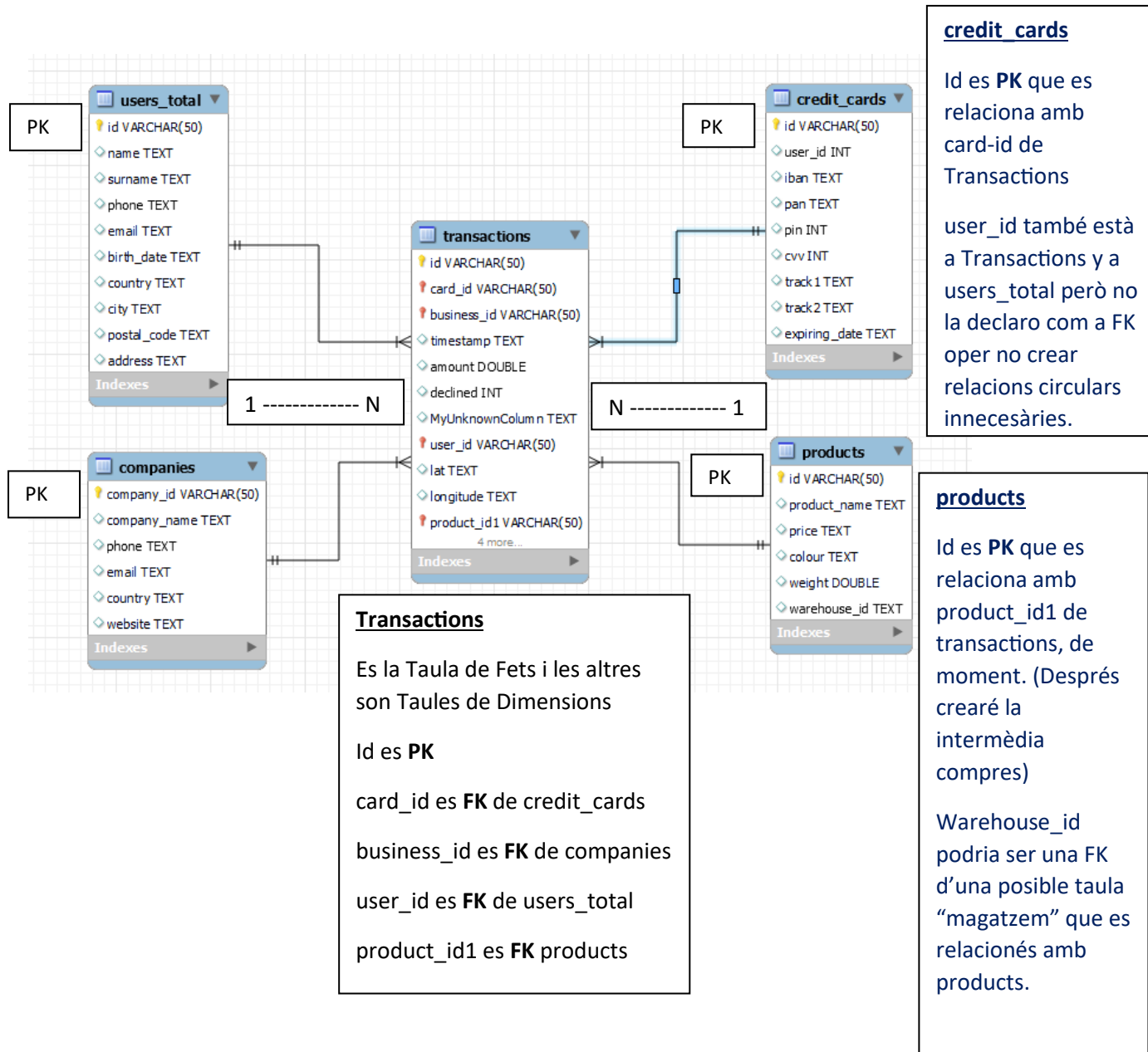


## SPRINT 4

### CORRECCIÓ:

En el esquema, fíjate que la tabla de hechos te dice 4 more, siempre muestra completo! La tabla credit\_cards y products tienen unas variables con una extensión \_id ¿se refieren a ids de otras tablas? ¿los tienes controlados?



