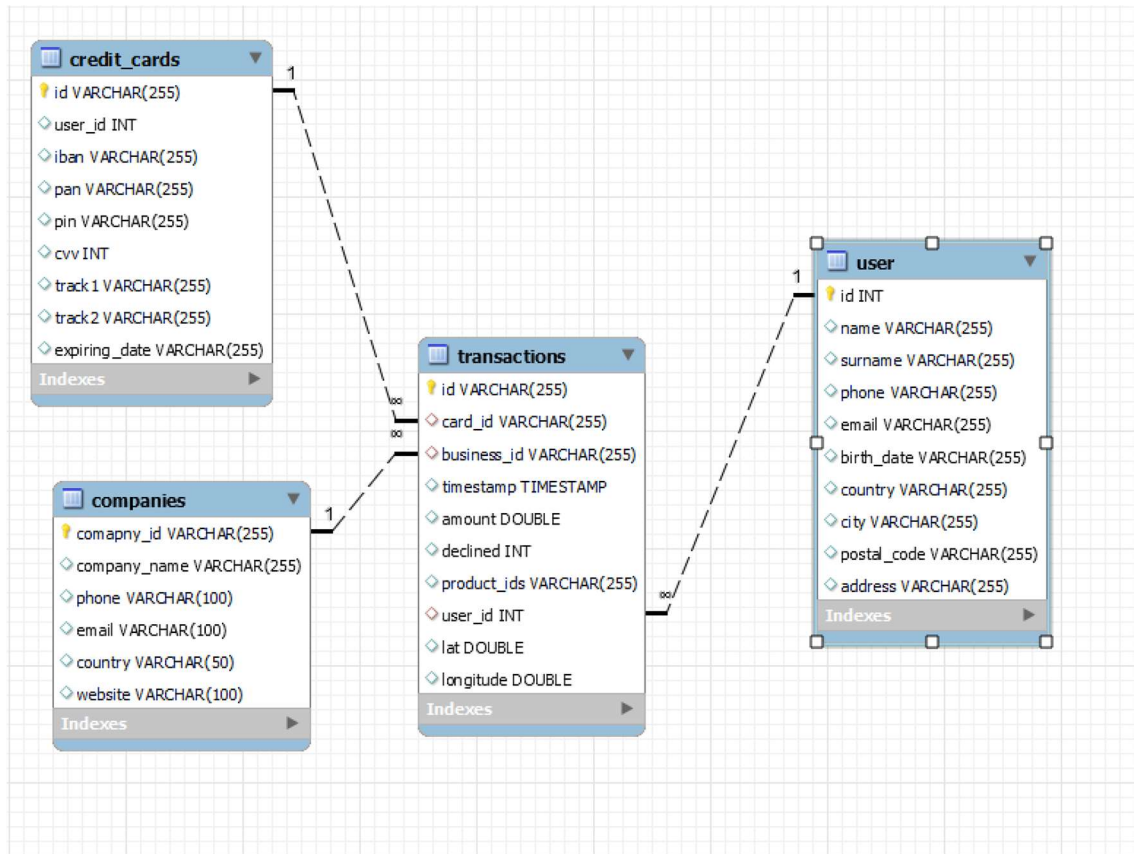
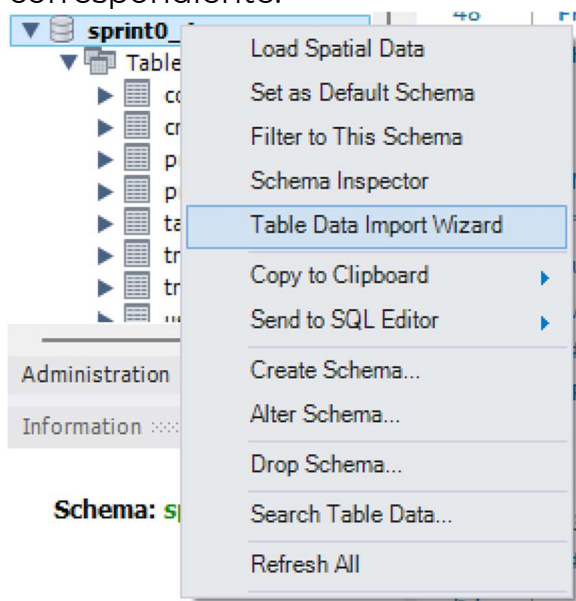


Nivell 1

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:



A través de la herramienta Wizard exportamos los archivos CSV correspondiente.



