

Matricola: _____

Cognome: _____

Nome: _____

Basi di Dati

Prova intermedia del giugno 2015

Avvertenze: è severamente vietato consultare libri e appunti.

Durata 2h

DOMANDE TEORIA (è obbligatorio rispondere ad almeno una domanda delle quattro di seguito elencate)

- a) (3) Illustrare l'architettura di un DMBS descrivendo in particolare il modulo di gestione dei buffer; si indichi inoltre, per ogni modulo dell'architettura, quali sono le proprietà delle transazioni che contribuisce a garantire.
- b) (2) Si presenti in dettaglio la definizione di Conflict-Serializzabilità (CSR).
- c) (2) Lo studente illustri la struttura di accesso ai dati denominata indice primario denso: caratteristiche della struttura, ricerca, inserimento e cancellazione di entry dall'indice.
- ~~d) (3) Lo studente illustri l'algoritmo di codifica di Huffman e mostri un esempio di codifica di un messaggio di lunghezza 4 sull'alfabeto {Z,W,Y,X}.~~
- d) (3) Domanda aggiuntiva di teoria (Transazioni, DBMS, Concorrenza, Indici, Ottimizzazione, XML)

ESERCIZI

Esecuzione concorrente

Dato il seguente schedule S:

- (2) indicare se è conflict-SR oppure no (calcolare l'insieme dei conflitti),
- (3) se non è CSR verificare se è view-SR oppure non-SR (giustificare dettagliatamente la risposta).

S: $r_0(t), r_2(z), r_3(z), w_1(z), r_3(x), r_2(x), w_3(x), w_3(y), w_2(y), w_0(y), w_1(t)$

Ottimizzazione

Si consideri il seguente schema relazionale contenente le ricette di una catena di ristoranti:

INGREDIENTE(Codice, Nome, Calorie);

COMPOSIZIONE(Ricetta, Ingrediente, Quantità)

RICETTA(CodiceRicetta, Nome, Regione, TempoPreparazione)

Nota: la quantità nella tabella COMPOSIZIONE è espressa in grammi

Vincoli di integrità: COMPOSIZIONE.Ricetta \rightarrow RICETTA, COMPOSIZIONE.Ingrediente \rightarrow INGREDIENTE

Formulare in SQL la seguente interrogazione:

(1) Trovare gli ingredienti usati in ricette della Regione Veneto, riportando, il codice della ricetta e il nome e le calorie dell'ingrediente.

(4) Calcolare il costo dell'interrogazione in termini di numero di accessi a memoria secondaria sotto le seguenti ipotesi:

- la selezione su ricetta richiede una scansione sequenziale della tabella RICETTA
- l'ordine di esecuzione del join è RICETTA \bowtie COMPOSIZIONE \bowtie INGREDIENTE
- le operazioni di join vengono eseguite con la tecnica "Nested Loop Join" con una pagina di buffer disponibile per ogni tabella
- NP(INGREDIENTE) = 40, NP(COMPOSIZIONE) = 200, NP(RICETTA) = 12
- NR(INGREDIENTE) = 1200, NR(COMPOSIZIONE) = 11000, NR(RICETTA) = 260
- VAL(Regione, RICETTA) = 20

(2) Come cambia il costo se è disponibile un indice B+-tree sull'attributo Codice della tabella INGREDIENTE che occupa 2 pagine di memoria secondaria.

SQL (è obbligatorio svolgere almeno due esercizi di questa sezione)

Si consideri il seguente schema relazionale contenente i dati concernenti le ricette di un ristorante:

INGREDIENTE(Codice, Nome, Calorie, Grassi, Proteine, Carboidrati);

COMPOSIZIONE(Ricetta, Ingrediente, Quantità)

RICETTA(CodiceRicetta, Nome, Regione, TempoPreparazione)

Nota: la quantità di grassi, proteine e carboidrati è in grammi su 100 grammi di ingrediente; la quantità nella tabella COMPOSIZIONE è espressa in grammi
Vincoli di integrità: COMPOSIZIONE.Ricetta → RICETTA, COMPOSIZIONE.Ingrediente → INGREDIENTE

