Alumna: Milene Quiles Gonçalves

Sprint 4

Nivell 1

- Ex 1. Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.

```
3       -- EX 1.
4
5  •    SELECT id, name, surname, totaltransactions
6       FROM users_all
7  ⊖    JOIN (
8           SELECT user_id, COUNT(transactions.id) AS totaltransactions
9           FROM transactions
10          GROUP BY user_id
11      ) AS transactions_count
12      ON transactions_count.user_id = users_all.id
13      WHERE totaltransactions > 30;
14
15
16      -- EX 2.
17
```

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

- Ex 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.

La media de la suma de las transactiones por IBAN és 2.

```
16        -- EX 2.
17  ●     SELECT iban, AVG(total_transactions) as avg_transactions
18    ⊖   FROM (
19            SELECT COUNT(transactions.ID) AS total_transactions, iban
20            FROM transactions
21            JOIN company ON company.company_id = transactions.business_id
22            JOIN credit_cards ON transactions.user_id = credit_cards.user_id
23            WHERE company_name = "Donec Ltd"
24            GROUP BY iban
25        ) AS counting_transactions
26        GROUP BY iban;
27        -- la media de transaciones por iban és 2 --
```

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

iban	avg_transactions
TR30195031221357681763866 1	2.0000

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:

Creo la tabla credit_card_status usando la función case.

```
30      -- Nivell 2 --
31      -- Ex 1.
32
33 •    CREATE TABLE `credit_card_status` AS
34      SELECT card_id,
35          CASE
36              WHEN SUM(declined) >= 3 THEN 'Inativo'
37              ELSE 'Ativo'
38          END AS status
39      FROM (
40          SELECT card_id, declined,
41              ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS rn
42          FROM transactions
43      ) AS last_transactions
44      WHERE rn <= 3
45      GROUP BY card_id;
```

Ex 1. Quantes targetes estan actives?

Hay 275 tarjetas activas.

```
47 •    SELECT COUNT(Card_id) AS active_cards
48      FROM credit_card_status
49      WHERE status = "Ativo"
50      HAVING active_cards;
51      -- 275 tarjetas están activas --
52
```

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:

Ex 1. Necessitem conèixer el nombre de vegades que s'ha venut cada producte.

Creo una tabla intermedia entre products y transactions llamada products_transactions para solventar la relación many to many entre las tablas.

```sql
56 ● ⊖ CREATE TABLE products (
57           id INT,
58           product_name VARCHAR(100),
59           price VARCHAR(100),
60           colour VARCHAR(100),
61           weight VARCHAR(100),
62           warehouse_id VARCHAR(100),
63           PRIMARY KEY (id)
64    );
65
66 ● ⊖ CREATE TABLE products_transactions (
67           transaction_id VARCHAR(100) NOT NULL,
68           product_id INT NOT NULL,
69           PRIMARY KEY (product_id, transaction_id),
70           FOREIGN KEY (transaction_id) REFERENCES transactions(id),
71           FOREIGN KEY (product_id) REFERENCES products(id)
72    );
73
```

Inserto los datos de products_transactions

```sql
75 ● INSERT INTO products_transactions
76   SELECT transactions.id, SUBSTRING_INDEX(SUBSTRING_INDEX(product_ids, ',', n.digit+1), ',', -1) AS product_id
77   FROM transactions
78   JOIN
79   (SELECT 0 as digit UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4) n
80   ON LENGTH(REPLACE(product_ids, ',' , '')) <= LENGTH(product_ids)-n.digit;
81
```

A continuación, hago una query para calcular o numero de ventas de cara producto

```sql
83   SELECT COUNT(transaction_id) as total_selling, product_name
84   FROM products_transactions
85   LEFT JOIN products ON products_transactions.product_id = products.id
86   GROUP BY product_id, product_name
87   ORDER BY total_selling DESC;
88
```

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

total_selling	product_name
68	riverlands north
68	Winterfell
66	Direwolf riverlands the
65	Tarly Stark
65	duel
62	Tully
61	Direwolf Stannis
61	skywalker ewok sith
61	jinn Winterfell
60	palpatine chewbacca
58	kingsblood Littlefinger the
57	Winterfell Lannister

Result 11 ×