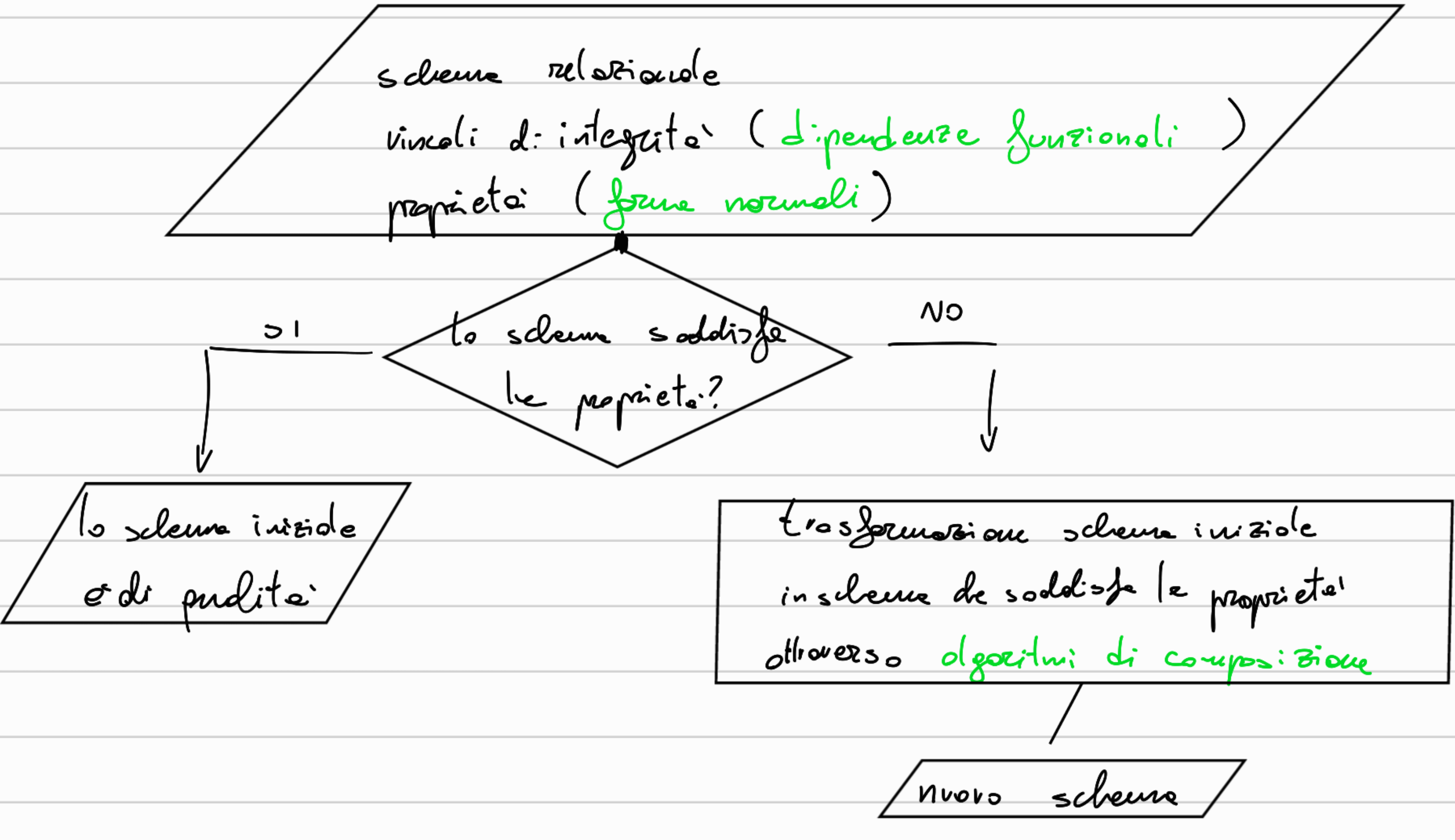
Teoria della normalizzazione / 26-04

Si ricerca uno schema di DB di "qualità"

La qualità si può definire sulle proprietà di un DB che vengono o meno soddisfatte tramite strategie

Utilizzando una metodologia di progettazione posso eseguire una verifica finale sul DB (può essere necessario se non creo io il DB)



Una forma normale è una proprietà (condizione) che se verificata garantisce l'assenza di determinati difetti

Un DB non è normalizzato se:

- presenta ridondanze
- presenta comportamenti atipici (anomalie)

L'assenza di questi sopra mi definisce un DB di qualità

Per ridondanza si intende la replicazione di stesse informazioni (dati nella relazione)

Le anomalie possono essere di vario genere:

- di aggiornamento; se eseguo una modifica e ho ridondanza devo aggiornare tutte le tuple

- di inserimento: se inserisco una tupla che ad esempio mette null una chiave

- di cancellazione: se cancello alcune tuple potrei perdere info utili es. cancellando gli studenti laureati (se questi erano legati a bdd e cancello tutti loro, perdo info su chi fa bdd)

Tali anomalie possono ad esempio capitare se unisco Studenti, Corso, Esami in un'unica tabella.

