Task 3.01.

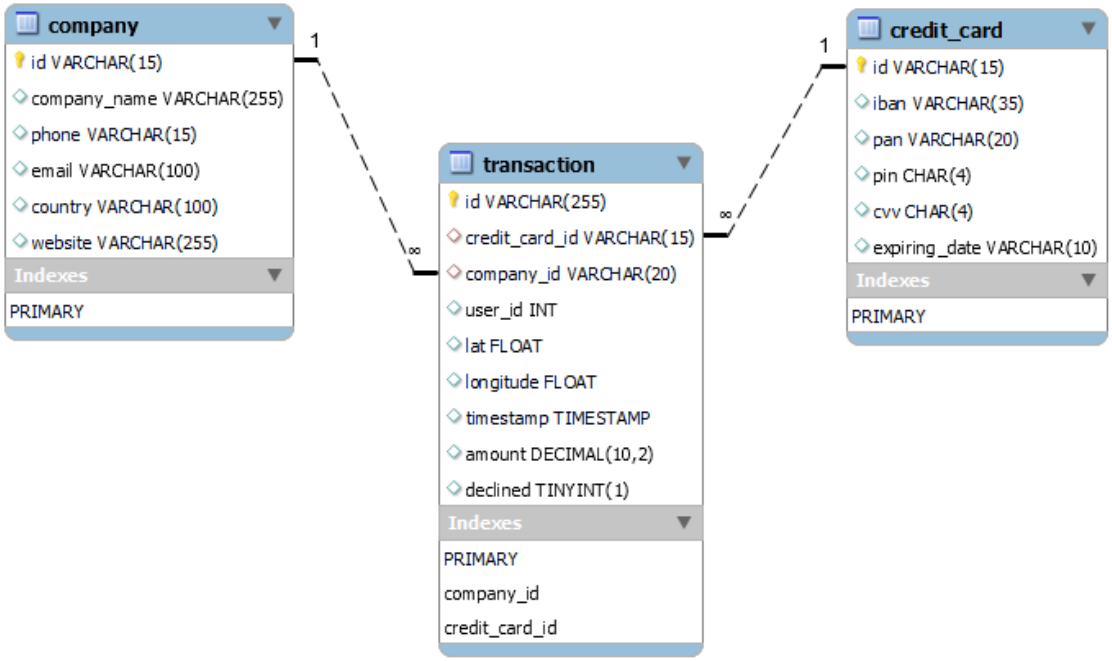
Level 1.

Exercise1.

Database '*transactions*' contains information from a company that sells products online.

The table *´credit_card´* was created inside the database '*transactions*' and populated with data from the provided file *´dades_introduir_credit.sql´* in MySQL Workbench.

ER diagram



Database consists from three tables (tables described in Sprint2 are denoted in grey):

- *´company´* - corporate information of the companies
  All variables are strings of different length
  Primary key - *´id´* - uniquely identify each company
- *´transaction´* - data related to the transactions performed by companies
  Variables *id, credit_card_id* and *company_id* are strings, *user_id* and *declined* have type integer, *lat* and *longitude* are floats, *amount* has decimal format (with up to 10 digits in total and 2 decimal places), and *timestamp* has datetime format.

Primary key - *´id´* - uniquely identify each transaction
Foreign key - *´company_id´* - links this table with table *´company´* (column *´id´*), links each transaction to the company that made the transaction.
Foreign key - *´credit_card_id´* - links this table with table *´credit_card´* (column *´id´*), links each transaction to the credit card used in that transaction. Each credit was used at least in one transaction.
Also *´user_id´* could be foreign key for table like *´user´* if existed.

- *´credit_card´* - stores crucial details about credit cards:
  - *id* – string allowing up to 15 characters (the same as transaction.credit_card_id)
  - *iban* - the International Bank Account Number, string allowing up to 35 characters (IBAN has up to 34 characters)
  - *pan* - the Primary Account Number, string allowing up to 20 characters (PAN can contain up to 19 digits)
  - *pin* – the personal Identification Number (usually 4-digit), fixed-length string up to 4 characters.
  - *cvv* - the Card Verification Value, which can be either 3 or 4 digits (American Express cards), fixed-length string up to 4 characters.
  - *expiring_date* - a date when the card expires, string up to 10 characters.

  Primary key - *´id´* - uniquely identify each credit card.

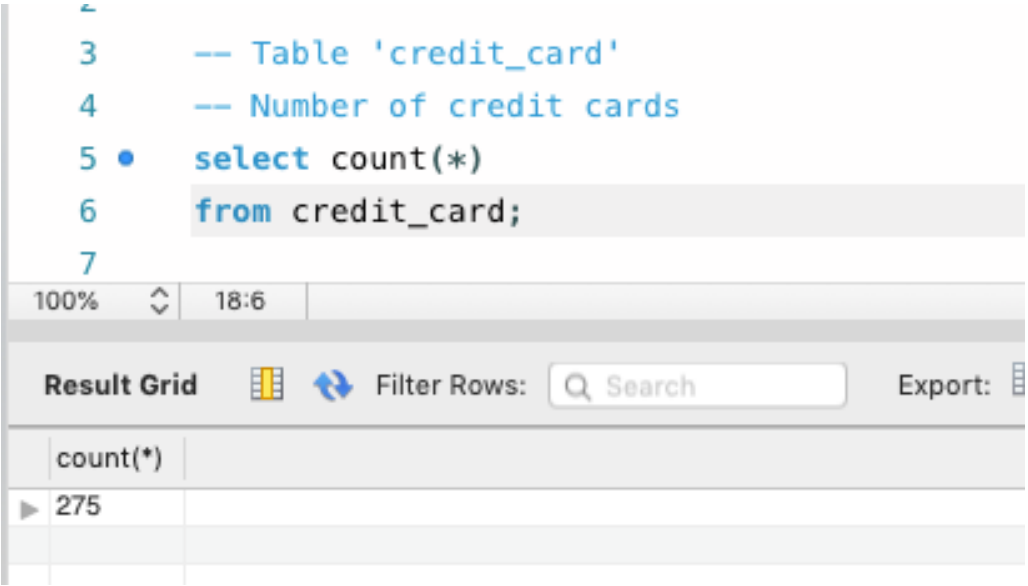
Tables ´company´ and ´credit_card´ have non-identifying relationships (one-to-many) to ´transaction´ table.

