

SPRINT 3. Manipulació de taules

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.

En primer lloc, per crear la taula credit_card és necessari explorar les dades del document 'dades_introduir_credit' per saber quins camps té la base de dades que hem de crear. A partir d'aquí sabem que existeixen 6 diferents camps i el tipus de dade que enmagatzemen. Per tant, creem:

- ID com a clau primària i amb valor alfanumèric (20 caràcters)
- IBAN com a camp alfanumèric de 50 caràcters i amb la característica de que no pot ser NULL
- pan (com a alfanumèric de 50 caràcters)
- pin (alfanumèric de 4 caràcters)*
- cvv (alfanumèric de 3 caràcters)*
- expiring_date (malgrat ser una data apareix com un camp alfanumèric, ja que presenta un format no-anglosaxó on les dades s'ordenen per dia-mes-any

```

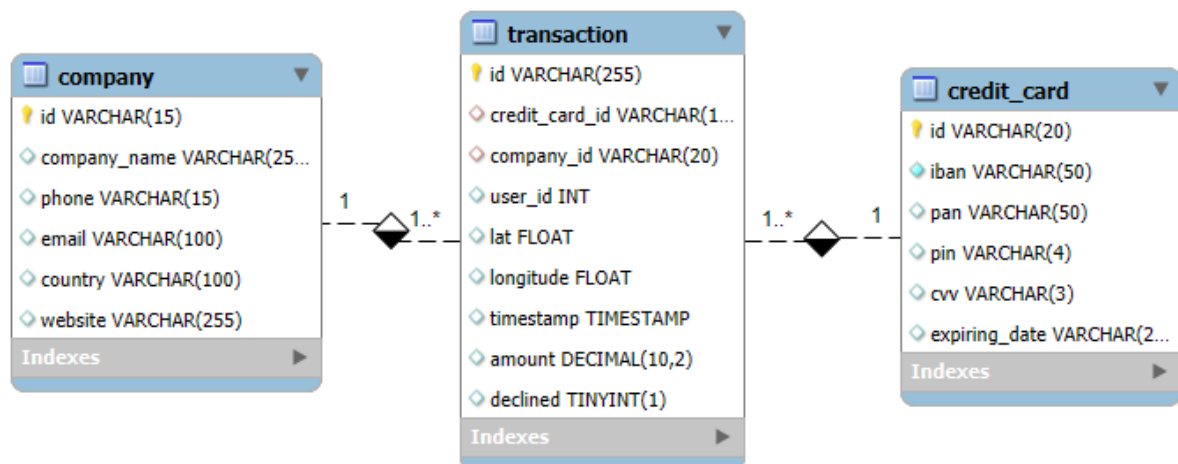
8 • USE transactions;
9
10 • CREATE TABLE credit_card (
11     id VARCHAR(20) PRIMARY KEY,
12     iban VARCHAR(50) NOT NULL UNIQUE,
13     pan VARCHAR (50),
14     pin VARCHAR (4),
15     cvv VARCHAR (3),
16     expiring_date VARCHAR(20),
17     FOREIGN KEY(id) REFERENCES transaction(credit_card_id)
18 );
19

```

Output

#	Time	Action	Message
22	13:02:37	CREATE TABLE credit_card (id VARCHAR(20) PRIMARY KEY, iban VARCHA...	0 row(s) affected

*Aquests camps els podríem considerar INT però en observar l'arxiu de dades, trobem que han estat introduïts com a string (').



El diagrama presenta la relació entre les taules, un cop creada la nova taula *credit_card*. Podem observar que és un **model d'estrella** on la taula de fets continua sent la taula *transaction* i la taula *company* i *credit_card* són taules de dimensions.

Les relacions entre les taules són:

- Relació d'1 a molts entre *credit_card* i *transaction* per mitjà de la PK *id* a *credit_card* / FK *credit_card_id*
- Relació d' 1 a molts entre *company* i *transaction* per mitjà de la PK *id* a *company* / FK *company_id*

Per tal d'establir la nova relació entre la taula *transaction* i la taula *credit_card* he fet servir els següents comandaments.

