

Cognome e nome: _____ Matricola: _____ Turno: _____

Riportare sui fogli i seguenti dati: cognome, nome, matricola e turno di laboratorio.

Esame di SQL

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 “**Incubi di cucina**” per gestire le comande (o ordinazioni) di un ristorante. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

STAFF(Matricola, Cognome, Nome)

PIATTO(Data, Descrizione, Prezzo, Tipo*)

COMANDA(Staff, Tavolo, Data, Descrizione, Quantità, Commenti*)

COMANDA(Staff) referencia STAFF(Matricola),

COMANDA(Data, Descrizione) referencia PIATTO(Data, Descrizione).

STAFF descrive il personale, PIATTO i piatti che compongono il menù del giorno e COMANDA le ordinazioni. PIATTO(Data) è una stringa nel formato ‘YYYYMMDD’.

PIATTO(Tipo) assume i valori NULL, ‘Vegetariano’, ‘Vegano’, ‘Crudista’.

COMANDA(Tavolo) indica quale tavolo effettua l’ordinazione. COMANDA(Commenti) contiene eventuali osservazioni sul piatto ordinato (per semplicità si consideri il testo tutto minuscolo) e può assumere valore nullo. Gli altri attributi sono autoesplicativi.

Domanda 1 (bassa complessità).

Elencare, senza duplicati e dalla più remota alla più recente, le date del 2020 in cui è stato ordinato almeno un piatto non vegano o di tipo sconosciuto.

Soluzione 1.

```
SELECT DISTINCT p.Data
FROM Piatto p JOIN Comanda c ON (p.Data=c.Data AND p.Descrizione=c.Descrizione)
WHERE (p.Tipo<>'Vegano' OR p.Tipo IS NULL) AND p.Data>='20200101' AND
p.Data<='20201231'
ORDER BY p.Data;
```

Domanda 2 (media complessità).

Considerando i camerieri che hanno preso comande per un totale di almeno 1000€ di piatti vegetariani, elencare il nome e il cognome di tali camerieri, il numero di tali comande e la quantità totale di piatti ordinati in tali comande. NON usare la clausola WITH.

Soluzione 2.

```
SELECT s.Nome, s.Cognome, count(*), SUM(c.Quantità)
FROM Staff s JOIN Comanda c ON (s.Matricola=c.Staff) JOIN Piatto p ON
(p.Data=c.Data AND p.Descrizione=c.Descrizione)
WHERE p.Tipo='Vegetariano'
GROUP BY s.Matricola
HAVING SUM(c.Quantità*p.Prezzo)>=1000;
```

Domanda 3 (alta complessità).

Nelle date (escluse le date dell'anno 2020, che non devono essere mostrate) in cui erano a menù esclusivamente piatti vegetariani, mostrare per ogni data il numero di tavoli per cui nel commento alla comanda è compresa la parola “bistecca”.

Soluzione 3.

```
WITH SoloVeganiNo2020 AS (  
    SELECT Data  
    FROM Piatto p  
    WHERE (Data<'20200101' OR Data>'20203112') AND Data NOT IN (  
        SELECT Data  
        FROM Piatto p  
        WHERE Tipo<>'Vegetariano' OR Tipo IS NULL)  
    )  
SELECT c.Data, COUNT(DISTINCT c.Tavolo)  
FROM SoloVeganiNo2020 s JOIN Comanda c ON (s.Data=c.Data)  
WHERE c.Commenti LIKE '%bistecca%'  
GROUP BY c.Data;
```

Esame di Teoria

Domanda 1 (9 punti).

Con riferimento alla base di dati “**Incubi di cucina**”:

A. (5 punti) Esprimere in Algebra Relazionale l’interrogazione:

Elencare le date e i tavoli per cui, nella stessa giornata, il tavolo è stato servito da almeno due camerieri diversi.

B. (4 punti) Esprimere, nel calcolo relazionale su tuple con dichiarazione di range, la seguente interrogazione:

Elencare il nome e il cognome dei camerieri per cui ogni ordine che hanno preso è consistito di almeno tre piatti crudisti.

Soluzione 1.

A.

$$\pi_{Tavolo}(\rho_{C1 \leftarrow COMANDA}(Comanda) \bowtie_{C1.Data=C2.Data \wedge C1.Tavolo=C2.Tavolo \wedge C1.Staff \neq C2.Staff} \rho_{C2 \leftarrow COMANDA}(Comanda))$$

B.

