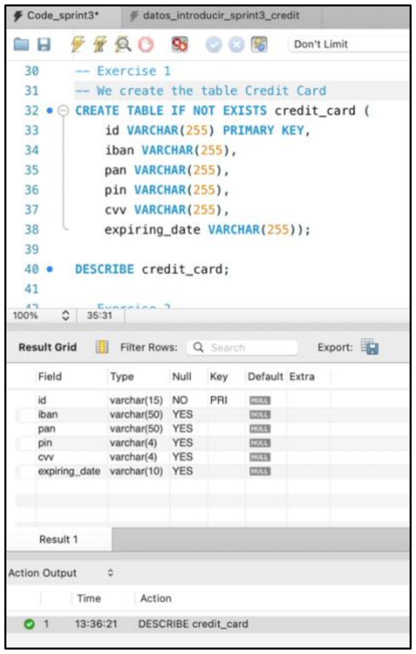
Sprint 3: Table Manipulation

LEVEL 1

- **EXERCISE 1**

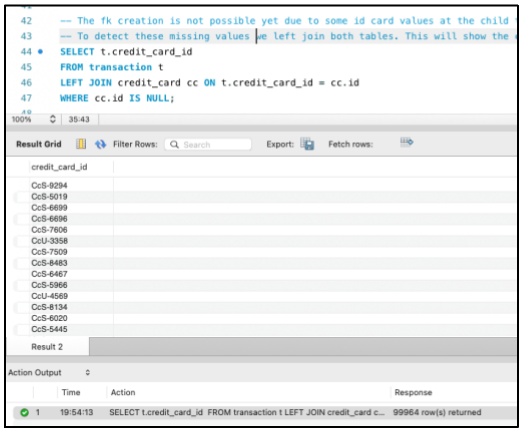


```sql
-- Exercise 1
-- We create the table Credit Card
CREATE TABLE IF NOT EXISTS credit_card (
    id VARCHAR(255) PRIMARY KEY,
    iban VARCHAR(255),
    pan VARCHAR(255),
    pin VARCHAR(255),
    cvv VARCHAR(255),
    expiring_date VARCHAR(255));

DESCRIBE credit_card;
```

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	NULL	
iban	varchar(50)	YES		NULL	
pan	varchar(50)	YES		NULL	
pin	varchar(4)	YES		NULL	
cvv	varchar(4)	YES		NULL	
expiring_date	varchar(10)	YES		NULL	

```
    Time       Action
1   13:36:21   DESCRIBE credit_card
```

- We created the **table *credit_card*** based on the column information given in the file *"dades_introduir_credit.sql".*
- The table has six columns from which "credit_card.id" is the Primary Key.

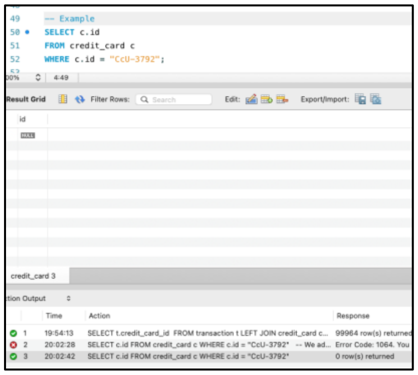


```sql
-- The fk creation is not possible yet due to some id card values at the child
-- To detect these missing values we left join both tables. This will show the
SELECT t.credit_card_id
FROM transaction t
LEFT JOIN credit_card cc ON t.credit_card_id = cc.id
WHERE cc.id IS NULL;
```

credit_card_id
CcS-9294
CcS-5019
CcS-6699
CcS-6696
CcS-7606
CcU-3358
CcS-7509
CcS-8483
CcS-6467
CcS-5966
CcU-4569
CcS-8134
CcS-6020
CcS-5445

```
    Time       Action                                                       Response
1   19:54:13   SELECT t.credit_card_id  FROM transaction t LEFT JOIN credit_card c...   99964 row(s) returned
```

- To set "credit_card.id" as a foreign key in the ***transaction*** table ("transaction.credit_card_id") first we look for null id values in ***credit_card*** that are not null in transaction.

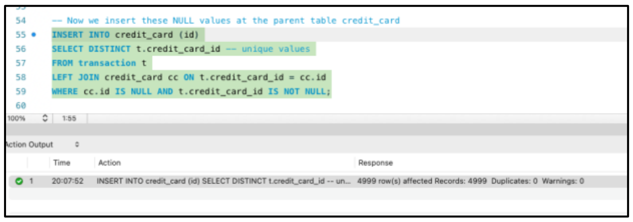


```sql
-- Example
SELECT c.id
FROM credit_card c
WHERE c.id = "CcU-3792";
```

id
NULL

```
    Time       Action                                                                  Response
1   19:54:13   SELECT t.credit_card_id  FROM transaction t LEFT JOIN credit_card c...  99964 row(s) returned
2   20:02:28   SELECT c.id FROM credit_card c WHERE c.id = "CcU-3792"  -- We ad...     Error Code: 1064. You
3   20:02:42   SELECT c.id FROM credit_card c WHERE c.id = "CcU-3792"                  0 row(s) returned
```

