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".

```sql
CREATE TABLE IF NOT EXISTS credit_card (
    id VARCHAR(15) PRIMARY KEY,
    iban VARCHAR(35),
    pan VARCHAR(25),
    pin VARCHAR(4),
    cvv VARCHAR(3),
    expiring_date VARCHAR(8)
    );
```

Output

Action Output

#	Time	Action	Message
1	13:54:21	CREATE TABLE IF NOT EXISTS credit_card (id VARCHAR(15) PRIMARY KEY, iban VARCHAR(35), pan VARCHAR(25), pin VARCHAR(4),...	0 row(s) affected

Se ejecuta "datos_introducir_credit.sql" para insertar los datos en la tabla creada “credit_card”. Este paso la de hacerse previamente a declarar foreign key la variable *credit_card_id* de la tabla “transaction” vinculada a la primary key *id* de la tabla credit_card:

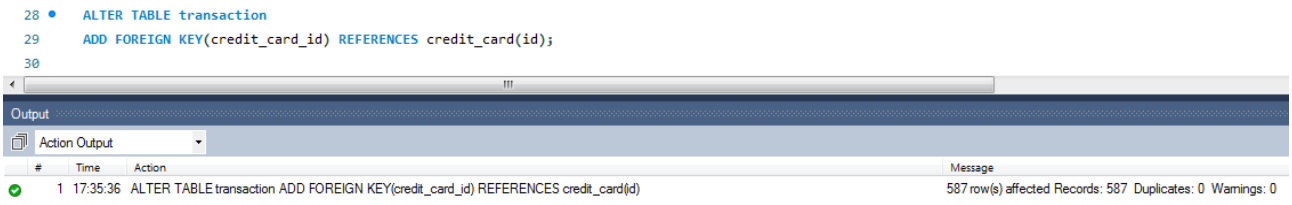
```sql
-- Insertamos datos de credit_card
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2938', 'TR301950312213576817638661', '5424465566813633', '3257', '984', '10/30/22');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2945', 'DO26854763748537475216568689', '5142423821948828', '9080', '887', '08/24/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2952', 'BG45IVQL52710525608255', '4556 453 55 5287', '4598', '438', '06/29/21');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2959', 'CR7242477244335841535', '372461377349375', '3583', '667', '02/24/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2966', 'BG72LKTQ15627628377363', '448566 886747 7265', '4900', '130', '10/29/24');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2973', 'PT878062281350924294563 46', '544 58654 54343 384', '8760', '887', '01/30/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2980', 'DE39241881883086277136', '402400 7145845969', '5075', '596', '07/24/22');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2987', 'GE89681434837748781813', '3763 747687 76666', '2298', '797', '10/31/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2994', 'BH6271442836806765294', '344283273252593', '7545', '595', '02/28/22');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3001', 'CY490874266547745812668321 10', '511722 924833 2244', '9562', '867', '09/16/22');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3008', 'LU5072166936161192 30', '4485744464433884', '1856', '740', '04/05/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3015', 'PS1193982162957159683424568 21', '3784 662233 17389', '3246', '822', '01/31/22');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3022', 'GT9169516285055697742312185 7', '5164 1379 4842 3951', '5610', '342', '04/25/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3029', 'AZ623174139824414181237397 46', '3429 279566 77631', '9708', '505', '09/02/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3036', 'AZ393360029258428658439419 94', '3768 451556 48766', '2232', '565', '10/27/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3043', 'TN6488143310514852179535', '455676 6437463635', '5969', '196', '06/07/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3050', 'FR5167744369175836831854477', '4024007123722', '4834', '126', '10/09/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3057', 'LU931822574697545215', '3484 621767 21237', '6805', '848', '09/14/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3064', 'PS1469655454492533776272731 33', '3467 732741 26810', '3865', '498', '06/03/25');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3071', 'NO8923814763512', '3464 789562 23352', '6625', '661', '12/20/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3078', 'IS0251271458846232795487 33', '4539 322 74 2377', '9405', '720', '03/08/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3085', 'BE63114723972437', '5266 3346 1135 1687', '7241', '413', '05/10/23');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3092', 'RO65LSOD1166122125447487', '3488 754223 46253', '9417', '594', '12/19/22');
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3099', 'PT2610527535682370553721 8', '448 55418 98863 789', '5612', '564', '01/22/23');
```

