

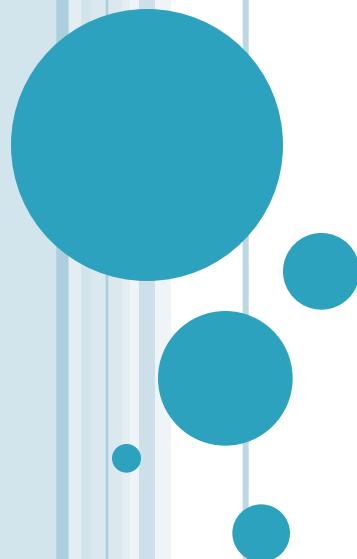
LA QUALITÀ DI UNA BASE DI DATI

FINORA

- **Ci siamo concentrati sul «come»**
- Come progettare una base di dati (livello concettuale, logico, DDL in SQL)
- Come interrogare una base di dati (algebra, interrogazioni in SQL)
- Come modificare una base di dati (modifiche in SQL)
- Come garantire l'integrità dei dati (vincoli in SQL)
- Come modificare una base di dati con «garanzie» sul risultato (transazioni in SQL, controllo della concorrenza e ripristino)
- Come definire il livello esterno (viste in SQL)
- Come controllare l'accesso a una base di dati (controllo dell'accesso in SQL)
- Come sviluppare applicazioni arbitrariamente complesse per basi di dati e regole di business (trigger)

ADESSO

- Ci concentriamo su «**come farlo bene**», introducendo il concetto di qualità
- **Qualità dei dati:** assenza di comportamenti indesiderati durante l'esecuzione degli aggiornamenti
 - Definizione del concetto di qualità dei dati
 - *Ingredienti già nel frigo:* schema logico, vincoli di integrità, modifiche
 - *Ingredienti da comprare:* teoria della normalizzazione (cenni)
- **Qualità del servizio:** come **migliorare le prestazioni del sistema** nell'eseguire le interrogazioni (livello logico/esterno, livello fisico, interrogazioni SQL) e nell'eseguire le transazioni (controllo della concorrenza e gestione del ripristino – **non lo vediamo**)
 - Definizione del concetto di qualità del servizio
 - *Ingredienti già nel frigo:* schema logico, schema esterno, interrogazioni e modifiche SQL
 - *Ingredienti da comprare:* schema fisico, indici, elaborazione delle interrogazioni