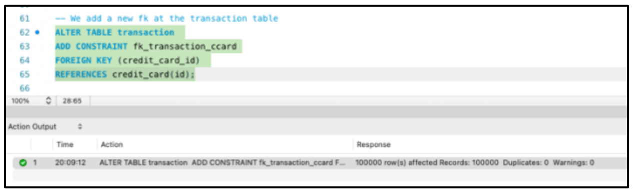
- When we test the existence of one of these null values, we confirm that is not present in ***credit_card*** table.



```
-- Now we insert these NULL values at the parent table credit_card
INSERT INTO credit_card (id)
SELECT DISTINCT t.credit_card_id -- unique values
FROM transaction t
LEFT JOIN credit_card cc ON t.credit_card_id = cc.id
WHERE cc.id IS NULL AND t.credit_card_id IS NOT NULL;
```

```
Action Output
     Time      Action                                                          Response
✓ 1  20:07:52  INSERT INTO credit_card (id) SELECT DISTINCT t.credit_card_id -- un...  4999 row(s) affected Records: 4999  Duplicates: 0  Warnings: 0
```

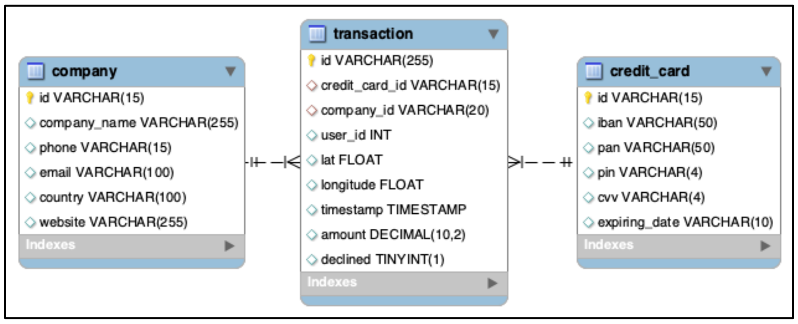
- We insert all the null id values in the ***credit_card*** table



```
-- We add a new fk at the transaction table
ALTER TABLE transaction
ADD CONSTRAINT fk_transaction_ccard
FOREIGN KEY (credit_card_id)
REFERENCES credit_card(id);
```

```
Action Output
     Time      Action                                                          Response
✓ 1  20:09:12  ALTER TABLE transaction  ADD CONSTRAINT fk_transaction_ccard F...  100000 row(s) affected Records: 100000  Duplicates: 0  Warnings: 0
```

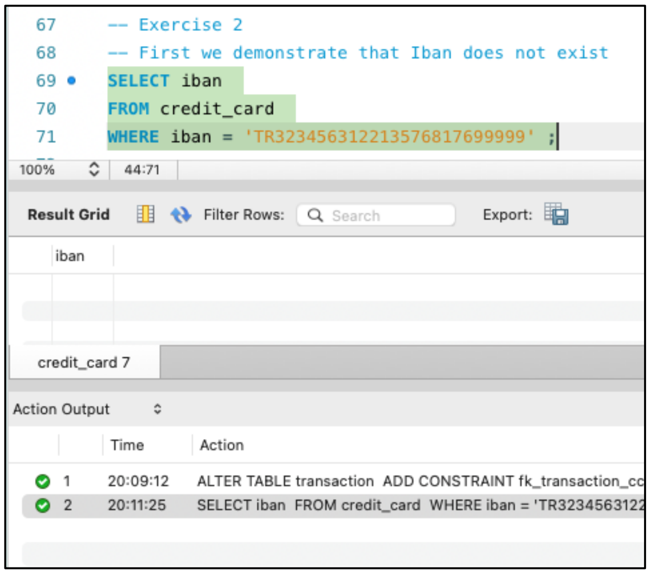
- Now we can set "credit_card_id" as a foreign key in ***transaction*** table, having ***credit_card*** as parent table.



The ER diagram shows the relationship **one to many (1:N)** between the ***credit_card*** and the ***transaction*** tables. This means that one credit card may have multiple transactions.

The table ***company*** has a similar relationship with ***transaction***: one company may have multiple transactions registered.

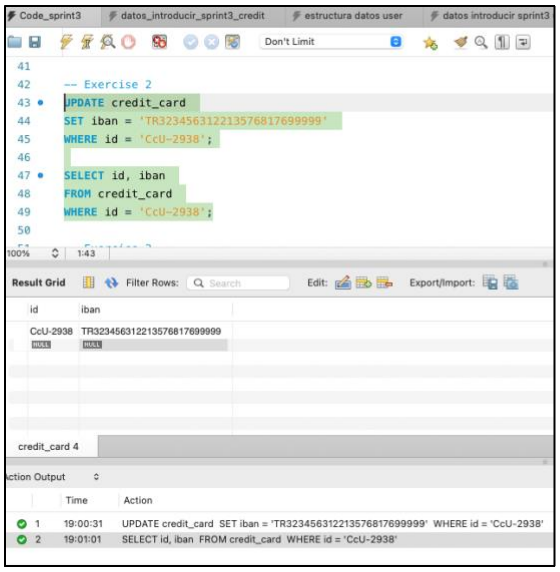
- **EXERCISE 2**



```
-- Exercise 2
-- First we demonstrate that Iban does not exist
SELECT iban
FROM credit_card
WHERE iban = 'TR32345631221357681769999' ;
```

```
Result Grid   Filter Rows:   Export:
iban

credit_card 7

Action Output
     Time      Action
✓ 1  20:09:12  ALTER TABLE transaction  ADD CONSTRAINT fk_transaction_cc
✓ 2  20:11:25  SELECT iban  FROM credit_card  WHERE iban = 'TR3234563122
```

