

# Sprint 3

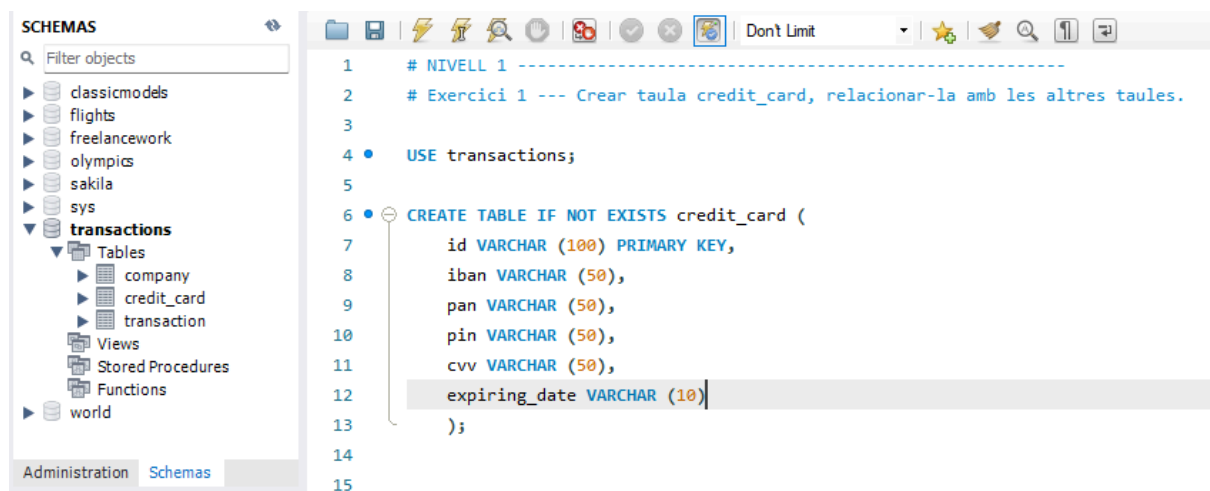
## Tasca S3.01. Manipulació de taules

### 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.

Se crea la tabla "credit\_card" con la siguiente sentencia:



The screenshot shows a MySQL IDE interface. On the left, the 'SCHEMAS' panel displays a tree view of databases, with 'transactions' expanded to show tables like 'company', 'credit\_card', and 'transaction'. The main editor on the right contains SQL code for creating the 'credit\_card' table. The code includes comments in Catalan and a 'USE transactions;' statement. The table definition specifies columns: 'id' as the primary key, 'iban', 'pan', 'pin', 'cvv', and 'expiring\_date', all as VARCHAR types with specific lengths.

```
1  # NIVELL 1 -----
2  # Exercici 1 --- Crear taula credit_card, relacionar-la amb les altres taules.
3
4  • USE transactions;
5
6  • CREATE TABLE IF NOT EXISTS credit_card (
7      id VARCHAR (100) PRIMARY KEY,
8      iban VARCHAR (50),
9      pan VARCHAR (50),
10     pin VARCHAR (50),
11     cvv VARCHAR (50),
12     expiring_date VARCHAR (10)
13 );
14
15
```

Se creó la fecha como VARCHAR y no como DATE porque el formato de fecha de los datos que debían importarse no coincidían con el formato de fecha determinado por MySQL y eso causaba error.

Se crea la FK credit\_card\_id en la tabla transaction para relacionarla con la tabla credit\_card.

The screenshot shows the SQL Developer interface with the following components:

- Navigator:** Displays the database schema structure. The 'transactions' schema is expanded, showing tables 'company', 'credit\_card', and 'transaction'. The 'credit\_card' table is selected.
- Table: credit\_card Columns:** Lists the columns: id (varchar(10) PK), iban (varchar(50)), pan (varchar(50)), pin (varchar(50)), and cvv (varchar(50)).
- SQL Editor:** Contains the following SQL script:

```
3
4 • USE transactions;
5
6 # Crear tabla
7 • CREATE TABLE IF NOT EXISTS credit_card (
8     id VARCHAR (100) PRIMARY KEY,
9     iban VARCHAR (50),
10    pan VARCHAR (50),
11    pin VARCHAR (50),
12    cvv VARCHAR (50),
13    expiring_date VARCHAR (10)
14 );
15
16 # Agregar FK a la tabla transaction para relacionarlas
17 • ALTER TABLE transaction
18   ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id)
19 ;
```
- Output:** Shows the execution results of the SQL script.