Sobre el schema, hacemos click derecho y seleccionamos table data import wizard

Table Data Import allows you to easily import CSV, JSON datafiles.
You can also create destination table on the fly.

File Path: C:\Users\Arkim\Downloads\transacciones_produtos.csv

Browse...

Luego especificamos la ruta de los archivos

Después seleccionamos la tabla o creamos una nueva

Select destination table and additional options.

☐ Use existing table: sprint0_4.companies

☒ Create new table: sprint0_4 transacciones_produtos

☐ Drop table if exists

Seleccionamos el tipo correspondiente a las columnas (en este caso es una prueba solo para realizar el procedimiento)

Detected file format: csv

Encoding: utf-8

Columns:

<input checked="" type="checkbox"/> Source Column	Field Type
<input checked="" type="checkbox"/> i»zid	text
<input checked="" type="checkbox"/> Suma de product_ids	int

i»zid	Suma de p...
02C6201E...	91
0466A42E...	187
063FBA79...	150
0668296C...	251
06CD9AA...	74

Por ultimo le damos a continuar y ejecutamos

The following tasks will now be performed. Please monitor the execution.

- ☐ Prepare Import
- ☐ Import data file

Click [Next >] to execute.

[Show Logs](#)

[← Back](#)

[Next >](#)

Cancel

- Exercici 1

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

Una vez exportado la base de datos con sus respectivas relaciones entre tablas ya podemos proceder a hacer consultas entre ellas. Esta es una consulta simple donde contamos las transacciones y solo mostramos aquellos que tengan mas de 30.

```
2 #Nivel 1 Ex 01
3 • SELECT * FROM user
4 WHERE id IN (SELECT user_id FROM transactions
5 GROUP BY user_id
6 HAVING count(transactions.id) >= 30
7 );
8
```

Result Grid									
Filter Rows:				Edit:		Export/Import:		Wrap Cell Content:	
id	name	surname	phone	email	birth_date	country	city	postal_code	address
92	Lynn	Riddle	1-387-985-4057	valie.aliqet@outlook.edu	Sep 21, 1984	United States	Bozeman	61871	P.O. Box 712, 7907 Est St.
267	Ocean	Nelson	079-481-2745	aenean@yahoo.com	Dec 26, 1991	Canada	Charlottetown	85X 3P4	Ap #732-8357 Pede, Rd.
272	Hedwig	Gilbert	064-204-8788	sem.eget@icloud.edu	Apr 16, 1991	Canada	Tuktoyaktuk	Q4C 3G7	P.O. Box 496, 5145 Sapien Road
275	Kenyon	Hartman	082-871-7248	convallis.ante.lectus@yahoo.com	Aug 3, 1982	Canada	Richmond	R8H 2K2	8564 Facilisi. St.

- Exercici 2

Mostra la mitjana de la suma de transaccions per IBAN de les targetes de crèdit en la companyia Donec Ltd. utilitzant almenys 2 taules.

En este caso he realizado dos consultas diferentes, una toma cada iban como único en cada transacción y hace la media contándolos en la división, esto significa que el mismo iban repetido dos veces dividiría entre dos. Por otro lado la otra consulta trata los mismo iban como mismos por lo que la división se hace entre el numero de iban diferentes.

Al final he optado por tratar cada iban como único aunque sean el mismo y enseñar la media:

```
9      #Ex02
10     #En este caso se suma las transacciones y se divide por el numero de iban aunque sean el mismo:
11 •    SELECT AVG(transactions.amount) as media_trans_iban_DonecLtd, iban
12     FROM transactions
13     JOIN credit_cards ON transactions.card_id = credit_cards.id
14     JOIN companies ON transactions.business_id = companies.comapny_id
15     WHERE companies.company_name= 'Donec Ltd'
16     GROUP BY iban;
17
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
media_trans_iban_DonecLtd	iban		
203.715	PT87806228135092429456346		


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:

En este caso lo que hemos hecho es crear una tabla con la columna id tarjeta y otra tabla haciendo una columna calculada en la que hacemos lo siguiente: coger por id de tarjeta los últimos tres timestamp y una vez cogidos mirar la columna declined, si esta columna suma "3" que significa que los últimos tres valores fueron declinados le asignamos "Desactivada" en la nueva columna "Estado_Tarjeta" si no suma tres le pondremos un "Activada"