Fig. 1 Testing the existence of iban



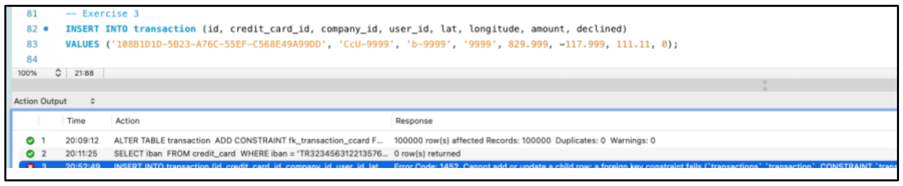
```
-- Exercise 2
UPDATE credit_card
SET iban = 'TR32345631221357681769999'
WHERE id = 'CcU-2938';

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

```
Result Grid   Filter Rows:   Edit:   Export/Import:
id        iban
CcU-2938  TR32345631221357681769999
NULL      NULL

credit_card 4

Action Output
     Time      Action
✓ 1  19:00:31  UPDATE credit_card  SET iban = 'TR32345631221357681769999'  WHERE id = 'CcU-2938'
✓ 2  19:01:01  SELECT id, iban  FROM credit_card  WHERE id = 'CcU-2938'
```

Fig. 2 Iban modification for specific credit card Id.

To modify the account number (Iban) value for the credit card id "CcU-2938" first we confirm that the new number does not exist already as a credit card "iban" (Fig. 1). Then, we use the UPDATE statement to set the new account number for the credit card id "CcU-2938" in the ***credit_card*** table and finally we confirm the modification (Fig. 2).

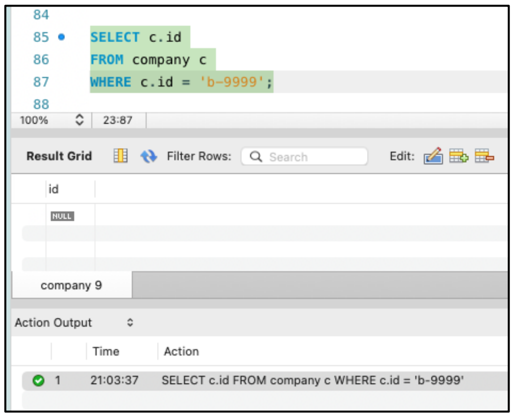
- **EXERCISE 3**



```
-- Exercise 3
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);
```

```
Action Output
   Time      Action                                                        Response
1  20:09:12  ALTER TABLE transaction  ADD CONSTRAINT fk_transaction_ccard F...  100000 row(s) affected Records: 100000  Duplicates: 0  Warnings: 0
2  20:11:25  SELECT iban  FROM credit_card  WHERE iban = 'TR323456312213576...  0 row(s) returned
3  20:52:49  INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat  Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('transactions'.'transaction', CONSTRAINT 'tran
```

When we try to insert a new value the error 1452 appears.

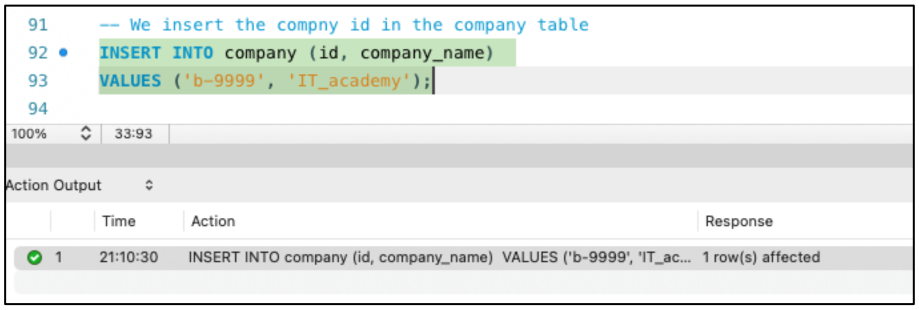


```
SELECT c.id
FROM company c
WHERE c.id = 'b-9999';
```

```
id
NULL
```

```
Action Output
   Time      Action
1  21:03:37  SELECT c.id FROM company c WHERE c.id = 'b-9999'
```

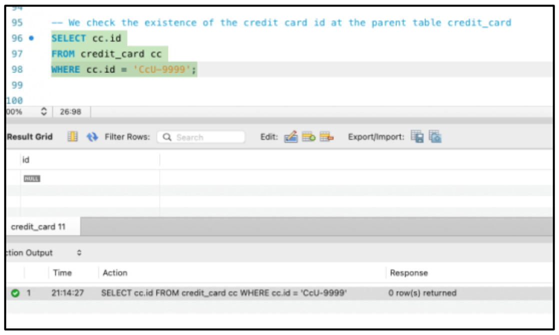
Given that the company id does not exist in the parent table ***company***, is not possible to insert a value for this company in the child table ***transaction***.