1. Establim el Foreign checks com a 0 per a que no executi la relació entre les taules pare i fill i ens permet crear la taula i introduir les dades. Aquest comandament

apareix per defecte com a 1, per tant l'hem de canviar sempre que introduïm dades codependents.

```
19
20 • SET FOREIGN_KEY_CHECKS = 0;
21
```

Output

Action Output

#	Time	Action	Message
✓ 23	13:02:40	SET FOREIGN_KEY_CHECKS = 0	0 row(s) affected

2. Introduïm les dades a partir de l'arxiu 'datos_introducir_credit'.

```
22 -- Insertamos datos de credit_card
23 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2938', 'TR301950312213576817638661', '5
24 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2945', 'D026854763748537475216568689',
25 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2952', 'B645IVQL52710525608255', '4556
26 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2959', 'CR7242477244335841535', '372461
27 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2966', 'BG72LKTQ15627628377363', '44856
28 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2973', 'PT87806228135092429456346', '54
29 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2980', 'DE39241881883086277136', '40246
30 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2987', 'GE89681434837748781813', '3763
31 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2994', 'BH62714428368066765294', '34428
32 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3001', 'CY49087426654774581266832110',
33 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3008', 'LU507216693616119230', '4485744
34 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3015', 'PS119398216295715968342456821',
35 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3022', 'GT91695162850556977423121857',
36 • INSERT INTO credit card (id, iban, pan, pin, cvv, expiring date) VALUES ( 'CcU-3029', 'AZ62317413982441418123739746',
```

Output

Action Output

#	Time	Action	Message
✓ 295	13:03:28	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2938', 'TR301950312213576817638661', '5	1 row(s) affected

3. Canviem el comandament FOREIGN KEY CHECKS a 1. Per tal que en introduïr pròximament dades, existeixi la vinculació entre les taules i es mantingui la integritat de les dades.

```
299 • SET FOREIGN_KEY_CHECKS = 1;
300
```

Output

Action Output

#	Time	Action	Message
✓ 299	13:03:43	SET FOREIGN_KEY_CHECKS = 1	0 row(s) affected

- 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.

```

16 • UPDATE credit_card
17     SET iban = 'R323456312213576817699999'
18     WHERE id = 'CcU-2938';
19

```

Output :

#	Time	Action	Message
9	10:32:55	UPDATE credit_card SET iban = 'R323456312213576817699999' WHERE id = 'CcU-2938';	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

Per a fer aquest modificació he fet servir el comandament UPDATE, filtrant per l'usuari que necessitava el canvi.

- 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

En el moment d'introduir aquest nou registre ens trobem amb un ERROR 1452 de base de dades, ja que els camps 'company_id' i credit_card_id (que serien les taules 'pare' de transaction) no comptem amb aquests registres.

```

22 • INSERT INTO transaction
23     (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD'
24     'CcU-9999', 'b-9999', 9999, 829.999, -117.999, 111.11, 0);
25

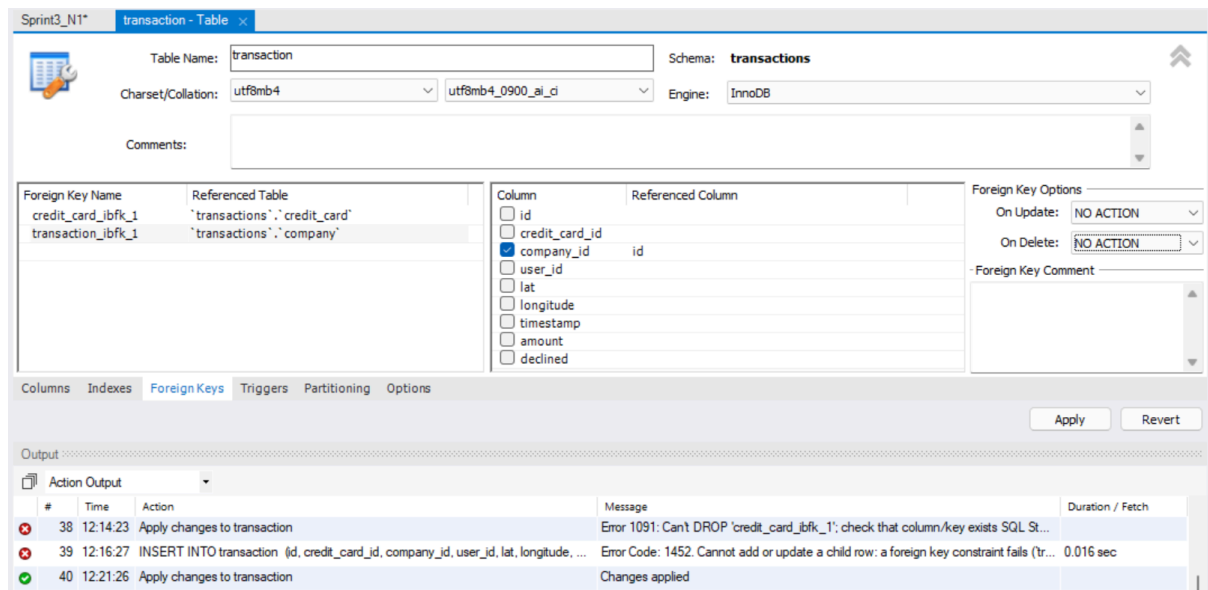
```

