

Cognome e nome: _____ Matricola: _____ Turno: _____

Esame di SQL (rispondere su fogli separati rispetto alla parte di teoria)

Punteggi massimi:

- Domande 1 e 2 svolte perfettamente: 23;
- Domande 1 e 3 svolte perfettamente: 25;
- Domande 2 e 3 svolte perfettamente: 28;
- Domande 1, 2 e 3 svolte perfettamente: 33.

Lo svolgimento corretto di una sola domanda non permette il raggiungimento della sufficienza.

Le seguenti relazioni definiscono una base di dati “Pizzeria” per gestire gli ordini di una pizzeria. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

ORDINE(Data, Tavolo, Pizza, Quantità, Note*)

PIZZA(NomePizza, Prezzo, Tipo)

CONTIENE(Pizza, Ingrediente)

INGREDIENTE(NomeIngrediente, Vegetariano, SenzaGlutine)

Vincoli di integrità referenziale: ORDINE(Pizza) referencia PIZZA(NomePizza), CONTIENE(Pizza) referencia PIZZA(NomePizza), CONTIENE(Ingrediente) referencia INGREDIENTE(NomeIngrediente).

“Tipo” può assumere i valori "Normale" e “Gourmet”. “Vegetariano” e “SenzaGlutine” sono booleani. Data è nel formato 'YYYYMMDD'.

I rimanenti attributi sono autoesplicativi. Note può essere NULL, mentre gli altri attributi sono NOT NULL.

Con riferimento alla base di dati "Pizzeria" esprimere in SQL le seguenti interrogazioni.

Domanda 1 (bassa complessità).

Trovare gli ordini (cioè tavoli e date) con almeno tre pizze uguali che costano singolarmente più di 10€ e che hanno tra gli ingredienti la salsiccia di Bra. Ordinare il risultato per data e tavolo crescenti.

Soluzione 1.

```
SELECT DISTINCT O.Data, O.Tavolo
FROM ordine O JOIN pizza P ON (O.Pizza=P.NomePizza)
      JOIN contiene C ON (P.NomePizza=C.Pizza)
WHERE O.Quantità>=3 AND P.Prezzo>10 AND C.Ingrediente='Salsiccia di Bra'
ORDER BY O.Data, O.Tavolo;
```

Domanda 2 (media complessità).

Trovare gli ordini (cioè tavoli e date) con almeno due pizze diverse che hanno entrambe almeno un ingrediente vegetariano. Non usare operatori aggregati né clausola WITH.

Soluzione 2.

```
SELECT DISTINCT Data, Tavolo
FROM (ordine O1 JOIN pizza P1 ON (O1.Pizza=P1.NomePizza)
      JOIN contiene C1 ON (P1.NomePizza=C1.Pizza)
      JOIN ingrediente I1 ON (C1.Ingrediente=I1.NomeIngrediente))
JOIN
(ordine O2 JOIN pizza P2 ON (O2.Pizza=P2.NomePizza)
      JOIN contiene C2 ON (P2.NomePizza=C2.Pizza)
      JOIN ingrediente I2 ON (C2.Ingrediente=I2.NomeIngrediente))
ON (O1.Tavolo=O2.Tavolo AND O1.Data=O2.Data AND O1.Pizza<>O2.Pizza)
WHERE I1.Vegetariano=True AND I2.Vegetariano=True;
```

Domanda 3 (alta complessità).

Trovare la spesa totale per l'anno 2023 degli ordini composti esclusivamente da pizze in cui tutti gli ingredienti sono senza glutine.

Soluzione 3.

```
WITH ContiSenzaGlutine AS (  
    SELECT O.Data, O.Tavolo, SUM(P.Prezzo*O.Quantità) Conto  
    FROM ordine O JOIN pizza P ON (O.Pizza=P.NomePizza)  
    WHERE (O.Data, O.Tavolo) NOT IN (  
        SELECT O1.Data, O1.Tavolo  
        FROM ordine O1 JOIN pizza P1 ON (O1.Pizza=P1.NomePizza)  
        JOIN contiene C ON (P1.NomePizza=C.Pizza)  
        JOIN ingrediente I ON (C.Ingrediente=I.NomeIngrediente)  
        WHERE I.SenzaGlutine=False)  
    GROUP BY O.Data, O.Tavolo)  
SELECT SUM(C.Conto)  
FROM ContiSenzaGlutine C  
WHERE C.Data>='20230101' AND C.Data<='20231231';
```

Esame di Teoria (rispondere su fogli separati rispetto alla parte di SQL)

Domanda 1 (9 punti).

