

# SPRINT4-CREACIÓN DE BASE DE DATOS

## JOSÉ JESÚS RODRIGUEZ V.

### 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:

The screenshot shows the SQL Developer interface. On the left, the 'SCHEMAS' pane displays a tree view with 'script4\_transactions' selected. The main editor shows SQL code for creating the database and the 'companies' table. The 'Output' pane at the bottom shows the execution results of the script.

```
1 -- estructura datos script4_transactions
2
3 -- Creamos la base de datos
4 • CREATE DATABASE IF NOT EXISTS script4_transactions;
5 • USE script4_transactions;
6
7
8 -- Creamos la tabla company
9 • CREATE TABLE IF NOT EXISTS companies (
10     company_id VARCHAR(15) PRIMARY KEY,
11     company_name VARCHAR(255),
12     phone VARCHAR(15),
13     email VARCHAR(100),
14     country VARCHAR(100),
15     website VARCHAR(255)
16 );
```

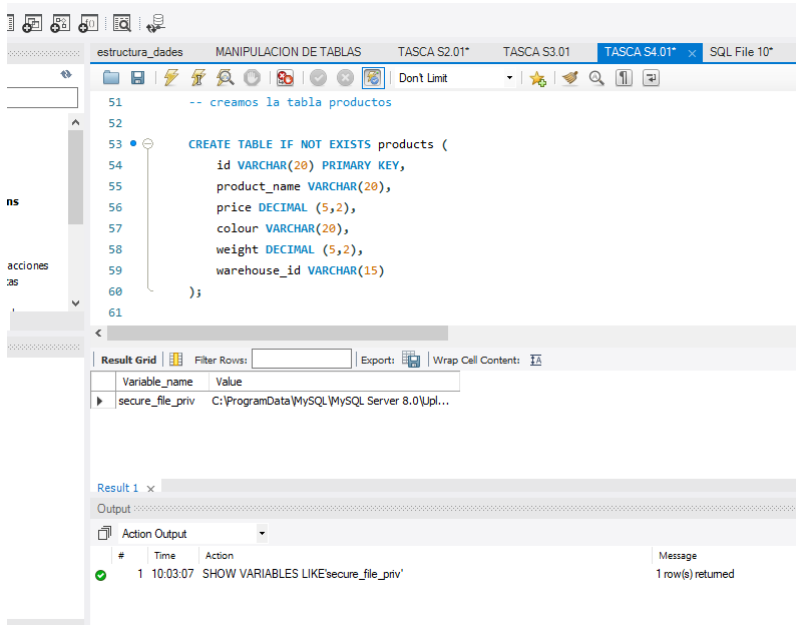
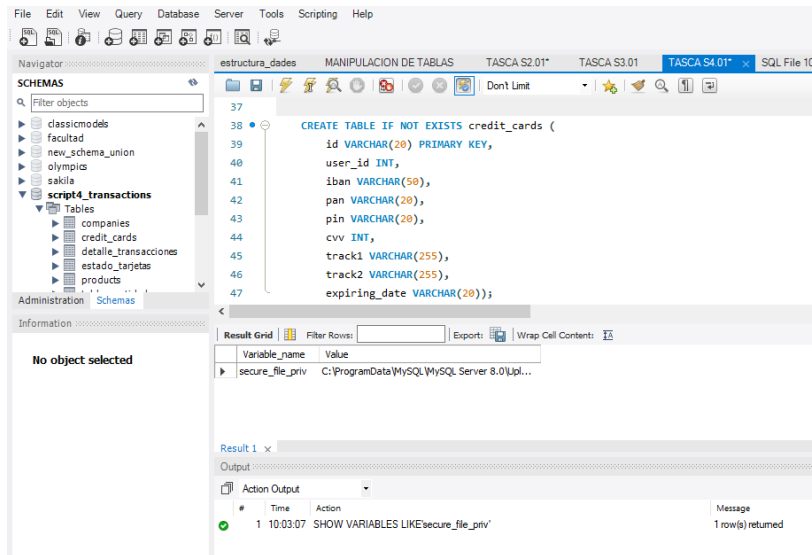
#	Time	Action	Message
2	12:59:39	USE script4_transactions	0 row(s) affected
3	12:59:39	CREATE TABLE IF NOT EXISTS companies ( company_id VARCHAR(15) PRIMARY KEY, ...	0 row(s) affected
4	12:59:40	CREATE TABLE IF NOT EXISTS users ( id INT PRIMARY KEY, name VARCHAR(100), ...	0 row(s) affected
5	12:59:40	CREATE TABLE IF NOT EXISTS credit_cards ( id VARCHAR(20) PRIMARY KEY, ...	0 row(s) affected
6	12:59:40	CREATE TABLE IF NOT EXISTS products ( id VARCHAR(20) PRIMARY KEY, pr...	0 row(s) affected
7	12:59:41	CREATE TABLE IF NOT EXISTS transaction ( id VARCHAR(255) PRIMARY KEY, ...	0 row(s) affected