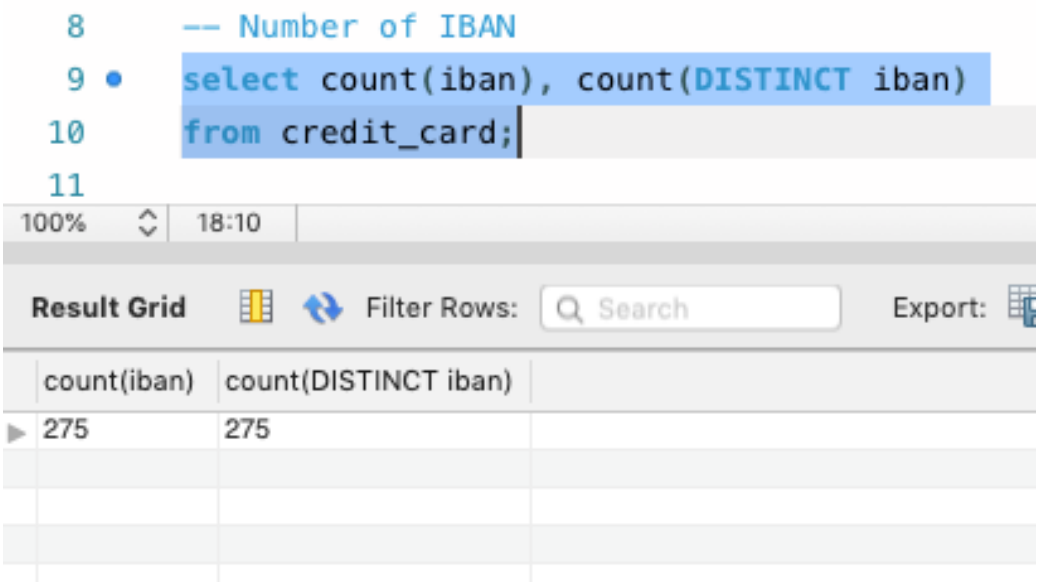
Tables overview

The overviews of tables *´transaction´* and *´company´* were provided in the previous task.

´credit_card´

There are 275 different credit cards (users?). All credit cards have different IBAN.





There are no missing values in credit card details.

```
12    -- Number of non-missing values
13 •  select count(iban), count(pan), count(pin), count(cvv), count(expiring_date)
14    from credit_card;
15
```

100% 32:12

Result Grid Filter Rows: Search Export:

count(iban)	count(pan)	count(pin)	count(cvv)	count(expiring_date)
275	275	275	275	275

All credit cards from `credit_card` table were used in transactions.

```
16    -- check if all credit cards were used
17 •  select *
18    from credit_card
19    where id not IN (
20    select credit_card_id from transaction);
21
```

100% 9:17

Result Grid Filter Rows: Search Edit:

id	iban	pan	pin	cvv	expiring_date
NULL	NULL	NULL	NULL	NULL	NULL

For all transactions there are credit card details.

```
22    -- check if all transactions have credit_card info
23 •  select *
24    from transaction
25    where credit_card_id not IN (
26    select id from credit_card);
27
```

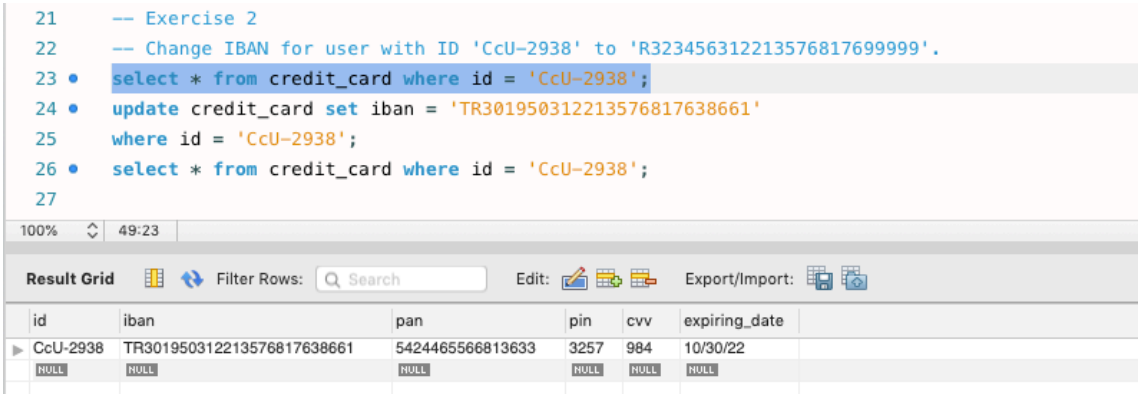
100% 29:26

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

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Exercise 2.
Change IBAN of user with credit card id ´CcU-2938´ to ´R323456312213576817699999´.