#	Time	Action	Message	Duration / Fetch
281	21:23:42	INSERT INTO credit_card (id, iban, pan, pin, cvv,...	1 row(s) affected	0.000 sec
282	21:31:16	ALTER TABLE transaction ADD FOREIGN KEY (...	587 row(s) affected Records: 587 Duplicates: 0 ...	0.141 sec

Luego se ejecuta el archivo datos\_introducir\_credit.sql para introducir los datos a la tabla

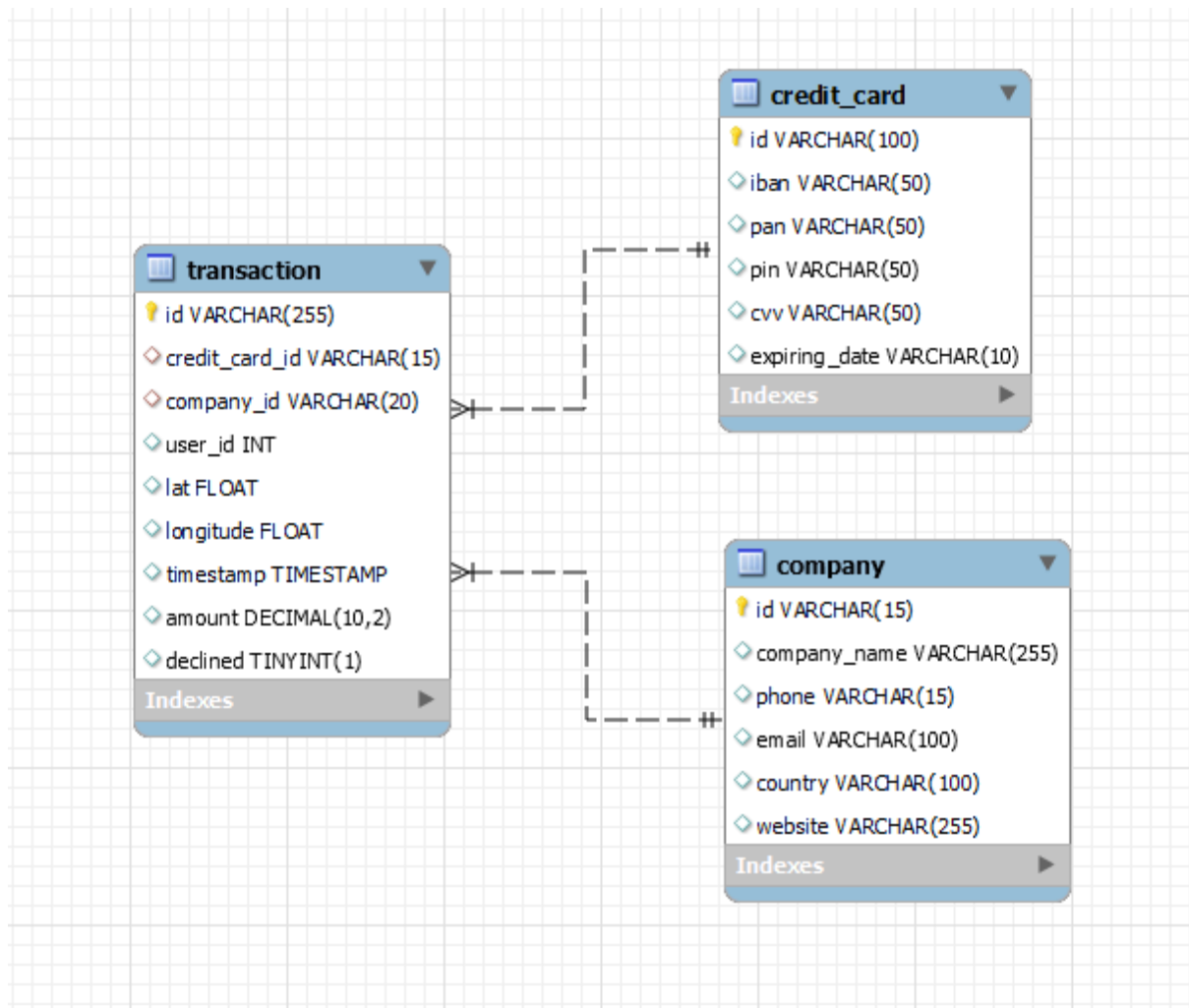
The screenshot shows the SQL Developer interface with the following components:

- SQL Editor:** Contains the following SQL script:

```
1
2 -- Insertamos datos de credit_card
3 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
4 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
5 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
6 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
7 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
8 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
9 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
10 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
11 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
12 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
13 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
14 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
15 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
16 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
17 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
18 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
```
- Output:** Shows the execution results of the script, displaying a list of inserted records with columns: id, iban, pan, pin, cvv, expiring\_date, and a message column.

id	iban	pan	pin	cvv	expiring_date	Message
CcU-2938	TR301950312213576817638661					'542446556681363
CcU-2945	D026854763748537475216568689					'5142423821948
CcU-2952	B645IVQL52710525608255					'4556 453 55 5287'
CcU-2959	CR7242477244335841535					'372461377349375', '3
CcU-2966	BG72LKTQ15627628377363					'448566 886747 7265'
CcU-2973	PT87806228135092429456346					'544 58654 54343
CcU-2980	DE39241881883086277136					'402400 7145845969'
CcU-2987	GE89681434837748781813					'3763 747687 76666'
CcU-2994	BH62714428368066765294					'344283273252593', '
CcU-3001	CY49087426654774581266832110					'511722 924833
CcU-3008	LU507216693616119230					'4485744464433884', '1
CcU-3015	PS119398216295715968342456821					'3784 662233
CcU-3022	GT91695162850556977423121857					'5164 1379 484
CcU-3029	AZ62317413982441418123739746					'3429 279566 7
CcU-3036	AZ39336002925842865843941994					'3768 451556 4
CcU-3043	TN6488143310514852179535					'455676 6437463635