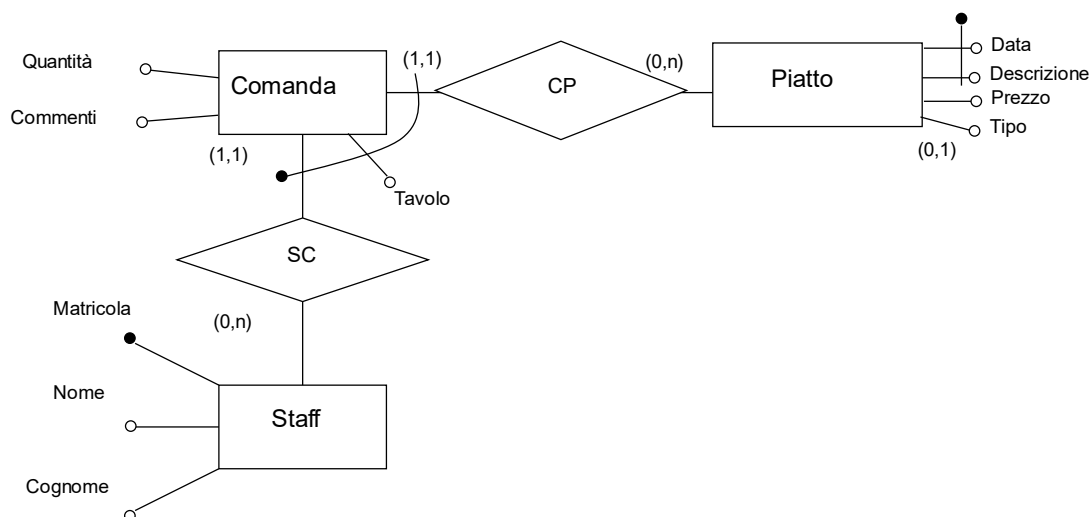
$$\{s.Nome, s.Cognome \mid s(Staff) \mid \forall c(Comanda)(c.Staff = s.Matricola \Rightarrow (c.Quantità \geq 3 \wedge \exists p(Piatto)(p.Data = c.Data \wedge p.Descrizione = c.Descrizione \wedge p.Tipo = 'Crudista')))\}$$

Domanda 2 (6 punti).

Disegnare uno schema Entity-Relationship che potrebbe avere generato lo schema logico della base di dati “**Incubi di cucina**”.

Soluzione 2.

Una possibile soluzione è la seguente:



Domanda 3 (6 punti).

Data una relazione $R(A,B,C,D,E,F,G)$ e l’insieme di dipendenze funzionali $F = \{AB \rightarrow EG, B \rightarrow F, A \rightarrow CDF, C \rightarrow B, E \rightarrow G\}$, decomporre in 3NF mostrando i vari passaggi e dire, motivando, se il risultato è anche BCNF.

Soluzione 3.

Ricavo le chiavi. Dato che A non è nei conseguenti, deve fare parte di ogni chiave. $A^+ = \{A \mid CDF \mid B \mid EG\}$, quindi A è superchiave e, essendo un attributo unico, è necessariamente minimale, quindi è l’unica chiave.

La forma canonica di F è $F' = \{AB \rightarrow E, AB \rightarrow G, B \rightarrow F, A \rightarrow C, A \rightarrow D, A \rightarrow F, C \rightarrow B, E \rightarrow G\}$.

Elimino gli attributi estranei. Dato che A è chiave, nelle prime due d.f. sicuramente B è estraneo. Le altre d.f. hanno un solo attributo a sinistra, che non può essere estraneo, quindi $F'' = \{A \rightarrow E, A \rightarrow G, B \rightarrow F, A \rightarrow C, A \rightarrow D, A \rightarrow F, C \rightarrow B, E \rightarrow G\}$.

Elimino le d.f. ridondanti. $A \rightarrow G$ è ottenibile da $A \rightarrow E$ e $E \rightarrow G$, quindi è ridondante e $F''' = \{A \rightarrow E, B \rightarrow F, A \rightarrow C, A \rightarrow D, A \rightarrow F, C \rightarrow B, E \rightarrow G\}$. Inoltre, $A \rightarrow F$ è ottenibile da $A \rightarrow C$, $C \rightarrow B$ e $B \rightarrow F$, quindi è ridondante e $F'''' = \{A \rightarrow E, B \rightarrow F, A \rightarrow C, A \rightarrow D, C \rightarrow B, E \rightarrow G\}$, che è l'insieme di copertura minimale.

