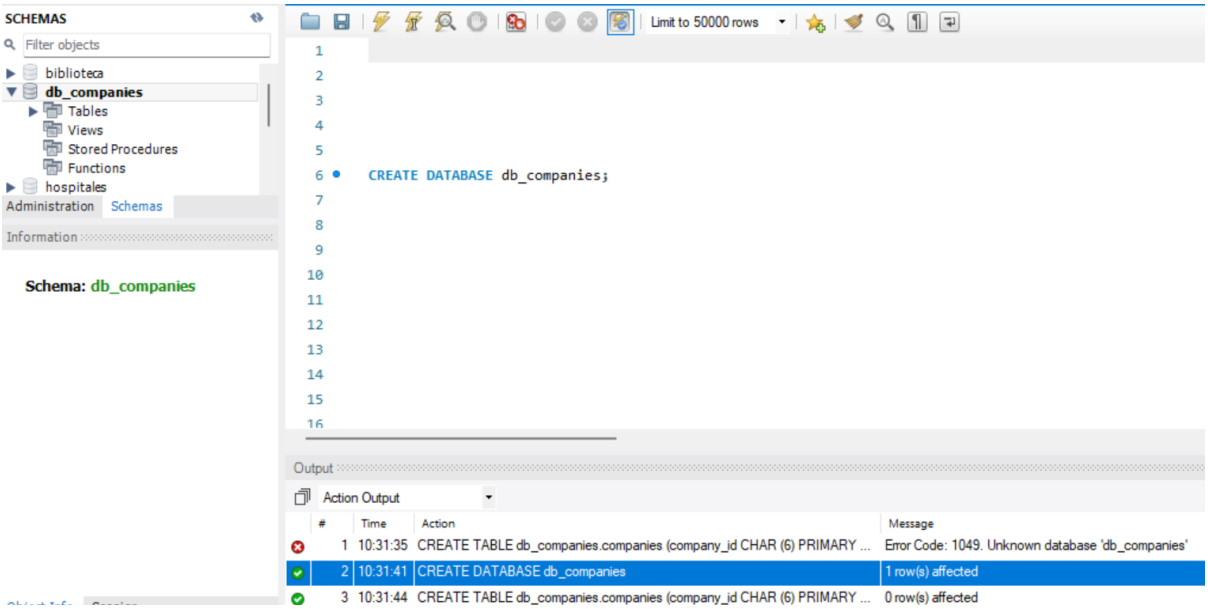
Sprint 4

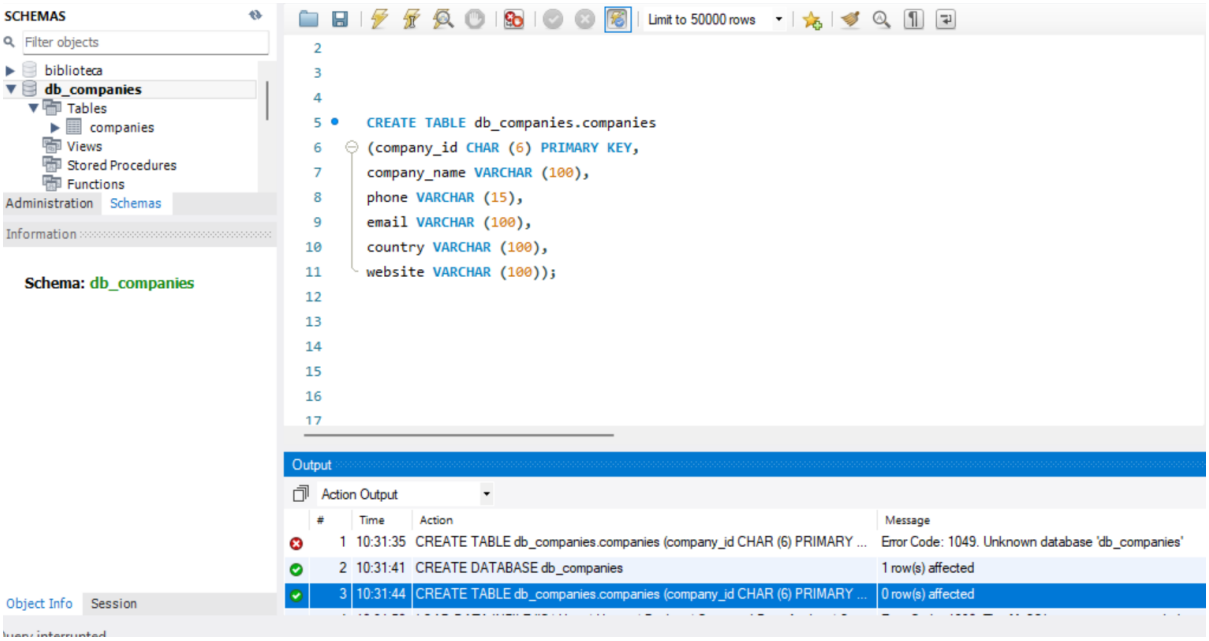
Creación de bases de datos

Nivell 1

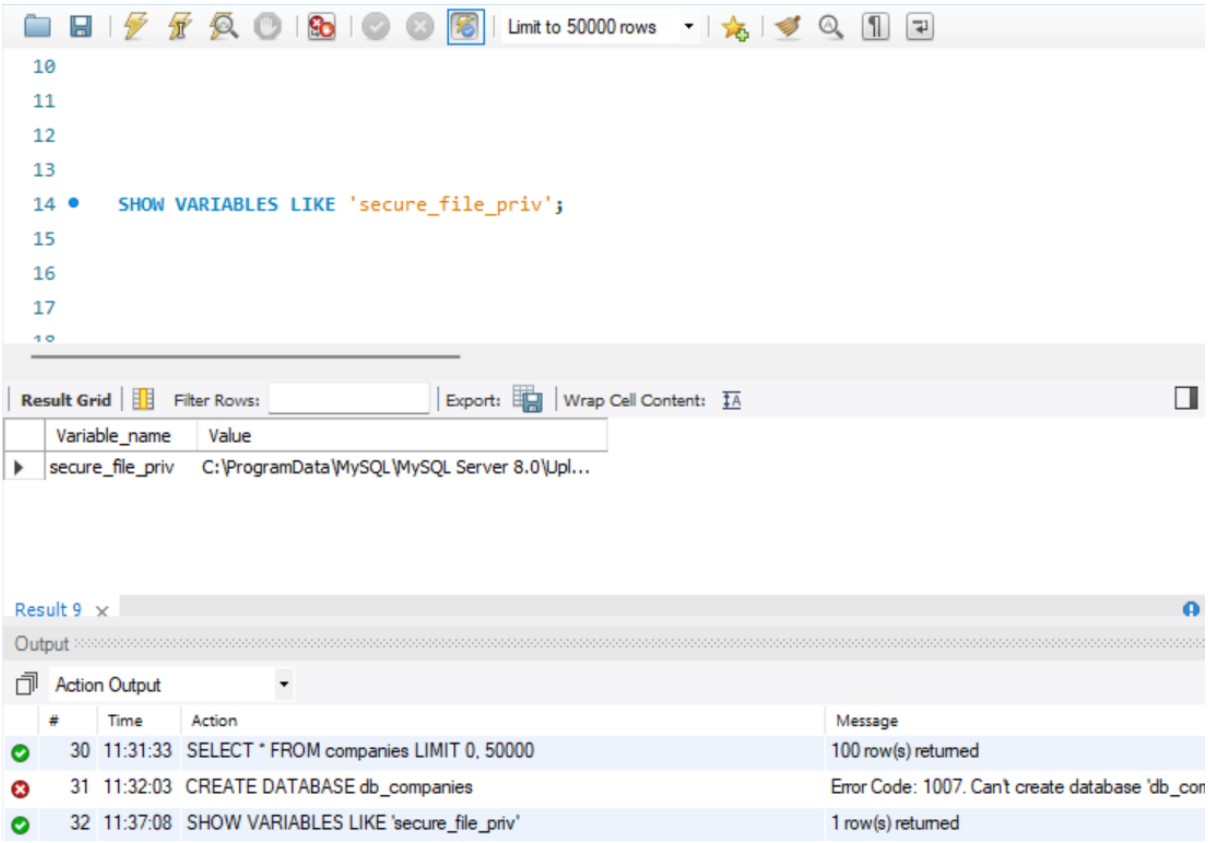
Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguis realitzar les següents consultes:



Se crea la base de datos llamada db_companies para guardar las 4 tablas a las que luego le queremos hacer consultas a través de un diagrama de estrella.