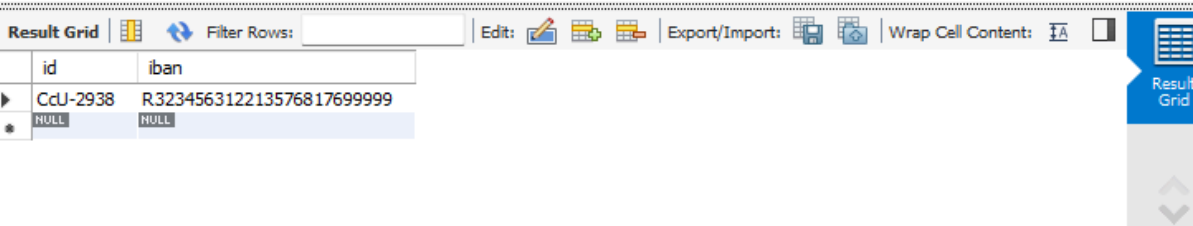
El diagrama muestra la tabla de hechos transaction con la PK id y dos FK que la relaciona con las tablas credit\_card y company en una relación de muchos a uno con el campo que es primary key en las tablas de dimensión.



## 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.

```
21      # EXERCICI 2 --- Error en el número de compte de l'usuari amb ID CcU-2938, hauria de ser:
      R323456312213576817699999
22 •    UPDATE credit_card
23      SET iban = 'R323456312213576817699999'
24      WHERE id = 'CcU-2938'
25      ;
26 •    SELECT id, iban
27      FROM credit_card
28      WHERE id = 'CcU-2938'
29      ;
```



The screenshot shows a database management tool interface. At the top, a SQL query is entered in a text area. Below the query, a 'Result Grid' is displayed, showing the results of the query. The grid has two columns: 'id' and 'iban'. The first row shows 'CcU-2938' and 'R323456312213576817699999'. The second row shows 'NULL' and 'NULL'. Below the result grid, there is a tab labeled 'credit\_card 2'. Below the tab, there is an 'Output' section. The 'Output' section shows a table with columns: '#', 'Time', 'Action', 'Message', and 'Duration / Fetch'. The table contains two rows of data, both marked with a green checkmark.

#	Time	Action	Message	Duration / Fetch
3	10:44:54	UPDATE credit_card SET iban = 'R32345631221...	1 row(s) affected Rows matched: 1 Changed: 1 ...	0.000 sec
4	10:45:00	SELECT id, iban FROM credit_card WHERE id = ...	1 row(s) returned	0.000 sec / 0.000 sec

### 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

```
30
31 # EXERCICI 3 --- Ingressar nou registre a la taula transaction
32 # Desabilitar FK constraints
33 • SET FOREIGN_KEY_CHECKS = 0;
34
35 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount,
declined)
36 VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999',
'-117.999', '111.11', '0')
37 ;
```

Output :

#	Time	Action	Message	Duration / Fetch
9	13:03:13	SET FOREIGN_KEY_CHECKS = 0	0 row(s) affected	0.000 sec
10	13:03:24	INSERT INTO transaction (id, credit_card_id, co...	1 row(s) affected	0.000 sec

```
40 # Habilitar FK constraints
41 • SET FOREIGN_KEY_CHECKS = 1;
42
43 # Verificar dades en la taula
44 • SELECT *
45 FROM transaction
46 WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD'
47 ;
48
```

[illegible]

Se agregan los valores del nuevo registro a las tablas de dimensiones para que haya coherencia en los datos.

```
49 # S'agrega el registre creat a la taula transaction a les altres taules perquè les dades siguin coherents en totes les taules
50 • INSERT INTO credit_card (id)
51 VALUES ('CcU-9999');
52
53 • INSERT INTO company (id)
54 VALUES ('b-9999');
55
56 • INSERT INTO user (id)
57 VALUES ('9999');
58
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 10	12:48:32	INSERT INTO company (id) VALUES (b-9999)	1 row(s) affected	0.015 sec
✓ 11	12:48:35	INSERT INTO user (id) VALUES (9999)	1 row(s) affected	0.015 sec

## Exercici 4

Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit\_card. Recordar mostrar el canvi realitzat.

```
49 # EXERCICI 4 --- Eliminar la columna 'pan' de la taula credit_card
50 • ALTER TABLE credit_card
51 DROP COLUMN pan
52 ;
53 • SELECT *
54 FROM credit_card
55 ;
```

Result Grid

	id	iban	pin	cvv	expiring_date
▶	CcU-2938	R323456312213576817699999	3257	984	10/30/22
	CcU-2945	DO26854763748537475216568689	9080	887	08/24/23
	CcU-2952	BG45IVQL52710525608255	4598	438	06/29/21
	CcU-2959	CR7242477244335841535	3583	667	02/24/23
	CcU-2966	BG72LKTQ15627628377363	4900	130	10/29/24

credit\_card 5 x Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 15	13:31:19	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Wami...	0.031 sec
✓ 16	13:31:34	SELECT * FROM credit_card	275 row(s) returned	0.000 sec / 0.000 sec

## Nivell 2

### Exercici 1

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

