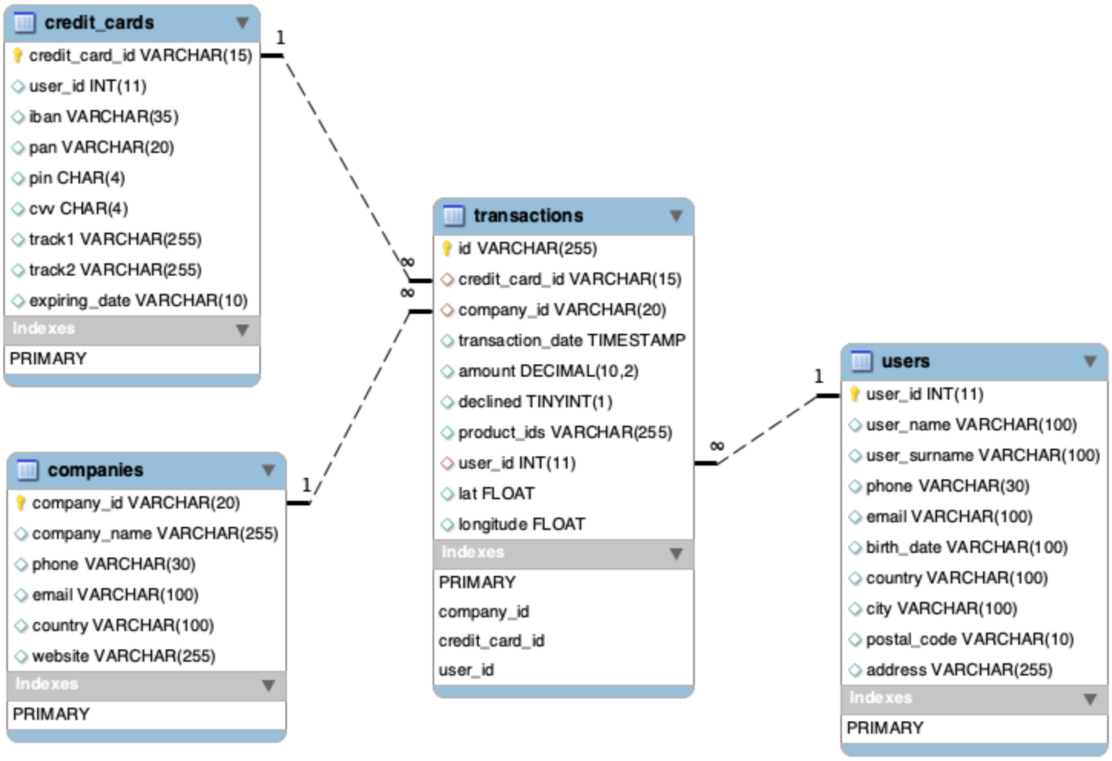
Task 4.01.

Level 1.

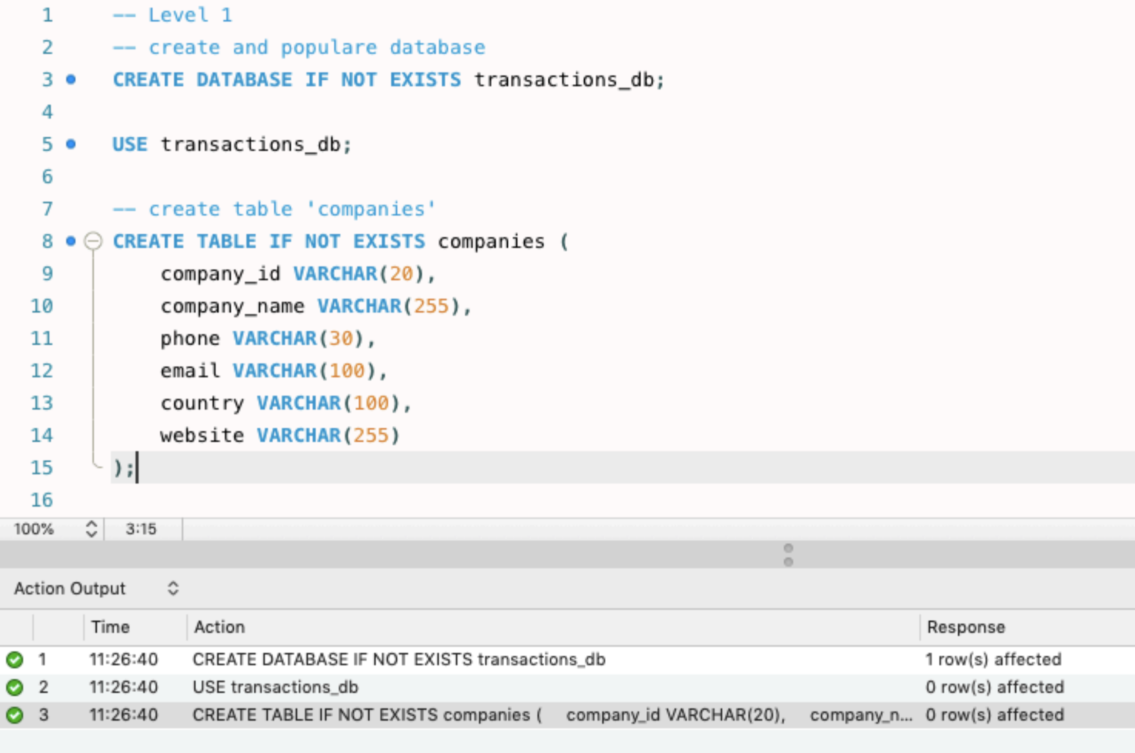
A database `'transactions_db'` containing information from a company selling products online was created and populated with data from the provided `.csv` files.

ER diagram



Process of database creation

First the database `transactions_db` was created, then first table `companies` was added to it.



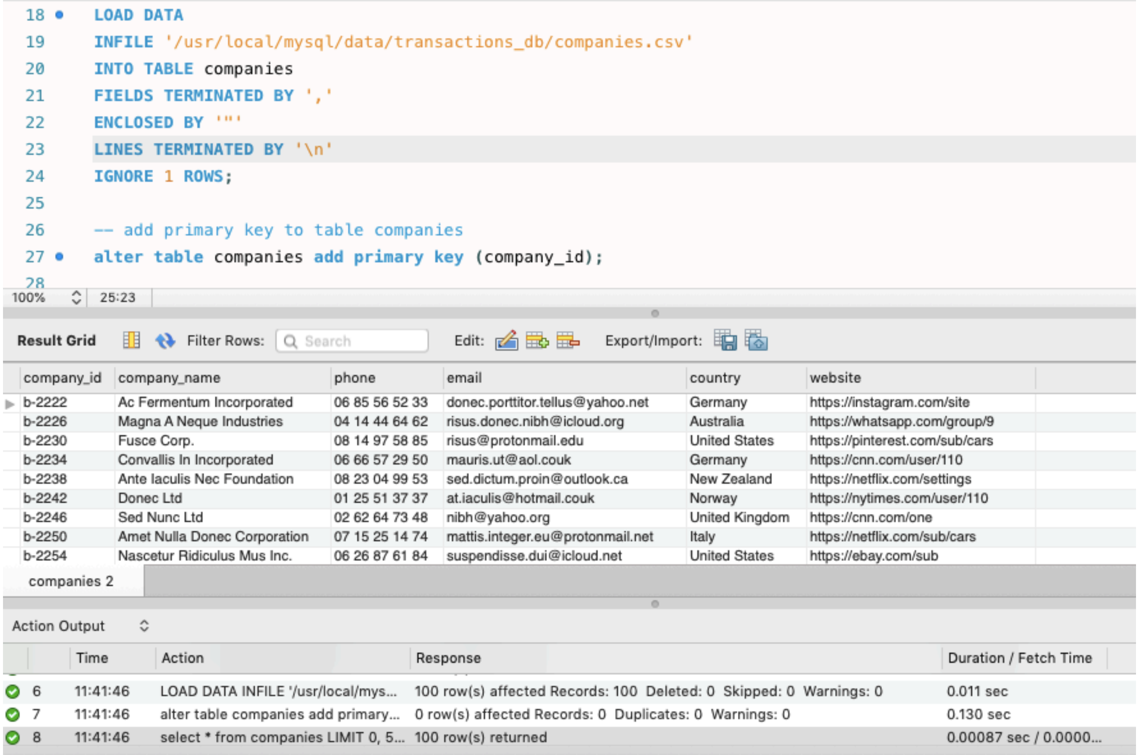
```sql
-- Level 1
-- create and populare database
CREATE DATABASE IF NOT EXISTS transactions_db;

USE transactions_db;

-- create table 'companies'
CREATE TABLE IF NOT EXISTS companies (
    company_id VARCHAR(20),
    company_name VARCHAR(255),
    phone VARCHAR(30),
    email VARCHAR(100),
    country VARCHAR(100),
    website VARCHAR(255)
);
```

		Time	Action	Response
✓	1	11:26:40	CREATE DATABASE IF NOT EXISTS transactions_db	1 row(s) affected
✓	2	11:26:40	USE transactions_db	0 row(s) affected
✓	3	11:26:40	CREATE TABLE IF NOT EXISTS companies (company_id VARCHAR(20), company_n...	0 row(s) affected

The table was populated with data from the `companies.csv` file, placed in the database folder.

