

José Messias Garcia da Silva Ferreira

SPRINT 4 - NIVEL 1

Totes les transformacions i importacions necessàries en aquesta tasca s'han de fer mitjançant codi SQL. NO ES PERMET utilitzar el Wizard. Cal descriure cada pas i fer una captura de pantalla.

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:

He comenzado analizando las tablas utilizando Google Sheets y Text Editor, y noté que muchos campos estaban en diferentes formatos, como *price* en la tabla *products* y *birth_date* en la tabla *users*. También revisé la presencia de valores nulos en las tablas. Otro inconveniente era la separación de registros, ya que en la tabla *transactions* se utilizaba ";", mientras que en otras tablas se empleaba ",".

Antes de trabajar con la base de datos, seleccioné las tablas que quería analizar: *transactions*, *credit_cards*, *products*, *companies*, *european_users* y *american_users*.

Posteriormente, comencé a trabajar en SQL, importando las tablas. Decidí trabajar con las seis tablas mencionadas, incluyendo *american_users* y *european_users*, que luego uniré mediante *SELECT ... UNION ALL*.

Al crear la tabla *transactions* (tabla principal del modelo estrella), añadí seis *foreign keys* con la opción ON DELETE CASCADE, de manera que si se elimina una entidad, sus registros relacionados también se borren automáticamente.

Creé la base de datos usando *CREATE DATABASE* y la nombré **EuroCommerce**. Dado que no estaba permitido limpiar o modificar los datos, asigné el tipo *VARCHAR* a la mayoría de los campos, excepto:

- *weight*: valores decimales.
- *lat* y *longitude*: tipo *FLOAT* en la tabla *transactions*.

Además, apliqué la restricción *UNIQUE* en el campo *email* de las tablas *companies*, *european_users* y *american_users*. Tras crear cada tabla con *CREATE TABLE*, verifiqué las columnas con *SHOW COLUMNS FROM*.

Para cargar los datos, utilicé LOAD DATA LOCAL INFILE ... INTO TABLE, considerando FIELDS TERMINATED BY ',' y ENCLOSED BY '\"'. Después de cargar european_users y american_users, añadí una columna con el continente correspondiente y unifiqué ambas tablas en data_users usando UNION ALL.

En la tabla transactions, el campo *product_ids* contenía múltiples valores (hasta 5 en algunos casos). Para manejar esta relación muchos a muchos, creé una tabla puente (transactions_products) entre transactions y products, utilizando *foreign keys*. Esta tabla se utiliza en el último ejercicio del nivel 3 por eso decidí utilizar todas las tablas de la base de datos...

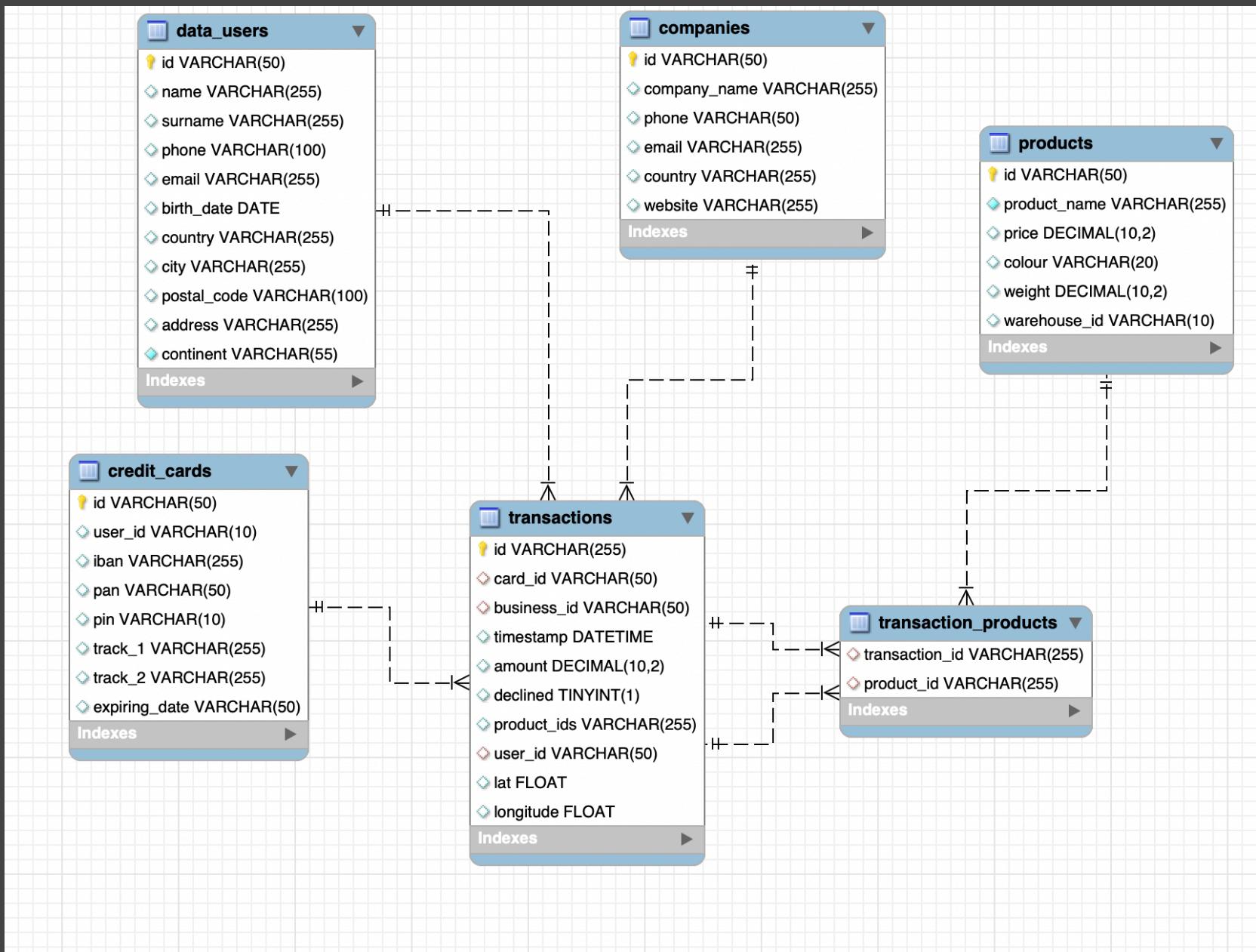
En la tabla data_users, añadí la *primary key* id y la *foreign key* user_id en transactions, y eliminé las tablas originales european_users y american_users.

Para procesar *product_ids*, utilicé CONCAT y REPLACE para eliminar espacios y añadir corchetes al inicio y final de los valores. Luego, con JSON, extraje los valores y los transferí a la tabla puente transactions_products.

Adicionalmente, realicé pequeñas modificaciones en las tablas:

- En products, eliminé el símbolo "\$" de price y cambié su tipo a DECIMAL.
- Normalicé los nombres de productos usando UPPER, LOWER y SUBSTRING, asegurando que la primera letra fuera mayúscula.
- En data_users, convertí birth_date a tipo DATE y apliqué STR_TO_DATE para un formato más legible.

Todos los pasos anteriores están claramente comentados en el script SQL, facilitando la comprensión y seguimiento.



DATABASE CREATION

```
1
2 • CREATE DATABASE EuroCommerce;
3 • USE EuroCommerce;
4
5 -- TABLES CREATION
6
7 • CREATE TABLE products (
8     id VARCHAR(50) PRIMARY KEY,
9     product_name VARCHAR(255) NOT NULL,
10    price VARCHAR(50),
11    colour VARCHAR(20),
12    weight DECIMAL(10,2),
13    warehouse_id VARCHAR(10)
14 );
15
16
17 • CREATE TABLE credit_cards (
18     id VARCHAR(50) PRIMARY KEY,
19     user_id VARCHAR(10),
20     iban VARCHAR(255),
21     pan VARCHAR(50),
22     pin VARCHAR(10),
23     track_1 VARCHAR(255),
24     track_2 VARCHAR(255),
25     expiring_date VARCHAR(50)
26 );
27
28 • CREATE TABLE companies (
29     id VARCHAR(50) PRIMARY KEY NOT NULL,
30     company_name VARCHAR(255),
31     phone VARCHAR(50),
32     email VARCHAR(255) UNIQUE,
33     country VARCHAR(255)
34 );
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Action	Output	Time	Action	Response	Duration / Fetch Time
✓ 292 17:55:48	SELECT cc.id AS card_id, CASE WHEN SUM(t3.declined) = 3 AND COUNT(t3.declined) = 3 THEN 'inactive' ELSE 'active' ...			5000 row(s) returned	0.604 sec / 0.115 sec
✓ 293 18:00:01	SELECT cc.id AS card_id, CASE WHEN SUM(t3.declined) = 3 AND COUNT(t3.declined) = 3 THEN 'inactive' ELSE 'active' ...			5000 row(s) returned	0.594 sec / 0.116 sec
✗ 294 18:00:06	SELECT cc.id AS card_id, CASE WHEN SUM(t.declined) = 3 AND COUNT(t.declined) = 3 THEN 'inactive' ELSE 'active' ...			Error Code: 1054. Unknown column 't.declined' in 'field list'	0.00054 sec
✓ 295 18:00:20	SELECT cc.id AS card_id, CASE WHEN SUM(t3.declined) = 3 AND COUNT(t3.declined) = 3 THEN 'inactive' ELSE 'active' ...			5000 row(s) returned	0.600 sec / 0.115 sec
✓ 296 18:09:15	SELECT * FROM TRANSACTIONS			100000 row(s) returned	0.0012 sec / 0.125 sec
✓ 297 18:21:07	SELECT * FROM data_users			5000 row(s) returned	0.0010 sec / 0.0096...
✓ 298 18:43:54	DROP DATABASE IF EXISTS EuroCommerce			7 row(s) affected	0.600 sec
✓ 299 18:44:05	CREATE DATABASE EuroCommerce			1 row(s) affected	0.0046 sec
✓ 300 18:44:06	USE EuroCommerce			0 row(s) affected	0.00052 sec

CREATING TABLES