```
-- We insert the compny id in the company table
INSERT INTO company (id, company_name)
VALUES ('b-9999', 'IT_academy');
```

```
Action Output
   Time      Action                                                          Response
1  21:10:30  INSERT INTO company (id, company_name)  VALUES ('b-9999', 'IT_ac...  1 row(s) affected
```

To solve this problem, we first add the new id at the ***company*** table

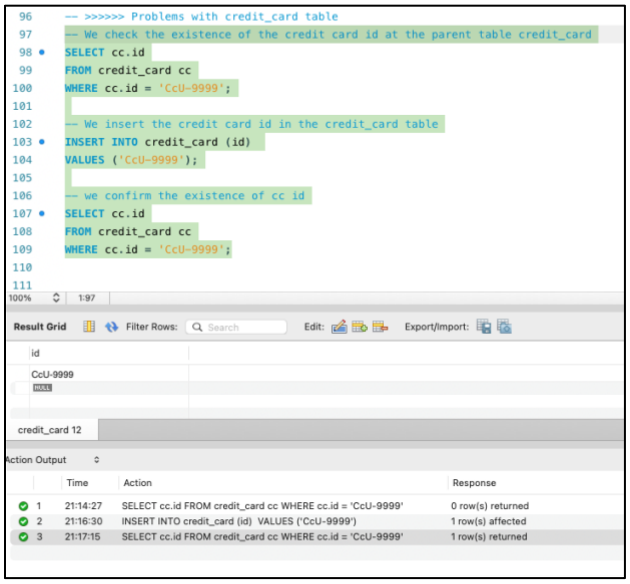


```
-- We check the existence of the credit card id at the parent table credit_card
SELECT cc.id
FROM credit_card cc
WHERE cc.id = 'CcU-9999';
```

```
id
NULL
```

```
Action Output
   Time      Action                                                  Response
1  21:14:27  SELECT cc.id FROM credit_card cc WHERE cc.id = 'CcU-9999'  0 row(s) returned
```

We also have a similar issue in the ***credit_card*** table. The id value that we want to insert does not exist in the parent table



```sql
-- >>>>>> Problems with credit_card table
-- We check the existence of the credit card id at the parent table credit_card
SELECT cc.id
FROM credit_card cc
WHERE cc.id = 'CcU-9999';

-- We insert the credit card id in the credit_card table
INSERT INTO credit_card (id)
VALUES ('CcU-9999');

-- we confirm the existence of cc id
SELECT cc.id
FROM credit_card cc
WHERE cc.id = 'CcU-9999';
```

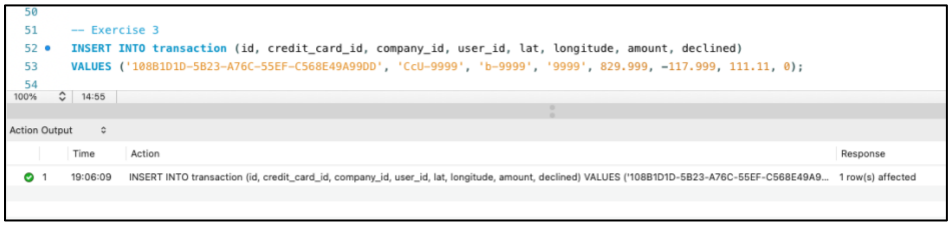
```
id
CcU-9999
NULL
```

credit_card 12

Action Output

```
   Time      Action                                                      Response
1  21:14:27  SELECT cc.id FROM credit_card cc WHERE cc.id = 'CcU-9999'   0 row(s) returned
2  21:16:30  INSERT INTO credit_card (id)  VALUES ('CcU-9999')           1 row(s) affected
3  21:17:15  SELECT cc.id FROM credit_card cc WHERE cc.id = 'CcU-9999'   1 row(s) returned
```

To solve this issue, we followed the same approach than above.
We insert the new id value at the ***credit_card*** parent table and test its existence.



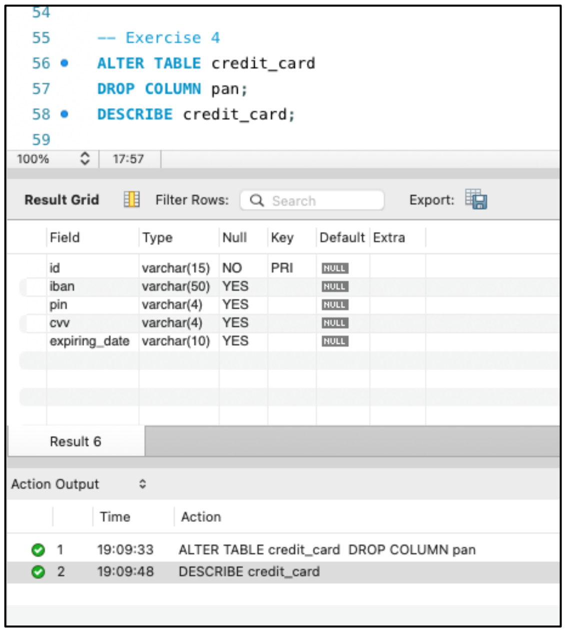
```sql
-- Exercise 3
INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', 829.999, -117.999, 111.11, 0);
```

Action Output