Output

#	Time	Action	Message
37	12:13:10	Apply changes to transaction	Error 1091: Can't DROP 'credit_card_ibfk_1'; check that column/key exists SQL St...
38	12:14:23	Apply changes to transaction	Error 1091: Can't DROP 'credit_card_ibfk_1'; check that column/key exists SQL St...
39	12:16:27	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, ...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (tr...

Per tant, hi ha dues formes de solucionar el problema:

- La més fàcil i ràpida és modificar les relacions entre la clau PK id a taula company i FK company_id, així com la clau PK id a credit_card / FK credit_card_id a transaction, per tal que quan es faci una actualització s'elimini el CONSTRAINT i la taula faci un NO ACTION ON UPDATE.



- La segona forma (i la que trobo que té més lògica) és afegir un nou registre a la taula company amb el codi de company facilitat a la taula anterior, ja que si ens trobem davant d'una empresa que està fent transaccions, ens interessa tenir un registre a la nostra base de dades amb la seva informació. Com que no disposem d'aquesta informació en aquest moment, introduïm un registre sense dades omplint el camp amb valors per evitar els constraints NOT NULL ja existents a les diferents taules.

```

25 • INSERT INTO company
26   (id, company_name, phone, email, country, website) VALUES ('b-9999', 'unknown', '000000000', 'unknown', 'unknown', 'unknown');
27
28 • INSERT INTO credit_card
29   (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-9999', '00000000000000000000', '0000000000', '0000', '000', '01-01-2028');
30
31 • INSERT INTO transaction
32   (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ('108B1D1D-5823-A76C-55EF-C568E49A990D',
33   'CcU-9999', 'b-9999', 9999, 829.999, -117.999, 111.11, 0);
34

```

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
53	12:33:53	INSERT INTO company (id, company_name, phone, email, country, website) VALU...	1 row(s) affected	0.015 sec
54	12:33:56	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (CcU-99...	1 row(s) affected	0.015 sec
55	12:33:59	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, ...	1 row(s) affected	0.016 sec

Considero que la segona sol·lució és més addient davant d'un problema real, per mantenir la integritat de les dades, sense manipular les característiques de les claus i per consistència amb les relacions preestablertes.

- Exercici 4

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

```
45 • ALTER TABLE credit_card DROP COLUMN pan;
```

Output

Action Output

#	Time	Action	Message
✓ 1	18:21:14	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Per a eliminar una columna he fet servir el comandament ALTER TABLE i, posteriorment DROP COLUMN.

Nivell 2

Exercici 1

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

```
26 #Elimina de la taula transaction el registre amb ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de dades.  
27 • DELETE FROM transaction  
28 WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

Output

Action Output

#	Time	Action	Message
✓ 1	10:29:38	DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'	1 row(s) affected

Per a eliminar la transacció mencionada he emprat el comandament DELETE FROM i he filtrat el registre exacte que volia eliminar amb un WHERE.

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.