Check if `credit_card` table has credit card with id ´CcU-2938´

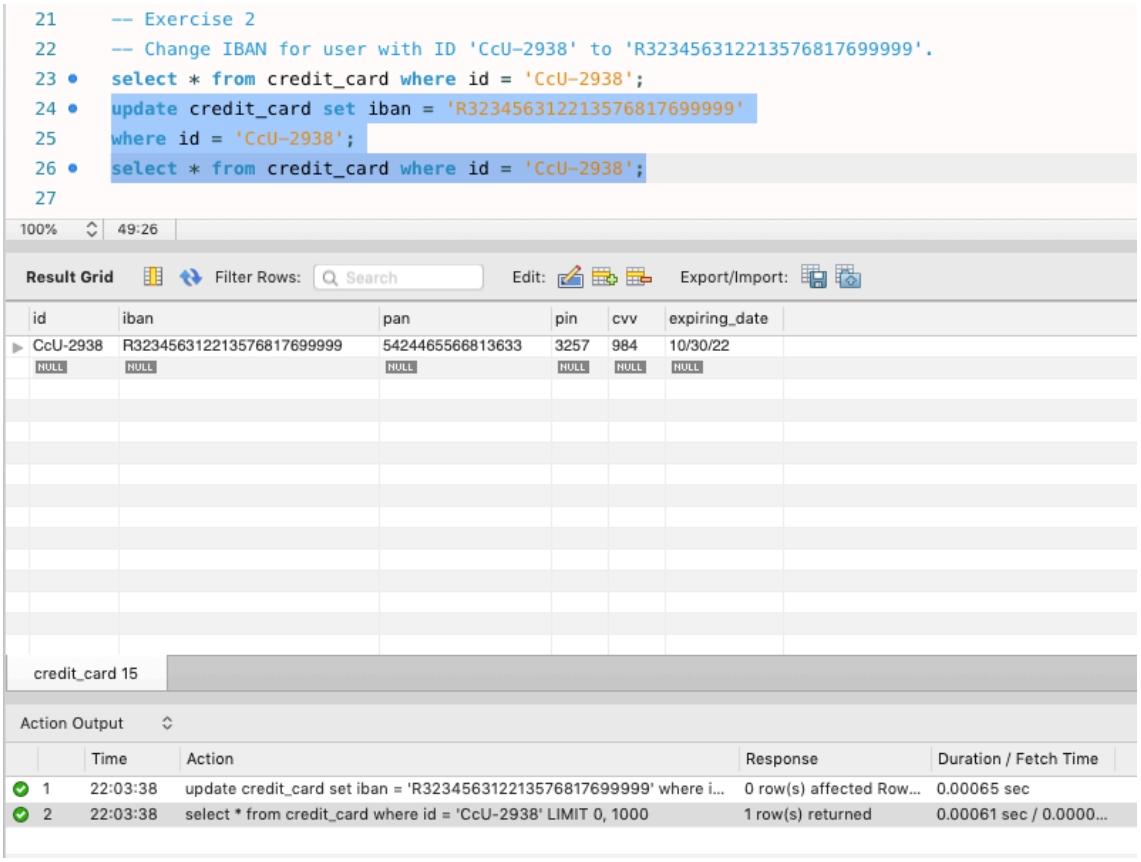


```
21    -- Exercise 2
22    -- Change IBAN for user with ID 'CcU-2938' to 'R323456312213576817699999'.
23 •  select * from credit_card where id = 'CcU-2938';
24 •  update credit_card set iban = 'TR3019503122135768176386 61'
25    where id = 'CcU-2938';
26 •  select * from credit_card where id = 'CcU-2938';
27
```

100% 49:23

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

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR3019503122135768176386 61	5424465566813633	3257	984	10/30/22
NULL	NULL	NULL	NULL	NULL	NULL



```
21    -- Exercise 2
22    -- Change IBAN for user with ID 'CcU-2938' to 'R323456312213576817699999'.
23 •  select * from credit_card where id = 'CcU-2938';
24 •  update credit_card set iban = 'R323456312213576817699999'
25    where id = 'CcU-2938';
26 •  select * from credit_card where id = 'CcU-2938';
27
```

100% 49:26

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

id	iban	pan	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	5424465566813633	3257	984	10/30/22
NULL	NULL	NULL	NULL	NULL	NULL

credit_card 15

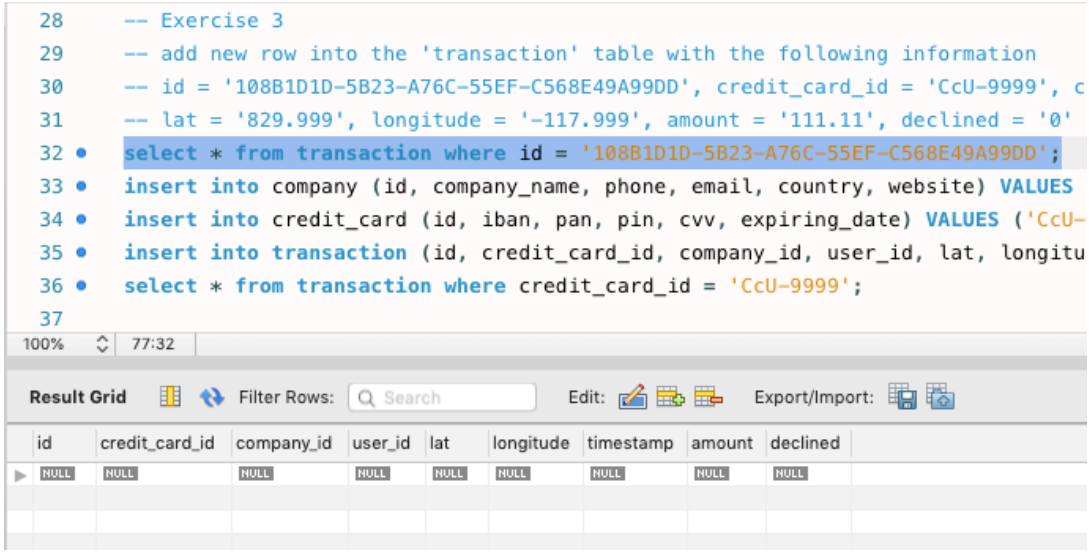
Action Output

	Time	Action	Response	Duration / Fetch Time
1	22:03:38	update credit_card set iban = 'R323456312213576817699999' where i...	0 row(s) affected Row...	0.00065 sec
2	22:03:38	select * from credit_card where id = 'CcU-2938' LIMIT 0, 1000	1 row(s) returned	0.00061 sec / 0.0000...

