

## Sprint 4

### NIVEL 01

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:

Primero creo las tablas para poder introducir los datos:

```
CREATE TABLE companies (  
  company_id VARCHAR(15),  
  company_name VARCHAR(100),  
  phone VARCHAR(15),  
  email VARCHAR(100),  
  country VARCHAR(100),  
  website VARCHAR(100),  
  PRIMARY KEY (company_id) );
```

```
CREATE TABLE credit_cards (  
  id VARCHAR(15),  
  user_id VARCHAR(15),  
  iban VARCHAR(30),  
  pan VARCHAR(20),  
  pin VARCHAR(4),  
  cvv INT,  
  track1 VARCHAR(100),  
  track2 VARCHAR(100),  
  expiring_date DATE,  
  PRIMARY KEY (id) );
```

```
CREATE TABLE products (  
  id VARCHAR(4),  
  product_name VARCHAR(20),  
  price DECIMAL(10,2),  
  colour VARCHAR(20),  
  weight VARCHAR(20),  
  warehouse_id VARCHAR(20),  
  PRIMARY KEY (id) );
```

```
CREATE TABLE transactions (
  id VARCHAR(50),
  card_id VARCHAR(15),
  bussines_id VARCHAR(15),
  fecha_hora timestamp,
  amount DECIMAL (10,2),
  declined BOOLEAN,
  products_ids VARCHAR(20),
  user_id INT,
  latitude VARCHAR(20),
  longitude VARCHAR(20),
  PRIMARY KEY (id) );
```

```
CREATE TABLE users (
  id VARCHAR (4),
  name VARCHAR (50),
  surname VARCHAR (50),
  phone VARCHAR(50),
  email VARCHAR(50),
  birth_date DATE,
  country VARCHAR(50),
  city VARCHAR(50),
  postal_code VARCHAR(10),
  address VARCHAR(50),
  PRIMARY KEY (id) );
```

El código para introducir un csv en una tabla es el siguiente:

```
LOAD DATA LOCAL INFILE "C:\Users\usuario\Documents\FADD\FADD CURS PRESENCIAL\datos sprint 03\users_ca.csv"
INTO TABLE users
FIELDS TERMINATED BY ","
LINES TERMINATED BY "\r\n"
IGNORE 1 ROWS;
#codigo para entrar documento csv, pero me da error: "file request rejected due to restrictions on access",
```

No me ha funcionado, por el siguiente error: "file request rejected due to restrictions on Access"

He intentado con Import Wizard, pero la tabla users tampoco me ha funcionado el resto de las tablas sí.

Tratando de encontrar soluciones primero he modificado el Excel, separando por columnas:

*Datos>Texto en Columnas*

*Elegir y seleccionar Delimitador (,)*

*Finalizar*

id	name	surname	phone	email	birth_date	country	city	postal_code	address		
201	Iola	Powers	018-139-4717	ante.blandit@	Mar 20, 2000	Canada	Rigolet	V6T 6M7	154-5415 Auctor St.		
202	Maxwell	Holden	045-402-7693	donec@hotmail	Dec 2, 1986	Canada	Murdochville	S7E 6E0	Ap #880-6372 Ultrices. St.		
203	Jarrold	Fields	010-741-8105	sit.amet@goo	Jan 6, 1982	Canada	Baddeck	K3X 6Z5	441-8969 Rhoncus Road		
204	Emerson	Sharp	068-138-9383	ante.laculis@	Oct 15, 1994	Canada	Maple Creek	Y2C 9E6	517-6759 Ut, Av.		
205	Sonya	Mckee	041-151-9737	magna.phase	May 7, 1983	Canada	Dieppe	E7S 4P8	Ap #916-8051 A St.		

Con esta modificación tampoco me dejaba importar los datos en MySQL, finalmente he creado una nueva tabla después de importar el documento con “Import Wizard” para la tabla users y así, si que me ha funcionado.