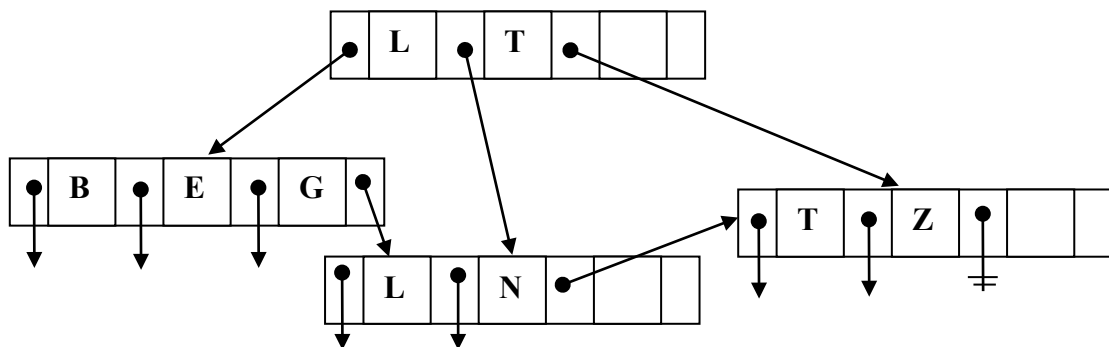
Formulare in SQL le seguenti interrogazioni (definire viste solo dove è necessario):

- (3) SQL.1 Trovare il nome e il tempo di preparazione delle ricette della Regione Liguria che contengono almeno un ingrediente con più del 30% di grassi e almeno un ingrediente con meno del 10% di proteine.
- (3) SQL.2 Trovare per ogni ricetta la quantità totale (in grammi) di proteine e carboidrati, riportando oltre alle quantità richieste anche il nome della ricetta.
- (3) SQL.3 Trovare gli ingredienti usati solo in ricette della Regione Veneto, riportando il nome e le calorie dell'ingrediente.
- (2) SQL.4 Trovare per ogni ricetta con tempo di preparazione maggiore di 60 minuti l'elenco degli ingredienti ordinati in senso crescente per nome della ricetta e decrescente per quantità; nel risultato riportare il nome della ricetta e tutti gli attributi della tabella ingrediente.

B⁺-tree

Dato il seguente B⁺-tree (fan-out=4), mostrare lo stato dell'albero:

- (2) dopo la cancellazione del valore chiave L e
- (3) dopo l'inserimento del valore chiave D supponendo di partire dall'albero ottenuto al punto a).



XML

- (6) Dato il seguente file XML e i seguenti requisiti si produca il file XML schema (XSD) che ne descrive la sintassi.

```
<?xml version="1.0"?>
<agenzia xmlns="http://www.agenzia.org"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.agenzia.org
                              agenzia.xsd">

  <edificio id="E002" >
    <proprietario>
      <nome>Mario</nome>
      <cognome>Rossi</cognome>
    </proprietario>
    <tipo>A3</tipo>
    <prezzoOfferto timestamp=" 2014-11-30 T 11:45:22">100000</PrezzoOfferto>
    <prezzoOfferto timestamp=" 2015-01-20 T 11:45:22">110000</PrezzoOfferto>
  </edificio>
  <edificio id="E0023" >
    <proprietario>
      <nome>Mario</nome>
      <cognome>Rossi</cognome>
    </proprietario>
    <proprietario>
      <nome>Giovanni</nome>
      <cognome>Bianchi</cognome>
    </proprietario>
  </edificio>
</agenzia>
```

```

        </proprietario>
        <tipo>A1</tipo>
        <descrizione>Descrizione edificio</descrizione>
    </edificio>

```

...

</agenzia>

Requisiti

L'attributo `id` e l'attributo `timestamp` sono obbligatori. L'elemento `tipo` può assumere solo i valori: A1, A2, A3.

XMLSchema `agenzia.xsd` da completare

```

<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
...
...
>
<xsd:complexType name="T_Proprietario">
    <xsd:sequence>
        <xsd:element ... />
        <xsd:element ... />
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="T_TipoEdificio">
    <xsd:restriction base="xsd:string">
        ...

    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="T_PrezzoOfferto">
    ...

</xsd:complexType>

<xsd:element name="edificio">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ... />
            <xsd:element ... />
            <xsd:element ... />
            <xsd:element ... />
        </xsd:sequence>
        <xsd:attribute ... />
    </xsd:complexType>
</xsd:element>
<xsd:element name="agenzia">
    <xsd:complexType><xsd:sequence>
        ...
    </xsd:sequence></xsd:complexType></xsd:element>
</xsd:schema>

```