

## Basi di Dati Modulo 1 Canali unificati

23 marzo 2023

1) 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(#idt, #idf, numero)

ORARIO(#idt, orapart)

NOTE

Gli attributi sottolineati costituiscono le chiavi delle relazioni

FERMATA.#id e TRATTA.#id sono identificatori (con significato diverso!)rispettivamente di una fermata e di una tratta

FERMATA.nome, FERMATA.via e FERMATA.comune sono informazioni aggiuntive sulla FERMATA (il nome è quello convenzionale assegnato dalla compagnia di trasporti)

TRATTA.#idp e TRATTA.#ida sono identificatori rispettivamente della fermata di partenza (la prima su una tratta) e della fermata di arrivo (l'ultima su una tratta)

Una TRATTA è caratterizzata da un PERCORSO, cioè da una sequenza di fermate

PERCORSO.#idt identifica una tratta, PERCORSO.#idf identifica una fermata sul percorso della tratta, PERCORSO.numero indica il numero d'ordine di una certa fermata in un dato percorso (un percorso contiene più fermate e una fermata si può trovare lungo più percorsi)

ORARIO riporta gli orari di partenza (più orari al giorno) per ogni tratta. ORARIO.#idt è il codice di una tratta ORARIO.orapart memorizza informazioni sugli orari di partenza (dalle 5:00 alle 22:00) delle corse relative alle varie tratte (una tratta viene effettuata in più orari e alla stessa ora partono corse relative a più tratte)

TRATTA è usato col significato di LINEA

Esprimere in algebra relazionale le seguenti interrogazioni:

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)

$$PARTONO_G1 = \sigma_{\substack{\text{comune} \\ \text{"GUBBIO"}}} (TRATTA \bowtie FERMATA) \quad IDP = 10$$

$$PARTONO_G1 \cap Q1 = \sigma_{\substack{Q1:00 \\ \wedge \\ Q1:00 \leq 3:00}} (PARTONO_G1 \bowtie_{\substack{TRATTA: 10 \\ IDP}} Q1)$$

$$ARRIVANO_A2 = \sigma_{\substack{\text{comune} \\ \text{"Rome"}}} (FERMATA)$$

$$OUT = \pi_{\substack{TRATTA: 10, \\ \text{Nome} \\ \text{via}}} (PARTONO_G1 \cap Q1 \bowtie_{\substack{IDA \\ IDP}} ARRIVANO_A2)$$

1b)

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

$$Q1 = \sigma_{\substack{Q1:00 \leq 6:00}} (Q1)$$

$$TRATTA_F_76 = \pi_{\substack{TRATTA: 10, \\ \text{FERMATA: Nome}}} (Q1 \bowtie_{\substack{IDT = ID \\ IDT = 10}} TRATTA \bowtie FERMATA)$$

$$ALL = \pi_{\substack{TRATTA: 10, \\ \text{FERMATA: Nome}}} (Q1 \bowtie_{\substack{IDT = ID \\ IDT = 10}} TRATTA \bowtie FERMATA)$$

$$OUT = ALL - TRATTA_F_76$$

2)

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

dire se  $\rho$  preserva  $F$  e illustrare il procedimento seguito per giungere alla risposta

2b) Dati lo schema di relazione  $R=ABCDEG$ , l'insieme di dipendenze funzionali

$F=\{A \rightarrow G, GA \rightarrow DB, B \rightarrow AC, BE \rightarrow A, AD \rightarrow EB\}$  e la decomposizione  $\rho=\{ABG, ADE, CDE\}$  di  $R$ ,

dire se  $\rho$  ha un join senza perdita e illustrare il procedimento seguito per giungere alla risposta

---

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<sup>+</sup>-tree). I blocchi sia del file principale sono pieni al massimo mentre i blocchi del file indice sono 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

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

dire se  $\rho$  preserva  $F$  e illustrare il procedimento seguito per giungere alla risposta

