

Introduzione alla Normalizzazione

La normalizzazione è una tecnica di progettazione di database che mira a ridurre la ridondanza dei dati e migliorare l'integrità attraverso l'organizzazione degli attributi in relazioni basate su dipendenze funzionali.

1. Calcolo delle Chiavi di una Relazione

Procedimento:

1. **Identificare gli attributi che non compaiono mai nella parte destra** di dipendenze funzionali (questi fanno parte di ogni chiave candidata)
2. **Calcolare la chiusura** di potenziali insiemi di attributi
3. **Verificare quali chiusure contengono tutti gli attributi** della relazione (queste sono le superchiavi)
4. **Verificare la minimalità** (nessun sottoinsieme proprio ha la stessa chiusura)

Esempio:

Per $R(A,B,C,D,E,F)$ con $G=\{C \rightarrow B, C \rightarrow D, C \rightarrow F, BE \rightarrow C, B \rightarrow A, AD \rightarrow E\}$:

Calcoliamo:

- $C^+ = \{C,B,D,F,A,E\} = R$ (C è una superchiave)
- $AD^+ = \{A,D,E\}$ (non è una superchiave)
- $BE^+ = \{B,E,C,A,D,F\} = R$ (BE è una superchiave)
- $B^+ = \{B,A\}$ (non è una superchiave)

Risultato: C e BE sono le chiavi di R.

2. Verifica della Terza Forma Normale (3NF)

Definizione:

Una relazione è in 3NF se per ogni dipendenza funzionale non banale $X \rightarrow A$:

- X è una superchiave, OPPURE
- A fa parte di qualche chiave

Procedimento:

1. Per ogni dipendenza funzionale $X \rightarrow A$, verificare se X è una superchiave
2. Se X non è una superchiave, verificare se A è parte di almeno una chiave

3. Se nessuna delle due condizioni è soddisfatta, la dipendenza viola la 3NF

Esempio:

Per $R(A,B,C,D,E,F)$ con $G=\{C\rightarrow B, C\rightarrow D, C\rightarrow F, BE\rightarrow C, B\rightarrow A, AD\rightarrow E\}$:

- $B\rightarrow A$ viola la 3NF perché B non è una superchiave e A non è parte di nessuna chiave

3. Decomposizione in 3NF

Algoritmo di sintesi per la 3NF:

1. **Determinare una copertura ridotta** G delle dipendenze funzionali
2. **Partizionare G in sottoinsiemi** G_1, G_2, \dots, G_n tali che le dipendenze funzionali $X\rightarrow A$ e $Y\rightarrow B$ sono insieme se e solo se $X^+ = Y^+$ (hanno la stessa chiusura)
3. **Creare una relazione per ogni sottoinsieme**, con attributi che sono l'unione dei lati sinistri e destri
4. **Eliminare relazioni ridondanti**: se una relazione $R_1(X)$ è contenuta in un'altra relazione $R_2(Y)$ (ovvero $X \subseteq Y$), R_1 viene eliminata
5. **Assicurarsi che almeno una relazione contenga una chiave** della relazione originale; altrimenti, aggiungere una relazione che contenga una chiave

Esempio:

Per $R(A,B,C,D,E,F)$ con $G=\{C\rightarrow B, C\rightarrow D, C\rightarrow F, BE\rightarrow C, B\rightarrow A, AD\rightarrow E\}$:

1. Partizionamento:
 - $\{C\rightarrow B, C\rightarrow D, C\rightarrow F, BE\rightarrow C\}$ (attributi con stessa chiusura)
 - $\{B\rightarrow A\}$
 - $\{AD\rightarrow E\}$
2. Creazione delle relazioni:
 - $R_1(C,B,D,F,E)$ con chiavi C e BE
 - $R_2(B,A)$ con chiave B
 - $R_3(A,D,E)$ con chiave AD
3. Nessuna relazione è contenuta in un'altra
4. R_1 contiene la chiave C , quindi non serve aggiungere ulteriori relazioni

4. Verifica della BCNF

Definizione:

Una relazione è in BCNF se per ogni dipendenza funzionale non banale $X\rightarrow A$, X è una superchiave.

Procedimento:

1. Per ogni relazione ottenuta dalla decomposizione in 3NF, verificare tutte le dipendenze funzionali applicabili
2. Per ogni dipendenza $X \rightarrow A$, verificare se X è una superchiave per quella relazione

Esempio:

- $R_1(C,B,D,F,E)$: $C \rightarrow B$, $C \rightarrow D$, $C \rightarrow F$ soddisfano BCNF poiché C è chiave; $BE \rightarrow C$ soddisfa BCNF poiché BE è chiave
- $R_2(B,A)$: $B \rightarrow A$ soddisfa BCNF poiché B è chiave
- $R_3(A,D,E)$: $AD \rightarrow E$ soddisfa BCNF poiché AD è chiave

Tutte le relazioni rispettano la BCNF, quindi la decomposizione è anche in BCNF.

5. Recovery Management (Esercizio su Log e Ripristino)

Comprensione del Log:

- $CK(T5,T6)$: Checkpoint con transazioni $T5$ e $T6$ attive
- $B(T7)$: Inizio transazione $T7$
- $U(T7,O6,B6,A6)$: $T7$ aggiorna $O6$ da $B6$ a $A6$
- $U(T6,O3,B7,A7)$: $T6$ aggiorna $O3$ da $B7$ a $A7$
- $B(T8)$: Inizio transazione $T8$
- $I(T8,O5,A5)$: $T8$ inserisce $A5$ in $O5$
- $C(T8)$: Commit di $T8$
- $A(T5)$: Abort di $T5$

Metodo di ripristino a caldo:

1. **Identificare transazioni da annullare (UNDO)** e transazioni da ripetere (REDO)
2. **Per le operazioni UNDO**: lavorare a ritroso, ripristinando i valori originali
3. **Per le operazioni REDO**: lavorare in avanti, riapplicando i cambiamenti

Analisi:

- UNDO per $T5$, $T6$ e $T7$ (non hanno fatto commit)
- REDO per $T8$ (ha fatto commit)

Prima operazione da eseguire (ordine di esecuzione):

La risposta corretta è **O3=B7** (opzione 2) perché:

1. Nella ripresa a caldo, si eseguono prima le operazioni UNDO
2. Si procede in ordine inverso (dall'ultima alla prima)
3. L'ultima operazione prima del guasto che coinvolge T5, T6 o T7 è $U(T6, O3, B7, A7)$
4. Quindi il primo UNDO è ripristinare O3 al suo valore originale B7

L'ordine corretto delle operazioni è:

1. $O3=B7$ (UNDO di $U(T6, O3, B7, A7)$)
2. $O6=B6$ (UNDO di $U(T7, O6, B6, A6)$)
3. $O5=A5$ (REDO di $I(T8, O5, A5)$)

Questo segue il protocollo ARIES (Algorithm for Recovery and Isolation Exploiting Semantics) dove si eseguono prima gli UNDO e poi i REDO per garantire la consistenza del database.