Dipendenze funzionali e vincoli di integrità

$R(X)$ schema di attributi $X = \{a_1, \dots, a_n\}$

$Y \subseteq X \quad Y \neq \emptyset \quad , \quad Z \subseteq X \quad Z \neq \emptyset$

Data r istanza di $R(X)$

Esiste in r dipendenza funzionale di $Y \rightarrow Z$ se per ogni coppia di tuple $t_1, t_2 \in r$ vale che se $t_1.Y = t_2.Z$ allora $t_1.Z = t_2.Z$

|| definita in termini di r

Ad esempio dip. funz. Matricola $\rightarrow$ (Nome, Cognome, Anno)

$Y \rightarrow Z$

se due tuple coincidono sulla matricola coincideranno anche su nome...
e vale per una qualsiasi istanza dello schema.

Y determina funzionalmente Z in $R(X)$ $[Y \xrightarrow[R(X)]{} Z]$

se qualunque istanza x di $R(X)$ soddisfa $Y \rightarrow Z$

$Y \xrightarrow[R(X)]{} Z$ diventa vincolo di integrità, cioè vale $\forall$ istanza $Y \rightarrow Z$

Esempi

questo implica

Matricola $\xrightarrow[R]{}$ Nome Studente (Matricola, Cod Corso → Nome)

entrambe chiavi

Matricola $\xrightarrow[R]{}$ Cognome S

Matricola $\xrightarrow[R]{}$ Anno Iscrizione

(equivale a

Matricola $\xrightarrow[R]{}$ {Nome, Cognome, Anno Iscrizione})

Matricola, Codice Corso $\xrightarrow[R]{}$ Voto

Nome Studente $\xrightarrow[\cancel{R}]{}$ Cognome Studente ?

↳ può determinare solo in alcune istanze
(es. per stessi nomi ho coincidenza di stessi cognomi)

Casi particolari

da cui intendo vincolo di integrità (R)

1) Matricola, Cod Corso → Cod Corso

Vale sempre ma non è banale se $A \notin Y$

vedi slide

$Y \to Z$ non è banale se per nessun attributo $A \in Z$ vale $A \notin Y$

non è chiave da solo

2) Matricola → Anno
Cod Corso → Crediti } genera ridondanza

Matricola, Cod Corso → Voto   non genera ridondanza perché ho una chiave

↳ è chiave

Dalle dipendenze funzionali alle "buone proprietà"
→ forme normali

Tipi di forme normali

- forma normale di Boyce e Codd (BCNF)

R(X) è in BCNF se per qualunque (tutte) dipendenza funzionale $X \to Y$ non banale che vale in R X contiene una chiave

- Garantiscono omogeneità e identificano univocamente le entità
- Almeno UNA chiave, qualsiasi superchiave è ok (contiene chiave) ↓ non necessariamente primaria (anche alternativa)

a) Il primo passo è determinare le chiavi
b) Verifico le condizioni di BCNF

Esiste una relazione fra il concetto di chiave e quello di dipendenza funzionale? Cioè se non conosco la chiave o ne voglio cercare di alternative posso usare le dipendenze funzionali (cioè ∃ df che la faccia)?

So che Matricola, Cod Corso → Voto
ma vale " " → Nome
anche " " → Cognome

E' possibile stabilire una relazione e ci forniscono un modo per ricercare le chiavi della relazione.

/03-05

Det. chiave/i relazioni dalle dipendenze funzionali

chiusura $\overline{X}_F^+ = \{$ tutti gli attributi che dipendono funzionalmente da $\overline{X}$, considerando $F\}$

Un insieme di attributi K è superchiave (unicità) se K_F^+ contiene tutti gli attributi (chiave se è anche minimale)

es. Calcola chiusura $\{Mat\}^+$; è chiave?

