



Politecnico di Milano

Facoltà di Ingegneria Civile, Ambientale e Territoriale

Informatica (ICA-LC) [091456] – Informatica A (ICA-LC) [091461]

Informatica [097256]

Prof. G. Boracchi

Allievi Ingegneria Civile e Ambientale

Il prova in itinere (5 Febbraio 2016)

<i>Cognome e nome</i>	
<i>Matricola</i>	
<i>Firma</i>	

Domanda	1	2	3	4	5	TOT
Punteggio max	4	6	8	10	4	32
Punteggio						

La **durata** della **prova** è di **2 ore**. Non è consentito consultare libri o appunti, non è consentito l'uso di calcolatrici.

Scrivere solo sui fogli distribuiti utilizzando il retro delle pagine in caso di necessità e cancellando le parti di brutta con un tratto di penna. Non separare questi fogli.

Per tutti gli esercizi non è sufficiente fornire il risultato, ma **è necessario mostrare il procedimento seguito. Non dilungarsi però in spiegazioni prolisse.**

Gli allievi sono invitati a privilegiare **chiarezza, proprietà di linguaggio e sinteticità** nelle risposte agli esercizi, con l'obiettivo di **dimostrare la loro conoscenza degli argomenti.**

Domanda 1. Il Sistema Operativo (4 punti).

Si considerino i seguenti sistemi tutti dotati di parole di 1 Byte.

sistema A dotato di memoria 4GB di memoria RAM con tempo di accesso medio di 200 ns

sistema B dotato di uno spazio di indirizzamento fisico a 30 bit e di 2 MB di memoria cache.

Il sistema è caratterizzato da HitRate 70% , HitTime 80ns, MissPenalty 400ns

Quale dei due sistemi dispone di maggiore memoria fisica? Motivare la risposta mostrando i calcoli

Quale sistema è dotato del minor tempo di accesso alla memoria? Motivare la risposta mostrando i calcoli

Si scriva una funzione Matlab *calcolaTempoMedio* che calcola del tempo medio di accesso alla memoria per un sistema dotato di cache e RAM. Si scriva quindi l'invocazione della funzione *calcolaTempoMedio* per calcolare il tempo medio di accesso dei sistemi A e B.

SOLUZIONE

Il sistema B ha meno memoria fisica perchè il suo spazio di indirizzamento permette di raggiungere 1GB che è 2^{30}

Il sistema B ha tempo di accesso medio pari a 176ns e quindi è migliore. Il calcolo è spiegato direttamente nella funzione Matlab

`function TDAMedio = calcolaTDAMedio(HitRate, HitTime, MissPenalty)`

$TDAMedio = HitRate * HitTime + (1 - HitRate) * MissPenalty;$

TDAMedioA = calcolaTDAMedio(0, 0, 200)
TDAMedioB = calcolaTDAMedio(0.7, 80, 400)

[per la prova di recupero]

Si assuma ora che

- *ogni MB di cache che aggiungo al sistema B comporta un aumento dell'HitRate del 5%. Un MB di cache per il sistema B costa 30Euro*
- *ogni MB di cache che aggiungo al sistema A comporta un aumento dell'HitRate del 3%. Un MB di cache per il sistema A costa 20Euro*

Domanda 2. Matlab: interpretazione del codice (6 punti).

Si consideri la seguente funzione Matlab

```
function out = funzRic(arg1, arg2)
    if(arg2 > 0)
        out = arg1 * funzRic(arg1, arg2 -1);
    else
        out = 1;
    end
```

sia k una variabile inizializzata con l'ultima cifra del vostro codice di matricola (se il vostro numero di matricola termina in 0 o 1 , mettete k = 2).

Si dica cosa viene visualizzato a schermo

disp(k)

funzRic(k, 0)

funzRic(k, 1)

funzRic(k, 2)

funzRic(k, -1)

Quante chiamate ricorsive vengono eseguite nella chiamata funzRic(7, 5)?

Si scrivano le invocazioni di funzRic specificando il valore dei parametri attuali in ingresso in ogni chiamata

Si descriva quindi brevemente (max 2 righe) cosa esegue la funzione funzRic

SOLUZIONE

```
k =  
  3  
>> funzRic(k, 0)  
ans =  
  1  
>> funzRic(k, 1)  
ans =  
  3  
>> funzRic(k, 2)  
ans =  
  9  
>> funzRic(k, -1)  
ans =  
  1
```

```
funzRic(7, 5)  
funzRic(7, 4)  
funzRic(7, 3)  
funzRic(7, 2)  
funzRic(7, 1)  
funzRic(7, 0)
```

La funzione calcola $\text{arg1}^{\text{arg2}}$

Domanda 3. Matlab (8 punti).