LA QUALITÀ DEI DATI: VERIFICA DI QUALITÀ DI SCHEMI RELAZIONALI E CENNI ALLA TEORIA DELLA NORMALIZZAZIONE

**Slide adattate a partire da quelle associate al
testo Atzeni, Ceri, Paraboschi, Torlone: Basi di
Dati, McGraw Hill**

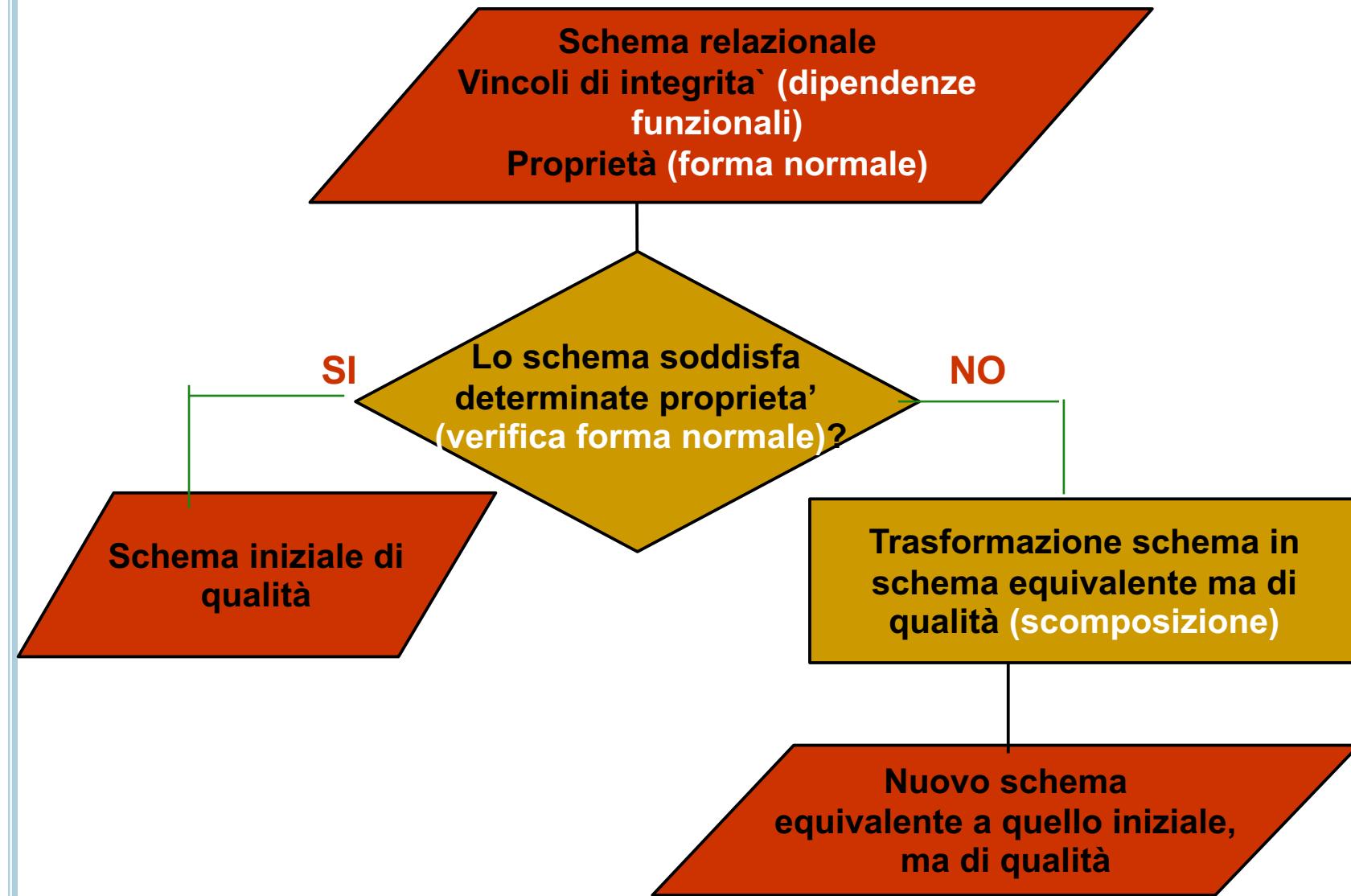
PROBLEMA

- Finora abbiamo imparato a progettare una base di dati partendo dall'analisi dei requisiti
- Ma come facciamo a garantire che lo schema ottenuto sia «di qualità»?
- E soprattutto, cosa vuol dire « qualità»?

SOLUZIONE

- La qualità di una base di dati si può definire in termini di proprietà che gli schemi (= le singole relazioni) possono soddisfare o meno
- Impatto su
 - Come lavorare su queste relazioni
 - Cosa queste relazioni rappresentano
- Strategie per verificare se le proprietà sono soddisfatte e, se non lo sono, per trasformare gli schemi
- Due applicazioni
 - Come ultima fase di progettazione (ma se seguiamo la metodologia, potremmo non averne bisogno)
 - Come tecnica di verifica dei risultati della progettazione di una base di dati (magari ottenuti non seguendo la metodologia che abbiamo introdotto in precedenza)

SOLUZIONE: NORMALIZZAZIONE



FORME NORMALI

- Una forma normale è una proprietà di una base di dati relazionale che ne garantisce la “qualità”, cioè l'assenza di determinati difetti
- Quando una relazione non è normalizzata:
 - presenta ridondanze
 - si presta a comportamenti poco desiderabili durante gli aggiornamenti → anomalie
- Quindi:
 - Qualità dei dati = assenza di (particolari tipi di) ridondanza = assenza di anomalie

UNA RELAZIONE CON ANOMALIE

Matricola Studente	NomeStudente	CognomeStudente	AnnoScr	CodiceCorso	SiglaCorso	NomeCorso	Crediti	Voto
4433038	Mario	Rossi	2	25880	BD	Basi Dati	9	30
4532453	Andrea	Verdi	3	25880	BD	Basi Dati	9	27
4532453	Andrea	Verdi	3	27054	BD2	Basi Dati 2	6	30
4446667	Bianca	Neri	2	80299	IP	Intro ...	12	24
4446667	Bianca	Neri	2	25880	BD	Basi Dati	9	27
4446667	Bianca	Neri	2	73026	EML	Elementi ...	12	21
4512345	Celeste	Mori	3	27054	BD2	Basi Dati 2	6	28
4512345	Celeste	Mori	3	25880	BD	Basi Dati	9	30
4455667	Viola	Bianchi	1	80299	IP	Intro ...	12	25
4455667	Viola	Bianchi	1	73026	EML	Elementi ...	12	26

UNA RELAZIONE CON ANOMALIE

Matricola Studente	NomeStudente	CognomeStudente	AnnoScr	CodiceCorso	SiglaCorso	NomeCorso	Crediti	Voto
4433038	Mario	Rossi	2	25880	BD	Basi Dati	9	30
4532453	Andrea	Verdi	3	25880	BD	Basi Dati	9	27
4532453	Andrea	Verdi	3	27054	BD2	Basi Dati 2	6	30
4446667	Bianca	Neri	2	80299	IP	Intro ...	12	24
4446667	Bianca	Neri	2	25880	BD	Basi Dati	9	27
4446667	Bianca	Neri	2	73026	EML	Elementi ...	12	21
4512345	Celeste	Mori	3	27054	BD2	Basi Dati 2	6	28
4512345	Celeste	Mori	3	25880	BD	Basi Dati	9	30
4455667	Viola	Bianchi	1	80299	IP	Intro ...	12	25
4455667	Viola	Bianchi	1	73026	EML	Elementi ...	12	26

Le informazioni di ciascuno studente (esempio nome, cognome, anno di corso) sono ripetute in tutte le tuple relative allo studente
→ ridondanza