```

335 • CREATE VIEW VistaMarketing AS
336 SELECT c.company_name, c.phone, c.country, ROUND(avg(t.amount),2) AS mitjana_compra
337 FROM company AS c
338 LEFT JOIN transaction AS t
339 ON c.id = t.company_id
340 GROUP BY 1, 3, 2
341 ORDER BY 4 DESC;

```

Output :

Action Output

#	Time	Action	Message
✓ 1	12:22:44	SELECT c.company_name, c.phone, c.country, ROUND(avg(t.amount),2) AS mitjana...	101 row(s) returned
✓ 2	12:22:50	CREATE VIEW VistaMarketing AS SELECT c.company_name, c.phone, c.country, ...	0 row(s) affected

Per a crear la view sol·licitada, primer he fet una query mitjançant un SELECT que obtingués les dades demanades. Com es pot veure a l'output anterior a la creació de la VIEW, els resultats de la query son 101 registres (el mateix número d'empreses de les quals disposem a la base de dades després de la creació de l'empresa de la qual no tenim dades encara). Posteriorment, un cop obtingudes les dades demandades, he creat la VIEW amb el comandament CREATE VIEW, sota el VistaMarketing. Per comprobar que la VIEW funciona correctament, he fet una query de totes les dades.

```

45 • SELECT *
46 FROM VistaMarketing;
47
48

```

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

	company_name	phone	country	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

VistaMarketing 3 x

Output :

Action Output

#	Time	Action	Message
✓ 33	12:05:28	SELECT * FROM VistaMarketing LIMIT 0, 5000	100 row(s) returned

Exercici 3

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

```

45 SELECT *
46 FROM VistaMarketing
47 WHERE country = 'Germany';
48
49
50

```

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

	company_name	phone	country	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

VistaMarketing 6 x

Output

Action Output

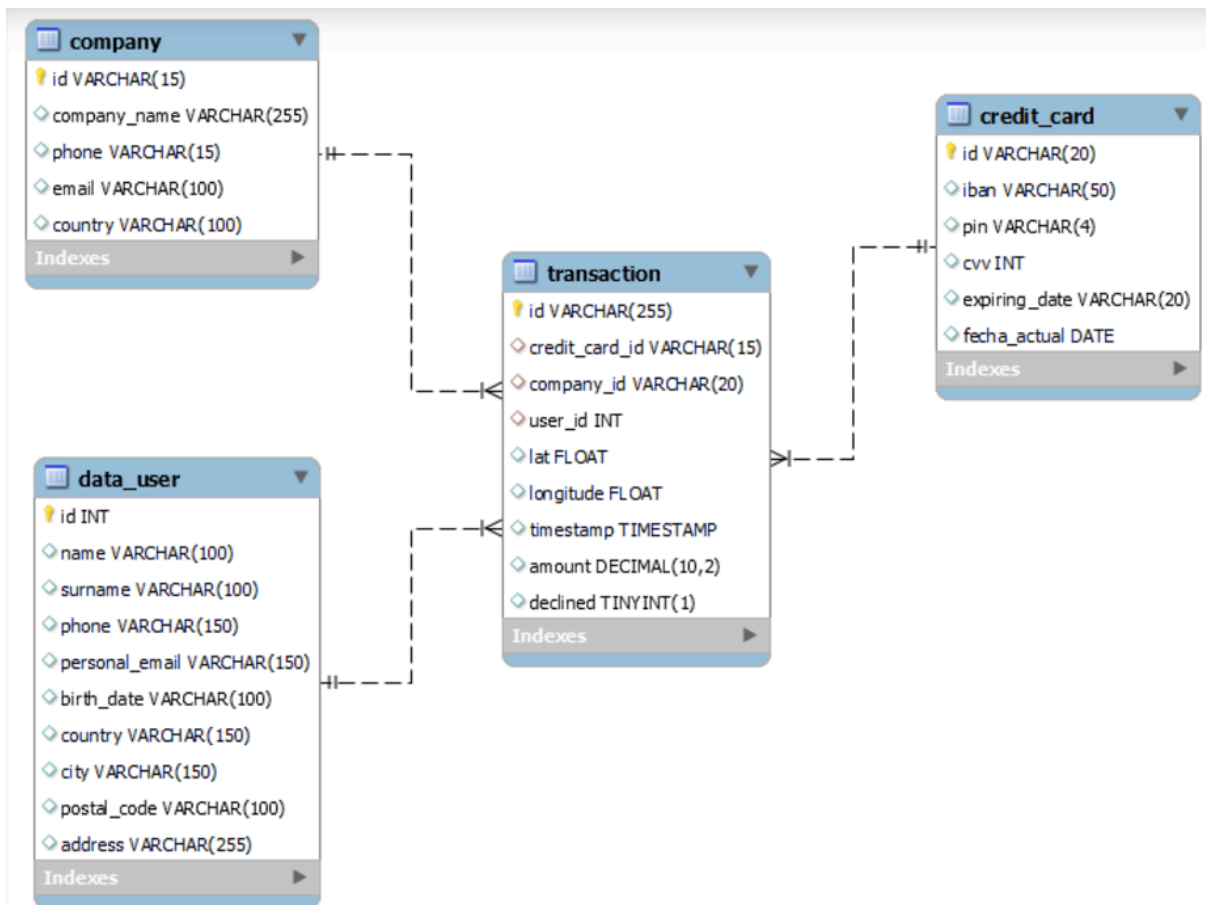
#	Time	Action	Message
✓ 28	10:44:10	SELECT c.company_name, c.phone, c.country, ROUND(avg(t.amount),2) AS mitja...	100 row(s) returned
✓ 29	10:44:27	CREATE VIEW VistaMarketing AS SELECT c.company_name, c.phone, c.count...	0 row(s) affected
✓ 30	10:45:51	SELECT * FROM VistaMarketing WHERE country = 'Germany' LIMIT 0, 5000	8 row(s) returned

Per a fer aquesta consulta he emprat la VIEW VistaMarketing i únicament he aplicat un filtre a país.

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"

En analitzar les modificacions realitzades pel meu company puc observar que hi han 4 accions principals que hem de dur a terme i el millor ordre per a realitzar-les és el següent:

1. De la taula credit_card cal modificar el format del camp CVV, el qual en un primer moment havia estat creat com un camp alfanumèric ja que en l'script on es troben els valors de la database apareixia en format d'*string*, però en el diagrama actualitzat volem aquest camp com a INT.