Depending on the data, the presence of quotation marks, commas inside fields, types of field separators, and non-printable characters at the end of lines, different values were used in the FIELDS TERMINATED BY and LINES TERMINATED BY clauses. The first line (column names) was always ignored.

After checking that the `company_id` column has non-null unique values, it was assigned as the primary key.

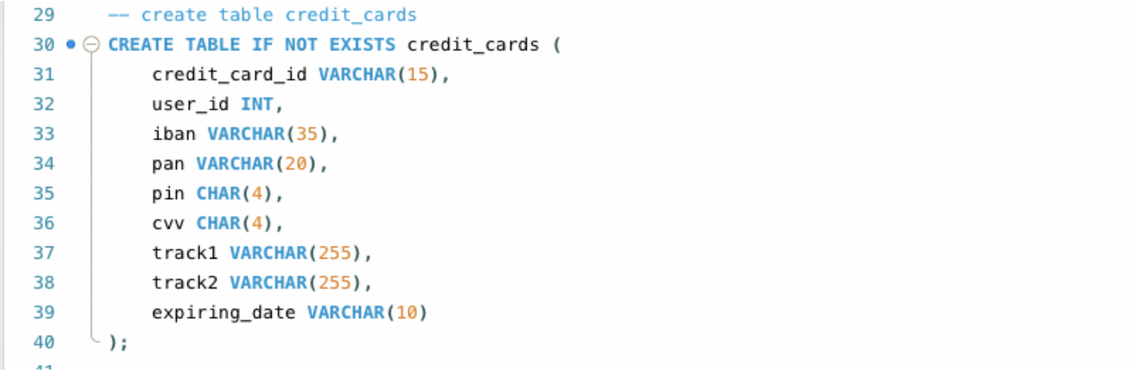


```
LOAD DATA
INFILE '/usr/local/mysql/data/transactions_db/companies.csv'
INTO TABLE companies
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

-- add primary key to table companies
alter table companies add primary key (company_id);
```

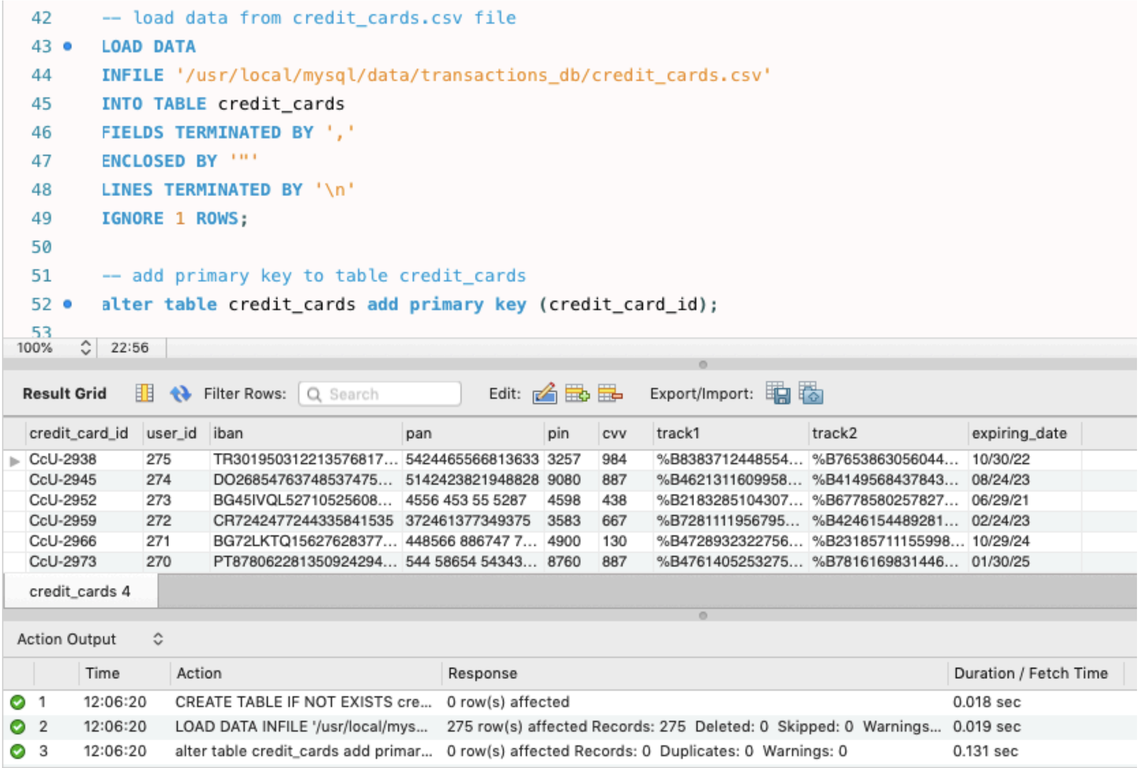
company_id	company_name	phone	email	country	website
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net	Germany	https://instagram.com/site
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia	https://whatsapp.com/group/9
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/sub/cars
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany	https://cnn.com/user/110
b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings
b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.couk	Norway	https://nytimes.com/user/110
b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
b-2250	Amet Nulla Donec Corporation	07 15 25 14 74	mattis.integer.eu@protonmail.net	Italy	https://netflix.com/sub/cars
b-2254	Nascetur Ridiculus Mus Inc.	06 26 87 61 84	suspendisse.dui@icloud.net	United States	https://ebay.com/sub

		Time	Action	Response	Duration / Fetch Time
✓	6	11:41:46	LOAD DATA INFILE '/usr/local/mys...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.011 sec
✓	7	11:41:46	alter table companies add primary...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.130 sec
✓	8	11:41:46	select * from companies LIMIT 0, 5...	100 row(s) returned	0.00087 sec / 0.0000...

Then the other tables (`credit_cards.csv`, `users.csv`, `transactions.csv`) were added in the same way.



```sql
-- create table credit_cards
CREATE TABLE IF NOT EXISTS credit_cards (
    credit_card_id VARCHAR(15),
    user_id INT,
    iban VARCHAR(35),
    pan VARCHAR(20),
    pin CHAR(4),
    cvv CHAR(4),
    track1 VARCHAR(255),
    track2 VARCHAR(255),
    expiring_date VARCHAR(10)
);
```

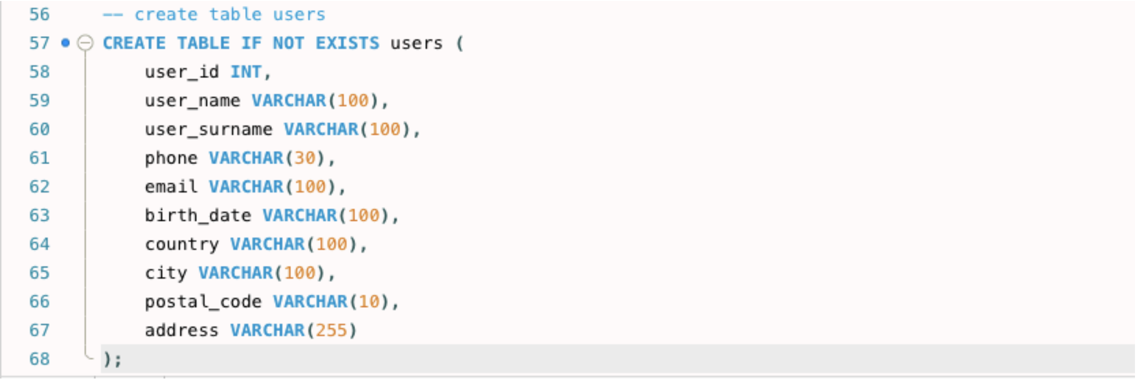


```sql
-- load data from credit_cards.csv file
LOAD DATA
INFILE '/usr/local/mysql/data/transactions_db/credit_cards.csv'
INTO TABLE credit_cards
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

-- add primary key to table credit_cards
alter table credit_cards add primary key (credit_card_id);
```