Si scriva un programma per analizzare i risultati di una gara di nuoto. In particolare, la gara in questione è i 400 metri stile libero in vasca olimpica (50 metri) e coinvolge 6 atleti.

I tempi con cui ogni atleta ha percorso le vasche della sua gara vengono registrati in una matrice **M**. In particolare **M(n,m)** indica il tempo con cui l'atleta **n**-simo ha percorso la sua **m**-sima vasca.

Ad esempio se la terza riga di **M** è 16, 16.5, 15.9,. vuol dire che il terzo atleta ha impiegato 16 secondi a fare la prima vasca, 16.5 a fare la seconda e così via.

Si sviluppino le seguenti funzioni Matlab per risolvere i problemi descritti. Ogni funzione deve prendere in ingresso solamente la matrice M e restituire le informazioni richieste in opportune variabili. Le funzioni non visualizzano quindi alcun valore delle variabili a schermo:

- *calcolaTempoMedioVasca* permette di calcolare il tempo medio con cui ogni atleta ha percorso le sue vasche
- *calcolaPrimoUltimo* permette di individuare il vincitore della gara, cioè il concorrente che ha completato per primo i 400 metri. La funzione restituisce anche l'atleta arrivato per ultimo.
- *disegnaGraficoVincitore* traccia un grafico in cui l'asse delle x rappresenta il numero di vasca e l'asse delle y il tempo con cui quella vasca è stata percorsa. Il grafico riporta
 - o una linea rossa per indicare i tempi del primo arrivato,
 - o una linea blu per indicare i tempi dell'ultimo arrivato,
 - o un cerchio giallo attorno tempi in cui l'ultimo arrivato è andato più veloce del primo arrivato.

Si apra la figura direttamente nel corpo della funzione.

N.B. Saranno valutate molto meno le soluzioni che NON sfruttano i vantaggi sintattici del linguaggio Matlab nella gestione degli array.

SOLUZIONE

```
function tt = calcolaTempoMedio(M)
% tt = sum(M, 2) / size(M,2);
tt = mean(M, 2);
```

```
function [primo, ultimo] = calcolaPrimoUltimo(M)
tt = sum(M, 2);
[~, primo] = min(tt);
[~, ultimo] = max(tt);
```

```
function visualizzaTempi(M)
[best, last] = calcolaPrimoUltimo(M);
```

```
tempiVincitore = M(best, :);
tempiUltimo = M(last, :);
```

```
indx = find(tempiUltimo < tempiVincitore);
```

```
figure(1),
plot([1 : 8], tempiVincitore, 'r');
hold on
plot([1 : 8], tempiUltimo, 'b');
plot(indx, tempiUltimo(indx), 'y*');
hold off
```

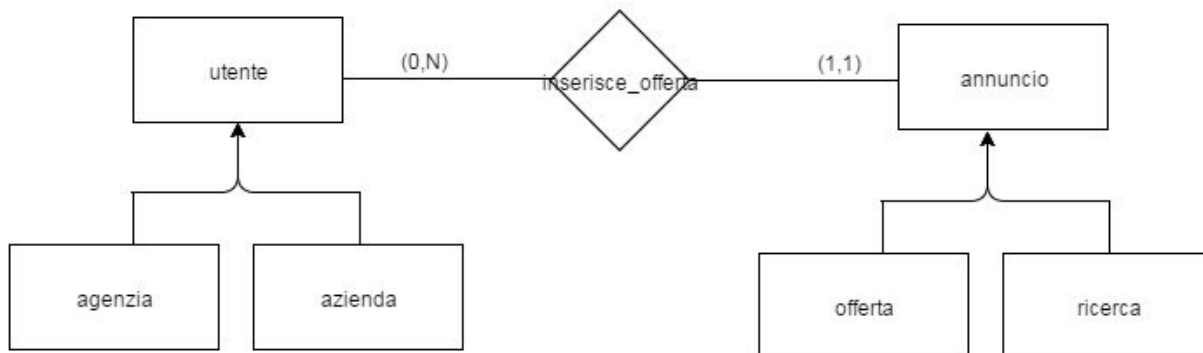
Domanda 4. Modello Concettuale + Schema Logico (10 punti).