UNA RELAZIONE CON ANOMALIE

Matricola Studente	NomeStudente	CognomeStudente	Annolscr	CodiceCorso	SiglaCorso	NomeCorso	Crediti	Voto
4433038	Mario	Rossi	2	25880	BD	Basi Dati	9	30
4532453	Andrea	Verdi	3	25880	BD	Basi Dati	9	27
4532453	Andrea	Verdi	3	27054	BD2	Basi Dati 2	6	30
4446667	Bianca	Neri	2	80299	IP	Intro ...	12	24
4446667	Bianca	Neri	2	25880	BD	Basi Dati	9	27
4446667	Bianca	Neri	2	73026	EML	Elementi ...	12	21
4512345	Celeste	Mori	3	27054	BD2	Basi Dati 2	6	28
4512345	Celeste	Mori	3	25880	BD	Basi Dati	9	30
4455667	Viola	Bianchi	1	80299	IP	Intro ...	12	25
4455667	Viola	Bianchi	1	73026	EML	Elementi ...	12	26

Se l'anno di iscrizione di uno studente varia, è necessario andarne a modificare il valore in diverse tuple (se non lo faccio, dati inconsistenti)

→ anomalia di aggiornamento

UNA RELAZIONE CON ANOMALIE

Matricola Studente	NomeStudente	CognomeStudente	AnnoScr	CodiceCorso	SiglaCorso	NomeCorso	Crediti	Voto
4433038	Mario	Rossi	2	25880	BD	Basi Dati	9	30
4532453	Andrea	Verdi	3	25880	BD	Basi Dati	9	27
4532453	Andrea	Verdi	3	27054	BD2	Basi Dati 2	6	30
4446667	Bianca	Neri	2	80299	IP	Intro ...	12	24
4446667	Bianca	Neri	2	25880	BD	Basi Dati	9	27
4446667	Bianca	Neri	2	73026	EML	Elementi ...	12	21
4512345	Celeste	Mori	3	27054	BD2	Basi Dati 2	6	28
4512345	Celeste	Mori	3	25880	BD	Basi Dati	9	30
4455667	Viola	Bianchi	1	80299	IP	Intro ...	12	25
4455667	Viola	Bianchi	1	73026	EML	Elementi ...	12	26
4476345	Chiara	Gialli	1	?				

Un nuovo studente che non abbia ancora dato esami non può essere inserito

→ anomalia di inserimento

UNA RELAZIONE CON ANOMALIE

Matricola Studente	NomeStudente	CognomeStudente	AnnoScr	CodiceCorso	SiglaCorso	NomeCorso	Crediti	Voto
4433038	Mario	Rossi	2	25880	BD	Basi Dati	9	30
4532453	Andrea	Verdi	3	25880	BD	Basi Dati	9	27
4532453	Andrea	Verdi	3	27054	BD2	Basi Dati 2	6	30
4446667	Bianca	Neri	2	80299	IP	Intro ...	12	24
4446667	Bianca	Neri	2	25880	BD	Basi Dati	9	27
4446667	Bianca	Neri	2	73026	EML	Elementi ...	12	21
4512345	Celeste	Mori	3	27054	BD2	Basi Dati 2	6	28
4512345	Celeste	Mori	3	25880	BD	Basi Dati	9	30
4455667	Viola	Bianchi	1	80299	IP	Intro ...	12	25
4455667	Viola	Bianchi	1	73026	EML	Elementi ...	12	26

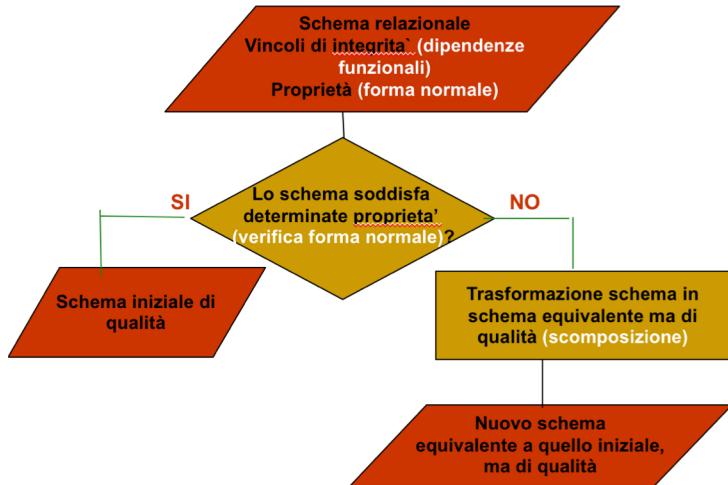
Se cancelliamo tutti gli esami registrati per un corso (ad es. perché gli studenti si laureano), perdiamo anche le informazioni sul corso

→ anomalia di cancellazione

PERCHÉ QUESTI FENOMENI INDESIDERABILI?