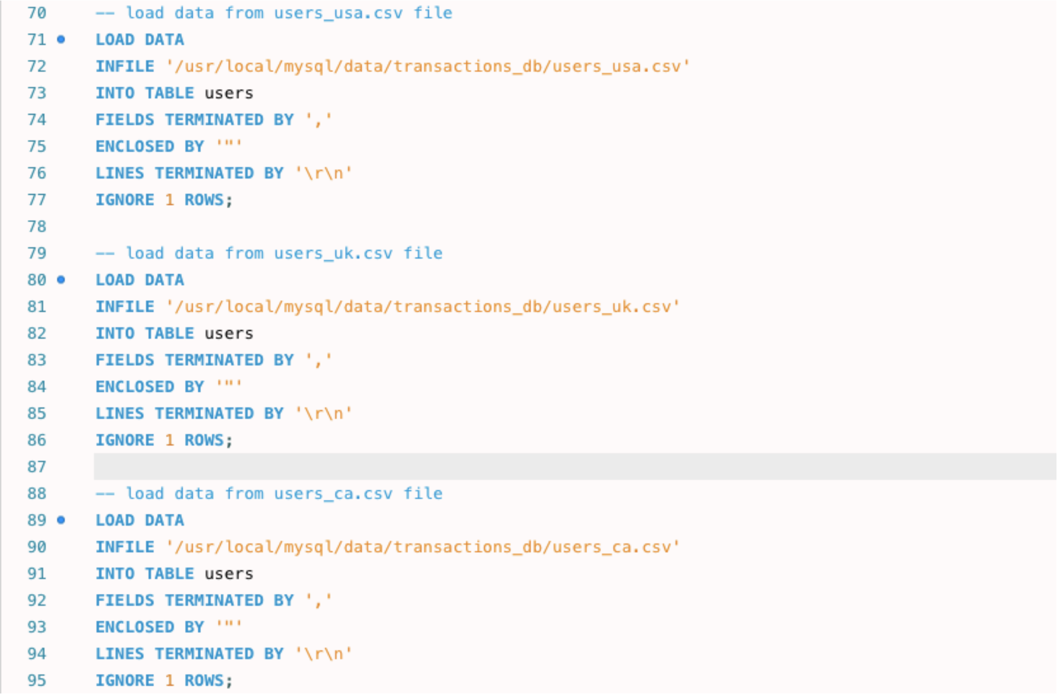
credit_card_id	user_id	iban	pan	pin	cvv	track1	track2	expiring_date
CcU-2938	275	TR301950312213576817...	5424465566813633	3257	984	%B8383712448554...	%B7653863056044...	10/30/22
CcU-2945	274	DO26854763748537475...	5142423821948828	9080	887	%B4621311609958...	%B4149568437843...	08/24/23
CcU-2952	273	BG45IVQL52710525608...	4556 453 55 5287	4598	438	%B2183285104307...	%B6778580257827...	06/29/21
CcU-2959	272	CR7242477244335841535	372461377349375	3583	667	%B7281111956795...	%B4246154489281...	02/24/23
CcU-2966	271	BG72LKTQ15627628377...	448566 886747 7...	4900	130	%B4728932322756...	%B23185711155998...	10/29/24
CcU-2973	270	PT878062281350924294...	544 58654 54343...	8760	887	%B4761405253275...	%B7816169831446...	01/30/25

		Time	Action	Response	Duration / Fetch Time
✓	1	12:06:20	CREATE TABLE IF NOT EXISTS cre...	0 row(s) affected	0.018 sec
✓	2	12:06:20	LOAD DATA INFILE '/usr/local/mys...	275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings...	0.019 sec
✓	3	12:06:20	alter table credit_cards add primar...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.131 sec

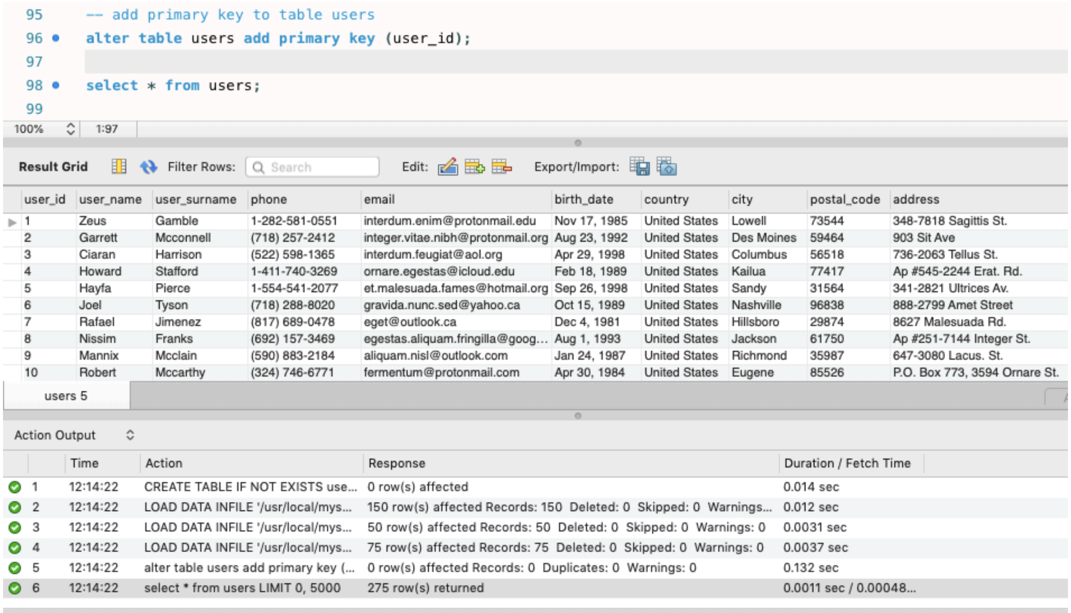
For the table `users`, data was loaded from three `.csv` files.



```sql
-- create table users
CREATE TABLE IF NOT EXISTS users (
    user_id INT,
    user_name VARCHAR(100),
    user_surname VARCHAR(100),
    phone VARCHAR(30),
    email VARCHAR(100),
    birth_date VARCHAR(100),
    country VARCHAR(100),
    city VARCHAR(100),
    postal_code VARCHAR(10),
    address VARCHAR(255)
);
```



```sql
-- load data from users_usa.csv file
LOAD DATA
INFILE '/usr/local/mysql/data/transactions_db/users_usa.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;

-- load data from users_uk.csv file
LOAD DATA
INFILE '/usr/local/mysql/data/transactions_db/users_uk.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;

-- load data from users_ca.csv file
LOAD DATA
INFILE '/usr/local/mysql/data/transactions_db/users_ca.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;
```

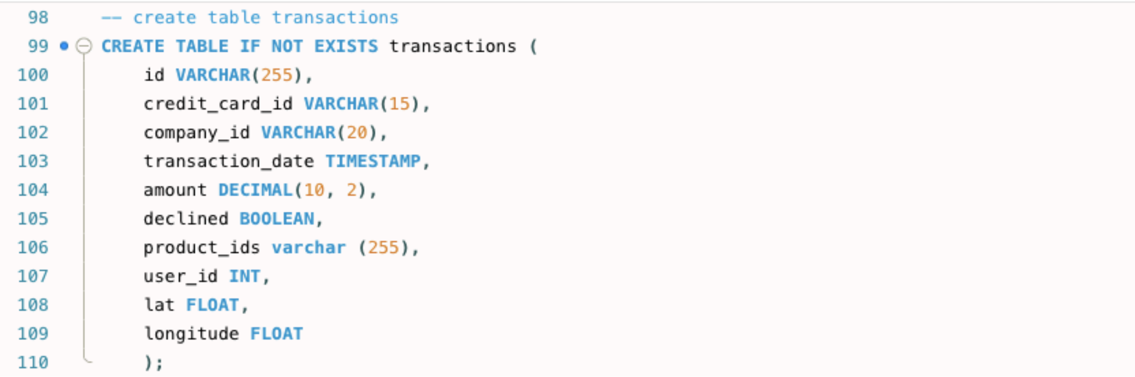


```sql
-- add primary key to table users
alter table users add primary key (user_id);

select * from users;
```

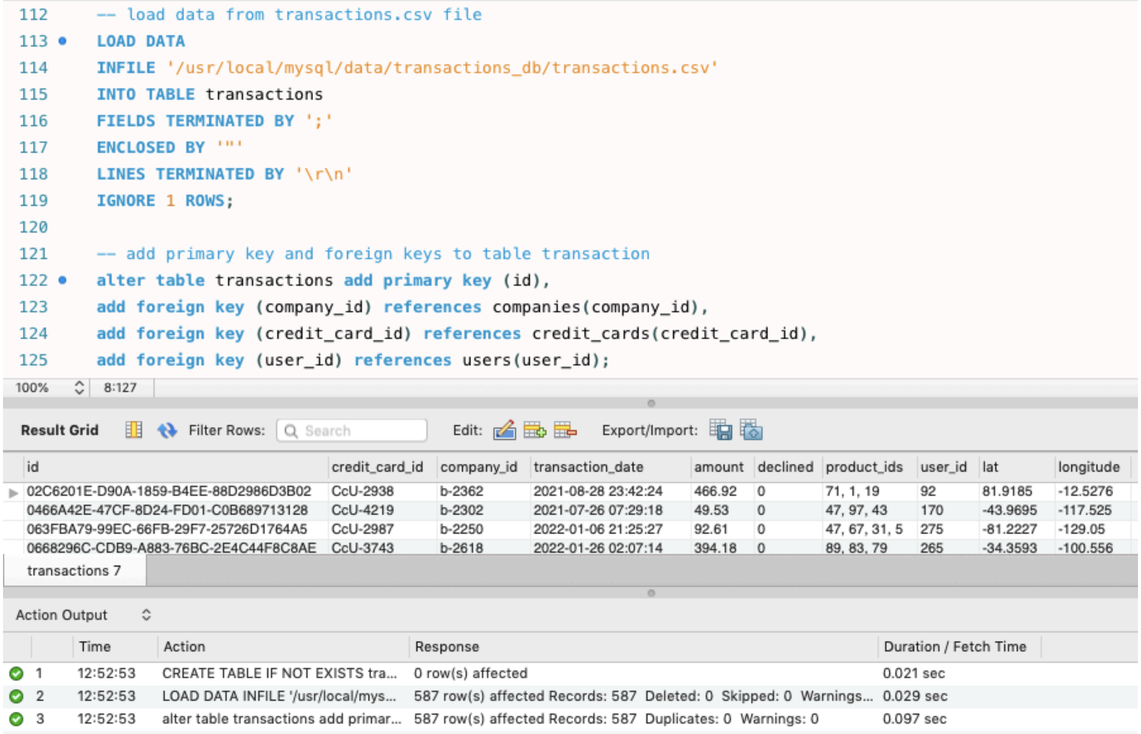
user_id	user_name	user_surname	phone	email	birth_date	country	city	postal_code	address
1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.edu	Nov 17, 1985	United States	Lowell	73544	348-7818 Sagittis St.
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	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@goog…	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.

