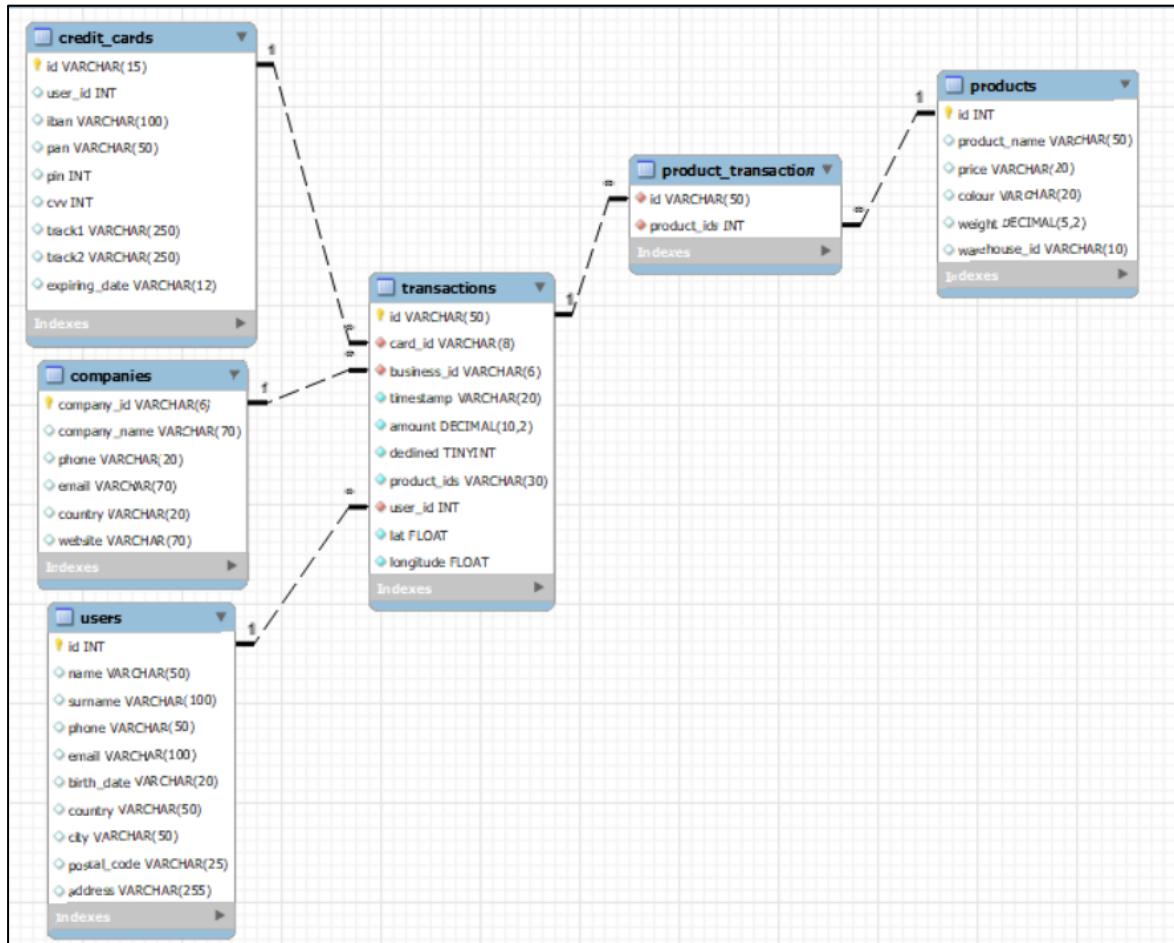


NIVEL 1



- Ejercicio 1

Realiza una subconsulta que muestre a todos los usuarios con más de 30 transacciones utilizando al menos 2 tablas.

