

Tasca S3.01. Manipulació de taules

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.

Previo a crear la tabla, me he asegurado de que en la tabla transaction columna credit_card_id al ser Foreign key sea NOT NULL, que estaba creada así de origen, y como se define una foreign key debe permitir valores repetidos. Por eso he aplicado el alter table al principio.

Posteriormente he creado la tabla credit_card con create table con los valores tipo varchar y NOT NULL (posteriormente lo he modificado a NULL, para el ejercicio 3)

Y después he introducido los datos que nos proporcionaban con INSERT INTO.

De primeras no me dejaba ingresar los datos por que la columna expiring_date lo trataba de ingresar con tipo de dato DATE, y me daba un error: 1 row(s) affected, 1 warning(s): 4095 Delimiter '/' in position 2 in datetime value '11/11/24' at row 1 is deprecated. Prefer the standard '-'. Esto significa que no permite el formato '11/11/24' y que se debe cambiar al formato YYYY-MM-DD, entonces opté por cambiar el tipo de dato a VARCHAR. Si no la otra opción sería editar los valores de fecha uno a uno en origen.

Al introducir los datos en la nueva tabla, debía relacionarla con la tabla transaction mediante la primary key de la tabla nueva (credit_card.id) y la foreign key de transaction.credit_card_id. Tenía que hacerlo con ADD CONSTRAINT.

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

```
1  -- Nivell 1
2  -- Exercici 1
3  -- La teva tasca és dissenyar i crear una taula anomenada "credit_card" que emmagatzemi detalls crucials sobre les targetes de crèdit.
4  -- 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").
5  -- Després de crear la taula serà necessari que ingressis la informació del document denominat "dades_introduir_credit".
6  -- Recorda mostrar el diagrama i realitzar una breu descripció d'aquest.
7
8  • ALTER TABLE transaction MODIFY credit_card_id VARCHAR(15) NOT NULL;
9
10 • CREATE TABLE credit_card (
11     id VARCHAR(15) PRIMARY KEY,
12     iban VARCHAR(34) NOT NULL,
13     pan VARCHAR(25) NOT NULL,
14     pin VARCHAR(10) NOT NULL,
15     cvv VARCHAR(5) NOT NULL,
16     expiring_date VARCHAR(255) NOT NULL
17 );
18
19 -- Insertamos datos de credit_card
```

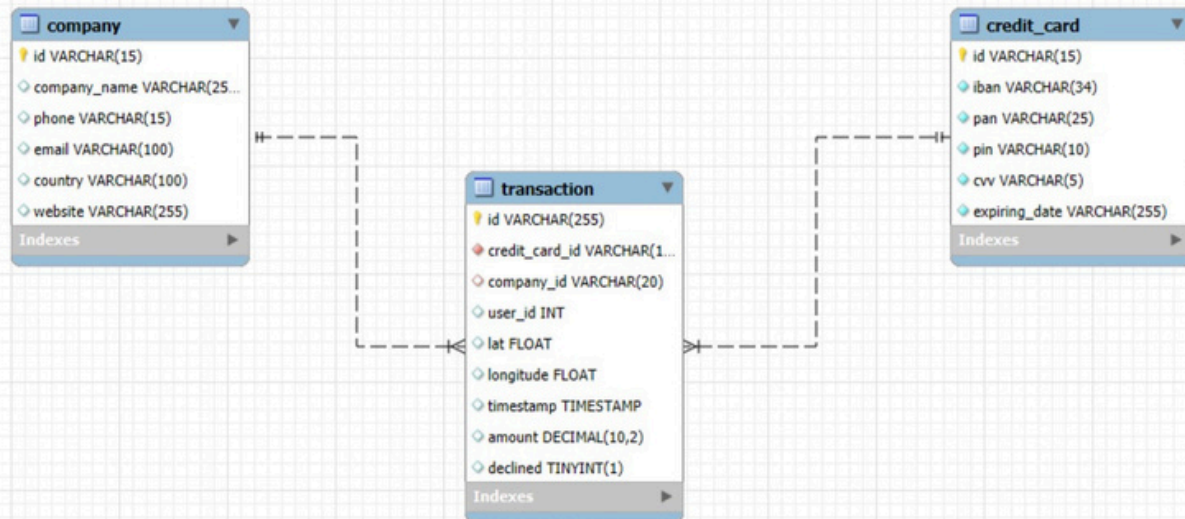
```

296 • ALTER TABLE transaction
297     ADD CONSTRAINT fk_transaction_credit_card
298     FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
299
300