	Time	Action	Response	Duration / Fetch Time
1	12:14:22	CREATE TABLE IF NOT EXISTS use…	0 row(s) affected	0.014 sec
2	12:14:22	LOAD DATA INFILE '/usr/local/mys…	150 row(s) affected Records: 150 Deleted: 0 Skipped: 0 Warnings…	0.012 sec
3	12:14:22	LOAD DATA INFILE '/usr/local/mys…	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0	0.0031 sec
4	12:14:22	LOAD DATA INFILE '/usr/local/mys…	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0	0.0037 sec
5	12:14:22	alter table users add primary key (…	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.132 sec
6	12:14:22	select * from users LIMIT 0, 5000	275 row(s) returned	0.0011 sec / 0.00048…

For the table `transactions` foreign keys were also added.



```sql
 -- create table transactions
CREATE TABLE IF NOT EXISTS transactions (
     id VARCHAR(255),
     credit_card_id VARCHAR(15),
     company_id VARCHAR(20),
     transaction_date TIMESTAMP,
     amount DECIMAL(10, 2),
     declined BOOLEAN,
     product_ids varchar (255),
     user_id INT,
     lat FLOAT,
     longitude FLOAT
     );
```



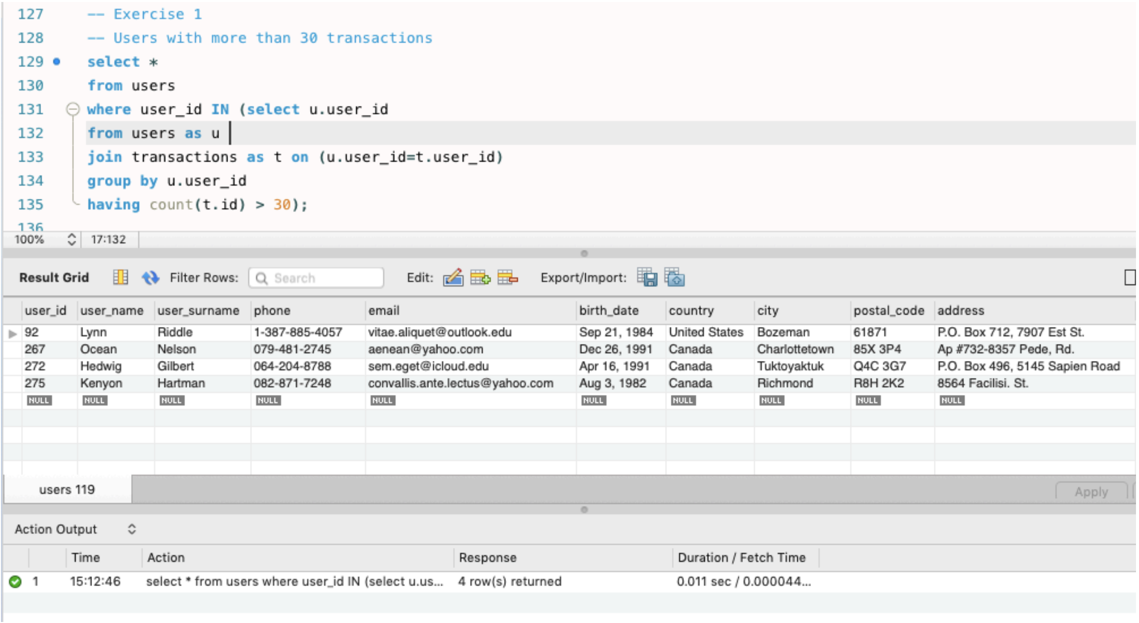
```sql
-- load data from transactions.csv file
LOAD DATA
INFILE '/usr/local/mysql/data/transactions_db/transactions.csv'
INTO TABLE transactions
FIELDS TERMINATED BY ';'
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;

-- add primary key and foreign keys to table transaction
alter table transactions add primary key (id),
add foreign key (company_id) references companies(company_id),
add foreign key (credit_card_id) references credit_cards(credit_card_id),
add foreign key (user_id) references users(user_id);
```

id	credit_card_id	company_id	transaction_date	amount	declined	product_ids	user_id	lat	longitude
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	2021-08-28 23:42:24	466.92	0	71, 1, 19	92	81.9185	-12.5276
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	2021-07-26 07:29:18	49.53	0	47, 97, 43	170	-43.9695	-117.525
063FBA79-99EC-66FB-29F7-25726D1764A5	CcU-2987	b-2250	2022-01-06 21:25:27	92.61	0	47, 67, 31, 5	275	-81.2227	-129.05
0668296C-CDB9-A883-76BC-2E4C44F8C8AE	CcU-3743	b-2618	2022-01-26 02:07:14	394.18	0	89, 83, 79	265	-34.3593	-100.556

		Time	Action	Response	Duration / Fetch Time
✓	1	12:52:53	CREATE TABLE IF NOT EXISTS tra...	0 row(s) affected	0.021 sec
✓	2	12:52:53	LOAD DATA INFILE '/usr/local/mys...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings...	0.029 sec
✓	3	12:52:53	alter table transactions add primar...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0	0.097 sec

Exercise 1.

Users with more than 30 transactions.



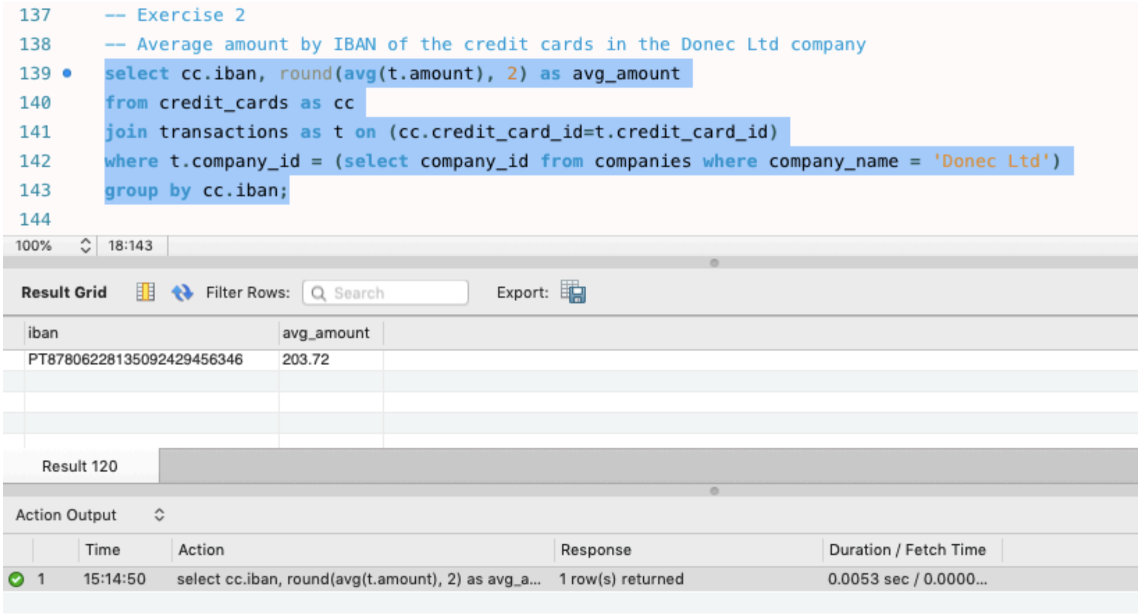
```
-- Exercise 1
-- Users with more than 30 transactions
select *
from users
where user_id IN (select u.user_id
from users as u
join transactions as t on (u.user_id=t.user_id)
group by u.user_id
having count(t.id) > 30);
```

user_id	user_name	user_surname	phone	email	birth_date	country	city	postal_code	address
92	Lynn	Riddle	1-387-885-4057	vitae.aliquet@outlook.edu	Sep 21, 1984	United States	Bozeman	61871	P.O. Box 712, 7907 Est St.
267	Ocean	Nelson	079-481-2745	aenean@yahoo.com	Dec 26, 1991	Canada	Charlottetown	85X 3P4	Ap #732-8357 Pede, Rd.
272	Hedwig	Gilbert	064-204-8788	sem.eget@icloud.edu	Apr 16, 1991	Canada	Tuktoyaktuk	Q4C 3G7	P.O. Box 496, 5145 Sapien Road
275	Kenyon	Hartman	082-871-7248	convallis.ante.lectus@yahoo.com	Aug 3, 1982	Canada	Richmond	R8H 2K2	8564 Facilisi. St.
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	Time	Action	Response	Duration / Fetch Time
✓ 1	15:12:46	select * from users where user_id IN (select u.us...	4 row(s) returned	0.011 sec / 0.000044...

