

Matricola: _____
 Cognome: _____
 Nome: _____
 Insegnamento: Basi di dati ☐
 Basi di dati e Web ☐

Basi di Dati

Prova scritta del 19 settembre 2012

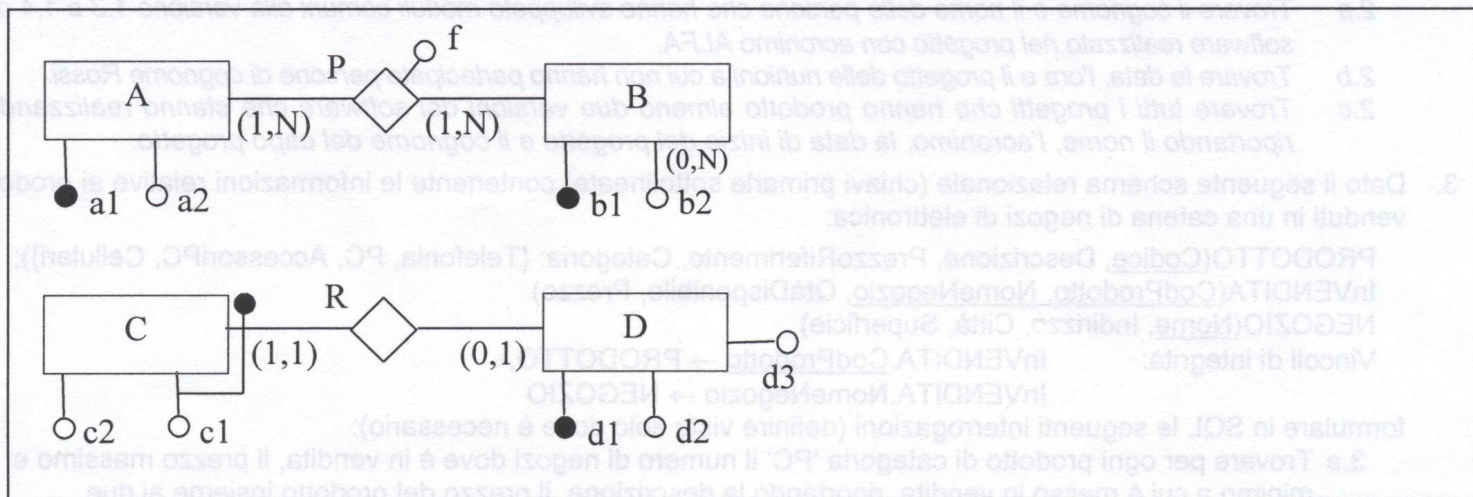
Durata 2h30m

Avvertenze: e' severamente vietato consultare libri e appunti.

DOMANDE PRELIMINARI (è necessario rispondere in modo sufficiente alle seguenti tre domande per poter superare la prova scritta con esito positivo; in caso di mancata o errata risposta a queste domande il resto del compito non verrà corretto)

a) Si illustri il costrutto di relazione del modello Entità-Relazioni

b) Dato il seguente schema concettuale nel modello ER, si produca la sua traduzione nel modello relazionale



c) Date le due seguenti relazioni: R1(A, B, C) e R2(D, E, F, G) (tutti gli attributi sono di tipo numerico) scrivere;

- un'espressione in algebra relazionale che restituisca l'insieme dei valori distinti contenuti nell'attributo A di R1 o nell'attributo G di R2;
- un'espressione ottimizzata dell'algebra relazionale che contenga un join naturale e una selezione su R2 e produca come risultato le tuple t di R2 tali che $t[E] > t[F]$ e tali che esista una tupla t' di R1 dove $t[D] = t'[C]$ (non sono ammesse altre selezioni oltre a quella su R2).

Punteggi esercizi: (1) 12 – (2.a,2.b, 2.c) 3 – (3a, 3.b) 3 – (4) 3 – (5) 3.

1. Si vuole progettare un sistema informativo relativo alla gestione dell'attività di un'azienda che produce software.

L'azienda organizza il proprio lavoro in progetti. Ogni progetto è caratterizzato da: codice univoco, un nome, un acronimo, data di inizio del progetto, eventuale data di inizio dell'implementazione ed eventuale data di conclusione del progetto; inoltre si registra lo stato del progetto: studio di fattibilità, progettazione iniziale, progettazione nuova versione, implementazione, test e rilascio.

