Basi di Dati
La teoria della normalizzazione
-- terza parte --

3NF

Outline

- **Introduzione**
- 3NF
- Insieme di copertura minimale
- Normalizzazione in 3NF
- Proprietà della normalizzazione in 3NF

Forma normale BCNF (richiamo)

Data una relazione *R(A)* in 1NF e l'insieme di dipendenze funzionali *F*, la relazione è in **BCNF** se per ogni $X \rightarrow Y \in F$ si verifica almeno una delle seguenti condizioni:

1. $Y \subseteq X$ ($X \rightarrow Y$ è una dipendenza **riflessiva**)
2. X è **superchiave** di R

Limitazioni di BCNF

Abbiamo visto che BCNF permette di definire schemi che eliminano le anomalie.

Però la BCNF ha una *limitazione*: per alcuni schemi non è possibile raggiungere la BCNF conservando le d.f.

Quindi è utile considerare un'altra forma normale, *la terza forma normale*, che:

- è meno restrittiva,
- non elimina tutte le anomalie,
- è sempre possibile raggiungere conservando le dipendenze funzionali.

Per definirla, prima introduciamo la definizione di *attributo primo*.

Outline

- Introduzione
- **3NF**
- Insieme di copertura minimale
- Normalizzazione in 3NF
- Proprietà della normalizzazione in 3NF

Attributi primi

Riconsideriamo questo esempio:

ESAMI(<u>MATR</u>, <u>Co</u>, Vo, CP)

F = {MATR, Co $\rightarrow$ Vo, CP; CP $\rightarrow$ Co}

con le chiavi {MATR,Co} e {MATR,CP}

Co è contenuto in una chiave della relazione ESAMI.

Si dice che Co è un **attributo primo** per la relazione ESAMI.

Attributi primi (definizione)

Data una relazione *R(A)*, gli attributi $Y \subseteq A$ sono detti **attributi primi** se e solo se $Y \subseteq K$, dove *K* è **una delle chiavi** di *R(A).*

N.B. Per determinare se un attributo è primo occorre quindi determinare tutte le chiavi della relazione.

Terza Forma Normale (3NF)

Una relazione *(R(A),F)* è in **3NF** (**terza forma normale**) se per ogni ogni $X \rightarrow Y \in F$ si verifica almeno una delle seguenti condizioni:

1. $Y \subseteq X$ ($X \rightarrow Y$ è **riflessiva**)
2. *X* è **superchiave**
3. Y sono **attributi primi**

Nota: 1. e 2. sono comuni con la definizione di BCNF.
3. è aggiuntiva di 3NF.

Terza Forma Normale (3NF)

Intuizione:

3. è un piccolo rilassamento della BCNF che permette la conservazione delle dipendenze.

3. permette ripetizioni su attributi che fanno parte di una chiave, che dovrebbero essere limitate.

Terza forma normale

ESAMI(<u>MATR</u>, <u>Co</u>, Vo, CP), con F = {MATR, Co → Vo, CP; CP → Co}
è un esempio di relazione che è in 3NF ma non in BCNF, infatti

- MATR,Co → Vo è una d.f. BCNF
- CP → Co è una d. f. 3NF non BCNF
  (non è BCNF perché CP non è superchiave, ma è 3NF perché Co è parte di una chiave)

La 3NF è *meno restrittiva della BCNF*, in quanto ammette come dipendenze funzionali, oltre a quelle ammesse da BCNF, anche quelle il cui conseguente è costituito da attributi primi.

Se una relazione *R(A)* è in BCNF, allora è anche in 3FN.
L’inverso non è necessariamente vero.

3NF e anomalie

- Una relazione in 3NF può avere anomalie di inserimento e di cancellazione.
- Ad esempio:

AGENZIE(<u>NAgenzia,CittaAgenzia</u>,Direttore)

$F_{AGENZIE}$ = { NAgenzia,CittaAgenzia → Direttore;
Direttore → CittaAgenzia }

AGENZIE non è in BCNF perché nella seconda d.f. Direttore non è superchiave, ma il conseguente, CittaAgenzia, è un attributo primo, quindi è in 3NF.

<u>NAgenzia</u>	**<u>CittaAgenzia</u>**	**Direttore**
341	Torino	Chiara
341	Roma	Susanna
343	Torino	Chiara
341	Milano	Mario
…	…	…

3NF e anomalie

<u>NAgenzia</u>	<u>CittaAgenzia</u>	Direttore
341	Torino	Chiara
341	Roma	Susanna
343	Torino	Chiara
341	Milano	Mario
…	…	…

$F_{AGENZIE}$ = { NAgenzia,CittaAgenzia → Direttore;
Direttore → CittaAgenzia }

In AGENZIE, a causa della d.f. Direttore → CittaAgenzia:

- CittaAgenzia è una *ridondanza* quando ho uno stesso direttore
- Se aggiungo un direttore devo aggiungere anche un'agenzia corrispondente perché CittaAgenzia è in chiave (*anomalia di inserimento*).
- Se cancello tutte le agenzie di un direttore, cancello anche ogni traccia del direttore (*anomalia di cancellazione*).
- Se cambio CittàAgenzia a un direttore, devo aggiornare tutte le agenzie che dirige (*anomalia di aggiornamento*), ma i cambiamenti delle chiavi primarie sono rari e limitati.

