

Esercizi SQL del 1 Dicembre 2023

Dati i seguenti schemi di relazione (le chiavi primarie sono sottolineate, le chiavi esterne sono date esplicitamente):

- pizze(codpizza, nome, tempoprep, prezzo)
- ingredienti(codingrediente, nome, quantiamagazzino, costobase)
- ricette(codpizza*, codingrediente*, quantita)
codpizza FK(pizze)
codingrediente FK(ingredienti)
- ordini(codordine, nomecliente, indirizzocliente, oraconsegna, codpizza*)
codpizza FK(pizze)

(i) Scrivere in SQL le seguenti interrogazioni:

1. trovare il nome delle pizze in cui almeno un ingrediente ha quantiamagazzino uguale a 0;
2. trovare il nome e il prezzo delle pizze che contengono sia Mozzarella che Prosciutto;
3. trovare il numero di pizze in cui non c'è il Pomodoro;
4. per ogni pizza, trovare il nome della pizza e quante volte è stata ordinata da Mario Rossi. Se una pizza non è mai stata ordinata da Mario Rossi restituire 0. Il risultato deve essere ordinato in ordine decrescente per il numero di pizze ordinate;
5. trovare i clienti che hanno ordinato almeno tre pizze di tipo diverso;
6. per ogni cliente, trovare quante volte ha ordinato la pizza Margherita, quante volte ha ordinato la pizza Capricciosa e quante volte ha ordinato la pizza Boscaiola. Restituire gli eventuali 0;
7. per ogni cliente, trovare il numero dei suoi ordini e il numero di pizze di tipo diverso che ha ordinato. Il risultato deve essere ordinato per il numero degli ordini in senso decrescente;
8. trovare il nome e il tempo di preparazione delle pizze che contengono almeno quattro ingredienti e non hanno alcun tipo di Funghi;
9. restituire il nome della pizza più costosa;
10. per ogni pizza per cui c'è almeno un ordine e il cui prezzo è inferiore a 9 euro, trovare il nome del cliente che l'ha ordinata il massimo numero di volte e restituire anche il nome della pizza e il numero di volte che tale cliente l'ha ordinata. Il risultato deve essere ordinato rispetto al nome della pizza;
11. diminuire del 10% il prezzo delle pizze se il costo di produzione è inferiore del 50% del prezzo della pizza.
Il costo di produzione di una pizza si calcola utilizzando il costobase degli ingredienti e la quantità prevista nella ricetta;
12. eliminare le pizze che sono state ordinate da meno di 3 persone, i relativi ordini e le relative ricette;

(ii) scrivere in Algebra relazionale la prima interrogazione.

Soluzione:

1. trovare il nome delle pizze in cui almeno un ingrediente ha quantiamagazzino uguale a 0;

```
SELECT DISTINCT p.codpizza, p.nome
FROM   pizze p NATURAL JOIN ricette r JOIN ingredienti i USING (codingrediente)
WHERE  i.quantiamagazzino = 0;
```

$$\pi_{\text{codpizza}, \text{nome}}(\text{pizze} \bowtie \text{ricette} \bowtie \rho_{\text{nome} \leftarrow \text{nomeing}}(\sigma_{\text{quantiamagazzino}=0}(\text{ingredienti})))$$

2. trovare il nome e il prezzo delle pizze che contengono sia Mozzarella che Prosciutto;

```
SELECT p.nome, p.prezzo
FROM pizze p NATURAL JOIN ricette r1 JOIN ingredienti i1 USING (codingrediente)
JOIN ricette r2 USING (codpizza) JOIN ingredienti i2 ON r2.codingrediente = i2.codingrediente
WHERE i1.nome = 'Mozzarella' AND i2.nome = 'Prosciutto';
```

3. Trovare il numero di pizze in cui non c'è il Pomodoro.

```
SELECT COUNT(*)
FROM pizze p
WHERE NOT EXISTS (SELECT *
FROM ricette r NATURAL JOIN ingredienti i
WHERE i.nome = 'Pomodoro' AND r.codpizza = p.codpizza);
```

4. per ogni pizza, trovare il nome della pizza e quante volte è stata ordinata da Mario Rossi. Se una pizza non è mai stata ordinata da Mario Rossi restituire 0. Il risultato deve essere ordinato in ordine decrescente per il numero di pizze ordinate;

```
SELECT p.nome, COUNT(o.codordine) AS numvolte
FROM pizze p LEFT JOIN ordini o ON p.codpizza = o.codpizza AND o.nomecliente = 'Mario Rossi'
GROUP BY p.codpizza, p.nome
ORDER BY numvolte DESC;
```