The screenshot shows the SQL Developer interface with the 'estructura\_datos' script open. The main editor displays the SQL code for creating the 'users' table. The 'Output' pane is empty.

```
17
18
19 -- Creamos la tabla users
20 -- id,name,surname,phone,email,birth_date,country,city,postal_code,address
21
22 • CREATE TABLE IF NOT EXISTS users (
23     id INT PRIMARY KEY,
24     name VARCHAR(100),
25     surname VARCHAR(100),
26     phone VARCHAR(150),
27     email VARCHAR(150),
28     birth_date VARCHAR(100),
29     country VARCHAR(150),
30     city VARCHAR(150),
31     postal_code VARCHAR(100),
32     address VARCHAR(255)
33 );
34
```

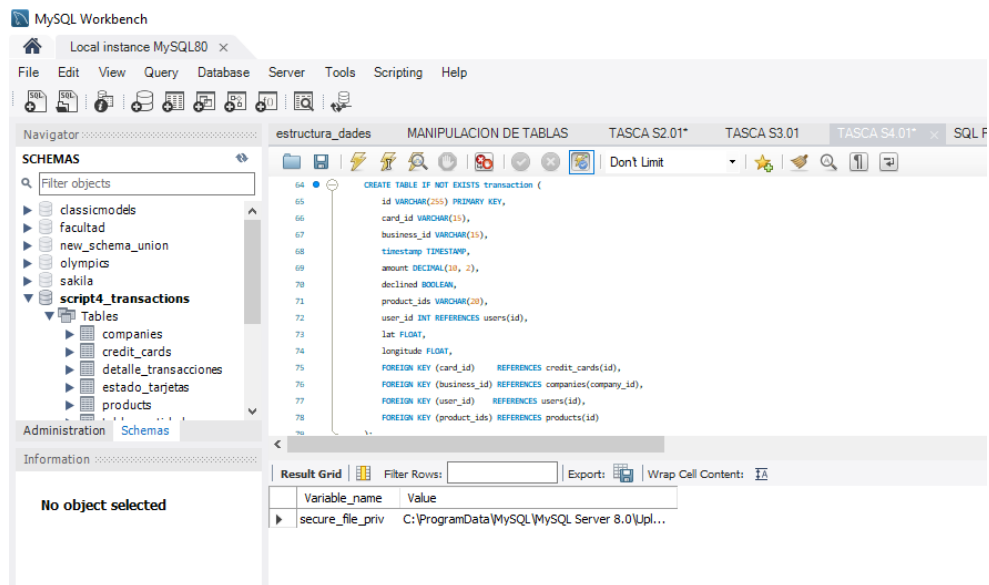
# SPRINT4-CREACIÓN DE BASE DE DATOS

## JOSÉ JESÚS RODRIGUEZ V.

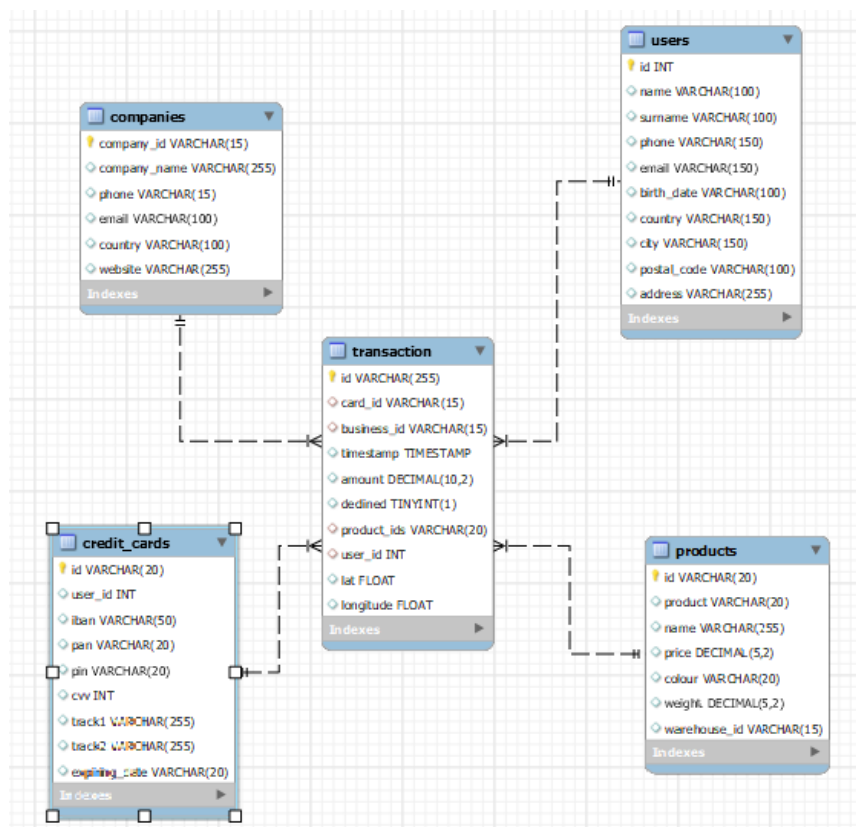


## SPRINT4-CREACIÓN DE BASE DE DATOS

### JOSÉ JESÚS RODRIGUEZ V.



El diagrama resultante es:



# SPRINT4-CREACIÓN DE BASE DE DATOS

## JOSÉ JESÚS RODRIGUEZ V.

Realizamos la carga de datos:

Tabla companies:

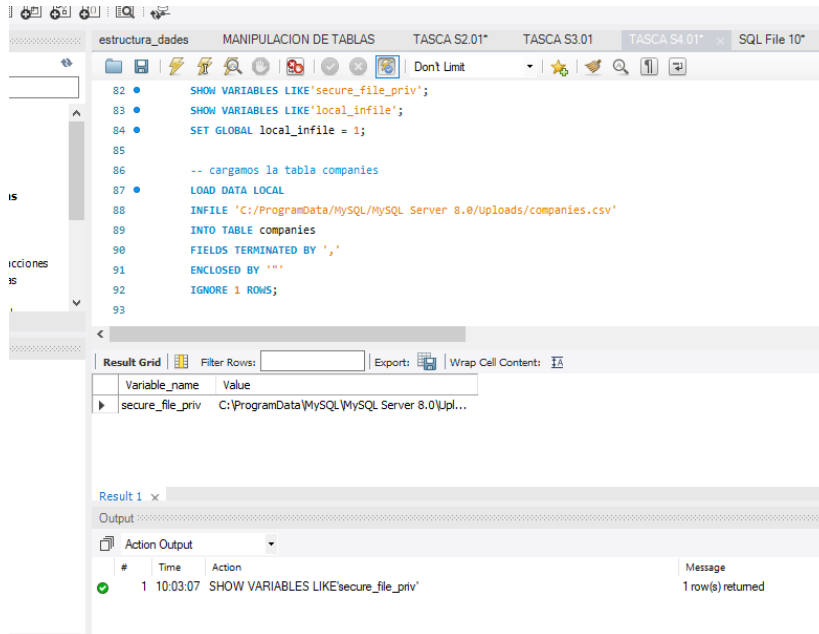
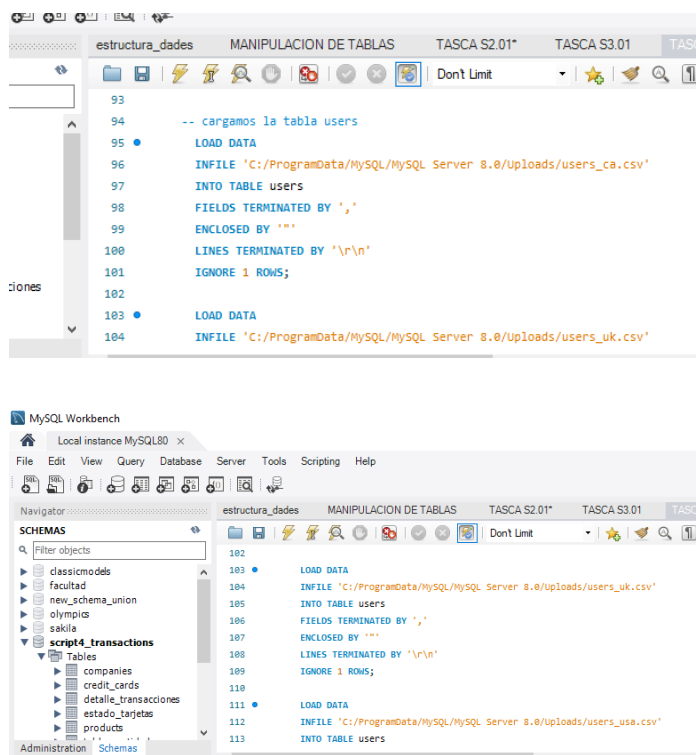
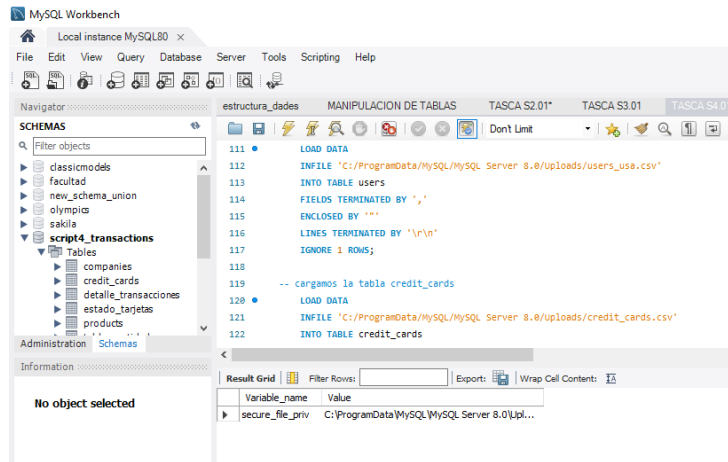


Tabla Users:

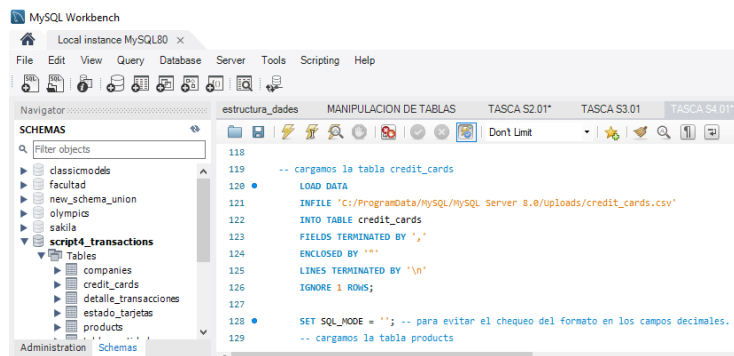


# SPRINT4-CREACIÓN DE BASE DE DATOS

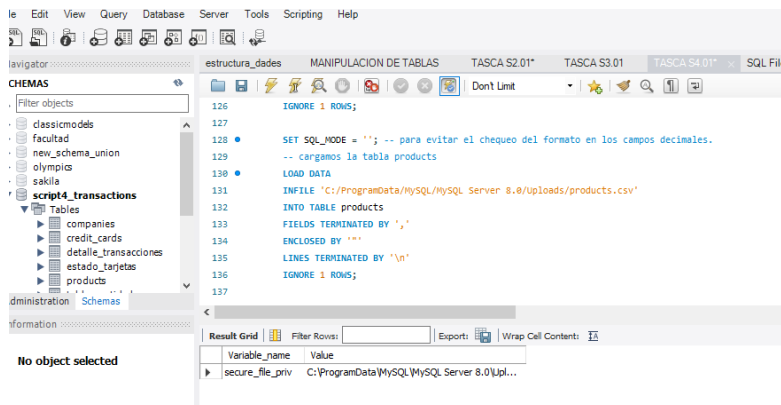
## JOSÉ JESÚS RODRIGUEZ V.



### Tabla credit\_cards



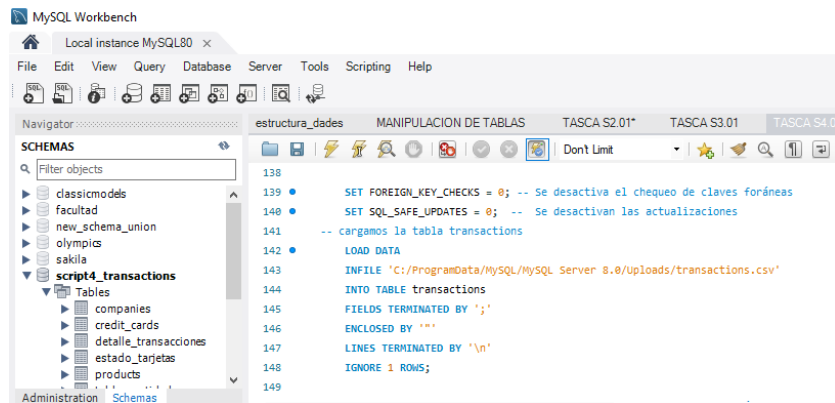
### Tabla products:



# SPRINT4-CREACIÓN DE BASE DE DATOS

## JOSÉ JESÚS RODRIGUEZ V.

### Tabla transactions



## SPRINT4-CREACIÓN DE BASE DE DATOS JOSÉ JESÚS RODRIGUEZ V.

### - Exercici 1

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

```
161 • SELECT u.name as nombre, u.surname as apellido
162 FROM users u
163 WHERE u.id IN (
164     SELECT user_id
165     FROM transactions t
166     WHERE t.declined = 0
167     GROUP BY user_id
168     HAVING COUNT(t.user_id) > 30 -- se usa la cláusula HAVING para revisar la condición del COUNT
169 );
170
```

Result Grid

nombre	apellido
Lynn	Riddle
Ocean	Nelson
Hedwig	Gilbert

users 15 x

Output

Action Output

#	Time	Action	Message
1	12:55:16	SELECT u.name as nombre, u.surname as apellido FROM users u WHERE u.id IN ( SE...	4 row(s) returned
2	12:55:59	SELECT u.name as nombre, u.surname as apellido FROM users u WHERE u.id IN ( SE...	3 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.

Este ejercicio fue realizado con sub-consulta. También puede ser hecho con JOIN