3NF e anomalie

- Dato che la BCNF non permette anomalie e la 3NF sì, queste derivano dalle d.f. di tipo 3.
- Storicamente è stata definita prima la 3NF e poi la BCNF proprio per evitare le anomalie permesse dalla 3NF.
- Comunque di solito sono anomalie limitate e tollerate.

Outline

- Introduzione
- 3NF
- **Insieme di copertura minimale**
- Normalizzazione in 3NF
- Proprietà della normalizzazione in 3NF

Normalizzazione in 3NF

La tecnica per normalizzare in 3NF parte da una manipolazione delle dipendenze funzionali.

Partendo da un insieme di dipendenze funzionali, dobbiamo trovare un altro insieme di d.f. equivalente e minimale.

Per introdurre la definizione di *insieme di copertura minimale*, dobbiamo definire le nozioni di:

- *Attributo estraneo*
- *Dipendenza ridondante*

Attributo estraneo

Un attributo in una d.f. in F è *estraneo* se e solo se possiamo rimuovere l'attributo dalla d.f. continuando ad avere un insieme di d.f. equivalente.

Es.: Abbiamo le seguenti dipendenze funzionali
F = $\{ABCD \rightarrow E, B \rightarrow C\}$

C è un attributo estraneo nella prima d.f. e si può cancellare: infatti C si può ricavare da B tramite le due d.f.

Quindi F diventa $\{ABD \rightarrow E, B \rightarrow C\}$.

Attributo estraneo

- Posso verificare se un attributo è estraneo calcolando la chiusura degli attributi.
- Dato un insieme di d.f. F e una d.f. $X \rightarrow Y \in F$ (per semplicità assumiamo che Y sia un attributo), l'attributo $B \in X$ è estraneo nella d.f. $X \rightarrow Y$ se e solo se $Y \in (X-B)^{+}_{F}$.
- Esempio: Dato F= {*ABCD* → *E, B* → *C*},
  - C è estraneo nella prima d.f. perché $E \in \{ABD\}^{+}_{F} = \{ABCDE\}$.
  - D non è estraneo nella prima d.f. perché $E \notin \{ABC\}^{+}_{F} = \{ABC\}$.

Dipendenze ridondanti

Una dipendenza funzionale è ridondante in un insieme di d.f. F se e solo se possiamo rimuoverla da F continuando ad avere un insieme di d.f. equivalente.

Dipendenze ridondanti

- Posso verificare se una dipendenza è ridondante usando la chiusura di un insieme di attributi.
- Dato un insieme di d.f. F, la d.f. $X \rightarrow Y \in F$ è ridondante (per semplicità assumiamo che Y sia un attributo) se e solo se $Y \in X^{+}_{F - \{X \rightarrow Y\}}$.
- Esempio. Consideriamo l'insieme di dipendenze $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$
  - $A \rightarrow C$ è ridondante perché $C \in \{A\}^{+}_{\{A \rightarrow B, B \rightarrow C\}} = \{ABC\}$.
  - $B \rightarrow C$ non è ridondante perché $C \notin \{B\}^{+}_{\{A \rightarrow B, A \rightarrow C\}} = \{B\}$.

Insieme di copertura minimale (definizione)

Un insieme F' di dipendenze funzionali è un **insieme di copertura minimale** rispetto a F quando:

i. $F' \equiv F$ (F' è equivalente a F)

ii. in ogni $X \rightarrow Y \in F'$ Y è un attributo singolo (si dice che è in *forma canonica*)

iii. ogni $X \rightarrow Y \in F'$ è priva di attributi estranei

iv. ogni $X \rightarrow Y \in F'$ non è ridondante

i. indica che F' è una ***copertura*** (per F).

ii., iii. e iv. significano che F' è ***minimale.***

In realtà ii. non è un requisito necessario, ma ci semplifica la trattazione.

Algoritmo per il calcolo di una copertura minimale

Ad alto livello:

1. Trasforma ogni d.f. in forma canonica
2. Elimina gli attributi estranei
3. Elimina le d.f. ridondanti

Algoritmo per il calcolo di una copertura minimale

Più nel dettaglio:

$F' := F$

1. **per ogni** d.f. $X \rightarrow A_1 \ldots A_n \in F'$

   sostituisci in F' la d.f. $X \rightarrow A_1 \ldots A_n$ con le d.f. $X \rightarrow A_1, \ldots, X \rightarrow A_n$

2. **per ogni** d.f. $X \rightarrow A_i \in F'$

   per ogni $B_j \in X$

   se $A_i \in (X - B_j)^+_{F'}$ **allora** cancella B_j da X e aggiorna F'

