

## Creacio de la base de dades i importacio de dades

```
26 CREATE SCHEMA IF NOT EXISTS company_sprint4;
27 USE company_sprint4;
28 SHOW GLOBAL VARIABLES LIKE 'local_infile';
29
30 SET GLOBAL local_infile=1;
31 SHOW GLOBAL VARIABLES LIKE 'local_infile';
32
33 CREATE TABLE IF NOT EXISTS users (
34     id INT NOT NULL PRIMARY KEY,
35     name VARCHAR(100),
36     surname VARCHAR(100),
37     phone VARCHAR(50),
38     email VARCHAR(255),
39     birth_date VARCHAR(50),
40     country VARCHAR(50),
41     city VARCHAR(100),
42 )
```

Result Grid

Variable_name	Value
local_infile	ON

Result 1 Result 2 ×

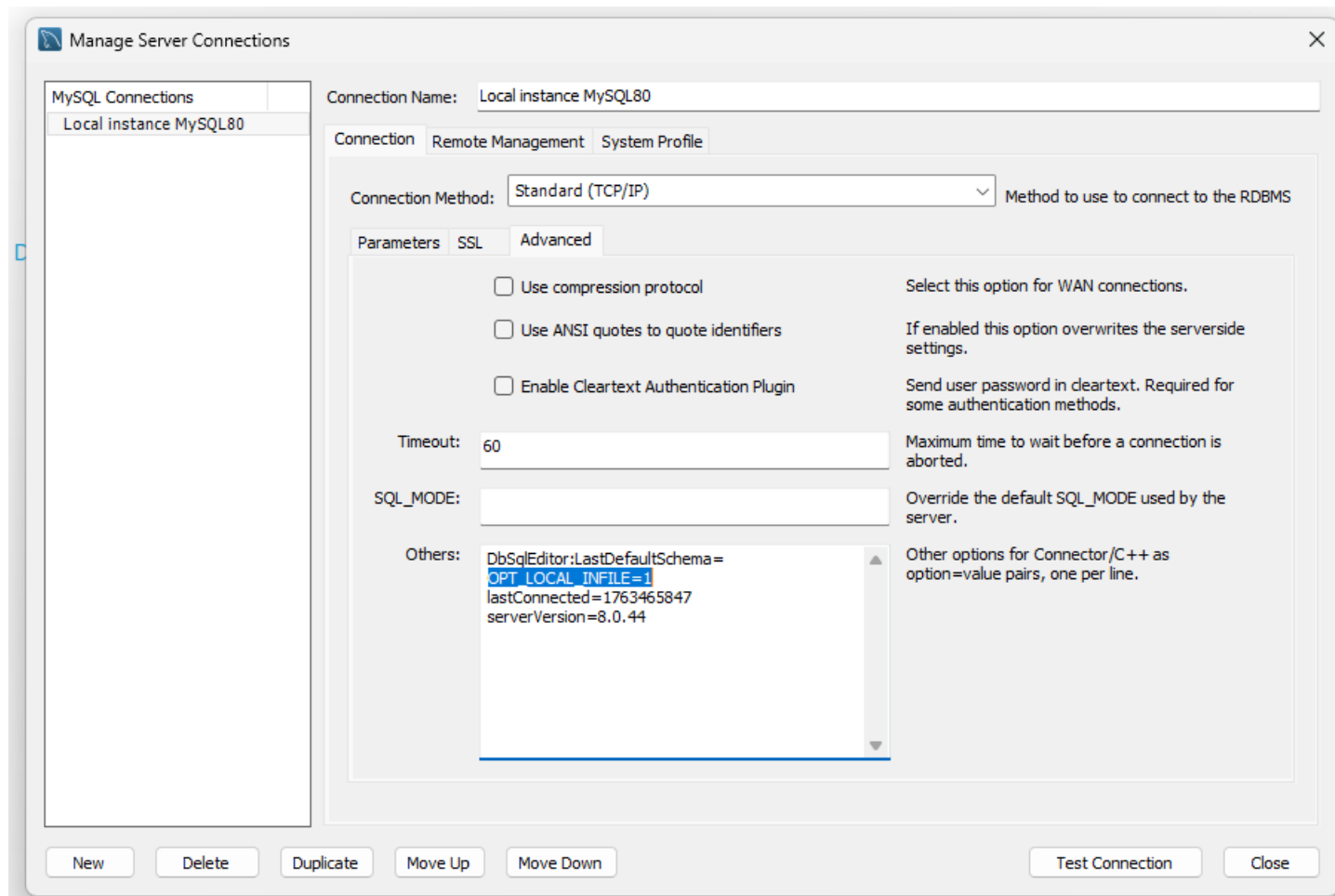
Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	17:36:56	CREATE SCHEMA IF NOT EXISTS company_sprint4	1 row(s) affected	0.000 sec
✓ 2	17:36:56	USE company_sprint4	0 row(s) affected	0.000 sec
✓ 3	17:36:56	SHOW GLOBAL VARIABLES LIKE 'local_infile'	1 row(s) returned	0.015 sec / 0.000 sec
✓ 4	17:36:56	SET GLOBAL local_infile=1	0 row(s) affected	0.000 sec
✓ 5	17:36:56	SHOW GLOBAL VARIABLES LIKE 'local_infile'	1 row(s) returned	0.016 sec / 0.000 sec

