:56:02	INSERT INTO company (id, company_name, phone, email, country, website) VALUES ('b-9999', NULL, NULL, NULL, ...	1 row(s) affected

```

75 • INSERT INTO credit_card (
76     id,
77     iban,
78     pan,
79     pin,
80     cvv,
81     expiring_date
82 ) VALUES (
83     'CcU-9999',
84     'TR304050312213576927638662',
85     '5101552687251312',
86     '3852',
87     '564',
88     '01/09/28'
89 );

```

Output

Action Output

#	Time	Action	Message
✓ 1	10:09:08	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-9999', 'TR3...	1 row(s) affected

Ara ja podem carregar el que demana l'exercici:

```

90 • INSERT INTO transaction (
91     id,
92     credit_card_id,
93     company_id,
94     user_id,
95     lat,
96     longitude,
97     amount,
98     declined
99 ) VALUES (
100     '10881D1D-5B23-A76C-55EF-C568E49A99DD',
101     'CcU-9999',
102     'b-9999',
103     9999,
104     829.999,
105     -117.999,
106     111.11,
107     0
108 );
109

```

Output

Action Output

#	Time	Action	Message
✓ 1	10:18:03	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, ...	1 row(s) affected

Exercici 4

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

```
112 • ALTER TABLE credit_card
113 DROP COLUMN pan;
```

Output			
Action Output			
#	Time	Action	Message
1	10:27:27	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Mostrar que s'ha realitzat el canvi:

```
115 • DESCRIBE credit_card;
```

Result Grid						
Filter Rows:						
Export: Wrap Cell Content:						
	Field	Type	Null	Key	Default	Extra
▶	id	varchar(10)	NO	PRI	HULL	
	iban	varchar(34)	NO		HULL	
	pin	char(4)	NO		HULL	
	cvv	char(3)	NO		HULL	
	expiring_date	varchar(10)	NO		HULL	

Result 1 x

Output			
Action Output			
#	Time	Action	Message
1	10:28:20	DESCRIBE credit_card	5 row(s) returned

Ara vaig a canviar el format de expiring_date per si en el futur he de fer càlculs amb la data. Després de canviar amb STR_TO_DATE hem de modificar la taula amb un ALTER.


```

118  /* UPDATE necessita una condició (és una opció que impedeix fer UPDATE o DELETE sense un WHERE amb clau primària o índex)
119  però em continua donant error 1175. Desactivo i activo de nou el safe Mode*/
120  • SET SQL_SAFE_UPDATES = 0;
121  • UPDATE credit_card
122    SET expiring_date = STR_TO_DATE(expiring_date, '%m/%d/%y');
123  • SET SQL_SAFE_UPDATES = 1; -- per tornar a activar-lo
124
125  • DESCRIBE credit_card; -- encara apareix com VARCHAR hem de modificar la taula amb expiring_date com DATE
126  • ALTER TABLE credit_card
127    MODIFY expiring_date DATE NOT NULL;
128  • DESCRIBE credit_card;
129

```

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

	Field	Type	Null	Key	Default	Extra
▶	id	varchar(10)	NO	PRI	NULL	
	iban	varchar(34)	NO		NULL	
	pin	char(4)	NO		NULL	
	cvv	char(3)	NO		NULL	
	expiring_date	date	NO		NULL	

Result 3 x

Output

Action Output

#	Time	Action	Message
✓ 3	10:54:07	SET SQL_SAFE_UPDATES = 1	0 row(s) affected
✓ 4	10:54:31	DESCRIBE credit_card	5 row(s) returned
✗ 5	11:04:35	CREATE TABLE credit_card (id VAR...	Error Code: 1050. Table 'credit_card' already exists
✓ 6	11:23:26	DESCRIBE credit_card	5 row(s) returned
✓ 7	11:28:35	ALTER TABLE credit_card MODIFY e...	5001 row(s) affected Records: 5001 Duplicates: 0 Warnings: 0
✓ 8	11:29:23	DESCRIBE credit_card	5 row(s) returned

Nivell 2

Exercici 1

Elimina de la taula transaction el registre amb ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de dades.

```

133  • DELETE FROM transaction
134  WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';

```

Output

Action Output

#	Time	Action	Message
✓ 1	12:41:53	DELETE FROM transaction WHERE id ...	1 row(s) affected

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.