5. trovare i clienti che hanno ordinato almeno tre pizze di tipo diverso;

```
SELECT nomecliente
FROM ordini
GROUP BY nomecliente
HAVING COUNT(DISTINCT codpizza) >= 3;
```

6. per ogni cliente, trovare quante volte ha ordinato la pizza Margherita, quante volte ha ordinato la pizza Capricciosa e quante volte ha ordinato la pizza Boscaiola. Restituire gli eventuali 0;

```
SELECT o.nomecliente,
SUM(CASE WHEN p.nome = 'Margherita' THEN 1 ELSE 0 END) AS nummarg,
SUM(CASE WHEN p.nome = 'Capricciosa' THEN 1 ELSE 0 END) AS numcap,
SUM(CASE WHEN p.nome = 'Boscaiola' THEN 1 ELSE 0 END) AS numbosc
FROM ordini o NATURAL JOIN pizze p
GROUP BY o.nomecliente;
```

7. per ogni cliente, trovare il numero dei suoi ordini e il numero di pizze di tipo diverso che ha ordinato. Il risultato deve essere ordinato per il numero degli ordini in senso decrescente;

```
SELECT nomecliente, COUNT(*) AS numOrdini,
COUNT(DISTINCT codpizza) AS numPizzeDiverse
FROM ordini
GROUP BY nomecliente
ORDER BY numOrdini DESC;
```

8. trovare il nome e il tempo di preparazione delle pizze che contengono almeno quattro ingredienti e non hanno alcun tipo di Funghi;

```

SELECT    p.nome, p.tempoprep
FROM      pizze p NATURAL JOIN ricette r
WHERE     p.codpizza NOT IN (SELECT r2.codpizza
                             FROM   ricette r2 NATURAL JOIN ingredienti i
                             WHERE  i.nome LIKE 'Funghi%')

GROUP BY  p.codpizza, p.nome, p.tempoprep
HAVING    COUNT(*) >= 4;

```

9. restituire il nome della pizza più costosa;

```

SELECT nome
FROM   pizze
WHERE  prezzo = (SELECT MAX(prezzo)
                 FROM   pizze);

```

10. per ogni pizza per cui c'è almeno un ordine e il cui prezzo è inferiore a 9 euro, trovare il nome del cliente che l'ha ordinata il massimo numero di volte e restituire anche il nome della pizza e il numero di volte che tale cliente l'ha ordinata. Il risultato deve essere ordinato rispetto al nome della pizza;

```

WITH pizzeeconomiche(codpizza, nomepizza, nomecliente, numvolte)
AS ( SELECT   pi.codpizza, pi.nome, o.nomecliente, COUNT(*)
      FROM     pizze pi NATURAL JOIN ordini o
      WHERE    pi.prezzo < 9
      GROUP BY pi.codpizza, pi.nome, o.nomecliente)

SELECT pc1.nomepizza, pc1.nomecliente, pc1.numvolte
FROM   pizzeeconomiche pc1
WHERE  pc1.numvolte = (SELECT MAX(pc2.numvolte)
                     FROM   pizzeeconomiche pc2
                     WHERE  pc2.codpizza = pc1.codpizza)

ORDER BY pc1.nomepizza;

```

11. diminuire del 10% il prezzo delle pizze se il costo di produzione è inferiore del 50% del prezzo della pizza.

Il costo di produzione di una pizza si calcola utilizzando il costobase degli ingredienti e la quantità prevista nella ricetta;

```

UPDATE pizze
SET     prezzo = prezzo * 0.9
WHERE  codpizza IN ( SELECT   p.codpizza
                    FROM     pizze p JOIN ricette r USING(codpizza)
                    JOIN     ingredienti i USING(codingrediente)
                    GROUP BY p.codpizza, p.prezzo
                    HAVING    SUM(i.costobase * r.quantita) < p.prezzo * 0.5);

```

12. eliminare le pizze che sono state ordinate da meno di 3 persone, i relativi ordini e le relative ricette.

```

DELETE FROM ordini
WHERE codpizza IN (SELECT   p.codpizza
                  FROM     pizze p NATURAL LEFT JOIN ordini o
                  GROUP BY p.codpizza
                  HAVING    COUNT(DISTINCT o.nomecliente) < 3);

```

Oppure

```
DELETE FROM ordini o
WHERE 3 > (SELECT COUNT(DISTINCT o1.nomecliente)
           FROM ordini o1
           WHERE o1.codpizza = o.codpizza);
```

Poi occorre eliminare le ricette e le pizze.

```
DELETE FROM ricette
WHERE codpizza NOT IN (SELECT codpizza
                       FROM ordini);
```

```
DELETE FROM pizze
WHERE codpizza NOT IN (SELECT codpizza
                       FROM ordini);
```