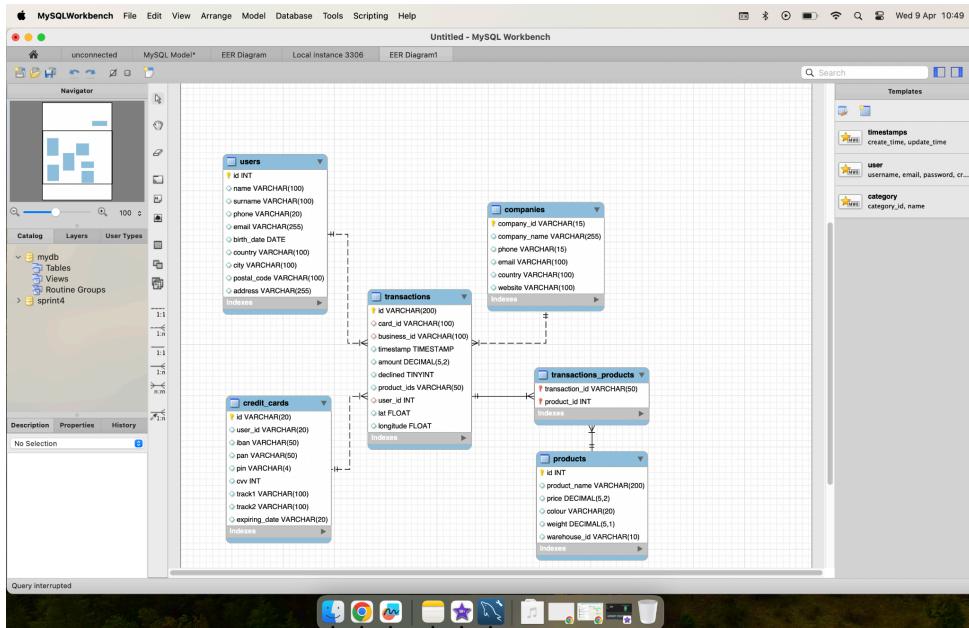


Nivell 1

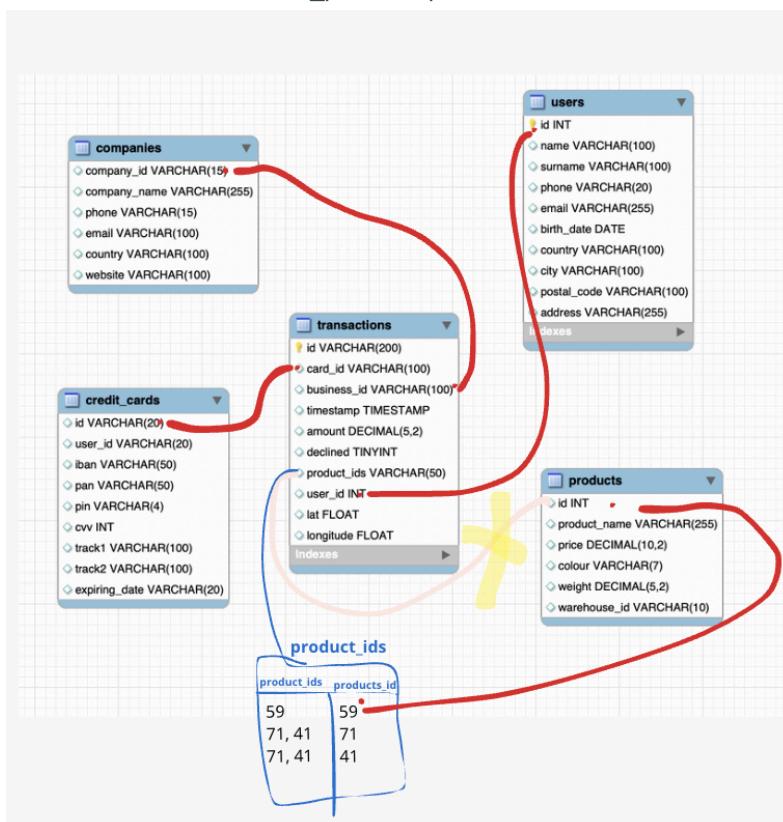
Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguis realitzar les següents consultes:

La base de datos creada va ser aquesta



✓ Actualización: enlace hecho entre transactions

Para crearla he pensado en las relaciones de la tabla de hecho transactions con la de dimensiones, comprendiendo que no era posible crear una relación directa entre la de transactions con products y que hacia falta una transactions_products para relacionar las duas.



Primero he creado la base de datos:

The screenshot shows the MySQL Workbench interface. In the top-left pane, under 'Administration' and 'Schemas', the 'sprint4' schema is selected. The main query editor window contains the following SQL code:

```
1 • CREATE DATABASE sprint4;
```

The status bar at the bottom indicates 'Query Completed'.

