

Esercitazione
normalizzazione

Algoritmo per chiusura di un insieme di attributi X (X^+)

1. $X^{(0)} := X, j = 0$
2. *REPEAT*
3. $X^{(j+1)} := X^{(j)} \cup \{A | \exists Y \rightarrow Z \in F | A \in Z \wedge Y \subseteq X^{(j)}\}$
4. *UNTIL* ($X^{(j+1)} = X^{(j)}$)
5. *SET* $X^+ := X^{(j)}$

Insiemi di Dipendenze Minimali

- Un insieme di dipendenze funzionali F è **minimale** se:

1. Ogni lato destro di una dipendenza è un singolo attributo.

2. Per ogni dipendenza $X \rightarrow A$ in F , $F \setminus \{X \rightarrow A\}$ non è equivalente a F

3. Per ogni $X \rightarrow A$ in F e $Z \subset X$, $F \setminus \{X \rightarrow A\} \cup \{Z \rightarrow A\}$ non è equivalente a F

- La regola 2 garantisce che nessuna dipendenza in F è ridondante. La regola 3 garantisce che nessun attributo in qualunque primo membro sia ridondante.

Algoritmo ricerca chiavi

Begin

$$ND = T - \bigcup_{X \rightarrow A \in F} A$$

$$SD = \bigcup_{X \rightarrow A \in F} X \cap \bigcup_{X \rightarrow A \in F} A$$

$$Cand = [ND :: (SD)]$$

$$Keys = \phi$$

while $Cand \neq \phi$ do

$$X :: (Y) = First(Cand)$$

$$Cand = Rest(Cand)$$

if $\nexists k \subset X \mid k \in Keys$ then

if $X^+ = T$ then

$$Keys = Keys \cup \{X\}$$

else

$$A_1, \dots, A_n = Y - X^+$$

for $i = 1 \dots n$ do

$$cand = cand \cup XA_i :: (A_{i+1} \dots A_n)$$

end for

end if

end if

end while

end

Decomposizioni che preservano i dati (loss-less joins)

- Dato uno schema R con un insieme F di dipendenze funzionali , una sua decomposizione $d=\{R_1, R_2, \dots, R_k\}$ si dice che **preserva i dati** (o che ha loss-less joins) se per ogni relazione r di R che soddisfa tutte le dipendenze di F si ha:

$$r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_k}(r)$$

Algoritmo per controllare se una decomposizione preserva i dati

Input:

$R = \{A_1, A_2, \dots, A_n\}$, F , $d = \{R_1, R_2, \dots, R_k\}$

Output:

Yes/No se d preserva i dati.

INIZIALIZZAZIONE:

Consideriamo una matrice

$$M = \{R_1, R_2, \dots, R_k\} \times \{A_1, A_2, \dots, A_n\}$$

dove nell'elemento $R_i A_j$ mettiamo a_j se A_j è in R_i

altrimenti mettiamo b_{ij} .

- ITERAZIONE:

Applichiamo finché è possibile ogni dipendenza $X \rightarrow Y$ in F nel seguente modo: se esistono due righe di M che coincidono su X allora facciamole coincidere anche in Y :

- se ho uno dei due a_j allora cambiamo l'altro (b_{ij}) in a_j ;
 - Altrimenti prendiamo uno dei due e lo facciamo uguale all'altro.
- Se durante il passo precedente si produce la riga $\mathbf{a_1 a_2 \dots a_n}$ allora rispondi *YES* (*La decomposizione preserva i dati*)
 - Altrimenti rispondi *NO*
(*La decomposizione non preserva i dati*)

Controllo di conservazione dati su decomposizione con 2 relazioni

Teorema

- *Se $d=\{R_1, R_2\}$ è una decomposizione di R con dipendenze F . Allora d preserva i dati se e solo se soddisfa una delle due:*

$$F \vdash R_1 \cap R_2 \rightarrow R_1 - R_2$$

$$F \vdash R_1 \cap R_2 \rightarrow R_2 - R_1$$

Proiezione di insieme di dipendenze

- La *proiezione* $\pi_Z(F)$ di F su un insieme Z di attributi è l'insieme delle dipendenze $X \rightarrow Y$ appartenenti a F^+ tali che $XY \subseteq Z$