Output

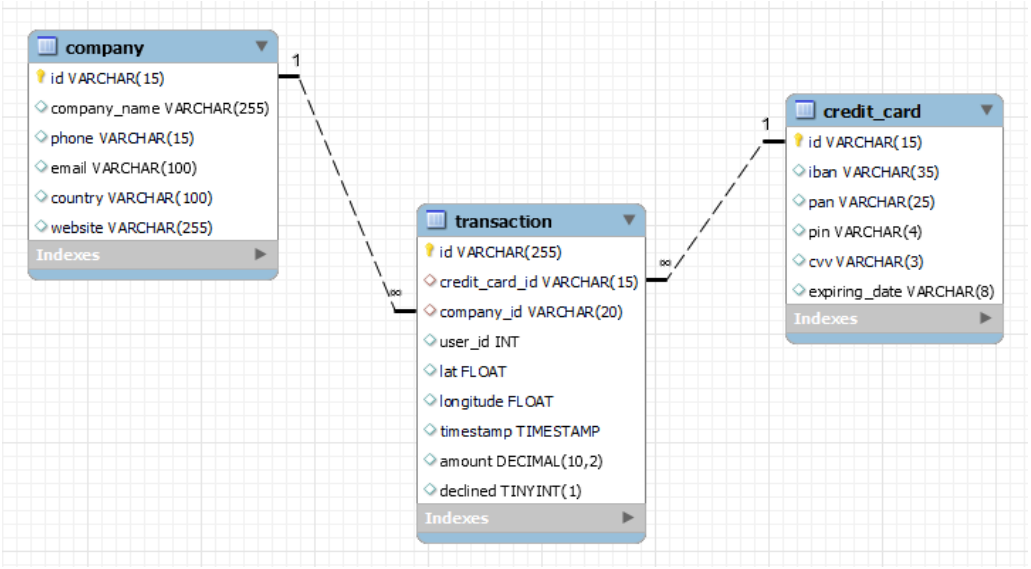
Action Output

#	Time	Action	Message
261	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4758', 'GB51GUVH61469185263634', '378486693428441', '52...	1 row(s) affected
262	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4765', 'SA2888713798782221436615', '448 51353 39347 393',...	1 row(s) affected
263	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4772', 'ME5983201545414812732 8', '455 63485 32288 611', '...	1 row(s) affected
264	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4779', 'FI9109231810971761', '513 92416 26288 645', '8788', '...	1 row(s) affected
265	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4786', 'SI5170310417317 6515', '557 97688 75435 755', '9002',...	1 row(s) affected
266	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4793', 'HU9521562774927657356555 6322', '471662 767641 7...	1 row(s) affected
267	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4800', 'SI9782433452216 1436', '5455 7952 5528 3322', '3745',...	1 row(s) affected
268	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4807', 'LB1929831871558085162567697 1', '4539 4326 8269 4...	1 row(s) affected
269	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4814', 'MR4845282437847152280636374', '37447161934335...	1 row(s) affected
270	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4821', 'LT2531475056864667 84', '453987 7873842836', '9000'...	1 row(s) affected
271	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4828', 'BG11ILMJ30149367569464', '4485252735942', '2789', '...	1 row(s) affected
272	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4835', 'PT3459217113176320013258 3', '3723 677744 22550', ...	1 row(s) affected
273	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4842', 'SA2156708581957118818229', '3774 636724 83250', '...	1 row(s) affected
274	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4849', 'SE2813123487163628531121', '5223363813491514', '...	1 row(s) affected
275	13:55:55	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4856', 'TR3738725583135456671242 86', '349528235713651',...	1 row(s) affected

Una vez introducidos los datos, se crea la relación entre las tablas “transaction” y “credit_card”, siendo la variable *credit_card_id* de la tabla “transaction” la foreign key que está vinculada a la primary key de la tabla “credit_card” que es variable *id*:



Recorda mostrar el diagrama i realitzar una breu descripció d'aquest.



