

Basi di Dati – Modulo 1

14 Febbraio 2023

Traccia B

Si consideri una base dati di una federazione sportiva che organizza tornei di scacchi:

GIOCATORE(ID, Nome, Cognome, DataNascita, Nazionalità, Elo)

TORNEO(Codice, Titolo, Città, Anno)

PARTECIPA(Torneo, Giocatore)

PARTITA(ID, CodTorneo, Data, Bianco, Nero, Risultato)

Note:

- GIOCATORE.ID è un codice identificativo del giocatore
- GIOCATORE.Elo è un numero intero che corrisponde alla forza di gioco del giocatore
- GIOCA.Torneo si riferisce ad un valore di TORNEO.Codice
- GIOCA.Giocatore si riferisce ad un valore di GIOCATORE.ID
- PARTITA.CodTorneo si riferisce ad un valore di TORNEO.Codice
- PARTITA.Data è una data in formato GG/MM/AAAA
- PARTITA.Bianco e PARTITA.Nero si riferiscono ad un valore di GIOCATORE.ID
- PARTITA.Risultato può contenere i valori "1-0", "0-1", oppure " $\frac{1}{2}$ - $\frac{1}{2}$ " se la partita è stata vinta dal bianco, dal nero, oppure è finita in parità, rispettivamente

1a) Trovare nome e cognome dei giocatori di nazionalità francese che hanno battuto giocatori che si chiamano Beth Harmon in qualche torneo giocato a Boston dopo il 2010.

2a) Trovare nome e cognome dei giocatori che, giocando coi pezzi bianchi, non hanno mai battuto giocatori di Elo almeno 2300 ad una qualunque edizione del torneo "Grand Prix".

2) Siano dati lo schema $R=ABCDEFG$ e l'insieme di dipendenze funzionali

$F=\{ G \rightarrow FA, BD \rightarrow EF, CD \rightarrow FA, C \rightarrow B, B \rightarrow DA, D \rightarrow GC, C \rightarrow D \}$

2a) Determinare le tre chiavi dello schema

2b) Dire se lo schema è 3NF e giustificare l'affermazione

2c) Calcolare una decomposizione ρ che ha i sottoschemi in 3NF, preserva le dipendenze e ha un join senza perdita, e descrivere il procedimento utilizzato giustificando i passaggi

3) E' dato un file di 2.125.800 record. Ogni record occupa 280 byte, di cui 130 per la chiave. Un blocco contiene 2048 byte. Un puntatore a blocco occupa 4 byte. Si utilizza una organizzazione B-TREE.

3a) Calcolare l'occupazione in blocchi del file principale quando l'albero ha altezza massima.

3b) Calcolare l'occupazione in blocchi del file indice (tutti i livelli) quando l'albero ha altezza massima.

3c) Calcolare il costo di una ricerca quando l'albero ha altezza massima.

Si consideri una base dati di una federazione sportiva che organizza tornei di scacchi:

GIOCATORE(ID, Nome, Cognome, DataNascita, Nazionalità, Elo)

TORNEO(Codice, Titolo, Città, Anno)

PARTECIPA(Torneo, Giocatore)

PARTITA(ID, CodTorneo, Data, Bianco, Nero, Risultato)

Note:

- GIOCATORE.ID è un codice identificativo del giocatore
- GIOCATORE.Elo è un numero intero che corrisponde alla forza di gioco del giocatore
- GIOCA.Torneo si riferisce ad un valore di TORNEO.Codice
- GIOCA.Giocatore si riferisce ad un valore di GIOCATORE.ID
- PARTITA.CodTorneo si riferisce ad un valore di TORNEO.Codice
- PARTITA.Data è una data in formato GG/MM/AAAA
- PARTITA.Bianco e PARTITA.Nero si riferiscono ad un valore di GIOCATORE.ID
- PARTITA.Risultato può contenere i valori "1-0", "0-1", oppure "½-½" se la partita è stata vinta dal bianco, dal nero, oppure è finita in parità, rispettivamente

1a) Trovare nome e cognome dei giocatori di nazionalità francese che hanno battuto giocatori che si chiamano Beth Harmon in qualche torneo giocato a Boston dopo il 2010.

