

TASK S3.01. Manipulation of Tables

LEVEL 1

Exercise 1: Your task is to design and create a table called "credit_card" that stores crucial details about credit cards. The new table must be able to uniquely identify each card and establish an adequate relationship with the other two tables ("transaction" and "company"). After creating the table, it will be necessary to enter the information of the document called "data_enter_credit". Remember to show the diagram and make a brief description of it.

The screenshot shows two sessions in MySQL Workbench. The top session is titled '#LEVEL 1' and '#Exercise 1'. It displays the SQL code for creating the 'credit_card' table:

```
1  #LEVEL 1
2  #Exercise 1
3 • CREATE TABLE IF NOT EXISTS credit_card (
4      id VARCHAR(20) PRIMARY KEY,
5      iban VARCHAR(50),
6      pan VARCHAR(19),
7      pin VARCHAR(4),
8      cvv INT,
9      expiring_date VARCHAR(20)
10 );
11
12
```

The bottom session is titled 'Insertamos datos de credit_card' and shows the execution of 13 INSERT INTO statements to populate the 'credit_card' table with sample data. The response indicates 1 row(s) affected for each insert.

Action	Time	Response
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-2938', 'TR301950312213576817638661')	10:55:35	1 row(s) affected
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-2945', 'D02685476374853747521656868')	10:55:35	1 row(s) affected
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-2952', 'BG45IVQL52710525608255')	10:55:35	1 row(s) affected
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-2959', 'CRT242477244335841535')	10:55:35	1 row(s) affected
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-2966', 'BG72LKT015627628377363')	10:55:35	1 row(s) affected
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-2973', 'PT87806228135092429456346')	10:55:35	1 row(s) affected
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-2980', 'DE39241881883086277136')	10:55:35	1 row(s) affected
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-2987', 'GE89681434837748781813')	10:55:35	1 row(s) affected
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-2994', 'BH62714428368066765294')	10:55:35	1 row(s) affected
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-3001', 'CY4908742665477458126683211')	10:55:35	1 row(s) affected
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-3008', 'LU507216693616119230')	10:55:35	1 row(s) affected
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CCU-3015', 'PS1193982162957159683424568')	10:55:35	1 row(s) affected

For the relationship between the transaction and credit_card, I need to alter the transaction table.

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a text area containing SQL code. The code includes creating a table 'credit_card' with columns id, iban, pan, pin, cvv, and expiring_date, and then adding a foreign key constraint 'fk_transaction_credit_card' to the 'transaction' table, linking the 'credit_card_id' column to the 'id' column of 'credit_card'. A message at the bottom indicates 1 error found. Below the code is a 'Action Output' section with a table showing a single successful execution row.

