

BD: prima prova di verifica del 2/11/2011 — Soluzioni

1. (Obbligatorio) Si considerino le relazioni $S(D :string, E :int)$ e $R(A :string, B :string, C :int)$:

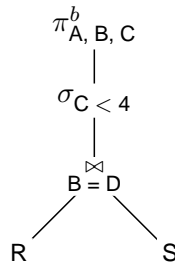
S	
D	E
b1	2
b2	4
b3	6

R		
A	B	C
a1	b2	1
a1	b2	2
a2	b2	2
a1	b3	3
a2	b3	4

Si diano (a) l'albero logico della seguente interrogazione, (b) il tipo del risultato e (c) il valore del risultato:

```
SELECT A, B, C
FROM S, R
WHERE D = B AND C < 4;
```

Albero logico



Tipo risultato: $\{(A :string, B :string, C :int)\}$

A	B	C
a1	b2	1
a1	b2	2
a2	b2	2
a1	b3	3

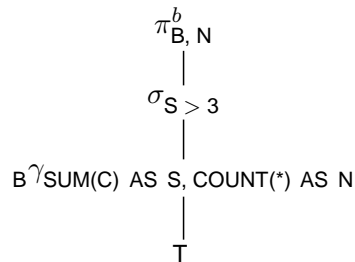
Sia T il risultato della precedente interrogazione. Si diano (a) l'albero logico della seguente interrogazione, (b) il tipo del risultato e (c) il valore del risultato:

```

SELECT    B, COUNT(*) AS N
FROM      T
GROUP BY  B
HAVING    SUM(C) > 3;

```

Albero logico



Tipo risultato: $\{(B : \text{string}, N : \text{int})\}$

B	N
b2	3

2. Si vuole progettare una base di dati per la gestione delle prenotazioni alberghiere di una stazione turistica.

Degli alberghi interessano il nome (unico) il telefono (unico), l'indirizzo (città, via, cap), la categoria (numero di stelle fra 1 e 5) e le modalità di pagamento accettate (bancomat, nome delle carte di credito).

Le camere di un albergo hanno un numero (unico per albergo), un tipo (singola, da due a quattro letti, o matrimoniali) e una restrizione (per fumatori o non fumatori).

Della prenotazione di un cliente interessano un numero progressivo che la identifica, la data di arrivo e di partenza, e la camera associata alla prenotazione. Una prenotazione prevede l'arrivo entro le 12 del primo giorno oppure deve essere garantita. Nel primo caso, se il cliente non arriva per le 12, la prenotazione viene cancellata senza penalità. Per avere una prenotazione valida dopo le 12, essa deve essere garantita da una carta di credito prevista dall'albergo, della quale interessano il nome (express, visa, ecc), il numero e la data di scadenza (la carta deve essere valida per tutto il periodo di soggiorno).

Di un cliente interessano il codice fiscale (unico), il nome, l'indirizzo e la email. Quando il cliente arriva all'albergo, si registra l'ora e la data di arrivo. Dal momento della registrazione, alla camera assegnata vengono addebitate le spese sostenute (ad. es. bar, pranzi, colazione, lavanderia).

I clienti chiudono il conto il giorno della partenza e si registra l'ora e la data della partenza.

- (a) Si definisca lo schema concettuale della della base di dati, indicando in modo testuale i vincoli non descritti graficamente (Figura 1).¹

Vincoli non descritti graficamente: (a) le camere hanno un numero unico per albergo, (b) una prenotazione può essere garantita da una carta di credito prevista dall'albergo.