La info que falta de la taula Transactions es aquesta:

Column	Type
id	VARCHAR(50)
card_id	VARCHAR(50)
business_id	VARCHAR(50)
timestamp	DATETIME
amount	DOUBLE
declined	INT
MyUnknownColumn	TEXT
user_id	INT
lat	TEXT
longitude	TEXT
product_id1	VARCHAR(50)
product_id2	VARCHAR(50)
product_id3	VARCHAR(50)
product_id4	VARCHAR(50)
product_id5	VARCHAR(50)

Index
PRIMARY
fk_card
fk_business
fk_user

En un primer moment separe els product id's en 5 columnes i declaro FK la primer columna product\_id1 , faig els primers exercicis així. Després quan he de calcular la quantitat de cada producte veig que no es suficient i creo una taula intermèdia i anul·lo la FK per no crear relacions circulars innecessàries.

```

/* Me doy cuenta del problema de product_ids y después de ver varias opciones decido modificarlo
directamente en excel (con la funcion texto en columnas) añadiendo columnas para cada articulo
product_id1,product_id2,3,4 i 5. Vuelvo a importar la tabla y aplico la PK y las FK's */
ALTER TABLE transactions
MODIFY COLUMN product_id2 VARCHAR(50),
MODIFY COLUMN product_id3 VARCHAR(50),
MODIFY COLUMN product_id4 VARCHAR(50),
MODIFY COLUMN product_id5 VARCHAR(50);
ALTER TABLE transactions
MODIFY COLUMN product_id1 VARCHAR(50),
ADD CONSTRAINT fk_product1 FOREIGN KEY (product_id1)
REFERENCES products(id);
/* Solo puedo referir como FK una de las columnas product pero no es problema para lo que se pide
de momento*/

```

Cuando creas la tabla ya le dices que id es tu PK ¿por qué se lo vuelves a decir en una actualización más adelante?

```

CREATE TABLE transactions (
  id VARCHAR(50) PRIMARY KEY,
  card_id VARCHAR(20),
  business_id VARCHAR(20),
  timestamp varchar(100),
  amount FLOAT,

  ALTER TABLE transactions
  MODIFY COLUMN id VARCHAR(50),
  ADD PRIMARY KEY (id);

```

Como te comenté en alguna ocasión y en el archivo sql , el problema es que no consigo importar creando la tabla e importando con LOAD DATA:

```
/*No consigo importar el contenido con LOAD DATA, Después de muchos intentos y modificaciones,  
al final consigo importarla con el import wizard creando tabla nueva con import wizard tb*/
```

La única manera en que lo consigo es desde import wizard y creando tabla nueva, por lo que el código de creación que escribo al inicio no es el de la tabla que consigo crear e introducir los datos.

```
CREATE TABLE `transactions` (  
  `id` varchar(50) NOT NULL,  
  `card_id` varchar(50) DEFAULT NULL,  
  `business_id` varchar(50) DEFAULT NULL,  
  `timestamp` datetime DEFAULT NULL,  
  `amount` double DEFAULT NULL,  
  `declined` int DEFAULT NULL,  
  `user_id` int DEFAULT NULL,  
  `lat` text,  
  `longitude` text  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Lo mismo con las tablas users. Las importo con Wizard y creo una nueva users\_total uniendo las otras.

```
/*Sigo sin poder importar el contenido con LOAD DATA, al final consigo importarla con el  
import wizard creando tabla nueva con import wizard tb. Hago lo mismo con todas las tablas.*/
```

```
CREATE TABLE users_total (  
  `id` int DEFAULT NULL,  
  `name` text,  
  `surname` text,  
  `phone` text,  
  `email` text,  
  `birth_date` text,  
  `country` text,  
  `city` text,  
  `postal_code` text,  
  `address` text  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
69 /* Empiezo por unir las 3 tablas de users ya que no veo ninguna ventaja en tenerlas separadas,  
70 manteniendo el orden del id para después poderlo cambiar a autoincremental*/  
71 • INSERT INTO users_total (id,name,surname,phone,email,birth_date,country,city,postal_code,address)  
72 SELECT * FROM users_usa  
73 UNION ALL  
74 SELECT * FROM users_uk  
75 UNION ALL  
76 SELECT * FROM users_ca;  
77 # compruebo que la tabla nueva es ok y elimino las anteriores
```