Algoritmo per il calcolo della proiezione di un insieme di dipendenze

Input: $(R(A_1, A_2, \dots, A_n), F)$

Output: Una copertura della proiezione di F su $T \subseteq (A_1, A_2, \dots, A_n)$

Begin

$$\pi_T(F) = \emptyset$$

for each $Y \subset T$ **do**

$$Z = Y_F^+ - Y$$

$$\pi_T(F) = \pi_T(F) \cup \{Y \rightarrow (Z \cap T)\}$$

end for

return $\pi_T(F)$

end

Algoritmo per controllare se una decomposizione preserva le dipendenze funzionali

- **Input:** $R=\{A_1, A_2, \dots, A_n\}$, F , $d=\{R_1, R_2, \dots, R_k\}$
- **Output:** Yes/No se d preserva F .
- **IDEA DELL'ALGORITMO:**
 - Sia $G = \bigcup_{i=1, \dots, k} \pi_{R_i}(F)$.
 - Dobbiamo provare che G è equivalente ad F (senza calcolare G).
 - Basta provare che per ogni $X \rightarrow Y$ in F , Y è contenuto in X^+ calcolato rispetto a G . Per questo calcolo basta fare la chiusura di X rispetto alle varie proiezioni di F sulle R_i .

FORMA NORMALE BCNF

- Uno schema relazionale R con dipendenze F si dice in **Forma Normale di Boyce-Codd** (BCNF) se per ogni $X \rightarrow A$ di F^+ , se A non appartiene ad X allora
 - X è una *superchiave* di R , cioè è o contiene una chiave.
- Si può dimostrare che se tutte le dipendenze di F sono del tipo $X \rightarrow A$ (F sia minimale), allora basta verificare la suddetta proprietà solo per gli elementi di F e non di F^+ .

TERZA FORMA NORMALE

- Uno schema relazionale R con dipendenze F si dice in **Terza Forma Normale** (3NF) se per ogni $X \rightarrow A$ di F^+ , se A non appartiene ad X allora
 - X è una *superchiave* di R oppure A è *primo*, cioè appartiene a qualche chiave.
- Si può dimostrare che se tutte le dipendenze di F sono del tipo $X \rightarrow A$, allora basta verificare la suddetta proprietà solo per gli elementi di F e non di F^+ .

INPUT: Schema $R(T)$ e dipendenze F

OUTPUT: Decomposizione che preserva i dati tale che ogni componente sia in BCNF rispetto alla proiezione di F su quella componente.

begin

$$\rho = \{(R_1(T_1), F_1) = (R(T), F)\}, n = 1$$

while $\exists (R_i(T_i), F_i) \in \rho$ non in BCNF per $X \rightarrow A$ **do**

$$n := n + 1$$

$$T' = X^+$$

$$F' = \pi_{T'}(F_i)$$

$$T'' = T_i - (T' - X)$$

$$F'' = \pi_{T''}(F_i)$$

$$\rho = \rho - \{(R_i(T_i), F_i)\} \cup \{(R_i(T'), F'), (R_n(T''), F'')\}$$

end while

end

ALGORITMO PER 3NF

- Se ci sono attributi *non presenti in F* essi possono essere raggruppati in un solo schema ed eliminati.
- Se una dipendenza di F coinvolge *tutti gli attributi* di R , allora ritorna (R) .
- Altrimenti ritorna la decomposizione fatta da tutti gli XA tali che $X \rightarrow A$ appartiene ad F .

E' possibile avere tutto?

- E' possibile avere decomposizioni che preservano i dati o le dipendenze ed in cui tutte le componenti sono in forma normale?
 - SI per *conservazione dei dati e BCNF*
 - SI per *conservazione dei dati e delle dipendenze e 3NF.*
 - **NO** per *la conservazione delle dipendenze e BCNF*

- Data la relazione $R(A, B, C, D, E)$ e le dipendenze funzionali $A \rightarrow B$, $BC \rightarrow D$ e $DE \rightarrow A$,
- determinare le chiavi di R ;
- specificare se R è in 3NF o in Boyce-Codd, motivando la risposta.
- Verificare se lo schema R è in BCNF ed eventualmente decomporlo

Si consideri lo schema relazionale $R(A,B,C,D,E)$ e il seguente insieme di dipendenze funzionali $F = \{A \rightarrow B, B \rightarrow C, AC \rightarrow D, D \rightarrow E\}$

- Trovare le chiavi
- Decomporre lo schema in BCNF

- Si Consideri il seguente schema Persona(NOME, CF, NumeroTelefonico, Città) con la seguente dipendenza funzionale $CF \rightarrow \text{Nome, Città}$. E' in BCNF (Suggerimento, fare un'istanza d'esempio)?
 - Indicare la chiave primaria e eventualmente decomporlo in BCNF.