```

Output			
Action Output			
#	Time	Action	Message
1	22:35:32	SELECT * FROM transactions.credit_card	275 row(s) returned
2	22:40:38	SHOW CREATE TABLE credit_card	1 row(s) returned
		Duration / Fetch	
		0.000 sec / 0.000 sec	
		0.016 sec / 0.000 sec	

Diagrama de la base de datos



El schema transactions tiene tres tablas:

- **company**: Está relacionado con la tabla transaction mediante la columna company.id (primary key) y la columna company_id que es la foreign key de la tabla transaction. Es una relación desde company a transaction, de 1 a muchos, es decir que una compañía puede tener muchas transacciones. En esta tabla almacena datos de las compañías que compran a la empresa.
- **Transaction**: Esta tabla almacena datos de las transacciones que se llevan a cabo, contiene dos foreign keys que hace posible que se relacione con las otras tablas (company_id que se relaciona con company, y credit_card_id que se relaciona con tabla credit_card)
- **credit_card**: Esta tabla se relaciona con transaction mediante la primary key credit.id y la columna credit_card_id (foreign key) de transaction. Contiene datos de las tarjetas de crédito con que se realizan las transacciones, y la relación desde credit_card a transaction es de 1 a muchos, es decir que con cada tarjeta de crédito se realizan muchas transacciones.

Exercici 2

El equipo de Recursos Humanos ha identificado un error en el número de cuenta del usuario con ID CcU-2938. El dato correcto que debe aparecer es:

R323456312213576817699999. ¡Recuerda resaltar que se ha realizado el cambio!

```
301 -- - Exercici 2
302 -- El departament de Recursos Humans ha identificat un error en el número de compte de l'usuari amb ID CcU-2938.
303 -- La informació que ha de mostrar-se per a aquest registre és: R323456312213576817699999. Recorda mostrar que el canvi es va realitzar.
304
305 • UPDATE credit_card
306 SET iban = 'R323456312213576817699999'
307 WHERE ID = 'CcU-2938';
308
309 • SELECT *
310 FROM credit_card
311 WHERE ID = 'CcU-2938';
312
313
314
315
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: I

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

credit_card 10 x

Output

Action Output

#	Time	Action	Message
1	00:29:55	UPDATE credit_card SET iban = 'R323456312213576817699999' WHERE ID = 'CcU-2938'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
2	00:31:31	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

CANTIDAD DE FILAS ANTES DE INTRODUCIR NUEVA LINEA: 587 filas

```

312 -- Exercici 3
313 -- En la taula "transaction" ingressa un nou usuari amb la següent informació:
314 -- id 108B1D1D-5B23-A76C-55EF-C568E49A99DD
315 -- credit_card_id CcU-9999
316 -- company_id b-9999
317 -- user_id 9999
318 -- lat 829.999
319 -- longitude -117.999
320 -- amount 111.11
321 -- declined 0
322
323 • SELECT COUNT(*)
324 FROM transaction;
325

```

Result Grid	Filter Rows:	Export:	Wrap Cell Contents:
COUNT(*)			
587			

- Antes de introducir nuevos datos en la tabla transaction, debemos tener en cuenta que:
 - Que en esta tabla transaction hay dos foreign keys que están referenciadas a las primary key de las otras tablas (company y credit_card)

transaction	
id VARCHAR(255)	
credit_card_id VARCHAR(100)	FK
company_id VARCHAR(20)	FK
user_id INT	
lat FLOAT	
longitude FLOAT	
timestamp TIMESTAMP	
amount DECIMAL(10,2)	
declined TINYINT(1)	
Indexes	

2. Al estar referenciadas estas primary keys, y por concepto no pueden tener valores nulos, entonces debemos tener datos en las tablas referencias y en las columnas de primary key, y en el resto de columnas que permitan nulos, porque no tenemos más datos. Por eso ingresamos datos de ese nuevo usuario en company(id) por que esta referenciada con transaction(company_id).

3. Y lo mismo que hicimos con la tabla company, ahora con la tabla credit_card en donde teníamos que crear un registro nuevo porque la pk de credit_card (id) esta referenciada en la tabla transaction (credit_card_id), el resto de los datos de las otras columnas de esta tabla serían nulos.

- INTRODUCIMOS NUEVO REGISTRO Y MOSTRAMOS LA NUEVA CANTIDAD DE FILAS DE LA TABLA TRANSACTION: 588 filas