```
7 • ⊖ CREATE TABLE products (
8     id VARCHAR(20) PRIMARY KEY,
9     product_name VARCHAR(50) NOT NULL,
10    price VARCHAR(10),
11    colour VARCHAR(20),
12    weight DECIMAL(10,2),
13    warehouse_id VARCHAR(10)
14
15 );
16
17 • ⊖ CREATE TABLE credit_cards (
18     id VARCHAR(50) PRIMARY KEY,
19     user_id VARCHAR(10),
20     iban VARCHAR(50),
21     pan VARCHAR(50),
22     pin VARCHAR(10),
23     track_1 VARCHAR(50),
24     track_2 VARCHAR(50),
25     expiring_date VARCHAR(30)
26 );
27
28 • ⊖ CREATE TABLE companies (
29     id VARCHAR(50) PRIMARY KEY NOT NULL,
30     company_name VARCHAR(40),
31     phone VARCHAR(70)
32 );
```

100% 1:16 | 1 error found

Action Output

	Time	Action	Response	Duration / Fetch Time	
✓	396	12:39:37	DROP DATABASE IF EXISTS euROCOMMERCE	0.346 sec	
✓	397	12:39:42	CREATE DATABASE EuroCommerce	0.0052 sec	
✓	398	12:39:48	USE EuroCommerce	0.0027 sec	
✗	399	12:43:15	CREATE DATABASE EuroCommerce	Error Code: 1007. Can't create database 'EuroCommerce'; database exists	0.0025 sec
✓	400	12:43:28	USE EuroCommerce	0 row(s) affected	0.00032 sec
✓	401	12:43:30	CREATE TABLE products (id VARCHAR(20) PRIMARY KEY, product_name VARCHAR(50)...)	0 row(s) affected	0.028 sec
✓	402	12:43:33	CREATE TABLE credit_cards (id VARCHAR(50) PRIMARY KEY, user_id VARCHAR(10),...	0 row(s) affected	0.019 sec
✓	403	12:43:39	CREATE TABLE companies (id VARCHAR(50) PRIMARY KEY NOT NULL, company_name V...	0 row(s) affected	0.019 sec

CREATING TABLES

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

- Code Editor:** Displays two SQL statements for creating tables.
 - european_users:** Columns include id (VARCHAR(50) PRIMARY KEY NOT NULL), name (VARCHAR(100)), surname (VARCHAR(100)), phone (VARCHAR(100)), email (VARCHAR(100) UNIQUE), birth_date (VARCHAR(100)), country (VARCHAR(40)), city (VARCHAR(40)), postal_code (VARCHAR(20)), and address (VARCHAR(150)).
 - american_users:** Columns include id (VARCHAR(50) PRIMARY KEY NOT NULL), name (VARCHAR(100)), surname (VARCHAR(100)), phone (VARCHAR(100)), email (VARCHAR(100) UNIQUE), birth_date (VARCHAR(100)), country (VARCHAR(40)), city (VARCHAR(40)), postal_code (VARCHAR(20)), and address (VARCHAR(150)).
- Status Bar:** Shows "100%" completion, "19:53" timestamp, and "1 error found".
- Action Output:** A table showing the execution history of the queries.

Action	Time	Response	Duration / Fetch Time
USE EuroCommerce	398 12:39:48	0 row(s) affected	0.0027 sec
CREATE DATABASE EuroCommerce	399 12:43:15	Error Code: 1007. Can't create database 'EuroCommerce'; database exists	0.0025 sec
USE EuroCommerce	400 12:43:28	0 row(s) affected	0.00032 sec
CREATE TABLE products (id VARCHAR(20) PRIMARY KEY, product_name VARCHAR(50)...	401 12:43:30	0 row(s) affected	0.028 sec
CREATE TABLE credit_cards (id VARCHAR(50) PRIMARY KEY, user_id VARCHAR(10)...	402 12:43:33	0 row(s) affected	0.019 sec
CREATE TABLE companies (id VARCHAR(50) PRIMARY KEY NOT NULL, company_name V...	403 12:43:39	0 row(s) affected	0.019 sec
CREATE TABLE european_users (id VARCHAR(50) PRIMARY KEY NOT NULL, name VARC...	404 12:44:40	0 row(s) affected	0.021 sec
CREATE TABLE american_users (id VARCHAR(50) PRIMARY KEY NOT NULL, name VARC...	405 12:44:42	0 row(s) affected	0.022 sec
- Message:** "Query Completed"

CREATING TABLES

```

10
11     colour VARCHAR(20),
12     weight DECIMAL(10,2),
13     warehouse_id VARCHAR(10)
14
15 );
16
17 • ⊕ CREATE TABLE credit_cards (
18     id VARCHAR(5) PRIMARY KEY,
19     user_id VARCHAR(10),
20     iban VARCHAR(255),
21     pan VARCHAR(50),
22     pin VARCHAR(10),
23     track_1 VARCHAR(255),
24     track_2 VARCHAR(255),
25     expiring_date VARCHAR(50)
26 );
27
28 • ⊕ CREATE TABLE companies (
29     id VARCHAR(50) PRIMARY KEY NOT NULL,
30     company_name VARCHAR(255),
31     phone VARCHAR(50),
32     email VARCHAR(255) UNIQUE,
33     country VARCHAR(255),
34     website VARCHAR(255)
35 );
36
37 • ⊕ CREATE TABLE european_users (
38     id VARCHAR(50) PRIMARY KEY NOT NULL,
39     name VARCHAR(255),
40     surname VARCHAR(255),
41     phone VARCHAR(100),
42     email VARCHAR(255) UNIQUE,
43     birth_date VARCHAR(100)
44
100% ◆ 19:17

```

Action Output

	Time	Action	Response	Duration / Fetch Time
✗	294 18:00:06	SELECT cc.id AS card_id, CASE WHEN SUM(t.declined) = 3 AND COUNT(t.declined) = 3 THEN 'inactive' ELSE 'active' END AS status FROM transactions t JOIN credit_cards cc ON cc.id = t.card_id GROUP BY cc.id	Error Code: 1054. Unknown column 't.declined' in 'field list'	0.00054 sec
✓	295 18:00:20	SELECT cc.id AS card_id, CASE WHEN SUM(t3.declined) = 3 AND COUNT(t3.declined) = 3 THEN 'inactive' ELSE 'active' END AS status FROM transactions t3 JOIN credit_cards cc ON cc.id = t3.card_id GROUP BY cc.id	5000 row(s) returned	0.600 sec / 0.115 sec
✓	296 18:09:15	SELECT * FROM TRANSACTIONS	100000 row(s) returned	0.0012 sec / 0.125 sec
✓	297 18:21:07	SELECT * FROM data_users	5000 row(s) returned	0.0010 sec / 0.0096...
✓	298 18:43:54	DROP DATABASE IF EXISTS EuroCommerce	7 row(s) affected	0.600 sec
✓	299 18:44:05	CREATE DATABASE EuroCommerce	1 row(s) affected	0.0046 sec
✓	300 18:44:06	USE EuroCommerce	0 row(s) affected	0.00052 sec
✓	301 18:44:29	CREATE TABLE products (id VARCHAR(50) PRIMARY KEY, product_name VARCHAR(255) NOT NULL, price VARCHAR(50), colu...	0 row(s) affected	0.024 sec
✓	302 18:44:43	CREATE TABLE credit_cards (id VARCHAR(50) PRIMARY KEY, user_id VARCHAR(10), iban VARCHAR(255), pan VARCHAR(50),...	0 row(s) affected	0.015 sec

CREATING TABLES

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

- SQL Editor:** Contains the following SQL code for creating tables:

```

20     pan VARCHAR(50),
21     pin VARCHAR(10),
22     track_1 VARCHAR(255),
23     track_2 VARCHAR(255),
24     expiring_date VARCHAR(50)
25 );
26
27
28 • ⊖ CREATE TABLE companies (
29     id VARCHAR(50) PRIMARY KEY NOT NULL,
30     company_name VARCHAR(255),
31     phone VARCHAR(50),
32     email VARCHAR(255) UNIQUE,
33     country VARCHAR(255),
34     website VARCHAR(255)
35 );
36
37 • ⊖ CREATE TABLE european_users (
38     id VARCHAR(50) PRIMARY KEY NOT NULL,
39     name VARCHAR(255),
40     surname VARCHAR(255),
41     phone VARCHAR(100),
42     email VARCHAR(255) UNIQUE,
43     birth_date VARCHAR(100),
44     country VARCHAR(255),
45     city VARCHAR(255),
46     postal_code VARCHAR(100),
47     address VARCHAR(255)
48 );
49
50 • ⊖ CREATE TABLE american_users (
51     id VARCHAR(50) PRIMARY KEY NOT NULL,
52     name VARCHAR(255),
53     surname VARCHAR(255)

```

- Action Output:** Shows a history of database actions with their times, descriptions, responses, and durations.

	Time	Action	Response	Duration / Fetch Time
✓ 295	18:00:20	SELECT cc.id AS card_id, CASE WHEN SUM(t3.declined) = 3 AND COUNT(t3.declined) = 3 THEN 'inactive' ELSE 'active'... FROM TRANSACTIONS	5000 row(s) returned 100000 row(s) returned	0.600 sec / 0.115 sec
✓ 296	18:09:15	SELECT * FROM TRANSACTIONS	5000 row(s) returned	0.0012 sec / 0.125 sec
✓ 297	18:21:07	SELECT * FROM data_users	5000 row(s) returned	0.0010 sec / 0.0096...
✓ 298	18:43:54	DROP DATABASE IF EXISTS EuroCommerce	7 row(s) affected	0.600 sec
✓ 299	18:44:05	CREATE DATABASE EuroCommerce	1 row(s) affected	0.0046 sec
✓ 300	18:44:06	USE EuroCommerce	0 row(s) affected	0.00052 sec
✓ 301	18:44:29	CREATE TABLE products (id VARCHAR(50) PRIMARY KEY, product_name VARCHAR(255) NOT NULL, price VARCHAR(50), colou... 0 row(s) affected	0 row(s) affected	0.024 sec
✓ 302	18:44:43	CREATE TABLE credit_cards (id VARCHAR(50) PRIMARY KEY, user_id VARCHAR(10), iban VARCHAR(255), pan VARCHAR(50),... 0 row(s) affected	0 row(s) affected	0.015 sec
✓ 303	18:45:02	CREATE TABLE companies (id VARCHAR(50) PRIMARY KEY NOT NULL, company_name VARCHAR(255), phone VARCHAR(50), em... 0 row(s) affected	0 row(s) affected	0.018 sec

CREATING TABLES

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

- SQL Editor:** Contains the following SQL code for creating tables:

```
58
59     address VARCHAR (150)
60 );
61
62
63 • ⊕ CREATE TABLE transactions (
64     id VARCHAR(100) PRIMARY KEY NOT NULL,
65     card_id VARCHAR(50),
66     business_id VARCHAR(50),
67     timestamp DATETIME,
68     amount DECIMAL(10,2),
69     declined BOOLEAN,
70     product_ids VARCHAR(100),
71     user_id VARCHAR(50),
72     lat FLOAT,
73     longitude FLOAT,
74     CONSTRAINT fk_credit_card_id
75         FOREIGN KEY (card_id) REFERENCES credit_cards(id)
76         ON DELETE CASCADE,
77     CONSTRAINT fk_business_id
78         FOREIGN KEY (business_id) REFERENCES companies(id)
79         ON DELETE CASCADE,
80     CONSTRAINT fk_european_users
81         FOREIGN KEY (user_id) REFERENCES european_users(id)
82         ON DELETE CASCADE
83 );
```

- Status Bar:** Shows "100%" completion, "20:67" time, and "1 error found".
- Action Output:** A table showing the history of actions:

Action	Time	Response	Duration / Fetch Time
CREATE DATABASE EuroCommerce	399 12:43:15	Error Code: 1007. Can't create database 'EuroCommerce'; database exists	0.0025 sec
USE EuroCommerce	400 12:43:28	0 row(s) affected	0.00032 sec
CREATE TABLE products (id VARCHAR(20) PRIMARY KEY, product_name VARCHAR(50)...)	401 12:43:30	0 row(s) affected	0.028 sec
CREATE TABLE credit_cards (id VARCHAR(50) PRIMARY KEY, user_id VARCHAR(10),...)	402 12:43:33	0 row(s) affected	0.019 sec
CREATE TABLE companies (id VARCHAR(50) PRIMARY KEY NOT NULL, company_name V...	403 12:43:39	0 row(s) affected	0.019 sec
CREATE TABLE european_users (id VARCHAR(50) PRIMARY KEY NOT NULL, name VARC...	404 12:44:40	0 row(s) affected	0.021 sec
CREATE TABLE american_users (id VARCHAR(50) PRIMARY KEY NOT NULL, name VARC...	405 12:44:42	0 row(s) affected	0.022 sec
CREATE TABLE transactions (id VARCHAR(100) PRIMARY KEY NOT NULL, card_id VAR...	406 12:45:34	0 row(s) affected	0.029 sec

LOADING DATA

```
95 -- TABLE PRODUCTS
96
97 • LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/products.csv'
98 INTO TABLE products
99 FIELDS TERMINATED BY ','
100 ENCLOSED BY '\"'
101 IGNORE 1 ROWS;
102
103 -- TABLE CREDIT_CARDS
104
105 • LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/credit_cards.csv'
106 INTO TABLE credit_cards
107 FIELDS TERMINATED BY ','
108 ENCLOSED BY '\"'
109 IGNORE 1 ROWS;
110
111 -- TABLE COMPANIES
112
113 • LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/companies.csv'
114 INTO TABLE companies
115 FIELDS TERMINATED BY ','
116 ENCLOSED BY '\"'
117 IGNORE 1 ROWS;
118
```

100% ◆ | 14:114 1 error found

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 402	12:43:33	CREATE TABLE credit_cards (id VARCHAR(50) PRIMARY KEY, user_id VARCHAR(10),...	0 row(s) affected	0.019 sec
✓ 403	12:43:39	CREATE TABLE companies (id VARCHAR(50) PRIMARY KEY NOT NULL, company_name V...	0 row(s) affected	0.019 sec
✓ 404	12:44:40	CREATE TABLE european_users (id VARCHAR(50) PRIMARY KEY NOT NULL, name VARC...	0 row(s) affected	0.021 sec
✓ 405	12:44:42	CREATE TABLE american_users (id VARCHAR(50) PRIMARY KEY NOT NULL, name VARC...	0 row(s) affected	0.022 sec
✓ 406	12:45:34	CREATE TABLE transactions (id VARCHAR(100) PRIMARY KEY NOT NULL, card_id VAR...	0 row(s) affected	0.029 sec
✓ 407	12:45:55	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/products.cs...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.012 sec
⚠ 408	12:45:57	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/credit_card...	5000 row(s) affected, 1024 warning(s): 1265 Data truncated for column 'expiring_date' at row 1 1262 Row 1 was truncat...	0.168 sec
✓ 409	12:45:58	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/companies...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.012 sec

LOADING DATA