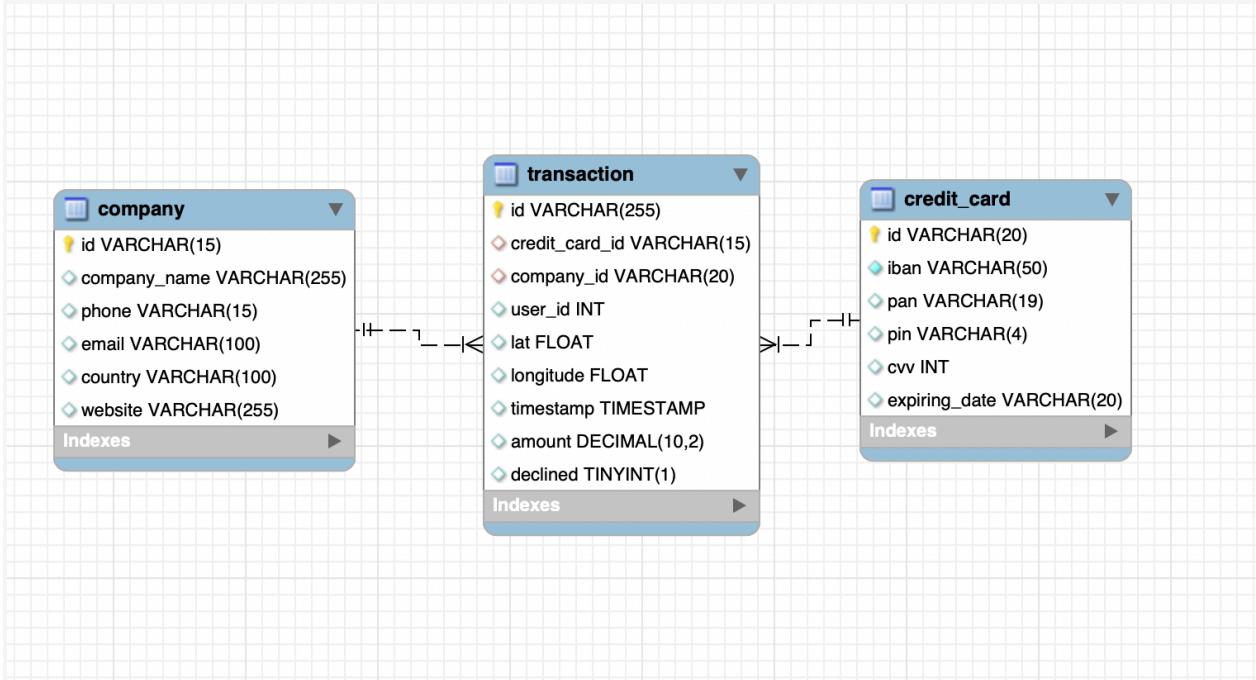
```

1  #LEVEL 1
2  #Exercise 1
3 • CREATE TABLE IF NOT EXISTS credit_card (
4      id VARCHAR(20) PRIMARY KEY,
5      iban VARCHAR(50) ,
6      pan VARCHAR(19),
7      pin VARCHAR(4),
8      cvv INT,
9      expiring_date VARCHAR(20)
10 );
11
12
13 • ALTER TABLE transaction
14   ADD CONSTRAINT fk_transaction_credit_card
15   FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
16
17
18
19
20
21

```

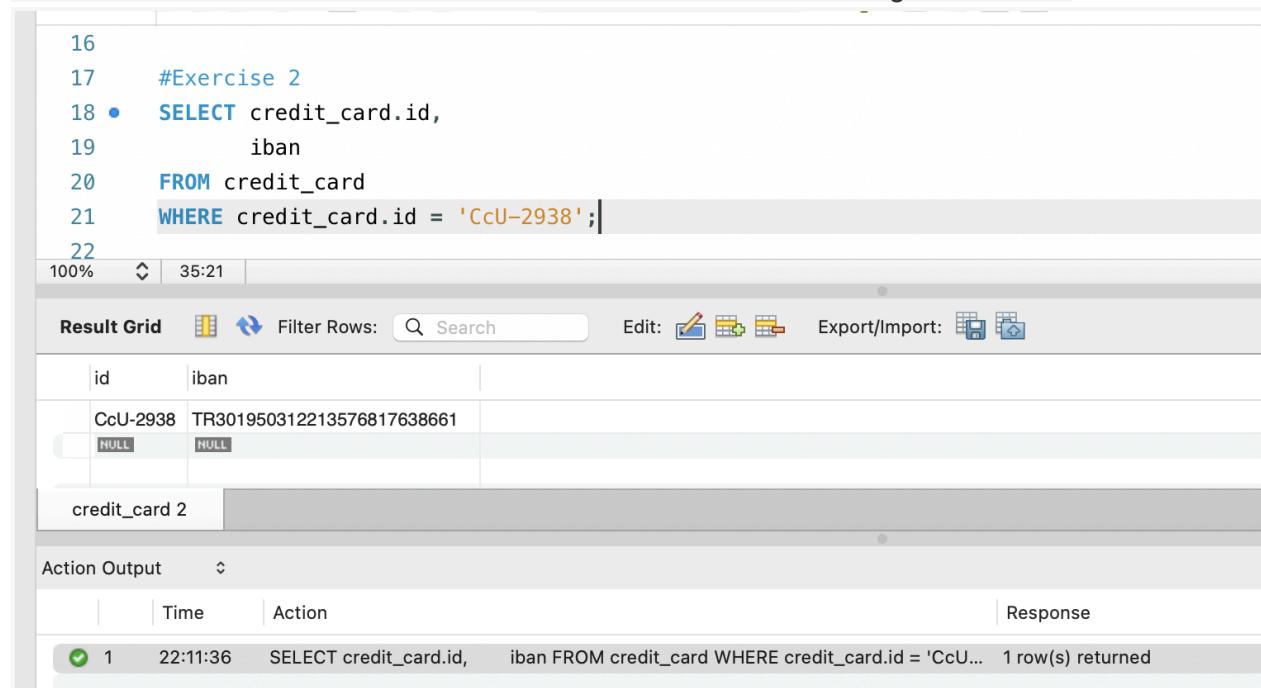
Action	Time	Action	Response
1	11:12:07	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_credit_c...	99999 row(s) affected Records: 99999 Duplicates: 0 Warnings: 0