144 • CREATE VIEW VistaMarketing AS
 145 SELECT c.company_name, c.phone, c.country, ROUND(AVG(t.amount),2) AS mitjanaCompra
 146 FROM company c
 147 LEFT JOIN transaction t
 148 ON c.id = t.company_id
 149 GROUP BY c.company_name, c.phone, c.country
 150 ORDER BY mitjanaCompra DESC;
 151
 152 • SHOW CREATE VIEW VistaMarketing;

View	Create View	character_set_client	collation_connection
vistamarketing	CREATE ALGORITHM=UNDEFINED DEFINER='r...'	utf8mb4	utf8mb4_0900_ai_ci

Result 8 ×

Output

#	Time	Action	Message
1	13:28:43	CREATE VIEW VistaMarketing AS SELECT c.company_name, c.phone, c.country, ROUND(AVG(t.amount),2) AS mitjanaCompra FROM company c LEFT JOIN transaction t ON c.id = t.company_id GROUP BY c.company_name, c.phone, c.country ORDER BY mitjanaCompra DESC;	0 row(s) affected
2	13:28:50	SHOW CREATE VIEW VistaMarketing	1 row(s) returned

Exercici 3

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

158 • SELECT *
 159 FROM vistaMarketing
 160 WHERE country='Germany';

company_name	phone	country	mitjanaCompra
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.87
Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.32
Convallis In Incorporated	06 66 57 29 50	Germany	257.75
Ac Industries	09 34 65 40 60	Germany	255.15
Rutrum Non Inc.	02 66 31 61 09	Germany	255.14
Auctor Mauris Corp.	05 62 87 14 41	Germany	254.77
Augue Foundation	06 88 43 15 63	Germany	253.51
Aliquam PC	01 45 73 52 16	Germany	253.14