3. **per ogni** $X \rightarrow A_i \in F'$

   $F^* := F' - \{X \rightarrow A_i\}$

   se $A_i \in X^+_{F^*}$ **allora** $F' := F^*$

return F'

Algoritmo per il calcolo di una copertura minimale

Più nel dettaglio:

Uso F' come variabile temporanea

$F' := F$

1. **per ogni** d.f. $X \rightarrow A_1 \ldots A_n \in F'$

   sostituisci in F' la d.f. $X \rightarrow A_1 \ldots A_n$ con le d.f. $X \rightarrow A_1, \ldots, X \rightarrow A_n$

2. **per ogni** d.f. $X \rightarrow A_i \in F'$

   per ogni $B_j \in X$

   se $A_i \in (X - B_j)^+_{F'}$ **allora** cancella B_j da X e aggiorna F'

3. **per ogni** $X \rightarrow A_i \in F'$

   $F^* := F' - (X \rightarrow A_i)$

   se $A_i \in X^+_{F^*}$ **allora** $F' := F^*$

return F'

Algoritmo per il calcolo di una copertura

Trasformo ogni d.f. in forma canonica applicando la regola di decomposizione.
F' sarà equivalente a F.

Più nel dettaglio:

$F' := F$

1. **per ogni** d.f. $X \rightarrow A_1 \ldots A_n \in F'$

    sostituisci in F' la d.f. $X \rightarrow A_1 \ldots A_n$ con le d.f. $X \rightarrow A_1, \ldots, X \rightarrow A_n$

2. **per ogni** d.f. $X \rightarrow A_i \in F'$

    per ogni $B_j \in X$

    se $A_i \in (X - B_j)^+_{F'}$ **allora** cancella B_j da X e aggiorna F'

3. **per ogni** $X \rightarrow A_i \in F'$

    $F^* := F' - (X \rightarrow A_i)$

    se $A_i \in X^+_{F^*}$ **allora** $F' := F^*$

return F'

Esempio passo 1

Considero la d.f. MATR,Co $\rightarrow$ Vo,CP

Possiamo ricavare d.f. equivalenti con un attributo singolo applicando la *regola di decomposizione*:

MATR,Co $\rightarrow$ Vo

MATR,Co $\rightarrow$ CP

Algoritmo per il calcolo di una copertura minimale

Più nel dettaglio:

$F' := F$

1. **per ogni** d.f. $X \rightarrow A_1 \ldots A_n \in F'$

sostituisci in F' la d.f. $X \rightarrow A_1 \ldots A_n$ con le d.f. $X \rightarrow A_1, \ldots, X \rightarrow A_n$

2. **per ogni** d.f. $X \rightarrow A_i \in F'$

per ogni $B_j \in X$

se $A_i \in (X - B_j)^+_{F'}$ **allora** cancella B_j da X e aggiorna F'

3. **per ogni** $X \rightarrow A_i \in F'$

$F^* := F' - (X \rightarrow A_i)$

se $A_i \in X^+_{F^*}$ **allora** $F' := F^*$

return F'

Se, data la d.f. $B_1, \ldots, B_m \rightarrow A_i$, riesco a ottenere A_i anche senza B_j, B_j è un attributo estraneo e posso eliminarlo dalla d.f.
F' sarà equivalente a F.

Esempio (attributi estranei, passo 2)

Dato F' = {ABCD → E, B → C}

L'algoritmo si chiede se **A** è estraneo in ABCD→E:

- Calcola la chiusura di (ABCD – A) rispetto a F': calcola quindi la chiusura di BCD
  - Si può applicare solo B → C, ma C è già nella chiusura, quindi $BCD^+ = BCD$
- Verifica se il conseguente (E) è contenuto nella chiusura
  - Non lo è, quindi A non è estraneo
- Esamina il successivo…

Esempio (attributi estranei, passo 2)

Dato F' = {A**B**CD → E, B → C}

L'algoritmo si chiede se **B** è estraneo in ABCD→E:

- Calcola la chiusura di (ABCD – B) rispetto a F': calcola quindi la chiusura di ACD
  - Non ha dipendenze applicabili, quindi ACD^+ = ACD
- Verifica se il conseguente (E) è contenuto nella chiusura
  - Non lo è, quindi B non è estraneo
- Esamina il successivo…

Esempio (attributi estranei, passo 2)

Dato F' = {AB**C**D → E, B → C}

L'algoritmo si chiede se **C** è estraneo in ABCD→E:

- Calcola la chiusura di (ABCD – C) rispetto a F': calcola quindi la chiusura di ABD
  - Applicando B → C, si trova ABCD
  - Applicando anche ABCD → E, si trova ABD^+ = ABCDE
- Verifica se il conseguente (E) è contenuto nella chiusura
  - E $\in$ ABCDE, quindi C è **estraneo e può essere rimosso dalla d.f.**
- Quindi **F' :=** {ABD → E, B → C}
- Esamina il successivo (e ultimo)…

Esempio (attributi estranei, passo 2)

Dato F' = {ABD → E, B → C}

L'algoritmo si chiede se **D** è estraneo in ABD→E:

- Calcola la chiusura di ABD – D rispetto a F': calcola quindi la chiusura di AB
  - È applicabile solo B → C, quindi AB^+ = ABC
- Verifica se il conseguente (E) è contenuto nella chiusura
  - Non lo è, quindi D non è estraneo
- Abbiamo analizzato ogni attributo a sinistra di ABD → E.
- B → C ha un solo attributo a sinistra, quindi non è sicuramente estraneo.