2a) Trovare nome e cognome dei giocatori che, giocando coi pezzi bianchi, non hanno mai battuto giocatori di Elo almeno 2300 ad una qualunque edizione del torneo "Grand Prix".

1A) $\Pi_{ID} \left(\sigma_{\left(\begin{array}{l} \text{Nome} = \text{'Beth'} \\ \text{Cognome} = \text{'Harmon'} \end{array} \right)} \text{GIOCATORE} \right)$

$\text{GIOCATORE F BIANCO V} = \sigma_{\left(\begin{array}{l} \text{Nazionalità} = \text{'franca'} \\ \text{Risultato} = \text{'1-0'} \\ \text{Nero} = ID_{\text{BETH}} \end{array} \right)} \left(\text{GIOCATORE} \bowtie \text{PARTITA} \right)$
ID = BIANCO

$\text{GIOCATORE F NERO V} = \sigma_{\left(\begin{array}{l} \text{Nazionalità} = \text{'franca'} \\ \text{Risultato} = \text{'0-1'} \\ \text{Bianco} = ID_{\text{BETH}} \end{array} \right)} \left(\text{GIOCATORE} \bowtie \text{PARTITA} \right)$
ID = NERO

$\text{TORNEO} = \sigma_{\left(\begin{array}{l} \text{Città} = \text{'Boston'} \\ \text{Anno} > 2010 \end{array} \right)} \left(\text{TORNEO} \right)$

$G_{\text{BFVT}} = \text{GIOCATORE F BIANCO V} \bowtie \text{TORNEO}$
CodTorneo = Codice

$G_{\text{NFVT}} = \text{GIOCATORE F NERO V} \bowtie \text{TORNEO}$
CodTorneo = Codice

$\text{OUT} = \Pi_{\left(\begin{array}{l} \text{Nome} \\ \text{Cognome} \end{array} \right)} \left(G_{\text{BFVT}} \cup G_{\text{NFVT}} \right)$

1B)

$$ELO_{ALTO} = \sigma_{ELO \approx 2300} (GLOTTARE)$$

$$TORNEO = \sigma_{\substack{\text{Titolo} \\ \rightarrow \\ \text{"Grand Prix"}}} (TORNEO)$$

$$PARTITA_{TORNEO} = \text{TORNEO} \wedge \text{PARTITA} \\ \text{codice} = \text{cod TORNEO}$$

$$GIOCATORI_{TORNEO} = (GLOTTARE \wedge \text{PARTECIPA} \wedge \text{TORNEO}) \\ \text{ID} = \text{GLOTTARE} \quad \text{TORNEO} = \text{COD}$$

$$GLOTTARE_{PARTE} = \sigma_{\substack{\text{ESULTATO} \\ = \\ \text{"a3"}}} (PARTITA_{TORNEO} \wedge ELO_{ALTO}) \\ \text{USID} = \text{ID}$$

$$QOT = \pi_{\substack{\text{VARE}, \\ \text{LOGGAC}}} (GIOCATORI_{TORNEO} - GLOTTARE_{PARTE})$$

2) Siano dati lo schema $R=ABCDEFG$ e l'insieme di dipendenze funzionali

$F=\{ G \rightarrow FA, BD \rightarrow EF, CD \rightarrow FA, C \rightarrow B, B \rightarrow DA, D \rightarrow GC, C \rightarrow D \}$

2a) Determinare le tre chiavi dello schema

2b) Dire se lo schema è 3NF e giustificare l'affermazione

2c) Calcolare una decomposizione ρ che ha i sottoschemi in 3NF, preserva le dipendenze e ha un join senza perdita, e descrivere il procedimento utilizzato giustificando i passaggi

2a)

$$(C)_F^+ = C B D A G F E \Rightarrow C \text{ chiave}$$

$$(D)_F^+ = D G C B A F E \Rightarrow D \text{ chiave}$$

$$(B)_F^+ = B D A G C F E \Rightarrow B \text{ chiave}$$

2b)