```
117    IGNORE 1 ROWS;
118
119
120    -- TABLE EUROPEAN_USERS
121
122 • LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/european_users.csv'
123     INTO TABLE european_users
124     FIELDS TERMINATED BY ','
125     ENCLOSED BY """
126     IGNORE 1 ROWS;
127
128    -- ADD COLUMN WITH CONTINENT NAME
129
130 • ALTER TABLE european_users
131     ADD COLUMN continent VARCHAR(55) NOT NULL DEFAULT 'Europe';
132
133
134    -- TABLE AMERICAN_USERS
135
136 • LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/american_users.csv'
137     INTO TABLE american_users
138     FIELDS TERMINATED BY ','
139     ENCLOSED BY """
140     IGNORE 1 ROWS;
```

100% 19:123 1 error found

Action Output

	Time	Action	Response	Duration / Fetch Time
✓	404 12:44:40	CREATE TABLE european_users (id VARCHAR(50) PRIMARY KEY NOT NULL, name VARC...	0 row(s) affected	0.021 sec
✓	405 12:44:42	CREATE TABLE american_users (id VARCHAR(50) PRIMARY KEY NOT NULL, name VARC...	0 row(s) affected	0.022 sec
✓	406 12:45:34	CREATE TABLE transactions (id VARCHAR(100) PRIMARY KEY NOT NULL, card_id VAR...	0 row(s) affected	0.029 sec
✓	407 12:45:55	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/products.cs...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.012 sec
⚠	408 12:45:57	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/credit_card...	5000 row(s) affected, 1024 warning(s): 1265 Data truncated for column 'expiring_date' at row 1 1262 Row 1 was truncat...	0.168 sec
✓	409 12:45:58	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/companies...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.012 sec
✗	410 12:46:42	CREATE DATABASE EuroCommerce	Error Code: 1007. Can't create database 'EuroCommerce'; database exists	0.00060 sec
✓	411 12:46:46	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/european_u...	3990 row(s) affected Records: 3990 Deleted: 0 Skipped: 0 Warnings: 0	0.134 sec

LOADING DATA

```

100
101 IGNORE 1 ROWS;
102
103 -- TABLE CREDIT_CARDS
104
105 • LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/credit_cards.csv'
106 INTO TABLE credit_cards
107 FIELDS TERMINATED BY ','
108 ENCLOSED BY ""
109 IGNORE 1 ROWS;
110
111 -- TABLE COMPANIES
112
113 • LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/companies.csv'
114 INTO TABLE companies
115 FIELDS TERMINATED BY ','
116 ENCLOSED BY ""
117 IGNORE 1 ROWS;
118
119
120 -- TABLE EUROPEAN_USERS
121
122 • LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/european_users.csv'
123 INTO TABLE european_users
124 FIELDS TERMINATED BY ','
125 ENCLOSED BY ""
126 IGNORE 1 ROWS;
127
128 -- ADD COLUMN WITH CONTINENT NAME
129
130 • ALTER TABLE european_users
131 ADD COLUMN continent VARCHAR(55) NOT NULL DEFAULT 'Europe';
132
133
100% 27:113

```

Action Output

	Time	Action	Response	Duration / Fetch Time
✓	302 18:44:43	CREATE TABLE credit_cards (id VARCHAR(50) PRIMARY KEY, user_id VARCHAR(10), iban VARCHAR(255), pan VARCHAR(50),... 0 row(s) affected		0.015 sec
✓	303 18:45:02	CREATE TABLE companies (id VARCHAR(50) PRIMARY KEY NOT NULL, company_name VARCHAR(255), phone VARCHAR(50), em... 0 row(s) affected		0.018 sec
✓	304 18:45:14	CREATE TABLE european_users (id VARCHAR(50) PRIMARY KEY NOT NULL, name VARCHAR(255), surname VARCHAR(255), phone V... 0 row(s) affected		0.017 sec
✓	305 18:45:29	CREATE TABLE american_users (id VARCHAR(50) PRIMARY KEY NOT NULL, name VARCHAR(255), surname VARCHAR(255), phone V... 0 row(s) affected		0.026 sec
✓	306 18:45:42	CREATE TABLE transactions (id VARCHAR(255) PRIMARY KEY NOT NULL, card_id VARCHAR(50), business_id VARCHAR(50), ti... 0 row(s) affected		0.032 sec
✓	307 18:46:19	SHOW COLUMNS FROM european_users 10 row(s) returned		0.0028 sec / 0.00002...
✓	308 18:47:04	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/products.csv' INTO TABLE products FIELDS TERMINATED... 100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0		0.015 sec
⚠	309 18:47:24	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/credit_cards.csv' INTO TABLE credit_cards FIELDS TERM... 5000 row(s) affected, 1024 warning(s): 1262 Row 1 was truncated; it contained more data than there were input columns... 0.153 sec		
✓	310 18:47:39	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/companies.csv' INTO TABLE companies FIELDS TERMINA... 100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0		0.011 sec

ADDING NEW COLUMN NAMED CONTINENT TO AMERICAN_USERS AND EUROPEAN_USERS

```

125
126
127
128 -- ADD COLUMN WITH CONTINENT NAME
129
130 • ALTER TABLE european_users
131   ADD COLUMN continent VARCHAR(55) NOT NULL DEFAULT 'Europe';
132
133
134 -- TABLE AMERICAN_USERS
135
136 • LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/american_users.csv'
137   INTO TABLE american_users
138   FIELDS TERMINATED BY ','
139   ENCLOSED BY ""
140   IGNORE 1 ROWS;
141
142 -- ADD COLUMN WITH THE CONTINENT NAME
143
144 • ALTER TABLE american_users
145   ADD COLUMN continent VARCHAR(55) NOT NULL DEFAULT 'America';
146
147 -- BRIDGE TABLE CREATION
148
149 • CREATE TABLE transaction_products (
150   transaction_id VARCHAR(100),
151   product_id VARCHAR(10)
100% 20:145 1 error found

```

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 407	12:45:55	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/products.cs...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.012 sec
⚠ 408	12:45:57	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/credit_card...	5000 row(s) affected, 1024 warning(s): 1265 Data truncated for column 'expiring_date' at row 1 1262 Row 1 was truncat...	0.168 sec
✓ 409	12:45:58	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/companies....	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.012 sec
✗ 410	12:46:42	CREATE DATABASE EuroCommerce	Error Code: 1007. Can't create database 'EuroCommerce'; database exists	0.00060 sec
✓ 411	12:46:46	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/european_u...	3990 row(s) affected Records: 3990 Deleted: 0 Skipped: 0 Warnings: 0	0.134 sec
✓ 412	12:48:10	ALTER TABLE european_users ADD COLUMN continent VARCHAR(55) NOT NULL DEFAULT '...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.046 sec
✓ 413	12:48:12	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/american_u...	1010 row(s) affected Records: 1010 Deleted: 0 Skipped: 0 Warnings: 0	0.040 sec
✓ 414	12:48:16	ALTER TABLE american_users ADD COLUMN continent VARCHAR(55) NOT NULL DEFAULT '...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.019 sec

CREATING BRIDGE TABLE transactions_products

```
-- ADD COLUMN WITH THE CONTINENT NAME

ALTER TABLE american_users
ADD COLUMN continent VARCHAR(55) NOT NULL DEFAULT 'America';

146
147 -- BRIDGE TABLE CREATION
148
149 • CREATE TABLE transaction_products (
150     transaction_id VARCHAR(100),
151     product_id VARCHAR(10),
152     FOREIGN KEY (transaction_id) REFERENCES transactions(id) ON DELETE CASCADE
153 );
154
155 -- SWITCH TEMPORARY OFF FOREIGN KEY TO AVOID ERROR 1452: PARENT TABLES ARE MISSING FOR THE FOREIGN KEYS IN TRANSACTIONS
156
157 ✘ SET FOREIGN_KEY_CHECKS = 0; ***
158
159 -- LOAD DATA TRANSACTIONS TABLE
160 LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/transactions.csv'
161 INTO TABLE transactions
162 FIELDS TERMINATED BY ','
163 ENCLOSED BY "'"
164 LINES TERMINATED BY '\n'
165 IGNORE 1 ROWS
166 (id_card_id_business_id_timestamp_amount_declined_product_ids_user_id_lat_longitude)
100% 27:151 1 error found
```

Action Output

	Time	Action	Response	Duration / Fetch Time
⚠ 408	12:45:57	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/credit_card...	5000 row(s) affected, 1024 warning(s): 1265 Data truncated for column 'expiring_date' at row 1 1262 Row 1 was truncat...	0.168 sec
✓ 409	12:45:58	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/companies....	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.012 sec
✗ 410	12:46:42	CREATE DATABASE EuroCommerce	Error Code: 1007. Can't create database 'EuroCommerce'; database exists	0.00060 sec
✓ 411	12:46:46	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/european_u...	3990 row(s) affected Records: 3990 Deleted: 0 Skipped: 0 Warnings: 0	0.134 sec
✓ 412	12:48:10	ALTER TABLE european_users ADD COLUMN continent VARCHAR(55) NOT NULL DEFAULT '...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.046 sec
✓ 413	12:48:12	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/american_u...	1010 row(s) affected Records: 1010 Deleted: 0 Skipped: 0 Warnings: 0	0.040 sec
✓ 414	12:48:16	ALTER TABLE american_users ADD COLUMN continent VARCHAR(55) NOT NULL DEFAULT '...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.019 sec
✓ 415	12:49:02	CREATE TABLE transaction_products (transaction_id VARCHAR(100), product_id VARC...	0 row(s) affected	0.021 sec

Query Completed

UNION ALL EUROPEAN_USERS AND AMERICAN_USERS

