

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

GIOCATOREID	Nome, Cognome, DataNascita, Nazionalità, Elo
TORNEO	(Codice, Titolo, Città, Anno)
PARTICIPAZIONE	(Torneo, Giocatore)
PARTITA	(ID, CodTorneo, Data, Bianco, Nero, Risultato)

- 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) $BlackWins = \pi_{NERO}(\sigma_{CITTA::BOSTON \wedge ANNO > 2010}(\sigma_{NOME::BETH \wedge COGNOME::HARMON}(\sigma_{RISULTATO::0-1}(PARTITAM_{BIANCO::GIOCATOREID} GIOCATORE)))M_{CORTORNEO::CODICE} TORNEO))$

WhiteWins = $\pi_{BIANCO}(\sigma_{CITTA::BOSTON \wedge ANNO > 2010}(\sigma_{NOME::BETH \wedge COGNOME::HARMON}(\sigma_{RISULTATO::1-0}(PARTITAM_{NERO::GIOCATOREID} GIOCATORE)))M_{CORTORNEO::CODICE} TORNEO))$

FPlayer = $\pi_{ID}(\sigma_{NAZIONALITA::FRANCIA} (GIOCATORE))$ Query finale $Q = \pi_{NOME, COGNOME} (GIOCATOREM(FPlayer \cap (BlackWins \cup WhiteWins)))$

1b) Prima trovo quelli che, hanno disputato partite da bianchi contro giocatori di elo almeno 2300 nel GrandPrix. $PTG = (PARTITAM_{CORTORNEO::CODICE} TORNEO)M_{GIOCATOREID::NERO} GIOCATORE$ Partite = $\sigma_{ELO \geq 2300 \wedge TITOLO::GRANDPRIX} (PTG)$

WhiteWins = $\pi_{GIOCATOREID}(\sigma_{RISULTATO::1-0} (Partite))$ AllPlayer = $\pi_{GIOCATOREID} (Partite)$ $Q' = AllPlayer - WhiteWins$

Query finale : $Q = \pi_{NOME, COGNOME} (GIOCATOREM Q')$

2) Siano dati lo schema R=ABCDEFG e l'insieme di dipendenze funzionali
 F={ G→FA, BD→EF, CD→FA, C→B, B→DA, D→GC, C→D}
 2a) Determinare le tre chiavi dello schema
 2b) Dire se lo schema è 3NF e giustificare l'affermazione
 2c) Calcolare una decomposizione p che ha i sottoschemi in 3NF, preserva le dipendenze e ha un join senza perdita, e descrivere il procedimento utilizzato giustificando i passaggi

BCD CDG BDC BCG

2a) Noto che AEF non sono mai determinanti ⇒ non sono nella chiave. Controllo i sottoinsiemi di BCDG.

$BCD_F^+ = R$, e' superchiave, controllo $BC_F^+ = R$ ma $C \rightarrow B \Rightarrow C_F^+ = R \Rightarrow C$ e' chiave, inoltre noto che $B \rightarrow DA$ quindi $B \rightarrow D$, e $D \rightarrow GC \Rightarrow B \rightarrow C \Rightarrow$ anche B e' chiave. Ho osservato che $D \rightarrow C$, quindi D e' chiave.

2b) ho $G \rightarrow FACF \Rightarrow G \rightarrow F$, ma F non e' primo e G non e' superchiave, quindi lo schema non e' in 3NF.

2c) Voglio trovare una cop. minimale, minimizzo i determinati :

$F = \{G \rightarrow F, G \rightarrow A, BD \rightarrow E, BD \rightarrow F, CD \rightarrow F, CD \rightarrow A, C \rightarrow B, B \rightarrow D, B \rightarrow A, D \rightarrow G, D \rightarrow C, C \rightarrow D\}$, vado al passo succ. :

- $BD \rightarrow E$, B e D sono chiavi, quindi ne tolgo una. • $BD \rightarrow F$ analogo. • $CD \rightarrow A$ analogo. • $CD \rightarrow F$ analogo. Ottengo:

$F = \{G \rightarrow F, G \rightarrow A, D \rightarrow E, D \rightarrow F, D \rightarrow A, C \rightarrow B, B \rightarrow D, B \rightarrow A, D \rightarrow G, D \rightarrow C, C \rightarrow D\}$, ora controllo le ridondanze:

$G_{F/AG}^+ = \{AG\}$. $G_{F/AG}^+ = \{FG\}$. $D_{F/DE}^+ \neq E$. $D_{F/DE}^+ \neq F$, e' di troppo. $D_{F/D \rightarrow A}^+ \neq A$, e' di troppo. $B \notin C_{F/BC}^+$. $D \notin B_{F/BD}^+$. $A \notin B_{F/BA}^+$ e' di troppo.

La copertura minimale e': $F = \{G \rightarrow F, G \rightarrow A, D \rightarrow E, C \rightarrow B, B \rightarrow D, D \rightarrow G, D \rightarrow C, C \rightarrow D\}$ applico l'algoritmo:

C'e' un attributo che non compare in $x \rightarrow y \in F$? no ⇒ $\exists x \rightarrow y | xy = R$? no ⇒ $p = \{FG, AG, DE, BC, BD, DG, CD\}$ contiene un elemento con una chiave, quindi ha un Join senza perdita.

3) E' dato un file di 2.125.800 record. Ogni record occupa 256 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) Se l'albero ha altezza massima, i blocchi sono riempiti a meta', in un blocco entrano $\lceil \frac{1024}{200} \rceil = 4$ record.

Per il file principale servono $\lceil \frac{2125800}{4} \rceil = 531450$ blocchi.

3b) $(Key \wedge pointer) \times Block = \lceil \frac{1024 \cdot 4}{130 \cdot 4} \rceil = 8 \Rightarrow 9$ puntatori a blocco. LIV 1 = $\lceil \frac{531450}{9} \rceil = 59050$ LIV 2 = $\lceil \frac{59050}{9} \rceil = 6562$

LIV 3 = $\lceil \frac{6562}{9} \rceil = 730$ LIV 4 = $\lceil \frac{730}{9} \rceil = 82$ LIV 5 = $\lceil \frac{82}{9} \rceil = 10$ LIV 7 = RADICE. MainFile = 66437

3c) Se la radice non e' in RAM, sono necessari 8 accessi