



PICCOLA GUIDA ALLA RISOLUZIONE DEL COMPITO DI INFORMATICA ALL'ESAME DI STATO

Sembrerà banale, ma per svolgere esattamente un compito di informatica è opportuno raggiungere la massima concentrazione possibile perché, fatto qualche errore di analisi iniziale, tale errore comprometterebbe l'intera esattezza e correttezza del compito nel suo prosieguo. Pertanto consiglio di perdere un po' più di tempo per la prima fase valutando il problema sotto ogni punto di vista; fatta bene tale fase, il resto verrà da sé!! Vediamo allora i vari steps (i primi cinque si devono sempre fare):

1. ANALISI DEI DATI (Analisi della realtà)

Individuare le entità, gli attributi e le associazioni (relazioni) motivando le scelte effettuate (molto importante farlo!! Specie in sede d'esame!);

2. MODELLO E/R (Modello Concettuale)

Disegnare il modello E/R con i versi delle associazioni;

3. CREAZIONE ARCHIVI (Modello Logico)

Definire poi gli archivi utilizzando le regole di derivazione e descrivere le caratteristiche di ciascun archivio con uno schema del tipo:

archivio | campo | chiave | formato | dimensione

.....

4. ANALISI DEL PROBLEMA e FUNZIONIGRAMMA

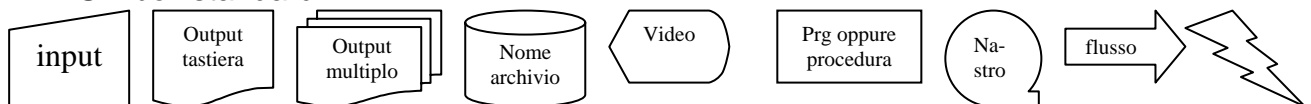
Analizzare il problema cercando di individuare la scomposizione del problema in un albero gerarchico e producendo il relativo funzionigramma (individuare quindi le funzioni madre e le funzioni figlie, ricordando che è preferibile non avere + di 6 funzioni figlie);

5. SCHEMA DELLE RISORSE DI SISTEMA

Lo schema delle risorse deve rappresentare con i simboli standard:

- gli archivi utilizzati dai diversi programmi
- il flusso delle informazioni, all'interno dell'elaborazione, dalle unità di input e verso quelle di output.

Simboli standard:



Conn.ne rete

6. EVENTUALI INTERROGAZIONI SQL

Implementare eventuali interrogazioni SQL con i comandi standard studiati, cercando di individuare eventuali interrogazioni nidificate;

7. EVENTUALE IMPLEMENATAZIONE IN VB 6.0

Implementare solo alcune procedure col linguaggio che meglio conosciamo (!) ovvero Visual Basic 6.0;

8. EVENTUALE REALIZZAZIONE PAGINA WEB

Implementare in HTML standard una semplice ed eventuale pagina web richiesta dalla traccia;

IN PARTICOLARE

Stiamo attenti per:

▪ Modellazione dei dati

1. Leggere più volte e attentamente il testo prima di decidere il modello dei dati.
2. L'ambito del problema non è un'entità (per esempio in un problema come il seguente:
"In un Istituto scolastico si vogliono gestire con un database le informazioni sui docenti e gli studenti". L'istituto non è un'entità)
3. Alcune regole che valgono (quasi) sempre: i sostantivi corrispondono alle entità, gli aggettivi e le proprietà corrispondono agli attributi, i verbi alle associazioni.
4. Leggere attentamente anche le richieste di output e il testo delle interrogazioni, per determinare correttamente e in modo completo gli attributi delle entità.
5. Gli attributi descrittivi che si ripeteranno con valori uguali per istanze diverse della stessa entità, per esigenze di normalizzazione, devono diventare entità (per esempio, Comuni, causali, tipologie, ecc.) legate da un'associazione 1:N con l'entità che devono descrivere.
Esse saranno poi derivate in tabelle di decodifica (codice, descrizione).
6. Motivare, se necessario, le scelte fatte, evidenziando ipotesi aggiuntive e vincoli introdotti.
7. Nella scelta del tipo di associazioni:
 - 1:1 molto rara, probabilmente si risolve con una sola entità con gli attributi opportuni; se poi nel modello E/R deve esserci per forza, si ricordi la regola di derivazione che risolve l'associazione in un'unica tabella con gli attributi della prima e della seconda entità
 - 1:N è la più frequente; di solito la parte statica (o anagrafica) sta vicino a 1 e la parte dinamica (o movimenti) sta vicino a N
 - N:N si può spezzare già nella fase di modellazione in due associazioni 1:N; in questo caso le due entità di partenza vanno vicino a 1 e l'entità "di legame" va vicino a N. In caso contrario, dal modello E/R con associazione N:N, la regola di derivazione crea tre tabelle: la terza tabella contiene la chiave della prima entità, la chiave della seconda e gli eventuali attributi dell'associazione.
8. Le entità di tipo dinamico (o movimenti) hanno sicuramente una data di registrazione; inoltre possono essere opportunamente caratterizzate da una chiave autoincrementale (ID di tipo contatore), numerico progressivo (numero di registrazione).
9. Dopo aver disegnato il modello E/R utilizzare le regole di lettura per controllare la correttezza delle frasi in linguaggio comune rispetto al problema trattato.

▪ Derivazione delle tabelle

1. Usare con rigore le regole di derivazione.
2. Ogni associazione 1:1 diventa un'unica tabella.
3. Ogni associazione 1:N porta una chiave esterna nella tabella dalla parte di N.
4. Ogni associazione N:N porta due chiavi esterne nella terza tabella derivata.

▪ Descrizioni dei dati

1. I campi che non vengono usati in calcoli sono di tipo testo (stringa); per esempio, telefono, partita IVA, CAP, pur essendo composti da cifre sono di tipo testo.
2. Per facilitare la comprensione a se stessi e al docente, si possono esplicitare alcuni dati di esempio per ciascuna delle tabelle.

▪ Interrogazioni e linguaggio SQL

1. Le query devono essere ottimizzate in modo da usare il minor numero di tabelle; se la richiesta riguarda solo un codice, non serve usare anche la tabella collegata.
2. Se si usano funzioni di aggregazione (COUNT, SUM, AVG, ...), quasi sempre c'è un GROUP BY. Quasi sempre, perché le funzioni si possono usare anche per contare le righe che rispettano una condizione oppure per sommare i valori di un campo numerico per le righe che rispettano una condizione.
3. Gli attributi scritti vicino a SELECT devono essere scritti vicino a GROUP BY e viceversa, a meno che siano usati come argomento di funzioni di aggregazione.
4. Se c'è HAVING, ci deve essere GROUP BY, perché HAVING pone condizioni sui gruppi.
5. Vicino a HAVING c'è sicuramente una condizione con una funzione di aggregazione.
6. Vicino a HAVING non ci può essere una condizione sulle singole righe: questa deve essere messa vicino a WHERE. Quindi vicino a WHERE non ci possono essere condizioni con funzioni di aggregazione.