```
83 # PAS 1. Modificació de tipus de camp. Passar CVV de VARCHAR(3) a INT
84 • ALTER TABLE credit_card MODIFY cvv INT;
85
```

Output

Action Output

#	Time	Action
1	13:23:20	ALTER TABLE credit_card MODIFY cvv INT

Message

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

2. Afegir l'atribut `fecha_actual` a la taula `credit_card` amb format `DATE`

```
86 #Afegir la columna fecha_actual a la taula credit_card amb format DATE
87 • ALTER TABLE credit_card ADD COLUMN fecha_actual DATE;
88
```

Output

#	Time	Action	Message
1	13:26:42	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

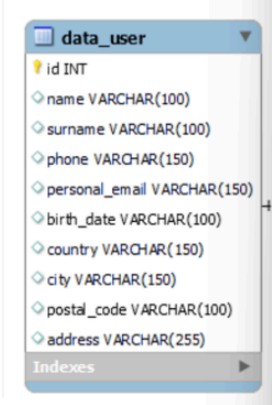
3. Eliminar l'atribut 'website' de la taula `company` ja que en el diagrama del nostre company aquesta dada no apareix.

```
89 #Eliminar l'atribut 'website' de la taula company.
90 • ALTER TABLE company DROP COLUMN website;
91
```

Output

#	Time	Action	Message
1	13:28:08	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

4. Creació de la taula `data_user`. Per a realitzar aquest canvi necessitem seguir un procés propi.
- a. Obrir l'script amb els valors que apareixen a l'arxiu 'estructura_dades_user' per a comprovar quines dades i de quin tipus són i comparar amb el diagrama que ens presenta el company per si hem de modificar algun tipus de dades. Es pot veure que no coincideixen ni el nom de la taula ni el de l'atribut 'email', els quals modificarem després d'introduir les dades per a que concordi amb el diagrama presentat.



