

Dato il seguente schema di una base di dati contenente dati relativi a trasporti extraurbani

FERMATA(id, nome, via, comune)  
 TRATTA(id, #idp, #ida)  
 PERCORSO(id, #idp, #idf, numero)  
 ORARIO(id, #idp, #idp, #idp)

1a) Trovare id della tratta, nome e via della fermata di partenza, nome e via della fermata di arrivo di tratte che partono dal Comune di Guidonia e arrivano nel comune di Roma con partenza tra le ore 8:00 e le ore 9:00  
 1b) Trovare id della tratta e nome della fermata di arrivo per le tratte che hanno la prima partenza prima delle 6:00 del mattino

1a)  $TP = \pi_{TRATTA.ID}(\sigma_{COMUNE = GUIDONIA}(FERMATAM_{FERMATA.ID = IDP}(TRATTA)))$   $TA = \pi_{TRATTA.ID}(\sigma_{COMUNE = ROMA}(FERMATAM_{FERMATA.ID = IDA}(TRATTA)))$

TROMAGUID =  $\pi_{TRATTA.ID}(\sigma_{ORAPART \geq 8.00 / ORAPARR \leq 9.00}((TP \bowtie TA) \bowtie ORARIO))$  ← id delle fermate in questione!

$F_c = \rho_{IDP = IDC, NONI = NOME_C, VIA = VIA_C}(FERMATA)$   $Q = \pi_{TRATTA.ID, NOME, VIA, NOME_C, VIA_C}((FERMATAM_{FERMATA.ID = IDP}(TRATTA \bowtie TROMAGUID)) \bowtie_{IDA = IDC} FC)$

1b)  $OC = \rho_{ORAPART > ORAPARR_C}(ORARIO)$   $COMP = \pi_{IDT, ORAPART}(ORARIO \bowtie OC - \sigma_{ORAPART > ORAPARR_C}(ORARIO \bowtie OC))$

in COMP ho tutte le tratte con l'orario della loro prima partenza! BEFORE6 =  $\sigma_{ORAPART < 6.00}(COMP)$

Query finale:  $Q = \pi_{IDT, NOME}((TRATTA \bowtie_{IDA = FERMATA.ID}(FERMATA)) \bowtie_{TRATTA.ID = IDT} BEFORE6)$

2a) Dati lo schema di relazione R=ABCDE, l'insieme di dipendenze funzionali F={AB→C, AD→E, C→BE, BD→E} e la decomposizione ρ={ABCE, CDE} di R,

dire se ρ preserva F e illustrare il procedimento seguito per giungere alla risposta

2a) Necessito di controllare se  $F \subseteq G$ , dove  $G = \bigcup_{i=1}^n \pi_{R_i}(F)$ ,  $\forall x \rightarrow y \in F$ , devo calcolare la chiusura di x rispetto a G, e controllare che ci sia y. Applico l'algoritmo, escludendo le  $x \rightarrow y \in F$  tali che  $xy \in R_i \in \rho$ .

•  $AD \rightarrow E$ ,  $Z_0 = AD$   $S_0 = \frac{(AD \cap ABCE)^+ \cap ABCE}{(AD \cap CDE)^+ \cap CDE} = \frac{A^+ \cap ABCE}{D^+ \cap CDE} = \frac{A}{D} = AD \Rightarrow S_0 \subseteq Z_0$ ? SI  $\Rightarrow AD^+ = \{AD\} \nsubseteq E \Rightarrow$  non preserva F.

2b) Data una decomposizione, applico l'algoritmo che costruisce la tabella:

	A	B	C	D	E	G
ABG	a	a	b	b	b	a
ADE	a	b	b	a	a	b
CDE	b	b	a	a	a	b

$\xrightarrow{A \rightarrow G}$

	A	B	C	D	E	G
ABG	a	a	b	b	b	a
ADE	a	b	b	a	a	a
CDE	b	b	a	a	a	b

$\xrightarrow{GA \rightarrow DB}$

	A	B	C	D	E	G
ABG	a	a	b	a	b	a
ADE	a	a	b	a	a	a
CDE	b	b	a	a	a	b

$\xrightarrow{B \rightarrow AC}$

	A	B	C	D	E	G
ABG	a	a	b	a	b	a
ADE	a	a	b	a	a	a
CDE	b	b	a	a	a	b

$\xrightarrow{AD \rightarrow EB}$

	A	B	C	D	E	G
ABG	a	a	b	a	b	a
ADE	a	a	b	a	a	a
CDE	b	b	a	a	a	b

$\Rightarrow$  l'algoritmo non fa modifiche e termina, non vi e' una riga di sole "a", quindi non ha un Join senza perdita.

3) Supponiamo di avere un file di 12.000.000 record. Ogni record occupa 275 byte, di cui 25 per il campo chiave. Ogni blocco contiene 2048 byte. Un puntatore a blocco occupa 5 byte. Usiamo una organizzazione B-tree in cui ogni blocco del file principale punta al prossimo blocco nel file principale (B+tree). I blocchi sia del file principale sono pieni al massimo mentre i blocchi del file indice sono pieni al minimo. Calcolare:

Da un blocco del MainFile, escludo 5 byte di puntatore, ogni blocco contiene  $\lfloor 2043/275 \rfloor = 7$  record quindi, si ha che il numero di blocchi del MainFile e':  $\lceil \frac{12000000}{7} \rceil = 1714286$ . Una coppia key-pointer pesa 30 byte, in un blocco riempito al minimo ce ne sono  $\lceil \frac{1024-5}{30} \rceil = 34$ , quindi vi sono 35 puntatori.

LIV1:  $\lceil \frac{1714286}{35} \rceil = 48980$  blocchi LIV2:  $\lceil \frac{48980}{35} \rceil = 1400$  blocchi LIV3:  $\lceil \frac{1400}{35} \rceil = 40$  LIV4:  $\lceil \frac{40}{35} \rceil = 2$  LIV5: 1 (RADICE)

Nel file indice ci sono  $48980 + 1400 + 40 + 2 + 1 = 50423$  blocchi. Supponendo che la radice non sia caricata in RAM, per accedere ad un record sono necessari 6 accessi.