And the diagram between the tables:



In the table company, company_id is the primary key; in the table credit_card, credit_card_id is the primary key and in the table transaction, transaction_id is the primary key, credit_card_id and company_id are the foreign keys.

Exercise 2: The Human Resources department has identified an error in the account number associated with the credit card with ID CcU-2938. The information to show for this record is: TR323456312213576817699999. Remember to show that the change was made.



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

```
16
17  #Exercise 2
18 •  SELECT credit_card.id,
19      iban
20  FROM credit_card
21  WHERE credit_card.id = 'CcU-2938';
22
```

The Result Grid pane displays the following data:

id	iban
CcU-2938	TR301950312213576817638661
NULL	NULL

The Action Output pane shows the following log entry:

Action	Time	Response
SELECT credit_card.id, iban FROM credit_card WHERE credit_card.id = 'CcU...'	22:11:36	1 row(s) returned

```

#Exercise 2
18 •  SELECT credit_card.id,
19      iban
20  FROM credit_card
21 WHERE credit_card.id = 'CcU-2938';
22
23 •  UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE credit_card.id = 'CcU-2938';
24
25

```

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

	id	iban
CcU-2938	NULL	TR323456312213576817699999
	NULL	NULL

credit_card 3

Action Output

	Time	Action	Response
1	22:11:36	SELECT credit_card.id, iban FROM credit_card WHERE credit_card.id = 'CcU...'	1 row(s) returned
2	22:14:39	UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE credit...	1 row(s) affected Rows matched: 1 Changed: 1
3	22:15:01	SELECT credit_card.id, iban FROM credit_card WHERE credit_card.id = 'CcU...'	1 row(s) returned

Exercise 3: In the table "transaction", enter a new user with the information.

```

26
27
28 #Exercise 3:
29 #INSERT INTO transaction (id,credit_card_id,company_id,user_id,lat,longitude,amount,timestamp,declined) VALUES ('108B1D1D-5B23-A76C-55E
30 #This code gave an error that's why I needed to insert company_id and credit_card_id to tables.
31 • INSERT INTO credit_card (id) VALUES ('CcU-9999');
32 • INSERT INTO company (id) VALUES ('b-9999');
33 • INSERT INTO transaction (id,credit_card_id,company_id,user_id,lat,longitude,amount,timestamp,declined)
34   VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD','CcU-9999','b-9999','9999','829.999','-117.999','111.11',current_timestamp,'0');
35

```

100% 23:3

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

id	credit_card...	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	2025-06-30 10:39:55	111.11	0
transaction 4								
Action Output								
	Time	Action					Response	
✓ 1	10:39:55	INSERT INTO transaction (id,credit_card_id,company_id,user_id,lat,longitude,am...					1 row(s) affected	
✓ 2	10:41:58	SELECT * FROM transaction WHERE transaction.id='108B1D1D-5B23-A76C-55EF...'					1 row(s) returned	

Exercise 4: From human resources, you are asked to remove the "pan" column from the credit_card table. Remember to show the change made.

filter objects

27 #Exercise 4:
28 • ALTER TABLE credit_card DROP COLUMN pan;
29
30
31
32

100% 1:30

Action Output

	Time	Action	Response
✓ 1	23:05:21	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

LEVEL 2

Exercise 1: Remove from the transaction table the record with ID000447FE-B650-4DCF-85DE-C7ED0EE1CAAD from the database.

```
30 #LEVEL 2
31 #Exercise 1:
32 • DELETE FROM transaction WHERE transaction.id= '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
33
34 • SELECT *
35 FROM transaction
36 WHERE transaction.id= '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
37
```