2b) Dati lo schema di relazione  $R=ABCDEG$ , l'insieme di dipendenze funzionali

$F=\{A \rightarrow G, GA \rightarrow DB, B \rightarrow AC, BE \rightarrow A, AD \rightarrow EB\}$  e la decomposizione  $\rho=\{ABG, ADE, CDE\}$  di  $R$ ,

dire se  $\rho$  ha un join senza perdita e illustrare il procedimento seguito per giungere alla risposta

2a) Ho alcune dipendenze problematiche:

-  $AD \rightarrow E$

-  $BD \rightarrow E$

Inizio ad applicare l'ALG con  $AD \rightarrow E$

$(AD)^+_F$ :

$Z_0 = AD$

$$S_0 = (AD \cap ABCE)^+_F \cap ABCE \cup (AD \cap CDE)^+_F \cap CDE =$$

$$= (A)^+_F \cap ABCE \cup (D)^+_F \cap CDE$$

$$= A \cap ABCE \cup D \cap CDE$$

$$= A \cup D = AD$$

So  $\nexists Z_c \rightarrow$  continuo

$Z_1 = AD$

$$S_1 = (AD \cap ABCE)^+_F \cap ABCE \cup (AD \cap CDE)^+_F \cap CDE =$$

$$= (A)^+_F \cap ABCE \cup (D)^+_F \cap CDE$$

$$= A \cap ABCE \cup D \cap CDE$$

$$= A \cup D = D$$

$$S_1 \subseteq Z_1 \Rightarrow m_1 \text{ fermo}$$

$$(AD)_G^T = \{AD\} \quad p \in \Rightarrow \text{NON PRESERVA}$$

2b)

	A	B	C	D	E	G
ABG	a	a	<del>b<sub>1</sub></del> <sub>b<sub>2</sub></sub>	<del>b<sub>1</sub></del> <sub>a<sub>1</sub></sub>	<del>b<sub>1</sub></del> <sub>a<sub>1</sub></sub>	a
ADE	a	<del>b<sub>2</sub></del> <sub>a<sub>1</sub></sub>	b <sub>2</sub>	a	a	<del>b<sub>2</sub></del> <sub>a<sub>1</sub></sub>
CDE	b <sub>3</sub>	b <sub>3</sub>	a	a	a	b <sub>3</sub>

I<sup>α</sup>      II<sup>o</sup>

- A → G ✓      ✓

- GA → DB ✓      ✓

NON CAMBIO NIENTE

↓

MI FERMO

↓

⚡ Diga ca tutte "a" ⇒ NON HA JON SORA  
PERDITA

- B → A ✓      ✓

- BE → A ✓      ✓

- AD → EB ✓      ✓

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:

- 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

$$2a) \text{ Tot Blocc FP: } \frac{\lceil \text{No Rec} \rceil}{\text{Recal} \times \text{Blocco}} = 1714286$$

$$\text{Recal} \times \text{Blocco} = \frac{\text{Block size} - P_{\text{size}}}{L_{\text{P size}}} = 7$$

$$2b) d = \frac{\lceil (\text{Block size}) - P_{\text{size}} \rceil}{P_{\text{size}} + \text{key size}} \rightarrow \frac{\lceil 1024 - 5 \rceil}{5 + 25} = 39 + 1 = 35 \text{ puntatori}$$

CHIAVI

$$\text{liv } d = \text{Tot Blocc FP}$$

$$\text{liv } 1 = \frac{\lceil 1714286 \rceil}{35} = 48980$$

$$\text{liv } 2 = \frac{\lceil 48980 \rceil}{8} = 1400$$

$$\text{liv } 3 = \frac{\lceil 1400 \rceil}{35} = 40$$

$$\text{liv } 5 = \frac{\lceil 2 \rceil}{35} = 1$$

$$\text{liv } 4 = \frac{\lceil 40 \rceil}{35} = 2$$

$$\text{Tot Blocc FI: } 48980 + 1400 + 40 + 2 + 1 = 50423$$

$$2c) \text{ Tot Accessi} = \text{no Liv} = 6 \text{ Accessi}$$