Ogni progetto produce diverse versioni del software. Ogni versione è caratterizzata da una data di rilascio e da un numero di versione (1.0, 1.4, 2.0, ...) ed è suddivisa in moduli. Ogni modulo viene identificato da un nome e una versione (ad esempio, il modulo Main 1.3 può far parte della versione 1.3 e 1.5 del software). Si registra inoltre per ogni versione del software lo schema concettuale dei dati salvato in un file di cui si memorizza il path.

Viene inoltre gestita l'allocazione del personale ai progetti, registrando per ogni progetto le persone che vi partecipano; nell'ambito di un progetto un dipendente assume il ruolo di capo progetto. Nel sistema ogni dipendente è caratterizzato da: matricola univoca, nome, cognome, data di nascita, data di assunzione, eventuale data di fine contratto e ruolo. Per i lavoratori a tempo determinato si memorizza anche la durata in mesi del contratto e l'importo del contratto. Per ogni modulo software si indicano inoltre gli sviluppatori del codice. Infine il sistema memorizza le varie riunioni delle persone coinvolte nel progetto; ogni riunione è identificata dal progetto e dalla data e ora della riunione. Si registrano inoltre le persone che vi partecipano e il tema della riunione.

Progettare lo schema concettuale utilizzando il modello entità-relazione e lo schema relazionale della base di dati (indicare esplicitamente per ogni relazione dello schema relazionale: le chiavi primarie, gli attributi che possono contenere valori nulli e i vincoli di integrità referenziale). Non aggiungere attributi non esplicitamente indicati nel testo.

2. Dato lo schema relazionale dell'esercizio 1, esprimere in algebra relazionale ottimizzata le seguenti interrogazioni:

- Trovare il cognome e il nome delle persone che hanno sviluppato moduli comuni alla versione 1.3 e 1.4 del software realizzato nel progetto con acronimo ALFA.
- Trovare la data, l'ora e il progetto delle riunioni a cui non hanno partecipato persone di cognome Rossi.
- Trovare tutti i progetti che hanno prodotto almeno due versioni del software che stanno realizzando, riportando il nome, l'acronimo, la data di inizio del progetto e il cognome del capo progetto.

3. Dato il seguente schema relazionale (chiavi primarie sottolineate) contenente le informazioni relative ai prodotti venduti in una catena di negozi di elettronica:

PRODOTTO(Codice, Descrizione, PrezzoRiferimento, Categoria: {Telefonia, PC, AccessoriPC, Cellulari});

InVENDITA(CodProdotto, NomeNegozio, QtàDisponibile, Prezzo)

NEGOZIO(Nome, Indirizzo, Città, Superficie)

Vincoli di integrità: InVENDITA.CodProdotto → PRODOTTO,

InVENDITA.NomeNegozio → NEGOZIO

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

3.a Trovare per ogni prodotto di categoria 'PC' il numero di negozi dove è in vendita, il prezzo massimo e minimo a cui è messo in vendita, riportando la descrizione, il prezzo del prodotto insieme ai due conteggi richiesti.

3.b Trovare il nome e la superficie dei negozi di Vicenza che non vendono prodotti di categoria 'Telefonia'.

4. Data la seguente lista di valori (W,G,K,E,P,A,C,D,F,I,R): (a) costruire un possibile B⁺-tree (fan-out=4) che contenga tutti i valori della lista ed almeno due nodi foglia con riempimento massimo; (b) mostrare l'albero dopo l'inserimento dei valori B e Z.

5. Dato il seguente frammento XML-schema generare una possibile istanza XML dell'elemento *Appello* conforme allo schema e contenente almeno 3 registrazioni. Si noti che l'elemento *Registrazione* contiene come valore la matricola dello studente.

```
<xsd:element name="Registrazione">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="voto" type="VotoType" use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:simpleType name="VotoType">
  <xsd:restriction base="xsd:nonNegativeInteger">
    <xsd:minInclusive value="18"/>
    <xsd:maxInclusive value="30"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
<xsd:element name="Appello">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Data" type="xsd:date" />
      <xsd:element name="Insegnamento" type="xsd:string" />
      <xsd:element name="Docente" type="xsd:string" />
      <xsd:element ref="VotoRegistrato"
        maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```