Posteriormente he creado la tabla users\_all, haciendo UNION de las tres tablas.

```
CREATE TABLE users_all AS
SELECT id, name, surname, phone, email, birth_date, country, city, postal_code, address FROM users_ca
UNION ALL
SELECT id, name, surname, phone, email, birth_date, country, city, postal_code, address FROM users_uk
UNION ALL
SELECT id, name, surname, phone, email, birth_date, country, city, postal_code, address FROM users_usa
ORDER BY id desc ;
```

De esta manera he unificado la información de las 3 tablas (users\_ca, users\_uk, users\_usa)




Table Name:

Charset/Collation:

Comments:

Index Name	Type
PRIMARY	PRIMARY
id_UNIQUE	UNIQUE
<input type="text"/>	

Index Columns

Column	#
--------	---

Al indicar que ID es unico se ha creado el Indice.

Posteriormente he dividido la columna product\_ids de la tabla transactions, para poder vincular la tabla products con los products\_ids, y he creado una tabla nueva llamada transactions2 con las 4 columnas de los products\_ids:

```

CREATE TABLE transactions2 AS
SELECT
id,
card_id,
bussines_id,
fecha_hora,
amount,
declined,
user_id,
latitude,
longitude,

SUBSTRING_INDEX(products_ids, ",", 1) AS product1,

> CASE WHEN LENGTH(products_ids) - LENGTH(REPLACE(products_ids, ',', '')) >= 1
- THEN SUBSTRING_INDEX(SUBSTRING_INDEX(products_ids, ",", 2), ",", -1) END AS product2,

> CASE WHEN LENGTH(products_ids) - LENGTH(REPLACE(products_ids, ',', '')) >= 2
- THEN SUBSTRING_INDEX(SUBSTRING_INDEX(products_ids, ",", 3), ",", -1) END AS product3,

> CASE WHEN LENGTH(products_ids) - LENGTH(REPLACE(products_ids, ',', '')) >= 3
- THEN SUBSTRING_INDEX(SUBSTRING_INDEX(products_ids, ",", 4), ",", -1) END AS product4

FROM transactions;

```

Información sobre el código:

Para contar items en un campo separados entre "," : `LENGTH(CampoTBL) - LENGTH(REPLACE(CampoTBL, ',', ''))`

Para separar los registros d'un campo, en diferentes columnas:

`SUBSTRING_INDEX(SUBSTRING_INDEX(products_ids, ",", 4), ",", -1)`

Posteriormente he creado las FK de las diferentes tablas con la nueva tabla transactions2:

