

Basi di Dati Title



Inotes

Argomenti

- Progettazione concettuale
- Modello E-R
- Modello relazionale
- Progettazione logica
- Algebra e calcolo relazionale
- Normalizzazione
- Indici B-Alberi

MODELLO E-R

Generalizzazioni

Totale se ogni occorrenza dell'entità genitore è occorrenza di **almeno una** delle entità figlie, altrimenti è **parziale**.

Esclusiva se ogni occorrenza dell'entità genitore è occorrenza di **al più una** delle entità figlie, altrimenti è **sovrapposta**.



CHIAVE

Sia $R(A_1, \dots, A_n)$ uno schema di relazione e $K \subseteq \{A_1, \dots, A_n\}$ sottoinsieme della relazione:




- K è **superchiave** di R se per ogni istanza r di R non esistono in r due n-uple t_1, t_2 tali che $t_1[K] = t_2[K]$
- K è una **superchiave minimale** di R se: 1) K è una superchiave di R ; 2) non esiste $K' \subset K$ tale che K' è superchiave di R .
- K è una **chiave** di R se è una superchiave minimale di R .

VINCOLO D'INTEGRITÀ REFERENZIALE

Un vincolo d'integrità referenziale tra un attributo A di una relazione R_1 e un attributo B di una relazione R_2 , impone che per ogni istanza della base di dati i valori dell'attributo A in tutte le n-uple di R_1 , compaiono come valore di qualche n-upla dell'attributo B in qualche n-upla di R_2 .

ALGEBRA RELAZIONALE

- Operazioni insiemistiche:

- Unione (medesimi attributi) 
- Intersezione (medesimi attributi) 
- Differenza (medesimi attributi) 

- Ridenominazione (ρ)

- $\rho(\text{Paternità})$
Genitore ← Padre

PADRE	FIGLIO
↓	
GENITORE	FIGLIO
- $\rho(\text{Paternità})$
Genitore ← Madre

MADRE	FIGLIO
↓	
GENITORE	FIGLIO

$$\rho_{B_1, \dots, B_n \leftarrow A_1, \dots, A_n}(r) = \{t \text{ su } B_1, \dots, B_n \mid \exists t' \in r (t[B_1..B_n] = t'[A_1..A_n])\}$$

- Selezione (restituisce un sottoinsieme) (σ)

- σ (Impiegati)
Filiale = "Milano" AND Stipendio > 50

$$\sigma_F(r) = \{t \mid t \in r \text{ e } t \text{ soddisfa le formule } F\}$$

- Proiezione (ignoriamo un insieme di attributi) (π)

- π (Impiegati)
MATRICOLA, COGNOME

MATRICOLA	NOME	COGNOME	STIPENDIO
-----------	-----------------	---------	----------------------

$$\pi_{A_1, \dots, A_k}(r) = \{t[A_1 \dots A_k] \mid t \in r\}$$

- Join (combina tabelle, no attributi in comune) (X)

• σ (Studenti x Esami)

Matricola = Studente

Studenti

MATRICOLA	NOME	COGNOME
1000	Aldo	Rossi
1010	Ana	Verdi
2000	Carlo	Bruni

Esami

STUDENTE	CORSO	VOTO
1000	DB	28
1010	DB	30
1000	Aldo	29

Studenti x Esami

MATRICOLA	NOME	COGNOME	STUDENTE	CORSO	VOTO
1000	Aldo	Rossi	1000	DB	28
1000	"	"	1000	Aldo	29

- Theta Join (attributi distinti) (\bowtie_{θ})

• $R_1 \bowtie_{\theta} R_2 \equiv \sigma_F(R_1 \times R_2)$ $F = \theta$

• Impiegati \bowtie Reparti

Reparti + Codice

IMPIEGATO	REPARTO	CODICE	CAPO
Rossi	A	A	Mari
Neri	B	B	Bruni
Bianchi	B	B	Bruni

Impiegati

IMPIEGATO	REPARTO
Rossi	A
Neri	B
Bianchi	B

Reparti

CODICE	CAPO
A	Mari
B	Bruni

- Join naturale (attributi da confrontare devono avere lo stesso nome) (\bowtie)