```
162
163
164 ENCLOSED BY ""
165 LINES TERMINATED BY '\n'
166 IGNORE 1 ROWS
167 (id, card_id, business_id, timestamp, amount, declined, product_ids, user_id, lat, longitude);
168
169 -- ON FOREIGN KEYS
170
171 • SET FOREIGN_KEY_CHECKS = 1;
172
173 -- UNION ALL EUROPEAN_USERS AND AMERICAN_USERS TABLES
174
175 • CREATE TABLE data_users AS
176   SELECT * FROM european_users
177   UNION ALL
178   SELECT * FROM american_users;
179
180 -- ADD A PRIMARY KEY TO THE NEW TABLE DATA_USERS
181
182 • ALTER TABLE data_users
183   ADD PRIMARY KEY (id);
184
185 -- ADD A FOREIGN KEY IN THE TABLE TRANSACTIONS
186
187 • ALTER TABLE transactions
188   ADD CONSTRAINT fk_user_id
100% 20:176
```

Action Output

Action	Time	Response	Duration / Fetch Time
433 12:52:48 ALTER TABLE european_users ADD COLUMN continent VARCHAR(55) NOT NULL DEFAULT ''		0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.023 sec
434 12:52:49 LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/american_u...		1010 row(s) affected Records: 1010 Deleted: 0 Skipped: 0 Warnings: 0	0.031 sec
435 12:52:52 ALTER TABLE american_users ADD COLUMN continent VARCHAR(55) NOT NULL DEFAULT ''		0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
436 12:52:54 CREATE TABLE transaction_products (transaction_id VARCHAR(100), product_id VARC...		0 row(s) affected	0.016 sec
437 12:52:57 SET FOREIGN_KEY_CHECKS = 0		0 row(s) affected	0.00047 sec
438 12:52:59 LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/transaction...		100000 row(s) affected Records: 100000 Deleted: 0 Skipped: 0 Warnings: 0	3.304 sec
439 12:53:09 SET FOREIGN_KEY_CHECKS = 1		0 row(s) affected	0.00046 sec
440 12:53:12 CREATE TABLE data_users AS SELECT * FROM european_users UNION ALL SELECT * FROM...		5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0	0.099 sec

ADDING FOREIGN KEYS

```

171
172    -- UNION ALL EUROPEAN_USERS AND AMERICAN_USERS TABLES
173
174 • CREATE TABLE data_users AS
175     SELECT * FROM european_users
176     UNION ALL
177     SELECT * FROM american_users;
178
179    -- ADD A PRIMARY KEY TO THE NEW TABLE DATA_USERS
180
181 • ALTER TABLE data_users
182     ADD PRIMARY KEY (id);
183
184    -- ADD A FOREIGN KEY IN THE TABLE TRANSACTIONS
185
186 • ALTER TABLE transactions
187     ADD CONSTRAINT fk_user_id
188     FOREIGN KEY (user_id) REFERENCES data_users(id);
189
190    -- CHECKING NEW TABLE
191
192 • SELECT * FROM data_users;
193
194    -- DELETE OLD TABLES BUT TO AVOID FK ERRORS, SWITCH TO OFF

```

100% ◇ 16:192

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

id	name	surname	phone	email	birth_date	country	city	postal_code	address	continent
100	Melodie	McLean	+1-677-221-7152	risus.varius@google.ca	Sep 15, 1989	United States	San Jose	95101	Ap #644-8492 Sagittis St.	America
1000	Amkjrvg	Qbulrxbp	+48-258-9936	amkjrvg.qbulrxbp@example.com	May 17, 1970	Germany	Stuttgart	70173	215 Qbulrxbp St	Europe
1001	Nfrvrib	Oydaiwbg	+94-121-2522	nfrvrib.oydaiwbg@example.com	Mar 4, 1994	Germany	Cologne	50667	121 Oydaiwbg St	Europe
1002	Jbmd	Jbddzhvp	+70-120-3668	jbmd.jbddzhvp@example.com	Sep 27, 2001	Germany	Munich	80331	412 Jbddzhvp St	Europe
1003	Uycig	Sfbдумzj	+58-123-6968	uycig.sfbдумzj@example.com	Jan 20, 1981	Germany	Stuttgart	70173	735 Sfbдумzj St	Europe

data_users 4 Apply Revert

Action Output ◇

Time	Action	Response	Duration / Fetch Time
315 18:49:56	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/american_users.csv' INTO TABLE american_users FIELDS...	0 row(s) affected, 1024 warning(s): 1261 Row 1 doesn't contain data for all columns 1062 Duplicate entry 'ame...' 0.019 sec	0.019 sec
316 18:50:12	CREATE TABLE transaction_products (transaction_id VARCHAR(255), product_id VARCHAR(255), FOREIGN KEY (transaction_id) R...	0 row(s) affected	0.030 sec
317 18:52:11	SET FOREIGN_KEY_CHECKS = 0	0 row(s) affected	0.0020 sec
318 18:52:13	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/transactions.csv' INTO TABLE transactions FIELDS TERMI...	100000 row(s) affected Records: 100000 Deleted: 0 Skipped: 0 Warnings: 0 5.038 sec	5.038 sec
319 18:52:43	SET FOREIGN_KEY_CHECKS = 1	0 row(s) affected	0.00036 sec
320 18:52:46	CREATE TABLE data_users AS SELECT * FROM european_users UNION ALL SELECT * FROM american_users	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0 0.124 sec	0.124 sec
321 18:53:14	ALTER TABLE data_users ADD PRIMARY KEY (id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.120 sec	0.120 sec
322 18:53:17	ALTER TABLE transactions ADD CONSTRAINT fk_user_id FOREIGN KEY (user_id) REFERENCES data_users(id)	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0 2.792 sec	2.792 sec
323 18:53:22	SELECT * FROM data_users	5000 row(s) returned 0.0021 sec / 0.0099...	0.0021 sec / 0.0099...

DELETE AMERICAN_USERS AND EUROPEAN_USERS TABLE

```

187 ADD CONSTRAINT fk_user_id
188 FOREIGN KEY (user_id) REFERENCES data_users(id);
189
190 -- CHECKING NEW TABLE
191
192 • SELECT * FROM data_users;
193
194 -- DELETE OLD TABLES BUT TO AVOID FK ERRORS, SWITCH TO OFF
195
196 • SET FOREIGN_KEY_CHECKS = 0;
197
198 -- DELETE TABLES
199
200 • DROP TABLE american_users;
201 • DROP TABLE european_users;
202
203 -- SWITCH TO ON FK
204
205 • SET FOREIGN_KEY_CHECKS = 1;
206
207 -- CHECK NEW TABLE
208
209 • SELECT * FROM data_users;
210
211 -- ADD THE FOREIGN KEYS TO THE BRIDGE TABLE
212 -- ON A BRIDGE TABLE WE ALWAYS HAVE JUST FOREIGN KEYS BECAUSE IS A MANY TO MANY RELATION
213
214 • ALTER TABLE transaction_products
215 ADD CONSTRAINT fk_tp_transactions
216 FOREIGN KEY (transaction_id) REFERENCES transactions(id)
217 ON DELETE CASCADE;
218
219 • ALTER TABLE transaction_products
100% 18:20:1

```

Action Output

	Time	Action	Response	Duration / Fetch Time
✓	318 18:52:13	LOAD DATA LOCAL INFILE '/Users/josemessiasferreira/Documents/sprint_4_data/transactions.csv' INTO TABLE transactions FIELDS TERMINATED BY ',' ENCLOSED BY '\"' LINES TERMINATED BY '\n';	100000 row(s) affected Records: 100000 Deleted: 0 Skipped: 0 Warnings: 0 0 row(s) affected	5.038 sec
✓	319 18:52:43	SET FOREIGN_KEY_CHECKS = 1		0.00036 sec
✓	320 18:52:46	CREATE TABLE data_users AS SELECT * FROM european_users UNION ALL SELECT * FROM american_users	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0	0.124 sec
✓	321 18:53:14	ALTER TABLE data_users ADD PRIMARY KEY (id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.120 sec
✓	322 18:53:17	ALTER TABLE transactions ADD CONSTRAINT fk_user_id FOREIGN KEY (user_id) REFERENCES data_users(id)	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	2.792 sec
✓	323 18:53:22	SELECT * FROM data_users	5000 row(s) returned	0.0021 sec / 0.0099...
✓	324 18:53:55	SET FOREIGN_KEY_CHECKS = 0	0 row(s) affected	0.00037 sec
✓	325 18:53:57	DROP TABLE american_users	0 row(s) affected	0.014 sec
✓	326 18:54:00	DROP TABLE european_users	0 row(s) affected	0.016 sec

ADD FOREIGN KEYS TO THE BRIDGE TABLE