```

# Para crear el modelo asigno las PK. Como he importado en automático tengo que cambiar el tipo de columna
ALTER TABLE transactions
MODIFY COLUMN id VARCHAR(50),
ADD PRIMARY KEY (id);
ALTER TABLE companies
MODIFY COLUMN company_id VARCHAR(50),
ADD PRIMARY KEY (company_id);
ALTER TABLE credit_cards
MODIFY COLUMN id VARCHAR(50),
ADD PRIMARY KEY (id);
ALTER TABLE products
MODIFY COLUMN id VARCHAR(50),
ADD PRIMARY KEY (id);
ALTER TABLE users_total
MODIFY COLUMN id INT AUTO_INCREMENT,
ADD PRIMARY KEY (id);

# ... y asigno las FK.
ALTER TABLE transactions
MODIFY COLUMN card_id VARCHAR(50),
ADD CONSTRAINT fk_card FOREIGN KEY (card_id)
REFERENCES credit_cards(id);
ALTER TABLE transactions
MODIFY COLUMN business_id VARCHAR(50),
ADD CONSTRAINT fk_business FOREIGN KEY (business_id)
REFERENCES companies(company_id);
ALTER TABLE transactions
MODIFY COLUMN user_id VARCHAR(50),
ADD CONSTRAINT fk_user FOREIGN KEY (user_id)
REFERENCES users_total(id);

```

## NIVELL 1

### EXERCICI 1

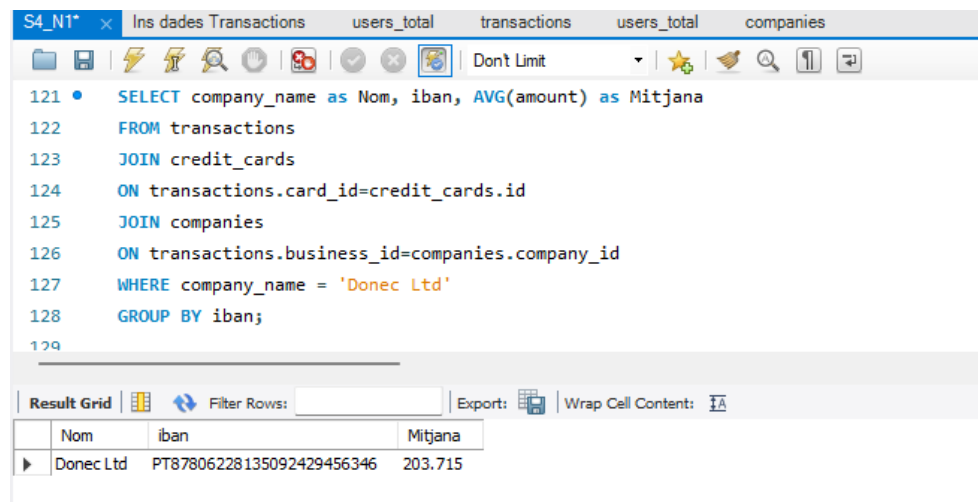
S4\_NT\* > Ins dades Transactions users\_total transactions users\_total

111 • `SELECT users_total.id, users_total.name, users_total.surname, COUNT(users_total.id)`  
 112 `FROM users_total`  
 113 `JOIN transactions`  
 114 `ON users_total.id = transactions.user_id`  
 115 `GROUP BY users_total.id`  
 116 `HAVING COUNT(users_total.id) > 30;`  
 117 `# Como he unido las 3 tablas user en una no puedo hacerlo sin subquerys`  
 118

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

	id	name	surname	COUNT(users_total.id)
▶	92	Lynn	Riddle	39
	267	Ocean	Nelson	52
	272	Hedwig	Gilbert	76
	275	Kenyon	Hartman	48

## EXERCICI 2



```
121 • SELECT company_name as Nom, iban, AVG(amount) as Mitjana
122 FROM transactions
123 JOIN credit_cards
124 ON transactions.card_id=credit_cards.id
125 JOIN companies
126 ON transactions.business_id=companies.company_id
127 WHERE company_name = 'Donec Ltd'
128 GROUP BY iban;
129
```

Result Grid

Nom	iban	Mitjana
Donec Ltd	PT87806228135092429456346	203.715

## 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 amb correcció