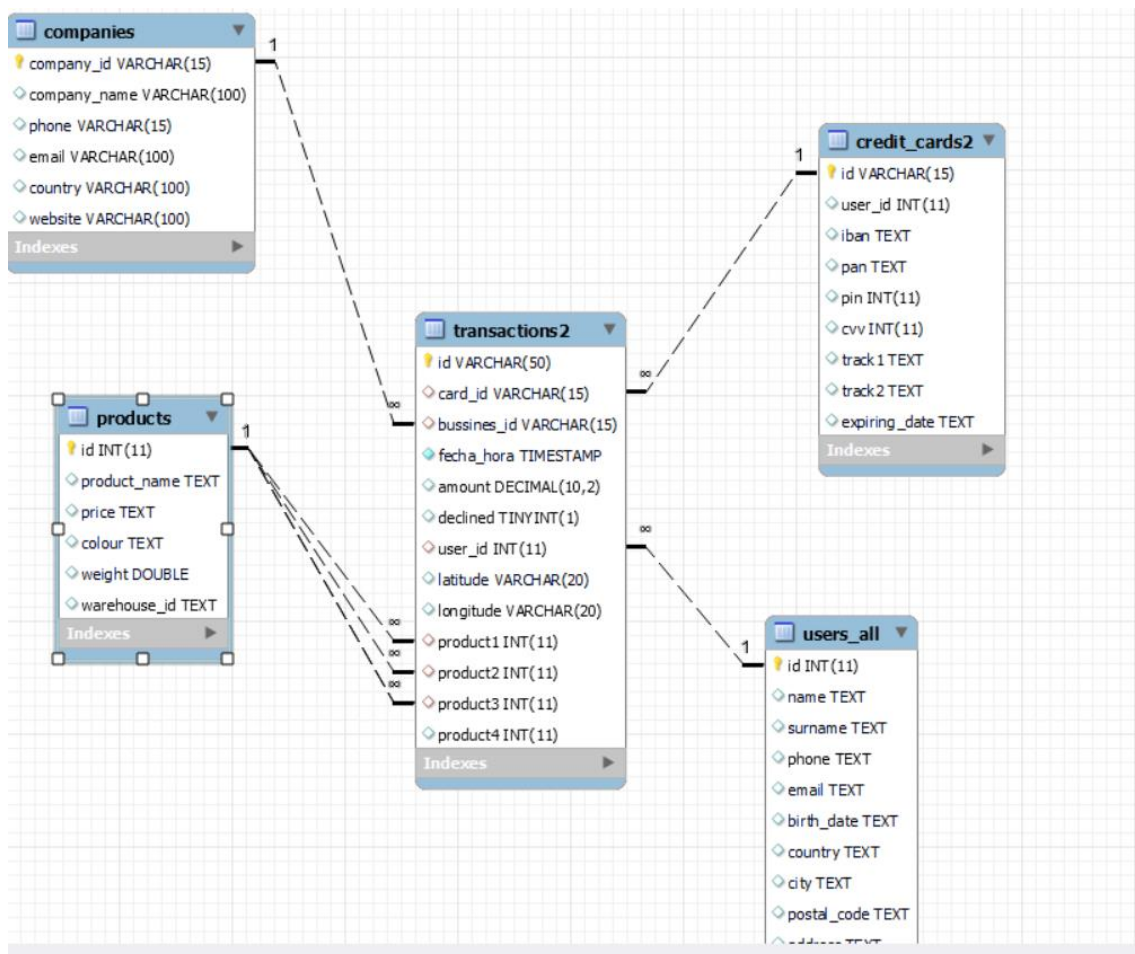
```
ALTER TABLE transactions2
ADD foreign key (bussines_id) REFERENCES companies(company_id);
```

```
ALTER TABLE transactions2
ADD foreign key (card_id) REFERENCES credit_cards2(id);
```

```
ALTER TABLE transactions2
ADD foreign key (user_id) REFERENCES users_all(id);
```

```
ALTER TABLE transactions2
ADD FOREIGN KEY (product1) REFERENCES products(id),
ADD FOREIGN KEY (product2) REFERENCES products(id),
ADD FOREIGN KEY (product3) REFERENCES products(id),
ADD FOREIGN KEY (product3) REFERENCES products(id);
```

Despues de crear las relaciones tengo el siguiente Diagrama , con el modelo estrella:



**EX1.Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.**

```
SELECT *
FROM users_all
JOIN (
    SELECT user_id, count(user_id) as transaction_count
    FROM transactions2
    GROUP BY user_id) subq1
ON users_all.id = subq1.user_id
WHERE subq1.transaction_count > 30;
```

Los usuarios con mas de 30 transacciones son los siguientes:

id	name	surname	phone	email	birth_date	country	city	postal_code	address	user_id
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.	92
267	Ocean	Nelson	079-481-2745	aenean@yahoo.com	Dec 26, 1991	Canada	Charlottetown	85X 3P4	Ap #732-8357 Pede, Rd.	267
272	Hedwig	Gilbert	064-204-8788	sem.eget@icloud.edu	Apr 16, 1991	Canada	Tuktoyaktuk	Q4C 3G7	P.O. Box 496, 5145 Sapient Road	272
275	Kenyon	Hartman	082-871-7248	convallis.ante.lectus@yahoo.com	Aug 3, 1982	Canada	Richmond	R8H 2K2	8564 Facilisi. St.	275

**EX2.Mostra la mitjana de la suma de transaccions per IBAN de les targetes de crèdit en la companyia Donec Ltd. utilitzant almenys 2 taules.**

```
SELECT avg(sumaAmountTransaccions) As AverageSumaTransaccions
FROM (
    SELECT SUM(amount) AS sumaAmountTransaccions, iban
    FROM transactions2
    JOIN credit_cards2 ON transactions2.card_id = credit_cards2.id
    WHERE transactions2.bussines_id = (
        SELECT company_id
        FROM companies
        WHERE company_name= "Donec Ltd")
    GROUP BY iban ) AS Subq1;
```

1º He creado la query para obtener la suma del amount de las transacciones por iban, posteriormente lo he relacionado con la tabla company con una subquery y he buscado el company\_name = Donec Ltd.