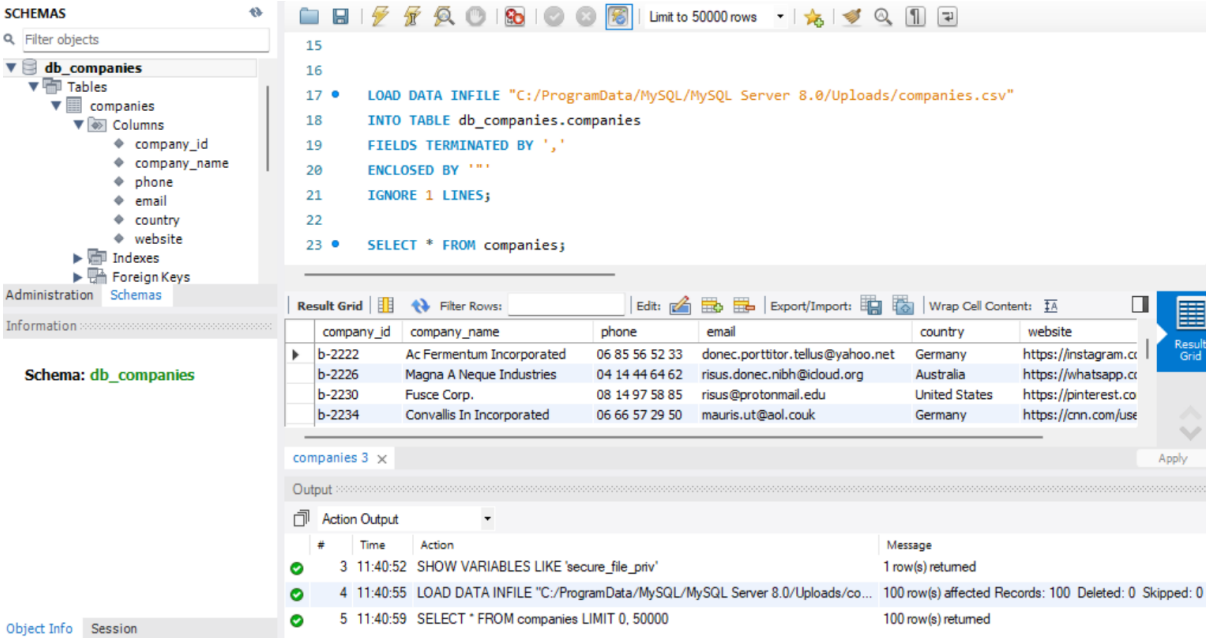


Se crea la tabla companies en la data base db_companies con las respectivas características de sus valores. (Con VARCHAR (100) se permiten todo tipo de valores hasta 100 características).



Para poder acceder a la carpeta que contiene el archivo de companies.csv se ha de crear un acceso seguro a través del comando secure_file_priv. Aqui se copia la ruta de acceso y se abre en carpetas, luego se guardan ahí los archivos a los que posteriormente vamos a querer acceder.

Despues se cargan estos datos, a través de esta ruta de acceso:



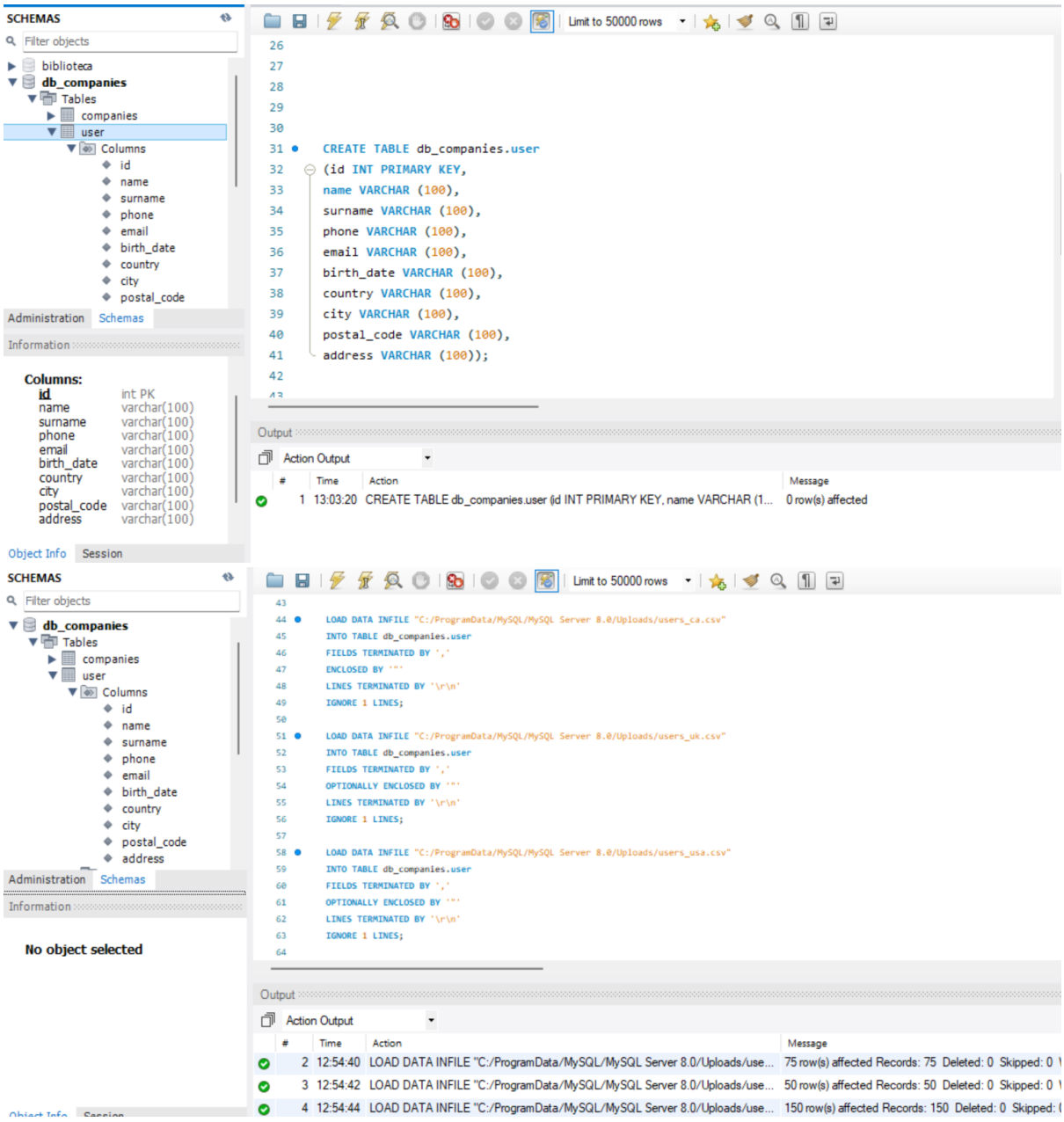
Se cargan adentro de la base de datos que hemos creado previamente y se establece la manera en la que queremos que Sql lea los valores contenidos en este archivo.

FIELDS TERMINATED BY ',' indica que los campos están terminados en coma (,).