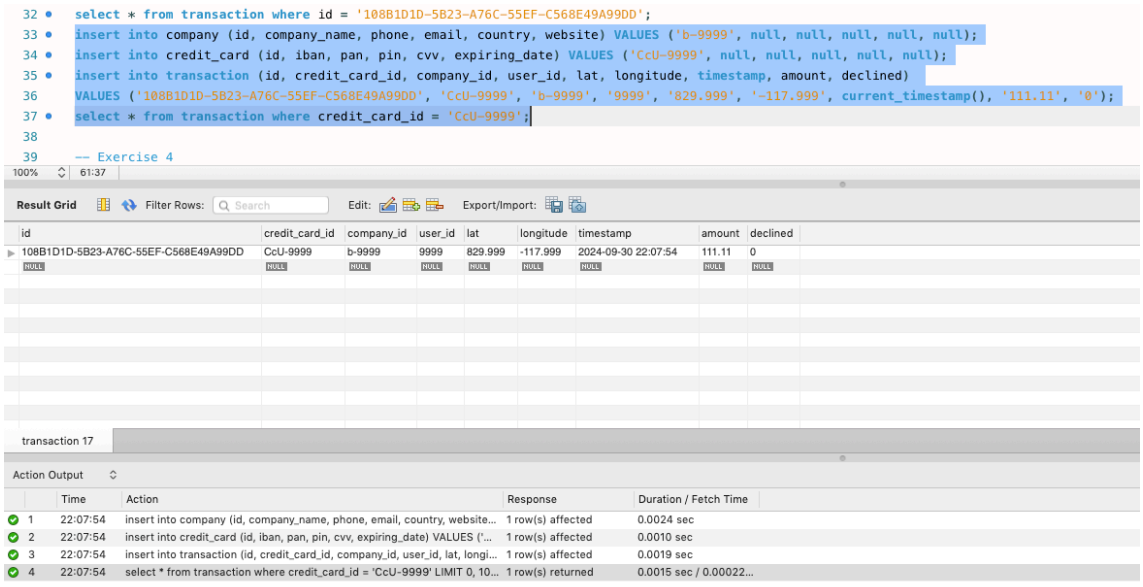
Exercise 3.
Add a new transaction into the table `transaction` with the following information:

`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

Check if `transaction` table has `id` ´108B1D1D-5B23-A76C-55EF-C568E49A99DD´



Add new row:



*** Since transaction time information was not provided, a null value was used.*

The `´transaction´` table is linked to the tables `´company´` and `´credit_card´`, which do not contain this new information. To maintain database integrity, rows with the new `credit_card_id` and `company_id` were added to the tables `credit_card` and `company`, respectively. Transaction cannot be performed without credit card and user, it is necessary to get data for user "9999" and credit card "CcU-9999", at the moment null values are used.

Exercise 4.
Remove the *´pan´* column from the *´credit_card´* table.

```
39    -- Exercise 4
40    -- delete the "pan" column from the credit_card table
41 •  alter table credit_card drop column pan;
42 •  describe credit_card;
```

100% 41:41

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	NULL	
iban	varchar(35)	YES		NULL	
pin	char(4)	YES		NULL	
cvv	char(4)	YES		NULL	
expiring_date	varchar(10)	YES		NULL	

Result 23

Action Output

	Time	Action	Response	Duration / Fetch Time
1	22:13:07	alter table credit_card drop column pan	0 row(s) affected Rec...	0.132 sec
2	22:13:07	describe credit_card	5 row(s) returned	0.0015 sec / 0.00001...

Level 2.

Exercise1.

Remove the record with ID ´02C6201E-D90A-1859-B4EE-88D2986D3B02´ from the table ´*transaction*´.

Check if `transaction` table has `id` 02C6201E-D90A-1859-B4EE-88D2986D3B02

```
44    -- Level 2
45    -- Exercise 1
46    -- Remove from the table 'transaction' the record with ID 02C6201E-D90A-1859-B4EE-88D2986D3B02
47 •  select * from transaction where id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
48 •  select * from transaction where credit_card_id = (select credit_card_id from transaction where id='02
49 •  select * from transaction where company_id = (select company_id from transaction where id='02C6201E-
50
```

100% 1:48

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

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	92	81.9185	-12.5276	2021-08-28 23:42:24	466.92	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Check if there are any other transactions with the same credit card

```
44    -- Level 2
45    -- Exercise 1
46    -- Remove from the table 'transaction' the record with ID 02C6201E-D90A-1859-B4EE-88D2986D3B02
47 •  select * from transaction where id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
48 •  select * from transaction where credit_card_id = (select credit_card_id from transaction where id='02C6201E-D90A-1859-B4EE-88D2986D3B02');
49 •  select * from transaction where company_id = (select company_id from transaction where id='02C6201E-D90A-1859-B4EE-88D2986D3B02'); -- if t
50
```

