

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
20 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2938', 'TR301950312213576817638661', '5424465566813633', '3257', '984', '10/30/22');
21 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2945', 'D026854763748537475216568689', '5142423821948828', '9080', '887', '08/24/23');
22 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2952', 'B645IQL52710525608255', '4556 453 55 5287', '4598', '438', '06/29/21');
23 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2959', 'CR7242477244335841535', '372461377349375', '3583', '667', '02/24/23');
24 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2966', 'B672LKTQ15627628377363', '448566 886747 7265', '4900', '130', '10/29/24');
25 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2973', 'PT87806228135092429456346', '544 58654 54343 384', '8760', '887', '01/30/25');
26 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2980', 'DE39241881883086277136', '402400 7145845969', '5075', '596', '07/24/22');
27 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2987', 'GE89681434837748781813', '3763 747687 76666', '2298', '797', '10/31/23');
28 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2994', 'BH62714428368066765294', '344283273252593', '7545', '595', '02/28/22');
29 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-3001', 'CY49087426654774581266832110', '511722 924833 2244', '9562', '867', '09/16/22');
30 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-3008', 'LU507216693616119230', '4485744464433884', '1856', '740', '04/05/25');
31 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-3015', 'PS119398216295715968342456821', '3784 662233 17389', '3246', '822', '01/31/22');
32 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-3022', 'FI1695162850556977423121857', '5164 1379 4842 3951', '5610', '342', '04/25/25');
33 INSERT INTO credit card (id, iban, pan, pin, cvv, expiring date) VALUES ('Ccu-3029', 'AZ62317413982441418123739746', '3429 279566 77631', '9708', '505', '09/02/23');

```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	22:35:32	SELECT * FROM transactions.credit_card	275 row(s) returned	0.000 sec / 0.000 sec

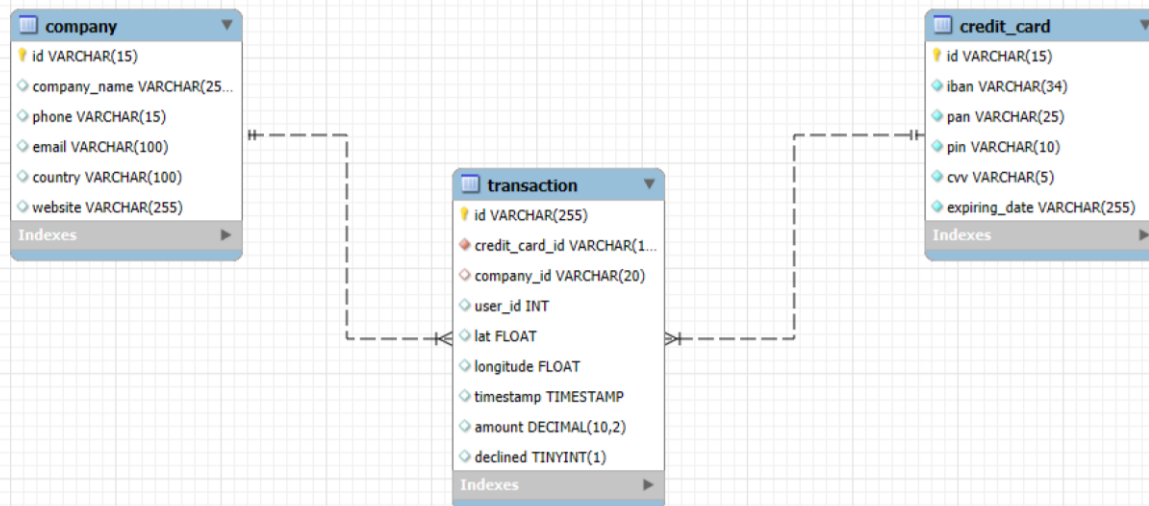
```
269 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4681', 'GE38261236401561281317', '448 57826 38974 722', '6486', '707', '04/13/22');
270 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4688', 'IS383255756743611896271082', '347635882232257', '3159', '350', '01/07/26');
271 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4695', 'GT44327784356731801238887639', '376558773427717', '8297', '734', '10/08/23');
272 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4702', 'DK7772824378841077', '348642676875332', '3880', '445', '09/19/22');
273 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4709', 'GE45504127369280160124', '453233 846716 4597', '4796', '233', '11/04/25');
274 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4716', 'PL75422719203842155307261926', '546684 833417 5543', '4252', '997', '08/14/25');
275 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4723', 'FI8248793454336573', '3414 376437 29590', '7936', '243', '07/31/20');
276 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4730', 'EE541536644818872885', '373396765877539', '5988', '988', '03/08/25');
277 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4737', 'AT278617748359359721', '3426 555216 37521', '9048', '509', '02/25/22');