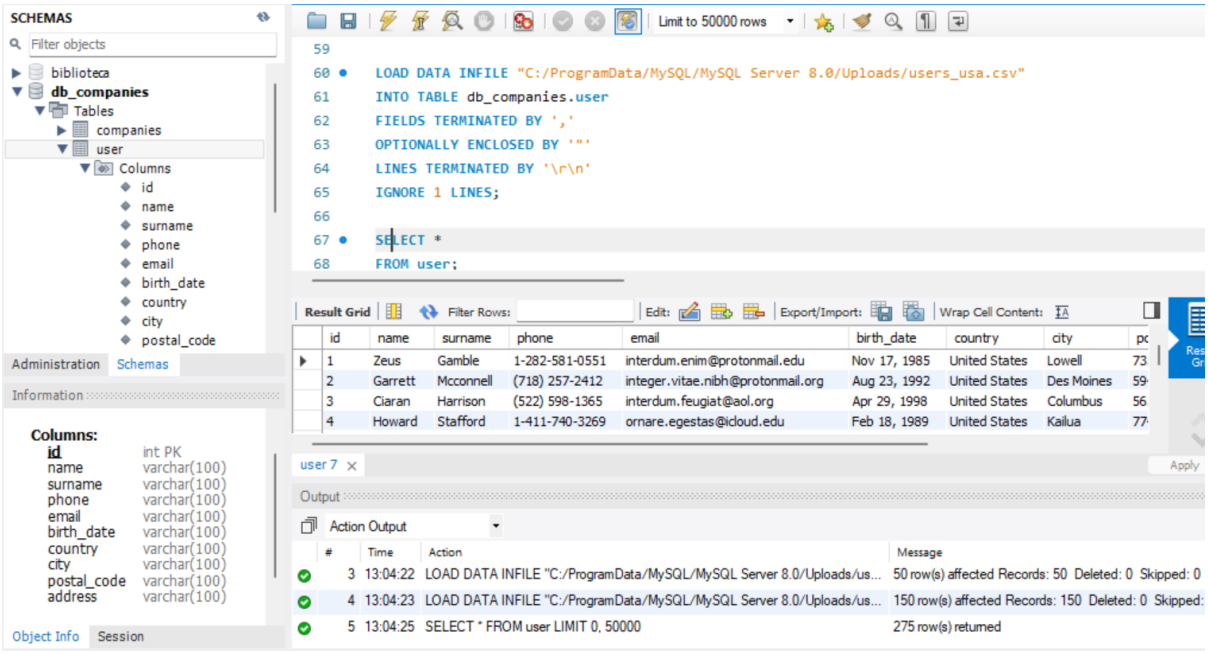
ENCLOSED BY '"" indica que otros están rodeados de comillas. Esto se establece para que a la hora de leer el archivo no se guarden los registros de manera incorrecta.

IGNORE 1 LINES indica que se ignore la primera línea de datos ya que esta contiene el nombre de las columnas y no los registros.
Despues leemos lo que contiene la nueva tabla companies con SELECT.
Estos pasos se repiten con las otras tablas que queremos insertar en la base de datos db_companies.

La **tabla user** incluye tres datasets de diferentes users de diferentes países:

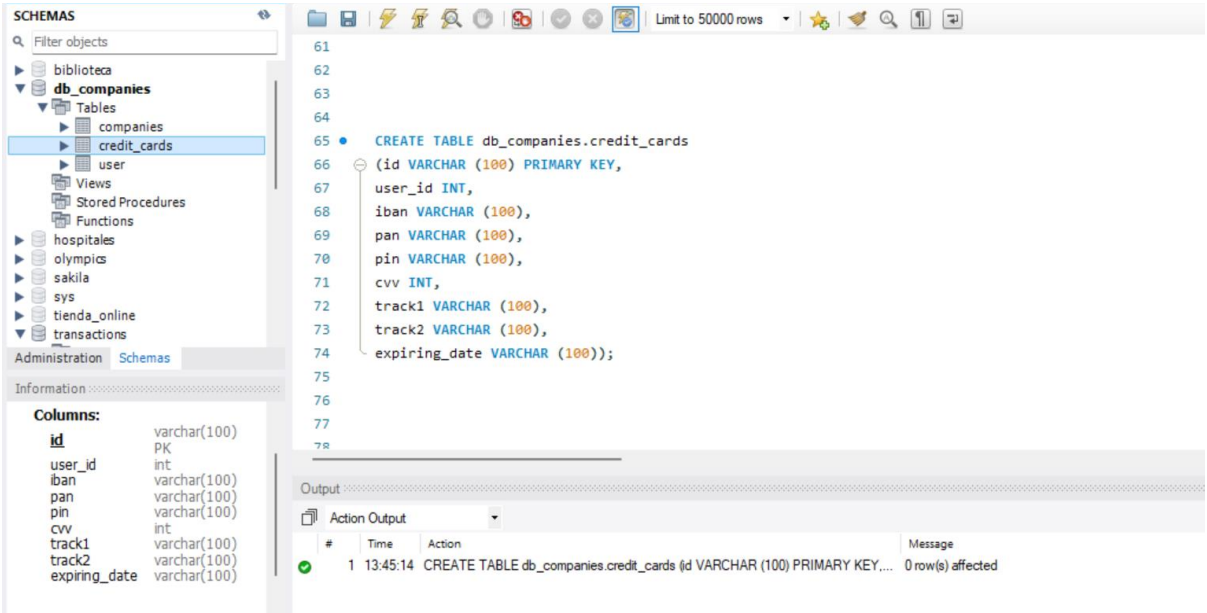


Aqui tuvimos que agregar la línea
LINES TERMINATED BY '\r\n', ya que los archivos csv proporcionados no culminan cada fila con una cierta estructura. Algunas terminan con comillas (”), otras con punto (.) Otras vacías, por lo cual hay que señalarle a sql que cada vez que encuentre un salto de línea (**\n**) → Carácter de salto de línea, se observa el siguiente registro.

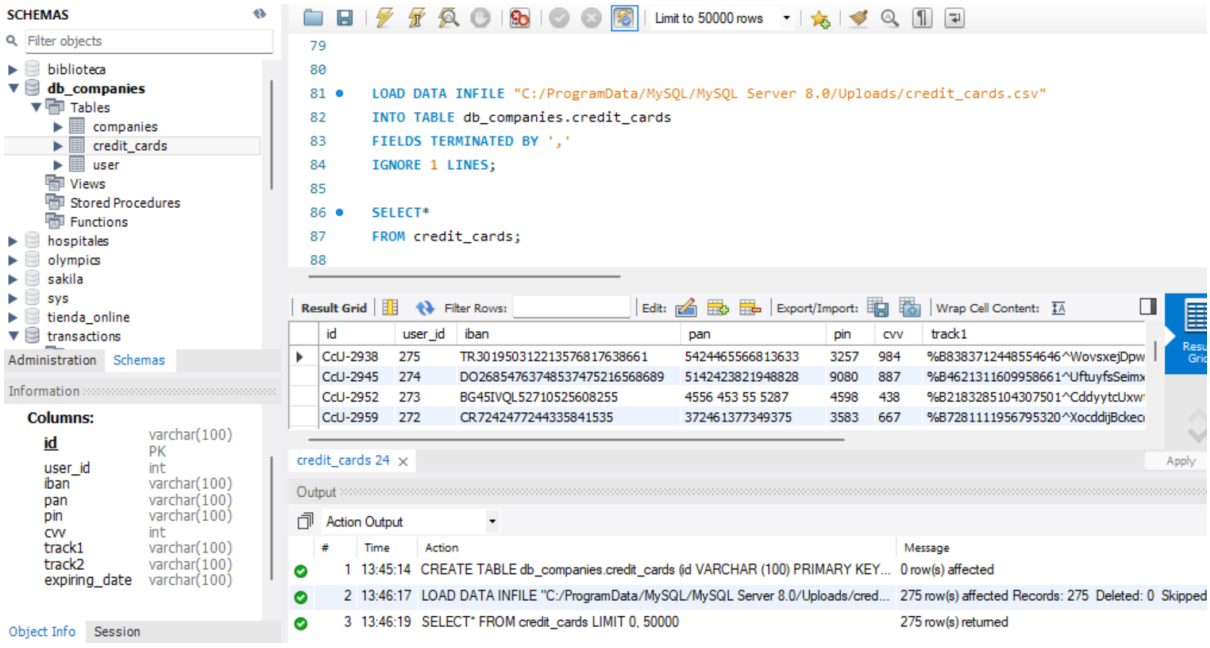


Se ve que todos los registros están ordenados según el numero de la Primary Key. El datatype de id de user es INT para poder garantizar este orden.

