

# S3.01. Manipulació de taules

Natalia Durán Gadea

28 de gener de 2025

## Descripció

En aquest sprint, es simula una situació empresarial en la qual has de realitzar diverses manipulacions en les taules de la base de dades. Al seu torn, hauràs de treballar amb índexs i vistes. En aquesta activitat, continuaràs treballant amb la base de dades que conté informació d'una empresa dedicada a la venda de productes en línia. En aquesta tasca, començaràs a treballar amb informació relacionada amb targetes de crèdit.

## Recursos

- **Objectius:** Manipulació de dades, treballar amb vistes i índexs.
- **Durada:** 3 dies.

## Lliurament

Emmagatzema en un repositori del teu GitHub una carpeta que contingui:

- L'arxiu `.sql` que contingui tots els scripts.
- Un PDF que contingui una captura de pantalla del workbench on es pugui observar el script de la consulta que vas fer i el resultat obtingut per a cada exercici.

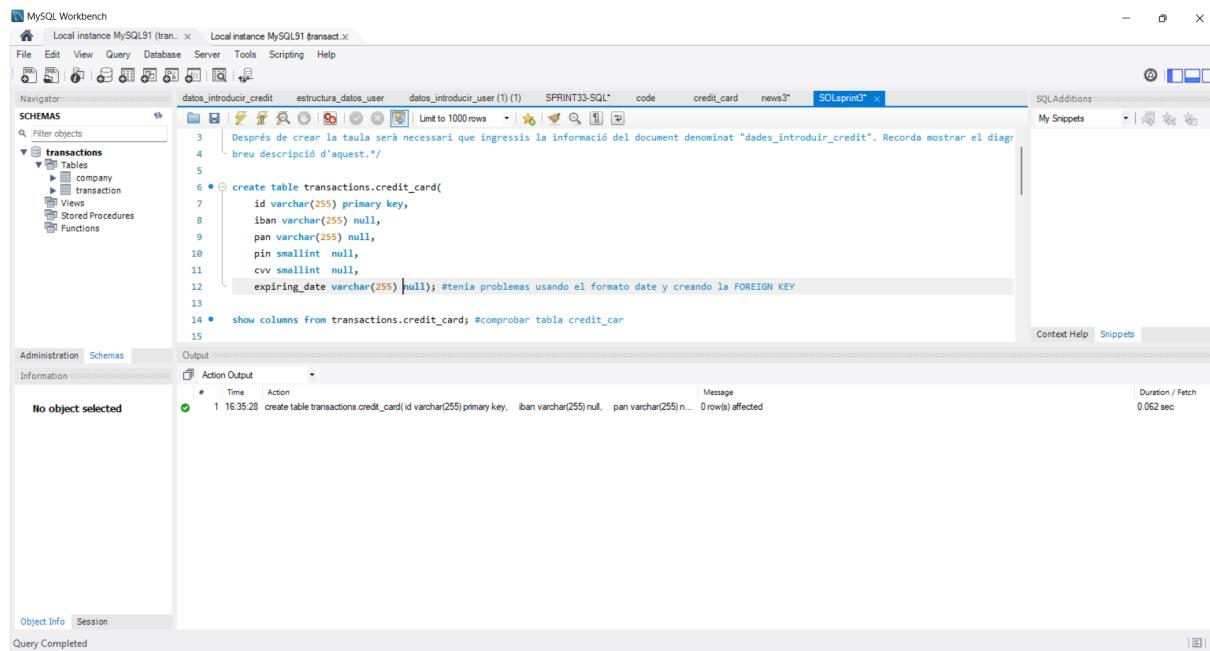
## Nivell 1

### 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.

- *La columna `id` actua com a clau principal de la taula, assegurant que cada registre sigui únic. A més, no permet valors nuls (`NOT NULL`), fet que garanteix que cada targeta de crèdit tingui un identificador obligatori i irrepetible.*

- La clau forana `id` estableix una relació amb la columna `id` de la taula `transaction`. Això assegura la integritat referencial entre ambdues taules, impedint que s'afegeixin registres a la taula `CREDIT_CARD` si no existeixen prèviament a la taula `transaction`.
- Creació de la taula `Credit_Card`. Especifico els tipus de dades, no he pogut crear la taula amb `FOREIGN KEY`. El tipus de dada `SMALLINT` s'utilitza per emmagatzemar números enters petits. És útil quan necessites emmagatzemar valors enters però vols optimitzar l'ús de memòria perquè el rang de valors que esperes és limitat. (Imatge 1).
- Utilitzo `SHOW COLUMNS` per visualitzar i comprovar l'estructura de la taula que acabo de crear. (Imatge 2)
- Compilo el codi de `dades_introduir_credit`. (Imatge 3)
- Vinculo `transaction.credit_card_id` amb `credit_card.id` fent el canvi de la dada perque tinguin el mateix tipus. (Imatge 4)



The screenshot shows the MySQL Workbench interface. In the center, there is a SQL editor window titled "credit\_card" containing the following SQL code:

```

3  Després de crear la taula serà necessari que ingressi la informació del document denominat "dades_introduir_credit". Recorda mostrar el diagr
4  àmbar descripció d'aquest.
5
6 *  create table transactions.credit_card(
7     id varchar(255) primary key,
8     iban varchar(255) null,
9     pan varchar(255) null,
10    pin smallint null,
11    cvv smallint null,
12    expiring_date varchar(255) null); #tenia problemas usando el formato date y creando la FOREIGN KEY
13
14 *  show columns from transactions.credit_card; #comprobar tabla credit_car
15

```

Below the SQL editor, the "Output" pane shows the results of the query:

#	Time	Action	Message	Duration / Fetch
1	16:35:28	create table transactions.credit_card(id varchar(255) primary key, iban varchar(255) null, pan varchar(255) n... 0 row(s) affected		0.062 sec

Figura 1: Nivell 1 Exercici 1. Creació taula Credit card

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL51 (transact.x)' is selected. The left sidebar shows the 'schemas' section with 'transactions' expanded, containing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main query editor window contains the following SQL code:

```

15
16
17 • show columns from transactions.credit_card; #comprobar tabla credit_card
18
19

```

The results grid shows the structure of the 'credit\_card' table:

Field	Type	Null	Key	Default	Extra
<b>id</b>	varchar(255)	NO	PRI		
<b>iban</b>	varchar(255)	YES			
<b>pan</b>	varchar(255)	YES			
<b>pin</b>	smallint	YES			
<b>cvv</b>	smallint	YES			

The output pane shows the execution history:

#	Time	Action	Message	Duration / Fetch
1	16:35:28	create table transactions.credit_card(id varchar(255) primary key, iban varchar(255) null, pan varchar(255), pin smallint, cvv smallint)	0 row(s) affected	0.062 sec
2	16:41:59	show columns from transactions.credit_card	6 row(s) returned	0.000 sec / 0.000 sec

Figura 2: Nivell 1 Exercici 1. Visualització de l'estructura de la taula

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL51 (transact.x)' is selected. The left sidebar shows the 'schemas' section with 'transactions' expanded, containing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main query editor window contains the following SQL code:

```

1
2 -- Insertamos datos de credit_card
3 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2938', 'TR301950312213576817638661', '5424465566813633', 'Ccu-2945', 'D026854763748537475216568689', '5142423821948826
4 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2952', 'B6451VQL5271052560255', '4556 453 55 5287', '45
5 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2959', 'CR7242477244335841535', '372461377349375', '3583
6 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2966', 'G672LKTQ15627628377363', '448586 886747 7265', 'Ccu-2973', 'T8780622813592429456346', '544 58654 54343 384
7 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2980', 'DE392418183806277136', '402400 7145845969', '5
8 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2987', 'GE09681434837748781813', '3763 747687 76666', '2
9 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-2994', 'BH62714428368066765294', '34428327325293', '754
10 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-3001', 'CY49887426654774581266832110', '511722 924833 22
11 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-3008', 'U50716693616119230', '4485744464433884', '1856
12 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-3015', 'S21567085819', '11722 924833 22
13 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-3022', 'TR37387255831...', '11722 924833 22

