

Sprint 4: Modelat SQL

Tasca S4.01. Creació de Base de Dades



Fede Labate



Haile Meneses

Descripció

Partint d'alguns arxius CSV dissenyaràs i crearàs la teva base de dades.

Objectius

o Disseny i creació d'una base de dades



4 dies.

XLliurament

Emmagatzema en un repositori del teu GitHub una carpeta que contingui:

- L'arxiu .sql que contingui tots els scripts.
- Un PDF que contingui una captura de pantalla del workbench on es pugui observar el script de la consulta que vas fer i el resultat obtingut per a cada exercici.

En el lliurament, col·loca el link al repositori.





Tasca S4.01. Creació de Base de Dades	
<u></u> Autor	1
P2P correcció	1
	1
⊚ Objectius	1
Surada	1
XLliurament	1
Aclariments	3
★Nivell 1	4
≪Estructura de la base de dades	6
Exercici 1	8
Exercici 2	9
He après	10
☐ He recordat	10
<u></u> ★★Nivell 2	11
Exercici 1	11
He après	12
★★★Nivell 3	13
Exercici 1	13
☐ He recordat	13
∲ Recursos	14
👰 Correcció	14



Aclariments

Por una cuestión de optimización de espacio en memoria, defino el límite de cada tipo de dato al valor más grande de la base de datos.

Esta técnica no es recomendable para tablas dinámicas donde se adquieren datos nuevos desconocidos, pero dado que estamos trabajando con tablas estáticas fijas, lo implementé de esta manera.

Retos a fragmentar y resolver. Cómo se me estaba complicando el hecho de tener valores no formateados en el campo transaction.product_id, me propuse dividir el problema en pequeñas acciones.
Retos a resolver:
☐ Definir estructura de bbdd
Transformar con SQL:
☐ transaction.product_id (convertir en orders)
☐ Ya tenés las tablas (sp4_Inicial)
☐ Entender cómo separar las comas.
☐ Tienen que ir a una tabla nueva:
Creá columnas para el máximo de datos (4 es la que más tiene). Pone un número en cada campo
Transpone cada campo con lo que sería mi order.id
☐ Borrar vacíos / NULLs
 □ En la tabla general ahora esa columna de comas se borra. Y se crea un order_id consecutivo que comience en 1.
☐ Identificá los ids para hacerlos int
☐ Eliminar el campo credit_card.user_id
☐ Formatea las fechas (no es tan urgente)
☐ Hacer UNION de las 3 tablas user
Con todo eso resuelto, crear FK
Lujo sería formatear los datos para que todas las tablas coincidan y sean eficientes de recursos.



★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:

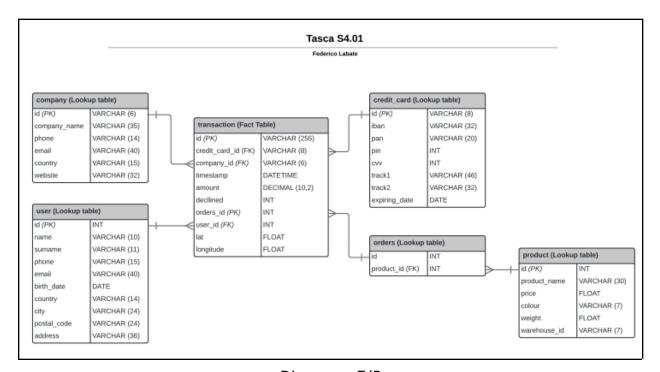


Diagrama E/R

El diseño se mantiene similar a los propuestos en los Sprint 2 y 3. **transaction** como tabla de hechos donde almacena las FK que se relacionan con las tablas de dimensiones **company**, **credit_card** (*), **user** (**) y **orders** (***).

(*) credit_card: Originalmente contenía una un campo llamado user_id el cual no me parecía relevante, incluso era problemático a la hora de relacionarlo con la tabla transaction, dado que el campo userd_id de la tabla transaction ya estaba relacionado con el campo id de la tabla user.

(**) user: Originalmente constaba de 3 tablas separadas por país (USA, UK y CA). Dado que las tres tenían la misma cantidad de campos y el mismo tipo de datos opté por hacer una UNION y unificarlas.

(***) **orders:** Creé esta tabla intermedia entre **transaction** y **product** dado que originalmente transaction contenía un campo llamado *product_id* donde justamente indicaba los ids de los productos vendidos en cada transacción pero separados por comas, ejemplo: 11, 13, 61, 29.



