

Esame di SQL

Le seguenti relazioni definiscono una base di dati “**Prenotazioni**” per gestire le prenotazioni di camere in un hotel di Torino. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

CLIENTE(NumTessera, Nome, DataNascita, CittàResidenza)

CAMERA(Num, Piano, NumPostiLetto, Tipo, Prezzo)

PRENOTA(Cliente, Num, Piano, DataArrivo, DataPartenza)

Vincoli di integrità referenziali: PRENOTA(Cliente) referencia CLIENTE(NumTessera) e PRENOTA(Num,Piano) referencia CAMERA(Num,Piano).

Significato degli attributi: “NumTessera” è il numero tessera del cliente accreditato; “Tipo” può assumere i valori “economica”, “normale” e “suite”; “Prezzo” è il prezzo di una notte in euro. I rimanenti attributi sono autoesplicativi. Gli attributi sono tutti NOT NULL.

Domanda 1 (bassa complessità).

Con riferimento alla base di dati "Prenotazioni" esprimere in SQL la seguente interrogazione:

Trovare i clienti che hanno almeno due prenotazioni di camere con un numero diverso di posti letto.

Soluzione 1.

```
SELECT DISTINCT p1.Cliente
FROM prenota p1 JOIN prenota p2 ON (p1.Cliente=p2.Cliente)
      JOIN camera c1 ON (p1.Num=c1.Num AND p1.Piano=c1.Piano)
      JOIN camera c2 ON (p2.Num=c2.Num AND p2.Piano=c2.Piano)
WHERE c1.NumPostiLetto<>c2.NumPostiLetto;
```

Domanda 2 (media complessità).

Con riferimento alla base di dati "Prenotazioni" esprimere in SQL la seguente interrogazione:

Mostrare i clienti che pagano, a notte, un prezzo maggiore del prezzo medio pagato, a notte, dai clienti provenienti dalla loro stessa città. Non usare la clausola WITH.

Soluzione 2.

```
SELECT DISTINCT c1.NumTessera, c1.Nome, c1.CittàResidenza
FROM cliente c1 JOIN prenota p ON (c1.Cliente=p.NumTessera)
      JOIN camera ca ON (p.Num=ca.Num AND p.Piano=ca.Piano)
WHERE ca.Prezzo > (SELECT AVG(Prezzo)
                  FROM cliente c11 JOIN prenota p1 ON (c11.Cliente=p1.NumTessera)
                  JOIN camera ca1 ON (ca1.Num=p1.Num AND ca1.Piano=p1.Piano)
                  WHERE c11.CittàResidenza=c1.CittàResidenza);
```

Domanda 3 (alta complessità).

Con riferimento alla base di dati "Prenotazioni" esprimere in SQL la seguente interrogazione:

Trovare, SENZA utilizzare gli operatori insiemistici, quanti clienti hanno sempre e soltanto prenotato camere di tipo economico.

Soluzione 3.

```
SELECT COUNT(DISTINCT Cliente)
FROM prenota
WHERE Cliente NOT IN
      (SELECT p.Cliente
       FROM prenota p JOIN camera c ON (p.Num=c.Num AND p.Piano=c.Piano)
       WHERE c.Tipo<>'economica');
```

Esame di Teoria

Domanda 1.

Con riferimento alla base di dati “**Prenotazioni**”:

A. Esprimere in Algebra Relazionale l'interrogazione

Elencare i clienti che hanno prenotato in ogni piano in cui è presente una suite.

B. Esprimere, nel calcolo relazionale su tuple con dichiarazione di range, la seguente domanda:

Elencare le camere con un solo posto letto che sono in overbooking. Si mostri Num e Piano.

(VERSIONE 1: Una camera in overbooking è una camera assegnata a clienti diversi con stessa data di arrivo).

(VERSIONE 2: Una camera in overbooking è una camera assegnata a clienti diversi per periodi che coincidono anche solo in parte).

Soluzione 1.

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

$$R(A,B) \div S(B)$$

dove

$$R(A,B) = \pi_{\text{Cliente, Piano}}(\text{PRENOTA})$$
$$S(B) = \pi_{\text{Piano}}(\sigma_{\text{Tipo}='suite'}(\text{CAMERA}))$$

B. Una possibile soluzione è la seguente:

VERSIONE 1:

$$\{p.\text{Num}, p.\text{Piano} \mid p(\text{PRENOTA}) \mid \exists c(\text{CAMERA})(c.\text{NumPostiLetto}=1 \wedge p.\text{Num}=c.\text{Num} \wedge p.\text{Piano}=c.\text{Piano} \wedge \\ \exists p'(\text{PRENOTA}) (p.\text{Num}=p'.\text{Num} \wedge p.\text{Piano}=p'.\text{Piano} \wedge p.\text{Cliente} \neq p'.\text{Cliente} \wedge \\ p.\text{DataArrivo}=p'.\text{DataArrivo}))\}$$

VERSIONE 2:

$$\{p.\text{Num}, p.\text{Piano} \mid p(\text{PRENOTA}) \mid \exists c(\text{CAMERA})(c.\text{NumPostiLetto}=1 \wedge p.\text{Num}=c.\text{Num} \wedge p.\text{Piano}=c.\text{Piano} \wedge \\ \exists p'(\text{PRENOTA}) (p.\text{Num}=p'.\text{Num} \wedge p.\text{Piano}=p'.\text{Piano} \wedge p.\text{Cliente} \neq p'.\text{Cliente} \wedge \\ (p.\text{DataArrivo} \leq p'.\text{DataArrivo} < p.\text{DataPartenza} \vee p'.\text{DataArrivo} \leq p.\text{DataArrivo} < p'.\text{DataPartenza}))\}$$

Domanda 2.