Con riferimento alla base di dati “Pizzeria”:

- A. (4 punti) Esprimere in Algebra Relazionale l'interrogazione
Elencare i tavoli che hanno ordinato (complessivamente, non necessariamente all'interno dello stesso ordine) tutte le pizze possibili.
- B. (5 punti) Esprimere, nel calcolo dei predicati su tuple con dichiarazione di range, la seguente domanda:
Elencare le pizze (si richiede nome e prezzo) di tipo gourmet che hanno esclusivamente ingredienti vegetariani.

Soluzione 1.

A. Una possibile soluzione (in algebra relazionale) è la seguente:

$$\pi_{Tavolo, Pizza}(Ordine) \div \rho_{Pizza \leftarrow NomePizza}(\pi_{NomePizza}(Pizza))$$

B. Una possibile soluzione è la seguente:

$$\{p.NomePizza, p.Prezzo \mid p(PIZZA) \mid p.Tipo='Gourmet' \wedge \forall c(CONTIENE)(c.Pizza=p.NomePizza \Rightarrow \exists i(INGREDIENTE)(c.Ingrediente=i.NomeIngrediente \wedge i.Vegetariano=True))\}$$

Domanda 2 (9 punti).

- A. Fornire la definizione di dipendenza funzionale.
Si considerino $R(A, B, C, D, E)$ e $F' = \{B \rightarrow D, D \rightarrow E\}$.
- B. Le dipendenze funzionali in F' non giustificano A come chiave di R : aggiungere a F' delle dipendenze funzionali in modo che A sia chiave di R .
- C. Considerando il nuovo insieme di dipendenze funzionali ottenuto nel punto precedente, dire se R è in 3FN e se non lo è decomporla in relazioni in 3FN esplicitando tutti i passaggi. Lo schema ottenuto è BCNF?

Soluzione 2.

- A. Vedere il libro di testo o le slide.
- B. Se aggiungiamo la d.f. $A \rightarrow BC$, la chiusura di A comprende tutti gli attributi di R quindi è una superchiave ed essendo composta da un solo attributo è minimale ed è una chiave.
- C. $R(A, B, C, D, E)$ e $F = \{A \rightarrow BC, B \rightarrow D, D \rightarrow E\}$ non è 3NF perché, ad esempio, nella d.f. $B \rightarrow D$ B non è una superchiave, D non è un attributo primo e la d.f. è non banale.

Calcolo l'insieme di copertura minimale.

Trasformo le d.f. in forma canonica: $F'' = \{A \rightarrow B, A \rightarrow C, B \rightarrow D, D \rightarrow E\}$.

Non ci sono attributi estranei perché ogni d.f. ha un solo attributo a sinistra.

Ogni d.f. ha a destra un attributo differente, quindi ogni d.f. è necessaria per avere quell'attributo nella chiusura e non ci sono d.f. ridondanti. F'' , quindi, è una copertura minimale.

Decompongo R in $R_1(\underline{A}, B, C)$, $R_2(\underline{B}, D)$, $R_3(\underline{D}, E)$. Non ci sono relazioni sottoinsieme di altre e R_1 comprende la chiave di R quindi ho terminato la normalizzazione in 3NF.

Lo schema ottenuto è BCNF perché ogni d.f. è rappresentata tramite la chiave primaria delle rispettive relazioni.

Domanda 3 (7 punti).

A. Spiegare in modo succinto ma preciso le principali differenze tra B-tree e B+-tree e quali vantaggi apportano i B+-tree rispetto ai B-tree nella gestione degli indici.

Soluzione 3.

A. Vedere il libro di testo o le slide.

Domanda 4 (8 punti).

A. Enunciare il criterio di serializzabilità.

B. Dire, giustificando la risposta, se la seguente storia è compatibile con il protocollo 2PL (lock a due fasi):

$S = r1(x), r2(y), w2(x), r3(y), w3(x), w1(z).$

Soluzione 4.

A. Vedere il libro di testo o le slide.

B. È compatibile con 2PL e una possibile storia è la seguente:

$S = LSI(x), LX1(z), r1(x), LS2(y), r2(y), UN1(x), LX2(x), w2(x), UN2(x), LS3(y), r3(y), LX3(x), w3(x), w1(z), UN2(y), UN3(y), UN1(z).$