NON È IN 3NF PER $G \rightarrow FA$ poiché G non è superchiave ed

FA non è primo

2c)

I° passo

Ogni determinante doppio (o più) viene diviso

$$F = \{ G \rightarrow F, G \rightarrow A, BD \rightarrow E, BD \rightarrow F, CD \rightarrow F, CD \rightarrow A, C \rightarrow B, B \rightarrow D, B \rightarrow A, D \rightarrow G, D \rightarrow C, C \rightarrow D \}$$

II° passo Copertura MINIMALE

Controlla i non singolari se possono essere ridotti applicando l'algoritmo

- $BD \rightarrow E$

$$(B)_F^+ = B D F A C G \text{ } \& E$$

$$(D)_F^+ = D G C F B A \text{ } \& E$$

$$- BD \rightarrow F$$

$$(B)_F^+ = BD \subseteq F \supset F \Rightarrow B \not\rightarrow F$$

$$- CD \rightarrow F$$

$$(C)_F^+ = CBDAGF \supset F \Rightarrow C \not\rightarrow F$$

$$- CD \rightarrow A$$

$$(C)_F^+ = CBADG \supset A \Rightarrow C \not\rightarrow A$$

Dopo Π^0

$$F = \{ G \rightarrow F, G \rightarrow A, B \rightarrow E, B \rightarrow F, C \rightarrow F, C \rightarrow A, C \rightarrow B, B \rightarrow D, B \rightarrow A, D \rightarrow G, D \rightarrow C, C \rightarrow D \}$$

N.B.

POICHE BD E CD CONTENGONO SOLO CHIAVI ROTTEGGIATE GIÀ DECISE DI TAGLIARE UNA.

III⁰ CERCO RIDONDANZE

$$- G \rightarrow F$$

$$(G)_{F \rightarrow G}^+ = GA \not\supset F \Rightarrow \text{NON TAGLIO}$$

$$- G \rightarrow A$$

$$(G)_F^+ = GF \not\supset A \Rightarrow \text{NON TAGLIO}$$

$$- B \rightarrow E$$

$$(B)_{F \rightarrow B}^+ = BFADCG \not\supset E \Rightarrow \text{NON TAGLIO}$$

$$- B \rightarrow F$$

$$(B)_{F \rightarrow B}^+ = BEDACF \supset F = \text{TAGLIO}$$

$$F = \{G \rightarrow F, G \rightarrow A, B \rightarrow E, \cancel{B \rightarrow F}, \cancel{C \rightarrow F}, \cancel{C \rightarrow A}, \cancel{C \rightarrow B}, B \rightarrow D, B \rightarrow A, D \rightarrow G, D \rightarrow C, C \rightarrow D\}$$

$$- C \rightarrow F$$

$$(C)^+_{F \rightarrow C \rightarrow F} = CABDGF \supset F \Rightarrow \text{To } LG_0$$

$$F = \{G \rightarrow F, G \rightarrow A, B \rightarrow E, \cancel{B \rightarrow F}, \cancel{C \rightarrow F}, \cancel{C \rightarrow A}, \cancel{C \rightarrow B}, B \rightarrow D, B \rightarrow A, D \rightarrow G, D \rightarrow C, C \rightarrow D\}$$

$$- C \rightarrow A$$

$$(C)^+_{F \rightarrow C \rightarrow A} = CBDFGA \supset A \Rightarrow \text{To } LG_0$$

$$F = \{G \rightarrow F, G \rightarrow A, B \rightarrow E, \cancel{B \rightarrow F}, \cancel{C \rightarrow F}, \cancel{C \rightarrow A}, \cancel{C \rightarrow B}, B \rightarrow D, B \rightarrow A, D \rightarrow G, D \rightarrow C, C \rightarrow D\}$$

$$- C \rightarrow B$$

$$(C)^+_{F \rightarrow C \rightarrow B} = CDGFAB \nexists B \Rightarrow \text{New Table}$$

$$- B \rightarrow D$$

$$(B)^+_{F \rightarrow B \rightarrow D} = BDA \nexists D \Rightarrow \text{New Table}$$