```
4
5 • CREATE TABLE IF NOT EXISTS user (
6     id INT PRIMARY KEY,
7     name VARCHAR(100),
8     surname VARCHAR(100),
9     phone VARCHAR(150),
10    email VARCHAR(150),
11    birth_date VARCHAR(100),
12    country VARCHAR(150),
13    city VARCHAR(150),
14    postal_code VARCHAR(100),
15    address VARCHAR(255),
16    FOREIGN KEY(id) REFERENCES transaction(user_id)
17 );
```

```

369 #PAS 4. Creació de la taula data_user
370 • CREATE TABLE IF NOT EXISTS user (
371     id INT PRIMARY KEY,
372     name VARCHAR(100),
373     surname VARCHAR(100),
374     phone VARCHAR(150),
375     email VARCHAR(150),
376     birth_date VARCHAR(100),
377     country VARCHAR(150),
378     city VARCHAR(150),
379     postal_code VARCHAR(100),
380     address VARCHAR(255),
381     FOREIGN KEY(id) REFERENCES transaction(user_id)
382 );

```

Output

Action Output

#	Time	Action	Message
1	12:26:03	CREATE TABLE IF NOT EXISTS user(id INT PRIMARY KEY, name VARC...	0 row(s) affected

- b. Afegim l'index que apareixia a l'inici del document d'estructura_dades_user per a vincular la PK de la taula data_user amb la taula transaction.

```

107 • CREATE INDEX idx_user_id ON transaction(user_id);
108
109

```

Output

Action Output

#	Time	Action	Message
1	13:44:01	CREATE INDEX idx_user_id ON transaction(user_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

- c. Afegim les dades de la taula data_user a la base de dades transactions per a poder treballar amb elles obrint l'script 'datos_introducir_user(1)' i fent córrer l'script. Tal com mostra l'output s'han afegit 277 registres a la taula user.

```

Sprint3_N1*  datos_introducir_user (1) x
Limit to 5000 rows

1 • SET foreign_key_checks = 0;
2
3 -- Insertamos datos de user
4 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "1", "Zeus", "Gamble",
5 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "2", "Garrett", "Mcconn
6 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "3", "Ciaran", "Harriso
7 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "4", "Howard", "Staffor
8 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "5", "Hayfa", "Pierce",
9 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "6", "Joel", "Tyson", "
10 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "7", "Rafael", "Jimenez
11 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "8", "Nissim", "Franks"
12 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "9", "Mannix", "Mcclain
13 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "10", "Robert", "Mccart
14 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "11", "Joan", "Baird",
15 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "12", "Benedict", "Whee

Output
Action Output
# Time Action Message Duration / Fetch
275 15:57:01 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal... 1 row(s) affected 0.016 sec
276 15:57:01 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal... 1 row(s) affected 0.000 sec
277 15:57:01 SET foreign_key_checks = 1 0 row(s) affected 0.000 sec

```

- d. Hem creat la taula com a *user* però volem modificar el nom de la taula per 'data_user' així que emprant el comandament RENAME ho modifiquem.

```

109 #Modificar el nom de la taula USER per data_user
110 • ALTER TABLE user RENAME data_user;
111

```

```

Output
Action Output
# Time Action Message
1 13:52:49 ALTER TABLE user RENAME data_user 0 row(s) affected

```

- e. Observem que hi ha una discrepància entre el nom de columna 'email' de l'script i el nom del model 'personal_email', així que modifiquem el nom de l'atribut.

```

395 #Modificar el nom de la columna email per personal_email
396 • ALTER TABLE data_user RENAME COLUMN email TO personal_email;
397