```
   Time      Action                                                                                                               Response
1  19:06:09  INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A9...  1 row(s) affected
```

Now, we can successfully add the new values in ***transaction*** table

- **EXERCISE 4**



```sql
-- Exercise 4
ALTER TABLE credit_card
DROP COLUMN pan;
DESCRIBE credit_card;
```

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	NULL	
iban	varchar(50)	YES		NULL	
pin	varchar(4)	YES		NULL	
cvv	varchar(4)	YES		NULL	
expiring_date	varchar(10)	YES		NULL	

Result 6

Action Output

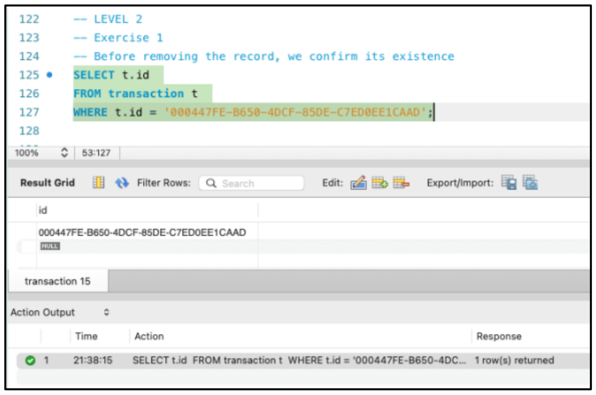
```
   Time      Action
1  19:09:33  ALTER TABLE credit_card  DROP COLUMN pan
2  19:09:48  DESCRIBE credit_card
```

Using ALTER statement we remove the column "pan" from the ***credit_card*** table

LEVEL 2

- **EXERCISE 1**

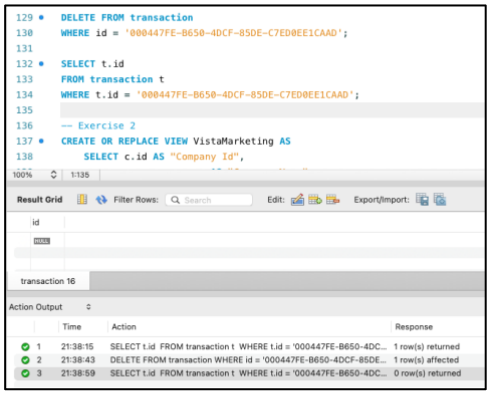
  To remove a record from the data base, first we use the id to detect its existence at the ***transaction*** table (Fig. 3), then we remove the record using the id as a condition and confirm that it does not exist anymore (Fig. 4).



```sql
-- LEVEL 2
-- Exercise 1
-- Before removing the record, we confirm its existence
SELECT t.id
FROM transaction t
WHERE t.id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
```

```
100%    53:127

Result Grid    Filter Rows: Search    Edit:    Export/Import:

id
000447FE-B650-4DCF-85DE-C7ED0EE1CAAD
NULL

transaction 15

Action Output
     Time      Action                                                        Response
✓ 1  21:38:15  SELECT t.id  FROM transaction t  WHERE t.id = '000447FE-B650-4DC...  1 row(s) returned
```

Fig. 3 We test the existence of the record using the id



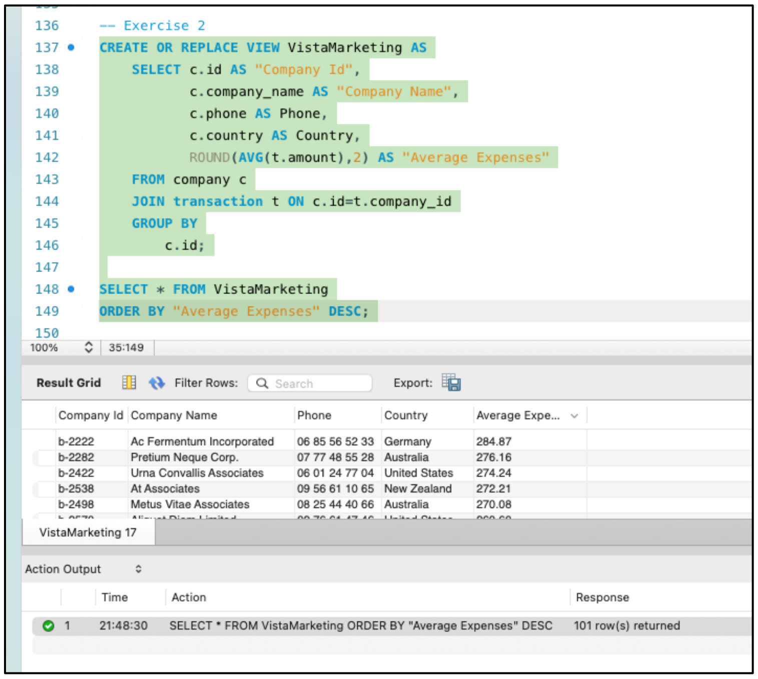
```sql
DELETE FROM transaction
WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';

SELECT t.id
FROM transaction t
WHERE t.id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';

-- Exercise 2
CREATE OR REPLACE VIEW VistaMarketing AS
    SELECT c.id AS "Company Id",
```