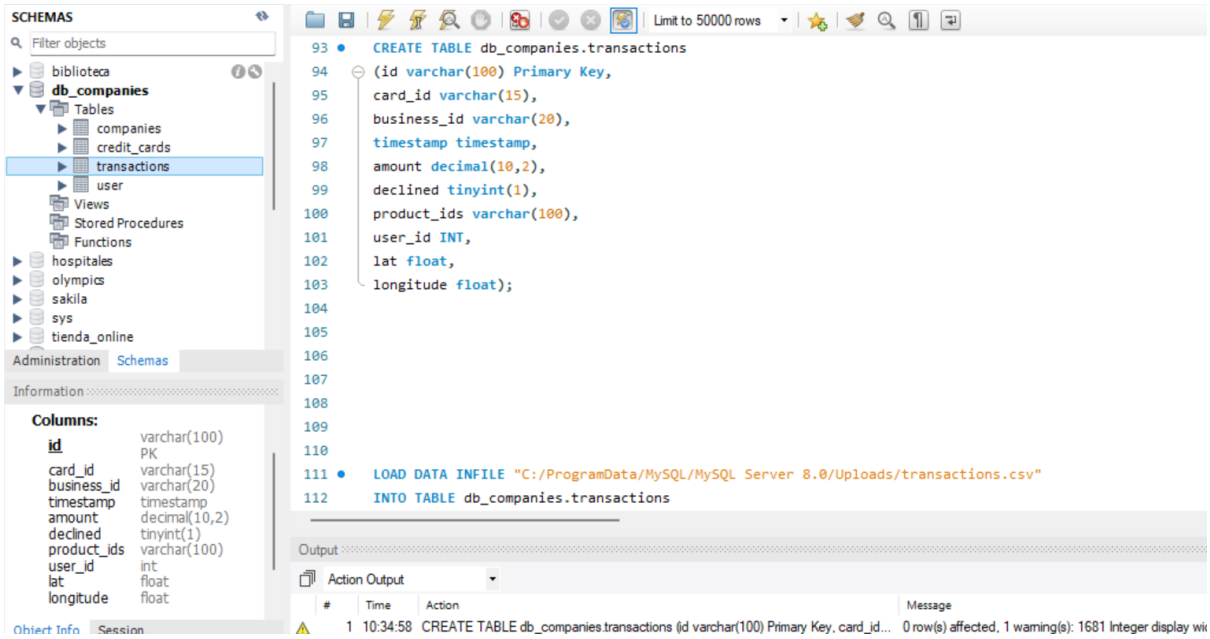
La tabla **credit_cards**:



Aquí cabe destacar que la columna user_id tiene la misma característica que la Primary Key de user para posteriormente poder hacer conexiones. (INT)



La tabla **transactions**:



Aqui opto por copiar el estilo de la tabla de transactions de otra data base.
Contiene valores como timestamp**,** con el datatype **timestamp** que es una hora y una fecha formateada.
Contiene un valor monetario (amount) a la que se le asigna **decimal (10,2)** lo que permite ingresar un total de 10 números con dos números después del decimal.

El datatype **tinyint(1)** es un boolean que almacena registros en función de que si son verdaderos o falsos.

Float permite valores numéricos con varios decimales cuando no se requiere una mayor precisión.

```
111 •  LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv"
112    INTO TABLE db_companies.transactions
113    FIELDS TERMINATED BY ';'
114    LINES TERMINATED BY '\r\n'
115    IGNORE 1 LINES;
116
117 •  SELECT *
118    FROM transactions;
119
120
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	card_id	business_id	timestamp	amount	declined	product_ids	u
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	2021-08-28 23:42:24	466.92	0	71, 1, 19	9
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	2021-07-26 07:29:18	49.53	0	47, 97, 43	1
063FBA79-99EC-66FB-29F7-25726D1764A5	CcU-2987	b-2250	2022-01-06 21:25:27	92.61	0	47, 67, 31, 5	2
0668296C-CDB9-A883-76BC-2E4C44F8C8AE	CcU-3743	b-2618	2022-01-26 02:07:14	394.18	0	89, 83, 79	2

transactions 26 | Apply

Output

Action Output

#	Time	Action	Message
1	13:58:03	CREATE TABLE db_companies.transactions (id varchar(100) Primary Key, card_j...	0 row(s) affected, 1 warning(s): 1681 Integer display wi
2	13:59:31	LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/tran...	587 row(s) affected Records: 587 Deleted: 0 Skipped
3	13:59:34	SELECT * FROM transactions LIMIT 0, 50000	587 row(s) returned

Se le agregan las Foreign Keys para obtener el diagrama de estrella:

Navigator

SCHEMAS

Filter objects

- biblioteca
- db_companies
  - Tables
    - companies
    - credit_cards
    - transactions
    - user
  - Views
  - Stored Procedures
  - Functions
- hospitales
- olympics
- sakila
- sys

Administration | Schemas

Information

Table: transactions

Columns:

id	varchar(100) PK
card_id	varchar(15)
business_id	varchar(20)
timestamp	timestamp
amount	decimal(10,2)
declined	tinyint(1)
product_ids	varchar(100)
user_id	int
lat	float

Object Info | Session

Sprint4* | Sprint 3 corregido

Limit to 50000 rows

```
105
106 •  ALTER TABLE transactions
107    ADD CONSTRAINT `transactionycredit`
108    FOREIGN KEY (`card_id`)
109    REFERENCES `credit_cards` (`id`);
110
111 •  ALTER TABLE transactions
112    ADD CONSTRAINT `transactionycompany`
113    FOREIGN KEY (`business_id`)
114    REFERENCES `companies` (`company_id`);
115
116 •  ALTER TABLE transactions
117    ADD CONSTRAINT `transactionyuser`
118    FOREIGN KEY (`user_id`)
119    REFERENCES `user` (`id`);
120
```

Output

Action Output

#	Time	Action	Message
1	10:34:00	ALTER TABLE transactions ADD CONSTRAINT `transactionycredit` FOREIGN ...	587 row(s) affected Records: 587 Duplicates: 0 Warnin
2	10:35:19	ALTER TABLE transaction ADD CONSTRAINT `transactionycompany` FOREIG...	Error Code: 1146. Table 'db_companies.transaction' doe
3	10:35:26	ALTER TABLE transactions ADD CONSTRAINT `transactionycompany` FOREI...	587 row(s) affected Records: 587 Duplicates: 0 Warnin
4	10:36:28	ALTER TABLE transactions ADD CONSTRAINT `transactionyuser` FOREIGN K...	587 row(s) affected Records: 587 Duplicates: 0 Warnin

Luego miramos los datos y regularizamos las columnas que contienen las fechas para poder trabajar con ellas:

```
123
124 •   ALTER TABLE transactions
125     MODIFY COLUMN timestamp date;
126
127 •   SELECT *
128     FROM transactions;
129
130
131
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	card_id	business_id	timestamp	amount	declined	product_ids	user_id
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	2021-08-28	466.92	0	71, 1, 19	92
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	2021-07-26	49.53	0	47, 97, 43	170
063FBA79-99EC-66FB-29F7-25726D1764A5	CcU-2987	b-2250	2022-01-06	92.61	0	47, 67, 31, 5	275
0668296C-CDB9-A883-76BC-2E4C44F8C8AE	CcU-3743	b-2618	2022-01-26	394.18	0	89, 83, 79	265

transactions 9 | Apply

Output

Action Output

#	Time	Action	Message
6	11:40:50	UPDATE transactions SET timestamp = STR_TO_DATE(timestamp, '%m/%d/%...	Error Code: 1175. You are using safe update r
7	11:41:16	ALTER TABLE transactions MODIFY COLUMN timestamp date	0 row(s) affected Records: 0 Duplicates: 0 W
8	11:41:39	SELECT * FROM transactions LIMIT 0, 50000	587 row(s) returned

SCHEMAS

Filter objects

- biblioteca
- db_companies
  - Tables
    - companies
    - credit_cards
    - estado_tarjetas
    - productos
    - transactions
    - transactions_products
    - user

Administration | Schemas

Information

Table: credit_cards

Columns:

id	varchar(100) PK
user_id	int
iban	varchar(100)
pan	varchar(100)
pin	varchar(100)
cvv	int
track1	varchar(100)
track2	varchar(100)
expiring_date	varchar(100)

