



Le dipendenze funzionali nel modello relazionale

Logica e Informatica





Indice

1

Introduzione alle
dipendenze funzionali

2

Un sistema formale
per le FD

3

Un algoritmo per
l'implicazione nelle FD

4

Le FD per i vincoli
nelle Basi di Dati

5

Le forme normali nelle
Basi di Dati

6

Conclusioni



Introduzione alle dipendenze funzionali



Relazione di esempio

Problemi riscontrati:

- Ridondanza
- Inconsistenza

Esiste una dipendenza funzionale
 $\text{DIPARTIMENTO} \rightarrow \text{SUPERVISORE}$,
ovvero il valore di DIPARTIMENTO
determina il valore di SUPERVISORE.

Studente	Dipartimento	Supervisore
Clelia	DIAG	Cristian
Amin	DIAG	Cristian
Andrea	DIET	Leonardo
Pietro	DIET	Leonardo
Elisabetta	DIAG	Cristian
Federico	DIET	Leonardo

Formalmente:

$X \rightarrow Y$ è una **dipendenza funzionale** (FD) su uno schema R
quando per ogni tupla $u, v \in I$ relazione su R , si ha $u[X] = v[X] \Rightarrow u[Y] = v[Y]$.



Proprietà delle FD

Data I una qualsiasi relazione legale senza valori nulli su uno schema R :

1

Soddisfacibilità

Dato un insieme Σ di FD, $I \models \Sigma$
se $\forall \sigma \in \Sigma$ si ha $I \models \sigma$.

3

Ridondanza

Σ è ridondante se esiste una FD $\sigma \in \Sigma$ tale
che $\Sigma - \{\sigma\} \models \sigma$.

2

Equivalenza

$\Delta \equiv \Sigma$
se e solo se $\Delta \models \sigma, \forall \sigma \in \Sigma$
e $\Sigma \models \delta, \forall \delta \in \Delta$.

4

Indipendenza

Un insieme di FD Σ è indipendente se **non è ridondante**,
cioè $\neg \exists \sigma \in \Sigma$ tale che $\Sigma - \{\sigma\} \models \sigma$.



Proiezione su relazioni

Sia I una qualsiasi relazione su R , una FD $X \rightarrow Y$ dove $X \cup Y \subset R$ è soddisfatta da I se e solo se $\pi_{XY}(I)$ soddisfa $X \rightarrow Y$.

Dipartimento	Supervisore
DIAG	Cristian
DIET	Leonardo

Studente	Dipartimento	Supervisore
Clelia	DIAG	Cristian
Amin	DIAG	Cristian
Andrea	DIET	Leonardo
Pietro	DIET	Leonardo
Elisabetta	DIAG	Cristian
Federico	DIET	Leonardo



La riduzione delle FD

Una FD $X \rightarrow Y$ è **reduced** se $\neg \exists W \subset X$ tale che $\Sigma \models W \rightarrow Y$.
Inoltre diremo che un insieme Σ di FD è **reduced** se tutte le FD al suo interno lo sono.

Algorithm 1: REDUCED

Data: $\Sigma \leftarrow$ un insieme di FD in input.

$\Delta \leftarrow \Sigma$

for *each* FD $X \rightarrow Y \in \Delta$ **do**

for *each* attribute $A \in X$ **do**

if $\Delta \models X - A \rightarrow Y$ **then**

$X \leftarrow X - A$ dove $X \rightarrow Y$

end

end

return Δ

Un sistema formale per le FD



Assiomi e regole di Inferenza per le FD

Elaborato da W.W Armstrong nel 1974.

FD1

Riflessività

$\vdash X \rightarrow X$

FD2

Transitività

$\{X \rightarrow Y, Y \rightarrow Z\} \vdash X \rightarrow Z$

FD3

Non ha un nome 😞

$X \rightarrow Y \vdash X \cup Z \rightarrow Y \cup Z$

Il sistema formale di Armstrong è **sound** e **complete**.



