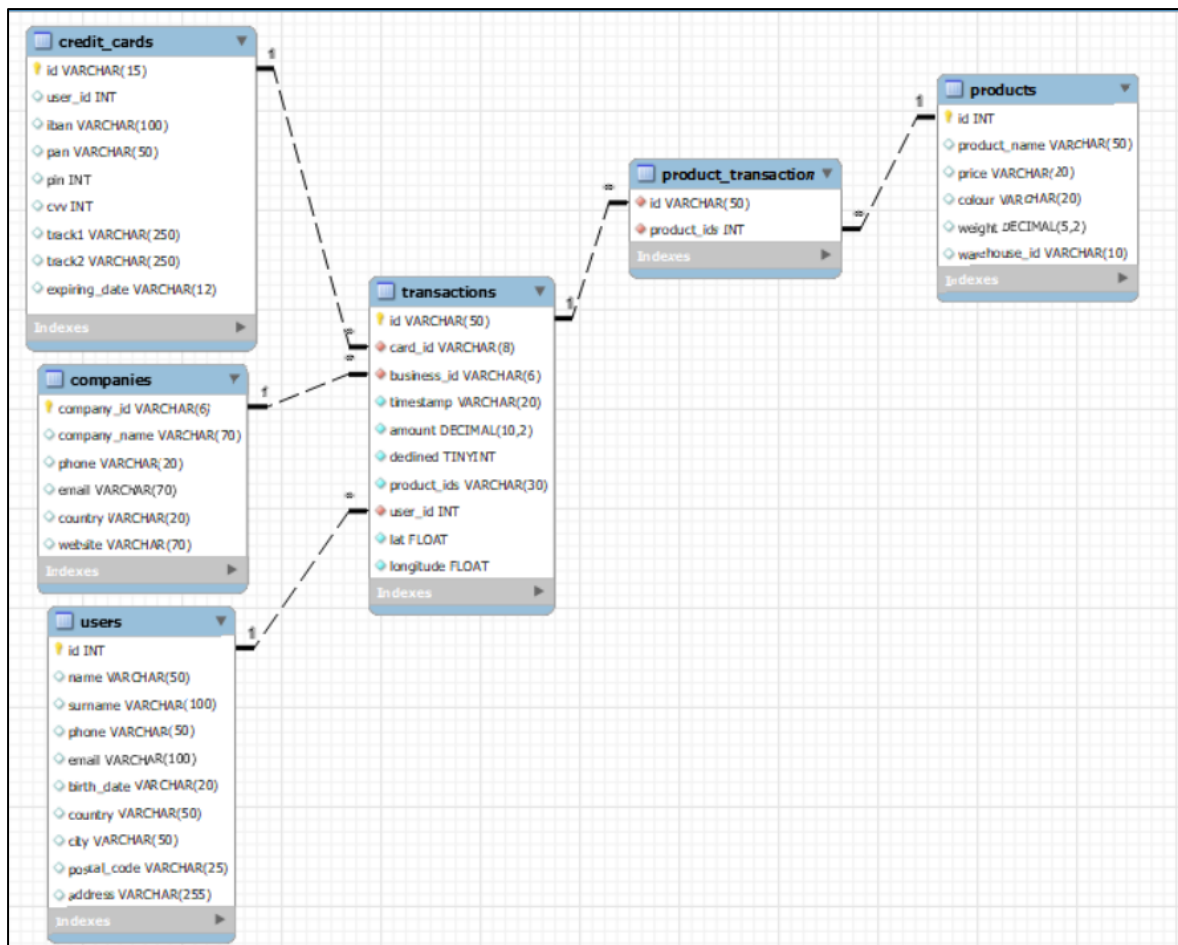


# NIVEL 1



## - Exercici 1

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

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

```

6 • SELECT u.*, total_transactions
7 FROM users u
8 JOIN (
9     SELECT user_id, COUNT(id) AS total_transactions
10    FROM transactions
11   GROUP BY user_id
12 ) AS conteo ON u.id = conteo.user_id
13 WHERE total_transactions > 30;
14
15
    
```

#2º Indico que quiero todo de la tabla users y el conteo de las transactions  
# que hice en el 1º paso que juntaré haciendo una JOIN

#1º Busco user\_id para poder agrupar el conteo de las transactions

#3º Despues de indicar las KEYS para hacer la JOIN filtro con un WHERE  
# indicando que quiero las que sean mayor a 30 transactions

id	name	surname	phone	email	birth_date	country	city	postal_code	address	total_transactions
92	Lynn	Riddle	1-387-885-4057	vitaе.аllquеt@outlook.edu	Sep 21, 1984	United States	Bozeman	61871	P.O. Box 712, 7907 Est St.	39
267	Ocean	Nelson	079-481-2745	aеnеan@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 Sapient Road	76
275	Kenyon	Hartman	082-871-7248	convallis.ante.lectus@yahoo.com	Aug 3, 1982	Canada	Richmond	R8H 2K2	8564 Facilis. St.	48