Luego he creado las tablas de dimensiones para luego relacionar con la tabla de hechos. Empeze con Companies:

The screenshot shows the MySQL Workbench interface. In the top-left pane, under 'Administration' and 'Schemas', the 'sprint4' schema is selected. The main query editor window contains the following SQL code for creating the 'companies' table:

```
1 • CREATE DATABASE sprint4;
2
3 -- Creando, importando y haciendo el check de la creacion de
4 -- Tabla COMPANIES: creacion, importacion y check
5 • CREATE TABLE IF NOT EXISTS companies(
6     company_id VARCHAR(15) PRIMARY KEY,
7     company_name VARCHAR(255),
8     phone VARCHAR(15),
9     email VARCHAR(100),
10    country VARCHAR (100),
11    website VARCHAR (100)
12 );
13
```

The status bar at the bottom indicates 'Query Completed'.

Y importado los datos de companies.csv via codigo en la tabla, haciendo el check visualizando los datos de la tabla:

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

- Schemas:** sprint4
- Tables:** companies
- Query Editor:**

```

13
14 • LOAD DATA INFILE '/Users/barbara/junqueira/Desktop/RecursosDataAnalyst/Sprint4/companies.csv'
15   INTO TABLE companies
16   FIELDS TERMINATED BY ','
17   ENCLOSED BY '\"'
18   LINES TERMINATED BY '\r\n'
19   IGNORE 1 LINES;
20
21 • SELECT * FROM companies;
22
    
```
- Result Grid:** Displays 100 rows of company data.
- Action Output:**

Time	Action	Response
126 11:52:10	LOAD DATA INFILE '/Users/barbara/junqueira/Desktop/RecursosDataAnalyst/Sprint4/companies.csv'	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
127 11:52:30	SELECT * FROM companies	100 row(s) returned

Exactamente lo mismo con la tabla de Credit_Cards: crear tabla, importar data y conferir importación. Aquí tuve un problema con la terminación de líneas y tuve que sacar o '\r' de la LINES TERMINATED BY.

Algunos archivos funcionaban con solo '\n' y otros necesitan ambos con '\r\n'. He pesquisado y entiendo que depende de donde se ha hecho la creación del archivo. Si el archivo ha sido creado en sistemas basados en Unix/Linux, incluidos macOS y otros sistemas derivados, se usa solo '\n'. Los archivos creados en sistemas Windows típicamente usan la secuencia '\r\n' para representar el final de una línea.

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

- Schemas:** sprint4
- Tables:** credit_cards
- Query Editor:**

```

22
23 -- Table CREDIT_CARDS: creamos, importacion y check
24 • CREATE TABLE IF NOT EXISTS credit_cards(
25   id VARCHAR(20) PRIMARY KEY,
26   user_id VARCHAR(30),
27   iban VARCHAR(30),
28   pin VARCHAR(10),
29   pin_type VARCHAR(10),
30   cvv INT,
31   track1 VARCHAR(100),
32   track2 VARCHAR(100),
33   expiring_date VARCHAR (20)
34 );
35
36 • LOAD DATA INFILE '/Users/barbara/junqueira/Desktop/RecursosDataAnalyst/Sprint4/credit_cards.csv'
37   INTO TABLE credit_cards
38   FIELDS TERMINATED BY ','
39   ENCLOSED BY '\"'
40   LINES TERMINATED BY '\r\n'
41   IGNORE 1 LINES;
42
43 • SELECT * FROM credit_cards;
44
    
```
- Result Grid:** Displays 275 rows of credit card data.
- Action Output:**

Time	Action	Response
128 11:53:33	CREATE TABLE IF NOT EXISTS credit_cards(id VARCHAR(20) PRIMARY KEY, user_...	0 row(s) affected
129 11:53:37	LOAD DATA INFILE '/Users/barbara/junqueira/Desktop/RecursosDataAnalyst/Sprint4/...	275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings: 0
130 11:53:40	SELECT * FROM credit_cards	275 row(s) returned