100% 140:48

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

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
17561134-37A4-E316-2C3C-C5541EDAD761	CcU-2938	b-2362	92	-10.8529	18.0969	2021-09-15 05:23:32	371.57	0
185C9F2C-67A2-8EC0-0BB6-93789CC59F05	CcU-2938	b-2362	92	9.55928	165.726	2021-07-25 12:34:59	285.24	0
18B2472B-DA8B-36F7-B0F2-7DDA688B73D1	CcU-2938	b-2362	87	-41.7722	98.2897	2021-09-24 08:33:44	391.38	0
26D8A738-B651-A620-1EEA-D8E2CD353BEB	CcU-2938	b-2362	92	15.6872	25.2507	2021-05-28 15:21:36	238.99	0
4DA5B1CE-E88C-23D1-AE6A-B7589E4FB2D5	CcU-2938	b-2362	82	39.2187	63.5563	2021-03-28 05:01:44	378.86	0
55166D02-D74C-6A63-6C54-8678467649B4	CcU-2938	b-2362	89	-31.517	-168.575	2022-02-24 11:01:42	186.12	0
644264C8-A268-C3C5-5929-958C49CBC926	CcU-2938	b-2362	92	20.9826	25.7996	2021-04-18 11:05:12	34.55	0
6575E457-EF8B-B50A-AD85-68ED6FE98BE1	CcU-2938	b-2362	92	-79.9882	-45.7529	2021-10-24 01:29:53	297.16	0
85068503-9D65-6978-D3C4-8E4A8A1B9A3C	CcU-2938	b-2362	85	82.348	-20.5866	2021-04-01 07:27:49	306.94	0
87D22108-C911-AC97-A3BE-B91C1DEEAE79	CcU-2938	b-2362	92	-25.6402	-74.192	2021-03-23 01:12:06	415.42	0
935EBFEE-B75C-6C68-4AC3-61351035C549	CcU-2938	b-2362	80	5.33058	-15.6521	2021-07-18 08:20:59	20.44	0

transaction 49 | Apply

Action Output

	Time	Action	Response	Duration / Fetch Time
1	13:11:13	select * from transaction where credit_card_id = (select credit_card_id from tra...	24 row(s) returned	0.00087 sec / 0.0000...

Check if there are any other transactions with the same company

```
44    -- Level 2
45    -- Exercise 1
46    -- Remove from the table 'transaction' the record with ID 02C6201E-D90A-1859-B4EE-88D2986D3B02
47 •  select * from transaction where id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
48 •  select * from transaction where credit_card_id = (select credit_card_id from transaction where id='02C6201E-D90A-1859-B4EE-88D2986D3B02');
49 •  select * from transaction where company_id = (select company_id from transaction where id='02C6201E-D90A-1859-B4EE-88D2986D3B02'); -- if t
50
```

100% 131:49

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

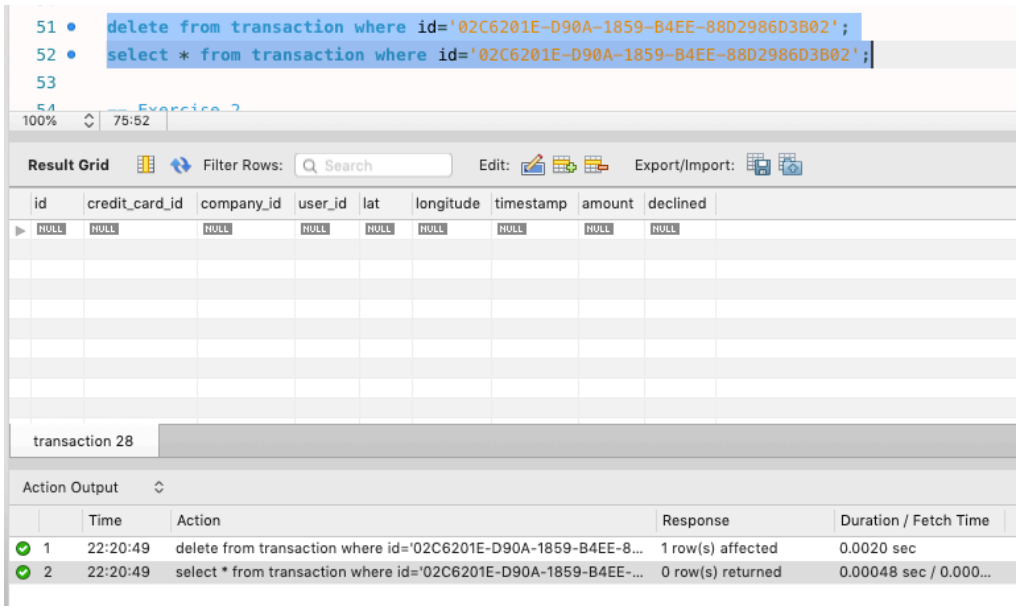
id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
17561134-37A4-E316-2C3C-C5541EDAD761	CcU-2938	b-2362	92	-10.8529	18.0969	2021-09-15 05:23:32	371.57	0
185C9F2C-67A2-8EC0-0BB6-93789CC59F05	CcU-2938	b-2362	92	9.55928	165.726	2021-07-25 12:34:59	285.24	0
18B1AD88-F155-45B7-D2EA-4738407435D6	CcU-3981	b-2362	65	36.2908	-128.821	2021-11-26 20:26:07	268.48	0
18B2472B-DA8B-36F7-B0F2-7DDA688B73D1	CcU-2938	b-2362	87	-41.7722	98.2897	2021-09-24 08:33:44	391.38	0
1998A483-B4CD-FAE6-745C-8F2F2EB77A8E	CcU-2959	b-2362	92	-6.77047	128.679	2021-08-29 10:53:03	372.58	0
2660D3B8-0B2C-AEBA-A891-95433968CF71	CcU-3981	b-2362	66	38.6937	1.87889	2021-12-01 02:29:51	381.17	0
26D8A738-B651-A620-1EEA-D8E2CD353BEB	CcU-2938	b-2362	92	15.6872	25.2507	2021-05-28 15:21:36	238.99	0
2A37F5B3-11D8-3BA4-6E94-B7407D91942C	CcU-2959	b-2362	92	-81.8177	-65.964	2022-01-10 20:31:41	83.19	0
3EB67ADC-B4F1-83C4-93E2-E8469D1A37F6	CcU-3981	b-2362	77	-64.6374	-99.9424	2021-12-23 20:25:16	33.10	0
44D0BBB1-975E-86AA-75DA-BDAFED65F368	CcU-4093	b-2362	62	34.5967	161.593	2021-07-20 08:47:44	133.39	0
4DA5B1CE-E88C-23D1-AE6A-B7589E4FB2D5	CcU-2938	b-2362	82	39.2187	63.5563	2021-03-28 05:01:44	378.86	0

transaction 50 | Apply

Action Output

	Time	Action	Response	Duration / Fetch Time
1	13:13:20	select * from transaction where company_id = (select company_id from transa...	54 row(s) returned	0.0012 sec / 0.00004...