Altro esempio (attributi estranei, passo 2)

F' = {AB → C, B → C}

L'algoritmo si chiede se **A** è estraneo in AB→C:

- Calcola la chiusura di AB – A rispetto a F': calcola quindi la chiusura di B
  - È applicabile solo B → C, quindi B^+ = BC
- Verifica se il conseguente (C) è contenuto nella chiusura
  - C $\in$ BC, quindi A è **estraneo e si aggiorna F'**
  - Quindi **F' :=** {B → C, B → C} = {B → C}
- B → C ha un solo attributo a sinistra, quindi non è sicuramente estraneo.

Algoritmo per il calcolo di una copertura minimale

Più nel dettaglio:

$F' := F$

1. **per ogni** d.f. $X \rightarrow A_1 \ldots A_n \in F'$

sostituisci in F' la d.f. $X \rightarrow$

2. **per ogni** d.f. $X \rightarrow A_i \in F'$

per ogni $B_j \in X$

se $A_i \in (X - B_j)^+_{F'}$ **allora** cancella B_j da X e aggiorna F'

3. **per ogni** $X \rightarrow A_i \in F'$

$F^* := F' - (X \rightarrow A_i)$

se $A_i \in X^+_{F^*}$ **allora** $F' := F^*$

return F'

Se, data la d.f. $X \rightarrow A_i$, riesco, tramite le altre d.f., a ottenere A_i anche senza usare $X \rightarrow A_i$, $X \rightarrow A_i$ è una d.f. ridondante e posso eliminarla. F' sarà equivalente a F.

Esempio (dipendenze ridondanti)

F' = {A → B, B → C, A → C}

L'algoritmo si chiede se **A → B** è ridondante:

- Rimuovendo A → B dall'insieme, si ottiene F*= {B → C, A → C}
- Calcola la chiusura di A rispetto a F*:
  - Applicando A→ C, si ottiene A^+ = AC
- Verifica se il conseguente (B) è contenuto in A^+
  - Non lo è, quindi A→ B non è ridondante

La verifica con **B → C** ha lo stesso risultato.

Esempio (dipendenze ridondanti)

F' = {A → B, B → C, **A → C**}

L'algoritmo si chiede se **A → C** è ridondante:

- Rimuovendo A → C dall'insieme, ottiene
  F*= {A → B, B → C}
- Calcola la chiusura di A rispetto a F*:
  - Applicando A → B, si ottiene AB
  - Applicando B → C, si ottiene A^+ = ABC
- Verifica se il conseguente (C) è contenuto in A^+
  - A $\in$ ABC, quindi A→ C **è ridondante**
  - Aggiorna F' := F* = {A → B, B → C}

Attenzione

Attenzione.

Bisogna sempre ***prima*** eliminare tutti gli attributi estranei **e** ***poi*** eliminare le dipendenze funzionali ridondanti

Altrimenti si rischia di non riconoscere tutte le dipendenze funzionali ridondanti

Controesempio

Consideriamo F'={AB → C, C → A, B → A}

Proviamo a invertire i due passi

Cerchiamo prima le dipendenze funzionali ridondanti:

AB→ C è ridondante?

Controesempio

Consideriamo F'={AB → C, C → A, B → A}

AB→ C è ridondante?

La chiusura di AB rispetto a F* = {C → A, B → A} è AB

C non è contenuto in AB quindi AB→ C non è ridondante

Controesempio

Consideriamo F'={AB → C, C → A, B → A}

C → A è ridondante?

La chiusura di C rispetto a F* = {AB → C, B → A} è C

A non è contenuto in C quindi C → A non è ridondante

Controesempio

Consideriamo F'={AB → C, C → A, B → A}

B → A è ridondante?

La chiusura di B rispetto a F* = {AB → C, C → A} è B

A non è contenuto in B quindi B → A non è ridondante

Controesempio

Consideriamo F'={AB → C, C → A, B → A}

Ora cerchiamo gli attributi estranei

Ovviamente ci concentriamo sulle d.f. con antecedenti composti da almeno due attributi

Consideriamo solo AB → C

Controesempio

Consideriamo F'={AB $\rightarrow$ C, C $\rightarrow$ A, B $\rightarrow$ A}

Consideriamo solo AB $\rightarrow$ C.