• Impiegati \bowtie Reparti

Impiegati

IMPIEGATO	REPARTO
Rossi	A
Neri	B
Bianchi	B

Reparti

REPARTO	CAPO
A	Mari
B	Bruni

IMPIEGATO	REPARTO	CAPO
Rossi	A	Mari
Neri	B	Bruni
Bianchi	B	Bruni

$$r_1 \bowtie r_2 = \{t_{su} x_1 \cup x_2 \mid \exists t_{12}, t_1 \in r_1 \wedge t_2 \in r_2 \wedge t = t_{12}^*\}$$

$$r_1 \bowtie r_2 = \{t_{su} x_1 \cup x_2 \mid t[x_1] \in r_1 \wedge t[x_2] \in r_2\}$$

$$* t[x_1] = t_1, t[x_2] = t_2$$

Se non ci sono attributi in comune il join naturale equivale al prodotto cartesiano.

SELEZIONE CON VALORI NULLI

Se il valore è "null" la condizione sarà falsa.

VISTE

Contenuto visualizzato diverso per utente.

Supervisione

IMPIEGATO	CAPO
Rossi	Mari
Neri	Bruni
Bianchi	Bruni
Neri	Long

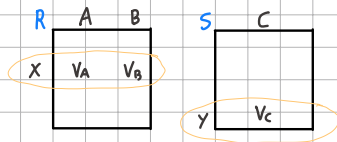
CALCOLO RELAZIONALE

Calcolo sulle tuple con restrizione di range. Nel calcolo abbiamo delle variabili che variano su una tupla.

$|x(R), y(S)|$ condizione

$x.A$ valore A della tupla x

$\{x.A, x.B, y.C \mid x(R), y(S) \mid \text{condizione}\}$
select from where
Target list Range list Formula



NORMALIZZAZIONE

Lo schema rispetta delle proprietà

Impiegato \rightarrow Stipendio (per ogni impiegato c'è un solo stipendio) $\forall t_1, t_2 (t_1[\text{Impiegati}] = t_2[\text{Impiegati}] \Rightarrow t_1[\text{Stipendio}] = t_2[\text{Stipendio}])$

- Dipendenze Funzionali

$R(A_1, \dots, A_n)$ $y, z \subseteq \{A_1, \dots, A_n\}$

$y \rightarrow z$ l'insieme y determina gli attributi in z

La dipendenza funzionale vale su R se per ogni istanza r di R $\forall t_1, t_2$ $t_1[y] = t_2[y]$ allora $t_1[z] = t_2[z]$

- BCNF (Boyce Codd Normal Form)

Dato uno schema di relazione R e una dipendenza funzionale $X \rightarrow Y$, R è in BCNF se per ogni dipendenza funzionale $X \rightarrow Y$, X contiene una chiave di R

- 3FN (Terza forma normale)

Forma normale più debole. Anche se a sinistra non ho una chiave, va bene se a destra ho un attributo appartenente alla chiave.

Una relazione r è in 3FN se per ogni dipendenza funzionale $X \rightarrow Y$ definita su r, è verificata

almeno una delle seguenti condizioni:

- X contiene una chiave K di r;

- ogni attributo in y è contenuto in almeno una chiave di r.

In caso negativo si usa l'algoritmo di decomposizione.

- Algoritmo per decomporre

Per ogni dipendenza creo una relazione.

Se faccio il Join naturale devo ottenere la tabella iniziale. (decomposizione senza perdita d'informazioni)

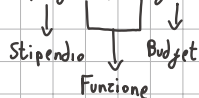
R (Impiegato, Stipendio, Progetto, Bilancio, Funzione)

1) Impiegato → Stipendio

2) Progetto → Budget

3) Impiegato, Progetto → Funzione

{Impiegato, Progetto}



R₁ (Impiegato, Progetto)

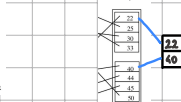
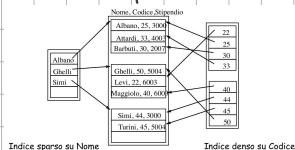
R₂ (Progetto, Budget)

R₃ (Impiegato, Progetto, Funzione)

B - ALBERI

Sulla chiave primaria i DBMS stabiliscono in automatico degli indici.