Decomponendo ottengo:

$R1(\underline{A}, C, D, E)$

$R2(\underline{B}, F)$

$R3(\underline{C}, B)$

$R4(\underline{E}, G)$

Nessuna relazione è sottoinsieme di un'altra e la chiave di R è già presente in R1.

La decomposizione è BCNF perché ogni relazione ha solo la d.f. implicita di chiave primaria, che è di tipo superchiave.

Domanda 4 (6 punti).

Con riferimento alla base di dati “**Incubi di cucina**” e i seguenti dati quantitativi:

$CARD(Staff) = 10$

$CARD(Comanda) = 10\,000$

Nel ristorante ci sono 30 tavoli, numerati da 1 a 30

$VAL(Descrizione, Comanda) = 100$

disegnare gli alberi sintattici prima e dopo l'ottimizzazione logica e calcolare il numero di tuple “mosse” prima e dopo l'ottimizzazione logica della seguente query:

$$\sigma_{Tavolo \leq 3 \wedge Matricola = '001' \wedge Descrizione = 'Uovo Sodo'}(Staff \bowtie_{Matricola=Staff} Comanda)$$

Soluzione 4.

La query ottimizzata dividendo la selezione e portandola verso le foglie è:

$$\sigma_{Matricola = '001'}(Staff) \bowtie_{Matricola=Staff} \sigma_{Tavolo \leq 3 \wedge Tipo = 'Uovo Sodo'}(Comanda)$$

Prima dell'ottimizzazione:

- Costo $r1 = (Staff \bowtie_{Matricola=Staff} Comanda)$: $CARD(Staff) \cdot CARD(Comanda) = 10 \cdot 10\,000 = 10^5$.
- Cardinalità $|r1| = CARD(Comanda) = 10^4$ (equijoin attraverso la chiave esterna)
- Costo della selezione: $|r1| = 10^4$
- Costo totale = $10^5 + 10^4 \sim 10^5$.

Dopo l'ottimizzazione:

- Costo $\sigma_1 = \sigma_{Matricola = '001'}(Staff) = CARD(Staff) = 10$
- Costo $\sigma_2 = \sigma_{Tavolo \leq 3 \wedge Descrizione = 'Uovo Sodo'}(Comanda) = CARD(Comanda) = 10^4$
- Tuple prodotte dalla selezione $|\sigma_1| = 1$ (selezione sulla chiave primaria)
- Tuple prodotte dalla selezione $|\sigma_2| = f_{tavolo \leq 3} \cdot f_{Descrizione = 'Uovo Sodo'} \cdot CARD(Comanda) = 3/30 \cdot 1/100 \cdot 10\,000 = 10$
- Costo join $r = \sigma_1 \bowtie_{Matricola=Staff} \sigma_2 = 1 \cdot 10 = 10$.
- Costo totale = $10 + 10^4 + 10 \sim 10^4$.

Domanda 5 (6 punti).

Si consideri il seguente file di log:

```
<T1, START>
<T1, BS(t1[B], 9), AS(t1[B], 11)>
<T2, START>
<T2, BS(t2[A], 11), AS(t2[A], 19)>
<T3, START>
<T3, BS(t3[A], 1), AS(t3[A], 2)>
<T2, ABORT>
<T1, COMMIT>
– CHECKPOINT –
```

Scrivere il contenuto del record di checkpoint e descrivere quali operazioni vengono eseguite per ripristinare il sistema se il crash avviene subito dopo la scrittura del checkpoint.

Soluzione 5.

Il record di checkpoint contiene le transazioni attive al momento del checkpoint con un puntatore al loro inizio:

<cp | T3, puntatore a <T3, START> | OK>

Il ripristino è una ripresa a caldo (hot restart) che considera il checkpoint per ricavare la lista AT delle transazioni attive, cioè T3, mentre la lista CT delle transazioni committed dopo il checkpoint in questo caso è vuota. Non c'è bisogno di considerare le transazioni committed e aborted prima del checkpoint.

Per la ripresa è sufficiente effettuare quindi UNDO(T3).