Dimostriamo al volo la soundness...

Che ricordiamo essere $\Sigma \vdash \sigma$ allora $\Sigma \models \sigma$,
la dimostrazione della completeness è più articolata...

Partiamo considerando le tuple $u, v \in I$ relazione su uno schema R contenente attributi X, Y, Z

FD1

Riflessività $\vdash X \rightarrow X$:
è banale...

FD2

Transitività $\{X \rightarrow Y, Y \rightarrow Z\} \vdash X \rightarrow Z$:
se $u[X] = v[X]$ allora $u[Y] = v[Y]$ ma allora anche $u[Z] = v[Z]$ e quindi è sound.

FD3

$X \rightarrow Y \vdash X \cup Z \rightarrow Y \cup Z$:
Supponiamo $X \rightarrow Y$ sia soddisfatta, e quindi $u[X \cup Z] = v[X \cup Z]$ ovviamente coincidono $u[X] = v[X]$ e $u[Z] = v[Z]$, data la FD abbiamo $u[Y] = v[Y]$ e di conseguenza $u[Z \cup Y] = v[Z \cup Y]$.



Nel caso dei valori nulli...

In questa relazione si hanno le FD

$A \rightarrow B$ e $B \rightarrow C$, e di conseguenza FD2 deduce $A \rightarrow C$

Se considerassimo i valori nulli come mancanza di informazione, la regola di inferenza FD2 non è più applicabile.

Ignoreremo pertanto la presenza di valori nulli nelle relazioni, poiché richiedono una trattazione più accurata ed estensiva.

A	B	C
a ₁	b ₁	c ₁
a ₂	b ₂	c ₂



Sostituiamo la colonna B con valori nulli.

A	B	C
a ₁	NULL	c ₁
a ₂	NULL	c ₂



Un algoritmo per l'implicazione delle FD





La chiusura di un insieme di FD

Per risolvere il problema dell'implicazione sfruttiamo il concetto di chiusura di un insieme, in particolare dato un insieme Σ di FD:

$$\Sigma \models X \rightarrow Y \iff Y \subseteq \text{cl}_\Sigma(X)$$

Pertanto per vedere se una FD $X \rightarrow Y$ è implicata da Σ ci basta costruire $\text{cl}_\Sigma(X)$.



Un algoritmo per la chiusura

Questo algoritmo computa la chiusura $cl_\Sigma(X)$ di un insieme Σ di FD, in tempo lineare rispetto alla lunghezza dell'input.
(Teor. C.Beer e P.A. Bernstein)

$\Sigma \models X \rightarrow A, \forall A \in X.$

Consideriamo $S \rightarrow T$ e $S \subseteq Y$, allora $Y \rightarrow S$ e da $S \rightarrow T$ otteniamo $Y \rightarrow T$.

Da cui $X \rightarrow Y$ e $X \rightarrow T$ si ottiene $X \rightarrow Y \cup T$.
E quindi $\Sigma \models X \rightarrow Y \cup T$.

Algorithm 2: CLOSURE

Data: $\Sigma \leftarrow$ un insieme di FD in input.

Data: $X \leftarrow$ un insieme di attributi in input

$Y \leftarrow X$

while *esiste una FD $S \rightarrow T$ tale che $S \subseteq Y$ e $T \not\subseteq Y$* **do**
 $Y \leftarrow Y \cup T$

end

return Y

Inoltre si dimostra di conseguenza che questo algoritmo non può avere come risultato, un $Y \subset cl_\Sigma(X)$.



Altre definizioni utili sulle FD

1

Closed

Una FD $X \rightarrow Y$ è **closed** se $Y = \text{cl}\Sigma(X)$
e un insieme Δ di FD è **closed** se tutte le FD
lo sono.

2

Forma Canonica

Una FD $X \rightarrow Y$ è in forma canonica
se $|Y| = 1$.