Luego la creacion de la tabla de users, que tuvo un aspecto especial porque quise insertar el dato de birthday como DATE y no como VARCHAR entonces he usado el SET para, al coger los datos, poder alterar la formatacion de string to date de esa forma: birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y'), en este orden porque esta asi en el archivo para hacer la transformacion. Tambien he juntado todos los 3 archivos de .csv de users de ca + usa + uk en una sola tabla de users.

```

-- Table users: creacion, importacion y check
CREATE TABLE users (
    id INT PRIMARY KEY,
    name VARCHAR(100),
    surname VARCHAR(100),
    phone VARCHAR(100),
    email VARCHAR(100),
    birth_date DATE,
    country VARCHAR(100),
    city VARCHAR(100),
    postal_code VARCHAR(100),
    address VARCHAR(255)
);

LOAD DATA INFILE '/Users/barbarajunqueira/Desktop/RecursosDataAnalyst/jets/Sprint4/users_ca.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY ''
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(@id, @name, @surname, @phone, @email, @birth_date, @country, @city, @postal_code, @address)
SET
    id = @id,
    name = @name,
    surname = @surname,
    phone = @phone,
    email = @email,
    birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y'), -- Convertiendo la fecha para el formato correcto
    country = @country,
    city = @city,
    postal_code = @postal_code,
    address = @address;
;

LOAD DATA INFILE '/Users/barbarajunqueira/Desktop/RecursosDataAnalyst/jets/Sprint4/users_usa.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY ''
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(@id, @name, @surname, @phone, @email, @birth_date, @country, @city, @postal_code, @address)
SET
    id = @id,
    name = @name,
    surname = @surname,
    phone = @phone,
    email = @email,
    birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y'),
    country = @country,
    city = @city,
    postal_code = @postal_code,
    address = @address;
;

LOAD DATA INFILE '/Users/barbarajunqueira/Desktop/RecursosDataAnalyst/jets/Sprint4/users_uk.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY ''
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(@id, @name, @surname, @phone, @email, @birth_date, @country, @city, @postal_code, @address)
SET
    id = @id,
    name = @name,
    surname = @surname,
    phone = @phone,
    email = @email,
    birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y'),
    country = @country,
    city = @city,
    postal_code = @postal_code,
    address = @address;
;

```

Una forma de cambiar los datos es cuando los cargas directamente con un SET, solo que no necesitas nombrar todas las columnas, basta con algo tipo: (id, name, surname, phone, email, @birth_date, country, city, postal_code, address)

SET birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y');

ACTUALIZADO:

```

LOAD DATA INFILE '/Users/barbarajunqueira/Desktop/RecursosDataAnalyst/jets/Sprint4/users_ca.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY ''
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(@id, @name, @surname, @phone, @email, @birth_date, @country, @city, @postal_code, @address)
SET
    birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y'), -- Convertiendo la fecha para el formato correcto
;

LOAD DATA INFILE '/Users/barbarajunqueira/Desktop/RecursosDataAnalyst/jets/Sprint4/users_usa.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY ''
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(@id, @name, @surname, @phone, @email, @birth_date, @country, @city, @postal_code, @address)
SET
    birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y');
;

LOAD DATA INFILE '/Users/barbarajunqueira/Desktop/RecursosDataAnalyst/jets/Sprint4/users_uk.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY ''
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(@id, @name, @surname, @phone, @email, @birth_date, @country, @city, @postal_code, @address)
SET
    birth_date = STR_TO_DATE(@birth_date, '%b %d, %Y');
;

```

Luego la tabla de productos:

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

- Administration** tab selected.
- Schemas** panel: Schema is set to **sprint4**. The **Tables** section lists **companies**, **credit_cards**, **products**, **users**, and **Views**. The **Stored Procedures** and **Functions** sections are empty. A **sys** entry is also present.
- Query Editor**:
 - Text area:

```
-- Tabla PRODUCTS: creacion, importacion y check, que tiene que ser creada posteriori de la tabla de TRANSACTION_PRODUCTS
CREATE TABLE products(
    id INT PRIMARY KEY,
    product_name VARCHAR(200),
    price DECIMAL (5, 2),
    colour VARCHAR(20),
    weight DECIMAL (5, 1),
    warehouse_id VARCHAR(10)
);

LOAD DATA INFILE '/Users/barbara junqueira/Desktop/RecursosDataAnalyst/Sprint4/products.csv'
INTO TABLE products
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';

IGNORE 1 LINES
(@id, @product_name, @price, @colour, @weight, @warehouse_id)
SET id = @id;
@product_name = @product_name,
colour = @colour,
weight = @weight,
warehouse_id = @warehouse_id;
;

SELECT * FROM products;
```
 - Status bar: 50% completion, 24:142 rows.
 - Result Grid: Shows the data from the products CSV file. The columns are **id**, **product_name**, **price**, **colour**, **weight**, and **warehouse_id**. The data includes entries like "Dwarrow Danhoz" with id 1, "Terry Stark" with id 2, and "north of Casterly" with id 7.
 - Action Output: Shows the execution history with three rows:
 - Row 136: 12:02:16 CREATE TABLE products(id INT PRIMARY KEY, ...) 0 row(s) affected
 - Row 137: 12:02:21 LOAD DATA INFILE '/Users/barbara junqueira/Desktop/RecursosDataAnalyst/Sprint4/products.csv' 100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
 - Row 138: 12:02:23 SELECT * FROM products 100 row(s) returned
 - Contextual help: A tooltip on the right side of the interface provides instructions for enabling automatic context help.