```
57 # NIVELL 2 -----
58 # EXERCICI 1 --- Elimina de la taula transaction el registre amb ID
    02C6201E-D90A-1859-B4EE-88D2986D3B02
59 • DELETE FROM transaction
60 WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02'
61 ;
```

Result Grid

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp
▶	0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:
	063FBA79-99EC-66FB-29F7-25726D1764A5	CcU-2987	b-2250	275	-81.2227	-129.05	2022-01-06 21:
	0668296C-CDB9-A883-76BC-2E4C44F8C8AE	CcU-3743	b-2618	265	-34.3593	-100.556	2022-01-26 02:
	06CD9AA5-9B42-D684-DDDD-A5E394FEBA99	CcU-2959	b-2346	92	33.7381	158.298	2021-10-26 23:

transaction 7 x Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 18	12:40:55	DELETE FROM transaction WHERE id = '02C62...	1 row(s) affected	0.000 sec






## 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.

```
74 • CREATE VIEW VistaMarketing AS
75     SELECT company.company_name AS Companyia,
76            company.phone AS Telefon,
77            company.country AS Pais,
78            ROUND(AVG(transaction.amount), 2) AS 'Mitjana compres'
79     FROM company
80     JOIN transaction
81     ON company.id = transaction.company_id
82     WHERE transaction.declined = 0
83     GROUP BY 1, 2, 3
84     ;
```

```
85 • SELECT *
86     FROM VistaMarketing
87     ORDER BY 'Mitjana compres' DESC
88     ;
```

Result Grid  Filter Rows:  Export:  Wrap Cell Content: 

	Companyia	Telefon	Pais	Mitjana compres
▶	Nunc Interdum Incorporated	05 18 15 48 13	Germany	242.95
	Amet Nulla Donec Corporation	07 15 25 14 74	Italy	92.61
	Non Institute	06 77 15 31 14	United Kingdom	297.04
	Ut Semper Foundation	01 60 36 33 06	Sweden	277.97
	Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.57
	NULL	NULL	NULL	111.11
	Lorem Eu Incorporated	01 83 66 62 07	Canada	258.64
	Enim Condimentum Ltd	09 55 51 66 25	United Kingdom	260.32
	Malesuada PC	01 74 85 68 70	Ireland	291.88
	Tortor Nunc Commodo Company	05 35 92 77 16	United States	447.11
	Arcu LLP	06 46 04 41 45	Norway	250.23

VistaMarketing 5 x

Output



Action Output

#	Time	Action	Message
✖ 16	13:00:38	SELECT * FROM VistaMarketing ORDER BY avg_amount DESC	Error Code: 1054. Unkn
✔ 17	13:01:08	SELECT * FROM VistaMarketing ORDER BY 'Mitjana compres' 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"