278 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4744', 'AZ58751886174806034766368322', '5256255735951122', '6209', '606', '10/12/21');
279 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4751', 'PK837304693330403264694', '3436 372241 88142', '6392', '760', '06/19/23');
280 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4758', 'GB516UHV61469185263634', '378486693428441', '5241', '571', '07/27/23');
281 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4765', 'SA2888713798782221436615', '448 51353 39347 393', '1667', '153', '02/28/21');
282 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4772', 'ME59832015454148127328', '455 63485 32288 611', '1201', '552', '09/21/20');
283 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4779', 'FI9109231810971761', '513 92416 26288 645', '8788', '131', '08/18/23');
284 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4786', 'SI51703104173167515', '557 97688 75435 755', '9002', '199', '05/15/20');
285 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4793', 'HU9521562774927657356556322', '471662 767641 7624', '7216', '848', '11/09/23');
286 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4800', 'SI97824334522161436', '5455 7952 5528 3322', '3745', '806', '05/23/20');
287 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4807', 'LB19298318715580851625676971', '4539 4326 8269 4216', '8596', '626', '04/07/22');
288 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4814', 'MR4845282437847152280636374', '374471619343357', '8790', '124', '12/19/20');
289 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4821', 'LT253147505686466784', '453987 7873842836', '9000', '867', '07/15/20');
290 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4828', 'BG11ILH30149367569464', '4485252735942', '2789', '942', '09/04/22');
291 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4835', 'PT34592171131763200132583', '3723 677744 22550', '1149', '680', '01/08/24');
292 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4842', 'SA2156708581957118818229', '3774 636724 83250', '4655', '750', '11/11/24');
293 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4849', 'SE2813123487163628531121', '52233663813491514', '9992', '779', '03/21/25');
294 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4856', 'TR373872558313545667124286', '349528235713651', '9006', '974', '05/19/23');
295
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	Duration / Fetch
1	22:35:32	SELECT * FROM transactions.credit_card	275 row(s) returned	0.000 sec / 0.000 sec
2	22:40:38	SHOW CREATE TABLE credit_card	1 row(s) returned	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 departament de Recursos Humans a 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.

```

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:

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

#	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                10881D1D-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 Content:
COUNT(*)			
587			

- Antes de introducir nuevos datos en la tabla transaction, debemos tener en cuenta que:
 1. 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(15)
company_id VARCHAR(20)
user_id INT
lat FLOAT
longitude FLOAT
timestamp TIMESTAMP
amount DECIMAL(10,2)
declined TINYINT(1)
Indexes

FK
FK

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. En la tabla credit_card, teníamos que modificar que permita nulos, ya que la creamos mal en un principio, entonces lo modificamos con alter table – modify.
 4. 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