Tot i que vaig aplicar el valor "1" a la configuració "local\_infile", encara no vaig poder carregar les dades a les taules.

Vaig haver d'introduir la línia subratllada per poder importar dades des de fitxers locals.



The screenshot displays a database management interface with a left-hand sidebar showing a schema tree. The main area contains a SQL script with four 'LOAD DATA LOCAL INFILE' statements. The bottom section shows the execution output for these statements.

**SQL Script:**

```

85 IGNORE 1 LINES;
86
87 • LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/european_users.csv'
88 INTO TABLE users
89 FIELDS TERMINATED BY ','
90 ENCLOSED BY '"'
91 LINES TERMINATED BY '\n'
92 IGNORE 1 LINES;
93
94 • LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/credit_cards.csv'
95 INTO TABLE credit_cards
96 FIELDS TERMINATED BY ','
97 ENCLOSED BY '"'
98 LINES TERMINATED BY '\n'
99 IGNORE 1 LINES;
100
101 • LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/companies.csv'
102 INTO TABLE companies
103 FIELDS TERMINATED BY ','
104 ENCLOSED BY '"'
105 LINES TERMINATED BY '\n'
106 IGNORE 1 LINES;
107
108 • LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/transactions.csv'
109 INTO TABLE transactions
110 FIELDS TERMINATED BY ';'
111 ENCLOSED BY '"'
112 LINES TERMINATED BY '\n'
113 IGNORE 1 LINES;

```

**Output:**

#	Time	Action	Message	Duration / Fetch
5	17:38:48	LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/american_users.csv' INTO TABLE users FIELDS TERMINATED BY ';' ENCLOSED BY '"'	1010 row(s) affected Records: 1010 Deleted: 0 Skipped: 0 Warnings: 0	0.047 sec
6	17:38:48	LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/european_users.csv' INTO TABLE users FIELDS TERMINATED BY ';' ENCLOSED BY '"'	3990 row(s) affected Records: 3990 Deleted: 0 Skipped: 0 Warnings: 0	0.188 sec
7	17:38:48	LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/credit_cards.csv' INTO TABLE credit_cards FIELDS TERMINATED BY ';' ENCLOSED BY '"'	5000 row(s) affected Records: 5000 Deleted: 0 Skipped: 0 Warnings: 0	0.234 sec
8	17:38:48	LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/companies.csv' INTO TABLE companies FIELDS TERMINATED BY ';' ENCLOSED BY '"'	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.016 sec
9	17:38:48	LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/transactions.csv' INTO TABLE transactions FIELDS TERMINATED BY ';' ENCLOSED BY '"'	100000 row(s) affected Records: 100000 Deleted: 0 Skipped: 0 Warnings: 0	3.593 sec

He suprimit els valors NOT NULL de les PRIMARY KEYS al fitxier SQL després que em comentessis que era un valor per defecte a la correcció del meu Sprint 3.

Vaig passar els fitxers per Gemini per determinar el llarg optimitzado per cada tipus de dades però he pres bona nota del teu comentari a la correccio del meu Sprint 3 que VARCHAR(255) és un datatype estàndard per integrar els valors en una taula.

Vaig assignar NOT NULL a camps específics de les taules on considerava que les dades eren crucials. Especialment, aplico això a la major part de la taula "credit\_card", ja que conté informació financera essencial que és interdependent entre si per permetre el pagament.

A la taula "transaccions", només he aplicat NOT NULL a les columnes indispensables per a identificar la transaccio, el seu import i les parts implicades.

He integrat les dades dels usuaris americans i europeus a la taula "users".

Per seguretat, tanco l'accés als fitxers locals :

```
92 • SET GLOBAL local_infile=0;
93 • SHOW GLOBAL VARIABLES LIKE 'local_infile';
94
95
```

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

Variable_name	Value
local_infile	OFF

Result 1 x | Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	14:00:13	SET GLOBAL local_infile=0	0 row(s) affected	0.000 sec
2	14:00:13	SHOW GLOBAL VARIABLES LIKE 'local_infile'	1 row(s) returned	0.000 sec / 0.000 sec

Afegeixo les FOREIGN KEYS que enllacen la taula de "transaccions" amb les altres.