- Abbiamo usato un'unica relazione per rappresentare informazioni eterogenee
 - gli studenti con i relativi anni di corso
 - i corsi con i relativi crediti
 - gli esami per i corsi sostenuti dagli studenti con le relative votazioni

Matricola Studente	NomeStudente	CognomeStudente	Annoscr	CodiceCorso	SiglaCorso	NomeCorso	Crediti	Voto
4433038	Mario	Rossi	2	25880	BD	Basi Dati	9	30
4532453	Andrea	Verdi	3	25880	BD	Basi Dati	9	27
4532453	Andrea	Verdi	3	27054	BD2	Basi Dati 2	6	30
4446667	Bianca	Neri	2	80299	IP	Intro ...	12	24
4446667	Bianca	Neri	2	25880	BD	Basi Dati	9	27
4446667	Bianca	Neri	2	73026	EML	Elementi ...	12	21
4512345	Celeste	Mori	3	27054	BD2	Basi Dati 2	6	28
4512345	Celeste	Mori	3	25880	BD	Basi Dati	9	30
4455667	Viola	Bianchi	1	80299	IP	Intro ...	12	25
4455667	Viola	Bianchi	1	73026	EML	Elementi ...	12	26



PER STUDIARE IN MANIERA SISTEMATICA QUESTI ASPETTI, È NECESSARIO INTRODURRE UN VINCOLO DI INTEGRITÀ:
LA DIPENDENZA FUNZIONALE

PROPRIETÀ

- Ogni studente ha un unico anno di corso (anche se sostiene più esami)
- Ogni corso ha un certo numero di crediti
- Ogni studente nell'esame di ogni corso prende un unico voto (anche se può prendere voti diversi negli esami di corsi diversi)

DIPENDENZE FUNZIONALI

- $R(X)$ schema di relazione, $X = \{A_1, A_2, \dots, A_n\}$
 - Y e Z sottoinsiemi non vuoti di X
 - r istanza di $R(A_1, A_2, \dots, A_n)$
-
- esiste in r una dipendenza funzionale (FD) da Y a Z se, per ogni coppia di tuple t_1 e t_2 in r con gli stessi valori su Y , risulta che t_1 e t_2 hanno gli stessi valori anche su Z

DIPENDENZE FUNZIONALI

- Y determina funzionalmente Z in $R(X)$, indicato con $Y \rightarrow_{R(X)} Z$ (o con $Y \rightarrow Z$, in assenza di ambiguità) se qualunque istanza r di $R(X)$ soddisfa $Y \rightarrow Z$
- $Y \rightarrow Z$ è un vincolo di integrità (vale per qualunque istanza di R) ed è chiamato **dipendenza funzionale** per R

ESEMPI

MatricolaStudente → NomeStudente
MatricolaStudente → CognomeStudente
MatricolaStudente → Anno
CodiceCorso → SiglaCorso
CodiceCorso → NomeCorso
CodiceCorso → Crediti
MatricolaStudente CodiceCorso → Voto

NomeStudente → CognomeStudente ?

non è una FD: vale solo su questa istanza, altre istanze potrebbero violare il vincolo

Matricola Studente	NomeStudente	CognomeStudente	Annolscr	<u>CodiceCorso</u>	SiglaCorso	NomeCorso	Crediti	Voto
4433038	Mario	Rossi	2	25880	BD	Basi Dati	9	30
4532453	Andrea	Verdi	3	25880	BD	Basi Dati	9	27
4532453	Andrea	Verdi	3	27054	BD2	Basi Dati 2	6	30
4446667	Bianca	Neri	2	80299	IP	Intro ...	12	24
4446667	Bianca	Neri	2	25880	BD	Basi Dati	9	27
4446667	Bianca	Neri	2	73026	EML	Elementi ...	12	21
4512345	Celeste	Mori	3	27054	BD2	Basi Dati 2	6	28
4512345	Celeste	Mori	3	25880	BD	Basi Dati	9	30
4455667	Viola	Bianchi	1	80299	IP	Intro ...	12	25
4455667	Viola	Bianchi	1	73026	EML	Elementi ...	12	26

ALTRE FD

- $\text{MatricolaStudente CodiceCorso} \rightarrow \text{CodiceCorso}$
- Si tratta però di una FD “**banale**” (sempre soddisfatta)
- $Y \rightarrow A$ è **non banale** se A non appartiene a Y
- $Y \rightarrow Z$ è **non banale** se nessun attributo in Z appartiene a Y

LE ANOMALIE SONO LEGATE AD ALCUNE FD

MatricolaStudente → Anno

CodiceCorso → Crediti

- causano anomalie

Le parti sinistre non corrispondono a chiavi e causano anomalie

MatricolaStudente CodiceCorso → Voto

- non causa anomalie

La parte sinistra corrisponde a una chiave e non causa anomalie

- Perché?