Luego la tabla de transactions que tiene todas las relaciones con las tablas anteriores con excepción de products, setadas en su creación:

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

- MySQL Model**: EER Diagram tab is selected.
- Administration**: Schemas tab is selected, showing the **sprint4** database.
- Tables**: A tree view of tables: companies, credit_cards, products, transactions, users.
- Views**, **Stored Procedures**, **Functions**, and **SYS** are also listed under Administration.
- Query Editor**: Contains three queries:
 - Query 1: `CREATE TABLE transactions (id VARCHAR(200) PRIMARY KEY, card_id VARCHAR(100), business_id VARCHAR(100), timestamp DATETIME, amount DECIMAL(15, 2), declined TINYINT, product_id VARCHAR(50), user_id INT, lat DECIMAL(10, 6), longitude FLOAT, FOREIGN KEY (business_id) REFERENCES companies(company_id), FOREIGN KEY (card_id) REFERENCES credit_cards(card_id), FOREIGN KEY (user_id) REFERENCES users(user_id))`
 - Query 2: `LOAD DATA INFILE '/Users/barbara junqueira/Desktop/RecursosDataAnalyst/sprint4/transactions.csv' INTO TABLE transactions FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n'`
 - Query 3: `SELECT * FROM transactions`Results for Query 3 show 1169 rows.
- Result Grid**: Shows the results of the SELECT query.
- Object Info** and **Session** tabs are visible.
- Schema: sprint4** is selected.
- Transactions**: A table showing transaction details with 48 rows.
- Action Output**: A table showing the execution of three queries with their times and results.
- Context... Snippet**: A snippet of code related to automatic context help.

Pensando en establecer las conexiones de la tabla de hechos con la de dimensiones me he dado cuenta de que la tabla de transactions no podria unirse con la de productos porque aunque la de transactions tuviera los productos comprados, estaban estos en un "pack" que era el product_ids. entonces hice la creacion de una tabla llamada transactions_products con el objetivo de separar los products_ids(transactions_id en la tabla en question) en product_id individual que se podría conectar con la tabla de productos. Como se trata de una tabla de unión entre la tabla de hechos y también la de productos, todas las relaciones en ambos lados se dan de muchos a muchos. por eso he creado un id para identificar de manera unica cada linea.

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

- Schemas:** The current schema is "sprint4".
- Tables:** The table "transactions_products" is selected.
- Script Editor:** The SQL code for creating the table and inserting data is visible. The table has columns: id_transaction_products (INT AUTO_INCREMENT PRIMARY KEY), transaction_id (INT), and product_id (VARCHAR(50)). The insert query uses a complex UNION ALL SELECT statement to split the comma-separated product_ids into individual rows.
- Result Grid:** The data grid shows 14 rows of data, mapping transaction IDs to various product IDs.
- Action Output:** The log shows three actions: the creation of the table, the insertion of 1457 rows, and a select query.
- Help:** A context help panel on the right provides information about automatic context help being disabled.

PD: al principio quise añadir una relación de many-to-many de transactions a transactions_products pero no he podido por el error 1822. como he leído que no tiene sentido crear esa relación en cuestión por esa misma cuestión de ser many-to-many dejé sin la relación como se puede ver en el pantallazo de diagrama.

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

- Action Output:** The log shows an attempt to alter the table "transactions" to add a foreign key constraint "transactions_ibfk_4" pointing to the "transaction_id" column in the "transactions_products" table. The operation failed with Error Code: 1822.
- Text:** A note in the text area says: "— alterando a tabla de transactions para crear o link com transactions_products — no he podido y no tiene sentido porque — quiero crear una conexión de many-to-many entonces dejé sin el enlace en el diagrama y esa tabla funciona como tabla de junción".

ROSER M'HA EXPLICAT COM FER-HO MILLOR AMB L'ID DE TRANSACCIÓ I HO HE FET:

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the tabs include Administration, Schemas, and the current tab, Sprint4*. Below the tabs, there's a toolbar with various icons. The main area displays a code editor with SQL queries and a results grid.

```

179 -- Tabla union entre products y transactions: TRANSACTIONS_PRODUCTS creacion, importacion y check
179 drop table transactions_products;
179
179 CREATE TABLE transactions_products (
179     transaction_id VARCHAR(50),
179     product_id INT
179 );
179
179
179 INSERT INTO transactions_products (transaction_id, product_id)
179
179 SELECT
179     t.id AS transaction_id,
179     CAST(SUBSTRING_INDEX(SUBSTRING_INDEX(t.product_ids, ',', n.n), ',', -1) AS UNSIGNED) AS product_id
179
179 FROM transactions t
179 JOIN (SELECT 1 AS n UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4) n
179 ON CHAR_LENGTH(t.product_ids) = CHAR_LENGTH(REPLACE(t.product_ids, ',', '')) + n.n - 1;
179
179
179 SELECT * FROM transactions_products;
179
    
```

The results grid shows the data inserted into the transactions_products table:

transaction_id	product_id
02C6201E-090A-1859-B4EE-...	1
02C6201E-090A-1859-B4EE-...	71
046442E5-470F-4D24-FD01...	43
046442E5-470F-4D24-FD01...	97
046442E5-470F-4D24-FD01...	47
063FB479-995C-6F6F-29F7...	5
063FB479-995C-6F6F-29F7...	31
063FB479-995C-6F6F-29F7...	67
063FB479-995C-6F6F-29F7...	47
066229C6-C3B9-4A83-7B8C...	79
066229C6-C3B9-4A83-7B8C...	85
066229C6-C3B9-4A83-7B8C...	89
06CD9A5A-9B42-0684-DDD0...	31
06CD9A5A-9B42-0684-DDD0...	43
n7a4frdr-11a5-7F47-6490-0...	23

At the bottom, the status bar indicates "Query Completed".