He suprimit els valors ON DELETE / ON UPDATE RESTRICT al fitxier SQL després que em comentessis que era un valor per defecte a la correcció del meu Sprint 3.

The screenshot displays a database management interface. On the left, a tree view shows the database structure for 'company\_sprint4', including tables like 'companies', 'credit\_cards', and 'transactions', along with their columns and foreign keys. The 'users' table is also visible in the 'Information' pane.

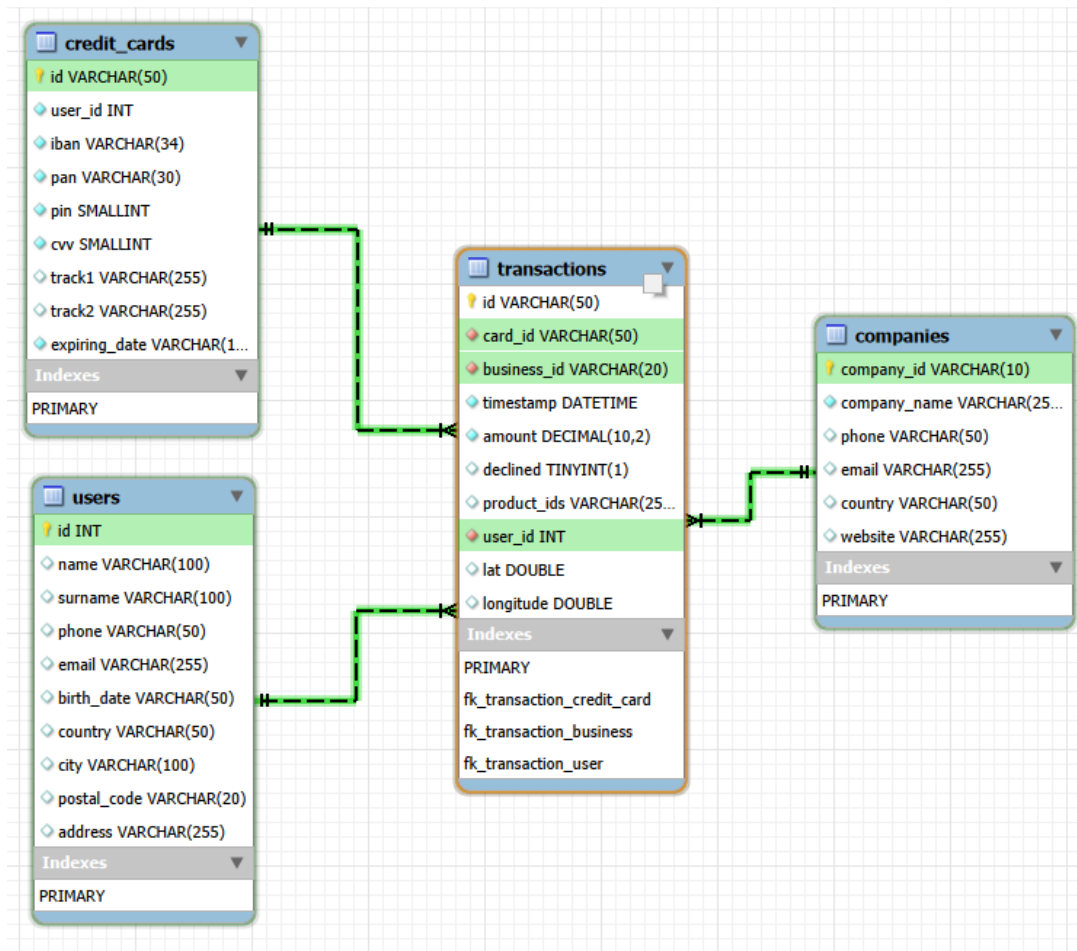
The main area shows a SQL script with the following queries:

```
109 INTO TABLE transactions
110 FIELDS TERMINATED BY ','
111 ENCLOSED BY '"'
112 LINES TERMINATED BY '\n'
113 IGNORE 1 LINES;
114
115 • ALTER TABLE transactions
116 ADD CONSTRAINT fk_transaction_credit_card
117 FOREIGN KEY (card_id)
118 REFERENCES credit_cards(id)
119 ON DELETE RESTRICT
120 ON UPDATE RESTRICT;
121
122 • ALTER TABLE transactions
123 ADD CONSTRAINT fk_transaction_business
124 FOREIGN KEY (business_id)
125 REFERENCES companies(company_id)
126 ON DELETE RESTRICT
127 ON UPDATE RESTRICT;
128
129 • ALTER TABLE transactions
130 ADD CONSTRAINT fk_transaction_user
131 FOREIGN KEY (user_id)
132 REFERENCES users(id)
133 ON DELETE RESTRICT
134 ON UPDATE RESTRICT;
```