estructura\_datos MANIPULACION DE TABLAS TASCA S2.01\* TASCA S3.01 TASCA S4.01\* SQL File 10\*

```
175
176
177 • SELECT ROUND(AVG(t.amount),2) -- Se calcula la media del amount
178 FROM transactions t
179 JOIN credit_cards cc
180 ON t.card_id = cc.id
181 JOIN companies c
182 ON t.business_id = c.company_id
183 WHERE t.declined = 0 AND
184 c.company_name = 'Donec Ltd' -- Se verifica la condición del cálculo en la empresa Donec Ltd.
185 GROUP BY cc.iban;
186
```

Result Grid

ROUND(AVG(t.amount),2)
42.82

Result 5 x

Output

Action Output

#	Time	Action	Message
4	11:22:21	SELECT u.name as nombre, u.surname as apellido FROM users u WHERE u.id IN ( ...	3 row(s) returned
5	11:28:51	SELECT ROUND(AVG(t.amount),2) -- Se calcula la media del amount FROM transacti...	Error Code: 1054. Unknown column 'c.company_name' in 'w'
6	11:29:16	SELECT ROUND(AVG(t.amount),2) -- Se calcula la media del amount FROM transacti...	1 row(s) returned

Table: companies

Columns:	
company_id	varchar(15)
company_name	PK
phone	varchar(255)
email	varchar(15)
country	varchar(100)
website	varchar(100)

# SPRINT4-CREACIÓN DE BASE DE DATOS

## JOSÉ JESÚS RODRIGUEZ V.

### 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:

#### Exercici 1

Quantes targetes estan actives?