vistaMarketing 9 ×

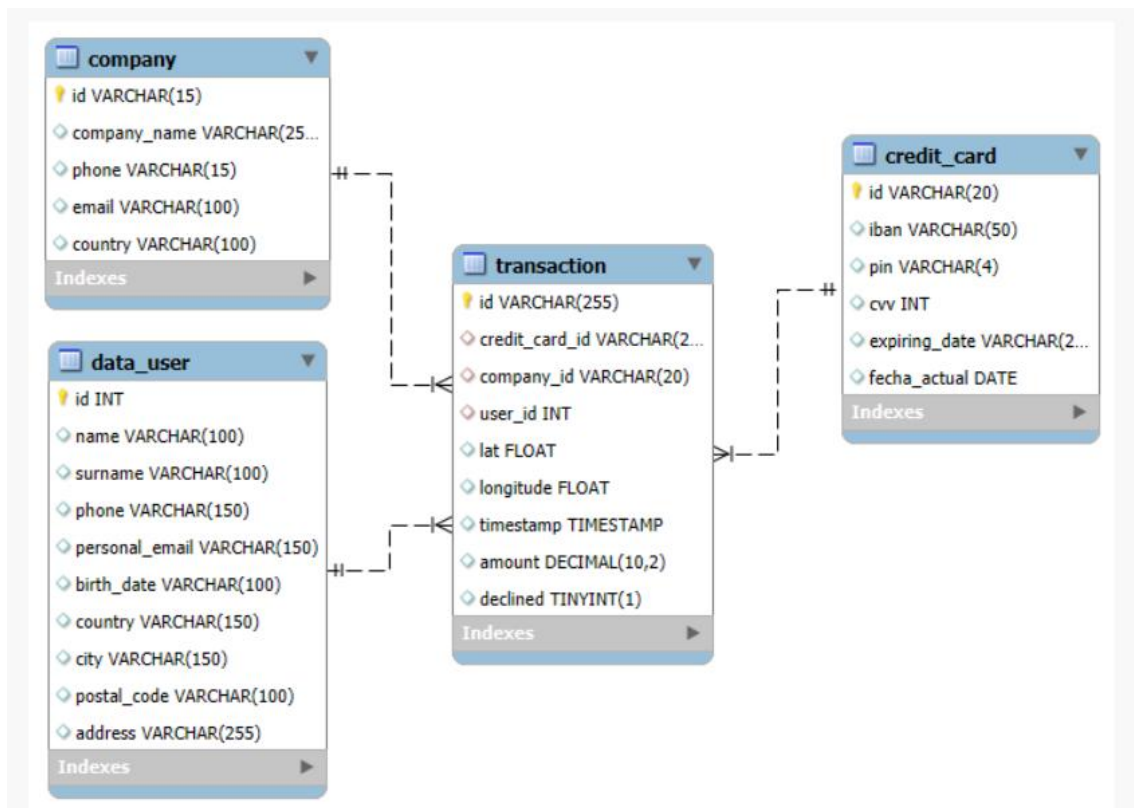
Output

#	Time	Action	Message
1	13:30:22	SELECT * FROM vistaMarketing WHERE country='Germany'	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"

Recorda continuar treballant sobre el model i les taules amb les quals ja has treballat fins ara.

- 1) Assumeixo que ja he fet tot el que he fet als exercicis anteriors
- 2) Creo la taula user executant el script de "estructura datos user"

```

1 • CREATE TABLE IF NOT EXISTS user (
2     id CHAR(10) PRIMARY KEY,
3     name VARCHAR(100),
4     surname VARCHAR(100),
5     phone VARCHAR(150),
6     email VARCHAR(150),
7     birth_date VARCHAR(100),
8     country VARCHAR(150),
9     city VARCHAR(150),
10    postal_code VARCHAR(100),
11    address VARCHAR(255)
12 );
13

```

Output			
Action Output			
#	Time	Action	Message
1	13:42:52	CREATE TABLE IF NOT EXISTS user (id CHAR(10) PRIMARY KEY, name VARCHAR(100), surname VARC...	0 row(s) affected

3) Carrego les dades de datos_introducir_sprint3user

Output			
Action Output			
#	Time	Action	Message
4996	14:00:10	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (...)	1 row(s) affected
4997	14:00:10	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (...)	1 row(s) affected
4998	14:00:10	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (...)	1 row(s) affected
4999	14:00:10	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (...)	1 row(s) affected
5000	14:00:10	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (...)	1 row(s) affected
5001	14:00:10	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (...)	1 row(s) affected
5002	14:00:10	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (...)	1 row(s) affected
5003	14:00:10	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (...)	1 row(s) affected

4) Comprovo que s'han carregat els 5000 registres

```
1 • SELECT * FROM transactions.user;
```

Result Grid

Filter Rows:

Edit

Export/Import:

Wrap Cell Content:

Fetch rows:

	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	New York	10001	348-7818 Sagittis St.
	10	Robert	Mccarthy	(324) 746-6771	fermentum@protonmail.com	Apr 30, 1984	United States	San Jose	95101	P.O. Box 773
	100	Melodie	Mclean	1-677-221-7152	risus.varius@google.ca	Sep 15, 1989	United States	San Jose	95101	Ap #644-8492 Sagittis St.
	1000	Amikrv	Qbulrxbp	+48-258-9936	amikrv.qbulrxbp@example.com	May 17, 1970	Germany	Stuttgart	70173	215 Qbulrxbp St
	1001	Nfvrjb	Oydaivbg	+94-121-2522	nfvrjb.oydaivbg@example.com	Mar 4, 1994	Germany	Cologne	50667	121 Oydaivbg St
	1002	Ijbfmd	Jbdddhyv	+70-120-3668	ijbfmd.jbdddhyv@example.com	Sep 27, 2001	Germany	Munich	80331	412 Jbdddhyv St