```

50 • INSERT INTO company (id) VALUES ('b-9999');
51
52 • INSERT INTO credit_card (id) VALUES ('CcU-9999');
53
54 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
55 values ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '111.11', '0');
56
57 • SELECT COUNT(*)
58 FROM transaction;
59

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
COUNT(*)			
588			

Result 6 x

Output

Action Output

#	Time	Action	Message
✓ 1	20:40:01	INSERT INTO company (id) VALUES (b-9999)	1 row(s) affected
✓ 2	20:40:30	INSERT INTO credit_card (id) VALUES (CcU-9999)	1 row(s) affected
✓ 3	20:40:46	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) values ('108B1D...	1 row(s) affected
✓ 4	20:40:51	SELECT COUNT(*) FROM transaction	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.

Eliminamos la columna pan con alter table y drop column, verificamos que se haya eliminado con show columns.

```

63 • ALTER TABLE credit_card DROP COLUMN pan;
64
65 • SHOW COLUMNS FROM credit_card;
66

```

Field	Type	Null	Key	Default	Extra
id	varchar(255)	NO	PRI	NULL	
iban	varchar(255)	YES		NULL	
pin	varchar(255)	YES		NULL	
cvv	varchar(255)	YES		NULL	
expiring_date	varchar(255)	YES		NULL	

Result 7 x

Output

Action Output

#	Time	Action	Message
✓ 1	20:47:09	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 2	20:47:13	SHOW COLUMNS FROM credit_card	5 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.

Contamos el número de filas de la tabla transaction antes de eliminar un registro, son 588 filas

```
358 • SELECT COUNT(*) FROM transaction;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
COUNT(*)			
588			

Aplicamos el código para eliminar un registro que se realiza con DELETE y a continuación se cuenta nuevamente los registros para verificar que se haya eliminado el registro solicitado, son 587 filas.

```
360 • DELETE FROM transaction
361 WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
362
363 • SELECT COUNT(*) FROM transaction;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
count(*)			
587			

2	19:11:14	SELECT * FROM transactions.transaction	587 row(s) returned
3	19:11:57	SELECT count(*) FROM transactions.transaction	1 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.

```
87 • CREATE VIEW VistaMarketing AS
88 SELECT c.company_name, c.phone, c.country, round(AVG(t.amount),2) AS MediaVentas
89 FROM company c
90 INNER JOIN transaction t ON c.id = t.company_id -- where company_name is not null
91 GROUP BY c.id
92 ORDER BY MediaVentas DESC;
93
94 • SELECT * FROM VistaMarketing;
95
```

company_name	phone	country	MediaVentas
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 11 x