Opté por reemplazar ese campo por el nombre *orders_id* asignarle un valor único incremental del 1 al n-1 el cual se vincula con la nueva tabla llamada **orders**, pudiendo hacer una referencia directa de la siguiente manera:

Tabla transaction:

id	order_id
108B1D1D-5B23-A76C-55EF-C568E49A05DD	1
7DC26247-20EC-53FE-E555-B6C2E55CA5D5	2
72997E96-DC2C-A4D7-7C24-66C302F8AE5A	3
AB069F53-965E-A2A8-CE06-CA8C4FD92501	4

Tabla orders:

labia orders:	
id	product_id
1	59
2	71
2	41
3	97
3	41
3	3
4	11
4	13
4	61
4	29



XEstructura de la base de dades

Ver <u>Estructura de la base de dades.sql.</u>

La estrategia utilizada fue crear manualmente las 7 tablas provistas siendo todos los campos VARCHAR(256) e insertarles los datos a través de la declaración **LOAD DATA:**

Ejemplo de creación e inserción de la tabla transaction:

```
#Creación tabla transaction
 1
 2 • ⊝ CREATE TABLE transaction (
 3
           id VARCHAR(255),
 4
           card_id VARCHAR(255),
           business_id VARCHAR(255),
 5
           timestamp VARCHAR(255),
 6
 7
           amount VARCHAR(255),
           declined VARCHAR(255),
 8
 9
           products_id VARCHAR(255),
           user_id VARCHAR(255),
10
11
           lat VARCHAR(255),
           longitude VARCHAR(255)
12
13
       );
14
       #inserción de datos tabla transaction
15 •
       LOAD DATA
       INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'
16
17
       INTO TABLE transaction
18
       FIELDS TERMINATED BY ';'
       ENCLOSED BY '"'
19
       IGNORE 1 ROWS:
```

Figura 1.0.1 - Creación + inserción de datos de la tabla transaction

Una vez creadas las 7 tablas originales con sus datos correctamente insertados, resuelvo el campo problemático: **transaction**.*product_id*Para ello conlleva una transformación que se encuentra en el documento <u>Estructura de la base de dades.sql</u> bajo la siguiente identificación:

Una vez terminada con la transformación de transaction.product_id, procedo a:

- Eliminar el campo credit_card.user_id
- Hacer UNION de las 3 tablas user

Entonces ahora doy comienzo a la redefinición de la estructura de datos de todas las tablas. Se encuentra en el documento <u>Estructura de la base de dades.sql</u> bajo la siguiente identificación:





###Redefino estructura de las tablas para hacer un uso eficiente del espacio #### Una vez resuelta toda la estructura, procedo a crear todas las Foreign Keys correspondientes y comenzar a ejecutar los ejercicios de consultas.



Exercici 1

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

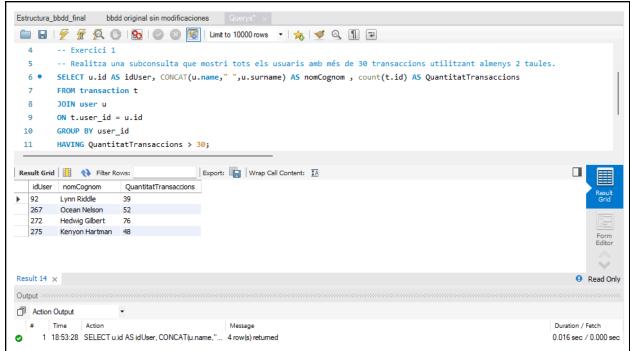


Figura 1.1.1 - Query + resultado



Exercici 2

Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.

#Tomando en cuenta todas las transacciones: transaction.declined = 1 OR 0.

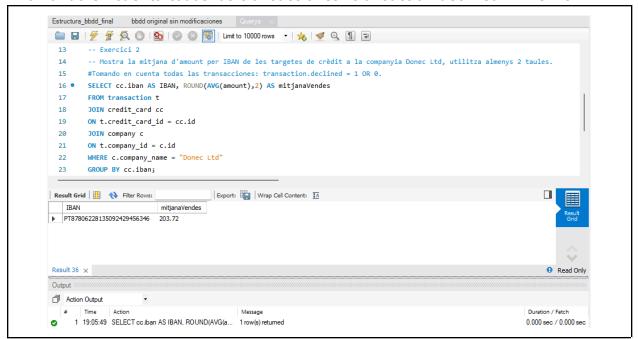


Figura 1.2.1 - Query + resultado

#Tomando en cuenta solo las ventas: transaction.declined = 0

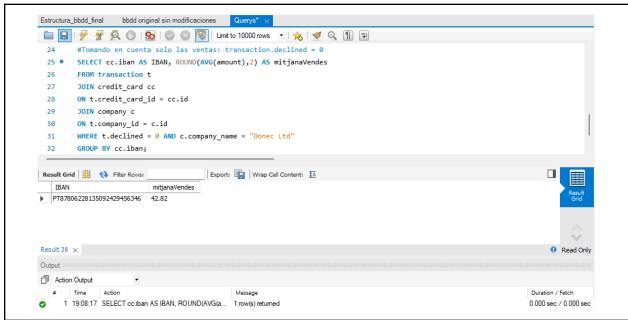


Figura 1.2.2 - Query + resultado



He après

- Una Foreign Key de una tabla no debe estar relacionada con dos o más variables de diferentes tablas. (Esto ocurría con user_id de la tabla credit card)
- SHOW VARIABLES LIKE "secure_file_priv"; Muestra cuál es la ruta donde debo ingresar las tablas.
- Evaluar cual es la separación de los .CSV en este caso eran ";"