user 1 ×

Output

Action Output

#

Time

Action

Message

1

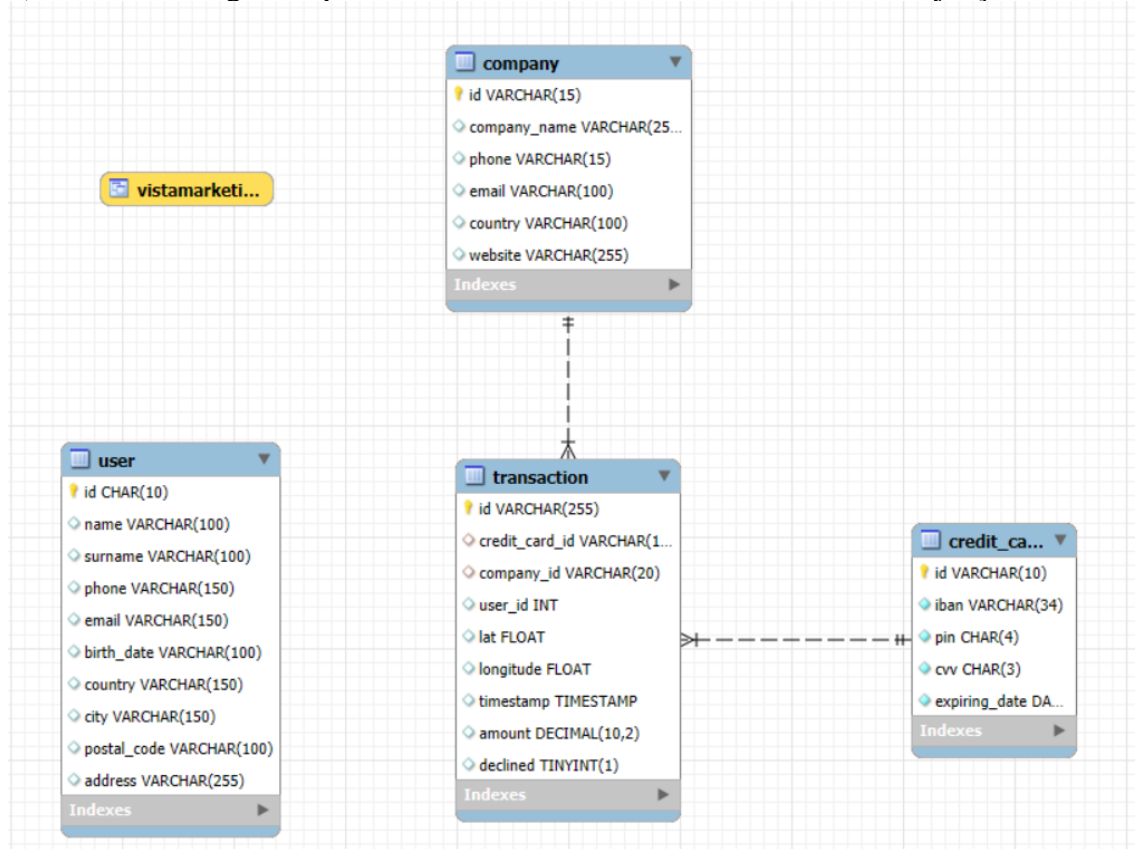
16:41:24

SELECT * FROM transactions user

5000 row(s) returned

Output			
Action Output			
#	Time	Action	Message
1	16:41:24	SELECT * FROM transactions.user	5000 row(s) returned

5) Ara tinc un diagrama que he de relacionar amb la taula transaction mitjançant user_id



6) **Nota:** transaction.user_id es INT i user.id és CHAR(10). Haurien de ser iguals per poder crear la clau forànea. Ho he fet després de comprovar que no hi ha registres “orfes” (Punt9). El més correcte hauria estat donar-se compte aquí

7) Abans de però crear la clau forànea comprovo que tots els valors de transaction.user_id tenen correspondència en user.id (trobar els valors “orfes”). Si hi haguessin s’haurien de corregir.

```

1 • SELECT t.user_id
2   FROM transaction t
3  LEFT JOIN user u
4    ON t.user_id = u.id
5  WHERE u.id IS NULL AND t.user_id IS NOT NULL;
6