Para este ejercicio utilizaremos la función ROW\_NUMBER. Esta es una función de ventana que asigna un número único, ordenado a cada fila dentro de una partición (punto o elemento de ruptura). Es usada para numerar filas en un conjunto de resultados, especialmente en combinación con consultas que requieren filas específicas.

En este caso usamos esta función como se muestra a continuación, enumerando las filas de acuerdo a las columnas card\_id y declined ordenando decuerdo a la fecha (timestamp) desde el más reciente.

ROW\_NUMBER() OVER (PARTITION BY card\_id, declined ORDER BY timestamp DESC)

```
1  -- Crear la tabla de estado de tarjetas (estado_tarjetas) e insertar los datos
2  • CREATE TABLE IF NOT EXISTS estado_tarjetas AS
3  SELECT *
4  FROM
5  (SELECT
6      card_id tarjeta,
7      declined estado,
8      ROW_NUMBER() OVER (PARTITION BY card_id, declined ORDER BY timestamp DESC) num_filas
9  FROM transactions) t;
10
11  -- Indicar cuántas tarjetas están activas
12  • SELECT estado, COUNT(tarjeta),
13  CASE
14      WHEN estado = 1 and num_filas = 3 THEN "Estas Tarjetas están inactivaa"
15      ELSE "Todas estas tarjetas están activas"
16  END AS texto
17  FROM estado_tarjetas
18  GROUP BY estado, texto;
```

estado	COUNT(tarjeta)	texto
0	500	Todas estas tarjetas están activas
1	87	Todas estas tarjetas están activas

#	Time	Action	Message
1	17:01:13	CREATE TABLE IF NOT EXISTS estado_tarjetas AS SELECT * FROM (SELECT	car... 587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
2	17:01:26	SELECT estado, COUNT(tarjeta), CASE WHEN estado = 1 and num_filas = 3 THEN "Estas	2 row(s) returned



## SPRINT4-CREACIÓN DE BASE DE DATOS

### JOSÉ JESÚS RODRIGUEZ V.

El ejecutar el script vemos que el numero total de registros de la tabla creada coincide con el número de activas, por tanto no existen tarjetas inactivas entre los datos proporcionados.

## Nivell 3

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

### Exercici 1

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

#### Paso 1: Crear una Tabla Auxiliar de Números

```
-- Insertar números secuenciales
• INSERT INTO tabla_cantidad (cantidad_productos_separados_coma)
  VALUES (1), (2), (3), (4), (5), (6)
;
```

#### Paso 2: Crear la Tabla Secundaria

```
-- Creamos la tabla detalle_transacciones para almacenar los valores divididos:
• CREATE TABLE detalle_transacciones (
  id INT AUTO_INCREMENT PRIMARY KEY,
  transaccion_id VARCHAR(255),
  producto_id VARCHAR(20)
);
```