LOAD DATA

INFILE 'C:/ProgramData/MySQL/MySQL Server

8.0/Uploads/transactions.csv'

INTO TABLE transaction

FIELDS TERMINATED BY ';'

ENCLOSED BY ""

IGNORE 1 ROWS;

- Hacer un CAST() significa asignar un tipo de dato a un valor
 - CAST(SUBSTRING_INDEX(products_id, ',', 1) AS UNSIGNED)
 (El resultado de la función SUBSTRING_INDEX la "Castea" a un entero.
- Crear una tabla a través de otra tabla:

CREATE TABLE new_tbl [AS] SELECT * FROM orig_tbl;

- Concatenar dos campos para visualizar un solo conjunto:
 - CONCAT(u.name," ",u.surname) AS nomCognom

☐He recordat

- Si quiero crear una FK que apunte a un campo que no es PK (ejemplo de transaction.order_id -> order.id puedo crear un INDICE que la el campo que no es PK (order.id)
- Oue los alias no necesitan "":
 - SELECT cc.iban AS IBAN, ROUND(AVG(amount),2) AS mitjanaVendes
- Lo imprescindible de usar CamelCase o "_" para generar los ALIAS y preguntar por ellos en un futuro.





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

Quantes targetes estan actives?

#Creación de tabla auxiliar temporal: credit_card_status_declined con transaciones rechazadas: transaction.declined = 1

#Visualizo cuantas tarjetas tienen las últimas 3 transacciones declinadas

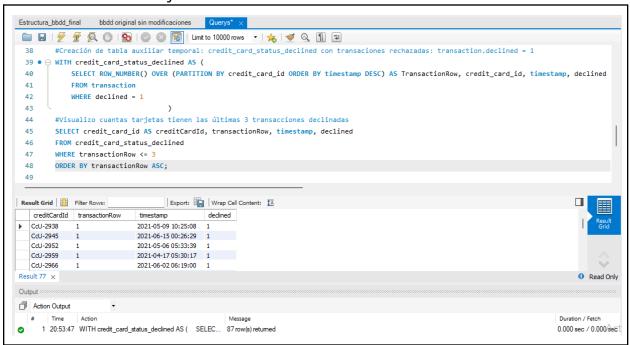


Figura 2.1.1 - CTE + query + resultado



#Creación de tabla auxiliar temporal: credit_card_status_approved con transaciones aceptadas: transaction.declined = 0 #Visualizo cuantas tarjetas están activas

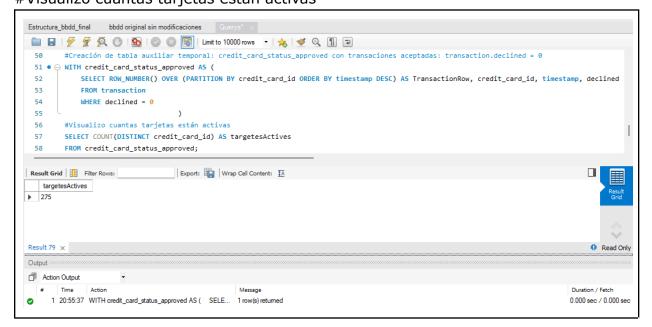


Figura 2.1.2 - CTE + query + resultado

He après

Envolver consultas en Common Table Expression (CTE) con la cláusula
 WITH, para poder aplicar filtros.





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

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

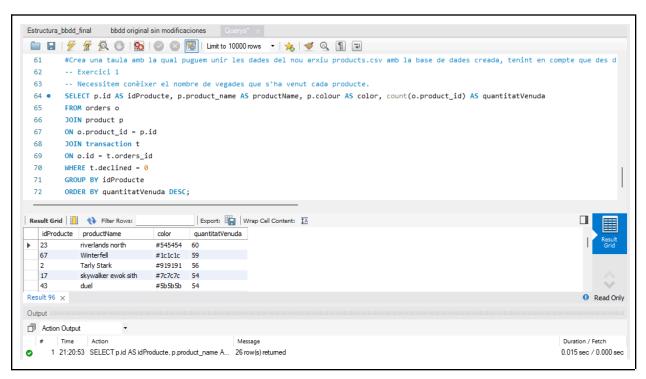


Figura 3.1.1 - Query + resultado

He recordat

 Lo importante que es agrupar por ID en lugar de por el nombre. En este ejemplo si agrupaba por nombre de producto estaba dando por hecho que los productos de distinto color eran los mismos.







companies.csv credit cards.csv products.csv transactions.csv users ca.csv users uk.csv users usa.csv