## - 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 ejercicio he tenido en cuenta 2 criterios:

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

29	•	SELECT ROUND(AVG(amount),2) FROM transactions WHERE card_id = 'CcU-2973' AND declined = 0 ; #Aquí solo tengo en cuenta las transacciones aprobadas.
30		
31		
32		

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

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

33	•	SELECT ROUND(AVG(amount),2) FROM transactions WHERE card_id = 'CcU-2973' ; #Aquí incluyo las aprobadas y rechazadas.
34		
35		

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

## NIVEL 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:*

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.

```
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   FROM transactions
49 )
50 SELECT card_id,                                                       #2º Seleccione 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'                                       #de ser así 'tarjeta operativa' de lo contrario 'tarjeta no operativa'
54 END AS status
55 FROM card_status                                                       #desde la CTE card_status
56 WHERE row_num <= 3                                                    #todo esto considerando que la row_num <=3 para que tome las 3 últimas transacciones
57 GROUP BY card_id                                                     #agrupando por card_id
58 HAVING COUNT(*) = 3                                                  #y con HAVING hago que cuente todas las filas de cada card_id y que esta sea =3 para asegurar
59 ;                                                                    #que la tarjeta tiene 3 transacciones
60
61
```

### Exercici 1

*Quantes targetes estan actives?*

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

## NIVEL 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:*

El proceso que he seguido ha sido:

1º He creado una tabla derivada de la tabla products que haga de nexo entre transactions y products a la cual he llamado product\_transaction .

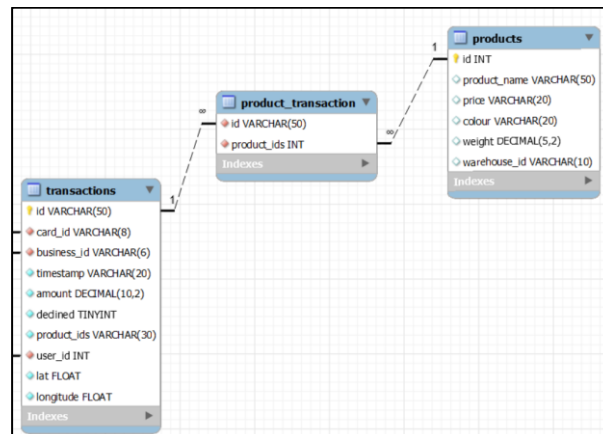
En la que he dejado únicamente dos campos que considero son los que necesito:

(id , product\_ids).

2º Al no tener las primary keys he indexado las dos columnas de esta nueva tabla para poder hacer la relación

3º He declarado como foreign key los dos campos de esta nueva tabla y hago referencia hacia las tablas con las que quiero crear la relación.

```
10 • CREATE TABLE product_transaction (  
11     id varchar(50) NOT NULL,  
12     product_ids int NOT NULL  
13 );  
14  
15 #CREACIÓN DE INDICES  
16 • ALTER TABLE product_transaction #agrego indice a product_transaction.id  
17 ADD INDEX idx_id (id);  
18  
19 • ALTER TABLE product_transaction #agrego indice a product_transaction.product_ids  
20 ADD INDEX idx_product_ids (product_ids);  
21  
22 #CREACION DE FOREIGN KEY'S  
23 • ALTER TABLE product_transaction  
24 ADD FOREIGN KEY (id) REFERENCES transactions(id); #creo foreign key product_transaction----->transactions  
25  
26 • ALTER TABLE product_transaction #ACHTUN...!!!!  
27 ADD FOREIGN KEY (product_ids) REFERENCES products(id); #creo foreign key product_transaction----->products
```



## Exercici 1

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

Hago una Subquery donde utilizo la función ROW NUMBER , el proceso está detallado al lado del código. Resultado en 26 líneas:

```
82 SELECT product_ids producto, COUNT(row_num) total_vendido FROM
83 (SELECT product_ids ,
84      ROW_NUMBER() OVER (PARTITION BY product_ids ) AS row_num
85      FROM product_transaction) dd
86 GROUP BY product_ids
87 ORDER BY producto ASC
88 ;
89
```

#2: La query que está por encima muestra el produc\_ids 'producto' y hace un conteo de la row\_num  
#1: Hago una subquery seleccionando product\_ids don una funcion  
# ROW NUMBER para que vaya generando un conteo de las filas por cada product\_ids  
#3: Agrupo el resultado por el campo product\_ids  
#4: Y lo ordeno para que sea mas visual

result Grid | Filter Rows: | Export: | Wrap Cell Content: |

producto	total_vendido
1	61
2	65
3	51
5	49
7	54