SPRINT 3

Nivell 1

Ejercicio 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.

Input 1: Creación esquema (advierto que expiring_date no tiene formato correcto para ser DATE y la transformo a VARCHAR para editarlo.)

```
CREATE TABLE IF NOT EXISTS credit_card(
        id VARCHAR(15) PRIMARY KEY,
        iban VARCHAR(100),
        pan  VARCHAR(50),
        pin  INT, #(10)
        cvv  INT, #(10)
        expiring_date DATE
    );

    # 1.Transformo a VARCHAR para editar la fecha para que sea compatible
    ALTER TABLE credit_card MODIFY COLUMN expiring_date varchar(10);
```

Output 1:

Input 2 ejecucion INSERT INTO

```
-- 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', 'DO28547637485374752165686689', '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', 'PT87806228135092429456346', '544 58654 54343 384', '8760', '887', '01/30/25');
```

Output 2

```
293  23:42:51  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-4821', 'LT253147505686466784', '453987 7873842836', '9000', '867', '07/15/20')  1 row(s) affected
294  23:42:51  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-4828', 'BG11ILMJ30149367569464', '4485252735942', '2789', '942', '09/04/22')  1 row(s) affected
295  23:42:51  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-4835', 'PT34592171131763200132583', '3723 677744 22550', '1149', '680', '01/0...  1 row(s) affected
296  23:42:51  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-4842', 'SA2156708581957118818229', '3774 636724 83250', '4655', '750', '11/11...  1 row(s) affected
297  23:42:51  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-4849', 'SE2813123487163628531121', '5223363813491514', '9992', '779', '03/21...  1 row(s) affected
298  23:42:51  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (  'CcU-4856', 'TR373872558313545667124286', '349528235713651', '9086', '974', '05/1...  1 row(s) affected
```

Input 3: En esta query inserto el formato de fecha que tenía mi string para que con el método STR_TO_DATE se corrija.

```
SET SQL_SAFE_UPDATES = 0;
UPDATE credit_card
SET expiring_date = STR_TO_DATE(expiring_date, '%m/%d/%y');
SET SQL_SAFE_UPDATES = 1;
```

Output 3:

```
299  23:44:15  SET SQL_SAFE_UPDATES = 0                                                                  0 row(s) affected
300  23:44:15  UPDATE credit_card    SET expiring_date = STR_TO_DATE(expiring_date, '%m/%d/%y')          275 row(s) affected Rows matched: 275  Changed: 275  Warnings: 0
301  23:44:15  SET SQL_SAFE_UPDATES = 1                                                                  0 row(s) affected
```

Input 4 Revisión de que los datos estén correctamente insertados y cambio a formato DATE.

```
#2.Despues tengo que devolverlo a DATE format
SELECT expiring_date FROM credit_card; # comprobacion de que las fechas eestan en el correcto formato para convertirse en DATE
ALTER TABLE credit_card MODIFY COLUMN expiring_date DATE; # Conversion en DATE
```

Output 4:

expiring_date
2022-10-30
2023-08-24
2021-06-29
2023-02-24

```
302  23:45:31  SELECT expiring_date FROM credit_card LIMIT 0, 50000   275 row(s) returned
303  23:45:31  ALTER TABLE credit_card MODIFY COLUMN expiring_date DATE   275 row(s) affected Records: 275  Duplicates: 0  Warnings: 0
```

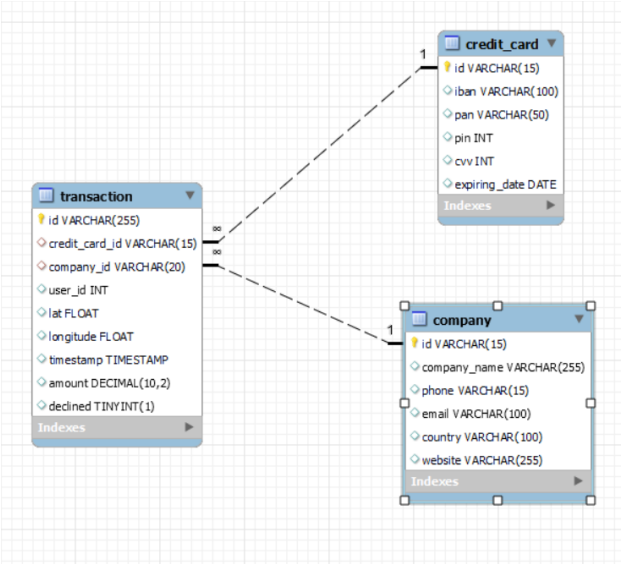
Input 5: Conecto FOREIGN KEY de transaction con primary key de credit_card.