## SPRINT4-CREACIÓN DE BASE DE DATOS JOSÉ JESÚS RODRIGUEZ V.

### Paso 3: Insertar los Valores Divididos:

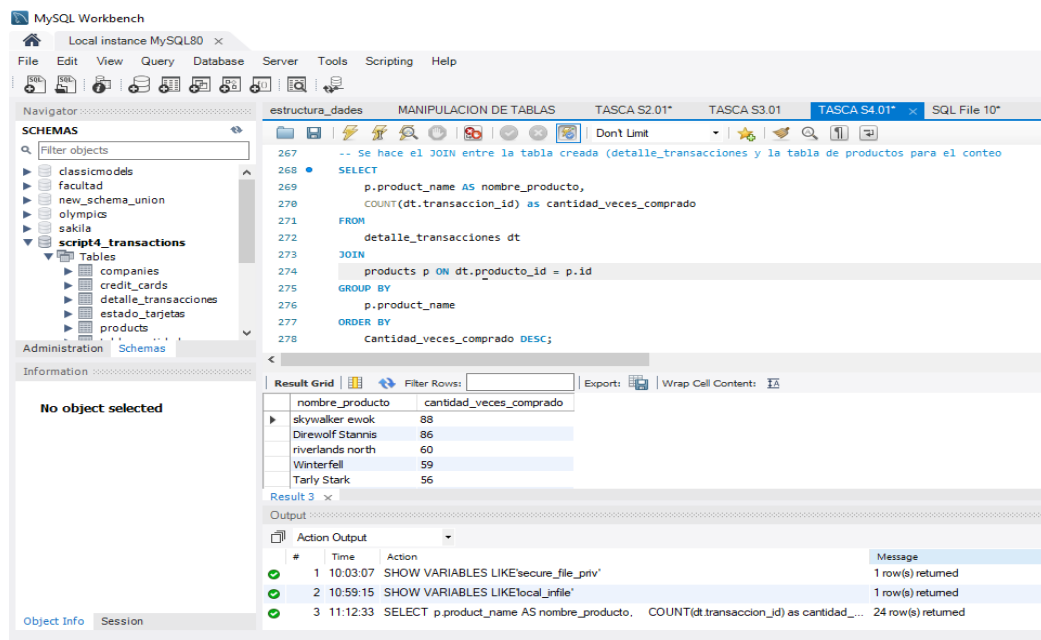
Insertar en detalle\_transacciones los codigos de productos de forma separada. Usamos una consulta basada en JOIN con la tabla tabla\_cantidad para dividir los valores separados por comas e insertarlos en la tabla detalle\_transacciones que usará como tabla secundaria.

```
-- Insertar en detalle_transacciones los codigos de productos de forma separada
-- Usamos una consulta basada en JOIN con la tabla tabla_cantidad para dividir
-- los valores separados por comas e insertarlos en la tabla detalle_transacciones
-- que usará como tabl asecundaria

INSERT INTO detalle_transacciones (transaccion_id, producto_id)
SELECT
    t.id AS transaccion_id,
    CAST(TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(t.product_ids, ',', tc.cantidad_productos_separados_coma), ',',
-1)) AS UNSIGNED) AS producto_id
FROM
    transactions t
JOIN
    tabla_cantidad tc ON tc.cantidad_productos_separados_coma <=
                        (1 + CHAR_LENGTH(t.product_ids) - CHAR_LENGTH(REPLACE(t.product_ids, ',', '')))
WHERE
    TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(t.product_ids, ',', tc.cantidad_productos_separados_coma), ',', -1)) <> ''
    AND t.declined = 0;
```

### Paso 4: Contar las veces que se ha comprado efectivamente cada producto:

Se hace el JOIN entre la tabla creada (detalle\_transacciones y la tabla de productos para el conteo



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'script4\_transactions' selected. The main editor shows a SQL query (lines 267-278) that joins 'detalle\_transacciones' and 'products' to count purchases. The 'Result Grid' at the bottom shows the output of this query, listing product names and their purchase counts. The 'Output' pane at the bottom shows the execution log with three messages: 'SHOW VARIABLES LIKE 'secure\_file\_priv'', 'SHOW VARIABLES LIKE 'local\_infile'', and the final 'SELECT' query which returned 24 rows.

nombre_producto	cantidad_veces_comprado
skywalker ewok	88
Direwolf Stannis	86
riverlands north	60
Winterfell	59
Tarly Stark	56

#	Time	Action	Message
1	10:03:07	SHOW VARIABLES LIKE 'secure_file_priv'	1 row(s) returned
2	10:59:15	SHOW VARIABLES LIKE 'local_infile'	1 row(s) returned
3	11:12:33	SELECT p.product_name AS nombre_producto, COUNT(dt.transaccion_id) as cantidad_...	24 row(s) returned