

Teoria di Armstrong

- Riflessività: se $Y \subseteq X$ allora $X \rightarrow Y$
- Unione: se $X \rightarrow Y$ e $X \rightarrow Z$ allora $X \rightarrow YZ$ dove $YZ = Y \cup Z$
- Transitività: se $X \rightarrow Y$ e $Y \rightarrow Z$ allora $X \rightarrow Z$

Regole:

- Espansione: Dati una dipendenza funzionale $X \rightarrow Y$ e un insieme di attributi W , allora $WX \rightarrow WY$
- Decomposizione: Se $X \rightarrow YZ$, allora $X \rightarrow Y$ e $X \rightarrow Z$
- Pseudotransitività: se $X \rightarrow Y$ e $WY \rightarrow Z$, allora $WX \rightarrow Z$
- Prodotto: $X \rightarrow Y$ e $W \rightarrow Z$ allora $XW \rightarrow YZ$

Chiusura di un insieme F

Algoritmo per il calcolo di X_F^+ dove X è un attributo

1. Parti da X
2. Finchè in F ci sono dipendenze funzionali applicabili ricavo i nuovi attributi e ripeto finchè non di ci sono più d.f. disponibili

Per trovare una superchiave calcolo la chiusura sugli attributi interessati, se la chiusura comprende tutti gli attributi della relazione allora ho una superchiave. Per calcolare la chiave minimale elimino degli attributi dalla superchiave se la chiusura

Decomposizione senza perdita def

Dato uno schema di relazione $R(A)$, dati due sottoinsiemi di attributi $A_1 \subseteq A$ e $A_2 \subseteq A$ con $A_1 \cup A_2 = A$, $\{R_1(A_1), R_2(A_2)\}$ è una decomposizione senza perdita di informazione se e solo se per ogni istanza $r(A)$ di $R(A)$ vale: $r(A) = r_1(A_1) \bowtie r_2(A_2)$ Dove $r_1(A_1) = \pi_{A_1}(r(A))$ e $r_2(A_2) = \pi_{A_2}(r(A))$

Algoritmo normalizzazione BCNF

1. Parto dallo schema di basi di dati SC
2. Cerco all'interno in SC una d.f. non in BCNF
3. Se non esiste una d.f. non BCNF mi fermo

4. Se esiste devo trasformare lo schema

1. Tolgo la relazione non in BCNF da SC e relativi attributi
2. Aggiungo ad SC due nuove relazioni con le relative dipendenze funzionali
3. Torno al passo 2

Verifica finale: unendo le restrizioni delle dipendenze delle nuove relazioni devo trovare la relazione di partenza. Se ciò non accade si può aggiungere un vincolo globale. Conviene iniziare dalla periferia delle dipendenze così da mantenerne il più possibile.

Calcolo di una copertura minimale

Def: un insieme F' di d.f. è un insieme di copertura minimale rispetto a F quando:

1. $F \equiv F'$
2. in ogni $X \rightarrow Y \in F'$ Y è un attributo singolo (non è un requisito necessario)
3. ogni $X \rightarrow Y \in F'$ è priva di attributi estranei
4. ogni $X \rightarrow Y \in F'$ non è ridondante
5. Deconpongo le d.f. in attributo singolo a destra applicando la regola di decomposizione
6. Elimino gli attributi estranei
 1. Calcolo la chiusura di $X \rightarrow Y$ dove X è un insieme di attributi e Y è l'attributo da esaminare
 2. Se il conseguente di X è presente nella chiusura allora l'attributo Y è estraneo e si può eliminare.
7. Elimino le d.f. ridondanti
 1. Prende un d.f. $X \rightarrow Y$ che voglio esaminare
 2. Rimuovo la d.f. dall'insieme
 3. Calcolo la chiusura di X sul nuovo insieme
 4. Se Y è presente nella chiusura di X allora la dipendenza è ridondante e posso eliminarla

Normalizzazione 3NF

1. Calcolo la copertura minimale
2. Creo una relazione per ogni insieme di d.f. che a sinistra hanno gli stessi attributi

3. Cerco sottoinsiemi di relazioni e li accorpo all'insieme maggiore aggiungendo la relativa dipendenza funzionale

Ottimizzazione logica

1. Decomposizione degli AND nelle selezioni
2. Trasferire le selezioni verso le foglie finché è possibile con le proprietà distributive della selezione
3. Trasferire le proiezioni verso le foglie finché è possibile con le proprietà distributive della proiezione.
4. Ridurre a un'unica selezione le selezioni multiple nello stesso nodo dell'albero
5. Riconoscere le sequenze di join
6. Ridurre ad un'unica proiezione le proiezioni multiple
7. Esaminare le varianti dell'albero sintattico dovute alle proprietà associative scegliendo la variante di costo minimo

$CARD(r) = |r|$: cardinalità della relazione $SIZE(t)$: ampiezza della tupla in byte

$VAL(A_i, r)$: il numero di valori distinti che appare nella colonna A_i all'interno della tavola r

$MIN(A_i, r)$: il valore minimo di A_i contenuto in r $MAX(A_i, r)$: il valore massimo di A_i

contenuto in r $NPAGE(r) = CARD(r)/blockfactor$: blockfactor è il numero massimo di tuple che una pagina può contenere.

Stima del costo della selezione: $|\sigma_p(r)| = f_p \cdot |r|$ Dove f_p è il fattore di selettività, varia tra 0 e 1

Predicato p	f_p
$A_i = v$	$1/\text{VAL}(A_i, r)$
$A_i \leq v$	$(v - \text{MIN}(A_i, r))/(\text{MAX}(A_i, r) - \text{MIN}(A_i, r))$
$A_i \geq v$	$(\text{MAX}(A_i, r) - v)/(\text{MAX}(A_i, r) - \text{MIN}(A_i, r))$
$v_1 \leq A_i \leq v_2$	$(v_2 - v_1)/(\text{MAX}(A_i, r) - \text{MIN}(A_i, r))$

Predicato p	f_p
$p_1 \wedge p_2 \wedge \dots \wedge p_n$	$f_{p_1} \cdot f_{p_2} \cdot \dots \cdot f_{p_n}$
$\neg p$	$1 - f_p$
$p_1 \vee p_2 \vee \dots \vee p_n$	$1 - ((1 - f_{p_1}) \cdot (1 - f_{p_2}) \cdot \dots \cdot (1 - f_{p_n}))$

Stima della cardinalità del join: Se esiste un vincolo di integrità referenziale la cardinalità dell'equi-join è: $|r(A) \bowtie_{A_i=B_j} s(B)| = |r(A)| = \text{CARD}(r)$ Se non esiste un vincolo di integrità referenziale: $|r(A) \bowtie_{A_i=B_j} s(B)| = \min\{1/\text{VAL}(A_i, r), 1/\text{VAL}(B_j, s)\} \cdot \text{CARD}(r) \cdot \text{CARD}(s)$