```
200 -- CHECK NEW TABLE
209
210 • SELECT * FROM data_users;
211
212 -- ADD THE FOREIGN KEYS TO THE BRIDGE TABLE
213 -- ON A BRIDGE TABLE WE ALWAYS HAVE JUST FOREIGN KEYS BECAUSE IS A MANY TO MANY RELATION
214
215 • ALTER TABLE transaction_products
216   ADD CONSTRAINT fk_tp_transactions
217     FOREIGN KEY (transaction_id) REFERENCES transactions(id)
218   ON DELETE CASCADE;
219
220 • ALTER TABLE transaction_products
221   ADD CONSTRAINT fk_tp_products
222     FOREIGN KEY (product_id) REFERENCES products(id)
223   ON DELETE CASCADE;
224
225 -- CHANGE THE DATATYPE OF THE FIELD PRODUCT_IDS TO JSON USING THE CONCAT FUNCTION
226
227 • UPDATE transactions
228   SET product_ids = CONCAT('[', REPLACE(product_ids, ' ', ''), ']');
229
230 /* 
231   USING JSON (JAVA SCRIPT OBJECT NOTATION) TO EXTRACT THE VALUES IN THE COLUMNS PRODUCTS_IDS
232   AND PAST THEM TO THE BRIDGE TABLE transaction_products
100% 15:222
```

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 443	12:53:41	SELECT * FROM data_users	5000 row(s) returned	0.0023 sec / 0.010 sec
✓ 444	12:54:08	SET FOREIGN_KEY_CHECKS = 0	0 row(s) affected	0.00030 sec
✓ 445	12:54:11	DROP TABLE american_users	0 row(s) affected	0.012 sec
✓ 446	12:54:13	DROP TABLE european_users	0 row(s) affected	0.014 sec
✓ 447	12:54:16	SET FOREIGN_KEY_CHECKS = 1	0 row(s) affected	0.00040 sec
✓ 448	12:54:18	SELECT * FROM data_users	5000 row(s) returned	0.00058 sec / 0.008...
✓ 449	12:54:22	ALTER TABLE transaction_products ADD CONSTRAINT fk_tp_transactions FOREIGN KEY (tra...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.039 sec
✓ 450	12:54:25	ALTER TABLE transaction_products ADD CONSTRAINT fk_tp_products FOREIGN KEY (produ...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.035 sec

Query Completed

```

212
213
214 • ALTER TABLE transaction_products
215   ADD CONSTRAINT fk_tp_transactions
216   FOREIGN KEY (transaction_id) REFERENCES transactions(id)
217   ON DELETE CASCADE;
218
219 • ALTER TABLE transaction_products
220   ADD CONSTRAINT fk_tp_products
221   FOREIGN KEY (product_id) REFERENCES products(id)
222   ON DELETE CASCADE;
223
224 -- CHANGE THE DATATYPE OF THE FIELD PRODUCT_IDS TO JSON USING THE CONCAT FUNCTION
225
226 • UPDATE transactions
227   SET product_ids = CONCAT('[', REPLACE(product_ids, ' ', ''), ']');
228
229 /* 
230  * USING JSON (JAVA SCRIPT OBJECT NOTATION) TO EXTRACT THE VALUES IN THE COLUMNS PRODUCTS_IDS
231  * AND PAST THEM TO THE BRIDGE TABLE transaction_products.
232 */
233
234 • INSERT INTO transaction_products (transaction_id, product_id)
235   SELECT
236     t.id AS transaction_id,
237     j.product_id
238   FROM transactions t
239   JOIN JSON_TABLE(
240     t.product_ids,
241     '$[*]' COLUMNS (
242       product_id INT PATH '$'
243     )
244   ) AS j;
245
100%  | 11:26 |

```

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 323	18:53:22	SELECT * FROM data_users	5000 row(s) returned	0.0021 sec / 0.0099...
✓ 324	18:53:55	SET FOREIGN_KEY_CHECKS = 0	0 row(s) affected	0.00037 sec
✓ 325	18:53:57	DROP TABLE american_users	0 row(s) affected	0.014 sec
✓ 326	18:54:00	DROP TABLE european_users	0 row(s) affected	0.016 sec
✓ 327	18:54:59	SET FOREIGN_KEY_CHECKS = 1	0 row(s) affected	0.00036 sec
✓ 328	18:55:02	SELECT * FROM data_users	5000 row(s) returned	0.00079 sec / 0.011 s...
✓ 329	18:55:06	ALTER TABLE transaction_products ADD CONSTRAINT fk_tp_transactions FOREIGN KEY (transaction_id) REFERENCES transactions(id) O...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.043 sec
✓ 330	18:55:09	ALTER TABLE transaction_products ADD CONSTRAINT fk_tp_products FOREIGN KEY (product_id) REFERENCES products(id) ON DELETE...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.042 sec
✓ 331	18:55:54	UPDATE transactions SET product_ids = CONCAT('[', REPLACE(product_ids, ' ', ''), ']')	100000 row(s) affected Rows matched: 100000 Changed: 100000 Warnings: 0	3.468 sec

USING JSON TO EXTRACT DATA FROM PRODUCT_IDS

```

225 ━ 226 • UPDATE transactions
227   SET product_ids = CONCAT('[', REPLACE(product_ids, ',', ''), ']');
228
229 /* 
230  USING JSON (JAVA SCRIPT OBJECT NOTATION) TO EXTRACT THE VALUES IN THE COLUMNS PRODUCTS_IDS
231  AND PAST THEM TO THE BRIDGE TABLE transaction_products.
232 */
233
234 • INSERT INTO transaction_products (transaction_id, product_id)
235   SELECT
236     t.id AS transaction_id,
237     j.product_id
238   FROM transactions t
239   JOIN JSON_TABLE(
240     t.product_ids,
241     '$[*]' COLUMNS (
242       product_id INT PATH '$'
243     )
244   ) AS j;
245
246 -- CHECK TRANSACTION_PRODUCTS TABLE
247
248 • SELECT * FROM transaction_products;
249
250 -- TABLE PRODUCTS TRANSFORMATIONS
251
252 • DESCRIBE products;
253 • SELECT *
254   FROM products;
255
256 -- FIELD PRICE TO DECIMAL (10,2)
257
258 • UPDATE products
100% 14:237

```

Action Output

	Time	Action	Response	Duration / Fetch Time
✓	324 18:53:55	SET FOREIGN_KEY_CHECKS = 0	0 row(s) affected	0.00037 sec
✓	325 18:53:57	DROP TABLE american_users	0 row(s) affected	0.014 sec
✓	326 18:54:00	DROP TABLE european_users	0 row(s) affected	0.016 sec
✓	327 18:54:59	SET FOREIGN_KEY_CHECKS = 1	0 row(s) affected	0.00036 sec
✓	328 18:55:02	SELECT * FROM data_users	5000 row(s) returned	0.00079 sec / 0.011 s...
✓	329 18:55:06	ALTER TABLE transaction_products ADD CONSTRAINT fk_tp_transactions FOREIGN KEY (transaction_id) REFERENCES transactions(id) O...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.043 sec
✓	330 18:55:09	ALTER TABLE transaction_products ADD CONSTRAINT fk_tp_products FOREIGN KEY (product_id) REFERENCES products(id) ON DELETE...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.042 sec
✓	331 18:55:54	UPDATE transactions SET product_ids = CONCAT('[', REPLACE(product_ids, ',', ''), ']')	100000 row(s) affected Rows matched: 100000 Changed: 100000 Warnings: 0	3.468 sec
✓	332 18:56:23	INSERT INTO transaction_products (transaction_id, product_id) SELECT t.id AS transaction_id, j.product_id FROM transactions t JOIN JSON_TABLE(t.product_ids, '\$[*]' COLUMNS (product_id INT PATH '\$')) AS j	253391 row(s) affected Records: 253391 Duplicates: 0 Warnings: 0	5.989 sec

CHECKING TRANSACTION_PRODUCTS TABLE