```
76 # EXERCICI 3 --- Filtra VistaMarketing per a mostrar companyies amb país 'Germany'
77 • SELECT *
78 FROM VistaMarketing
79 WHERE country = 'Germany'
80 ;
81
```

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

	company_name	phone	country	avg_amount
▶	Ac Fermentum Incorporated	06 85 56 52 33	Germany	206.465000
	Convallis In Incorporated	06 66 57 29 50	Germany	156.730000
	Nunc Interdum Incorporated	05 18 15 48 13	Germany	244.025238
	Augue Foundation	06 88 43 15 63	Germany	240.800000
	Ac Industries	09 34 65 40 60	Germany	289.645000

VistaMarketing 13 x Read Only

Output

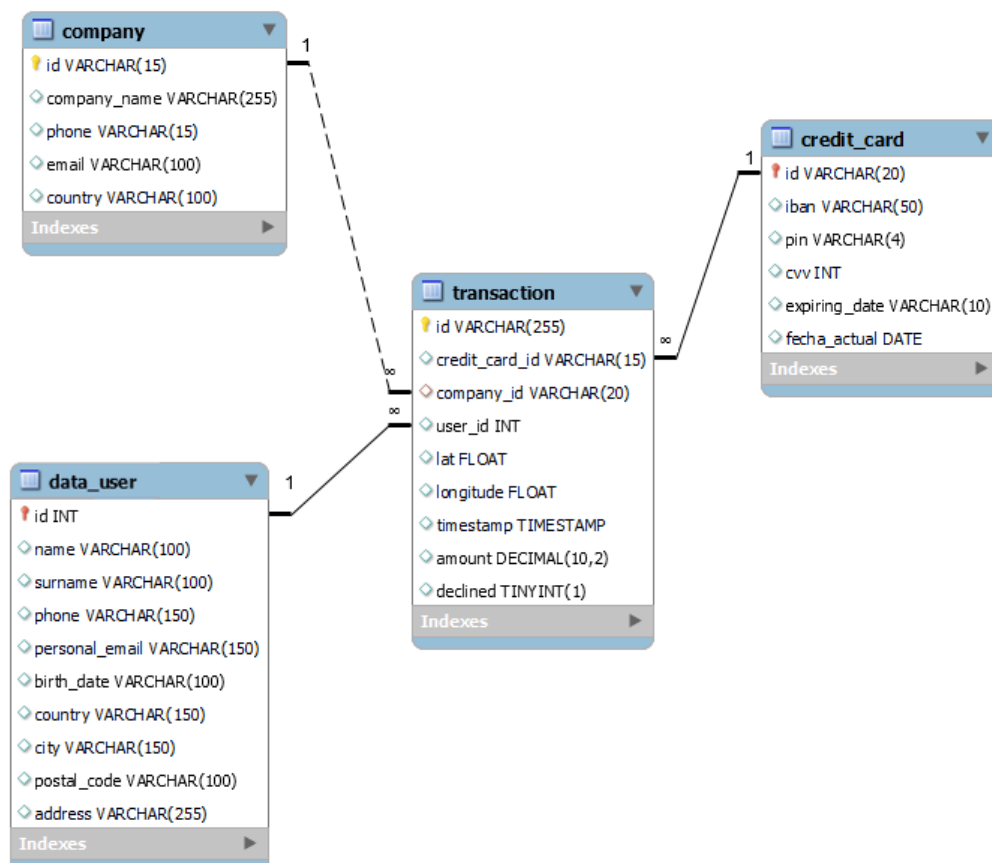
Action Output

#	Time	Action	Message	Duration / Fetch
✓ 26	14:26:10	SELECT * FROM VistaMarketing ORDER BY avg...	100 row(s) returned	0.000 sec / 0.000 sec
✓ 27	14:26:52	SELECT * FROM VistaMarketing WHERE countr...	8 row(s) returned	0.000 sec / 0.000 sec

## Nivell 3

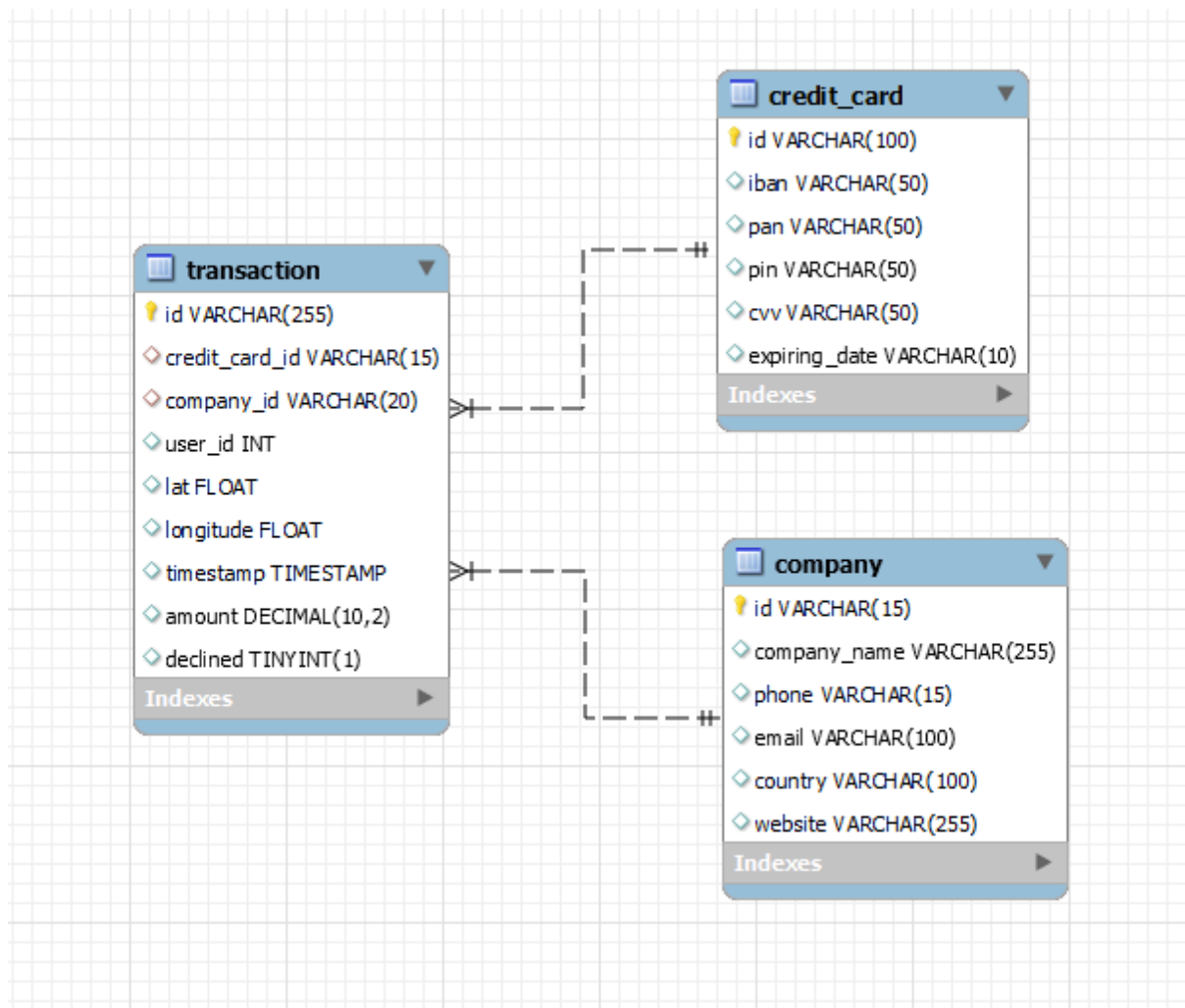
### Exercici 1

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:



En aquesta activitat, és necessari que descriguis el "pas a pas" de les tasques realitzades.

Se parte de el siguiente esquema y se realizan los cambios listados a continuación.



En la tabla `credit_card`:

1. Se cambia el tipo del campo `cvv`, de VARCHAR a INT
2. Se crea una columna: *actual\_date* (en inglés para mantener coherencia)

En la tabla `company`:

3. Se elimina el campo *website*

En la tabla `data_user`:

4. Se creó la tabla `data_user` según el archivo `estructura_datos_user.sql`, pero luego de revisar las relaciones con la tabla `transaction` se detectó que había un error en el código que creaba la foreign key y hacía que la relación entre las tablas fuera la inversa. Por esto se eliminó la tabla, se modificó el código y volvió a crearse.
5. Se cargaron los datos de la tabla desde el archivo `datos_introducir_user(1).sql`
6. Se cambió el nombre de la columna *email* a *personal\_email*.
7. Se cambió el nombre de la tabla de `user` a `data_user`

### 1. Se cambia el tipo del campo cvv, de VARCHAR a INT

Table: **credit\_card**

Columns:

id	varchar(100)
iban	PK
pin	varchar(50)
cvv	varchar(50)
expiring_date	int
actual_date	varchar(10)
	date