- Si consideri lo schema di relazione $R(A, B, C, D, E, F, G, H, I, J)$ ed il relativo insieme di dipendenze funzionali $F = \{ABD \rightarrow E, AB \rightarrow G, B \rightarrow F, C \rightarrow J, CJ \rightarrow I, G \rightarrow H\}$.
 - Determinare l'insieme delle chiavi di R .
 - Decomporlo in BCNF

- Si considerino lo schema di relazione $R(A, B, C, D, E, F)$ e l'insieme di dipendenze associato: $F = \{A \rightarrow B, C \rightarrow AD, AF \rightarrow EC\}$.
 - Si determinino le chiavi candidate di R .
 - Si stabilisca se R è in 3NF. Qualora non lo sia, si definisca una decomposizione di R in 3NF che conservi le dipendenze date.

Si considerino lo schema di relazione $R(A, B, C, D, E, G)$ e l'insieme di dipendenze associato: $F = \{E \rightarrow D, C \rightarrow B, CE \rightarrow G, B \rightarrow A\}$.

- Si stabilisca se R è in 3NF. Qualora non lo sia, si definisca una decomposizione di R in 3NF che conservi le dipendenze date.
- Se R non è in BCNF, determinare una decomposizione di R in BCNF

Dato il seguente schema relazionale che memorizza il sistema informativo di una scuola che registra informazioni relative agli iscritti, ai docenti, alle classi e ai programmi insegnati in ciascuna classe:

SCUOLA(Classe, CFDocente, NomeDocente, NomeMateria, CFStudente, NomeStudente)

E sapendo che:

- Il programma di una classe è costituito dall'elenco delle materie insegnate
- Sono registrati gli studenti suddivisi in classi (non esiste storico, dunque per ogni studente si memorizza solo la classe che sta attualmente frequentando)
- Ciascuna materia in ogni classe viene insegnata da un solo docente
- Non viene registrato l'orario delle lezioni

*Qual è la chiave della relazione SCUOLA? Motivando la risposta mediante le dipendenze funzionali. **Decomporre** lo schema in terza forma normale.*

- Dato lo schema di relazione di una discoteca:
DISCOTECA(NomeSala, TipoMusica, CapacitàSala, CodDJ, NomeDJ, SpecialitàDJ, Serata, FasciaOraria, DataInizioStagione, DataFineStagione)
- Sapendo che:
 1. La discoteca deve stampare i programmi della stagione che hanno validità solo in un certo periodo (es. Apertura Estiva 2014: 1 maggio 2014 – 30 settembre 2014)
 2. La discoteca è composta da diverse sale, ognuna dedicata a un solo tipo di musica in una certa serata (il mercoledì rock in sala blu e house in sala rossa, il sabato...)
 3. Per ogni serata di apertura settimanale vengono indicati nel volantino il tipo di musica e i DJ di ogni sala
 4. In una serata, in ogni sala si alternano più DJ nelle diverse fasce orarie.
- Evidenziare la **chiave** della relazione, elencare tutte le **dipendenze funzionali non banali** presenti nello schema, **decomporre lo schema in 3NF**.

Si consideri il seguente schema relativo a dei terreni di una provincia:

Terreni(IdTerreno, Comune, NumTerreno, Dimensione, PrezzoTotale, Proprietario)

con le seguenti assunzioni:

- l'attributo IdTerreno identifica un terreno nell'intero database
- l'attributo NumTerreno è un numero progressivo assegnato al terreno da ciascun comune della provincia indipendentemente (quindi può ripetersi in comuni diversi)
- l'attributo Dimensione rappresenta i metri quadri del terreno
- un terreno può avere più proprietari, ma una persona può possedere al massimo un terreno

Scrivere le dipendenze funzionali e identificare le chiavi.