3

Cover tra insiemi di FD

Dati due insiemi Σ e Δ di FD, una è
cover per l'altra se $\Delta \equiv \Sigma$.
Inoltre sarà una cover minima se contiene il
numero minimo di FD.

4

Cover Canonica

Una cover canonica Δ di Σ è una cover **non
ridondante** in cui ogni FD è sia **reduced** che
in forma canonica

Teorema:

Sia Δ un **cover non ridondante** per Σ
allora se Δ è **closed** è anche **minimo**.





La conversione in forme canoniche

Teorema:

Ogni FD $X \rightarrow Y$ può essere convertita in **forma canonica**.

Dimostrazione. Sia $Y = A_1, \dots, A_k$, allora certamente $X \rightarrow Y \models X \rightarrow A_j$,
e l'insieme $\{\forall j : X \rightarrow A_j\} \models X \rightarrow Y$.



La non ridondanza per le FD

Questo algoritmo, dato un insieme Σ di FD, ne trova uno Δ **non ridondante**, che risulta essere una **cover** per Σ .

Non trova necessariamente **cover minimi**.

Algorithm 3: NONREDUND

Data: $\Sigma \leftarrow$ un insieme di FD in input.

$\Delta \leftarrow \Sigma$

for *each* FD $\sigma \in \Delta$ **do**

if $\Delta - \{\sigma\} \models \sigma$ **then**
 $\Delta \leftarrow \Delta - \{\sigma\}$

end

return Δ

Esempio:

Input: $\{X \rightarrow Y, Y \rightarrow Z, X \rightarrow Z\}$

Un possibile output: $\{X \rightarrow Y, Y \rightarrow Z\}$



I cover minimi per gli insiemi di FD

A differenza dello scorso algoritmo a ogni rimozione verifichiamo che la **dipendenza rimossa** Y si trovi nella **chiusura**, e in caso negativo la calcoliamo e l'aggiungiamo come nuova FD a Δ .

Algorithm 4: MINCOVER

Data: $\Sigma \leftarrow$ un insieme di FD in input.

$\Delta \leftarrow \Sigma$

for *each* FD $\sigma = X \longrightarrow Y \in \Sigma$ **do**

$\Delta \leftarrow \Delta - \{\sigma\}$

if $Y \not\subseteq \text{CLOSURE}(\Delta, X)$ **then**

$Z \leftarrow \text{CLOSURE}(\Delta, Y)$

$\Delta \leftarrow \Delta \cup \{X \longrightarrow Z\}$

end

return Δ

Il costo dell'algoritmo è $O(|\Sigma| \times \tau)$ dove τ è il costo dell'esecuzione dell'algoritmo 2 (Closure).



Il problema degli insiemi di FD minimi

Sia lo schema di relazione $R = \{A_1, \dots, A_k\} \subset U$ su cui definiamo tutte le relazioni, prese ora tutte le possibili relazioni legali che soddisfano un insieme Σ di FD, calcoliamo l'insieme Δ il quale è **cover closed non ridondante**, cioè **minimo**.

Poniamo $K = \{X_k\}$ tale che $\forall X \in K, \Delta \vdash X \rightarrow R$ e non esiste altra Y che invece $\Delta \vdash Y \rightarrow R$, cioè abbiamo trovato un **insieme di FD minimo su tutto lo schema R** a partire da K . Inoltre **non esiste altro Y con questa stessa proprietà in R** .

Trovare questo K risulta essere il più piccolo ed elementare insieme di FD per R nonché unico è un problema **NP-Completo**.



Le FD per i vincoli nelle Basi di Dati



Definiamo gli schemi

Schema di relazione:

(R, Σ) è formato da un insieme di relazioni $R \subseteq U$,
dove $U = \bigcup R_k$, e da un insieme Σ di FD.