```
93 ;
94 • ALTER TABLE credit_card
95 MODIFY COLUMN cvv INT(3)
96 ;
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
---	------	--------	---------	------------------

### 2. Se crea una columna: *actual\_date* (en inglés para mantener coherencia). Este campo se basa en el campo *expiring\_date*, pero es de tipo DATE.

```
87 • ALTER TABLE credit_card
88 ADD COLUMN actual_date DATE
89 AS (STR_TO_DATE(expiring_date, '%m/%d/%y')) STORED
90 ;
91 • SELECT *
92 FROM credit_card;
```

Result Grid						
	id	iban	pin	cvv	expiring_date	actual_date
▶	CcU-2938	R323456312213576817699999	3257	984	10/30/22	2022-10-30
	CcU-2945	DO26854763748537475216568689	9080	887	08/24/23	2023-08-24
	CcU-2952	BG45IVQL52710525608255	4598	438	06/29/21	2021-06-29
	CcU-2959	CR7242477244335841535	3583	667	02/24/23	2023-02-24
	CcU-2966	BG72LKTQ15627628377363	4900	130	10/29/24	2024-10-29

credit\_card 14 x

Apply

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 312	11:54:41	ALTER TABLE credit_card ADD COLUMN actual...	275 row(s) affected Records: 275 Duplicates: 0 ...	0.063 sec
✓ 313	11:55:28	SELECT * FROM credit_card	275 row(s) returned	0.000 sec / 0.000 sec

### 3. Se elimina el campo *website* de la tabla *company*

```
84 • ALTER TABLE company
85 DROP COLUMN website
86 ;
87
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 309	10:15:20	SELECT * FROM transactions.user	275 row(s) returned	0.000 sec / 0.000 sec
✓ 310	10:18:38	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Wami...	0.031 sec

4. Se creó la tabla user según el archivo estructura\_datos\_user.sql, pero luego de revisar las relaciones con la tabla transaction se detectó que había un error en el código que creaba la foreign key y hacía que la relación entre las tablas fuera la inversa. Por esto se eliminó la tabla, se modificó el código y volvió a crearse. (También se cambió el nombre de la tabla y de la columna *email*, lo cual tuvo que revertirse para cargar los datos. Esta modificación se hará más adelante).

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Schemas' pane is open, showing the 'transactions' schema. The 'Tables' folder is expanded, showing 'company', 'credit\_card', 'transaction', and 'user'. The 'user' table is selected. The main pane displays the SQL script for creating the 'user' table and its foreign key relationship with the 'transaction' table.

```

1  -- Creamos la tabla user
2
3  CREATE INDEX idx_user_id ON transaction(user_id);
4
5  CREATE TABLE IF NOT EXISTS user (
6      id INT PRIMARY KEY,
7      name VARCHAR(100),
8      surname VARCHAR(100),
9      phone VARCHAR(150),
10     email VARCHAR(150),
11     birth_date VARCHAR(100),
12     country VARCHAR(150),
13     city VARCHAR(150),
14     postal_code VARCHAR(100),
15     address VARCHAR(255),
16     FOREIGN KEY(id) REFERENCES transaction(user_id)
17 );
18
19
20
21
22

```

The 'Output' pane at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
30	10:11:03	CREATE INDEX idx_user_id ON transaction(user...	0 row(s) affected Records: 0 Duplicates: 0 Warni...	0.047 sec
31	10:11:03	CREATE TABLE IF NOT EXISTS user ( id I...	0 row(s) affected	0.031 sec

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Schemas' pane is open, showing the 'transactions' schema. The 'Tables' folder is expanded, showing 'company', 'credit\_card', 'data\_user', and 'transaction'. The 'data\_user' table is selected. The main pane displays the SQL script for creating the 'data\_user' table without a foreign key relationship.