Remove the record



Exercise2.

Create a view *´VistaMarketing´* that provides key details about companies and their transactions: company name, phone number, country of residence, average purchase made by each company. Present the created view, sorting the data from highest to lowest purchase average.

```
54    -- Exercise 2
55    -- Create a veiw VistaMarketing that contains the following information:
56    -- company name, phone number, country, average purchase made by each company.
57    -- Present the created view, sorting the data from highest to lowest purchase average.
58    create view VistaMarketing as
59    select c.company_name, c.phone, c.country, round(AVG(t.amount), 2) as avg_purchase
60    from company as c
61    join transaction as t on c.id=t.company_id
62    where t.declined = 0 and c.company_name is not null
63    group by c.id;
64
65    select * from VistaMarketing
66    order by avg_purchase desc;
67
```

100% 15:63

Result Grid Filter Rows: Search Export:

company_name	phone	country	avg_purchase
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
Maecenas Malesuada Fringilla Inc.	09 38 53 76 61	Netherlands	451.29
Erat LLP	03 18 88 77 79	Netherlands	448.44
Tortor Nunc Commodo Company	05 35 92 77 16	United States	447.11
Justo Eu Arcu Ltd	08 42 56 71 52	Italy	444.16
Pede Cum Ltd	07 62 26 48 38	Norway	442.32
Vestibulum Lorem PC	02 02 87 33 40	Belgium	428.40
Mauris Institute	05 29 60 36 87	Sweden	427.71
Aliquet Diam Limited	02 76 61 47 46	United States	425.64
Mus Aenean Eget Foundation	06 25 15 52 43	Sweden	419.97
Sed LLC	01 63 16 26 52	Belgium	416.66
Viverra Donec Foundation	03 33 12 32 73	United Kingdom	414.53
Eget Tincidunt Dui Institute	05 35 93 32 44	Netherlands	413.50
Amet Institute	06 23 40 21 33	Australia	412.48

VistaMarketing 33

Action Output

	Time	Action	Response	Duration / Fetch Time
1	22:28:53	create view VistaMarketing as select c.company_name, c.phone, c....	0 row(s) affected	0.027 sec
2	22:28:53	select * from VistaMarketing order by avg_purchase desc LIMIT 0,...	100 row(s) returned	0.042 sec / 0.00016...

*** Only successful transactions were considered as purchases, since if a transaction is declined, the purchase cannot be considered completed.*

Companies without names are not included in the View, since in this case their contact information is also missing.

Exercise3.
Companies from ´*VistaMarketing*´ that have their country of residence in "Germany".

```
68    -- Exercise 3
69    -- Companies from 'VistaMarketing' that have their country of residence in "Germany".
70 •  select * from VistaMarketing
71    where country = 'Germany';
72
```

100% 46:69

Result Grid Filter Rows: Search Export:

company_name	phone	country	avg_purchase
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 66 31 61 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
Auctor Mauris Corp.	05 62 87 14 41	Germany	308.99
Augue Foundation	06 88 43 15 63	Germany	15.05
Aliquam PC	01 45 73 52 16	Germany	280.34

VistaMarketing 35

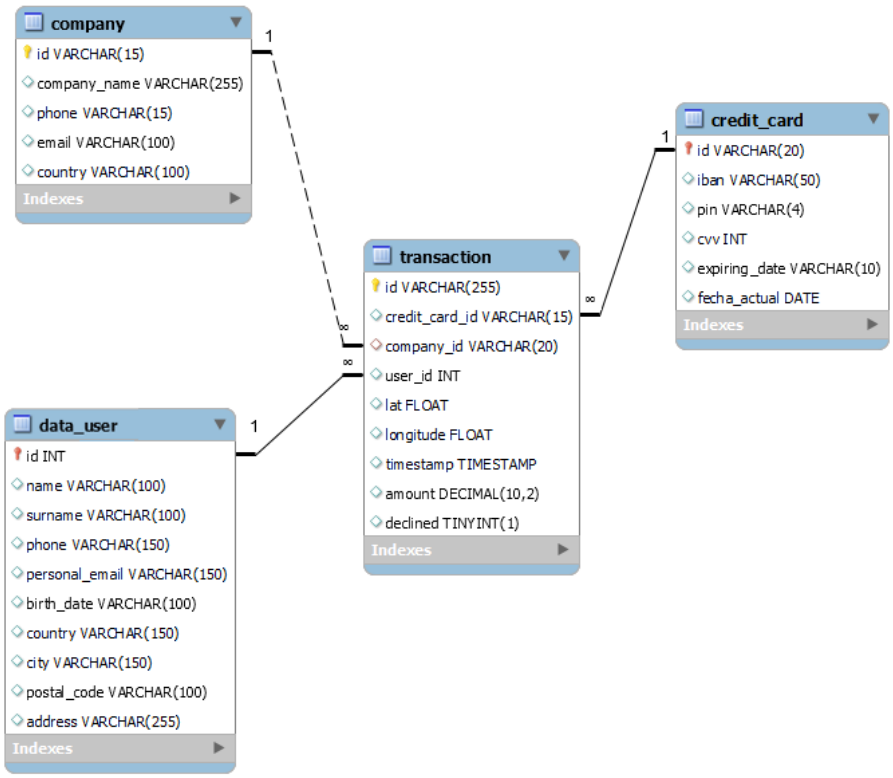
Action Output

	Time	Action	Response	Duration / Fetch Time
1	22:30:20	select * from VistaMarketing where country = 'Germany' LIMIT 0, 1...	8 row(s) returned	0.0048 sec / 0.0000...