```

Result Grid

user_id
9999

Output

Action Output

#	Time	Action	Message
1	17:52:06	SELECT t.user_id FROM transaction t LEFT JOIN user u ON t.user_id = u.id WHERE u.id ...	1 row(s) returned

8) Elimino el registre user.id=9999 en mode “segur” (Nota: s’havia creat a l’exercici3 Nivell 1). Un altre opció seria demanar la informació als departaments corresponents per omplir les altres dues taules (user & credit_card)

```

186 • SET SQL_SAFE_UPDATES = 0;
187 • DELETE FROM transaction
188     WHERE user_id=9999;
189 • SET SQL_SAFE_UPDATES = 1;
190

```

Output				
Action Output				
#	Time	Action	Message	
✓ 1	18:09:26	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	
✓ 2	18:09:31	DELETE FROM transaction WHERE user_id=9999	1 row(s) affected	
✓ 3	18:09:38	SET SQL_SAFE_UPDATES = 1	0 row(s) affected	

9) Comprovo que no queda cap registre “orfe”.

```

1 • SELECT *
2   FROM transaction t
3  LEFT JOIN user u
4    ON t.user_id = u.id
5  WHERE u.id IS NULL AND t.user_id IS NOT NULL;
6

```

Result Grid

Filter Rows:

Export

Wrap Cell Content:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	id	name	surname	phone	email	birth_date	country	city	postal_code	address
----	----------------	------------	---------	-----	-----------	-----------	--------	----------	----	------	---------	-------	-------	------------	---------	------	-------------	---------

10) Nota: transaction.user_id es INT i user.id és CHAR(10). Haurien de ser iguals per poder crear la clau forànea. Els índexs sobre INT són més ràpids i ocupen menys espai que CHAR(10) i al diagrama presentat està com INT. Ho canviem eliminant primer la clau primària que ja està creada:

```

182 • ALTER TABLE user DROP PRIMARY KEY;

```

Output				
Action Output				
#	Time	Action	Message	
✓ 1	23:33:45	ALTER TABLE user ...	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0	

11) Crear la clau primària de nou amb AUTOINCREMENT

```

184 • ALTER TABLE user
185     MODIFY id INT AUTO_INCREMENT PRIMARY KEY;
186

```

Output				
Action Output				
#	Time	Action	Message	
✓ 1	23:40:09	ALTER TABLE user ...	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0	

12) Crear la clau forànea


```

206 • ALTER TABLE transaction
207     ADD CONSTRAINT fk_transaction_user_id
208     FOREIGN KEY (user_id)
209     REFERENCES user(id);

```

Output					
Action Output					
#	Time	Action	Message	Duration / F	
✓ 1	23:45:42	ALTER TABLE transa...	99999 row(s) affected Records: 99999 Duplicates: 0 Warnings: 0	7.547 sec	