```

The output pane shows the execution history:

#	Time	Action	Message	Duration / Fetch
265	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4772', 'ME59832015454...', '1 row(s) affected		0.000 sec
266	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4779', 'F1910923181097...', '1 row(s) affected		0.015 sec
267	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4786', 'S1517031041731...', '1 row(s) affected		0.000 sec
268	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4793', 'H95215627749...', '1 row(s) affected		0.000 sec
269	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-47973', 'T8780622813592429456346', '544 58654 54343 384		0.000 sec
270	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4807', 'LB19293135221...', '1 row(s) affected		0.016 sec
271	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4814', 'MRA845282437...', '1 row(s) affected		0.000 sec
272	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4821', 'LT25314750563...', '1 row(s) affected		0.000 sec
273	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4829', 'BG11ULM30149...', '1 row(s) affected		0.016 sec
274	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4836', 'PT34592171131...', '1 row(s) affected		0.000 sec
275	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4842', 'SA21567085819...', '1 row(s) affected		0.000 sec
276	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4849', 'SE28132124871...', '1 row(s) affected		0.000 sec
277	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Ccu-4856', 'TR37387255831...', '1 row(s) affected		0.000 sec

Figura 3: Nivell 1 Exercici 1. Compilació del codi de dades\_introducir\_credit

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database is set to 'Local instance MySQL91 (transact.x)'. The main window has two tabs: 'credit\_card' and 'news3\*'. The 'credit\_card' tab is active, displaying the following SQL code:

```
37
38      #Vinculación de transaction.credit_card_id con credit_card.id;
39      ## Cambio el tipo de dato para que sean iguales: VARCHAR(8)
40 •  ALTER TABLE transaction CHANGE credit_card_id credit_card_id VARCHAR(8) NOT NULL;
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
```

In the bottom right corner of the code editor, there are two status indicators: 'Context Help' and 'Snippets'.

The 'Output' pane at the bottom shows the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
277	16:42:52	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-4856', TR37387255831...)	1 row(s) affected	0.000 sec
278	16:51:32	ALTER TABLE transaction CHANGE credit_card_id credit_card_id VARCHAR(8) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec

The status bar at the bottom of the interface indicates 'Query Completed'.

Figura 4: Nivell 1 Exercici 1. Cambiar el tipus de dada per vincular les dues taules

- Genero el Diagrama E/R. (Imatge 5)

### 1. Relació entre transaction i credit\_card

- La taula **transaction** està vinculada a **credit\_card** mitjançant el camp **credit\_card\_id**.
- Aquesta és una relació de molts-a-un, és a dir, una targeta de crèdit pot estar associada a moltes transaccions, però cada transacció només està vinculada a una targeta específica.

### 2. Relació entre transaction i company

- La taula **transaction** es relaciona amb **company** mitjançant el camp **company\_id**.
- Aquesta relació també és de molts-a-un, on una empresa pot tenir moltes transaccions, però cada transacció pertany a una sola empresa.

### 3. Estructura general de les taules

- **credit\_card**: Conté informació de targetes de crèdit, com l'ID, l'IBAN, el número PAN, el PIN, el CVV i la data de caducitat.
- **transaction**: Emmagatzema dades de transaccions, com la geolocalització (latitud i longitud), l'usuari, l'import, i si ha estat denegada.
- **company**: Inclou informació sobre les empreses, com el nom, el correu electrònic, el país i el lloc web.

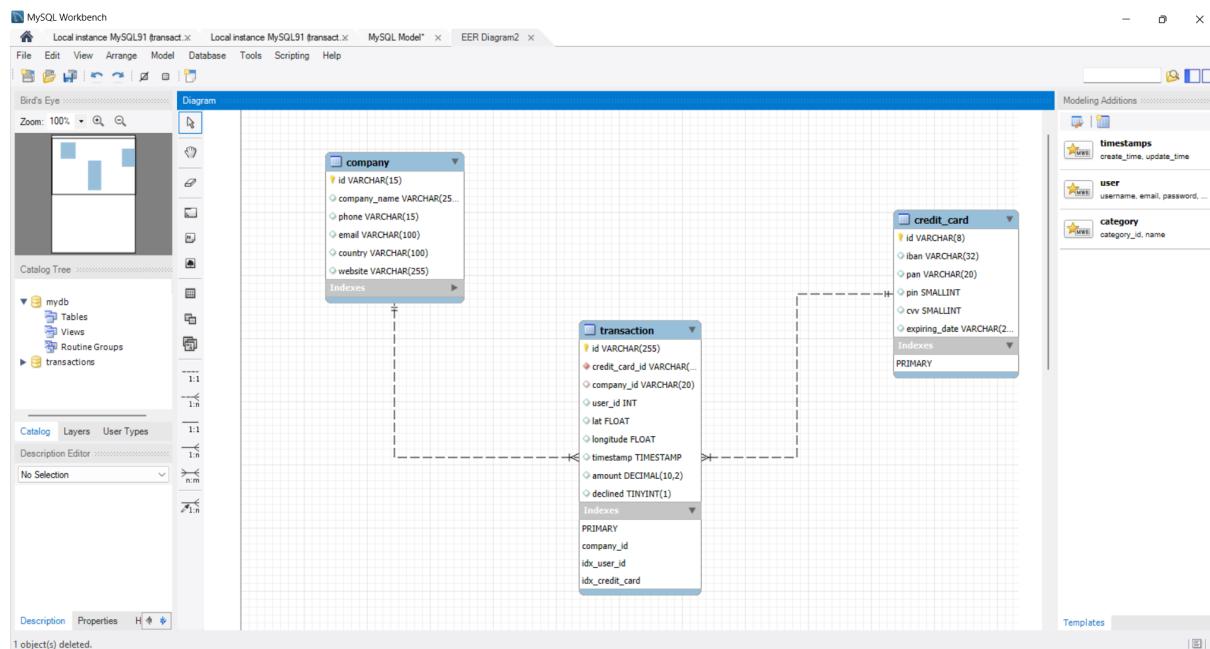


Figura 5: Nivell 1 Exercici 1. Diagrama E/R

## 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.

- R/Consulta de l'usuari CcU-2938. Verifiquem que hi ha un error.(Imatge 6)
- R/Cambi del registre(Imatge 7)
- R/Confirmació del canvi del registre de l'usuari CcU-2938(Imatge 8)

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane shows the 'transactions' schema with tables like 'company', 'credit\_card', and 'transaction'. The main area displays a SQL editor window with the following code:

```

61 -- El departament de Recursos Humans ha identificat un error en el número
62 -- de compte de l'usuari amb ID CcU-2938. La informació que ha de mostrarse
63 -- per a aquest registre és: R323456312213576817699999. Recorda mostrar que el canvi es va realitzar.
64
65 • SELECT *
66   FROM credit_card
67 WHERE id = "CcU-2938";
68
69
70

```

Below the code, the Result Grid shows a single row of data:

	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22
●	NULL	NULL	NULL	NULL	NULL	NULL

At the bottom, the Action Output pane shows two log entries:

#	Time	Action	Message	Duration / Fetch
278	16:51:32	ALTER TABLE transaction CHANGE credit_card_id credit_card_id VARCHAR(8) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
279	17:12:17	SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec

Figura 6: Nivell 1 Exercici 2. Consulta de l'usuari CcU-2938

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator Schemas Administration Information
No object selected
Object Info Session Query Completed

Schemas
transactions
  Tables
    company
    credit_card
    transaction
  Views
  Stored Procedures
  Functions

Schemas
datas_introducir_credit estructura_datos_user datas_introducir_user(1)(1) SPRINT33-SQL* code credit_card news3* SOLprint3*
Limit to 1000 rows
79
80  #Cambio:
81  • UPDATE credit_card set
82    iban = 'R323456312213576817699999'
83    WHERE id = "CcU-2938";
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

Output
Action Output
# Time Action
279 17:12:17 SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 1000
280 17:57:01 UPDATE credit_card set iban = 'R323456312213576817699999' WHERE id = "CcU-2938"