Comprendi que si yo tenia el product_ids estos se podrían repetir porque dos empresas/usuarios diferentes podrían hacer una compra igual. Pero el ID de la transacció de la tabla de transactions siempre sería único y esa es una buena manera de hacerlo mejor!

Luego quise mejorar la selección de elementos para “cuantos haya” y no solo 4 (porque pude hacer el check manual pero entendiendo que habrá archivos en los cuales no podré hacerlo quiero garantizar que todos los elementos de la columna product_ids entre dentro de la nueva tabla, entonces aplique una recursividad que lo que hace es coger la cantidad de elementos que tiene la columna y coletar los datos esa cantidad X de veces.

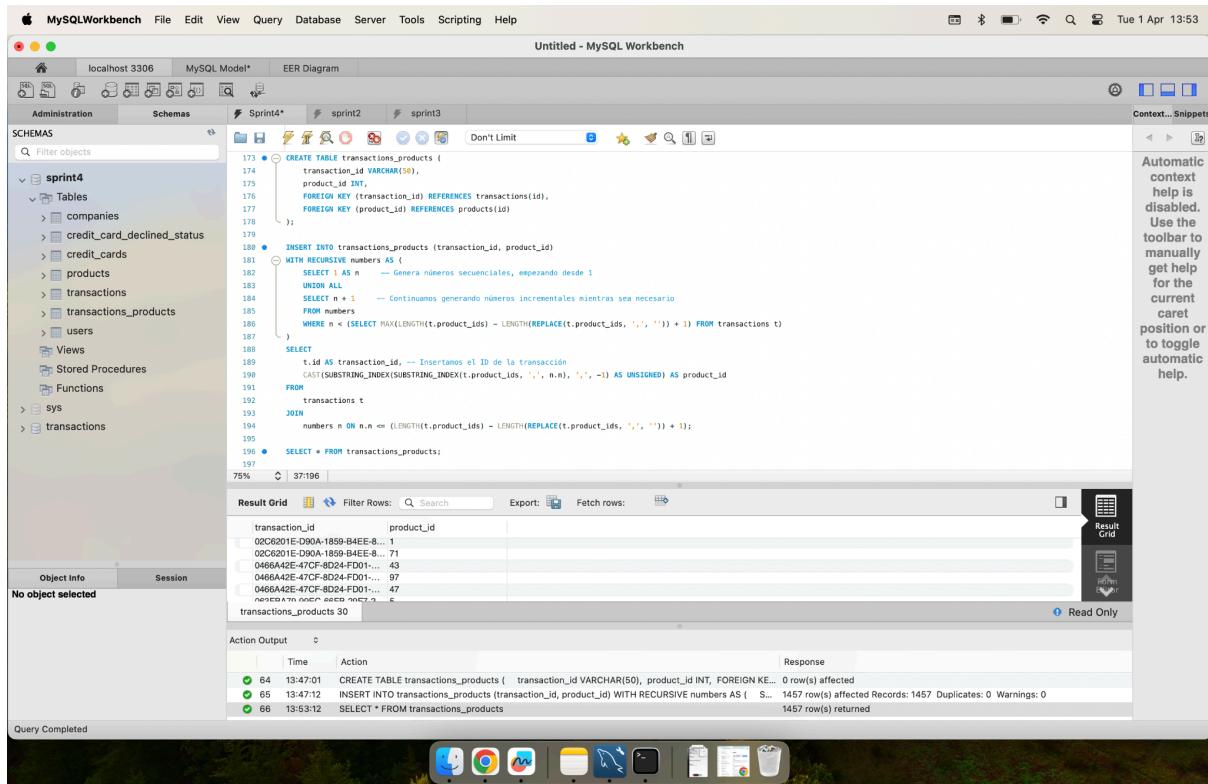
Explicacion

WHERE n < (SELECT MAX(LENGTH(t.product_ids) - LENGTH(REPLACE(t.product_ids, ',', '')) + 1) FROM transactions t)

1. LENGTH(t.product_ids): Calcula la longitud total de la cadena product_ids. Por ejemplo, si tienes product_ids = '1,2,3,5,9,12', esto devolvería 11 (porque hay 11 caracteres en total, incluyendo las comas).
2. REPLACE(t.product_ids, ',', ''): Elimina todas las comas en la cadena, dejando solo los números. Usando el mismo ejemplo de product_ids = '1,2,3,5,9,12', después de aplicar REPLACE, obtienes '1235912' (longitud de 7).
3. LENGTH(t.product_ids) - LENGTH(REPLACE(t.product_ids, ',', '')): Esto da como resultado el número de comas en la cadena. En nuestro ejemplo, hay 5 comas, por lo que el resultado de la resta es 5.
4. MAX es usado para calcular el número máximo de productos en una transacción.
5. + 1: Finalmente, sumamos 1, porque el número de elementos es una coma menos que el número de valores. En nuestro caso, como hay 5 comas, sumando 1 nos da 6, que es el número total de elementos separados por comas en la lista.
6. La condición WHERE n < (SELECT LENGTH(...) + 1) asegura que la recursión se detenga cuando el número n llegue a más de los valores que hay en la lista de product_ids. Si la cadena tiene 6 elementos, solo se generarán números hasta el 6, y no más.