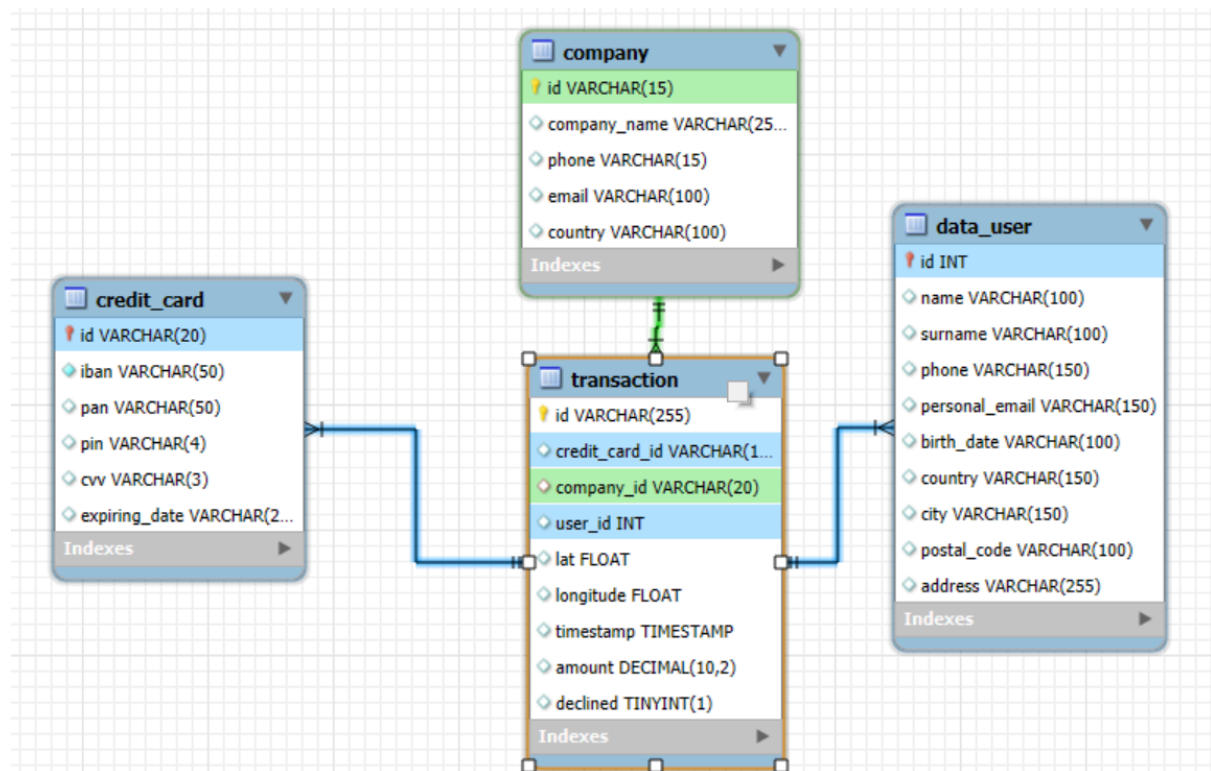
```

```

Output
Action Output
# Time Action Message
1 16:01:59 ALTER TABLE data_user RENAME COLUMN email TO personal_email 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

```

Per a verificar que les taules han estat creades correctament, segons el diagrama de l'exercici, podem consultar el nostre propi diagrama on apareixen tots els atributs i totes les relacions entre taules.



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.

```

684 • CREATE VIEW InformeTecnico AS
685 SELECT t.id AS id_transaction, du.name, du.surname, c.company_name, cd.iban
686 FROM transaction AS t
687 LEFT JOIN data_user AS du
688     ON du.id = t.user_id
689     LEFT JOIN company AS c
690         ON c.id = t.company_id
691         LEFT JOIN credit_card AS cd
692             ON cd.id = t.credit_card_id
693 ORDER BY t.id DESC;

```

Output

Action Output

#	Time	Action	Message
✓ 1	11:33:30	SELECT t.id AS id_transaction, du.name, du.surname, c.company_name, cd.iban FR...	587 row(s) returned
✓ 2	11:33:50	CREATE VIEW InformeTecnico AS SELECT t.id AS id_transaction, du.name, du.surname...	0 row(s) affected

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

```

695 • SELECT *
696 FROM InformeTecnico;
697

```

Result Grid Filter Rows: Export: Wrap Cell Content:

	id_transaction	name	surname	company_name	iban
▶	FE96CE47-8D59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	Magna A Neque Industries	DO26854763748537475216568689
	FE809ED4-2DB6-55AC-C915-929516E46468	Molly	Gilliam	Nunc Interdum Incorporated	SE2813123487163628531121
	FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	Nunc Interdum Incorporated	KW9485332754781757886242955643
	FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	Malesuada PC	LT053237077744561475
	FD2E8957-4148-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	Neque Tellus Imperdiet Corp.	GE84848451582810541526

InformeTecnico 2 x

Output

Action Output

#	Time	Action	Message
✓ 1	11:33:30	SELECT t.id AS id_transaction, du.name, du.surname, c.company_name, cd.iban FR...	587 row(s) returned
✓ 2	11:33:50	CREATE VIEW InformeTecnico AS SELECT t.id AS id_transaction, du.name, du.surname...	0 row(s) affected
✓ 3	11:34:46	SELECT * FROM InformeTecnico LIMIT 0, 50000	587 row(s) returned