```

237     j.product_id
238   FROM transactions t
239   JOIN JSON_TABLE(
240     t.product_ids,
241     '$[*]' COLUMNS (
242       product_id INT PATH '$'
243     )
244   ) AS j;
245
246 -- CHECK TRANSACTION_PRODUCTS TABLE
247
248 • SELECT * FROM transaction_products;
249
250 -- TABLE PRODUCTS TRANSFORMATIONS
251
252 • DESCRIBE products;
253 • SELECT *
254   FROM products;
255
256 -- FIELD PRICE TO DECIMAL (10,2)
257
258 • UPDATE products
259   SET price = REPLACE(price, '$', '');
260
261 ALTER TABLE products
100% 26:248

```

Result Grid Filter Rows: Search: Export: Fetch rows:

transaction_id	product_id
00043A49-2949-494B-A5DD-A5BAE3BB19DD	16
00043A49-2949-494B-A5DD-A5BAE3BB19DD	26
00043A49-2949-494B-A5DD-A5BAE3BB19DD	97
00043A49-2949-494B-A5DD-A5BAE3BB19DD	87
000447FE-B650-4DCF-85DE-C7ED0EE1CAAD	66

transaction_products 6 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 325	18:53:57	DROP TABLE american_users	0 row(s) affected	0.014 sec
✓ 326	18:54:00	DROP TABLE european_users	0 row(s) affected	0.016 sec
✓ 327	18:54:59	SET FOREIGN_KEY_CHECKS = 1	0 row(s) affected	0.00036 sec
✓ 328	18:55:02	SELECT * FROM data_users	5000 row(s) returned	0.00079 sec / 0.011 s...
✓ 329	18:55:06	ALTER TABLE transaction_products ADD CONSTRAINT fk_tp_transactions FOREIGN KEY (transaction_id) REFERENCES transactions(id) O...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.043 sec
✓ 330	18:55:09	ALTER TABLE transaction_products ADD CONSTRAINT fk_tp_products FOREIGN KEY (product_id) REFERENCES products(id) ON DELETE...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.042 sec
✓ 331	18:55:54	UPDATE transactions SET product_ids = CONCAT('[', REPLACE(product_ids, ',', ','), ']')	100000 row(s) affected Rows matched: 100000 Changed: 100000 Warnings: 0	3.468 sec
✓ 332	18:56:23	INSERT INTO transaction_products (transaction_id, product_id) SELECT t.id AS transaction_id, j.product_id FROM transactions t JOIN JSON_TABLE(t.product_ids, '\$[*]' COLUMNS (product_id INT PATH '\$')) j ON t.id = j.transaction_id	253391 row(s) affected Records: 253391 Duplicates: 0 Warnings: 0	5.989 sec
✓ 333	18:57:32	SELECT * FROM transaction_products	253391 row(s) returned	0.0010 sec / 0.261 sec

CHANGE FIELD PRICE TO DECIMAL

```

243     )
244 ) AS j;
245
246 -- CHECK TRANSACTION_PRODUCTS TABLE
247
248 • SELECT * FROM transaction_products;
249
250 -- TABLE PRODUCTS TRANSFORMATIONS
251
252 • DESCRIBE products;
253 • SELECT *
254   FROM products;
255
256 -- FIELD PRICE TO DECIMAL (10,2)
257
258 • UPDATE products
259   SET price = REPLACE(price, '$', '');
260
261 • ALTER TABLE products
262   MODIFY price DECIMAL(10,2);
263
264 -- FIRST LETTER UPPER OF PRODUCT NAME
265
266 • UPDATE products
267   SET product_name = CONCAT(
268     UPPER(LEFT(product_name, 1)),
269     LOWER(SUBSTRING(product_name, 2)))
270 ;
271
272 -- TABLE DATA_USERS TRANSFORMATIONS
273
274 • SELECT * FROM data_users;
275 • DESCRIBE data_users
100% 6:288

```

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 327	18:54:59	SET FOREIGN_KEY_CHECKS = 1	0 row(s) affected	0.00036 sec
✓ 328	18:55:02	SELECT * FROM data_users	5000 row(s) returned	0.00079 sec / 0.011 s...
✓ 329	18:55:06	ALTER TABLE transaction_products ADD CONSTRAINT fk_tp_transactions FOREIGN KEY (transaction_id) REFERENCES transactions(id) O...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.043 sec
✓ 330	18:55:09	ALTER TABLE transaction_products ADD CONSTRAINT fk_tp_products FOREIGN KEY (product_id) REFERENCES products(id) ON DELETE...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.042 sec
✓ 331	18:55:54	UPDATE transactions SET product_ids = CONCAT('[', REPLACE(product_ids, ',', ','), ']')	100000 row(s) affected Rows matched: 100000 Changed: 100000 Warnings: 0	3.468 sec
✓ 332	18:56:23	INSERT INTO transaction_products (transaction_id, product_id) SELECT t.id AS transaction_id, j.product_id FROM transactions t JOIN...	253391 row(s) affected Records: 253391 Duplicates: 0 Warnings: 0	5.989 sec
✓ 333	18:57:32	SELECT * FROM transaction_products	253391 row(s) returned	0.0010 sec / 0.261 sec
✓ 334	18:58:12	DESCRIBE products	6 row(s) returned	0.0059 sec / 0.00001...
✓ 335	18:58:17	UPDATE products SET price = REPLACE(price, '\$', '')	100 row(s) affected Rows matched: 100 Changed: 100 Warnings: 0	0.0071 sec

FIRST LETTER CAPITALIZED ON PRODUCTS NAME

```

245
246 -- CHECK TRANSACTION_PRODUCTS TABLE
247
248 • SELECT * FROM transaction_products;
249
250 -- TABLE PRODUCTS TRANSFORMATIONS
251
252 • DESCRIBE products;
253 • SELECT *
254   FROM products;
255
256 -- FIELD PRICE TO DECIMAL (10,2)
257
258 • UPDATE products
259   SET price = REPLACE(price, '$', '');
260
261 • ALTER TABLE products
262   MODIFY price DECIMAL(10,2);
263
264 -- FIRST LETTER UPPER OF PRODUCT NAME
265
266 • UPDATE products
267   SET product_name = CONCAT(
268     UPPER(LEFT(product_name, 1)),
269     LOWER(SUBSTRING(product_name, 2)))
270   ;
271
272 • SELECT * FROM products;
273
274 -- TABLE DATA_USERS TRANSFORMATIONS
275
276 • SELECT * FROM data_users;
277 • DESCRIBE data_users;
100% 1:273

```

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

Result Grid Form Editor

id	product_name	price	colour	weight	warehouse_id
100	South duel	40.43	#666666	3.00	WH-95
11	Karstark dome	49.70	#141414	2.70	WH-6
12	Duel direwolf	181.60	#a8a8a8	2.10	WH-7
13	Palpatine chewbacca	139.59	#2b2b2b	1.00	WH-8
14	Direwolf	147.53	#c4c4c4	2.00	WH-9
15	Stannis warden	194.29	#d9d9d9	1.50	WH-10
16	The duel warden	180.91	#666666	3.00	WH-11
17	Skywalker ewok sith	91.89	#7c7c7c	3.20	WH-12

products 12 Apply Revert

CHANGE DATATYPE OF BIRTH_DATE TO DATE

```
260
261 • ALTER TABLE products
262   MODIFY price DECIMAL(10,2);
263
264   -- FIRST LETTER UPPER OF PRODUCT NAME
265
266 • UPDATE products
267   SET product_name = CONCAT(
268     UPPER(LEFT(product_name, 1)),
269     LOWER(SUBSTRING(product_name, 2)))
270 );
271
272 • SELECT * FROM products;
273
274   -- TABLE DATA_USERS TRANSFORMATIONS
275
276 • SELECT * FROM data_users;
277 • DESCRIBE data_users;
278
279   -- CHANGE DATATYPE FROM BIRTH_DATE FIELD TO DATE
280
281 • UPDATE data_users
282   SET birth_date = STR_TO_DATE(birth_date, '%b %d, %Y');
283
284 • ALTER TABLE data_users
285   MODIFY birth_date DATE;
286
287 • SELECT * FROM data_users;
288
289
```

100% ◇ 16:282 |

Action Output ◇

	Time	Action	Response	Duration / Fetch Time
✓ 334	18:58:12	DESCRIBE products	6 row(s) returned	0.0059 sec / 0.00001...
✓ 335	18:58:17	UPDATE products SET price = REPLACE(price, '\$', '')	100 row(s) affected Rows matched: 100 Changed: 100 Warnings: 0	0.0071 sec
✓ 336	18:58:59	ALTER TABLE products MODIFY price DECIMAL(10,2)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0	0.033 sec
✓ 337	18:59:02	UPDATE products SET product_name = CONCAT(UPPER(LEFT(product_name, 1)), LOWER(SUBSTRING(product_name, 2)))	77 row(s) affected Rows matched: 100 Changed: 77 Warnings: 0	0.0068 sec
✓ 338	18:59:07	SELECT * FROM data_users	5000 row(s) returned	0.00070 sec / 0.010 s...
✓ 339	18:59:10	DESCRIBE data_users	11 row(s) returned	0.0014 sec / 0.00006...
✓ 340	18:59:13	DESCRIBE data_users	11 row(s) returned	0.0018 sec / 0.00002...
✓ 341	18:59:25	UPDATE products SET product_name = CONCAT(UPPER(LEFT(product_name, 1)), LOWER(SUBSTRING(product_name, 2)))	0 row(s) affected Rows matched: 100 Changed: 0 Warnings: 0	0.00060 sec
✓ 342	18:59:31	SELECT * FROM data_users	5000 row(s) returned	0.00075 sec / 0.0087...
✓ 343	18:59:49	SELECT * FROM products	100 row(s) returned	0.00061 sec / 0.0000...
✓ 344	19:00:35	UPDATE data_users SET birth_date = STR_TO_DATE(birth_date, '%b %d, %Y')	5000 row(s) affected Rows matched: 5000 Changed: 5000 Warnings: 0	0.102 sec

José Messias Garcia da Silva Ferreira

SPRINT 4 - NIVEL 1

Exercici 1

Realitza una subconsulta que mostri tots els usuaris amb més de 80 transaccions utilitzant almenys 2 taules.