Una vez he obtenido esta informacion he hecho la query principal para calcular el Average.

La media de la suma de las cantidades de las transacciones de la empresa Donec Ltd:

Result Grid		Filter Rows:
	AverageTransaccions	
	203.715000	

### CORRECCIÓN ( codigo mas sencillo):

```
SELECT avg(amount),iban
FROM transactions2
JOIN credit_cards2 ON transactions2.card_id = credit_cards2.id
JOIN companies ON companies.company_id = transactions2.bussines_id
WHERE company_name= "Donec Ltd"
GROUP BY iban;
```

Result Grid		Filter Rows:	Export:
	avg(amount)	iban	
	203.715000	PT87806228135092429456346	

## NIVELL2

### EX1.

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:

Quantes targetes estan actives?



```

CREATE TABLE status_CC
SELECT count(card_id) AS NumCC, card_status
FROM (
SELECT card_id,
CASE
WHEN SUM(declined) >= 3 THEN 'credit card not working'
ELSE 'credit card working'
END AS card_status
FROM transactions2
GROUP BY card_id
) Subq1
group by Subq1.card_status;

```

Todas las tarjetas son activas:

NumCC	card_status
275	credit card working

Esta nueva tabla la puedo vincular con la tabla Credit Cards, uniendo CC tabla credit cards con CC de la tabla Card\_Status. (ALTER TABLE CREDIT\_CARDS REFERENCES NUM CC)

No puedo seleccionar las ultimas 3 transacciones (por fecha) porque no me funciona las siguientes funciones por la versión de MySQL ( 5.7.44-log):

```

SELECT*
FROM transactions2
WHERE card_id IN (
SELECT card_id
FROM transactions2
ORDER BY fecha_hora DESC LIMIT 3) ;

##no funciona row_number()
SELECT card_id,fecha_hora,declined
ROW_NUMBER() OVER(PARTITION BY card_id ORDER BY fecha_hora DESC) AS rownum
FROM transactions2;

```

**Corrección:**

```
CREATE TABLE status_CC2
SELECT card_id, card_status
FROM (
SELECT card_id,
CASE
WHEN SUM(declined) >= 3 THEN 'credit card not working'
ELSE 'credit card working'
END AS card_status
FROM transactions2
GROUP BY card_id
) Subq1 ;
```

```
SELECT count(card_id), card_status
FROM status_CC2
GROUP BY card_status;
```

### NIVELL3

#### EX1

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:

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

```
CREATE TABLE sales_products
SELECT count(id) as NumProducte, product_name AS NomProducte, id AS NumId
FROM products
JOIN (
SELECT product1, product2, product3, product4
FROM transactions2
WHERE declined= 0 ) Subq1
ON Subq1.product1 = products.id or Subq1.product2= products.id or Subq1.product3= products.id or Subq1.product4= products.id
GROUP BY product_name, id;
```

En la siguiente tabla podemos ver cuantas veces se han vendido los distintos productos:

NumProducte	NomProducte	NumId
51	Direwolf Stannis	1
56	Tarly Stark	2
43	duel tourney Lannister	3
42	skywalker ewok	5
44	north of Casterly	7
40	Karstark Dorne	11
51	palpatine chewbacca	13
54	skywalker ewok sith	17
44	dooku solo	19
60	riverlands north	23

Para comprobar que es correcto he hecho la siguiente query:

```
SELECT count(id)
FROM transactions2
where declined=0 and (product1=13 or product2=13 or product3=13 or product4=13);
```

Result Grid	
	count(id)
▶	51