Exercise 2.

Average amount by IBAN of the credit cards in the Donec Ltd company.



```sql
-- Exercise 2
-- Average amount by IBAN of the credit cards in the Donec Ltd company
select cc.iban, round(avg(t.amount), 2) as avg_amount
from credit_cards as cc
join transactions as t on (cc.credit_card_id=t.credit_card_id)
where t.company_id = (select company_id from companies where company_name = 'Donec Ltd')
group by cc.iban;
```

iban	avg_amount
PT87806228135092429456346	203.72

	Time	Action	Response	Duration / Fetch Time
✓ 1	15:14:50	select cc.iban, round(avg(t.amount), 2) as avg_a...	1 row(s) returned	0.0053 sec / 0.0000...

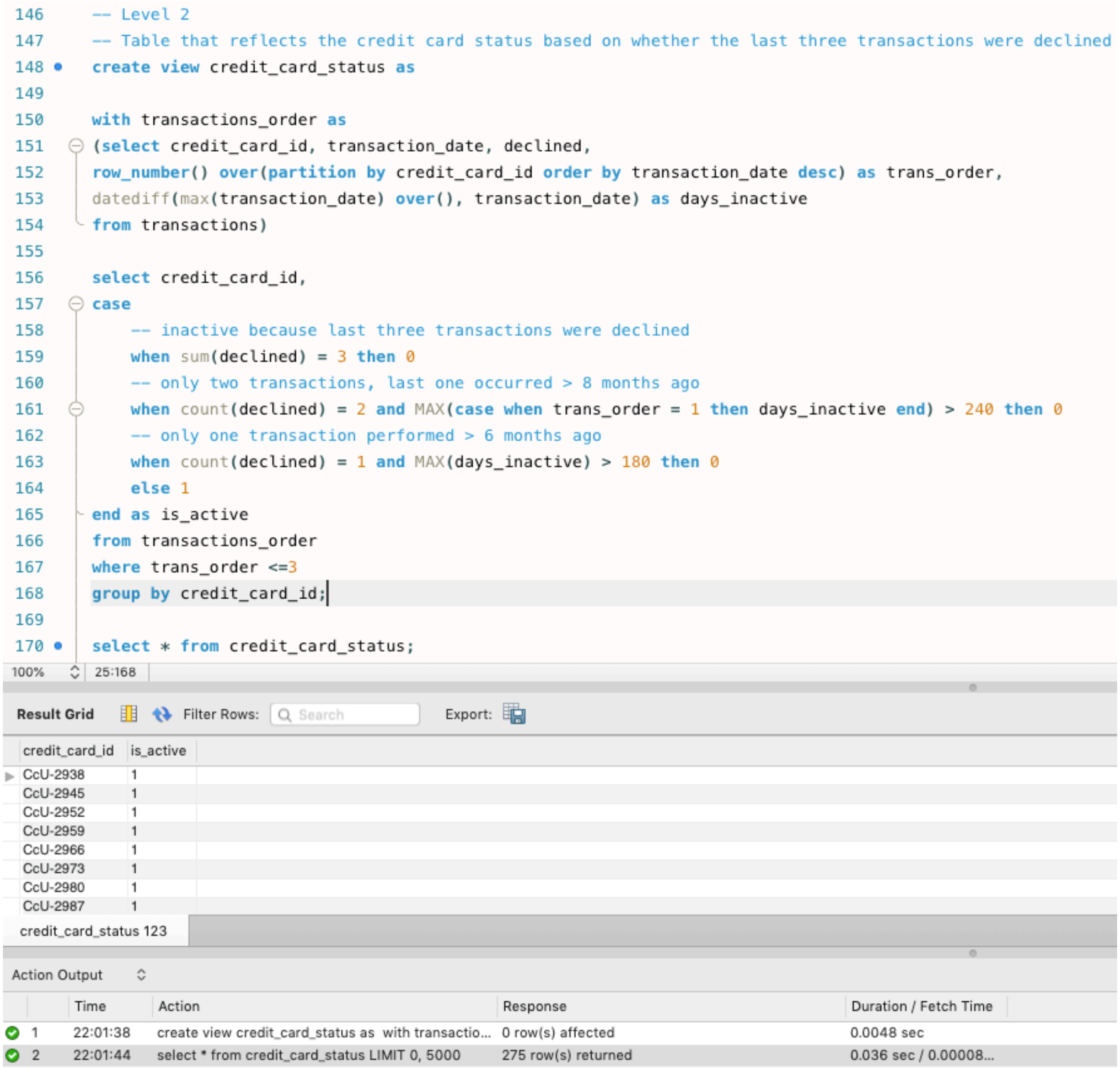
Level 2.

To maintain “updatable” table, a view ´credit_card_status´ was created. This view ensures that the credit card status is always current, reflecting any changes in the transactions table (since the credit card status is calculated based on transaction data). The view determines the status of a credit card using several criteria:

- according to the task requirements, credit card is considered "inactive" if the last three transactions were declined.

However, many credit cards may have only two or even one transaction. To account for these cases, two additional conditions were implemented to mark a card as inactive:

- if the credit card has only two transactions, and the last one occurred more than 8 months ago;
- If the credit card has only one transaction, and it occurred more than 6 months ago.



```sql
-- Level 2
-- Table that reflects the credit card status based on whether the last three transactions were declined
create view credit_card_status as

with transactions_order as
(select credit_card_id, transaction_date, declined,
row_number() over(partition by credit_card_id order by transaction_date desc) as trans_order,
datediff(max(transaction_date) over(), transaction_date) as days_inactive
from transactions)

select credit_card_id,
case
    -- inactive because last three transactions were declined
    when sum(declined) = 3 then 0
    -- only two transactions, last one occurred > 8 months ago
    when count(declined) = 2 and MAX(case when trans_order = 1 then days_inactive end) > 240 then 0
    -- only one transaction performed > 6 months ago
    when count(declined) = 1 and MAX(days_inactive) > 180 then 0
    else 1
end as is_active
from transactions_order
where trans_order <=3
group by credit_card_id;

select * from credit_card_status;
```

credit_card_id	is_active
CcU-2938	1
CcU-2945	1
CcU-2952	1
CcU-2959	1
CcU-2966	1
CcU-2973	1
CcU-2980	1
CcU-2987	1

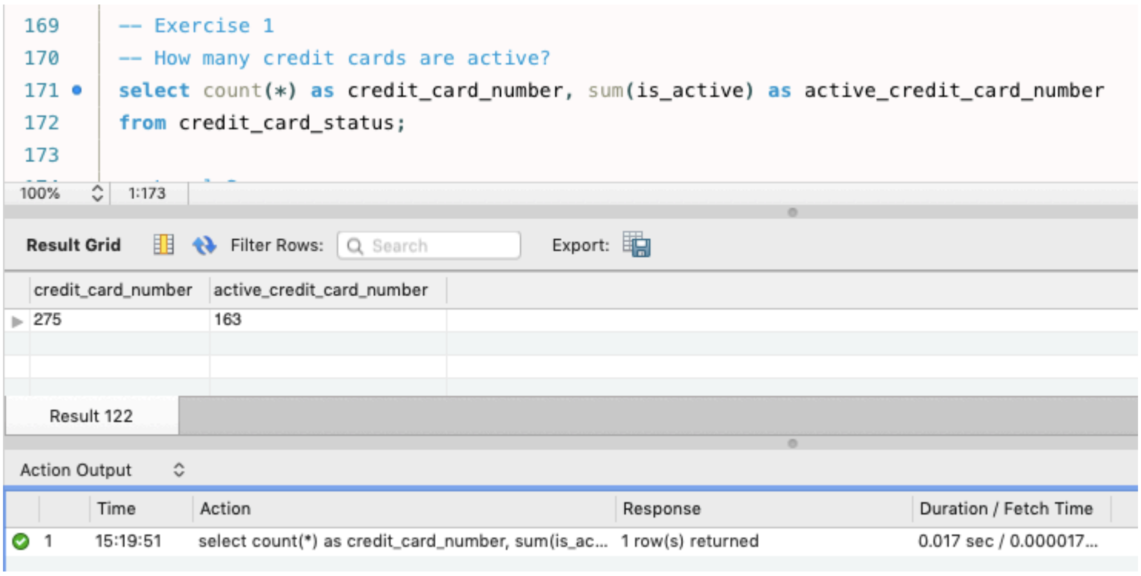
	Time	Action	Response	Duration / Fetch Time
1	22:01:38	create view credit_card_status as with transactio...	0 row(s) affected	0.0048 sec
2	22:01:44	select * from credit_card_status LIMIT 0, 5000	275 row(s) returned	0.036 sec / 0.00008...