```

326
327 • INSERT INTO company (id) VALUES ('b-9999');
328
329 • ALTER TABLE credit_card MODIFY iban VARCHAR(34) DEFAULT NULL;
330 • ALTER TABLE credit_card MODIFY pan VARCHAR(25) DEFAULT NULL;
331 • ALTER TABLE credit_card MODIFY pin VARCHAR(4) DEFAULT NULL;
332 • ALTER TABLE credit_card MODIFY cvv VARCHAR(5) DEFAULT NULL;
333 • ALTER TABLE credit_card MODIFY expiring_date VARCHAR(65) DEFAULT NULL;
334
335 • INSERT INTO credit_card (id) VALUES ('CcU-9999');
336
337 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
338 values ('10881D1D-5823-A76C-55EF-C568E49A990D', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '111.11', '0');
339
340 • SELECT COUNT(*)
341 FROM transaction;
342
343

```

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

COUNT(*)
588

Result 5 x

Output

#	Time	Action	Message	Duration / Fetch
9	12:42:32	ALTER TABLE credit_card MODIFY cvv VARCHAR(5) DEFAULT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.140 sec
10	12:42:32	ALTER TABLE credit_card MODIFY expiring_date VARCHAR(65) DEFAULT NULL	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0	0.094 sec
11	12:44:02	INSERT INTO credit_card (id) VALUES ('CcU-9999')	1 row(s) affected	0.000 sec
12	12:44:13	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) values ('108...	1 row(s) affected	0.015 sec
13	12:44:26	SELECT COUNT(*) FROM transaction	1 row(s) returned	0.000 sec / 0.000 sec

- Exercici 4

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

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

340 • **SELECT** COUNT(*)

341 **FROM** transaction;

342

343 -- Exercici 4

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

345

346 • **ALTER TABLE** credit_card **DROP COLUMN** pan;

347

348 • **SHOW COLUMNS** **FROM** credit_card;

349

350

351

352

353

354

355

356

Result Grid

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	NULL	
iban	varchar(34)	YES		NULL	
pin	varchar(4)	YES		NULL	
cvv	varchar(5)	YES		NULL	
expiring_date	varchar(65)	YES		NULL	

Result 2 x

Output

Action Output

#	Time	Action	Message
1	17:18:42	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
2	17:18:47	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;**

359

360 • **DELETE FROM transaction**

361 **WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';**

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			

Result 7 

Output

 Action Output 

#	Time	Action	Message
	1 19:11:14	DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'	1 row(s) affected
	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.

Para este ejercicio tengo dos soluciones:

1. Una es creando una vista intermedia y posteriormente usarla para crear otra vista principal, que sería la que piden.

```

373 • CREATE VIEW Vista_Media_Ventas AS
374 SELECT c.company_name, round(AVG(t.amount),2) AS MediaVentas
375 FROM company c
376 INNER JOIN transaction t ON c.id = t.company_id
377 GROUP BY c.company_name
378 ORDER BY MediaVentas DESC;
379
380 • SELECT * FROM Vista_Media_Ventas;
381

```

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

	company_name	MediaVentas
▶	Eget Ipsum Ltd	473.08
	Non Magna LLC	468.35
	Sed Id Limited	461.21
	Justo Eu Arcu Ltd	443.64
	Eget Tincidunt Dui Institute	442.52
	Viverra Donec Foundation	442.28
	Vestibulum Lorem PC	434.06
	Aliquet Diam Limited	425.64

Vista_Media_Ventas 5 x

Output

Action Output

#	Time	Action	Message
▶	1 11:53:29	DROP VIEW Vista_Media_Ventas	0 row(s) affected
▶	2 11:53:42	CREATE VIEW Vista_Media_Ventas AS SELECT c.company_name, round(AVG(t.amount),2) AS MediaVentas FROM ...	0 row(s) affected
▶	3 11:53:42	SELECT * FROM Vista_Media_Ventas	101 row(s) returned