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

Output 5:

```
999  23:50:59  ALTER TABLE transaction ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id)   587 row(s) affected Records: 587  Duplicates: 0  Warnings: 0
```

Diagrama



Ejercicio 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.

Input 1: Actualizo la cuenta iban donde el id es CcU-2938. Y hago la comprobación del cambio.

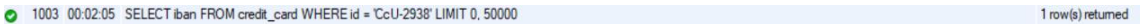
```sql
UPDATE credit_card
SET iban = 'R323456312213576817699999'
WHERE id = 'CcU-2938';

SELECT iban
FROM credit_card
WHERE id = 'CcU-2938';
```

Output 1:

```
1000  00:00:41  UPDATE credit_card SET iban = 'R323456312213576817699999' WHERE id = 'CcU-2938'   1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0
```

iban
R323456312213576817699999



Ejercicio 3

En la taula "transaction" ingressa un nou usuari amb la següent informació:

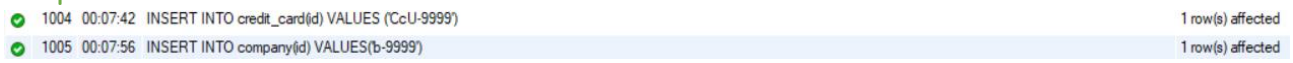
Id	108B1D1D-5B23-A76C-55EF-C568E49A99DD
credit_card_id	CcU-9999
company_id	b-9999
user_id	9999
lat	829.999
longitude	-117.999
amount	111.11
declined	0

Input 1: Para insertar los datos que quiero insertar en transaction primero tengo que crear las claves primarias a las que hace referencia en sus respectivas tablas de dimensiones. (Ahora las tablas credit_card y Company).

```
INSERT INTO credit_card(id) VALUES ('CcU-9999');
```

```
INSERT INTO company(id) VALUES('b-9999');
```

Output 1:



Input 2: Creación tabla user (Aquí me adelante creando la tabla user y más adelante la relación de esta tabla con transaction, dando por hecho que era lo esperado al ser necesario que haya coincidencia entre la clave foránea de transaction y las claves primarias de las tablas relacionadas "Esta tarea tendría que realizarse en Nivell 3.1").

```
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 2:

1006	00:17:04	CREATE INDEX idx_user_id ON transaction(user_id)		0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
1007	00:17:04	CREATE TABLE IF NOT EXISTS user (id INT PRIMARY KEY, name VARCHAR(100), surname VARCHAR(100), phone VARCHAR(150), email...		0 row(s) affected

Input 3: Inserto los datos en la tabla user

```
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", "Gamble", "1-282-581-0551", "interdum.enim@protonmail.edu", "Nov 17, 1985",  "United
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@protonmail.org", "Aug 23, 1992",
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.org", "Apr 29, 1998",  "United
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.edu", "Feb 18, 1989",  "United
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (  "5", "Havfa", "Pierce", "1-554-541-2077", "et.malesuada.fames@hotmail.org", "Sep 26, 1998",  "Unit
```

Output 3:

1281	00:21:35	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (	"273", "Hilary", "Ferguson", "060-710-1604", "sapi...	1 row(s) affected
1282	00:21:35	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (	"274", "Jameson", "Hunt", "024-732-2321", "fringill...	1 row(s) affected
1283	00:21:35	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (	"275", "Kenyon", "Hartman", "082-871-7248", "co...	1 row(s) affected
1284	00:21:35	SET foreign_key_checks = 1		0 row(s) affected

Input 4: INSERT en tabla user del nuevo registro de user(id) para que se relacione con la nueva user_id de la tabla transaction, que se insertara a continuación de ésta.