*** Additional conditions are subjective and should be adjusted based on the specific business model. Depending on business needs, another condition for credit card “inactivity” can also be added: if the credit card has only one or two transactions and all of them were declined.*

The type of goods being sold is another important factor to consider. For instance, if the goods have a longer sales cycle, the period of "inactivity" may extend beyond 6 months. However, based on the product prices and names, it can be assumed that these are not expensive, infrequently purchased items. Therefore, the current inactivity periods seem reasonable for this context.

Exercise1.

How many credit cards are active?



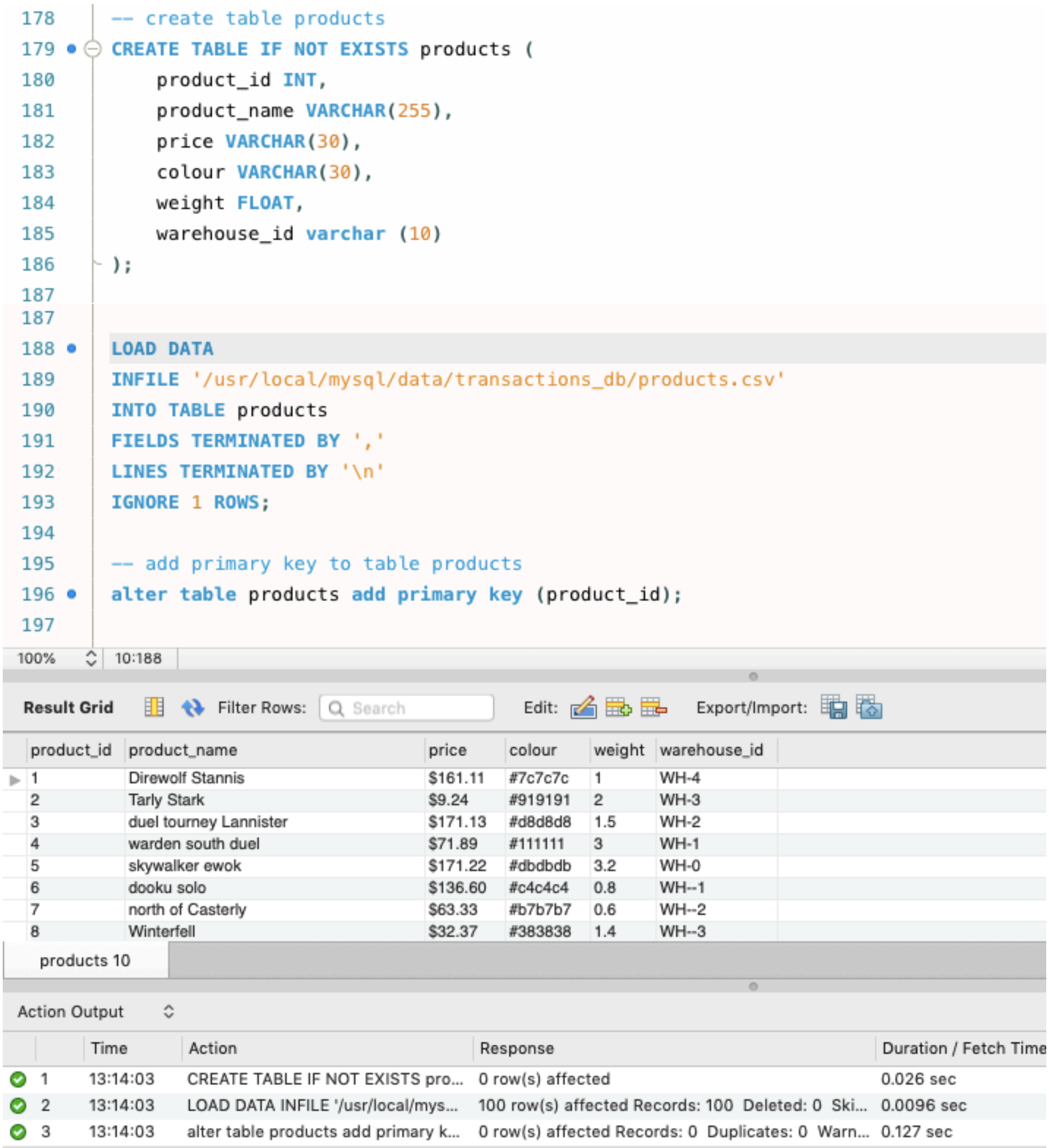
```
-- Exercise 1
-- How many credit cards are active?
select count(*) as credit_card_number, sum(is_active) as active_credit_card_number
from credit_card_status;
```

credit_card_number	active_credit_card_number
275	163

	Time	Action	Response	Duration / Fetch Time
1	15:19:51	select count(*) as credit_card_number, sum(is_ac...	1 row(s) returned	0.017 sec / 0.000017...

Level 3.

To link product data to the corresponding transaction, a `products` table was first created and populated with data from the `products.csv` file.



```sql
-- create table products
CREATE TABLE IF NOT EXISTS products (
    product_id INT,
    product_name VARCHAR(255),
    price VARCHAR(30),
    colour VARCHAR(30),
    weight FLOAT,
    warehouse_id varchar (10)
);


LOAD DATA
INFILE '/usr/local/mysql/data/transactions_db/products.csv'
INTO TABLE products
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

-- add primary key to table products
alter table products add primary key (product_id);
```

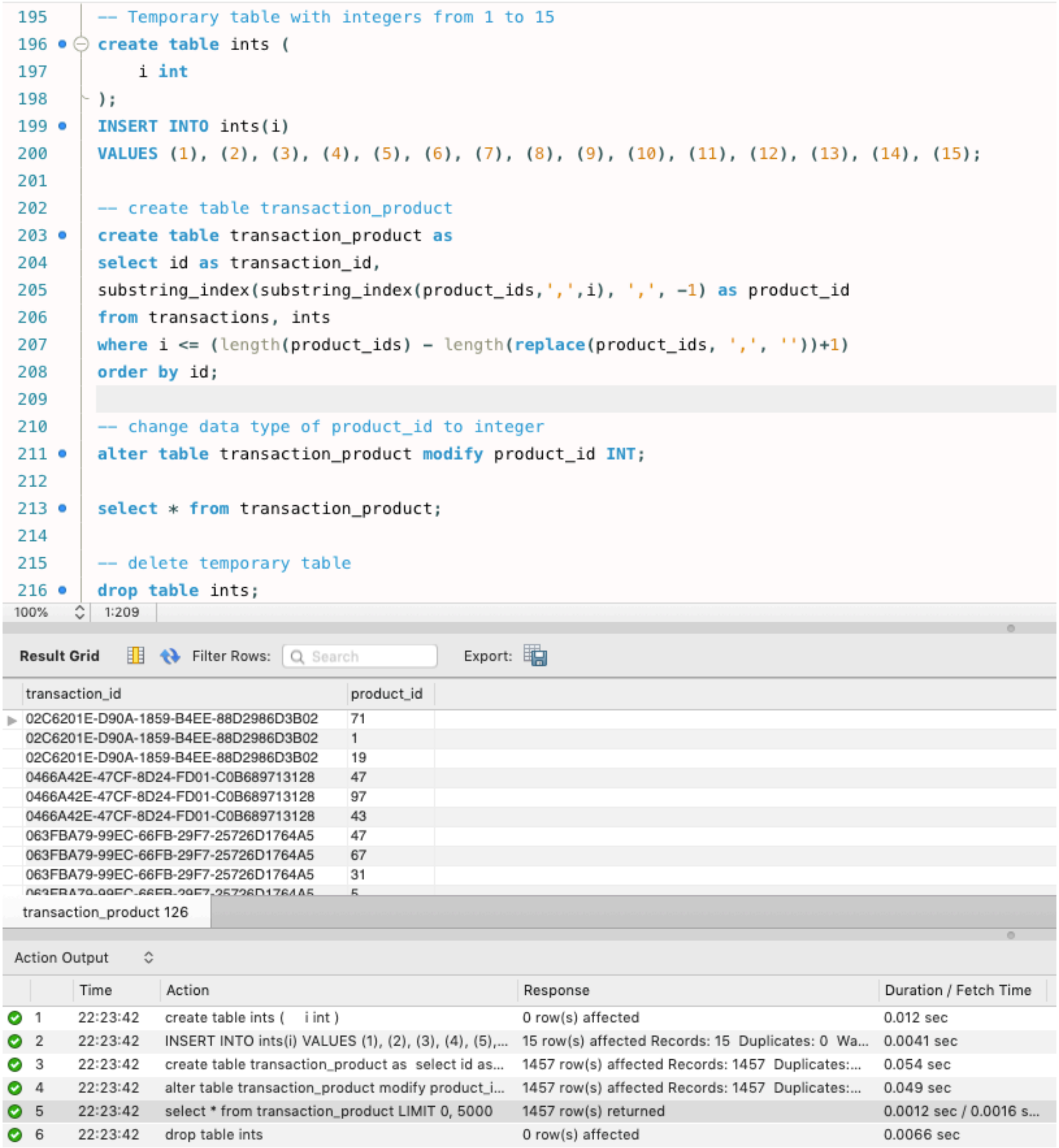
product_id	product_name	price	colour	weight	warehouse_id
1	Direwolf Stannis	$161.11	#7c7c7c	1	WH-4
2	Tarly Stark	$9.24	#919191	2	WH-3
3	duel tourney Lannister	$171.13	#d8d8d8	1.5	WH-2
4	warden south duel	$71.89	#111111	3	WH-1
5	skywalker ewok	$171.22	#dbdbdb	3.2	WH-0
6	dooku solo	$136.60	#c4c4c4	0.8	WH--1
7	north of Casterly	$63.33	#b7b7b7	0.6	WH--2
8	Winterfell	$32.37	#383838	1.4	WH--3