```
28 • CREATE TABLE tarjetas_activadas AS
29 SELECT t.card_id,
30 CASE
31     WHEN (SELECT COUNT(*)
32           FROM (SELECT declined
33                 FROM transactions t2
34                 WHERE t2.card_id = t.card_id
35                 ORDER BY t2.timestamp DESC
36                 LIMIT 3) AS last_three
37           WHERE declined = 1) = 3 THEN "Desactivada" ELSE "Activada"
38     END AS Estado_Tarjeta
39 FROM transactions t
40 GROUP BY t.card_id;

42 #Comprobamos
43 • SELECT *
44 FROM tarjetas_activadas;
```

Result Grid |  Filter Rows: | Exp

card_id	Estado_Tarjeta
CcU-2938	Activada
CcU-2945	Activada
CcU-2952	Activada
CcU-2959	Activada
CcU-2966	Activada
CcU-2973	Activada
CcU-2980	Activada
CcU-2987	Activada

Exercici 1

Quantes targetes estan actives?

```
81      #ex 03
82 •    SELECT COUNT(CARD_ID) AS CANTIDAD_TARJETAS_ACTIVAS
83      FROM TARJETAS_ACTIVADAS
84      WHERE ACTIVADA = 1;
85
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	CANTIDAD_TARJETAS_ACTIVAS			
▶	275			

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 creación de la tabla la hemos hecho a través de Excel con power query al resultar más sencillo.

Para ello, cogemos una copia de la tabla de transacciones, borramos todas las columnas menos el ID y el products_ids. Después sobre la columna products_ids, la damos a la opción de dividir celda por delimitador “,” además de añadirle la opción en filas.

IMPORTANTE: Debido a debates y comentarios en clase he editado el archivo sql para hacer el mismo proceso en mysql.

Para ello haremos lo siguiente:

Creamos la tabla de enlace con ambas columnas (Id de la transacción e id de los productos)

```
##Primer Paso crear tabla de enlace:  
CREATE TABLE productos_transacciones_enlace (  
    transaction_id VARCHAR(255),  
    product_id VARCHAR(10)  
);
```

Una vez hecho tenemos que introducir los valores correspondientes en dicha tabla para ello usaremos la siguiente consulta:

```
INSERT INTO productos_transacciones_enlace (Transaction_ID, Product_ID) #Los valores seran introducidos  
SELECT id, SUBSTRING_INDEX(SUBSTRING_INDEX(Product_ids, ',', numbers.n), ',', -1) as Product_ID #Select  
FROM Transactions  
JOIN (  
    SELECT 1 as n UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 #unimos  
    ) numbers ON CHAR_LENGTH(Product_ids) - CHAR_LENGTH(REPLACE(Product_ids, ',', '')) >= numbers.n - 1;
```

La función INSERT INTO como ya hemos usado en anteriores ejercicios es para introducir los valores e indicar donde

En el SELECT indicamos que seleccionamos el ID (de transactions) y usamos substring_index para extraer trozos de cadenas de texto con un delimitador específico, al usarla dos veces nos aseguramos de extraer todos los valores

Como vemos en el SELECT usamos una tabla “numbers” que creamos en el JOIN donde estamos creando una columna “n” con 5 valores posibles

Luego restamos el valor total de la cadena menos el mismo valor sin tener en cuenta las comas

Prácticamente tenemos que crear una tabla de enlace en la que cojamos el id de la transacción y el id de los productos, pero duplicando la transacción por cada producto para poder tener un valor en cada registro.