```
3 | Exercici 1
4 | Quantes targetes estan actives?*/
5 • CREATE TABLE estat_tarjetes (
6 |     id VARCHAR (50) primary key,
7 |     trans_ultima int,
8 |     trans_penultima int,
9 |     trans_avantpenultima int
10 | );
11 • UPDATE transactions
12 | SET timestamp = STR_TO_DATE(timestamp, '%d/%m/%Y %H:%i');
13 • ALTER TABLE transactions MODIFY COLUMN timestamp DATETIME;
14 • INSERT INTO estat_tarjetes (id, trans_ultima, trans_penultima, trans_avantpenultima)
15 | SELECT card_id,
16 | (SELECT declined FROM transactions as t1 WHERE t1.card_id = tarjetes_uniques.card_id
17 | ORDER BY timestamp DESC LIMIT 1) as trans_ultima,
18 | (SELECT declined FROM transactions as t2 WHERE t2.card_id = tarjetes_uniques.card_id
19 | ORDER BY timestamp DESC LIMIT 1 OFFSET 1) as trans_penultima,
20 | (SELECT declined FROM transactions as t3 WHERE t3.card_id = tarjetes_uniques.card_id
21 | ORDER BY timestamp DESC LIMIT 1 OFFSET 2) as trans_avantpenultima
22 | FROM (SELECT DISTINCT card_id FROM transactions) as tarjetes_uniques;
23 • SELECT COUNT(id) as tarjetes_actives
24 | FROM estat_tarjetes
25 | WHERE trans_ultima = 0;
```

Així queda la taula:

10 • `SELECT * FROM transactions.estat_tarjetas;`

id	trans_ultima	trans_penultima	trans_avantpenultima
CcU-2938	0	0	0
CcU-2945	0	1	NULL
CcU-2952	0	1	NULL
CcU-2959	0	0	0
CcU-2966	0	1	NULL

estat\_tarjetas 3 x Apply Revert

Output

Action Output

#	Time	Action	Message
✓ 27	12:35:03	SELECT * FROM transactions.estat_tarjetas	276 row(s) returned

23 • `SELECT COUNT(id) as tarjetas_activas`  
 24 `FROM estat_tarjetas`  
 25 `WHERE trans_ultima = 0;`  
 26  
 27

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
tarjetas_activas			
236			

Result 4 x Read Only

Output

Action Output

#	Time	Action	Message
✓ 27	12:35:03	SELECT * FROM transactions.estat_tarjetas	276 row(s) returned
✓ 28	12:35:55	SELECT COUNT(id) as tarjetas_activas FROM estat_tarjetas WHERE trans_ultima = 0	1 row(s) returned

## CORRECCIONES:

En el nivel 2, ejercicio 1 cuidado en el orden en que colocas las ejecuciones, si en este momento ejecuto todo me da error, no puede insertar nada porque la tabla no existe. Fíjate primero la creas, luego la eliminas y luego insertas!

```

S4_N1 | Ins datos Transactions | transactions | users_total | companies | S4_N2 | transactions | estat_tarjetas
-----|-----|-----|-----|-----|-----|-----|-----
3 | Ejercicio 1
4 | ¿Cuántas tarjetas están activas?*/
5 | CREATE TABLE estat_tarjetas (
6 |   id VARCHAR(30) PRIMARY KEY,
7 |   trans_ultima int,
8 |   trans_penultima int,
9 |   trans_avantpenultima int
10 | )
11 |
12 | drop table estat_tarjetas;
  
```

Usé el drop table para una corrección y se me olvidó borrarlo, lo elimino.

Cuidado con el código, finalmente estás mostrando cuando `trans_ultima = 0`, pero no estás generando un estado activo inactivo, tienes 3 columnas, necesito que resuelvas lo siguiente, genera la clasificación si está o no activa tomando en cuenta esas tres últimas transacciones.

Aquí me muestras que 40 compañías tienen la última transacción rechazada y 235 la tienen aceptada: las inactivas deben ser las que estas **tres ultimas** fueron declinadas (`declined = 1`)

Mi razonamiento para hacerlo de esta manera es el siguiente:

Como la pregunta es cuantas tarjetas están activas y la única manera de que estén inactivas es que las últimas 3 transacciones hayan sido declinadas, todas aquellas que tengan la última

transacción OK estarán activas. Lo hice así porque me pareció la manera más simple de resolver el enunciado.

No tuve en cuenta que si la última es KO pero las anteriores son OK también estará activa.

Entiendo que lo que me pides es que cree una columna que indique el estado activo o inactivo tomando en cuenta la última transacción. Lo haría así:

```
27 • ALTER TABLE estat_tarjetas
28   ADD COLUMN activa BOOLEAN DEFAULT 1;
29
30 • UPDATE estat_tarjetas
31   SET activa = CASE
32     WHEN trans_ultima = 0 AND trans_penultima = 0 AND trans_avantpenultima = 0 THEN 0
33     ELSE 1
34   END;
```

```
10 • SELECT * FROM transactions.estat_tarjetas;
```

Result Grid				
Filter Rows:				
Edit: Export/Import: Wrap Cell Content:				
id	trans_ultima	trans_penultima	trans_avantpenultima	activa
CdJ-2938	0	0	0	0
CdJ-2945	0	1	NULL	1
CdJ-2952	0	1	NULL	1
CdJ-2959	0	0	0	0
CdJ-2966	0	1	NULL	1

Output			
Action Output			
#	Time	Action	Message
35	13:04:25	UPDATE estat_tarjetas SET activa = CASE WHEN trans_ultima = 0 AND trans_penulti...	0 row(s) affected Rows matched: 276 Changed: 0 Warnings: 0
36	13:05:01	SELECT * FROM transactions.estat_tarjetas	276 row(s) returned

```
36 • SELECT COUNT(id)
37   FROM estat_tarjetas
38   WHERE activa = 1;
```

Result Grid	
Filter Rows:	
Export: Wrap Cell Content:	
COUNT(id)	
▶	268

El nivel 3 no puedo corregírtelo hasta que no vea lo que hiciste en el nivel 1. Pero es información que tienes ya en tu tabla transaction, parte de ahí para crear una nueva tabla o tablas.

Un saludo,

Alana

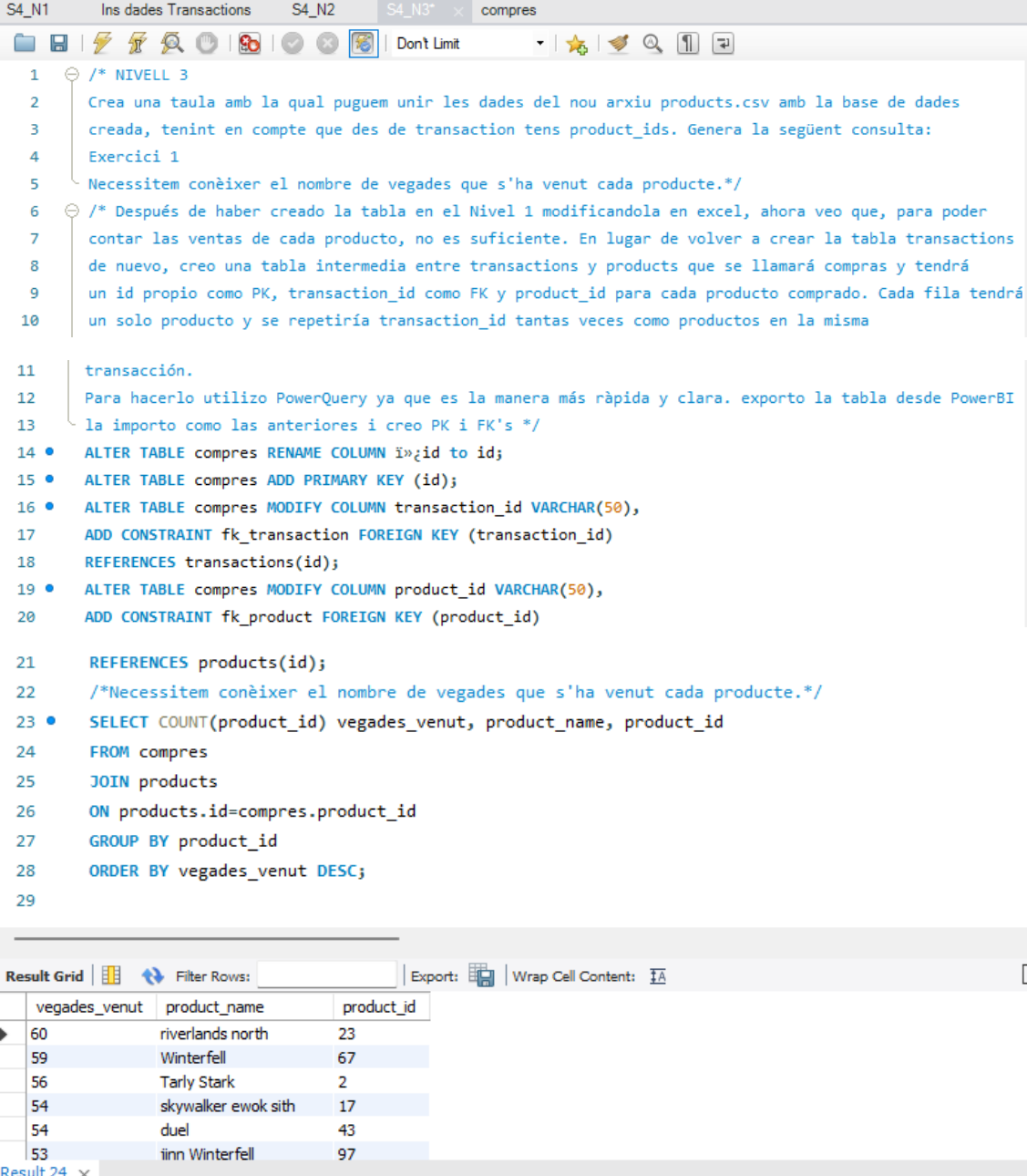
Espero haber aclarado todo lo necesario para que puedas corregir el Nivel 3.

Gracias.

## 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



The screenshot shows a SQL IDE with a script editor and a result grid. The script editor contains the following SQL code:

```
1  /* NIVELL 3
2  Crea una taula amb la qual puguem unir les dades del nou arxiu products.csv amb la base de dades
3  creada, tenint en compte que des de transaction tens product_ids. Genera la següent consulta:
4  Exercici 1
5  Necessitem conèixer el nombre de vegades que s'ha venut cada producte.*/
6  /* Después de haber creado la tabla en el Nivel 1 modificandola en excel, ahora veo que, para poder
7  contar las ventas de cada producto, no es suficiente. En lugar de volver a crear la tabla transactions
8  de nuevo, creo una tabla intermedia entre transactions y products que se llamará compras y tendrá
9  un id propio como PK, transaction_id como FK y product_id para cada producto comprado. Cada fila tendrá
10 un solo producto y se repetiría transaction_id tantas veces como productos en la misma
11 transacción.
12 Para hacerlo utilizo PowerQuery ya que es la manera más rápida y clara. exporto la tabla desde PowerBI
13 la importo como las anteriores i creo PK i FK's */
14 • ALTER TABLE compres RENAME COLUMN i»¿id to id;
15 • ALTER TABLE compres ADD PRIMARY KEY (id);
16 • ALTER TABLE compres MODIFY COLUMN transaction_id VARCHAR(50),
17 ADD CONSTRAINT fk_transaction FOREIGN KEY (transaction_id)
18 REFERENCES transactions(id);
19 • ALTER TABLE compres MODIFY COLUMN product_id VARCHAR(50),
20 ADD CONSTRAINT fk_product FOREIGN KEY (product_id)
21 REFERENCES products(id);
22 /*Necessitem conèixer el nombre de vegades que s'ha venut cada producte.*/
23 • SELECT COUNT(product_id) vegades_venut, product_name, product_id
24 FROM compres
25 JOIN products
26 ON products.id=compres.product_id
27 GROUP BY product_id
28 ORDER BY vegades_venut DESC;
29
```

The result grid shows the following data:

vegades_venut	product_name	product_id
60	riverlands north	23
59	Winterfell	67
56	Tarly Stark	2
54	skywalker ewok sith	17
54	duel	43
53	inn Winterfell	97



## MODEL FINAL

```

30  /* Retoco las FK para dar coherencia al modelo final*/
31  • ALTER TABLE `trans_sprint4`.`transactions`
32    DROP FOREIGN KEY `fk_product1`;
33  • ALTER TABLE `trans_sprint4`.`transactions`
34    DROP INDEX `fk_product1` ;
35
36  • ALTER TABLE credit_cards
37    ADD CONSTRAINT fk_estat_tarjetas FOREIGN KEY (id)
38    REFERENCES estat_tarjetas(id);
39

```