Para este ejercicio he utilizado las tablas users y transactions, el resultado me da 4 líneas:

```

6 •  SELECT u.*, total_transactions          #2º Indico que quiero todo de la tabla users y el conteo de las transactions
7   FROM users u                            # que hice en el 1º paso que juntaré haciendo una JOIN
8   JOIN (
9     SELECT user_id, COUNT(id) AS total_transactions      #1º Busco user_id para poder agrupar el conteo de las transactions
10    FROM transactions
11   GROUP BY user_id
12  ) AS conteo ON u.id = conteo.user_id           #3º Despues de indicar las KEYS para hacer la JOIN filtro con un WHERE
13   WHERE total_transactions > 30;                # indicando que quiero las que sean mayor a 30 transactions
14
15
result Grid | Filter Rows: | Export: | Wrap Cell Content: | 


| ID  | Name   | Surname | Phone          | Email                           | Birth Date   | Country       | City          | Postal Code | Address                        | Total Transactions |
|-----|--------|---------|----------------|---------------------------------|--------------|---------------|---------------|-------------|--------------------------------|--------------------|
| 92  | Lynn   | Riddle  | 1-387-885-4057 | vitae.aliquet@outlook.edu       | Sep 21, 1984 | United States | Bozeman       | 61871       | P.O. Box 712, 7907 Est St.     | 39                 |
| 267 | Ocean  | Nelson  | 079-481-2745   | aenean@yahoo.com                | Dec 26, 1991 | Canada        | Charlottetown | 85X 3P4     | Ap #732-8357 Pede, Rd.         | 52                 |
| 272 | Hedwig | Gilbert | 064-204-8788   | sem.eget@icloud.edu             | Apr 16, 1991 | Canada        | Tuktoyaktuk   | Q4C 3G7     | P.O. Box 496, 5145 Sapien Road | 76                 |
| 275 | Kenyon | Hartman | 082-871-7248   | corvallis.ante.lectus@yahoo.com | Aug 3, 1982  | Canada        | Richmond      | R8H 2K2     | 8564 Facilis, St.              | 48                 |


```

- Ejercicio 2

Muestra el promedio de la suma de transacciones por IBAN de las tarjetas de crédito en la compañía Donec Ltd. utilizando al menos 2 tablas.

Lo primero que hago es encontrar el iban que corresponda con el nombre de la compañía que nos proporciona el ejercicio.

```
22 •   SELECT cc.iban                                #Por medio de esta consulta llego a encontrar el iban correspondiente a la empresa
23     FROM credit_cards cc
24   JOIN transactions t ON cc.id = t.card_id
25   JOIN companies c ON t.business_id = c.company_id
26 WHERE c.company_name = "Donec Ltd";
```

He tenido en cuenta 2 criterios:

1º Considerando únicamente las transacciones que no han sido rechazadas.

```
28   #Una vez encontrado el iban que es: 'PT87806228135092429456346' hago la busquede de la media utilizando 3 tablas.
29 •   SELECT ROUND(AVG(t.amount),2)
30     FROM transactions t
31   JOIN credit_cards cc ON cc.id = t.card_id
32   JOIN companies c ON t.business_id = c.company_id
33 WHERE cc.iban = 'PT87806228135092429456346' AND t.declined = 0 ;    #Aquí solo tengo en cuenta las transacciones aprobadas.
```

Result Grid	Filter Rows:	Exports:	Wrap Cell Content:
ROUND(AVG(t.amount),2)			
► 42.82			

2º Considerando todas las transacciones indistintamente si fueron rechazadas o no.

```
35   #Una vez encontrado el iban que es: 'PT87806228135092429456346' hago la busquede de la media utilizando 3 tablas.
36 •   SELECT ROUND(AVG(t.amount),2)
37     FROM transactions t
38   JOIN credit_cards cc ON cc.id = t.card_id
39   JOIN companies c ON t.business_id = c.company_id
40 WHERE cc.iban = 'PT87806228135092429456346' ;                      #Aquí incluyo las aprobadas y rechazadas.
```

Result Grid	Filter Rows:	Exports:	Wrap Cell Content:
ROUND(AVG(t.amount),2)			
► 203.72			

NIVEL 2

Crea una nueva tabla que refleje el estado de las tarjetas de crédito basado en si las últimas tres transacciones fueron declinadas y genera la siguiente consulta:

La explicación está puesta paso a paso al lado del código, considerando el criterio indicado de tomar en cuenta solo aquellas tarjetas con un mínimo de 3 transacciones hechas en total. Para luego revisar las 3 últimas transacciones si se rechazaron o no y con base en esto: Si 1 de las 3 últimas transacciones se ha aprobado, se considera la tarjeta operativa de lo contrario no.