100% 63:36

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

id	credit_card...	company_id	user_id	lat	longitude	timestamp	amount	declined
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 5

Action Output

	Time	Action	Response
✓ 1	23:16:45	SELECT * FROM transaction WHERE transaction.id= '000447FE-B650-4DCF-85...	1 row(s) returned
✓ 2	23:17:48	DELETE FROM transaction WHERE transaction.id= '000447FE-B650-4DCF-85DE... 1 row(s) affected	
✓ 3	23:18:23	SELECT * FROM transaction WHERE transaction.id= '000447FE-B650-4DCF-85...	0 row(s) returned

Exercise 2: The marketing section wants to have access to specific information to carry out effective analysis and strategies. It has been requested to create a view that provides key details about companies and their transactions. You will need to create a view called VistaMarketing that contains the following information: Name of the company. Contact telephone. Country of residence. The average purchase made by each company. Presents the view created, sorting the data from highest to lowest average purchase.

Schemas

```

47
48
49 • CREATE VIEW marketing_view AS
50     SELECT c.company_name,
51             c.phone,
52             c.country,
53             AVG(t.amount) AS avg_sales
54     FROM company c
55     INNER JOIN transaction t
56     ON c.id = t.company_id
57     GROUP BY c.company_name,c.phone,c.country;
58
59
60 • SELECT *
61     FROM marketing_view
62     ORDER BY avg_sales DESC;
63

```

Result Grid

company_name	phone	country	avg_sales
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.867160
Prelium Neque Corp.	07 77 48 55 28	Australia	276.158330
Urna Convallis Associates	06 01 24 77 04	United States	274.235011
At Associates	09 56 61 10 65	New Zealand	272.214870
Metus Vitae Associates	08 25 44 40 66	Australia	270.080965
Aliquet Diam Limited	02 76 61 47 46	United States	269.599181
Nec Luctus LLC	02 14 71 75 73	Norway	268.604837
Neque Tellus Incorporated	04 43 18 34 19	Ireland	267.850372
Tortor Nunc Commodo Company	05 35 92 77 16	United States	267.836085
Cras Consulting	07 50 10 85 63	Belgium	267.439409
Fringilla Insumi In	03 67 44 56.72	United States	267.060541

Action Output

Time	Action	Response
11:19:27	CREATE VIEW marketing_view AS SELECT c.company_name, c.phone, c....	0 row(s) affected
11:19:35	SELECT * FROM marketing_view ORDER BY avg_sales DESC LIMIT 0, 1000	101 row(s) returned

Exercise 3: Filter Marketing to show only companies that have their country of residence in "Germany".

```

64    #Exercise 3:
65 • SELECT *
66     FROM marketing_view
67     WHERE country = 'Germany';
68

```

Result Grid

company_name	phone	country	avg_sales
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.867160
Convallis In Incorporated	06 66 57 29 50	Germany	257.745376
Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.319156
Augue Foundation	06 88 43 15 63	Germany	253.505000
Ac Industries	09 34 65 40 60	Germany	255.147288
Auctor Mauris Corp.	05 62 87 14 41	Germany	254.765518
Aliquam PC	01 45 73 52 16	Germany	253.136923
Rutrum Non Inc.	02 66 31 61 09	Germany	255.136927

Action Output

Time	Action	Response
11:27:50	SELECT * FROM marketing_view WHERE country = 'Germany' LIMIT 0, 1000	8 row(s) returned

LEVEL 3

Exercise 1: Next week, you will have a new meeting with marketing managers. A colleague of your team made modifications to the database, but does not remember how it made them. He asks you to help him leave the commands executed to obtain the diagram.