Limit to 50000 rows

```
127
128 •   UPDATE credit_cards
129     SET expiring_date = STR_TO_DATE(expiring_date, '%m/%d/%Y');
130
131 •   SELECT *
132     FROM credit_cards;
133
134
135
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	pin	cvv	track1	track2	expiring_date
5813633	3257	984	%B8383712448554646^WovsxejDpwiev^8604...	%B7653863056044187=80071633363?3	2022-10-30
1948828	9080	887	%B4621311609958661^UftuyfsSeimxn^06106...	%B4149568437843501=51071403303?1	2023-08-24
5 5287	4598	438	%B2183285104307501^CddyytcUxwfdq^5907...	%B6778580257827162=69068597400?7	2021-06-29
349375	3583	667	%B7281111956795320^XocddjBckecd^09016...	%B4246154489281853=28052239167?8	2023-02-24

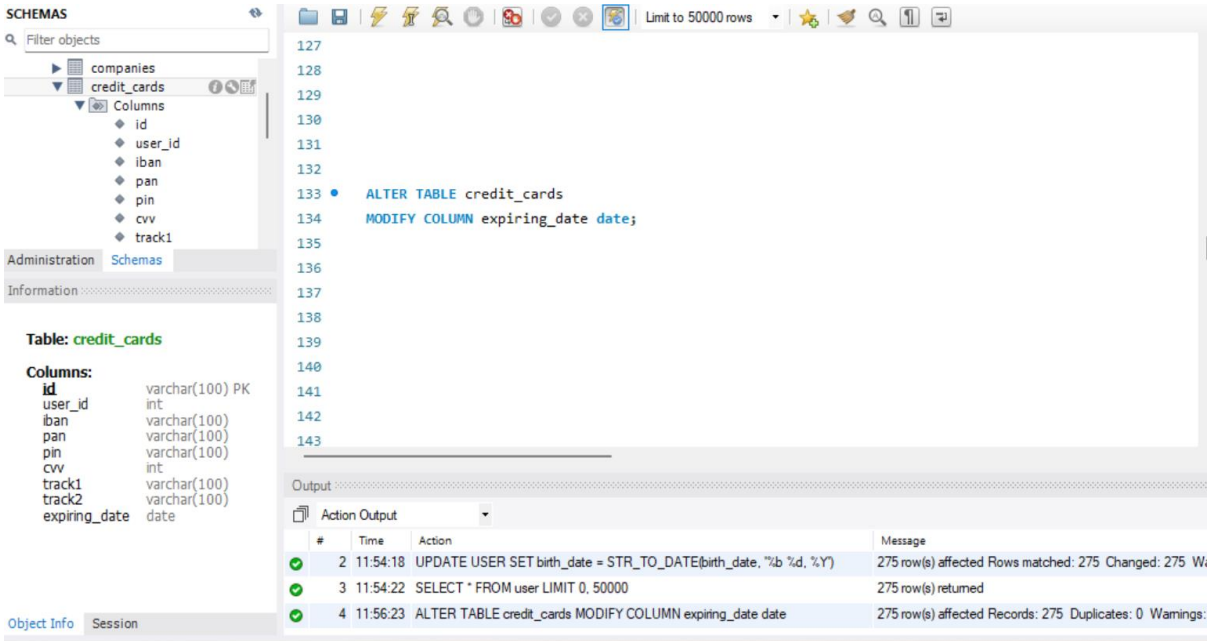
credit_cards 1 | Apply

Output

Action Output

#	Time	Action	Message
1	11:49:56	UPDATE credit_cards SET expiring_date = STR_TO_DATE(expiring_date, '%m/...	275 row(s) affected Rows matched: 275 Ch
2	11:49:58	SELECT * FROM credit_cards LIMIT 0, 50000	275 row(s) returned

Luego de formatear la manera en la que se escribieron las fechas, modificamos el datatype en la que se guardaron para que sean fechas




```
131
132
133 •  UPDATE USER
134    SET birth_date = STR_TO_DATE(birth_date, '%b %d, %Y');
135
136 •  SELECT *
137    FROM user;
138
139
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

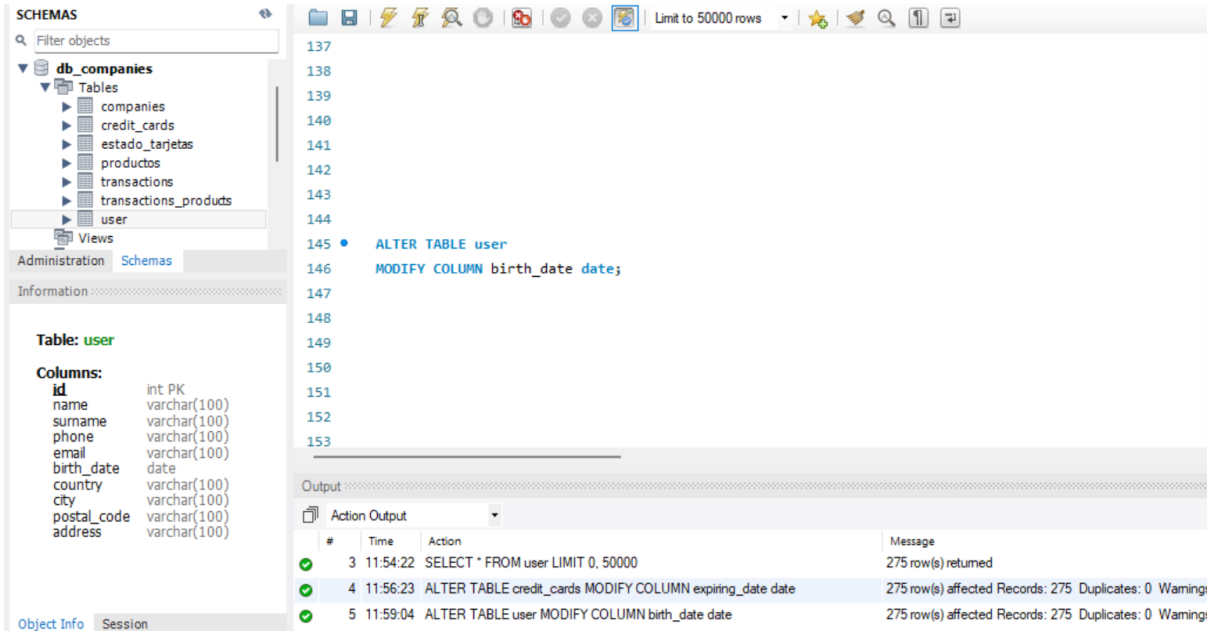
name	phone	email	birth_date	country	city	postal_code	address
ible	1-282-581-0551	interdum.enim@protonmail.edu	1985-11-17	United States	Lowell	73544	348-7818 Sag
onnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	1992-08-23	United States	Des Moines	59464	903 Sit Ave
ison	(522) 598-1365	interdum.feugiat@aol.org	1998-04-29	United States	Columbus	56518	736-2063 Tel
ford	1-411-740-3269	ornare.egestas@icloud.edu	1989-02-18	United States	Kailua	77417	Ap #545-224

