Sprint 4

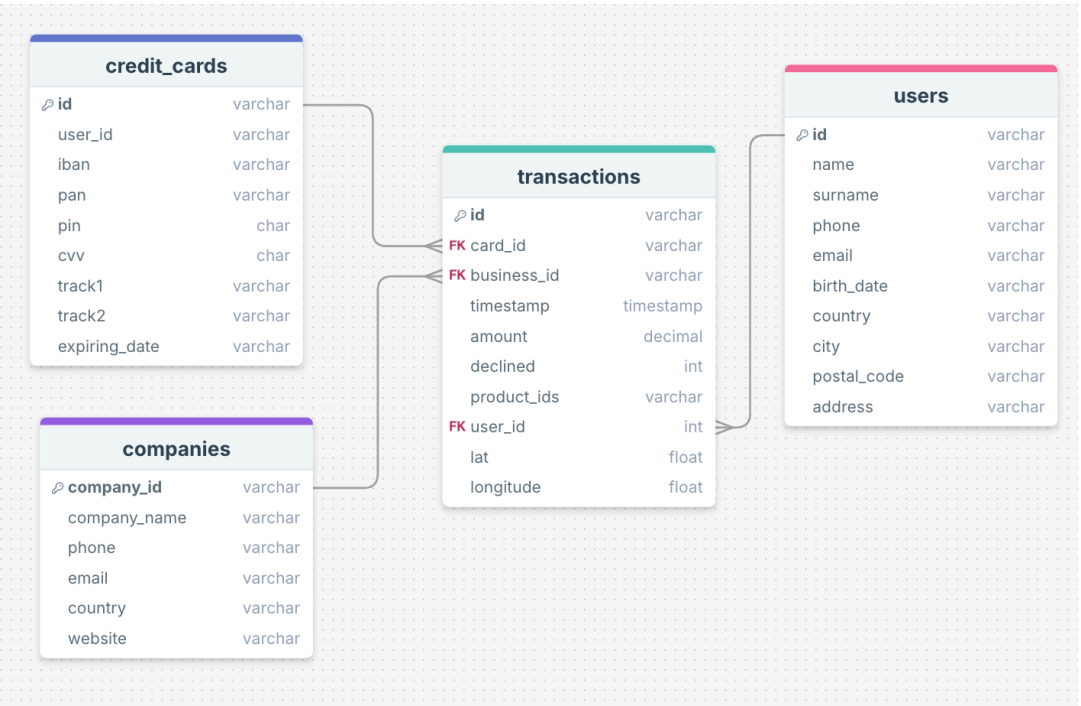
Descripció

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

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:

- Lo primero que hice fue abrir los archivos en Numbers (similar a Excel pero para Mac) para ver con qué tipo de datos estaba trabajando y poder determinar cómo va a ser la estructura de la base de datos y qué tablas la van a componer.
- Después creé la base de datos y la llamé sprint4, y procedí a crear las tablas tal y como se muestra en el script de SQL.
- Usando el Table Data Import Wizard subí los archivos con los datos de cada tabla.
- Decidí subir los archivos de users_ca, users_uk y users_usa en una sola tabla llamada users.
- Tuve problemas con el archivo CSV users_ca, ya que me daba el siguiente error al subirlo: Unhandled exception: 'ascii' codec can't decode byte 0xc3 in position 733: ordinal not in range(128). Decidí convertir el archivo a formato JSON para poder eludir el error y funcionó.
- De momento el modelo de datos ha quedado así:



- El modelo de datos tiene forma de estrella dónde ‘transactions’ es la tabla de hechos, mientras que las tablas de dimensiones están compuestas por ‘credit_cards’, ‘users’ y ‘companies’. La relación entre las tablas de dimensiones es de 1:N con la tabla de hechos.

- Exercici 1

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

- He hecho un JOIN entre las tablas users y transactions para vincular cada usuario con sus transacciones.
- Luego, he contado las transacciones de cada usuario usando la función COUNT a la cuál he dado el alias ‘num_trans’.
- He agrupado los resultados usando la función GROUP BY por id, name y surname de cada usuario.
- Finalmente, he añadido el filtro HAVING para que solo aparezcan los usuarios que tienen más de 30 transacciones en total.

```
15 •  SELECT users.id, users.name, users.surname, COUNT(transactions.id) AS num_trans
16    FROM users
17    JOIN transactions ON users.id = transactions.user_id
18    GROUP BY users.id, users.name, users.surname
19    HAVING COUNT(transactions.id) > 30;
```

100% 53:17

Result Grid | Filter Rows: Search | Export:

id	name	surname	num_trans
92	Lynn	Riddle	39
275	Kenyon	Hartman	48
272	Hedwig	Gilbert	76
267	Ocean	Nelson	52

Result 2 — Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
183	11:02:55	SELECT users.id, users.name, users.surname, COUNT(transactions.id) AS num_trans FROM users...	4 row(s) returned	0.0035 sec / 0.00002...