```
SET foreign_key_checks = 0;

INSERT INTO user(id) VALUES('9999');

SET foreign_key_checks = 1;
```

Output 4:

```
1286  00:24:29  SET foreign_key_checks = 0                          0 row(s) affected
1287  00:24:29  INSERT INTO user(id) VALUES('9999')                 1 row(s) affected
1288  00:24:29  SET foreign_key_checks = 1                          0 row(s) affected
```

Input 5: Inserto los datos de las distintas columnas en transaction entre ellos user_id

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

Output 5:

```
1289  00:26:21  INSERT INTO transaction(id,credit_card_id,company_id,user_id,lat,longitude,amount,declined) VALUES (  '108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', '...  1 row(s) affected
```

Input 6: Me percato que user venía con una clave foránea y quiero eliminar la foreign key para crear una relación correcta. En la cual transaction tenga relación de varios a uno y no a la inversa. Ahora busco la dirección de la FOREIGN KEY

```
SELECT CONSTRAINT_NAME
FROM information_schema.KEY_COLUMN_USAGE
WHERE TABLE_NAME = 'user'
AND COLUMN_NAME = 'id'
AND REFERENCED_TABLE_NAME = 'transaction'
AND REFERENCED_COLUMN_NAME = 'user_id';
```

Output 6:

```
1296  06:48:48  SELECT CONSTRAINT_NAME  FROM information_schema.KEY_COLUMN_USAGE  WHERE TABLE_NAME = 'user'  AND COLUMN_NAME = 'id'  AND REFERENCE...  3 row(s) returned
1297  06:50:25  SELECT CONSTRAINT_NAME  FROM information_schema.KEY_COLUMN_USAGE  WHERE TABLE_NAME = 'user'  AND COLUMN_NAME = 'id'  AND REFERENCE...  3 row(s) returned
1298  06:53:09  SELECT CONSTRAINT_NAME  FROM information_schema.KEY_COLUMN_USAGE LIMIT 0, 50000                                                          288 row(s) returned
```

Input 7: Elimino FOREIGN KEY

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

Output 7:

```
1302  07:00:27  ALTER TABLE user DROP FOREIGN KEY user_ibfk_1   0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

Input 8: Creación nueva FOREIGN KEY en transaction

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

Output 8:

```
1304  07:12:42  ALTER TABLE transaction ADD FOREIGN KEY transaction (user_id) REFERENCES user(id)   588 row(s) affected Records: 588  Duplicates: 0  Warnings: 0
```

Aquí termino toda la configuración de tabla user

Ejercicio 4

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

Input 1: Elimino la columna pan de la tabla credit_card.

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

Output 1:

```
1290  00:28:17  ALTER TABLE credit_card  DROP COLUMN pan   0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

Input 2: Testeo

```
SELECT * FROM credit_card
```

Output 2:

id	iban	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	3257	984	2022-10-30
CcU-2945	DO26854763748537475216568689	9080	887	2023-08-24
CcU-2952	BG45IVQL52710525608255	4598	438	2021-06-29
CcU-2959	CR72424772443358415 35	3583	667	2023-02-24
CcU-2966	BG72LKTQ15627628377363	4900	130	2024-10-29

```
298  00:20:13  SELECT * FROM credit_card LIMIT 0, 50000   276 row(s) returned
```

Nivell 2

Ejercicio 1

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

Input 4: Elimino la transacción con el id en cuestión.

```
DELETE FROM transaction
WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

Output 4:

```
1305  07:16:02  DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'  1 row(s) affected
```

Ejercicio 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.

Input 1: Creación VistaMarketing. Esta vista organiza las empresas destacando aquellas con un promedio por compra superior.

```
CREATE VIEW VistaMarketing AS
SELECT company_name, phone, country, AVG(amount) AS promedio_compras
FROM company c
JOIN transaction t
ON c.id = t.company_id
GROUP BY company_name,phone,country
ORDER BY promedio_compras DESC;
```

Output 1:

```
1307  07:18:04  CREATE VIEW VistaMarketing AS  SELECT company_name, phone, country, AVG(amount) AS promedio_compras FROM company c JOIN transaction t ON c.id = t.co...  0 row(s) affected
```

Input 2: Testeo

```
SELECT * FROM VistaMarketing
```

Output 2:

company_name	phone	country	promedio_compras
Eget Ipsum Ltd	03 67 44 56 72	United States	473.075000
Non Magna LLC	06 71 73 13 17	United Kingdom	468.345000
Sed Id Limited	07 28 18 18 13	United States	461.210000
Justo Eu Arcu Ltd	08 42 56 71 52	Italy	443.635000
Eget Tincidunt Dui Institute	05 35 93 32 44	Netherlands	442.520000