user 2 | Apply

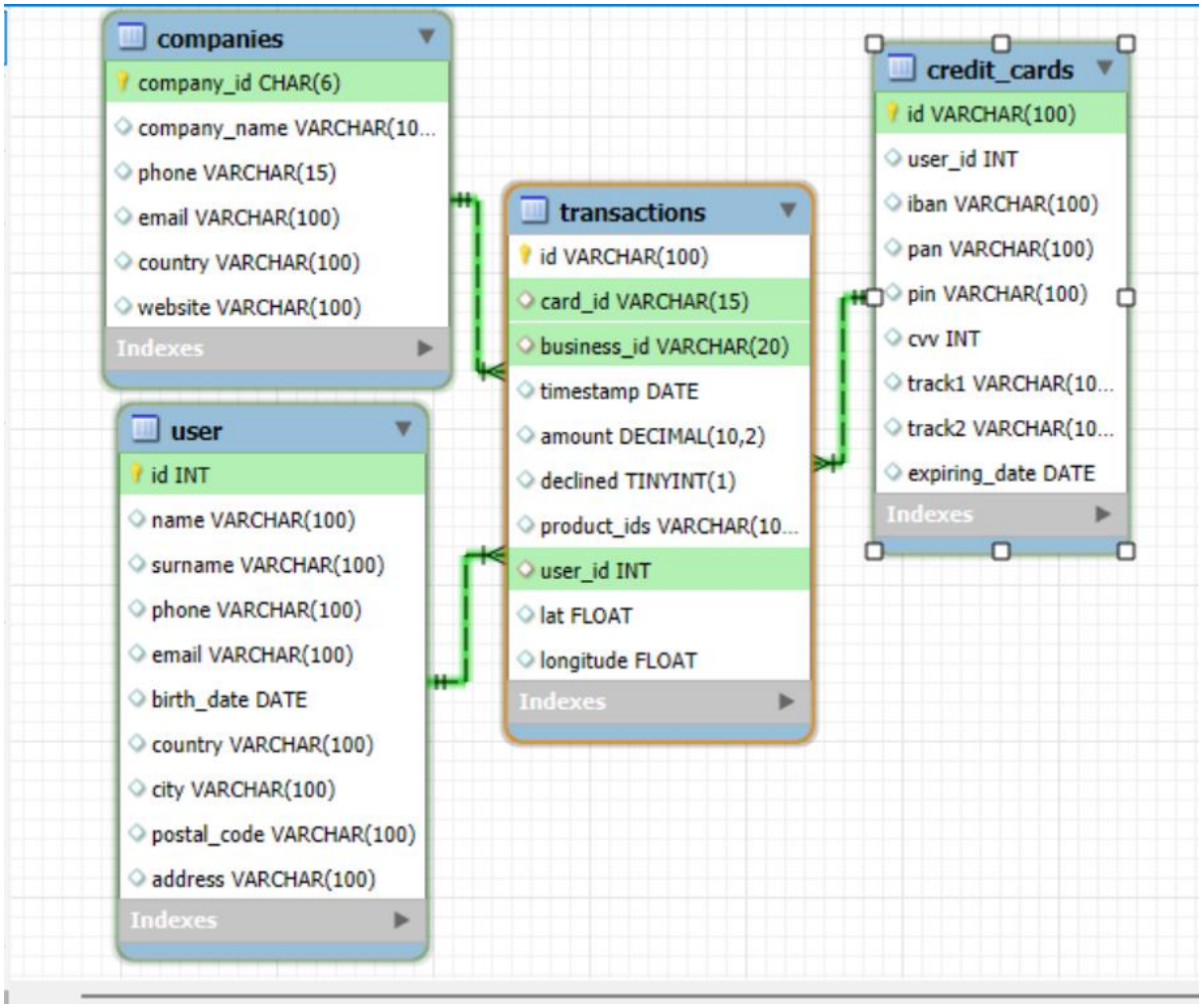
Output

Action Output

#	Time	Action	Message
1	11:54:05	UPDATE USER SET birth_date = STR_TO_DATE(birth_date, '%b %d %Y')	Error Code: 1411. Incorrect datetime value:
2	11:54:18	UPDATE USER SET birth_date = STR_TO_DATE(birth_date, '%b %d, %Y')	275 row(s) affected Rows matched: 275 Ch
3	11:54:22	SELECT * FROM user LIMIT 0, 50000	275 row(s) returned



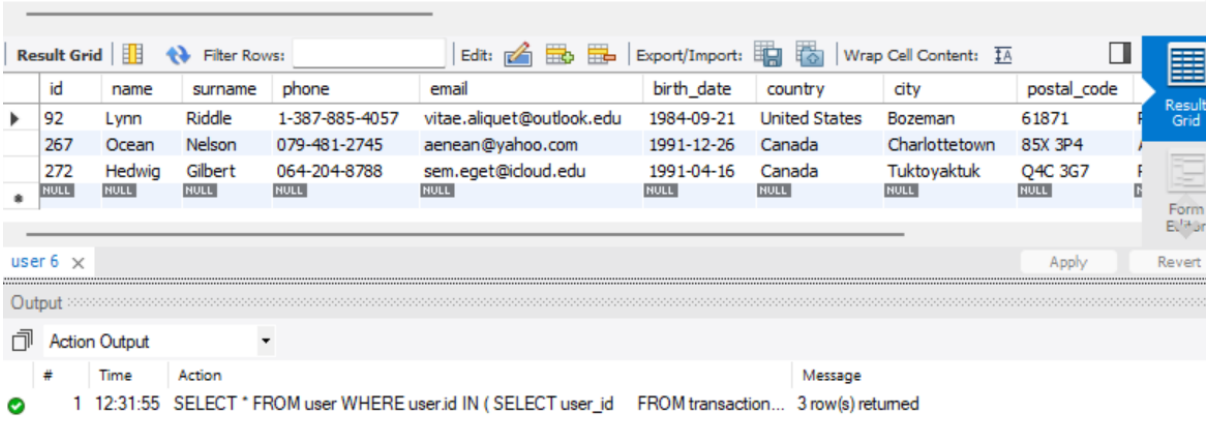
Llegamos a este diagrama en la que la tabla de transacciones se sitúa en la mitad y las otras tablas de dimensiones con las Primary Keys están alrededor:



- Exercici 1

Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.

```
155
156 •   SELECT *
157     FROM user
158   WHERE user.id IN ( SELECT user_id
159         FROM transactions
160         WHERE declined = 0
161         GROUP BY user_id
162         HAVING COUNT(id) > 30);
163
```

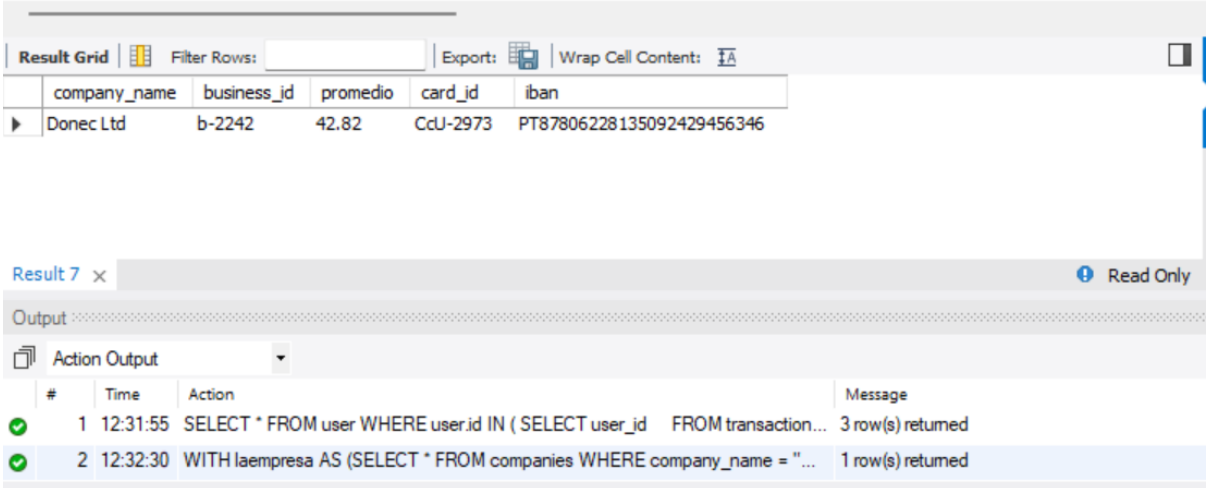


Aqui podemos ver que hay 3 usuarios que han efectuado (declined = 0) más que 30 transacciones.

- Exercici 2

Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.

```
167
168 •   WITH laempresa AS (SELECT *
169     FROM companies
170     WHERE company_name = "Donec Ltd")
171     SELECT company_name, business_id, round(avg(amount),2) as promedio, card_id, iban
172     FROM transactions
173     JOIN laempresa ON laempresa.company_id = transactions.business_id
174     JOIN credit_cards ON credit_cards.id = transactions.card_id
175     WHERE declined = 0
```



Aqui podemos ver que solo 1 credit_card ha efectuado (declined=0) una transaccion a la empresa Donec Ltd.