LE ANOMALIE SONO LEGATE ALL'ETEROGENEITÀ DELLO SCHEMA

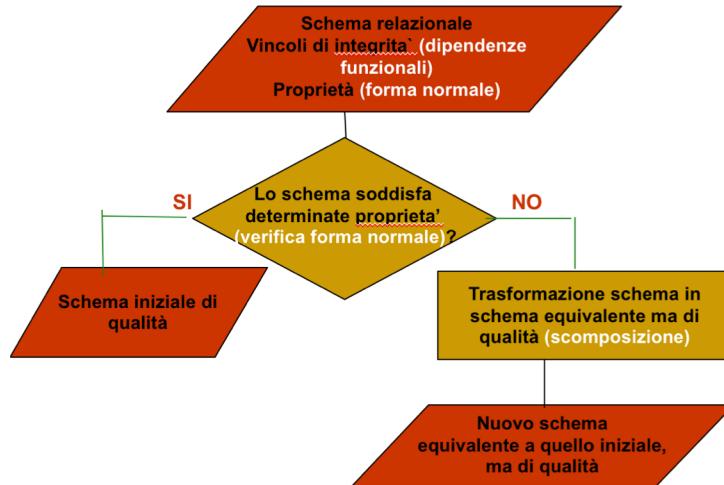
MatricolaStudente → Anno

CodiceCorso → Crediti

MatricolaStudente CodiceCorso → Voto

- MatricolaStudente CodiceCorso è chiave
- MatricolaStudente da solo no
- CodiceCorso da solo no

- Le anomalie sono causate dalla presenza di concetti eterogenei nella relazione:
 - proprietà degli studenti (l'anno di corso)
 - proprietà di corsi (i crediti)
 - proprietà della chiave (coppia **studente-corso**, il voto)



COME DEFINIRE LE BUONE PROPRIETÀ DI UNO SCHEMA PARTENDO DALLE DIPENDENZE FUNZIONALI

FORMA NORMALE DI BOYCE E CODD (BCNF)

- Una relazione **R** è in forma normale di Boyce e Codd se, per ogni dipendenza funzionale (non banale) $X \rightarrow Y$ definita su di essa, **X** contiene una chiave **K** di **R**
- La forma normale richiede che i concetti in una relazione siano omogenei (solo proprietà direttamente associate alla chiave)

MA DIPENDENZE FUNZIONALI E CHIAVI SONO COLLEGATE?

- Ovviamente sì
- Dalla definizione di chiave, una chiave è un insieme K di attributi di una relazione r di schema R(X)
 - Minimale
 - t.c. per qualunque istanza r di R, non esiste in r coppia di tuple t_1 e t_2 di r con gli stessi valori su K
 - Possiamo vederlo come:
per qualunque r,
per ogni coppia di tuple t_1 e t_2 di r con gli stessi valori su K, risulta che t_1 e t_2 hanno gli stessi valori anche su tutti gli attributi in X

POSSIAMO DETERMINARE LE CHIAVI DALLE DIPENDENZE FUNZIONALI?

- La **chiusura** di un insieme di attributi X rispetto a un insieme di dipendenze funzionali F, indicato come X^+_F , è l'insieme degli attributi che dipendono funzionalmente da X
- Un insieme di attributi K è **superchiave** (proprietà di unicità) se e solo se K^+_F contiene tutti gli attributi della relazione
 - È chiave se è anche minimale

CALCOLO DELLA CHIUSURA

Insieme di dipendenze F

Insieme di attributi X

Calcolo chiusura X^+_F

BEGIN

$X(0) := X$

 REPEAT

$S := \bigcup_{Y \rightarrow Z \in F} Y \subseteq X(i)$

$X(i+1) := X(i) \cup S$

 UNTIL $X(i+1) = X(i)$

 RETURN $X(i)$

END

CALCOLO DELLA CHIUSURA - ESEMPIO

- MatricolaStudente+
- MatricolaStudente(0) =
MatricolaStudente
- MatricolaStudente(1) =
MatricolaStudente NomeStudente
CognomeStudente AnnoIscr
- MatricolaStudente(2) =
MatricolaStudente(1)
- È il nostro risultato

MatricolaStudente → NomeStudente
MatricolaStudente → CognomeStudente
MatricolaStudente → Anno
CodiceCorso → SiglaCorso
CodiceCorso → NomeCorso
CodiceCorso → Crediti
MatricolaStudente CodiceCorso → Voto

Matricola studente non è chiave

CALCOLO DELLA CHIUSURA - ESEMPIO

- $\text{MatricolaStudente CodiceCorso} \} +$
- $\{\text{MatricolaStudente CodiceCorso}\}(0) = \{\text{MatricolaStudente CodiceCorso}$
- $\{\text{MatricolaStudente CodiceCorso}\}(1) = \{\text{MatricolaStudente CodiceCorso}$
 $\text{NomeStudente CognomeStudente}$
 $\text{AnnoIscr SiglaCorso NomeCorso Crediti}$
 Voto
- $\{\text{MatricolaStudente CodiceCorso}\}(2) = \{\text{MatricolaStudente CodiceCorso}\}(1)$
- È il nostro risultato e corrisponde allo schema della relazione

MatricolaStudente → NomeStudente
MatricolaStudente → CognomeStudente
MatricolaStudente → Anno
CodiceCorso → SiglaCorso
CodiceCorso → NomeCorso
CodiceCorso → Crediti
MatricolaStudente CodiceCorso → Voto

{MatricolaStudente CodiceCorso} è chiave

VERIFICA FORMA NORMALE DI BOYCE CODD

- Per ogni relazione R nello schema, individuo dipendenze funzionali a partire dal documento di analisi requisiti (sono vincoli)
- Determino la chiave
- Verifico condizione forma normale di Boyce Codd



SCOMPOSIZIONI E LORO PROPRIETÀ

CHE FACCIAMO SE UNA RELAZIONE NON SODDISFA LA BCNF?