The 'Output' pane at the bottom shows the execution results of these queries:

#	Time	Action	Message	Duration / Fetch
1	17:44:19	ALTER TABLE transactions ADD CONSTRAINT fk_transaction_credit_card FOREIGN KEY (card_id) REFERENCES credit_cards(id) ON DELETE RESTRICT ON UPDATE RESTRICT;	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	2.625 sec
2	17:44:21	ALTER TABLE transactions ADD CONSTRAINT fk_transaction_business FOREIGN KEY (business_id) REFERENCES companies(company_id) ON DELETE RESTRICT ON UPDATE RESTRICT;	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	3.078 sec
3	17:44:25	ALTER TABLE transactions ADD CONSTRAINT fk_transaction_user FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE RESTRICT ON UPDATE RESTRICT;	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	3.781 sec

Estat final de l'esquema :



## Nivell 1 - Exercici 1

Treballa en totes les transaccions que requereix l'exercici, no només en les vendes.

```
116 -- ***** Nivell 1 - Exercici 1 *****
117
118 • SELECT u.id AS "ID client", u.name AS "Nom", u.surname AS "Cognom", u.country AS "Pais", (
119     SELECT COUNT(t.id)
120     FROM transactions t
121     WHERE t.user_id = u.id
122 ) AS "Nombre de transaccions"
123 FROM users u
124 WHERE u.id IN (
125     SELECT t.user_id
126     FROM transactions t
127     GROUP BY t.user_id
128     HAVING COUNT(t.id) > 80
129 );
130
```

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

	ID client	Nom	Cognom	Pais	Nombre de transaccions
▶	185	Molly	Gilliam	United Kingdom	110
	289	Dxwgi	Hwcru	Germany	94
	318	Bnyr	Astuw	Italy	91
	454	Sfzzoh	Xgvfridxs	Poland	81

Result 2 x

Output					Read Only	
Action Output						
#	Time	Action	Message		Duration / Fetch	
✓ 1	11:09:41	SELECT u.id AS "ID client", u.name AS "Nom", u.surname AS "Cognom", u.country AS "Pais", ( SELECT COUNT(t.id) FROM transactions t ...	4 row(s) returned		0.063 sec / 0.000 sec	

## Nivell 1 - Exercici 2

El nom de l'empresa s'utilitza com a filtre, de manera que per diferenciar qualsevol possible nom duplicat entre dues empreses diferents, utilitzo "company\_id" per identificar els resultats i agrupar-los. També vaig ordenar els resultats de manera creixent, ja que em semblava que tenia sentit a partir de les dades sol·licitades, és a dir, les mitjanes d'import de les transaccions per targeta.

```
130 -- ***** Nivell 1 - Exercici 2 *****
131
132 • SELECT c.company_id AS "ID Companya", cc.iban AS "IBAN", ROUND(AVG(t.amount), 2) AS "Mitjana d'amount per IBAN"
133 FROM transactions t
134 JOIN companies c
135 ON t.business_id = c.company_id
136 JOIN credit_cards cc
137 ON t.card_id = cc.id
138 WHERE c.company_name = 'Donec Ltd'
139 GROUP BY c.company_id, cc.iban
140 ORDER BY AVG(t.amount) DESC;
141
```

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

	ID Companya	IBAN	Mitjana d'amount per IBAN
▶	b-2242	XX383017813919620199366352	680.69
	b-2242	XX637706357397570394973913	680.01
	b-2242	XX971393971465292202312259	645.46
	b-2242	XX171847116928892375969307	628.89
	b-2242	XX225424638818542406223575	608.68
	b-2242	XX748890729057195711766071	607.29
	b-2242	TN9614563570667381893122	605.41
	b-2242	XX481908034037364242591185	605.36
	b-2242	XX194675519739256335753508	597.19
	b-2242	XX215962766061967195493437	594.26
	b-2242	XX449322320826890721001443	591.61
	b-2242	XX535185492735704229474237	570.09
	b-2242	CH9552373968796160224	566.38
	b-2242	XX347605377125637880303131	561.80
	b-2242	XX688471446697921912860304	543.42
	b-2242	XX605533964582458704105956	542.00

Result 5 x

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	14:09:26	SELECT c.company_id AS "ID Companya", cc.iban AS "IBAN", ROUND(AVG(t.amount), 2) AS "Mitjana d'amount per IBAN" FROM transactions t J...	371 row(s) returned	0.000 sec / 0.000 sec



## Nivell 2 - Exercici 1

Faig servir CTE per crear un informe/taula filtrat. En el tercer pas, el d'assignació de valors, afegeixo els noms de les columnes en català, utilitzant comiats invertits en el cas de la segona columna. Això és per assegurar-me que la taula “cc\_status” estarà llesta per utilitzar tal com jo la creo.