```

382 • CREATE VIEW VistaMarketing AS
383 SELECT c.company_name, c.phone, c.country, vm.MediaVentas
384 FROM company c
385 INNER JOIN Vista_Media_Ventas vm ON c.company_name = vm.company_name;
386
387 • SELECT * FROM VistaMarketing;
388

```

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

	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
	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

VistaMarketing 6 x

Output

Action Output

#	Time	Action	Message
✓	1 11:56:31	CREATE VIEW VistaMarketing AS SELECT c.company_name, c.phone, c.country, vm.MediaVentas FROM company ...	0 row(s) affected
✓	2 11:56:31	SELECT * FROM VistaMarketing	100 row(s) returned

2. La otra solución es hacer una subquery dentro de la creación de la Vista:

```

390 • CREATE VIEW VistaMarketingII AS
391 SELECT c.company_name, c.phone, c.country, Medias.MediaVentas
392 FROM company c
393 INNER JOIN (SELECT c.company_name, round(AVG(t.amount),2) AS MediaVentas
394             FROM company c
395             INNER JOIN transaction t ON c.id = t.company_id
396             GROUP BY c.company_name
397             ORDER BY MediaVentas DESC) AS Medias
398 ON c.company_name = Medias.company_name;
399
400 • SELECT * FROM VistaMarketingII;
401

```

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

	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
	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

VistaMarketingII 7 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

- 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

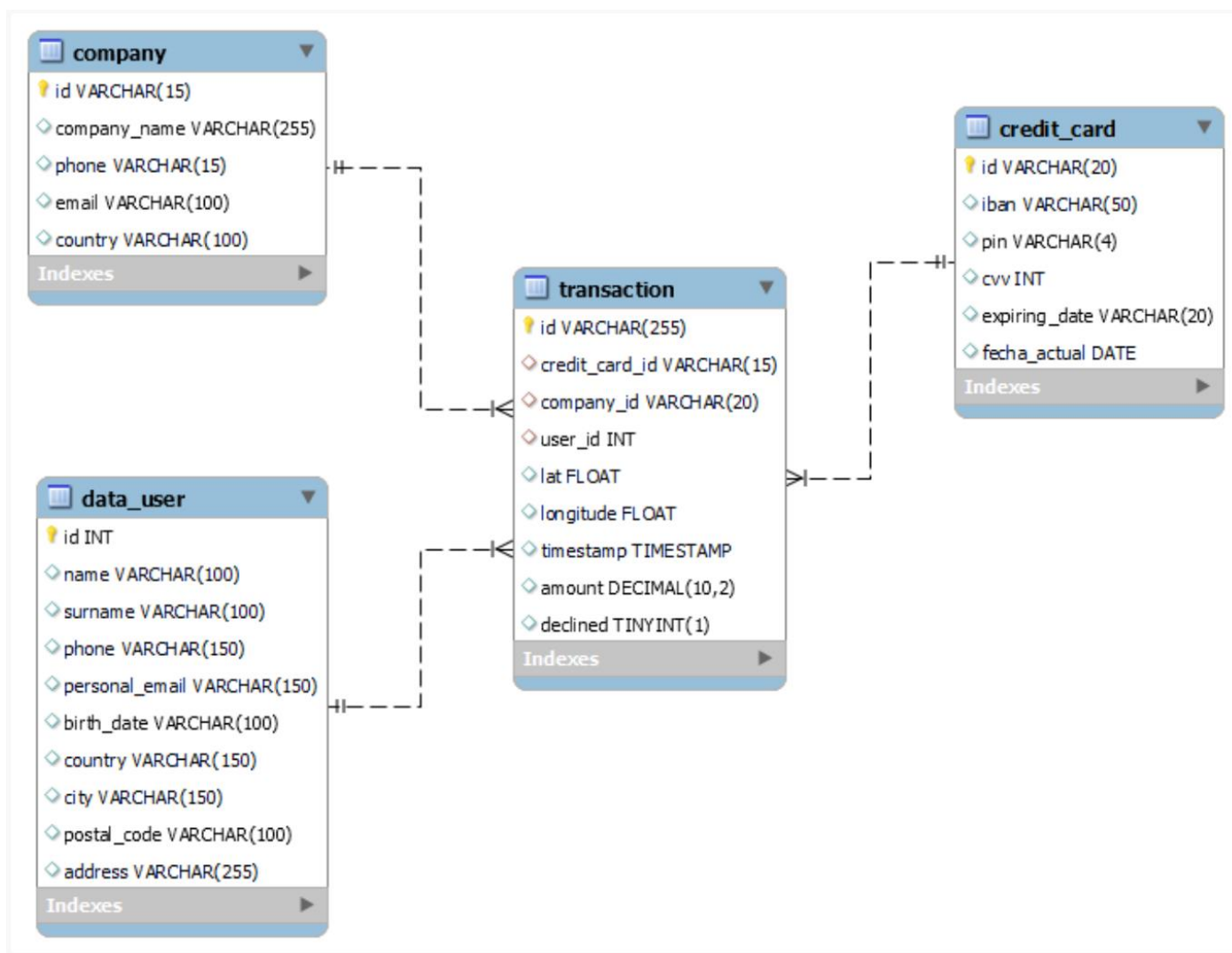
Action Output

#	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.