Schema di Basi di Dati:

è un insieme di schemi di relazioni
 $\{(R_1, \Sigma_1), \dots, (R_k, \Sigma_k)\}$, dove $\Sigma = \bigcup \Sigma_j$

Ricordiamo che in logica del prim'ordine abbiamo definito per uno **Schema di Base di Dati D**:
un analogo **Linguaggio L_D** (in cui i predicati corrispondono alle tabelle/relazioni, ecc)
+ i **vincoli di integrità** cioè query booleane (senza variabili libere in uscita) in calcolo relazionale.

Vedremo come in particolare possiamo definire le chiavi e superchiavi usando le FD.





Una Base di Dati è...

Con le FD

Un'assegnazione ragionevole B sullo schema D di ogni relazione.

In FOL

Un'interpretazione legale (cioè non violi i vincoli)
 $J_D = \langle \Delta, I \rangle$ di L_D

Queste definizioni vivono in simbiosi, e possiamo immaginare le FD come uno strumento aggiunto alla FOL



Definizioni sugli schemi di relazione

Dato uno schema di relazione (R, Σ) , e un insieme di attributi $X \subseteq R$

1

Determinante

X è un determinante per R se esiste almeno un attributo $A \in R - X$ tale che $\Sigma \models X \rightarrow A$.

2

Superchiave

Se $\forall A \in R$ abbiamo $\Sigma \models X \rightarrow A$ allora X è una superchiave per R .

3

Chiave

Sia X una superchiave, se $\forall B \in X, \Sigma \models X - B \not\rightarrow A$ allora X è una chiave per R .



Un algoritmo per la ricerca di una chiave

Questo algoritmo, a partire da tutti gli attributi di una relazione, anche grazie alle FD, ne cerca il più piccolo insieme che sia una chiave.

Algorithm 5: MINIMAL - KEY

Data: $R \leftarrow$ un insieme di relazioni in input.

Data: $\Sigma \leftarrow$ un insieme di FD in input.

$\Delta \leftarrow \Sigma$

$X \leftarrow R$

for *each* $A \in R$ **do**

if $\Delta \models X - A \rightarrow R$ **then**
 $X \leftarrow X - A$

end

return X

Con FD:

$\{B \rightarrow C\}$

Deduce A come **chiave**.

A	B	C
a ₁	b ₁	c ₁
a ₂	b ₁	c ₁
a ₃	b ₃	c ₃





Approccio brute-force per la ricerca di tutte le chiavi.

Può essere interessante sapere tutte le chiavi presenti in uno schema di relazioni, questo algoritmo le calcola brutalmente.

Algorithm 6: ALL - MINIMAL - KEYS

Data: $R \leftarrow$ un insieme di relazioni in input.

Data: $\Sigma \leftarrow$ un insieme di FD in input.

$Q \leftarrow \mathcal{P}^R$

$K \leftarrow \emptyset$

while $Q \neq \emptyset$ do

$X \in Q$

 if $\Sigma \models X \longrightarrow R$ then

$Z \leftarrow \text{NONREDUND}(X)$

$K \leftarrow K \cup \{Z\}$

 Siano $S_1, \dots, S_r \in Q$ tali che $X \subset S_i, \forall i = 1, \dots, r$

$Q \leftarrow Q - X - S_1 - \dots - S_r$

 end

 else

$Q \leftarrow Q - X$

 end

end

return K



Le forme normali nelle Basi di Dati





Cosa sono le forme normali?

Sono una serie di regole che uno schema di Base di Dati deve rispettare.

Ce ne sono diverse, e noi ne vedremo soltanto alcune (le prime due), un elenco non completo delle più importanti include:

- BCNF (Boyce-Codd)
- 3-NF
- 2-NF
- 1-NF

Questo elenco è decrescente in base alla stringenza delle condizioni, e ognuna verifica anche quella successiva.





La forma normale di Boyce-Codd

Uno schema di basi di dati **D** soddisfa la **BCNF** quando:

Se X è un **determinante** di R allora X è una **superchiave** per R , dove R fa parte di un qualunque schema di relazione in D .