```
100%    1:135

Result Grid    Filter Rows: Search    Edit:    Export/Import:

id
NULL

transaction 16

Action Output
     Time      Action                                                           Response
✓ 1  21:38:15  SELECT t.id  FROM transaction t  WHERE t.id = '000447FE-B650-4DC...    1 row(s) returned
✓ 2  21:38:43  DELETE FROM transaction WHERE id = '000447FE-B650-4DCF-85DE...         1 row(s) affected
✓ 3  21:38:59  SELECT t.id  FROM transaction t  WHERE t.id = '000447FE-B650-4DC...    0 row(s) returned
```

Fig. 4 We remove the record and re confirm the action

- **EXERCISE 2**

  We create a view using the needed information



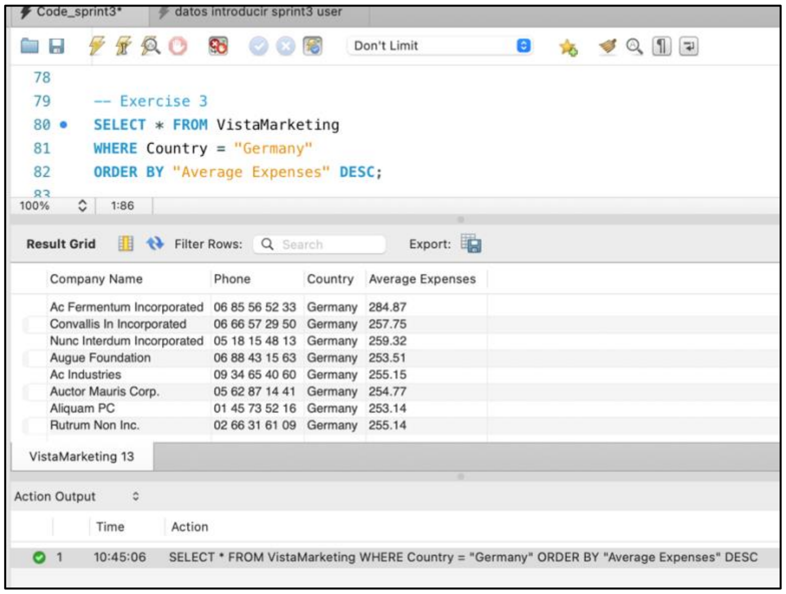
```sql
-- Exercise 2
CREATE OR REPLACE VIEW VistaMarketing AS
    SELECT c.id AS "Company Id",
           c.company_name AS "Company Name",
           c.phone AS Phone,
           c.country AS Country,
           ROUND(AVG(t.amount),2) AS "Average Expenses"
    FROM company c
    JOIN transaction t ON c.id=t.company_id
    GROUP BY
        c.id;

SELECT * FROM VistaMarketing
ORDER BY "Average Expenses" DESC;
```

```
100%    35:149

Result Grid    Filter Rows: Search    Export:
```

Company Id	Company Name	Phone	Country	Average Expe...
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.87
b-2282	Pretium Neque Corp.	07 77 48 55 28	Australia	276.16
b-2422	Urna Convallis Associates	06 01 24 77 04	United States	274.24
b-2538	At Associates	09 56 61 10 65	New Zealand	272.21
b-2498	Metus Vitae Associates	08 25 44 40 66	Australia	270.08

```
VistaMarketing 17

Action Output
     Time      Action                                                        Response
✓ 1  21:48:30  SELECT * FROM VistaMarketing ORDER BY "Average Expenses" DESC  101 row(s) returned
```

- **EXERCISE 3**