	Time	Action	Response	Duration / Fetch Time
✓ 1	13:14:03	CREATE TABLE IF NOT EXISTS pro...	0 row(s) affected	0.026 sec
✓ 2	13:14:03	LOAD DATA INFILE '/usr/local/mys...	100 row(s) affected Records: 100 Deleted: 0 Ski...	0.0096 sec
✓ 3	13:14:03	alter table products add primary k...	0 row(s) affected Records: 0 Duplicates: 0 Warn...	0.127 sec

Then the `transaction_product` table was created to associate product data with the `transactions_db` database.

To link each product to the corresponding transaction in which it was purchased, the `transactions.product_ids` strings, containing comma-separated values, needed to be "split". Since MySQL does not support data types like arrays and lacks built-in functions for array manipulation, an alternative approach was used. A temporary table containing integers from 1 to 15 was created and joined with the `transactions` table. This allowed each transaction to be repeated as many times as the number of products it contains. A new column was then generated with extracted individual product IDs from the `product_ids` string.

*** This approach has several limitations. The primary limitation is the fixed maximum number of products allowed per transaction, as the `product_ids` string cannot contain more than 15 products.*
Moreover, joining the `transactions` table with the temporary table results in repetitive rows for each transaction, increasing the computational cost. For large datasets, this approach can lead to low performance.



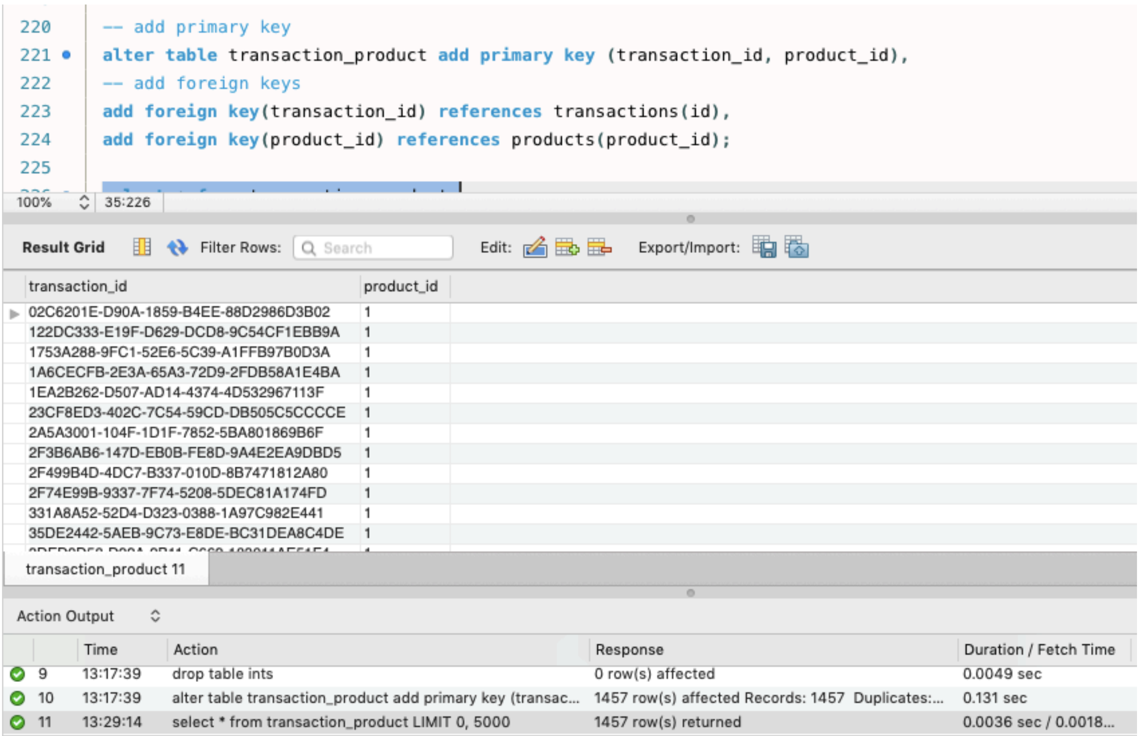
```sql
-- Temporary table with integers from 1 to 15
create table ints (
    i int
);
INSERT INTO ints(i)
VALUES (1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14), (15);

-- create table transaction_product
create table transaction_product as
select id as transaction_id,
substring_index(substring_index(product_ids,',',i), ',', -1) as product_id
from transactions, ints
where i <= (length(product_ids) - length(replace(product_ids, ',', '')))+1)
order by id;

-- change data type of product_id to integer
alter table transaction_product modify product_id INT;

select * from transaction_product;

-- delete temporary table
drop table ints;
```

transaction_id	product_id
02C6201E-D90A-1859-B4EE-88D2986D3B02	71
02C6201E-D90A-1859-B4EE-88D2986D3B02	1
02C6201E-D90A-1859-B4EE-88D2986D3B02	19
0466A42E-47CF-8D24-FD01-C0B689713128	47
0466A42E-47CF-8D24-FD01-C0B689713128	97
0466A42E-47CF-8D24-FD01-C0B689713128	43
063FBA79-99EC-66FB-29F7-25726D1764A5	47
063FBA79-99EC-66FB-29F7-25726D1764A5	67
063FBA79-99EC-66FB-29F7-25726D1764A5	31
063FBA79-99EC-66FB-29F7-25726D1764A5	5

		Time	Action	Response	Duration / Fetch Time
✓	1	22:23:42	create table ints (i int)	0 row(s) affected	0.012 sec
✓	2	22:23:42	INSERT INTO ints(i) VALUES (1), (2), (3), (4), (5),...	15 row(s) affected Records: 15 Duplicates: 0 Wa...	0.0041 sec
✓	3	22:23:42	create table transaction_product as select id as...	1457 row(s) affected Records: 1457 Duplicates:...	0.054 sec
✓	4	22:23:42	alter table transaction_product modify product_i...	1457 row(s) affected Records: 1457 Duplicates:...	0.049 sec
✓	5	22:23:42	select * from transaction_product LIMIT 0, 5000	1457 row(s) returned	0.0012 sec / 0.0016 s...
✓	6	22:23:42	drop table ints	0 row(s) affected	0.0066 sec

To establish connection between tables primary and foreign keys were added to the table `transaction_products`.

The `transaction_product` table resolves the many-to-many relationship between `transactions` and `products`. It ensures that each transaction and product pair is recorded uniquely (combination of `transaction_id` and `product_id` forms a composite primary key), and the foreign key constraints enforce referential integrity between the tables.