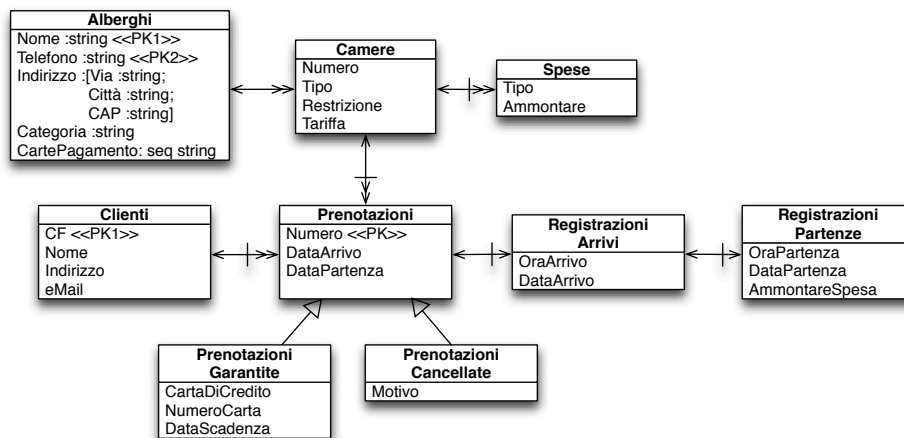


Figura 1: Schema concettuale

- (b) Si traduca lo schema concettuale in uno schema relazionale grafico, definendo gli attributi delle chiavi primarie e delle chiavi esterne (Figura 2).

¹È da considerarsi corretta anche una soluzione che riporti invece le informazioni su arrivo e partenza nelle prenotazioni, o in una sottoclasse. Per quanto riguarda le spese, è da considerarsi corretta una soluzione che le associ alla registrazione di arrivo anziché alla camera.

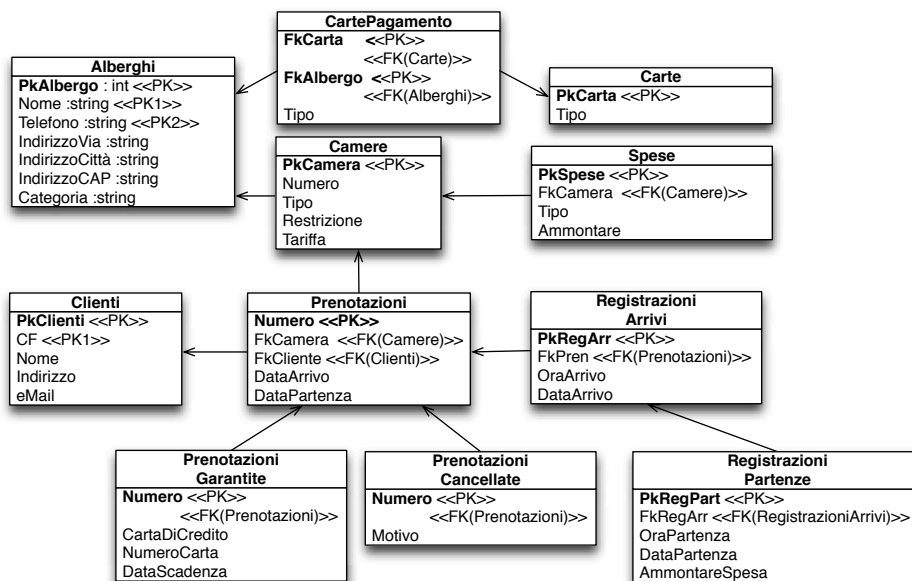


Figura 2: Schema Logico

3. Si consideri il seguente schema relazionale che riguarda rappresentazioni di opere teatrali:

Opere(CodOp, Titolo, Autore, Lingua, Nazione, AnnoComposizione)
 Rappresentazioni(CodR, CodOp*, Regista, CodTeatro*, Data, Anno)
 RappresentazioniAttori(CodA*, CodR*, ÈProtagonista)
 Attori(CodA, Nome, AnnoNascita, Nazione)
 Teatri(CodTeatro, Nome, Città)

Scrivere le interrogazioni SQL che restituiscono le seguenti informazioni:

(a) Il nome di tutti gli attori che hanno recitato nel 2009

```

SELECT DISTINCT A.CodA, d.Nome
FROM Attori A, RappresentazioniAttori RA, Rappresentazioni R
WHERE A.CodA = RA.CodA AND RA.CodR = R.CodR
AND R.Anno > 2009;
  