- La rimpiazziamo con altre relazioni che soddisfano la BCNF

Come?

- Decomponendo sulla base delle dipendenze funzionali, al fine di separare i concetti

<u>Matricola</u> <u>Studente</u>	<u>NomeStu</u> <u>dente</u>	<u>CognomeStudente</u>	<u>Annolscr</u>	<u>CodiceCorso</u>	<u>SiglaCorso</u>	<u>NomeCorso</u>	<u>Crediti</u>	<u>Voto</u>
4433038	Mario	Rossi	2	25880	BD	Basi Dati	9	30
4532453	Andrea	Verdi	3	25880	BD	Basi Dati	9	27
4532453	Andrea	Verdi	3	27054	BD2	Basi Dati 2	6	30
4446667	Bianca	Neri	2	80299	IP	Intro ...	12	24
4446667	Bianca	Neri	2	25880	BD	Basi Dati	9	27
4446667	Bianca	Neri	2	73026	EML	Elementi ...	12	21
4512345	Celeste	Mori	3	27054	BD2	Basi Dati 2	6	28
4512345	Celeste	Mori	3	25880	BD	Basi Dati	9	30
4455667	Viola	Bianchi	1	80299	IP	Intro ...	12	25
4455667	Viola	Bianchi	1	73026	EML	Elementi ...	12	26



<u>Matricola</u> <u>Studente</u>	<u>CodiceCorso</u>	<u>Voto</u>
4433038	25880	30
4532453	25880	27
4532453	27054	30
4446667	80299	24
4446667	25880	27
4446667	73026	21
4512345	27054	28
4512345	25880	30
4455667	80299	25
4455667	73026	26

<u>Matricola</u> <u>Studente</u>	<u>Nome</u> <u>Studente</u>	<u>Cognome</u> <u>Studente</u>	<u>Annolscr</u>	<u>CodiceCorso</u>	<u>SiglaCorso</u>	<u>NomeCorso</u>	<u>Crediti</u>
4433038	Mario	Rossi	2	25880	BD	Basi Dati	9
4532453	Andrea	Verdi	3	27054	BD2	Basi Dati 2	6
4446667	Bianca	Neri	2	73026	EML	Elementi ...	12
4512345	Celeste	Mori	3	80299	IP	Intro ...	12
4455667	Viola	Bianchi	1				

COME ABBIAMO FATTO?

- Abbiamo creato una relazione per ogni dipendenza funzionale, contenente gli attributi coinvolti nella dipendenza
- Dopo aver unificato le dipendenze con la stessa parte sinistra

<u>Matricola</u> <u>Studente</u>	<u>NomeStu</u> <u>dente</u>	<u>CognomeStud</u> <u>ente</u>	<u>AnnoIscr</u>	<u>CodiceCorso</u>	<u>SiglaCorso</u>	<u>NomeCorso</u>	<u>Crediti</u>	<u>Voto</u>
4433038	Mario	Rossi	2	25880	BD	Basi Dati	9	30
4532453	Andrea	Verdi	3	25880	BD	Basi Dati	9	27
4532453	Andrea	Verdi	3	27054	BD2	Basi Dati 2	6	30
4446667	Bianca	Neri	2	80299	IP	Intro ...	12	24
4446667	Bianca	Neri	2	25880	BD	Basi Dati	9	27
4446667	Bianca	Neri	2	73026	EML	Elementi ...	12	21
4512345	Celeste	Mori	3	27054	RD2	Basi Dati 2	6	28
4512345	Celeste	Mori	3	25880	BD	Basi Dati	9	27
4455667	Viola	Bianchi	1	80299	IP	Intro ...	12	24
4455667	Viola	Bianchi	1	73026	EML	Elementi ...	12	21

MatricolaStudente → NomeStudente

MatricolaStudente → CognomeStudente

MatricolaStudente → Anno

CodiceCorso → SiglaCorso

CodiceCorso → NomeCorso

CodiceCorso → Crediti

MatricolaStudente CodiceCorso → Voto

<u>Matricola</u> <u>Studente</u>	<u>CodiceCorso</u>	<u>Voto</u>
4433038	25880	30
4532453	25880	27
4532453	27054	30
4446667	80299	24
4446667	25880	27
4446667	73026	21
4512345	27054	28
4512345	25880	30
4455667	80299	25
4455667	73026	26