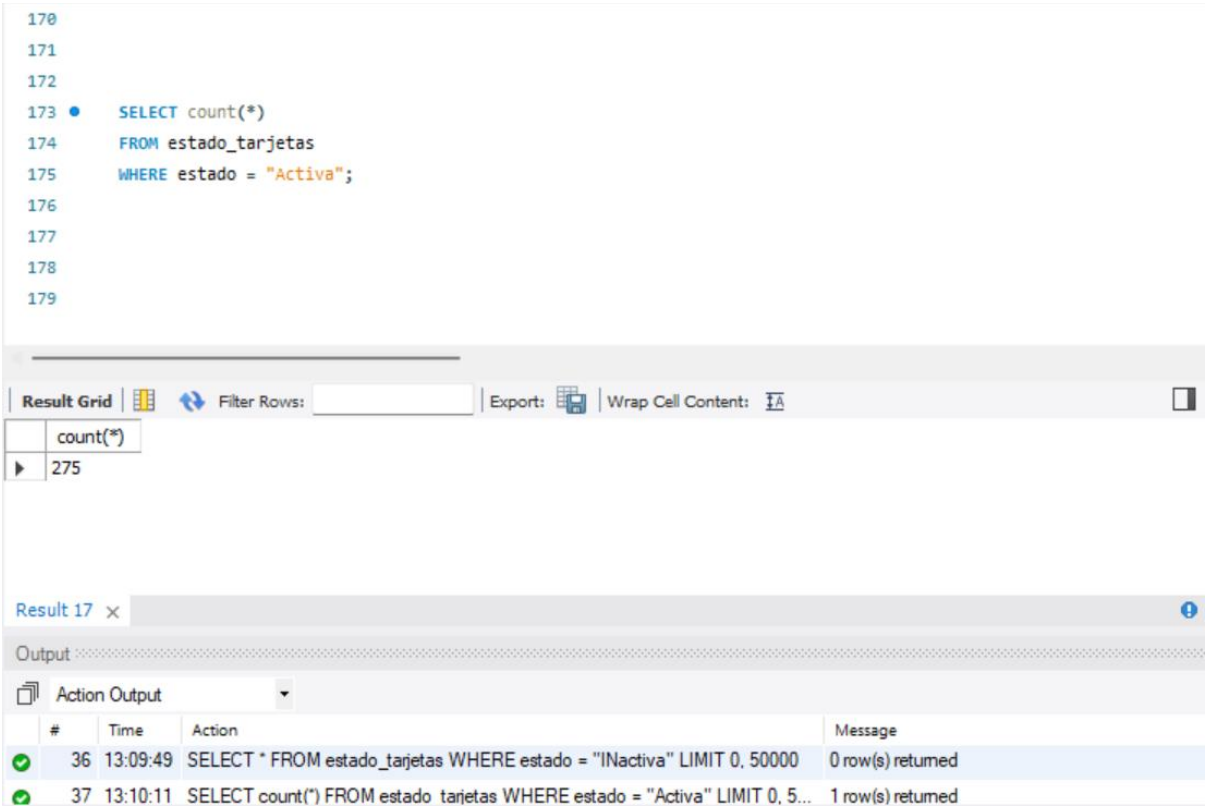
Aqui lo primero que hacemos es **CREATE** la tabla **estado_tarjeta** que adopta los valores de las consultas que siguen. Esta tabla contendrá card_id y asignara una etiqueta de **estado** a cada card_id según si fueron declinadas las últimas tres veces o no.

(WHEN **COUNT**(*) = 3 AND SUM(declined = 1) = 3 THEN '**Inactiva**'
ELSE **'Activa'**
END AS **estado)**

Posteriormente se agrega con la segunda subconsulta de la que se quiere sacar los datos. En esta solo se tienen en cuenta las últimas tres filas según su timestamp.

FROM (SELECT card_id, declined, ROW_NUMBER() OVER (PARTITION BY card_id
ORDER BY timestamp DESC) AS rn
FROM transactions) AS ultimas
WHERE rn <= 3
GROUP BY card_id;

Solo se busca las ultimas rows que tienen que tener el numero 1, 2 y 3. **WHERE rn <= 3**



275 tarjetas están activas. (todas)

También se crea una Foregin Key con la tabla de credit_cards:

```
167
168
169
170
171
172
173
174 ●   ALTER TABLE estado_tarjetas
175     ADD CONSTRAINT `creditcardyestado`
176     FOREIGN KEY (`card_id`)
177     REFERENCES `credit_cards` (`id`);
178
179
180
181
182
183
184
```

Output

Action Output

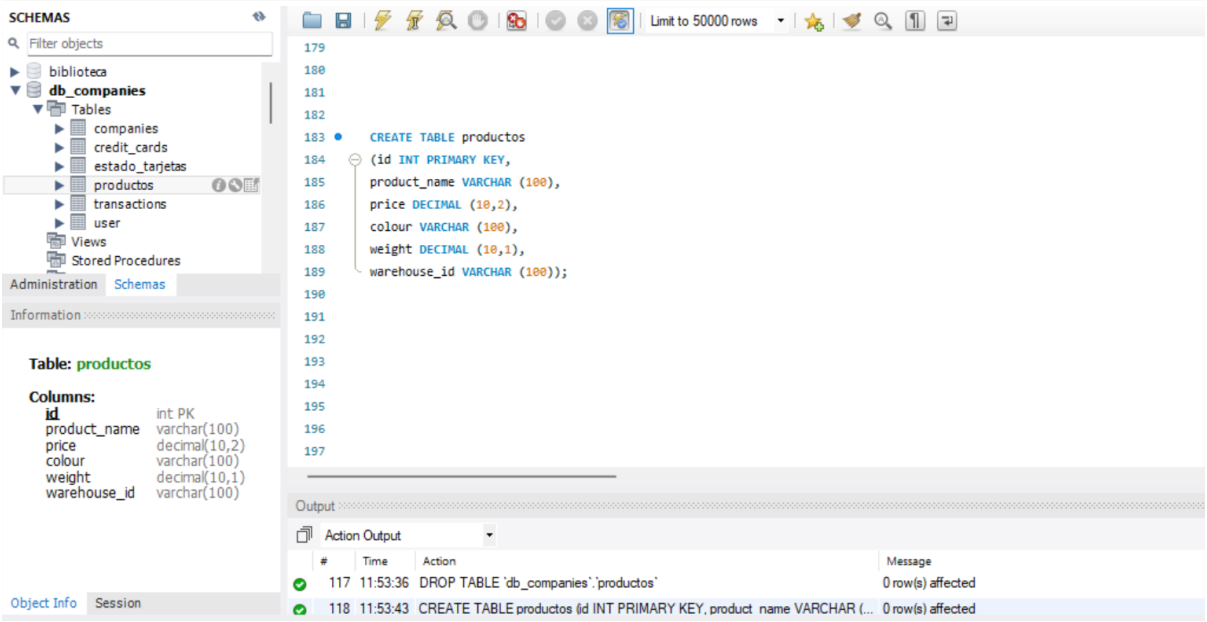
#	Time	Action	Message
151	12:58:24	CREATE TABLE estado_tarjetas AS SELECT card_id, CASE WHEN COUNT(*)...	Error Code: 1064. You have an error in your SQL synt
152	13:01:53	ALTER TABLE estado_tarjetas ADD CONSTRAINT `creditcardyestado` FOREIGN ...	275 row(s) affected Records: 275 Duplicates: 0 War

Nivell 3

Crea una taula amb la qual puguem unir les dades del nou arxiu products.csv amb la base de dades creada, tenint en compte que des de transaction tens product_ids. Genera la següent consulta:

Exercici 1

Necessitem conèixer el nombre de vegades que s'ha venut cada producte.



```
191
192 •  LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv"
193    INTO TABLE db_companies.productos
194    FIELDS TERMINATED BY ','
195    LINES TERMINATED BY '\n'
196    IGNORE 1 LINES
197    (id, product_name, @precio, colour, weight, warehouse_id)
198    SET price = REPLACE(@precio, '$', '');
199
200 •  SELECT *
201    FROM productos;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	product_name	price	colour	weight	warehouse_id
1	Direwolf Stannis	161.11	#7c7c7c	1.0	WH-4
2	Tarly Stark	9.24	#919191	2.0	WH-3
3	duel tourney Lannister	171.13	#d8d8d8	1.5	WH-2
4	warden south duel	71.89	#111111	3.0	WH-1
5	skywalker ewok	171.22	#dbdbdb	3.2	WH-0

productos 48 ×

Output

Action Output

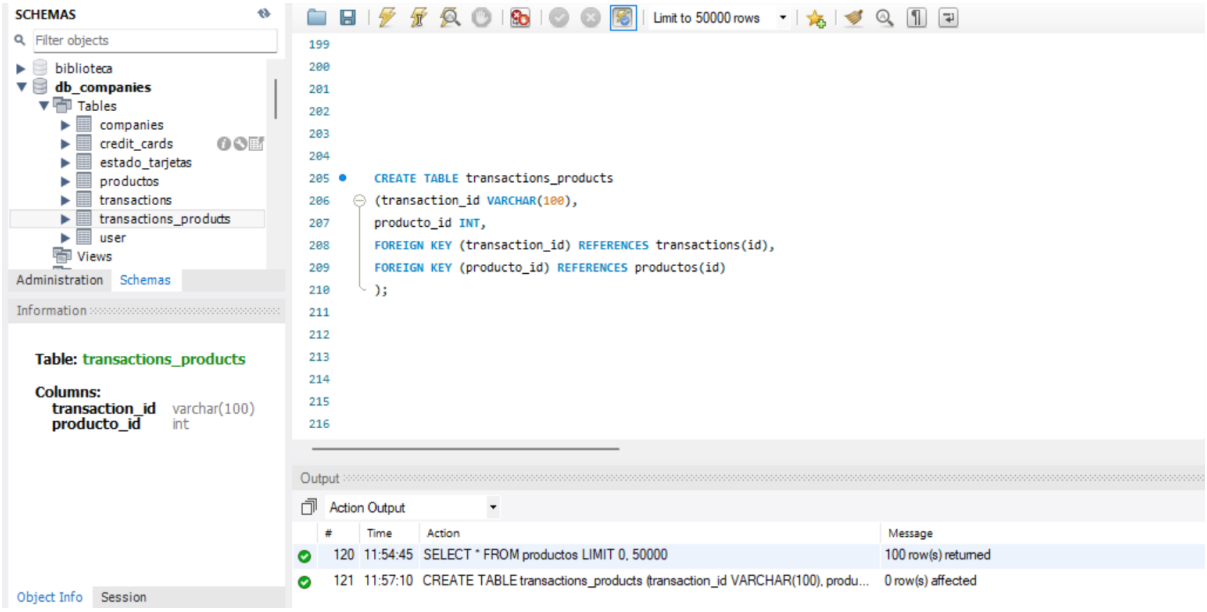
#	Time	Action	Message
119	11:54:42	LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/pr...	100 row(s) affected Records: 100 Deleted: 0
120	11:54:45	SELECT * FROM productos LIMIT 0, 50000	100 row(s) returned

Primero se crea la tabla en la que se quieren insertar los datos de productos.csv. Luego se insertan los datos teniendo en cuenta que el precio que esta guardado en el archivo con $, se reemplaza con un espacio en blanco para que no genere problemas.

(id, product_name, @precio, colour, weight, warehouse_id)

SET price = REPLACE(@precio, '$', '');

Se verifica que se hayan insertado correctamente.



Se crea una nueva tabla intermediaria para guardar las transacciones separadas por sus product_ids. Ya que en la tabla original de transacciones los product_ids están

guardadas en una sequencia separada por ,.

