

Dato il seguente schema di una base di dati contenente informazioni su eventi culturali

```

EVENTO(Id, Titolo, DataInizio, DataFine, CostoBiglietto)
ARTISTA(Id, Nome, Cognome, NomeArte, CittàNascita)
PARTECIPA(IdEvento, IdArtista)
SPETTATORE(Ce, Nome, Cognome, DataNascita, CittàNascita)
BIGLIETTO(IdSpettatore, IdEvento)
  
```

1a) Dati (nome, cognome e nome d'arte) degli artisti che hanno partecipato ad eventi iniziati nel 2022 e terminati nel 2023 con costo del biglietto compreso tra 80 e 150 euro oppure minore di 50 euro

1b) Dati degli eventi a cui hanno partecipato solo artisti e spettatori di Roma.

1a) $E = \sigma_{(10/1/2022 \leq \text{DataInizio} \leq 31/12/2022) \wedge (10/1/2023 \leq \text{DataFine} \leq 31/12/2023) \wedge ((80 \leq \text{CostoBiglietto} \leq 150) \vee (\text{CostoBiglietto} < 50))}$ EVENTO

Query finale: $Q = \pi_{\text{Nome, Cognome, NomeArte}}((\sigma_{\text{IdEvento} = \text{IdEvento}}(\text{PARTECIPA})) \bowtie \sigma_{\text{IdArtista} = \text{IdArtista}}(\text{ARTISTA}))$

1b) $\text{ANR} = \sigma_{\text{CittàNascita} \neq \text{ROMA}}(\text{ARTISTA})$ $\text{SNR} = \sigma_{\text{CittàNascita} \neq \text{ROMA}}(\text{SPETTATORE})$

Eventi ConNRA = $\pi_{\text{IdEvento}}(\text{PARTECIPA} \bowtie \sigma_{\text{IdArtista} = \text{IdArtista}}(\text{ANR}))$ + eventi con almeno un artista non romano

Eventi ConNRS = $\pi_{\text{IdEvento}}(\text{BIGLIETTO} \bowtie \sigma_{\text{IdSpettatore} = \text{IdSpettatore}}(\text{SNR}))$ + eventi con almeno uno spettatore non romano

Eventi ConUnNonR = $\text{EVENTO} \bowtie_{\text{Id} = \text{IdEvento}}(\text{Eventi ConNRA} \cup \text{Eventi ConNRS})$ Query finale: $Q = \text{EVENTI} - \text{Eventi ConUnNonR}$

2a) Dati lo schema di relazione $R = ABCDE$, l'insieme di dipendenze funzionali $F = \{BD \rightarrow A, ED \rightarrow AC, DA \rightarrow C\}$ e la decomposizione $\rho = \{ABCD, ADE\}$ di R , dire se ρ preserva F e illustrare il procedimento seguito per giungere alla risposta

2b) Dati lo schema di relazione $R = ABCDEG$, l'insieme di dipendenze funzionali $F = \{B \rightarrow AC, A \rightarrow G, G \rightarrow C, CG \rightarrow B\}$ e la decomposizione $\rho = \{ABC, BEG, ADE\}$ di R , dire se ρ ha un join senza perdita e illustrare il procedimento seguito per giungere alla risposta

2a) Devo controllare che $F \subseteq G :: \bigcup_{i=1}^2 \pi_{R_i}(F)$, ossia che $G \subseteq F \wedge F \subseteq G$, la prima è banale data la costruzione di G , per la seconda, verifico che, se $X \rightarrow Y \in F$, allora $Y \in X_G^+$. Applico l'algoritmo:

$$\text{ED} \rightarrow \text{AC}: Z_0 = \text{ED} \quad S_0 = \frac{(\text{ED} \cap \text{ABCD})_F^+ \cap \text{ABCD}}{(\text{ED} \cap \text{ADE})_F^+ \cap \text{ADE}} = \frac{D_F^+ \cap \text{ABCD}}{E_D^+ \cap \text{ADE}} = \frac{D \cap \text{ABCD}}{A \cap \text{ADE}} = \text{ADE} \subseteq Z_0? \text{ NO} \Rightarrow Z_1 = \text{ADU} \cap \text{ADE} = \text{ADE}$$

$$S_1 = \frac{(\text{ADE} \cap \text{ABCD})_F^+ \cap \text{ABCD}}{(\text{ADE} \cap \text{ADE})_F^+ \cap \text{ADE}} = \frac{((\text{AD})_F^+ \cap \text{ABCD}) \cap \text{ADE}}{A \cap \text{ADE}} = \text{ADCU} \cap \text{ADE} = \text{ADCE} \subseteq Z_1? \text{ NO} \Rightarrow \text{non continuo, so già che } \text{AC} \in \text{ED}_G^+$$

$BD \rightarrow A$ e $DA \rightarrow C$ non vanno controllate, dato che vedono tutti in un elemento della decomposizione. ρ preserva F .

2b) Applico l'algoritmo che sfrutta un istanza, che verrà resa legale:

	A	B	C	D	E	G
ABC	a	a	a	b	b	b
BEG	b	a	a	b	a	a
ADE	a	b	b	a	a	b

 \Rightarrow

	A	B	C	D	E	G
ABC	a	a	a	b	b	b
BEG	a	a	a	b	a	a
ADE	a	b	b	a	a	b

 \Rightarrow

	A	B	C	D	E	G
ABC	a	a	a	b	b	a
BEG	a	a	a	b	a	a
ADE	a	b	b	a	a	a

 \Rightarrow

	A	B	C	D	E	G
ABC	a	a	a	b	b	a
BEG	a	a	a	b	a	a
ADE	a	b	a	a	a	a

 \Rightarrow

	A	B	C	D	E	G
ABC	a	a	a	b	b	a
BEG	a	a	a	b	a	a
ADE	a	b	a	a	a	a

 \Rightarrow

\Rightarrow C'è una riga con sole a, ρ ha un Join senza perdita.

3) Supponiamo di avere un file di 16.500.000 record. Ogni record occupa 240 byte, di cui 25 per il campo chiave. Ogni blocco contiene 2048 byte. Un puntatore occupa 5 byte. Usiamo una organizzazione B-tree con i blocchi sia del file principale che del file indice pieni al minimo. Calcolare:

- il numero di blocchi del file principale
- il numero di blocchi del file indice
- il numero di accessi necessari per ricercare un record del file principale

In un blocco foglia pieno al minimo, entrano $\lceil \frac{1024}{240} \rceil = 5$ record, quindi per il MainFile sono necessari $\lceil \frac{16.500.000}{5} \rceil = 3.300.000$ blocchi. Una coppia chiave-puntatore occupa 30 byte, in un blocco

pieno al minimo ne entrano $\lceil \frac{1024-5}{30} \rceil = 34 \Rightarrow$ quindi ci sono 35 puntatori. LIV1: $\lceil \frac{3.300.000}{35} \rceil = 94.286$

LIV2: $\lceil \frac{94.286}{35} \rceil = 2694$ LIV3: $\lceil \frac{2694}{35} \rceil = 77$ LIV4: $\lceil \frac{77}{35} \rceil = 3$ LIV5: $\lceil \frac{3}{35} \rceil = 1$ RADICE.

I blocchi del File indice sono $94.286 + 2694 + 77 + 3 + 1 = 97.061$. Supponendo che la radice non sia caricata in RAM, sono necessari 6 accessi.