```

DESC credit_card;

ALTER TABLE credit_card MODIFY id VARCHAR(20);
ALTER TABLE credit_card MODIFY iban VARCHAR(50);
ALTER TABLE credit_card MODIFY cvv INT;
ALTER TABLE credit_card MODIFY expiring_date VARCHAR(65);

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

Un cambio de última hora en la columna expiring_date a varchar(20)

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

```
426 • DESC credit_card;
```

```
427
```

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

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

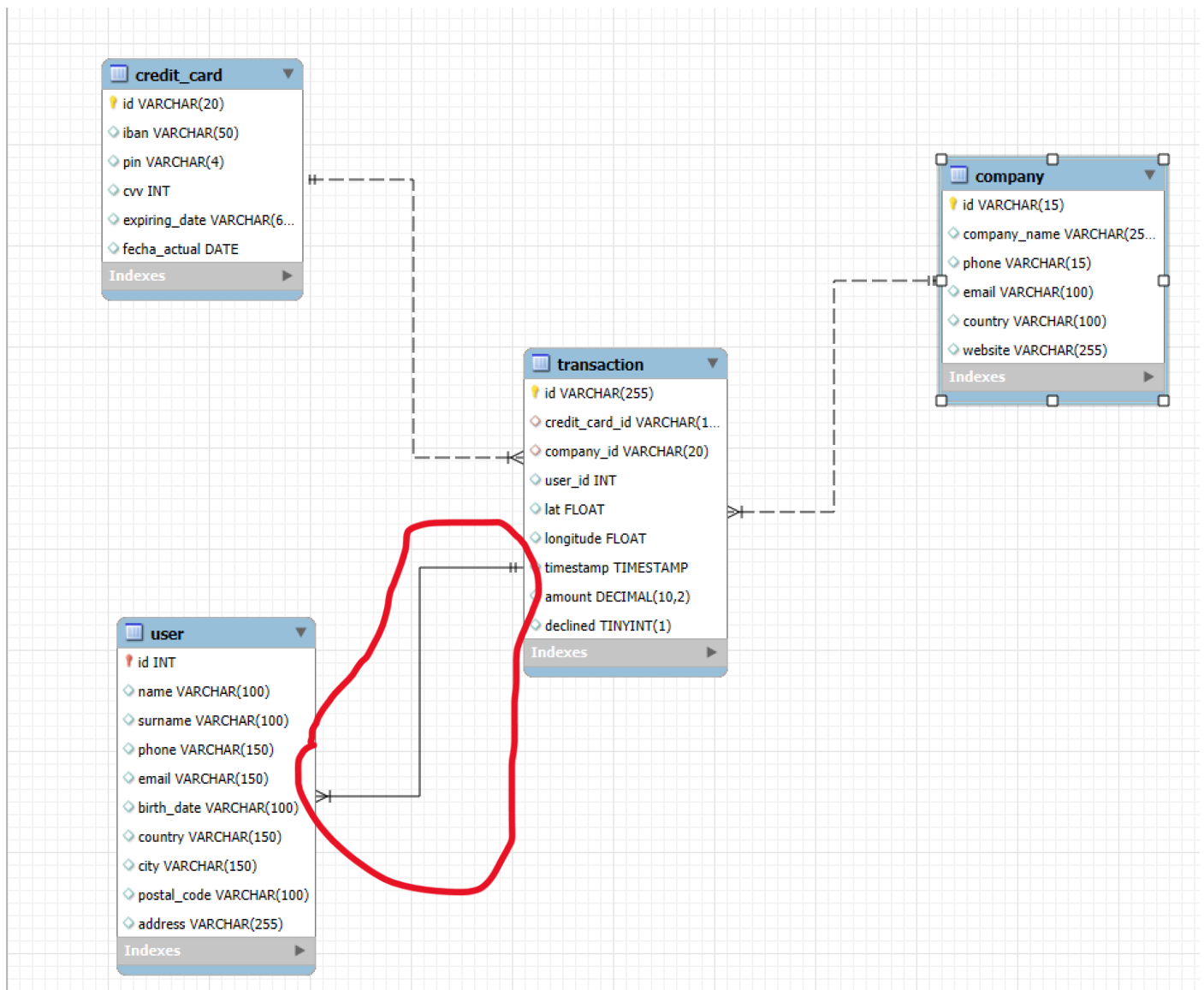

```

426 -- Creamos la tabla user
427
428 • CREATE INDEX idx_user_id ON transaction(user_id);
429
430 • CREATE TABLE IF NOT EXISTS user (
431     id INT PRIMARY KEY,
432     name VARCHAR(100),
433     surname VARCHAR(100),
434     phone VARCHAR(150),
435     email VARCHAR(150),
436     birth_date VARCHAR(100),
437     country VARCHAR(150),
438     city VARCHAR(150),
439     postal_code VARCHAR(100),
440     address VARCHAR(255),
441     FOREIGN KEY(id) REFERENCES transaction(user_id)
442
443 );
444
445
446 • SET foreign_key_checks = 0;
447
448 -- Insertamos datos de user
449 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "1", "Zeus", "Gamble", "1-282-581-0551", "interdum.enim@protonmail.",
450 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "2", "Garrett", "Mcconnell", "(718) 257-2412", "integer.vitae.nibh@
451 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "3", "Ciaran", "Harrison", "(522) 598-1365", "interdum.feugiat@aol.c
452 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "4", "Howard", "Stafford", "1-411-740-3269", "ornare.egestas@icloud.
453 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "5", "Hayfa", "Pierce", "1-554-541-2077", "et.malesuada.fames@hotma
454 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "6", "Joel", "Tyson", "(718) 288-8020", "gravida.nunc.sed@yahoo.ca".
455 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "7", "Rafael", "Jimenez", "(817) 689-0478", "eget@outlook.ca", "Dec
456 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "8", "Nissim", "Franks", "(692) 157-3469", "egestas.aliquam.fringill

```

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
275	21:48:03	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (...	1 row(s) affected	0.000 sec
276	21:48:03	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (...	1 row(s) affected	0.000 sec
277	21:48:03	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (...	1 row(s) affected	0.016 sec
278	21:48:03	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (...	1 row(s) affected	0.000 sec
279	21:48:03	SET foreign_key_checks = 1	0 row(s) affected	0.000 sec

Hasta aquí tenemos este diagrama de las relaciones de las tablas, pero la relación entre las tablas transaction y user está en el sentido contrario, cuando debería ser de user a transaction de uno a muchos, ya que cada usuario de esta tabla puede tener muchas transacciones.