Output

Action Output

#	Time	Action	Message
3	21:49:38	CREATE VIEW VistaMarketing AS SELECT c.company_name, c.phone, c.country, round(AVG(t.amount),2) AS Med...	0 row(s) affected
4	21:49:47	SELECT * FROM VistaMarketing	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"

```
399 -- Exercici 3
400 -- Filtra la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"
401
402 • SELECT vm.company_name
403 FROM vistamarketing vm
404 WHERE vm.country = 'Germany';
405
406
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

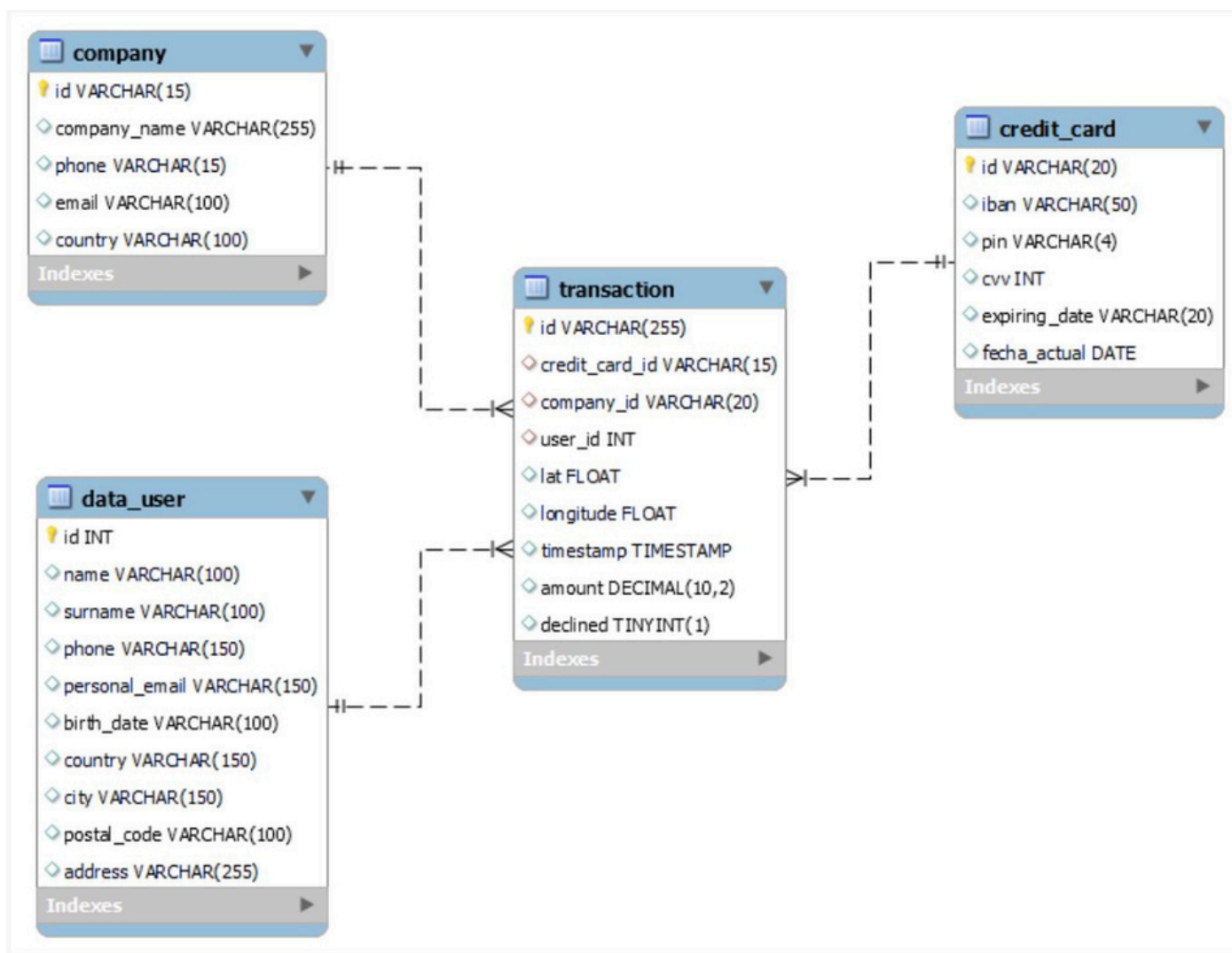
company_name
Aliquam PC
Ac Industries
Rutrum Non Inc.
Nunc Interdum Incorporated
Augue Foundation
Ac Fermentum Incorporated
Auctor Mauris Corp.
Convallis In Incorporated

vistamarketing 33 x			
Output			
Action Output			
#	Time	Action	Message
1	13:24:44	SELECT vm.company_name FROM vistamarketing vm WHERE vm.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:



Comenzamos rectificando los tipos de datos de las columnas de la tabla credit_card, para que queden como en el grafico dado.

Con desc credit_card podemos ver la columnas o campos de esa tabla, y según esto vamos modificando los tipos de datos y lo hacemos con Alter Table y Modify.