Level 3.

Exercise1.

Update DB ´transactions´ to obtain the following diagram:



There are different changes that we should made to update the DB:

1) Modify **existing** tables, including:
   - delete column ´`website`´ from table ´`company`´
   - add column ´`fecha_actual`´ to table ´`credit_card`´
   - change data type of different columns in the table ´`credit_card`´
2) Add new table '`user`'
3) Modify table '`user`':
   - change name to '`data_user`'
   - change column name '`email`' to '`personal_email`'

4) Modify relationship between tables '`data_user`' and '`transaction`' (*since each user can have multiple transactions, and each transaction can only be executed by one user, the foreign key should only exist in the* `transaction` *table (on* `transaction.user_id`*), referencing* `user.id`*).*

- drop foreign key in the table ´data_user´
- add foreign key to the table 'transaction' that links unique users (data_user.id) to the transactions they have made (transaction.user_id) with a non-identifying one-to-many relationship.

Table overview

´user´

There are 275 different users.

```
28
29    -- Table 'user'
30    -- Number of users
31 •  select count(*), count(DISTINCT id)
```

100% 11:32

Result Grid | Filter Rows: Search | Export:

count(*)	count(DISTINCT id)
275	275

There are no missing values in the user data.

```
34    -- Number of non-missing values
35 •  select count(*) as total_rows, count(name), count(surname),
36          count(phone), count(email), count(birth_date), count(country),
37          count(city), count(postal_code), count(address)
38    from user;
```

100% 1:39

Result Grid | Filter Rows: Search | Export:

total_rows	count(name)	count(surname)	count(phone)	count(email)	count(birth_date)	count(country)	count(city)	count(postal_code)	count(address)
275	275	275	275	275	275	275	275	275	275

There is transaction with no info in the 'user' table.

```
40    -- check if there is user info for all transactions
41 •  select *
42    from transaction
43    where user_id not IN (
44    select id from user);
```

100% 22:44

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

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	2024-09-30 22:12:56	111.11	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

To maintain database integrity, the user with ´9999´ was added.

```
46    -- add user '9999' into 'user' table
47 •  INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address)
48    VALUES ("9999", Null, Null, Null, Null, Null, Null, Null, Null, Null);
49 •  select * from user where id='9999';
```

100% 1:48

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

id	name	surname	phone	email	birth_date	country	city	postal_code	address
9999	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Not all users in the table `'user'` made purchases.

```
51    -- check if all users made purchases
52 •  select *
53    from user
54    where id not IN (
55    select user_id from transaction);
```

100% 10:53

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

id	name	surname	phone	email	birth_date	country	city	postal_code	address
1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.edu	Nov 17, 1985	United States	Lowell	73544	348-7818 Sagittis St.
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Des Moines	59464	903 Sit Ave
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Columbus	56518	736-2063 Tellus St.
4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.edu	Feb 18, 1989	United States	Kailua	77417	Ap #545-2244 Erat. Rd.
5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	Sep 26, 1998	United States	Sandy	31564	341-2821 Ultrices Av.
6	Joel	Tyson	(718) 288-8020	gravida.nunc.sed@yahoo.ca	Oct 15, 1989	United States	Nashville	96838	888-2799 Amet Street
7	Rafael	Jimenez	(817) 689-0478	eget@outlook.ca	Dec 4, 1981	United States	Hillsboro	29874	8627 Malesuada Rd.
8	Nissim	Franks	(692) 157-3469	egestas.aliquam.fringilla@google.ca	Aug 1, 1993	United States	Jackson	61750	Ap #251-7144 Integer St.
9	Mannix	Mcclain	(590) 883-2184	aliquam.nisl@outlook.com	Jan 24, 1987	United States	Richmond	35987	647-3080 Lacus. St.
10	Robert	Mccarthy	(324) 746-6771	fermentum@protonmail.com	Apr 30, 1984	United States	Eugene	85526	P.O. Box 773, 3594 Ornare St.
11	Joan	Baird	(981) 429-8106	et@outlook.net	Feb 25, 1990	United States	Lincoln	35211	P.O. Box 687, 8917 Ligula St.
12	Benedict	Wheeler	1-515-824-2855	tincidunt.donec.vitae@hotmail.couk	Aug 6, 1999	United States	Lewiston	92393	748-8694 Porttitor Avenue
13	Allegra	Stanton	1-927-753-6488	proin.eget@protonmail.ca	May 19, 1990	United States	Kearney	14947	4457 Ante. Av.