```sql

-- Exercise 3
SELECT * FROM VistaMarketing
WHERE Country = "Germany"
ORDER BY "Average Expenses" DESC;
```

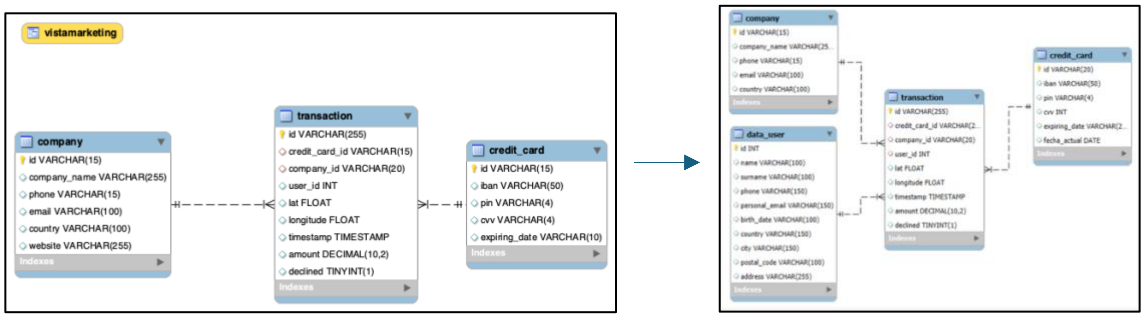
Company Name	Phone	Country	Average Expenses
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.87
Convallis In Incorporated	06 66 57 29 50	Germany	257.75
Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.32
Augue Foundation	06 88 43 15 63	Germany	253.51
Ac Industries	09 34 65 40 60	Germany	255.15
Auctor Mauris Corp.	05 62 87 14 41	Germany	254.77
Aliquam PC	01 45 73 52 16	Germany	253.14
Rutrum Non Inc.	02 66 31 61 09	Germany	255.14

	Time	Action
1	10:45:06	SELECT * FROM VistaMarketing WHERE Country = "Germany" ORDER BY "Average Expenses" DESC

LEVEL 3

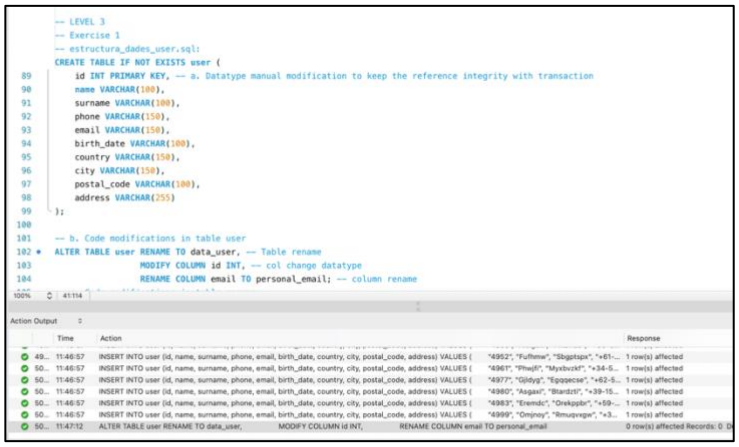
- **EXERCISE 1**

The ER diagrams below show the current and new design of the ER diagram.



To obtain the entire code based on the new ER diagram we made the following modifications:

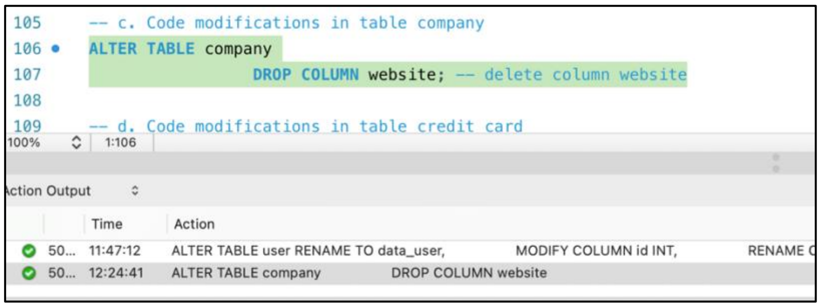
a. We create a new table ***user*** using estructura_dades_user.sql script
b. Insert the records



```
-- LEVEL 3
-- Exercise 1
-- estructura_dades_user.sql:
CREATE TABLE IF NOT EXISTS user (
    id INT PRIMARY KEY, -- a. Datatype manual modification to keep the reference integrity with transaction
    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)
);

-- b. Code modifications in table user
ALTER TABLE user RENAME TO data_user, -- Table rename
                 MODIFY COLUMN id INT, -- col change datatype
                 RENAME COLUMN email TO personal_email; -- column rename
```

```
100%   41:114

Action Output
      Time      Action                                                                                                                         Response
49... 11:46:57  INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (  "4952", "Fufhmw", "Sbgptspx", "+61-...  1 row(s) affected
50... 11:46:57  INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (  "4961", "Phwjfi", "Myxbvzkf", "+34-5...  1 row(s) affected
50... 11:46:57  INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (  "4977", "Gjldyg", "Egqqecse", "+62-5...  1 row(s) affected
50... 11:46:57  INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (  "4980", "Asgaxi", "Btardzti", "+39-15...  1 row(s) affected
50... 11:46:57  INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (  "4983", "Eremdc", "Orekppbr", "+59-...  1 row(s) affected
50... 11:46:57  INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (  "4999", "Omjnoy", "Rmuqvxgw", "+3...  1 row(s) affected
50... 11:47:12  ALTER TABLE user RENAME TO data_user,   MODIFY COLUMN id INT,   RENAME COLUMN email TO personal_email   0 row(s) affected Records: 0 D
```

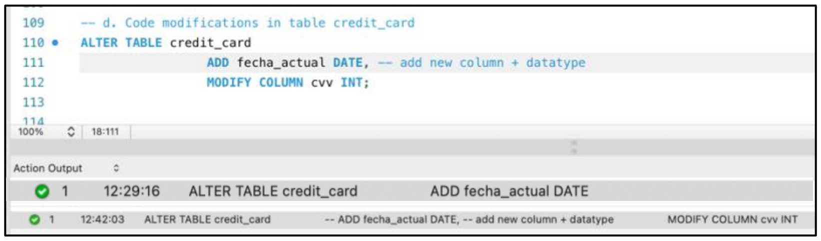
c. We change the name of the **table** *user* to *data_user*, the **datatype** from “CHAR(10)” to “INT” at the column *id* and rename **column** *email* to *personal_email* in the same table.