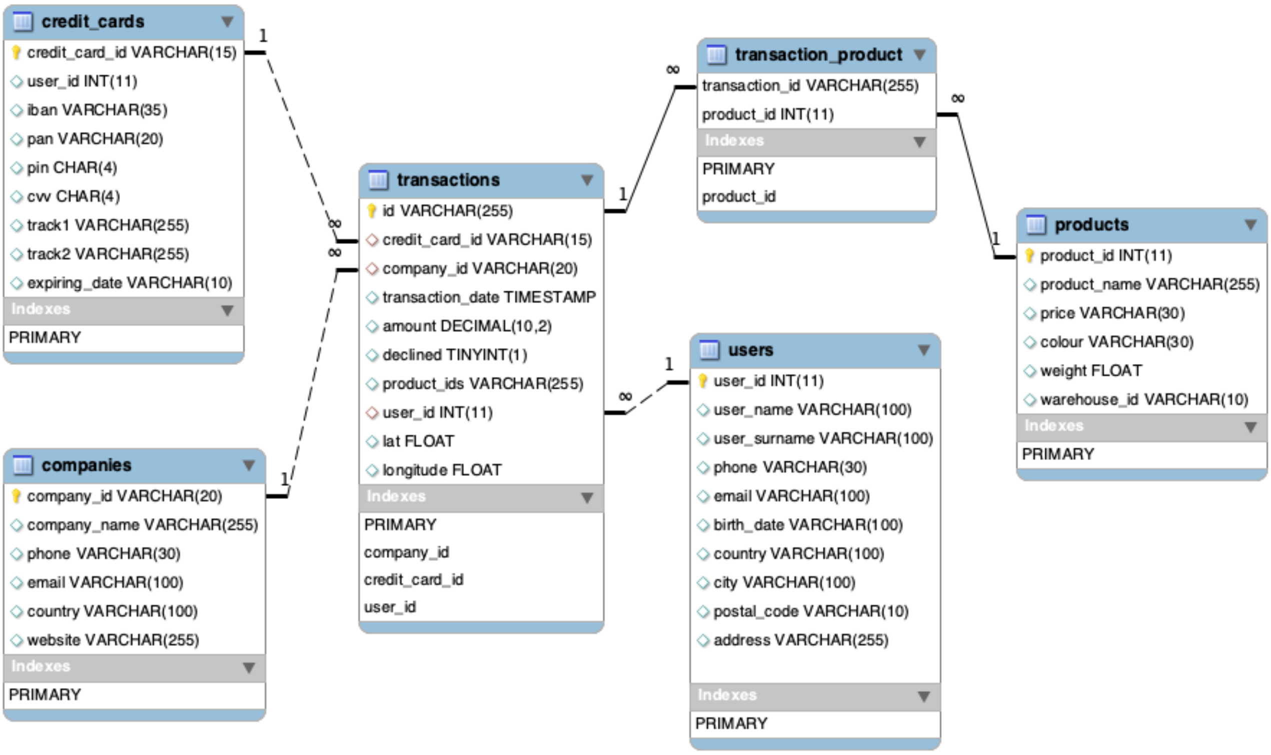


```
-- add primary key
alter table transaction_product add primary key (transaction_id, product_id),
-- add foreign keys
add foreign key(transaction_id) references transactions(id),
add foreign key(product_id) references products(product_id);
```

transaction_id	product_id
02C6201E-D90A-1859-B4EE-88D2986D3B02	1
122DC333-E19F-D629-DCD8-9C54CF1EBB9A	1
1753A288-9FC1-52E6-5C39-A1FFB97B0D3A	1
1A6CECFB-2E3A-65A3-72D9-2FDB58A1E4BA	1
1EA2B262-D507-AD14-4374-4D532967113F	1
23CF8ED3-402C-7C54-59CD-DB505C5CCCCE	1
2A5A3001-104F-1D1F-7852-5BA801869B6F	1
2F3B6AB6-147D-EB0B-FE8D-9A4E2EA9DBD5	1
2F499B4D-4DC7-B337-010D-8B7471812A80	1
2F74E99B-9337-7F74-5208-5DEC81A174FD	1
331A8A52-52D4-D323-0388-1A97C982E441	1
35DE2442-5AEB-9C73-E8DE-BC31DEA8C4DE	1

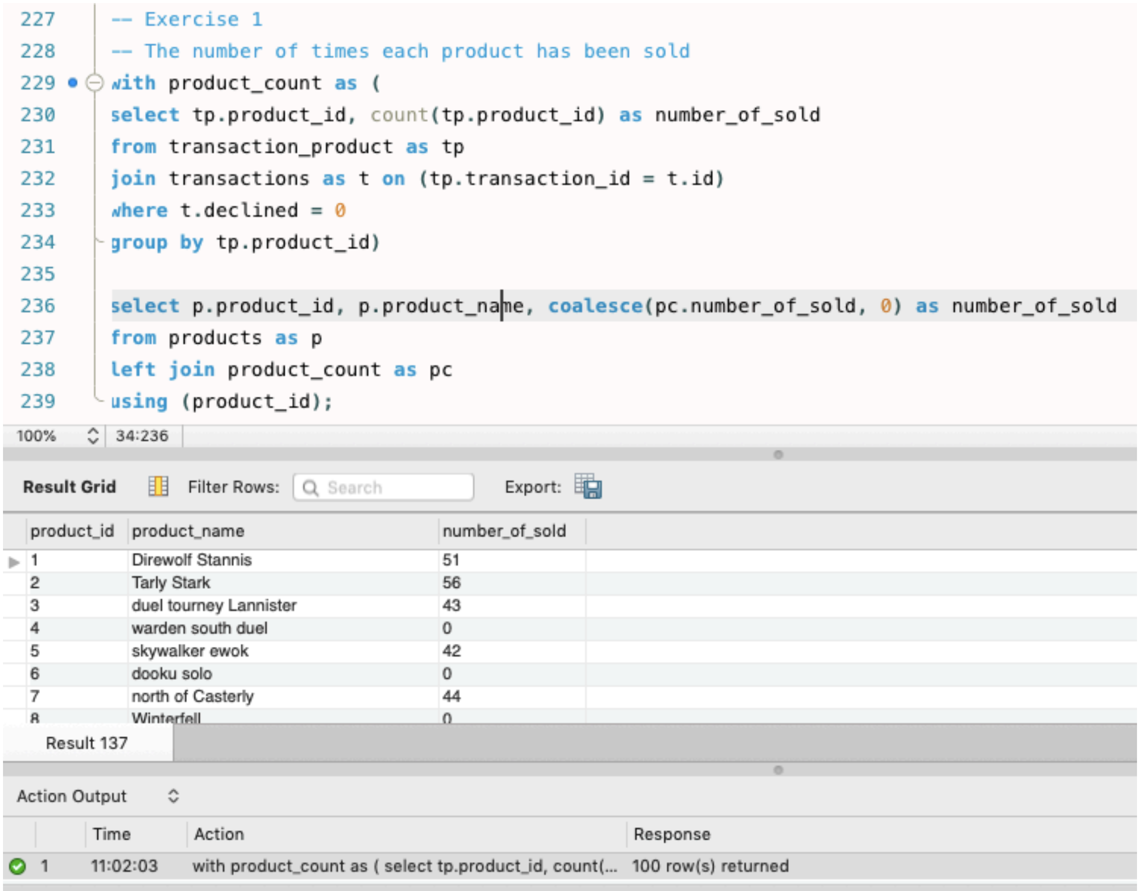
		Time	Action	Response	Duration / Fetch Time
✓	9	13:17:39	drop table ints	0 row(s) affected	0.0049 sec
✓	10	13:17:39	alter table transaction_product add primary key (transac...	1457 row(s) affected Records: 1457 Duplicates:...	0.131 sec
✓	11	13:29:14	select * from transaction_product LIMIT 0, 5000	1457 row(s) returned	0.0036 sec / 0.0018...

ER diagram


credit_cards
credit_card_id VARCHAR(15)
user_id INT(11)
iban VARCHAR(35)
pan VARCHAR(20)
pin CHAR(4)
cvv CHAR(4)
track1 VARCHAR(255)
track2 VARCHAR(255)
expiring_date VARCHAR(10)
Indexes
PRIMARY
1
∞
∞
companies
company_id VARCHAR(20)
company_name VARCHAR(255)
phone VARCHAR(30)
email VARCHAR(100)
country VARCHAR(100)
website VARCHAR(255)
Indexes
PRIMARY
1
transactions
id VARCHAR(255)
credit_card_id VARCHAR(15)
company_id VARCHAR(20)
transaction_date TIMESTAMP
amount DECIMAL(10,2)
declined TINYINT(1)
product_ids VARCHAR(255)
user_id INT(11)
lat FLOAT
longitude FLOAT
Indexes
PRIMARY
company_id
credit_card_id
user_id
1
∞
transaction_product
transaction_id VARCHAR(255)
product_id INT(11)
Indexes
PRIMARY
product_id
∞
1
∞
users
user_id INT(11)
user_name VARCHAR(100)
user_surname VARCHAR(100)
phone VARCHAR(30)
email VARCHAR(100)
birth_date VARCHAR(100)
country VARCHAR(100)
city VARCHAR(100)
postal_code VARCHAR(10)
address VARCHAR(255)
Indexes
PRIMARY
1
products
product_id INT(11)
product_name VARCHAR(255)
price VARCHAR(30)
colour VARCHAR(30)
weight FLOAT
warehouse_id VARCHAR(10)
Indexes
PRIMARY

Exercise 1.

The number of times each product has been sold.



```sql
-- Exercise 1
-- The number of times each product has been sold
with product_count as (
select tp.product_id, count(tp.product_id) as number_of_sold
from transaction_product as tp
join transactions as t on (tp.transaction_id = t.id)
where t.declined = 0
group by tp.product_id)

select p.product_id, p.product_name, coalesce(pc.number_of_sold, 0) as number_of_sold
from products as p
left join product_count as pc
using (product_id);
```

product_id	product_name	number_of_sold
1	Direwolf Stannis	51
2	Tarly Stark	56
3	duel tourney Lannister	43
4	warden south duel	0
5	skywalker ewok	42
6	dooku solo	0
7	north of Casterly	44
8	Winterfell	0

	Time	Action	Response
1	11:02:03	with product_count as (select tp.product_id, count(...	100 row(s) returned

*** Products from declined transactions were counted because, stricly speaking, they were not sold.*

Unsold products were assigned a zero value.