```

(b) Per ogni attore, il nome ed il numero di rappresentazioni di opere di Goldoni in cui ha recitato

```

SELECT    A.Nome, COUNT(*) AS NumeroRappresentazioniGoldoni
FROM      Attori A, RappresentazioniAttori RA, Rappresentazioni R, Opere O
WHERE     A.CodA = RA.CodA AND RA.CodR = R.CodR AND R.CodOp = O.CodOp
          AND O.Autore = 'Goldoni'
GROUP BY  A.CodA, A.Nome;

```

- (c) Per ogni teatro e per ogni anno, il nome del teatro, la città, l'anno, ed il numero di rappresentazioni che sono avvenute in quell'anno ed in quel teatro, a patto che tale numero sia superiore a 10.

```

SELECT    T.Nome, T.Città, , R.Anno, COUNT(*) AS NumeroRappresentazioni
FROM      Teatri T, Rappresentazioni R
WHERE     T.CodTeatro = R.CodTeatro
GROUP BY  T.CodTeatro, T.Nome, T.Città, R.Anno
HAVING    COUNT(*) > 10;

```

- (d) Il nome di tutti gli attori che hanno recitato solo nel 2009 (assumendo che ogni attore abbia recitato in almeno una rappresentazione):

```

SELECT    A.Nome
FROM      Attori A
WHERE     FOR ALL RA IN RappresentazioniAttori, R IN Rappresentazioni
          WHERE A.CodA = RA.CodA AND RA.CodR = R.CodR
          : r.Anno = 2009;

```

```

SELECT    A.Nome
FROM      Attori A
WHERE     NOT EXISTS(
          SELECT *
          FROM  RappresentazioniAttori RA, Rappresentazioni R
          WHERE A.CodA = RA.CodA AND RA.CodR = R.CodR
          AND NOT(r.Anno = 2009) );

```

- (e) Il nome di tutti gli attori che, nel 2009, hanno recitato solo da protagonisti (assumendo che tutti gli attori abbiano recitato almeno una volta nel 2009)

```

SELECT    A.Nome
FROM      Attori A
WHERE     FOR ALL RA IN RappresentazioniAttori, R IN Rappresentazioni
          WHERE A.CodA = RA.CodA AND RA.CodR = R.CodR AND R.Anno = 2009
          : RA.ÈProtagonista;

```

```

SELECT A.Nome
FROM Attori A
WHERE NOT EXISTS(
  SELECT *
  FROM RappresentazioniAttori RA, Rappresentazioni R
  WHERE A.CodA = RA.CodA AND RA.CodR = R.CodR AND R.Anno = 2009
  AND NOT(RA.ÈProtagonista));

```

(f) (Opzionale) Per ogni attore, il nome ed il numero di opere di Goldoni che ha recitato

```

SELECT A.Nome, COUNT(DISTINCT O.CodOp) AS NumOpereGoldoni
FROM Attori A, RappresentazioniAttori RA, Rappresentazioni R, Opere O
WHERE A.CodA = RA.CodA AND RA.CodR = R.CodR AND R.CodOp = O.CodOp
AND O.Autore = 'Goldoni'
GROUP BY A.CodA, A.Nome;

```

oppure

```

SELECT A.Nome, COUNT(*) AS NumOpereGoldoni
FROM Attori A, Opere O
WHERE O.Autore = 'Goldoni' AND EXISTS
  (SELECT *
  FROM RappresentazioniAttori RA, Rappresentazioni R
  WHERE A.CodA = RA.CodA AND RA.CodR = R.CodR AND R.CodOp = O.CodOp)
GROUP BY A.CodA, A.Nome;

```

oppure

```

SELECT A.Nome, COUNT(*) AS NumOpereGoldoni
FROM Attori A, Opere O
WHERE O.Autore = 'Goldoni' AND CodA IN
  (SELECT CodA
  FROM RappresentazioniAttori RA, Rappresentazioni R
  WHERE A.CodA = RA.CodA AND RA.CodR = R.CodR AND R.CodOp = O.CodOp)
GROUP BY A.CodA, A.Nome;

```