```
-- c. Code modifications in table company
ALTER TABLE company
                 DROP COLUMN website; -- delete column website

-- d. Code modifications in table credit card
```

```
100%   1:106

Action Output
      Time      Action
50... 11:47:12  ALTER TABLE user RENAME TO data_user,   MODIFY COLUMN id INT,   RENAME C
50... 12:24:41  ALTER TABLE company   DROP COLUMN website
```

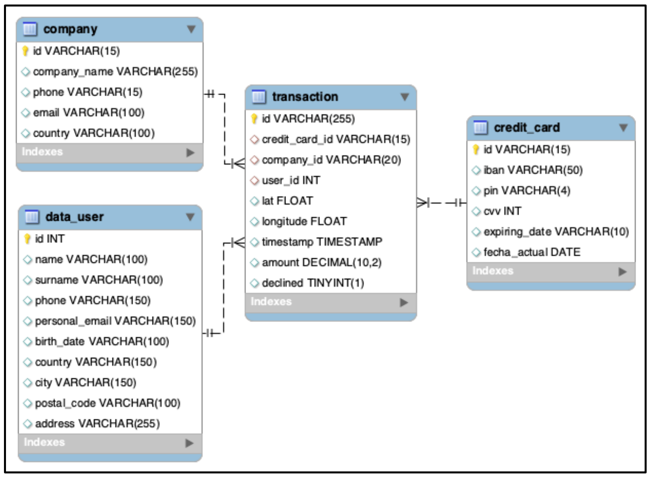
d. We remove **column** *website* from **table** *company*



```
-- d. Code modifications in table credit_card
ALTER TABLE credit_card
                 ADD fecha_actual DATE, -- add new column + datatype
                 MODIFY COLUMN cvv INT;
```

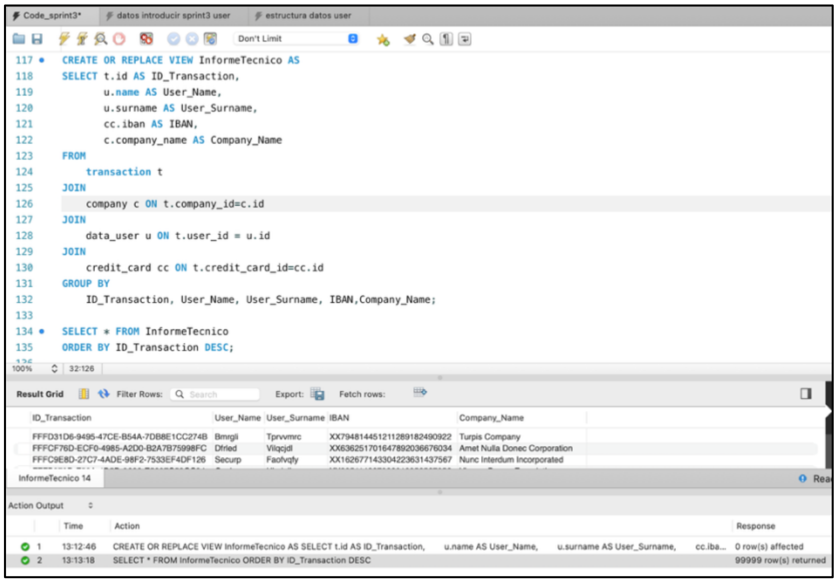
```
100%   18:111

Action Output
1  12:29:16  ALTER TABLE credit_card   ADD fecha_actual DATE
1  12:42:03  ALTER TABLE credit_card   -- ADD fecha_actual DATE, -- add new column + datatype   MODIFY COLUMN cvv INT
```

e. We add a new **column** named *fecha_actual* with DATE datatype and modify the datatype of column *cvv* from VARCHAR to INT at the *credit_card* **table**



Final ER diagram after all code modifications to match the initial ER diagram. The relationship between the ***data_user*** is also 1:N with ***transaction*** table, one user can have many different transactions.

- **EXERCISE 2**



```
CREATE OR REPLACE VIEW InformeTecnico AS
SELECT t.id AS ID_Transaction,
       u.name AS User_Name,
       u.surname AS User_Surname,
       cc.iban AS IBAN,
       c.company_name AS Company_Name
FROM
    transaction t
JOIN
    company c ON t.company_id=c.id
JOIN
    data_user u ON t.user_id = u.id
JOIN
    credit_card cc ON t.credit_card_id=cc.id
GROUP BY
    ID_Transaction, User_Name, User_Surname, IBAN,Company_Name;

SELECT * FROM InformeTecnico
ORDER BY ID_Transaction DESC;
```

ID_Transaction	User_Name	User_Surname	IBAN	Company_Name
FFFD31D6-9495-47CE-B54A-7DB8E1CC274B	Bmrgli	Tprvvmrc	XX794814451211289182490922	Turpis Company
FFFCF76D-ECF0-4985-A2D0-B2A7B75998FC	Dfrled	Vilqcjdl	XX636251701647892036676034	Amet Nulla Donec Corporation
FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	Securp	Faofvqfy	XX162677143304223631437567	Nunc Interdum Incorporated

```
Action Output
    Time      Action                                                                                                   Response
 1  13:12:46  CREATE OR REPLACE VIEW InformeTecnico AS SELECT t.id AS ID_Transaction,  u.name AS User_Name,  u.surname AS User_Surname,  cc.iba...  0 row(s) affected
 2  13:13:18  SELECT * FROM InformeTecnico ORDER BY ID_Transaction DESC                                                99999 row(s) returned
```

To create the view *"InformeTecnico"* we joined the tables ***data_user, company and credit_card*** with ***transaction*** including all asked information.