```
8 •  SELECT
9     du.id,
10    CONCAT(name, ' ', surname) AS full_name,
11  (SELECT COUNT(*)
12   FROM transactions t
13  WHERE t.user_id = du.id) AS number_of_transactions -- SUBQUERY WITH THE NUMBER_OF_TRANSACTIONS
14  FROM data_users du
15 WHERE (SELECT COUNT(*)
16   FROM transactions t
17  WHERE t.user_id = du.id) > 80 -- SUBQUERY TO FIND THE NUMBER OF THE TRANSACTION GREATER THAN 80
18 ORDER BY number_of_transactions;
19
20
21 •  SELECT
22     du.id,
23     CONCAT(du.name, ' ', du.surname) AS full_name
24  FROM data_users du
25 WHERE (
26   SELECT COUNT(*)
27   FROM transactions t
28  WHERE t.user_id = du.id
29 ) > 80;
30
```

100% | 27:28 |

Result Grid Filter Rows: Search Export:

id	full_name
185	Molly Gilliam
289	Dxwgi Hwcru
318	Bnyr Astuw
454	Sfizzoh Xavfridxs

Result 171 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
480	13:11:15	SELECT ROUND(AVG(t.amount),2) AS avg....	371 row(s) returned	0.0084 sec / 0.00012...
481	13:19:10	SELECT du.id, CONCAT(du.name, ' ', du.s... 4 row(s) returned		0.071 sec / 0.00009...


```

28
29
30
31
32 /*
33 Exercise 2 - Sprint 4 - Level 1
34 Show the average amount per IBAN of the credit cards for the company Donec Ltd, using at least 2 tables.
35 */
36
37 • SELECT
38     ROUND(AVG(t.amount),2) AS avg_sales,
39     cc.iban,
40     co.company_name
41 FROM transactions t
42 JOIN credit_cards cc ON t.card_id = cc.id
43 JOIN companies co ON t.business_id = co.id
44 WHERE co.company_name = 'Donec Ltd'
45 GROUP BY cc.iban;
46
47 /*
48 Exercise 1 - Sprint 4 - Level 2
49 Create a new table that reflects the status of credit cards based on: if the last three transactions have been declined,
50 then it is inactive; if at least one is not declined, then it is active. Based on this table, answer:
51 How many cards are active?
52 */
100% 26:42

```

Result Grid Filter Rows: Search Export:

avg_sales	iban	company_name
257.37	XX776752917845952975555640	Donec Ltd
139.59	XX413827362289719304908990	Donec Ltd
240.41	XX347787246070769610780308	Donec Ltd
188.58	XX688768436543090894854602	Donec Ltd

Result Grid Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 481	13:19:10	SELECT du.id, CONCAT(du.name, ' ', du.s...	4 row(s) returned	0.071 sec / 0.000009...
✓ 482	13:21:39	SELECT ROUND(AVG(t.amount),2) AS avg....	371 row(s) returned	0.0069 sec / 0.0000...

José Messias Garcia da Silva Ferreira

SPRINT 4 - NIVEL 2

Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en: si les tres últimes transaccions han estat declinades aleshores és inactiu, si almenys una no és rebutjada aleshores és actiu. Partint d'aquesta taula respon:

```
Create a new table that reflects the status of credit cards based on: if the last three transactions have been declined, then it is inactive; if at least one is not declined, then it is active. Based on this table, answer:  
51  */  
52  
53 • CREATE TABLE card_status AS  
54     SELECT  
55         card_id,  
56     CASE  
57         WHEN SUM(CASE WHEN number_of_row <= 3 AND declined THEN 1 ELSE 0 END) = 3 THEN 'inactive'  
58             ELSE 'active'  
59         END AS status_card  
60     FROM (  
61         SELECT  
62             card_id,  
63             declined,  
64             ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC, id DESC) AS number_of_row  
65         FROM transactions) t  
66     GROUP BY card_id;  
67  
68 /*  
69 How many cards are active?  
70 */  
100% 23:53 |  
  
> SELECT cc.id,  
ROW_NUMBER() OVER(ORDER BY cc.id) AS row_numberw,  
CASE  
    WHEN SUM(t.declined) = 3 AND COUNT(t.declined) = 3 THEN 'inactive'  
    ELSE 'active'  
END AS card_status  
FROM credit_cards cc  
JOIN transactions t ON cc.id = t.card_id  
GROUP BY cc.id  
HAVING card_status = 'active'  
Vertical Output Vertical Output Apply Revert  
  
Action Output   


|       | Time     | Action                                                                                                                                 | Response          | Duration / Fetch Time |
|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------|-------------------|-----------------------|
| ✓ 492 | 13:38:35 | DROP TABLE card_status                                                                                                                 | 0 row(s) affected | 0.018 sec             |
| ✓ 493 | 13:38:49 | CREATE TABLE card_status AS SELECT card_id, CASE WHEN SUM(CASE WHEN nu... 5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0 |                   | 0.396 sec             |


```

Exercici 1

Quantes targetes estan actives?

```
61   SELECT
62       card_id,
63       declined,
64       ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC, id DESC) AS number_of_row
65   FROM transactions) t
66 GROUP BY card_id;
67
68 /*
69 How many cards are active?
70 */
71 • SELECT * FROM card_status;
72
73 • SELECT COUNT(*) AS active_cards
74   FROM card_status
75   WHERE status_card = 'Active';
76
77
78 /*
79 Exercise 1 - Sprint 4 - Level 3
80 Create a table that allows us to join the data from the new products.csv file with the database,
81 taking into account that transaction contains product ids. Generate the following query:
```

100% 31:73

Result Grid Filter Rows: Search Export:

active_cards
4995

Result Grid Form Editor

Vertical Output Vertical Output Result 178 Read Only

Action Output

Time	Action	Response	Duration / Fetch Time
495 13:39:36	SELECT COUNT(*) AS active_cards FROM card_status WHERE status_card = 'Active'	1 row(s) returned	0.0045 sec / 0.00001...
496 13:39:49	SELECT COUNT(*) AS active_cards FROM card_status WHERE status_card = 'Active'	1 row(s) returned	0.0040 sec / 0.0000...

José Messias Garcia da Silva Ferreira

SPRINT 4 - NIVEL 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.

```
Exercise 1 - Sprint 4 - Level 3
Create a table that allows us to join the data from the new products.csv file with the database,
taking into account that transaction contains product_ids. Generate the following query:

87
88 | We need to know the number of times each product has been sold.
89 | /*
90 |
91 • SELECT
92     p.id,
93     p.product_name,
94     COUNT(DISTINCT t.id) AS total_sales
95     FROM transaction_products tp
96     JOIN products p ON tp.product_id = p.id
97     JOIN transactions t ON tp.transaction_id = t.id
98     WHERE t.declined = 0
99     GROUP BY p.id, p.product_name
100    ORDER BY p.product_name;
101
102 /**
103 | THE BRIDGE TABLE ALREADY ESTABLISHES THE RELATIONSHIP BETWEEN PRODUCTS AND TRANSACTIONS.
104 | I JUST ADDED THE INFORMATION USING JOINS AND COUNT TO GET THE TOTAL SALES.
105 /**
106
100% 18:96
```

Result Grid Filter Rows: Search Export:

id	product_name	total_sales
85	Bastard of north	2534
22	Chewbacca mustafar	2437
28	Chewbacca mustafar	2572
64	Direwolf	2562

Vertical Output Vertical Output Result 173 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
✗	483 13:23:50	SELECT COUNT(*) AS active_cars FROM card_status WHERE status_card = 'Active'	Error Code: 1146. Table 'eurocommerce.card_status' doesn't exist	0.0014 sec
✓	484 13:27:34	SELECT p.id, p.product_name, COUNT(DISTINCT t.id) AS total_sales FROM transacti...	100 row(s) returned	1.691 sec / 0.000019...