Para enmendar este error, eliminaremos esta relación:

726

727 • **ALTER TABLE user**

728 **DROP CONSTRAINT user_ibfk_1;**

729

730

Y crearemos una nueva tabla llamada data_user, con los datos una subquery de los usuarios de la tabla user que están o coinciden con la tabla transaction.

De ese modo no crearemos datos null en la tabla transaction, que además no se puede introducir por que la columna transaction.id es Primary key y además no permite el autoincrement ya que es de tipo Varchar y no INT.

```

729
730 -- CREAR NUEVA TABLA DATA_USER
731 -- con DATOS DE UNA subquery de USUARIOS DE LA TABLA USER QUE ESTAN EN LA TABLA TRANSACTION
732 • CREATE TABLE data_user AS
733 SELECT *
734 FROM user u
735 WHERE u.id = any (SELECT DISTINCT u.id
736                  FROM user u
737                  LEFT JOIN transaction t ON u.id = t.user_id
738                  WHERE user_id IS NOT NULL);
739
740

```

Output			
Action Output			
#	Time	Action	Message
✓ 1	13:29:41	CREATE TABLE data_user AS SELECT * FROM user u WHERE u.id = any (SELECT DISTINCT u.id FROM user u LE...	216 row(s) affected Records: 216 Duplicates: 0 Warnings: 0

Hacemos cambio de nombre en la columna email por personal_email y verificamos las columnas que se hayan creado en la tabla data_user y vemos que hay que hacer a la columna id, primary key:

```

743 • ALTER TABLE data_user CHANGE email personal_email VARCHAR(150);
744 • DESC data_user;
745

```

Result Grid						
Filter Rows:						
Export: Wrap Cell Content: IA						
	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	
	name	varchar(100)	YES		NULL	
	surname	varchar(100)	YES		NULL	
	phone	varchar(150)	YES		NULL	
	personal_email	varchar(150)	YES		NULL	
	birth_date	varchar(100)	YES		NULL	
	country	varchar(150)	YES		NULL	
	city	varchar(150)	YES		NULL	
	postal_code	varchar(100)	YES		NULL	
	address	varchar(255)	YES		NULL	