Crec que també podria haver estat útil crear aquest informe com a VIEW, ja que la taula en si no pertany directament a l'esquema de dades, pot ser útil de manera regular, i que la VIEW hauria permès el mateix filtratge que estic executant a la pàgina següent.

The screenshot shows a SQL IDE interface. On the left, a tree view displays the database structure, including tables like 'cc\_status', 'companies', 'credit\_cards', 'transactions', and 'users'. The main editor area contains the following SQL code:

```
170
171 -- ***** Nivell 2 - Exercici 1 *****
172
173 CREATE TABLE cc_status AS WITH OrderedTransactions AS (
174     SELECT t.card_id, t.declined, t.timestamp, ROW_NUMBER() OVER (
175         PARTITION BY t.card_id
176         ORDER BY t.timestamp DESC) AS transactions_order
177     FROM transactions t
178 ),
179
180 LastThree AS (
181     SELECT ot.card_id, SUM(ot.declined) AS declined_in_last_three
182     FROM OrderedTransactions ot
183     WHERE ot.transactions_order <= 3
184     GROUP BY ot.card_id
185 )
186
187 SELECT lt.card_id AS ID_targeta,
188     CASE
189         WHEN lt.declined_in_last_three = 3 THEN "Inactiva"
190         ELSE "Activa"
191     END AS `Estat targeta`
192 FROM LastThree lt;
```

Below the code editor, the 'Output' pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	08:57:15	CREATE TABLE cc_status AS WITH OrderedTransactions AS ( SELECT t.card_id, t.declined, t.timestamp, ROW_NUMBER() OVER ( PARTITION ...	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0	0.516 sec

Des d'aquesta nova taula, faig una consulta COUNT filtrada per respondre a l'exercici.

196 •  
197  
198

```
SELECT COUNT(*) AS "Total targetes actives"
FROM cc_status
WHERE `Estat targeta` = 'Activa';
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Total targetes actives
4995

Result 19

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	08:57:15	CREATE TABLE cc_status AS WITH OrderedTransactions AS ( SELECT t.card_id, t.declined, t.timestamp, ROW_NUMBER() OVER ( PARTITION ...	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0	0.516 sec
✓ 2	09:02:27	SELECT COUNT(*) AS "Total targetes actives" FROM cc_status WHERE `Estat targeta` = 'Activa'	1 row(s) returned	0.000 sec / 0.000 sec

## Nivell 3 - Exercici 1

Torno a obrir l'accés a les fitxers locals.

```
175  -- ***** Nivell 3 - Exercici 1 *****
176
177
178 • SHOW GLOBAL VARIABLES LIKE 'local_infile';
179 • SET GLOBAL local_infile=0;
180 • SHOW GLOBAL VARIABLES LIKE 'local_infile';
181
182
```

Result Grid Filter Rows: Export: Wrap Cell Content:

Variable_name	Value
local_infile	OFF

Result 6 x

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	14:17:03	SHOW GLOBAL VARIABLES LIKE 'local_infile'	1 row(s) returned	0.000 sec / 0.000 sec

```

175  -- ***** Nivell 3 - Exercici 1 *****
176
177
178 • SHOW GLOBAL VARIABLES LIKE 'local_infile';
179 • SET GLOBAL local_infile=1;
180 • SHOW GLOBAL VARIABLES LIKE 'local_infile';
181
182

```

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

Variable_name	Value
local_infile	ON

Result Grid

Form Editor

Field Types

Query Stats

Result 8 x

Read Only

Output					
Action Output					
#	Time	Action	Message	Duration / Fetch	
1	14:17:59	SET GLOBAL local_infile=1	0 row(s) affected	0.000 sec	
2	14:17:59	SHOW GLOBAL VARIABLES LIKE 'local_infile'	1 row(s) returned	0.000 sec / 0.000 sec	

Creacio taula "products".

The screenshot displays the SQL Server Enterprise Manager interface on the left and the SQL Server Enterprise Edition query window on the right.

**SQL Server Enterprise Manager (Left Panel):**

- Server: company
- Database: company\_sprint4
- Tables:
  - cc\_status
  - companies
  - credit\_cards
  - transactions
  - users
- Views
- Stored Procedures
- Functions
- hola
- sakila
- sys
- test
- transactions
- world

Administration | Schemas | Information

No object selected

**SQL Server Enterprise Edition (Right Panel):**

Query window showing the execution of SQL commands:

```
174
175
176
177 -- ***** Nivell 3 - Exercici 1 *****
178
179
180 CREATE TABLE IF NOT EXISTS products (
181     id INT NOT NULL PRIMARY KEY,
182     product_name VARCHAR(255),
183     price DECIMAL(10, 2),
184     colour VARCHAR(50),
185     weight DECIMAL(5, 2),
186     warehouse_id VARCHAR(10)
187 );
188
189 LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/products.csv'
190 INTO TABLE products
191 FIELDS TERMINATED BY ','
192 ENCLOSED BY '"'
193 LINES TERMINATED BY '\n'
194 IGNORE 1 LINES
195 (id, product_name, @price_string, colour, weight, warehouse_id)
196 SET price = REPLACE(@price_string, '$', '');
197
```

**Output Window (Bottom):**

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	09:10:07	CREATE TABLE IF NOT EXISTS products ( id INT NOT NULL PRIMARY KEY, product_name VARCHAR(255), price DECIMAL(10, 2), co...	0 row(s) affected	0.032 sec

Importacio dades a la taula "products". La columna "preu" inclou un símbol \$ a cada fila que cal eliminar per importar les dades correctament, que és l'objectiu de les línies 195 i 196 de la captura de pantalla següent.

The screenshot displays the SQL Server Enterprise Manager interface on the left, showing a tree view of the 'company\_sprint4' database. The 'Products' table is highlighted under the 'Tables' folder. The main window shows the SQL script for creating and loading the 'products' table.