CAST(SUBSTRING_INDEX(SUBSTRING_INDEX(t.product_ids, ',', n.n), ',', -1) AS UNSIGNED) AS product_id

La columna de product_id será rellenada por la columna de product_ids pero separando cada una que este separada por virgula hasta el enésimo valor. El -1 es para coger la última entrada de valor (vuelve la casilla anterior). El CAST convierte la subcadena extraída en un número entero sin signo (UNSIGNED). Si fueran números negativos usaría el SIGNED para mantener la señal



The screenshot shows the MySQL Workbench interface with the EER Diagram tab selected. A query window displays the creation of the `transactions_products` table:

```

CREATE TABLE transactions_products (
    transaction_id VARCHAR(50),
    product_id INT,
    FOREIGN KEY (transaction_id) REFERENCES transactions(id),
    FOREIGN KEY (product_id) REFERENCES products(id)
);

INSERT INTO transactions_products (transaction_id, product_id)
WITH RECURSIVE numbers AS (
    SELECT 1 AS n  -- Genera números secuenciales, empezando desde 1
    UNION ALL
    SELECT n + 1  -- Continuamos generando números incrementales mientras sea necesario
    FROM numbers
    WHERE n < (SELECT MAX(LENGTH(t.product_ids)) - LENGTH(REPLACE(t.product_ids, ',', '')) + 1) FROM transactions t
)
SELECT
    t.id AS transaction_id, -- Insertamos el ID de la transacción
    CAST(SUBSTRING_INDEX(SUBSTRING_INDEX(t.product_ids, ',', n), ',', '-1') AS UNSIGNED) AS product_id
FROM
    transactions t
JOIN
    numbers n ON n.n <= (LENGTH(t.product_ids) - LENGTH(REPLACE(t.product_ids, ',', '')) + 1);
SELECT * FROM transactions_products;

```

The Result Grid shows the generated data:

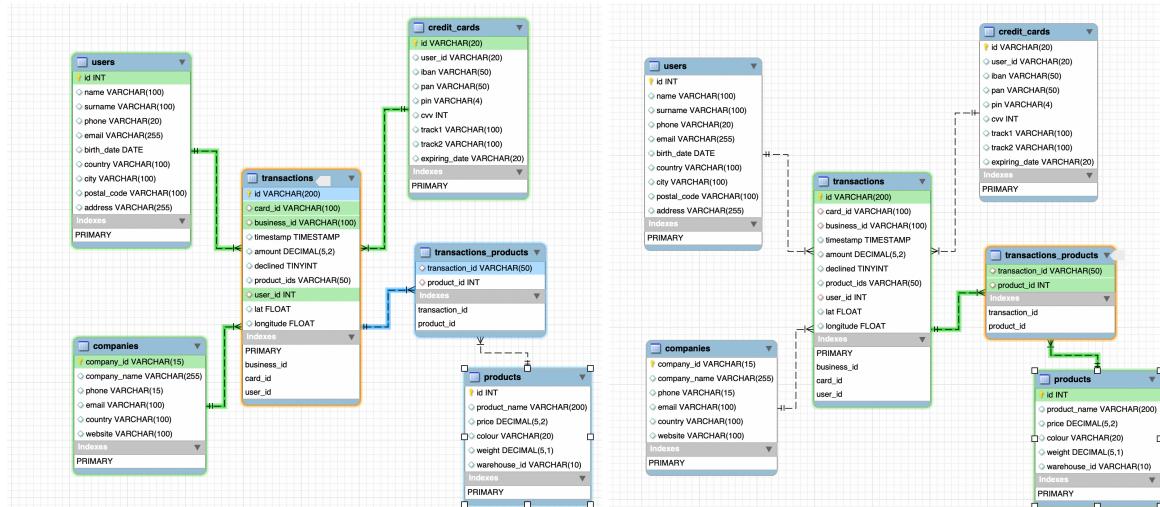
transaction_id	product_id
02C6201E-090A-1859-B4EE-8...	1
02C6201E-090A-1859-B4EE-8...	71
046842E-47CF-BD24-FD01...	43
046842E-47CF-BD24-FD01...	97
046842E-47CF-BD24-FD01...	47

The Action Output pane shows the execution log:

- CREATE TABLE transactions_products (transaction_id VARCHAR(50), product_id INT, FOREIGN KEY (transaction_id) REFERENCES transactions(id), FOREIGN KEY (product_id) REFERENCES products(id)) affected: 0 rows(s)
- INSERT INTO transactions_products (transaction_id, product_id) WITH RECURSIVE numbers AS (SELECT 1 AS n -- Genera números secuenciales, empezando desde 1 UNION ALL SELECT n + 1 -- Continuamos generando números incrementales mientras sea necesario FROM numbers WHERE n < (SELECT MAX(LENGTH(t.product_ids)) - LENGTH(REPLACE(t.product_ids, ',', '')) + 1) FROM transactions t) SELECT t.id AS transaction_id, -- Insertamos el ID de la transacción CAST(SUBSTRING_INDEX(SUBSTRING_INDEX(t.product_ids, ',', n), ',', '-1') AS UNSIGNED) AS product_id FROM transactions t JOIN numbers n ON n.n <= (LENGTH(t.product_ids) - LENGTH(REPLACE(t.product_ids, ',', '')) + 1); affected: 1457 rows(s) affected Records: 1457 Duplicates: 0 Warnings: 0
- SELECT * FROM transactions_products affected: 1457 row(s) returned

Entonces ahora tiene sentido hacer la relación de

1 a muchos de `transactions.id` a `transactions_products.transaction_id` y también
muchos a 1 de `transactions_products.product_id` a `products.id` 😊😊😊



- Exercici 1

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

✓ ACTUALIZADO

```
209  -- Nivell 1 Exercici 1
210  -- Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.
211  SELECT u.name AS userName, u.surname AS userSurname, (SELECT SUM(t.amount) FROM transactions t WHERE t.user_id = u.id) AS totalAmount
212  FROM users u
213  WHERE u.id IN (SELECT t.user_id
214    FROM transactions t
215    GROUP BY t.user_id
216    HAVING COUNT(t.id) > 30
217 );
75%  18:204
```

Result Grid Filter Rows: Search Export:

userName	userSurname	totalAmou...
Lynn	Riddle	11451.57
Ocean	Nelson	13052.24
Hedwig	Gilbert	18351.30
Kenyon	Hartman	12011.56

Action Output

Time	Action	Response
11 10:25:24	SELECT u.name AS userName, u.s...	4 row(s) returned

- Exercici 2

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

✓ ACTUALIZADO:

```
193  -- Nivell 1 Exercici 2
194  -- Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.
195  -- pre consulta para entender si seria un avg alto o bajo, y se puede calcular manualmente. tendria que ser (364,61 + 42,82) /2 = 203,715
196  SELECT *
197  FROM credit_cards cc
198  JOIN transactions t ON t.card_id = cc.id
199  JOIN companies c ON t.business_id = c.company_id
200  WHERE c.company_name = "Donec Ltd";
201
202  -- query en question:
203  SELECT ROUND(AVG(amount), 2) as avgAmount, company_name as companyName, card_id as iban
204  FROM transactions t
205  JOIN companies c ON t.business_id = c.company_id
206  WHERE c.company_name = "Donec Ltd"
207  GROUP BY iban, companyName;
208
```

28:207

Result Grid Filter Rows: Search Export:

avgAmount	companyName	iban
203.72	Donec Ltd	CcU-2973

Action Output

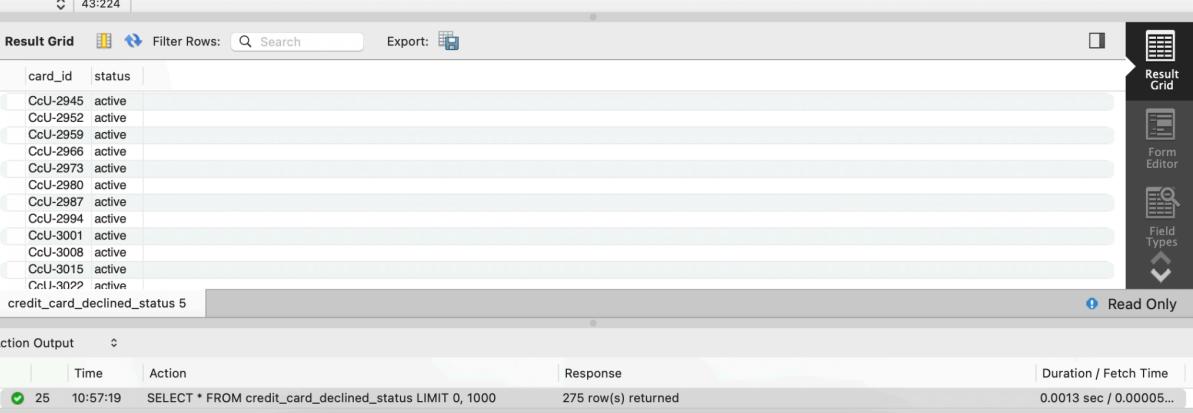
Time	Action	Response
21 10:54:49	SELECT ROUND(AVG(amount), 2) as avgAmount, company_name a...	1 row(s) returned