La tabla de hechos es “transaction” y las tablas de dimensiones son “company” y “credit_card”. En este caso, las tablas de dimensiones ampliarían la información de algunas de sus variables. La relación de las dos tablas de dimensiones es de 1 a muchos, es decir, cada compañía puede realizar múltiples transacciones y se pueden realizar diversas transacciones con la misma tarjeta de crédito.
La tabla “transaction” contiene información sobre las transacciones realizadas (id. tarjeta credito, id. compañía, id. usuario, latitud, longitud, fecha transaccion, cantidad vendida y rechazada s/n). La primary key es *id* que es el identificador de cada transacción y tienes dos foreign key: *company_id* que es el identificador de compañía y *credit_card_id* que es el identificador de la tarjeta de crédito.

La tabla “company” contiene una ampliación de datos de las compañías (telefono, email, pais, y website). La primary key es *id* que es el identificador de cada compañía.

La tabla “credit_card” contiene una ampliación de información sobre las tarjetas de crédito utilizadas en las transacciones (iban, pan, pin, cvv, fecha caducidad). La primary key es *id* que es el identificador de cada tarjeta de crédito.

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.

```
38 •  UPDATE credit_card
39    SET iban='R323456312213576817699999'
40    WHERE id = 'CcU-2938';
41
42
43 •  SELECT id, iban
44    FROM credit_card
45    WHERE id = 'CcU-2938';
```

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

id	iban
CcU-2938	R323456312213576817699999
NULL	NULL

credit_card 1

Output

Action Output

#	Time	Action	Message
1	17:40:40	UPDATE credit_card SET iban='R323456312213576817699999' WHERE id = 'CcU-2938'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
2	17:40:40	SELECT id, iban 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

Previamente a introducir los datos del nuevo usuario en la tabla “transaction”, ha de registrarse la información vinculada a la variable *company_id* en la tabla “company” y la de la variable *credit_card_id* en la tabla “credit_card”.
Se comprueba si ya está introducida la compañía con identificador b-9999 en la tabla “company” y el identificador de la tarjeta de crédito CcU-9999 en la tabla “credit_card”.

```
55 •  SELECT *
56    FROM company
57    WHERE id = 'b-9999';
58
```

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

id	company_name	phone	email	country	website
NULL	NULL	NULL	NULL	NULL	NULL

company 5

Output

Action Output

#	Time	Action	Message
1	11:17:35	SELECT * FROM company WHERE id = 'b-9999'	0 row(s) returned

```
59 •  SELECT *
60    FROM credit_card
61    WHERE id = 'CcU-9999';
```

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

id	iban	pan	pin	cvv	expiring_date
NULL	NULL	NULL	NULL	NULL	NULL

credit_card 6

Output

Action Output

#	Time	Action	Message
1	11:26:26	SELECT * FROM credit_card WHERE id = 'CcU-9999'	0 row(s) returned

Como la información no está incluida, se introducen estos valores en las tablas “company” y “credit_card”:

```
52 •  INSERT INTO company (id)
53    VALUES ('b-9999');
54
55 •  INSERT INTO credit_card (id)
56    VALUES ('CcU-9999');
```

Output

Action Output

#	Time	Action	Message
1	17:45:00	INSERT INTO company (id) VALUES ('b-9999')	1 row(s) affected
2	17:45:00	INSERT INTO credit_card (id) VALUES ('CcU-9999')	1 row(s) affected

Ahora ya pueden introducirse los datos del nuevo usuario en la tabla “transaction”.

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

Output

Action Output

#	Time	Action	Message
1	17:46:06	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ('108B1D1D-5B23-A76C-55EF-C568E...	1 row(s) affected

Exercici 4.-

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

```
65 •  ALTER TABLE credit_card
66    DROP COLUMN pan;
67
68 •  SHOW COLUMNS
69    FROM credit_card;
70
```

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

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	NULL	
iban	varchar(35)	YES		NULL	
pin	varchar(4)	YES		NULL	
cvv	varchar(3)	YES		NULL	
expiring_date	varchar(8)	YES		NULL	

Result 2

Output

Action Output

#	Time	Action	Message
1	17:47:19	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
2	17:47:19	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.

```
78 •  DELETE FROM transaction
79    WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
80
81 •  SELECT id
82    FROM transaction
83    WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
84
85
```

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

id
NULL

transaction 3

Output

Action Output

#	Time	Action	Message
1	17:48:26	DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'	1 row(s) affected
2	17:48:26	SELECT id FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'	0 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.