Ci chiediamo se A è estraneo

- Calcoliamo la chiusura di B rispetto a F'
  - con B $\rightarrow$ A abbiamo AB
  - con AB $\rightarrow$ C abbiamo B^+ = ABC
- Il conseguente C è contenuto nella chiusura, quindi A è un attributo estraneo e

F' = {B $\rightarrow$ C, C $\rightarrow$ A, B $\rightarrow$ A}

Controesempio

Il risultato è quindi F' = {B → C, C → A, B → A}

Ma F' non è minimale perché B → A è ridondante per transitività.

Quindi i due passi **non sono invertibili**!

Non unicità di F’

L’algoritmo non determina necessariamente un’unica copertura minimale F’.

Considerare gli attributi e le d.f. in ordine diverso può portare a eliminare attributi estranei o d.f. ridondanti diversi.

Le diverse coperture minimali trovate sono comunque equivalenti.

Non unicità di F’

Vediamo un *esempio* di non unicità relativa alle dipendenze ridondanti.

Consideriamo una relazione R(CF,MATR,SCU), cioè codice fiscale (CF), matricola (MATR) e credenziali SCU (SCU)

Abbiamo le dipendenze funzionali seguenti:

CF → MATR

CF → SCU

MATR → CF

MATR → SCU

SCU → CF

SCU → MATR

Non unicità di F'

È chiaro che c'è ridondanza.

Per esempio possiamo cancellare MATR → SCU e SCU → MATR perché possiamo ottenerle per transitività. Si verifica che le dipendenze funzionali rimaste non sono ridondanti (**esercizio**)

CF → MATR

CF → SCU

MATR → CF

~~MATR → SCU~~

SCU → CF

~~SCU → MATR~~

Non unicità di F’

In alternativa potrei però cancellare CF → MATR e MATR → CF perché posso ottenerle per transitività.

Si verifica che le dipendenze funzionali rimaste non sono ridondanti (**esercizio**) e abbiamo un insieme F’ diverso

~~CF → MATR~~

CF → SCU

~~MATR → CF~~

MATR → SCU

SCU → CF

SCU → MATR

Non unicità di F’

Possiamo generare ancora un altro insieme F’ di copertura minimale con il ciclo

CF → SCU

SCU → MATR

MATR → CF

È una copertura minimale più semplice ma è equivalente agli altri casi (**esercizio**)

Altro esempio

Consideriamo la relazione ESAMI con le dipendenze funzionali

F = {MATR,CF,Co → Vo, MATR → CF, CF → MATR}

Mostreremo la non unicità di F' in relazione agli attributi estranei (prima invece l'abbiamo mostrata in relazione alle dipendenze ridondanti)

Altro esempio

Consideriamo la relazione ESAMI con le dipendenze funzionali
$F = \{MATR,CF,Co \rightarrow Vo, MATR \rightarrow CF, CF \rightarrow MATR\}$

Verifichiamo solo $MATR,CF,Co \rightarrow Vo$

Si vede immediatamente che MATR è estraneo se calcoliamo la chiusura di CF,Co utilizzando $CF \rightarrow MATR$

Quindi abbiamo $F' = \{CF,Co \rightarrow Vo, MATR \rightarrow CF, CF \rightarrow MATR\}$

Altro esempio

Consideriamo la relazione ESAMI con le dipendenze funzionali
F = {MATR,CF,Co → Vo, MATR → CF, CF → MATR}

Verifichiamo solo MATR,CF,Co → Vo

Ma anche CF è estraneo se calcoliamo la chiusura di MATR,Co utilizzando MATR → CF

Quindi abbiamo F' = {MATR,Co→Vo, MATR→CF, CF→MATR}

Proprietà (complessità)

La complessità dell’algoritmo per il calcolo dell’insieme di copertura minimale è **polinomiale**, infatti contiene cicli su F e cicli su X con all’interno il calcolo della chiusura di attributi, che – come abbiamo già visto – è **polinomiale**.

Outline

- Introduzione
- 3NF
- Insieme di copertura minimale
- **Normalizzazione in 3NF**
- Proprietà della normalizzazione in 3NF

Normalizzazione in 3NF

Ad alto livello:

Data una relazione *(R(A),F)*,

1. calcola la copertura minimale *F'* di *F*
2. decomponi R: costruisci una relazione per ogni d.f. in F' mettendo insieme le d.f. con lo stesso antecedente
3. elimina le relazioni i cui schemi sono contenuti negli schemi di altre relazioni (conservando le relative d.f.)
4. se una relazione non la contiene già, aggiungi una relazione con la chiave di R

Normalizzazione in 3NF

Più in dettaglio:

Data una relazione *(R(A),F)*,

1. **calcola** la copertura minimale *F'* di *F*
2. **per ogni** insieme di d.f. $\{X \rightarrow A_1, \ldots, X \rightarrow A_n\} \subseteq F'$ che contiene tutte le d.f. che hanno a sinistra gli stessi attributi X

   crea la relazione $(R_X(\underline{X}A_1 \ldots A_n), \{X \rightarrow A_1 \ldots A_n\})$
3. **per ogni** coppia di relazioni $R_X(\underline{X}Y)$, $R_{X'}(\underline{X'}Y')$ in cui $XY \supseteq X'Y'$

   elimina la relazione $R_{X'}$ e aggiungi le d.f. di $R_{X'}$ a quelle di R_X
4. **se** nessuna relazione contiene una chiave *K* qualsiasi di *R(A)*

   trova K tale che $K^+ = A$ e crea una nuova relazione $R_K(\underline{K})$

Esempio

ESAMI(MATR,NS,DN,Co,Vo,DE,CP,NP)

F={MATR,Co$\rightarrow$ Vo,DE,CP; MATR$\rightarrow$NS,DN; CP$\rightarrow$NP}

Esempio

ESAMI(MATR,NS,DN,Co,Vo,DE,CP,NP)

F={MATR,Co→ Vo,DE,CP; MATR→NS,DN; CP→NP}

Prima di tutto ricaviamo tutte le chiavi: ci serve per verificare se ESAMI è 3NF e per eseguire il passo 4.

Si può verificare che l'unica chiave è Matr,Co.

ESAMI non è né 3NF né BCNF: per es. CP→NP non è riflessiva, di tipo superchiave o attributi primi. Decomponiamo in 3NF…

Esempio

ESAMI(MATR,NS,DN,Co,Vo,DE,CP,NP)

F={MATR,Co→ Vo,DE,CP; MATR→NS,DN; CP→NP}

1. *calcola la copertura minimale F' di F:*

- Scrivo le d.f. in F in forma canonica:
  F = {MATR → NS; MATR→ DN; MATR,Co→ Vo;
  MATR,Co → DE; MATR,Co→CP; CP → NP}
- Non ci sono attributi estranei né d.f. ridondanti, quindi F è già minimale

Esempio

F' = {MATR → NS; MATR→ DN; MATR,Co→ Vo; MATR,Co → DE; MATR,Co→CP; CP → NP}

2. ***per ogni*** *insieme di d.f.* $\{X \rightarrow A_1, \ldots, X \rightarrow A_n\} \subseteq F'$ *che contiene tutte le d.f. che hanno a sinistra gli stessi attributi X*

   crea *la relazione* $(R_X(\underline{X}A_1 \ldots A_n), \{X \rightarrow A_1 \ldots A_n\})$:

- Da {MATR → NS; MATR→ DN} ottengo R_1(<u>MATR,</u>NS,DN)
- Da {MATR,Co→ Vo; MATR,Co → DE; MATR,Co→CP} ottengo R_2(<u>MATR,Co</u>,Vo,DE,CP)
- Da {CP → NP} ottengo R_3(<u>CP</u>,NP)

Tutte le d.f. iniziali sono rappresentate dalle chiavi primarie di R_1, R_2 e R_3

Esempio

F' = {MATR → NS; MATR→ DN; MATR,Co→ Vo; MATR,Co → DE; MATR,Co→CP; CP → NP}

3. ***per ogni*** *coppia di relazioni* $R_X(\underline{X}Y)$, $R_{X'}(\underline{X'}Y')$ *in cui* $XY \supseteq X'Y'$
elimina *la relazione* $R_{X'}$ *e aggiungi le d.f. di* $R_{X'}$ *a quelle di* R_X

- Nessuna relazione è un sottoinsieme di un'altra.

Esempio

F' = {MATR → NS; MATR→ DN; MATR,Co→ Vo; MATR,Co → DE; MATR,Co→CP; CP → NP}

4. ***se*** *nessuna relazione contiene una chiave K qualsiasi di R(A)*

trova *K tale che* $K^+=A$ *e crea una nuova relazione* $R_K(\underline{K})$

Verifico se in almeno una delle relazioni è presente una chiave per ESAMI

- R_1(<u>MATR</u>,NS,DN)
- **R_2(<u>MATR,Co</u>,Vo,DE,CP)**
- R_3(<u>CP</u>,NP)

R_2 contiene MATR,Co, che è una chiave di ESAMI, quindi non devo aggiungere un'ulteriore relazione.

Esempio

Ho terminato la normalizzazione in 3NF di

ESAMI(MATR,NS,DN,Co,Vo,DE,CP,NP)

F={MATR,Co→ Vo,DE,CP; MATR→NS,DN; CP→NP}

in

R_1(<u>MATR</u>,NS,DN)

R_2(<u>MATR,Co</u>,Vo,DE,CP)

R_3(<u>CP</u>,NP)

Non è necessario che rappresenti le d.f. di R_1, R_2 e R_3 perché sono tutte rappresentate dalle chiavi primarie.

Posso constatare che quindi tutte le d.f. sono di tipo 2 e quindi il nuovo schema è anche in BCNF.