$Mat(0) = Mat$ $\quad$ $Mat(1) = Mat, Nome, Cognome, Anno$
$\quad Mat(2) = Mat(1)$

→ non contiene tutti gli attributi, no chiave

es. Calc. chiusura $\{Mat, Cod, Corso\}$ ... contiene tutti → superk.

chiave? → ∃ sottoinsieme che contiene tutti? Se sì ed è minimale è chiave

Verifica BCNF? ∀R individuo FD a partire dai vincoli

→ det chiave (semplifica con euristiche) →
→ BCNF (parte sx ⊆ chiave (∀))

Scomposizione

Decompongo in relazioni equivalenti con schemi omogenei, guidati dalle FD.

1) Unifico le dipendenze con la stessa parte sx
2) ∀ dipend. "unificate" → creo tabella separata
↓
ma non funziona sempre: se nessuna relazione decomposta ⊆ la chiave iniziale, potrei avere perdita
→ per lossless join creo una tabella con la chiave (nell'es in slide Mat, Corso → Voto contiene già la chiave).

Una decomposizione conserva le dipendenze se ciascuna delle FD coinvolge attributi che compaiono assieme dopo la decomposizione.

Proprietà:
- lossless join
- conservazione delle dipendenze

N.B. BCNF non sempre possibile

es. City, Street, CAP (Zip) ho 2 chiavi: CS e ZS

con chiusura = R per entrambe (con dipendenze che coinvolgono tutti gli A ↛ BCNF)

Se lo schema ammette > 1 chiave è possibile ∄ BCNF (troppo rigida)

3NF

E' sempre possibile in 3NF trovare decomposizioni che soddisfino le 2 proprietà

Se $\forall$ FD $X \to Y$ vale se almeno una delle seguenti condizioni è vera:

1) X contiene una chiave di R (BCNF)

2) ogni attributo di Y è contenuto in una chiave di R (attributi primi)

|| parte dx

prudosi, non necessariamente la stessa

l'appartenenza a una chiave limita le ridondanze

La 3NF funziona sempre se le dipendenze sono minimali

es. chi è chiave?

$R(A,B,C,D) \quad F: \{AB \to C, \quad BC \to D\}$

$A^+ = A$ non è chiave, stesso motivo per B, C, D

$AB \subseteq \{A\}$? no
$BC \subseteq \{A\}$? no

$AB^+ = A\,B$ + dipendenze $\qquad$ AB è chiave

$= A\,\underbrace{B\,C}$ posso ancora espandere

$= A\,B\,C\,D$

si vede bene che è chiave perchè A non è a destra e nemmeno B, non posso ottenerlo dalle dipendenze ma allora AB chiave

C e D compaiono solo a destra, quindi posso ottenerlo dalle dipendenze, non ∈ alla chiave

es quale decomposizione con perdite sicuramente?

↓

ovvero x qualunque istanza delle relazioni decomposte, se faccio il join ottengo un'istanza della R originale (è una proprietà sullo schema)

- $R_1(A,B)$ $\quad$ $R_2(B,C,D)$
- $R_1(A,B)$ $\quad$ $R_2(C,D)$
- $R_1(A,B)$ $\quad$ $R_2(A,B,C)$ $\quad$ $R_3(B,C,D)$
- $R_1(A)$ $\quad$ $R_2(B)$ $\quad$ $R_3(C)$ $\quad$ $R_4(D)$

- se vado a fare il join eseguo per forza un prod cartesiano, sono schemi disgiunti PERDO i legami!

- per stessi motivi ho schemi disgiunti = prod cart

→ negli altri schemi ho dei legami e non ho altre informazioni es Circa il primo esempio quale scomposizione ottengo?

Sappiamo che la chiave è AB
l'insieme di dipendenze precedenti è minimale
F

Creo una relazione per ogni dipendenza
R_1 : $\underline{AB}C$ almeno uno dei due schemi contiene la chiave? Sì
R_2 : $\overline{BC}D$ allora sono giunto alla decomposizione

oltrimenti avrei dovuto
aggiungere la chiave
(poi sarebbe ridondante)

-> è in BCNF? Devo determinare le dipendenze di prima presenti nelle nuove relazioni
$F_1 = \{ AB \to C \}$
$F_2 = \{ BC \to D \}$

Quindi chiave di R_1 è AB (non sto a destra, ho una sola dipendenza)
" R_2 è BC

*
Tutte le dip. di R_1 contengono la chiave? sì, ce n'è una
stessa cosa per R_2

Quindi la scomposizione è in BCNF

Se F è minimale -> la scomposizione ottenuta
↓
- è in 3NF
è in BCNF? Non lo - lossless join
so, devo verificarlo - preserva le dipendenze
come sopra *