```

111 • DESC credit_card;
112
113 • ALTER TABLE credit_card MODIFY id VARCHAR(20);
114 • ALTER TABLE credit_card MODIFY iban VARCHAR(50);
115 • ALTER TABLE credit_card MODIFY pin VARCHAR(4);
116 • ALTER TABLE credit_card MODIFY cvv INT;
117 • ALTER TABLE credit_card MODIFY expiring_date VARCHAR(20);
  
```

```
• ALTER TABLE credit_card ADD fecha_actual DATE;
```

tion Output

Time	Action	Message
1 18:26:49	ALTER TABLE credit_card MODIFY iban VARCHAR(50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
2 18:26:49	ALTER TABLE credit_card MODIFY cvv INT	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
3 18:26:49	ALTER TABLE credit_card MODIFY expiring_date VARCHAR(65)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

También creamos una nueva columna en la tabla credit card llamada fecha actual como tipo de dato DATE, crea con Alter Table y ADD. Para comprobar que vamos bien encaminados, lo hacemos con DESC a la tabla credit card

Por último añadimos la columna expiring_date a varchar(20)

```
425 • ALTER TABLE credit_card ADD fecha_actual DATE;
426 • DESC credit_card;
427
```

Result Grid						
	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 8 x			
Output			
Action Output			
#	Time	Action	Message
✓ 1	12:00:03	CREATE VIEW VistaMarketingII AS SELECT c.company_name, c.phone, c.country, Medias.MediaVentas FROM com...	0 row(s) affected
✓ 2	12:00:03	SELECT * FROM VistaMarketingII	100 row(s) returned
✓ 3	12:07:32	ALTER TABLE credit_card MODIFY expiring_date VARCHAR(20)	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
✓ 4	12:10:23	DESC credit_card	6 row(s) returned

Acontinuación, creamos una nueva tabla llamada user e ingresamos los datos que tendrá dicha tabla

```
128 • CREATE TABLE IF NOT EXISTS user (
129     id INT PRIMARY KEY,
130     name VARCHAR(100),
131     surname VARCHAR(100),
132     phone VARCHAR(150),
133     email VARCHAR(150),
134     birth_date VARCHAR(100),
135     country VARCHAR(150),
136     city VARCHAR(150),
137     postal_code VARCHAR(100),
138     address VARCHAR(255)
139 );
140
141 -- Insertamos datos de user
142
```

Output			
Action Output			
#	Time	Action	Message
✓ 1	11:30:26	CREATE TABLE IF NOT EXISTS user (id INT PRIMARY KEY, name VARCHAR(100), sumame VAR...	0 row(s) affected
✓ 2	11:31:30	INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, address) VALUES (...	1 row(s) affected
✓ 3	11:31:31	INSFRT INTO user (id name sumame phone email birth date country city postal code address) VAI IIFS (1 row(s) affected

Cambiamos el nombre a la tabla user a data_user.

Tambien cambiamos el nombre ala columna email por personal_email.

Todo esto para adecuarlos al modelo que nos sugieren

```
147 -- cambiamos nombre a tabla
148 • RENAME TABLE user to data_user;
149
150 -- cambiamos nombre a columna
151 • ALTER TABLE data_user CHANGE email personal_email VARCHAR(150);
152 • DESC data_user;
153
```

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 15 x

Output

#	Time	Action	Message
✓ 1	12:09:41	RENAME TABLE user to data_user	0 row(s) affected
✓ 2	12:10:40	ALTER TABLE data_user CHANGE email personal_email VARCHAR(150)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 3	12:10:40	DESC data_user	10 row(s) returned

Creamos la columna primary key en id:

```
746 -- agregamos el constraint de primary key
747 • ALTER TABLE data_user ADD CONSTRAINT PRIMARY KEY(id);
748 • DESC data_user;
```

749

750

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 2 x

Output

#	Time	Action	Message
✓ 1	15:43:19	ALTER TABLE data_user ADD CONSTRAINT PRIMARY KEY(id)	0 row(s) affected
✓ 2	15:43:19	DESC data_user	10 row(s) returned

Al crear la relación entre las tablas data_user y transaction , se debería hacer con alter table y add constraint, pero dicha sentencia da un error:

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (transactions.#sql-16fc_8d, CONSTRAINT fk_trans_data_user FOREIGN KEY (user_id) REFERENCES data_user (id))

Este error nos indica que hay un valor en transaction que no está referido en la tabla data_user (id), que es primary key

