

10 Prova del 04/07/2019

Schema base di dati

Si consideri il seguente schema relazionale (chiavi primarie sottolineate):

NEGOZI(codice, nome, città) --città deve essere selezionabile da una tabella
PRODOTTI(codice, nome, marca) --marca deve essere selezionabile da una tabella
LISTINO(negozio, prodotto, prezzo)

con vincoli di integrità (oltre a quelli indirettamente indicati nei commenti) fra gli attributi di LISTINO e le chiavi di NEGOZI e PRODOTTI.

Domanda 1 [7 punti]

- 1) Indicare i 4 vincoli di integrità usando la notazione '→'.
- 2) Scrivere il codice PostgreSQL che generi TUTTE le tabelle per rappresentare lo schema relazionale specificando tutti gli attributi, almeno 2 controlli di correttezza dei dati usando il CHECK, MA SENZA I vincoli di integrità referenziale.
- 3) Scrivere tutti i comandi ALTER TABLE che aggiungono SOLO tutti i vincoli di integrità referenziale.

Domanda 2 [7 punti]

Scrivere il codice PostgreSQL che determina le soluzioni alle seguenti due interrogazioni usando il minor numero di JOIN e NON usando il DATA BINDING:

- (a) Per ciascuna città, determinare i prodotti con prezzo maggiore (nella città) visualizzando il nome della città, il codice del negozio dove viene venduto il prodotto, il codice e nome del prodotto, e il suo prezzo.
- (b) Trovare i prodotti che vengono venduti in una sola città mostrando codice e nome prodotto e il nome della città, ordinati per nome città.

Domanda 3 [7 punti]

Data una base di dati PostgreSQL che contenga le tabelle e viste della Domanda 1, scrivere un programma Python che, leggendo il codice di un prodotto da console, visualizzi la città risultato della query dell'esercizio 2(b) (adattata al fatto che si fornisce il codice del prodotto). In particolare, se il prodotto non è venduto o venduto in più città, il programma deve segnalarlo. Se, invece, viene venduto solo in una città, il programma deve visualizzare la città. Se il codice del prodotto non esiste, il programma deve segnalarlo e richiedere il dato.

Domanda 4 [6 punti]

Si scrivano due transazioni in cui: 1) nella prima transazione si aumenta del 5% il prezzo a tutti i prodotti dei negozi della città Verona, mentre 2) nell'altra si abbassa il prezzo del 10% a tutti i farmaci che hanno prezzo maggiore o uguale a 1050 euro. Che tipo di errore si può verificare se le due transazioni sono concorrenti? Che tipo di isolamento si deve dichiarare per eliminare l'errore? In quale transazione?

Domanda 5 [5 punti]

Si consideri il seguente risultato del comando ANALYZE su una query inerenti a 3 tabelle:

```
GroupAggregate (cost=8994..9232 ROWS=7709)
  GROUP KEY: i.nomeins
  Filter: (COUNT(DISTINCT d.id_persona) > 2)
  -> Sort (cost=8994..9026 ROWS=12927)
    Sort KEY: i.nomeins
    -> Hash JOIN (cost=6474..8111 ROWS=12927)
      Hash Cond: (ie.id_insegn = i.id)
      -> Hash JOIN (cost=6186..7645 ROWS=12927)
        Hash Cond: (d.id_inserogato = ie.id)
        -> Seq Scan ON docenza d (cost=0..1139
        ROWS=50867)
      -> Hash (cost=5970..5970 ROWS=17285)
        -> Seq Scan ON inserogato ie
        (cost=0..5970 ROWS=17285)
      Filter:
      ((annoaccademico)::TEXT > '2010/2011'::TEXT)
      -> Hash (cost=185..185 ROWS=8169)
        -> Seq Scan ON insegn i (cost=0..185
        ROWS=8169)
```

```
GroupAggregate (cost=8770.13..9009.12 ROWS=7709)
  GROUP KEY: i.nomeins
  Filter: (COUNT(DISTINCT d.id_persona) > 2)
  -> Sort (cost=8770..8802 ROWS=12952)
    Sort KEY: i.nomeins
    -> Hash JOIN (cost=6247..7885 ROWS=12952)
      Hash Cond: (ie.id_insegn = i.id)
      -> Hash JOIN (cost=5959..7419 ROWS=12952)
        Hash Cond: (d.id_inserogato = ie.id)
        -> Seq Scan ON docenza d (cost=0..1139
        ROWS=50867)
      -> Hash (cost=5743..5743 ROWS=17319)
        -> Bitmap Heap Scan ON inserogato
        ie (cost=406..5743 ROWS=17319)
          Recheck Cond:
          ((annoaccademico)::TEXT > '2010/2011'::TEXT)
          -> Bitmap INDEX Scan ON i1
          (cost=0..402 ROWS=17319)
          INDEX Cond:
          ((annoaccademico)::TEXT > '2010/2011'::TEXT)
          -> Hash (cost=185..185 ROWS=8169)
            -> Seq Scan ON insegn i (cost=0..185
            ROWS=8169)
```

A sinistra la query originale, a destra la query dopo la creazione di uno o più indici.



Desumere il testo della query e scrivere il codice del/degli indici creati.

Suggerimento: La clausola SELECT ha una parte che si può desumere perché molto probabile, una parte no.

*Scrivere solo la parte che si può desumere senza aggiungere * o altro!*