

Prova scritta del 16 Febbraio 2018

Regole

Non è ammesso nessun supporto o ausilio come appunti, schemi, manuali, personal computer o smartphone per lo svolgimento della prova. Rispondere alle domande in ordine usando uno o più fogli formato A4 distribuiti dal docente. Ciascun foglio deve riportare in alto a sinistra il proprio nome, cognome e numero di matricola. Consegnare solo i fogli che contengono le risposte, non i fogli usati per prove o brutte copie.

Se non si raggiunge il punteggio di 13 con le prime tre domande, la prova non è superata e le rimanenti domande non vengono valutate.

Schema base di dati

Si consideri il seguente schema relazionale parziale e semplificato (chiavi primarie sottolineate) contenente le informazioni relative alla programmazione di convegni:

1 CONVEGNO(nome, dataInizio, dataFine, numeroSessioni, tipo, luogo);
4 INTERVENTO_IN_CONVEGNO(nomeConvegno, idIntervento, nomeSessione, orarioInizio);
3 SESSIONE(nome, nomeConvegno, data, orarioInizio, orarioFine);

2 INTERVENTO(id, titolo, relatore, durata);

dove numeroSessioni è un intero, tipo può assumere valori {seminario, simposio, conferenza}, luogo è generico (VARCHAR è sufficiente), INTERVENTO_IN_CONVEGNO.orarioInizio è l'orario di inizio dell'intervento all'interno della sessione. Gli attributi inerenti al tempo devono avere il fuso orario.

Domanda 1 [5 punti]

Scrivere il codice PostgreSQL che generi i domini e le tabelle per rappresentare lo schema relazionale con i controlli di integrità e di correttezza dei dati opportuni.

Domanda 2 [6 punti]

Dato il nome 'X' di un convegno, scrivere una query che visualizzi il programma del convegno. Il programma è costituito da un elenco ordinato per giorno e ora degli interventi: sessione, titoloIntervento, relatore e orario inizio intervento.

Domanda 3 [8 punti]

Assumendo di avere una base di dati PostgreSQL che contenga le tabelle di questo tema d'esame, scrivere un programma Python che, leggendo uno o più tuple del tipo 'nomeConvegno', 'nomeSessione', 'idIntervento', 'orarioInizio' da console, inserisca una o più tuple nella tabella INTERVENTO_IN_CONVEGNO. L'inserimento deve garantire la correttezza della base di dati: controllo preventivo che le eventuali dipendenze siano rispettate e che l'orario dell'intervento sia compatibile con la sua durata e la presenza di altri interventi. Se una dipendenza non è rispettata, il programma deve richiedere di reinserire il dato associato alla dipendenza prima di procedere a inserire la tupla. Il programma deve visualizzare l'esito di ogni singolo inserimento. È richiesto che il programma suggerisca il tipo di dati da inserire, che non ammetta possibilità di SQL Injection e che sia eseguibile in concorrenza con altre istanze del programma stesso.

Domanda 4 [7 punti]

Scrivere il codice PostgreSQL, definendo anche eventuali viste, per rispondere alle seguenti due interrogazioni nel modo più efficace:

- Trovare per ogni convegno il numero totale e la durata totale degli interventi per ciascuna sessione di convegno. Il risultato deve visualizzare nome convegno, giorno e i conteggi richiesti.
- Trovare per ogni convegno distribuito su almeno 3 giorni il numero totale di interventi e la durata totale degli interventi per ciascun giorno di convegno. Il risultato deve visualizzare nome convegno, giorno e i conteggi richiesti.

Domanda 5 [7 punti]

Si consideri il seguente piano di esecuzione:

```
HashAggregate (cost=1859.67..1866.24 ROWS=657 width=60)
  GROUP KEY: i.nomeins, d.descrizione
  -> Hash JOIN (cost=319.54..1856.39 ROWS=657 width=60)
    Hash Cond: (ie.id_insegn = i.id)
    -> Hash JOIN (cost=31.74..1559.55 ROWS=657 width=25)
      Hash Cond: (ie.id_discriminante = d.id)
      -> Bitmap Heap Scan ON interrogato ie (cost=26.68..1543.97 ROWS=1053 width=8)
        Recheck Cond: (((annoaccademico)::TEXT = '2009/2010'::TEXT) AND (crediti = ANY
        ('{3,5,12}'::NUMERIC))) AND (modulo = '0'::NUMERIC)
        -> Bitmap INDEX Scan ON ins_aa (cost=0.00..26.42 ROWS=1053 width=0)
          INDEX Cond: (((annoaccademico)::TEXT = '2009/2010'::TEXT) AND (crediti = ANY
          ('{3,5,12}'::NUMERIC))) AND (modulo = '0'::NUMERIC)
          -> Hash (cost=3.36..3.36 ROWS=136 width=25)
            -> Seq Scan ON discriminante d (cost=0.00..3.36 ROWS=136 width=25)
            -> Hash (cost=185.69..185.69 ROWS=8169 width=43)
              -> Seq Scan ON insegn i (cost=0.00..185.69 ROWS=8169 width=43)
```

- Indicare quanti e quali indici sono stati usati per risolvere la query. Per indicare quali sono gli indici, scrivere il codice SQL per crearli.
- Supponendo che i dati non varino nel tempo e che le chiavi primarie hanno indice, in base al piano di esecuzione conviene creare degli altri indici? Se sì, quali?