⚠ SI QUISIERA CONSIDERAR TODAS LAS TARJETAS (INCLUYENDO AQUELLAS QUE TIENEN MENOS DEL MÍNIMO DE 3 TRANSACCIONES HECHAS) SOLO TENDRÍA QUE QUITAR LA PARTE FINAL DEL CÓDIGO (HAVING COUNT(*) = 3)

```
43 • CREATE TABLE last_card_movements AS          #1º Creo la tabla last_card_movements
44   WITH card_status AS (                         #2º Defino una CTE llamada card_status
45     SELECT card_id, timestamp, declined,        #selecciono las columnas card_id, timestamp, y declined de la tabla transactions.
46     ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) #con la función ROW NUMBER asigno un número de fila a cada transacción dentro de cada card_id,
47     AS row_num                                #ordenando las transacciones por timestamp en orden descendente.
48   )
49
50   SELECT card_id,                            #2º Selecciono card_id de la CTE card_status
51   CASE                                         #utilizo CASE para calcular el estado de la tarjeta
52     WHEN SUM(declined) <= 2 THEN 'tarjeta operativa'  #basándome en la suma total de (declined) donde indico que la suma del campo declined sea <= 2
53     ELSE 'tarjeta no operativa'                 #debe ser así 'tarjeta operativa' de lo contrario 'tarjeta no operativa'
54   END AS status                               #desde la CTE card_status
55   FROM card_status                           #todo esto considerando que la row_num <=3 para que tome las 3 últimas transacciones
56   WHERE row_num <= 3                          #agrupando por card_id
57   GROUP BY card_id                          #y con HAVING hago que cuente todas las filas de cada card_id y que esta sea =3 para asegurar
58   HAVING COUNT(*) = 3                      #que la tarjeta tiene 3 transacciones
59
60
61
```

Ejercicio 1

¿Cuántas tarjetas están activas?

Esta query me da como resultado 9 tarjetas

Teniendo la tabla anterior solo tenemos que hacer un conteo de los card_id filtrando por aquellos que hayan sido considerados como 'tarjeta operativa'

```
64 •   SELECT COUNT(card_id) as 'total tarjetas operativas'      #1º Hago un COUNT de los card_id de la tabla que he creado (last_card_movements)
65     FROM last_card_movements                                # para ver cuantos encuentra.....
66     WHERE status = 'tarjeta operativa';                      # filtrando solo las que el campo status sea = 'tarjeta operativa'
67
```

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

total tarjetas operativas
9

Esta es la query en la que se muestra la lista completa de las tarjetas y su estado:

```
107  
108 •   SELECT *  
109     FROM operations.last card movements:  
  
Result Grid | Filter Rows: | Export: | Wrap  


|   | card_id  | status            |
|---|----------|-------------------|
| ▶ | CcU-2938 | tarjeta operativa |
|   | CcU-2959 | tarjeta operativa |
|   | CcU-2994 | tarjeta operativa |
|   | CcU-3393 | tarjeta operativa |
|   | CcU-3981 | tarjeta operativa |
|   | CcU-4093 | tarjeta operativa |
|   | CcU-4191 | tarjeta operativa |
|   | CcU-4219 | tarjeta operativa |
|   | CcU-4849 | tarjeta operativa |


```

NIVEL 3

Crea una tabla con la que podamos unir los datos del nuevo archivo products.csv con la base de datos creada, teniendo en cuenta que desde transaction tienes product_ids. Genera la siguiente consulta:

! PARA PODER ESTABLECER LA RELACION CON LA TABLA PRODUCTS HE TENIDO QUE CREAR UNA TABLA PUENTE

ENTRE LAS TABLAS TRANSACTIONS Y PRODUCTS A LA QUE HE LLAMADO PRODUCT_TRANSACTION.