La sentencia ORDER BY, tanto puede ir incluida dentro de la creación de la vista como una vez se presentan los datos.

```
95 •  CREATE VIEW VistaMarketing AS
96    SELECT company_name, phone, country, AVG(amount) AS MediaCompañia
97    FROM transaction t, company c
98    WHERE t.company_id = c.id AND t.declined = 0
99    GROUP BY company_name, phone, country;
100
101 •  SELECT *
102   FROM VistaMarketing
103   ORDER BY MediaCompañia DESC;
104
```

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

company_name	phone	country	MediaCompañia
Eget Ipsum Ltd	03 67 44 56 72	United States	481.860000
Sed Id Limited	07 28 18 18 13	United States	477.510000
Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.100000
Nunc Sit Incorporated	07 28 42 63 63	Norway	461.830000
Non Magna LLC	06 71 73 13 17	United Kingdom	458.740000

VistaMarketing 4

Output

Action Output

#	Time	Action	Message
1	17:49:37	CREATE VIEW VistaMarketing AS SELECT company_name, phone, country, AVG(amount) AS MediaCompañia FROM transaction t, company c ...	0 row(s) affected
2	17:49:37	SELECT * FROM VistaMarketing ORDER BY MediaCompañia DESC	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".

Como la sentencia ORDER BY no se incluyó en la vista VistaMarketing, en la presentación de datos no se muestran ordenados.

```
110 •   SELECT *
111     FROM VistaMarketing
112     WHERE country = 'Germany';
113
```

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

company_name	phone	country	MediaCompañia
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.570000
Convallis In Incorporated	06 66 57 29 50	Germany	60.990000
Nunc Interdum Incorporated	05 18 15 48 13	Germany	242.947692
Augue Foundation	06 88 43 15 63	Germany	15.050000
Ac Industries	09 34 65 40 60	Germany	396.150000