- Posso aggiungere livelli di indici per gli indici densi.



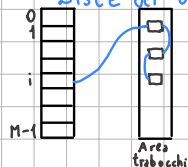
- B⁺-alberi

Nei B⁺-alberi nelle foglie sono contenute le chiavi e puntatori all'area della tabella, tutte le chiavi compaiono nelle foglie, le foglie sono tra loro doppiamente linkate

- Struttura procedurale, Hash table, statica

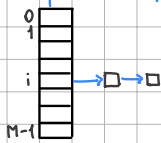
Posso usare un'area dei trabocchi dove conservare le n-uple che non riesco ad inserire nella tabella normale.

• Liste dei trabocchi



Le liste sono collegate all'indice in cui dovrebbero essere inserite.

• Sequenza di pagine



Più efficiente delle liste.
In ogni pagina ci sono più tuple

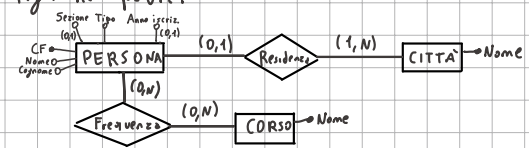
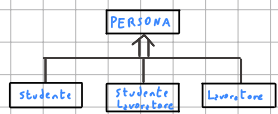
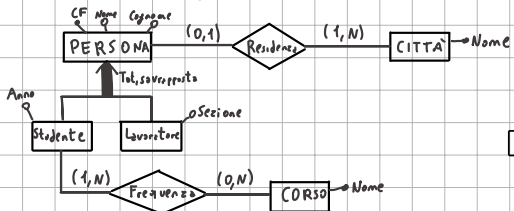
Se la dimensione dei dati varia nel tempo conviene utilizzare i B⁺-Alberi

RISTRUTTURAZIONE

- 1) Analisi Ridondanze
- 2) Eliminazione delle generalizzazioni.
- 3) Partizionamento/accorpamento di entità e relazioni
- 4) Scelta degli identificatori primari

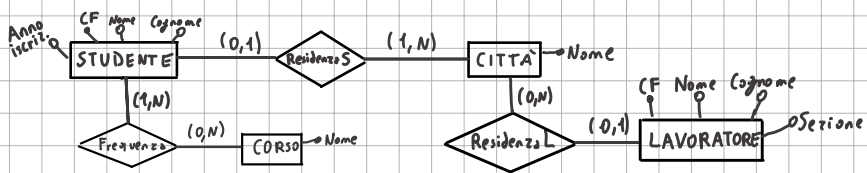
2

1- Trasforma le generalizzazioni da sovrapposte ad esclusive. Poi accorpo i figli nel padre.

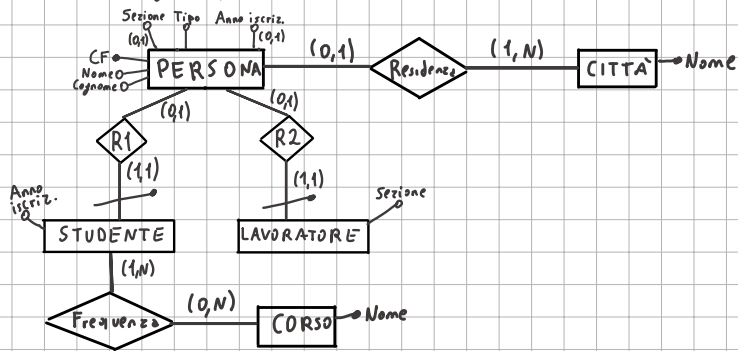


2-Elimino l'entità padre (solo se la generalizzazione è totale)

No relazioni ternarie



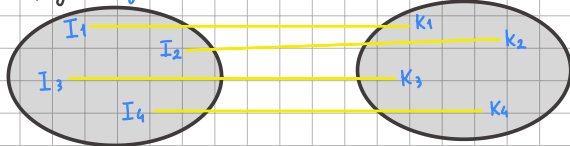
3- Mantengo le entità Figlie padre



3

Impiegato (anagrafico)

Impiegato (lavorativo)



4