A modo de comprobación añado lo siguiente para comprobar que ambos métodos me dan el mismo resultado de registros:

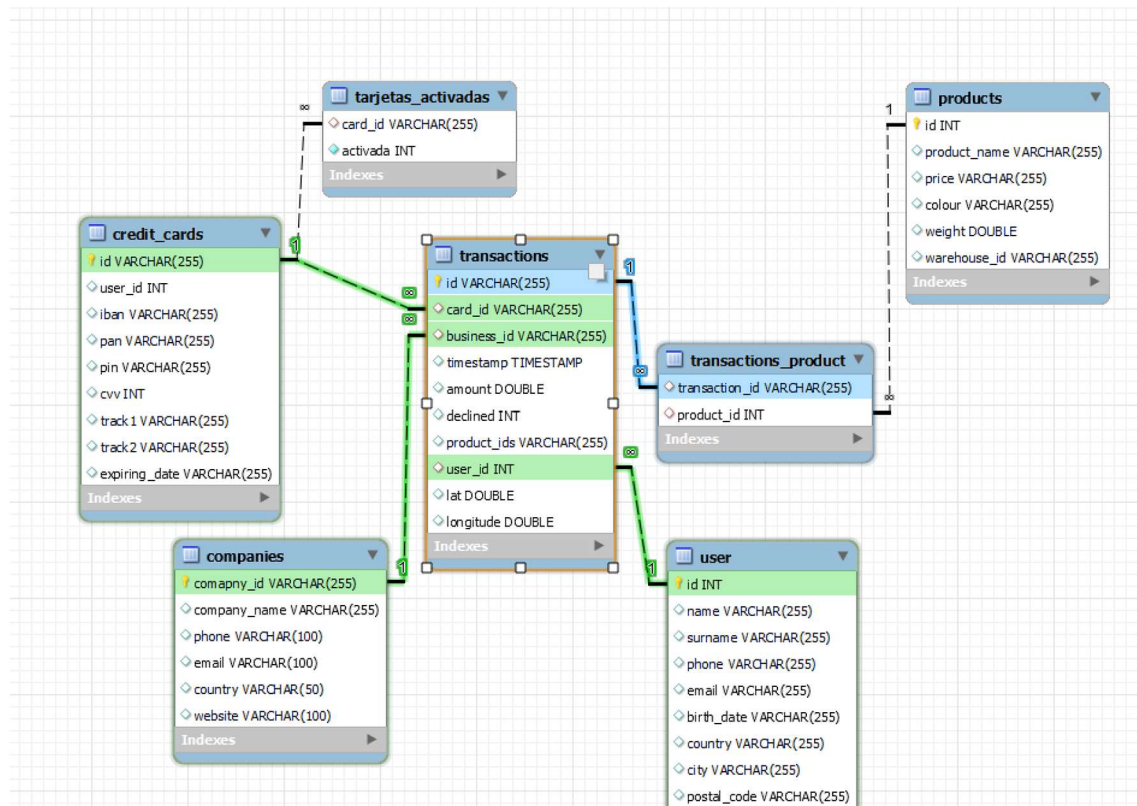
```
#Comprobamos la tabla
SELECT *
FROM productos_transacciones_enlace;
##AÑADO COMO INTERES QUE TANTO LA TABLA CREADA CON POWER QUERY COMO ESTA TIENEN LOS MISMOS RESULTADOS:
SELECT *
FROM transactions_product;
```

“productos_transacciones_enlace” es la tabla creada con SQL
“transactions_product” es la tabla creada con power query.