Nivell 2

Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions van ser declinades i genera la següent consulta:



```
209  -- Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions van ser declinades
210  ● CREATE TABLE credit_card_declined_status AS
211      WITH lastThreeTransactions AS (
212          SELECT t.card_id, t.declined,ROW_NUMBER() OVER (PARTITION BY t.card_id ORDER BY t.timestamp DESC) AS numRow
213      )
214      SELECT card_id,
215          CASE
216              WHEN SUM(declined) = 3 THEN 'declined'
217              ELSE 'active'
218          END AS status
219      FROM lastThreeTransactions
220      WHERE numRow <= 3
221      GROUP BY card_id;
222
223
224  ● SELECT * FROM credit_card_declined_status;
225
```



The screenshot shows a database query interface with the following details:

- Code Area:** Displays the SQL code for creating the `credit_card_declined_status` table.
- Result Grid:** Shows the data returned by the query. The table has two columns: `card_id` and `status`. The data includes:

card_id	status
CcU-2945	active
CcU-2952	active
CcU-2959	active
CcU-2966	active
CcU-2973	active
CcU-2980	active
CcU-2987	active
CcU-2994	active
CcU-3001	active
CcU-3008	active
CcU-3015	active
CcU-3022	active
- Action Output:** Shows the execution details of the query:

Action	Time	Action	Response	Duration / Fetch Time
25	10:57:19	SELECT * FROM credit_card_declined_status LIMIT 0, 1000	275 row(s) returned	0.0013 sec / 0.00005...
- Right Panel:** A sidebar with icons for Result Grid, Form Editor, and Field Types.
- Status Bar:** Shows "Read Only".

Exercici 1

Quantes targetes estan actives?

```
256 • | SELECT count(status) as tarjetasActivas
257 | FROM credit_card_declined_status
258 | WHERE status = 'active';
75% | 13:241 | Result Grid | Filter Rows: | Search | Export: |
```

tarjetasActivas
275

Result 21

```
Action Output | Time | Action | Response
38 11:59:38 | SELECT count(status) as tarjetasActivas FROM credit_card_declined_status WHERE status = 'active' | 1 row(s) returned
```

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:

La tabla ya habia sido creada a principio:

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, Help, and a system tray with icons for battery, signal, and search. The title bar says "MySQL Workbench".

The left sidebar displays the "SCHEMAS" tree, which includes a "Schemas" section and a "sprint4" database containing tables like companies, credit_card_declined_status, credit_cards, products, transactions, users, Views, Stored Procedures, Functions, sys, and transactions.

The main area has tabs for "Query 2", "sprint2", "sprint3*", and "Sprint4*". The "Sprint4*" tab is active, showing a complex SQL script. The script creates a table "transactions_products" and inserts data into it using a recursive common table expression (CTE) to generate sequential numbers for each transaction. It also performs a self-join on the transactions table and uses a WHERE clause with LENGTH and REPLACE functions to handle varying lengths of product IDs.

The "Result Grid" tab shows the results of the last query, displaying columns for transaction_id and product_id. The results are as follows:

transaction_id	product_id
00000000-0000-0000-0000-000000000000	1
00000001-0000-0000-0000-000000000000	2
00000002-0000-0000-0000-000000000000	3
00000003-0000-0000-0000-000000000000	4
00000004-0000-0000-0000-000000000000	5
00000005-0000-0000-0000-000000000000	6
00000006-0000-0000-0000-000000000000	7
00000007-0000-0000-0000-000000000000	8
00000008-0000-0000-0000-000000000000	9
00000009-0000-0000-0000-000000000000	10
0000000A-0000-0000-0000-000000000000	11
0000000B-0000-0000-0000-000000000000	12
0000000C-0000-0000-0000-000000000000	13
0000000D-0000-0000-0000-000000000000	14
0000000E-0000-0000-0000-000000000000	15

The bottom status bar indicates "Query Completed".

Exercici 1

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

```
268 -- Necesitem conèixer el nombre de vegades que s'ha venut cada producte.
269 •   SELECT product_id as producte, count(product_id) as nombreVegadesVenut
270   FROM transactions_products tp
271   JOIN transactions t ON tp.transaction_id = t.id
272   WHERE declined = 0
273   GROUP BY producte
274   ORDER BY producte;
275
```

25:266

Result Grid Filter Rows: Search Export:

producte	nombreVegadesVenut
1	61
2	56
3	43
5	42
7	44
11	40
13	51
17	54

Result 9

Action Output

Time	Action	Response	Duration / Filtre Time
16 10:42:39	SELECT product_id as producte, count(product_id) as nombreVegadesVenut FROM transactions_products tp JOIN transactions t ON tp.transaction_id = t.id WHERE declined = 0 GROUP BY producte;	26 row(s) returned	0.013 sec / 0.000030...

Result Grid Form Editor Read Only