Si vuole realizzare una base dati per gestire una piattaforma web gratuita di annunci lavorativi. Ogni utente che desidera inserire un annuncio deve registrarsi al sito e sarà identificato con la sua mail (che sarà il suo nome utente) e una password. Per registrarsi, l'utente deve inserire il suo nome, cognome, data e luogo di nascita, indirizzo di residenza e numero di telefono. Se l'utente è una azienda, questi stessi campi saranno inseriti relativamente ai dati dell'azienda e ci sarà anche il campo partita IVA aggiuntivo. L'utente potrebbe anche essere un'agenzia interinale e a questo punto occorre inserire il codice filiale.

Ogni utente può inserire uno o più annunci ed il sistema prevede due tipi di annunci: ricerca o di offerta di lavoro. Ogni annuncio è identificato da un numero progressivo ed è descritto con un titolo, un testo esplicativo (che non deve superare 300 caratteri), una categoria lavorativa. Nel caso di annunci di ricerca, l'utente avrà la possibilità di associare al suo annuncio il proprio curriculum. Per quanto riguarda gli annunci di offerta di lavoro, l'annuncio dovrà contenere la data di inizio dell'offerta, la durata del contratto e la sua tipologia (tempo parziale o tempo pieno).

Per ogni inserimento di un annuncio da parte dell'utente sarà memorizzata la data e ora di inserimento.

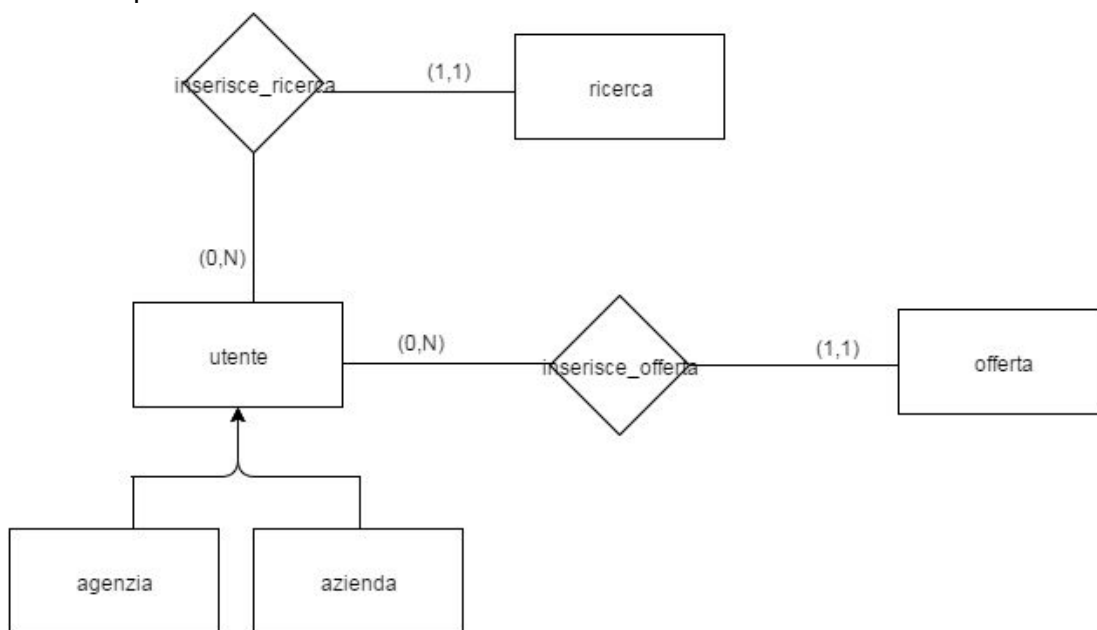
1. Progettare lo schema **Entità-relazione** per progettare la base dati del sistema sopra descritto (si inseriscano brevi commenti solo se necessari per giustificare alcune scelte progettuali)
2. Progettare lo **schema relazionale** della base dati. (si inseriscano brevi commenti solo se necessari per giustificare alcune scelte progettuali)
3. Sulla base dello schema relazionale ottenuto, si definiscano in algebra relazionale le seguenti interrogazioni:
 - Selezionare il nome e cognome dell'utente che ha inserito un annuncio di ricerca lavoro sia nel mese di gennaio 2013 sia nel mese di giugno 2013.
 - Stampare tutte le offerte di lavoro pubblicate nel mese di Agosto 2013
 - Visualizzare gli utenti registrati che non hanno mai inserito un annuncio di ricerca di lavoro



generalizzazione su utente parziale ed esclusiva
 generalizzazione su annuncio totale ed esclusiva