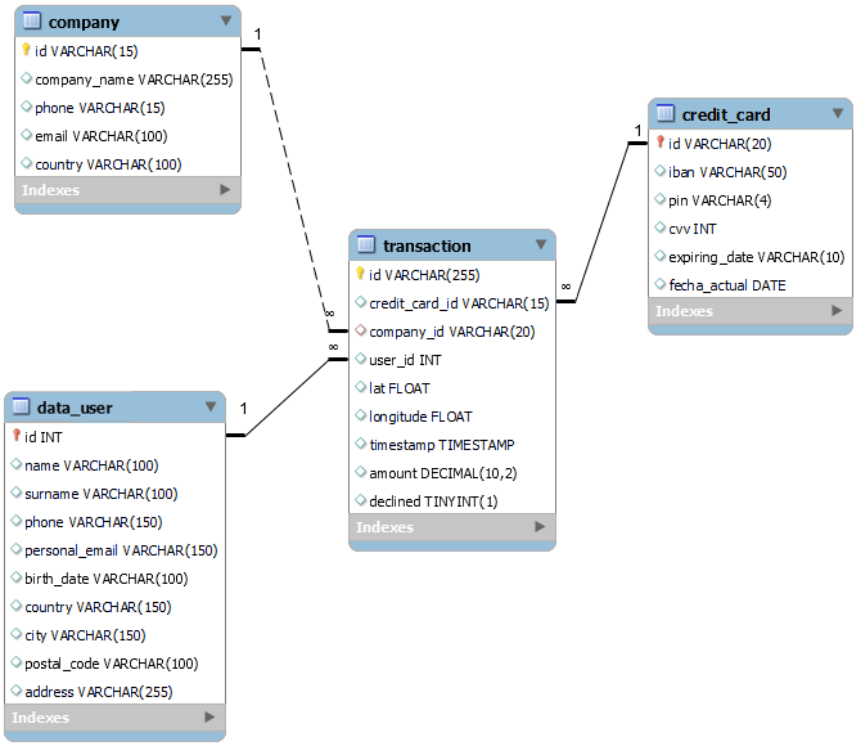
user 15

Action Output

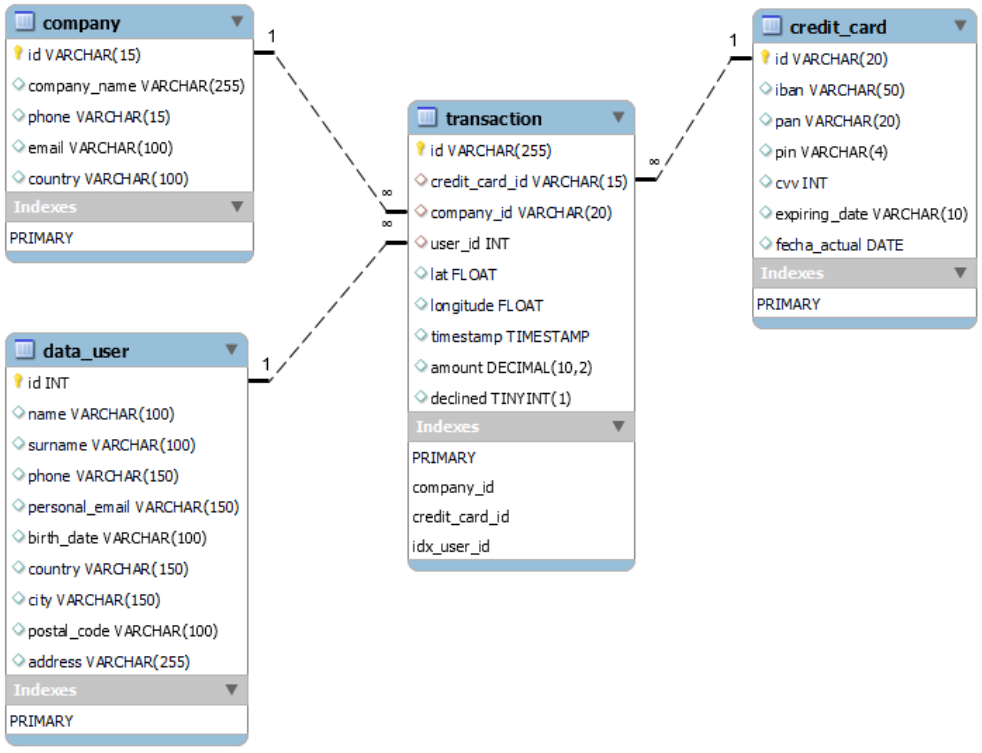
	Time	Action	Response	Duration / Fetch Time
1	08:19:24	select * from user where id not IN (select user_id from transaction...	59 row(s) returned	0.0016 sec / 0.00014...

ER Diagrams

Desired



Achieved



*** There are still some differences between ER diagrams.*

- *In the tables `´data_user´` and `´credit_card´`, the ID columns are both primary keys and foreign keys (red coloured keys). In the current case, this circular relationship seems unnecessary since there is no hierarchy in the data.*
- *Tables `´data_user´` and `´credit_card´` have identifying relationships (solid lines) to table `´transaction´`. But it seems redundant because each transaction is uniquely identified by its ID. In the case of identifying relationships each transaction will be uniquely identified by its id, user_id and credit_card_id. This means that `transaction.user_id` and `transaction.credit_card_id` are parts of the primary key of the `transaction` table.*
- *In the `transaction` table, `user_id` and `credit_card_id` do not have red icons indicating that they are foreign keys, which may be a MySQL Workbench "bug".*

Exercise2.

Create a view "Technical Report" that contains the following information: transaction ID, name of the user, surname of the user, IBAN of the credit card used, name of the company of the transaction carried out.

Present the created view, sorting the data in descending order based on the transaction ID variable.

```
108    -- Exercise 2
109    -- Create a view "Technical Report" that contains the following information:
110    -- transaction ID, name of the user, surname of the user, IBAN of the credit card used,
111    -- name of the company of the transaction carried out.
112    -- Present the created view, sorting the data in descending order based on the transaction ID.
113    create view TechnicalReport as
114    select t.id as transaction_id, u.name as user_name, u.surname as user_surname, cc.iban, c.company_name
115    from transaction as t
116    join data_user as u on t.user_id=u.id
117    join credit_card as cc on t.credit_card_id=cc.id
118    join company as c on t.company_id=c.id;
119
120    select * from TechnicalReport
121    order by transaction_id desc;
```

100% 49:117

Result Grid Filter Rows: Search Export:

transaction_id	user_name	user_surname	iban	company_name
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84484515828105415 26	Neque Tellus Imperdiet Corp.

TechnicalReport 40

Action Output

	Time	Action	Response	Duration / Fetch Time
1	08:42:42	create view TechnicalReport as select t.id as transaction_id, u.name as user_n...	0 row(s) affected	0.0025 sec
2	08:42:42	select * from TechnicalReport order by transaction_id desc LIMIT 0, 1000	587 row(s) returned	0.017 sec / 0.0021 sec