```

112 ;
113
114 # 4. Se crea la tabla data_user sin la línea de foreign key
115 CREATE INDEX idx_user_id ON transaction(user_id);
116
117 CREATE TABLE IF NOT EXISTS data_user (
118     id INT PRIMARY KEY,
119     name VARCHAR(100),
120     surname VARCHAR(100),
121     phone VARCHAR(150),
122     personal_email VARCHAR(150),
123     birth_date VARCHAR(100),
124     country VARCHAR(150),
125     city VARCHAR(150),
126     postal_code VARCHAR(100),
127     address VARCHAR(255)
128 );
129

```

The 'Output' pane at the bottom shows the execution results:

#	Time	Action	Message
1	11:27:55	CREATE TABLE IF NOT EXISTS data_user ( id INT PRIMARY KEY, name ...	0 row(s) affected

A continuación se creó la FK en la tabla transaction para relacionarla con la tabla user\_data. La tabla transaction es la tabla hija, la que contiene las foreign keys de las demás y está en el centro del diagrama de estrella.

Para crear la FK se debieron desactivar los FK checks, que vuelven a activarse tras agregar la FK.

```
130      # Se agrega la FK a la tabla transaction
131 •    SET FOREIGN_KEY_CHECKS = 0;
132
133 •    ALTER TABLE transaction
134      ADD FOREIGN KEY (user_id) REFERENCES data_user(id)
135      ;
136 •    SET FOREIGN_KEY_CHECKS = 1;
137
138
```

Output			
Action Output			
#	Time	Action	Message
✓ 2	13:43:53	ALTER TABLE transaction ADD FOREIGN KEY (user_id) REFERENCES data_user(...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 3	13:43:56	SET FOREIGN_KEY_CHECKS = 1	0 row(s) affected

## 5. Se cargan los datos de la tabla ejecutando el archivo datos\_introducir\_user(1).sql

```

1 • SET foreign_key_checks = 0;
2
3 -- Insertamos datos de user
4 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "1", "Zeus", "Gamble",
5 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "2", "Garrett", "Mcconn
6 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "3", "Ciaran", "Harriso
7 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "4", "Howard", "Staffor
8 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "5", "Hayfa", "Pierce",
9 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "6", "Joel", "Tyson", "
10 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "7", "Rafael", "Jimenez
11 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "8", "Nissim", "Franks"
12 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "9", "Mannix", "Mcclain
13 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "10", "Robert", "Mccart
14 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "11", "Joan", "Baird",
15 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "12", "Benedict", "Whee
16 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "13", "Allegra", "Stant
17 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "14", "Sara", "Flynn",
18 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "15", "Noelani", "Patri

```

Y se verifica que se hayan cargado los registros

```

1 • SELECT * FROM transactions.user;

```

	id	name	surname	phone	email	birth_date	country	city
▶	1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.edu	Nov 17, 1985	United States	Lowell
	2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Des Moi
	3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Columbi
	4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.edu	Feb 18, 1989	United States	Kailua

user 1 x Apply Revert

Output

#	Time	Action	Message	Duration / Fetch
✓ 308	10:15:06	SET foreign_key_checks = 1	0 row(s) affected	0.000 sec
✓ 309	10:15:20	SELECT * FROM transactions.user	275 row(s) returned	0.000 sec / 0.000 sec

## 6. Se actualiza el nombre de la columna personal\_email