Gli attributi non sono stati riportati (ma erano richiesti)

Nella semplificazione dello schema si arriva a



dove agenzia e azienda vengono riportate in utente con l'aggiunta di un campo tipo

Schema Logico

Utente(nomeUtente, password, dataNascita, luogoNascita, indirizzo, nTelefono, nome, cognome, tipo, Piva, codice_filiale)

AnnuncioRicerca(id, titolo, testo, categoria, curriculum, nomeUtente, data, ora)

AnnuncioOfferta(id, titolo, testo, categoria, data_inizio, durata, nomeUtente, data, ora)

vincoli integrità referenziale

AnnuncioRicerca.nomeUtente ----> Utente.nomeUtente

AnnuncioOfferta.nomeUtente ----> Utente.nomeUtente

Queries

PROJ_(Utente.nome, Utente.cognome) [(SEL_(AnnuncioRicerca.data > 1.1.2013 and
AnnuncioRicerca.data < 31.1.2013) (AnnuncioRicerca JOIN Utente))] INTERSEZ

PROJ_(Utente.nome, Utente.cognome) [(SEL_(AnnuncioRicerca.data > 1.6.2013 and
AnnuncioRicerca.data < 30.6.2013) (AnnuncioRicerca JOIN Utente))]

PROJ_(Titolo,Testo) (SEL_(data>1.8.2013 and data < 31.8.2013) AnnuncioOfferta)

PROJ_(nomeUtente) (Utente) \ PROJ_(nomeUtente) AnnuncioOfferta

Domanda 5. Schema Logico e Interrogazioni in SQL (4pti)

Dato lo schema logico:

REGISTA (Nome, DataNascita, Nazionalità)

ATTORE (Nome, DataNascita, Nazionalità)

INTERPRETA (Attore, Film, Personaggio)

FILM (Titolo, NomeRegista, Anno)

PROIEZIONE (NomeCin, CittaCin, TitoloFilm)

CINEMA (Citta, NomeCinema, Sale, Posti)

Sottolineare le chiavi primarie ed indicare i vincoli di integrità referenziale.

Definire le seguenti interrogazione in SQL:

i. Elencare i nomi di registri che hanno diretto almeno 3 film.

ii. Registi che hanno recitato in (almeno) un loro film.

*iii. Selezionare le Nazionalità dei registi che hanno diretto qualche film nel 1992 **ma non** hanno diretto **alcun** film nel 1993.*

SOLUZIONE

REGISTA (Nome, DataNascita, Nazionalità)

ATTORE (Nome, DataNascita, Nazionalità)

INTERPRETA (Attore, Film, Personaggio)

FILM (Titolo, NomeRegista, Anno)

PROIEZIONE (NomeCin, CittaCin, TitoloFilm)

CINEMA (Citta, NomeCinema, Sale, Posti)

vincoli di integrità referenziale:

interpeta.Attore ----> attore.Nome

interpreta.Film -----> film.Titolo

Film.NomeRegista ----> Regista.Nome

Proiezione.NomeCin ---> Cinema.NomeCinema

Proiezione.TitoloFilm ----> Film.Titolo

Proiezione.CittaCin ----> Cinema.Citta

```
SELECT Nome
FROM REGISTA,FILM
WHERE FILM.NomeRegista=REGISTA.Nome
GROUP BY REGISTA.Nome
HAVING COUNT(*)>2
```

```
SELECT DISTINCT NomeRegista
FROM FILM join INTERPRETA
      on Titolo=Film
WHERE Regista.NomeRegista=Interpreta.Attore
```

```
SELECT DISTINCT Nazionalità
FROM REGISTA, FILM
WHERE Nome = NomeRegista AND Anno='1992'
AND Nome NOT IN
(SELECT NomeRegista
 FROM FILM WHERE Anno='1993')
```