VistaMarketing 5

Output

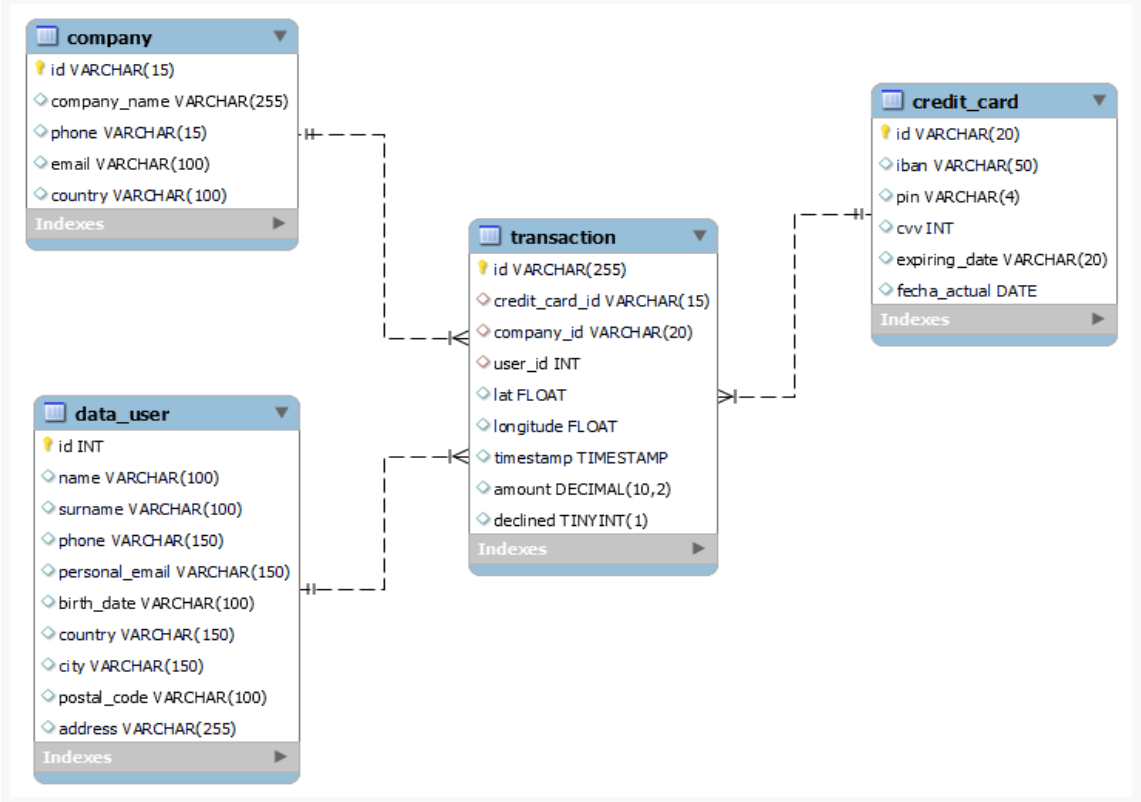
Action Output

#	Time	Action	Message
1	17:51:16	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:



Han de realizarse diversos cambios en el diagrama actual para llegar a este nuevo diagrama:

- La tabla “data_user” ha de crearse nueva puesto que no la tenemos. Será una nueva tabla de dimensiones que va a aportar más información de los datos del usuario (nombre, apellido, teléfono, mail, fecha nacimiento, país, ciudad, código postal y dirección). Al igual que las otras tablas de dimensiones, la relación con la tabla de hechos es de 1 a muchos, es decir, que un usuario puede llevar a cabo múltiples transacciones.
- La tabla “transaction” es correcta tal cual está.
- La tabla “company” tiene una variable más que es *website* que ha de eliminarse y las demás variables están ya bien.
- La tabla “credit_card” tiene la variable *id* de tipo varchar con 15 caracteres y ha de tener 20 caracteres; la variable *iban* de tipo varchar tiene 35 y ha de tener 50 caracteres; la variable *pin* no ha de modificarse; la variable *cvv* de tipo varchar ha de convertirse a tipo entero; la variable *expiring_date* de tipo varchar con 8 caracteres y ha de tener 20 caracteres; y incluir una nueva variable llamada *fecha_actual* que ha de ser de tipo fecha.

Se ejecuta "estructura_datos_user.sql" que nos viene dada.

```sql
-- Creamos la tabla user

CREATE INDEX idx_user_id ON transaction(user_id);

CREATE TABLE IF NOT EXISTS user (
        id INT PRIMARY KEY,
        name VARCHAR(100),
        surname VARCHAR(100),
        phone VARCHAR(150),
        email VARCHAR(150),
        birth_date VARCHAR(100),
        country VARCHAR(150),
        city VARCHAR(150),
        postal_code VARCHAR(100),
        address VARCHAR(255),
        FOREIGN KEY(id) REFERENCES transaction(user_id)
    );
```

```
Output
Action Output
#  Time      Action                                                                                                  Message
1  17:56:32  CREATE INDEX idx_user_id ON transaction(user_id)                                                         0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
2  17:56:33  CREATE TABLE IF NOT EXISTS user (  id INT PRIMARY KEY,  name VARCHAR(100),  surname VARCHAR(100),  phone VARC...  0 row(s) affected
```

Aunque no da error en su ejecución, contiene un error puesto que declara la variable *id* de la tabla “user” como foreign key, cuando en realidad es al revés: la variable *user_id* de la tabla “transaction” es la foreign key que ha de quedar vinculada a la primary key de la tabla “user”, que es la variable *id*.
Ahora se elimina la foreign key de la tabla “user”:

```sql
ALTER TABLE user
DROP FOREIGN KEY user_ibfk_1;
```

```
Output
Action Output
#  Time      Action                                    Message
1  16:24:19  ALTER TABLE user DROP FOREIGN KEY user_ibfk_1  0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

Y se ejecuta "datos_introducir_user(1).sql" para introducir los datos:

```
SET foreign_key_checks = 0;