#	Time	Action	Message
1	23:01:33	SELECT * FROM VistaMarketing LIMIT 0, 50000	101 row(s) returned

Ejercició 3

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

Input 1: Query que retorna el nombre de las compañías residentes en Alemania dentro de la VistaMarketing.

```
SELECT company_name,country FROM VistaMarketing WHERE country = 'Germany';
```

Output 1:

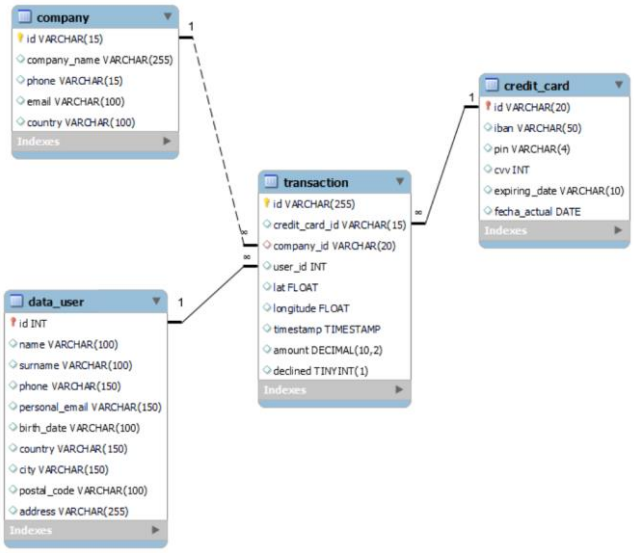
company_name	country
Aliquam PC	Germany
Ac Industries	Germany
Rutrum Non Inc.	Germany
Nunc Interdum Incorporated	Germany
Augue Foundation	Germany

2	23:03:59	SELECT company_name,country FROM VistaMarketing WHERE country = 'Germany' LIMIT 0, 50000	8 row(s) returned

Nivell 3

Ejercicio 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:



La tabla user ya está creada y vinculada a las demás tablas.

Input 1: Elimino columna website de la tabla Company, la cual estaba vacía.

```
ALTER TABLE company DROP COLUMN website;
```

Output 1:

```
1310  07:41:12  ALTER TABLE company DROP COLUMN website    0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

Input 2: cambio el nombre de la tabla user a data_user.

```
RENAME TABLE user TO data_user;
```

Output 2:

```
1312  07:47:15  RENAME TABLE user TO data_user    0 row(s) affected
```

Input 3: Añado columna fecha_actual que toma la fecha del dia en que se crea.

```
ALTER TABLE credit_card ADD COLUMN fecha_actual DATE DEFAULT(CURRENT_DATE);
```

Output 3:

```
293  00:04:03  ALTER TABLE credit_card ADD COLUMN fecha_actual DATE DEFAULT(CURRENT_DATE)    276 row(s) affected Records: 276  Duplicates: 0  Warnings: 0
```

Input 4: Cambio el formato de pin a VARCHAR(4)

```
ALTER TABLE credit_card MODIFY COLUMN pin VARCHAR(4);
```

Output 4:

Input 5: Cambio el formato del iban a VARCHAR(50)

```
ALTER TABLE credit_card MODIFY COLUMN iban VARCHAR(50);
```

Output 5:

Ejercicio 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.

Input: Creación vista InformeTecnico

```
CREATE VIEW InformeTecnico AS
SELECT t.id, u.name, u.surname, cre.iban, c.company_name
FROM transaction t
JOIN company c ON t.company_id = c.id
JOIN credit_card cre ON t.credit_card_id = cre.id
JOIN data_user u ON t.user_id = u.id;
```

Output 1:

Input 2: Testeo

```
SELECT * FROM informetecnico;
```

Output 2:

id	name	surname	iban	company_name
108B1D1D-5B23-A76C-55EF-C568E49A05DD	Kenyon	Hartman	R3234563122135768176999999	Ac Fermentum Incorporated
EA2C3281-C9C1-A387-44F8-729FB4B51C76	Kenyon	Hartman	R3234563122135768176999999	Ac Fermentum Incorporated
7DC26247-20EC-53FE-E555-B6C2E55CA5D5	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries

1315 07:56:04 SELECT * FROM informetecnico LIMIT 0, 50000 587 row(s) returned