-First, create the table user and insert the data:

The screenshot shows a database interface with a left sidebar containing 'schemas' like 'sys' and 'transactions'. Under 'transactions', there's a 'Tables' section with 'company', 'credit_card', 'transaction', and 'user'. The 'user' table is selected and highlighted with a blue border. In the main pane, a SQL code editor displays the creation of the 'user' table:

```

CREATE TABLE IF NOT EXISTS user (
    id CHAR(10) PRIMARY KEY,
    name VARCHAR(100),
    surname VARCHAR(100),
    phone VARCHAR(150),
    email VARCHAR(150),
    birth_date VARCHAR(100),
    country VARCHAR(150),
    city VARCHAR(150),
    postal_code VARCHAR(100),
    address VARCHAR(255)
);

```

Below the code editor is an 'Action Output' table showing a single log entry:

Action	Time	Response
CREATE TABLE IF NOT EXISTS user (id CHAR(10) PRIMARY KEY, name VARCHAR...	11:43:49	0 row(s) affected

The screenshot shows the same database interface. The 'user' table is now expanded to show its columns: 'id', 'name', 'surname', 'phone', 'email', 'birth_date', 'country', 'city', 'postal_code', and 'address'. Below the table, the 'Action Output' table shows 19 rows of INSERT statements, each with a timestamp from '22:46:15' to '22:46:17' and a response of '1 row(s) affected'.

Action	Time	Response
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected
INSERT INTO user (id, name, surname, phone, email, birth_date, coun...	22:46:15	1 row(s) affected

- To establish the relationship between the transaction table and the user table, we need to use a foreign key constraint on user_id. But first we need to add also user_id = 9999. Because we added it before.

```

83
84
85
86
87 • INSERT INTO user(id) VALUES('9999');
88 • ALTER TABLE transaction
89   ADD CONSTRAINT fk_transaction_user
90   FOREIGN KEY (user_id) REFERENCES user(id);
91
92
93
94
95
```

Action Output

	Time	Action	Response
✓ 1	11:17:23	RENAME TABLE data_user to user	0 row(s) affected
✓ 2	11:20:42	INSERT INTO user(id) VALUES('9999')	1 row(s) affected
✓ 3	11:22:05	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_user FO...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0

- Change the name of the table user to data_user and the data type of id column:

-Add a column as fecha_actual to the table credit_card:

CHEMANS

Filter objects

- > Foreign Keys
- > Triggers
- credit_card**
 - Columns
 - id
 - iban
 - pin
 - cvv
 - expiring_date
 - fecha_actual
 - Indexes
 - Foreign Keys
 - Triggers
- data_user**
 - Columns
 - Indexes
 - Foreign Keys

89

```

90
91
92
93
94 #Change the name of the table user to data_user and the data type of id column:
95 • RENAME TABLE user TO data_user;
96 • ALTER TABLE data_user
97   MODIFY COLUMN id INT;
98
99 #Add the fecha_actual column to the credit_card table:
100 • ALTER TABLE credit_card ADD fecha_actual DATE;
```

Action Output

	Time	Action	Response
✓ 1	11:51:25	ALTER TABLE credit_card ADD fecha_actual DATE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

-Delete the website column from the table company:

SCHEMAS

Filter objects

- > sys
- > transactions
- Tables**
 - company**
 - Columns
 - id
 - company_name
 - phone
 - email
 - country
 - Indexes

90

```

91 #Delete website column from the company table :
92 • ALTER TABLE company DROP COLUMN website;
```

Action Output

	Time	Action	Response
✓ 1	12:21:10	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

-In the data_user table change the name of email to personnel_email:

The screenshot shows the MySQL Workbench interface. In the left sidebar, under 'data_user' columns, there is a list of fields: id, name, surname, phone, personal_email, birth_date, country, city, postal_code, and address. In the main query editor area, the following SQL code is displayed:

```
105
106
107  #In the data_user table change the name of email to personnel_email
108 • ALTER TABLE data_user CHANGE email personal_email VARCHAR(150);
109
110
111
```

Below the code, the 'Action Output' section shows the results of the execution:

Action	Time	Response
1	12:04:33	ALTER TABLE data_user CHANGE email personal_email VARCHAR(150) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

-To establish the relationship between the transaction table and the data_user table, we need to use a foreign key constraint on user_id:

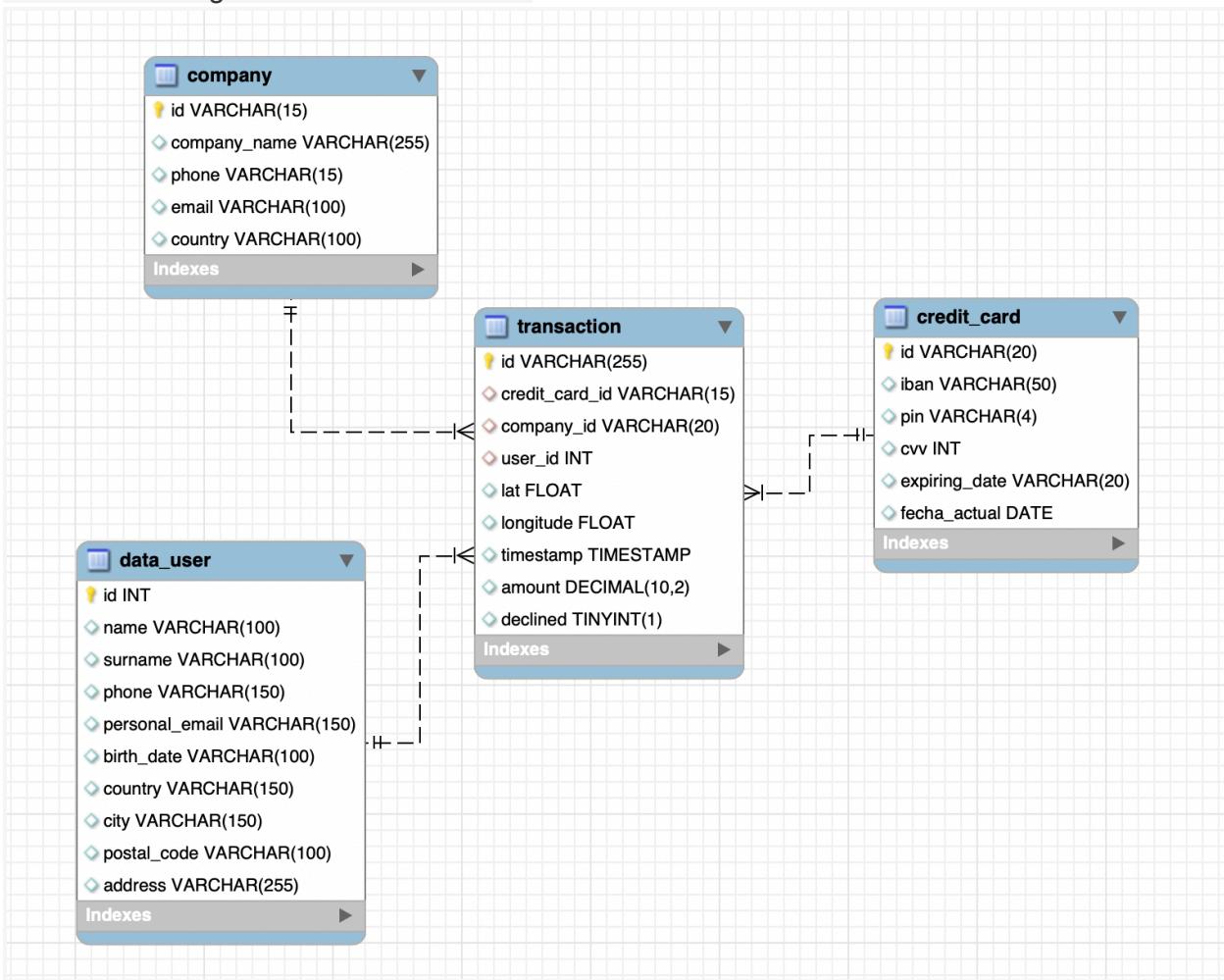
The screenshot shows the MySQL Workbench interface. In the main query editor area, the following SQL code is displayed:

```
104 •
105
106
107
108  #To establish the relationship between the transaction table and the data_user table, we need to use a fo
109 • ALTER TABLE transaction
110  DROP FOREIGN KEY fk_transaction_user;
111
112 • ALTER TABLE transaction
113  ADD CONSTRAINT fk_transaction_data_user
114  FOREIGN KEY (user_id) REFERENCES data_user(id);
```

Below the code, the 'Action Output' section shows the results of the execution:

Action	Time	Response
1	11:51:25	ALTER TABLE credit_card ADD fecha_actual DATE 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
2	11:54:43	ALTER TABLE transaction DROP FOREIGN KEY fk_transaction_user 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3	11:54:53	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_data_us... 100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0

And the final diagram between the tables:



Exercise 2: The company also asks you to create a view called "Technical Report" containing the following information:

- Transaction ID
- Username
- User Surname
- IBAN of the used credit card.
- Name of the company of the transaction made.