MatricolaStudente → NomeStudente CognomeStudente

AnnoIscr

CodiceCorso → SiglaCorso NomeCorso Crediti

MatricolaStudente CodiceCorso → Voto

<u>Matricola</u> <u>Studente</u>	<u>Nome</u> <u>Studente</u>	<u>Cognome</u> <u>Studente</u>	<u>AnnoIscr</u>	<u>CodiceCorso</u>	<u>SiglaCorso</u>	<u>NomeCorso</u>	<u>Crediti</u>
4433038	Mario	Rossi	2	25880	BD	Basi Dati	9
4532453	Andrea	Verdi	3	27054	BD2	Basi Dati 2	6
4446667	Bianca	Neri	2	73026	EML	Elementi ...	12
4512345	Celeste	Mori	3	80299	IP	Intro ...	12
4455667	Viola	Bianchi	1				

FUNZIONA SEMPRE? (1)

- Se nessuna relazione ottenuta contiene la chiave, va aggiunta una relazione corrispondente alla chiave
- Questo serve a permetterci di ricostruire senza perdita il contenuto della relazione originale combinando (join) il contenuto delle relazioni in cui l'abbiamo scomposta

DECOMPOSIZIONE SENZA PERDITA

- Una relazione R si **decomponе senza perdita** su X_1 e X_2 se il join delle proiezioni di R su X_1 e X_2 è uguale a R stessa (cioè non contiene tuple spurie)
- Una decomposizione con schemi **disgiunti** è certamente **con perdita**
- Avere degli attributi in comune non basta a garantire che la decomposizione sia senza perdita
- La decomposizione senza perdita è garantita se gli attributi comuni alle relazioni contengono una chiave per almeno una delle relazioni composte

ESEMPIO

Studente	Corso	CorsodiStudi
Rossi	IP	Informatica
Verdi	EML	Informatica
Verdi	IP	Informatica
Neri	ALGA	SMID
Neri	SD	SMID

Studente → CorsodiStudi

Corso → CorsodiStudi

Chiave = {Studente, Corso}

DECOMPONIAMO SULLA BASE DELLE DIPENDENZE

Studente	Corso	Corsodi Studi
Rossi	IP	Informatica
Verdi	EML	Informatica
Verdi	IP	Informatica
Neri	ALGA	SMID
Neri	SD	SMID

Studente	Corsodi Studi
Rossi	Informatica
Verdi	Informatica
Neri	SMID

Corso	Corsodi Studi
IP	Informatica
EML	Informatica
ALGA	SMID
SD	SMID

Studente → CorsodiStudi
Corso → CorsodiStudi

PROVIAMO A RICOSTRUIRE

Studente	Corsodi Studi
Rossi	Informatica
Verdi	Informatica
Neri	SMID

Corso	Corsodi Studi
IP	Informatica
EML	Informatica
ALGA	SMID
SD	SMID

Studente	Corso	Corsodi Studi
Rossi	IP	Informatica
Verdi	EML	Informatica
Verdi	IP	Informatica
Neri	ALGA	SMID
Neri	SD	SMID
Rossi	EML	Informatica

Diversa dalla
relazione di
partenza!

Studente	Corso	Corsodi Studi
Rossi	IP	Informatica
Verdi	EML	Informatica
Verdi	IP	Informatica
Neri	ALGA	SMID
Neri	SD	SMID

PROVIAMO A DECOMPORRE SENZA PERDITA

Studente	Corso
Rossi	IP
Verdi	EML
Verdi	IP
Neri	ALGA
Neri	SD

Studente	Corsodi Studi
Rossi	Informatica
Verdi	Informatica
Neri	SMID

Corso	Corsodi Studi
IP	Informatica
EML	Informatica
ALGA	SMID
SD	SMID

Studente → CorsodiStudi
Corso → CorsodiStudi

UN ALTRO PROBLEMA

- Supponiamo di voler inserire una nuova tupla che specifica la partecipazione dello studente di Informatica Rossi al corso di ALGA

Studente	Corso
Rossi	ALGA
Rossi	IP
Verdi	EML
Verdi	IP
Neri	ALGA
Neri	SD

Studente	CorsodiStudi
Rossi	Informatica
Verdi	Informatica
Neri	SMID

Studente → CorsodiStudi
Corso → CorsodiStudi

Corso → CorsodiStudi
Non vale più!

Studente	Corso	CorsodiStudi
Rossi	IP	Informatica
Verdi	EML	Informatica
Verdi	IP	Informatica
Neri	ALGA	SMID
Neri	SD	SMID
Rossi	ALGA	Informatica

CONSERVAZIONE DELLE DIPENDENZE

- Una decomposizione **conserva le dipendenze** se ciascuna delle dipendenze funzionali dello schema originario coinvolge attributi che compaiono tutti insieme in uno degli schemi decomposti
- Corso → CorsodiStudi non è conservata

QUALITÀ DELLE DECOMPOSIZIONI

- Una decomposizione dovrebbe sempre soddisfare:
 - la **decomposizione senza perdita (lossless join)**, che garantisce la ricostruzione delle informazioni originarie
 - la **conservazione delle dipendenze**, che garantisce il mantenimento dei vincoli di integrità originari
- Esiste sempre una decomposizione in BCNF che soddisfa queste proprietà?