-- Insertamos datos de user
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "1", "Zeus", "Gam
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "2", "Garrett", "
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "3", "Ciaran", "H
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "4", "Howard", "S
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "5", "Hayfa", "Pi
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "6", "Joel", "Tys
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "7", "Rafael", "J
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "8", "Nissim", "F
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "9", "Mannix", "M
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "10", "Robert", "
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "11", "Joan", "Ba
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "12", "Benedict",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "13", "Allegra",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "14", "Sara", "Fl
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "15", "Noelani",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "16", "Eric", "Ro
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "17", "Bruce", "G
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "18", "Russell",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "19", "Nicholas",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "20", "Kelsey", "
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "21", "Hall", "Re
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "22", "Allistair"
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "23", "Kelsie", "
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "24", "Nolan", "C
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "25", "Wanda", "C
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (        "26", "Aguila", "
```

Output

Action Output

#	Time	Action	Message
271	18:00:58	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("270", "Elton", "Roberson", ...	1 row(s) affected
272	18:00:58	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("271", "Leandra", "Cherry", ...	1 row(s) affected
273	18:00:58	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("272", "Hedwig", "Gilbert", "...	1 row(s) affected
274	18:00:58	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("273", "Hilary", "Ferguson", ...	1 row(s) affected
275	18:00:58	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("274", "Jameson", "Hunt", "...	1 row(s) affected
276	18:00:58	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("275", "Kenyon", "Hartman"...	1 row(s) affected
277	18:00:58	SET foreign_key_checks = 1	0 row(s) affected

Se comprueba si hay algún usuario de la tabla “transaction” que no está en la tabla “user” porque, de ser así, no podrían vincularse ambas tablas.

```
SELECT DISTINCT transaction.user_id
FROM transaction
LEFT JOIN user
ON transaction.user_id = user.id
WHERE user.id IS NULL;

```

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

user_id
9999

Result 6

Output

Action Output

#	Time	Action	Message
1	18:04:13	SELECT DISTINCT transaction.user_id FROM transaction LEFT JOIN user ON transaction.user_id = user.id WHERE user.id IS NULL	1 row(s) returned

El usuario ‘9999’ no está en la tabla “user”, así que hay que incluirlo:

```
INSERT INTO user (id)
VALUES ("9999");
```

Output

Action Output

#	Time	Action	Message
1	18:05:52	INSERT INTO user (id) VALUES ("9999")	1 row(s) affected

Ahora ya se puede crear la foreign key, *user_id*, de la tabla “transaction” y vincularla a la primary key, *id,* de la tabla “user”:

```
ALTER TABLE transaction
ADD FOREIGN KEY(user_id) REFERENCES user(id);
```

Output

Action Output

#	Time	Action	Message
1	18:07:05	ALTER TABLE transaction ADD FOREIGN KEY(user_id) REFERENCES user(id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

Finalmente realizamos todos los cambios en las variables de las distintas tablas para llegar al nuevo diagrama.
También son necesarios dos cambios más referidos a la tabla “user”: la variable *email* que modifica su nombre a *personal_email* y el propio nombre de la tabla que ha de llamarse “data_user”.

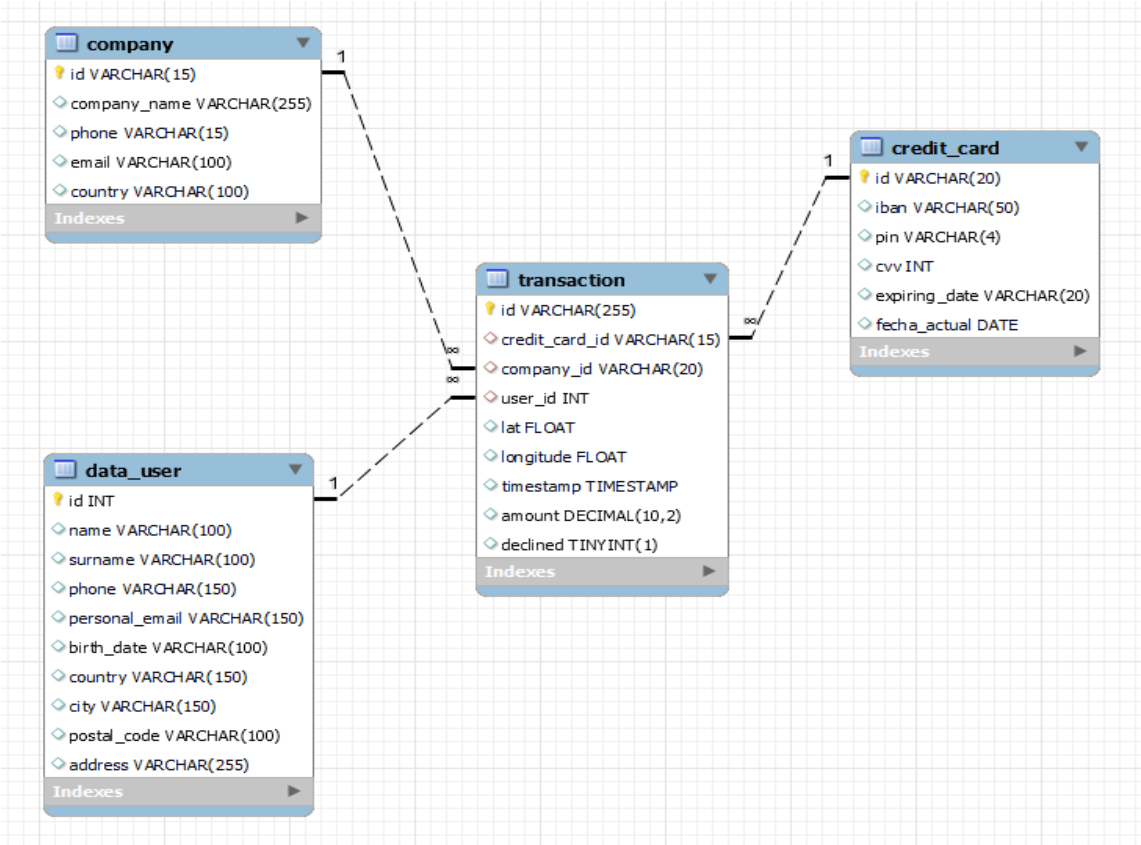
```
# Cambios de las variables de la tabla COMPANY
ALTER TABLE company
DROP COLUMN website;