Ciò implica che per $A \in R - X$ dove $X \rightarrow A$ è una FD, si ha **necessariamente** X chiave.



Una tecnica per la verifica della BCNF

Proiezione su insiemi di FD

Dato un insieme di attributi $X \subset R$ nel contesto (R, Σ) definiamo la proiezione di Σ su X come
 $\pi_X(\Sigma) = \{W \rightarrow A \in \Sigma \text{ e inoltre } W \cup A \subseteq X\}$

Cioè prendiamo le sole FD che riguardano l'insieme di attributi X .

Verifica della BCNF

Supponiamo che da D vogliamo ottenere E in BCNF.
pertanto, sia $\Sigma_j = \pi_{R_j}(\Sigma)$.

Se $\forall X \rightarrow A \in \Sigma_j$ si ottiene $\Sigma_j \models X \rightarrow R_j$,
allora (R_j, Σ_j) è in **BCNF**.

Cioè Σ_j è un insieme di FD che rende X una superchiave per R_j

Teorema di C. Beeri e P.A. Bernstein:
Verificare che ci sia una **violazione** alla **BCNF** è un problema **NP-Completo**.





Trasformazioni su uno schema

Trasformazione Lossless

Sia I una relazione su U che soddisfa Σ , allora la decomposizione di I in $\pi_{R_j}(I)$, $j = 1, \dots, k$ è lossless.

Trasformazione Dependency-Preserving

Sia $\Delta = \bigcup \Sigma_j$, allora $\Sigma \models \Delta$ e $\Delta \models \Sigma$.

Uno schema di base di dati E a partire da D rappresenta D se essa subisce una trasformazione sia **Lossless** che **Dependency-Preserving**.



Proiezioni Lossless come conseguenza delle FD

Dato $R = ABCDE$ uno schema di relazione tale che per ogni relazione si abbia la FD $E \rightarrow CD$ valida. Considereremo le decomposizioni di R su ABE e CDE .

Si nota che $\pi_{ABE}(I) * \pi_{CDE}(I) = I$ è vero $\forall I$ grazie alla presenza della FD che garantisce decomposizioni lossless.

A	B	C	D	E
0	0	0	0	1
0	1	0	0	1
1	0	1	1	0
1	1	1	1	0

Teorema:

Sia I definita su $R = XYZ$ tale che I soddisfi la FD $X \rightarrow Y$ allora $\pi_{XY}(I) * \pi_{XZ}(I) = I$, cioè le proiezioni sono lossless.



Decomposizione per ottenere la BCNF

Assumiamo $D = \{(R_1, \Sigma_1), \dots, (R_k, \Sigma_k)\}$ non in **BCNF** e che (R_j, Σ_j) sia la **causa della violazione** e che quindi esista un X che sia **determinante** per R_j ma che **non sia superchiave**,

cioè esiste un attributo $A \in R_j - X$ tale che $\Sigma \models X \rightarrow A$ dove $\Sigma = \bigcup \Sigma_j$,

usando il teorema precedente sostituiamo (R_j, Σ_j) con $D1 = (\pi_{XA}(R_j), \Sigma_{j_1})$ e $D2 = (\pi_{R_j-A}(R_j), \Sigma_{j_2})$ e continuiamo questo processo finché $D2$ non è in **BCNF**.

Tuttavia questo processo **non è Dependency-Preserving**.





Cosa possiamo fare?

A partire da uno schema di base di dati **D** ottenere **un altro schema in BCNF** che lo rappresenti potrebbe **non essere fattibile**.

Possiamo cercare di rilassare queste condizioni, tentando di fermarci a una forma più debole, cioè la **Terza Forma Normale (3NF)**.



Definizioni aggiuntive per la 3NF

1

Primalità

Un attributo $A \in R_j$ è detto **primo** se esiste una chiave Z di R_j tale che $A \in Z$.
Determinare la primalità di un attributo è un **problema NP-Completo**.