UNA RELAZIONE NON-NORMALIZZATA

City	<u>Street</u>	Zip
Genova	Vernazza	16131
Genova	Sturla	16131
Genova	Garibaldi	16121
Milano	Garibaldi	20121
Milano	Assietta	20161

City Street → Zip
Zip → City

CS → Z
Z → C

Chiavi CS, ZS

non BCNF

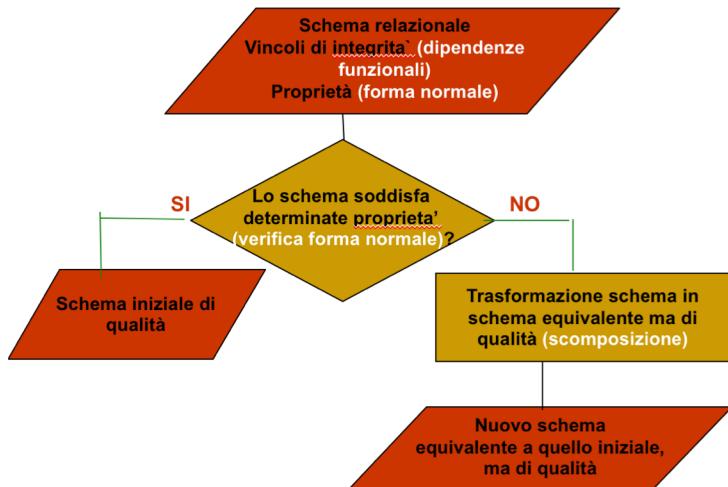
QUALI PROBLEMI?

- Provate per esercizio a

- Individuare le possibili scomposizioni
- Individuare le proprietà soddisfatte da ciascuna scomposizione

LA DECOMPOSIZIONE È PROBLEMATICA

- **City Street → Zip** coinvolge tutti gli attributi e quindi nessuna decomposizione può preservare tale dipendenza
- quindi in alcuni casi la BCNF **non è raggiungibile**
- per alcuni schemi che hanno almeno due chiavi non esiste una decomposizione in BCNF che sia
 - senza perdita (lossless join)
 - preservi le dipendenze



ALTRE FORME NORMALI, ALTRE PROPRIETÀ

UNA NUOVA FORMA NORMALE

- Una relazione R è in **terza forma normale** se, per ogni FD (non banale) $X \rightarrow Y$ definita su R , è verificata almeno una delle seguenti condizioni:
 - X contiene una chiave K di R (**soddisfa BCNF**)
 - ogni attributo in Y è contenuto in almeno una chiave di R (**Y contiene solo attributi primi**)

ESEMPIO

- CSZ
- Dipendenze $CS \rightarrow Z$, $Z \rightarrow C$
- Chiavi CS , ZS
- Attributi primi C , S , Z

- $CS \rightarrow Z$ soddisfa BCNF
- $Z \rightarrow C$ non soddisfa BCNF ma C è primo

- CSZ con $CS \rightarrow Z$, $Z \rightarrow C$ è in 3NF

BCNF E TERZA FORMA NORMALE

- la terza forma normale è meno restrittiva della forma normale di Boyce e Codd (e ammette relazioni con alcune anomalie)
 - Aumenta i casi di ridondanza
- ha il vantaggio però di essere sempre raggiungibile
- è sempre possibile trovare una decomposizione in 3NF che sia lossless join e preservi le dipendenze

COME SCOMPORRE?

- L'approccio alla scomposizione visto prima
 - si crea una relazione per ogni gruppo di attributi coinvolti in una dipendenza funzionale
 - si verifica che alla fine una relazione contenga una chiave della relazione originaria
- produce una decomposizione **in terza forma normale**

FUNZIONA SEMPRE?

- Funziona sempre?
- Dipende dalle dipendenze individuate
- Se le dipendenze sono in forma minimale (= non ottenibili da altre dipendenze), sì!
 - Non ci sono dipendenze inutili
 - $A \rightarrow B, B \rightarrow C$
 - $A \rightarrow C$ è inutile
 - Le dipendenze non contengono attributi inutili a sinistra
 - $A \rightarrow B, AB \rightarrow C$
 - B in $AB \rightarrow C$ è inutile perché se in due tuple A coincide, allora coincide anche B
 - Le dipendenze non contengono attributi inutili a destra
 - $A \rightarrow B, A \rightarrow C, A \rightarrow CD$
 - In $A \rightarrow CD$, C è ridondante

UNA POSSIBILE STRATEGIA

- se la relazione non è normalizzata si decomponе in terza forma normale
 - alla fine si verifica se lo schema ottenuto è anche in BCNF
-
- Se una relazione ha una sola chiave allora le due forme normali coincidono
 - Se esistono almeno due chiavi (come in CSZ), potrebbe non esistere un decomposizione BCNF che sia lossless join e preservi le dipendenze

PROGETTAZIONE E NORMALIZZAZIONE

- la teoria della normalizzazione può essere usata nella progettazione logica per verificare lo schema relazionale finale
- il fatto di utilizzare la metodologia di progettazione vista aiuta a produrre schemi normalizzati (spesso in 3NF), ma non lo garantisce
- in alcuni casi può essere accettabile utilizzare schemi non normalizzati, per motivi di efficienza, ma è importante che la scelta sia consapevole e documentata