Make sure to include relevant information about the tables you will know and use alias to rename columns as needed.

Displays the results of the view, sorts the results downward based on the transaction ID variable.

transaction

```

121
122 #Exercise 2:
123 #Create a view called technical_report:
124 • CREATE VIEW technical_report AS
125   SELECT t.id AS transaction_id,
126         d.name AS user_name,
127         d.surname AS user_surname,
128         cc.iban,
129         c.company_name
130   FROM transaction_t
131   INNER JOIN data_user d
132   ON t.user_id = d.id
133   INNER JOIN company c
134   ON t.company_id = c.id
135   INNER JOIN credit_card cc
136   ON t.credit_card_id = cc.id
137   GROUP BY transaction_id,user_name,user_surname,cc.iban,c.company_name;
138
139
140
141
142

```

Action Output

Time	Action	Response
12:16:24	CREATE VIEW technical_report AS SELECT t.id AS transaction_id,...	Error Code: 1054. Unknown column 'd.surname' in 'group
12:17:48	CREATE VIEW technical_report AS SELECT t.id AS transaction_id,...	0 row(s) affected

transaction

```

138
139 #Displays the results of the view, sorts the results downward based on the transaction ID variable
140 • SELECT *,
141   ROW_NUMBER() OVER (ORDER BY transaction_id DESC) AS row_num
142   FROM technical_report;
143
144
145
146
147

```

Result Grid

transaction_id	user_name	user_surname	iban	company_name	row_num
FFFD31D6-9495-47CE-B54A-7DB8E1CC274B	Bmrgli	Tprvrmrc	XX794814451211289182490922	Turpis Company	1
FFFFCF76D-E0F0-4985-A2D0-B2A7B75989FC	Dlfled	VlglcjdI	XX636251701647892036876034	Amet Nulla Donec Corporation	2
FFFCE9EB0-27C7-4ADE-98F2-7533EF4DF126	Securp	FaoIvqfy	XX162677143304223631437567	Nunc Interdum Incorporated	3
FFFBB270D-F53A-4D5D-9666-E5307C53C84	Gzjipa	Uirzjulh	XX395114267082019952567052	Viverra Donec Foundation	4
FFF9E3CE-234E-408C-A8EF-F9CAD577224A	Yshimr	Zpsjsleed	XX8845462156537570367941	Convallis In Incorporated	5
FFF9E178-8CD2-4DF9-99B0-49AE068809B1	Jevpx	Xwczwznm	XX321405515711654384711481	Mus Aenean Eget Foundation	6
FFF867C9-17B5-4B1F-AFD9-F8023AAA449E	Fqlngd	LvhfqlxI	XX278446342932680979729426	Cras Vehicula Aliquet Industries	7
FFF7042D-18C6-4DDD-823C-4D90A4AC8F26	Njraa	Egsquili	XX405009272572550082027209	Placerat LLP	8
FFF660DA-4244-47F6-9210-ESD1DCB99D80	Lopzaj	Itgrfay	XX63376659736627454015125	Pede Cum Ltd	9
FFF5C660-4441-436D-BD27-EBC53B618622	Gmnbru	Oxdvnkll	XX237820256172646394016483	At Associates	10
FFF54F54-B439-41F0-BD2D-F732DC1ACAD	Gqcfy	Mplfrln	XX802723943240147612158718	Enim Condimentum Ltd	11
FFF42F7D-7A0D-4E2-AF72-5996903F8AA9	Ddkuqg	Ycsbpnyu	XX92644230155195974199541	At Pede Corp.	12
FFF42620-1968-4A1F-B9D7-27843CB097A5	Sbjrvp	Eiqnzdjf	XX395457232638959102336873	Amet Nulla Donec Corporation	13
FFF2900EB-79D2-497C-BDCE-2EC63FE3E9AD	Wcfzys	Dkktghfd	XX45898904822223071751935	Integer Mollis Corp.	14

Action Output

Time	Action	Response
12:24:30	SELECT *, ROW_NUMBER() OVER (ORDER BY transaction_id DESC)...	100000 row(s) returned