$$- B \rightarrow A$$

$$(B)^+_{F \rightarrow B \rightarrow A} = BDCGFABE \supset A \Rightarrow \text{To } LG_0$$

$$F = \{G \rightarrow F, G \rightarrow A, B \rightarrow E, \cancel{B \rightarrow F}, \cancel{C \rightarrow F}, \cancel{C \rightarrow A}, \cancel{C \rightarrow B}, \cancel{B \rightarrow D}, \cancel{B \rightarrow A}, D \rightarrow G, D \rightarrow C, C \rightarrow D\}$$

$$- D \rightarrow G$$

$$(D)^+_{F \rightarrow D \rightarrow G} = DCBFE \nexists G \Rightarrow \text{New Table}$$

$\neg D \rightarrow C$

$$(C \vee D)^+_{F \rightarrow D \rightarrow C} = DG \vee C \Rightarrow \text{NOW Take}$$

$\neg C \rightarrow D$

$$(C \vee D)^+_{F \rightarrow C} = C \vee D \vee G \vee A \vee D \Rightarrow \text{Take}$$

$$F = \{G \rightarrow F, G \rightarrow A, B \rightarrow E, B \rightarrow F, C \rightarrow F, C \rightarrow A, C \rightarrow B, B \rightarrow D, B \rightarrow A, D \rightarrow G, D \rightarrow C, C \rightarrow D\}$$

Dopo 3° PASSO

$$F = \{G \rightarrow F, G \rightarrow A, B \rightarrow E, C \rightarrow B, B \rightarrow D, D \rightarrow G, D \rightarrow C\}$$

METTO INSIGNE DETERMINANTE E DETERMINATO

$$\rho = \{GF, GA, BE, CB, BD, DG, DC\} \text{ E' IN 3NF E PRESENZA F}$$

MA CHE' CONTIENE ANCHE QUALI HA SON SENZA PERDITA

3) E' dato un file di 2.125.800 record. Ogni record occupa 280 byte, di cui 130 per la chiave. Un blocco contiene 2048 byte. Un puntatore a blocco occupa 4 byte. Si utilizza una organizzazione B-TREE.

3a) Calcolare l'occupazione in blocchi del file principale quando l'albero ha altezza massima.

3b) Calcolare l'occupazione in blocchi del file indice (tutti i livelli) quando l'albero ha altezza massima.

3c) Calcolare il costo di una ricerca quando l'albero ha altezza massima.

$$3a) \text{ Tot Blochi FP} = \frac{\lceil N^{\circ} \text{ Record} \rceil}{\text{Record} \times \text{Block}} = \frac{\lceil \frac{2125800}{4} \rceil}{4} = 531450$$

$$\text{Record} \times \text{Block} = \frac{\text{Block size}}{\text{P size}} = \frac{\lceil \frac{2048}{280} \rceil}{280} = 4$$

$$3b) d = \left\lceil \frac{(\text{Psize} / 2) - \text{Psize}}{\text{Psize} + \text{key size}} \right\rceil + 1 = \left\lceil \frac{1024}{130} \right\rceil + 1 = 8 + 1 = 9$$

$$\text{Liv 0} = \frac{\lceil \text{Tot Blochi FP} \rceil}{d} = 531450$$

$$\text{Liv 1} = \frac{\lceil 531450 \rceil}{9} = 6562$$

$$\text{Liv 2} = \frac{\lceil 6562 \rceil}{9} = 730$$

$$\text{Liv 3} = \frac{\lceil 730 \rceil}{9} = 82$$

$$\text{Liv 4} = \frac{\lceil 82 \rceil}{9} = 10$$

$$\text{Liv 5} = \frac{\lceil 10 \rceil}{9} = 2$$

$$\text{Liv 6} = \frac{\lceil 2 \rceil}{9} = 1$$

$$\text{Tot Blochi FI} = \text{Tot Blochi FP} + 531450 + 6562 + 730 + 82 + 10 + 2 + 1 =$$

7 66 437

2d) 7 AKES1