Altro esempio

CC(Titolare,NConto,NAgenzia,CittaAgenzia,Saldo)

Con F = {NConto $\rightarrow$ Nagenzia; NConto $\rightarrow$ CittaAgenzia; NConto $\rightarrow$ Saldo}

Altro esempio

CC(Titolare,NConto,NAgenzia,CittaAgenzia,Saldo)

Con F = {NConto → Nagenzia; NConto → CittaAgenzia; NConto → Saldo}

Ricaviamo tutte le chiavi. Si può verificare che l'unica chiave è Titolare,NConto.

CC non è né 3NF né BCNF: per es. NConto → Nagenzia non è riflessiva, di tipo superchiave o attributi primi. Decomponiamo in 3NF…

Altro esempio

CC(Titolare,NConto,NAgenzia,CittaAgenzia,Saldo)

Con F = {NConto → Nagenzia; NConto → CittaAgenzia; NConto → Saldo}

1) Trovo la copertura minimale: F lo è già.

2) Decompongo:

- R_1(<u>NConto</u>,NAgenzia,CittaAgenzia,Saldo)

3) Dato che c'è una sola relazione, non cerco sottoinsiemi

4) Considerando F, la chiave di CC è (Titolare, NConto). In R_1 non c'è Titolare quindi sicuramente manca la chiave di CC, quindi aggiungo la relazione:

- R_2(<u>Titolare,NConto</u>)

Ulteriore esempio

CC(Titolare, NConto, NAgenzia, CittaAgenzia, Saldo, Direttore, Qualifica, Stipendio)

F' = {

1. NConto → NAgenzia
2. NConto → CittaAgenzia
3. NConto → Saldo
4. NAgenzia, CittaAgenzia → Direttore
5. Qualifica → Stipendio
6. Direttore → CittaAgenzia

}

Ulteriore esempio

CC(Titolare, NConto, NAgenzia, CittaAgenzia, Saldo, Direttore, Qualifica, Stipendio)

F' = {NConto → NAgenzia; NConto → CittaAgenzia; NConto → Saldo; NAgenzia, CittaAgenzia → Direttore; Qualifica → Stipendio; Direttore → CittaAgenzia}

Cerchiamo tutte le chiavi: si può verificare che Titolare,NConto,Qualifica è l’unica chiave.

La relazione non è BCNF né 3NF: per es. NConto → Nagenzia non è riflessiva, di tipo superchiave o attributi primi.

Decomponiamo in 3NF…

Esempio

1) Calcoliamo una copertura minimale (F’ è già minimale)

2) Costruiamo una relazione per ogni dipendenza funzionale raggruppando quelle con lo stesso antecedente

- R1(<u>NConto</u>,NAgenzia,CittaAgenzia,Saldo)
- R2(<u>NAgenzia,CittaAgenzia</u>,Direttore)
- R3(<u>Qualifica</u>,Stipendio)
- R4(<u>Direttore</u>,CittaAgenzia)

Esempio

3) Cerchiamo se una relazione è un sottoinsieme di un'altra:

R1(NConto,NAgenzia,CittaAgenzia,Saldo)

R2(NAgenzia,CittaAgenzia,Direttore)

R3(Qualifica,Stipendio)

~~**R4(Direttore,CittaAgenzia)**~~

- R4 è sottoinsieme di R2, quindi eliminiamo R4 e aggiungiamo la d.f. Direttore→CittaAgenzia

Esempio

4) Verifichiamo se qualche relazione contiene una chiave per CC(Titolare, NConto, NAgenzia, CittaAgenzia, Saldo, Direttore, Qualifica, Stipendio):

R1(NConto,NAgenzia,CittaAgenzia,Saldo)

R2(NAgenzia,CittaAgenzia,Direttore) con Direttore→CittaAgenzia

R3(Qualifica,Stipendio)

(Titolare,NConto,Qualifica) è chiave di CC. Dato che nessuna relazione ha Titolare, sicuramente non è presente una chiave per CC, quindi creiamo una relazione:

– R5(Titolare,NConto,Qualifica)

Outline

- Introduzione
- 3NF
- Insieme di copertura minimale
- Normalizzazione in 3NF
- **Proprietà della normalizzazione in 3NF**

Proprietà della normalizzazione 3NF

La normalizzazione 3NF:

- Nelle relazioni $R_i(XA_1 \ldots A_n)$ che genera dalle d.f. $X \rightarrow A_i$, X è chiave di R_i.
- Genera relazioni in 3NF.
- Ha complessità polinomiale.
- Conserva le dipendenze, infatti troviamo ogni dipendenza di F' all'interno della relazione corrispondente.
- Garantisce la decomposizione con join senza perdita.

Proprietà della normalizzazione 3NF

- Nelle dimostrazioni delle proprietà per semplicità ignoriamo alcune ottimizzazioni introdotte dall'algoritmo.
- Supponiamo che lo schema generato sia di questa forma $\{R_1(X_1A_1), R_2(X_2A_2), \ldots, R_n(X_nA_n), [R_{n+1}(K)]\}$.
- Non consideriamo cioè che, nei casi in cui alcuni X_i siano uguali tra di loro, abbiamo formato un'unica relazione.