```
197
198
199
200
201 ●    SELECT *
202      FROM transactions;
203
204
205
206
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	card_id	business_id	timestamp	amount	declined	product_ids	user_id	lat	longitude
2986D3B02	CcU-2938	b-2362	2021-08-28 23:42:24	466.92	0	71, 1, 19	92	81.9185	-12.5276
689713128	CcU-4219	b-2302	2021-07-26 07:29:18	49.53	0	47, 97, 43	170	-43.9695	-117.525
26D1764A5	CcU-2987	b-2250	2022-01-06 21:25:27	92.61	0	47, 67, 31, 5	275	-81.2227	-129.05
C44F8C8AE	CcU-3743	b-2618	2022-01-26 02:07:14	394.18	0	89, 83, 79	265	-34.3593	-100.556

transactions 49 ×

Output

Action Output

#	Time	Action	Message
121	11:57:10	CREATE TABLE transactions_products (transaction_id VARCHAR(100), produ...	0 row(s) affected
122	11:59:34	SELECT * FROM transactions LIMIT 0, 50000	587 row(s) returned

Para que se separen los product_ids hay que aplicar una función:

SCHEMAS

Filter objects

- companies
- credit_cards
- estado_tarjetas
- productos
- transactions
- transactions_products
- user
- Views
- Stored Procedures

Administration | Schemas

Information

Table: transactions_products

Columns:
transaction_id varchar(100)
producto_id int

Object Info | Session

Limit to 50000 rows

```
235 ●  INSERT INTO transactions_products (transaction_id, producto_id)
236    SELECT transactions.id AS transaction_id,
237    productos.id AS producto_id
238    FROM transactions
239    JOIN productos
240    ON FIND_IN_SET(productos.id, REPLACE(transactions.product_ids, ' ', ''))
241    WHERE declined = 0;
242
243 ●  SELECT *
244    FROM transactions_products;
```

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

transaction_id	producto_id
02C6201E-D90A-1859-B4EE-88D2986D3B02	1
02C6201E-D90A-1859-B4EE-88D2986D3B02	19
02C6201E-D90A-1859-B4EE-88D2986D3B02	71
0466A42E-47CF-8D24-FD01-C0B689713128	43
0466A42E-47CF-8D24-FD01-C0B689713128	47

transactions_products 8 ×

Output

Action Output

#	Time	Action	Message
2	12:32:30	WITH laempresa AS (SELECT * FROM companies WHERE company_name = ...	1 row(s) returned
3	12:35:49	SELECT * FROM transactions_products LIMIT 0, 50000	1236 row(s) returned

Query Completed

Insertamos directamente en la tabla posteriormente creada cada transaccion y cada producto_id.

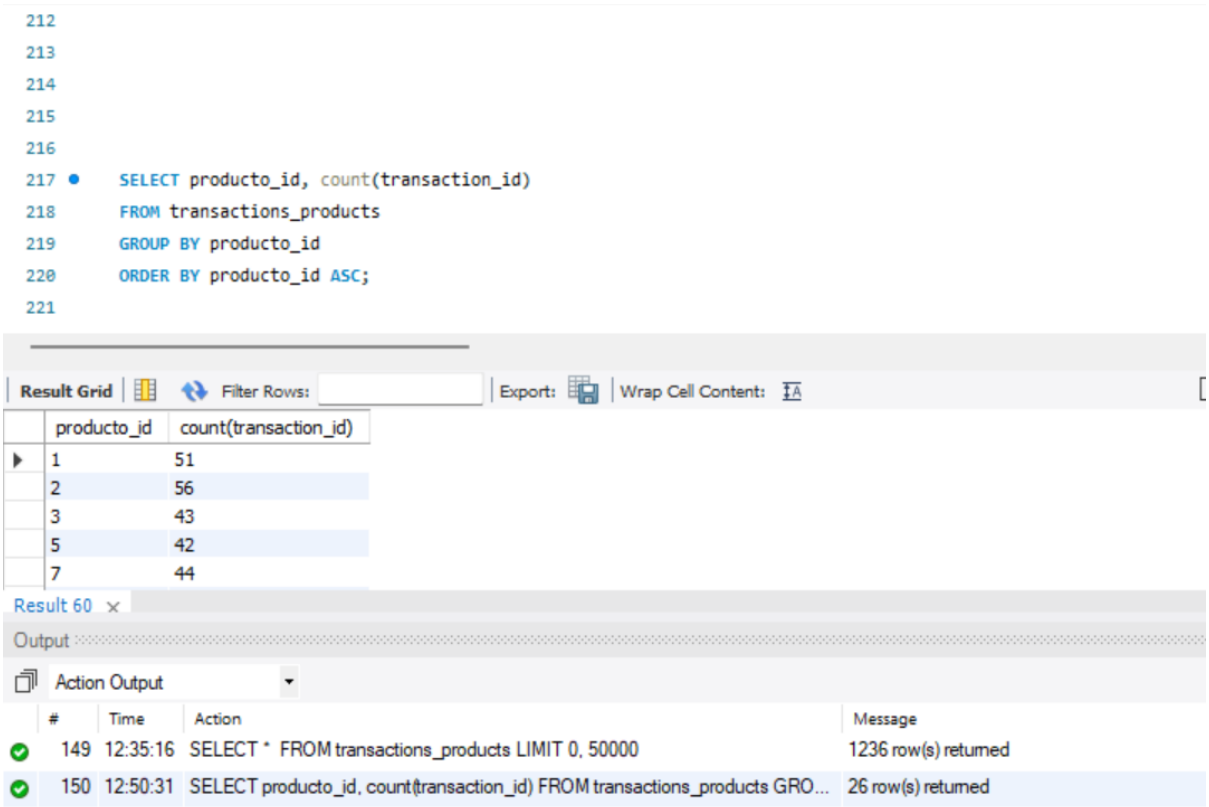
SELECT transactions.id AS transaction_id,

productos.id AS producto_id
FROM transactions
JOIN productos
ON FIND_IN_SET(productos.id,
REPLACE(transactions.product_ids, ' ', ''))
WHERE declined = 0;

Aquí el **JOIN** se hace para que la tabla resultante contenga valores contrastados a través de los productos.id. Si un numero de productos.id está contenida en transactions los encuentra en la columna y cadena de valores product_id de transactions (**FIND_IN_SET**).

Sin embargo, para que SQL pueda leer la cadena de valores hay que quitarles el espacio después de cada coma. (**REPLACE(transactions.product_ids, ' ', '')**

La tabla resultante es una que contiene varias veces 1 transaction.id (transaction_id) con varios valores de productos separados, si es que ese producto estaba inicialmente en la tabla de productos.



Luego se cuentan las transaction_id de la nueva tabla y se ordena por producto_id. Eso cuenta cuantas veces aparece una transaction_id por cada numero (producto_id).