Message
1 row(s) returned
1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
Duration / Fetch
0.016 sec / 0.000 sec
0.016 sec

Context Help Snippets

```

Figura 7: Nivell 1 Exercici 2. Canvi del registre de l'usuari CcU-2938

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator Schemas Administration Information
No object selected
Object Info Session Query Completed

Schemas
transactions
  Tables
    company
    credit_card
    transaction
  Views
  Stored Procedures
  Functions

Schemas
datas_introducir_credit estructura_datos_user datas_introducir_user(1)(1) SPRINT33-SQL* code credit_card news3* SOLprint3*
Limit to 1000 rows
85
86  #Confirmación del cambio:
87  • SELECT *
88    FROM credit_card
89    WHERE id = "CcU-2938";
90
91
92
93
94

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: 
id iban pan pin cvv expiring_date
CcU-2938 R323456312213576817699999 502446556612633 3257 994 10/30/22
* 0000 0000 0000 0000 0000 0000

credit_card 4 x
Output
Action Output
# Time Action
280 17:57:01 UPDATE credit_card set iban = 'R323456312213576817699999' WHERE id = "CcU-2938"
281 17:58:01 SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 1000

Message
1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
1 row(s) returned
Duration / Fetch
0.016 sec
0.000 sec / 0.000 sec

Context Help Snippets

```

Figura 8: Nivell 1 Exercici 2. Confirmació i verificació del canvi de l'usuari CcU-2938

## 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
```

- *R/Consulta de l'usuari CcU-2938(Imatge 9)*
  - *R/Creació de l'usuari CcU-2938 tant a la taula Company como a la taula de Credit card(Imatge 10)*
  - *R/Introducció de les dades(Imatge 11)*

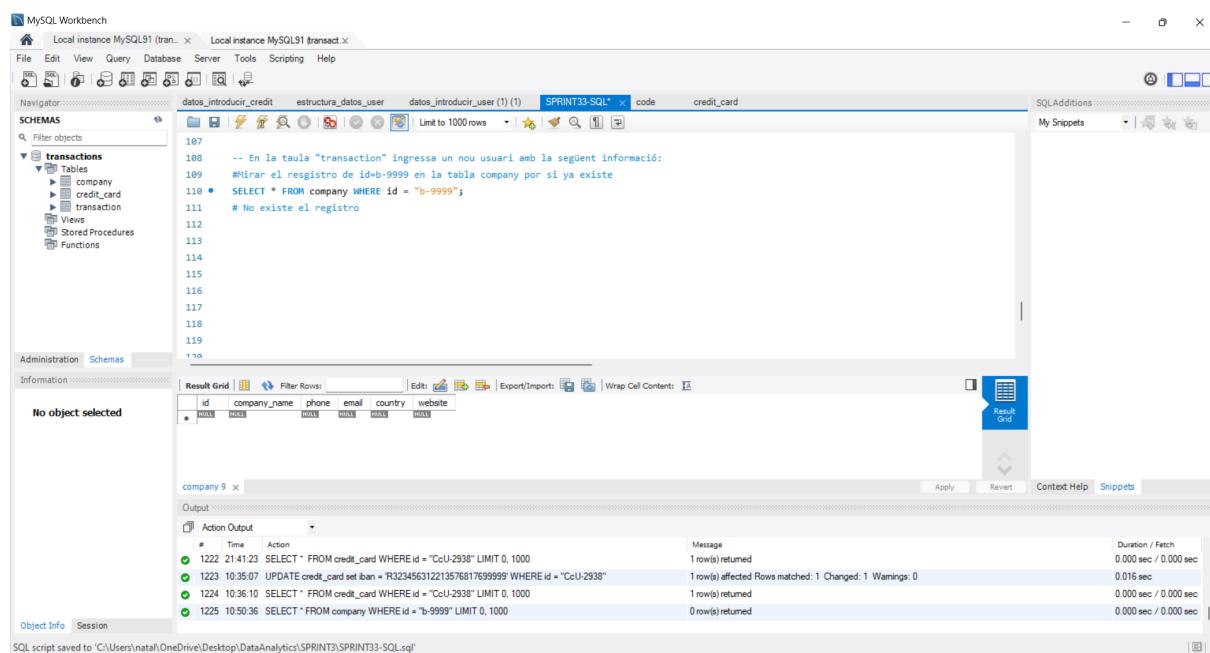


Figura 9: Nivell 1 Exercici 3. Confirmació de la inexistència de l'usuari CcU-2938

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, Help, and a toolbar with various icons. The left sidebar displays the Navigator, Schemas (with 'transactions' selected), Tables, Views, Stored Procedures, and Functions. The main area shows a query editor with the following SQL code:

```
100  
101  
102  
103  
104    #Debo crear primero los registros company.id = "b-9999" y credit_card.id="CcU-9999" por ser claves externas a la tabla transaction:  
105 • INSERT INTO company (id)  
106     VALUES ("b-9999");  
107 • INSERT INTO credit_card (id)  
108     VALUES ("CcU-9999");  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122
```

The bottom right corner shows the Output pane with the following log entries:

#	Time	Action	Message	Duration / Fetch
282	18.05.05	SELECT * FROM company WHERE id = 'b-9999' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
283	18.06.09	INSERT INTO credit_card (id) VALUES ('CcU-9999')	1 row(s) affected	0.016 sec

Figura 10: Nivell 1 Exercici 3. Creació de l'usuari CcU-2938

The screenshot shows the MySQL Workbench interface. The top navigation bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar displays the Navigator, Schemas (with transactions selected), Tables (company, credit\_card, transaction), Views, Stored Procedures, and Functions. The main area shows a SQL editor with the following code:

```
138
139
140
141
142
143
144
145    #Visualizo la nueva transacción
146 • SELECT * FROM transaction
147 WHERE id = "108B1D1D-5B23-A76C-55EF-C568E49A99D0";
```

The Result Grid shows the following data:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A99D0	CCU-9999	b-9999	9999	829.999	-117.999	NULL	111.11	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The bottom pane shows the transaction log with the following entries:

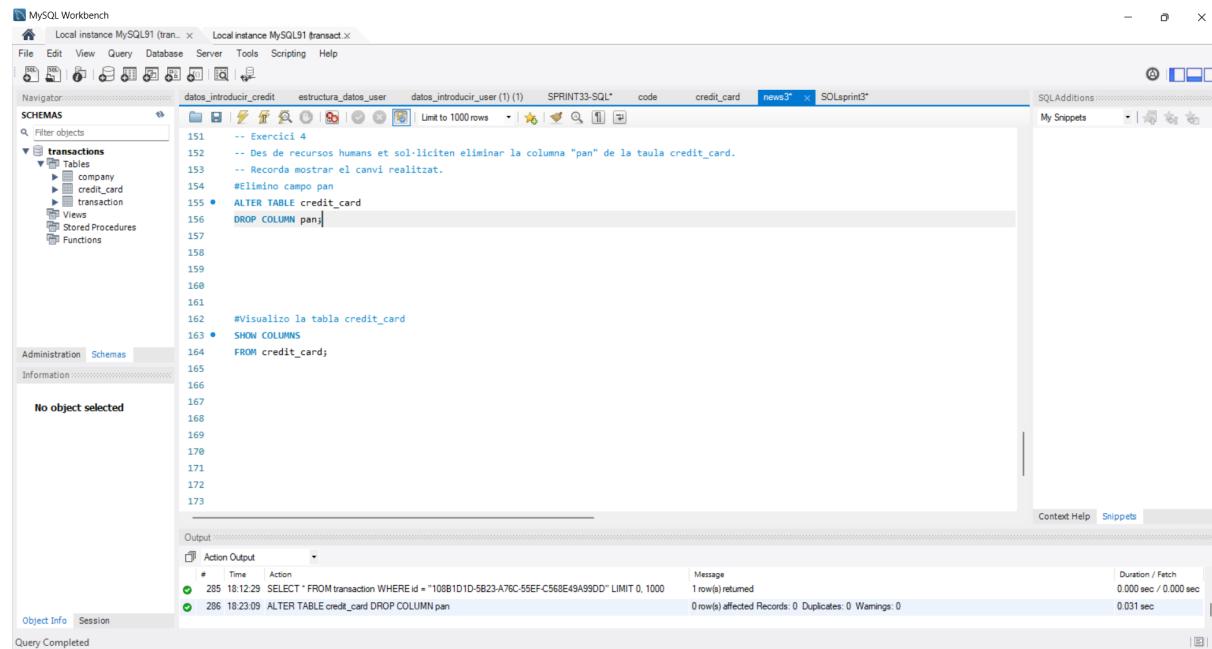
Output	Action	Time	Message	Duration / Fetch
transaction 6	Insert into transaction(id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) values ("108B1D1D-5B23-A76C-55EF-C568E49A99D0")	18:12:17	Error Code: 1062. Duplicate entry '108B1D1D-5B23-A76C-55EF-C568E49A99D0' for key 'transaction.PRIMARY'	0.000 sec
	SELECT * FROM transaction WHERE id = "108B1D1D-5B23-A76C-55EF-C568E49A99D0" LIMIT 0, 1000	18:12:29	1 row(s) returned	0.000 sec / 0.000 sec

Figura 11: Nivell 1 Exercici 3. Introducció de les dades

## Exercici 4

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

- R/ Eliminar la columna PAN (Imatge 12)
- R/ Visualització del canvi. (Imatge 13)



The screenshot shows the MySQL Workbench interface. In the top navigation bar, there are tabs for 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. Below the tabs, the 'Navigator' pane shows the database schema with 'SCHEMAS' expanded, revealing 'transactions' and its tables: 'company', 'credit\_card', and 'transaction'. The 'credit\_card' table is selected. The main 'Code' tab contains the following SQL code:

```
-- Exercici 4
-- Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card.
-- Recorda mostrar el canvi realitzat.
#Elimino campo pan
ALTER TABLE credit_card
DROP COLUMN pan;
```

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

#	Action	Time	Message	Duration / Fetch
265	SELECT * FROM transaction WHERE id = "108B1D1D-5B23-A76C-55EF-C568E49A99DD" LIMIT 0, 1000	18:12:29	1row(s) returned	0.000 sec / 0.000 sec
266	ALTER TABLE credit_card DROP COLUMN pan	18:23:09	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec

Figura 12: Nivell 1 Exercici 4. Eliminar la columna pan

The screenshot shows the MySQL Workbench interface. In the center, there is a SQL editor window titled 'credit\_card' containing the following SQL code:

```
161
162
163
164     #Visualizo la tabla credit_card
165 •   SHOW COLUMNS
166     FROM credit_card;
167
168
169
170
```

Below the SQL editor is a 'Result Grid' table showing the columns of the 'credit\_card' table:

Field	Type	Null	Key	Default	Extra
id	varchar(255)	NO	PRI		
iban	varchar(255)	YES			
pin	smallint	YES			
cvv	smallint	YES			
expiring_date	varchar(255)	YES			

At the bottom of the interface, the 'Action Output' section displays the following log entries:

#	Time	Action	Message	Duration / Fetch
286	18:23:09	ALTER TABLE credit_card DROP COLUMN pin	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
287	18:23:39	SHOW COLUMNS FROM credit_card	5 row(s) returned	0.016 sec / 0.000 sec

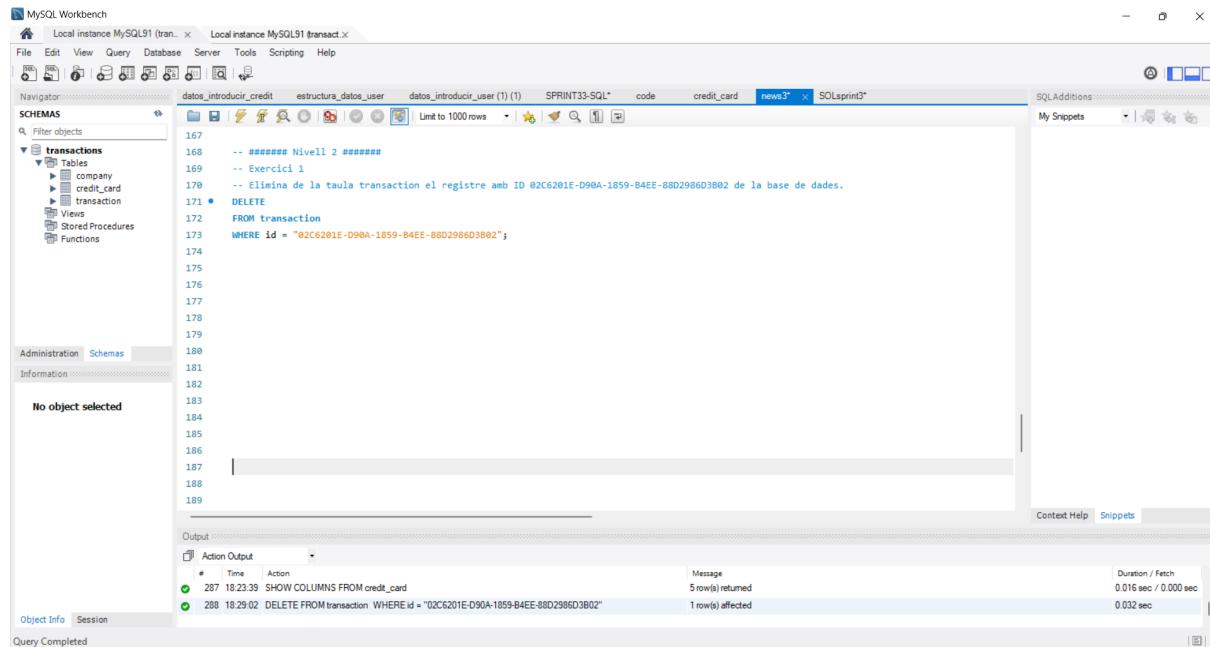
Figura 13: Nivell 1 Exercici 4. Visualització del canvi

## Nivell 2

### Exercici 1

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

- R/Visualització del ID: 02C6201E-D90A-1859-B4EE-88D2986D3B02 (Imatge 14)
- R/Visualització del canvi. (Imatge 15)



The screenshot shows the MySQL Workbench interface. In the top navigation bar, there are tabs for 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. Below the tabs, the 'Navigator' pane shows the database schema with a tree view under 'SCHEMAS'. The 'transactions' schema is selected, and its tables ('company', 'credit\_card', and 'transaction') are visible. The main workspace contains a SQL editor window titled 'SPRINT33-SQL\*' with the following code:

```

167
168 -- ##### Nivell 2 #####
169 -- Exercici 1
170 -- Elimina de la taula transaction el registre amb ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de dades.
171 • DELETE
172   FROM transaction
173   WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189

```

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

#	Time	Action	Message	Duration / Fetch
287	18:23:39	SHOW COLUMNS FROM credit_card	5 row(s) returned	0.016 sec / 0.000 sec
288	18:29:02	DELETE FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02"	1 row(s) affected	0.032 sec

Figura 14: Nivell 2 Exercici 1. Visualització del ID: 02C6201E-D90A-1859-B4EE-88D2986D3B02

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database is set to 'Local instance MySQL91 (transact.x)'. The main area displays a SQL editor window with the following code:

```
179
180
181
182
183
184    #visualizo el id eliminado.
185 •   SELECT * FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
186
187
188
189
190
191
192
193
```

Below the editor, the 'Result Grid' pane shows a table structure and an empty data row:

	<u>id</u>	<u>credit_card_id</u>	<u>company_id</u>	<u>user_id</u>	<u>lat</u>	<u>longitude</u>	<u>timestamp</u>	<u>amount</u>	<u>declined</u>
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

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

#	Time	Action	Message	Duration / Fetch
285	18:29:02	DELETE FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02"	1 row(s) affected	0.032 sec
289	18:29:51	SELECT * FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	0 row(s) returned	0.016 sec / 0.000 sec

Figura 15: Nivell 2 Exercici 1. Visualització del canvi

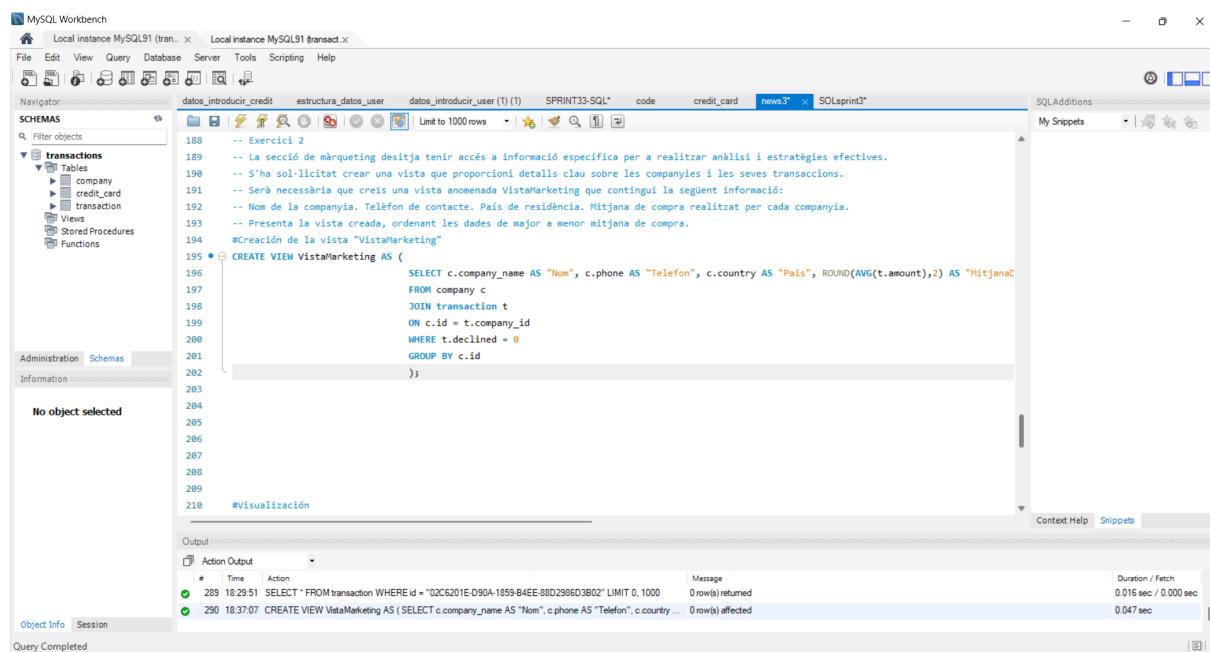
## Exercici 2

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 realitzada per cada companyia.

Presenta la vista creada, ordenant les dades de major a menor mitjana de compra.

- R/Creació de la vista (Imatge 16)
- R/Visualització de la vista (Imatge 17)



The screenshot shows the MySQL Workbench interface with the SQL editor tab active. The code pane contains the following SQL script:

```

-- 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.

#Creació de la vista "VistaMarketing"
CREATE VIEW VistaMarketing AS (
    SELECT c.company_name AS "Nom", c.phone AS "Telefon", c.country AS "País", ROUND(AVG(t.amount),2) AS "MitjanaC"
    FROM company c
    JOIN transaction t
    ON c.id = t.company_id
    WHERE t.declined = 0
    GROUP BY c.id
);

#Visualización

```

The output pane shows the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
289	18:29:51	SELECT * FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	0 row(s) returned	0.016 sec / 0.000 sec
290	18:37:07	CREATE VIEW VistaMarketing AS ( SELECT c.company_name AS "Nom", c.phone AS "Telefon", c.country ...	0 row(s) affected	0.047 sec

Figura 16: Nivell 2 Exercici 2. Creació de la vista

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database is set to 'Local instance MySQL91 (transact.x)'. The main area displays a SQL editor window with the following code:

```
206
207
208
209
210    #Visualización
211 •   SELECT *
212     FROM VistaMarketing
213     ORDER BY MitjanaDeCompra DESC;
214
215
216
217
218
219
220
221
```

The result grid shows the following data:

Nom	Telefon	País	MitjanaDeCompra
Eget Ipsum Ltd	03 67 44 56 72	United States	481.86
Sed Id Limited	07 28 18 18 13	United States	477.51
Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.10
Nunc Sit Incorporated	07 28 42 63 63	Norway	461.83
Non Magna LLC	06 71 73 13 17	United Kingdom	458.74

The output pane shows the following log entries:

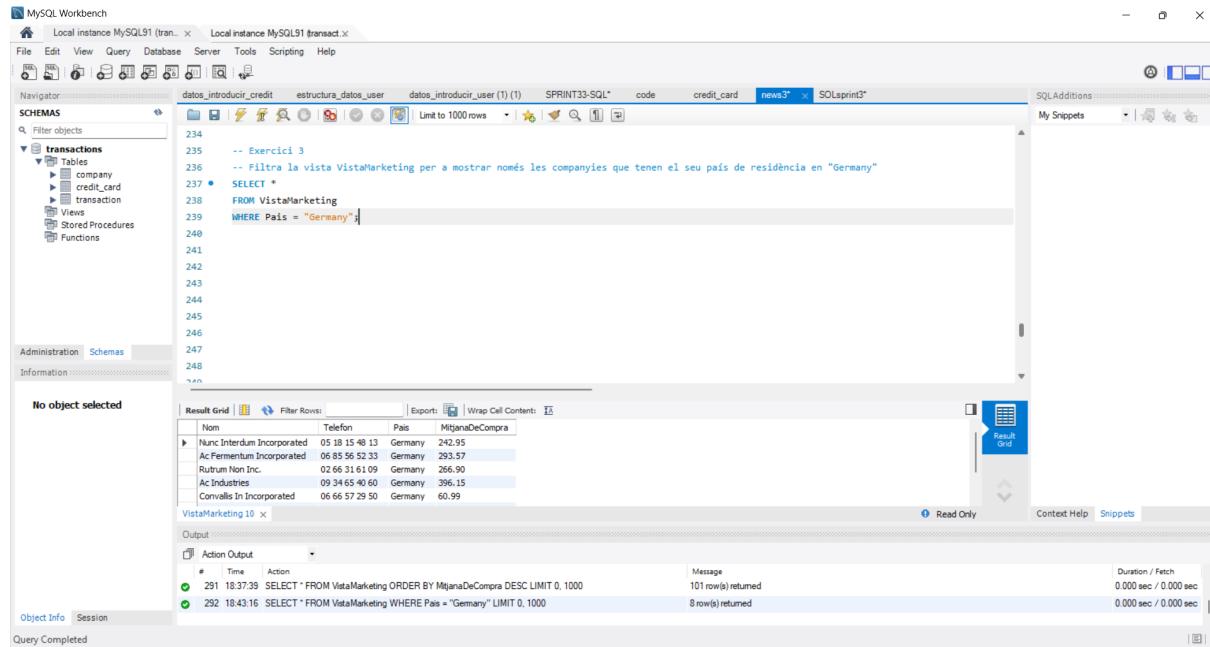
#	Time	Action	Message	Duration / Fetch
290	18:37:07	CREATE VIEW VistaMarketing AS ( SELECT c.company_name AS "Nom", c.phone AS "Telefon", c.country AS "País", AVG(p.price) AS "MitjanaDeCompra" FROM company c JOIN product p ON c.id = p.company_id GROUP BY c.id )	0 row(s) affected	0.047 sec
291	18:37:39	SELECT * FROM VistaMarketing ORDER BY MitjanaDeCompra DESC LIMIT 0, 1000	101 row(s) returned	0.000 sec / 0.000 sec

Figura 17: Nivell 2 Exercici 2. Visualització de la vista

### Exercici 3

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

- R/Filtrem fent servir un WHERE només les companyies que tenen el seu país de residència en Germany (Imatge 18)



The screenshot shows the MySQL Workbench interface. In the SQL editor tab, a query is written to filter companies from the VistaMarketing view where the country is Germany:

```

235 -- Exercici 3
236 -- Filtra la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"
237 • SELECT *
238 FROM VistaMarketing
239 WHERE Pais = "Germany";
240
241
242
243
244
245
246
247
248
249

```

The results grid displays the filtered data:

Nom	Telefon	Pais	MitjanaDeCompra
Nunc Interdum Incorporated	05 18 15 48 13	Germany	242.95
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.57
Rutrum Non Inc.	02 63 16 10 09	Germany	266.90
Ac Industries	09 34 65 40 60	Germany	396.15
Convallis In Incorporated	06 66 57 29 50	Germany	60.99

The output pane shows the executed queries and their results:

#	Time	Action	Message	Duration / Fetch
291	18:37:39	SELECT * FROM VistaMarketing ORDER BY MitjanaDeCompra DESC LIMIT 0, 1000	101 row(s) returned	0.000 sec / 0.000 sec
292	18:43:16	SELECT * FROM VistaMarketing WHERE Pais = "Germany" LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec

Figura 18: Nivell 2 Exercici 3. Filtratge de les companyies segons el país de residència en Germany

## 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 (Imatge 19).

Recordatori: En aquesta activitat, és necessari que descrigui el "pas a pas" de les tasques realitzades. És important realitzar descripcions senzilles, simples i fàcils de comprendre. Per a realitzar aquesta activitat, haurà de treballar amb els arxius denominats *estructura\_dades\_user* i *dades\_introducir\_user*.

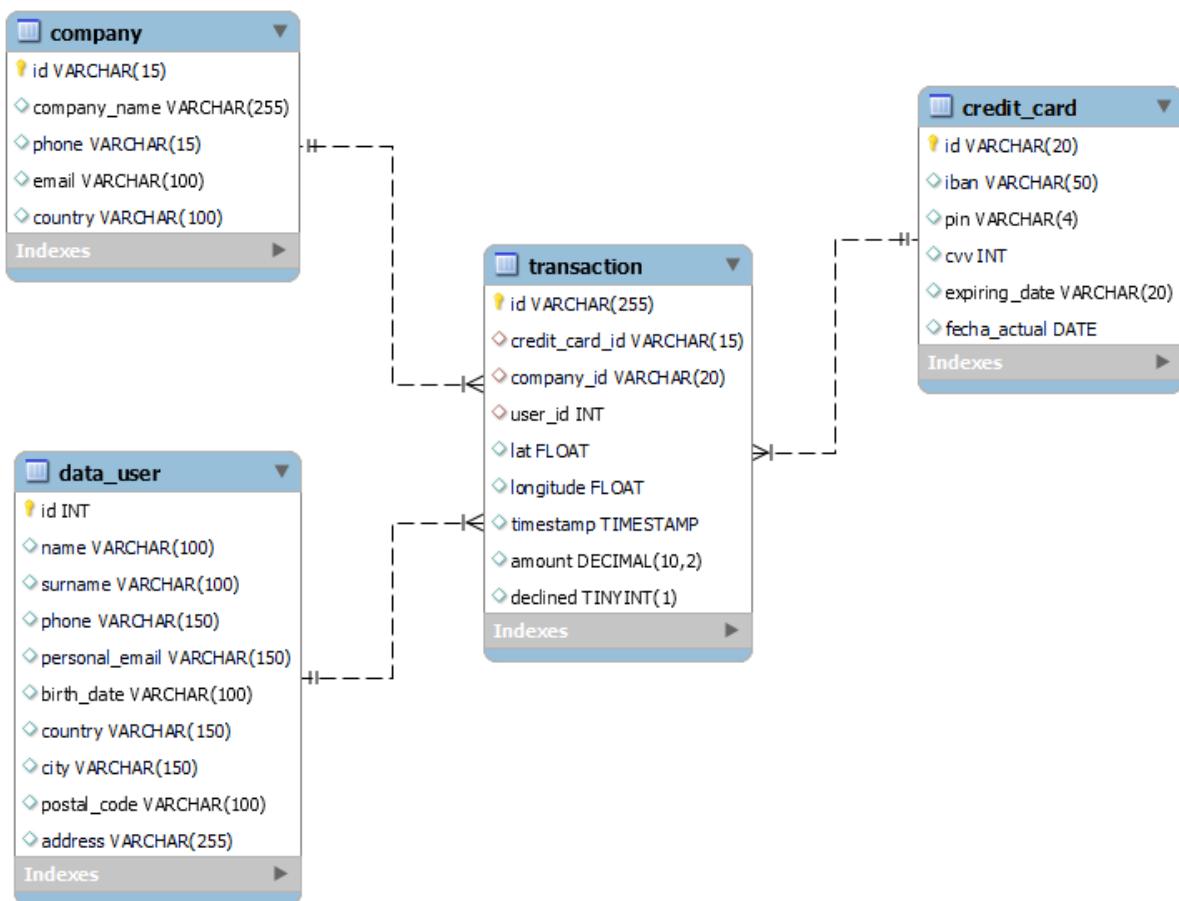
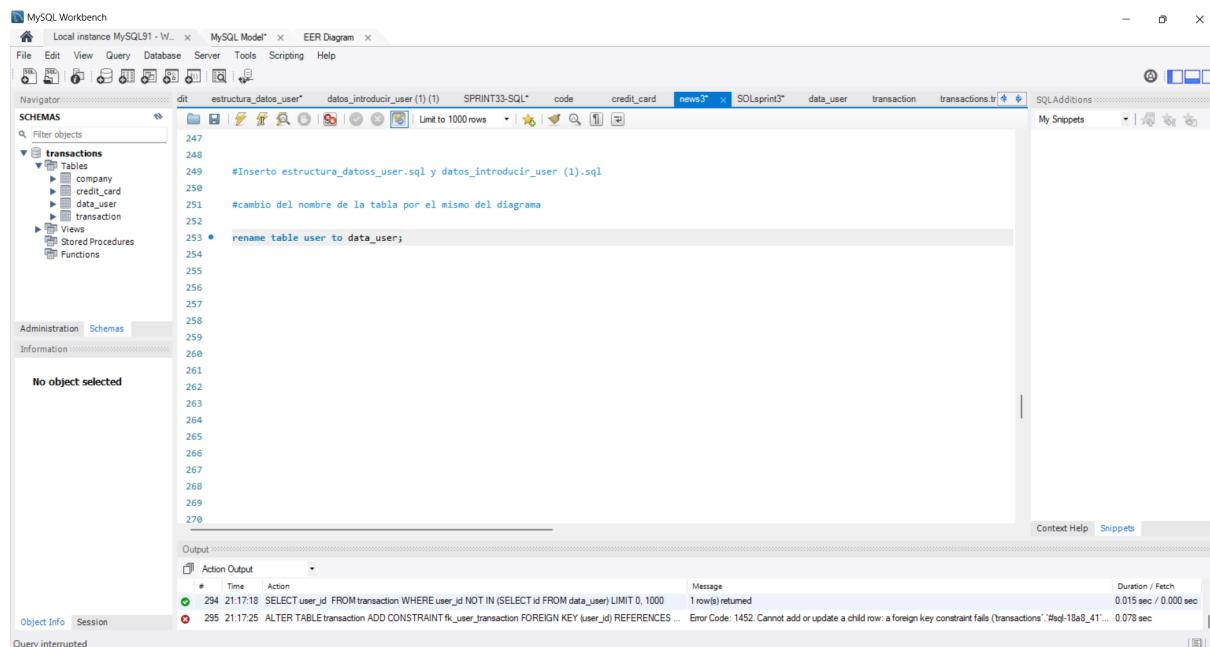


Figura 19: Nivell 3 Exercici 1. Diagrama E/R d'un company

- R/ Canvi de nom de la taula **user** per **data\_user** per coherència amb el model E/R de l'exercici (Imatge 20).
- R/ Intent de creació d'una clau forana per relacionar la taula **data\_user** amb la taula **transaction** (Imatge 21).
- R/ Comprovació que l'usuari 9999 no existeix a la taula **data\_user** (Imatge 22).
- R/ Eliminació de les files problemàtiques en transaction (Imatge 23).
- R/ Creació d'una clau forana per relacionar la taula **data\_user** amb la taula **transaction** (Imatge 24).
- R/ Modificació de les taules per adaptar-se al model de l'exercici. S'ha eliminat la columna **website** de la taula **transactions.company** i s'ha afegit la columna **fecha\_actual** de tipus data a **transactions.credit\_card**. A més, s'ha renombrat la columna **email** a **personal\_email** i s'ha definit una longitud màxima de 150 caràcters a **transactions.data\_user**. (Imatge 25).
- R/ El codi desactiva les verificacions de claus foranes amb `set foreign_key_checks=0` per permetre modificar l'estructura de la taula sense restriccions. Es modifica la columna **id** de la taula **transactions.credit\_card** canviant el seu tipus a `varchar(20)`. Finalment, es tornen a activar les claus foranes amb `set foreign_key_checks=1` per mantenir la integritat referencial. (Imatge 26).



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following code:

```

247
248
249 #Inserto estructura_datoss_user.sql y datos_introducir_user (1).sql
250
251 #cambio del nombre de la tabla por el mismo del diagrama
252
253 • rename table user to data_user;
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270

```

The Output pane shows the results of the query:

Action	Time	Message	Duration / Fetch
254	21:17:18	SELECT user_id FROM transaction WHERE user_id NOT IN (SELECT id FROM data_user) LIMIT 0, 1000	1 row(s) returned 0.015 sec / 0.000 sec
255	21:17:25	ALTER TABLE transaction ADD CONSTRAINT fk_user_transaction FOREIGN KEY (user_id) REFERENCES ...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (transactions'.#sql-18a8_41'... 0.078 sec

Figura 20: Nivell 3 Exercici 1. Renombrar la taula **user** per **data\_user**

The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'schemas' section, there is a 'transactions' schema expanded, showing tables like company, credit\_card, data\_user, and transaction. In the main editor area, a SQL script is being run:

```

256
257
258
259
260
261
262
263
264     #Aparece un error debido al usuario 9999 que no aparece en la tabla data_user
265 • ALTER TABLE transaction
266     ADD CONSTRAINT fk_user_transaction
267     FOREIGN KEY (user_id)
268     REFERENCES data_user(id);
269
270
271
272
273
274
275
276
277
278
279

```

The output window at the bottom shows the results of the last two statements:

Action	Time	Message	Duration / Fetch
296 21:26:11 SELECT user_id FROM transaction WHERE user_id NOT IN (SELECT id FROM data_user) LIMIT 0, 1000		1 row(s) returned	0.000 sec / 0.000 sec
297 21:32:40 ALTER TABLE transaction ADD CONSTRAINT fk_user_transaction FOREIGN KEY (user_id) REFERENCES data_user(id)		Error Code: 1452 Cannot add or update a child row: a foreign key constraint fails (transactions' '#obj-1ba8_41')	0.047 sec

Object Info Session

Query interrupted

Figura 21: Nivell 3 Exercici 1. Error al crear la foreign key per relacionar les taules

The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'schemas' section, there is a 'transactions' schema expanded, showing tables like company, credit\_card, data\_user, and transaction. In the main editor area, a SQL query is being run:

```

265
266
267
268
269
270
271     #Comprobación
272 • SELECT user_id
273     FROM transaction
274     WHERE user_id NOT IN (SELECT id FROM data_user);
275
276
277
278
279
280

```

The result grid shows the output of the query:

user_id
9999

In the bottom right corner of the result grid, there is a 'Result Grid' button.

The output window at the bottom shows the results of the last two statements:

Action	Time	Message	Duration / Fetch
295 21:17:25 ALTER TABLE transaction ADD CONSTRAINT fk_user_transaction FOREIGN KEY (user_id) REFERENCES ...		Error Code: 1452 Cannot add or update a child row: a foreign key constraint fails (transactions' '#obj-1ba8_41' ...	0.078 sec
296 21:26:11 SELECT user_id FROM transaction WHERE user_id NOT IN (SELECT id FROM data_user) LIMIT 0, 1000		1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

Figura 22: Nivell 3 Exercici 1. Comprovació de la inexistència de dades de l'usuari 9999 a la taula data\_user

The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'transactions' schema, the 'Tables' section lists 'company', 'credit\_card', 'data\_user', and 'transaction'. The 'transaction' table is selected. The main pane displays a SQL script:

```

269
270      #Desactivació del mode segur per poder eliminar el registre
271      SET SQL_SAFE_UPDATES = 0;
272
273
274      #Eliminació de les dades de l'usuari 9999 per solucionar el problema
275      DELETE FROM transaction
276      WHERE user_id NOT IN (SELECT id FROM data_user);
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292

```

The output pane shows the results of the query execution:

#	Time	Action	Message	Duration / Fetch
299	11:08:19	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.000 sec
300	11:08:23	DELETE FROM transaction WHERE user_id NOT IN (SELECT id FROM data_user)	1 row(s) affected	0.000 sec

At the bottom, the status bar says 'Query Completed'.

Figura 23: Nivell 3 Exercici 1. Eliminació de les dades de l'usuari 9999

The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'transactions' schema, the 'Tables' section lists 'company', 'credit\_card', 'data\_user', and 'transaction'. The 'transaction' table is selected. The main pane displays a SQL script:

```

296
297
298
299
300      alter table transaction add constraint fk_data_user_id foreign key (user_id) references data_user(id);
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319

```

The output pane shows the results of the query execution:

#	Time	Action	Message	Duration / Fetch
300	11:08:23	DELETE FROM transaction WHERE user_id NOT IN (SELECT id FROM data_user)	1 row(s) affected	0.000 sec
301	11:14:58	alter table transaction add constraint fk_data_user_id foreign key (user_id) references data_user(id)	596 row(s) affected Records: 596 Duplicates: 0 Warnings: 0	0.076 sec

At the bottom, the status bar says 'Query Completed'.

Figura 24: Nivell 3 Exercici 1. Creació de la foreign key per relacionar la taula transaction amb data\_user

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `transactions` with tables: company, credit\_card, data\_user, and transaction.
- SQL Editor:** Displays the following SQL code (lines 305-325):
 

```

305
306
307
308  #Modificación de las tablas para adaptarse al modelo del ejercicio
309 • alter table transactions.company drop website;
310 • alter table transactions.credit_card add column fecha_actual DATE;
311 • alter table transactions.data_user change email personal_email varchar(150);
312
313
314
315
316
317
318
319
320
321
322
323
324
325
--
```
- Output:** Shows the execution results for each query:
 

#	Time	Action	Message	Duration / Fetch
301	11:14:58	alter table transaction add constraint fk_data_user_id foreign key (user_id) references data_user(id)	586 row(s) affected Records: 586 Duplicates: 0 Warnings: 0	0.078 sec
302	11:19:43	alter table transactions.company drop website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
303	11:19:45	alter table transactions.credit_card add column fecha_actual DATE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.000 sec
304	11:19:47	alter table transactions.data_user change email personal_email varchar(150)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec

Figura 25: Nivell 3 Exercici 1. Modificació de les taules per adaptar-se al model de l'exercici

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `transactions` with tables: company, credit\_card, data\_user, and transaction.
- SQL Editor:** Displays the following SQL code (lines 321-341):
 