```
750
751 -- Creamos la relacion entre la tabla data_user y transaction de uno a muchos
752 • ALTER TABLE transaction
753 ADD CONSTRAINT fk_trans_data_user
754 FOREIGN KEY(user_id) REFERENCES data_user(id);
755
756
```

Output

#	Time	Action	Message
1	16:24:57	ALTER TABLE transaction ADD CONSTRAINT fk_trans_data_user FOREIGN KEY(user_id) REFERENCES data_user(...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (transactions' '#sql-16fc_8d', CONST... 0

Entonces buscamos el valor que no existe en la tabla data_user y que sí está en transaction:

```
747 • SELECT user_id
748 FROM transaction
749 WHERE user_id NOT IN (SELECT id FROM data_user);
750
```

Result Grid

user_id
9999

transaction 7 x

Output

#	Time	Action	Message
1	16:50:00	SELECT user_id FROM transaction WHERE user_id NOT IN (SELECT id FROM data_user)	1 row(s) returned

Y es el user_id = 9999

Agregamos este valor en la tabla data_user

```
751 • INSERT INTO data_user (id)
752 SELECT t.user_id
753 FROM transaction t
754 WHERE t.user_id NOT IN (SELECT d.id FROM data_user d);
755
```

Output

#	Time	Action	Message
1	16:57:24	INSERT INTO data_user (id) SELECT t.user_id FROM transaction t WHERE t.user_id NOT IN (SELECT d.id FROM d...	1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0

Y verificamos que no haya datos en la tabla transaction que no esten en la tabla data_user

```
756 • SELECT t.user_id
757 FROM transaction t
758 WHERE t.user_id NOT IN (SELECT d.id FROM data_user d);
759
760 -- Creamos la relacion entre la tabla data_user y transaction de uno a muchos
```

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

user_id

transaction 10 x			
Output			
Action Output			
#	Time	Action	Message
✓ 1	17:06:40	SELECT t.user_id FROM transaction t WHERE t.user_id NOT IN (SELECT d.id FROM data_user d)	0 row(s) returned

Y como no hay datos, podemos volver a ejecutar la sentencia anterior de agregar la constraint de la foreign key:

```
760 -- Creamos la relacion entre la tabla data_user y transaction de uno a muchos
761 • ALTER TABLE transaction
762 ADD CONSTRAINT fk_trans_data_user
763 FOREIGN KEY(user_id) REFERENCES data_user(id);
764
765
```

Output

Action Output

#	Time	Action	Message
✓ 1	17:08:37	ALTER TABLE transaction ADD CONSTRAINT fk_trans_data_user FOREIGN KEY(user_id) REFERENCES data_user(...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

En la tabla company hay que eliminar una columna website para que quede como el modelo esperado.

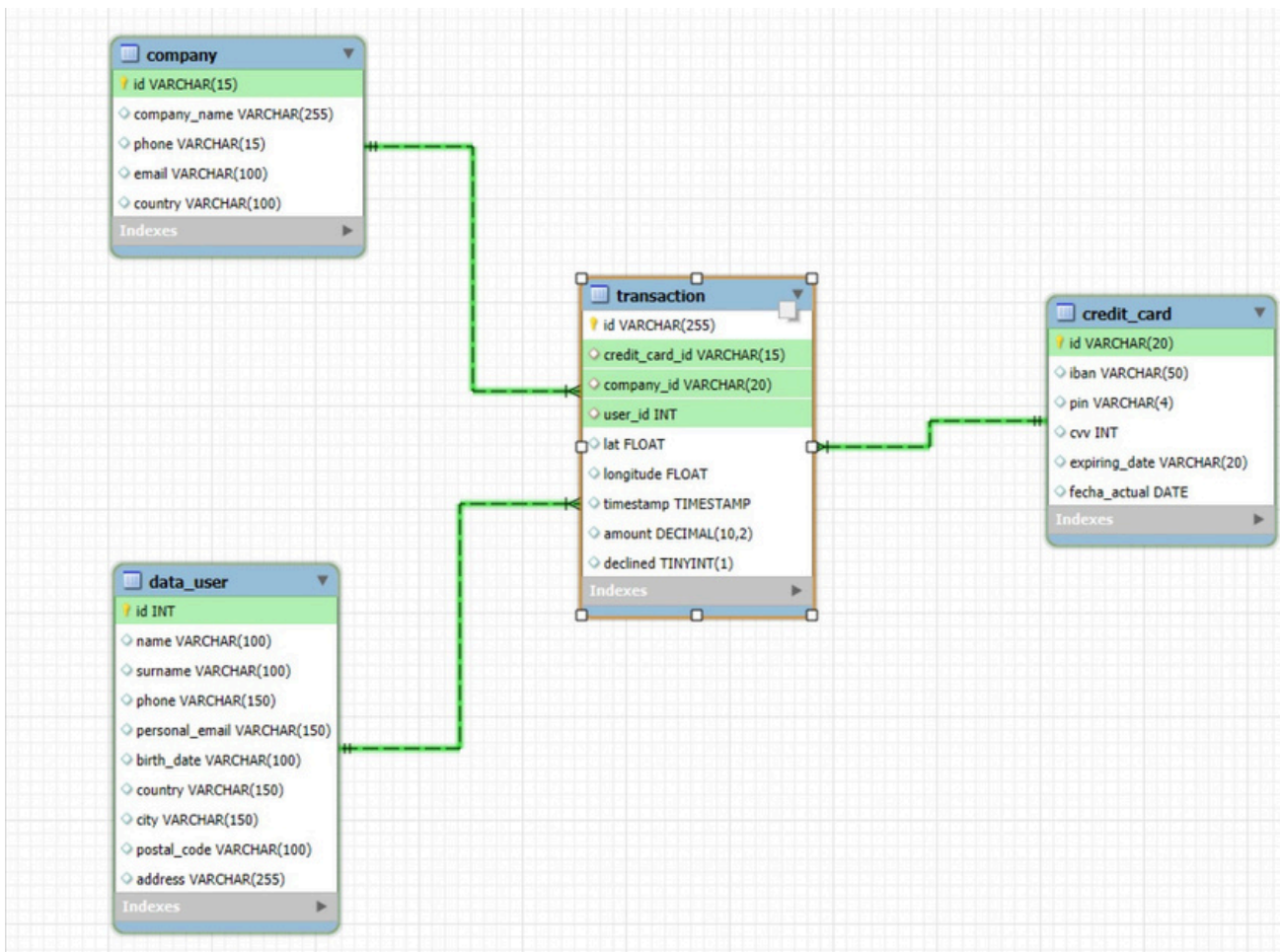
```
769 • ALTER TABLE company DROP website;
770
```

Output

Action Output

#	Time	Action	Message
✓ 1	12:37:40	ALTER TABLE company DROP website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Al final el diagrama de las tablas queda así:



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.

Para obtener esta vista se han hecho inner join de las tablas transaction, credit_card, data_user y company.

```
779 • CREATE VIEW InformeTecnico AS
780 SELECT t.id AS Transaccion, d.name AS Nombre_usuario, d.surname AS Apellido_usuario, cr.iban AS Numero_tarjeta, c.company_name AS Nombre_company
781 FROM transaction t
782 INNER JOIN data_user d ON t.user_id = d.id
783 INNER JOIN credit_card cr ON t.credit_card_id = cr.id
784 INNER JOIN company c ON t.company_id = c.id
785 ORDER BY t.id DESC;
786
787 • SELECT *
788 FROM InformeTecnico;
789
```

Transaccion	Nombre_usuario	Apellido_usuario	Numero_tarjeta	Nombre_company
FE96CE474BD59-381C-4E18-E3CA3D44EBFF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gillam	5E2813123487163628531121	Nunc Interdum Incorporated
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-88083FB03187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-8EEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCE2AB9A-271D-2BDC-9E49-8D092A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated
FBD7E0D6-8A6B-F5BC-OCA9-EA4B8760100C	Hedwig	Gilbert	MU4132333444534342541344788855	Mauris Id Inc.
FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF58868200575771634583813	Arou LLP

InformeTecnico 23 x

Output

#	Time	Action	Message	Duration / Fets
1	19:37:35	CREATE VIEW InformeTecnico AS SELECT t.id AS Transaccion, d.name AS Nombre_usuario, d.surname AS Apellid...	0 row(s) affected	0.031 sec
2	19:37:35	SELECT * FROM InformeTecnico	587 row(s) returned	0.000 sec / 0