HE OPTADO POR ESTA OPCION YA QUE EN TRANSACTIONS EN EL CAMPO PRODUCTS_ID HAY MAS DE UN ID EN MUCHOS CASOS SEPARADOS POR,

1º Creo la tabla que utilizaré en el modelo:

```
CREATE TABLE product_transaction (
    id varchar(50) NOT NULL,
    product_ids int NOT NULL,
    FOREIGN KEY(id) REFERENCES transactions(id),
    FOREIGN KEY(product_ids) REFERENCES products(id)
);
```

! Los datos para esta tabla los he obtenido de la siguiente manera:

2º Cargo el CSV de la tabla transactions en **Excel** y elimino las columnas que no voy a utilizar hasta quedarme sólo con id y products_id

3º En **Worbench** creo una tabla temporal para poder hacer los cambios previos.

```
CREATE TABLE temporal (
    id VARCHAR(50),
    product_id VARCHAR(100)
);
```

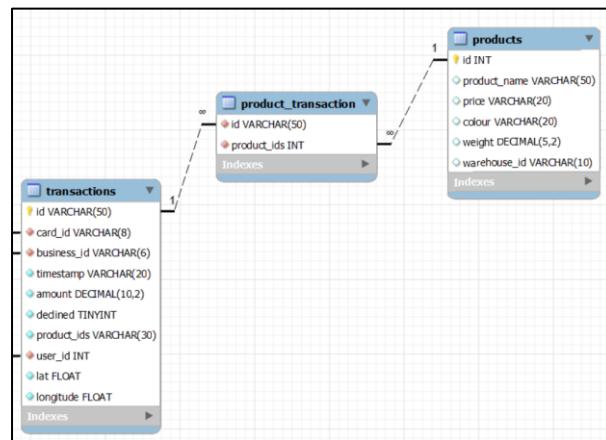
4º Cargo los datos con **WIZARD** desde el CSV que he utilizado en el paso anterior.

5º Separo los valores de product_ids separados por , en filas desde la tabla temporal a una definitiva (product_transaction)

```
INSERT INTO product_transaction (id, product_ids)
(SELECT id,
SUBSTRING_INDEX(SUBSTRING_INDEX(product_ids, ',', numbers.n), ',', -1) AS product_id
FROM temporal
JOIN ( SELECT 1 AS n
UNION ALL SELECT 2
UNION ALL SELECT 3
UNION ALL SELECT 4 ) AS numbers
ON CHAR_LENGTH(product_ids) - CHAR_LENGTH(REPLACE(product_ids, ',', '')) >= n - 1);
```

6º Finalmente elimino la tabla temporal.

La relación entre las tablas queda de esta manera:



Ejercicio 1

Necesitamos conocer el número de veces que se ha vendido cada producto.

Hago una query donde selecciono todo de product y un conteo de product_ids de la tabla product_transaction , hago la JOIN y para que el conteo sea por cada p.id agrupo por p.id. Resultado en 26 líneas:

```
100 •  SELECT p.*, count(product_ids) unidades_vendidas
101    FROM product_transaction pt
102   JOIN products p ON pt.product_ids = p.id
103  GROUP BY p.id
104 ;
```

Result Grid							
	id	product_name	price	colour	weight	warehouse_id	unidades_vendidas
▶	1	Direwolf Stannis	\$161.11	#7c7c7c	1.00	WH-4	61
	2	Tarly Stark	\$9.24	#919191	2.00	WH-3	65
	3	duel tourney Lannister	\$171.13	#d8d8d8	1.50	WH-2	51
	5	skywalker ewok	\$171.22	#dbbdb	3.20	WH-0	49
	7	north of Casterly	\$63.33	#b7b7b7	0.60	WH--2	54
	11	Karstark Dorne	\$49.70	#141414	2.70	WH-6	48