- Exercici 2

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

- He hecho un JOIN entre las tablas transactions, credit_cards y companies para poder relacionar cada transacción con el IBAN de la tarjeta de crédito usada en esa transacción y con la compañía involucrada en cuestión.
- Luego, he usado la función AVG para calcular el promedio de las transacciones realizadas con cada tarjeta de crédito asociada a la empresa que nos interesa.
- He usado un filtro WHERE para que la consulta solo muestre datos de la compañía Donec Ltd.
- Finalmente, he agrupado los resultados por credit_cards.iban con la función GROUP BY.

```sql
SELECT companies.company_name, credit_cards.iban, AVG(transactions.amount) AS avg_amount
FROM transactions
JOIN credit_cards ON transactions.card_id = credit_cards.id
JOIN companies ON transactions.business_id = companies.company_id
WHERE companies.company_name = 'Donec Ltd'
GROUP BY credit_cards.iban;
```

100% 28:25

Result Grid | Filter Rows: Search | Export:

company_name	iban	avg_amount
Donec Ltd	PT87806228135092429456346	203.715000

Result 10 | Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
204	10:32:29	SELECT companies.company_name, credit_cards.iban, AVG(transactions.amount) AS avg_amoun...	1 row(s) returned	0.039 sec / 0.00066...

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:

- He creado una nueva tabla llamada credit_card_status con las columnas ‘card_id’ y ‘status’.
- La primary key es ‘card_id’.
- La columna ‘status’ es un campo tipo ENUM que solo acepta los valores 'active' o ‘inactive’. Esto se ha hecho para ver el estado de las tarjetas, según lo que se me pide en los siguientes ejercicios.

```sql
CREATE TABLE IF NOT EXISTS credit_card_status (
    card_id VARCHAR(50) PRIMARY KEY,
    status ENUM('active', 'inactive') NOT NULL
);
```

100% 7:50

Action Output

	Time	Action	Response	Duration / Fetch Time
195	00:47:00	CREATE TABLE IF NOT EXISTS credit_card_status (card_id VARCHAR(50) PRIMARY KEY, sta...	0 row(s) affected	0.044 sec

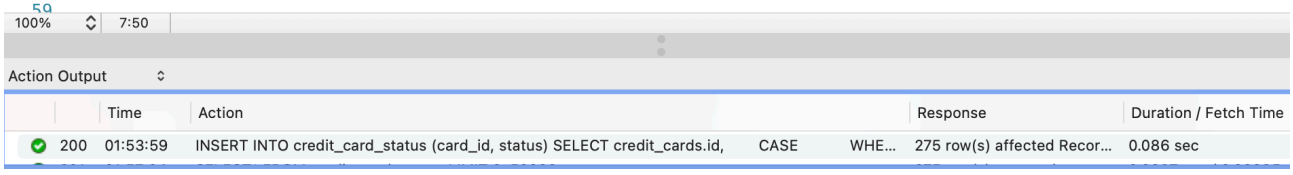
YI HUANG LAI

Exercici 1

Quantes targetes estan actives?

- Me ha costado mucho obtener la información que se pedía en el enunciado ya que inicialmente lo he intentado hacer con una subconsulta correlacionada y me aparecía el siguiente error al usar LIMIT dentro de una subconsulta con la cláusula IN: Error Code: 1235. This version of MySQL doesn't yet support 'LIMIT & IN/ALL/ANY/SOME subquery'.
- Para poder eludir este error sin tener que actualizar la versión que tengo de MySQL (ya que lo he intentado varias veces y las versiones más nuevas no son compatibles con el sistema operativo de mi ordenador), lo he tenido que hacer de una forma bastante más rebuscada.
- Primero, he realizado una subconsulta usando la función ROW_NUMBER() para asignar un número secuencial de fila a cada transacción, empezando desde 1. Este numerado se reinicia para cada tarjeta de crédito, contando cada transacción de forma independiente dentro de cada partición creada por la cláusula PARTITION BY.
- De esta forma, he ordenado las transacciones de la más reciente a la más antigua (ORDER BY timestamp DESC), ya que he podido numerar las transacciones y seleccionar solo las tres últimas para cada tarjeta (WHERE row_num <= 3).
- Luego, he sumado el valor de declined en estas tres transacciones. Si la suma es igual a 3, significa que las tres últimas transacciones fueron rechazadas (declined = 1), y en ese caso, se le asigna el valor 'inactive' a la tarjeta. Si alguna de las tres transacciones es exitosa (no rechazada), entonces el status aparece como 'active'.
- Los resultados han sido insertados en la tabla credit_card_status usando INSERT INTO.

```sql
INSERT INTO credit_card_status (card_id, status)
SELECT credit_cards.id,
       CASE
           WHEN SUM(recent_transactions.declined) = 3 THEN 'inactive'
           ELSE 'active'
       END AS status
FROM credit_cards
JOIN (
    SELECT card_id, declined
    FROM (
        SELECT card_id, declined,
               ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS row_num
        FROM transactions
    ) AS ranked_transactions
    WHERE row_num <= 3
) AS recent_transactions ON credit_cards.id = recent_transactions.card_id
GROUP BY credit_cards.id;
```



- Finalmente, para poder responder a lo que me pide el enunciado, he realizado una nueva consulta dónde he usado la función COUNT para contar la cantidad de tarjetas activas, a la cuál he dado el alias de 'active_cards'.
- Para poder filtrar solamente por las tarjetas activas he usado un filtro WHERE.

```
60 •  SELECT COUNT(*) AS active_cards
61    FROM credit_card_status
62    WHERE status = 'active';
63
```

100% 1:58

Result Grid Filter Rows: Search Export:

active_cards
275

Result 12 Read Only

ction Output

	Time	Action	Response	Duration / Fetch Time
206	12:49:50	SELECT COUNT(*) AS active_cards FROM credit_card_status WHERE status = 'active' LIMIT 0, 5...	1 row(s) returned	0.0039 sec / 0.00002...

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

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