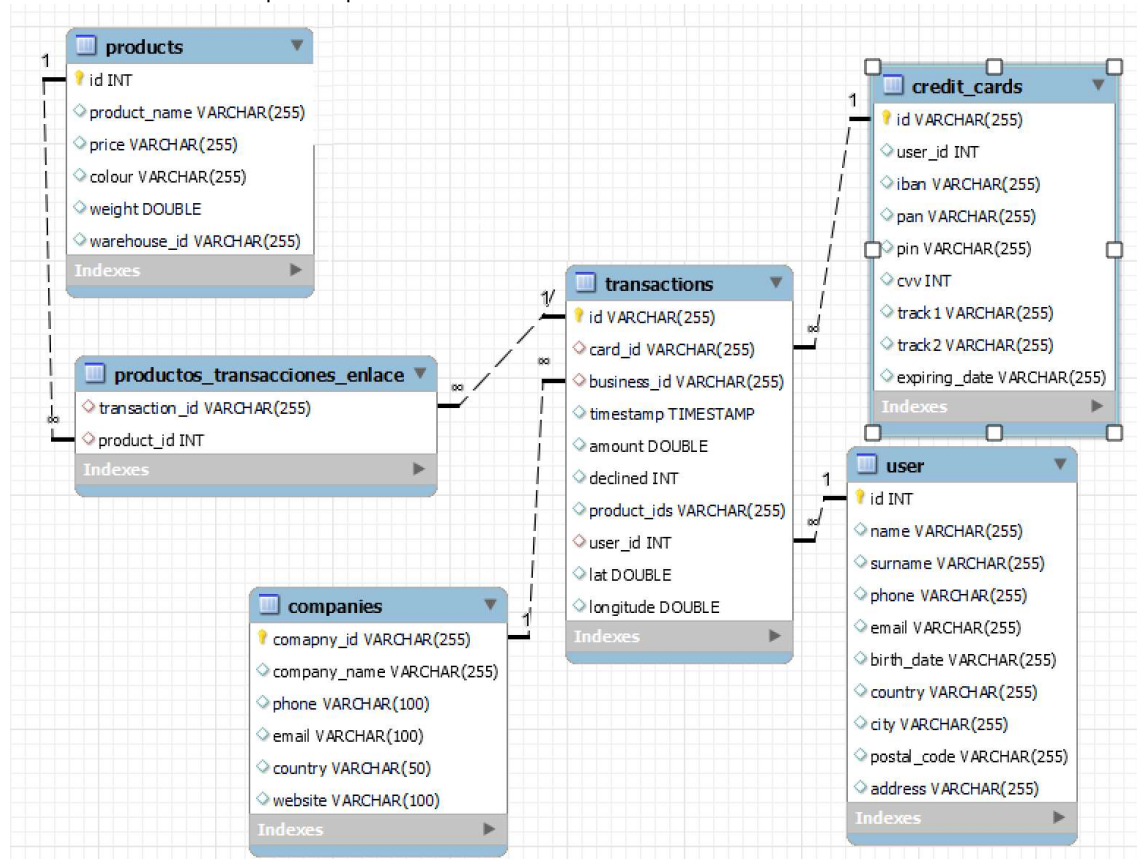
Resultados:

✓	35	14:26:08	SELECT * FROM productos_transacciones_enlace LIMIT 0, 50000	1457 row(s) returned
✓	36	14:26:13	SELECT * FROM transactions_product LIMIT 0, 50000	1457 row(s) returned

El modelo final queda como el siguiente (esta imagen es de la tabla creada con power query, sin embargo quedaría igual después de la explicación anterior con sql)



El modelo creado por sql:



Exercici 1

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

Para esto tendremos que crear una consulta donde haga un recuento del id agrupado por nombre

Aquí a modo de corroborar que ambos paso anterior estén bien, también he hecho esta query sobre las dos tablas (solo en el pdf se muestra)

Sobre la tabla creada en sql:

```
82 #Nivel 3 Ex01
83 • SELECT p.product_name, count(tp.product_id) as recuento_ventas
84 FROM products p
85 JOIN products_transacciones_enlace tp ON p.id = tp.product_id
86 GROUP BY p.product_name;
87
```

product_name	recuento_ventas
Direwolf Stannis	106
Tarly Stark	65
duel tourney Lannister	51
skywalker ewok	100
north of Casterly	54
Karstark Dorne	48
palpatine chewbacca	60
skywalker ewok sith	61

Sobre la tabla creada en power query:

```
50 #La tabla de enlace fue creada en powerquery en el pdf esta explicado.
51 #Nivel 3 Ex01
52 • SELECT p.product_name, count(tp.product_id) as recuento_ventas
53 FROM products p
54 JOIN transactions_product tp ON p.id = tp.product_id
55 GROUP BY p.product_name;
```

product_name	recuento_ventas
Direwolf Stannis	106
Tarly Stark	65
duel tourney Lannister	51
skywalker ewok	100
north of Casterly	54
Karstark Dorne	48
palpatine chewbacca	60
skywalker ewok sith	61
dooku solo	49