13) Afegir la columna que falta a la taula crèdit_card

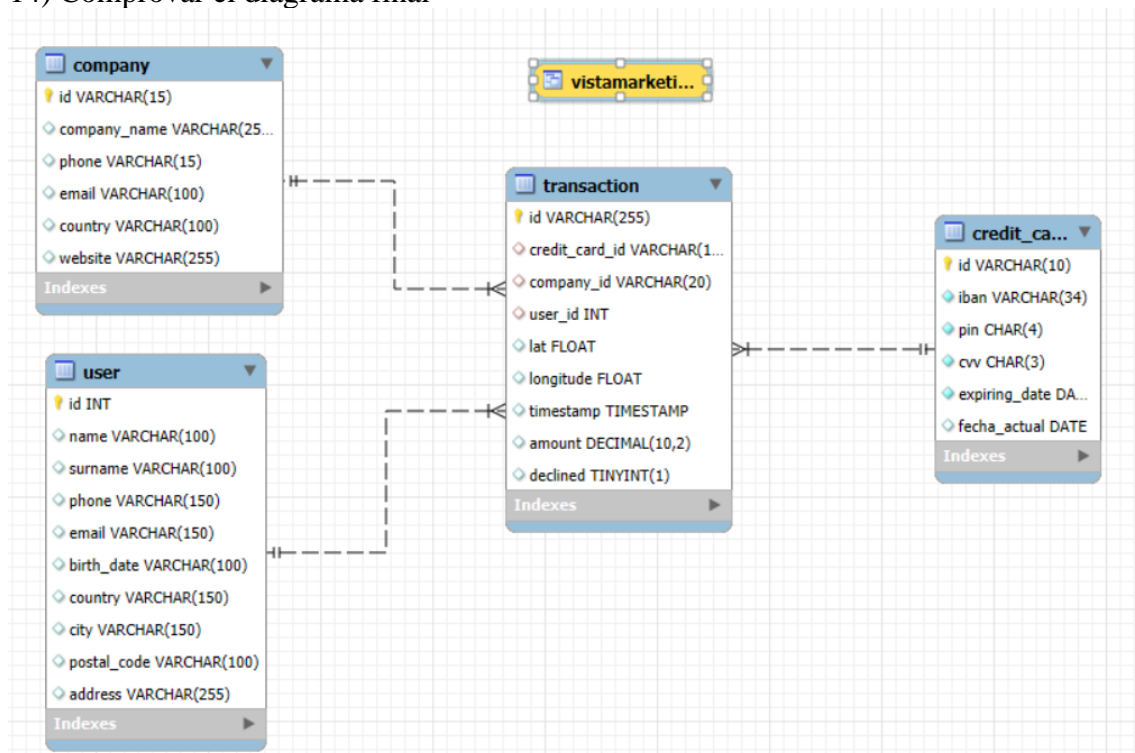
```

212 • ALTER TABLE credit_card
213     ADD COLUMN fecha_actual DATE;

```

Output					
Action Output					
#	Time	Action	Message		
✓ 1	23:58:52	ALTER TABLE credit_card ADD COLUMN fecha_act...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0		

14) Comprovar el diagrama final



Exercici 2

L'empresa també us demana 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.
- Assegureu-vos d'incloure informació rellevant de les taules que coneixereu i utilitzeu àlies per canviar de nom columnes segons calgui.

Mostra els resultats de la vista, ordena els resultats de forma descendent en funció de la variable ID de transacció.

```

225 • CREATE VIEW InformeTecnico AS
226 SELECT t.id, u.name, u.surname, cc.iban, c.company_name
227 FROM company c
228 LEFT JOIN transaction t
229 ON c.id = t.company_id
230 LEFT JOIN user u
231 ON t.user_id = u.id
232 LEFT JOIN credit_card cc
233 ON t.credit_card_id = cc.id
234 ORDER BY t.id DESC;
235
236

```

Output			
Action Output			
#	Time	Action	Message
✓ 1	00:14:54	CREATE VIEW InformeTecnico AS SELECT t.id, u.na...	0 row(s) affected

Mostrar resultats de la vista:

236 • `SELECT * FROM InformeTecnico;`

Result Grid

Filter Rows:

Export:

Wrap Cell Contents:

Fetch rows:

	id	name	surname	iban	company_name
▶	FFFD31D6-9495-47CE-B54A-7DB8E1CC274B	Bmrgli	Tprvmrc	XX794814451211289182490922	Turpis Company
	FFFCF76D-ECF0-4985-A2D0-B2A7B75998FC	Dfired	Vilqcdl	XX636251701647892036676034	Amet Nulla Donec Corporation
	FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	Securp	Faofvqfy	XX162677143304223631437567	Nunc Interdum Incorporated
	FFFB270D-F53A-4D5D-9666-E5307C53CC84	Ggzjpa	Uirzjulh	XX395114267082019952567052	Viverra Donec Foundation
	FFF9E3CE-234E-408C-A8EF-F9CAD577224A	Yshimq	Zpsjsleed	XX8845462156537570367941	Convallis In Incorporated

InformeTecnico 12 x

Output

Action Output

#	Time	Action	Message
✓ 1	00:16:24	SELECT * FROM InformeTecnico	100000 row(s) returned