Result 10			
Output			
Action Output			
#	Time	Action	Message
✓ 1	12:25:38	ALTER TABLE data_user CHANGE email personal_email VARCHAR(150)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 2	12:25:38	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

```

Result Grid						
		Filter Rows:			Export:	Wrap Cell Content:
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				
Output				
Action Output				
#	Time	Action	Message	
✓ 1	15:43:19	ALTER TABLE data_user ADD CONSTRAINT PRIMARY KEY(id)	0 row(s) affected Re	
✓ 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

Action 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

Action 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

Action 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:
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

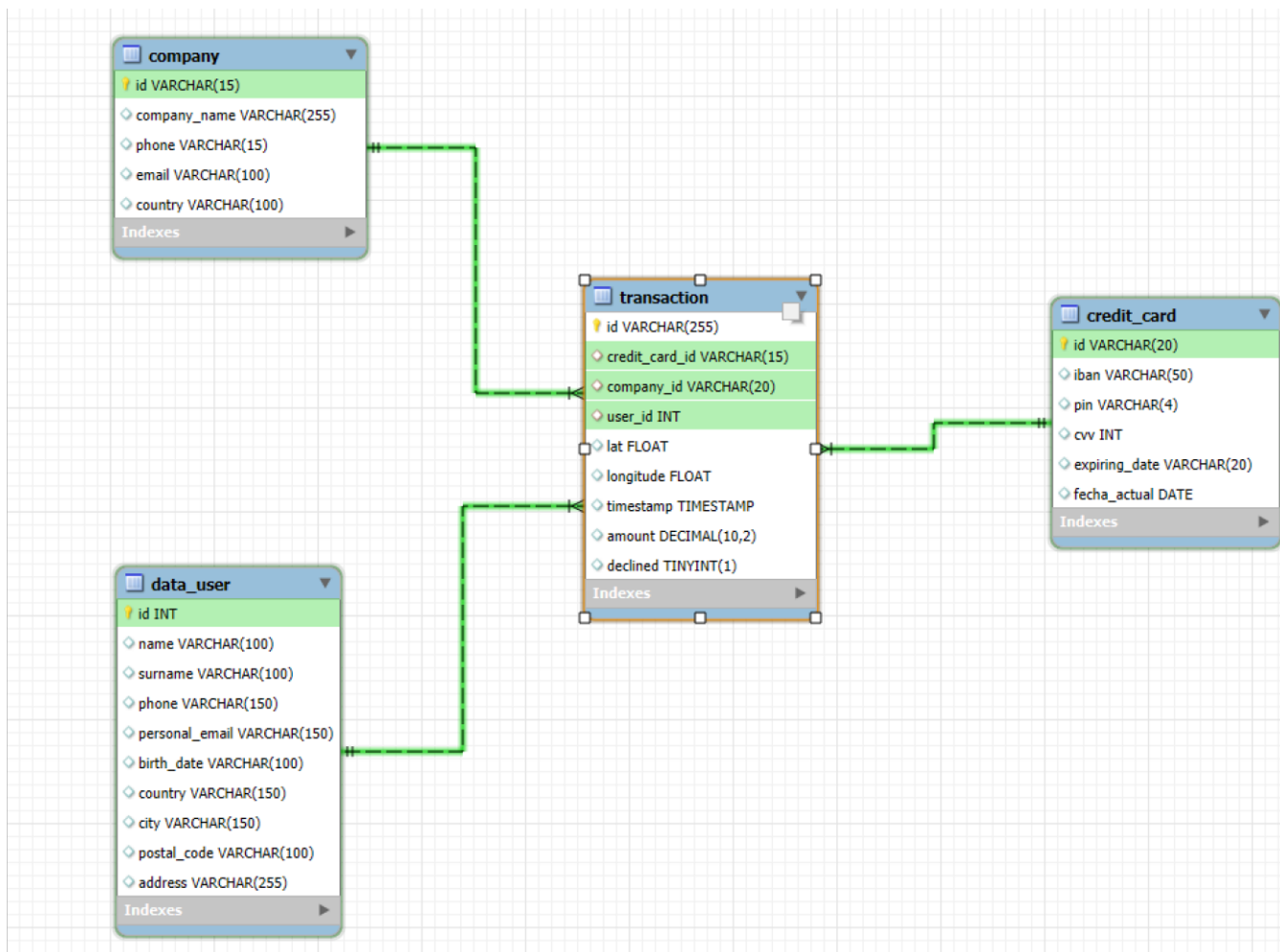
```

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:



La tabla user finalmente la eliminamos con Drop table.

765 • **DROP TABLE user;**

766

767 **Exercici 2**

Output

Action Output

#	Time	Action	Message
2	17:36:07	SELECT * FROM transactions.data_user	217 row(s) returned
3	17:36:18	SELECT * FROM transactions.transaction	587 row(s) returned
4	17:36:30	SELECT * FROM transactions.company	101 row(s) returned
5	17:36:36	SELECT * FROM transactions.credit_card	276 row(s) returned
6	17:41:20	drop table user	0 row(s) affected

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

```
// 0
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
```

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

Transaccion	Nombre_usuario	Apellido_usuario	Numero_tarjeta	Nombre_company
FE96CE47-8D59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2D86-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-88083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCE2AB9A-271D-2BDC-9E49-80D92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated
FBD7E0D6-8A6B-F8BC-OCA9-EA4B8760100C	Hedwig	Gilbert	MU4132333444534342541344788855	Mauris Id Inc.
FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF58868200575771634583813	Arcu LLP

InformeTecnico 23 x

Output

Action Output

#	Time	Action	Message	Duration / Fetc
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.