Lemma: X è chiave di R_X

Consideriamo una qualsiasi relazione $R_i(X_i A_i)$:
la relazione R_i è stata generata da una d.f. $X_i \rightarrow A_i$ presente nella copertura minimale F'.
Allora X_i è **chiave di R_i**.

Dimostrazione

Consideriamo una qualsiasi relazione $R_i(X_iA_i)$: la relazione è stata generata da $X_i \rightarrow A_i$, quindi X_i è superchiave di R_i, ma si dimostra che X_i è **chiave di R_i**, non solo superchiave, di R_i.

Se vale $X_i \rightarrow A_i$ allora $X_iA_i \subseteq X_i^+$, cioè X_i è superchiave di R_i.

Assumiamo per assurdo che X_i non sia una superchiave minimale e quindi esista un sottoinsieme $W \subset X_i$ che sia chiave per R_i, cioè $X_iA_i \subseteq W^+_{F'}$.

Allora gli attributi in $X_i - W$ sono attributi estranei per la d.f. $X_i \rightarrow A_i$, infatti, se $X_iA_i \subseteq W^+_{F'}$ otteniamo che $A_i \subseteq W^+_{F'}$ ed è sufficiente W per ottenere A_i (cioè $W \rightarrow A_i \in F'^+$).

Questo è in contraddizione con l'ipotesi che $X_i \rightarrow A_i$ appartenga alla copertura minimale perché gli attributi estranei sarebbero stati rimossi.

Di conseguenza X_i è **chiave**.

Proprietà dell'algoritmo: genera solo schemi 3NF

Si può dimostrare che l'algoritmo di normalizzazione 3NF genera relazioni in 3NF.

Dimostrazione

Per assurdo assumiamo che, in una relazione *$R_i(\underline{X_i}A_i)$*, sia presente una d.f. proveniente dalla copertura minimale in forma canonica che non sia 3NF $Y \rightarrow B_j$ con $YB_j \subseteq X_iA_i$.

Allora $B_j = A_i$, altrimenti $B_j \in X_i$ e B_j sarebbe un attributo primo e $Y \rightarrow B_j$ sarebbe 3NF. Quindi la d.f. che vìola la 3NF è $Y \rightarrow A_i$.

$Y \neq X_i$ altrimenti la d.f. sarebbe ridondante (la chiave primaria implica già una d.f. $X_i \rightarrow A_i$).

Quindi $Y \subset X_i$ per costruzione, ma, dato che $Y \rightarrow A_i$, Y sarebbe una chiave per R_i e X_i non sarebbe una chiave (ma solo una superchiave). Assurdo per il lemma dimostrato prima.

[CVD]

Schemi BCNF

La normalizzazione 3NF in alcuni casi genera schemi BCNF.

Infatti:

– il passo 2 genera schemi BCNF $R(\underline{X_i}A_i)$ con la d.f. $X_i \rightarrow A_i$, che è BCNF perché l'antecedente è una chiave,

– il passo 4 genera schemi $R(\underline{K})$ con d.f. banali.

L'unico passo che può generare schemi non BCNF è il passo 3.

Proprietà dell’algoritmo: complessità

Ogni passo dell’algoritmo è polinomiale:

- il calcolo della copertura minimale è polinomiale (passo 1).
- la decomposizione è polinomiale e itera su tutte le d.f., che sono di lunghezza polinomiale (passo 2).
- il controllo del contenimento è polinomiale (passo 3).
- la ricerca della chiave itera su ogni attributo calcolando la chiusura ed è polinomiale (passo 4).

Proprietà dell'algoritmo: conservazione delle dipendenze

Ogni dipendenza funzionale $X \rightarrow A_i$ della copertura minimale F' si ritrova nella restrizione della corrispondente relazione $R_i(XA_i)$ per costruzione (passo 2) oppure in una relazione $R_{i'}(X'A_{i'})$ che la contiene (passo 3).

In tutti e due casi la decomposizione **conserva le dipendenze**.

Proprietà dell'algoritmo: decomposizione con join senza perdita

La normalizzazione in 3NF garantisce la decomposizione con join senza perdita.

Non vediamo la dimostrazione.

Conclusione

Alcuni degli schemi prodotti dalla normalizzazione in 3NF sono anche in BCNF: sono i casi in cui non sono state generate dipendenze di tipo 3.

Conviene quindi normalizzare sempre in 3NF per via dei vantaggi rispetto alla normalizzazione in BCNF:

- conservazione delle dipendenze (BCNF non la garantisce)
- complessità polinomiale (BCNF è esponenziale)

Conclusione

Alla fine, si può analizzare quanto ottenuto: alcune relazioni saranno in BCNF e le anomalie saranno annullate.

Nelle relazioni con una dipendenza funzionale di tipo 3, invece, rimangono invece le anomalie.

Ad esempio in AGENZIE ci possono essere anomalie introdotte dalla d.f. Direttore $\rightarrow$ CittaAgenzia.