```

321
322
323
324
325
326
327  #deshabilitar las FK
328 • set foreign_key_checks=0;
329 • alter table transactions.credit_card modify id varchar(20);
330  #habilitar las FK
331 • set foreign_key_checks=1;
332
333
334
335
336
337
338
339
340
341
--
```
- Output:** Shows the execution results for each query:
 

#	Time	Action	Message	Duration / Fetch
304	11:19:47	alter table transactions.data_user change email personal_email varchar(150)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
305	12:14:07	set foreign_key_checks=0	0 row(s) affected	0.000 sec
306	12:14:12	alter table transactions.credit_card modify id varchar(20)	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0	0.047 sec
307	12:14:20	set foreign_key_checks=1	0 row(s) affected	0.000 sec

Figura 26: Nivell 3 Exercici 1. Desactivació, modificació y activació de la FK

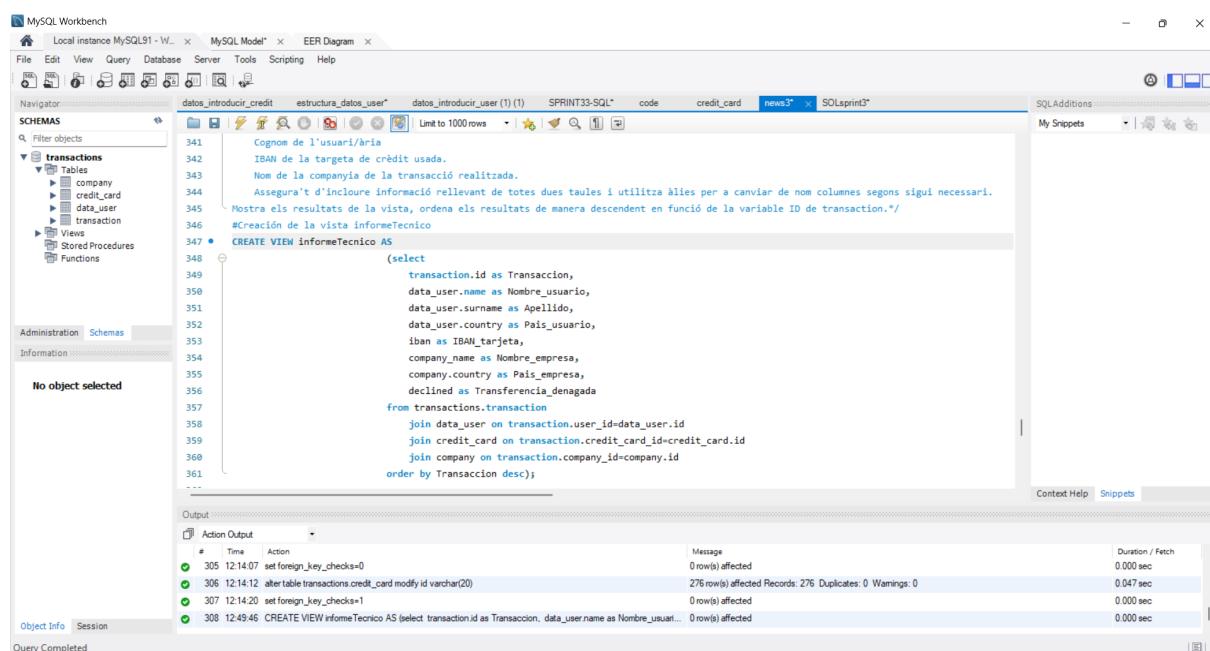
## Exercici 2

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**.

- R/ Aquest codi crea una vista anomenada **informeTecnico** que mostra informació sobre transaccions financeres. Inclou dades com l'ID de la transacció, el nom, cognom i país de l'usuari, l'IBAN de la targeta, el nom i país de l'empresa, i si la transferència ha estat denegada. Les dades es combinen de diverses taules relacionades mitjançant **JOIN** i s'ordenen en ordre descendent segons l'ID de la transacció. (Imatge 27).
- R/ Visualització de la vista **InformeTecnico** (Imatge 28).
- R/ Comprovació del diseny E/R generat per Workbench (Imatge 29).



The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema **transactions** with tables: **company**, **credit\_card**, **data\_user**, and **transaction**.
- SQL Editor:** Contains the SQL code for creating the **informeTecnico** view. The code includes comments explaining the purpose of the view and the query structure.
- Output:** Shows the results of the query execution, including the creation of the view and the execution of the SELECT statement.

```

CREATE VIEW informeTecnico AS
    (select
        transaction.id as Transaccion,
        data_user.name as Nombre_usuario,
        data_user.surname as Apellido,
        data_user.country as Pais_usuario,
        iban as IBAN_tarjeta,
        company.name as Nombre_empresa,
        company.country as Pais_empresa,
        company.country as Pais_empresa,
        declined as Transferencia_denegada
    from transactions.transaction
    join data_user on transaction.user_id=data_user.id
    join credit_card on transaction.credit_card_id=credit_card.id
    join company on transaction.company_id=company.id
    order by Transaccion desc);
  
```

Figura 27: Nivell 3 Exercici 2. Creació de la vista **InformeTecnico**

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the tabs "File", "Edit", "View", "Query", "Database", "Server", "Tools", "Scripting", and "Help" are visible. The "Query" tab is active. Below the tabs, the "Navigator" pane shows the schema "transactions" containing tables like "company", "credit\_card", "data\_user", and "transaction", along with views "informetecnico" and "vistamarketing". The main workspace displays a SQL editor with the following code:

```

365
366
367
368     #Visualización
369 •     SELECT *
370     FROM InformeTecnico
371
372
373
374
375
376
377
378
  
```

Below the code, the "Result Grid" pane shows the output of the query, listing transactions from various companies with their details and transfer amounts. The "Action Output" pane at the bottom shows the log of actions taken during the query execution.

Figura 28: Nivell 3 Exercici 2. Visualització de la vista InformeTecnico

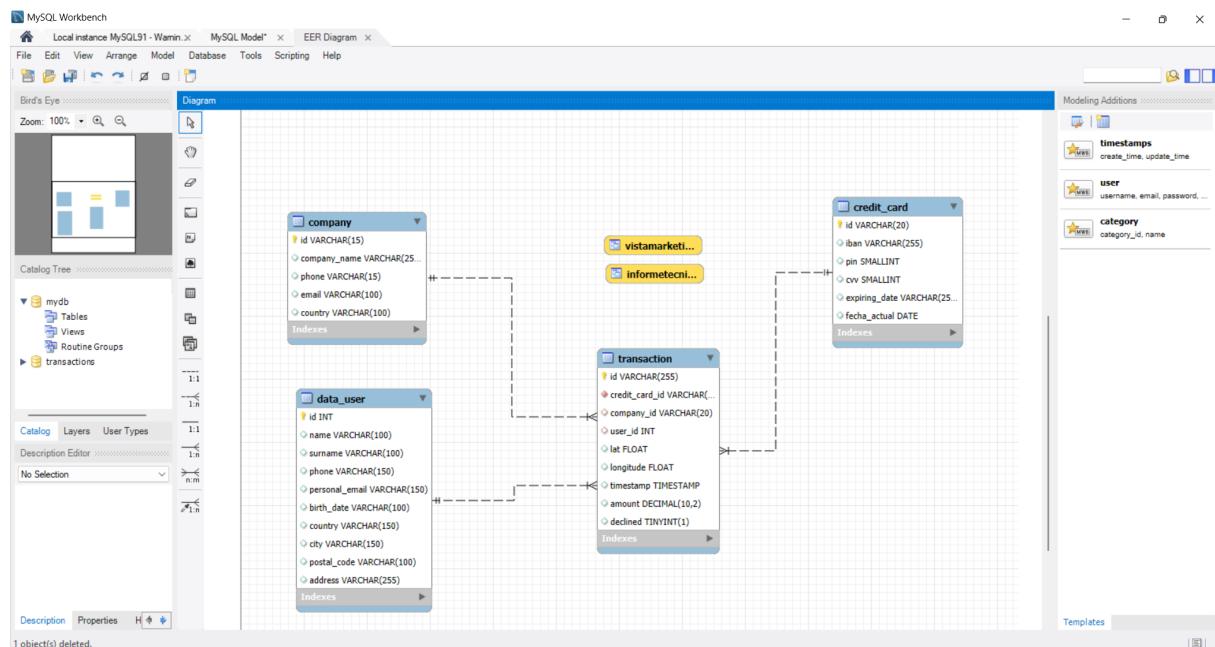


Figura 29: Nivell 3 Exercici 2. Visualització del diagrama E/R