# Cambios de las variables de la tabla CREDIT_CARD
ALTER TABLE credit_card
MODIFY COLUMN id VARCHAR(20);

ALTER TABLE credit_card
MODIFY COLUMN iban VARCHAR(50);

ALTER TABLE credit_card
MODIFY COLUMN cvv INT;

ALTER TABLE credit_card
MODIFY COLUMN expiring_date VARCHAR(20);

ALTER TABLE credit_card
ADD fecha_actual DATE;

# Cambios de las variables de la tabla DATA_USER
ALTER TABLE user
RENAME COLUMN email to personal_email;

RENAME TABLE user TO data_user;
```

Output

Action Output

#	Time	Action	Message
1	18:08:53	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
2	18:08:53	ALTER TABLE credit_card MODIFY COLUMN id VARCHAR(20)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3	18:08:53	ALTER TABLE credit_card MODIFY COLUMN iban VARCHAR(50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
4	18:08:53	ALTER TABLE credit_card MODIFY COLUMN cvv INT	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
5	18:08:55	ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR(20)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
6	18:08:55	ALTER TABLE credit_card ADD fecha_actual DATE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
7	18:08:55	ALTER TABLE user RENAME COLUMN email to personal_email	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
8	18:08:56	RENAME TABLE user TO data_user	0 row(s) affected

Así queda el nuevo diagrama resultante:



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 tarjeta de crèdit usada, nom de la compañía de la transacció realitzada.

Assegura't d'incloure información rellevant de les taules i utilitza àlies per a canviar de nom columnas segons sigui necessari.

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

```
214 •  CREATE VIEW InformeTecnico AS
215    SELECT t.id AS IdTransaccion, du.name AS Nombre_Usuario, du.surname AS Apellido_Usuario, cc.iban AS IBAN, c.company_name AS Nombre_Compañia
216    FROM transaction t, data_user du, credit_card cc, company c
217    WHERE t.company_id = c.id AND t.user_id = du.id AND t.credit_card_id = cc.id;
218
219 •  SELECT *
220    FROM InformeTecnico
221    ORDER BY IdTransaccion DESC;
```

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

IdTransaccion	Nombre_Usuario	Apellido_Usuario	IBAN	Nombre_Compañia
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.

InformeTecnico 7

Output

Action Output

#	Time	Action	Message
1	18:13:00	CREATE VIEW InformeTecnico AS SELECT t.id AS IdTransaccion, du.name AS Nombre_Usuario, du.surname AS Apellido_Usuario, cc.iban AS I...	0 row(s) affected
2	18:13:00	SELECT * FROM InformeTecnico ORDER BY IdTransaccion DESC	587 row(s) returned