2

Determinante Forte

Z sarà un determinante forte di R_j se $Z \subseteq R_j$ ed esiste un A non primo tale che $A \in R_j - Z$ soddisfi $\Sigma \models Z \rightarrow A$.
Non è necessario che Z sia una superchiave.



Propagazione della primalità nelle catene di FD

Teorema:

Sia (R, Σ) uno schema di relazione, $\forall j: 1 \leq j \leq n$ sia $A_j \rightarrow A_{j+1} \in \Sigma$, sia dato K superchiave, se $A_{n+1} \in K$ allora $A_1 \in K$.

Dimostrazione:

Data una FD $A \rightarrow B$, si ha che A è primo se B lo è. Per questo senza perdita di generalità supponiamo $B \twoheadrightarrow A$ la primalità di B significa che $K - B \twoheadrightarrow R$ per una superchiave K tale che $B \in K$, se $A \in K$ il risultato segue. se $A \notin K$ allora si avrebbe che $\{K \rightarrow A, B \twoheadrightarrow A\} \models K - B \rightarrow A$, cioè una contraddizione.





La Terza Forma Normale

Uno schema di base di dati D soddisfa la Terza Forma Normale (3NF) quando se Z è un **determinante forte** per R in D , allora Z è una **superchiave** per R .

Come si verifica la 3NF?

- Mostrare che per tutte le FD $X \rightarrow A \in \Delta_j$, X è una **superchiave**, o A è **primo**, dove $\Delta_j = \text{CANONICAL}(\Sigma_j)$, e quindi la **3NF** è rispettata.
- Trovare una FD $X \rightarrow A \in \Delta_j$, tale che X non è una **superchiave** e A non è **primo**, e quindi la **3NF** è violata.

Relazione tra 3NF e BCNF:

Sia (R, Σ) uno schema di relazione in **3NF**, se per ogni coppia di chiavi K_1, K_2 si ha $K_1 \cap K_2 = \emptyset$ allora (R, Σ) è in **BCNF**.



Algoritmo per la 3NF

L'algoritmo inizialmente calcola una cover canonica per Σ , poi ne cerca una chiave. Successivamente genera uno schema di relazione per ogni FD, e infine aggiunge la chiave a $\mathcal{D}$.

Algorithm 7: 3NF

Data: $U \leftarrow$ un insieme di relazioni in input.

Data: $\Sigma \leftarrow$ un insieme di FD in input.

$\mathcal{D} \leftarrow \phi$

$\Delta \leftarrow \text{CANONICAL}(U, \Sigma)$

$X \leftarrow \text{MINIMAL-KEY}(U, \Delta)$

for *each* FD $Y \leftarrow A \in \Delta$ **do**

$\mathcal{D} \leftarrow \mathcal{D} \cup (YA, \pi_{YA}(\Delta))$

end

$\mathcal{D} \leftarrow \mathcal{D} \cup (X, \phi)$

return $\mathcal{D}$

Conclusioni





Le Forme normali dalla progettazione nel modello ER

Come si presenta tutto questo nelle fasi di progetto di una Base di Dati?

Un'ottima **progettazione concettuale** così come una eccellente **progettazione logica** a partire dai requisiti di progetto, portano alla realizzazione di Schemi di Basi di Dati già in **forme normali**.

Si ricorda infatti che nella **ristrutturazione** della progettazione logica, dove nel caso di **tabelle lascamente accoppiate**, era previsto un **accorpamento per denormalizzazione** per facilitare l'accesso ai dati e ridurre il costo computazionale.





Le FD nel ciclo di vita della Base di Dati

In via definitiva le FD sono un **mattone fondamentale** della **progettazione** di basi di dati nel modello relazionale, dove appaiono silenziosamente, e sono una **conseguenza di una ben svolta progettazione**.

Pertanto averne un'accurata conoscenza aiuta in tutte le fasi del ciclo di vita della base di dati.





Grazie per l'attenzione!