A. Dare la definizione di insieme di copertura minimale.

B. Dati: $R(L, M, N, O, P, Q, R)$ e $F = \{ O \rightarrow MR, N \rightarrow LM, MN \rightarrow OP, R \rightarrow NO \}$

dire se R è in 3FN e se non lo è decomporla in relazioni in 3FN esplicitando tutti i passaggi. Il risultato è BCNF?

Soluzione 2.

A. Si vedano gli appunti/testo/slide.

B. Per prima cosa è necessario identificare le chiavi candidate. In questo caso abbiamo tre chiavi $K1=\{O\}$, $K2=\{N\}$, $K3=\{R\}$ (il calcolo della chiusura lo dimostra).

La relazione R è già in 3FN e in BCNF, in quanto tutte le dipendenze funzionali sono del tipo “superchiave”. Non è necessario decomporre.

Domanda 3.

Considerare la seguente storia interfogliata $S = r1(x), r1(y), r2(x), r3(y), w1(y), r2(y), w2(x)$
 S è compatibile con il protocollo 2PL? Giustificare la risposta.

Soluzione 3.

È compatibile con 2PL perché, per esempio, è possibile aggiungere i lock ottenendo la storia seguente, compatibile con 2PL:

$S = LS1(x), r1(x), LS1(y), r1(y), LS2(x), r2(x), LS3(y), r3(y), UN3(y), LX1(y), w1(y), UN1(x), UN1(y), LS2(y), r2(y), LX2(x), w2(x), UN2(y), UN2(x)$

Domanda 4.

Si consideri la base di dati col seguente schema:

AUTORE(Nome, Nazionalità, Qualifica)

ARTICOLO(Titolo, Anno, Conferenza)

PUBBLICAZIONE(TitoloArticolo, NomeAutore)

dove TitoloArticolo e NomeAutore in PUBBLICAZIONE sono in vincolo di chiave esterna, rispettivamente, con ARTICOLO e AUTORE.

e i seguenti dati quantitativi:

CARD(pubblicazione) = 100 000

VAL(NomeAutore, pubblicazione) = 2 000

CARD(articolo) = 6 000

VAL(Nazionalità, autore) = 150

MIN(Anno, articolo) = 1972

MAX(Anno, articolo) = 2022

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:

$$\sigma_{(Anno \geq 2002 \wedge Anno \leq 2022) \wedge NomeAutore = 'Alan Turing'}(articolo \bowtie_{Titolo = TitoloArticolo} pubblicazione)$$

Soluzione 4.

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

$$\sigma_{Anno \geq 2002 \wedge Anno \leq 2022}(articolo) \bowtie_{Titolo = TitoloArticolo} \sigma_{NomeAutore = 'Alan Turing'}(pubblicazione)$$

Prima dell'ottimizzazione:

- Costo $r1 = articolo \bowtie_{Titolo = TitoloArticolo} pubblicazione$: $6 \cdot 10^3 \cdot 10^5 = 6 \cdot 10^8$.
- Cardinalità $|r1| = CARD(pubblicazione) = 10^5$ (equijoin attraverso la chiave esterna).
- Costo della selezione $= |r1| = 10^5$.
- Costo totale $= 6 \cdot 10^8 + 10^5 \cong 6 \cdot 10^8$.

Dopo l'ottimizzazione:

- Costo $\sigma_1 = \sigma_{Anno \geq 2002 \wedge Anno \leq 2022}(articolo) = CARD(articolo) = 6 \cdot 10^3$.
- Tuple prodotte dalla selezione $|\sigma_1| = (2022 - 2002) / (MAX(Anno, articolo) - MIN(Anno, articolo)) \cdot CARD(articolo) = (2022 - 2002) / (2022 - 1972) \cdot 6 \cdot 10^3 = 20/50 \cdot 6 \cdot 10^3 = 2,4 \cdot 10^3$
- Costo $\sigma_2 = \sigma_{NomeAutore = 'Alan Turing'}(pubblicazione) = CARD(pubblicazione) = 10^5$.
- Tuple prodotte dalla selezione $|\sigma_2| = 1 / VAL(NomeAutore, pubblicazione) \cdot CARD(pubblicazione) = \frac{1}{2000} \cdot 10^5 = 0,5 \cdot 10^2$.
- Costo join $r = \sigma_1 \bowtie_{Titolo = TitoloArticolo} \sigma_2 = 2,4 \cdot 10^3 \cdot 0,5 \cdot 10^2 = 1,2 \cdot 10^5$.
- Costo totale $= 6 \cdot 10^3 + 10^5 + 1,2 \cdot 10^5 \cong 2 \cdot 10^5$.