```
174
175
176
177 -- ***** Nivell 3 - Exercici 1 *****
178
179
180 CREATE TABLE IF NOT EXISTS products (
181     id INT NOT NULL PRIMARY KEY,
182     product_name VARCHAR(255),
183     price DECIMAL(10, 2),
184     colour VARCHAR(50),
185     weight DECIMAL(5, 2),
186     warehouse_id VARCHAR(10)
187 );
188
189 LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/products.csv'
190 INTO TABLE products
191 FIELDS TERMINATED BY ','
192 ENCLOSED BY '"'
193 LINES TERMINATED BY '\n'
194 IGNORE 1 LINES
195 (id, product_name, @price_string, colour, weight, warehouse_id)
196 SET price = REPLACE(@price_string, '$', '');
197
```

The Output window at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	09:10:07	CREATE TABLE IF NOT EXISTS products ( id INT NOT NULL PRIMARY KEY, product_name VARCHAR(255), price DECIMAL(10, 2), co...	0 row(s) affected	0.032 sec
2	09:10:32	LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/products.csv' INTO TABLE products FIELDS TERMINATED BY ',' ENCLOSED BY '"' LIN...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.015 sec

Torno a tancar l'accès a les fitxers locals.

202 • SET GLOBAL local\_infile=0;

203 • SHOW GLOBAL VARIABLES LIKE 'local\_infile';

204

205

...

Result Grid

Filter Rows:

Export:

Wrap Cell Contents

Variable_name	Value
local_infile	OFF

Result 10

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	14:19:24	SET GLOBAL local_infile=0	0 row(s) affected	0.000 sec
✓ 2	14:19:24	SHOW GLOBAL VARIABLES LIKE 'local_infile'	1 row(s) returned	0.000 sec / 0.000 sec

Cal una taula intermèdia per extreure la informació per producte, que es troba en una sola columna de la taula de transaccions.

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'company\_sprint4' database is expanded, showing a schema with tables like 'products', 'transactions', and 'users'. The 'products' table structure is detailed below the schema tree:

Column	DataType	PK
id	int	PK
product_name	varchar(255)	
price	decimal(10,2)	
colour	varchar(50)	
weight	decimal(5,2)	
warehouse_id	varchar(10)	

The main pane shows a SQL script with the following lines:

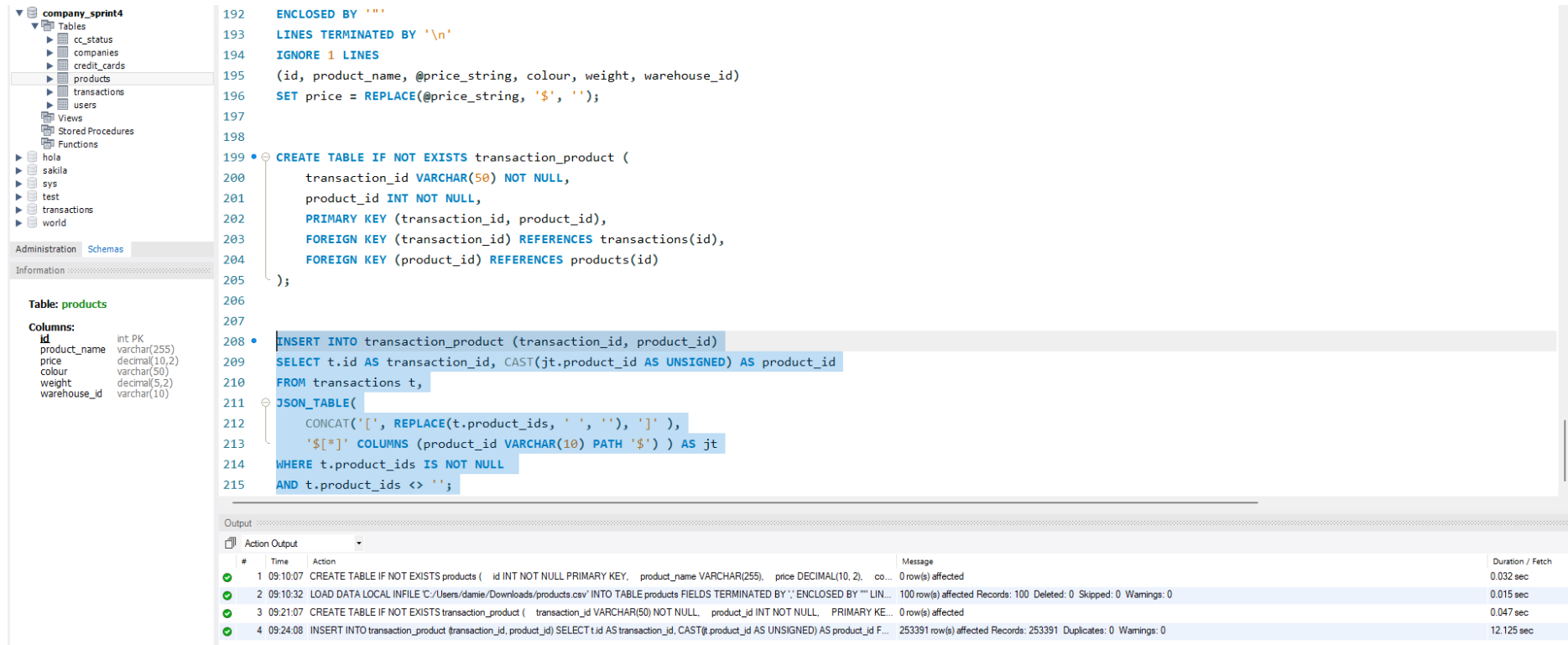
```
183 price DECIMAL(10, 2),
184 colour VARCHAR(50),
185 weight DECIMAL(5, 2),
186 warehouse_id VARCHAR(10)
187 );
188
189 • LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/products.csv'
190 INTO TABLE products
191 FIELDS TERMINATED BY ','
192 ENCLOSED BY '"'
193 LINES TERMINATED BY '\n'
194 IGNORE 1 LINES
195 (id, product_name, @price_string, colour, weight, warehouse_id)
196 SET price = REPLACE(@price_string, '$', '');
197
198
199 • CREATE TABLE IF NOT EXISTS transaction_product (
200     transaction_id VARCHAR(50) NOT NULL,
201     product_id INT NOT NULL,
202     PRIMARY KEY (transaction_id, product_id),
203     FOREIGN KEY (transaction_id) REFERENCES transactions(id),
204     FOREIGN KEY (product_id) REFERENCES products(id)
205 );
206
```

The bottom pane shows the 'Output' window with the following results:

#	Time	Action	Message	Duration / Fetch
1	09:10:07	CREATE TABLE IF NOT EXISTS products ( id INT NOT NULL PRIMARY KEY, product_name VARCHAR(255), price DECIMAL(10, 2), co...	0 row(s) affected	0.032 sec
2	09:10:32	LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/products.csv' INTO TABLE products FIELDS TERMINATED BY ',' ENCLOSED BY '"' LIN...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.015 sec
3	09:21:07	CREATE TABLE IF NOT EXISTS transaction_product ( transaction_id VARCHAR(50) NOT NULL, product_id INT NOT NULL, PRIMARY KE...	0 row(s) affected	0.047 sec



La funció `JSON_TABLE()` per a la divisió de cadenes a MySQL 8.0+ permet minimitzar l'ús de la CPU i escalar molt millor amb taules de transaccions grans en comparació amb mètodes iteratius com ara els CTE recursius.



The screenshot displays the MySQL Workbench interface. On the left, the 'Schemas' tab is active, showing a tree view of databases including 'company\_sprint4'. Under 'company\_sprint4', the 'Tables' folder is expanded, listing tables like 'cc\_status', 'companies', 'credit\_cards', 'products', 'transactions', and 'users'. The 'products' table is selected, and its structure is shown in the 'Table: products' section:

Column	Type
id	int PK
product_name	varchar(255)
price	decimal(10,2)
colour	varchar(50)
weight	decimal(5,2)
warehouse_id	varchar(10)

The main SQL editor contains the following script:

```
192 ENCLOSED BY ''''
193 LINES TERMINATED BY '\n'
194 IGNORE 1 LINES
195 (id, product_name, @price_string, colour, weight, warehouse_id)
196 SET price = REPLACE(@price_string, '$', '');
197
198
199 CREATE TABLE IF NOT EXISTS transaction_product (
200     transaction_id VARCHAR(50) NOT NULL,
201     product_id INT NOT NULL,
202     PRIMARY KEY (transaction_id, product_id),
203     FOREIGN KEY (transaction_id) REFERENCES transactions(id),
204     FOREIGN KEY (product_id) REFERENCES products(id)
205 );
206
207
208 INSERT INTO transaction_product (transaction_id, product_id)
209 SELECT t.id AS transaction_id, CAST(jt.product_id AS UNSIGNED) AS product_id
210 FROM transactions t,
211 JSON_TABLE(
212     CONCAT('[', REPLACE(t.product_ids, ' ', ''), ']' ),
213     '$[*]' COLUMNS (product_id VARCHAR(10) PATH '$') ) AS jt
214 WHERE t.product_ids IS NOT NULL
215 AND t.product_ids <> '';
```

The 'Output' tab at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	09:10:07	CREATE TABLE IF NOT EXISTS products ( id INT NOT NULL PRIMARY KEY, product_name VARCHAR(255), price DECIMAL(10, 2), co...	0 row(s) affected	0.032 sec
2	09:10:32	LOAD DATA LOCAL INFILE 'C:/Users/damie/Downloads/products.csv' INTO TABLE products FIELDS TERMINATED BY ';' ENCLOSED BY '' LIN...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0	0.015 sec
3	09:21:07	CREATE TABLE IF NOT EXISTS transaction_product ( transaction_id VARCHAR(50) NOT NULL, product_id INT NOT NULL, PRIMARY KE...	0 row(s) affected	0.047 sec
4	09:24:08	INSERT INTO transaction_product (transaction_id, product_id) SELECT t.id AS transaction_id, CAST(t product_id AS UNSIGNED) AS product_id F...	253391 row(s) affected Records: 253391 Duplicates: 0 Warnings: 0	12.125 sec

Ara es pot executar una consulta SELECT per respondre l'exercici.

Faig servir dos JOIN per poder accedir també al camp "declined", ja que l'exercici només requereix treballar en les vendes. També he ordenat els resultats de manera creixent, ja que semblava que tenia sentit per a un informe d'aquest tipus.

```
228 • SELECT p.id AS "ID producte", p.product_name AS "Nom producte", COUNT(tp.transaction_id) AS "Suma total vendes"
229 FROM transaction_product tp
230 JOIN products p
231 ON tp.product_id = p.id
232 JOIN transactions t
233 ON tp.transaction_id = t.id
234 WHERE t.declined = 0
235 GROUP BY p.id, p.product_name
236 ORDER BY COUNT(tp.transaction_id) DESC;
```

Result Grid			
Filter Rows:			
Exports:   Wrap Cell Content:			
	ID producte	Nom producte	Suma total vendes
▶	52	riverlands the duel	2642
	29	Tully maester Tarly	2627
	21	duel Direwolf	2603
	16	the duel warden	2602
	33	duel warden	2593
	87	sith Jade	2591
	66	mustafar jinn	2590
	48	rock Renly in	2589
	68	Stark Karstark	2587
	23	riverlands north	2586
	88	Stannis warden so...	2582
	73	Dorne bastard	2579
	34	of north	2574
	31	Lannister	2573
	4	warden south duel	2573

Result 11 x

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	14:19:24	SET GLOBAL local_infile=0	0 row(s) affected	0.000 sec
✓ 2	14:19:24	SHOW GLOBAL VARIABLES LIKE 'local_infile'	1 row(s) returned	0.000 sec / 0.000 sec
✓ 3	14:21:49	SELECT p.id AS "ID producte", p.product_name AS "Nom producte", COUNT(tp.transaction_id) AS "Suma total vendes" FROM transaction_produ...	100 row(s) returned	1.500 sec / 0.000 sec

Estat final de l'esquema :