```

140 # Se cambia el nombre de la columna email a personal_email.
141 • ALTER TABLE user
142 RENAME COLUMN email TO personal_email
143 ;

```

Output


#	Time	Action	Message
✓ 2	11:27:00	ALTER TABLE user RENAME COLUMN email TO personal_email	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 3	11:27:07	SELECT * FROM transactions.user	275 row(s) returned



7. Se cambió el nombre de la tabla de user a data\_user

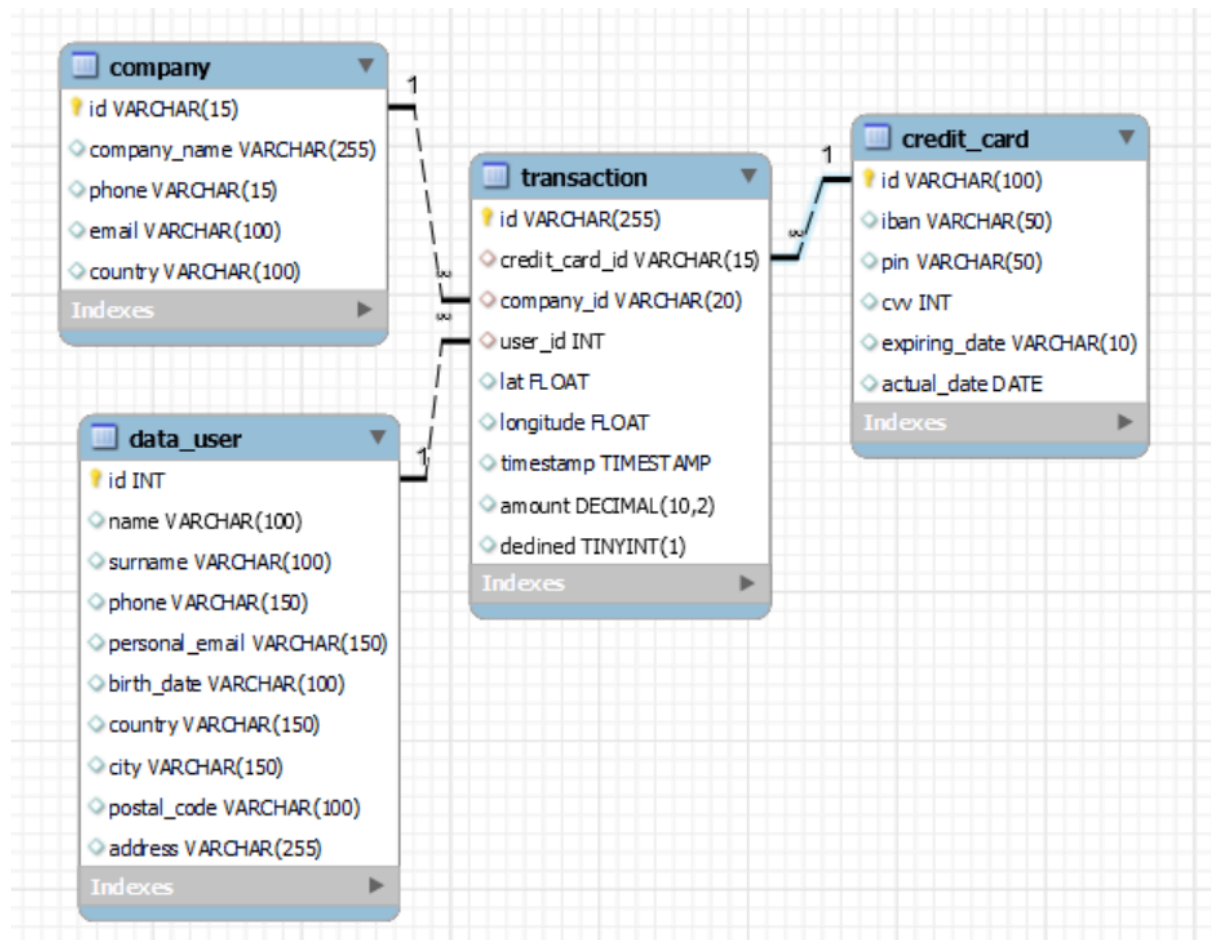
```
145      # Se cambió el nombre de la tabla de user a data_user
146 •    RENAME TABLE user TO data_user
147      ;
148
```

Output

 Action Output ▼

	#	Time	Action
✓	3	11:27:07	SELECT * FROM transactions.user
✓	4	11:29:57	RENAME TABLE user TO data_user

Diagrama finalizado:



## 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.

```
98 # EXERCICI 2 --- Crear vista anomenada 'InformeTecnico'
99 • CREATE VIEW InformeTecnico AS
100 SELECT transaction.id AS 'ID de la transacció',
101         user.name AS Nom,
102         user.surname AS Cognom,
103         credit_card.iban AS IBAN,
104         company.company_name AS Companyia
105 FROM transaction
106 JOIN user
107 ON transaction.user_id = user.id
108 JOIN credit_card
109 ON transaction.credit_card_id = credit_card.id
110 JOIN company
111 ON transaction.company_id = company.id
112 ;
113 • SELECT *
114 FROM InformeTecnico
115 ORDER BY 'ID de la transacció' DESC
116 ;
```

ID de la transacció	Nom	Cognom	IBAN	Companyia
108B1D1D-5B23-A76C-55EF-C568E49A05DD	Kenyon	Hartman	R323456312213576817699999	Ac Fermentum Incorpo
EA2C3281-C9C1-A387-44F8-729FB4B51C76	Kenyon	Hartman	R323456312213576817699999	Ac Fermentum Incorpo
7DC26247-20EC-53FE-E555-B6C2E55CA5D5	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Indust
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Indust
72997E96-DC2C-A4D7-7C24-66C302F8AE5A	Kenyon	Hartman	BG45IVQL52710525608255	Fusce Corp.
8766F5E3-A331-B81E-8C44-E7DCD12C4F03	Kenyon	Hartman	BG45IVQL52710525608255	Fusce Corp.

InformeTecnico 16 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 318	14:07:26	CREATE VIEW InformeTecnico AS SELECT tran...	0 row(s) affected	0.016 sec
✗ 319	14:08:55	SELECT * FROM InformeTecnico ORDER BY tra...	Error Code: 1054. Unknown column 'transaction.id'...	0.000 sec
✓ 320	14:09:14	SELECT * FROM InformeTecnico ORDER BY 'ID...	586 row(s) returned	0.000 sec / 0.016 sec